

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

Dhiya Fauziza dan Zani Kolina

Jurusan Teknik Elektro, Politeknik Negeri Semarang
E-mail: dhiyafauziza99@gmail.com, zanikolina@gmail.com

Abstrak

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. Pengunjung masih melakukan interaksi secara langsung kepada pengrajin untuk memperoleh informasi batik tersebut. Penelitian 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 mendeteksi fitur yang ada pada gambar dan membandingkan fitur gambar tersebut dengan yang ada di dalam *dataset*, gambar akan terus dideteksi selama dalam bidang pandang kamera. Metode yang digunakan dalam pembuatan sistem ini adalah metode scrum yang meliputi *product backlog*, *sprint planning meeting*, *sprint backlog*, dan pengerjaan *sprint*. 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%. Hasil pengujian dilakukan menggunakan kain batik asli dengan sepuluh kali percobaan pada setiap kain batik dan didapatkan kebenaran dari hasil deteksi batik sebesar 100%.

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

Abstract

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 study 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 while it is in the camera's field of view. The method used in making this system is the Scrum method which includes the product backlog, sprint planning meeting, sprint backlog, and sprint work. This application uses a dataset consisting of 12 batik motifs and each motif has 600 images, with a total motif 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 test results were carried out using the original batik cloth with ten experiments on each batik cloth and the truth was obtained from the batik detection results of 100%.

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

I. PENDAHULUAN

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 pameran, 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.

Beberapa penelitian yang telah dilakukan sebelumnya khususnya tentang metode pengenalan motif batik. Kusanti dan Agus [1] menerapkan pengidentifikasian motif batik tradisional Surakarta dengan metode *Grey Level Co-Occurrence Matrix*. Hasil tes memperoleh tingkat akurasi 95%, dengan tingkat kesalahan 0,05%. Gultom dkk. [2] menggunakan metode *ConvNet* yang mempelajari representasi data pada klasifikasi pola batik. Nuraedah dan Bakri [3] menggunakan metode *Gray Level Co-occurrence matrices (GLCM)* dalam sistem klasifikasi motif kain tradisional Batik Bomba Kaili berdasarkan fitur tekstur. Hasil akurasi klasifikasi pada tahap pelatihan menjadi 80.65% dan pada tahap pengujian menghasilkan akurasi sebesar 77.14%. Sari Y. [4] menggunakan *image retrieval* yang diterapkan pada proses pengenalan motif batik. Hasil akurasi dalam klasifikasi motif batik sebesar 44.44%. Kahamdany [5] menerapkan metode L-System untuk menyediakan fitur pengembangan motif karang jenis *leptosiris papyracea* pada aplikasi batik. Dewi [6] menggunakan *Convolutional Neural Network (CNN)* untuk pendeteksian klasifikasi meja dan kursi. Tingkat akurasi model berkisar antara 70% hingga 99%. Danukusumo [7] juga menggunakan *Convolutional Neural Network (CNN)* untuk pengenalan motif ukiran Jepara menggunakan *framework tensorflow*. Hasil pengujian menunjukkan akurasi sebesar 98,99% pada *training* set dan 85,57%. Nurhikmat [8] menggunakan metode *Convolutional Neural Network (CNN)* dalam mengklasifikasikan wayang golek dengan tingkat akurasi dalam melakukan klasifikasi gambar Wayang Golek sebesar 93%. Eka Putra [9] menggunakan metode CNN dalam mengklasifikasikan unggas diantaranya dengan kategori emu, flamingo, ibis, *pigeon*, dan *roaster*. Hasil dari 5 kategori unggas menunjukkan bahwa persentase keberhasilan 20% sedangkan untuk 3 kategori lainnya menunjukkan persentase keberhasilan

50%. Abu dkk. [10] menerapkan metode *Deep Neural Network (DNN)* dengan input kategori bunga. Hasil akurasi klasifikasi gambar, Mawar mendapatkan 90,585% dan sama dengan bunga jenis lain di mana rata-rata hasilnya mencapai 90% ke atas. Jalled dan Voronkov [11] menerapkan algoritma Haar Cascade untuk deteksi sebuah *Unmanned Ariel Vehicle (UAV)* untuk menghindari hal yang tidak diinginkan tersebut tabrakan dan kerusakan UAV. A. Elrefaei dkk.[12] menggunakan algoritma Douglas-Peucker untuk mendeteksi bentuk geometris dua dimensi termasuk lingkaran, segitiga, kotak, dan cocok dengan gambar pemandangan. Dalam kondisi yang berbeda, juga sudah dilakukan peneliti dengan menggunakan OpenCV untuk segmentasi warna dengan tahapan *captures frames* didalam RGBA, lalu dikonversi 8 bit piksel matrik dan mengonversikan warna RGBA ke HSV. Taskin dkk.[13] menggunakan OpenCV pada penelitian untuk membedakan warna dengan tahapan *captures frames* didalam RGBA, lalu dikonversi 8 bit piksel matrik dan mengonversikan warna RGBA ke HSV. Maulana[14] menggunakan metode *color blob detection* dalam pengolahan citra pada deteksi daging memiliki akurasi daging sapi 84.28%, jenis daging babi: 92.86% dan jenis daging kambing: 94.28%. Masdiyasa dkk. [15] menggunakan algoritma *threshold, contour* dan *k-Nearest Neighbors* pada aplikasi digitalisasi plat nomor dapat membaca dan mengenali segmen tertentu pada *image* sebagai plat nomor. Tingkat akurasi pengenalan plat mencapai 95% dengan menggunakan perangkat *smartphone*. Goma [16] menyatakan pengujian *tensorflow* yang merupakan *framework deep learning* dari Google yang menjadi sebuah alternatif dari *deep neural network*. Shaha dan Pawar [17] menyatakan bahwa *Convolutional neural network (CNN)* digunakan untuk berbagai aplikasi seperti pengenalan objek, gambar beresolusi besar, segmentasi semantik, dll. Arsitektur CNN diantaranya AlexNet, VGG16. Pangestu dan Bunyamin [18] menggunakan metode CNN untuk deteksi ras anjing dengan *pre-trained* model dari Keras yang diujikan ResNet50, Xception dan VGG16, Xception memiliki performa keakuratan yang lebih baik daripada dua model lainnya. Shu [19] menggunakan InceptionResNet V2 yang 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*.

