

Penerapan Metode Deep Learning Menggunakan Algoritma CNN based Recommendation pada Aplikasi E-Commerce Gols (Studi Kasus : PT. Cipta Giri Sentosa)

Dedy Putra Romadhon¹, Ricky Eka Putra²

^{1,2} Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya

¹dedy.17051204060@mhs.unesa.ac.id

²rickyeka@unesa.ac.id

Abstrak — Perkembangan zaman Teknologi informasi dan komunikasi yang sangat pesat di era sekarang, menggiring masyarakat untuk melakukan banyak hal secara *online*. Efektivitas dan efisiensi yang sangat tinggi menjadi hal yang dapat diandalkan bagi masyarakat untuk mempermudah dalam kegiatan sehari-hari. Termasuk pada *trend* berbelanja masyarakat dimana sebelumnya dilakukan secara *offline* yang mengharuskan masyarakat untuk datang ke lokasi perbelanjaan / toko untuk membeli barang yang diinginkan. Kini, masyarakat dapat berbelanja dimanapun dan kapanpun hanya dengan bermodalkan perangkat (HP/Laptop). Tak hanya masyarakat dapat berbelanja, namun masyarakat yang ingin menjual barang apapun, tidak perlu repot untuk membuat gerai/toko, cukup dengan mendaftarkan pada *platform* yang menyediakan jasa jual/beli. Namun terdapat tantangan bagi pembeli dimana dari banyaknya kategori dan jenis barang yang tentunya akan memberi kesulitan bagi pembeli untuk mencari yang diinginkan.

Dalam mengatasi hal tersebut, maka berkembanglah sistem rekomendasi yang memudahkan pembeli untuk memfilter secara spesifik produk yang diinginkan. Penelitian mengenai sistem rekomendasi ini juga mengalami banyak perkembangan. *Deep Learning* merupakan salah satu contoh metode yang dikembangkan untuk sistem rekomendasi ini, dimana metode ini turunan dari *Artificial Neural Network* yang bertujuan untuk membangun saraf visual mirip dengan jaringan otak manusia yang dapat menganalisis dan memproses data besar secara otomatis.

Pada penelitian kali ini metode *Deep Learning* yang digunakan yaitu algoritma CNN (*Convolutional Neural Network*) yang dirancang untuk menganalisis dan memproses 8 kategori *fashion* dengan total 6000 *training data* gambar (750 data gambar per kategori) dan 1000 *test data* gambar (125 data gambar per kategori). Hasil akurasi terbaik yang didapatkan dari *training data* sebanyak 70 epochs yaitu di angka 86,42%.

Kata Kunci - Teknologi, E-Commerce, Deep Learning, Convolutional Neural Network (CNN).

I. PENDAHULUAN

Perkembangan teknologi komunikasi dan informasi kini tiap tahunnya sangatlah berkembang. Dimana didukungnya juga pada perubahan kultur dan kehidupan manusia yang saat ini semua beralih pada digital. Kini media elektronik menjadi salah satu media yang sangat gemar dipakai dan diandalkan. Orang dapat dengan cepat melakukan berbagai jenis aktivitas dimana waktu dan jarak mempunyai dampak yang signifikan terhadap kehidupan sehari-hari. Setiap orang mengharapkan segala aktivitasnya dapat selesai dengan cepat dan cepat. Dan adanya internet menciptakan manusia makin cepat mengenal dan mengetahui dunia karena jangkauan yang sangat luas

dengan waktu yang singkat. Perkembangan pada internet sangatlah banyak memberikan perubahan dalam berbagai kegiatan, termasuk salah satunya kegiatan berbelanja.

Seiring dengan perkembangan teknologi, tren berbelanja saat ini orang-orang mulai beralih ke arah *online*. dimana dulunya orang-orang berbelanja secara langsung di toko kini berpindah menggunakan teknologi yang dapat dioperasikan di rumahnya masing-masing. yang biasanya disebut dengan istilah *e-commerce*. Definisi lain dari perdagangan elektronik adalah penggunaan teknologi pemrosesan informasi dan komunikasi digital dalam melakukan transaksi komersial untuk menciptakan, mengubah, dan mendefinisikan kembali hubungan baru antara penjual dan pembeli. Meningkatnya jumlah informasi produk di Internet menimbulkan tantangan unik bagi kami para pembeli di lingkungan e-commerce. Pembeli sendiri akan sering mengalami kesulitan pada saat mencari suatu produk yang dikarenakan banyaknya suatu produk. Menurut [1] “Sistem rekomendasi merupakan salah satu fungsi dari perangkat lunak yang sangat berguna untuk mempermudah pengguna”. Dengan sistem rekomendasi ini pengguna akan dimudahkan dengan rekomendasi-rekomendasi produk yang sesuai dengan apa yang dia inginkan. Sistem rekomendasi sendiri digunakan oleh di hampir semua area bisnis online atau *e-commerce* di mana konsumen perlu mengetahui informasi untuk mengambil keputusan. Sistem ini dibandingkan dengan sebagai gambaran keinginan dan kebutuhan konsumen melalui pendekatan metodologi rekomendasi di mana item ditemukan. Teknologi sistem rekomendasi ini banyak digunakan oleh perusahaan besar di dunia *e-commerce* diantaranya Shopee, Tokopedia, Bukalapak, dan masih banyak yang lainnya.

Menurut [2] “Secara umum, sistem rekomendasi memiliki tiga kategori dari model yang dapat digunakan, yaitu *Collaborative Filtering*, *Content Based Filtering*, dan *Hybrid Recommendation System*”. *Collaborative Filtering*, memiliki peran penting dalam proses sistem rekomendasi, dan oleh karena itu merupakan metode penyaringan kolaboratif yang paling umum digunakan untuk merancang sistem rekomendasi. Dalam pendekatan ini, rekomendasi untuk setiap pengguna aktif diperoleh dengan membandingkannya dengan preferensi pengguna lain yang menilai produk dengan cara yang mirip dengan pengguna aktif [11]. *Content Based*, Untuk sistem rekomendasi ini adalah sistem yang berdasarkan pilihan pengguna sebelumnya. Deskripsi item dan profil penargetan pengguna memainkan peran penting dalam pemfilteran berbasis konten. Algoritma pemfilteran berbasis konten merekomendasikan item berdasarkan kesamaan. Sedangkan,

Sistem rekomendasi *hybrid* adalah dimana pengombinasian dari 2 metode diatas untuk meningkatkan akurasi lebih tinggi dan efisiensi dari sistem rekomendasi [3]. Penelitian akan sistem rekomendasi ini terus mengalami perkembangan tiap tahunnya, hingga metode *deep learning* juga merupakan salah satu metode yang populer di sistem rekomendasi. Metode *deep learning* ialah turunan dari sebuah *artificial neural network* yang bertujuan untuk membangun saraf visual jaringan otak manusia untuk menganalisis dan memproses data besar yang dapat dipakai secara otomatis mengekstrak representasi data secara hirarkis [4].

