

**APLIKASI INTERAKTIF PENGENALAN BATIK
UNTUK PENGUNJUNG PAMERAN BERBASIS
ANDROID DAN *IMAGE PROCESSING***



Disusun oleh

Dhiya Fauziza 3.34.17.1.09

Zani Kolina 3.34.17.1.26

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI SEMARANG
2020**

**APLIKASI INTERAKTIF PENGENALAN BATIK
UNTUK PENGUNJUNG PAMERAN BERBASIS
ANDROID DAN *IMAGE PROCESSING***



**Tugas akhir ini disusun untuk melengkapi sebagian persyaratan
menjadi Ahli Madya**

Disusun oleh

Dhiya Fauziza 3.34.17.1.09

Zani Kolina 3.34.17.1.26

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI SEMARANG**

2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini, saya:

Nama / NIM / Kelas : Dhiya Fauziza / 3.34.17.1.09 / IK-3B

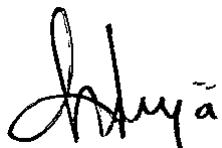
Nama / NIM / Kelas : Zani Kolina / 3.34.17.1.26 / IK-3B

Jurusan : Teknik Elektro

Program Studi : Teknik Informatika

Menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul APLIKASI INTERAKTIF PENGENALAN BATIK UNTUK PENGUNJUNG PAMERAN BERBASIS ANDROID DAN *IMAGE PROCESSING* yang dibuat untuk melengkapi sebagian persyaratan menjadi Ahli Madya pada Program Studi Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Semarang, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari tugas akhir yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar Ahli Madya di lingkungan Politeknik Negeri Semarang maupun di perguruan tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Semarang, 13 Oktober 2020



Dhiya Fauziza
NIM 3.34.17.1.09



Zani Kolina
NIM 3.34.17.1.26

HALAMAN PERSETUJUAN

Tugas akhir dengan judul APLIKASI INTERAKTIF PENGENALAN BATIK UNTUK PENGUNJUNG PAMERAN BERBASIS ANDROID DAN *IMAGE PROCESSING* dibuat untuk melengkapi sebagian persyaratan menjadi Ahli Madya pada Program Studi Teknik Informatika, Jurusan Teknik Elektro Politeknik Negeri Semarang dan disetujui untuk diajukan dalam sidang ujian tugas akhir.

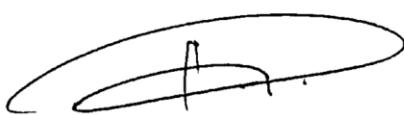
Semarang, 13 Oktober 2020

Pembimbing I,



Mardiyono, S.Kom., M.Sc.
NIP. 197403112000121001

Pembimbing II,



Lilik Triyono, S.T., M.Kom.
NIP. 198404202015041003

Mengetahui,

Ketua Program Studi Teknik Informatika



Sukamto, S.Kom., M.T.
NIP. 197101172003121001

HALAMAN PENGESAHAN

Tugas akhir dengan judul APLIKASI INTERAKTIF PENGENALAN BATIK UNTUK PENGUNJUNG PAMERAN BERBASIS ANDROID DAN *IMAGE PROCESSING* telah dipertahankan dalam ujian wawancara dan diterima sebagai syarat untuk menjadi Ahli Madya pada Program Studi Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Semarang pada tanggal 13 Oktober 2020.

TIM PENGUJI

Penguji I,

Wahyu Sulistiyo, S.T., M.Kom.
NIP. 197704012005011001

Penguji II,

Idhawati H, S.Kom., M.Kom.
NIP. 197711192008012013

Penguji III,

Tri Raharjo Yudantoro, S.Kom., M.Kom.
NIP. 196810252000121001

Ketua Penguji,

Mardiyono, S.Kom., M.Sc.
NIP. 197403112000121001

Sekretaris,

Afandi Nur Aziz Thohari, S.T., M.Cs.
NIP. 199004112019031014



Afandi Nur Aziz Thohari, S.T., M.Cs.
NIP. 199004112019031014

ABSTRAK

Dhiya Fauziza dan Zani Kolina, “Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan Image Processing”, Tugas akhir D III Program Studi Teknik Informatika, Jurusan Teknik Elektro, Politeknik Negeri Semarang, di bawah bimbingan Mardiyono, S.Kom., M.Sc. dan Liliek Triyono, S.T, M.Kom., Oktober 2020, 193.

Salah satu permasalahan yang dialami oleh masyarakat mengenai batik yaitu minimnya pengetahuan mengenai citra batik yang ada di Indonesia. Keterbatasan tersebut dikarenakan kurangnya media yang membantu pengguna dalam mendapatkan informasi batik yang ingin diketahuinya, khususnya ketika diadakannya event pameran batik. Dimana pada saat event pameran batik, umumnya pengunjung masih melakukan interaksi secara langsung kepada pengrajin untuk memperoleh informasi batik tersebut. Tugas akhir ini bertujuan untuk membangun aplikasi untuk mengenali motif batik secara cepat dan akurat menggunakan image processing dengan metode Convolutional Neural Network (CNN). Proses pengenalan motif dilakukan dengan mendekripsi fitur yang ada pada gambar dan membandingkan fitur gambar tersebut dengan yang ada di dalam dataset, gambar akan terus didekripsi selama dalam pandang kamera. Aplikasi ini menggunakan dataset yang terdiri dari 12 motif batik dan setiap motif terdapat 600 gambar, dengan total keseluruhan motif yaitu 7200 gambar. Data tersebut dibagi dalam data training sebesar 6480 gambar (90%) dan data testing sejumlah 720 gambar (10%). Akurasi yang dihasilkan dari proses testing sebesar 99,86%. Metode yang digunakan dalam pembuatan sistem ini adalah metode scrum yang meliputi product backlog, sprint planning meeting, sprint backlog, dan pengerjaan sprint.

Kata kunci: Batik, Convolutional Neural Network (CNN), Image Processing, Metode Scrum, Pengenalan Motif Batik

ABSTRACT

Dhiya Fauziza and Zani Kolina, “An Introduction to Batik Interactive Application for Exhibition Visitors Based on Android and Image Processing”, Final Project Diploma III Department of Electrical Engineering State Polytechnic of Semarang, under guidance of Mr. Mardiyono, S.Kom., M.Sc. and Mr. Liliek Triyono, S.T, M.Kom., October 2020, 193.

One of the problems experienced by the community regarding batik is the lack of knowledge about the image of batik in Indonesia. This limitation is due to the lack of media that helps users get the batik information they want to know, especially when the batik exhibition is held. Where at the batik exhibition event, visitors generally still interact directly with the craftsmen to obtain the batik information. This final project aims to build an application to quickly and accurately recognize batik motifs using image processing with the Convolutional Neural Network (CNN) method. Motive recognition process is done by detecting the features in the image and comparing the image features with those in the dataset, the image will continue to be detected as long as it is in the camera's field of view. This application uses a dataset consisting of 12 batik motifs and each motif has 600 images, with a total of 7200 images. The data is divided into training data of 6480 images (90%) and testing data of 720 images (10%). The resulting accuracy from the testing process is 99.86%. The method used in making this system is the Scrum method which includes the product backlog, sprint planning meeting, sprint backlog, and sprint work.

Keywords: Batik, Convolutional Neural Network (CNN), Image Processing, Scrum Method, Introduction to Batik Motifs

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas akhir ini dengan judul **APLIKASI INTERAKTIF PENGENALAN BATIK UNTUK PENGUNJUNG PAMERAN BERBASIS ANDROID DAN IMAGE PROCESSING** disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Program Studi Diploma III Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Semarang.

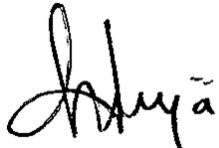
Pada kesempatan ini penulis menyampaikan terimakasih kepada:

1. Bapak Ir. Supriyadi, M.T, selaku Direktur Politeknik Negeri Semarang.
2. Bapak Yusnan Badruzzaman, S.T., M.Eng selaku Ketua Jurusan Teknik Elektro.
3. Bapak Sukamto, S.Kom., M.T., selaku Ketua Program Studi D3 Teknik Informatika.
4. Bapak Mardiyono, S.Kom., M.Sc. dan Bapak Liliek Triyono, S.T, M.Kom. selaku Dosen Pembimbing yang memberikan masukan dan solusi kepada penulis mengenai pembuatan aplikasi dan penyusunan laporan.
5. Kedua orang tua beserta saudara-saudara yang senantiasa mendoakan penulis dan memberikan bantuan secara moril maupun materil dengan penuh keikhlasan.
6. Teman-teman Kelas IK-3B yang telah membantu dan mendukung Penulis.

Semoga tugas akhir ini dapat bermanfaat bagi Penulis khususnya dan pihak yang berkepentingan.

Semarang, 13 Oktober 2020

Mahasiswa,



Dhiya Fauziza

NIM 3.34.17.1.09

Mahasiswa,



Zani Kolina

NIM 3.34.17.1.26

DAFTAR ISI

halaman

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
ABSTRAK	v
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiv
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.1.1. Perumusan Masalah	2
1.1.2. Tujuan	2
1.1.3. Manfaat	2
1.2. Batasan Masalah.....	3
1.3. Sistematika	3
BAB II TINJAUAN PUSTAKA.....	5
2.1. Penelitian Terkait	5
2.2. Teknologi yang digunakan.....	15
2.2.1. <i>Deep Learning</i>	15
2.2.2. CNN (<i>Convolutional Neural Network</i>).....	16
2.2.3. <i>Transfer Learning</i>	21
2.2.4. MobileNet V2	21
2.2.5. Tensorflow	22
2.2.6. Tf.lite	22
2.2.7. Python	22
2.2.8. Android Studio.....	22
2.2.9. MySQL	23

BAB III KEGIATAN PELAKSANAAN	24
3.1. <i>Product Backlog</i>	25
3.2. <i>Sprint Planning Meeting</i>	25
3.3. <i>Sprint Backlog</i>	27
3.3.1. <i>Sprint 1</i> Analisis Kebutuhan Perangkat	27
3.3.2. <i>Sprint 2</i> Analisis Perancangan Sistem	27
3.3.3. <i>Sprint 3</i> Perancangan Model Dataset.....	28
3.3.4. <i>Sprint 4</i> Pembuatan dan Pengujian Model Dataset.....	29
3.3.5. <i>Sprint 5</i> Perancangan Antarmuka Aplikasi.....	29
3.3.6. <i>Sprint 6</i> Pembuatan dan Pengujian Fitur Aplikasi.....	32
3.3.7. <i>Sprint 7</i> Perancangan Uji Kepuasan Pengguna	35
3.4. Pengerajan <i>Task</i>	36
3.4.1. <i>Sprint 1:</i> Analisis Kebutuhan Perangkat.....	36
3.4.2. <i>Sprint 2:</i> Analisis Perancangan Sistem	36
3.4.3. <i>Sprint 3:</i> Perancangan Model Dataset.....	57
3.4.4. <i>Sprint 4:</i> Pembuatan dan Pengujian Model Dataset	60
3.4.5. <i>Sprint 5:</i> Perancangan Antarmuka Aplikasi	67
3.4.6. <i>Sprint 6:</i> Pembuatan Dan Pengujian Fitur Aplikasi.....	94
3.4.7. <i>Sprint 7:</i> Perancangan Uji Kepuasan Pengguna	152
BAB IV ANALISIS DAN PEMBAHASAN	154
4.1. Analisis dan Pembahasan Sistem	154
4.1.1. Analisis Model.....	154
4.1.2. Analisis Aplikasi.....	159
4.2. Pembahasan Sistem	160
4.2.1. Rekapitulasi Pengujian Fungsionalitas Model.....	160
4.2.1. Rekapitulasi Pengujian Deteksi Batik	162
4.2.2. Rekapitulasi Pengujian Tampilan Android.....	175
4.2.2. Rekapitulasi Pengujian Kompatibilitas Aplikasi	180
4.2.3. Rekapitulasi Uji Fungsionalitas Aplikasi	182
4.2.4. Pengujian Tingkat Kepuasan Pengguna	187
BAB V KESIMPULAN	190
DAFTAR PUSTAKA	191

DAFTAR GAMBAR

	halaman
Gambar 2. 1 Arsitektur <i>Convolutional Neural Network</i>	16
Gambar 2. 2 Proses Konvolusi.....	17
Gambar 2. 3 Proses <i>Pooling Layer</i>	20
Gambar 2. 4 Proses <i>Fully Connected Layer</i>	20
Gambar 2. 5 Arsitektur MobileNet-V2	21
Gambar 3. 1 Tahapan Metode Agile menggunakan Kerangka Kerja Scrum.....	24
Gambar 3. 2 Gambaran Umum Sistem	37
Gambar 3. 3 <i>Block Diagram</i> Sistem.....	38
Gambar 3. 4 <i>Block Diagram</i> Pembuatan Model.....	38
Gambar 3. 5 <i>Use Case Diagram</i> Sistem	39
Gambar 3. 6 <i>Activity Diagram Login</i>	41
Gambar 3. 7 <i>Activity Diagram</i> Menampilkan Dashboard.....	42
Gambar 3. 8 <i>Activity Diagram</i> Motif Batik	43
Gambar 3. 9 <i>Activity Diagram</i> Manajemen Berita	44
Gambar 3. 10 <i>Activity Diagram</i> Manajemen Pengunjung	45
Gambar 3. 11 <i>Activity Diagram</i> Manajemen Penyelenggara.....	46
Gambar 3. 12 <i>Activity Diagram</i> Manajemen <i>Event</i> Pameran	47
Gambar 3. 13 <i>Activity Diagram</i> Mengedit Profil.....	48
Gambar 3. 14 <i>Activity Diagram</i> Klasifikasi Batik	49
Gambar 3. 15 <i>Activity Diagram</i> Menampilkan <i>Event</i>	50
Gambar 3. 16 <i>Activity Diagram</i> Mengikuti <i>Event</i>	51
Gambar 3. 17 <i>Activity Diagram</i> Memberikan <i>Like</i> dan <i>Comment</i>	52
Gambar 3. 18 <i>Activity Diagram</i> Menampilkan Berita	53
Gambar 3. 19 <i>Activity Diagram</i> Mengedit Profil.....	54
Gambar 3. 20 <i>Activity Diagram</i> Manajemen <i>Event</i> Penyelenggara.....	55
Gambar 3. 21 ERD Sistem Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan <i>Image Processing</i>	56
Gambar 3. 22 Diagram Blok Perancangan dan Pembuatan Model Dataset.....	57
Gambar 3. 23 Data Batik.....	58

Gambar 3. 24 Tahapan Pre-processing Citra Batik	58
Gambar 3. 25 Data Batik yang Dirotasi dan Diflipping	59
Gambar 3. 26 Hasil Import Package Model Maker Tensorflow	60
Gambar 3. 27 Hasil Pembagian Data Training dan Data Testing.....	61
Gambar 3. 28 Hasil Menampilkan Sampel dari Dataset	62
Gambar 3. 29 Hasil Training Model	63
Gambar 3. 30 Hasil Testing Model	64
Gambar 3. 31 Hasil Export Model	66
Gambar 3. 32 Tampilan Halaman Onboarding.....	67
Gambar 3. 33 Tampilan Halaman Splash Screen.....	68
Gambar 3. 34 Tampilan Halaman Login Admin	68
Gambar 3. 35 Tampilan Halaman Register.....	69
Gambar 3. 36 Tampilan Halaman Pengisian Data Pengguna	70
Gambar 3. 37 Tampilan Halaman Home.....	70
Gambar 3. 38 Tampilan Halaman Event Pameran	71
Gambar 3. 39 Tampilan Halaman Komentar Event	72
Gambar 3. 40 Tampilan Halaman Mengubah Komentar Event	72
Gambar 3. 41 Tampilan Halaman Menghapus Komentar Event	73
Gambar 3. 42 Tampilan Halaman Scan Qrcode Event	74
Gambar 3. 43 Tampilan Halaman Konfirmasi Event.....	75
Gambar 3. 44 Tampilan Halaman Deteksi Batik	75
Gambar 3. 45 Tampilan Halaman Profil	76
Gambar 3. 46 Tampilan Halaman Login Admin	77
Gambar 3. 47 Tampilan Halaman Dashboard	77
Gambar 3. 48 Tampilan Halaman Menampilkan Informasi Motif Batik.....	78
Gambar 3. 49 Tampilan Halaman Menambah Informasi Motif Batik.....	79
Gambar 3. 50 Tampilan Halaman Mengedit Informasi Motif Batik	79
Gambar 3. 51 Tampilan Halaman Menampilkan Berita	80
Gambar 3. 52 Tampilan Halaman Menambah Berita	81
Gambar 3. 53 Tampilan Halaman Mengedit Berita	81
Gambar 3. 54 Tampilan Halaman Menampilkan All Event Pameran	82
Gambar 3. 55 Tampilan Halaman Menampilkan Detail Event	83

Gambar 3. 56 Tampilan Halaman Konfirmasi <i>Event</i>	83
Gambar 3. 57 Tampilan Halaman Data Pengunjung	84
Gambar 3. 58 Tampilan Halaman Menambah Data Pengunjung	85
Gambar 3. 59 Tampilan Halaman Mengedit Data Pengunjung	86
Gambar 3. 60 Tampilan Halaman Data Penyelenggara	87
Gambar 3. 61 Tampilan Halaman Menambah Data Penyelenggara	87
Gambar 3. 62 Tampilan Halaman Mengedit Data Penyelenggara.....	88
Gambar 3. 63 Tampilan Halaman Profil	89
Gambar 3. 64 Tampilan Halaman Menambahkan <i>Event</i>	89
Gambar 3. 65 Tampilan Halaman Mengedit <i>Event</i>	90
Gambar 3. 66 Tampilan Halaman Menampilkan <i>MyEvent</i>	91
Gambar 3. 67 Tampilan Halaman Menampilkan <i>QRCode</i>	92
Gambar 3. 68 Tampilan Halaman Menampilkan Buku Tamu	92
Gambar 3. 69 Tampilan Halaman Menambahkan Kehadiran Pengunjung.....	93
Gambar 3. 70 Tampilan Halaman Menampilkan Status <i>Event</i>	94
Gambar 4. 1 Hasil Perbandingan Ukuran Model	159

DAFTAR TABEL

halaman

Tabel 2. 1 Perbandingan Sistem Aplikasi	11
Tabel 3. 1 Daftar <i>Sprint</i>	26
Tabel 3. 2 <i>Sprint</i> 1: Analisis Kebutuhan Perangkat	27
Tabel 3. 3 <i>Sprint</i> 2: Analisis Perancangan Sistem	28
Tabel 3. 4 <i>Sprint</i> 3: Perancangan Model Dataset.....	28
Tabel 3. 5 <i>Sprint</i> 4: Pembuatan dan Pengujian Model Dataset.....	29
Tabel 3. 6 <i>Sprint</i> 5: Perancangan Antarmuka Aplikasi.....	29
Tabel 3. 7 <i>Sprint</i> 6: Pembuatan dan Pengujian Fitur Aplikasi.....	32
Tabel 3. 8 <i>Sprint</i> 7 : Perancangan Uji Kepuasan Pengguna.....	35
Tabel 3. 9 Pengujian Fungsionalitas <i>Import Package</i> dan <i>Library Model</i>	61
Tabel 3. 10 Pengujian Fungsionalitas Pembagian Data	62
Tabel 3. 11 Pengujian Fungsionalitas <i>Train Model</i>	64
Tabel 3. 12 Pengujian Fungsionalitas <i>Testing Model</i>	65
Tabel 3. 13 Pengujian Fungsionalitas <i>Export Model</i>	66
Tabel 3. 14 Pengujian Kompatibilitas Halaman <i>Onboarding</i>	95
Tabel 3. 15 Pengujian Fungsionalitas Halaman <i>Onboarding</i>	96
Tabel 3. 16 Pengujian Kompatibilitas Halaman <i>Splash Screen</i>	97
Tabel 3. 17 Pengujian Fungsionalitas Halaman <i>Splash Screen</i>	97
Tabel 3. 18 Pengujian Kompatibilitas Halaman <i>Login</i>	98
Tabel 3. 19 Pengujian Fungsionalitas Halaman <i>Login</i>	99
Tabel 3. 20 Pengujian Kompatibilitas Halaman <i>Register</i>	100
Tabel 3. 21 Pengujian Fungsionalitas Halaman <i>Register</i>	100
Tabel 3. 22 Pengujian Kompatibilitas Halaman Pengisian Data Pengguna.....	101
Tabel 3. 23 Pengujian Fungsionalitas Halaman Pengisian Data Pengguna	102
Tabel 3. 24 Pengujian Kompatibilitas Halaman <i>Home</i>	103
Tabel 3. 25 Pengujian Fungsionalitas Halaman <i>Home</i>	103
Tabel 3. 26 Pengujian Kompatibilitas Halaman <i>Event Pameran</i>	105
Tabel 3. 27 Pengujian Fungsionalitas Halaman <i>Event Pameran</i>	105
Tabel 3. 28 Pengujian Kompatibilitas Halaman <i>Komentar Event</i>	107

Tabel 3. 29 Pengujian Fungsionalitas Halaman Komentar <i>Event</i>	107
Tabel 3. 30 Pengujian Kompatibilitas Halaman Mengedit Komentar <i>Event</i>	108
Tabel 3. 31 Pengujian Fungsionalitas Halaman Mengedit Komentar <i>Event</i>	109
Tabel 3. 32 Pengujian Kompatibilitas Halaman Menghapus Komentar <i>Event</i> ...	110
Tabel 3. 33 Pengujian Fungsionalitas Halaman Menghapus Komentar <i>Event</i> ...	110
Tabel 3. 34 Pengujian Kompatibilitas Halaman <i>Scan Qrcode Event</i>	111
Tabel 3. 35 Pengujian Fungsionalitas Halaman <i>Scan Qrcode Event</i>	112
Tabel 3. 36 Pengujian Kompatibilitas Halaman Konfirmasi <i>Event</i>	113
Tabel 3. 37 Pengujian Fungsionalitas Halaman Konfirmasi <i>Event</i>	113
Tabel 3. 38 Pengujian Kompatibilitas Halaman Deteksi Batik.....	114
Tabel 3. 39 Pengujian Fungsionalitas Halaman Deteksi Batik	115
Tabel 3. 40 Pengujian Kompatibilitas Halaman Profil	116
Tabel 3. 41 Pengujian Fungsionalitas Halaman Profil.....	116
Tabel 3. 42 Pengujian Kompatibilitas Halaman <i>Login</i>	118
Tabel 3. 43 Pengujian Fungsionalitas Halaman <i>Login</i>	118
Tabel 3. 44 Pengujian Kompatibilitas Halaman <i>Dashboard Admin</i>	119
Tabel 3. 45 Pengujian Fungsionalitas Halaman <i>Dashboard Admin</i>	119
Tabel 3. 46 Pengujian Kompatibilitas Menampilkan Informasi Motif Batik	121
Tabel 3. 47 Pengujian Fungsionalitas Menampilkan Informasi Motif Batik.....	121
Tabel 3. 48 Pengujian Kompatibilitas Menambahkan Informasi Motif Batik....	122
Tabel 3. 49 Pengujian Fungsionalitas Menambahkan Informasi Motif Batik	122
Tabel 3. 50 Pengujian Kompatibilitas Mengedit Informasi Motif Batik	123
Tabel 3. 51 Pengujian Fungsionalitas Mengedit Informasi Motif Batik.....	124
Tabel 3. 52 Pengujian Kompatibilitas Halaman Menampilkan Berita	125
Tabel 3. 53 Pengujian Fungsionalitas Halaman Menampilkan Berita.....	125
Tabel 3. 54 Pengujian Kompatibilitas Halaman Menambahkan Berita.....	126
Tabel 3. 55 Pengujian Fungsionalitas Halaman Menambahkan Berita	127
Tabel 3. 56 Pengujian Kompatibilitas Halaman Mengedit Berita	128
Tabel 3. 57 Pengujian Fungsionalitas Halaman Mengedit Berita.....	128
Tabel 3. 58 Pengujian Kompatibilitas Menampilkan <i>All Event</i> Pameran.....	129
Tabel 3. 59 Pengujian Fungsionalitas Menampilkan <i>All Event</i> Pameran	129
Tabel 3. 60 Pengujian Kompatibilitas Halaman Menampilkan Detail <i>Event</i>	130

Tabel 3. 61 Pengujian Fungsionalitas Halaman Menampilkan Detail <i>Event</i>	131
Tabel 3. 62 Pengujian Kompatibilitas Menampilkan Konfirmasi <i>Event</i>	132
Tabel 3. 63 Pengujian Fungsionalitas Menampilkan Konfirmasi <i>Event</i>	132
Tabel 3. 64 Pengujian Kompatibilitas Menampilkan Data Pengunjung	133
Tabel 3. 65 Pengujian Fungsionalitas Menampilkan Data Pengunjung	134
Tabel 3. 66 Pengujian Kompatibilitas Menambahkan Data Pengunjung	135
Tabel 3. 67 Pengujian Fungsionalitas Menambahkan Data Pengunjung	135
Tabel 3. 68 Pengujian Kompatibilitas Halaman Mengedit Data Pengunjung	136
Tabel 3. 69 Pengujian Fungsionalitas Halaman Mengedit Data Pengunjung	136
Tabel 3. 70 Pengujian Kompatibilitas Menampilkan Data Penyelenggara.....	137
Tabel 3. 71 Pengujian Fungsionalitas Menampilkan Data Penyelenggara	138
Tabel 3. 72 Pengujian Kompatibilitas Menambahkan Data Penyelenggara	139
Tabel 3. 73 Pengujian Fungsionalitas Menambahkan Data Penyelenggara	139
Tabel 3. 74 Pengujian Kompatibilitas Halaman Mengedit Data Penyelenggara	140
Tabel 3. 75 Pengujian Fungsionalitas Halaman Mengedit Data Penyelenggara	140
Tabel 3. 76 Pengujian Kompatibilitas Halaman Profil	141
Tabel 3. 77 Pengujian Fungsionalitas Halaman Profil.....	142
Tabel 3. 78 Pengujian Kompatibilitas Halaman Menambahkan <i>Event</i>	143
Tabel 3. 79 Pengujian Fungsionalitas Halaman Menambahkan <i>Event</i>	143
Tabel 3. 80 Pengujian Kompatibilitas Halaman Mengedit <i>Event</i>	144
Tabel 3. 81 Pengujian Fungsionalitas Halaman Mengedit <i>Event</i>	144
Tabel 3. 82 Pengujian Kompatibilitas Halaman Menampilkan <i>My Event</i>	145
Tabel 3. 83 Pengujian Fungsionalitas Halaman Menampilkan <i>My Event</i>	146
Tabel 3. 84 Pengujian Kompatibilitas Halaman Menampilkan <i>Qrcode</i>	147
Tabel 3. 85 Pengujian Fungsionalitas Halaman Menampilkan <i>Qrcode</i>	147
Tabel 3. 86 Pengujian Kompatibilitas Halaman Menampilkan Buku Tamu	148
Tabel 3. 87 Pengujian Fungsionalitas Halaman Menampilkan Buku Tamu.....	148
Tabel 3. 88 Pengujian Kompatibilitas Menambahkan Kehadiran Pengunjung ..	149
Tabel 3. 89 Pengujian Fungsionalitas Menambahkan Kehadiran Pengunjung...	150
Tabel 3. 90 Pengujian Kompatibilitas Halaman Melihat Konfirmasi <i>Event</i>	151
Tabel 3. 91 Pengujian Fungsionalitas Halaman Melihat Konfirmasi <i>Event</i>	151
Tabel 3. 92 Daftar Pertanyaan.....	152

Tabel 3. 93 Indikator Penilaian	153
Tabel 3. 94 Indikator Kepuasan Pengguna	153
Tabel 4. 1 Hasil Grafik <i>Training Model</i>	156
Tabel 4. 2 Hasil Akurasi <i>Training Model</i>	158
Tabel 4. 3 Hasil Akurasi <i>Testing Model</i>	158
Tabel 4. 4 Hasil Pengujian Fungsionalitas Model	160
Tabel 4. 5 Rekapitulasi Pengujian Deteksi Batik.....	162
Tabel 4. 6 Spesifikasi <i>Smartphone</i> Penguji Aplikasi Android.....	176
Tabel 4. 7 Hasil Rekapitulasi Tampilan Android.....	176
Tabel 4. 8 Rekapitulasi Pengujian Kompabilitas Aplikasi.....	180
Tabel 4. 9 Hasil Pengujian Fungsionalitas Aplikasi	182
Tabel 4. 10 Hasil Kuisioner Responden.....	188

DAFTAR LAMPIRAN

halaman

Lampiran 1 Dokumentasi Pengujian	195
Lampiran 2 Lembar Kontrol Bimbingan Tugas Akhir	196
Lampiran 3 Form Revisi Tugas Akhir	202
Lampiran 4 Surat Keterangan Selesai Revisi Tugas Akhir.....	205
Lampiran 5 Kode Pengujian <i>Import Package</i> dan <i>Library</i>	206
Lampiran 6 Kode Pengujian Pembagian Data	206
Lampiran 7 Kode Pengujian <i>Train Model</i>	207
Lampiran 8 Kode Pengujian <i>Testing Model</i>	207
Lampiran 9 Kode Pengujian <i>Export Model</i>	207
Lampiran 10 Kode Halaman <i>Onboarding</i>	207
Lampiran 11 Kode Halaman <i>Splash Screen</i>	208
Lampiran 12 Kode Halaman <i>Login</i>	208
Lampiran 13 Kode Halaman <i>Register</i>	209
Lampiran 14 Kode Halaman Data Pengguna.....	210
Lampiran 15 Kode Halaman <i>Home</i>	211
Lampiran 16 Kode Halaman <i>Event</i> Pameran	212
Lampiran 17 Kode Halaman Komentar <i>Event</i>	213
Lampiran 18 Kode Halaman Mengedit Komentar <i>Event</i>	214
Lampiran 19 Kode Halaman Menghapus Komentar <i>Event</i>	215
Lampiran 20 Kode Halaman <i>Scan Qrcode Event</i>	216
Lampiran 21 Kode Halaman Konfirmasi <i>Event</i>	216
Lampiran 22 Kode Halaman Deteksi Batik	217
Lampiran 23 Kode Halaman Profil	219
Lampiran 24 Kode Halaman <i>Login</i>	220
Lampiran 25 Kode Halaman <i>Dashboard</i>	221
Lampiran 26 Kode Halaman Menampilkan Informasi Motif Batik.....	221
Lampiran 27 Kode Halaman Menambahkan Informasi Motif Batik	221
Lampiran 28 Kode Halaman Mengedit Informasi Motif Batik	222
Lampiran 29 Kode Halaman Menampilkan Berita	222

Lampiran 30 Kode Halaman Menambahkan Berita.....	222
Lampiran 31 Kode Halaman Mengedit Berita	223
Lampiran 32 Kode Halaman Menampilkan <i>All Event</i> Pameran	224
Lampiran 33 Kode Halaman Menampilkan Detail <i>Event</i>	224
Lampiran 34 Kode Halaman Menampilkan Konfirmasi <i>Event</i>	224
Lampiran 35 Kode Halaman Menampilkan Data Pengunjung	224
Lampiran 36 Kode Halaman Menambahkan Data Pengunjung.....	225
Lampiran 37 Kode Halaman Mengedit Data Pengunjung	225
Lampiran 38 Kode Halaman Menampilkan Data Penyelenggara.....	226
Lampiran 39 Kode Halaman Menambahkan Data Penyelenggara	226
Lampiran 40 Kode Halaman Mengedit Data Penyelenggara.....	227
Lampiran 41 Kode Halaman Profil	228
Lampiran 42 Kode Halaman Menambahkan <i>Event</i>	228
Lampiran 43 Kode Halaman Mengedit <i>Event</i>	229
Lampiran 44 Kode Halaman Menampilkan <i>My Event</i>	230
Lampiran 45 Kode Halaman Menampilkan <i>Qrcode</i>	231
Lampiran 46 Kode Halaman Menampilkan Buku Tamu	231
Lampiran 47 Kode Halaman Menambahkan Kehadiran Pengunjung.....	231
Lampiran 48 Kode Halaman Melihat Konfirmasi <i>Event</i>	232

BAB I PENDAHULUAN

1.1. Latar Belakang

Indonesia memiliki berbagai macam pola atau motif batik yang berbeda disetiap daerah. Pendataan budaya selama 9 tahun terakhir yang dilakukan oleh Bandung Fe Institute dan Sobat Budaya telah tercatat sedikitnya terdapat 5.849 motif batik dari Aceh sampai Papua. Batik menjadi salah satu permasalahan yang dialami oleh masyarakat mengenai minimnya pengetahuan citra batik yang ada di Indonesia. Keterbatasan tersebut dikarenakan kurangnya media yang membantu pengguna dalam mendapatkan informasi batik yang ingin diketahuinya, khususnya ketika diadakannya *event* pameran batik. Dimana pada saat *event* pameran batik, umumnya pengunjung masih melakukan interaksi secara langsung kepada pengrajin untuk memperoleh informasi batik tersebut. Bahkan di beberapa pameran batik sering tidak ditemukannya penjaga pemeran, sehingga menyulitkan masyarakat dalam mengetahui informasi mengenai batik tersebut. Disisi lain, untuk memperoleh informasi dari pengrajin membutuhkan waktu yang lama, sehingga diperlukannya media yang mampu menghasilkan informasi secara cepat dan akurat dalam mengenali citra batik.

Saat ini sudah terdapat beberapa penelitian yang diciptakan untuk menangani masalah pengenalan motif batik. Pengidentifikasi motif batik tradisional Surakarta dengan klasifikasi menggunakan *backpropagation* (Kusanti & Agus, 2019). Sistem yang lain yaitu menggunakan metode *ConvNet* yang mempelajari representasi data pada klasifikasi pola batik (Gultom dkk., 2018). Metode yang memanfaatkan sudut juga digunakan dalam sistem klasifikasi (Nuraedah & Bakri, 2017). Penggunaan *image retrieval* juga sudah diterapkan pada proses pengenalan motif batik (Sari, 2018). Penelitian lainnya yaitu menyediakan fitur penggambaran motif batik (Kahamdany dkk., 2018). Namun, dari sistem yang dibangun tersebut belum ada yang terintegrasi dengan *smartphone* serta belum adanya fitur untuk menampilkan detail dari motif dan fitur manajemen *event* pameran batik.

Laporan tugas akhir ini membahas mengenai pembangunan aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* sebagai solusi dari permasalahan diatas. Pada aplikasi ini terdapat fitur deteksi untuk

mengenali motif batik pada suatu kain yang mengandalkan kamera *smartphone* untuk mendeteksi motif batik, sehingga memudahkan pengunjung pameran dalam mengenali batik dengan cepat. Selain itu, aplikasi ini mempunyai fitur manajemen *event* pameran batik yang menjadi salah satu keunggulan dalam mengelola *event* agar memperoleh data minat pengunjung pameran batik. Gagasan tersebut menjadi solusi dalam mempertahankan citra batik di kalangan masyarakat Indonesia dan membantu para pengrajin batik dalam memasarkan produk batik yang dijual.

1.1.1. Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada laporan tugas akhir ini adalah sebagai berikut:

- 1) Masih terbatasnya informasi mengenai batik pada saat pameran.
- 2) Masih sulitnya mengetahui seberapa besar minat pengunjung terhadap *event* pameran.

1.1.2. Tujuan

Tujuan pembuatan aplikasi ini adalah:

- 1) Membangun aplikasi yang dapat mengenali motif batik dan menampilkan informasi secara langsung berbasis android dan *image processing*.
- 2) Membangun aplikasi yang mampu menampung respon pengunjung terhadap pameran batik.

1.1.3. Manfaat

Manfaat pembuatan aplikasi ini adalah:

- 1) Memudahkan pengunjung pameran dalam memperoleh informasi motif batik sesuai dengan objek yang dideteksinya menggunakan perangkat android.
- 2) Memudahkan pengrajin batik dalam menangani pertanyaan dari pengunjung pameran batik.
- 3) Membantu pengelola *event* pameran dalam meningkatkan kualitas pameran batik.

1.2. Batasan Masalah

Dalam pembuatan Tugas Akhir ini, masalah yang akan dibahas terbatas pada:

- 1) Dataset batik menggunakan data sekunder yang diperoleh melalui google dengan format jpg/png.
- 2) Jumlah sampel yang digunakan adalah 7200 citra batik yang dibagi dalam 12 kelas.
- 3) Dataset batik difokuskan pada batik daerah jawa.
- 4) Aplikasi terbatas hanya dapat mendeteksi satu motif didalam satu kain utuh yang mempunyai banyak motif.

1.3. Sistematika

Sistematika penulisan pada tugas akhir ini dibagi menjadi beberapa bab yang meliputi:

BAB I PENDAHULUAN

Berisi latar belakang yang juga meliputi perumusan masalah, tujuan, dan manfaat pembuatan tugas akhir, batasan masalah sistem, dan sistematika penulisan laporan tugas akhir.

BAB II TINJAUAN PUSTAKA

Berisi beberapa penelitian yang telah dilakukan oleh ahli dan berkaitan dengan tugas akhir serta membantu dalam pembuatan tugas akhir. Selain itu, terdapat perbandingan sistem yang telah ada dengan tugas akhir yang dikerjakan dengan adanya beberapa indikator.

BAB III KEGIATAN PELAKSANAAN

Berisi kegiatan pelaksanaan pembuatan sistem menggunakan metode *scrum*.

BAB IV ANALISIS DAN PEMBAHASAN

Berisi hasil dari analisis dan rekapitulasi pengujian terhadap sistem yang telah dikerjakan dalam tugas akhir ini.

BAB V KESIMPULAN

Berisi kesimpulan dibuatnya laporan tugas akhir.

DAFTAR PUSTAKA

Daftar yang menampilkan sumber pustaka yang dijadikan panduan penggerjaan laporan tugas akhir.

LAMPIRAN

Dokumen tambahan yang ditambahkan (dilampirkan) ke laporan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas mengenai penelitian terkait dengan konsep serupa yang telah dilakukan oleh peneliti sebelumnya sebagai pembanding dalam pembuatan tugas akhir aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing*. Selain itu, akan dipaparkan beberapa teknologi yang digunakan dalam pembuatan tugas akhir ini.

2.1. Penelitian Terkait

Pada bagian ini dipaparkan beberapa penelitian terkait sistem *detection* yang telah dibuat oleh peneliti. Selain itu, terdapat perbandingan antara sistem *detection* dengan sistem lain yang sudah ada. Beberapa penelitian terkait sistem *detection* diantaranya metode yang digunakan sistem mempengaruhi keakuratan masing-masing pola pada citra batik Surakarta. Data gambar digunakan untuk data pelatihan adalah 100 data gambar dengan ukuran 256 x 256 piksel, dengan data gambar uji digunakan sebanyak 20 data gambar. Objek yang diperoleh ditambahkan oleh proses dan dihitung menggunakan *Grey Level Co-Occurrence Matrix* untuk mendapatkan karakteristik masing-masing gambar. Karakteristik yang digunakan adalah empat variabel (energi, kontras, homogenitas, dan korelasi). Variabel yang dihasilkan digunakan sebagai input ke klasifikasi menggunakan *backpropagation*. Hasil tes memperoleh tingkat akurasi 95%, dengan tingkat kesalahan 0,05% (Kusanti & Agus, 2019). Namun, penggunaan *dataset* dalam sistem ini masih terlalu sedikit sedangkan untuk proses deteksi batik membutuhkan data yang besar tergantung dengan banyaknya kelas yang akan digunakan.

Uji coba sistem menggunakan dataset 2.092 batik (5 kelas), percobaan menunjukkan bahwa model yang diusulkan, yang menggunakan ConvNet VGG16 sebagai ekstraktor fitur (*transfer learning*), mencapai rata-rata sedikit lebih baik dengan akurasi $89 \pm 7\%$ (Gultom dkk., 2018). Namun, dalam proses klasifikasi motif batik parang dan lereng hasil akurasinya rendah, jadi *output* yang dihasilkan terkadang tertukar.

Lapisan *Fully-Connected* di mana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya seperti halnya jaringan saraf

tiruan biasa. Penelitian ini menggunakan metode *Multi Layer Perceptron* (MLP) untuk mengolah data sehingga bisa diklasifikasikan (Danukusumo, 2017). Namun, penelitian ini menggunakan metode MLP yang kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik sedangkan aplikasi Batik.in lebih cocok menggunakan *sparsely connected*.

Penelitian hasil dari pendekripsi klasifikasi meja dan kursi pada suatu citra digital menggunakan *Convolutional Neural Network*. Tingkat akurasi model berkisar antara 70% hingga 99%. Teknik pengumpulan datanya menggunakan *crawling image* yang didapat di *google image* dengan memanfaatkan *console* pada Google Chrome. Pengenalan objek motif ukiran Jepara menggunakan *framework tensorflow* dengan dataset sebanyak 500 gambar. Proses pengklasifikasian menggunakan arsitektur dari CNN (Dewi, 2018). Namun, pada *training* tidak menggunakan metode *transfer learning* yang sebenarnya berguna untuk penggunaan kembali pengetahuan dari model yang telah dilatih sebelumnya untuk melakukan tugas baru. Model *transfer learning (pre-trained model)* merupakan *image feature extraction* yang sudah dilatih di ImageNet.

Mengklasifikasi motif kain tradisional Batik Bomba Kaili dengan tahapan akuisisi, praproses, ekstraksi fitur tekstur dan tahapan klasifikasi. Data yang digunakan dalam penelitian ini yaitu 128 dengan 93 data latih 35 data uji. Proses ekstraksi menggunakan metode *Gray Level Co-occurrence Matrices* (GLCM) dengan sudut 0° , 45° , 90° dan 135° . Fitur tekstur setiap motif batik Bomba akan diklasifikasi dengan metode *Linear Support Vector Machine* (LSVM). Hasil akurasi klasifikasi jika digabungkan pada tahap pelatihan menjadi 80.65% dan pada tahap pengujian menghasilkan akurasi sebesar 77.14% (Nuraedah & Bakri, 2017). Namun, penggunaan *deep learning* dengan metode *convolutional neural network* akan lebih membantu dalam klasifikasi berbentuk gambar dengan pemodelan abstraksi tingkat tinggi dengan menggunakan sekumpulan fungsi transformasi non-linier.

Model CNN pada penelitian ini menggunakan *input shape* berukuran 64x64, nilai *learning rate* 0.001, ukuran filter 3x3, Jumlah Epoch 20, Data *training* 240, dan

data *testing* 60. Menghasilkan tingkat akurasi *training* dan *testing* dalam melakukan klasifikasi gambar wayang golek sebesar 95 % *training* dan 90 % *testing*. Penelitian ini menggunakan data *testing* baru sebanyak 60 untuk diujikan kedalam model yang telah dibuat. Hasil *testing* menghasilkan tingkat akurasi baru dalam melakukan klasifikasi gambar Wayang Golek sebesar 93% (Nurhikmat, 2018). Metode ini cocok untuk proses klasifikasi gambar, dan tepat untuk diterapkan pada aplikasi Batik.in. Namun, perlu direlasikan dengan *platform android* agar dapat digunakan dengan mudah.

Image Processing menggunakan metode *deep learning* untuk mengklasifikasikan unggas diantaranya dengan kategori emu, flamingo, ibis, *pigeon*, dan *roaster* yang terdiri dari 150 citra. Selain kategori tersebut dihasilkan pula 3 kategori yaitu *cougar*, *crocodile*, dan *face*. Hasil dari 5 kategori unggas menunjukkan bahwa persentase keberhasilan 20% sedangkan untuk 3 kategori lainnya menunjukkan persentase keberhasilan 50%. Penggunaan metode CCN dengan tahap pertama yaitu algoritma *feedforward* dan tahapan kedua yaitu *backpropagation* (Eka Putra, 2016). Namun, penggunaan *feedforward* dalam pembuatan aplikasi Batik.in tidak terlalu tepat dan lebih menitikberatkan pada algoritma *backpropagation* dalam proses klasifikasi.

Mempelajari tentang klasifikasi gambar dengan menggunakan *deep neural network* (DNN) atau dikenal juga dengan *deep learning* dengan menggunakan *framework tensorflow*. Python digunakan sebagai bahasa pemrograman karena dilengkapi dengan kerangka kerja *tensorflow*. Data *input* terutama berfokus pada kategori bunga yang ada lima (5) jenis bunga. *Deep neural network* (DNN) telah memilih sebagai pilihan terbaik untuk proses pelatihan karena menghasilkan persentase akurasi yang tinggi. Hasil dibahas dalam hal akurasi klasifikasi gambar dalam persentase. Mawar mendapatkan 90,585% dan sama dengan bunga jenis lain di mana rata-rata hasilnya mencapai 90% ke atas (Abu dkk., 2019). Aplikasi Batik.in dalam proses klasifikasi model akan menggunakan metode yang dengan penelitian tersebut yaitu penggunaan CNN, *transfer learning* dan MobileNet.

Penelitian sebuah *Unmanned Ariel Vehicle* (UAV) yang memiliki tujuan utama untuk mengembangkan OpenCV-Python. Saat ini, UAV digunakan untuk mendeteksi dan menyerang target darat yang tersembunyi. Kelemahan utama untuk jenis UAV adalah bahwa terkadang objeknya tidak terdeteksi dengan benar. Proyek ini bertujuan untuk menghindari hal yang tidak diinginkan tersebut tabrakan dan kerusakan UAV. Langkah deteksi yang pertama yaitu mendeteksi bagian wajah, mengidentifikasi fitur, dan *face tracking* (Jalled & Voronkov, 2016). Penggunaan metode ini jika diterapkan pada klasifikasi Batik.in tidak terlalu tepat, karena metode yang tepat untuk pengolahan data piksel dan spasial lebih cenderung menggunakan *convolutional neural network*.

Hasil performa dengan penggunaan 5 fitur gabungan hasil GLCM berfungsi baik sebagai ekstraksi fitur, hal ini ditandai dengan hasil performa yang diberikan dari seluruh percobaan yang dilakukan lebih dari 50%. Hasil perbesaran citra sebelum fitur ekstraksi dianggap berkerja dengan baik, mengingat hasil performa yang dihasilkan dari percobaan tersebut. Perbesaran citra memberikan dampak terhadap persebaran fitur setiap objek ikan yang akan dilakukan pengenalan. Hasil dari kinerja *image retrieval Euclidean distance* dalam klasifikasi motif kain batik memberi nilai akurasi 44.44% (Sari, 2018). Akurasi dengan nilai terlalu rendah belum siap untuk diimplementasikan secara langsung karena *output* yang dihasilkan masih memiliki kemungkinan untuk tertukar datanya.

Aplikasi android yang mampu mendeteksi objek dalam gambar yang diambil dari galeri seluler, berdasarkan warna, bentuk, atau fitur lokal yang dikembangkan. Deteksi warna gambar diproses dalam domain warna HSV untuk deteksi warna yang lebih baik. Berdasarkan hasil percobaan, aplikasi ini mampu mendeteksi dengan benar sebelas warna berbeda, mendeteksi bentuk geometris dua dimensi termasuk lingkaran, segitiga, kotak, dan cocok dengan gambar pemandangan untuk kondisi yang berbeda (A. Elrefaei dkk., 2017). Namun, konsep aplikasi ini kurang tepat jika diterapkan pada aplikasi Batik.in karena aplikasi yang akan dibuat menggunakan model warna RGB.

Pengujian terhadap aplikasi android menggunakan *library* yaitu *computer vision* untuk membedakan warna. Aplikasi ini bekerja secara *real time camera* dan dapat membedakan warna sesuai yang dipilih oleh pengguna. Penggunaan OpenCV untuk segmentasi warna dengan tahapan *captures frames* didalam RGBA, lalu dikonversi 8 bit piksel matrik dan mengonversikan warna RGBA ke HSV (Taskin dkk., 2017). Namun, aplikasi Batik.in akan menggunakan model warna RGB untuk melakukan tahapan ekstraksi.

Proses pengolahan citra dilakukan dengan menghitung nilai persentasi *red, green* dan *blue* dari setiap jenis daging dengan menggunakan metode *color blob detection*. Nilai parameter diperoleh dari penelitian sebelumnya dan berdasarkan hasil pengolahan citra yang dilakukan langsung menggunakan aplikasi. Aplikasi memiliki tingkat akurasi yaitu pada jenis daging sapi: 84.28%, jenis daging babi: 92.86% dan jenis daging kambing: 94.28% (Maulana, 2018). Konsep model RGB juga akan diaplikasi pada Batik.in karena untuk tahap ekstraksi aplikasi ini juga menggunakan model RGB.

Penggunaan algoritma dalam pengembangan aplikasi ini yaitu algoritma *threshold*, *contour* dan *kNearestNeighbors*. Aplikasi digitalisasi plat nomor ini dapat membaca dan mengenali segmen tertentu pada *image* sebagai plat nomor. Tingkat akurasi pengenalan plat mencapai 95% dengan menggunakan perangkat *smartphone* yang memiliki fitur kamera beresolusi 5 *Megapixel* atau lebih (Masdiyasa dkk., 2019). Namun, metode yang digunakan dalam pembuatan aplikasi ini kurang tepat untuk diterapkan pada Batik.in karena aplikasi yang akan kami buat akan menggunakan model *convolutional neural network*.

Pengujian *tensorflow* yang merupakan *framework deep learning* dari Google yang menjadi sebuah alternatif dari *deep neural network*. Performansi model menggunakan GPU untuk proses klasifikasi data, dengan *error* 10.37% (Gomà, 2016). Pada pembuatan aplikasi Batik.in juga menggunakan *tensorflow* untuk mempermudah proses klasifikasi.

Convolutional neural network (CNN) digunakan untuk berbagai aplikasi seperti pengenalan objek, gambar beresolusi besar, segmentasi semantik, dll. Arsitektur

CNN diantaranya AlexNet, VGG16 dan VGG19 untuk melakukan pengenalan objek. Penggunaan *transfer learning* juga berpengaruh terhadap performansi VGG19. Proses klasifikasinya sendiri menggunakan *Support Vector Machine* (SVM) (Shaha & Pawar, 2018). Penggunaan *transfer learning* juga akan diimplementasikan pada proses klasifikasi Batik.in.

Pengujian dilakukan pada ketiga model yang sudah diterapkan pada sistem yang telah dikembangkan, yaitu ResNet50, Xception, dan VGG16. Analisis performa menggunakan *pre-trained CNN model*. Kumpulan gambar yang akan digunakan untuk pengujian adalah 79 gambar anjing dari berbagai ras yang telah diberi label untuk memudahkan pengujian. Dari tiga *pre-trained* model dari Keras yang diujikan (ResNet50, Xception dan VGG16), Xception memiliki performa keakuratan yang lebih baik daripada dua model lainnya (Pangestu & Bunyamin, 2018). Proses klasifikasi batik akan menggunakan *tensorflow* versi 2.0 yang didalamnya sudah terdapat Keras.

Proyek ini bertujuan untuk menunjukkan seberapa dalam model seperti, VGG16, VGG19, Inception V3, InceptionResNet V2 yang dapat digunakan pada data berukuran sangat kecil dengan total 6000 gambar yang terbagi menjadi 3000 data pelatihan, 2000 data validasi dan 1000 data pengujian tanpa adanya *overfitting*. Semua model sudah dilatih dengan ImageNet (Shu, 2019). Jika dilihat dari penggunaan metode dan hasil dari akurasi yang bagus bisa diterapkan dalam proses klasifikasi model pada aplikasi Batik.in, karena metode pemodelan CNN lebih tepat diimplementasikan pada gambar piksel.

Pembahasan mengenai potensi penerapan arsitektur model VGG16 untuk klasifikasi tumbuhan. Pengerajan sebelumnya menggunakan *transfer learning*, tetapi ada masalah dalam penerapan CNN, ketika jumlah data yang terbatas menyebabkan *overfitting*. Percobaan berikutnya sukses dengan 2800 gambar bunga dan menghasilkan akurasi percobaan 96.25%, 93.93% validasi dan 89.96% pengujian (Abas dkk., 2018). Namun, terjadinya *overfitting* akan menyebabkan akurasi data yang kurang bagus pada proses *testing*.

Pendekatan untuk mendeteksi *note object* menggunakan CNN melalui *smartphone* dengan memanfaatkan beberapa model dasar secara *realtime*. Data dapat dilatih dengan jumlah data yang kecil tetapi tetap mencapai keakuratan yang tinggi menggunakan *transfer learning*. Peningkatan model *deep learning* dengan heuristik jauh lebih tinggi jika menggunakan DL karena menghasilkan peningkatan sebesar 117.7% sedangkan heuristik sebesar 26.1% (Alsing, 2018). Jadi, dapat dipastikan bahwa model *deep learning* memang tepat jika diterapkan pada objek yang mempunyai kekompleksan pada data atau objek.

Dari beberapa penelitian diatas yang dilakukan oleh ahli, belum terdapat sebuah sistem yang mempunyai fitur untuk menampilkan deskripsi dari batik yang telah dideteksi dan belum adanya sistem interaktif seperti pemberian nilai ataupun komentar pada setiap batik. Oleh karena itu, tugas akhir aplikasi *object detection* ini akan mewujudkan sebuah sistem dengan fitur yang telah disebutkan sebelumnya. Selain itu, *database* yang digunakan adalah MySQL yang akan menyimpan informasi seputar aplikasi.