Abas dkk. [20] menerapkan metode CNN dengan arsitektur model VGG16 untuk mendeteksi bunga. Pengerjaan 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. Alsing[21] menyatakan peningkatan model *deep learning* dengan heuristik jauh lebih tinggi jika menggunakan DL karena menghasilkan peningkatan sebesar 117.7% sedangkan heuristik sebesar 26.1%.

Berbeda dengan [1] dan [3] dalam proses pendeteksian gambar yang menggunakan metode metode *Grey Level Co-Occurrence Matrix* (GLCM), penelitian ini menggunakan metode CNN untuk mendapatkan hasil akurasi yang lebih baik. Berbeda dengan [17] dan [18] dalam menggunakan algoritma InceptionResNet V2, ResNet50, Xception dan VGG16, penelitian ini menggunakan arsitektur MobileNet yang mempunyai akurasi yang lebih tinggi dengan ukuran file yang relatif lebih kecil untuk dipasang pada android.

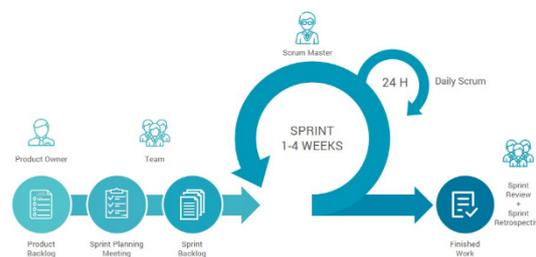
Dari beberapa penelitian diatas, belum terdapat fitur deteksi untuk mengenali motif batik pada suatu kain yang mengandalkan kamera *smartphone* untuk mendeteksi motif batik dengan menampilkan informasi batik meliputi nama batik, asal daerah batik, ciri khas motif batik, dan filosofi batik. Selain itu, belum terdapat fitur manajemen *event* pameran batik yang menjadi salah satu keunggulan dalam mengelola *event* agar memperoleh data minat pengunjung pameran batik.

Tujuan dari penelitian ini adalah membangun aplikasi yang dapat mendeteksi motif batik menggunakan android dan *image processing*. Selain itu penelitian ini bertujuan untuk menampung respon pengunjung terhadap *event* pameran batik, sehingga dapat mempertahankan citra batik di kalangan masyarakat Indonesia dan membantu para pengrajin batik dalam memasarkan produk batik yang dijual.

II. METODE PENELITIAN

Metode pembuatan aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* menggunakan metode *Agile* dengan kerangka kerja *Scrum*. Metode ini merupakan model

pengembangan sistem yang cepat dan lebih mementingkan interaksi dengan user daripada proses dan aplikasi. Kerangka kerja yang digunakan adalah *Scrum* berupa kerangka kerja yang dapat mengatasi masalah adaptif 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. Ken Schwaber dan Jeff Sutherland telah menulis *The Scrum Guide* untuk menjelaskan Scrum dengan jelas dan ringkas [22]. Dalam kerangka kerja scrum memiliki beberapa tahapan seperti *product backlog*, *sprint planning meeting*, *sprint backlog*, pengerjaan *task*.



Gambar 1 Tahapan Metode Agile menggunakan Kerangka Kerja Scrum

2.1 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.

2.2 Sprint Planning Meeting

Pada tahap ini, tim pengembang mendiskusikan pembuatan *sprint* pada projek yang akan dikerjakan. Pembuatan *sprint* didasarkan pada tingkat kesulitan *backlog* yang ada.