Klasifikasi objek dengan menggunakan teknologi *deep learning* telah merubah banyak aspek teknologi dan industri, termasuk pembuatan aplikasi sistem rekomendasi. Dengan kemampuannya untuk mempelajari fitur-fitur dari data secara otomatis, model *deep learning*, khususnya *Convolutional Neural Networks* (CNN), telah menjadi alat yang sangat kuat untuk mengenali dan mengklasifikasikan objek dalam gambar. Penerapan ini dalam aplikasi sistem rekomendasi memungkinkan sistem untuk menawarkan saran yang lebih relevan pada pengguna berdasarkan analisis visual dari item-item yang direkomendasikan. *Deep Learning* adalah bentuk pembelajaran yang melibatkan penggunaan algoritma pemodelan abstraksi tingkat tinggi. Dalam *Deep Learning*, fungsi transformasi *non-linear* yang terdiri dari berbagai lapisan digunakan untuk memproses data secara mendalam. *Neural network* merupakan dasar dari model *Deep Learning*. Prinsip kerja *Deep Learning* melibatkan hierarki proses yang dapat diskalakan, memungkinkannya untuk secara otomatis mengekstraksi fitur-fitur dari data [5].

Penggunaan teknik *deep learning* pada sistem rekomendasi juga dinilai lebih efisien dan tepat sasaran. Pembelajaran mendalam telah secara drastis merevolusi arsitektur rekomendasi dan menawarkan lebih banyak cara untuk meningkatkan kinerja rekomendasi. Kemajuan terbaru oleh sistem pembelajaran mendalam berbasis rekomendasi telah menarik perhatian yang signifikan untuk mengatasi keterbatasan konvensional dan mencapai kualitas rekomendasi yang tinggi. Diantara banyaknya algoritma dari metode *deep learning* yang banyak digunakan yaitu *convolutional neural network* (CNN), *fully convolutional network* (FCN), *autoencoder*, dan *long short-term memory* (LSTM), CNN dan turunannya adalah model yang populer untuk memproses masalah terkait visual termasuk klasifikasi gambar, deteksi objek, dan segmentasi semantik [6]. Algoritma CNN dipilih karena dapat bekerja secara optimal pada data dengan jumlah yang banyak [7]. CNN sendiri juga terdiri dari banyak lapisan yang mengubah input dengan filter konvolusi dari Sebagian kecil. Ada beberapa varian CNN telah diusulkan, seperti MobileNet, LeNet, AlexNet, ZFNet, GoogleNet, ResNet, Dan lain-lainnya. MobileNet, merupakan salah satu varian dari metode CNN yang dapat digunakan untuk mengatasi akan computing yang berlebihan [8].

Kotlin adalah bahasa JVM modern, diadopsi dengan cepat dan menjadi bahasa pemrograman resmi Android. Dengan penggunaannya yang luas, kebutuhan akan analisis kode Kotlin semakin meningkat. Mengekspos semantik kode secara eksplisit dengan format yang terstruktur dengan baik adalah langkah pertama dalam analisis kode dan konstruksi representasi tersebut adalah dasar untuk tugas-tugas hilir. Baru-

baru ini, pendekatan berbasis grafik menjadi cara yang menjanjikan untuk menyandikan semantik kode sumber. Namun, pekerjaan ini terutama berfokus pada pembelajaran representasi dengan interpretasi terbatas dan pengetahuan domain dangkal. Perkembangan semantik kode yang diketahui dalam bahasa pemrograman generasi baru telah diabaikan. Cara membuat pemetaan yang efektif antara kode sumber Kotlin yang ringkas secara alami dengan representasi berbasis grafik perlu dipelajari dengan menganalisis fitur bahasa yang diketahui. Selain itu, kelayakan untuk meningkatkan pemetaan dengan semantik kode yang dipelajari secara otomatis dari program perlu dieksplorasi [9].

TensorFlow sebuah *framework open-source* untuk *machine learning* yang dikembangkan oleh Google, telah menjadi salah satu pilihan utama untuk pembuatan aplikasi sistem rekomendasi. Dengan kemampuan untuk memproses data dalam jumlah besar dan algoritma pembelajaran mesin yang canggih, TensorFlow memungkinkan pengembang dan data scientist untuk merancang, melatih, dan menerapkan model yang dapat memberikan rekomendasi yang relevan dan personal kepada pengguna. TensorFlow adalah perpustakaan pembelajaran mesin tingkat tinggi, dan bahasa yang didukung adalah C++ dan Python. Pengguna tidak perlu menulis kode yang rumit untuk membangun struktur jaringan saraf. Algoritma dan fungsi pengoptimalan yang akan digunakan dalam pemodelan cukup memanggil fungsi yang sesuai di pustaka TensorFlow, sehingga sangat mengurangi ambang batas pembelajaran mendalam. Dengan pesatnya perkembangan teknologi komputer dan peningkatan daya komputasi yang luar biasa, komputer biasa juga dapat membangun model pembelajaran mendalam pada platform TensorFlow, yang mengurangi biaya pembelajaran mendalam dan mempermudah verifikasi algoritma [10].

Berdasarkan penelitian terdahulu yang pernah dilakukan oleh Jinzhu Lu, et. al (2021) yang berjudul "*Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification*" menyimpulkan bahwa metode DL telah diterapkan secara luas dalam deteksi dan klasifikasi penyakit tanaman. Ini telah memecahkan atau menyelesaikan sebagian masalah metode pembelajaran mesin tradisional. DL, yang merupakan cabang pembelajaran mesin, terutama digunakan untuk klasifikasi gambar, deteksi target, dan segmentasi gambar. Dalam makalah ini, kami meninjau jaringan CNN terbaru yang berkaitan dengan klasifikasi penyakit daun tanaman. Kami memperkenalkan proses metode CNN yang diterapkan pada klasifikasi penyakit tanaman dan merangkum prinsip-prinsip DL yang terlibat dalam klasifikasi penyakit tanaman. Selanjutnya, kami membahas arah pengembangan masa depan dalam klasifikasi penyakit tanaman, misalnya elektrofisiologi tanaman dan kombinasi klien telepon seluler dan program sisi server akan menjadi arah penelitian yang baik di masa depan. Kombinasi seperti ini baik untuk praktik dan penerapan metode DL secara real-time dalam klasifikasi penyakit tanaman [11].

Berdasarkan analisis dari beberapa penelitian tersebut, peneliti berkeinginan untuk membuat penelitian yang berjudul "Penerapan Metode *Deep Learning* Menggunakan Algoritma CNN Based Recommendation pada Aplikasi *E-Commerce* GOLs (Studi Kasus : PT. Cipta Giri Sentosa)". Penelitian ini akan menggunakan algoritma CNN dalam penerapan metode

deep learning pada aplikasi e-commerce GOLS di PT. Cipta Giri Sentosa sehingga diharapkan untuk dapat melakukan pengembangan sistem layanan informasi produk e-commerce yang lebih tertata dengan rapi, efisien, dan juga menarik.

II. METODOLOGI PENELITIAN

A. Studi Literatur

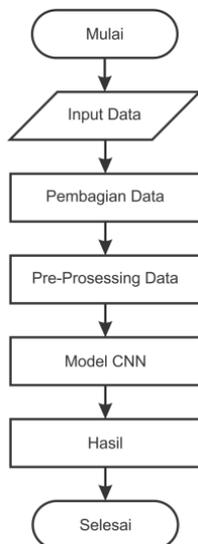
Metode penelitian merujuk pada serangkaian langkah yang dilakukan oleh seorang peneliti untuk mengumpulkan informasi dan data, serta mengolah data yang telah diperoleh. Dalam penelitian ini, teknik yang digunakan termasuk studi literatur dan observasi. Studi literatur dilakukan untuk mencari informasi terkait masalah penelitian, metode yang telah digunakan dalam penelitian sebelumnya, dan penelitian terkait lainnya. Selain itu, observasi dilakukan di PT. Cipta Giri Sentosa dan beberapa situs web yang terdaftar di Google.