Tabel 2. 1 Perbandingan Sistem Aplikasi

No.	Sistem	Metode/Fitur/Teknologi						
		1	2	3	4	5	6	7
1	<i>Application System for Identification of Surakarta Traditional Batik Images (SABATARA)</i>	√	-	-	-	-	-	-
2	<i>Batik Classification Using Deep Convolutional Network Transfer Learning</i>	√	√	√	-	-	-	-
3	<i>Implementasi Deep Learning Menggunakan Convolutional Neural</i>	√	√	-	√	-	-	-

	<i>Network Untuk Klasifikasi Aplikasi Citra Candi Berbasis GPU</i>							
4	<i>Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network</i>	√	√	-	√	-	-	-
5	Klasifikasi Motif Kain Tradisional Batik Bomba Kaili	√	-	-	-	-	-	-
6	<i>Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma Convolutional Neural Network pada Citra Wayang Golek</i>	√	√	-	√	-	-	-
7	Klasifikasi Citra Menggunakan Convolutional Neural Network (<i>Cnn</i>) pada Caltech 101	√	√	-	-	√	-	-
8	<i>A study on Image Classification based on Deep Learning and Tensorflow</i>	√	√	-	√	-	-	-

9	<i>Object Detection using Image Processing</i>	√	-	-	-	-	-	-
10	Klasifikasi Pengenalan Motif Batik Berbasis <i>Image Retrieval</i>	√	-	-	-	-	-	-
11	<i>Development of an Android Application for Object Detection Based on Color, Shape, or Local Features</i>	√	-	-	-	-	-	-
12	<i>Image Processing on Android Devices With Opencv</i>	√	√	-	-	-	-	-
13	Identifikasi Jenis Daging Menggunakan <i>Image Processing</i> Berbasis Android	√	√	-	-	√	-	-
14	Identifikasi Plat Nomor Kendaraan Bermotor Menggunakan Metode <i>Multi-Step Image Processing</i> Berbasis Android	√	-	-	-	-	-	-
15	<i>Image Recognition with Deep Learning Techniques and Tensorflow</i>	√	√	-	√	-	-	-
16	<i>Transfer Learning for Image Classification</i>	√	√	√	-	-	-	-
17	Analisis Performa dan Pengembangan Sistem	√	√	-	√	-	-	-

	Deteksi Ras Anjing pada Gambar dengan Menggunakan <i>Pre-Trained CNN Model</i>							
18	<i>Deep learning for image classification on very small datasets using transfer learning</i>	√	√	√	√	-	-	-
19	<i>VGG16 for plant image classification with transfer learning and data augmentation</i>	√	√	√	√	-	-	-
20	<i>Mobile Object Detection using TensorFlow Lite and Transfer Learning</i>	√	√	√	√	-	-	-
21	Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android Dan <i>Image Processing</i> .	√	√	√	√	√	√	√

Keterangan:

- 1) *Object detection*
- 2) Metode *convolutional neural network*
- 3) *Transfer learning*
- 4) *Tensorflow*
- 5) Detail informasi dari objek
- 6) *Like* dan *comment*
- 7) Manajemen event pameran

Berdasarkan Tabel 2.1 dapat dilihat bahwa dari beberapa sistem yang dibuat oleh peneliti yang berkaitan dengan tugas akhir ini belum ada yang membuat aplikasi batik dengan menampilkan informasi batik meliputi nama batik, asal daerah batik, ciri khas motif batik, dan filosofi batik. Selain itu, interaktif pada aplikasi seperti pemberian *like* dan komentar akan ditambahkan dalam aplikasi yang akan dibuat. Metode yang digunakan dalam proses klasifikasi yaitu *convolutional neural network* yang merupakan salah satu dari model *deep learning*. Penggunaan *transfer learning* juga membantu dalam *training* ulang model yang sudah dilatih. *Framework* yang disediakan oleh Google yaitu *tensorflow* juga digunakan dalam membantu proses pemodelan data. Beberapa kelebihan lainnya yang dimiliki aplikasi deteksi ini yaitu menyediakan fitur untuk memanajemen *event* pameran batik dimana dapat menjadi objek secara nyata untuk ujicoba aplikasi. Menampilkan deskripsi atau detail dari motif batik yang terdeteksi membutuhkan bantuan *database* untuk menyimpan informasi. Label model akan berelasi dengan *database* yang sudah dibuat, dalam aplikasi ini menggunakan *database MySQL*.

2.2. Teknologi yang digunakan

Pada bagian ini membahas teknologi yang digunakan dalam pembuatan tugas akhir aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* yang diantaranya sebagai berikut.

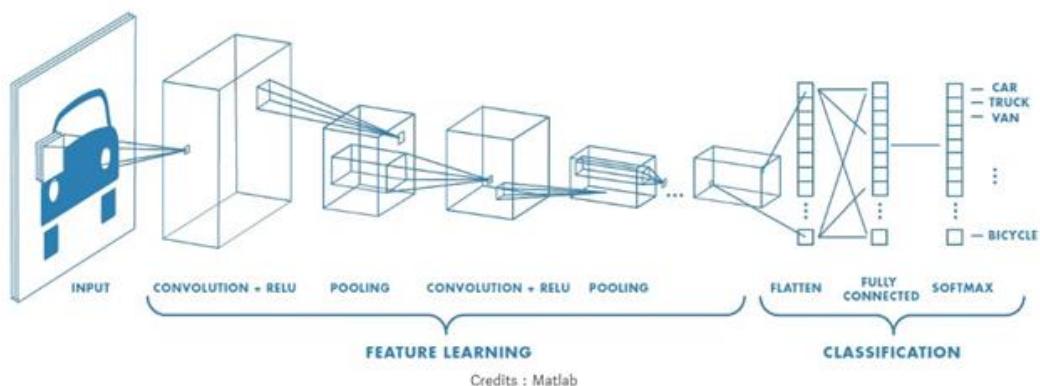
2.2.1. Deep Learning

Deep Learning merupakan salah satu cabang ilmu dari *machine learning* yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data dengan menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam (Hadinisa dkk., 2018). Algoritma pada *Deep Learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*) (Lorentius dkk., 2017).

Deep Learning mempunyai algoritma yang beragam dengan spesifikasi yang berbeda diantaranya ada *Deep Neural Network* (*DNN*) dengan struktur algoritma berlapis-lapis dengan model sirkuit saraf kranial manusia dan hewan yang dirancang untuk mengenali pola yang disebut *neural network* (*NN*). Kemudian ada *Convolutional Neural Network* (*CNN*) yang menggunakan *neural network* tipe propagasi maju dengan abstraksi informasi lokal dan universalitas posisi. Dan *Recurrent Neural Network* (*RNN*) yang menggunakan *neural network* di mana sinyal dua arah disebarluaskan dengan struktur rekursif di lapisan tengah agar dapat menangani variabel data panjang seperti data suara dan gambar bergerak. Dalam pembuatan tugas akhir ini digunakan algoritma *Convolutional Neural Network* (*CNN*).

2.2.2. CNN (*Convolutional Neural Network*)

Convolutional Neural Network merupakan salah satu jenis *neural network* yang biasanya digunakan dalam pengolahan data *image*. Konvolusi atau biasa yang disebut dengan *convolution* adalah matriks yang memiliki fungsi melakukan *filter* pada gambar. *Convolutional Neural Network* memiliki beberapa layer yang difungsikan untuk melakukan *filter* pada setiap prosesnya. Prosesnya disebut dengan proses *training*. Pada proses *training* terdapat tiga tahapan yaitu *Convolutional layer*, *Pooling layer*, dan *Fully connected layer* (Santoso & Ariyanto, 2018). Berikut adalah jaringan arsitektur *Convolutional Neural Network*.



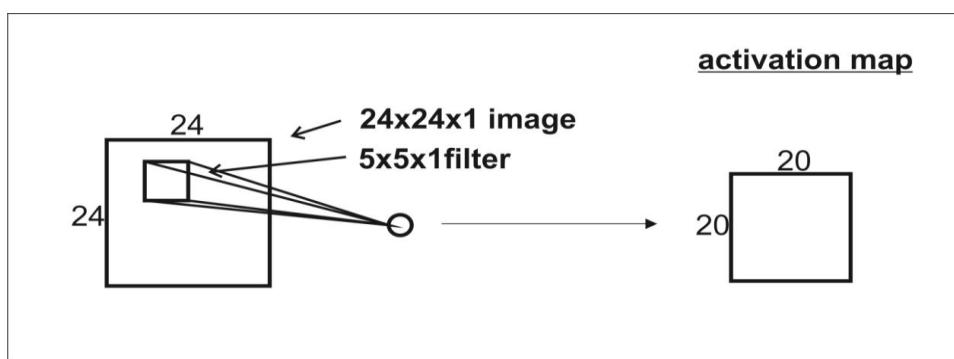
Gambar 2. 1 Arsitektur *Convolutional Neural Network*

(Santoso & Ariyanto, 2018)

Berdasarkan gambar tersebut, tahap pertama pada arsitektur CNN adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*), Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*. Pada aplikasi yang dibuat dalam tugas akhir ini, CNN digunakan untuk mengklasifikasi motif batik. Penjelasan tahapan *Convolutional Neural Network* (CNN) sebagai berikut:

2.2.2.1. Convolution Layer

Convolutional Layer merupakan *layer* pertama yang menerima *input* gambar langsung pada arsitektur. *Convolutional Layer* bagian yang melakukan operasi konvolusi yaitu mengkombinasikan linier *filter* terhadap daerah lokal. *Layer* ini yang pertama kali menerima gambar yang diinputkan pada arsitektur. Bentuk *layer* ini adalah sebuah *filter* dengan panjang (piksel), tinggi (piksel), dan tebal sesuai dengan *channel image* data yang diinputkan. Ketiga *filter* ini akan bergeser keseluruhan bagian gambar. Pergeseran tersebut akan melakukan operasi “dot” antara input dan nilai dari *filter* tersebut sehingga akan menghasilkan output yang disebut sebagai *activation map* atau *feature map*. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input*. Gambar 2.2 menampilkan proses konvolusi yang ada di dalam *convolutional layer*.



Gambar 2. 2 Proses Konvolusi

Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada output fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *Feature Map* dari *input* citra. *Input* dan *output* ini dapat dilihat sebagai dua argumen bernilai riil. Operasi konvolusi dapat dituliskan seperti persamaan 2.1

$$s(t) = (x * t)(t) = \sum_{\alpha=-\infty}^{\infty} x(\alpha) * w(t - \alpha) \quad (2.1)$$

Keterangan :

$S(t)$ = Fungsi hasil operasi konvolusi

X = Input

W = bobot (kernel)

Fungsi $s(t)$ memberikan output tunggal berupa *feature map*. Argumen pertama adalah input yang merupakan x dan argumen kedua w sebagai kernel atau filter. Apabila dilihat input sebagai citra dua dimensi, maka bisa dikatakan t sebagai piksel dan menggantinya dengan i dan j . Maka dari itu, operasi untuk konvolusi ke input dengan lebih dari satu dimensi dapat ditulis seperti pada persamaan 2.2 dan 2.3

$$s(i,j) = (K * I)(i,j) = \sum_{\infty} I(i-m, j-n) K(m,n) \quad (2.2)$$

$$s(i,j) = (K * I)(i,j) = \sum_{\infty} I(i-m, j+n) K(m,n) \quad (2.3)$$

Keterangan :

$s(i,j)$ = Fungsi hasil operasi konvolusi

I = Input

K = Kernel

i,j = pixel dari Input

m,n = pixel dari Kernel

Berdasarkan persamaan 2.2 dan 2.3 merupakan perhitungan dasar dalam operasi konvolusi, dengan i dan j adalah sebuah piksel dari citra. Perhitungan tersebut bersifat komulatif dan muncul saat K sebagai kernel, kemudian I sebagai input dan kernel yang dapat dibalik relatif terhadap input. Sebagai alternatif operasi konvolusi dapat dilihat sebagai perkalian perkalian matriks antara citra masukan dan kernel

dimana keluarannya dihitung dengan *dot product*. Selain itu, penentuan *volume output* juga dapat ditentukan dari masing-masing lapisan dengan hyperparameters. Hyperparameter yang digunakan pada persamaan 4 digunakan untuk menghitung banyaknya neuron aktivasi dalam sekali *output*. Perhatikan persamaan 2.4.

$$W - F + 2P / (S + 1) \quad (2.4)$$

Keterangan :

W = Ukuran volume gambar

F = Ukuran Filter

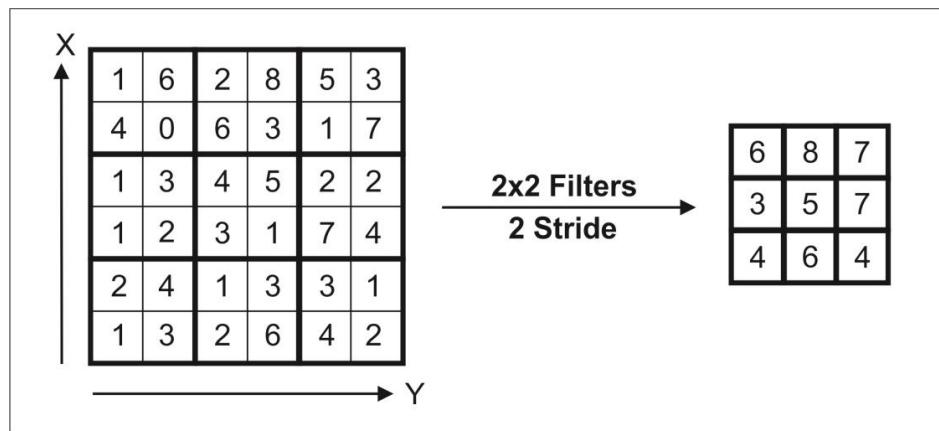
P = Nilai Padding yang digunakan

S = Ukuran Pergeseran (*Stride*)

Berdasarkan persamaan 2.4, dapat dihitung ukuran spasial dari volume output dimana hyperparameter yang dipakai adalah ukuran *volume* (W), *filter* (F), *Stride* yang diterapkan (S) dan jumlah *padding* nol yang digunakan (P). *Stride* merupakan nilai yang digunakan untuk menggeser *filter* melalui *input* citra dan Zero Padding adalah nilai untuk mendapatkan angka nol di sekitar *border* citra.

2.2.2.2. Pooling Layer

Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. Pada dasarnya *pooling layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max-pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *Pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran *volume output* pada *Feature Map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *Overfitting* (Nurhikmat, 2018). Gambar 2.3 menampilkan proses yang ada pada *pooling layer*.

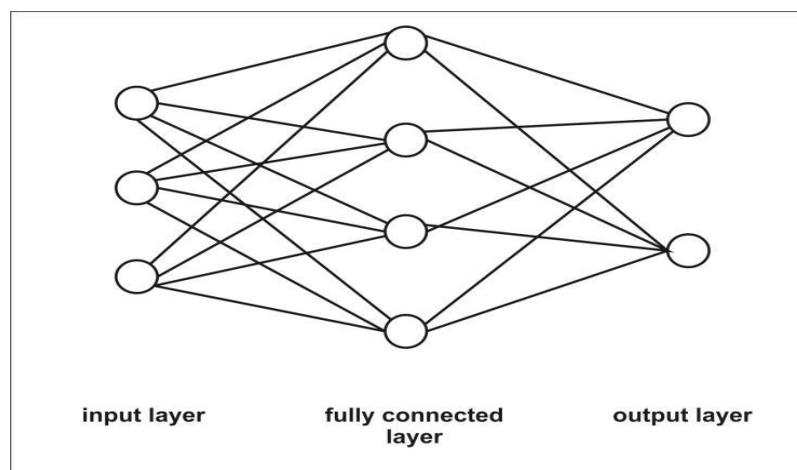


Gambar 2. 3 Proses *Pooling Layer*

(Santoso & Ariyanto, 2018)

2.2.2.3. *Fully Connected Layer*

Fully connected layer mengambil input dari hasil output *pooling layer* yang berupa *feature map*. *Feature map* tersebut masih berbentuk multidimensional array maka lapisan ini akan melakukan *reshape feature map* dan menghasilkan vektor sebanyak n-dimensi dimana n adalah jumlah kelas *output* yang harus dipilih program. Misalnya lapisan terdiri dari 500 neuron, maka akan diterapkan softmax yang mengembalikan daftar probabilitas terbesar untuk masing-masing 10 label kelas sebagai klasifikasi akhir dari jaringan (Dutt Anuj, & Dutt Aashi, 2017). Berikut ini merupakan proses yang terjadi di dalam *fully connected layer*.



Gambar 2. 4 Proses *Fully Connected Layer*

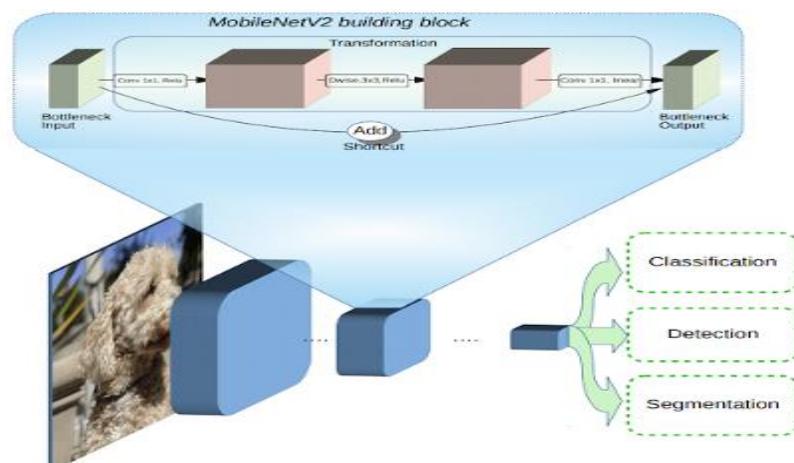
(Santoso & Ariyanto, 2018)

2.2.3. Transfer Learning

Transfer learning merupakan penggunaan kembali pengetahuan dari model yang telah dilatih sebelumnya untuk melakukan tugas baru. Model *transfer learning* (*pre-trained model*) yang digunakan adalah *image feature extraction* dengan arsitektur InceptionV3 yang telah dilatih di ImageNet. Model tersebut memuat contoh jaringan yang terlatih yang dikemas untuk mendapatkan *feature vectors* dari gambar. *Feature vectors* itu sendiri merupakan tensor 1-D yang merepresentasikan keseluruhan gambar yang biasanya digunakan untuk klasifikasi (Hadinisa et al., 2018). Pada pembangunan aplikasi ini *transfer learning* digunakan untuk mempercepat *training* model.

2.2.4. MobileNet V2

MobileNet merupakan salah satu arsitektur *convolutional neural network* (CNN) yang digunakan untuk *computing* pada *resource* yang besar. MobileNet membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*. Pada MobileNet V2 ini ditambahkan fitur *linier bottleneck* dan *shortcut connections* antar *bottlenecks*. Pada bagian *bottleneck* terdapat *input* dan *output* antara model sedangkan lapisan atau layer bagian dalam meng-enkapsulasi kemampuan model untuk mengubah input dari konsep tingkat yang lebih rendah ke deskriptor tingkat yang lebih tinggi. MobileNet V2 digunakan pada sistem ini dikarenakan tingkat akurasi yang tinggi dan kapasitas file yang kecil. Berikut ini merupakan arsitektur MobilNet V2.



Gambar 2. 5 Arsitektur MobileNet-V2

2.2.5. Tensorflow

Tensorflow merupakan perpustakaan perangkat lunak yang dikembangkan oleh Tim Google Brain dalam organisasi penelitian Mesin Cerdas Google, untuk tujuan melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam. *Tensorflow* menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana permasalahannya adalah waktu yang dibutuhkan untuk melakukan perhitungan (Dewi, 2018).

2.2.6. Tf-lite

TensorFlow Lite terdiri dari software untuk menjalankan model yang sudah ada sebelumnya, dan seperangkat alat yang dapat digunakan untuk menyiapkan model untuk digunakan pada perangkat seluler tertanam. *TensorFlow* dapat berjalan dengan model klasifikasi gambar yang umum termasuk *inception* dan *mobilene*. Model yang sudah dibuat akan diunduh dalam bentuk file berformat *tf.lite* untuk dipasang di android.

2.2.7. Python

Python merupakan bahasa pemrograman dengan tujuan umum yang dikembangkan secara khusus untuk membuat *source code* mudah dibaca. Python juga memiliki *library* yang lengkap sehingga memungkinkan *programmer* untuk membuat aplikasi yang mutakhir dengan menggunakan *source code* yang tampak sederhana (Dewi, 2018). Bahasa pemrograman python digunakan untuk pembuatan model, dari tahapan *processing* data sampai mengubah model menjadi ekstensi *.tf* dalam pembuatan aplikasi ini.

2.2.8. Android Studio

Android Studio merupakan sebuah Integrated Development Environment (IDE) khusus untuk membangun aplikasi yang berjalan pada *platform* android. Android studio ini berbasis pada IntelliJ IDEA, sebuah IDE untuk bahasa pemrograman Java. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa XML. Android studio juga terintegrasi dengan Android Software Development Kit (SDK) untuk *deploy* ke

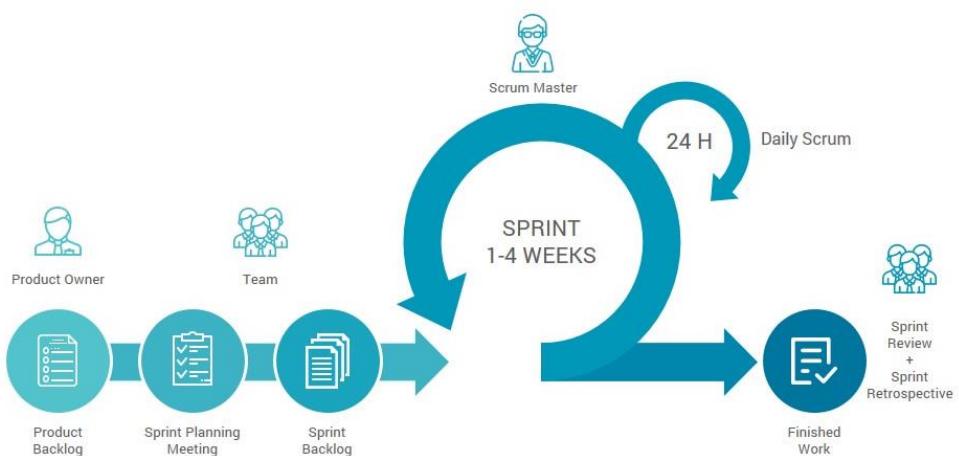
perangkat android (Al Fikri, 2016). Android studio digunakan sebagai *software* untuk membuat aplikasi pendekripsi batik ini.

2.2.9. MySQL

MySQL (*My Structure Query Language*) adalah salah satu Basis Data Management System (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postgre SQL, dan lainnya. MySQL berfungsi untuk mengolah basis data menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis (Nadia et al., 2016). Pada pembuatan aplikasi ini MySQL digunakan sebagai media penyimpanan *database* dan relasi antara label dengan modul.

BAB III KEGIATAN PELAKSANAAN

Pada bab ini akan dibahas mengenai kegiatan pelaksanaan tugas akhir dengan menggunakan metode agile. Metode ini merupakan model pengembangan sistem yang cepat dan lebih mementingkan interaksi dengan user daripada proses dan alat (Taufik, 2018). Sedangkan kerangka kerja yang digunakan adalah scrum, scrum adalah sebuah kerangka kerja untuk pengembangan secara inkremental dengan menggunakan satu atau lebih tim yang *cross-functional* dan *self-organizing* yang terdiri dari kurang lebih tujuh orang pada tiap-tiap tim. Scrum menggunakan iterasi tetap bernama *sprint*, yang dijalankan dalam waktu dua minggu atau tiga puluh hari. Menurut Ken Schwaber dan Jeff Sutherland Scrum adalah suatu metodologi atau kerangka kerja yang terstruktur untuk mendukung pengembangan produk yang kompleks. Scrum terdiri dari sebuah tim yang memiliki peran dan tugas masing-masing. Setiap komponen dalam kerangka melayani tujuan tertentu dan sangat penting untuk kesuksesan penggunaan scrum (Ananda, 2018).



Gambar 3. 1 Tahapan Metode Agile menggunakan Kerangka Kerja Scrum

Gambar 3.1 merupakan tahap – tahap metode agile menggunakan kerangka kerja scrum. Dalam pengembangannya kerangka kerja *scrum* memiliki beberapa tahapan yang berurut yaitu: *product backlog*, *sprint planning meeting*, *sprint backlog*, *finished work*. Tahapan – tahapan dari metode *scrum* adalah sebagai berikut:

1) *Product Backlog*

Tahap ini pengembang aplikasi mendapatkan daftar *backlog* (kebutuhan) yang akan dikelompokkan dalam beberapa *sprint* pada tahap berikutnya. Kebutuhan – kebutuhan ini didapatkan dari hasil wawancara atau diskusi dengan pengrajin batik di Kampung Batik bernama Rini Sari Handayani yang beralamat di Jl. Batik No.698A, Rejomulyo, Kec. Semarang Timur., Kota Semarang. Selain itu hasil wawancara juga didapatkan dari *event* pameran batik yang berada di Java Mall Semarang dengan penjaga *stand* pameran bernama Ning dari Pekalongan, Retno dari Kulonprogo, dan Niken dari Semarang.

2) *Sprint Planning Meeting*

Tahap ini tim pengembang akan mendiskusikan mengenai pembuatan *sprint* pada projek yang akan dikerjakan. Pembuatan *sprint* didasarkan pada tingkat kesulitan *backlog* yang ada.

3) *Sprint Backlog*

Tahap ini merupakan proses penentuan *task* yang akan dimasukan kedalam *sprint*. *Task* disini merupakan pemecahan permasalahan ke bentuk yang lebih kecil dan bertujuan agar mudah dalam memantau perkembangan serta pengembangannya.

4) Penggerjaan *Task*

Pada tahap ini dilakukan penggerjaan *task – task* yang ada pada *sprint backlog* dan waktu penggerjaan diharuskan mengikuti waktu yang sudah ditetapkan pada *sprint* tersebut.

3.1. *Product Backlog*

Pada bagian ini akan ditentukan kebutuhan – kebutuhan yang diperlukan untuk penggerjaan aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing*. Berikut daftar *product backlog*:

- 1) Analisis kebutuhan perangkat
- 2) Analisis perancangan sistem
- 3) Perancangan model dataset

- 4) Pembuatan dan pengujian model dataset
- 5) Perancangan antarmuka aplikasi
- 6) Pembuatan dan pengujian fitur aplikasi
- 7) Perencangan uji kepuasan pengguna

3.2. Sprint Planning Meeting

Pada bagian ini akan ditentukan estimasi waktu penggerjaan *sprint* dan jumlah *sprint* yang diurutkan berdasarkan urutan kebutuhan pembuatan projek. Berikut merupakan daftar *sprint*:

Tabel 3. 1 Daftar *Sprint*

No.	Sprint	Product Backlog	Prioritas	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
1.	1	Analisis kebutuhan perangkat	Sangat Tinggi	1	1
2.	2	Analisis perancangan sistem	Sangat Tinggi	24	20
3.	3	Perancangan model dataset	Sangat Tinggi	50	45
4.	4	Pembuatan dan Pengujian model dataset	Sangat Tinggi	20	16
5.	5	Perancangan antarmuka aplikasi	Sangat Tinggi	72	68
6.	6	Pembuatan dan pengujian fitur aplikasi	Sangat Tinggi	470	410
7.	7	Perancangan uji kepuasan pengguna	Tinggi	31	26
TOTAL				668	586

Dari Tabel 3.1 didapatkan jumlah *sprint* sebanyak tujuh *sprint* dengan mempertimbangkan prioritas serta estimasi waktu yang digunakan dalam pengerjaan *sprint*. Estimasi waktu yang digunakan untuk pengerjaan tujuh *sprint* adalah tiga minggu lebih enam hari dan waktu pengerjaan yang digunakan sebanyak tiga minggu lebih tiga hari untuk menyelesaikan *sprint* yang sudah direncanakan.

3.3. Sprint Backlog

Pada tahap ini ditentukan *task* pada setiap *sprint* yang ditentukan dari tahap sebelumnya.

3.3.1. Sprint 1 Analisis Kebutuhan Perangkat

Pada *sprint* ini akan mengerjakan bagian dari *product backlog* yaitu analisis kebutuhan perangkat, *task* yang dikerjakan antara lain:

Tabel 3. 2 *Sprint 1: Analisis Kebutuhan Perangkat*

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Analisis kebutuhan perangkat	Analisis kebutuhan perangkat keras	0,5	0,5
	Analisis kebutuhan perangkat lunak	0,5	0,5
TOTAL		1	1

Dari Tabel 3.2 didapatkan jumlah *task*, estimasi waktu pengerjaan, dan waktu pengerjaan sebenarnya dalam mengerjakan *task – task* pada *backlog* analisis kebutuhan perangkat.

3.3.2. Sprint 2 Analisis Perancangan Sistem

Pada *sprint* ini akan mengerjakan bagian dari *product backlog* yaitu analisis perancangan sistem, *task* yang dikerjakan antara lain:

Tabel 3. 3 *Sprint 2: Analisis Perancangan Sistem*

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Analisis perancangan sistem	Analisis gambaran umum sistem	1	1
	Analisis <i>block diagram</i>	2	1
	Analisis <i>use case diagram</i>	5	4
	Analisis <i>activity diagram</i>	14	12
	Analisis <i>entity relationship diagram</i>	2	2
TOTAL		24	20

Dari Tabel 3.3 didapatkan jumlah *task*, estimasi waktu penggerjaan, dan waktu penggerjaan sebenarnya dalam mengerjakan *task – task* pada *backlog* analisis perancangan sistem.

3.3.3. *Sprint 3 Perancangan Model Dataset*

Pada *sprint* ini akan mengerjakan bagian dari *product backlog* yaitu perancangan model dataset *task* yang dikerjakan antara lain:

Tabel 3. 4 *Sprint 3: Perancangan Model Dataset*

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Perancangan model dataset	Pengumpulan Dataset Batik	6	6
	Perancangan Implementasi Model	2	2
	<i>Pre-processing</i>	5	5
	<i>Training Model</i>	35	30
	<i>Testing Model</i>	2	2
TOTAL		50	45

Dari Tabel 3.4 didapatkan jumlah *task*, estimasi waktu penggerjaan, dan waktu penggerjaan sebenarnya dalam mengerjakan *task – task* pada *backlog* perancangan model dataset.

3.3.4. Sprint 4 Pembuatan dan Pengujian Model Dataset

Pada *sprint* ini akan mengerjakan bagian dari *product backlog* yaitu pembuatan dan pengujian model dataset, *task* yang dikerjakan antara lain.

Tabel 3. 5 *Sprint 4*: Pembuatan dan Pengujian Model Dataset

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Pembuatan dan Pengujian Model	<i>Import Package dan Library</i>	1	1
	Persiapan Data	1	1
	<i>Train Model</i>	15	12
	<i>Testing Model</i>	2	1,5
	<i>Export Model</i>	1	0,5
TOTAL		20	16

Dari Tabel 3.5 didapatkan jumlah *task*, estimasi waktu penggerjaan, dan waktu penggerjaan sebenarnya dalam mengerjakan *task – task* pada *backlog* pembuatan dan pengujian model dataset.

3.3.5. Sprint 5 Perancangan Antarmuka Aplikasi

Pada *sprint* ini akan mengerjakan bagian dari *product backlog* yaitu perancangan antarmuka aplikasi, *task* yang dikerjakan antara lain.

Tabel 3. 6 *Sprint 5*: Perancangan Antarmuka Aplikasi

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Perancangan antarmuka aplikasi	Antarmuka Aplikasi Android		
	Halaman <i>Onboarding</i>	1	1
	Halaman <i>Splash Screen</i>	1	1

	Halaman <i>Login</i>	1	1
	Halaman <i>Register</i>	1	1
	Halaman Pengisian Data Pengguna	1,5	1
	Halaman <i>Home</i>	1,5	1
	Halaman <i>Event</i> Pameran	3	3
	Halaman Komentar <i>Event</i>	1	1
	Halaman Mengubah Komentar <i>Event</i>	1	1
	Halaman Menghapus Komentar <i>Event</i>	1	1
	Halaman <i>Scan Qrcode Event</i>	2	2
	Halaman Konfirmasi <i>Event</i>	2	2
	Halaman Deteksi Batik	3	3
	Halaman Profil	1	1
	Antarmuka Website		
	Halaman <i>Login</i>	1	1
	Halaman Dashboard	2	2
	Halaman Menampilkan Informasi Motif Batik	2	2
	Halaman Menambah Informasi Motif Batik	2	2
	Halaman Mengedit Informasi Motif Batik	2	2
	Halaman Menampilkan Berita	2	2
	Halaman Menambah Berita	2	1,5
	Halaman Mengedit Berita	2	1,5
	Halaman Menampilkan All <i>Event</i> Pameran	2	2
	Halaman Menampilkan Detail <i>Event</i>	2	2

	Halaman Konfirmasi <i>Event</i>	3	2
	Halaman Menampilkan Data Pengunjung	2	2
	Halaman Menambah Data Pengunjung	2	1,5
	Halaman Mengedit Data Pengunjung	2	1,5
	Halaman Menampilkan Data Penyelenggara	2	2
	Halaman Menambah Data Penyelenggara	2	2
	Halaman Mengedit Data Penyelenggara	2	2
	Halaman Profil	2	2
	Halaman Menambahkan <i>Event</i>	2	2
	Halaman Mengedit <i>Event</i>	2	2
	Halaman Menampilkan <i>MyEvent</i>	2	2
	Halaman Menampilkan <i>QRCode</i>	3	3
	Halaman Menampilkan Buku Tamu	2	2
	Halaman Menambahkan Kehadiran Pengunjung	2	2
	Halaman Melihat Status <i>Event</i>	2	2
TOTAL		72	68

Dari Tabel 3.6 didapatkan jumlah *task*, estimasi waktu penggerjaan, dan waktu penggerjaan sebenarnya dalam mengerjakan *task – task* pada backlog perancangan antarmuka aplikasi.

3.3.6. Sprint 6 Pembuatan dan Pengujian Fitur Aplikasi

Pada *sprint* ini akan mengerjakan bagian dari *product backlog* yaitu pembuatan dan pengujian fitur aplikasi, *task* yang dikerjakan antara lain:

Tabel 3. 7 *Sprint 6: Pembuatan dan Pengujian Fitur Aplikasi*

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Pembuatan dan pengujian fitur aplikasi	Antarmuka Aplikasi Android		
	Pembuatan dan pengujian fitur Halaman <i>Onboarding</i>	5	4
	Pembuatan dan pengujian fitur Halaman <i>Splash Screen</i>	5	4
	Pembuatan dan pengujian fitur Halaman <i>Login</i>	10	8
	Pembuatan dan pengujian fitur Halaman <i>Register</i>	10	9
	Pembuatan dan pengujian fitur Halaman Pengisian Data Pengguna	10	9
	Pembuatan dan pengujian fitur Halaman <i>Home</i>	10	8
	Pembuatan dan pengujian fitur Halaman <i>Event</i> Pameran	15	14
	Pembuatan dan pengujian fitur Halaman Komentar <i>Event</i>	10	9
	Pembuatan dan pengujian fitur Halaman Mengubah Komentar <i>Event</i>	10	9
	Pembuatan dan pengujian fitur Halaman Menghapus Komentar <i>Event</i>	7	6

	Pembuatan dan pengujian fitur Halaman <i>Scan Qrcode Event</i>	15	13
	Pembuatan dan pengujian fitur Halaman Konfirmasi <i>Event</i>	18	15
	Halaman Deteksi Batik	30	27
	Halaman Profil	10	8
Antarmuka Website			
	Pembuatan dan pengujian fitur Halaman <i>Login</i>	10	8
	Pembuatan dan pengujian fitur Halaman Dashboard	15	12
	Pembuatan dan pengujian fitur Halaman Menampilkan Informasi Motif Batik	10	8
	Pembuatan dan pengujian fitur Halaman Menambah Informasi Motif Batik	15	13
	Pembuatan dan pengujian fitur Halaman Mengedit Informasi Motif Batik	15	13
	Pembuatan dan pengujian fitur Halaman Menampilkan Berita	10	9
	Pembuatan dan pengujian fitur Halaman Menambah Berita	15	13
	Pembuatan dan pengujian fitur Halaman Mengedit Berita	10	9
	Pembuatan dan pengujian fitur Halaman Menampilkan <i>All</i> <i>Event</i> Pameran	15	12

	Pembuatan dan pengujian fitur Halaman Menampilkan Detail <i>Event</i>	10	9
	Pembuatan dan pengujian fitur Halaman Konfirmasi <i>Event</i>	15	13
	Pembuatan dan pengujian fitur Halaman Menampilkan Data Pengunjung	10	9
	Pembuatan dan pengujian fitur Halaman Menambah Data Pengunjung	15	13
	Pembuatan dan pengujian fitur Halaman Mengedit Data Pengunjung	15	13
	Halaman Menampilkan Data Penyelenggara	10	9
	Pembuatan dan pengujian fitur Halaman Menambah Data Penyelenggara	10	9
	Pembuatan dan pengujian fitur Halaman Mengedit Data Penyelenggara	10	9
	Pembuatan dan pengujian fitur Halaman Profil	10	9
	Pembuatan dan pengujian fitur Halaman Menambahkan <i>Event</i>	15	13
	Pembuatan dan pengujian fitur Halaman Mengedit <i>Event</i>	15	13
	Pembuatan dan pengujian fitur Halaman Menampilkan <i>MyEvent</i>	10	9

	Pembuatan dan pengujian fitur Halaman Menampilkan <i>QRCode</i>	15	15
	Pembuatan dan pengujian fitur Halaman Menampilkan Buku Tamu	10	9
	Pembuatan dan pengujian fitur Halaman Menambahkan Kehadiran Pengunjung	10	10
	Pembuatan dan pengujian fitur Halaman Melihat Status <i>Event</i>	10	8
TOTAL		470	410

Dari Tabel 3.7 didapatkan jumlah *task*, estimasi waktu penggerjaan, dan waktu penggerjaan sebenarnya dalam mengerjakan *task – task* pada *backlog* pembuatan dan pengujian fitur aplikasi

3.3.7. Sprint 7 Perancangan Uji Kepuasan Pengguna

Pada *sprint* ini akan mengerjakan bagian dari *product backlog* yaitu perancangan uji kepuasan pengguna, *task* yang dikerjakan antara lain:

Tabel 3. 8 *Sprint 7* : Perancangan Uji Kepuasan Pengguna

Product Backlog	Task	Estimasi Waktu (jam)	Waktu yang digunakan (jam)
Perancangan uji kepuasan pengguna	Pembuatan pertanyaan	3	2
	Pembuatan indikator penilaian	2	1
	Pembuatan indikator kepuasan pengguna	2	1
	Eksekusi lapangan	24	22
TOTAL		31	26

Dari Tabel 3.8 didapatkan jumlah *task*, estimasi waktu penggerjaan, dan waktu penggerjaan sebenarnya dalam mengerjakan *task – task* pada *backlog* perancangan uji kepuasan pengguna.

3.4. Pengerjaan Task

Pengerjaan *task* dilakukan secara berurutan sesuai nama urutan *sprint* yang ada.

3.4.1. Sprint 1: Analisis Kebutuhan Perangkat

Pada *sprint* ini akan dibahas mengenai analisis kebutuhan perangkat berupa kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*) yang diperlukan dalam pembuatan tugas akhir aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing*.

1) Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk membangun aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* adalah sebagai berikut:

- a. Laptop
- b. Mouse

2) Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membangun aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* adalah sebagai berikut:

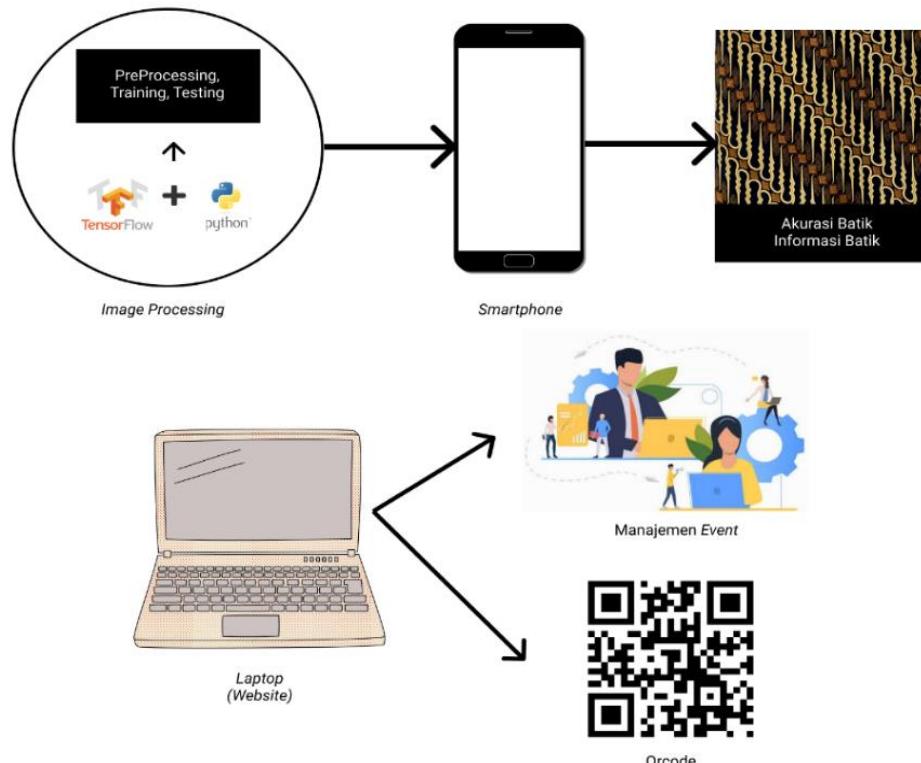
- a. Visual Studio Code
- b. Android Studio
- c. Windows
- d. Xampp

3.4.2. Sprint 2: Analisis Perancangan Sistem

Pada *sprint* ini membahas perancangan diagram sistem seperti gambaran umum sistem, *block diagram*, *usecase diagram*, *activity diagram*, dan ERD.

1) Gambaran Umum Sistem

Gambaran umum sistem digunakan untuk mengetahui bagaimana aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* bekerja secara umum. Penjelasan mengenai aplikasi akan ditujukan pada Gambar 3.2.



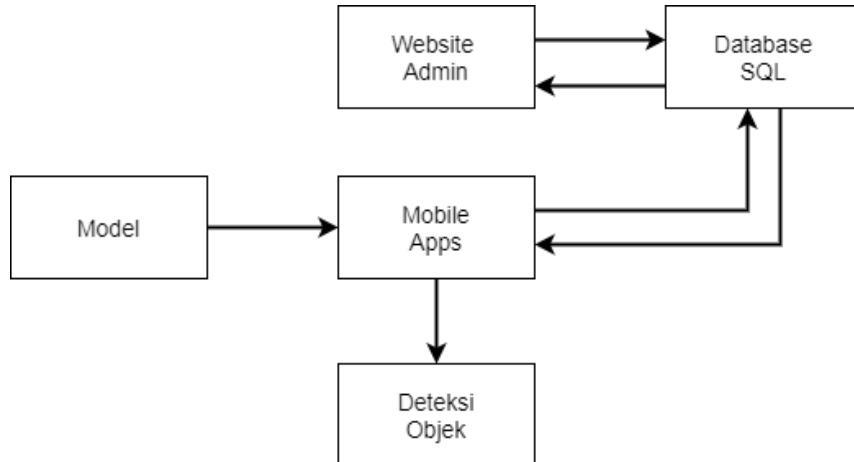
Gambar 3. 2 Gambaran Umum Sistem

Pada Gambar 3.2 dapat diketahui bahwa tahapan *image processing* meliputi proses *preprocessing*, *training*, dan *testing* yang dilakukan menggunakan tensorflow dan python. Hasil dari proses *image processing* tersebut akan diunduh berupa file .tflite untuk dipasang pada android. Android digunakan untuk mengenali motif batik beserta informasi motif batik. Pada website digunakan untuk manajemen *event* pameran dan menampilkan *qrcode* untuk buku tamu pengunjung.

2) Block Diagram

Block Diagram digunakan untuk mengetahui gambaran umum tentang bagaimana aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android

dan *image processing* ini bekerja dalam bentuk blok – blok yang saling terhubung seperti pada Gambar 3.3.



Gambar 3. 3 *Block Diagram Sistem*

Pada Gambar 3.3 dapat diketahui bahwa model yang sudah dihasilkan akan disimpan dan diproses kedalam sebuah aplikasi *mobile* yang nantinya akan berisikan informasi mengenai motif batik yang dideteksi. Aplikasi *mobile* ini memiliki fungsi *capture* untuk mendeteksi objek yaitu motif batik. Selain itu aplikasi *mobile* ini digunakan untuk memberikan respon dari pengunjung pameran terhadap *event* pameran batik yang diadakan. Untuk menyimpan model ini digunakan database SQL dan database ini terintegrasi dengan website untuk Admin sebagai *monitoring event* dari pameran batik.

Berikut ini merupakan *block diagram* pembuatan model dalam aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* yang terdapat pada Gambar 3.4.



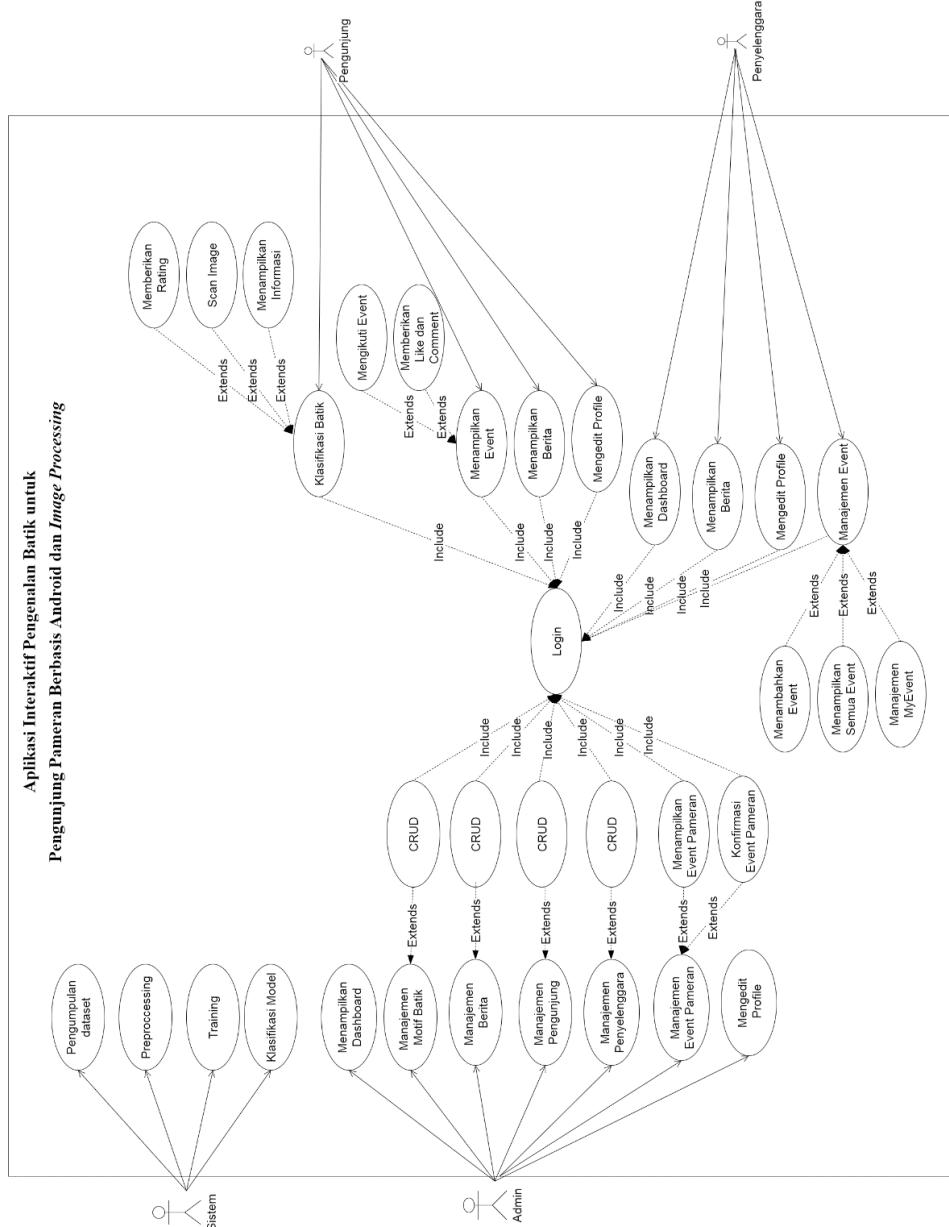
Gambar 3. 4 *Block Diagram Pembuatan Model*

Pada Gambar 3.4 proses pembuatan model dilakukan dengan mencari dataset batik. Dataset akan dirotasi, di *resize*, diubah ke warna keabuan, kemudian dilabeli sesuai dengan kelas yang sudah ditentukan, kemudian data akan di *split* menjadi data *train* dan data *test* agar memiliki akurasi yang baik. Selanjutnya data yang sudah di *split*

akan di *training* untuk kemudian akan dilakukan proses *testing* dataset. *Testing* dataset ini dilakukan menggunakan metode CNN MobileNet yang nantinya akan menghasilkan sebuah model.

3) Use Case Diagram

Use Case Diagram merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem. *Use case diagram* aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* akan dipaparkan pada Gambar 3.5.



Gambar 3. 5 *Use Case Diagram* Sistem

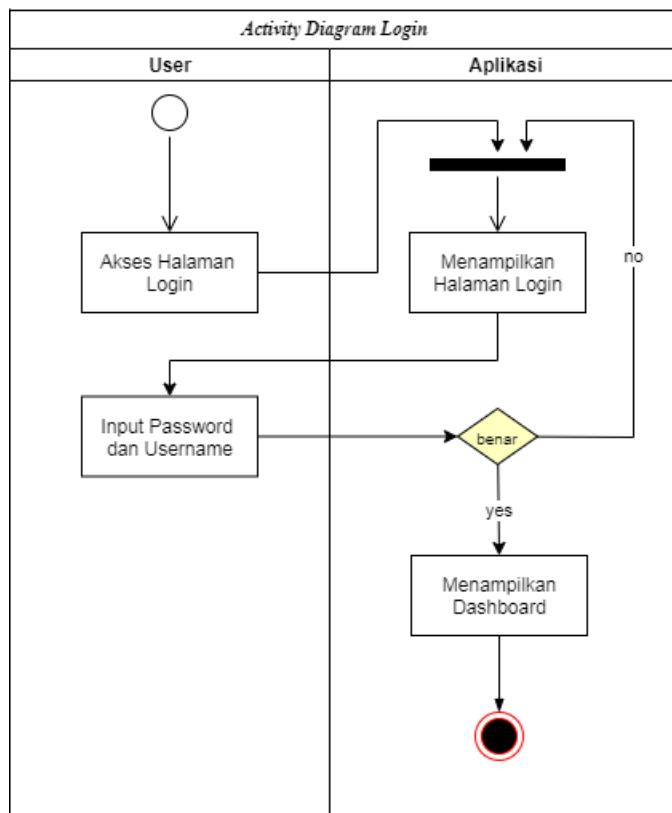
Berdasarkan Gambar 3.5 terdapat empat aktor yaitu sistem, admin, pengunjung, dan penyelenggara dimana aktor merupakan seseorang atau sesuatu yang berinteraksi dengan sistem. Sistem memproses dan mengolah dataset yang dimulai dari pencarian dataset, *preprocessing*, *training* dan *testing*. Sedangkan admin dapat menampilkan dashboard, melakukan manajemen motif batik, manajemen berita, manajemen pengunjung, manajemen penyelenggara, manajemen *event* pameran, dan mengedit profil. Pengunjung dapat melihat hasil klasifikasi batik dari *scan* yang dilakukan menggunakan kamera *handphone* dan memberikan *rating* terhadap batik yang dideteksi, selain itu pengunjung dapat melihat berita, mengikuti *event* pameran dengan melakukan *scan barcode*, memberikan *like* dan komentar terhadap *event* dan mengedit profil. Sedangkan penyelenggara dapat menampilkan dashboard, memanajemen *event* seperti melihat pengunjung, melihat berita, dan *edit* profil. Admin, pengunjung, dan penyelenggara wajib melakukan *log in* dengan *username* dan *password* yang telah ditentukan sebelum melakukan proses – proses tersebut.