2.3 Sprint Backlog

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

2.4 Pengerjaan Task

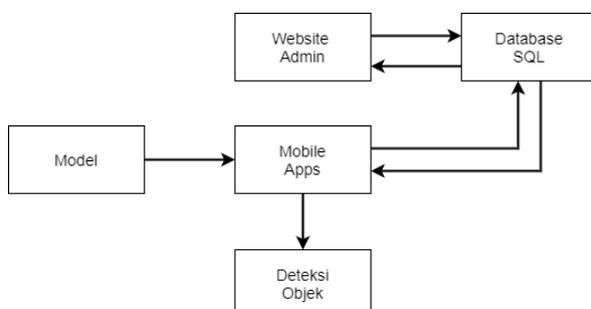
Tahap ini melakukan pengerjaan *task – task* yang ada pada *sprint backlog* dan waktu pengerjaan diharuskan mengikuti waktu yang sudah ditetapkan pada *sprint* tersebut. Tahap ini merupakan tahapan pembuatan dan pengujian *task* yang dilakukan secara berurutan sesuai *sprint* yang meliputi analisis kebutuhan perangkat, analisis perancangan diagram sistem, analisis perancangan sistem, perancangan model dataset, pembuatan dan pengujian model dataset, perancangan antarmuka aplikasi, pembuatan dan pengujian fitur aplikasi serta perancangan uji kepuasan pengguna.

2.4.1 Analisis Kebutuhan Perangkat

Bagian menganalisis kebutuhan perangkat berupa kebutuhan perangkat keras (*hardware*) dan kebutuhan perangkat lunak (*software*) yang diperlukan dalam pembuatan aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing*.

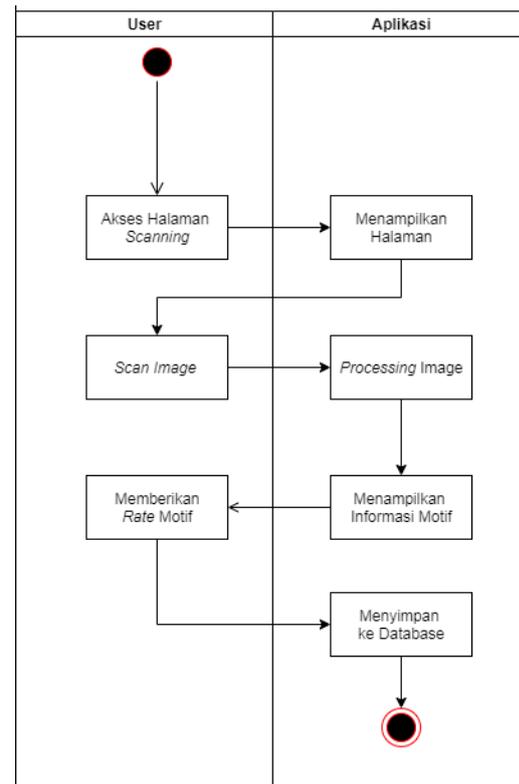
2.4.2 Analisis Perancangan Sistem

Bagian ini membahas perancangan sistem seperti analisis gambaran umum sistem, *block diagram*, *usecase diagram*, *activity diagram*, *Entity Relationship Diagram (ERD)*. Model dataset yang sudah dihasilkan disimpan dan diproses ke dalam sebuah aplikasi *mobile* yang nantinya 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. Gambar 2 menunjukkan blok diagram sistem.



Gambar 2 Blok Diagram Sistem

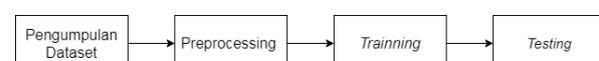
Gambar 3 menunjukkan *activity diagram* proses deteksi batik. Pengguna dapat mengklasifikasi motif batik dengan mengakses halaman *scanning*. Pengguna melakukan *scan* terhadap batik dengan mengarahkan kamera ke gambar batik, selanjutnya aplikasi akan memproses *image* dan menampilkan informasi mengenai batik yang sesuai dengan hasil deteksi. Selanjutnya pengguna akan diarahkan untuk memberikan *rating* terhadap batik yang dideteksi dan disimpan pada database.



Gambar 3 Activity Diagram Deteksi Batik

2.4.3 Perancangan Dataset

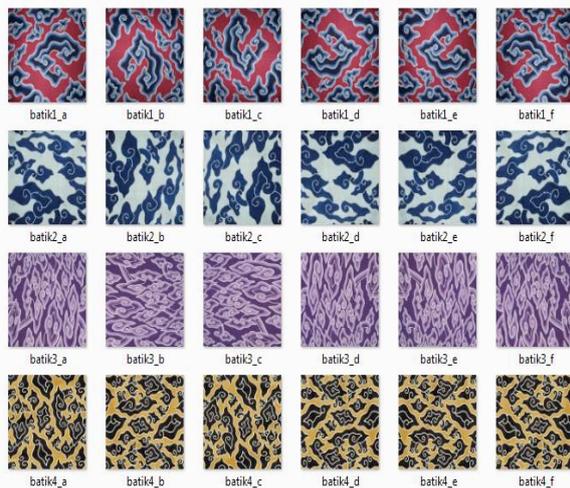
Pada bagian ini dibahas mengenai perancangan model data yang diawali dengan pengambilan citra motif batik, kemudian dilakukan tahap pengolahan citra digital yang meliputi pra-pengolahan, dan *training* untuk menghasilkan nilai ciri yang terkandung dari citra sebagai vektor masukan. Tahap selanjutnya yaitu pengujian atau *testing* dengan data yang berbeda. Berikut ini merupakan gambaran perancangan model dataset yang akan dilakukan.