B. Jenis Penelitian

Jenis penelitian ini adalah penelitian eksperimen, yaitu metode penelitian yang ditujukan pada pengembangan penerapan metode deep learning dengan menggunakan algoritma CNN serta mempermudah user aplikasi GOLS dalam mencari barang fashion yang menggunakan metode deep learning dengan algoritma CNN. Algoritma CNN akan digunakan untuk memproses data dua dimensi yang dapat dianggap sebagai data citra. Metode dalam konteks ini, mengacu pada prosedur yang dirancang dan digunakan untuk mencapai tujuan tertentu serta bekerja secara optimal.

C. Analisis Dan Gambaran Umum Sistem

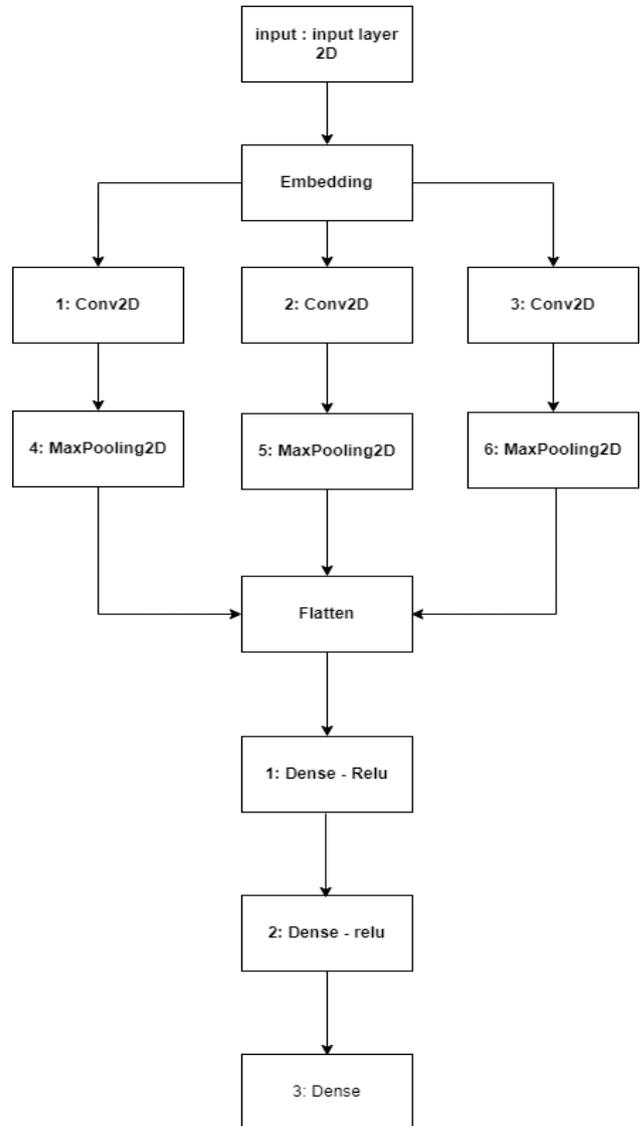
Metodologi penelitian digunakan untuk memastikan bahwa jalur dan hasil penelitian sesuai dengan tujuan awal yang ditetapkan. Metodologi penelitian dapat dijelaskan melalui serangkaian langkah-langkah seperti yang terlihat pada Gambar 1:



Gbr 1. Langkah-langkah pembuatan model CNN

Pada penelitian ini, sistem pencarian data fashion memakai metode Deep Learning algoritma CNN. Proses klasifikasi kategori fashion dimulai dengan mengambil gambar/dataset dari library tensorflow sebagai input data. Lalu, data tersebut akan mengalami proses pre-processing yang meliputi cropping

dan pembersihan noise. Setelah itu, dilakukan pembuatan data training dan testing. Selanjutnya, algoritma CNN akan diimplementasikan untuk memproses data fashion yang telah diinputkan. Setelah proses pemrosesan selesai, hasil kinerja model CNN akan ditampilkan dalam bentuk angka yang dimana setiap nilai memiliki label kategori tersendiri (Kaos atas, Celana Panjang, Sweater, Sandal, Kemeja, Sneaker, Tas, dan Sepatu boot), sehingga dapat dilakukan analisis terhadap gambar-gambar yang diproses menggunakan algoritma CNN. Berikut adalah alur yang akan diikuti dalam penerapan algoritma CNN oleh peneliti.



Gbr 2. Struktur Jaringan CNN

Gambar 2 merupakan Flowchart dari struktur jaringan CNN yang digunakan oleh peneliti. Proses diawali dengan input layer yang sudah di reshape menjadi 2 dimensi. Lalu memasuki proses selanjutnya pada embedding dan memasuki proses Feature Extraction dimana dilakukan konvolusi (Conv2D) yang berfungsi sebagai ekstraksi pada berbagai fitur gambar input untuk dianalisis. Proses selanjutnya Pooling (MaxPooling2D) berfungsi untuk mengurangi biaya komputasi dengan cara memperkecil ukuran peta pada fitur gambar. Setelah proses Feature Extraction yang diakhiri dengan proses Pooling (MaxPooling2D), alur proses selanjutnya masuk pada

proses *Fully Connected Layer* yang diawali dengan Proses *Flatten*, proses tersebut mengubah hasil ekstraksi fitur 2D menjadi 1D, lalu pada proses *Dense 1-2 (Dense - Relu)* digunakan karena data *output* merupakan *multiclass*, dan proses *Dense 3* sebagai pemetaan akhir.

D. Tempat Dan Waktu Penelitian

1. Tempat Penelitian

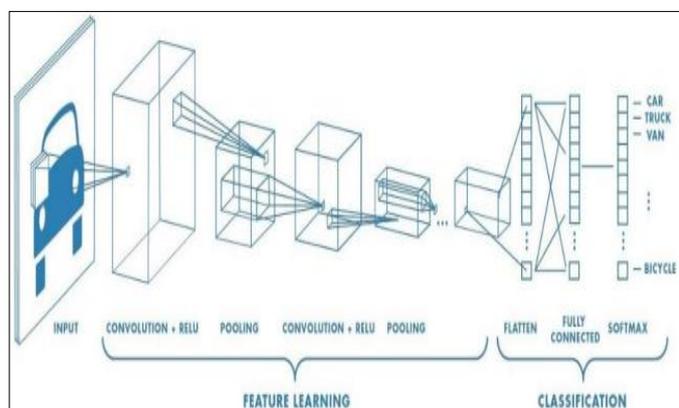
Tempat penelitian merupakan tempat dimana peneliti mendapatkan informasi mengenai data-data yang diperlukan dalam menjalankan penelitian. Dalam hal ini, penelitian dilaksanakan pada PT. Cipta Giri Sentosa yang beralamat di jalan Gubernur Suryo No. 149, Lumpur, Telogo Pojok, Kecamatan Gresik, Kabupaten Gresik.