4) *Activity Diagram*

Activity diagram merupakan sebuah diagram yang digunakan untuk menggambarkan alur dari aktifitas – aktifitas yang ada dalam sistem ini. Berikut beberapa diagram yang terdapat pada sistem ini.

a) *Activity Diagram Login*

Activity diagram login pada admin, pengguna, dan penyelenggara akan ditampilkan pada Gambar 3.6

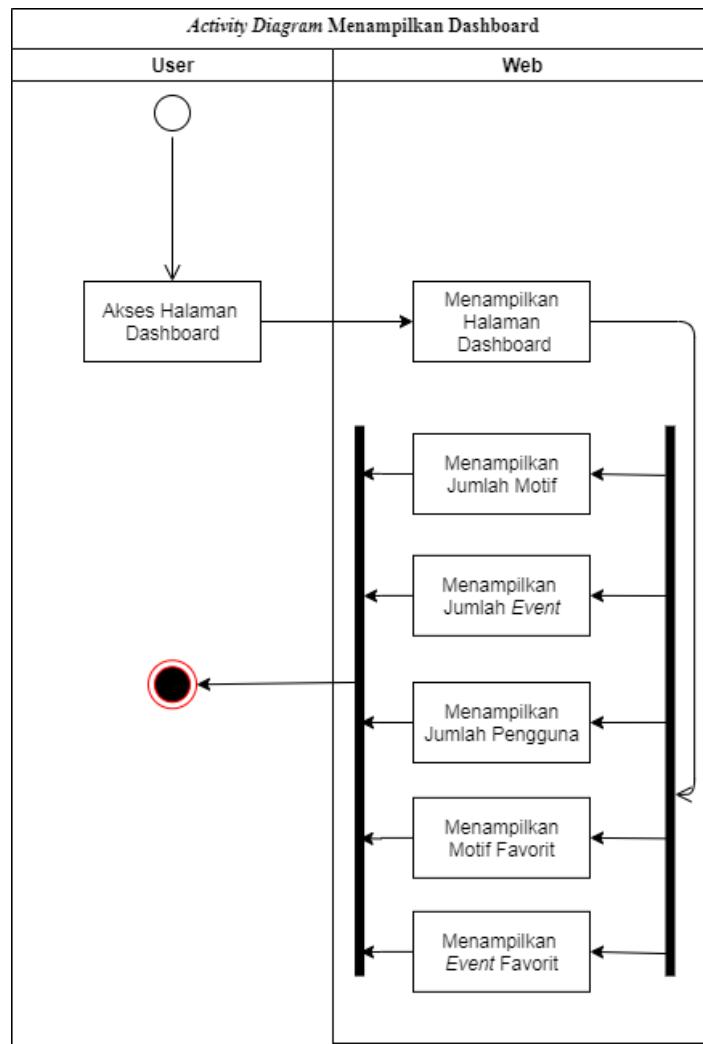


Gambar 3. 6 Activity Diagram Login

Berdasarkan Gambar 3.6 pengguna dapat melakukan *login* dengan mengakses halaman *login* terlebih dahulu. Setelah itu pengguna dapat memasukkan *username* dan *password* yang dimilikinya. Jika proses *login* telah berhasil, sistem akan menampilkan dashboard awal pengguna. Namun jika proses *login* gagal, pengguna akan tetap berada di halaman *login*.

b) Activity Diagram Menampilkan Dashboard

Activity diagram menampilkan dashboard pada admin dan pengguna akan ditampilkan pada Gambar 3.7

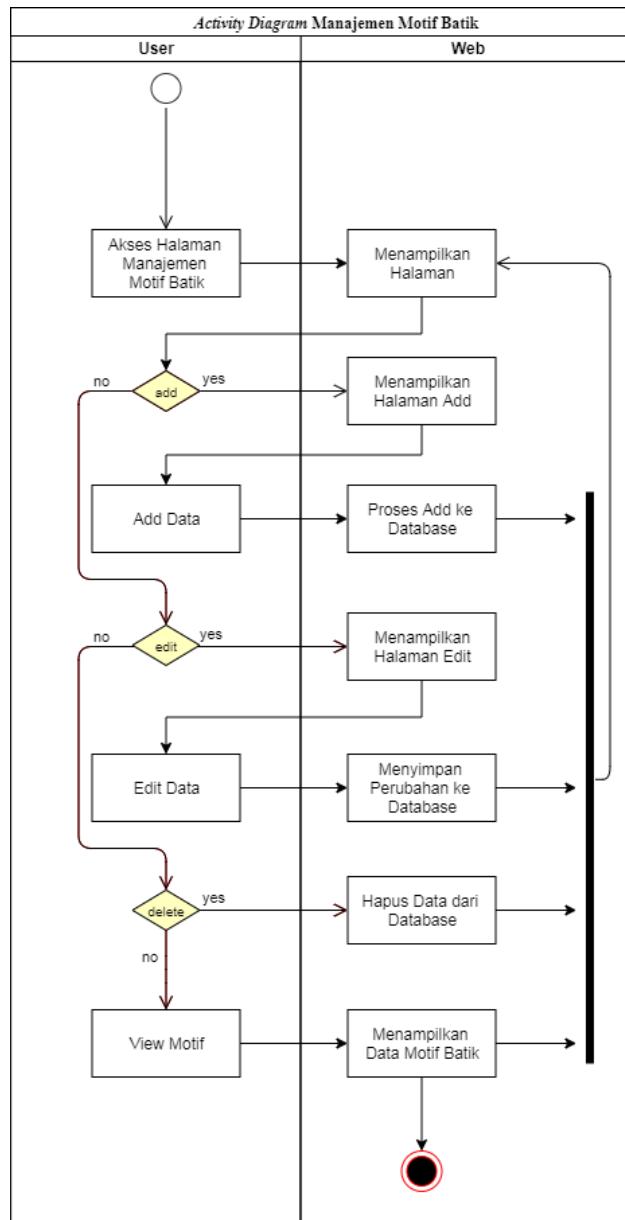


Gambar 3. 7 *Activity Diagram Menampilkan Dashboard*

Berdasarkan Gambar 3.7 admin dapat melihat kalkulasi jumlah motif, jumlah *event*, jumlah pengguna, motif terfavorit dan *event* terfavorit. Kalkulasi pada dashboard akan bertambah seiring adanya penambahan pada motif batik, *event* pameran, pengguna, hasil *rating* batik yang diberikan pengguna, dan hasil *like* yang diberikan pengguna terhadap *event* batik.

c) Activity Diagram Manajemen Motif Batik

Activity diagram manajemen motif batik akan ditampilkan pada Gambar 3.8.

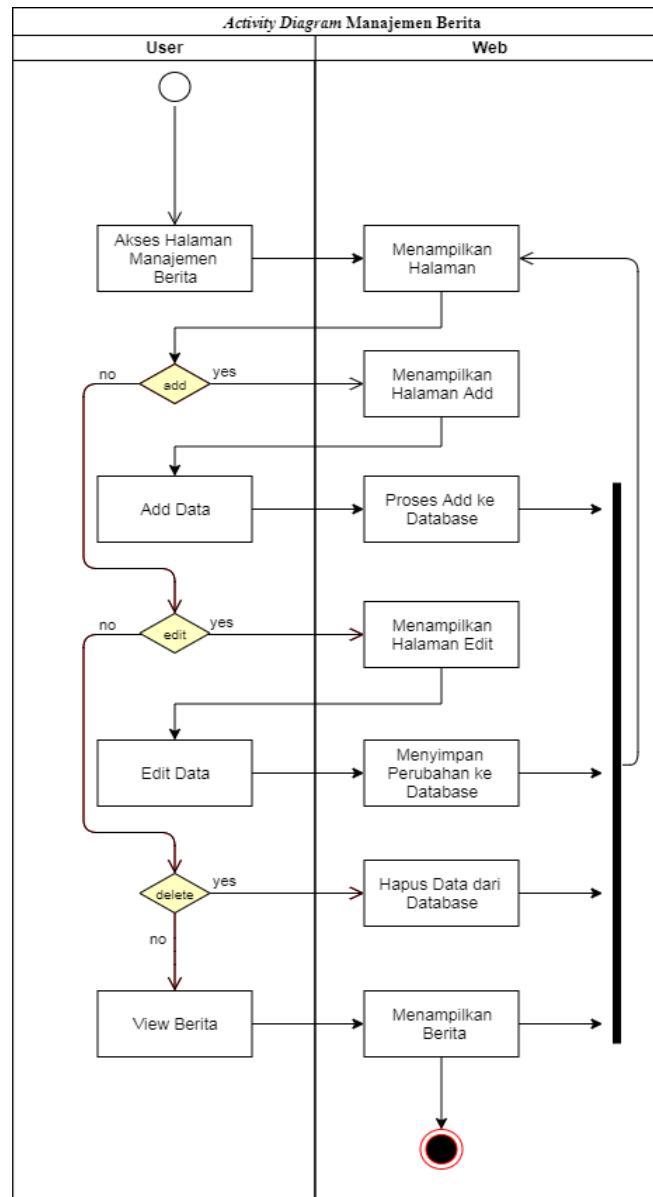


Gambar 3. 8 Activity Diagram Motif Batik

Berdasarkan Gambar 3.8 ketika admin melakukan manajemen motif batik dengan mengakses halaman manajemen motif batik terlebih dahulu. Setelah ditampilkan halaman motif batik, admin dapat menambahkan, mengedit, dan menghapus motif batik yang akan disimpan oleh database. Jika admin tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman motif batik.

d) Activity Diagram Manajemen Berita

Activity diagram manajemen berita akan ditampilkan pada Gambar 3.9.

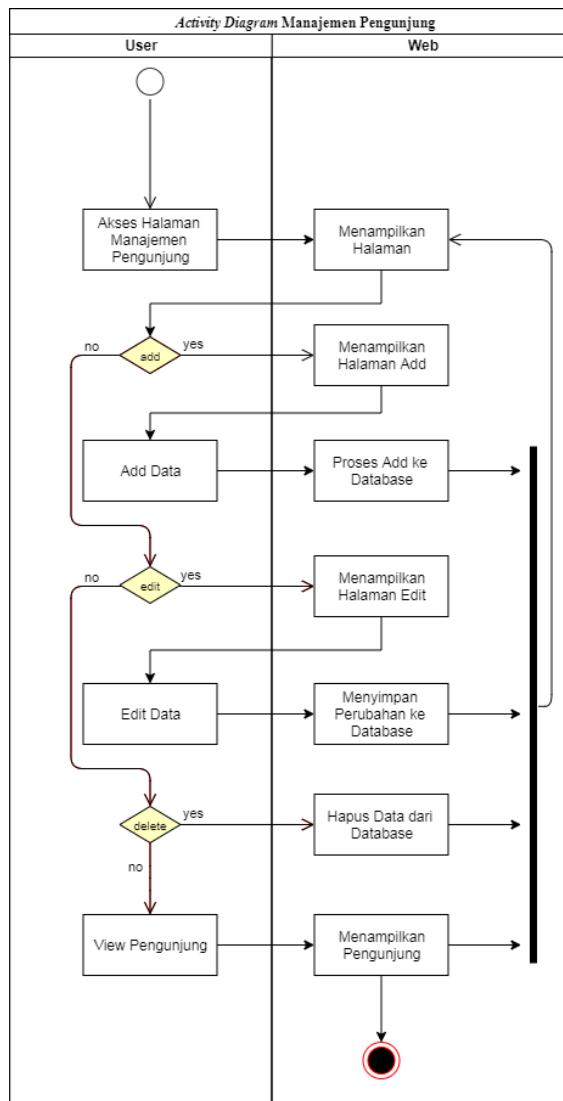


Gambar 3. 9 Activity Diagram Manajemen Berita

Berdasarkan Gambar 3.9 admin melakukan manajemen berita dengan mengakses halaman manajemen berita terlebih dahulu. Setelah ditampilkan halaman manajemen berita, admin dapat menambahkan, mengedit, dan menghapus berita yang akan disimpan oleh database. Jika admin tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman manajemen berita.

e) Activity Diagram Manajemen Pengunjung

Activity diagram manajemen pengunjung akan ditampilkan pada Gambar 3.10.

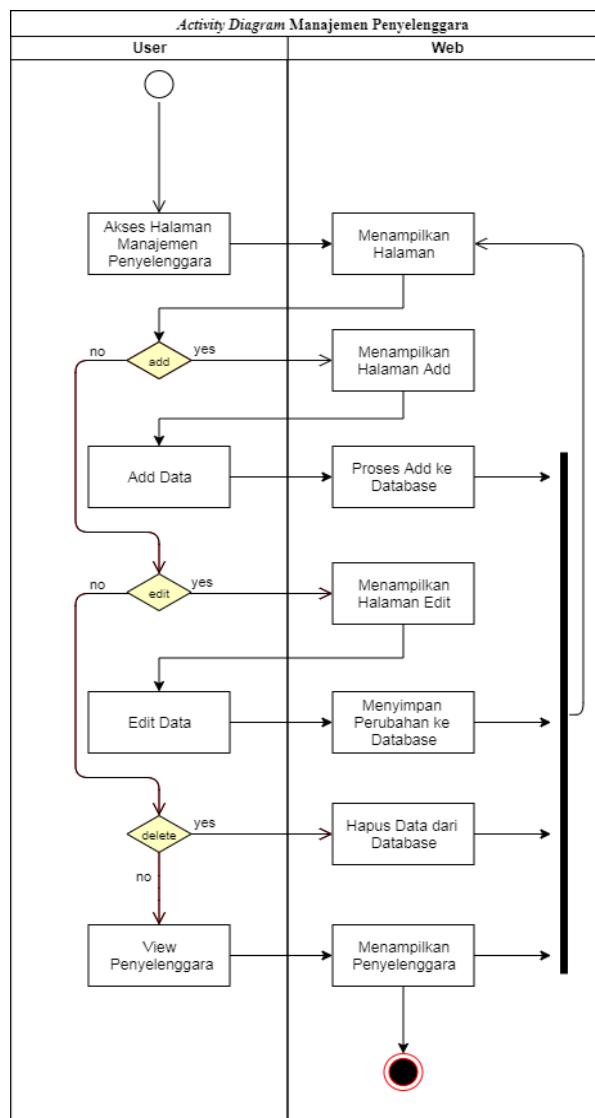


Gambar 3. 10 *Activity Diagram* Manajemen Pengunjung

Berdasarkan Gambar 3.10 admin melakukan manajemen pengunjung dengan mengakses halaman manajemen pengunjung terlebih dahulu. Setelah ditampilkan halaman manajemen pengunjung, admin dapat menambahkan, mengedit, dan menghapus pengunjung yang akan disimpan oleh database. Jika admin tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman manajemen pengunjung.

f) Activity Diagram Manajemen Penyelenggara

Activity diagram manajemen manajemen penyelenggara akan ditampilkan pada Gambar 3.11.

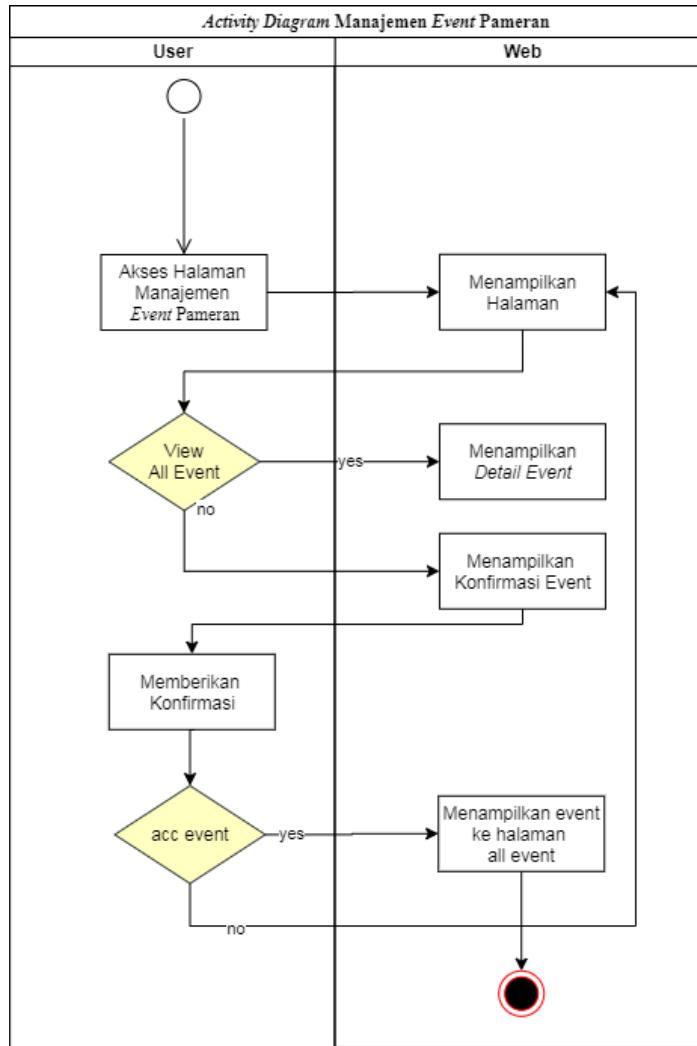


Gambar 3. 11 *Activity Diagram* Manajemen Penyelenggara

Berdasarkan Gambar 3.11 admin melakukan manajemen penyelenggara dengan mengakses halaman manajemen penyelenggara terlebih dahulu. Setelah ditampilkan halaman manajemen penyelenggara, admin dapat menambahkan, mengedit, dan menghapus penyelenggara yang akan disimpan oleh database. Jika admin tidak melakukan penambahan atau penghapusan maka akan ditampilkan kembali halaman manajemen penyelenggara.

g) Activity Diagram Manajemen Event Pameran

Activity diagram manajemen event pameran akan ditampilkan pada Gambar 3.12.

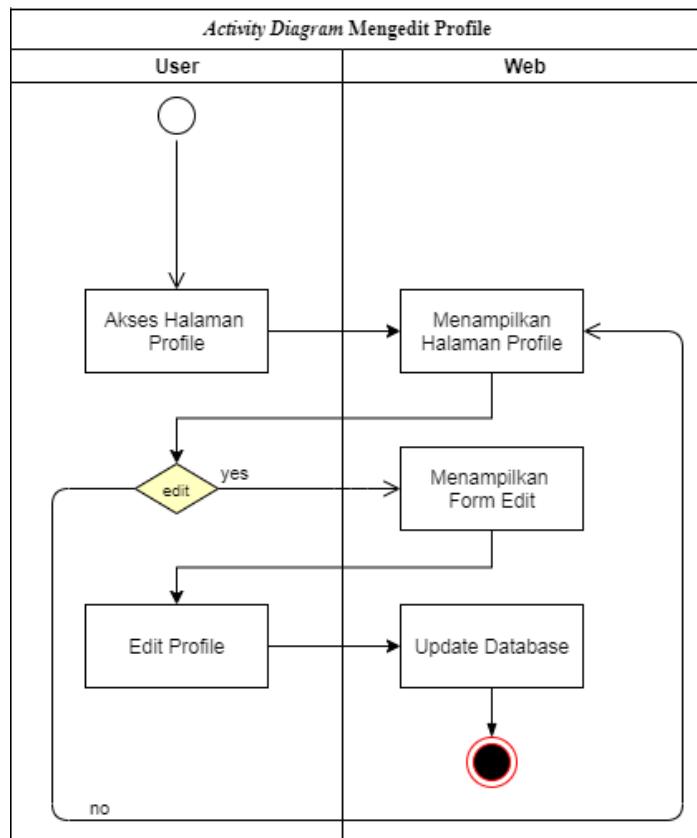


Gambar 3. 12 Activity Diagram Manajemen Event Pameran

Berdasarkan Gambar 3.12 admin dapat melakukan manajemen *event* pameran dengan mengakses halaman manajemen *event* pameran, setelah berhasil maka akan ditampilkan halaman manajemen *event* pameran. Jika admin memilih *view all event*, maka akan ditampilkan detail dari *event*, apabila tidak maka akan diarahkan ke halaman konfirmasi *event* yang sebelumnya didaftarkan oleh penyelenggara. Apabila admin menyetujui *event* untuk ditampilkan maka *event* yang didaftarkan akan ditampilkan pada halaman *all event*, jika tidak maka akan kembali ke halaman manajemen *event* pameran.

h) Activity Diagram Mengedit Profil

Activity diagram mengedit profil akan ditampilkan pada akan ditampilkan pada Gambar 3.13.

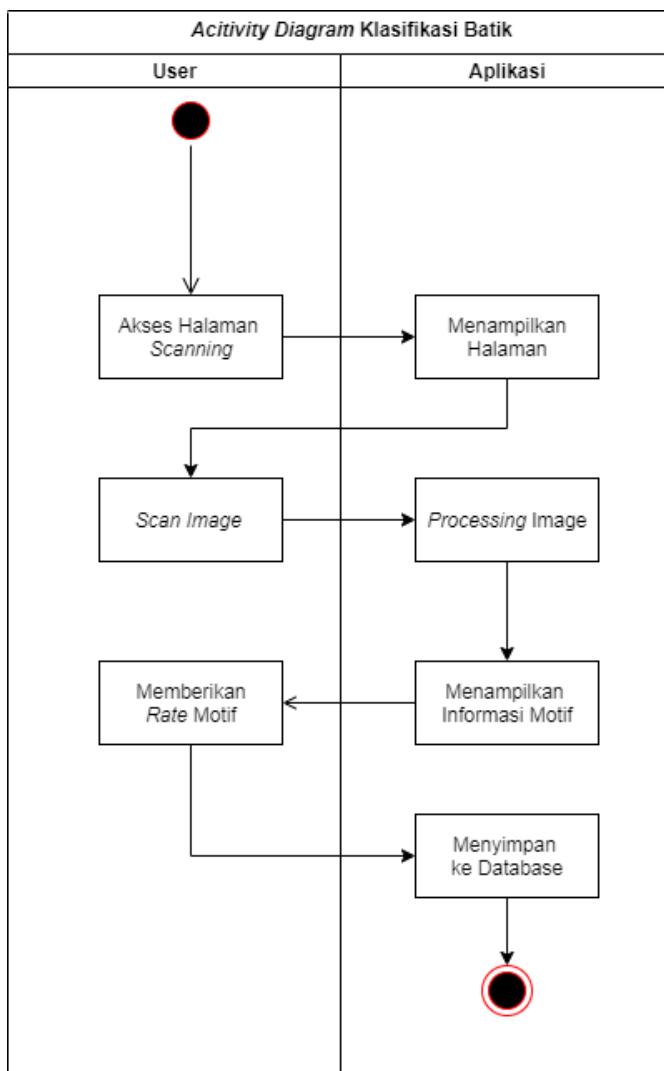


Gambar 3. 13 *Activity Diagram* Mengedit Profil

Berdasarkan Gambar 3.13 admin dapat mengedit profil dengan mengakses halaman profil terlebih dahulu. Setelah ditampilkan halaman profil apabila admin akan mengubah informasi profil maka akan disimpan ke database. Jika tidak maka akan ditampilkan kembali halaman profil.

i) *Activity Diagram Klasifikasi Batik*

Activity diagram klasifikasi batik akan ditampilkan pada akan ditampilkan pada Gambar 3.14.

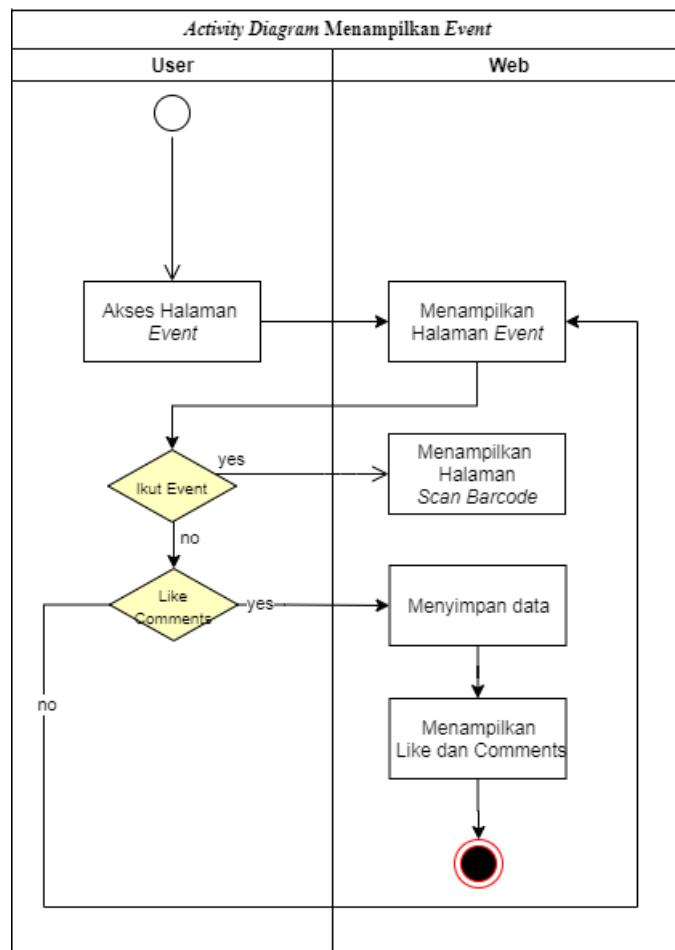


Gambar 3. 14 *Activity Diagram Klasifikasi Batik*

Berdasarkan Gambar 3.14 pengguna dapat mengklasifikasi motif batik dengan mengakses halaman *scanning*. Setelah mengakses halaman *scanning* maka akan ditampilkan halaman *scanning*. Pengguna melakukan *scan* terhadap batik dengan mengarahkan kamera ke batik, selanjutnya aplikasi akan memproses *image* dan menampilkan informasi mengenai batik yang sesuai dengan hasil deteksi. Pengguna akan diarahkan untuk memberikan *like* terhadap tersebut dan akan disimpan pada database.

j) Activity Diagram Menampilkan Event

Activity diagram menampilkan *event* akan ditampilkan pada akan ditampilkan pada Gambar 3.15.

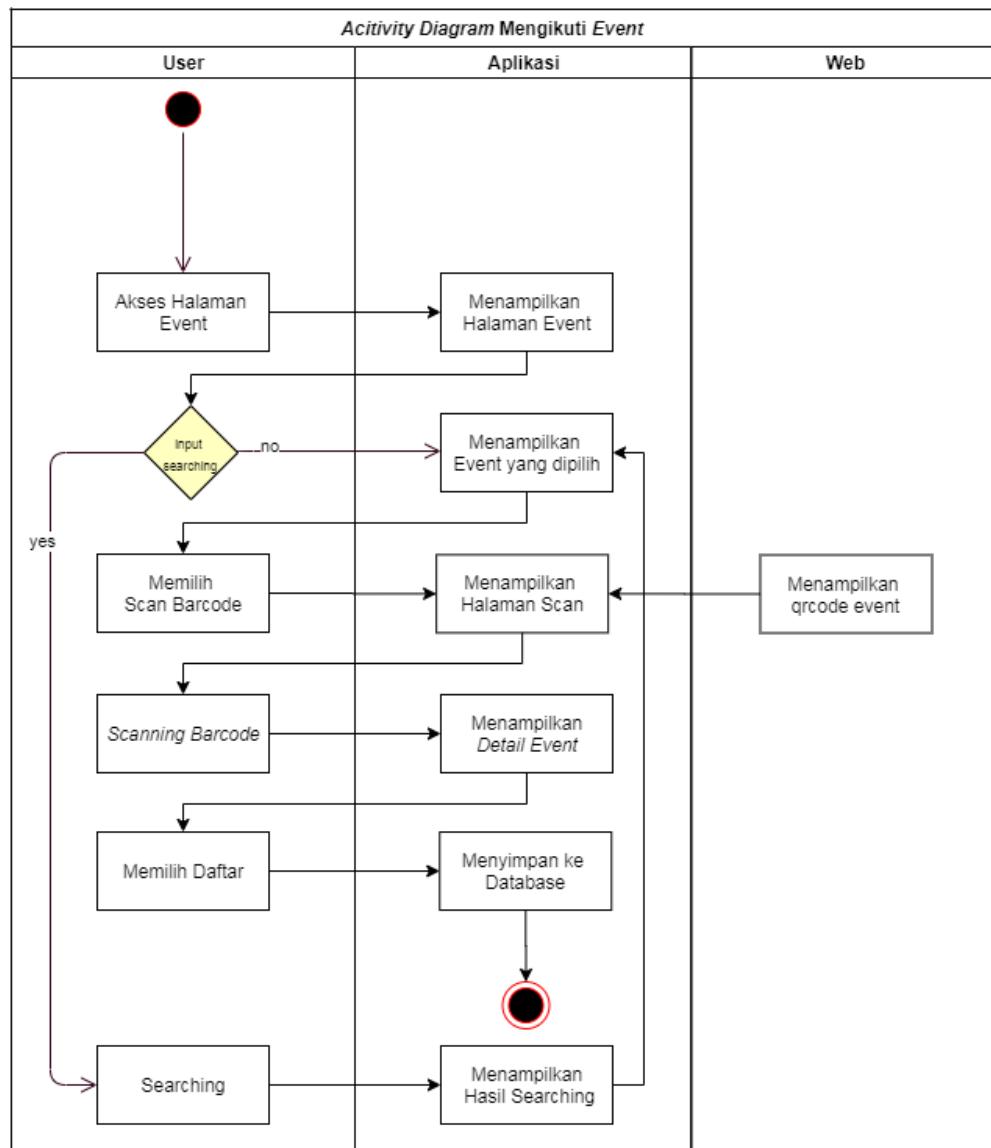


Gambar 3. 15 *Activity Diagram* Menampilkan *Event*

Berdasarkan Gambar 3.15 pengguna dapat menampilkan *event* dengan mengakses halaman *event* terlebih dahulu. Setelah menampilkan halaman *event* pengguna dapat mengikuti *event* dengan mengakses halaman *scan barcode* kemudian *scan barcode*. Apabila berhasil maka akan ada konfirmasi *event*. Jika pengguna memberikan *like* dan komentar maka akan disimpan ke database, jika tidak maka akan diarahkan kembali pada halaman *event*.

k) Activity Diagram Mengikuti Event

Activity diagram mengikuti *event* akan ditampilkan pada akan ditampilkan pada Gambar 3.16.



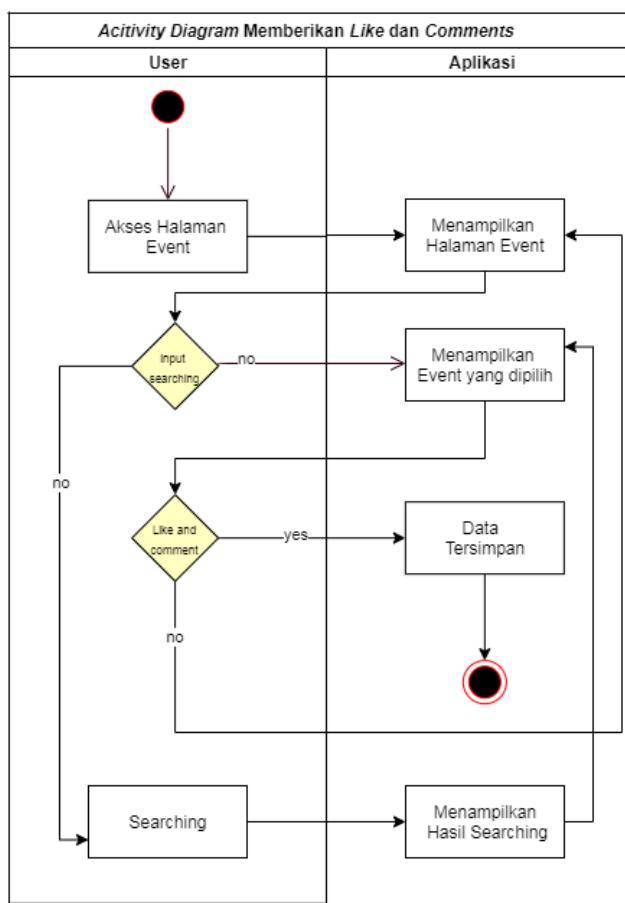
Gambar 3. 16 Activity Diagram Mengikuti Event

Berdasarkan Gambar 3.16 pengguna dapat mengikuti *event* dengan mengakses halaman *event* terlebih dahulu. Jika pengguna akan melakukan *searching event* maka sistem akan melakukan *searching* dengan data yang dicari, jika berhasil maka akan ditampilkan hasil dari pencarian. Jika pengguna tidak melakukan pencarian maka akan *scan barcode*. Untuk mengikuti *event* pengguna melakukan *scan*

barcode, jika berhasil sistem akan menampilkan informasi *event* yang sesuai dengan *barcode* yang *discan*. Jika berhasil pengguna memilih tombol daftar untuk konfirmasi dan akan disimpan ke database.

1) Activity Diagram Memberikan Like dan Comment

Activity diagram memberikan *like* dan *comment* akan ditampilkan pada akan ditampilkan pada Gambar 3.17.

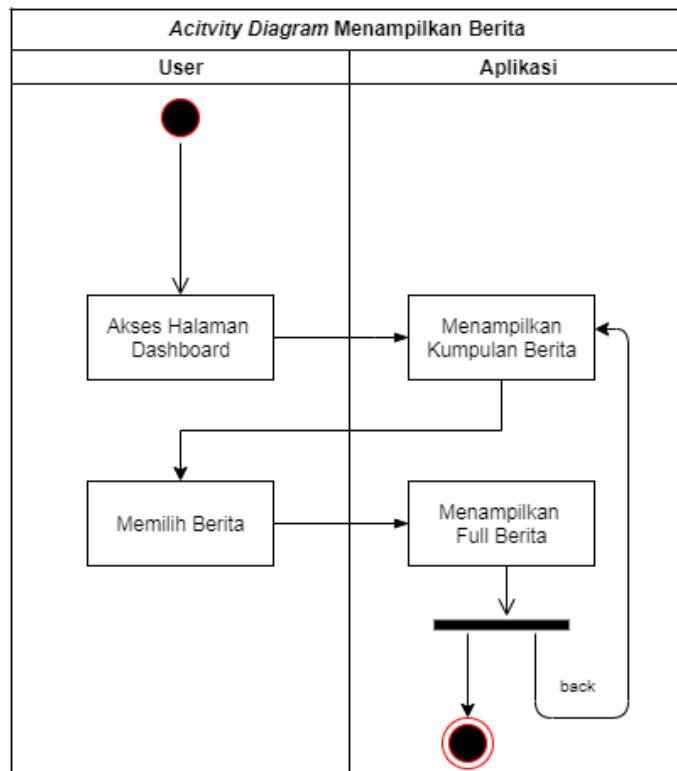


Gambar 3. 17 Activity Diagram Memberikan Like dan Comment

Berdasarkan Gambar 3.17 pengguna dapat mengikuti *event* dengan mengakses halaman *event* terlebih dahulu. Jika pengguna akan melakukan *searching event* maka sistem akan melakukan *searching* dengan data yang dicari, jika berhasil maka akan ditampilkan hasil dari pencarian. Apabila pengguna memberikan *like* dan komentar maka data akan disimpan, jika tidak maka akan diarahkan ke halaman *event*.

m) Activity Diagram Menampilkan Berita

Activity diagram menampilkan berita akan ditampilkan pada akan ditampilkan pada Gambar 3.18.

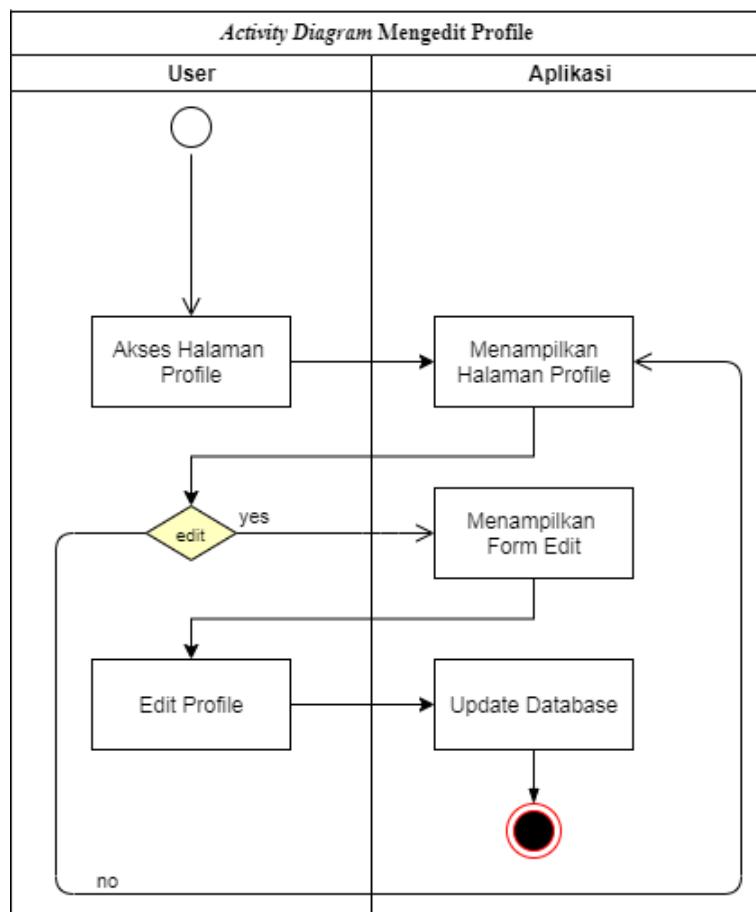


Gambar 3. 18 *Activity Diagram* Menampilkan Berita

Berdasarkan Gambar 3.18 pengguna dapat menampilkan berita dengan mengakses halaman berita terlebih dahulu. Setelah menampilkan halaman berita pengguna dapat memilih berita yang akan dibaca, kemudian aplikasi akan menampilkan informasi lengkap mengenai berita yang dipilih. Jika pengguna tidak ingin menampilkan informasi lengkap berita maka akan kembali ke halaman berita.

n) *Activity Diagram Mengedit Profil*

Activity diagram mengedit profil akan ditampilkan pada akan ditampilkan pada Gambar 3.19.

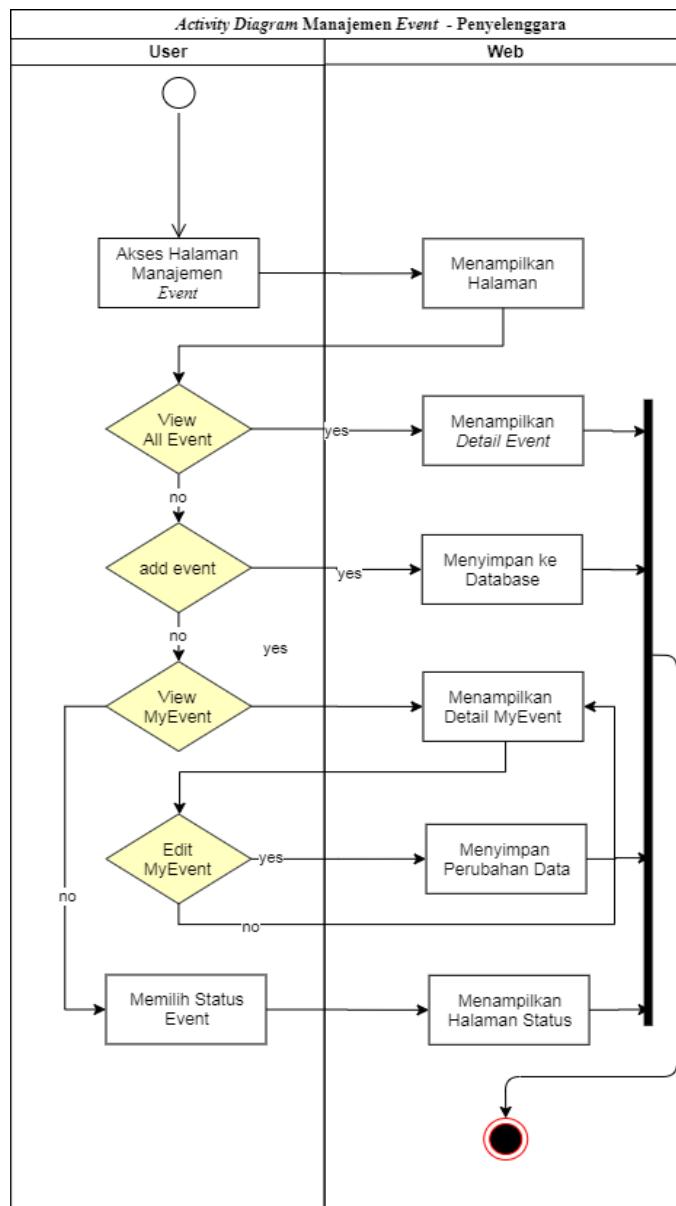


Gambar 3. 19 *Activity Diagram* Mengedit Profil

Berdasarkan Gambar 3.19 pengguna dapat mengedit profil dengan mengakses halaman profil terlebih dahulu. Setelah ditampilkan halaman profil apabila pengguna akan mengubah informasi profil maka akan disimpan ke database. Jika tidak maka akan ditampilkan kembali halaman profil.

o) *Activity Diagram* Manajemen *Event* Penyelenggara

Activity diagram manajemen *event* penyelenggara akan ditampilkan pada akan ditampilkan pada Gambar 3.20.



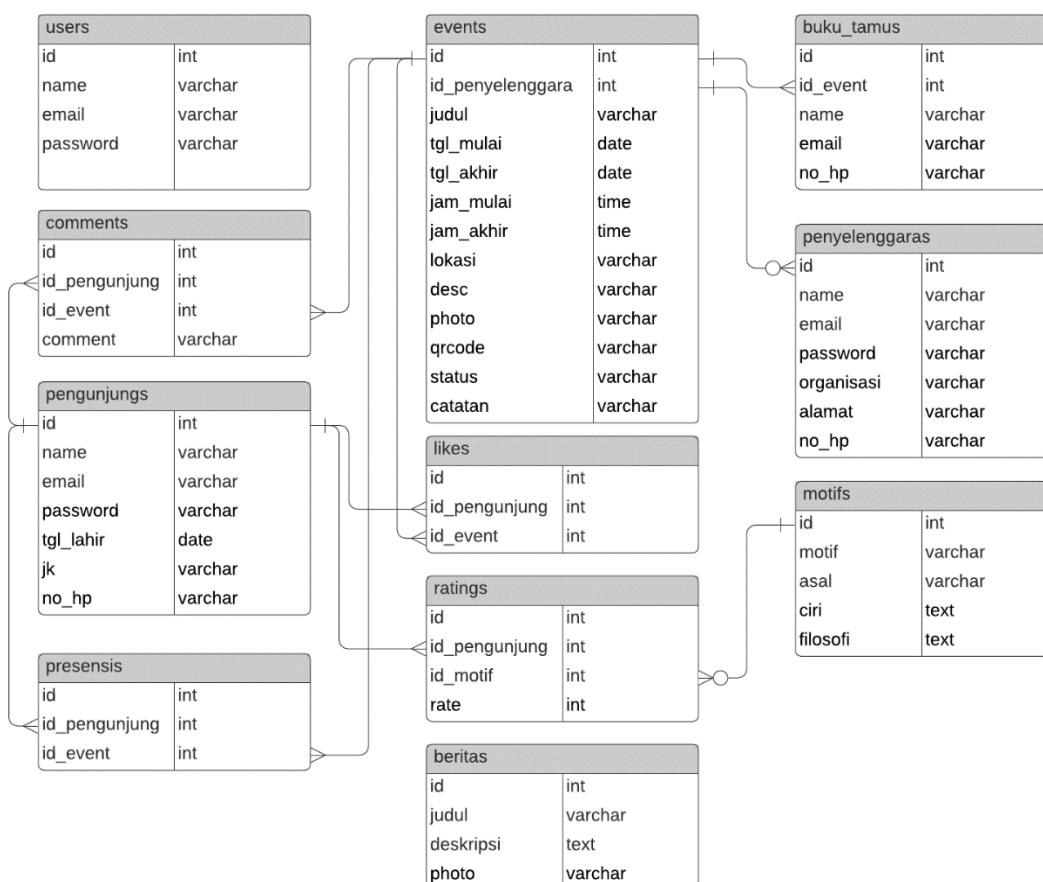
Gambar 3. 20 *Activity Diagram* Manajemen *Event* Penyelenggara

Berdasarkan Gambar 3.20 penyelenggara dapat melakukan manajemen *event* dengan terlebih dahulu mengakses halaman manajemen *event*. Setelah berhasil maka akan ditampilkan halaman manajemen *event*. Jika penyelenggara akan melihat keseluruhan *event* maka akan ditampilkan detail *event* apabila memilih

halaman status *event* maka akan ditampilkan halaman status *event* apakah berhasil atau gagal. Apabila berhasil maka sudah disetujui oleh admin, dan apabila gagal *event* tidak disetujui admin. Penyelenggara dapat menambahkan *event* lagi dan akan disimpan ke database. Apabila penyelenggara ingin melihat dari *eventnya* sendiri, kemudian memilih halaman *MyEvent* maka akan ditampilkan detail *event*-nya sendiri. Penyelenggara dapat melakukan pengubahan data pada *event* maka akan disimpan ke dalam database. Jika tidak maka akan dialihkan kembali ke halaman *MyEvent*.

5) ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* ditampilkan pada Gambar 3.21.



Gambar 3. 21 ERD Sistem Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan *Image Processing*

Dalam Gambar 3.21 ERD aplikasi interaktif pengenalan batik memiliki sebelas buah entitas yaitu users, events, comments, beritas, motifs, likes, presesis, ratings, buku_tamus, penyelenggaras, dan pengunjungs. Entitas pengunjungs memiliki relasi *one-to-many* terhadap entitas presensis, ratings, likes, dan comments . Entitas motifs memiliki relasi *one-to-one* terhadap entitas ratings. Entitas *events* memiliki relasi *one-to-many* terhadap entitas likes, presensis, penyelenggaras, buku_tamus, dan comments. Sedangkan entitas penyelenggaras memiliki relasi *one-to-one* terhadap entitas *events*.

3.4.3. Sprint 3: Perancangan Model Dataset

Pada *sprint* ini akan dibahas mengenai perancangan model data yang diawali dengan pengambilan citra motif batik, kemudian dilakukan tahap pengolahan citra digital yang meliputi pra-pengolahan, *training* untuk menghasilkan nilai ciri yang terkandung dari citra sebagai vektor masukan. Kemudian melakukan pengujian atau *testing* dengan data yang berbeda. Secara garis besar proses yang dilakukan dapat dilihat pada Gambar 3.22.



Gambar 3. 22 Diagram Blok Perancangan dan Pembuatan Model Dataset

Dalam tahap pengumpulan dataset batik, hal yang dilakukan adalah mencari gambar batik yang diperoleh melalui *search engine* google. Dataset batik yang didapatkan kemudian dikelompokkan menjadi 12 kelas dengan satu kelas motif batik terdapat 600 gambar, dimana keseluruhan total dataset batik yaitu 7.200 gambar yaitu batik Ceplok, Jlamprang, Kawung, Lasem, Mega Mendung, Parang, Sekar Jagad, Semen, Sidomukti, Tambal, Truntum, dan Udan Liris. Citra batik yang diambil menggunakan tipe gambar jpg atau png. Gambar 3.23 menampilkan contoh dari dataset yang digunakan.



Gambar 3. 23 Data Batik

1) Perancangan Implementasi Model

Pada tahapan perancangan implementasi model ini, tahap yang dilakukan yaitu data input, *pre-processing* data, *training* model, dan *testing* model. Dataset sebagai data input yang akan dibagi menjadi data *training* dan data *testing*, setelah data input masuk akan dilakukan *pre-processing* data. Dan pada proses *training* model dilakukan pendekatan dengan *transfer learning* yang berfungsi untuk melatih data yang tidak terlalu banyak sehingga akan lebih efektif jika digunakan *pre-trained* model. Proses implementasi model menggunakan *framework* tensorflow dan dilatih dengan arsitektur CNN MobileNet. Kemudian hasil dari proses tersebut akan berupa model.tflite yang akan diimplementasikan pada *device android*.

Tahap – tahap yang dilakukan pada implementasi model sebagai berikut.

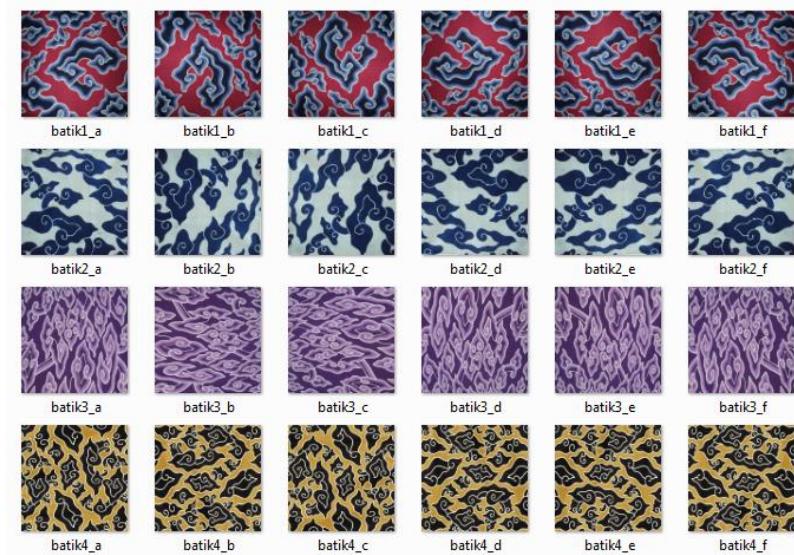
a) Pre-processing Data

Tahap *pre-processing* merupakan proses awal yang dilakukan untuk mengolah citra batik agar proses *training* berlangsung seacara optimal. Pada tahap *pre-processing* ini dilakukan *rotate*, *flipping*, *resize*, *greyscale*, dan *labelling* pada gambar. Tahapan *pre-processing* ditunjukkan pada Gambar 3.24.



Gambar 3. 24 Tahapan Pre-processing Citra Batik

Tahap awal citra batik asli yang didapatkan kemudian diubah menjadi ukuran 500×500 pixel dan dilakukan rotasi dengan sudut 90° , 180° , 270° serta *flipping* horizontal dan *flipping* vertikal. Contoh citra batik yang dirotasi dan *diflipping* ditunjukkan pada Gambar 3.25.



Gambar 3. 25 Data Batik yang Dirotasi dan *Diflipping*

Pada tahap *pre-processing* ini dilakukan *resize* kembali oleh *package* yang sudah disediakan tensorflow sehingga menjadi 224×224 pixel. Citra asli yang diambil masih dalam RGB (*Red*, *Green*, *Blue*) dan diubah ke dalam citra yang memiliki warna dalam *grayscale* (keabuan) agar bisa diproses dengan mudah pada proses *training*. Kemudian dilakukan pelabelan pada dataset sesuai dengan kelas dari masing – masing motif batik.

b) Training

Pada tahapan *training*, dataset citra batik yang digunakan sebanyak 7.200 sampel, dibagi menjadi 12 dataset kelas batik. Pada setiap dataset kelasnya ada sebanyak 600 sampel citra batik. Kemudian untuk data *training* diambil sebanyak 6480 gambar (90%) dan data *testing* 720 gambar (10%). Pada tahap ini model yang dibangun menggunakan arsitektur MobileNet yang memiliki layer-layer konvolusi dengan input awal 224×224 dan menggunakan *feature learning* yang berfungsi untuk mengubah suatu input menjadi fitur berdasarkan citra dari input tersebut yang berbentuk angka dalam vektor. Pada *feature learning* terdapat *pooling layer* yang

digunakan untuk mengurangi dimensi citra. Selain *feature learning* terdapat layer *fully connected* dan *softmax* yang merupakan layer untuk klasifikasi yang berguna untuk mengklasifikasikan tiap neuron yang telah diekstraksi sebelumnya.

c) *Testing Model*

Setelah proses *training* model, tahap selanjutnya yaitu proses *testing* model pada data *testing*. Hasil dari *testing* divisualisasikan menggunakan tabel untuk mengukur performa dari model. Model dengan tingkat akurasi yang tinggi akan digunakan dan diubah dalam bentuk .tflite.

3.4.4. Sprint 4: Pembuatan dan Pengujian Model Dataset

Pada *sprint* ini membahas tentang pembuatan dan pengujian terhadap model *dataset* yang digunakan dalam proses deteksi motif batik.

1) Import Package dan Library

Berikut hasil pembuatan dan pengujian *import package* dan *library model* :

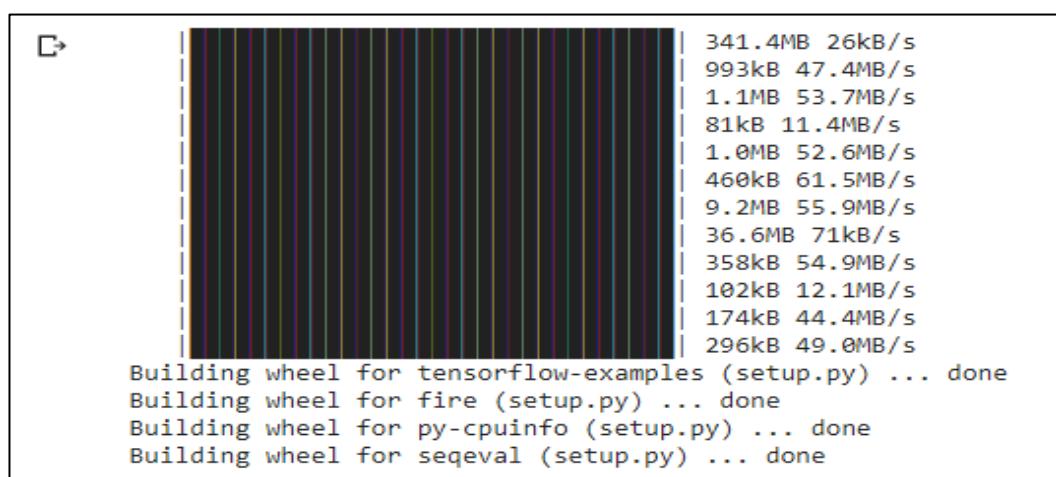
a) Kode Program

Keterangan Kode Program sebagai berikut:

(Lampiran 5 Kode Pengujian Import Package dan Library)

b) Pengujian fungsionalitas

Pengujian fungsionalitas dilakukan berdasarkan hasil menjalankan kode program pada Google Colab. Hasil pengujian sebagai berikut:



```

341.4MB 26kB/s
993kB 47.4MB/s
1.1MB 53.7MB/s
81kB 11.4MB/s
1.0MB 52.6MB/s
460kB 61.5MB/s
9.2MB 55.9MB/s
36.6MB 71kB/s
358kB 54.9MB/s
102kB 12.1MB/s
174kB 44.4MB/s
296kB 49.0MB/s

Building wheel for tensorflow-examples (setup.py) ... done
Building wheel for fire (setup.py) ... done
Building wheel for py-cpuinfo (setup.py) ... done
Building wheel for seqeval (setup.py) ... done

```

Gambar 3. 26 Hasil *Import Package Model Maker Tensorflow*

Pada Gambar 3.26 menunjukkan bahwa *package* dari *tensorflow maker* sudah berhasil dimasukkan kedalam program.

Tabel 3. 9 Pengujian Fungsionalitas *Import Package* dan *Library Model*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Memasukkan <i>Package</i> dan <i>Library</i>	Agar dapat menggunakan fungsi dari <i>package</i> yang sudah dideklarasikan	<i>Import package</i> dan <i>library</i> → jalankan program	Menampilkan hasil dari <i>package</i> yang terdownload	✓

Dari hasil pengujian menurut Tabel 3.9 didapatkan hasil bahwa fungsionalitas memasukkan *library* ke program berjalan dengan baik.

2) Pembagian Data

Berikut hasil pembuatan dan pengujian pembagian data :

a) Kode Program

Keterangan Kode Program sebagai berikut:

(Lampiran 6 Kode Pengujian Pembagian Data)

b) Pengujian fungsionalitas

Pengujian fungsionalitas dilakukan berdasarkan hasil menjalankan kode program pada Google Colab. Hasil pengujian sebagai berikut:

```
1 data = ImageClassifierDataLoader.from_folder(base_dir)
2 train_data, test_data = data.split(0.9)
3
4 INFO:tensorflow:Load image with size: 7200, num_label: 12, labels: Batik Ceplok, Batik Jlamprang, Batik Kawung, Batik Kembang, Batik Parang, Batik Tulis, Batik Wayang, Batik Wadon, Batik Wato, Batik Wulungan, Batik Yogyakarta, Batik Yos Sudarmo
```

Gambar 3. 27 Hasil Pembagian Data *Training* dan *Data Testing*

Pada Gambar 3.27 menunjukkan bahwa data yang didapat dari *base_dir* dibagi menjadi *train data* dan *test data* dengan akumulasi 90% untuk *train data* dan 10% untuk *test data*.