Gambar 4 Proses Perancangan Model Dataset

Proses pembuatan model dataset batik didapatkan dari *search engine* google yang 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.

Pada tahap *pre-processing* ini dilakukan *rotate*, *flipping*, *resize*, dan *labelling* pada gambar. Tahap awal citra batik asli yang didapatkan kemudian diubah menjadi ukuran 500x500 pixel dan dilakukan rotasi dengan sudut 90°, 180°, 270° serta *flipping* horizontal dan *flipping* vertikal. Contoh citra batik yang dirotasi dan *flipping* ditunjukkan pada Gambar 5.



Gambar 5 Data Batik yang Dirotasi dan *Flipping*

Pada tahapan *training*, *dataset* citra batik yang digunakan sebanyak 7.200 sampel, dibagi menjadi 12 *dataset* kelas batik. Pada setiap kelas *dataset* terdapat 600 sampel citra batik. Data *training* yang digunakan sebanyak 6480 gambar (90%) dan data *testing* 720 gambar (10%).

Pada tahap ini model yang dibangun menggunakan arsitektur CNN MobileNet yang memiliki layer-layer konvolusi dengan input awal 224x224 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 diset 60 untuk mengurangi dimensi citra. Tahapan *testing* dilakukan terhadap 720 gambar untuk mendapatkan hasil akurasi yang lebih baik.