2. Waktu Pelaksanaan Penelitian

Waktu pelaksanaan penelitian merupakan waktu yang dilakukan dari awal mulai hingga berakhirnya penelitian. Dalam hal ini, waktu pelaksanaan pada penelitian ini dilakukan selama semester ganjil pada tahun 2023.

E. Model Algoritma CNN

Algoritma CNN merupakan salah satu algoritma yang menerapkan teknologi *deep learning*. CNN dapat mengklasifikasi gambar. CNN terdiri dari layer dengan susunan neuron 3D lebar, tinggi, dan kedalaman. Susunan lebar dan tinggi merupakan ukuran lapisan, dan kedalaman adalah jumlah layer [12]. Arsitektur CNN juga dibagi menjadi dua bagian: Feature Extraction Layer dan Fully Connected Layer (FC). Seperti yang terlihat pada gambar 3:



Gbr 3. Proses Convolutional Neural Network (CNN).

Jaringan saraf konvolusional. Kedengarannya seperti kombinasi aneh antara biologi dan matematika dengan sedikit ilmu komputer, namun jaringan ini adalah salah satu inovasi paling berpengaruh dalam visi komputer dan pemrosesan gambar. Jaringan saraf konvolusional adalah versi reguler dari multilayer perceptron (MLP). Ini dikembangkan berdasarkan cara kerja neuron di korteks visual hewan.

F. Flask Library

Flask merupakan *web framework* yang ditulis dalam bahasa Python. Flask memiliki dua *external libraries* yaitu *WSGI toolkit* dan *Jinja2 template engine*. Flask merupakan jenis *micro framework* yang tidak memerlukan library tertentu dalam penggunaannya. Flask dapat menggunakan ekstensi untuk menambahkan fitur dan komponen yang sudah disediakan oleh

pihak ketiga dan tidak terpasang secara standar pada Flask seperti *Form Validation*, *Upload Handling*, dan *Database*. Flask, juga bisa disebut *micro-framework* untuk pengembangan aplikasi web di Python, telah menjadi pilihan populer untuk pembuatan aplikasi sistem rekomendasi berkat kesederhanaan, fleksibilitas, dan kemudahan penggunaannya. Flask menyediakan alat yang diperlukan untuk membangun aplikasi web dengan cepat tanpa memerlukan banyak konfigurasi atau dependensi *eksternal* yang berat. Dalam konteks aplikasi sistem rekomendasi, Flask memungkinkan pengembang untuk dengan mudah membuat API yang dapat melayani rekomendasi kepada pengguna.

Flask dapat dengan mudah terintegrasi dengan library Python untuk *machine learning* seperti TensorFlow, PyTorch, atau Scikit-Learn untuk memproses data dan menghasilkan rekomendasi. Flask juga mendukung penggunaan database SQL dan NoSQL untuk menyimpan data pengguna dan rekomendasi, memudahkan pengelolaan data dalam skala besar. Salah satu pola umum dalam pembuatan sistem rekomendasi adalah menggunakan Flask untuk membangun API yang menerima permintaan dari *frontend*, memproses permintaan tersebut menggunakan model rekomendasi, dan mengembalikan hasil kepada pengguna. Flask adalah pilihan yang menarik bagi pengembang yang ingin membangun sistem rekomendasi yang responsif dan dapat diandalkan. Kombinasi antara kemudahan penggunaan dan fleksibilitas membuat Flask menjadi pilihan yang tepat baik untuk proyek kecil maupun solusi enterprise yang kompleks [13].

G. Firebase Database

Teknologi firebase database adalah *database backend* untuk aplikasi android, iOS dan web. Firebase menyediakan API untuk membuat *database* dan menjemputnya secara *real time* hanya dengan beberapa baris kode. Data disimpan sebagai JSON dan dapat diakses dari semua *platform*. Firebase menyediakan library untuk berbagai *client platform* yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Objective-C dan Node aplikasi Js dan dapat juga disebut sebagai layanan *DbaaS (Database as a Service)* dengan konsep *realtime firebase* digunakan untuk mempermudah dalam penambahan fitur-fitur yang akan dibangun oleh *developer*. Teknologi *smartphone* dengan teknologi *firebase realtime database* digunakan untuk membantu memberikan informasi dengan cepat dan mudah. *Firebase Database* menawarkan solusi yang kuat dan fleksibel untuk penyimpanan dan sinkronisasi data dalam aplikasi sistem rekomendasi. Dengan kemampuan *real-time*, skalabilitas, dan integrasi yang mudah dengan layanan Firebase lainnya, Firebase Database menyederhanakan proses pengembangan dan memungkinkan pembuatan sistem rekomendasi yang responsif dan personal. Bagi pengembang yang mencari cara untuk mempercepat pengembangan aplikasi mereka sambil tetap mempertahankan kualitas dan skalabilitas, Firebase Database merupakan pilihan yang layak dipertimbangkan [14].

H. Proses Pre-Processing Data

Agar diperoleh hasil akurasi yang maksimal maka perlu dilakukan *preprocessing*. *Preprocessing* akan melakukan transformasi data agar data sesuai dengan apa yang peneliti inginkan. Pada penelitian ini ada tiga proses yaitu:

1. Resizing

Proses *resizing* dibutuhkan sebelum masuk ke *model training*, yaitu dengan cara mengecilkan ukuran dari image yang akan di *training*. hal ini guna untuk membantu mempercepat proses *training*.

2. Normalise

Proses *normalise* disini untuk memudahkan gambar dengan cara nilai intensitas cahaya tinggi diubah menjadi putih dan nilai intensitas cahaya rendah diubah menjadi hitam, sehingga kontras gambar teregangkan.

3. Background Removal

Proses *background removal* dibutuhkan untuk memudahkan proses *training*, dimana tidak ada *noise background* di *training* yang dapat membuat proses lebih lambat dan data *training* yang kurang akurat karena ketidakefisienan data *training*.