Gambar 3. 28 Hasil Menampilkan Sampel dari *Dataset*

Pada Gambar 3.28 menunjukkan bahwa data yang terdapat pada *drive* sudah siap untuk dilakukan pengolahan data. Data yang akan diolah yaitu berjumlah 12 kelas.

Tabel 3. 10 Pengujian Fungsionalitas Pembagian Data

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Membagi Data	Agar <i>dataset</i> dapat terbagi menjadi <i>train data</i> dan <i>test data</i>	Masukkan fungsi <i>Image Classifier DataLoader</i> → Bagi data menjadi 0.9 yang berarti 90% <i>training</i> 10% <i>testing</i>	Menampilkan jumlah data yaitu 7200, jumlah label yaitu 12 dan menampilkan label dari batik	✓
Menampilkan Sampel Data	Agar dapat melihat sampel dari data batik	Menentukan ukuran gambar dari sampel → menentukan	Motif batik akan ditampilkan berjumlah 25	✓

	yang digunakan	jumlah dari sampel yang ditampilkan	gambar secara <i>random</i>	
--	----------------	-------------------------------------	-----------------------------	--

Dari hasil pengujian menurut Tabel 3.10 didapatkan hasil bahwa fungsionalitas membagi data kedalam *data train* dan *data test* berjalan dengan baik dan data sampel motif batik juga dapat ditampilkan pada layer Google Colab.

3) *Train Model*

Berikut hasil pembuatan dan pengujian *train model* :

a) Kode Program

Keterangan Kode Program sebagai berikut:

(Lampiran 7 Kode Pengujian *Train Model*)

b) Pengujian fungsionalitas

Pengujian fungsionalitas dilakukan berdasarkan hasil menjalankan kode program pada Google Colab. Hasil pengujian sebagai berikut:

```
...
INFO:tensorflow:Retraining the models...
INFO:tensorflow:Retraining the models...
Model: "sequential_1"
-----  

Layer (type)          Output Shape         Param #
-----  

hub_keras_layer_viv2_1 (HubK (None, 1280))      2257984  

dropout_1 (Dropout)        (None, 1280)          0  

dense_1 (Dense)           (None, 12)            15372  

-----  

Total params: 2,273,356
Trainable params: 2,239,244
Non-trainable params: 34,112
-----  

None
INFO:tensorflow:Use default resize_bicubic.
INFO:tensorflow:Use default resize_bicubic.
INFO:tensorflow:Use customized resize method bilinear
INFO:tensorflow:Use customized resize method bilinear
Epoch 1/10
202/202 [=====] - 1088s 5s/step - loss: 1.6851 - accuracy: 0.5847
Epoch 2/10
202/202 [=====] - 1084s 5s/step - loss: 1.0054 - accuracy: 0.8687
Epoch 3/10
202/202 [=====] - 1135s 6s/step - loss: 0.8311 - accuracy: 0.9391
Epoch 4/10
202/202 [=====] - 1130s 6s/step - loss: 0.7786 - accuracy: 0.9637
Epoch 5/10
202/202 [=====] - 1091s 5s/step - loss: 0.7520 - accuracy: 0.9728
Epoch 6/10
202/202 [=====] - 1083s 5s/step - loss: 0.7245 - accuracy: 0.9821
Epoch 7/10
202/202 [=====] - 1106s 5s/step - loss: 0.7130 - accuracy: 0.9873
Epoch 8/10
202/202 [=====] - 1093s 5s/step - loss: 0.7037 - accuracy: 0.9872
Epoch 9/10
202/202 [=====] - 1134s 6s/step - loss: 0.6945 - accuracy: 0.9894
Epoch 10/10
172/202 [=====>....] - ETA: 2:52 - loss: 0.6855 - accuracy: 0.9920
```

Gambar 3. 29 Hasil *Training Model*

Pada Gambar 3.29 menunjukkan bahwa hasil akurasi dari *training* model yaitu 0.9920 dan *loss* akurasinya 0.6855. *Epoch* yang diberikan yaitu 10, yang mana disetiap *epoch* terdapat 202 iterasi dan perjalanan *epochnya* terdapat kenaikan akurasi. Proses ini menggunakan teknik klasifikasi MobileNet.

Tabel 3. 11 Pengujian Fungsionalitas *Train Model*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Train Model</i>	Agar model dapat dilatih sebelum dilakukan <i>testing</i> model	Masukkan banyaknya <i>epoch</i> → melakukan klasifikasi model dengan MobileNet	Menampilkan informasi mengenai model dan akurasi dari model tersebut yaitu 0.9920	✓

Dari hasil pengujian menurut Tabel 3.11 didapatkan hasil bahwa fungsionalitas melakukan *training data* berjalan dengan baik yaitu menghasilkan akurasi sebesar 0.9920 yang berarti nilai mendekati nilai sempurna yaitu 1.

4) *Testing Model*

Berikut hasil pembuatan dan pengujian *testing* model :

a) Kode Program

Keterangan Kode Program sebagai berikut:

(Lampiran 8 Kode Pengujian *Testing Model*)

b) Pengujian fungsionalitas

Pengujian fungsionalitas dilakukan berdasarkan hasil menjalankan kode program pada Google Colab. Hasil pengujian sebagai berikut:

```
INFO:tensorflow:Use customized resize method bilinear
INFO:tensorflow:Use customized resize method bilinear
23/23 [=====] - 40s 1s/step - loss: 0.6538 - accuracy: 0.9986
```

Gambar 3. 30 Hasil *Testing Model*

Pada Gambar 3.30 menunjukkan bahwa hasil akurasi dari *testing* model yaitu 0.9986 dan 0.6538 untuk *loss accuracy*. Proses berjumlah 23 iterasi per-*epoch* dan ditempuh dalam waktu 40s.

Tabel 3. 12 Pengujian Fungsionalitas *Testing Model*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Testing Model</i>	Agar model dapat dikenali saat motif dideteksi secara langsung	Masukkan kode untuk <i>testing</i> data	Menampilkan akurasi dari data <i>testing</i> yaitu 0.9986	✓

Dari hasil pengujian menurut Tabel 3.12 didapatkan hasil bahwa fungsionalitas melakukan *testing* data berjalan dengan baik yaitu menghasilkan akurasi sebesar 0.9986 yang berarti nilai tersebut mendekati nilai sempurna yaitu 1.

5) *Export Model*

Berikut hasil pembuatan dan pengujian *export model* :

a) Kode Program

Keterangan Kode Program sebagai berikut:

(Lampiran 9 Kode Pengujian *Export Model*)

b) Pengujian fungsionalitas

Pengujian fungsionalitas dilakukan berdasarkan hasil menjalankan kode program pada Google Colab. Hasil pengujian sebagai berikut:

```

● for model, name in zip(models_list, model_names):
    name = name.replace(" ", "_").lower()

    # Export as TensorFlow Lite model.
    model.export(export_dir='./'+name)

□ WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/training/tracking/tracking.py:109: Model.state_updates (from tensorflow.python.keras.engine.base_layer)
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/training/tracking/tracking.py:109: Model.state_updates (from tensorflow.python.keras.engine.base_layer)
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/training/tracking/tracking.py:109: Layer.updates (from tensorflow.python.keras.engine.base_layer)
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/training/tracking/tracking.py:109: Layer.updates (from tensorflow.python.keras.engine.base_layer)
Instructions for updating:
This property should not be used in TensorFlow 2.0, as updates are applied automatically.
INFO:tensorflow:Assets written to: /tmp/tmpdpxj7vun/assets
INFO:tensorflow:Assets written to: /tmp/tmpdpxj7vun/assets
INFO:tensorflow:Label file is inside the TFLite model with metadata.
INFO:tensorflow:Label file is inside the TFLite model with metadata.
INFO:tensorflow:Saving labels in /tmp/tmpaahn3q00/labels.txt.
INFO:tensorflow:Saving labels in /tmp/tmpaahn3q00/labels.txt.

```

Gambar 3. 31 Hasil *Export Model*

Pada Gambar 3.31 menunjukkan bahwa model dari MobileNet sudah berhasil di*export* menjadi .tflite, model akan tersimpan secara sementara dibagian folder google colab. Untuk proses *download* model dapat dilakukan dengan membuka folder pada google colab dan download model.tflite.

Tabel 3. 13 Pengujian Fungsionalitas *Export Model*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Export Model</i>	Agar model dapat berbentuk .tflite	Masukkan kode nama model → <i>export model</i>	Model akan terbuat dan akan disimpan sementara di folder Google Colab	✓
<i>Download Model</i>	Agar model dapat didownload	Membuka folder dari Google Colab → buka file dan lakukan <i>download model</i>	Model akan didownload dan masuk kedalam penyimpan komputer lokal	✓

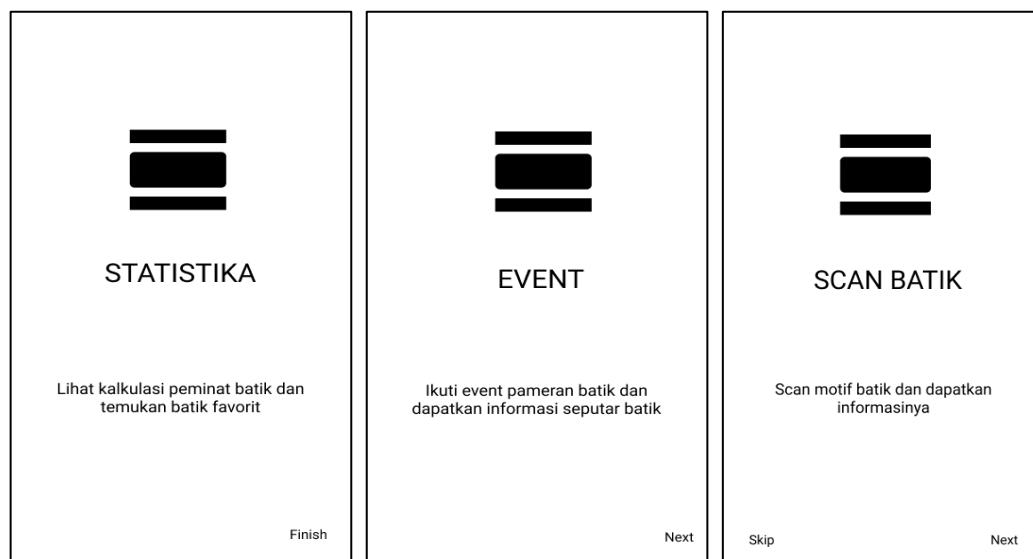
Dari hasil pengujian menurut Tabel 3.13 didapatkan hasil bahwa fungsionalitas melakukan *export model* kedalam bentuk .tflite dapat berjalan dengan baik. Hal tersebut dapat dibuktikan bahwa model dapat diunduh dan dipasangkan kedalam aplikasi android.

3.4.5. Sprint 5: Perancangan Antarmuka Aplikasi

Pada bagian *sprint* ini akan dibahas mengenai perancangan tampilan *User Interface* (UI) dari aplikasi pengenalan batik untuk pengunjung pameran berbasis android dan *image processing*.

1) Halaman *Onboarding*

Halaman *onboarding* aplikasi ini sebagai berikut:



Gambar 3. 32 Tampilan Halaman *Onboarding*

Gambar 3.32 merupakan tampilan pembuka dari aplikasi ini, tampilan tersebut akan muncul pada saat aplikasi dijalankan pertama kali oleh pengguna. Tujuannya memberikan informasi kepada pengguna tentang cara kerja aplikasi.

2) Halaman *Splash Screen*

Halaman *splash screen* aplikasi ini sebagai berikut:



Gambar 3. 33 Tampilan Halaman *Splash Screen*

Gambar 3.33 merupakan tampilan dari halaman *splash screen*, tampilan ini akan muncul setelah tampilan dari halaman *onboarding*. Halaman *splash screen* menunjukkan logo utama dari aplikasi yang bertujuan memberikan kesan pertama mengenai fungsi dari aplikasi tersebut.

3) Halaman *Login*

Halaman *login* aplikasi ini sebagai berikut:



Gambar 3. 34 Tampilan Halaman *Login Admin*

Gambar 3.34 merupakan halaman *login* yang digunakan *user* untuk masuk ke aplikasi. Untuk bisa *login* pengguna harus memasukkan akun yang sudah terdaftar sebelumnya dengan memasukkan *email* dan *password*. Apabila data yang dimasukkan sesuai maka akan diteruskan akses menuju halaman selanjutnya.

4) Halaman *Register*

Halaman *register* aplikasi ini sebagai berikut:

BATIK.IN

Register

Email

Password

Confirm

Sign Up

[Back to Login](#)

Gambar 3. 35 Tampilan Halaman *Register*

Gambar 3.35 menunjukkan tampilan halaman *register*. Halaman ini berguna untuk menambahkan user baru dengan mendaftarkan alamat *email*, *password*, dan konfirmasi *password*. Agar berhasil melakukan *register* akun, perlu mengklik tombol *Sign Up*.

5) Halaman Pengisian Data Pengguna

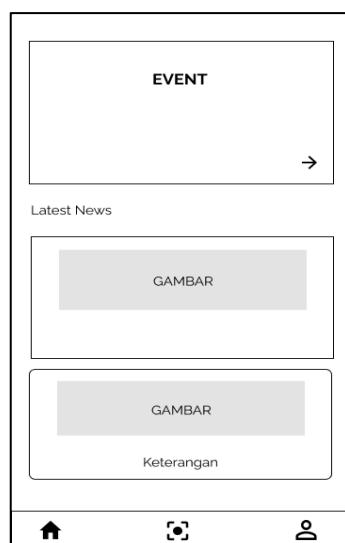
Halaman pengisian data pengguna aplikasi ini sebagai berikut:

Gambar 3. 36 Tampilan Halaman Pengisian Data Pengguna

Gambar 3.36 menunjukkan tampilan halaman pengisian data pengguna dari aplikasi. Pada halaman ini pengguna baru akan melengkapi data seperti foto profil, nama, no. hp, tanggal lahir, dan jenis kelamin. Setelah mengisikan data dengan lengkap maka pengguna akan diarahkan ke halaman utama.

6) Halaman Home

Halaman *home* aplikasi ini sebagai berikut:

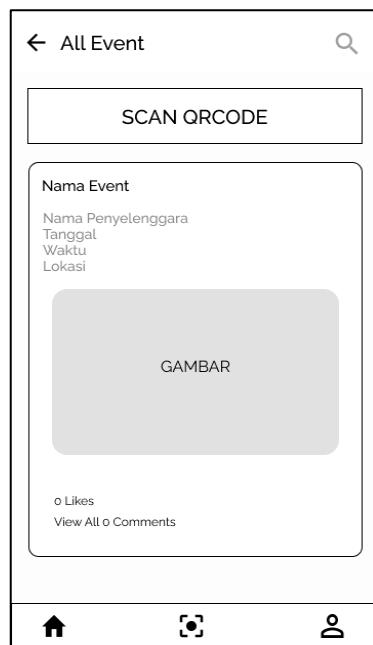


Gambar 3. 37 Tampilan Halaman *Home*

Gambar 3.37 menunjukkan tampilan *home* dari aplikasi. Pada halaman ini akan ditampilkan menu untuk mengakses halaman *event* pameran serta berita mengenai batik. Berita dapat berupa *event* pameran, produk batik, maupun hal yang berkaitan dengan batik. Dalam menu *event* tersebut ketika pengguna mengklik tombol panah akan diarahkan ke halaman *event* dimana pengunjung pameran dapat melihat *event* batik apa saja yang diselenggarakan.

7) Halaman *Event* Pameran

Halaman *event* pameran aplikasi ini sebagai berikut:

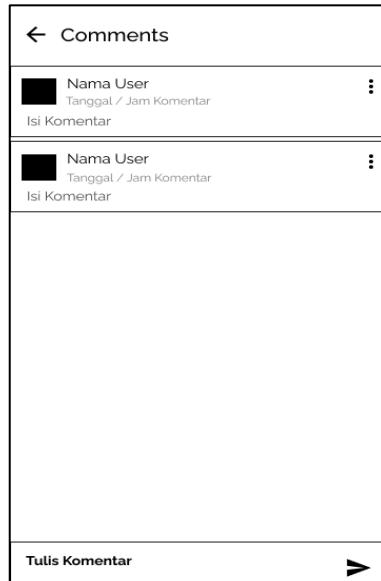


Gambar 3. 38 Tampilan Halaman *Event* Pameran

Gambar 3.38 merupakan tampilan dari *event* pameran, halaman ini menunjukkan semua pameran batik yang sedang dilaksanakan sekarang ataupun yang akan dilaksanakan di waktu mendatang. Pada setiap blok *event* pameran akan menampilkan nama dari pameran batik, lokasi, tanggal, waktu, dan penyelenggara pameran tersebut. Di dalam halaman ini pengguna atau pengujung pameran dapat memberikan *like* atau komentar yang ditunjukkan untuk pameran yang dipilih.

8) Halaman Komentar *Event*

Halaman komentar *event* aplikasi ini sebagai berikut:

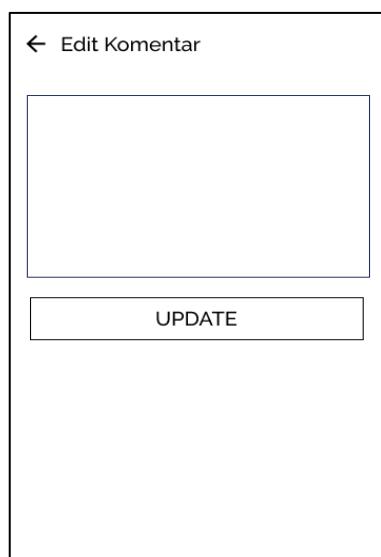


Gambar 3. 39 Tampilan Halaman Komentar *Event*

Gambar 3.39 menampilkan tampilan dari komentar *event*. Halaman ini berisikan komentar – komentar yang diberikan pengguna kepada *event* batik. Pengguna juga dapat mengedit atau menghapus komentar yang sudah diberikan ke *event* tersebut.

9) Halaman Mengubah Komentar *Event*

Halaman mengubah komentar *event* aplikasi ini sebagai berikut:

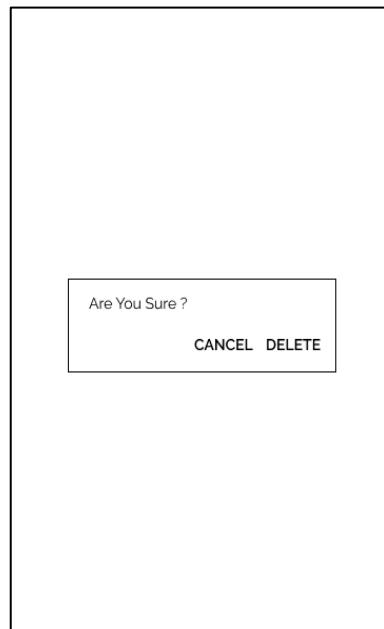


Gambar 3. 40 Tampilan Halaman Mengubah Komentar *Event*

Gambar 3.40 menampilkan tampilan dari mengubah komentar *event*. Halaman ini digunakan untuk mengubah komentar yang sudah diberikan. Agar komentar dapat berubah pengguna menekan tombol *update*.

10) Halaman Menghapus Komentar *Event*

Halaman menghapus komentar *event* aplikasi ini sebagai berikut:

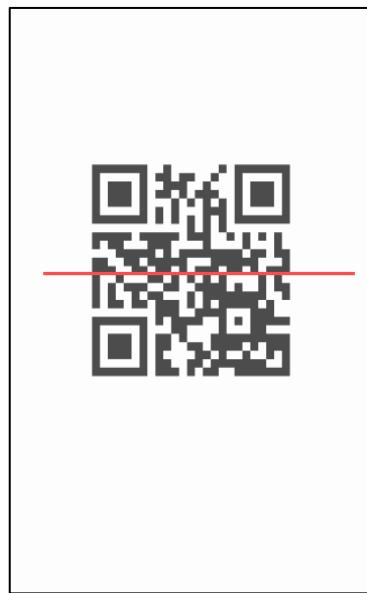


Gambar 3. 41 Tampilan Halaman Menghapus Komentar *Event*

Gambar 3.41 menampilkan tampilan dari menghapus komentar *event*. Halaman ini digunakan untuk menghapus komentar yang sudah diberikan. Agar komentar dapat berubah pengguna mengklik tombol *delete*.

11) Halaman *Scan Qrcode Event*

Halaman *scan qrcode event* aplikasi ini sebagai berikut:

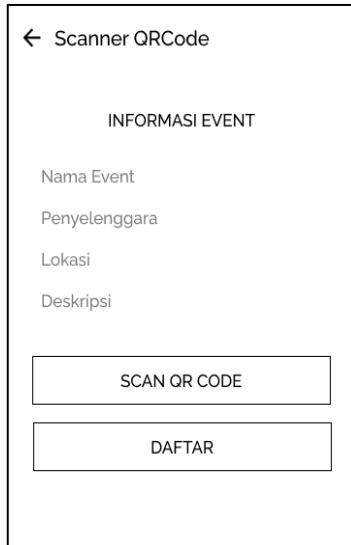


Gambar 3. 42 Tampilan Halaman *Scan Qrcode Event*

Gambar 3.42 merupakan tampilan *scan qrcode event*. Untuk konfirmasi kehadiran terhadap *event* yang diikuti, pengguna memilih tombol *scan qrcode* dan mengarahkan kamera ke *qrcode* yang sebelumnya sudah disediakan oleh penyelenggara *event*. Jika berhasil registrasi kehadiran, pengguna akan mendapatkan notifikasi bahwa data sudah masuk. *Event* pameran yang diikuti ini akan tersimpan pada halaman profil pengguna.

12) Halaman Konfirmasi *Event*

Halaman konfirmasi *event* aplikasi ini sebagai berikut:

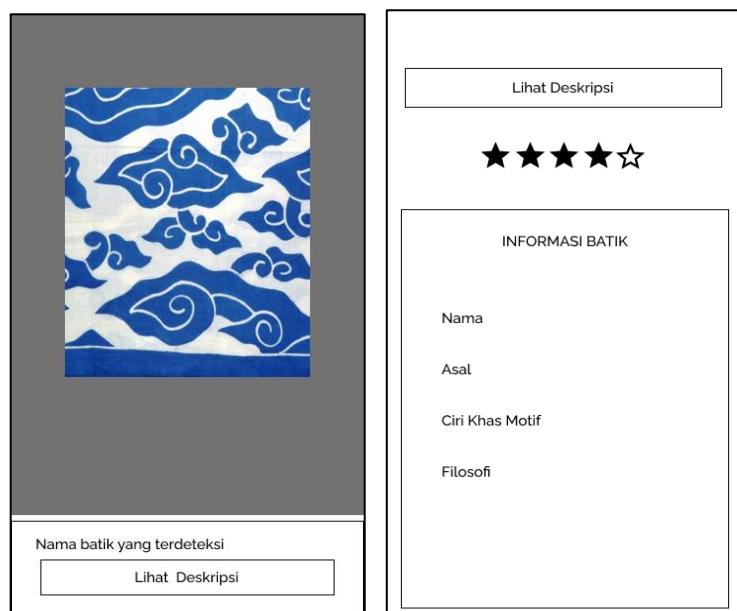


Gambar 3. 43 Tampilan Halaman Konfirmasi *Event*

Gambar 3.43 merupakan tampilan halaman konfirmasi *event*. Pada halaman ini setelah *qrcode* berhasil dideteksi kemudian pengguna memilih tombol daftar untuk konfirmasi kehadiran pada *event* tersebut.

13) Halaman Deteksi Batik

Halaman deteksi batik aplikasi ini sebagai berikut:

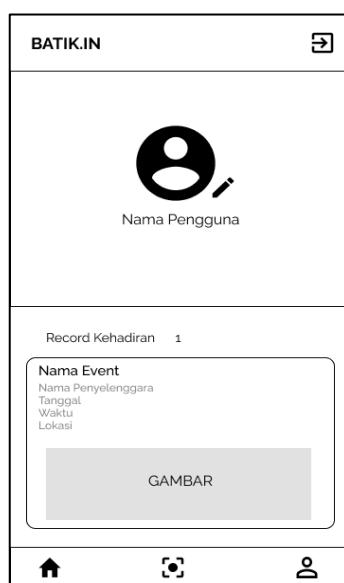


Gambar 3. 44 Tampilan Halaman Deteksi Batik

Gambar 3.44 menunjukkan tampilan dari deteksi batik. Tampilan utama dari halaman ini berupa tombol *scan* yang digunakan pengguna untuk mendeteksi batik yang diinginkan. Ketika pengguna mengeklik tombol *scan* maka akan diarahkan ke kamera utama. Pengguna mengarahkan ke batik yang akan dideteksi. Kemudian akan ditampilkan nama batik yang dideteksi beserta akurasi dari hasil *scanning* tersebut. Untuk melihat deskripsi batik yang di *scan* tersebut pengguna harus memilih tombol *view deskripsi*, maka akan ditampilkan *card view* berisi nama batik, asal batik, ciri khas batik, dan filosofi dari batik yang berhasil dideteksi. Pengguna dapat memberikan *like* pada batik tersebut dan akan disimpan pada admin sebagai data batik favorit pilihan pengunjung pameran.

14) Halaman Profil

Halaman profil aplikasi ini sebagai berikut:



Gambar 3. 45 Tampilan Halaman Profil

Gambar 3.45 menunjukkan tampilan dari profil pengguna. Halaman ini berisikan informasi pengguna dan *record* kehadiran *event* yang diikuti. Dalam halaman ini pengguna juga dapat mengubah informasi pribadi dengan menekan tombol *edit*.

15) Halaman *Login-Admin*

Halaman *login admin* aplikasi ini sebagai berikut:

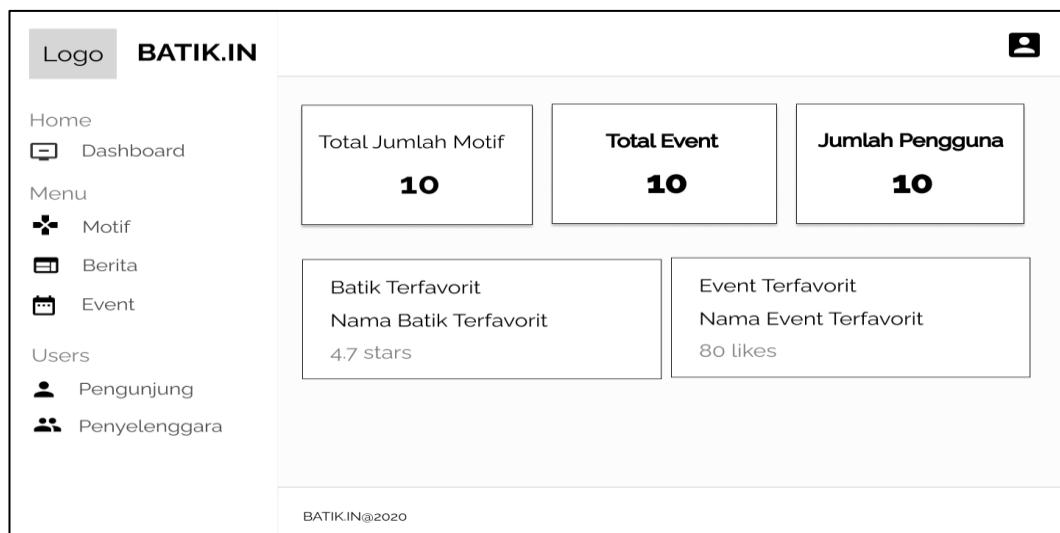
The screenshot shows a login interface for an application named BATIK.IN. The title 'BATIK.IN' is at the top. Below it is a form with the title 'Login'. The form includes fields for 'Email' and 'Password', and a 'LOGIN' button.

Gambar 3. 46 Tampilan Halaman *Login Admin*

Gambar 3.46 menunjukkan tampilan halaman *login* yang digunakan untuk melakukan proses manajemen data dari pengguna aplikasi. Apabila data yang dimasukkan sesuai maka akan diteruskan akses menuju halaman selanjutnya. Dalam *login* ini hanya dapat dilakukan oleh *user* yang sebelumnya sudah diberikan *username* serta *password*.

16) Halaman *Dashboard*

Halaman *dashboard* aplikasi ini sebagai berikut:



Gambar 3. 47 Tampilan Halaman *Dashboard*

Gambar 3.47 menunjukkan tampilan halaman *dashboard*. Pada halaman ini akan ditampilkan kalkulasi dari jumlah motif batik, jumlah *event* pameran, jumlah pengguna aplikasi, motif terfavorit dan *event* terfavorit.

17) Halaman Menampilkan Informasi Motif Batik

Halaman menampilkan informasi motif batik aplikasi ini sebagai berikut:

Motif	Asal Motif	Ciri Khas	Filosofi	Rating	Action
					EDIT DELETE

Gambar 3. 48 Tampilan Halaman Menampilkan Informasi Motif Batik

Gambar 3.48 menunjukkan tampilan halaman menampilkan informasi motif batik. Pada halaman ini berisi informasi mengenai nama motif, asal daerah, serta detail dari motif batik. Dalam halaman informasi *dataset* ini admin juga dapat melakukan *editing* atau penghapusan data motif batik dengan *action edit* dan *delete*.

18) Halaman Menambah Informasi Motif Batik

Halaman menambah informasi motif batik aplikasi ini sebagai berikut:

Gambar 3. 49 Tampilan Halaman Menambah Informasi Motif Batik

Gambar 3.49 menunjukkan tampilan halaman menambah informasi motif batik. Untuk dapat menambahkan motif baru *user* mengisikan nama motif, asal daerah motif, ciri khas dari motif tersebut, dan filosofi yang terkandung dalam suatu motif batik. Tombol *submit* digunakan untuk menyimpan data, dan tombol *reset* digunakan untuk menghapus data keseluruhan atau membatalkan pengisian data informasi motif batik.

19) Halaman Mengedit Informasi Motif Batik

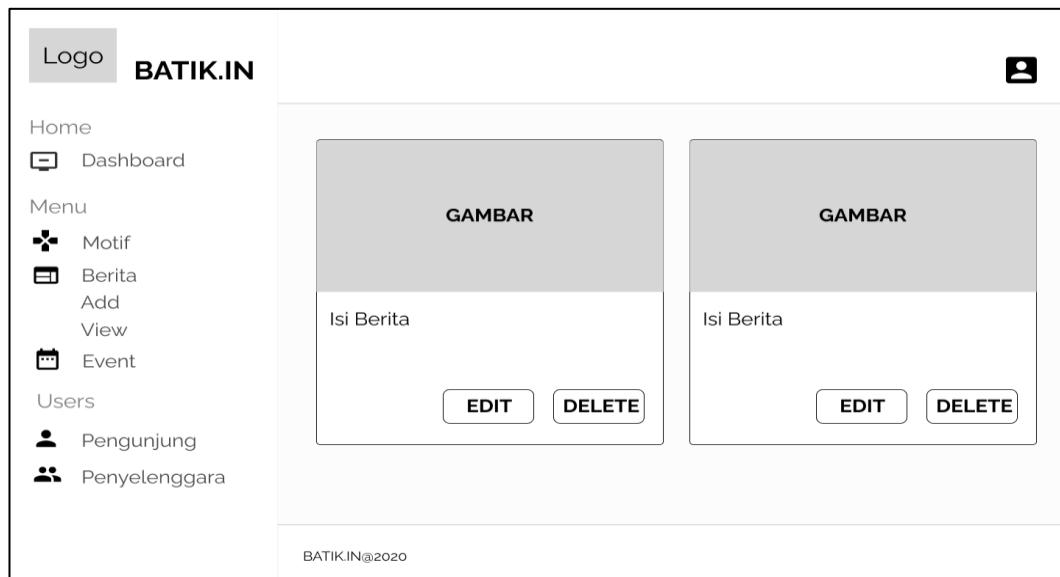
Halaman informasi motif batik aplikasi ini sebagai berikut:

Gambar 3. 50 Tampilan Halaman Mengedit Informasi Motif Batik

Gambar 3.50 menunjukkan tampilan halaman mengedit informasi motif batik. Untuk mengubah keterangan dari informasi batik, admin mengisikan kembali informasi apa saja yang akan diubah dan menekan tombol *submit* untuk konfirmasi pengubahan data.

20) Halaman Menampilkan Berita

Halaman menampilkan berita batik aplikasi ini sebagai berikut:



Gambar 3. 51 Tampilan Halaman Menampilkan Berita

Gambar 3.51 menunjukkan tampilan halaman menampilkan berita. Pada halaman ini berisikan berita-berita mengenai batik yang sudah ditambahkan. Berita yang sudah ditambahkan pada halaman ini akan ditampilkan pada halaman utama aplikasi deteksi pada android. Admin dapat mengubah isi berita atau menghapus berita.

21) Halaman Menambah Berita

Halaman menambah berita batik aplikasi ini sebagai berikut:

Gambar 3. 52 Tampilan Halaman Menambah Berita

Gambar 3.52 menunjukkan tampilan halaman menambah berita. Untuk menambahkan berita admin mengisi judul berita, deskripsi berita, dan gambar yang akan ditampilkan. Tombol *submit* digunakan untuk menyimpan berita, dan tombol *reset* digunakan untuk melakukan pembatalan pengisian data berita.

22) Halaman Mengedit Berita

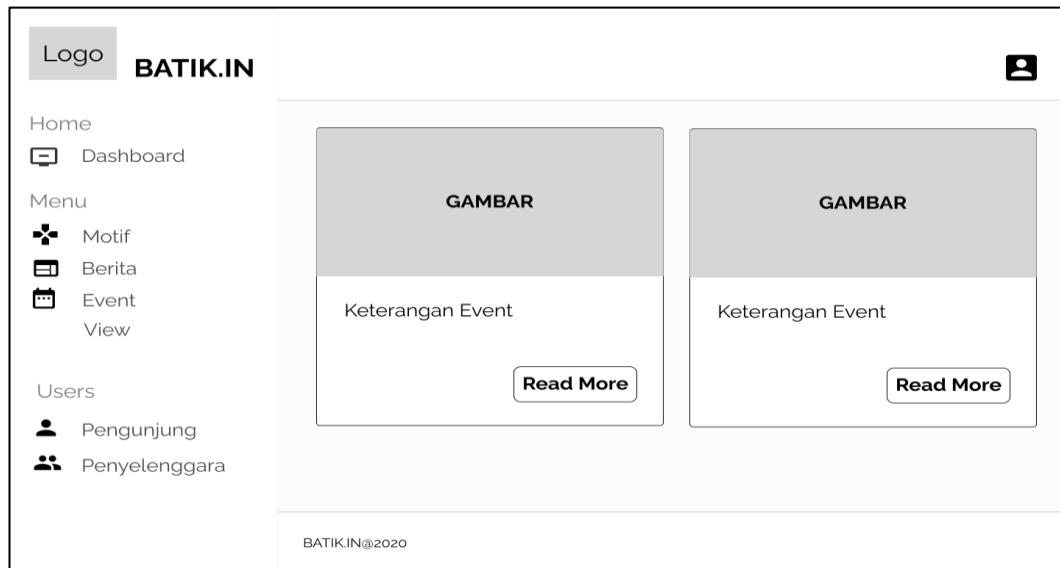
Halaman mengedit berita batik aplikasi ini sebagai berikut:

Gambar 3. 53 Tampilan Halaman Mengedit Berita

Gambar 3.53 menunjukkan tampilan halaman mengedit berita. Untuk melakukan perubahan data pada berita, admin memperbarui data yang akan diubah kemudian menekan tombol *submit*.

23) Halaman Menampilkan All Event Pameran

Halaman menampilkan *all event* pameran aplikasi ini sebagai berikut:

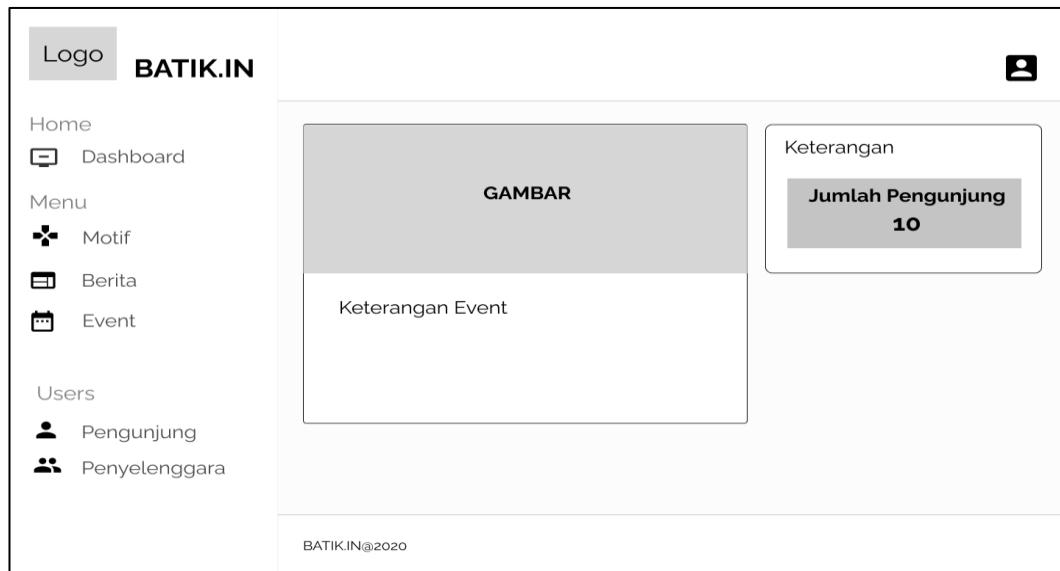


Gambar 3. 54 Tampilan Halaman Menampilkan All Event Pameran

Gambar 3.54 menunjukkan tampilan halaman menampilkan *event* pameran. Pada halaman ini akan ditampilkan *event* pameran batik yang sudah terdaftar. Admin dapat melihat detail dari *event* dengan memilih tombol *read more*.

24) Halaman Menampilkan Detail Event

Halaman menampilkan detail *event* aplikasi ini sebagai berikut:

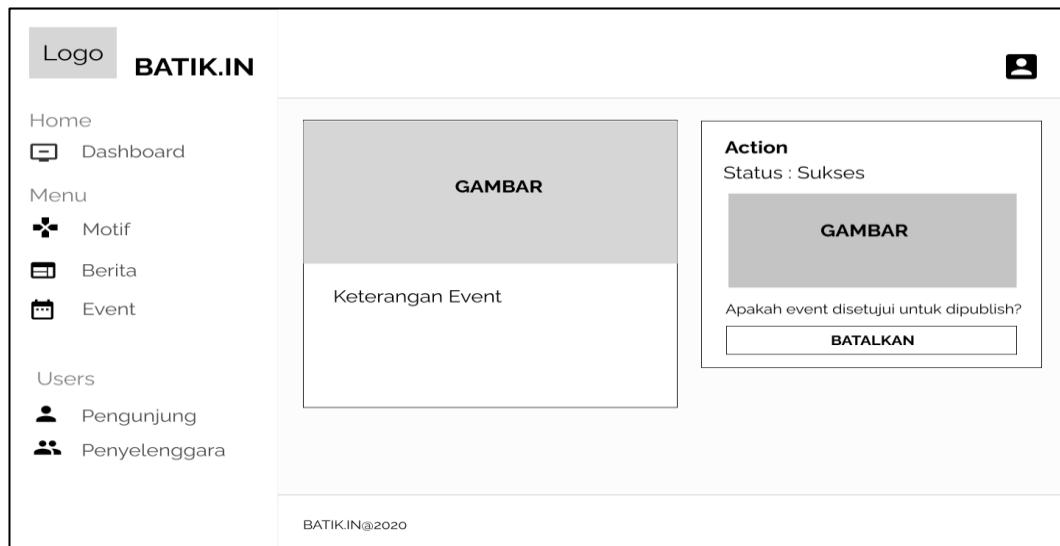


Gambar 3. 55 Tampilan Halaman Menampilkan Detail Event

Gambar 3.55 menunjukkan tampilan halaman menampilkan detail *event*. Pada halaman ini akan ditampilkan informasi mengenai *event* tersebut beserta keterangan jumlah pengunjung yang hadir dalam *event* tersebut. Jumlah pengunjung ini didapatkan dari proses *scanning qrcode* yang dilakukan pengguna android.

25) Halaman Konfirmasi Event

Halaman konfirmasi *event* aplikasi ini sebagai berikut:



Gambar 3. 56 Tampilan Halaman Konfirmasi Event

Gambar 3.56 menunjukkan tampilan halaman konfirmasi *event*. Pada halaman ini dapat dilihat status *event* yang akan ditampilkan. Admin dapat membatalkan persetujuan *event* menggunakan tombol batalkan.

26) Halaman Data Pengunjung

Halaman data pengunjung aplikasi ini sebagai berikut:

Nama	Email	Tgl Lahir	Gender	No Hp	Profile	Action
						EDIT DELETE

Gambar 3. 57 Tampilan Halaman Data Pengunjung

Gambar 3.57 menunjukkan tampilan halaman data pengunjung. Data pengunjung diambil dari aplikasi Batik.in. Dalam halaman ini berisikan data pengunjung *event* pameran seperti nama, email, tanggal lahir, gender, no.hp dan *profile* dari pengunjung pameran. Jumlah data pengunjung akan bertambah seiring dengan *registrasi* akun pada aplikasi Batik.In. Admin dapat mengedit ataupun menghapus data pengunjung dengan *action edit* dan *delete*.

27) Halaman Menambah Data Pengunjung

Halaman menambah data pengunjung aplikasi ini sebagai berikut:

The screenshot shows a modal window titled 'ADD - PENGUNJUNG'. It contains fields for 'Nama Lengkap', 'Email', 'Password', 'Tanggal Lahir', 'Jenis Kelamin' (with radio buttons for 'Laki - Laki' and 'Perempuan', where 'Perempuan' is selected), 'No. Hp', 'Foto Profile' (with a 'Choose File' button), and two buttons at the bottom: 'SUBMIT' and 'RESET'.

Gambar 3. 58 Tampilan Halaman Menambah Data Pengunjung

Gambar 3.58 menunjukkan tampilan halaman menambah data pengunjung. Pada halaman ini digunakan untuk menambahkan pengunjung yang tidak mempunyai aplikasi Batik.In. Untuk menambahkan pengunjung admin mengisikan data seperti nama lengkap, email, password, tanggal lahir, jenis kelamin, no.hp, dan foto profile. Tombol *submit* digunakan untuk proses pemyimpanan data, dan tombol *reset* digunakan untuk mengulangi pengisian data.

28) Halaman Mengedit Data Pengunjung

Halaman mengedit data pengunjung aplikasi ini sebagai berikut:

The screenshot shows a modal window titled 'ADD - EDIT'. It contains fields for 'Nama Lengkap' (Full Name), 'Email', 'Password', and 'Tanggal Lahir' (Birth Date). Below these are gender selection buttons ('Laki - Laki' and 'Perempuan'), a 'No. Hp' (Phone Number) field, and a 'Foto Profile' (Profile Photo) input field with a 'Choose File' button. At the bottom is a 'SUBMIT' button.

Gambar 3. 59 Tampilan Halaman Mengedit Data Pengunjung

Gambar 3.59 menunjukkan tampilan mengedit halaman data pengunjung. Pada halaman ini digunakan untuk mengedit data pengunjung. Admin melakukan perubahan pada data yang akan diedit dan menekan tombol *submit* untuk konfirmasi perubahan data.

29) Halaman Data Penyelenggara

Halaman data penyelenggara aplikasi ini sebagai berikut:

DATA PENYELENGGARA						
Nama	Email	Organisasi	Alamat	No Hp	Profile	Action
						EDIT DELETE

BATIK.IN@2020

Gambar 3. 60 Tampilan Halaman Data Penyelenggara

Gambar 3.60 menunjukkan tampilan halaman data penyelenggara. Dalam halaman ini berisikan data penyelenggara *event* pameran seperti nama, email, organisasi, alamat, no.hp dan *profile* dari penyelenggara pameran. Admin dapat mengedit ataupun menghapus data pengunjung dengan *action edit* dan *delete*.

30) Halaman Menambah Data Penyelenggara

Halaman menambah data penyelenggara aplikasi ini sebagai berikut:

ADD - PENYELENGGARA ×

Nama Lengkap

Email

Alamat Rumah/Organisasi/Perusahaan

Nama Organisasi Perusahaan

No. Hp/Telepon

Password

Foto Profile

SUBMIT **RESET**

Gambar 3. 61 Tampilan Halaman Menambah Data Penyelenggara

Gambar 3.61 menunjukkan tampilan halaman data penyelenggara. Pada halaman ini digunakan untuk menambahkan penyelenggara *event* pameran batik. Untuk menambahkan penyelenggara admin mengisikan data seperti nama lengkap, email, password, alamat organisasi/perusahaan/rumah, nama organisasi/perusahaan, no.hp/telepon, dan foto profile. Tombol *submit* digunakan untuk proses pemyimpanan data, dan tombol *reset* digunakan untuk mengulangi pengisian data.

31) Halaman Mengedit Data Penyelenggara

Halaman mengedit data penyelenggara aplikasi ini sebagai berikut:

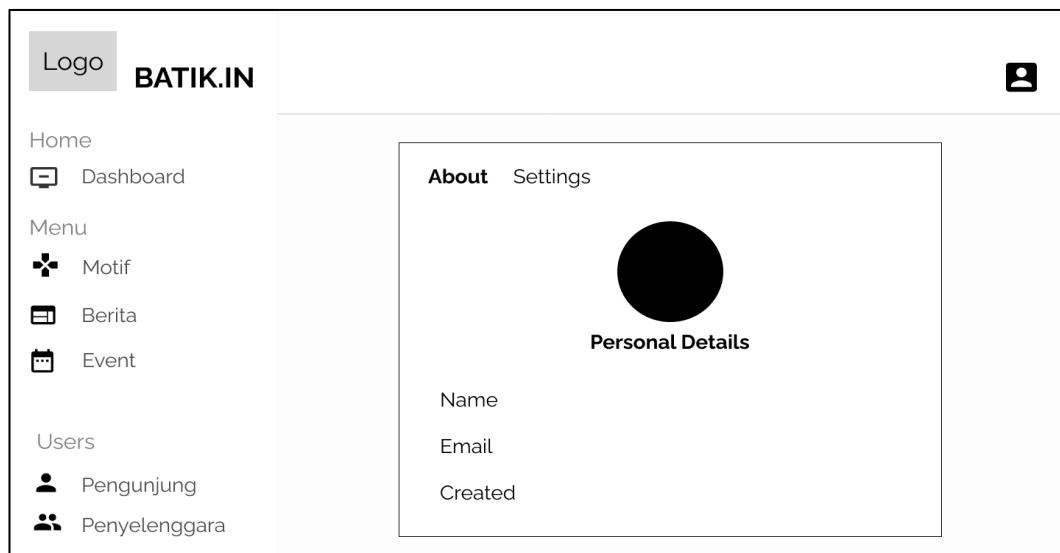
The screenshot shows a modal window titled "EDIT - PENYELENGGARA". Inside the window, there are seven input fields: "Nama Lengkap", "Email", "Alamat Rumah/Organisasi/Perusahaan", "Nama Organisasi Perusahaan", "No. Hp/Telepon", "Password", and "Foto Profile". Below the "Foto Profile" field is a "Choose File" button. At the bottom of the window is a "SUBMIT" button.

Gambar 3. 62 Tampilan Halaman Mengedit Data Penyelenggara

Gambar 3.62 menunjukkan tampilan halaman data penyelenggara. Pada halaman ini digunakan untuk mengedit data penyelenggara. Admin melakukan perubahan pada data yang akan diedit dan menekan tombol *submit* untuk konfirmasi perubahan data.

32) Halaman Profil

Halaman profil aplikasi ini sebagai berikut:



Gambar 3. 63 Tampilan Halaman Profil

Gambar 3.63 menunjukkan tampilan profil. Dalam halaman ini berisikan detail seperti nama, email, dan tanggal pembuatan akun. Admin dapat merubah informasi akun pada bagian *settings*.

33) Halaman Menambahkan Event

Halaman menambahkan *event* aplikasi ini sebagai berikut:

The screenshot shows a modal window titled 'ADD - EVENT' with an 'X' button at the top right. The form contains the following fields:

- Nama:** An input field.
- Tanggal Mulai:** A date input field.
- Tanggal Selesai:** A date input field.
- Jam Mulai:** A time input field.
- Jam Selesai:** A time input field.
- Lokasi:** An input field.
- Deskripsi:** A text input field.
- Image:** A file input field labeled 'Choose File'.

At the bottom of the form are two buttons: 'SUBMIT' and 'RESET'.

Gambar 3. 64 Tampilan Halaman Menambahkan *Event*

Gambar 3.64 menunjukkan halaman menambahkan *event*. Dalam halaman ini penyelenggara dapat menambahkan *event* pameran batik yang nantinya akan disetujui terlebih dahulu oleh admin. Untuk menambahkan *event*, penyelenggara mengisikan informasi seperti nama *event*, tanggal mulai dan tanggal selesai, jam mulai dan jam selesai, lokasi, deskripsi, serta gambar dari *event*. Tombol *submit* digunakan untuk menyimpan data.

34) Halaman Mengedit *Event*

Halaman mengedit *event* aplikasi ini sebagai berikut:

The screenshot shows a modal window titled "EDIT - EVENT". Inside the window, there are several input fields and labels. At the top right is a close button (X).
Fields include:

- Nama (Name): An input field.
- Tanggal Mulai (Start Date): An input field.
- Tanggal Selesai (End Date): An input field.
- Jam Mulai (Start Time): An input field.
- Jam Selesai (End Time): An input field.
- Lokasi (Location): An input field.
- Deskripsi (Description): An input field.
- Image: A "Choose File" button.

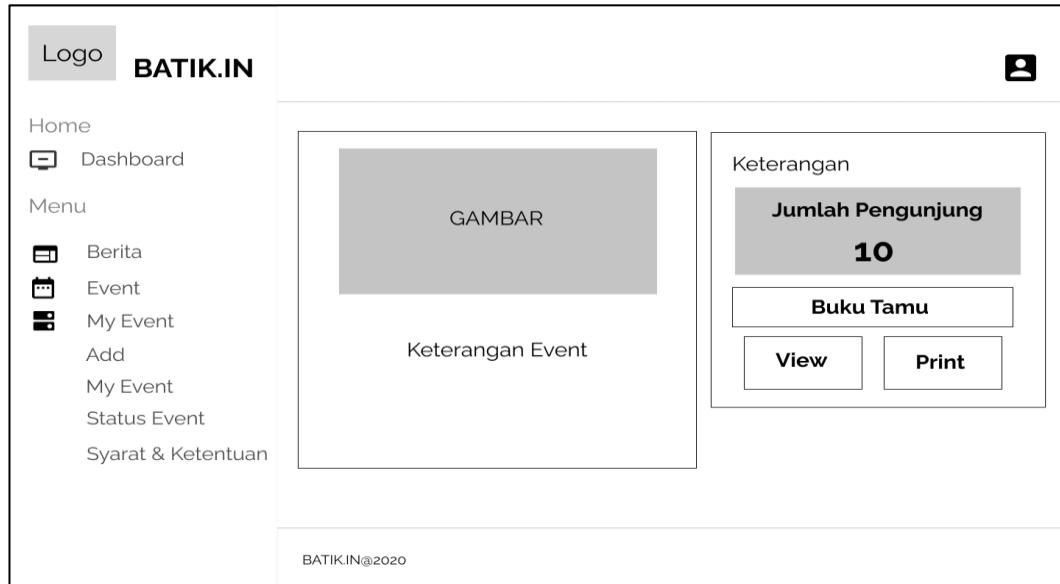
A "SUBMIT" button is located at the bottom of the form.

Gambar 3. 65 Tampilan Halaman Mengedit *Event*

Gambar 3.65 menunjukkan tampilan halaman mengedit *event*. Penyelenggara dapat melakukan perubahan data pada *event* yang ditambahkan. Untuk melakukan perubahan data digunakan tombol *submit*.

35) Halaman Menampilkan *MyEvent*

Halaman menampilkan *myevent* aplikasi ini sebagai berikut:



Gambar 3. 66 Tampilan Halaman Menampilkan *MyEvent*

Gambar 3.66 menunjukkan halaman tampilan *MyEvent*. Dalam halaman ini penyelenggara dapat melihat *event* yang ditambahkan beserta keterangan jumlah pengunjung yang mengikuti *event*. Dalam hal ini penyelenggara dapat melihat data pengunjung yang mengikuti *event* dengan memilih tombol buku tamu. Penyelenggara juga dapat menampilkan *qrcode* untuk proses kehadiran pengunjung serta dapat mencetak hasil *qrcode*.

36) Halaman Menampilkan QRCode

Halaman menampilkan *qrcode* aplikasi ini sebagai berikut:

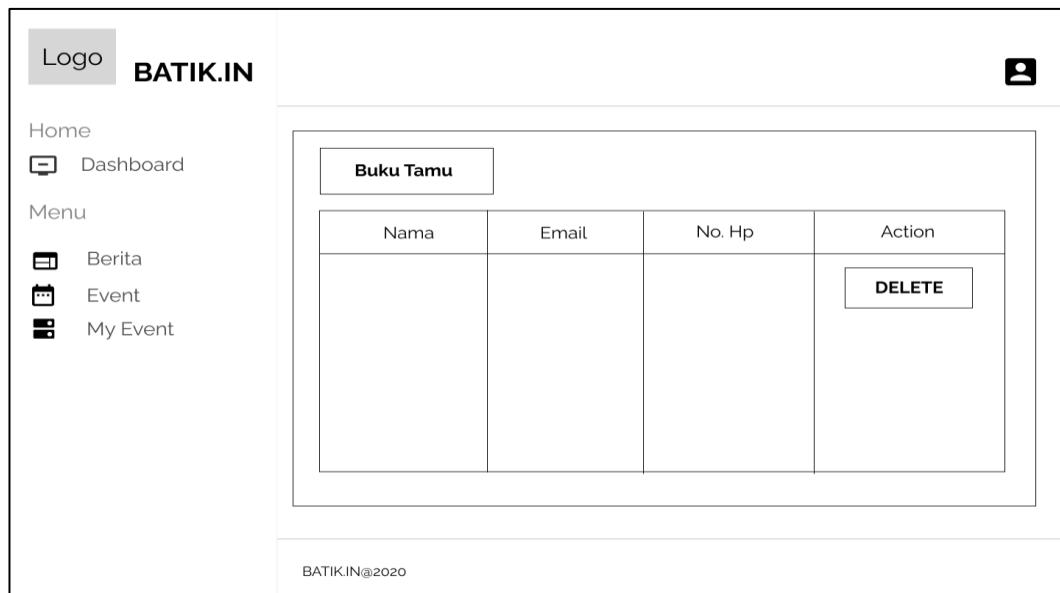


Gambar 3. 67 Tampilan Halaman Menampilkan QRCode

Gambar 3.67 menunjukkan tampilan halaman *QRCode*. Dalam halaman ini akan ditampilkan *qrcode* yang digunakan untuk konfirmasi kehadiran pengunjung. Qrcode ini akan *discan* pengunjung melalui aplikasi pada android.