2.4.4 Pembuatan dan Pengujian Model Dataset

Pengujian dan pembuatan dilakukan menggunakan google collab untuk melakukan proses *training* dan proses *testing*. Selanjutnya model akan diubah ke dalam bentuk .tflite untuk dipasang ke dalam aplikasi android. Pada tahapan pengujian model ini dilakukan pengujian terhadap empat arsitektur yang berbeda diantaranya EfficientNet Lite 0, EfficientNet Lite 1, EfficientNet Lite 2, dan MobileNet. Dari ke empat model yang berbeda akan dipilih model dengan hasil akurasi *training* dan *testing* yang lebih baik. Selain itu kapasitas jumlah file yang dihasilkan akan dipilih model dengan ukuran file yang lebih kecil untuk dipasang pada android. Berikut merupakan kode program pada proses *testing* model yang ditunjukkan pada Gambar 6.

```
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 6 Kode Program *Testing* Model

2.4.5 Perancangan Antarmuka Aplikasi

Perancangan antarmuka pada android berupa fitur deteksi batik menggunakan kamera *smartphone* pengguna, fitur menampilkan deskripsi informasi dari batik yang dideteksi, dan halaman untuk mengisi buku tamu pada saat *event* pameran batik. Untuk perancangan antarmuka pada website berupa fitur manajemen *event* pameran seperti menampilkan *qr code* yang digunakan untuk validasi kehadiran pengunjung pameran, serta untuk menampilkan kalkulasi dari jumlah pengunjung *event*, batik terfavorit, dan *event* pameran terfavorit berdasarkan pilihan pengguna android atau pengunjung pameran.

2.4.6 Pembuatan dan Pengujian Fitur Aplikasi

Bagian ini membahas tentang pembuatan fitur – fitur yang ada pada halaman antarmuka aplikasi pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* dari tampilan, kode program, dan pengujian fitur.

2.4.7 Perancangan Uji Kepuasan Pengguna

Pada tahapan ini dilakukan proses perancangan pengujian aplikasi yang sudah jadi kepada pengunjung pameran. Dalam hal ini penulis menggunakan sampel 16 responden untuk melakukan pengujian sistem. Dalam pengujian sistem ini didapatkan data dengan cara memberikan beberapa poin pertanyaan seputar

sistem ini serta mengharapkan kritik dan saran demi perkembangan selanjutnya.

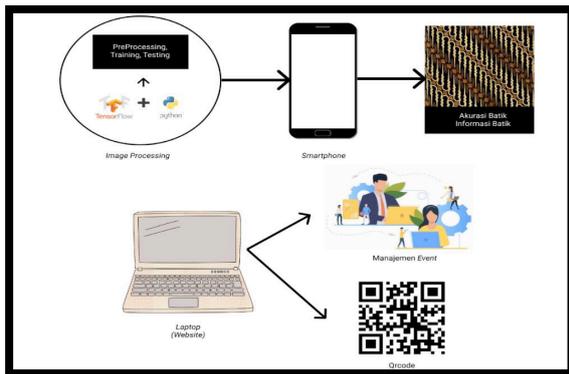
III. HASIL DAN PEMBAHASAN

Aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* dibangun menggunakan metode pengklasifikasian yaitu *Convolutional Neural Network* (CNN) yang digunakan dalam pengolahan data *image*. Arsitektur yang digunakan dalam pengolahan dataset batik yaitu menggunakan MobileNet dengan model dataset yang direlasikan menggunakan database MySQL yang bertujuan untuk menampilkan informasi ketika proses *scan* batik telah selesai.

Berikut hasil dan pembahasan penelitian ini yang meliputi hasil perancangan, model dataset, aplikasi android dan sistem manajemen *event* pada web dan kepuasan pengguna.

3.1 Hasil Perancangan

Hasil perancangan berupa gambaran umum sistem, blok diagram, *activity diagram*, dan ERD.



Gambar 7 Gambaran Umum Sistem

Gambar 7 merupakan gambaran umum sistem dimana 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 *qr code* untuk buku tamu pengunjung.

3.2 Analisis Model

Pada penelitian ini, peneliti membandingkan empat model yang berbeda untuk diterapkan pada aplikasi android diantaranya MobileNet, EfficientNet Lite 0, EfficientNet Lite 1, EfficientNet Lite 2, dan Efficientnet Lite 3.

MobileNet merupakan salah satu arsitektur *Convolutional Neural Network* (CNN) yang

digunakan untuk *computing* pada *resource* yang besar. 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. 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. 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.

3.3 Hasil Pengujian Model Dataset

Pada tahap uji coba model digunakan empat arsitektur model yang berbeda diantaranya MobileNet, EfficientNet Lite 0, EfficientNet Lite 1, dan EfficientNet Lite 2. Pengujian dilakukan dengan menghasilkan akurasi *training*, akurasi *testing* dan ukuran model yang akan divisualisasikan kedalam tabel.

3.3.1 Perbandingan Akurasi Training Model

Sebelum menentukan model yang tepat untuk diaplikasikan pada aplikasi, berikut merupakan perbandingan akurasi training model yang divisualisasikan dalam Tabel 1.

TABEL 1 PERBANDINGAN AKURASI *TRAINING* MODEL

| Model | Epoch | Akurasi |
|---------------------|-------|---------|
| MobileNet-V2 | 10 | 99.16 % |
| EfficientNet Lite 0 | 10 | 97.70 % |
| EfficientNet Lite 1 | 10 | 98.36 % |