I. Pelatihan Model

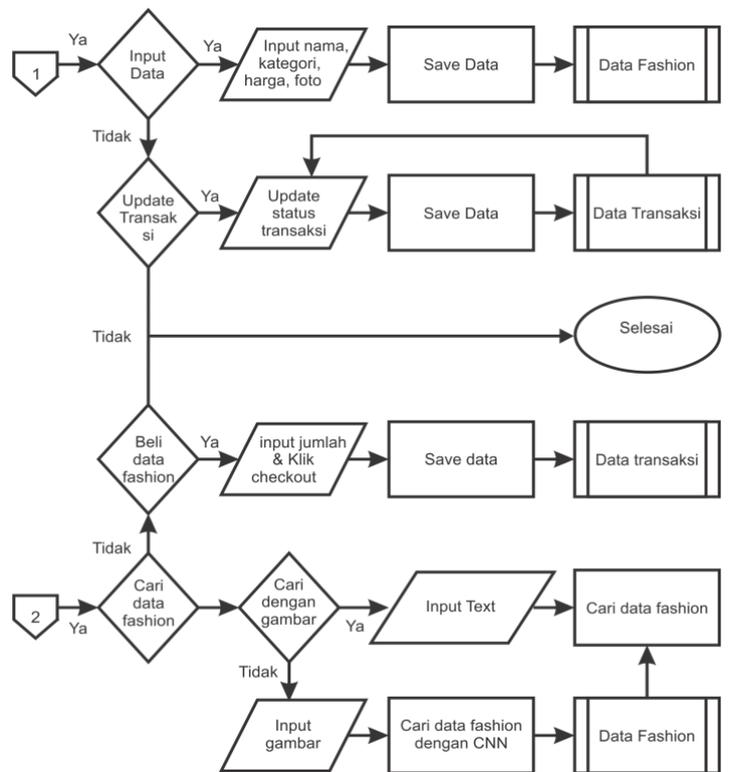
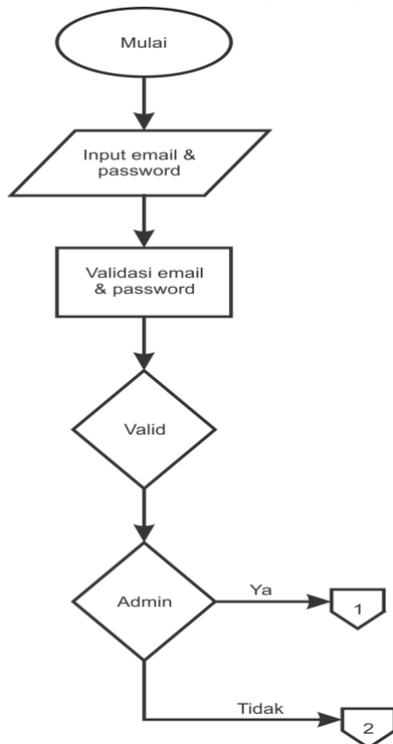
Pelatihan model menggunakan data pelatihan dan data validasi. Dalam penelitian ini, peneliti menggunakan 70 epoch dan ukuran batch 64. Hasil pelatihan model berupa nilai akurasi (accuracy) dan kesalahan klasifikasi (loss).

J. Uji Validasi Data

Pemilihan kumpulan dataset testing disesuaikan dengan urutan iterasi dalam urutan fold. Artinya, iterasi pertama fold 1 dan iterasi kedua fold 2. Proses training dijalankan sebanyak 70 kali pada setiap iterasi. Setelah setiap training selesai, segera dilakukan testing untuk menentukan nilai rata-rata. Kemudian tinggal menghitung rata-rata tingkat akurasinya.

K. Penerapan Pada Aplikasi Android

Untuk penerapan pada aplikasi android akan dijelaskan lewat *flowchart* seperti yang tertera pada gambar 4:



Gbr 4. Flowchart Aplikasi GOLS

Pada gambar 4 merupakan *Flowchart* aplikasi, menjelaskan alur berjalannya aplikasi. Dimulai dari inialisasi “*Start*” lalu dimulai dengan pengisian variabel “*input email dan password*” dilanjutkan dengan proses validasi apakah variabel tersebut valid terdaftar, jika tidak maka akan kembali ke pengisian variabel “*input email dan password*”. validasi memiliki 2 kondisi yaitu “*admin*” dan “*bukan admin*” (bisa dinyatakan sebagai *customer*).

1. Variabel Tervalidasi Sebagai “Admin”

Kondisi “*admin*” dimulai dengan 2 kondisi apakah *admin* akan *input data* atau tidak *input data*. jika *admin* memilih *input data* maka proses selanjutnya yaitu *input data* (nama, kategori, harga dan foto) lalu melakukan penyimpanan pada proses “*save data*” dan tersimpan pada “*data fashion*”. namun ketika memilih kondisi “*tidak input data*” maka alur selanjutnya masuk ke dalam 2 kondisi yaitu *update transaksi* atau tidak *update transaksi*. jika memilih untuk *update transaksi*, *admin* akan melakukan input “*update status transaksi*” dan melakukan proses penyimpanan “*save data*” dan tersimpan pada “*data transaksi*”. Jika *admin* memilih tidak *update transaksi*, maka *flowchart* dinyatakan selesai.

2. Variabel Tervalidasi Sebagai “Tidak Admin”

Kondisi “*tidak admin*” atau *customer* memiliki 2 kondisi awal yaitu apakah akan “*cari data fashion*” atau “*tidak cari data fashion*”. Ketika memilih “*cari data fashion*” maka masuk ke dalam kondisi kedua yaitu “*cari dengan gambar*” atau “*tidak cari dengan gambar*”. Jika *customer* memilih “*cari dengan gambar*” maka akan masuk ke dalam proses input “*input gambar*” lalu akan diproses “*cari data fashion dengan CNN*” dan data akan muncul pada “*data fashion*” sedangkan jika *customer* memilih “*input text*” maka proses

dilanjutkan pada proses “cari data fashion” dan akan muncul pada “data fashion”. Bagi customer yang memilih kondisi “tidak cari data fashion” maka akan masuk ke 2 kondisi “beli data fashion” atau “tidak beli data fashion”. Ketika memilih “beli data fashion” alur berlanjut ke dalam proses input “input jumlah dan klik checkout” akan dilakukan proses “save data” dan masuk ke dalam sub-proses “data transaksi”. Ketika customer memilih “tidak beli data fashion” maka flowchart dinyatakan selesai.

III. HASIL DAN PEMBAHASAN

A. Deskripsi Data

1. Deskripsi Data

Data yang dipakai ialah Dataset MNIST Fashion yang di berikan oleh PT. Cipta Giri Sentosa, kumpulan data yang diberikan terdiri dari 7000 gambar berukuran 28x28, yang masing-masing diklasifikasikan ke dalam satu dari 8 kategori pakaian. Dataset ini sering dipakai sebagai pengganti dari dataset MNIST asli yang berisi angka tulisan tangan untuk berbagai tugas pembelajaran mesin klasik, khususnya untuk pengujian algoritma dalam pengenalan gambar.

2. Pembagian / Perbandingan Data

Penelitian ini menggunakan 8 kategori fashion untuk memenuhi kebutuhan data maka diambil data public dalam proses identifikasi. Dalam pembuatan data train dan test dibagi menggunakan perbandingan 8:2 yang dimana menghasilkan data train sebanyak 6000 (750 data per kategori) dan test sebanyak 1000 (125 per kategori). Kategori data fashion terbagi seperti pada tabel 1:

TABEL I

Kategori	Training	Testing
Kaos Atas	750	125
Celana Panjang	750	125
Sweater	750	125
Sandal	750	125
Kemeja	750	125
Sneakers	750	125
Tas	750	125
Sepatu Boot	750	125

KATEGORI DATASET TRAINING

Pada tabel 1 terdapat kategori dan juga training. Untuk kategori terdapat kaos atas, celana Panjang, sweater, sandal, kemeja, sneakers, tas dan sepatu boot. Lalu untuk training dari tiap kategori secara keseluruhannya memiliki jumlah yang sama, yaitu 750 dataset.