37) Halaman Menampilkan Buku Tamu

Halaman menampilkan buku tamu aplikasi ini sebagai berikut:



Gambar 3. 68 Tampilan Halaman Menampilkan Buku Tamu

Gambar 3.68 menunjukkan tampilan halaman buku tamu. Dalam halaman ini berisikan data pengunjung yang sudah melakukan konfirmasi kehadiran melalui *scan barcode*. Penyelenggara dapat menghapus data dan menambahkan pengunjung apabila mengalami kendala saat *scan barcode*.

38) Halaman Menambahkan Kehadiran Pengunjung

Halaman menambahkan kehadiran pengunjung aplikasi ini sebagai berikut:

EVENT			
Nama	Email	No. Hp	Action

← ADD

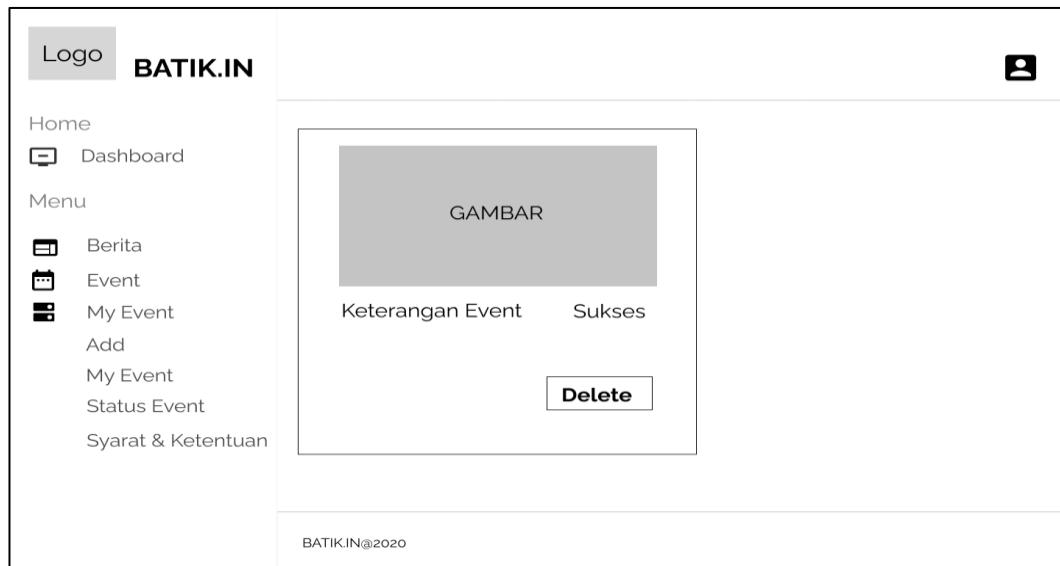
Submit
Reset

Gambar 3. 69 Tampilan Halaman Menambahkan Kehadiran Pengunjung

Gambar 3.69 menunjukkan tampilan halaman menambahkan kehadiran pengunjung. Dalam halaman ini penyelenggara *event* dapat menambahkan kehadiran atau buku tamu bagi pengunjung yang melalui kendala saat *scan barcode*. Data yang ditambahkan berupa nama, email, dan no.hp. Tombol *submit* digunakan untuk menyimpan data dan tombol *reset* digunakan untuk membatalkan pengisian data.

39) Halaman Menampilkan Status *Event*

Halaman menampilkan status *event* aplikasi ini sebagai berikut:



Gambar 3. 70 Tampilan Halaman Menampilkan Status *Event*

Gambar 3.70 menunjukkan tampilan halaman menampilkan status *event*. Apabila *event* yang didaftarkan ke admin berhasil maka akan tertulis sukses, namun apabila gagal akan muncul tampilan gagal. Penyelenggara juga dapat menghapus *event*.

3.4.6. Sprint 6: Pembuatan Dan Pengujian Fitur Aplikasi

Pada *sprint* ini membahas tentang pembuatan fitur – fitur yang ada pada halaman antarmuka aplikasi dari tampilan, kode program, dan pengujian fitur.

1) Halaman *Onboarding*

Berikut hasil pembuatan dan pengujian halaman *onboarding*:

a) Kode Program

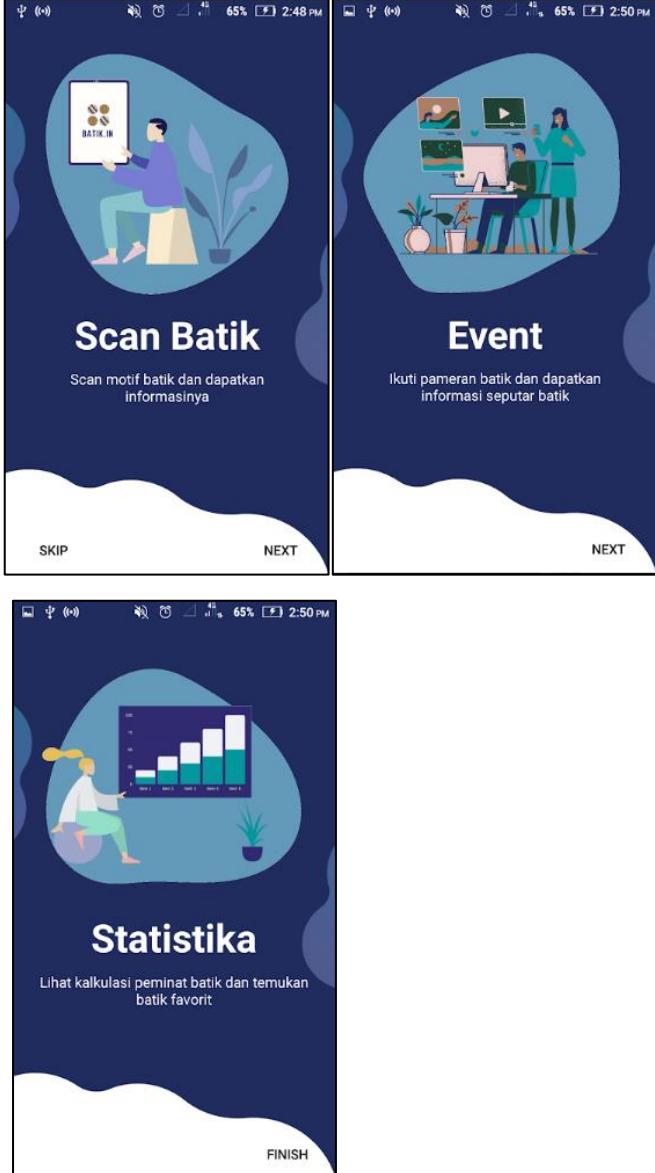
Keterangan kode program sebagai berikut:

(**Lampiran 10 Kode Halaman *Onboarding***)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 14 Pengujian Kompatibilitas Halaman *Onboarding*

Aplikasi Batik.In (Halaman)	Android
<i>Onboarding</i>	
	<p>Keterangan: Tampilan <i>onboarding</i> pada android sesuai dengan desain yang direncanakan yaitu terdapat tiga halaman.</p>

Tabel 3. 15 Pengujian Fungsionalitas Halaman *Onboarding*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Onboarding</i>	Agar pengguna dapat melihat informasi yang ditawarkan oleh aplikasi	Pengguna melakukan instalasi aplikasi dan <i>onboarding</i> akan tampil	Pengguna bisa masuk ke halaman <i>login</i>	✓

Dari hasil pengujian menurut Tabel 3.14 dan Tabel 3.15 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan pada aplikasi *mobile*. Fungsionalitas ketika berpindah halaman juga berjalan dengan baik.

2) Halaman *Splash Screen*

Berikut hasil pembuatan dan pengujian halaman *splash screen*:

a) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 11 Kode Halaman *Splash Screen*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 16 Pengujian Kompatibilitas Halaman *Splash Screen*

Aplikasi Batik.In (Halaman)	Android
<i>Splash Screen</i>	
Keterangan: Tampilan <i>splash</i> pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 17 Pengujian Fungsionalitas Halaman *Splash Screen*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Splash screen</i>	Agar pengguna dapat melihat kesan pertama dari aplikasi	Pengguna membuka aplikasi yang ada pada <i>smartphone</i>	Pengguna bisa masuk ke halaman aplikasi	✓

Dari hasil pengujian menurut Tabel 3.16 dan Tabel 3.17 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan pada aplikasi *mobile*, dan fungsionalitas ketika berpindah halaman juga berjalan dengan baik.

3) Halaman *Login*

Berikut hasil pembuatan dan pengujian halaman *login*:

a) Kode Program

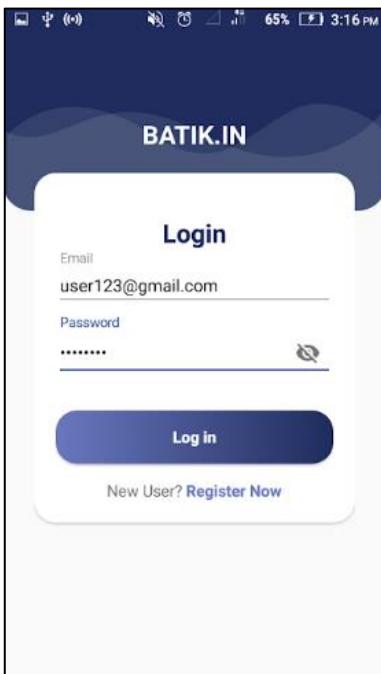
Keterangan kode program sebagai berikut:

(Lampiran 12 Kode Halaman *Login*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 18 Pengujian Kompatibilitas Halaman *Login*

Aplikasi Batik.In (Halaman)	Android
<i>Login</i>	 <p>Keterangan: Tampilan <i>login</i> pengguna pada android sesuai dengan desain yang direncanakan.</p>

Tabel 3. 19 Pengujian Fungsionalitas Halaman *Login*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Login</i>	Agar pengguna dapat masuk ke dalam aplikasi	Pengguna harus memasukkan akun yang sudah terdaftar sebelumnya dengan memasukkan <i>username</i> dan <i>password</i>	Pengguna bisa masuk ke halaman pengguna	✓

Dari hasil pengujian menurut Tabel 3.18 dan Tabel 3.19 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti notifikasi *username* dan *password* kosong juga berjalan dengan baik.

4) Halaman *Register*

Berikut hasil pembuatan dan pengujian halaman *Register*:

a) Kode Program

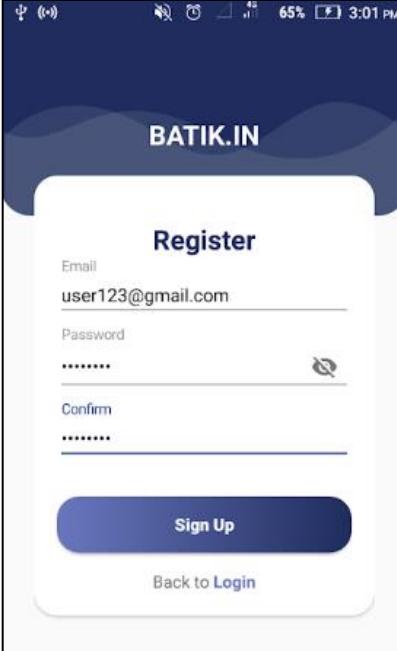
Keterangan kode program sebagai berikut:

(Lampiran 13 Kode Halaman *Register*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 20 Pengujian Kompatibilitas Halaman *Register*

Halaman Aplikasi	Android
<i>Register</i>	

Keterangan: Tampilan registrasi pengguna pada android sesuai dengan desain yang direncanakan.

Tabel 3. 21 Pengujian Fungsionalitas Halaman *Register*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Register</i>	Agar pengguna dapat mendaftarkan akun baru ke aplikasi	Pengguna memasukkan alamat <i>email</i> , <i>password</i> , dan konfirmasi <i>password</i> . Pengguna harus mengklik tombol <i>signup</i>	Pengguna akan mempunyai akun yang bisa digunakan untuk <i>login</i>	✓

Dari hasil pengujian menurut Tabel 3.20 dan Tabel 3.21 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti notifikasi ketika *input* kosong juga bisa ditampilkan.

5) Halaman Pengisian Data Pengguna

Berikut hasil pembuatan dan pengujian halaman pengisian data pengguna:

a) Kode Program

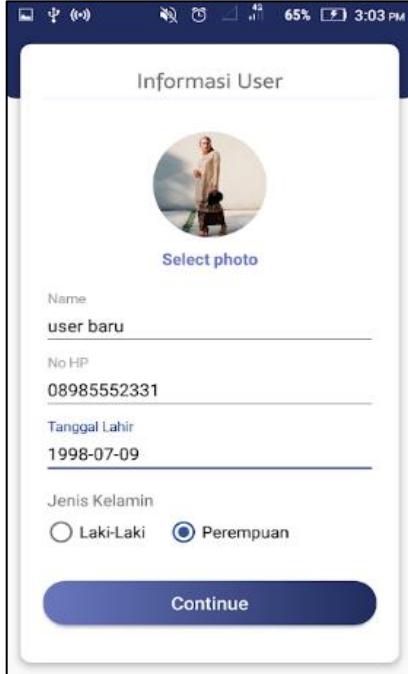
Keterangan kode program sebagai berikut:

(Lampiran 14 Kode Halaman Data Pengguna)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 22 Pengujian Kompatibilitas Halaman Pengisian Data Pengguna

Halaman Aplikasi	Android
Pengisian Data Pengguna	

Keterangan: Tampilan halaman pengisian data pengguna pada android sesuai dengan desain yang direncanakan.

Tabel 3. 23 Pengujian Fungsionalitas Halaman Pengisian Data Pengguna

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Halaman Pengisian Data Pengguna	Agar sistem mendapatkan informasi mengenai pengguna	Pengguna memasukkan foto profil, nama, no hp, tanggal lahir dan jenis kelamin	Pengguna akan bisa masuk ke halaman <i>home</i>	✓

Dari hasil pengujian menurut Tabel 3.22 dan Tabel 3.23 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti memunculkan notifikasi ketika ada inputan yang masih kosong juga berjalan dengan baik.

6) Halaman Informasi *Home*

Berikut hasil pembuatan dan pengujian halaman *home*:

a) Kode Program

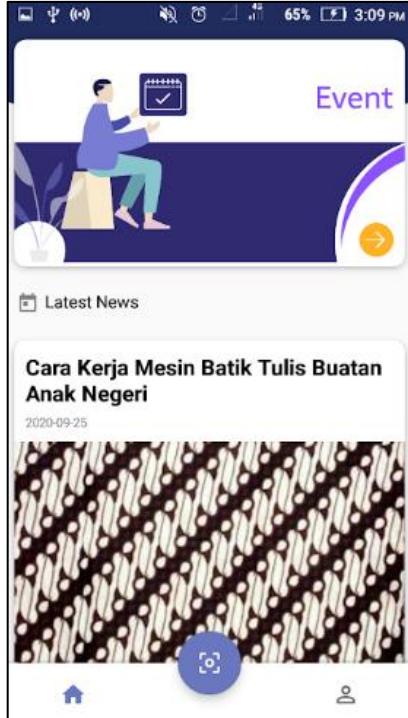
Keterangan kode program sebagai berikut:

(Lampiran 15 Kode Halaman *Home*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 24 Pengujian Kompatibilitas Halaman *Home*

Halaman Aplikasi	Android
<i>Home</i>	 <p>The screenshot shows the home screen of a mobile application. At the top, there is a banner featuring a person sitting at a desk with a laptop, labeled 'Event'. Below the banner, there is a section titled 'Latest News' with an article about a 'Cara Kerja Mesin Batik Tulis Buatan Anak Negeri' from 2020-09-25. The article includes a thumbnail image of a batik pattern. At the bottom of the screen are three navigation icons: a house icon, a magnifying glass icon, and a person icon.</p>
Keterangan: Tampilan halaman <i>home</i> pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 25 Pengujian Fungsionalitas Halaman *Home*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan halaman <i>event</i>	Agar pengguna dapat melihat atau mengakses <i>event</i> yang diselenggarakan	Pengguna mengklik tombol <i>left button</i> pada <i>event card</i>	Pengguna dapat masuk ke halaman <i>event</i>	✓
Menampilkan berita	Agar pengguna dapat melihat berita mengenai	Pengguna melakukan <i>scroll</i> pada	Pengguna dapat melihat berita yang	✓

	<i>event</i> ataupun batik	berita yang ditampilkan dan dapat mengklik <i>view more</i> berita	ditampilkan dan <i>view more</i> deskripsinya	
--	----------------------------	--	---	--

Dari hasil pengujian menurut Tabel 3.24 dan Tabel. 3.25 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat berita dan berpindah halaman ke *event* juga berjalan dengan baik.

7) Halaman *Event* Pameran

Berikut hasil pembuatan dan pengujian halaman *event* pameran:

a) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 16 Kode Halaman *Event* Pameran)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 26 Pengujian Kompatibilitas Halaman *Event* Pameran

Halaman Aplikasi	Android
<i>Event</i> Pameran	
Keterangan: Tampilan halaman <i>event</i> pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 27 Pengujian Fungsionalitas Halaman *Event* Pameran

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan <i>event</i>	Agar pengguna dapat melihat detail dari <i>event</i>	Pengguna dapat melakukan scroll secara vertikal	Pengguna dapat melihat detail dari <i>event</i> seperti nama dari pameran batik, lokasi, tanggal, waktu	✓

			dan penyelenggara pameran	
<i>Like event</i> pameran	Agar pengguna dapat menyukai <i>event</i> pameran yang diselenggarakan	Melakukan klik pada tombol <i>like</i> untuk menyukai <i>event</i>	Tombol <i>like</i> akan berwarna merah jika disukai dan akan berwarna abu – abu jika tidak disukai	✓

Dari hasil pengujian menurut Tabel 3.26 dan Tabel 3.27 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti *scrolldown* blok pameran batik, mengklik tombol *like*, dan melakukan perpindahan antar halaman juga berjalan dengan baik.

8) Halaman Komentar *Event*

Berikut hasil pembuatan dan pengujian halaman komentar *event*:

a) Kode Program

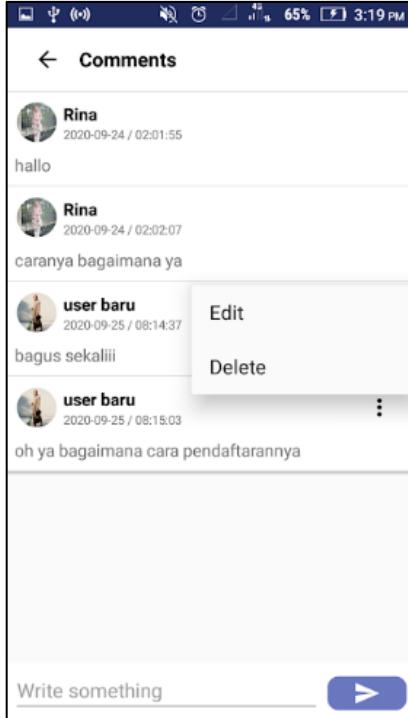
Keterangan kode program sebagai berikut:

(Lampiran 17 Kode Halaman Komentar *Event*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 28 Pengujian Kompatibilitas Halaman Komentar *Event*

Halaman Aplikasi	Android
Komentar <i>Event</i>	 <p>Keterangan: Tampilan halaman komentar <i>event</i> pada android sesuai dengan desain yang direncanakan.</p>

Tabel 3. 29 Pengujian Fungsionalitas Halaman Komentar *Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Komentar <i>Event</i>	Agar pengguna dapat memberikan komentar terhadap <i>event</i> pameran batik	Klik pada <i>textbox</i> yang sudah disediakan sistem, setelah itu tulis komentar dan klik tombol kirim	Komentar akan terkirim dan ditampilkan di halaman komentar	✓

Dari hasil pengujian menurut Tabel 3.28 dan Tabel 3.29 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat komentar, dan mengklik *list button* komentar juga berjalan dengan baik.

9) Halaman Mengedit Komentar Event

Berikut hasil pembuatan dan pengujian halaman mengedit komentar *event*:

a) Kode Program

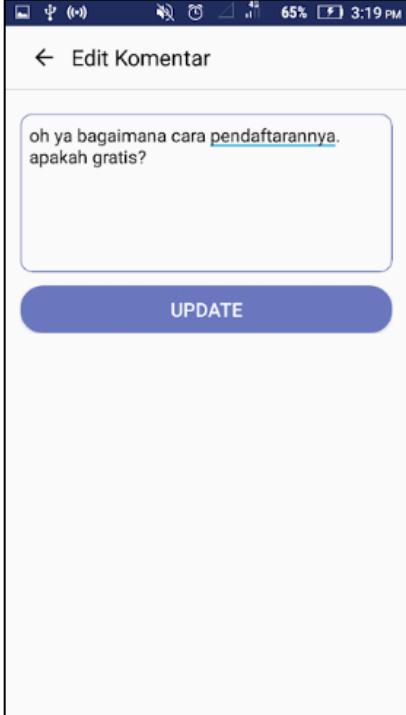
Keterangan kode program sebagai berikut:

(Lampiran 18 Kode Halaman Mengedit Komentar Event)

c) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 30 Pengujian Kompatibilitas Halaman Mengedit Komentar *Event*

Halaman Aplikasi	Android
Mengedit Komentar <i>Event</i>	

Keterangan: Tampilan halaman mengedit komentar *event* pada android sesuai dengan desain yang direncanakan.

Tabel 3. 31 Pengujian Fungsionalitas Halaman Mengedit Komentar *Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Edit</i> Komentar	Agar pengguna dapat mengubah komentar yang sudah dibuatnya	Klik tombol <i>list</i> , pilih <i>edit</i> komentar, menuliskan komentar yang akan diubah	Komentar akan diperbarui dan ditampilkan di halaman komentar	✓

Dari hasil pengujian menurut Tabel 3.30 dan Tabel 3.31 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat komentar, mengubah komentar dan menghapus komentar juga berjalan dengan baik.

10) Halaman Menghapus Komentar *Event*

Berikut hasil pembuatan dan pengujian halaman menghapus komentar *event*:

a) Kode Program

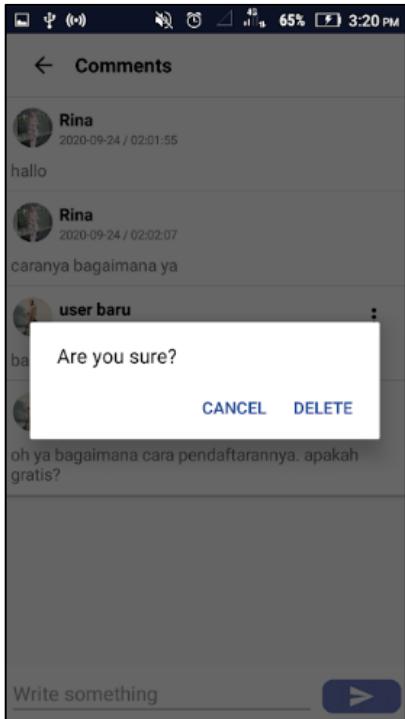
Keterangan kode program sebagai berikut:

(Lampiran 19 Kode Halaman Menghapus Komentar *Event*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 32 Pengujian Kompatibilitas Halaman Menghapus Komentar *Event*

Halaman Aplikasi	Android
Menghapus Komentar <i>Event</i>	
Keterangan: Tampilan halaman komentar <i>event</i> pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 33 Pengujian Fungsionalitas Halaman Menghapus Komentar *Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Hapus Komentar	Agar pengguna dapat menghapus komentar yang sudah dibuat	Klik tombol <i>list</i> , pilih komentar, dan klik tombol <i>delete</i>	Komentar akan dihapus	✓

Dari hasil pengujian menurut Tabel 3.32 dan Tabel 3.33 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti menghapus komentar juga berjalan dengan baik.

11) Halaman *Scan Qrcode Event*

Berikut hasil pembuatan dan pengujian halaman *scan qrcode event*:

a) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 20 Kode Halaman *Scan Qrcode Event*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 34 Pengujian Kompatibilitas Halaman *Scan Qrcode Event*

Halaman Aplikasi	Android
<i>Scan Qrcode Event</i>	
Keterangan: Tampilan halaman <i>scan qrcode event</i> pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 35 Pengujian Fungsionalitas Halaman *Scan Qrcode Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Scan Qrcode</i>	Agar pengguna dapat menscan <i>qrcode event</i> dan mendapatkan akses masuk ke pameran	Memilih tombol <i>scan qrcode</i> dan mengarahkan kamera ke <i>qrcode</i> registrasi	Menampilkan detail dari <i>event</i>	✓

Dari hasil pengujian menurut Tabel 3.34 dan Tabel 3.35 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti mengakses kamera dan menscan *qrcode* juga dapat bekerja dengan baik.

12) Halaman Konfirmasi *Event*

Berikut hasil pembuatan dan pengujian halaman konfirmasi *event*:

a) Kode Program

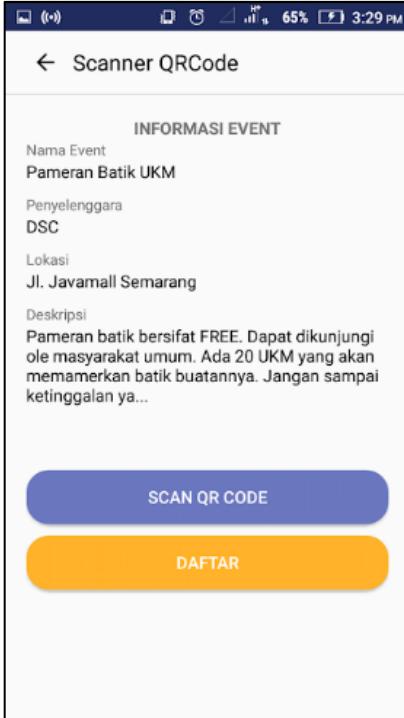
Keterangan kode program sebagai berikut:

(Lampiran 21 Kode Halaman Konfirmasi *Event*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 36 Pengujian Kompatibilitas Halaman Konfirmasi *Event*

Halaman Aplikasi	Android
Konfirmasi <i>Event</i>	
Keterangan: Tampilan halaman konfirmasi <i>event</i> pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 37 Pengujian Fungsionalitas Halaman Konfirmasi *Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Konfirmasi <i>Event</i>	Agar pengguna dapat mengkonfirmasi kehadiran	Pengguna dapat mengklik tombol daftar	Menampilkan notifikasi berhasil mendaftar	✓

Dari hasil pengujian menurut Tabel 3.36 dan Tabel 3.37 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat detail *event* dan menampilkan notifikasi berhasil mendaftar juga bekerja dengan baik.

13) Halaman Deteksi Batik

Berikut hasil pembuatan dan pengujian halaman deteksi batik:

a) Kode Program

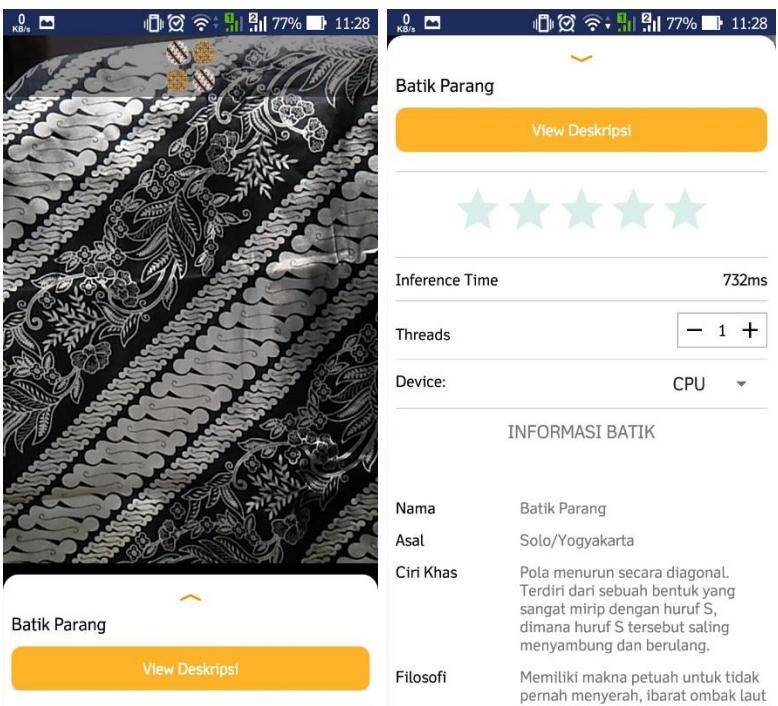
Keterangan kode program sebagai berikut:

(Lampiran 22 Kode Halaman Deteksi Batik)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 38 Pengujian Kompatibilitas Halaman Deteksi Batik

Halaman Aplikasi	Android								
Deteksi Batik	 <p>Deteksi Batik</p> <p>Android</p> <p>Batik Parang</p> <p>Inference Time 732ms</p> <p>Threads - 1 +</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <table> <tr> <td>Nama</td> <td>Batik Parang</td> </tr> <tr> <td>Asal</td> <td>Solo/Yogyakarta</td> </tr> <tr> <td>Ciri Khas</td> <td>Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang.</td> </tr> <tr> <td>Filosofi</td> <td>Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut</td> </tr> </table> <p>Keterangan: Tampilan halaman deteksi batik pada android sesuai dengan desain yang direncanakan.</p>	Nama	Batik Parang	Asal	Solo/Yogyakarta	Ciri Khas	Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang.	Filosofi	Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut
Nama	Batik Parang								
Asal	Solo/Yogyakarta								
Ciri Khas	Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang.								
Filosofi	Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut								

Tabel 3. 39 Pengujian Fungsionalitas Halaman Deteksi Batik

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Deteksi Batik	Untuk mendeteksi batik	Arahkan kamera pada objek batik yang akan dideteksi	Menampilkan label dari batik yang dideteksi secara <i>realtime</i>	✓
Menampilkan Deskripsi	Agar pengguna dapat melihat deskripsi dari motif batik	Klik tombol <i>view</i> deskripsi dan tekan geser keatas tombol panah	Menampilkan deskripsi seperti nama, asal, ciri khas, dan filosofi dari batik	✓
<i>Rating</i>	Agar pengguna dapat memberikan <i>rating</i> pada batik	Klik jumlah <i>rating</i> yang akan diberikan	Menampilkan <i>rating</i> yang diberikan oleh pengguna	✓

Dari hasil pengujian menurut Tabel 3.38 dan Tabel 3.39 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti mendeteksi motif batik, menampilkan label dari motif yang terdeteksi, menampilkan deskripsi dari motif, dan memberikan *rating* pada batik juga bekerja dengan baik.

14) Halaman Profil

Berikut hasil pembuatan dan pengujian halaman profil:

a) Kode Program

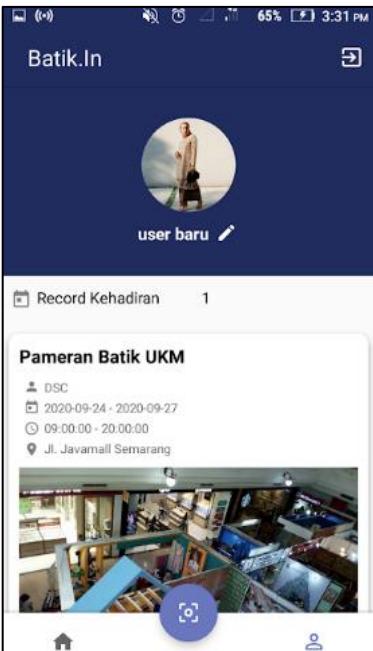
Keterangan kode program sebagai berikut:

(Lampiran 23 Kode Halaman Profil)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 40 Pengujian Kompatibilitas Halaman Profil

Halaman Aplikasi	Android
Profil	
Keterangan: Tampilan halaman profil pada android sesuai dengan desain yang direncanakan.	

Tabel 3. 41 Pengujian Fungsionalitas Halaman Profil

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan Profil	Agar pengguna dapat melihat profil dan melihat histori <i>event</i> yang pernah diikuti	Klik menu profil dan scroll secara vertikal	Menampilkan profil dan histori dari <i>event</i> yang diikuti	✓

<i>Edit Profil</i>	Agar pengguna dapat mengubah informasi data diri	Klik tombol edit, lakukan perubahan sesuai yang diinginkan dan klik simpan	Menampilkan informasi sesuai data yang diperbarui	✓
<i>Logout</i>	Agar pengguna dapat keluar dari aplikasi	Klik tombol logout pada kanan atas profil	Menampilkan konfirmasi <i>logout</i> dan menampilkan halaman <i>login</i>	✓

Dari hasil pengujian menurut Tabel 3.40 dan Tabel 3.41 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat profil, melihat histori *event* yang pernah diikuti, mengubah profil, dan melakukan *logout* juga dapat berkerja dengan baik.

15) Halaman *Login*

Berikut hasil pembuatan dan pengujian halaman *login*:

a) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 24 Kode Halaman *Login*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 42 Pengujian Kompatibilitas Halaman *Login*

Halaman Aplikasi	Website
<i>Login</i>	<p>Keterangan: Tampilan halaman <i>login</i> pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 43 Pengujian Fungsionalitas Halaman *Login*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Login Admin</i>	Agar admin dapat masuk ke halaman selanjutnya	Masukkan <i>username</i> dan <i>password</i>	Menampilkan halaman beranda dari admin	✓
<i>Login Penyelenggara</i>	Agar penyelenggara <i>event</i> dapat masuk ke halaman selanjutnya	Masukkan <i>username</i> dan <i>password</i>	Menampilkan halaman beranda dari penyelenggara	✓

Dari hasil pengujian menurut Tabel 3.42 dan Tabel. 3.43 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti validasi inputan juga bekerja dengan baik.

16) Halaman *Dashboard*

Berikut hasil pembuatan dan pengujian halaman *dashboard*:

a) Kode Program

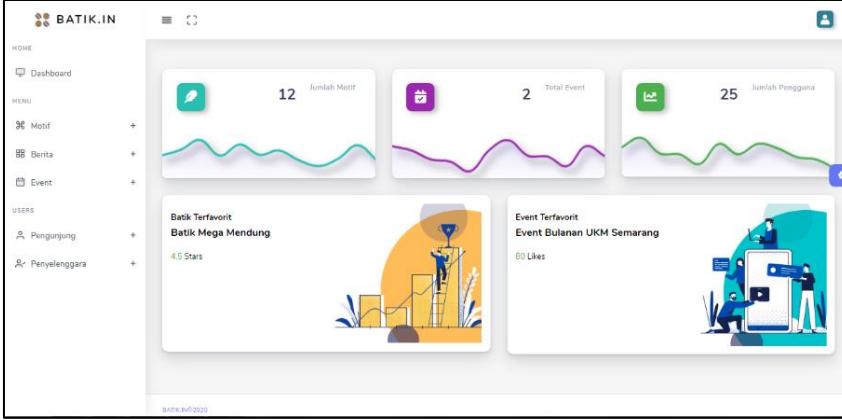
Keterangan kode program sebagai berikut:

(Lampiran 25 Kode Halaman *Dashboard*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 44 Pengujian Kompatibilitas Halaman *Dashboard Admin*

Halaman Aplikasi	Website
<i>Dashboard</i>	 <p>The screenshot shows the website's dashboard with the following data:</p> <ul style="list-style-type: none"> Jumlah Motif: 12 Total Event: 2 Jumlah Pengguna: 25 Batik Terfavorit: Batik Mega Mendung (4.5 Stars) Event Terfavorit: Event Bulanan UKM Semarang (60 Likes)

Keterangan: Tampilan halaman *dashboard* pada website sesuai dengan desain yang direncanakan.

Tabel 3. 45 Pengujian Fungsionalitas Halaman *Dashboard Admin*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Dashboard Admin</i>	Agar admin dapat melihat informasi	Pengguna dapat mengklik menu <i>dashboard</i> pada sidebar	Menampilkan kalkulasi	✓

<i>Dashboard Penyelenggara</i>	Agar penyelenggara dapat melihat informasi	Pengguna dapat mengklik menu <i>dashboard</i> pada <i>sidebar</i>	Menampilkan kalkulasi	✓
--------------------------------	--	---	-----------------------	---

Dari hasil pengujian menurut Tabel 3.44 dan Tabel. 3.45 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti menampilkan kalkulasi jumlah motif batik, jumlah *event* pameran, jumlah pengguna aplikasi, motif terfavorit dan *event* terfavorit juga dapat bekerja dengan baik.

17) Halaman Menampilkan Informasi Motif Batik

Berikut hasil pembuatan dan pengujian halaman informasi motif batik:

a) Kode Program

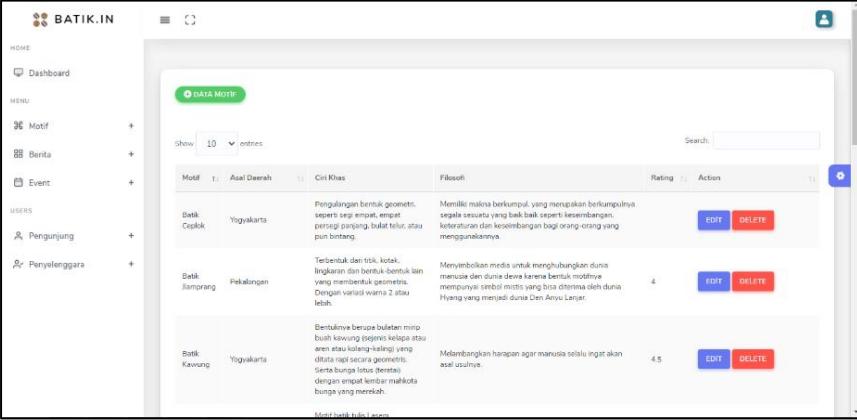
Keterangan kode program sebagai berikut:

(Lampiran 26 Kode Halaman Menampilkan Informasi Motif Batik)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 46 Pengujian Kompatibilitas Menampilkan Informasi Motif Batik

Halaman Aplikasi	Website
Informasi Motif Batik	 <p>Keterangan: Tampilan halaman menampilkan informasi batik pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 47 Pengujian Fungsionalitas Menampilkan Informasi Motif Batik

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan Informasi Motif Batik	Agar admin dapat melihat data dari motif batik	Admin dapat mengklik fitur <i>view motif batik</i>	Menampilkan data dari motif batik	✓
Menghapus Data	Agar admin dapat menghapus data	Klik tombol <i>delete</i>	Menampilkan konfirmasi hapus data dan data akan terhapus	✓

Dari hasil pengujian menurut Tabel 3.46 dan Tabel. 3.47 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti menampilkan nama motif, asal, ciri khas, filosofi dan *rating* juga bekerja dengan baik.

18) Halaman Menambahkan Informasi Motif Batik

Berikut hasil pembuatan dan pengujian halaman menambahkan informasi motif batik:

a) Kode Program

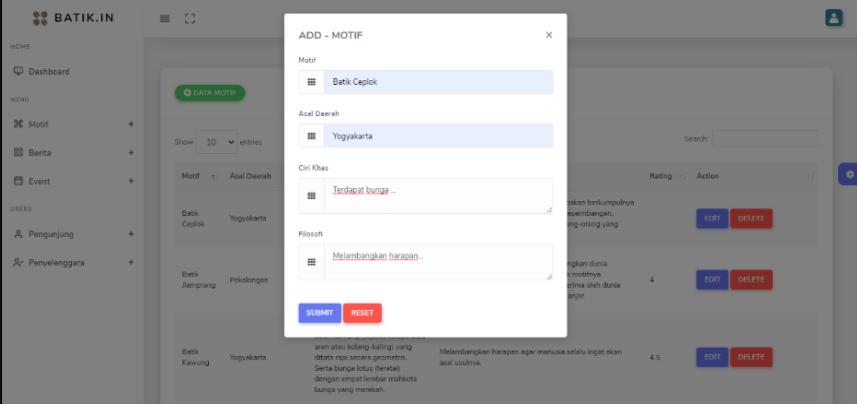
Keterangan kode program sebagai berikut:

(Lampiran 27 Kode Halaman Menambahkan Informasi Motif Batik)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 48 Pengujian Kompatibilitas Menambahkan Informasi Motif Batik

Halaman Aplikasi	Website
Tambah Data Motif Batik	 <p>Keterangan: Tampilan halaman menambahkan motif batik pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 49 Pengujian Fungsionalitas Menambahkan Informasi Motif Batik

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menambahkan Data	Agar admin dapat menambahkan	Mengisi <i>form input</i> seperti motif, asal,	Menampilkan notifikasi dan data akan	✓

	data motif batik	ciri khas dan filosofi	masuk kedalam tabel	
--	------------------	------------------------	---------------------	--

Dari hasil pengujian menurut Tabel 3.48 dan Tabel. 3.49 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti Menampilkan notifikasi dan data akan masuk kedalam tabel juga bekerja dengan baik.

19) Halaman Mengedit Informasi Motif Batik

Berikut hasil pembuatan dan pengujian halaman mengedit informasi batik:

a) Kode Program

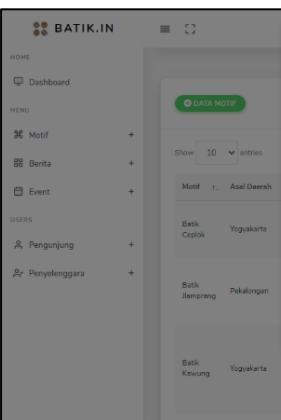
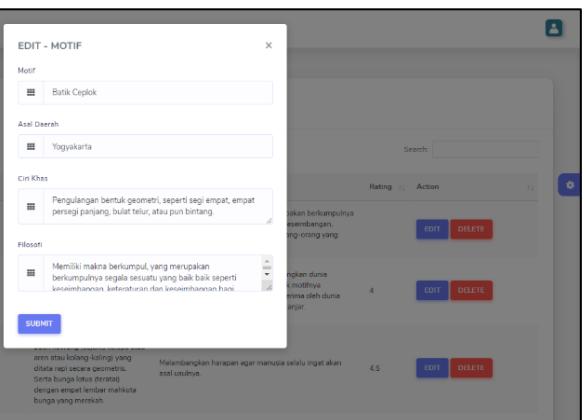
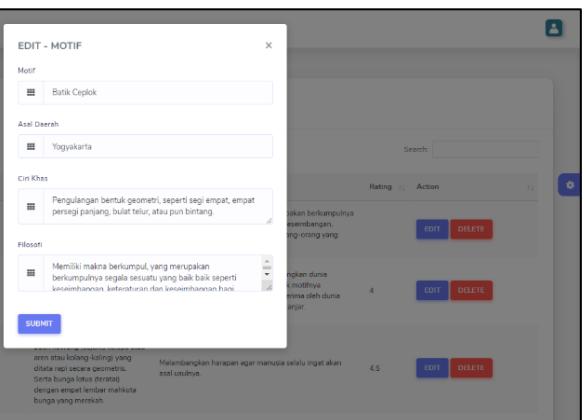
Keterangan kode program sebagai berikut:

(Lampiran 28 Kode Halaman Mengedit Informasi Motif Batik)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 50 Pengujian Kompatibilitas Mengedit Informasi Motif Batik

Halaman Aplikasi	Website
Edit Data Motif Batik  	Website 

Keterangan: Tampilan halaman mengedit motif batik pada website sesuai dengan desain yang direncanakan.

Tabel 3. 51 Pengujian Fungsionalitas Mengedit Informasi Motif Batik

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Mengedit Data	Agar admin dapat mengubah data motif batik	Mengubah isi <i>form input</i> seperti motif, asal, ciri khas dan filosofi	Menampilkan notifikasi dan data akan diperbarui	✓

Dari hasil pengujian menurut Tabel 3.50 dan Tabel. 3.51 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti mengubah data dan menampilkan informasi *update* juga bekerja dengan baik.

20) Halaman Menampilkan Berita

Berikut hasil pembuatan dan pengujian halaman menampilkan berita:

a) Kode Program

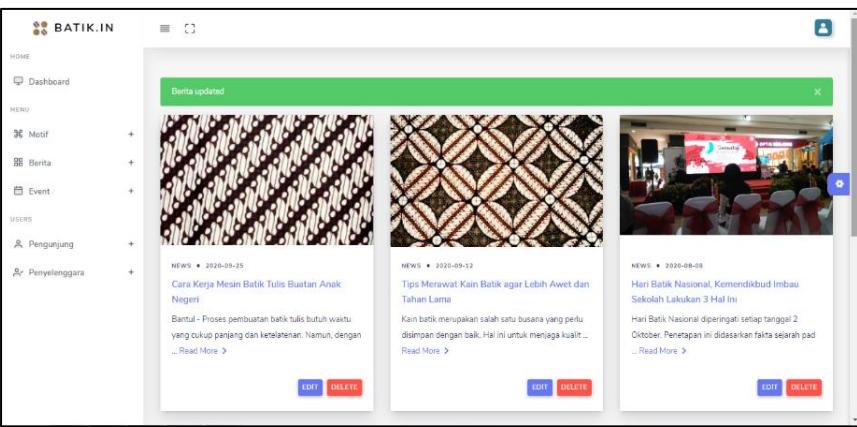
Keterangan kode program sebagai berikut:

(Lampiran 29 Kode Halaman Menampilkan Berita)

c) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 52 Pengujian Kompatibilitas Halaman Menampilkan Berita

Halaman Aplikasi	Website
Tampil Berita	 <p>Keterangan: Tampilan halaman menampilkan berita pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 53 Pengujian Fungsionalitas Halaman Menampilkan Berita

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan Berita (Admin)	Agar admin dapat memantau berita yang diinputkan	Admin dapat mengklik fitur <i>view berita</i>	Menampilkan berita didalam tampilan blok	✓
Menghapus Berita (Admin)	Agar admin dapat menghapus data berita	Klik tombol <i>delete</i>	Menampilkan konfirmasi hapus data dan data akan terhapus	✓
Menampilkan Berita (Penyelenggara)	Agar penyelenggara dapat melihat berita	penyelenggara dapat mengklik fitur <i>view berita</i>	Menampilkan berita	✓

Dari hasil pengujian menurut Tabel 3.52 dan Tabel. 3.53 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat berita didalam *cardview*, menghapus berita juga dapat berkerja dengan baik.

21) Halaman Menambahkan Berita

Berikut hasil pembuatan dan pengujian halaman menambahkan berita:

a) Kode Program

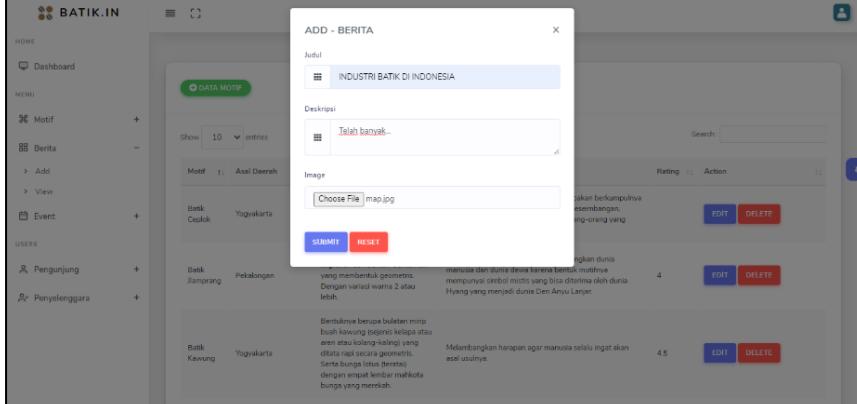
Keterangan kode program sebagai berikut:

(Lampiran 30 Kode Halaman Menambahkan Berita)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 54 Pengujian Kompatibilitas Halaman Menambahkan Berita

Halaman Aplikasi	Website
Tambah Berita	 <p>Keterangan: Tampilan halaman tambah berita pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 55 Pengujian Fungsionalitas Halaman Menambahkan Berita

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menambahkan Data	Agar admin dapat menambahkan data berita	Mengisi <i>form input</i> seperti judul berita, deskripsi dan foto	Menampilkan notifikasi dan data akan masuk kedalam <i>cardview</i>	✓

Dari hasil pengujian menurut Tabel 3.54 dan Tabel. 3.55 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti Menampilkan notifikasi dan data akan masuk kedalam tabel berbentuk blok juga bekerja dengan baik.

22) Halaman Mengedit Berita

Berikut hasil pembuatan dan pengujian halaman mengedit berita:

a) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 31 Kode Halaman Mengedit Berita)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 56 Pengujian Kompatibilitas Halaman Mengedit Berita

Halaman Aplikasi	Website
Edit Data Berita	<p>Keterangan: Tampilan halaman mengedit berita pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 57 Pengujian Fungsionalitas Halaman Mengedit Berita

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Mengedit Data	Agar admin dapat mengubah data berita	Mengubah isi <i>form input</i> seperti judul, deskripsi dan foto berita	Menampilkan notifikasi dan data berita akan diperbarui	✓

Dari hasil pengujian menurut Tabel 3.56 dan Tabel. 3.57 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti mengubah data dan menampilkan informasi *update* juga bekerja dengan baik.

23) Halaman Menampilkan All Event Pameran

Berikut hasil pembuatan dan pengujian halaman menampilkan *all event* pameran

a) Kode Program

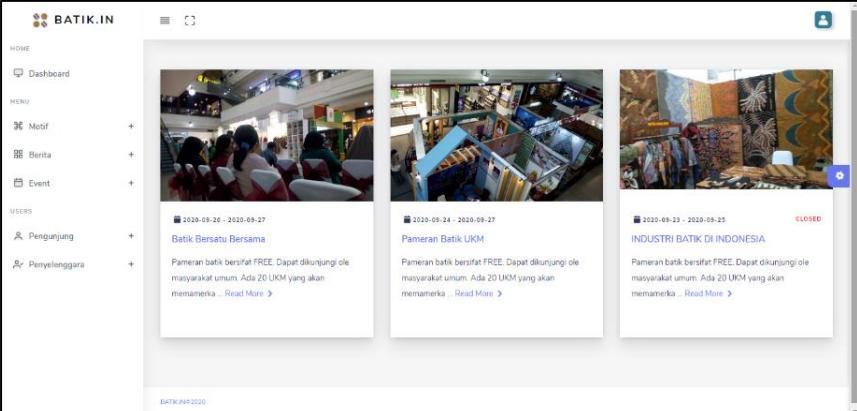
Keterangan kode program sebagai berikut:

(Lampiran 32 Kode Halaman Menampilkan All Event Pameran)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 58 Pengujian Kompatibilitas Menampilkan *All Event* Pameran

Halaman Aplikasi	Website
<i>All Event</i>	

Keterangan: Tampilan halaman *all event* pada website sesuai dengan desain yang direncanakan.

Tabel 3. 59 Pengujian Fungsionalitas Menampilkan *All Event* Pameran

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan <i>all event</i>	Agar admin maupun penyelenggara dapat melihat semua <i>event</i> yang sudah terdaftar	Klik fitur <i>view all event</i>	Menampilkan <i>event</i> kedalam blok	✓
<i>Readmore</i>	Agar admin maupun penyelenggara dapat melihat	Klik tulisan <i>readmore</i> pada <i>event</i>	Menampilkan detail dari <i>event</i>	✓

	informasi <i>event</i> dengan lengkap			
--	--	--	--	--

Dari hasil pengujian menurut Tabel 3.58 dan Tabel. 3.59 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti menampilkan semua *event* yang sudah terdaftar kedalam *cardview* dan menampilkan detail dari *event* juga bekerja dengan baik.

24) Halaman Menampilkan Detail *Event*

Berikut hasil pembuatan dan pengujian halaman menampilkan detail *event*:

a) Kode Program

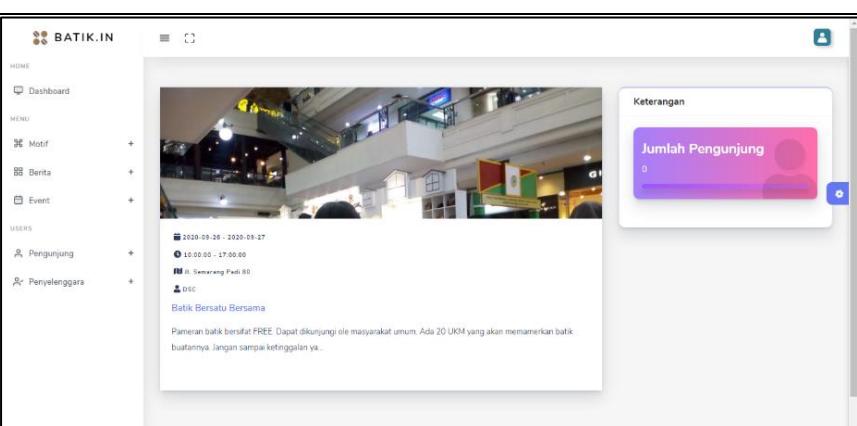
Keterangan kode program sebagai berikut:

(Lampiran 33 Kode Halaman Menampilkan Detail *Event*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 60 Pengujian Kompatibilitas Halaman Menampilkan Detail *Event*

Halaman Aplikasi	Website
Detail <i>Event</i>	 <p>Keterangan: Tampilan halaman detail <i>event</i> pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 61 Pengujian Fungsionalitas Halaman Menampilkan Detail *Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan Detail <i>Event</i>	Agar admin maupun penyelenggara dapat melihat dari <i>event</i>	Klik link <i>readmore</i>	Menampilkan detail dari <i>event</i>	✓
Kalkulasi Pengunjung	Agar admin maupun penyelenggara dapat melihat total pengunjung yang datang pada pameran	Klik link <i>readmore</i>	Menampilkan hasil dari kalkulasi pengunjung yang datang di pameran	✓

Dari hasil pengujian menurut Tabel 3.60 dan Tabel. 3.61 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat detail event dan menampilkan jumlah pengunjung yang datang ke pameran juga bekerja dengan baik.