| EfficientNet Lite 2 | 10 | 98.79 % |

Berdasarkan Tabel 1, MobileNet memiliki tingkatan akurasi *training* yang lebih tinggi dibandingkan dengan 3 model lainnya yaitu sebesar 99.16 %.

3.3.2 Perbandingan Akurasi Testing Model

Sebelum menentukan model yang tepat untuk diaplikasikan pada aplikasi, berikut merupakan perbandingan akurasi testing model yang divisualisasikan dalam Tabel 2.

Tabel 2 memperlihatkan hasil testing model MobileNet 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. Selain itu pada arsitektur MobileNet tidak terjadi *overfitting* dan *underfitting* sehingga saat proses deteksi motif batik akan menghasilkan tingkat keberhasilan deteksi batik yang baik dan sesuai.

TABEL 2 PERBANDINGAN 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 |

3.3.3 Perbandingan Ukuran Model

Model yang dipasang di android digunakan ukuran yang paling kecil untuk meminimalisir ukuran aplikasi android, sehingga aplikasi akan berjalan lancar di *smartphone* pengguna. Ukuran perbandingan model dataset akan ditampilkan pada Tabel 3.

TABEL 3 PERBANDINGAN UKURAN MODEL

| Model | Ukuran (M) |
|---------------------|------------|
| MobileNet-V2 | 8.57 M |
| EfficientNet Lite 0 | 12.91 M |
| EfficientNet Lite 1 | 15.83 M |
| EfficientNet Lite 2 | 18.4 M |

Berdasarkan Tabel 3 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.

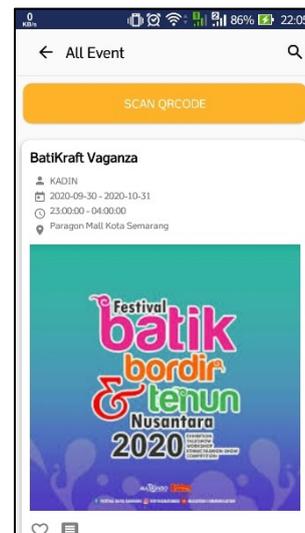
3.4 Aplikasi Deteksi Batik dan Manajemen Event

Aplikasi interaktif pengenalan batik untuk pengunjung pameran berbasis android dan *image processing* dibangun sebagai antar muka sistem ke pengguna dan diberi nama Aplikasi Batik.In. Batik.In mempunyai tampilan – tampilan guna memudahkan pengguna dalam melakukan deteksi batik serta memperoleh informasi mengenai batik yang dideteksi. Pengguna aplikasi dapat memberikan *like* dan komentar pada suatu *event* pameran dan juga dapat memberikan *rating* terhadap batik yang dideteksi. Aplikasi ini juga digunakan pengguna pameran dalam mengisi buku tamu ketika menghadiri suatu pameran batik dengan cara *scan qr code* yang sudah disediakan oleh sistem *web*. Data pengunjung pameran akan

direkap pada sistem web guna memperoleh data pengunjung yang menghadiri *event* pameran.

3.4.1 Halaman Event Pameran

Gambar 8 menyajikan tampilan *event* pameran yang sedang dilaksanakan dan yang akan dilaksanakan. Pada setiap blok *event* pameran ditampilkan nama dari pameran batik, lokasi, tanggal, waktu, dan penyelenggara pameran tersebut. Di dalam halaman ini, pengguna atau pengunjung pameran dapat memberikan *like* atau komentar yang ditunjukkan untuk pameran yang dipilih.



Gambar 8 Halaman Event Pameran

3.4.2 Halaman Konfirmasi Event

Gambar 9 menyajikan halaman konfirmasi *event* dari aplikasi Batik.In. Halaman ini digunakan untuk mengkonfirmasi kehadiran pengunjung pameran terhadap *event* yang didatangi dari hasil *scan qr code* yang disediakan penyelenggara. Tombol daftar digunakan untuk konfirmasi kehadiran.



Gambar 9 Halaman Konfirmasi Event

3.4.3 Halaman Deteksi Batik

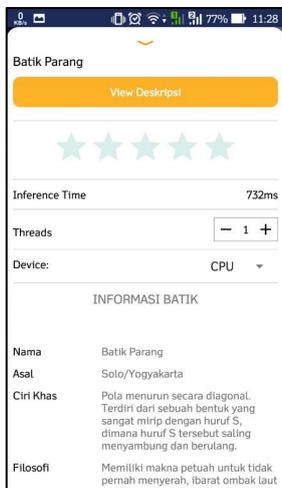
Gambar 10 menampilkan halaman deteksi batik. Halaman deteksi dapat diakses dengan mengklik *button* kamera, maka secara otomatis akan membuka kamera *smartphone* dan mengarahkan kamera ke batik yang ingin dideteksi dengan jarak 30 - 50 cm. Maka aplikasi akan menampilkan nama batik yang dideteksi. Apabila pengguna ingin melihat informasi mengenai batik yang dideteksi dapat melihatnya pada *button view* deskripsi.



Gambar 10 Halaman Deteksi Batik

3.4.4 Halaman Informasi Batik

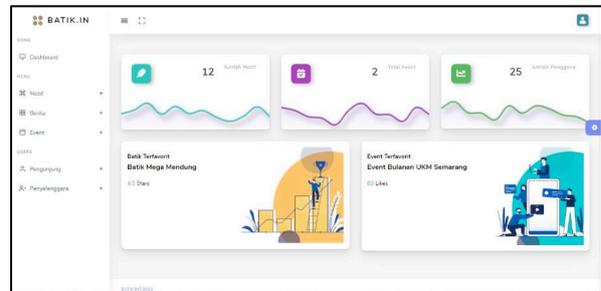
Gambar 11 menyajikan halaman informasi batik dari aplikasi Batik.In. Dalam halaman ini, informasi batik ditampilkan seperti nama batik, asal, ciri khas motif batik tersebut, dan filosofi batik tersebut. Pada halaman ini pengguna dapat memberikan *rating* terhadap batik tersebut yang nantinya dapat mengetahui batik terfavorit dalam *event* tersebut.



Gambar 11 Halaman Informasi Batik

3.4.5 Halaman Dashboard Web

Gambar 12 menampilkan halaman *dashboard* yang berisi kalkulasi dari jumlah motif batik, total *event* pameran, total pengunjung dalam pameran tersebut, *event* favorit, dan batik terfavorit.



Gambar 12 Halaman Dashboard

3.4.6 Halaman Menampilkan Qrcode Web

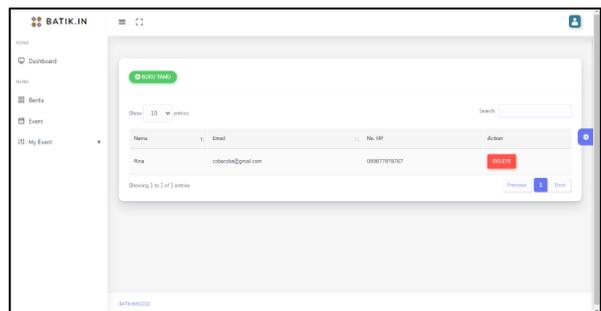