B. Preprocessing Data

1. Resizing Dan Normalisasi Dataset

```

1 # Load Fashion MNIST dataset
2 (train_images, train_labels), (test_images, test_labels) =
3 fashion_mnist.load_data()
4
5
6 # Preprocess the data
    
```

```

7 train_images = train_images.reshape((60000, 28, 28, 1))
8 train_images = train_images.astype('float32') / 255
9
10
11 test_images = test_images.reshape((10000, 28, 28, 1))
12 test_images = test_images.astype('float32') / 255
    
```

Gbr 5. Pengambilan Dan Pemisahan Dataset

Pada gambar 5 diatas di line 2 data train dan data test di load dengan function “fashion_mnist.load_data()” yang sudah ada pada library tensorflow. dari pengambilan data tersebut, data train dan data test perlu melakukan reshape terlebih dahulu. variabel “float32” disini digunakan untuk meminimalisir resource, dimana nantinya akan memudahkan proses training. Data train menjadi 6000 dan data test menjadi 1000. Dalam setiap data dibagi menjadi 8 kategori, data train 6000 (750 per kategori) dan data test 1000 (125 per kategori).

2. Remove Background Dataset

```

# Define function to remove background
def remove_background(image):
    # Convert image to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply Gaussian blur to reduce noise
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    # Perform edge detection
    edges = cv2.Canny(blurred, 50, 150)

    # Find contours in the edge-detected image
    contours, _ = cv2.findContours(edges.copy(),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    
```

Gbr 6. Proses Background Removal

Proses removal background pada gambar 6 memiliki beberapa tahap, yaitu mengkonversi gambar asli menjadi grayscale sehingga memudahkan untuk membedakan warna karena warna RGB diubah menjadi grayscale. Lalu mengaplikasikan Gaussian blur yang berfungsi untuk mengurangi noise gambar. Lalu pada proses selanjutnya yaitu edge detection, dimana secara output akan menampilkan tepi garis sesuai dengan bentuk pada image dengan tujuan hasil gambar yang jelas dan memudahkan proses training. lalu hasil edge tersebut dicari contour dan disimpan.

C. Proses Pembuatan Model

1. Pembuatan Model CNN

```

# Define the model architecture
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'), # Added another
convolutional layer
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'), # Increased units to 128
    Dropout(0.5),
    Dense(64, activation='relu'), # Added another fully connected
layer
    Dropout(0.5),
    Dense(10, activation='softmax')
])
    
```

Gbr 7. Implementasi Alur CNN

Pada gambar 7, implementasi alur CNN diatas terbagi menjadi 2 arsitektur yaitu *Feature Extracting Layer* dan *Fully Connected Layer*. model di training secara sekuensial. Pada *Feature Extracting Layer* digunakan 3 layer *convolutional* untuk hasil training terbaik dan masing-masing menggunakan *activation* “relu” yang berfungsi untuk menormalisasi nilai negatif dari *output* konvolusi. Pada bagian. Pada *Fully Connected Layer* digunakan 3 *Layer Dense*, *Layer* pertama sebanyak 128, *Layer* kedua sebanyak 64 dan yang terakhir yaitu 8 (jumlah *kategori*). Menggunakan fungsi aktivasi *softmax* dikarenakan *class* nya bersifat *multiclass*.

2. Training Model

```
# Define early stopping
early_stopping = EarlyStopping(monitor='val_loss', =5,
restore_best_weights=True)

# Compile the model
model.compile(optimizer=Adam(lr=0.0001), # Lowered learning
rate
                loss='sparse_categorical_crossentropy',
                metrics=['accuracy'])
# Data augmentation
datagen = ImageDataGenerator(rotation_range=15, # Increased
rotation range
                            width_shift_range=0.15, # Added width shift
range
                            height_shift_range=0.15, # Added height shift
range
                            shear_range=0.15, # Added shear range
                            zoom_range=0.15, # Added zoom range
                            horizontal_flip=True) # Added horizontal flip
datagen.fit(train_images) # Fit the data generator
# Train the model
history = model.fit(datagen.flow(train_images, train_labels,
batch_size=64),
                    epochs=70, # Increased number of epochs
                    validation_data=(test_images, test_labels), # Added
validation data
                    callbacks=[early_stopping]) # Added early stopping
```

Gbr 8. Training Model CNN

Pada gambar 8 model di *compile* menggunakan *optimizer* “Adam” dengan *learning rate* = 0.0001. *Data Augmentation* dilakukan untuk menambah *dataset* yang dilatih, sehingga menghindari kemungkinan terjadinya *underfitting model* dikarenakan *dataset* yang di *training* kurang, adapun yang digunakan *rotation range*, *width & height shift range*, *Shear range*, *zoom range*, dan *horizontal flip*. Data dilatih dengan *batch_size* = 64, *epochs* = 70. Data *Training* akan melakukan *callback (Early Stopping)* jika hasil dari tingkat *accuracy* sudah mencapai hasil yang terbaik untuk mempersingkat waktu untuk memproses mencari *accuracy*.

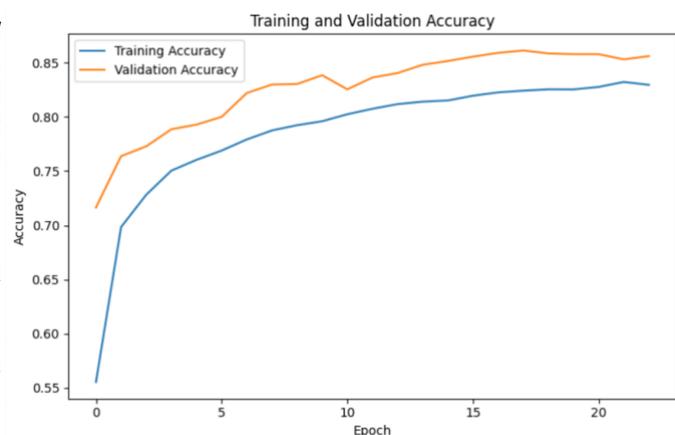
D. Hasil Perhitungan Akurasi

Untuk mengevaluasi keberhasilan model CNN dalam penelitian ini, dilakukan perhitungan akurasi. Akurasi perhitungan dapat dijadikan tolak ukur untuk membandingkan atau mengembangkan model CNN yang dibangun di masa depan. Tabel II menunjukkan hasil akurasi. Pada penelitian ini diperoleh akurasi sebesar 0,8644 atau 86,44%.