25) Halaman Menampilkan Konfirmasi *Event*

Berikut hasil pembuatan dan pengujian halaman menampilkan konfirmasi *event*:

a) Kode Program

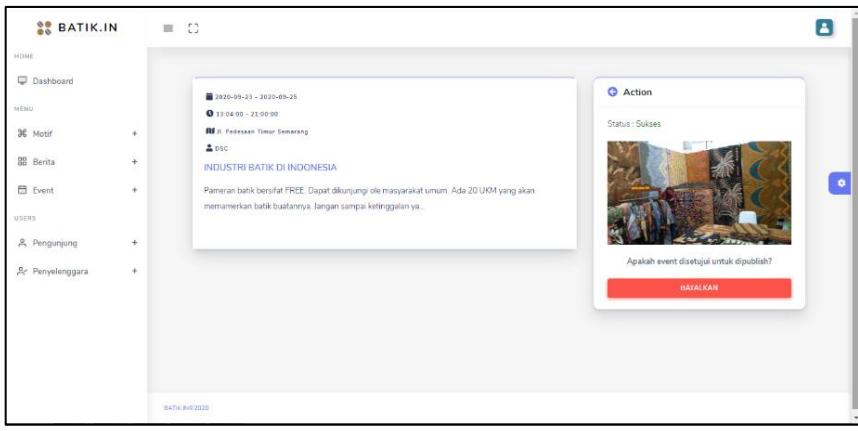
Keterangan kode program sebagai berikut:

(Lampiran 34 Kode Halaman Menampilkan Konfirmasi *Event*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 62 Pengujian Kompatibilitas Menampilkan Konfirmasi *Event*

Halaman Aplikasi	Website
Konfirmasi <i>Event</i>	 <p>Keterangan: Tampilan halaman konfirmasi <i>event</i> pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 63 Pengujian Fungsionalitas Menampilkan Konfirmasi *Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Konfirmasi <i>Event</i>	Agar admin dapat mengonfirmasi atau membatalkan <i>event</i> yang dibuat oleh penyelenggara	Admin mengklik tombol yang terletak pada <i>card action</i>	Jika admin setuju dengan <i>event</i> maka <i>event</i> akan tampil di halaman <i>all event</i> , sedangkan jika tidak disetujui maka <i>event</i> akan gagal dipublikasikan	✓

Dari hasil pengujian menurut Tabel 3.62 dan Tabel. 3.63 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas mengonfirmasi atau membatalkan *event* juga bekerja dengan baik.

26) Halaman Menampilkan Data Pengunjung

Berikut hasil pembuatan dan pengujian halaman menampilkan data pengunjung:

a) Kode Program

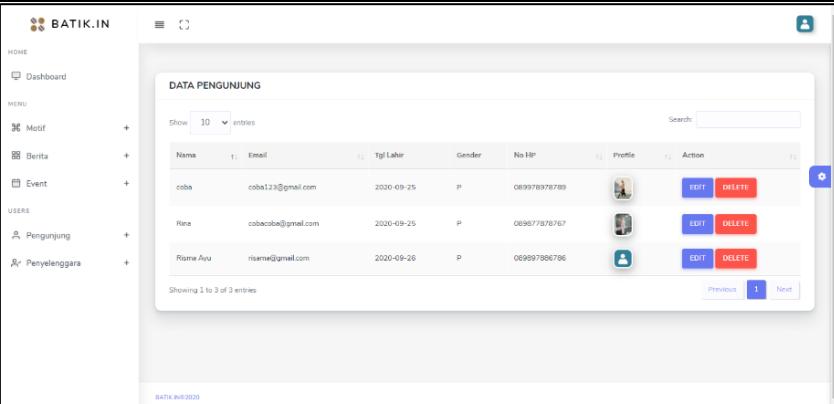
Keterangan kode program sebagai berikut:

(Lampiran 35 Kode Halaman Menampilkan Data Pengunjung)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 64 Pengujian Kompatibilitas Menampilkan Data Pengunjung

Halaman Aplikasi	Website
Data Pengunjung	 <p>Keterangan: Tampilan halaman menampilkan data pengunjung pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 65 Pengujian Fungsionalitas Menampilkan Data Pengunjung

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan Data Pengunjung	Agar admin dapat melihat data dari pengunjung	Admin dapat mengklik fitur <i>view</i> pengunjung	Menampilkan data dari pengunjung	✓
Menghapus Data	Agar admin dapat menghapus data	Klik tombol <i>delete</i>	Menampilkan konfirmasi hapus data dan data akan terhapus	✓

Dari hasil pengujian menurut Tabel 3.64 dan Tabel. 3.65 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat data pengunjung didalam tabel, dan menghapus data juga dapat berkerja dengan baik.

27) Halaman Menambahkan Data Pengunjung

Berikut hasil pembuatan dan pengujian halaman menambahkan data pengunjung:

a) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 36 Kode Halaman Menambahkan Data Pengunjung)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 66 Pengujian Kompatibilitas Menambahkan Data Pengunjung

Halaman Aplikasi	Website
Tambah Data Pengunjung	

Keterangan: Tampilan halaman tambah data pengunjung pada website sesuai dengan desain yang direncanakan.

Tabel 3. 67 Pengujian Fungsionalitas Menambahkan Data Pengunjung

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menambahkan Data	Agar admin dapat menambahkan data pengunjung	Mengisi <i>form input</i> seperti nama, alamat email, password, tanggal lahir, jenis kelamin, no hp, dan foto	Menampilkan notifikasi dan data akan masuk kedalam <i>tabel</i>	✓

Dari hasil pengujian menurut Tabel 3.66 dan Tabel. 3.67 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti terdapat validasi inputan, menampilkan notifikasi dan data akan masuk kedalam tabel juga bekerja dengan baik.

28) Halaman Mengedit Data Pengunjung

Berikut hasil pembuatan dan pengujian halaman mengedit data pengunjung:

a) Kode Program

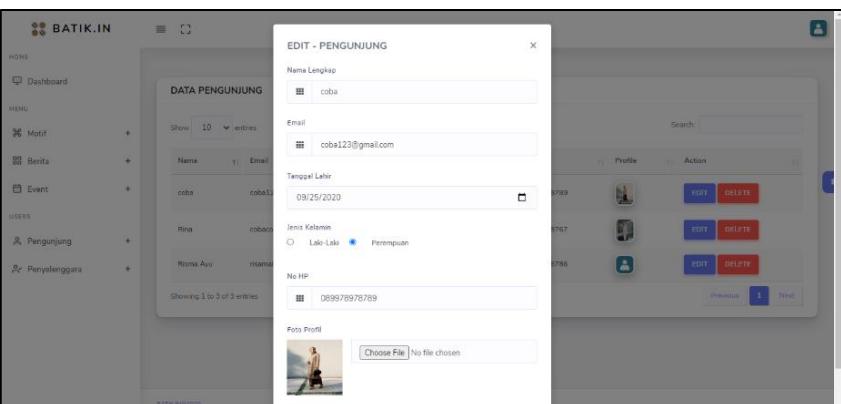
Keterangan kode program sebagai berikut:

(Lampiran 37 Kode Halaman Mengedit Data Pengunjung)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 68 Pengujian Kompatibilitas Halaman Mengedit Data Pengunjung

Halaman Aplikasi	Website
Edit Data Pengunjung	 <p>Keterangan: Tampilan halaman mengedit data pengunjung pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 69 Pengujian Fungsionalitas Halaman Mengedit Data Pengunjung

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Mengedit Data	Agar admin dapat mengubah data pengunjung	Mengubah isi <i>form input</i> seperti nama, alamat <i>email</i> , <i>password</i> ,	Menampilkan notifikasi dan data pengunjung	✓

		tanggal lahir, jenis kelamin, no hp, dan foto	akan diperbarui	
--	--	--	--------------------	--

Dari hasil pengujian menurut Tabel 3.68 dan Tabel. 3.69 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti mengubah data dan menampilkan informasi *update* juga bekerja dengan baik.

29) Halaman Menampilkan Data Penyelenggara

Berikut hasil pembuatan dan pengujian halaman menampilkan data penyelenggara:

a) Kode Program

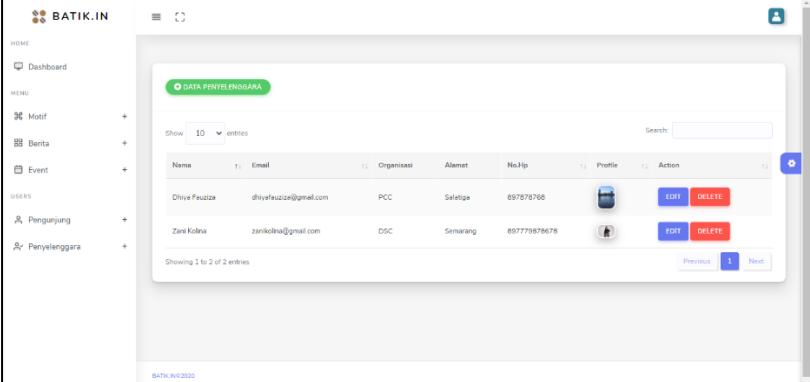
Keterangan kode program sebagai berikut:

(Lampiran 38 Kode Halaman Menampilkan Data Penyelenggara)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 70 Pengujian Kompatibilitas Menampilkan Data Penyelenggara

Halaman Aplikasi	Website
Data Penyelnggara	 <p>Keterangan: Tampilan halaman menampilkan data penyelenggara pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 71 Pengujian Fungsionalitas Menampilkan Data Penyelenggara

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan Data Penyelenggara	Agar admin dapat melihat data penyelenggara	Admin dapat mengklik fitur <i>view data</i> penyelenggara	Menampilkan data dari penyelenggara	✓
Menghapus Data	Agar admin dapat menghapus data	Klik tombol <i>delete</i>	Menampilkan konfirmasi hapus data dan data akan terhapus	✓

Dari hasil pengujian menurut Tabel 3.70 dan Tabel. 3.71 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat data penyelenggara didalam tabel, dan menghapus data juga dapat berkerja dengan baik.

30) Halaman Menambahkan Data Penyelenggara

Berikut hasil pembuatan dan pengujian halaman menambahkan data penyelenggara

a) Kode Program

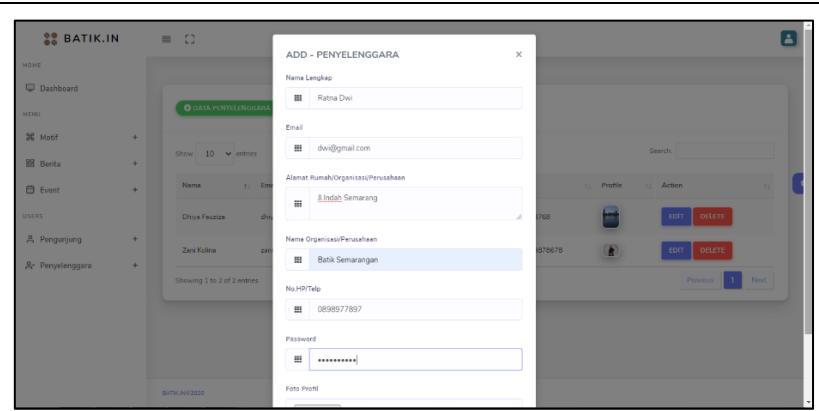
Keterangan kode program sebagai berikut:

(Lampiran 39 Kode Halaman Menambahkan Data Penyelenggara)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 72 Pengujian Kompatibilitas Menambahkan Data Penyelenggara

Halaman Aplikasi	Website
Tambah Data Penyelenggara	 <p>Keterangan: Tampilan halaman tambah data penyelenggara pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 73 Pengujian Fungsionalitas Menambahkan Data Penyelenggara

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menambahkan Data	Agar admin dapat menambahkan data penyelenggara	Mengisi <i>form input</i> seperti nama, alamat <i>email</i> , <i>password</i> , organisasi, no hp, dan foto	Menampilkan notifikasi dan data akan masuk kedalam tabel	✓

Dari hasil pengujian menurut Tabel 3.72 dan Tabel. 3.73 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti terdapat validasi inputan, menampilkan notifikasi dan data akan masuk kedalam tabel juga bekerja dengan baik.

31) Halaman Mengedit Data Penyelenggara

Berikut hasil pembuatan dan pengujian halaman mengedit data penyelenggara:

a) Kode Program

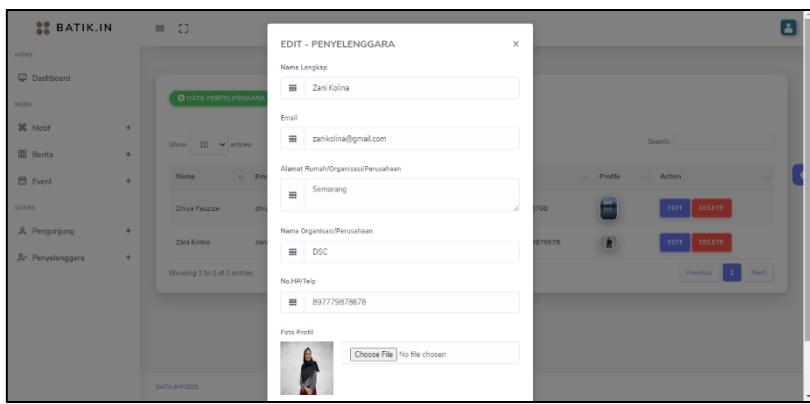
Keterangan kode program sebagai berikut:

(Lampiran 40 Kode Halaman Mengedit Data Penyelenggara)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 74 Pengujian Kompatibilitas Halaman Mengedit Data Penyelenggara

Halaman Aplikasi	Website
Edit Data Penyelenggara	 <p>Keterangan: Tampilan halaman mengedit data penyelenggara pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 75 Pengujian Fungsionalitas Halaman Mengedit Data Penyelenggara

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Mengedit Data	Agar admin dapat mengubah data penyelenggara	Mengubah isi <i>form input</i> nama, alamat <i>email</i> , <i>password</i> ,	Menampilkan notifikasi dan data penyelenggara	✓

		organisasi, no hp, dan foto	akan diperbarui	
--	--	--------------------------------	--------------------	--

Dari hasil pengujian menurut Tabel 3.74 dan Tabel. 3.75 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti mengubah data dan menampilkan informasi *update* juga bekerja dengan baik.

32) Halaman Profil

Berikut hasil pembuatan dan pengujian halaman profil:

a) Kode Program

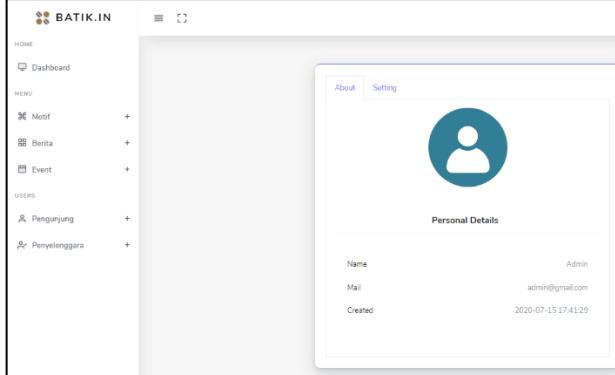
Keterangan kode program sebagai berikut:

(Lampiran 41 Kode Halaman Profil)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 76 Pengujian Kompatibilitas Halaman Profil

Halaman Aplikasi	Website
Profil	 <p>Keterangan: Tampilan halaman profil pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 77 Pengujian Fungsionalitas Halaman Profil

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Lihat Profil	Agar pengguna maupun penyelenggara dapat melihat profil masing – masing	Klik tombol profil pada logo dikanan atas	Menampilkan informasi profil	✓
Setting	Agar pengguna maupun penyelenggara dapat mengubah informasi profil	Klik menu <i>setting</i> pada profil, lakukan pengubahan data	Data profil akan diperbarui	✓

Dari hasil pengujian menurut Tabel 3.76 dan Tabel. 3.77 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat profil serta memperbarui data profil juga bekerja dengan baik.

33) Halaman Menambahkan *Event*

Berikut hasil pembuatan dan pengujian halaman menambahkan *event*:

a) Kode Program

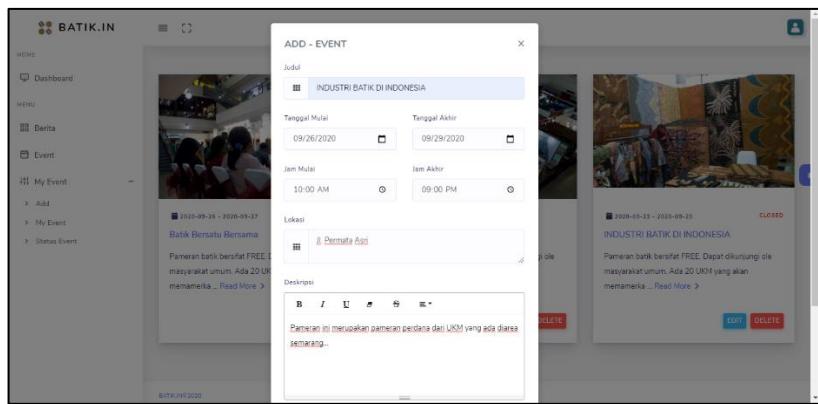
Keterangan kode program sebagai berikut:

(Lampiran 42 Kode Halaman Menambahkan *Event*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 78 Pengujian Kompatibilitas Halaman Menambahkan *Event*

Halaman Aplikasi	Website
Tambah <i>Event</i>	 <p>Keterangan: Tampilan halaman tambah <i>event</i> pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 79 Pengujian Fungsionalitas Halaman Menambahkan *Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menambahkan Data	Agar penyelenggara dapat menambahkan <i>event</i>	Mengisi form <i>input</i> seperti nama <i>event</i> , tanggal, waktu, lokasi, deskripsi dan foto.	Menampilkan notifikasi dan data akan masuk kedalam <i>waiting list</i> konfirmasi <i>event</i>	✓

Dari hasil pengujian menurut Tabel 3.78 dan Tabel. 3.79 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti terdapat validasi inputan, menampilkan notifikasi dan data akan masuk kedalam *waiting list* konfirmasi *event* juga bekerja dengan baik.

34) Halaman Mengedit Event

Berikut hasil pembuatan dan pengujian halaman mengedit event:

a) Kode Program

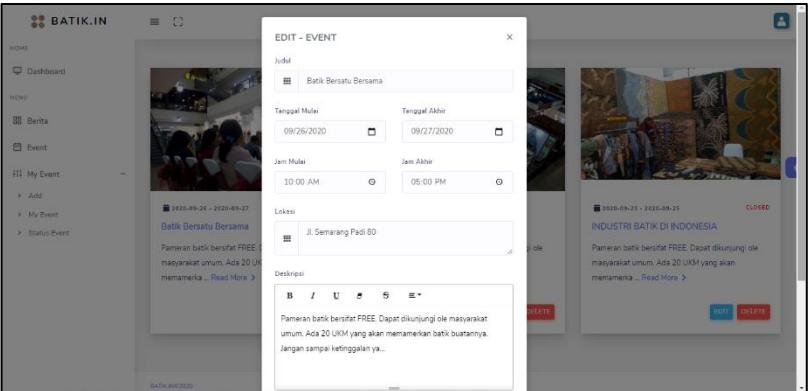
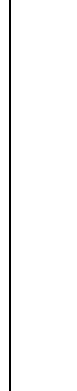
Keterangan kode program sebagai berikut:

(Lampiran 43 Kode Halaman Mengedit Event)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 80 Pengujian Kompatibilitas Halaman Mengedit Event

Halaman Aplikasi	Website
<i>Edit Event</i>	 

Keterangan: Tampilan halaman mengedit *event* pada website sesuai dengan desain yang direncanakan.

Tabel 3. 81 Pengujian Fungsionalitas Halaman Mengedit Event

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Mengedit Data	Agar penyelenggara dapat mengubah informasi <i>event</i>	Mengubah isi <i>form input</i> nama <i>event</i> , tanggal, waktu, lokasi, deskripsi, foto.	Menampilkan notifikasi dan data berita akan diperbarui	✓

Dari hasil pengujian menurut Tabel 3.80 dan Tabel. 3.81 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti mengubah data dan menampilkan informasi *update* juga bekerja dengan baik.

35) Halaman Menampilkan *My Event*

Berikut hasil pembuatan dan pengujian halaman menampilkan *my event*:

a) Kode Program

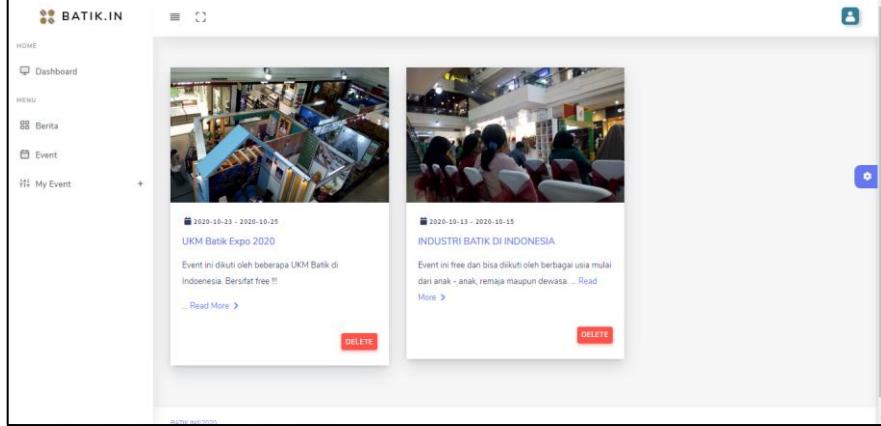
Keterangan kode program sebagai berikut:

(Lampiran 44 Kode Halaman Menampilkan *My Event*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 82 Pengujian Kompatibilitas Halaman Menampilkan *My Event*

Halaman Aplikasi	Website
<i>My Event</i>	 <p>Keterangan: Tampilan halaman <i>my event</i> pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 83 Pengujian Fungsionalitas Halaman Menampilkan *My Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan <i>My Event</i>	Agar penyelenggara dapat melihat <i>event</i> yang sudah dibuat	Penyelenggara dapat mengklik fitur <i>view my event</i>	Menampilkan <i>event</i> kedalam <i>cardview</i>	✓
Menghapus Data	Agar penyelenggara dapat menghapus data	Klik tombol <i>delete</i>	Menampilkan konfirmasi hapus data dan data akan terhapus	✓

Dari hasil pengujian menurut Tabel 3.82 dan Tabel. 3.83 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat *event* yang sudah didaftarkan dan sudah diverifikasi oleh admin juga bekerja dengan baik.

36) Halaman Menampilkan *Qrcode*

Berikut hasil pembuatan dan pengujian halaman menampilkan *qrcode*:

a) Kode Program

Keterangan kode program sebagai berikut:

(Lampiran 45 Kode Halaman Menampilkan *Qrcode*)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform android* berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 84 Pengujian Kompatibilitas Halaman Menampilkan *Qrcode*

Halaman Aplikasi	Website
<i>Qrcode</i>	 A screenshot of a website interface. On the left, there's a sidebar with a logo, navigation links for 'HOME' (Dashboard), 'MENU' (Berita, Event, My Event), and a '+' button. The main content area has a title 'QRCode' and displays a large QR code.
Keterangan: Tampilan halaman <i>qrcode event</i> pada website sesuai dengan desain yang direncanakan.	

Tabel 3. 85 Pengujian Fungsionalitas Halaman Menampilkan *Qrcode*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan <i>Qrcode</i>	Agar penyelenggara dapat melihat <i>qrcode</i> dari event	Pada bagian detail <i>event</i> dapat mengklik fitur <i>view qrcode</i>	<i>Qrcode</i> akan ditampilkan	✓
Mencetak <i>Qrcode</i>	Agar penyelenggara dapat mencetak <i>qrcode</i> kedalam bentuk kertas	Pada bagian detail <i>event</i> dapat mengklik fitur <i>print qrcode</i>	<i>Qrcode</i> akan ditampilkan dalam bentuk <i>print preview</i>	✓

Dari hasil pengujian menurut Tabel 3.84 dan Tabel. 3.85 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat *qrcode* dalam bentuk *view* maupun *print* juga bekerja dengan baik.

37) Halaman Menampilkan Buku Tamu

Berikut hasil pembuatan dan pengujian halaman menampilkan buku tamu:

a) Kode Program

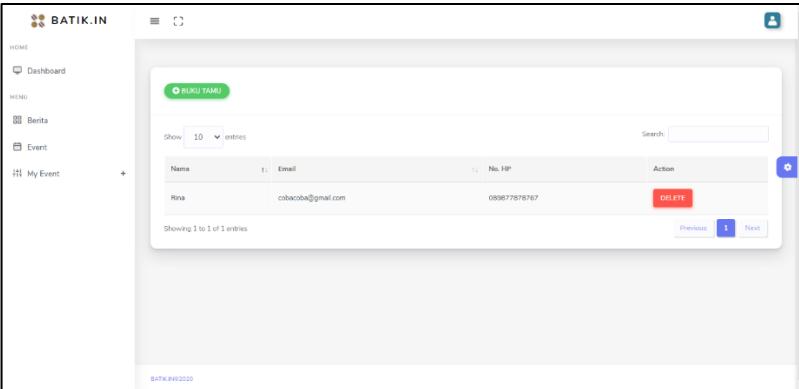
Keterangan Kode Program sebagai berikut:

(Lampiran 46 Kode Halaman Menampilkan Buku Tamu)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 86 Pengujian Kompatibilitas Halaman Menampilkan Buku Tamu

Halaman Aplikasi	Website
Buku Tamu	 <p>Keterangan: Tampilan halaman buku tamu pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 87 Pengujian Fungsionalitas Halaman Menampilkan Buku Tamu

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menampilkan Buku Tamu	Agar penyelenggara dapat melihat data pengunjung	Penyelenggara dapat mengklik tombol buku	Menampilkan pengunjung yang sudah hadir dengan menscan	✓

	yang sudah datang ke pameran	tamu pada event	qrcode pada aplikasi android	
--	------------------------------	-----------------	------------------------------	--

Dari hasil pengujian menurut Tabel 3.86 dan Tabel. 3.87 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti melihat pengunjung yang sudah hadir melalui *scan qrcode* juga bekerja dengan baik.

38) Halaman Menambahkan Kehadiran Pengunjung

Berikut hasil pembuatan dan pengujian halaman menambahkan kehadiran pengunjung:

a) Kode Program

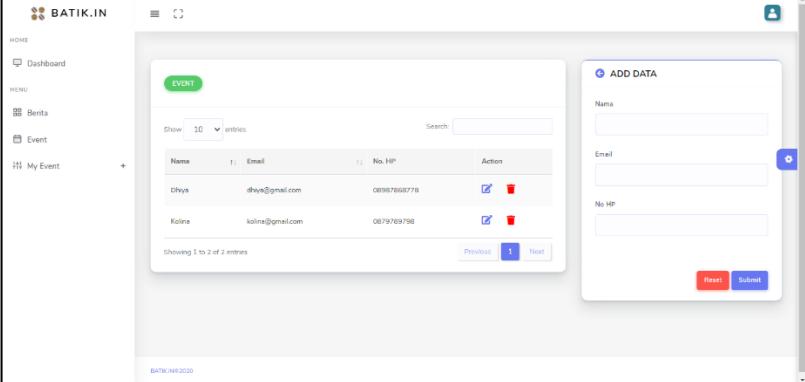
Keterangan kode program sebagai berikut:

(Lampiran 47 Kode Halaman Menambahkan Kehadiran Pengunjung)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 88 Pengujian Kompatibilitas Menambahkan Kehadiran Pengunjung

Halaman Aplikasi	Website
Tambah Kehadiran Pengunjung	 <p>The screenshot shows a website interface for managing events. On the left, there's a sidebar with 'HOME', 'Dashboard', 'Berita', 'Event', and 'My Event'. The main area displays a table titled 'EVENT' with two entries: 'Dhyas' and 'Koline'. Each entry has columns for Name, Email, No HP, and Action (checkboxes). To the right, a modal window titled 'ADD DATA' is open, containing fields for 'Nama', 'Email', and 'No HP', along with 'Reset' and 'Submit' buttons.</p> <p>Keterangan: Tampilan halaman tambah kehadiran pengunjung pada website sesuai dengan desain yang direncanakan.</p>

Tabel 3. 89 Pengujian Fungsionalitas Menambahkan Kehadiran Pengunjung

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Menambahkan Data	Agar penyelenggara dapat menambahkan pengunjung ke buku tamu	Masukkan nama, alamat <i>email</i> , dan no hp kedalam <i>form</i>	Menampilkan data yang sudah diinput kedalam tabel	✓
Mengedit Data	Agar penyelenggara dapat mengubah data pengunjung jika terdapat kesalahan saat penginputan	Masukkan nama, alamat <i>email</i> , dan no hp yang akan diubah	Data akan diperbarui dan ditampilkan kedalam tabel	✓
Menghapus Data	Agar penyelenggara dapat menghapus data	Klik tombol <i>delete</i>	Data akan terhapus	✓

Dari hasil pengujian menurut Tabel 3.88 dan Tabel. 3.89 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti menambahkan, mengedit, menghapus, dan validasi *input* juga bekerja dengan baik.

39) Halaman Melihat Konfirmasi Event

Berikut hasil pembuatan dan pengujian halaman melihat konfirmasi *event*:

a) Kode Program

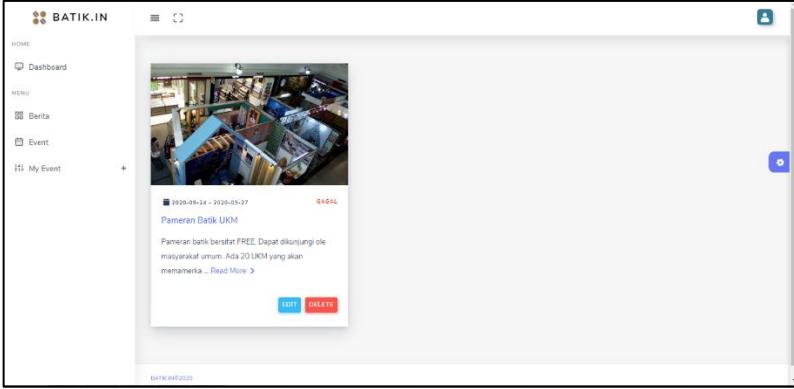
Keterangan kode program sebagai berikut:

(Lampiran 48 Kode Halaman Melihat Konfirmasi Event)

b) Pengujian

Pengujian dilakukan berdasarkan tampilan dari *platform* android berdasarkan fungsi yang ada didalam halaman tersebut. Hasil pengujian sebagai berikut:

Tabel 3. 90 Pengujian Kompatibilitas Halaman Melihat Konfirmasi *Event*

Halaman Aplikasi	Website
Status Konfirmasi <i>Event</i>	 <p>Keterangan: Tampilan halaman status konfirmasi <i>event</i> pada android sesuai dengan desain yang direncanakan.</p>

Tabel 3. 91 Pengujian Fungsionalitas Halaman Melihat Konfirmasi *Event*

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Melihat status konfirmasi <i>event</i>	Agar penyelenggara dapat melihat status <i>event</i> apakah sudah terkonfirmasi	Pilih menu status konfirmasi <i>event</i>	Menampilkan status gagal jika <i>event</i> tidak disetujui dan menampilkan <i>waiting</i> jika <i>event</i> belum dicek oleh admin	✓

Dari hasil pengujian menurut Tabel 3.90 dan Tabel. 3.91 didapatkan hasil bahwa tampilan bekerja dengan baik ketika dijalankan dan fungsionalitas seperti menampilkan status konfirmasi juga bekerja dengan baik.

3.4.7. Sprint 7: Perancangan Uji Kepuasan Pengguna

Pada tahapan ini dilakukan proses pengujian aplikasi kepada pengunjung pameran dan penyelenggara *event* pameran. Dalam pengujian aplikasi ini didapatkan data dengan cara memberikan beberapa poin pertanyaan seputar aplikasi serta mengharapkan kritik dan saran demi perkembangan selanjutnya.

Dalam pengujian aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* ini diberikan beberapa pertanyaan untuk mendapatkan hasil perhitungan yang digunakan sebagai acuan untuk mengetahui tingkat kepuasan pengguna dalam penggunaannya. Untuk daftar pertanyaan dapat dilihat pada Tabel 3.92 sedangkan untuk penilaian pengukuran tingkat kepuasan pengguna didasarkan pada Tabel 3.93 dan Tabel 3.94

Tabel 3. 92 Daftar Pertanyaan

No	Kriteria Pertanyaan
1	Seberapa puas kebermanfaatan Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan <i>Image Processing</i> ?
2	Seberapa puas kebermanfaatan fitur <i>monitoring event</i> pameran pada Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan <i>Image Processing</i> ?
3	Seberapa puas kebermanfaatan fitur kalkulasi aplikasi Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan <i>Image Processing</i> ?
4	Seberapa puas kemudahan penggunaan Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan <i>Image Processing</i> ?
5	Seberapa puas tampilan Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan <i>Image Processing</i> ?

Tabel 3. 93 Indikator Penilaian

Predikat	Nilai
Sangat Puas	5
Puas	4
Cukup	3
Tidak Puas	2
Sangat Tidak Puas	1

Tabel 3. 94 Indikator Kepuasan Pengguna

Kategori	Presentasi
Sangat Memuaskan	81% - 100%
Memuaskan	61% - 80%
Cukup Memuaskan	41% - 60%
Tidak Memuaskan	21% - 40%
Sangat Tidak Memuaskan	0 - 20 %

Untuk menghitung tingkat kepuasan pengguna menggunakan rumus sebagai berikut:

Presentase Kepuasan Pengguna (%) =

$$\frac{\text{Total kepuasan pengguna}}{\text{Total kepuasan maksimum pengguna}} \times 100\%$$

BAB IV ANALISIS DAN PEMBAHASAN

Bab ini akan membahas mengenai analisis dan pembahasan mengenai sistem dan model data yang sudah dirancang sebelumnya. Analisis dan pembahasan terdiri dari pengujian model data, pengujian deteksi batik, pengujian tampilan pada android, pengujian kompatibilitas dan fungsionalitas aplikasi, dan kepuasan pengguna.

4.1. Analisis dan Pembahasan Sistem

Pada bagian ini membahas mengenai analisis aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* sebagai sebuah sistem yang dapat digunakan untuk mengenali motif batik dengan memanfaatkan fitur *capture* pada aplikasi *mobile* yang secara otomatis akan menampilkan informasi dari motif batik yang di-*capture*. Selain itu sistem ini dapat memonitoring *event* sebuah pameran batik dengan berdasarkan respon yang diberikan oleh pengunjung pameran.

4.1.1. Analisis Model

Pembuatan model yang digunakan dalam aplikasi ini menerapkan ilmu dari *deep learning* yang merupakan cabang dari *machine learning*. Pemecahan yang digunakan dalam ilmu ini beraneka ragam salah satunya yaitu *supervised*. Jenis *supervised* ini yang nantinya akan digunakan dalam pembuatan model.

Langkah awal yang perlu dilakukan sebelum menganalisis pembuatan model yaitu melakukan pengumpulan data yang berkaitan dengan sistem. Data yang digunakan berupa data gambar motif batik. Dataset motif dibatasi khusus area motif Jawa, agar kelas yang digunakan tidak terlalu kompleks dan lebih fokus terhadap daerah tertentu. Pada proses klasifikasi menggunakan metode pengklasifikasian yaitu CNN (*Convolutional Neural Network*) yang digunakan dalam pengolahan data *image*. Proses *training* terdapat 3 tahapan yaitu *convolutional layer*, *pooling layer*, dan *fully connected layer*.

Penggunaan model *transfer learning* (*pre-trained model*) yang telah dilatih pada *ImageNet* sangat membantu dalam proses *training*. Pada pembangunan aplikasi ini *transfer learning* digunakan untuk mempercepat *training* model.

Framework yang digunakan dalam pembuatan model yaitu menggunakan *Tensorflow*. Bahasa yang digunakan yaitu menggunakan bahasa *python* yang digunakan untuk pembuatan model, dari tahapan *processing* data sampai mengubah model menjadi ekstensi .tf dalam pembuatan aplikasi ini.

Workflow object detection dari aplikasi ini yaitu dimulai dari pengumpulan data, *labeling data*, membagi data latih dan uji, melakukan *training process* (terdapat proses *convolution*, *pooling*, aktivasi dengan ReLu, *fully connected layer*, dan hasil klasifikasi), melakukan *testing process* (terdapat *load* model, menginputkan gambar, proses konvolusi, dan hasil klasifikasi), setelah proses tersebut tahap terakhir yaitu *download* model dan interpretasi hasil.

Setelah proses pembuatan model selesai, model bisa didownload dan dipasangkan pada aplikasi *mobile*. Model tersebut akan direlasikan dengan database MySql yang bertujuan untuk menampilkan informasi ketika proses *capture* telah selesai. Dalam pelatihan model tidak hanya menggunakan MobileNet, ada beberapa model yang sudah dilakukan uji coba pada google codelab. Hal ini bertujuan untuk mencari model yang tepat untuk diterapkan pada aplikasi. Kualifikasi untuk model yaitu model memiliki akurasi tinggi dan memiliki ukuran file yang tidak besar, karena model akan diterapkan pada android. Berikut merupakan penjelasan dan perbandingan dari model:

4.1.1.1. Penjelasan Model

Metode yang diuji coba pada sistem ini diantaranya MobileNet, EfficientNet Lite 0, EfficientNet Lite 1, EfficientNet Lite 2, dan Efficientnet Lite 3. Penjelasan mengenai model – model tersebut dapat dilihat dibawah ini:

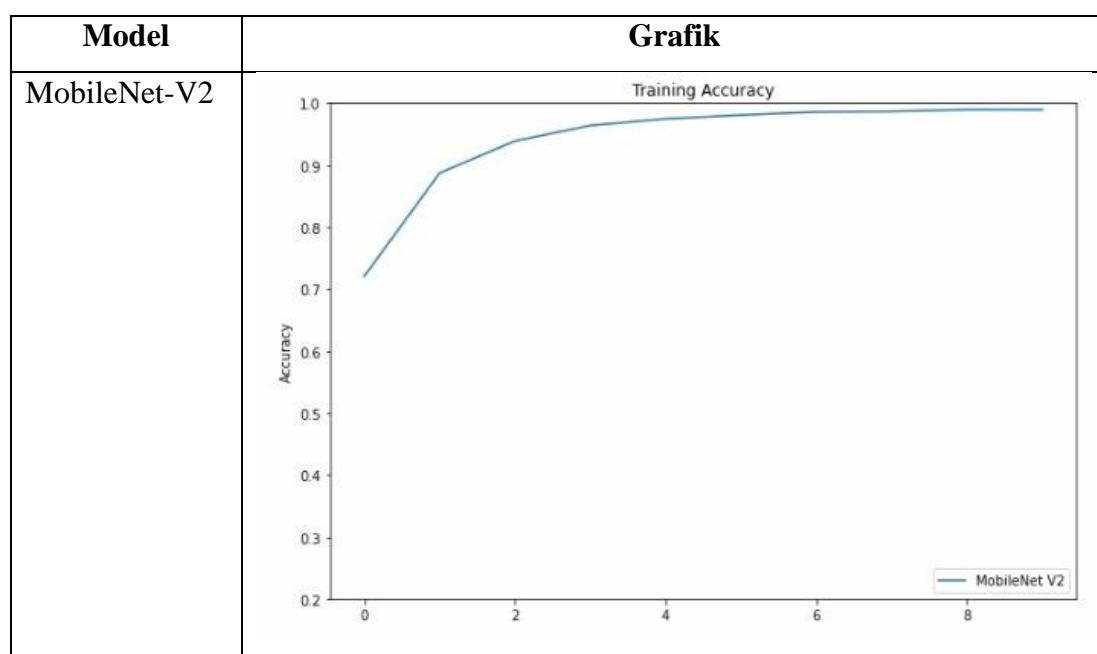
- 1) MobileNet merupakan salah satu arsitektur *convolutional neural network* (CNN) yang digunakan untuk *computing* pada *resource* yang besar.
2. EfficientNet Lite 0 merupakan salah satu arsitektur *convolutional neural network* (CNN) dengan model klasifikasi gambar yang mencapai akurasi mutakhir. EfficientNet Lite 0 mempunyai resolusi input model 224.

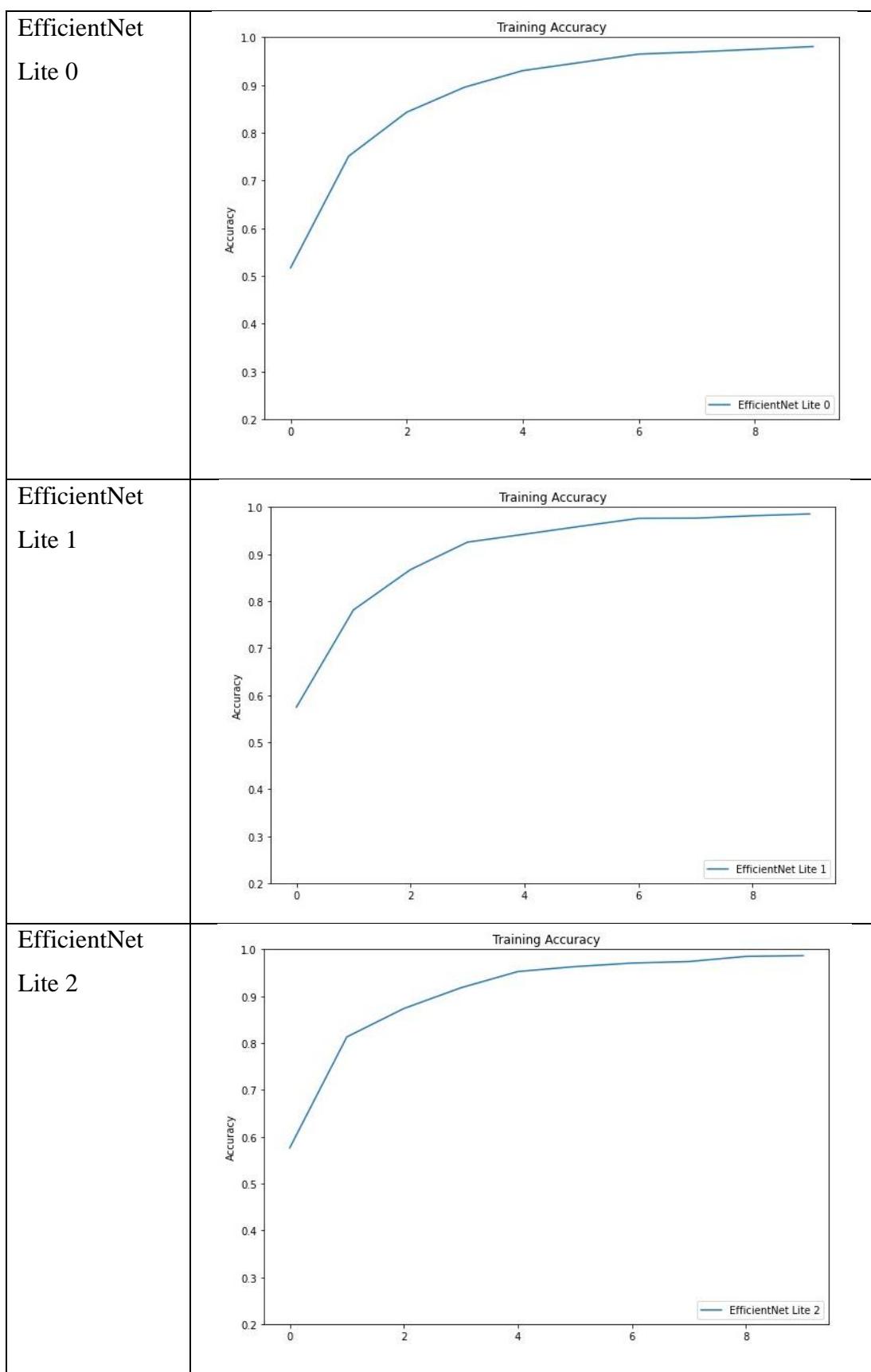
3. EfficientNet Lite 1 merupakan salah satu arsitektur *convolutional neural network* (CNN) dengan model klasifikasi gambar yang mencapai akurasi mutakhir. EfficientNet Lite 1 memiliki jumlah params yang lebih besar dibandingkan EfficientNet Lite 0 dan mempunyai resolusi input model 240.
4. EfficientNet Lite 2 merupakan salah satu arsitektur *convolutional neural network* (CNN) dengan model klasifikasi gambar yang mencapai akurasi mutakhir. EfficientNet Lite 2 memiliki jumlah params yang lebih besar dibandingkan EfficientNet Lite 1 dan mempunyai resolusi input model 260.

4.4.1.2. Perbandingan Grafik *Training Model*

Sebelum menentukan model yang tepat untuk diaplikasikan pada aplikasi, berikut merupakan perbandingan yang divisualisasikan menggunakan grafik:

Tabel 4. 1 Hasil Grafik *Training Model*





Berdasarkan Tabel 4.1 didapatkan hasil bahwa MobileNet V2 memiliki tingkatan akurasi grafik yang lebih tinggi dibandingkan dengan 3 model lainnya. Perbandingan dari setiap model memiliki perbedaan yang tidak jauh berbeda, tetapi untuk diaplikasikan pada sistem harus memilih model yang tepat. Pada model diatas sama-sama menggunakan 10 *epoch*.

4.4.1.3. Perbandingan Akurasi *Training Model*

Sebelum menentukan model yang tepat untuk diaplikasikan pada aplikasi, berikut merupakan perbandingan akurasi *training model* yang divisualisasikan menggunakan data tabel:

Tabel 4. 2 Hasil Akurasi *Training Model*

Model	Epoch	Akurasi	Waktu
MobileNet-V2	10 <i>Epoch</i>	99.16 %	234522 /s
EfficientNet Lite 0	10 <i>Epoch</i>	97.70 %	195334 /s
EfficientNet Lite 1	10 <i>Epoch</i>	98.36 %	301788 /s
EfficientNet Lite 2	10 <i>Epoch</i>	98.79 %	401374 /s

Berdasarkan Tabel 4.2 didapatkan hasil bahwa MobileNet memiliki tingkatan akurasi *trainning* yang lebih tinggi dibandingkan dengan 3 model lainnya yaitu sebesar 99.16 %. Selain itu waktu yang dibutuhkan untuk *training* model sebanyak 10 *epoch* tidak terlalu besar dibandingkan dengan model lainnya.

4.1.1.4. Perbandingan Akurasi *Testing Model*

Sebelum menentukan model yang tepat untuk diaplikasikan pada aplikasi, berikut merupakan perbandingan akurasi *testing* model yang divisualisasikan menggunakan data tabel:

Tabel 4. 3 Hasil Akurasi *Testing Model*

Model	Akurasi	Waktu
MobileNet-V2	99.86 %	920 /s
EfficientNet Lite 0	98.89 %	989 /s
EfficientNet Lite 1	99.03 %	1288 /s
EfficientNet Lite 2	99.72 %	1633 /s

Berdasarkan Tabel 4.3 didapatkan hasil bahwa *testing* model yang dilakukan menghasilkan model MobileNet yang memiliki tingkatan akurasi lebih tinggi dibandingkan dengan 3 model lainnya yaitu dengan tingkat akurasi *testing* sebesar 99.86 %. Selain itu waktu yang dibutuhkan untuk *testing* model tidak terlalu besar dibandingkan dengan model lainnya.

4.4.1.5. Perbandingan Ukuran Model

Ukuran model pada *file* sangat berpengaruh untuk menjadi bahan pertimbangan pada aplikasi ini. Model yang memiliki ukuran lebih kecil dan akurasi yang maksimal akan diterapkan pada aplikasi. Berikut merupakan perbandingan ukuran dari setiap model:

Model	Size
MobileNet V2	8.57M
EfficientNet Lite 0	12.91M
EfficientNet Lite 1	15.83M
EfficientNet Lite 2	18.4M

Gambar 4. 1 Hasil Perbandingan Ukuran Model

Berdasarkan Gambar 4.1. perbandingan ukuran model yang paling kecil yaitu MobileNet sedangkan ukuran model yang paling besar yaitu EfficientNet Lite 2. Maka dari itu model MobileNet adalah model yang tepat untuk diterapkan pada aplikasi, selain memiliki ukuran model yang kecil MobileNet juga memiliki akurasi yang tinggi.

4.1.2. Analisis Aplikasi

Aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* dibuat menggunakan android studio. Target pengguna dari aplikasi android yaitu untuk pengunjung dan penyelenggara *event* pameran. Sedangkan untuk menghandle bagian admin yaitu menggunakan sistem berbasis *website*. Penyimpanan yang digunakan dalam aplikasi ini menggunakan *database mysql*. Tampilan pada *mobile* menggunakan XML sedangkan *logic* yang digunakan dalam aplikasi android menggunakan PHP. Pada sisi *website admin* dibuat

menggunakan *framework Laravel*. Peran *website admin* pada aplikasi ini adalah untuk memanajemen data yang dibutuhkan dalam proses pertukaran data. Database yang bersifat *local* akan dihosting, agar proses pengambilan dan pengiriman data dapat berjalan dengan mudah.

Model yang telah didapatkan dalam proses klasifikasi akan dihubungkan pada aplikasi android. Ketika pengguna melakukan *scanning* motif batik, maka secara otomatis aplikasi tersebut akan memproses motif batik apa yang telah *discanning* dan menampilkan informasi terkait dengan motif tersebut.

4.2. Pembahasan Sistem

Pada tahap pembahasan sistem ini dilakukan pengujian performa aplikasi dan rekapitulasi mengenai beberapa pengujian pada aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* yang sudah dilakukan pada tahapan *sprint*.

4.2.1. Rekapitulasi Pengujian Fungsionalitas Model

Rekapitulasi pengujian fungsionalitas model dilakukan berdasarkan pada tahap pelaksanaan *sprint* sebelumnya. Berikut hasil rekapitulasi uji fungsionalitas model:

Tabel 4. 4 Hasil Pengujian Fungsionalitas Model

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
Memasukkan <i>Package</i> dan <i>Library</i>	Agar dapat menggunakan fungsi dari <i>package</i> yang sudah dideklarasikan	<i>Import package</i> dan <i>library</i> → jalankan program	Menampilkan hasil dari <i>package</i> yang terdownload	Berhasil
Membagi Data	Agar <i>dataset</i> dapat terbagi menjadi <i>train</i>	Masukkan fungsi <i>Image Classifier</i> <i>DataLoader</i> →	Menampilkan jumlah data yaitu 7200, jumlah label	Berhasil

	<i>data</i> dan <i>test data</i>	Bagi data menjadi 0.9 yang berarti 90% <i>training</i> 10% <i>testing</i>	yaitu 12 dan menampilkan label dari batik	
<i>Training</i> Model	Agar model dapat dilatih sebelum dilakukan <i>testing</i> model	Masukkan banyaknya <i>epoch</i> → melakukan klasifikasi model dengan MobileNet	Menampilkan informasi mengenai model dan akurasi dari model tersebut yaitu 0.9920	Berhasil
<i>Testing</i> Model	Agar model dapat dikenali saat motif dideteksi secara langsung	Masukkan kode untuk <i>testing</i> data	Menampilkan akurasai dari data <i>testing</i> yaitu 0.9986	Berhasil
<i>Export</i> Model	Agar model dapat berbentuk .tflite	Masukkan kode nama model → <i>export model</i>	Model akan terbuat dan akan disimpan sementara di folder google colab	Berhasil
<i>Download</i> Model	Agar model dapat didownload	Membuka folder dari google colab → buka file dan lakukan <i>download model</i>	Model akan didownload dan masuk kedalam penyimpanan komputer lokal	Berhasil

Berdasarkan hasil rekapitulasi pengujian fungsionalitas model pada Tabel 4.4 didapatkan hasil bahwa model pada aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* yang sudah menghasilkan akurasi yang tinggi dan berdampak baik pada proses pendekripsi.

4.2.1. Rekapitulasi Pengujian Deteksi Batik

Rekapitulasi pengujian deteksi batik dilakukan menggunakan kain batik asli dengan 4 motif batik. Pengambilan data per motif dilakukan sebanyak 10 kali pengambilan gambar batik dengan posisi yang berbeda - beda. Syarat yang dilakukan dalam pengujian ini antara lain posisi kain tidak terlipat, pengambilan gambar dengan keadaan cahaya terang, dan posisi kamera untuk melakukan *scan* batik dalam jangkauan 20-50 cm, untuk menghasilkan kebenaran motif batik yang dideteksi. Hasil deteksi batik akan ditampilkan dalam Tabel 4.5.