Gambar 13 menampilkan halaman *qrcode* yang digunakan untuk konfirmasi kehadiran pada suatu pameran batik. Halaman ini dapat diakses oleh penyelenggara pameran pada menu *MyEvent*. Penyelenggara dapat menampilkan *qrcode* langsung atau dapat mencetak *qrcode* untuk ditempelkan pada pintu masuk pameran.

3.4.7 Halaman Menampilkan Buku Tamu

Gambar 14 menyajikan halaman buku tamu. Dalam halaman ini, penyelenggara dapat melihat siapa saja yang hadir dalam *event* pameran tersebut. Halaman ini dapat diakses pada menu *MyEvent* kemudian memilih *button* buku tamu.



Gambar 13 Halaman Menampilkan qrcode



Gambar 14 Halaman Menampilkan Buku Tamu

3.5 Pengujian Fungsionalitas

Aplikasi ini juga melalui beberapa pengujian guna mengetahui fungsionalitas dari tiap – tiap fitur yang sudah dibuat.

3.5.1 Pengujian Fungsionalitas Model

Pada pengujian ini, empat model diuji dengan menggunakan google collab pada setiap tahapan membagi data, proses *training* model, *testing* model, *export* model dan *download* model guna memastikan fungsi dari setiap kode python berhasil dan didapatkan akurasi *training* dan *testing* yang lebih baik. Hasil yang didapat dari pengujian ini adalah model yang digunakan adalah MobileNet dengan akurasi *training* dan *testing* yang lebih tinggi, serta ukuran model yang relatif lebih kecil.

3.5.2 Pengujian Deteksi Batik

Pada bagian ini, batik dideteksi dengan aplikasi menggunakan empat sampel batik asli yaitu Batik Semen, Batik Mega Mendung, Batik Sidomukti, dan Batik Parang. 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 yang didapat pada pengujian ini akan ditampilkan pada Tabel 4.

3.5.3 Pengujian Fungsionalitas Aplikasi

Pada pengujian ini aplikasi android dan web diuji pada setiap fitur guna memastikan fungsi dari fitur tersebut berjalan dengan baik dan sesuai dengan rencana awal. Hasil yang didapat dari pengujian ini adalah fitur yang terdapat dalam aplikasi android dan web berjalan dengan baik dan sesuai rencana.

3.5.4 Pengujian Kompatibilitas Aplikasi

Pada pengujian ini aplikasi android dilakukan menggunakan tiga buah *smartphone* android dengan spesifikasi yang berbeda – beda. Tampilan mempunyai desain yang dinamis dan mengikuti ukuran layar namun tidak merubah tampilannya.

3.6 Kepuasan Pengguna

Berikut panduan perhitungan skor kuesioner dari 16 responden, seperti pada Tabel 5.

TABEL 4 PENGUJIAN DETEKSI BATIK

| Nama Batik | Data <i>Testing</i> | Hasil Deteksi | | Infor masi Batik |
|---|---------------------|---------------|-------|------------------|
| | | Benar | Salah | |
| Batik Semen  | 10 | 10 | 0 | Benar |
| Batik Sidomukti  | 10 | 10 | 0 | Benar |
| Batik Mega Mendung  | 10 | 10 | 0 | Benar |
| Batik Parang  | 10 | 10 | 0 | Benar |

TABEL 5 PANDUAN PERHITUNGAN SKOR KUESIONER

| Predikat | Skor |
|-------------------|------|
| Sangat Puas | 5 |
| Puas | 4 |
| Cukup | 3 |
| Tidak Puas | 2 |
| Sangat Tidak Puas | 1 |

Untuk hasil perhitungan total kepuasan pengguna sesuai dengan tabel inidikator kepuasan pengguna pada Tabel 5.

TABEL 6 INDIKATOR PENILIAIAN PERSENTASI KEPUASAN PENGGUNA

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

Berikut perhitungan skor kepuasan pengguna:
 Jumlah maksimal indikator penilaian = $5 \times 5 = 25$
 Jumlah responden yang mengisi kuesioner = 16

Jumlah kepuasan maksimal = $16 \times 25 = 347$

Skoring Kepuasan

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

Dari hasil perhitungan *score* kepuasan pengguna adalah 86,75%. Dari hasil perhitungan tingkat kepuasan pengguna tersebut berdasarkan Tabel 5 didapatkan kategori penilaian “sangat memuaskan”.

Metode CNN MobileNet pada proses pendeteksian batik memberikan kinerja yang lebih baik dengan menghasilkan akurasi *training* sebesar 99.16% dan *testing* sebesar 99.86%. Berbeda pada penelitian yang dilakukan oleh Nuraedah dan Bakri [3] hasil deteksi Batik Bomba Kaili mendapatkan akurasi pelatihan sebesar 80,65% dan pengujian sebesar 77.16%. Pada penelitian Dewi [6] klasifikasi meja dan kursi menghasilkan akurasi model berkisar antara 70% hingga 99%. Penelitian Danukusumo [7] pengenalan motif ukiran Jepara menggunakan metode CNN menghasilkan akurasi sebesar 98,99% pada pengujian dan *training* set 85,57%. Penelitian Nurhikmat [8] yang menggunakan metode *Convolutional Neural Network* (CNN) dalam mengklasifikasikan wayang golek diperoleh hasil akurasi dalam melakukan klasifikasi gambar Wayang Golek sebesar 93%.