TABEL II
HASIL PERHITUNGAN AKURASI DENGAN 100 EPOCH

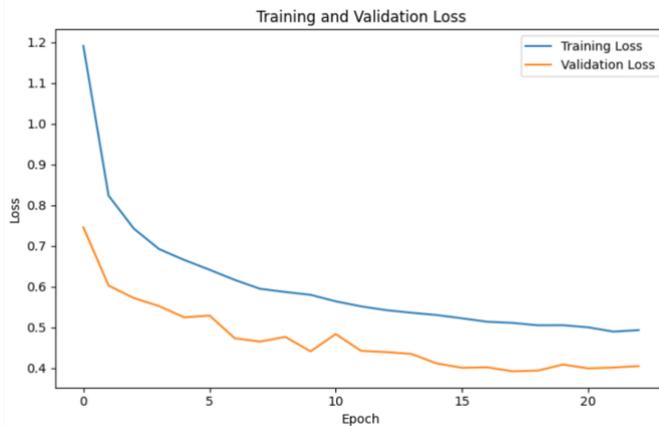
Epoch	Val_Accuracy	Val_Loss
Epoch 1	0.7197	0.7204
Epoch 2	0.7527	0.6352
Epoch 3	0.7709	0.5877
Epoch 4	0.7876	0.5602
Epoch 5	0.7971	0.5426
Epoch 6	0.8066	0.5128
Epoch 7	0.8108	0.5058
Epoch 8	0.8239	0.4940
Epoch 9	0.8284	0.4654
Epoch 10	0.8283	0.4733
Epoch 11	0.8366	0.4428
Epoch 12	0.8397	0.4366
Epoch 13	0.8403	0.4527
Epoch 14	0.8299	0.4581
Epoch 15	0.8404	0.4289
Epoch 16	0.8506	0.4066
Epoch 17	0.8201	0.4788
Epoch 18	0.8531	0.4052
Epoch 19	0.8521	0.4009
Epoch 20	0.8545	0.4025
Epoch 21	0.8546	0.4019
Epoch 22	0.8573	0.3983
Epoch 23	0.8519	0.4067
Epoch 24	0.8590	0.3900
Epoch 25	0.8551	0.3937
Epoch 26	0.8644	0.3816
Epoch 27	0.8578	0.3837
Epoch 28	0.8603	0.3837
Epoch 29	0.8613	0.3837
Epoch 30	0.8529	0.3837
Epoch 31	0.8642	0.3837

Pada tabel 2 diatas terdapat epoch dan juga *accuracy* dari epoch tersebut. Untuk epoch 27 dan 28 mendapatkan masing-masing *accuracy* 0.8578 dan 0.8603. Untuk epoch 29 dan 30 mendapatkan masing-masing *accuracy* 0.8613 dan 0.8529. Dalam program digunakan *EarlyStopping* yang dimana ketika *val_loss* 5 kali perulangan tetapi memiliki nilai yang sama atau tetap maka akan berhenti secara otomatis. kasus ini terjadi pada epoch 31 dimana *val_loss* 0.3837 mendapat 5x perulangan sehingga training model berhenti pada akurasi 0.8642. Berikut ialah hasil dari gambar grafik data *training validation Accuracy* dan data *training validation Loss* dari hasil perhitungan pengujian validasi:



Gbr 9. Grafik Training And Validation Accuracy

Pada gambar 9 merupakan grafik training and validation akurasi yang memiliki hasil akhir akurasi sebesar 86.42% dan cenderung konstan naik grafiknya sehingga termasuk hasil yang bagus karena tidak fluktuatif, dan dengan adanya grafik validation diatas grafik training, menunjukkan bahwa hasil tersebut tidak mengalami adanya underfitting.



Gbr 10. Grafik Training And Validation Loss

Hasil grafik pada gambar 10 menunjukkan bahwa data validation berada pada nilai 0.3837%, sama halnya dengan grafik akurasi bahwa grafik validation loss berada dibawah grafik training loss sehingga dari model grafik tersebut tidak terjadi adanya underfitting, dimana kondisi underfitting dapat dipengaruhi dari dataset yang kurang memadai.

E. Hasil Implementasi Aplikasi

1. Halaman Splash Screen



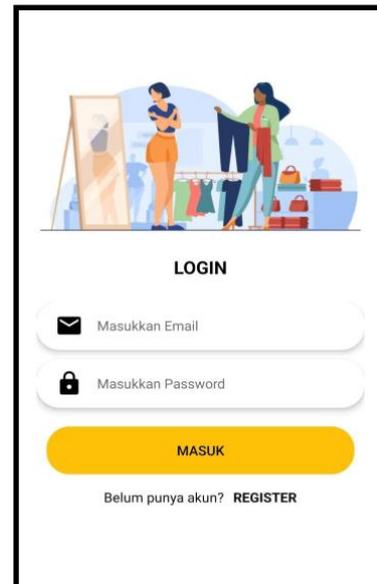
Gbr 11. Halaman Awal Splash Screen

Pada gambar 11 merupakan tampilan halaman awal (*splash screen*) pada aplikasi. salah satu *pattern standard* dalam *Android Development* yang digunakan untuk menampilkan logo dari *brand* aplikasi tersebut.

2. Halaman Login

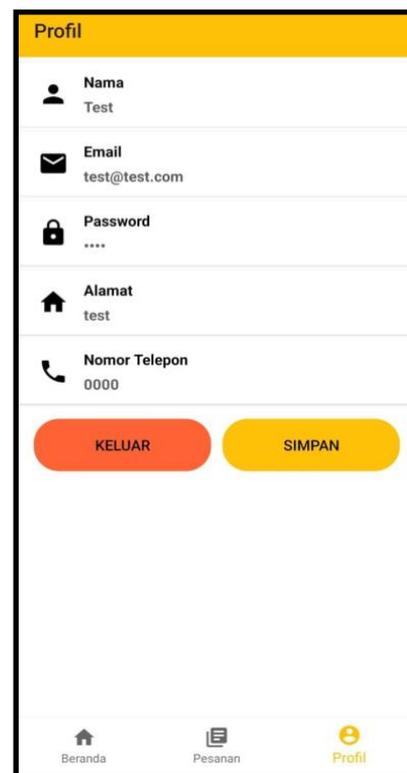
Setelah *splash screen*, akan menuju ke halaman *login* dari aplikasi. Jika *user* yang sudah memiliki akun, terdapat menu *user* (menggunakan email) dan *password* sebagai *authentication* masuk pada aplikasi. Bagi *user* yang belum

memiliki akun bisa menekan menu “REGISTER” untuk membuat akun. Seperti yang terlihat pada gambar 12.



Gbr 12. Halaman Login

3. Halaman Profile

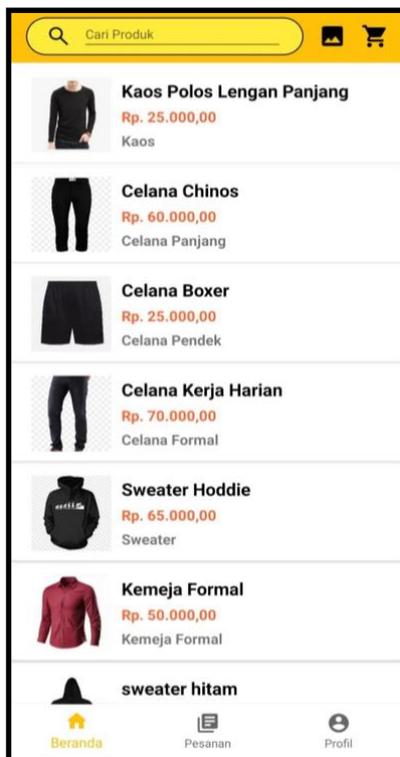


Gbr 13. Halaman Profile

Pada gambar 13 merupakan halaman *profile* yang spesifik menampilkan data dari akun *user* sebagai konfirmasi jika data sudah sesuai dengan data *user* dan profil yang ditampilkan yaitu Nama (nama dari pemilik akun tersebut), Email (email yang dipakai sebagai user pada akun tersebut), Password (sebagai kata sandi *authentication* masuk aplikasi), Alamat (Alamat rumah yang saat ini ditinggali oleh pemilik akun) dan Nomor Telepon (Nomor

telepon yang dapat dihubungi / masih aktif dari pemilik akun). Tombol “SIMPAN” digunakan untuk menyimpan data (jika data sudah sesuai) sedangkan tombol “KELUAR” digunakan untuk keluar dari halaman profil aplikasi.