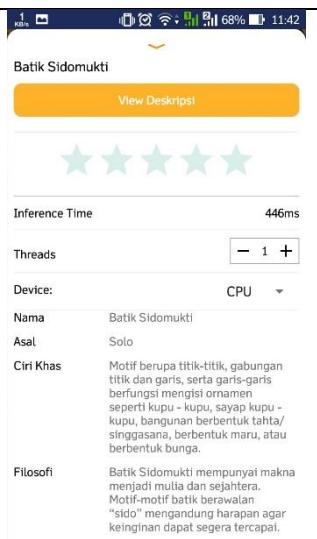
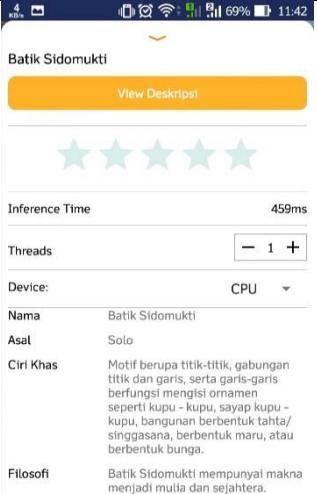
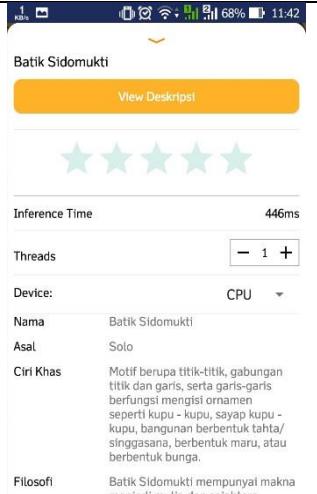
Tabel 4. 5 Rekapitulasi Pengujian Deteksi Batik

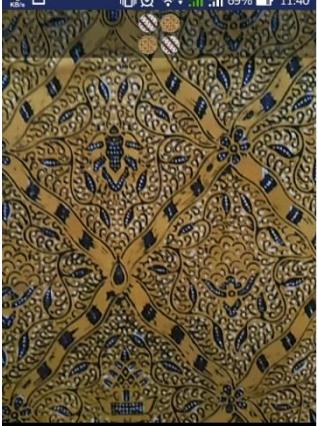
No.	Nama Batik	Hasil Deteksi	Informasi Batik
1.	Batik Semen	 	<p>Batik Semen</p> <p>Inference Time: 414ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Semen</p> <p>Asal: Surakarta</p> <p>Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuh-tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, semar, dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak.</p> <p>Filosofi: Penggambaran dari "kehidupan yang semi" (kehidupan yang</p>

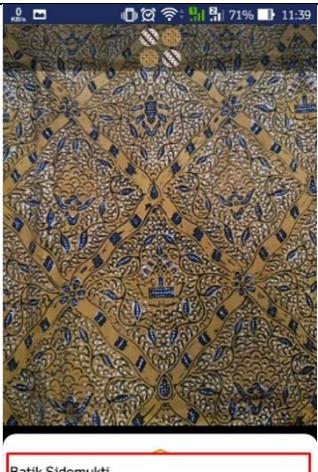
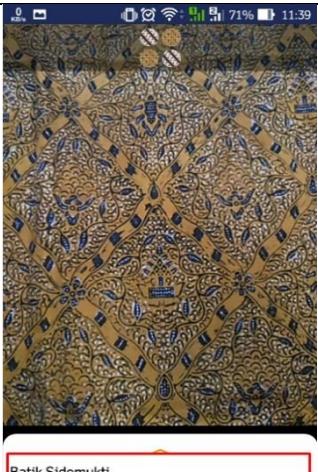
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Semen</p> <p>View Deskripsi</p> <p>5 stars (yellow)</p> <p>Inference Time: 414ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Semen Asal: Surakarta Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuh-tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak.</p> <p>Filosofi: Penggambaran dari "kehidupan yang semai" (kehidupan yang</p>
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Semen</p> <p>View Deskripsi</p> <p>5 stars (yellow)</p> <p>Inference Time: 434ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Semen Asal: Surakarta Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuh-tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak.</p> <p>Filosofi: Penggambaran dari "kehidupan yang semai" (kehidupan yang</p>
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Semen</p> <p>View Deskripsi</p> <p>5 stars (yellow)</p> <p>Inference Time: 598ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Semen Asal: Surakarta Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuh-tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak.</p> <p>Filosofi: Penggambaran dari "kehidupan yang semai" (kehidupan yang</p>

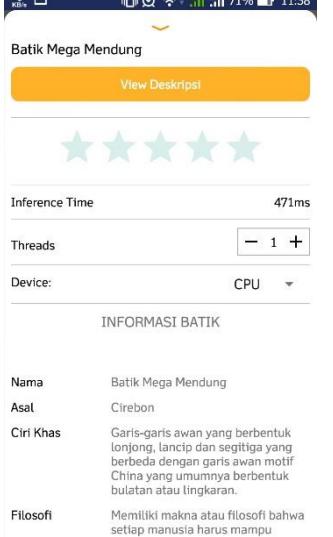
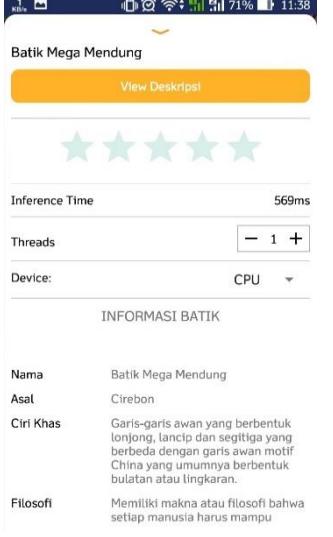
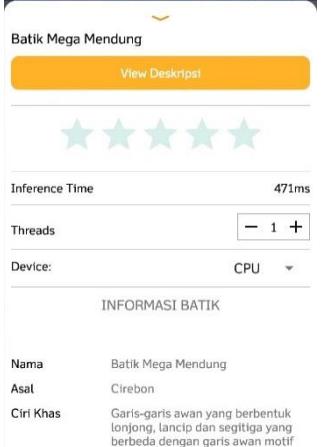
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Semen</p> <p>View Deskripsi</p> <p>5 KB 67% 11:48</p> <p>★★★★★</p> <p>Inference Time 452ms</p> <p>Threads - 1 +</p> <p>Device: CPU</p> <p>Nama: Batik Semen</p> <p>Asal: Surakarta</p> <p>Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak.</p> <p>Filosofi: Penggambaran dari "kehidupan yang sem" (kehidupan yang</p>
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Sidomukti</p> <p>View Deskripsi</p> <p>5 KB 67% 11:47</p> <p>★★★★★</p> <p>Inference Time 520ms</p> <p>Threads - 1 +</p> <p>Device: CPU</p> <p>Nama: Batik Semen</p> <p>Asal: Surakarta</p> <p>Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak.</p> <p>Filosofi: Penggambaran dari "kehidupan yang sem" (kehidupan yang</p>
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Semen</p> <p>View Deskripsi</p> <p>0 KB 67% 11:47</p> <p>★★★★★</p> <p>Inference Time 532ms</p> <p>Threads - 1 +</p> <p>Device: CPU</p> <p>Nama: Batik Semen</p> <p>Asal: Surakarta</p> <p>Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak.</p> <p>Filosofi: Penggambaran dari "kehidupan yang sem" (kehidupan yang</p>

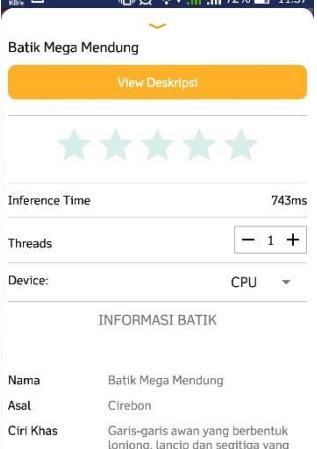
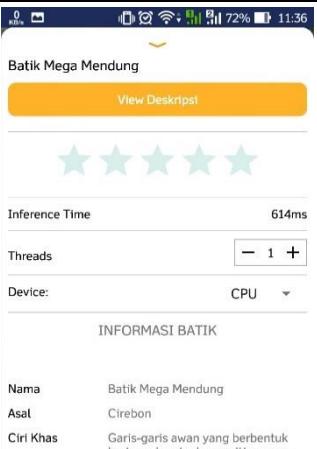
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Semen</p> <p>View Deskripsi</p> <p>532ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Semen Asal: Surakarta Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak. Filosofi: Penggambaran dari "kehidupan yang sem" (kehidupan yang</p>
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Semen</p> <p>View Deskripsi</p> <p>488ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Semen Asal: Surakarta Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak. Filosofi: Penggambaran dari "kehidupan yang sem" (kehidupan yang</p>
		 <p>Batik Semen</p> <p>View Deskripsi</p>	<p>Batik Semen</p> <p>View Deskripsi</p> <p>492ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Semen Asal: Surakarta Ciri Khas: Ornamen yang berhubungan dengan daratan, seperti tumbuhan atau binatang berkaki empat. Kedua adalah ornament yang berhubungan dengan udara, seperti garuda, burung dan megamendung. Sedangkan yang ketiga adalah ornament yang berhubungan dengan laut atau air, seperti ular, ikan dan katak. Filosofi: Penggambaran dari "kehidupan yang sem" (kehidupan yang</p>

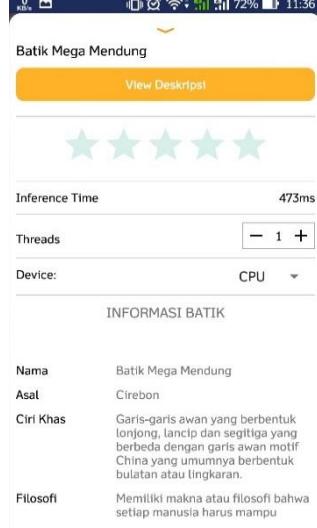
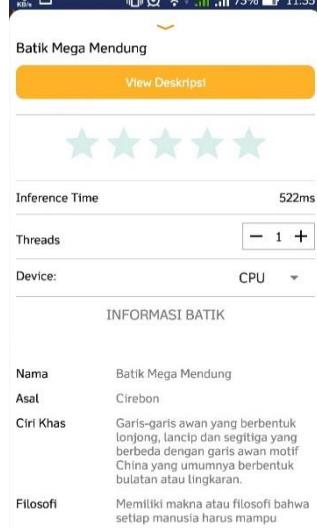
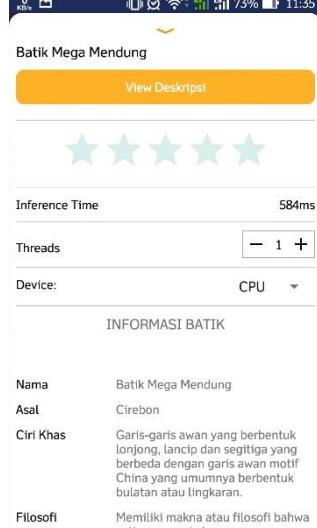
2.	Batik Sidomukti	   	  
----	-----------------	---	--

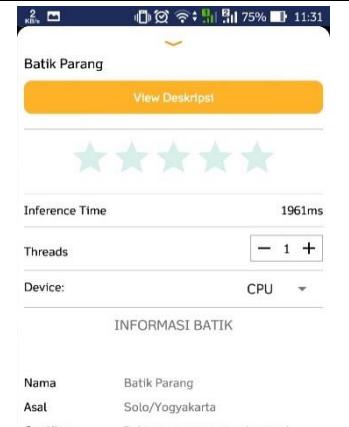
		 <p>Batik Sidomukti</p> <p>View Deskripsi</p>	<p>Batik Sidomukti</p> <p>View Deskripsi</p> <p>5 stars</p> <p>Inference Time: 435ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Sidomukti Asal: Solo Ciri Khas: Motif berupa titik-titik, gabungan titik dan garis, serta garis-garis berfungsi mengisi ornamen seperti kupu - kupu, sayap kupu - kupu, bangunan berbentuk tahta/ singgasana, berbentuk maru, atau berbentuk bunga. Filosofi: Batik Sidomukti mempunyai makna menjadi mulia dan sejahtera. Motif-motif batik berawalan "sido" mengandung harapan agar keinginan dapat segera tercapai.</p>
		 <p>Batik Sidomukti</p> <p>View Deskripsi</p>	<p>Batik Sidomukti</p> <p>View Deskripsi</p> <p>5 stars</p> <p>Inference Time: 494ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>Nama: Batik Sidomukti Asal: Solo Ciri Khas: Motif berupa titik-titik, gabungan titik dan garis, serta garis-garis berfungsi mengisi ornamen seperti kupu - kupu, sayap kupu - kupu, bangunan berbentuk tahta/ singgasana, berbentuk maru, atau berbentuk bunga. Filosofi: Batik Sidomukti mempunyai makna menjadi mulia dan sejahtera. Motif-motif batik berawalan "sido" mengandung harapan agar keinginan dapat segera tercapai.</p>
		 <p>Batik Sidomukti</p> <p>View Deskripsi</p>	<p>Batik Sidomukti</p> <p>View Deskripsi</p> <p>5 stars</p> <p>Inference Time: 424ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Sidomukti Asal: Solo Ciri Khas: Motif berupa titik-titik, gabungan titik dan garis, serta garis-garis berfungsi mengisi ornamen seperti kupu - kupu, sayap kupu - kupu, bangunan berbentuk tahta/ singgasana, berbentuk maru, atau berbentuk bunga.</p>

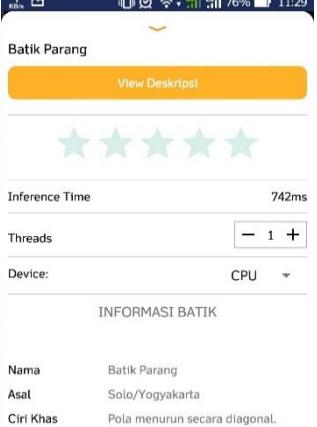
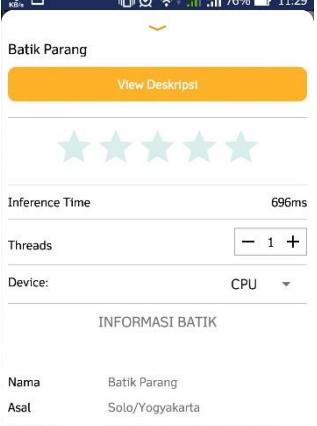
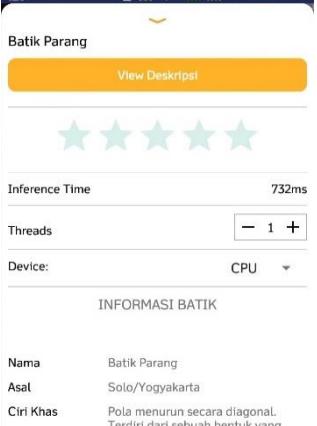
		  	<p>Batik Sidomukti</p> <p>View Deskripsi</p> <p></p> <p>Inference Time: 501ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Sidomukti Asal: Solo Ciri Khas: Motif berupa titik-titik, gabungan titik dan garis, serta garis-garis berfungsi mengisi ornamen seperti kupu - kupu, sayap kupu - kupu, bangunan berbentuk tahta/ singgasana, berbentuk maru, atau berbentuk bunga.</p>
		  	<p>Batik Sidomukti</p> <p>View Deskripsi</p> <p></p> <p>Inference Time: 437ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Sidomukti Asal: Solo Ciri Khas: Motif berupa titik-titik, gabungan titik dan garis, serta garis-garis berfungsi mengisi ornamen seperti kupu - kupu, sayap kupu - kupu, bangunan berbentuk tahta/ singgasana, berbentuk maru, atau berbentuk bunga.</p>
		  	<p>Batik Sidomukti</p> <p>View Deskripsi</p> <p></p> <p>Inference Time: 435ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Sidomukti Asal: Solo Ciri Khas: Motif berupa titik-titik, gabungan titik dan garis, serta garis-garis berfungsi mengisi ornamen seperti kupu - kupu, sayap kupu - kupu, bangunan berbentuk tahta/ singgasana, berbentuk maru, atau berbentuk bunga.</p> <p>Filosofi: Batik Sidomukti mempunyai makna menjadi mula dan sejahtera. Motif-motif batik berawalan "sido" mengandung harapan agar keinginan dapat segera tercapai.</p>

3.	<p>Batik Mega Mendung</p> 	 <p>Batik Mega Mendung</p> <p>View Deskripsi</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>Batik Mega Mendung</p> <p>View Deskripsi</p>
		 <p>Batik Mega Mendung</p> <p>View Deskripsi</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>Batik Mega Mendung</p> <p>View Deskripsi</p>
		 <p>Batik Mega Mendung</p> <p>View Deskripsi</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>Batik Mega Mendung</p> <p>View Deskripsi</p>

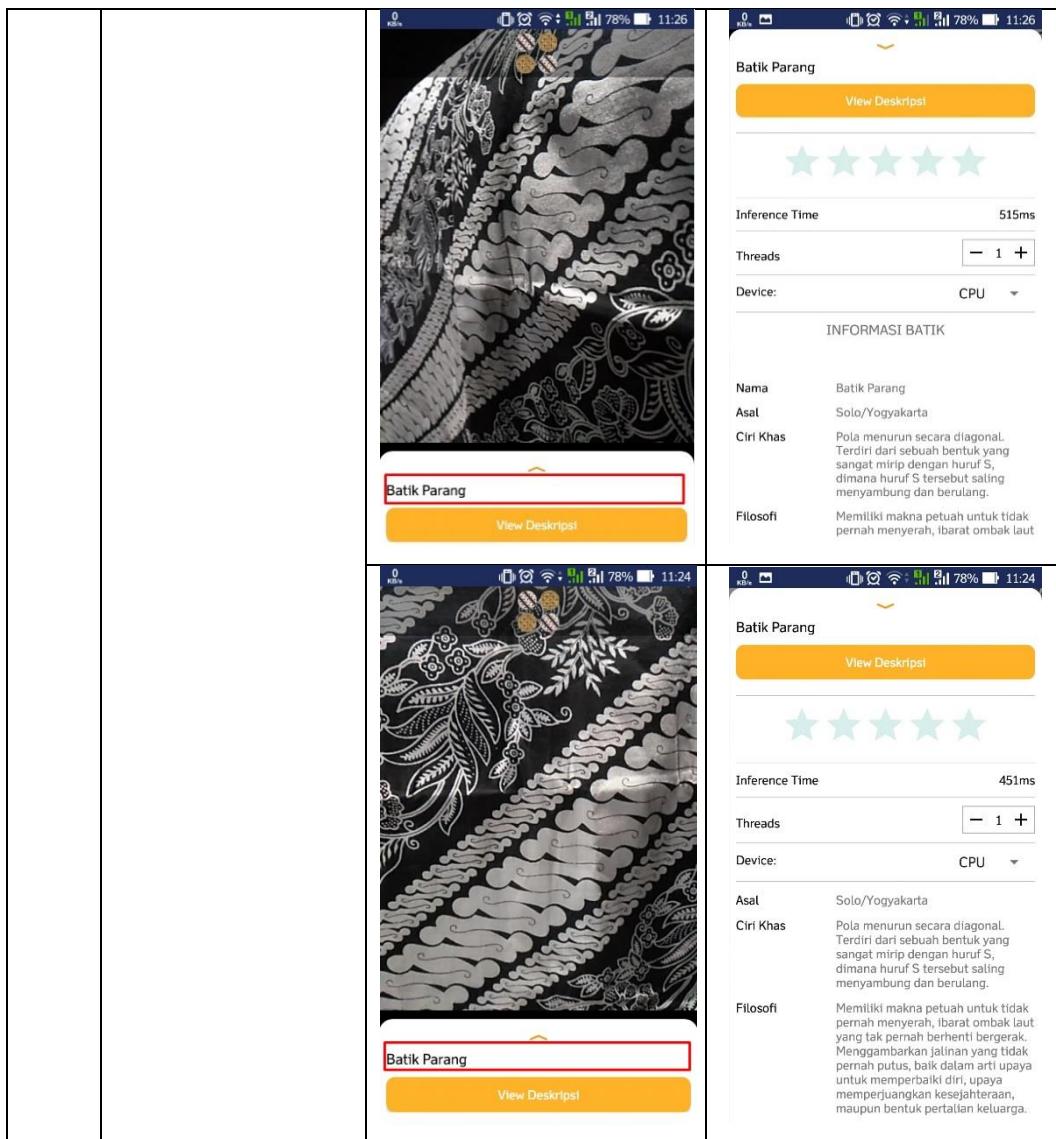
		 <p>Batik Mega Mendung</p> <p>View Deskripsi</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>5 stars</p> <p>Inference Time: 743ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Mega Mendung Asal: Cirebon Ciri Khas: Garis-garis awan yang berbentuk lonjong, lancip dan segitiga yang berbeda dengan garis awan motif China yang umumnya berbentuk bulatan atau lingkaran. Filosofi: Memiliki makna atau filosofi bahwa setiap manusia harus mampu</p>
		 <p>Batik Mega Mendung</p> <p>View Deskripsi</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>5 stars</p> <p>Inference Time: 614ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Mega Mendung Asal: Cirebon Ciri Khas: Garis-garis awan yang berbentuk lonjong, lancip dan segitiga yang berbeda dengan garis awan motif China yang umumnya berbentuk bulatan atau lingkaran. Filosofi: Memiliki makna atau filosofi bahwa setiap manusia harus mampu</p>
		 <p>Batik Mega Mendung</p> <p>View Deskripsi</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi</p> <p>5 stars</p> <p>Inference Time: 472ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Mega Mendung Asal: Cirebon Ciri Khas: Garis-garis awan yang berbentuk lonjong, lancip dan segitiga yang berbeda dengan garis awan motif China yang umumnya berbentuk bulatan atau lingkaran. Filosofi: Memiliki makna atau filosofi bahwa setiap manusia harus mampu</p>

		 <p>Batik Mega Mendung</p> <p>View Deskripsi!</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi!</p> <p>Inference Time: 473ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Mega Mendung Asal: Cirebon Ciri Khas: Garis-garis awan yang berbentuk lonjong, lancip dan segitiga yang berbeda dengan garis awan motif China yang umumnya berbentuk bulatan atau lingkaran. Filosofi: Memiliki makna atau filosofi bahwa setiap manusia harus mampu</p>
		 <p>Batik Mega Mendung</p> <p>View Deskripsi!</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi!</p> <p>Inference Time: 522ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Mega Mendung Asal: Cirebon Ciri Khas: Garis-garis awan yang berbentuk lonjong, lancip dan segitiga yang berbeda dengan garis awan motif China yang umumnya berbentuk bulatan atau lingkaran. Filosofi: Memiliki makna atau filosofi bahwa setiap manusia harus mampu</p>
		 <p>Batik Mega Mendung</p> <p>View Deskripsi!</p>	 <p>Batik Mega Mendung</p> <p>View Deskripsi!</p> <p>Inference Time: 584ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Mega Mendung Asal: Cirebon Ciri Khas: Garis-garis awan yang berbentuk lonjong, lancip dan segitiga yang berbeda dengan garis awan motif China yang umumnya berbentuk bulatan atau lingkaran. Filosofi: Memiliki makna atau filosofi bahwa setiap manusia harus mampu</p>

			
4.	Batik Parang		 <p>Batik Parang</p> <p>View Deskripsi</p> <p>Inference Time: 1961ms</p> <p>Threads: - 1 +</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Parang Asal: Solo/Yogyakarta Ciri Khas: Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang. Filosofi: Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut.</p>

		 <p>Batik Parang</p> <p>View Deskripsi</p>	 <p>Batik Parang</p> <p>View Deskripsi</p> <p>5 stars (green)</p> <p>Inference Time: 742ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Parang Asal: Solo/Yogyakarta Ciri Khas: Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang. Filosofi: Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut</p>
		 <p>Batik Parang</p> <p>View Deskripsi</p>	 <p>Batik Parang</p> <p>View Deskripsi</p> <p>5 stars (green)</p> <p>Inference Time: 696ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Parang Asal: Solo/Yogyakarta Ciri Khas: Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang. Filosofi: Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut</p>
		 <p>Batik Parang</p> <p>View Deskripsi</p>	 <p>Batik Parang</p> <p>View Deskripsi</p> <p>5 stars (green)</p> <p>Inference Time: 732ms</p> <p>Threads: 1</p> <p>Device: CPU</p> <p>INFORMASI BATIK</p> <p>Nama: Batik Parang Asal: Solo/Yogyakarta Ciri Khas: Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang. Filosofi: Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut</p>

		 <p>Batik Parang</p> <p>View Deskripsi</p>	<p>Batik Parang</p> <p>View Deskripsi</p> <p>INFORMASI BATIK</p> <p>Nama Batik Parang</p> <p>Asal Solo/Yogyakarta</p> <p>Ciri Khas Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang.</p> <p>Filosofi Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut</p>
		 <p>Batik Parang</p> <p>View Deskripsi</p>	<p>Batik Parang</p> <p>View Deskripsi</p> <p>INFORMASI BATIK</p> <p>Nama Batik Parang</p> <p>Asal Solo/Yogyakarta</p> <p>Ciri Khas Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang.</p> <p>Filosofi Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut</p>
		 <p>Batik Parang</p> <p>View Deskripsi</p>	<p>Batik Parang</p> <p>View Deskripsi</p> <p>INFORMASI BATIK</p> <p>Nama Batik Parang</p> <p>Asal Solo/Yogyakarta</p> <p>Ciri Khas Pola menurun secara diagonal. Terdiri dari sebuah bentuk yang sangat mirip dengan huruf S, dimana huruf S tersebut saling menyambung dan berulang.</p> <p>Filosofi Memiliki makna petuh untuk tidak pernah menyerah, ibarat ombak laut</p>



Berdasarkan Tabel 4.5. didapatkan hasil bahwa deteksi batik yang dilakukan menggunakan aplikasi android berhasil mendeteksi batik dengan keseluruhan tingkat kebenaran batik mencapai 100% atau berhasil. Batik yang dideteksi dapat menampilkan informasi yang sesuai dengan batik tersebut.

4.2.2. Rekapitulasi Pengujian Tampilan Android

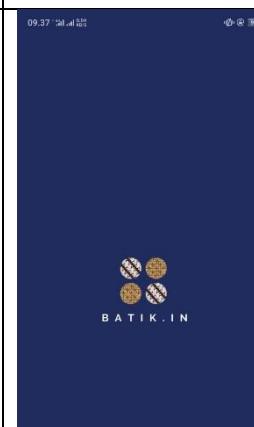
Pengujian tampilan aplikasi android dilakukan menggunakan tiga buah *smartphone* android dengan spesifikasi yang berbeda – beda seperti yang terlihat pada Tabel 4.6.

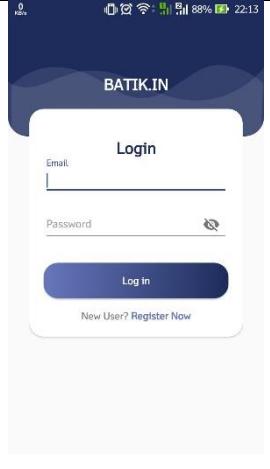
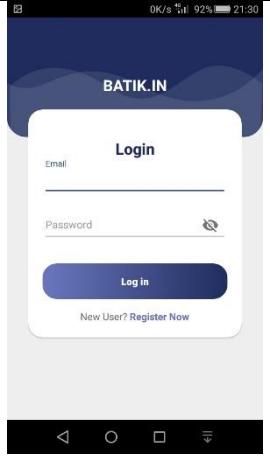
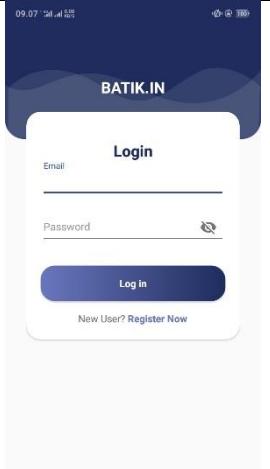
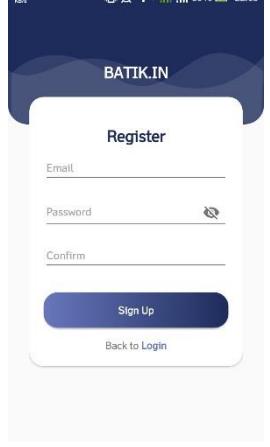
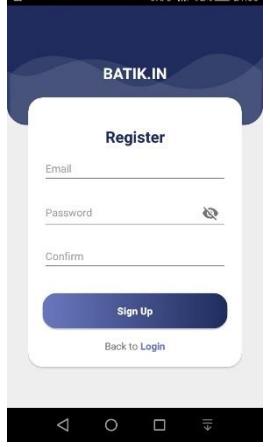
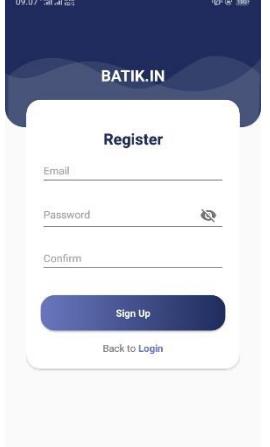
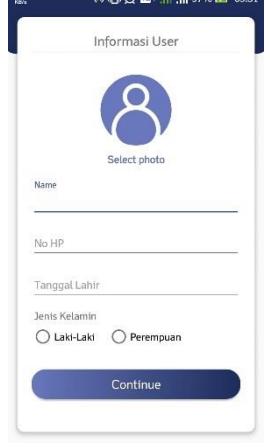
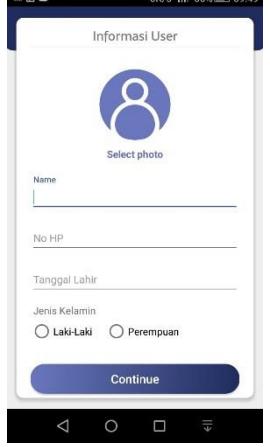
Tabel 4. 6 Spesifikasi Smartphone Penguji Aplikasi Android

Nama Perangkat	Ukuran Layar	Versi Android	Kamera
ASUS_X007D	720 x 1280 piksel	Marshmallow	13 MP
Huawei Cun-L22	720 x 1280 piksel	Lollipop	8 MP
Oppo A5s	720 x 1520 piksel	Oreo	13 MP

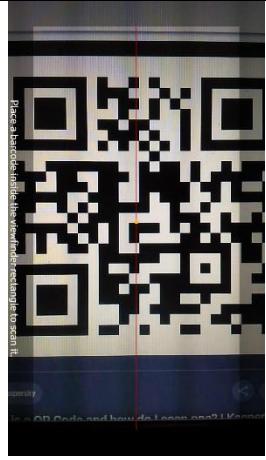
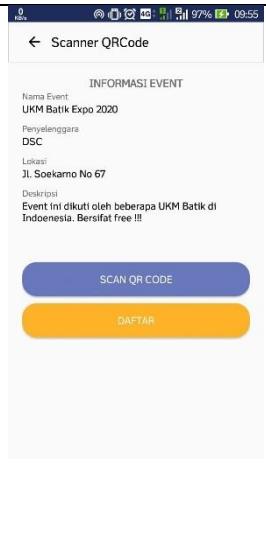
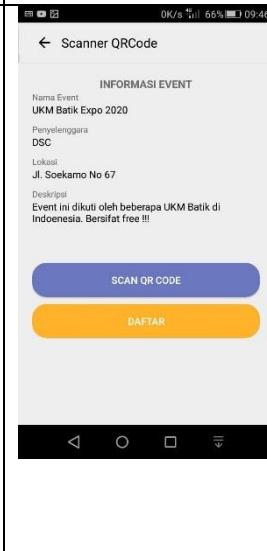
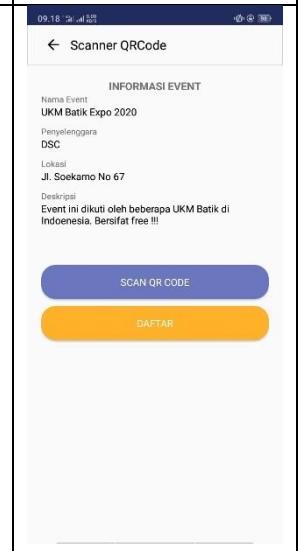
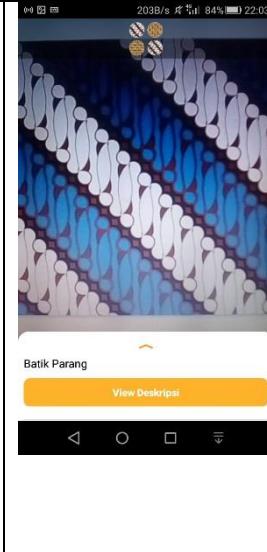
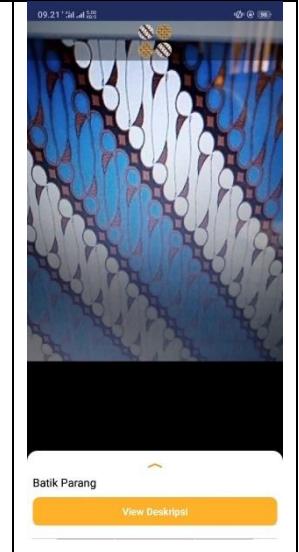
Hasil rekap tampilan aplikasi android dapat dilihat pada Tabel 4.7.

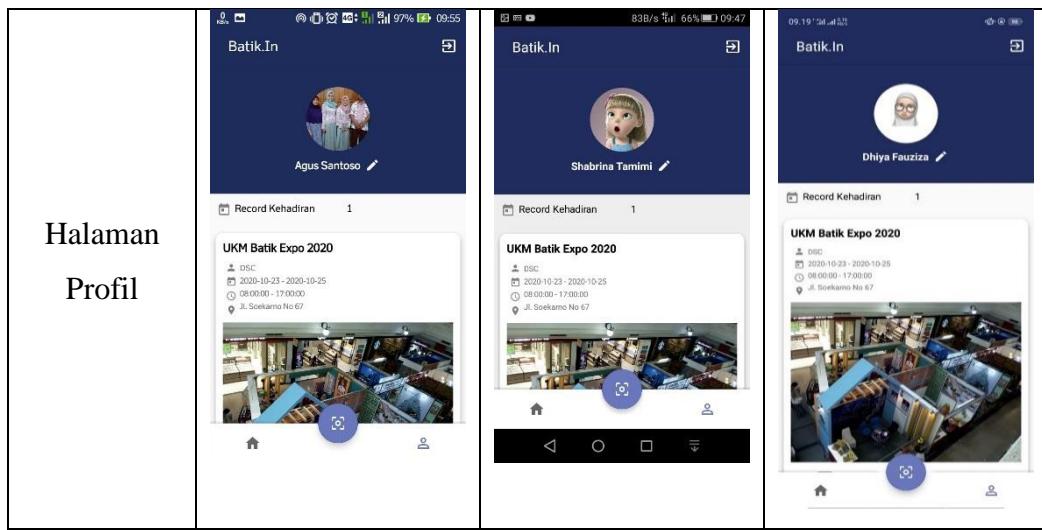
Tabel 4. 7 Hasil Rekapitulasi Tampilan Android

Pengujian	Smartphone		
	Asus_X007D	Huawei Cun-L22	Oppo A5s
Halaman <i>Onboarding</i>			
Halaman <i>Splash Screen</i>			

Halaman <i>Login</i>			
Halaman <i>Register</i>			
Halaman Pengisian Data Pengguna			

Halaman <i>Home</i>			
Halaman <i>Event</i> Pameran			
Halaman <i>Komentar</i> <i>Event</i>			

<p>Halaman Scan Qrcode Event</p>			
<p>Halaman Konfirmasi Event</p>			
<p>Halaman Deteksi Batik</p>			



Berdasarkan Tabel 4.7. dapat disimpulkan bahwa uji coba tampilan android yang dilakukan menggunakan tiga android yang berbeda memiliki tampilan yang sesuai dengan UI android yang asli dan tidak berubah. Tampilan mempunyai desain yang dinamis dan mengikuti ukuran layar namun tidak merubah tampilannya.

4.2.2. Rekapitulasi Pengujian Kompatibilitas Aplikasi

Rekapitulasi pengujian kompatibilitas aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* yang sudah dilakukan pada tahapan *sprint* berdasarkan hasil dari tahap pelaksanaan *sprint* sebelumnya. Berikut hasil rekapitulasi pengujian kompatibilitas aplikasi:

Tabel 4. 8 Rekapitulasi Pengujian Kompabilitas Aplikasi

Tampilan	Hasil Pengujian Tampilan	
	Android	Website
Halaman <i>Onboarding</i>	✓	
Halaman <i>Splash Screen</i>	✓	
Halaman <i>Login</i>	✓	
Halaman <i>Register</i>	✓	
Halaman Pengisian Data Pengguna	✓	
Halaman <i>Home</i>	✓	
Halaman <i>Event Pameran</i>	✓	
Halaman Komentar <i>Event</i>	✓	

Halaman Scan Qrcode Event	✓	
Halaman Konfirmasi Event	✓	
Halaman Deteksi Batik	✓	
Halaman Profil	✓	
Halaman Login		✓
Halaman Dashboard		✓
Halaman Menampilkan Informasi Motif Batik		✓
Halaman Menambah Informasi Motif Batik		✓
Halaman Mengedit Informasi Motif Batik		✓
Halaman Menampilkan Berita		✓
Halaman Menambah Berita		✓
Halaman Mengedit Berita		✓
Halaman Menampilkan All Event Pameran		✓
Halaman Menampilkan Detail Event		✓
Halaman Konfirmasi Event		✓
Halaman Menampilkan Data Pengunjung		✓
Halaman Menambah Data Pengunjung		✓
Halaman Mengedit Data Pengunjung		✓
Halaman Menampilkan Data Penyelenggara		✓
Halaman Menambah Data Penyelenggara		✓
Halaman Mengedit Data Penyelenggara		✓
Halaman Profil		✓
Halaman Menambahkan Event		✓
Halaman Mengedit Event		✓
Halaman Menampilkan MyEvent		✓
Halaman Menampilkan QRCode		✓
Halaman Menampilkan Buku Tamu		✓
Halaman Menambahkan Kehadiran Pengunjung		✓
Halaman Melihat Status Event		✓

Berdasarkan tabel 4.8. dapat disimpulkan bahwa pengujian kompatibilitas dari dua *platform* yang berbeda berhasil berjalan dengan baik serta memiliki tampilan yang sesuai dengan rencana perancangan.

4.2.3. Rekapitulasi Uji Fungsionalitas Aplikasi

Rekapitulasi pengujian fungsionalitas aplikasi dilakukan berdasarkan pada tahap penggerjaan *sprint* sebelumnya. Berikut hasil rekapitulasi uji fungsionalitas aplikasi:

Tabel 4. 9 Hasil Pengujian Fungsionalitas Aplikasi

Pengujian	Tujuan	Cara Menguji	Indikator	Hasil Pengujian
<i>Onboarding</i>	Agar pengguna dapat melihat informasi yang ditawarkan oleh aplikasi	Pengguna melakukan instalasi aplikasi dan <i>onboarding</i> akan tampil	Pengguna bisa masuk ke halaman <i>login</i>	Berhasil
<i>Splash screen</i>	Agar pengguna dapat melihat kesan pertama dari aplikasi	Pengguna membuka aplikasi yang ada pada <i>smartphone</i>	Pengguna bisa masuk ke halaman aplikasi	Berhasil
<i>Login</i>	Agar pengguna dapat masuk ke dalam aplikasi pengguna lebih jauh	Pengguna harus memasukkan akun yang sudah terdaftar sebelumnya dengan memasukkan	Pengguna bisa masuk ke halaman pengguna	Berhasil

		<i>username</i> dan <i>password</i>		
<i>Register</i>	Agar pengguna dapat mendaftarkan akun baru ke aplikasi	Pengguna memasukkan alamat <i>email</i> , <i>password</i> , dan konfirmasi <i>password</i> . Pengguna harus mengeklik tombol <i>signup</i>	Pengguna akan mempunyai akun yang bisa digunakan untuk <i>login</i>	Berhasil
Halaman Pengisian Data Pengguna	Agar sistem mendapatkan informasi mengenai pengguna	Pengguna memasukkan foto profil, nama, no hp, tanggal lahir dan jenis kelamin	Pengguna akan bisa masuk ke halaman <i>home</i>	Berhasil
Halaman <i>Home</i>	Agar pengguna dapat melihat <i>event</i> dan berita	Pengguna mengklik tombol <i>left button</i> pada <i>event card</i> dan melakukan <i>scroll</i> pada berita yang ditampilkan dan dapat mengklik <i>view more</i> berita	Pengguna dapat masuk ke halaman <i>event</i> dan pengguna dapat melihat berita yang ditampilkan dan <i>view more</i> deskripsinya	Berhasil

Menampilkan <i>event</i>	Agar pengguna dapat melihat detail dari <i>event</i>	Pengguna dapat melakukan <i>scroll</i> secara vertikal	Pengguna dapat melihat detail dari <i>event</i> seperti nama dari pameran batik, lokasi, tanggal, waktu dan penyelenggara pameran	Berhasil
Komentar <i>Event</i>	Agar pengguna dapat memberikan komentar terhadap <i>event</i> pameran batik	Klik pada <i>textbox</i> yang sudah disediakan sistem, setelah itu tulis komentar dan klik tombol kirim	Komentar akan terkirim dan ditampilkan di halaman komentar	Berhasil
Konfirmasi <i>Event</i>	Agar pengguna dapat mengkonfirmasi kehadiran	Pengguna dapat mengklik tombol daftar	Menampilkan notifikasi berhasil mendaftar	Berhasil
<i>Scan Qrcode</i>	Agar pengguna dapat menscan <i>qrcode event</i> dan mendapatkan akses masuk ke pameran	Memilih tombol <i>scan qrcode</i> dan mengarahkan kamera ke <i>qrcode</i> registrasi	Menampilkan detail dari <i>event</i>	Berhasil

Deteksi Batik	Untuk mendeteksi batik	Arahkan kamera pada objek batik yang akan dideteksi	Menampilkan label dari batik yang dideteksi secara <i>realtime</i>	Berhasil
Menampilkan Profil	Agar pengguna dapat melihat profil dan melihat histori <i>event</i> yang pernah diikuti	Klik menu profil dan scroll secara vertikal	Menampilkan profil dan histori dari <i>event</i> yang diikuti	Berhasil
Manajemen Motif Batik	Agar admin dapat melihat, menambah, mengubah dan menghapus data motif	Admin dapat mengklik fitur <i>view</i> motif batik	Menampilkan manajemen dari motif batik	Berhasil
Manajemen Berita	Agar admin dapat melihat, menambah, mengubah dan menghapus data berita	Admin dapat mengklik fitur <i>view</i> berita	Menampilkan manajemen dari berita	Berhasil
Halaman Manajemen Data Pengunjung	Agar admin dapat melihat, menambah, mengubah dan menghapus data pengunjung	Admin dapat mengklik fitur <i>view</i> pengunjung	Menampilkan manajemen dari pengunjung	Berhasil

Halaman Manajemen Data Penyelenggara	Agar admin dapat melihat, menambah, mengubah dan menghapus data penyelenggara	Admin dapat mengklik fitur <i>view</i> penyelenggara	Menampilkan manajemen dari penyelenggara	Berhasil
Halaman Manajemen Data <i>Event</i>	Agar admin dapat melihat, menambah, mengubah dan menghapus data <i>event</i>	Admin dapat mengklik fitur <i>view event</i>	Menampilkan manajemen dari <i>event</i>	Berhasil
Mencetak <i>Qrcode</i>	Agar penyelenggara dapat mencetak <i>qrcode</i> kedalam bentuk kertas	Pada bagian detail <i>event</i> dapat mengklik fitur <i>print qrcode</i>	<i>Qrcode</i> akan ditampilkan dalam bentuk <i>print preview</i>	Berhasil
Menampilkan Buku Tamu	Agar penyelenggara dapat melihat data pengunjung yang sudah datang ke pameran	Penyelenggara dapat mengklik tombol buku tamu pada <i>event</i>	Menampilkan pengunjung yang sudah hadir dengan menscan <i>qrcode</i> pada aplikasi android	Berhasil
Menambahkan Data Kehadiran	Agar penyelenggara dapat menambahkan pengunjung ke buku tamu	Masukkan nama, alamat <i>email</i> , dan no hp kedalam <i>form</i>	Menampilkan data yang sudah diinput kedalam tabel	Berhasil

Melihat status konfirmasi <i>event</i>	Agar penyelengara dapat melihat status <i>event</i> apakah sudah terkonfirmasi	Pilih menu status konfirmasi <i>event</i>	Menampilkan status gagal jika <i>event</i> tidak disetujui dan menampilkan <i>waiting</i> jika <i>event</i> belum dicek oleh admin	Berhasil
--	--	---	--	----------

Berdasarkan hasil rekapitulasi pengujian fungsionalitas aplikasi pada Tabel 4.9. didapatkan hasil bahwa aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* yang sudah berhasil menjalankan semua fitur sesuai tujuan yang sudah ditetapkan.

4.2.4. Pengujian Tingkat Kepuasan Pengguna

Pengujian ini dilakukan pada tanggal 07 Oktober 2020 sampai 08 Oktober 2020 pada pengunjung pameran. Berikut adalah pertanyaan yang diajukan pada 16 responden yang berbeda dengan variasi pertanyaan sebagai berikut:

- 1) Seberapa puas kebermanfaatan Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan *Image Processing*?
- 2) Seberapa puas kebermanfaatan fitur *monitoring* event pameran pada Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan *Image Processing*?
- 3) Seberapa puas kebermanfaatan fitur kalkulasi aplikasi Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan *Image Processing*?
- 4) Seberapa puas kemudahan penggunaan Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan *Image Processing*?
- 5) Seberapa puas tampilan Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan *Image Processing*?

Berikut adalah hasil penilaian responden mengenai aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* dari 16 responden yang berbeda, seperti yang ditunjukkan pada Tabel 4.10.

Tabel 4. 10 Hasil Kuisioner Responden

No.	Nama Responden	Event Batik yang Dikunjungi	Pertanyaan					Total Nilai
			1	2	3	4	5	
1.	Qiswatul Ulfah	Batik Semarang 16	4	4	4	4	4	20
2.	Siska Meilinda	Museum Batik Festival	5	4	4	4	4	21
3.	Nabilla Citra	Museum Batik Kuala Lumpur, Malaysia	4	4	4	4	4	20
4.	Kartika Maharani	Pameran tahunan SMA	3	5	4	5	5	22
5.	Resmito Nurul	Kampung Batik Semarang	4	3	4	4	4	19
6.	Maratus Sholekhah	Kampung Batik Semarang	5	5	5	5	5	25
7.	Nadila Larasati	Kelas Kewirausahaan	4	3	4	5	5	21
8.	Deby Ayu	Pameran Busana Batik, Kota Lama	4	4	4	3	3	19
9.	Dyah Ayu Christanti	Kampung Batik Semarang	5	4	3	4	3	19
10.	Trischa Adila	Kampung Batik Semarang	3	4	4	3	5	19
11.	Fahmi Anwar Kusuma	Pameran Laweyan Solo	5	4	4	4	5	22
12.	Aditia Prasetio	Kampung Batik Semarang	5	5	5	5	5	25

13.	Bobi	Museum Batik Pekalongan	5	5	5	5	5	25
14.	Dyta Sandy P	Pameran Batik Mall Ciputra	5	5	5	5	5	25
15.	Alessandra FA	Pekan Batik Nasional Kota Pekalongan	4	5	5	5	5	24
16.	Ardani Nur Istiqomah	Kampung Batik Semarang	5	4	5	5	4	23
TOTAL			70	68	69	69	71	347

Jumlah nilai maksimal untuk indikator penilaian = $5 \times 5 = 25$

Jumlah responden yang mengisi kuesioner = 16

Jumlah kepuasan responden = 347

Jumlah kepuasan maksimal = $16 \times 25 = 400$

$$\begin{aligned} \text{Skoring Kepuasan} &= \frac{\text{Total Kepuasan Responden}}{\text{Total Kepuasan Maksimum Responden}} \times 100\% \\ &= \frac{347}{400} \times 100\% = 86,75\% \end{aligned}$$

Dari hasil jajak pendapat pada Tabel 4.10 didapatkan hasil bahwa tingkat kepuasan pengguna adalah 86,75 %. Dari hasil perhitungan tingkat kepuasan pengguna tersebut berdasarkan Tabel 3.94 didapatkan indikator penilaian “sangat memuaskan”.

BAB V KESIMPULAN

Dari hasil pembuatan Tugas Akhir Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan *Image Processing*, didapatkan kesimpulan sebagai berikut:

- 1) Tugas akhir ini berhasil membuat aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing*.
- 2) Tugas akhir ini berhasil membuat aplikasi untuk manajemen *event* pameran yang dapat menampung respon pengunjung.
- 3) Pembuatan tugas akhir ini menggunakan kerangka kerja scrum dengan estimasi waktu penggeraan tiga minggu enam hari dan waktu penggeraan dapat diselesaikan dalam waktu tiga minggu tiga hari.
- 4) Proses *training* model dengan menggunakan pendekatan *transfer learning*, implementasi model menggunakan arsitektur CNN MobileNet. *Dataset* terdapat 12 kelas yang berjumlah 7200 motif batik yang dibagi menjadi 90% untuk data *training* dan 10% untuk data *testing*. Akurasi yang dihasilkan yaitu 99.16% untuk *training model* dan 99.86% untuk *testing* model.
- 5) Proses *testing* yang dilakukan menggunakan kain batik dengan 4 motif berbeda dan dilakukan 10 kali pengambilan gambar batik tiap motif dari berbagai sudut dihasilkan akurasi kebenaran dari motif Batik Semen sebesar 100%, Batik Sidomukti sebesar 100%, Batik Mega Mendung sebesar 100%, dan Batik Parang sebesar 100 %.

DAFTAR PUSTAKA

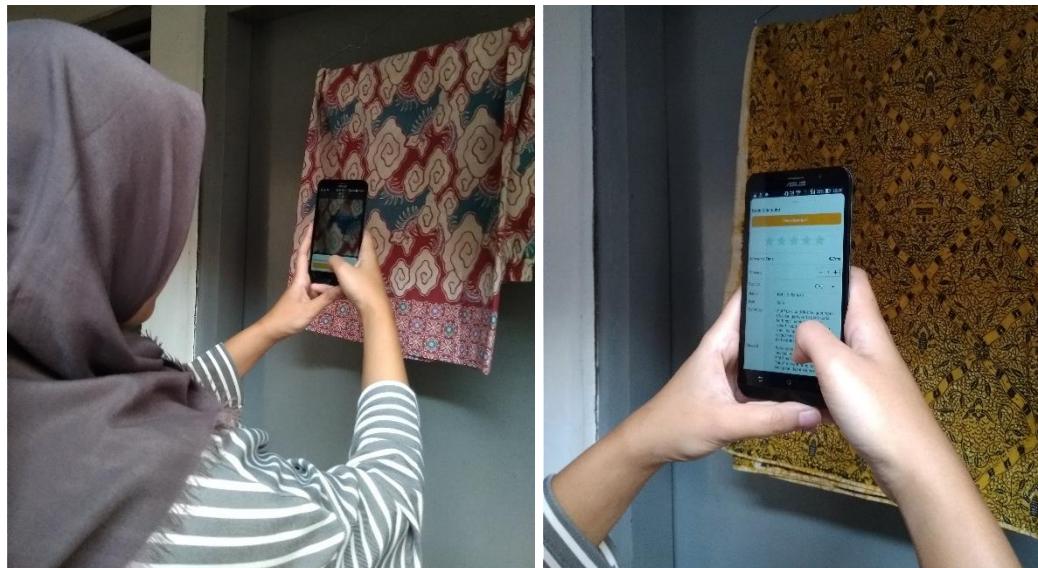
- A. Elrefaei, L., Omar Al-musawa, M., & Abdullah Al-gohany, N. (2017). Development of an Android Application for Object Detection Based on Color, Shape or Local Features. *The International Journal of Multimedia & Its Applications*, 9(1), 21–30. <https://doi.org/10.5121/ijma.2017.9103>
- Abas, M. A. H., Ismail, N., Yassin, A. I. M., & Taib, M. N. (2018). VGG16 for plant image classification with transfer learning and data augmentation. *International Journal of Engineering and Technology(UAE)*, 7(4), 90–94. <https://doi.org/10.14419/ijet.v7i4.11.20781>
- Abu, M. A., Indra, N. H., Rahman, A. H. A., Sapiee, N. A., & Ahmad, I. (2019). A study on image classification based on deep learning and tensorflow. *International Journal of Engineering Research and Technology*, 12(4), 563–569.
- Al Fikri, I. (2016). Aplikasi Navigasi Berbasis Perangkat Bergerak dengan Menggunakan Platform Wikitude untuk Studi Kasus Lingkungan ITS. *Jurnal Teknik ITS*, 5(1), 48–51. <https://doi.org/10.12962/j23373539.v5i1.14511>
- Alsing, O. (2018). *Mobile Object Detection using TensorFlow Lite and Transfer Learning*. KTH Royal Institute of Technology.
- Ananda, R. R. (2018). *Pengembangan Media Pembelajaran Scrum Interaktif Berbasis Aplikasi Android*. UIN Alauiddin Makassar.
- Danukusumo, K. P. (2017). In *Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU* (Vol. 91). UNIVERSITAS ATMA JAYA YOGYAKARTA.
- Dewi, S. R. (2018). Deep Learning Object Detection Pada Video. In *Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network*. https://dspace.uii.ac.id/bitstream/handle/123456789/7762/11242_Syarifah Rosita Dewi_Statistika.pdf?sequence=1
- Eka Putra, W. S. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, 5(1). <https://doi.org/10.12962/j23373539.v5i1.15696>
- Fitrianda, M. I. (2013). *Digital Repository Universitas Jember*. Jember Digital Repository Universitas Jember.
- Gomà, M. Y. (2016). *Image Recognition with Deep Learning Techniques and Tensorflow* [Universitat Politècnica de Catalunya (UPC) - BarcelonaTech]. <https://doi.org/10.1109/iciibms.2018.8549986>

- Gultom, Y., Arymurthy, A. M., & Masikome, R. J. (2018). Batik Classification using Deep Convolutional Network Transfer Learning. *Jurnal Ilmu Komputer Dan Informasi*, 11(2), 59. <https://doi.org/10.21609/jiki.v11i2.507>
- Hadinisa, S., Koyimatu, M., Irawan, A., Nugroho, H., Penelitian, M., Dasar, T., & Learning, M. (2018). *Analisis Learning Rate pada Metode Transfer Learning untuk Sistem Pendekripsi Api*. 8–11.
- Jalled, F., & Voronkov, I. (2016). *Object Detection using Image Processing*. 1–6.
- Kahamdany, H., Kusuma, D. P. D., & Osmond, A. B. (2018). Pengembangan Motif Karang Jenis Leptoseris Papyracea Pada Aplikasi Batik Berbasis Web. *E-Proceeding of Engineering*, 5(1), 1065–1071.
- Kusanti, J., & Agus, R. (2019). Application System for Identification of Surakarta Traditional Batik Images (SABATARA). *SinkrOn*, 4(1), 5. <https://doi.org/10.33395/sinkron.v4i1.10202>
- Lorentius, C. A., Adipranata, R., Tjondrowiguno, A., Studi, P., Informatika, T., Industri, F. T., Petra, U. K., & Siwalankerto, J. (2017). *Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network*.
- Masdiyasa, I. G. S., Bhirawa, S., & Winardi, S. (2019). *Identifikasi Plat Nomor Kendaraan Bermotor Menggunakan Metode Multi-Step Image Processing Berbasis Android*. V.
- Maulana, A. (2018). *Identifikasi Jenis Daging Menggunakan Image Processing Android* (Vol. 2). Universitas Lampung.
- Nadia, W. S., Ibrani, Y. D. H. E. Y., Eva, K., Dewi, C., Masripah, Vera, D., Fitriasari, D., Misran, M., Chabachib, M., Harum Sari Dewi, N., Pande Dwiana Putra, I., Subandono, S., Mikami, K., Mohri, H., Itoh, N., Miyashita, H., Hayashi, N., Sano, H., Lestari, P., ... Danamon, B. (2016). Rancang Bangun E-Voting Berbasis Website di Universitas Negeri Surabaya. *E-Jurnal Akuntansi*, 15(2), 1–23. <https://doi.org/10.33395/sinkron.v4i1.10202>
- Nuraedah, & Bakri, M. (2017). Klasifikasi Motif Kain Tradisional Batik Bomba Kaili. *Seminar Nasional Sistem Informasi UNMER Malang, 14 September*, 715–723.
- Nurhikmat, T. (2018). *Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek*. UNIVERSITAS ISLAM INDONESIA YOGYAKARTA.
- Pangestu, M. A., & Bunyamin, H. (2018). Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model. *Teknik Informatika Dan Sistem Informasi*, 4(2), 337–344.

- Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, 18(01), 15–21. <https://doi.org/10.23917/emitor.v18i01.6235>
- Sari, Y. (2018). Klasifikasi Pengenalan Motif Batik Berbasis Image Retrieval. *Jukung Jurnal Teknik Lingkungan*, 4(2), 27–33.
- Shaha, M., & Pawar, M. (2018). *Transfer Learning for Image Classification*. <https://doi.org/10.1109/ICECA.2018.8474802>
- Shu, M. (2019). Deep learning for image classification on very small datasets using transfer learning. *Creative Components*.
- Taskin, C. E. M., Yigit, A., Taskin, D., & Taskin, A. C. (2017). Image Processing on Android Devices With Opencv. *Techsys 2017, May*.
- Taufik, G. (2018). Extreme Programming Guna Rancang Bangun Sistem Informasi Pengarsipan Dokumen. *Bina Insani ICT Journal*, 5(1), 11–20.

LAMPIRAN

Lampiran 1 Dokumentasi Pengujian



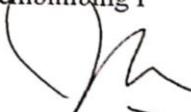
Lampiran 2 Lembar Kontrol Bimbingan Tugas Akhir

	LEMBAR KONTROL BIMBINGAN TUGAS AKHIR/SKRIPSI	No. PM Revisi Tanggal Halaman	7.5.18/L1 2 1 Juli 2019 1 / 4
---	---	--	--

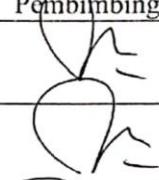
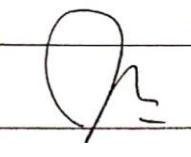
Nama : DHIYA FAUZIZA Kelas : IK-3B NIM : 3.34.17.1.09		Judul Tugas Akhir/Skripsi : APLIKASI INTERAKTIF PENGENALAN BATIK UNTUK PENGUNJUNG PAMERAN BERBASIS ANDROID DAN IMAGE PROCESSING	
No	Tanggal	Uraian	Tandatangan Pembimbing
1.	09 Juni 2020	- Pembuatan Logbook - Konsultasi Survei Lapangan	
2.	15 Juli 2020	- Alur kerja sistem	
3.	03 Agustus 2020	- Fitur vote untuk batik - Rujukan daftar pustaka harus jelas - Mencari di pustaka/jurnal tentang dataset	
4.	17 Agustus 2020	- Demo fitur event pada android - Dataset harus harus dilakukan rotate - Menambahkan informasi yang ditampilkan saat capture batik seperti filosofi dan ciri khas	
5.	24 Agustus 2020	- Mencari metode untuk pembagian dataset training dan testing - Penambahan fitur edit pada komentar	
6.	07 September 2020	- Penambahan informasi yang ditampilkan saat proses scan qrcode event - Menambah pengolahan dataset seperti gambar harus dirotate dan flip	

7.	14 September 2020	<ul style="list-style-type: none"> - Menambahkan informasi berupa email dan no telp pada buku tamu - Mengubah flow tentang sistem qrcode event, jadi generate qrcode berada pada penyelenggara bukan pengunjung 	
8.	16 September 2020	<ul style="list-style-type: none"> - Menambahkan fitur agar qrcode dapat dicetak 	
9.	19 September 2020	<ul style="list-style-type: none"> - Laporan Bab 1 - Perbaikan pada abstrak - Laporan Bab 2 - Laporan Bab 3 	
10.	26 September 2020	<ul style="list-style-type: none"> - Laporan Bab 4 dan 5 - Menambahkan hasil pengujian secara real di objek batik 	
11.	02 Oktober 2020	<ul style="list-style-type: none"> - Revisi laporan Bab 4 - Revisi laporan Bab 5 	
12.	05 Oktober 2020	<ul style="list-style-type: none"> - Perbaikan mengenai cara pengambilan data pengujian di objek batik - Perbaikan tampilan mengenai besar akurasi tidak usah ditampilkan di android 	
13.	08 Oktober 2020	<ul style="list-style-type: none"> - Revisi akhir laporan 	

Semarang, 09 Oktober 2020
Pembimbing I

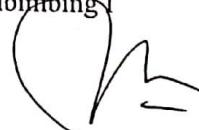

Mardiyono, S.Kom., M.Sc.
NIP. 196810252000121001

	LEMBAR KONTROL BIMBINGAN TUGAS AKHIR/SKRIPSI	No. PM Revisi Tanggal Halaman	7.5.18/L1 2 1 Juli 2019 2 / 4
---	---	--	--

Nama : ZANI KOLINA Kelas : IK-3B NIM : 3.34.17.1.26		Judul Tugas Akhir/Skripsi : APLIKASI INTERAKTIF PENGENALAN BATIK UNTUK PENGUNJUNG PAMERAN BERBASIS ANDROID DAN <i>IMAGE PROCESSING</i>	
No	Tanggal	Uraian	Tandatangan Pembimbing
1.	09 Juni 2020	<ul style="list-style-type: none"> - Pembuatan Logbook - Konsultasi Survei Lapangan 	
2.	15 Juli 2020	<ul style="list-style-type: none"> - Alur kerja sistem 	
3.	03 Agustus 2020	<ul style="list-style-type: none"> - Fitur vote untuk batik - Rujukan daftar pustaka harus jelas - Mencari di pustaka/jurnal tentang dataset 	
4.	17 Agustus 2020	<ul style="list-style-type: none"> - Demo fitur event pada android - Dataset harus harus dilakukan rotate - Menambahkan informasi yang ditampilkan saat capture batik seperti filosofi dan ciri khas 	
5.	24 Agustus 2020	<ul style="list-style-type: none"> - Mencari metode untuk pembagian dataset training dan testing - Penambahan fitur edit pada komentar 	
6.	07 September 2020	<ul style="list-style-type: none"> - Penambahan informasi yang ditampilkan saat proses scan qrcode event - Menambah pengolahan dataset seperti gambar harus dirotate dan flip 	
7.	14 September 2020	<ul style="list-style-type: none"> - Menambahkan informasi berupa email dan no telp pada 	

		<ul style="list-style-type: none"> - buku tamu - Mengubah flow tentang sistem qrcode event, jadi generate qrcode berada pada penyelenggara bukan pengunjung 	
8.	16 September 2020	<ul style="list-style-type: none"> - Menambahkan fitur agar qrcode dapat dicetak 	
9.	19 September 2020	<ul style="list-style-type: none"> - Laporan Bab 1 - Perbaikan pada abstrak - Laporan Bab 2 - Laporan Bab 3 	
10.	26 September 2020	<ul style="list-style-type: none"> - Laporan Bab 4 dan 5 - Menambahkan hasil pengujian secara real di objek batik 	
11.	02 Oktober 2020	<ul style="list-style-type: none"> - Revisi laporan Bab 4 - Revisi laporan Bab 5 	
12.	05 Oktober 2020	<ul style="list-style-type: none"> - Perbaikan mengenai cara pengambilan data pengujian di objek batik - Perbaikan tampilan mengenai besar akurasi tidak usah ditampilkan di android 	
13.	08 Oktober 2020	<ul style="list-style-type: none"> - Revisi akhir laporan 	

Semarang, 09 Oktober 2020
 Pembimbing I

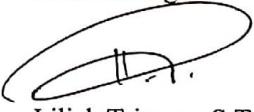


Mardiyono, S.Kom., M.Sc.
 NIP. 196810252000121001

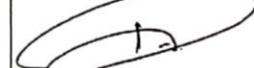
	LEMBAR KONTROL	No. PM	7.5.18/L1
	BIMBINGAN	Revisi	2
	TUGAS AKHIR/SKRIPSI	Tanggal	1 Juli 2019
		Halaman	3 / 4

Nama : DHIYA FAUZIZA		Judul Tugas Akhir/Skripsi : APLIKASI INTERAKTIF PENGENALAN BATIK UNTUK PENGUNJUNG PAMERAN BERBASIS ANDROID DAN IMAGE PROCESSING	
No	Tanggal	Uraian	Tandatangan Pembimbing
1.	14 Juni 2020	<ul style="list-style-type: none"> - Konsultasi mengenai alur sistem - Penyampaian progress website - Revisian form modal input 	
2.	13 Agustus 2020	<ul style="list-style-type: none"> - Penyampaian perbaikan website - Diskusi tentang dataset - Penambahan dataset 	
3.	27 Agustus 2020	<ul style="list-style-type: none"> - Penyampaian progress perbaikan dataset - Menampilkan akurasi dan teknik dari dataset 	
4.	3 September 2020	<ul style="list-style-type: none"> - Mengubah flow buku tamu event - Revisian dataset 	
5.	10 September 2020	<ul style="list-style-type: none"> - Penyampaian progress aplikasi event - Demo aplikasi android 	
6.	24 September 2020	<ul style="list-style-type: none"> - Demo qrcode event - Demo website 	
7.	05 Oktober 2020	<ul style="list-style-type: none"> - Revisian laporan bab 1-5 	
8.	07 Oktober 2020	<ul style="list-style-type: none"> - Revisian laporan bab 1-5 	

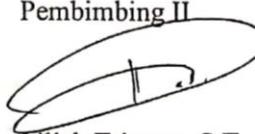
Semarang, 09 Oktober 2020
Pembimbing II


Lilik Triyono, S.T., M.Kom
NIP. 198404202015041003

	LEMBAR KONTROL BIMBINGAN TUGAS AKHIR/SKRIPSI	No. PM	7.5.18/L1
		Revisi	2
		Tanggal	1 Juli 2019
		Halaman	4 / 4

Nama : ZANI KOLINA Kelas : IK-3B NIM : 3.34.17.1.26		Judul Tugas Akhir/Skripsi : APLIKASI INTERAKTIF PENGENALAN BATIK UNTUK PENGUNJUNG PAMERAN BERBASIS ANDROID DAN <i>IMAGE PROCESSING</i>	
No	Tanggal	Uraian	Tandatangan Pembimbing
4.	14 Juni 2020	<ul style="list-style-type: none"> - Konsultasi mengenai alur sistem - Penyampaian progress website - Revisian form modal input 	
5.	13 Agustus 2020	<ul style="list-style-type: none"> - Penyampaian perbaikan website - Diskusi tentang dataset - Penambahan dataset 	
6.	27 Agustus 2020	<ul style="list-style-type: none"> - Penyampaian progress perbaikan dataset - Menampilkan akurasi dan teknik dari dataset 	
4.	3 September 2020	<ul style="list-style-type: none"> - Mengubah flow buku tamu event - Revisian dataset 	
5.	10 September 2020	<ul style="list-style-type: none"> - Penyampaian progress aplikasi event - Demo aplikasi android 	
6.	24 September 2020	<ul style="list-style-type: none"> - Demo qrcode event - Demo website 	
7.	05 Oktober 2020	<ul style="list-style-type: none"> - Revisian laporan bab 1-5 	
8.	07 Oktober 2020	<ul style="list-style-type: none"> - Revisian laporan bab 1-5 	

Semarang, 09 Oktober 2020
Pembimbing II


Liliek Triyono, S.T., M.Kom

NIP. 198404202015041003

Lampiran 3 Form Revisi Tugas Akhir

	LEMBAR REVISI TUGAS AKHIR	No. PM 7.5.21/L3 Revisi 2 Tanggal 1 Juli 2010 Halaman 1/ 3
---	----------------------------------	---

No	Uraian	Tandatangan Penguji
1.	Waktu sprint diubah	

Semarang, 13 Oktober 2020

Penguji I,



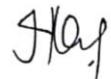
Wahyu Sulistiyo, S.T., M.Kom.
NIP. 197704012005011001

	LEMBAR REVISI TUGAS AKHIR	No. PM	7.5.21/L3
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	2/ 3

No	Uraian	Tandatangan Penguji
1.	Estimasi waktu dan waktu yang digunakan dikonversi ke minggu, di masukkan dalam kesimpulan	

Semarang, 13 Oktober 2020

Penguji II,



Idhawati Hestiningsih, S.Kom., M.Kom.
NIP. 197711192008012013

	LEMBAR REVISI TUGAS AKHIR	No. PM	7.5.21/L3
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	3 / 3

No	Uraian	Tandatangan Pengaji
1.	Tambahkan di dataset batik Semarang, Pekalongan, Lasem	
2.	Backlog di metode sprint diperkecil supaya waktunya tidak terlalu lama dalam penyelesaian 1 sprint	

Semarang, 13 Oktober 2020

Pengaji III,



Tri Raharjo Yudantoro, S.Kom.,M.Kom.
NIP. 196810252000121001

Lampiran 4 Surat Keterangan Selesai Revisi Tugas Akhir

	REVISI TUGAS AKHIR	No. FPM	7.5.21/L3
		Revisi	2
		Tanggal	1 Juli 2010
		Halaman	1 / 1

Yang bertanda tangan di bawah ini, Pengaji I/II/III menerangkan bahwa:

No.	Nama	NIM	Kelas
1	Dhiya Fauziza	3.34.17.1.09	IK-3B
2	Zani Kolina	3.34.17.1.26	IK-3B

Dengan judul Tugas Akhir:

Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan *Image Processing*

Benar-benar telah melaksanakan revisi tugas akhir. Surat keterangan ini dapat dipergunakan sebagaimana mestinya.

Semarang, 13 Oktober 2020

Pengaji I,



Wahyu Sulistiyo, S.T., M.Kom.
NIP. 197704012005011001

Pengaji II,



Idhwati H, S.Kom., M.Kom.
NIP. 197711192008012013

Pengaji III,



Tri Ral tarjo Yudantoro, S.Kom., M.Kom.
NIP. 196810252000121001

Lampiran 5 Kode Pengujian Import Package dan Library

```

!pip install -
q git+https://github.com/tensorflow/examples.git#egg=tensorflow-
examples[model_maker]

import numpy as np
import tensorflow as tf
assert tf.__version__.startswith('2')

from tensorflow_examples.lite.model_maker.core.data_util.image_dataloader imp
ort ImageClassifierDataLoader
from tensorflow_examples.lite.model_maker.core.task import image_classifier
from tensorflow_examples.lite.model_maker.core.task.configs import Quantizati
onConfig
from tensorflow_examples.lite.model_maker.core.task.model_spec import (mobile
net_v2_spec,
    efficientnet_lite0_spec,
    efficientnet_lite1_spec,
    efficientnet_lite2_spec,
    efficientnet_lite3_spec,
    efficientnet_lite4_spec,
    resnet_50_spec)
from tensorflow_examples.lite.model_maker.core.task.model_spec import ImageMo
delSpec

import matplotlib.pyplot as plt
import zipfile
from google.colab import drive
drive.mount('/content/drive/')

zip_ref = zipfile.ZipFile("/content/drive/My Drive/Dataset_Batik/BATIK.zip",
'r')
zip_ref.extractall("/tmp")
zip_ref.close()

import os
base_dir = '/tmp/BATIK'
print(os.listdir(base_dir))

```

Lampiran 6 Kode Pengujian Pembagian Data

```

data = ImageClassifierDataLoader.from_folder(base_dir)
train_data, test_data = data.split(0.9)

plt.figure(figsize=(10,10))
for i, (image, label) in enumerate(data.dataset.take(25)):
    plt.subplot(5,5,i+1)

```

```

plt.xticks([])
plt.yticks([])
plt.grid(False)
plt.imshow(image.numpy(), cmap=plt.cm.gray)
plt.xlabel(data.index_to_label[label.numpy()])
plt.show()

```

Lampiran 7 Kode Pengujian *Train Model*

```

EPOCHS = 10

model1 = image_classifier.create(train_data, model_spec=mobilenet_v2_spec,
                                 epochs=EPOCHS, use_augmentation=True, train_
whole_model=True)

```

Lampiran 8 Kode Pengujian *Testing Model*

```

# Evaluate the model
loss_list = list()
acc_test_list = list()

for model in models_list:
    loss, accuracy = model.evaluate(test_data)
    loss_list.append(loss)
    acc_test_list.append(accuracy)

```

Lampiran 9 Kode Pengujian *Export Model*

```

for model, name in zip(models_list, model_names):
    name = name.replace(" ", "_").lower()

    # Export as TensorFlow Lite model.
    model.export(export_dir='./'+name)

```

Lampiran 10 Kode Halaman *Onboarding*

File OnboardingActivity.java

```

dotsLayout.removeAllViews();
dots = new TextView[3];
for (int i = 0; i < dots.length; i++) {
    dots[i] = new TextView(this);
    //this html code creates dot
    dots[i].setText(Html.fromHtml("&#8226"));
    dots[i].setTextSize(35);
    dots[i].setTextColor(getResources().getColor(R.color.colorWhite));
    dotsLayout.addView(dots[i]);
}
if(dots.length>0){
dots[position].setTextColor(getResources().getColor(R.color.colorWhite));
}
}

```

Lampiran 11 Kode Halaman *Splash Screen*

File SplashActivity.java

```
public void run() {
    SharedPreferences userPref =
    getApplicationContext().getSharedPreferences("user", Context.MODE_PRIVATE);
    boolean isLoggedIn = userPref.getBoolean("isLoggedIn", false);
    if (isLoggedIn){
        startActivity(new
Intent(SplashActivity.this,HomeActivity.class));
        finish();
    }
    else {
        isFirstTime();
    }
},1500);
```

Lampiran 12 Kode Halaman *Login*

File AuthController.php

```
public function login(Request $request){
    $creds = $request->only(['email','password']);
    if(!$token=auth()->attempt($creds)){
        return response()->json([
            'success' => false,
            'message' => 'invalid credentials'
        ]);
    }
    return response()->json([
        'success' =>true,
        'token' => $token,
        'pengunjung' => Auth::user('pengunjung')
    ]);
}
```

File SignInFragment.java

```
private void login(){
    dialog.setMessage("Logging in");
    dialog.show();
    StringRequest request = new StringRequest(Request.Method.POST,
Constant.LOGIN,response -> {
        try {
            JSONObject object = new JSONObject(response);
            if (object.getBoolean("success")){
                JSONObject pengunjung = object.getJSONObject("pengunjung");
                SharedPreferences userPref =
getActivity().getApplicationContext().getSharedPreferences("pengunjung",getCo
ntext().MODE_PRIVATE);
                SharedPreferences.Editor editor = userPref.edit();
                editor.putString("token",object.getString("token"));
                editor.putString("name",pengunjung.getString("name"));
                editor.putInt("id",pengunjung.getInt("id"));
                editor.putString("email",pengunjung.getString("email"));
            }
        }
    });
}
```

```

editor.putString("tgl_lahir",pengunjung.getString("tgl_lahir"));
    editor.putString("jk",pengunjung.getString("jk"));
    editor.putString("path",pengunjung.getString("path"));
    editor.putBoolean("isLoggedIn",true);
    editor.apply();
    startActivity(new Intent(((AuthActivity)getContext()),
HomeActivity.class));
    ((AuthActivity) getContext()).finish();
    Toast.makeText(getContext(),"Login sukses",
Toast.LENGTH_SHORT).show();
}
} catch (JSONException e) {
    e.printStackTrace();
}
dialog.dismiss();
RequestQueue queue = Volley.newRequestQueue(getContext());
queue.add(request);
}
}

```

Lampiran 13 Kode Halaman Register

File AuthController.php

```

public function register(Request $request){
    $encryptedPass = Hash::make($request->password);
    $pengunjung = new Pengunjung();

    try{
        $pengunjung->email = $request->email;
        $pengunjung->password = $encryptedPass;
        $pengunjung->save();
        return $this->login($request);
    }
    catch(Exception $e){
        return response()->json([
            'success' => false,
            'message' => ''.$e
        ]);
    }
}

```

File SignUpFragment.java

```

private void register(){
    dialog.setMessage("Registering");
    dialog.show();
    StringRequest request = new StringRequest(Request.Method.POST,
Constant.REGISTER, response -> {
        try {
            JSONObject object = new JSONObject(response);
            if (object.getBoolean("success")){
                JSONObject pengunjung = object.getJSONObject("pengunjung");
                Sharedpreferences userPref =
getActivity().getApplicationContext().getSharedpreferences("pengunjung",getCo
nText().MODE_PRIVATE);
                Sharedpreferences.Editor editor = userPref.edit();
                editor.putString("token",object.getString("token"));
            }
        }
    });
}

```

```

        editor.putString("name", pengunjung.getString("name"));
        editor.putInt("id", pengunjung.getInt("id"));
        editor.putString("email", pengunjung.getString("email"));

        editor.putString("tgl_lahir", pengunjung.getString("tgl_lahir"));
        editor.putString("jk", pengunjung.getString("jk"));
        editor.putString("path", pengunjung.getString("path"));
        editor.putBoolean("isLoggedIn", true);
        editor.apply();
        startActivity(new Intent(((AuthActivity)getContext()),
        UserInfoActivity.class));
        ((AuthActivity) getContext()).finish();
        Toast.makeText(getContext(), "Register Success",
        Toast.LENGTH_SHORT).show();
    }
} catch (JSONException e) {
    e.printStackTrace();
}
dialog.dismiss();
RequestQueue queue = Volley.newRequestQueue(getContext());
queue.add(request);
}
}

```

Lampiran 14 Kode Halaman Data Pengguna

File AuthController.php

```

public function saveUserInfo(Request $request){
    $pengunjung = Pengunjung::find(Auth::user('pengunjung')->id);
    $pengunjung->name = $request->name;
    $pengunjung->tgl_lahir = $request->tgl_lahir;
    $pengunjung->jk = $request->jk;
    $pengunjung->no_hp = $request->no_hp;
    $path = '';
    if($request->path!=''){
        $path = time().'.' .jpg';
        file_put_contents('storage/pengunjungs/' . $path, base64_decode($request->path));
        $pengunjung->path = $path;
    }
    $pengunjung->update();
    return response()->json([
        'success' => true,
        'path' => $path
    ]);
}

```

File UserInfoActivity.java

```

private void init() {
    dialog = new ProgressDialog(this);
    dialog.setCancelable(false);
    userPref = getApplicationContext().getSharedPreferences("pengunjung",
    Context.MODE_PRIVATE);
    layoutTglLahir = findViewById(R.id.txtLayoutTglLahirUserInfo);
}

```

```
layoutName = findViewById(R.id.txtLayoutNameUserInfo);
layoutJK = findViewById(R.id.txtLayoutJKUserInfo);
layoutHp = findViewById(R.id.txtLayoutHpUserInfo);
txtName = findViewById(R.id.txtNameUserInfo);
txtHp = findViewById(R.id.txtHpUserInfo);
male = findViewById(R.id.txtmale);
female = findViewById(R.id.txtfamale);
txtSelectPhoto = findViewById(R.id.txtSelectPhoto);
btnContinue = findViewById(R.id.btnContinue);
circleImageView = findViewById(R.id.imgUserInfo);

//pick photo from gallery
txtSelectPhoto.setOnClickListener(v->{
    Intent i = new Intent(Intent.ACTION_PICK);
    i.setType("image/*");
    startActivityForResult(i,GALLERY_ADD_PROFILE);
});

btnContinue.setOnClickListener(v->{
    // validate fields
    if(validate()){
        saveUserInfo();
    }
});
});
```

Lampiran 15 Kode Halaman *Home*

File BeritaController.php

```
public function beritas(){
    return response()->json([
        'success' => true,
        'beritas' => $beritas
    ]);
}
```

File HomeActivity.java

```
public class HomeActivity extends AppCompatActivity{
    private FragmentManager fragmentManager;
    private FloatingActionButton fab;
    private BottomNavigationView navigationView;
    private static final int GALLERY_ADD_POST = 2;
    private ImageView btnEventt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        fragmentManager = getSupportFragmentManager();
        fragmentManager.beginTransaction().replace(R.id.frameHomeContainer,new
HomeFragment(),HomeFragment.class.getSimpleName()).commit();
        init();
    }
}
```

Lampiran 16 Kode Halaman *Event* Pameran

File EventController.php

```

public function events(){
    $events = Event::where('status', '1')->orderBy('tgl_akhir','desc')-
>orderBy('id','desc')->get();

    foreach($events as $event){
        $event->penyelenggara;
        $event['commentsCount'] = count($event->comments);
        $event['likesCount'] = count($event->likes);
        $event['daftarCount'] = count($event->presensis);
        $event['selfLike'] = false;

        $dt = Carbon::now();
        if($dt > $event['tgl_akhir']){
            $event['closed'] = true;
        }else{
            $event['closed'] = false;
        }
        $event['today']= $dt->toDateString();

        foreach($event->likes as $like){
            if($like->id_pengunjung == Auth::user('pengunjung')->id){
                $event['selfLike'] = true;
            }
        }
        $event['selfPresensi'] = false;
        foreach($event->presensis as $presensi){
            if($presensi->id_pengunjung == Auth::user('pengunjung')->id){
                $event['selfPresensi'] = true;
            }
        }
    }
    return response()->json([
        'success' => true,
        'events' => $events
    ]);
}

```

File FirstEventActivity.java

```

if (object.getBoolean("success")) {
    JSONArray array = new
JSONArray(object.getString("events"));
    for (int i = 0; i < array.length(); i++) {
        JSONObject eventObject = array.getJSONObject(i);
        JSONObject eventPenyelenggaraObject =

```

```

eventObject.getJSONObject("penyelenggara");
    Penyelenggara penyelenggara = new Penyelenggara();
    penyelenggara.setId(eventPenyelenggaraObject.getInt("id"));
    penyelenggara.setOrganisasi(eventPenyelenggaraObject.getString("organisasi"))
;
    Event event = new Event();
    event.setId(eventObject.getInt("id"));
    event.setPenyelenggara(penyelenggara);
    event.setJudul(eventObject.getString("judul"));
    event.setTgl_mulai(eventObject.getString("tgl_mulai")
+ " - " + eventObject.getString("tgl_akhir"));
    event.setJam_mulai(eventObject.getString("jam_mulai")
+ " - " + eventObject.getString("jam_akhir"));
    event.setLokasi(eventObject.getString("lokasi"));
    event.setPhoto(eventObject.getString("photo"));
    event.setLikes(eventObject.getInt("likesCount"));
    event.setComments(eventObject.getInt("commentsCount"));
    event.setSelfLike(eventObject.getBoolean("selfLike"));
    event.setSelfPresensi(eventObject.getBoolean("selfPresensi"));
    event.setClosed(eventObject.getString("closed"));
    arrayList.add(event);
}
eventAdapter = new EventAdapter(getApplicationContext(), arrayList);
recyclerView.setAdapter(eventAdapter);
}

```

Lampiran 17 Kode Halaman Komentar Event

File CommentController.php

```

public function comments(Request $request){
    $comments = Comment::where('id_event',$request->id_event)->get();
    foreach($comments as $comment){
        $comment->pengunjung;
    }
    return response()->json([
        'success' => true,
        'comments' => $comments
    ]);
}

```

File CommentActivity.java

```

private void getComments() {
    list = new ArrayList<>();
    StringRequest request = new
StringRequest(Request.Method.POST,Constant.COMMENTS,res->{
    try {
        JSONObject object = new JSONObject(res);
        if (object.getBoolean("success")){
            JSONArray comments = new
JSONArray(object.getString("comments"));
            for (int i = 0; i < comments.length(); i++) {
                JSONObject comment = comments.getJSONObject(i);
                JSONObject pengunjung = comment.getJSONObject("pengunjung");
                Pengunjung mPengunjung = new Pengunjung();
                mPengunjung.setId(pengunjung.getInt("id"));
                mPengunjung.setPhoto(Constant.URL+"storage/pengunjungs/"+pengunjung.getString

```

```

("path"));

        mPengunjung.setUsername(pengunjung.getString("name"));
        Comment mComment = new Comment();
        mComment.setId(comment.getInt("id"));
        mComment.setPengunjung(mPengunjung);
        mComment.setDatae(comment.getString("created_at"));
        mComment.setComment(comment.getString("comment"));
        list.add(mComment);
    }
}

adapter = new CommentAdapter(this,list);
recyclerView.setAdapter(adapter);

} catch (JSONException e) {
    e.printStackTrace();
}

RequestQueue queue = Volley.newRequestQueue(CommentActivity.this);
queue.add(request);
}
}

```

Lampiran 18 Kode Halaman Mengedit Komentar *Event*

File CommentController.php

```

public function update(Request $request){
    $comment = Comment::find($request->id_comment);
    if($comment->id_pengunjung != Auth::user('pengunjung')->id){
        return response()->json([
            'success' => false,
            'message' => 'unauthorize access'
        ]);
    }
    $comment->comment = $request->comment;
    $comment->update();

    return response()->json([
        'success' => true,
        'message' => 'comment edited'
    ]);
}

```

File CommentAdapter.java

```

public void updateComment(int position,Comment comment){

    list.remove(position);
    list.add(comment);
    notifyDataSetChanged();
    notifyDataSetChanged();
}

```

Lampiran 19 Kode Halaman Menghapus Komentar Event

File CommentController.php

```
public function delete(Request $request){
    $comment = Comment::find($request->id_comment);
    if($comment->id_pengunjung != Auth::user('pengunjung')->id){
        return response()->json([
            'success' => false,
            'message' => 'unauthorize access'
        ]);
    }
    $comment->delete();

    return response()->json([
        'success' => true,
        'message' => 'comment deleted'
    ]);
}
```

File CommentAdapter.java

```
holder.btnDelete.setOnClickListener(v->{
    PopupMenu popupMenu = new PopupMenu(context,holder.btnDelete);
    popupMenu.inflate(R.menu.menu_comment_options);
    popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem item) {

            switch (item.getItemId()){
                case R.id.item_delete: {
                    AlertDialog.Builder builder = new
AlertDialog.Builder(context);
                    builder.setMessage("Are you sure?");
                    builder.setPositiveButton("Delete", new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            deleteComment(comment.getId(),position);
                        }
                    });
                    builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            } });
                    builder.show();
                    return true;
                } }
            return false;
        }});
    popupMenu.show();
});
```

Lampiran 20 Kode Halaman *Scan Qrcode Event*

File ScanActivity.java

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    IntentResult result = IntentIntegrator.parseActivityResult(requestCode,
    resultCode, data);
    if (result != null){
        if (result.getContents() == null){
            Toast.makeText(this, "Hasil tidak ditemukan",
            Toast.LENGTH_SHORT).show();
        }else{
            try{
                JSONObject object = new JSONObject(result.getContents());
                textViewID.setText(object.getString("id"));
                textViewEvent.setText(object.getString("judul"));
                textViewPenyelenggara.setText(object.getString("penyelenggara"));
                textViewLokasi.setText(object.getString("lokasi"));
                textViewDeskripsi.setText(Html.fromHtml(object.getString("deskripsi")));
                buttonDaftar.setVisibility(View.VISIBLE);
            }catch (JSONException e){
                e.printStackTrace();
            }
        }
    }
    super.onActivityResult(requestCode, resultCode, data);
}
}

```

Lampiran 21 Kode Halaman Konfirmasi *Event*

File PresensiController.php

```

public function daftarr(Request $request){
    $presensi = Presensi::where('id_event',$request->id_event)->where('id_pengunjung',Auth::user('pengunjung')->id)->get();

    if(count($presensi)>0){
        return response()->json([
            'success' => false,
            'message' => 'Sudah Terdaftar'
        ]);
    }
    $presensi = new Presensi();
    $presensi->id_pengunjung = Auth::user('pengunjung')->id;
    $presensi->id_event = $request->id_event;
    $presensi->save();
}

```

```

        return response()->json([
            'success' => true,
            'message' => 'hadir',
            'presensis' => $presensi
        ]);
    }
}

File ScanActivity.java

private void daftar(){
    dialog.setMessage("Daftar");
    dialog.show();

    StringRequest request = new
StringRequest(Request.Method.POST,Constant.DAFTARR, response -> {

        try {
            JSONObject object = new JSONObject(response);
            if (object.getBoolean("success") == true){
                JSONObject presensiObject =
object.getJSONObject("presensis");
                startActivity(intent);
                Toast.makeText(this, "Terdaftar",
Toast.LENGTH_SHORT).show();
                finish();
            }else{
                Toast.makeText(this, "Anda Sudah Terdaftar!!!",
Toast.LENGTH_SHORT).show();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        dialog.dismiss();
    }
    RequestQueue queue = Volley.newRequestQueue(ScanActivity.this);
    queue.add(request);
}

```

Lampiran 22 Kode Halaman Deteksi Batik

File MotifController.php

```

class MotifController extends Controller
{
    public function motifs(){
        $motifs = Motif::orderBy('id','desc')->get();

        return response()->json([
            'success' => true,
            'motifs' => $motifs
        ]);
    }

    public function rating(Request $request){

```

```

$rating = Rating::where('id_motif',$request->id_motif)-
>where('id_pengunjung',Auth::user('pengunjung')->id)->get();

if(count($rating)>0){

    $rating[0]->delete();
    $rating = new Rating();
    $rating->id_pengunjung = Auth::user('pengunjung')->id;
    $rating->id_motif = $request->id_motif;
    $rating->rate = $request->rate;
    $rating->save();
    return response()->json([
        'success' => true,
        'message' => 'updated',
        'ratings' => $rating
    ]);
}

$rating = new Rating();
$rating->id_pengunjung = Auth::user('pengunjung')->id;
$rating->id_motif = $request->id_motif;
$rating->rate = $request->rate;
$rating->save();

return response()->json([
    'success' => true,
    'message' => 'successfully',
    'ratings' => $rating
]);
}

public function viewRating(Request $request){
    $ratings = Rating::where('id_motif',$request->id_motif)-
>where('id_pengunjung',Auth::user('pengunjung')->id)->get();
    foreach($ratings as $rating){
        $rating->motif;
    }
    return response()->json([
        'success' => true,
        'ratings' => $ratings
    ]);
}
}

```

File CameraActivity.java

```

@UiThread
protected void showResultsInBottomSheet(List<Recognition> results) {
    if (results != null && results.size() >= 3) {
        Recognition recognition = results.get(0);
        if (recognition != null) {
            if (recognition.getTitle() != null)
                recognitionTextView.setText(recognition.getTitle());
            if (recognition.getConfidence() != null)
                recognitionValueTextView.setText(
                    String.format("%.2f", (100 * recognition.getConfidence())) + "%");
        }
    }

    private void ratingBatik(){
        String IDMotif = tv_id.getText().toString();
        String ratinggg = String.valueOf(ratingBar.getRating());

        StringRequest request = new
StringRequest(Request.Method.POST, Constant.MOTIF_RATING,res->{
            try {
                JSONObject object = new JSONObject(res);
                if (object.getBoolean("success")){
                    Toast.makeText(getApplicationContext(),
ratinggg,Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            dialog.dismiss();
        }
        RequestQueue queue = Volley.newRequestQueue(CameraActivity.this);
        queue.add(request);
    }
}
}

```

Lampiran 23 Kode Halaman Profil**File AuthController.php**

```

public function logout(Request $request){
    try{
        JWTAuth::invalidate(JWTAuth::parseToken($request->token));
        return response()->json([
            'success' => true,
            'message' => 'logout success'
        ]);
    }
    catch(Exception $e){
        return response()->json([
            'success' => false,
            'message' => ''.$e
        ]);
    }
}

```

File AccountFragment.java

```

private void init() {
    preferences = getContext().getSharedPreferences("pengunjung",
Context.MODE_PRIVATE);
    toolbar = view.findViewById(R.id.toolbarAccount);
    ((HomeActivity)getContext()).setSupportActionBar(toolbar);
    setHasOptionsMenu(true);
    imgProfile = view.findViewById(R.id.imgAccountProfile);
    txtName = view.findViewById(R.id.txtAccountName);
    txtPostsCount = view.findViewById(R.id.txtAccountPresensiCount);
    recyclerView = view.findViewById(R.id.recyclerAccount);
    btnEditAccount = view.findViewById(R.id.btnEditAccount);
    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
}

btnEditAccount.setOnClickListener(v->{
    Intent i = new Intent((HomeActivity)getContext(),
EditUserInfoActivity.class);
    i.putExtra("imgUrl", imgUrl);
    startActivity(i);
});
}
}

```

Lampiran 24 Kode Halaman Login**File LoginController.php**

```

public function postLogin(Request $request)
{
    $validator = $this->validate($request, [
        'email' => 'required|email',
        'password' => 'required'
    ]);
    if (Auth::guard('penyelenggara')->attempt(['email' => $request-
>email, 'password' => $request->password])) {
        return redirect()->intended('/penyelenggaraa');
    } else if (Auth::guard('user')->attempt(['email' => $request-
>email, 'password' => $request->password])) {
        return redirect()->intended('/user');
    } else{
        return redirect()
            ->back()
            ->withInput()
            ->with('statusError','Incorrect user login details!');
    }
}
}

```

Lampiran 25 Kode Halaman *Dashboard*

File HomeController.php

```
public function index()
{
    $motif = Motif::count();
    $event = Event::count();
    $user = Pengunjung::count();
    return view('home', compact('motif', 'event', 'user'));
}
```

Lampiran 26 Kode Halaman Menampilkan Informasi Motif Batik

File MotifBatikController.php

```
public function index()
{
    $motifs = Motif::orderBy('created_at', 'DESC')->paginate();
    foreach($motifs as $motif){
        $motif['countRating'] = $motif->ratings->avg('rate');
    }
    return view('pages.motif.view', compact('motifs'));
}
```

Lampiran 27 Kode Halaman Menambahkan Informasi Motif Batik

File MotifBatikController.php

```
public function store(Request $request)
{
    $this->validate($request, [
        'motif' => ['required'],
        'asal' => ['required'],
        'ciri' => ['required'],
        'filosofi' => ['required'],
        'created_by',
        'updated_by'
    ]);
    $fill_data = ([
        'motif' => $request->motif,
        'asal' => $request->asal,
        'ciri' => $request->ciri,
        'filosofi' => $request->filosofi,
        'created_by' => Auth::user()->id,
        'updated_by' => Auth::user()->id
    ]);
    $motif = Motif::create($fill_data);
    return redirect()->route('motif.index')-
>with('statusEdit', 'Data Motif Created!');
```

Lampiran 28 Kode Halaman Mengedit Informasi Motif Batik

File MotifBatikController.php

```
public function update(Request $request, $id)
{
    $motif = Motif::findOrFail($id);
    $this->validate($request, [
        'motif' => ['required'],
        'asal' => ['required'],
        'ciri' => ['required'],
        'filosofi' => ['required'],
        'created_by',
        'updated_by'
    ]);
    $fill_data = ([
        'motif' => $request->motif,
        'asal' => $request->asal,
        'ciri' => $request->ciri,
        'filosofi' => $request->filosofi,
        'created_by' => Auth::user()->id,
        'updated_by' => Auth::user()->id
    ]);
    $motif->update($fill_data);
    return redirect()->route('motif.index')-
>with('statusEdit', 'Data Motif Updated');
}
```

Lampiran 29 Kode Halaman Menampilkan Berita

File BeritaController.php

```
public function index()
{
    $beritas = Berita::orderBy('created_at', 'DESC')->paginate();
    return view('pages.berita.view', compact('beritas'));
}
```

Lampiran 30 Kode Halaman Menambahkan Berita

File BeritaController.php

```
public function store(Request $request)
{
    $this->validate($request, [
        'judul' => ['required', 'min:3'],
        'deskripsi' => ['required', 'min:5'],
        'photo' => ['required', 'mimes:jpg,png,jpeg', 'max:2000']
    ]);
}
```

```

$fill_data = ([  

    'judul' => $request->judul,  

    'deskripsi' => $request->deskripsi  

]);  

$berita = Berita::create($fill_data);  

$file = $request->file('photo');  

$path = Storage::disk('public')->putFile('beritas', $file);  

$berita->photo = $path;  

$berita->save();  
  

return redirect()->route('berita.index')-  

>with('statusEdit', 'Berita created!');  

}

```

Lampiran 31 Kode Halaman Mengedit Berita

File BeritaController.php

```

public function update(Request $request, $id)  

{  

    $berita = Berita::findOrFail($id);  

    $this->validate($request, [  

        'judul' => ['required', 'min:3'],  

        'deskripsi' => ['required', 'min:5'],  

        'photo' => ['mimes:jpg,png,jpeg','max:2000']  

    ]);  

    if($request->hasFile('photo'))  

    {  

        $file = $request->file('photo');  

        $oldFilename = $berita->photo;  

        $del = Storage::delete('public/'.$oldFilename);  

        $berita->$del;  

        $path=Storage::disk('public')->putfile('beritas', $file);  

    }  

    else  

    {  

        $path = $berita->photo??null;  

    }  

    $berita->judul = $request->judul;  

    $berita->deskripsi = $request->deskripsi;  

    $berita->photo = $path;  

    $berita->update();  

    return redirect()->route('berita.index')-  

>with('statusEdit', 'Berita updated');  

}

```

Lampiran 32 Kode Halaman Menampilkan All Event Pameran

File EventAdminController.php

```
public function index()
{
    $events = Event::orderBy('tgl_akhir', 'desc')-
>orderBy('created_at', 'DESC')->paginate();

    foreach($events as $event){
        $dt = Carbon::now();
        if($dt > $event['tgl_akhir']){
            $event['closed'] = true;
        }else{
            $event['closed'] = false;
        }
    }
    return view('pages.eventAdmin.view', compact('events'));
}
```

Lampiran 33 Kode Halaman Menampilkan Detail Event

File EventAdminController.php

```
public function showReadmore($id)
{
    $event = Event::findOrFail($id);
    $data = Presensi::where('id_event', $id)->get();
    $data['countDaftar'] = count($data);
    return view('pages.eventAdmin.readmore-event', compact('event', 'data'));
}
```

Lampiran 34 Kode Halaman Menampilkan Konfirmasi Event

File EventAdminController.php

```
public function eventKonfirmasi()
{
    $events = Event::orderBy('id', 'desc')->paginate();
    return view('pages.eventAdmin.view-konfirmasi', compact('events'));
}
```

Lampiran 35 Kode Halaman Menampilkan Data Pengunjung

File PengunjungController.php

```
public function index()
{
    $pengunjungs = Pengunjung::orderBy('created_at', 'DESC')->paginate();
    return view('pages.pengunjung.view', compact('pengunjungs'));
}
```

Lampiran 36 Kode Halaman Menambahkan Data Pengunjung

File PengunjungController.php

```
public function store(Request $request)
{
    $this->validate($request, [
        'name' => ['required', 'min:3'],
        'email' => ['required', 'min:5', 'unique:pengunjungs'],
        'password' => ['required'],
        'photo' => ['mimes:jpg,png,jpeg', 'max:2000']
    ]);
    if($request->hasFile('photo')!=""){
        $file = $request->file('photo');
        $path = Storage::disk('public')->putFile('pengunjungs', $file);
        $fill_data = ([[
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
            'tgl_lahir' => $request->tgl_lahir,
            'jk' => $request->jk,
            'no_hp' => $request->no_hp,
            'path' => $path
        ]]);
    }else{
        $fill_data = ([[
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
            'tgl_lahir' => $request->tgl_lahir,
            'jk' => $request->jk,
            'no_hp' => $request->no_hp,
        ]]);
    }
    $pengunjung = Pengunjung::create($fill_data);
    return redirect()->route('pengunjung.index')-
>with('statusEdit', 'Data pengunjung created!');
}
```

Lampiran 37 Kode Halaman Mengedit Data Pengunjung

File PengunjungController.php

```
public function update(Request $request, $id)
{
    $pengunjung = Pengunjung::findOrFail($id);
    $this->validate($request, [
        'name' => ['required', 'min:3'],
        'email' => ['min:5'],
    ]);
```

```

'photo' => ['mimes:jpg,png,jpeg', 'max:2000']
]);
if($request->hasFile('photo'))
{
$file = $request->file('photo');
$oldFilename = $pengunjung->path;
$del = Storage::delete('public/'.$oldFilename);
$pengunjung->$del;
$path=Storage::disk('public')->putfile('pengunjungs', $file);
$pengunjung->path = $path;
}
else
{
$path = $pengunjung->path??null;
}
$fill_data = [
'name' => $request->name,
'email' => $request->email,
'tgl_lahir' => $request->tgl_lahir,
'jk' => $request->jk,
'no_hp' => $request->no_hp,
'path' => $path
];
$pengunjung->update($fill_data);
return redirect()->route('pengunjung.index')-
>with('statusEdit', 'Data Pengunjung Updated');}

```

Lampiran 38 Kode Halaman Menampilkan Data Penyelenggara

File PenyelenggaraController.php

```

public function index()
{
$penyelenggaras = Penyelenggara::orderBy('created_at', 'DESC')->paginate();
return view('pages.penyelenggara.view', compact('penyelenggaras'));
}

```

Lampiran 39 Kode Halaman Menambahkan Data Penyelenggara

File PenyelenggaraController.php

```

public function store(Request $request)
{
$this->validate($request, [
'name' => ['required', 'min:3'],
'email' => ['required', 'min:5', 'unique:penyelenggaras'],
'password' => ['required'],
'organisasi' => ['required'],
'alamat' => ['required', 'min:5'],
]

```

```

        'no_hp' => ['required', 'max:15'],
        'photo' => ['required', 'mimes:jpg,png,jpeg', 'max:2000'],
        'created_by',
        'updated_by'
    ]);
$file = $request->file('photo');
$path = Storage::disk('public')->putFile('penyelenggaras', $file);
$fill_data = [
    'name' => $request->name,
    'email' => $request->email,
    'password' => Hash::make($request->password),
    'path' => $path,
    'organisasi' => $request->organisasi,
    'alamat' => $request->alamat,
    'no_hp' => $request->no_hp,
    'created_by' => $request = Auth::user()->id,
    'updated_by' => $request = Auth::user()->id,
];
$penyelenggara = Penyelenggara::create($fill_data);
return redirect()->route('penyelenggara.index')-
>with('statusEdit', 'Data Penyelenggara Created!');
}

```

Lampiran 40 Kode Halaman Mengedit Data Penyelenggara

File PenyelenggaraController.php

```

public function update(Request $request, $id)
{
    $penyelenggara = Penyelenggara::findOrFail($id);
    $this->validate($request, [
        'name' => ['required', 'min:3'],
        'email' => ['required', 'min:5', 'unique:penyelenggaras'],
        'organisasi' => ['required'],
        'alamat' => ['required', 'min:5'],
        'no_hp' => ['required', 'max:15'],
        'photo' => ['mimes:jpg,png,jpeg', 'max:2000'],
        'updated_by'
    ]);
    if($request->hasFile('photo'))
    {
        $file = $request->file('photo');
        $oldFilename = $penyelenggara->path;
        $del = Storage::delete('public/'.$oldFilename);
        $penyelenggara->$del;
        $path=Storage::disk('public')-
>putfile('penyelenggaras', $file);
        $penyelenggara->path = $path;
    }
}

```

```

        }
    else
    {
        $path = $penyelenggara->path??null;
    }
$fill_data = [
    'name' => $request->name,
    'email' => $request->email,
    'organisasi' => $request->organisasi,
    'alamat' => $request->alamat,
    'no_hp' => $request->no_hp,
    'path' => $path,
    'updated_by' => $request = Auth::user()->id
];
$penyelenggara->update($fill_data);
return redirect()->route('penyelenggara.index')-
>with('statusEdit', 'Data Penyelenggara Updated');
}

```

Lampiran 41 Kode Halaman Profil

File ProfilController.php

```

public function index()
{
    return view('pages.profil.profil');
}

```

Lampiran 42 Kode Halaman Menambahkan *Event*

File EventController.php

```

public function store(Request $request)
{
    $this->validate($request, [
        'photo' => ['required', 'mimes:jpg,png,jpeg', 'max:2000']
    ]);

    $event= new Event;
    $event->id_penyelenggara = Auth::user('penyelenggara')->id;
    $event->judul = $request->judul;
    $event->tgl_mulai = $request->tgl_mulai;
    $event->tgl_akhir = $request->tgl_akhir;
    $event->jam_mulai = $request->jam_mulai;
    $event->jam_akhir = $request->jam_akhir;
    $event->lokasi = $request->lokasi;
    $note = $request->desc;
    $dom = new \domdocument();

```

```

$dom->loadHtml($note, LIBXML_HTML_NOIMPLIED | LIBXML_HTML_NODEFDTD);

$note = $dom->savehtml();
$file = $request->file('photo');
$path = Storage::disk('public')->putFile('events', $file);
$event->photo = $path;
$event->desc = $note;
$event->save();

$QRCode = "storage/qrcodes/".$event->id.".png";
QRCode::text('{"id":"' . $event->id. '", "judul":"' . $event-
>judul. '", "penyelenggara":"' . $event->penyelenggara-
>organisasi. '", "deskripsi":"' . $event->desc. '", "lokasi":"' . $event-
>lokasi. '"}')
->setSize(64)
->setMargin(2)
->setOutFile($QRCode)
->png();

$event = Event::where(['id'=>$event->id])
->update(['qrcode'=>$QRCode]);

return redirect('/myEvent')->with('statusEdit', 'Data event created!');
}

```

Lampiran 43 Kode Halaman Mengedit *Event*

File EventController.php

```

public function edit($id)
{
    $event = Event::findOrFail($id);
    return view('pages.event.edit', compact('event'));
}

public function update(Request $request, $id)
{
    $event = Event::findOrFail($id);

    $this->validate($request, [
        'photo' => ['mimes:jpg,png,jpeg', 'max:2000']
    ]);
    if($request->hasFile('photo'))
    {
        $file = $request->file('photo');
        $oldFilename = $event->photo;
        $del = Storage::delete('public/'.$oldFilename);
        $event->$del;
    }
}

```

```

        $path=Storage::disk('public')-
>putfile('events', $file);
    }
else
{
    $path = $event->photo??null;
}

$event->id_penyelenggara = Auth::user('penyelenggara')-
>id;
$event->judul = $request->judul;
$event->tgl_mulai = $request->tgl_mulai;
$event->tgl_akhir = $request->tgl_akhir;
$event->jam_mulai = $request->jam_mulai;
$event->jam_akhir = $request->jam_akhir;
$event->lokasi = $request->lokasi;
$event->photo = $path;
$event->desc= $request->desc;
$event->update();

$QRCODE = "storage/qrcodes/".$event->id.".png";
QRCode::text('{"id":"' . $event->id. '", "judul":"' . $event-
>judul. '", "penyelenggara":"' . $event->penyelenggara-
>organisasi. '", "deskripsi":"' . $event->desc. '", "lokasi":"' . $event-
>lokasi. '"}')
->setSize(64)
->setMargin(2)
->setOutFile($QRCODE)
->png();

$event = Event::where(['id'=>$event->id])
->update(['qrcode'=>$QRCODE]);

return redirect('/myEvent')->with('statusEdit', 'Data Event Updated');
}

```

Lampiran 44 Kode Halaman Menampilkan *My Event*

File EventController.php

```

public function myEvent(){
    $events = Event::where('id_penyelenggara',Auth::user('penyelenggara')-
->id)->orderBy('tgl_akhir','desc')->orderBy('created_at','desc')->get();
    foreach($events as $event){
        $dt = Carbon::now();
        if($dt > $event['tgl_akhir']){
            $event['closed'] = true;
        }else{

```

```

        $event['closed'] = false;
    }
}

return view('pages.event.myevent-view', compact('events'));
}

```

Lampiran 45 Kode Halaman Menampilkan Qrcode

File PresensiController.php

```

public function showQrcode($id){
    $show = Event::findOrFail($id);

    return view('pages.event.show-qrcode', compact('show'));
}

public function viewQrcode($id){
    $show = Event::findOrFail($id);
    return view('pages.event.view-qrcode', compact('show'));
}

```

Lampiran 46 Kode Halaman Menampilkan Buku Tamu

File BukuTamuController.php

```

public function show($id)
{
    $event = Event::findOrFail($id);

    $bukutamus = BukuTamu::where('id_event', $id)->orderBy('created_at', 'DESC')->paginate();

    return view('pages.bukutamu.view', compact('bukutamus', 'event'));
}

```

Lampiran 47 Kode Halaman Menambahkan Kehadiran Pengunjung

File BukuTamuController.php

```

public function store(Request $request)
{
    $this->validate($request, [
        'name' => ['required'],
        'email' => ['required'],
        'no_hp',
        'id'
    ]);
}

```

```
$bukutamu = New BukuTamu();
$bukutamu->name = $request->name;
$bukutamu->email = $request->email;
$bukutamu->no_hp = $request->no_hp;
$bukutamu->id_event = $request->id;
$bukutamu->save();

return redirect()->back()->with('statusEdit', 'Data Motif Created!');
}
```

Lampiran 48 Kode Halaman Melihat Konfirmasi *Event*

File EventController.php

```
public function eventStatus(){
    $events = Event::where('id_penyelenggara',Auth::user('penyelenggara')
->id)->orderBy('tgl_akhir','desc')->orderBy('created_at','desc')->get();

    return view('pages.event.myevent-status', compact('events'));
}
```