Dari beberapa sistem yang dibuat oleh peneliti sebelumnya 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 manajemen *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.

IV. KESIMPULAN

Dari penelitian ini, dapat disimpulkan bahwa penelitian ini berhasil membuat aplikasi untuk mendeteksi motif batik menggunakan android dan *image processing*, serta dapat membuat sistem yang dapat menampung respon pengunjung terhadap *event* pameran batik. Aplikasi dapat mendeteksi batik pada 4 kain batik dengan dilakukan 10 kali pengambilan gambar batik tiap motif dari berbagai sudut dan menghasilkan deteksi kebenaran pada batik tersebut 100%. Proses *training* model dengan menggunakan pendekatan *transfer learning*, implementasi model menggunakan arsitektur CNN MobileNet dengan menghasilkan akurasi yang dihasilkan yaitu 99.16% untuk *training model* dan 99.86% untuk *testing model*. Kepuasan pengguna yang didapatkan dari pembuatan aplikasi ini sebesar 86,75% dengan kategori memuaskan.

Dalam pengembangan selanjutnya perlu ditambahkan dataset batik kontemporer dengan menambahkan dataset dalam jumlah yang lebih banyak.

DAFTAR PUSTAKA

- [1] J. Kusanti and R. Agus, “Application System for Identification of Surakarta Traditional Batik Images (SABATARA),” *Sinkron*, vol. 4, no. 1, p. 5, 2019, doi: 10.33395/sinkron.v4i1.10202.
- [2] Y. Gultom, A. M. Arymurthy, and R. J. Masikome, “Batik Classification using Deep Convolutional Network Transfer Learning,” *J. Ilmu Komput. dan Inf.*, vol. 11, no. 2, p. 59, 2018, doi: 10.21609/jiki.v11i2.507.
- [3] Nuraedah and M. Bakri, “Klasifikasi Motif Kain Tradisional Batik Bomba Kaili,” *Semin. Nas. Sist. Inf. UNMER Malang*, no. 14 September, pp. 715–723, 2017.
- [4] Y. Sari, “Klasifikasi Pengenalan Motif Batik Berbasis Image Retrieval,” *Jukung J. Tek. Lingkungan*, vol. 4, no. 2, pp. 27–33, 2018.
- [5] H. Kahamdany, D. P. D. Kusuma, and A. B. Osmond, “Pengembangan Motif Karang Jenis *Leptoseria Papyracea* Pada Aplikasi Batik Berbasis Web,” *E-Proceeding Eng.*, vol. 5, no. 1, pp. 1065–1071, 2018.
- [6] S. R. Dewi, “Deep Learning Object Detection Pada Video,” 2018.

- [7] K. P. Danukusumo, Title, " UNIVERSITAS ATMA JAYA YOGYAKARTA, 2017.
- [8] T. Nurhikmat, "Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek," UNIVERSITAS ISLAM INDONESIA YOGYAKARTA, 2018.
- [9] W. S. Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.
- [10] M. A. Abu, N. H. Indra, A. H. A. Rahman, N. A. Sapiee, and I. Ahmad, "A study on image classification based on deep learning and tensorflow," *Int. J. Eng. Res. Technol.*, vol. 12, no. 4, pp. 563–569, 2019.
- [11] F. Jalled and I. Voronkov, "Object Detection using Image Processing," pp. 1–6, 2016.
- [12] L. A. Elrefaei, M. Omar Al-musawa, and N. Abdullah Al-gohany, "Development of an Android Application for Object Detection Based on Color, Shape or Local Features," *Int. J. Multimed. Its Appl.*, vol. 9, no. 1, pp. 21–30, 2017, doi: 10.5121/ijma.2017.9103.
- [13] C. E. M. Taskin, A. Yigit, D. Taskin, and A. C. Taskin, "Image Processing on Android Devices With Opencv," *Techsyst 2017*, no. May, 2017.
- [14] A. Maulana, "Identifikasi Jenis Daging Menggunakan Image Processing Android," Universitas Lampung, 2018.
- [15] I. G. S. Masdiyasa, S. Bhirawa, and S. Winardi, "Identifikasi Plat Nomor Kendaraan Bermotor Menggunakan Metode Multi-Step Image Processing Berbasis Android," vol. V, 2019.
- [16] M. Y. Gomà, "Image Recognition with Deep Learning Techniques and Tensorflow," Universitat Politècnica de Catalunya (UPC) - BarcelonaTech, 2016.
- [17] M. Shaha and M. Pawar, "Transfer Learning for Image Classification," 2018, doi: 10.1109/ICECA.2018.8474802.
- [18] M. A. Pangestu and H. Bunyamin, "Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model," *Tek. Inform. dan Sist. Inf.*, vol. 4, no. 2, pp. 337–344, 2018.
- [19] M. Shu, "Deep learning for image classification on very small datasets using transfer learning," *Creat. Components*, 2019.
- [20] M. A. H. Abas, N. Ismail, A. I. M. Yassin, and M. N. Taib, "VGG16 for plant image classification with transfer learning and data augmentation," *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 90–94, 2018, doi: 10.14419/ijet.v7i4.11.20781.
- [21] O. Alsing, "Mobile Object Detection using TensorFlow Lite and Transfer Learning," KTH Royal Institute of Technology, Stocckholm, Sweden, 2018.
- [22] R. R. Ananda, "Pengembangan Media Pembelajaran Scrum Interaktif Berbasis Aplikasi Android," UIN Alauiddin Makassar, 2018.