4. Halaman Beranda



Gbr 14. Halaman Beranda

Halaman beranda ditunjukkan pada gambar 14 dimana merupakan tampilan *default* awal dari aplikasi tersebut, tampak produk yang dijual tampil secara *default*. pada halaman ini tersedia menu “search” yang dapat digunakan untuk mencari secara spesifik barang yang diinginkan.



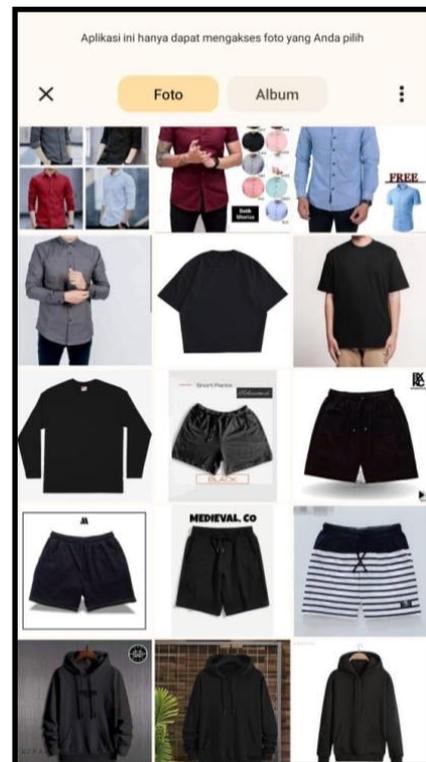
Gbr 15. Tampilan Fitur Picture

Pada gambar 15 terdapat fitur *search by “import picture”* yaitu yang diberi tanda lingkaran merah. fungsi dari fitur tersebut yaitu dapat melakukan pencarian dengan cari membuka galeri *picture* pada perangkat user yang nantinya bisa dipilih untuk di *import* pada aplikasi sebagai *search input*.



Gbr 16. Tampilan Fitur Keranjang

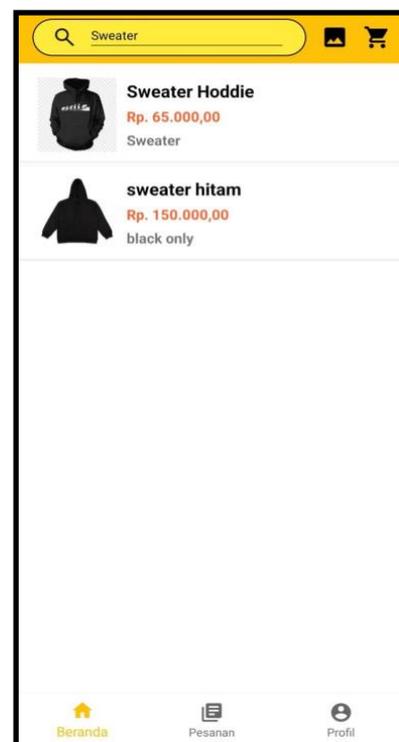
Pada gambar 16 merupakan fitur keranjang yang berguna untuk menampilkan produk-produk yang sudah dipilih oleh user.



Gbr 17. Halaman Search Engine

Gambar 17 merupakan halaman *search engine* menggunakan fitur “*import picture*”, *search engine* akan *generate* sesuai dengan gambar yang dipilih oleh user. sebagai contoh gambar diatas *user* memilih gambar “Sweater” untuk dicari pada aplikasi. maka gambar yang dipilih tersebut akan dijadikan parameter *search engine* untuk filter produk sesuai dengan pilihan user.

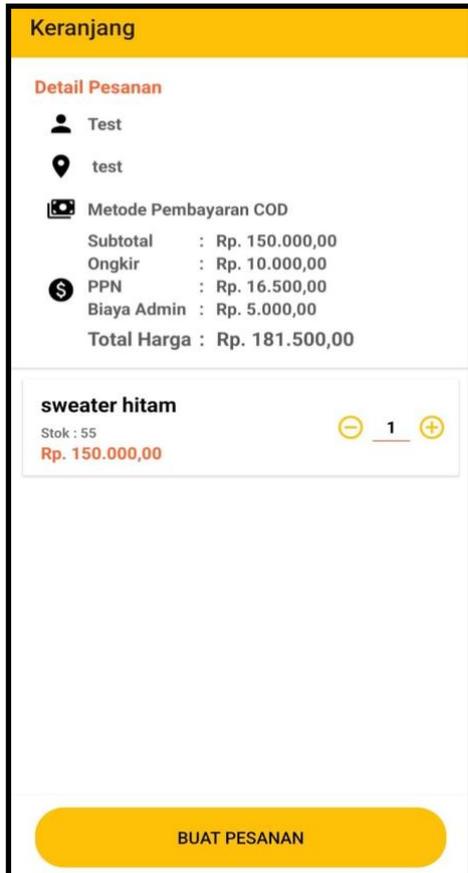
5. Halaman Hasil Generate Search Engine



Gbr 18. Tampilan Hasil Generate Search Engine

Gambar 18 merupakan hasil dari *generate search engine* dari gambar 17 user memilih gambar “sweater” melalui galeri di *smartphone* pribadi. Hal tersebut adalah hasil dari proses *training* model dengan algoritma CNN yang dapat mengenali gambar lalu mengubahnya menjadi *text* sesuai dengan gambar yang dipilih. Sehingga user dapat mencari produk tanpa harus menuliskan secara manual melainkan dapat dengan “*Import Picture*” dari galeri gambar user.

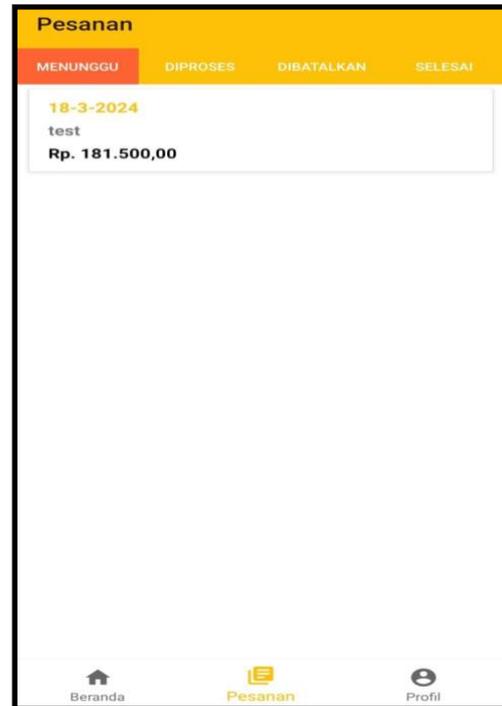
6. Halaman Keranjang



Gbr 19. Halaman Keranjang

Gambar 19 merupakan tampilan dari halaman keranjang jika user telah selesai memilih produk. menu tersebut dapat diakses melalui menu keranjang pada gambar 16. Tampilan tersebut merupakan konfirmasi pembelian (detail pesanan) produk disertai dengan spesifik nama penerima, alamat penerima, rincian total harga serta kuantitas dan tampilan produk yang dipilih oleh user. User dapat menekan tombol “BUAT PESANAN” jika detail pesanan sudah sesuai.

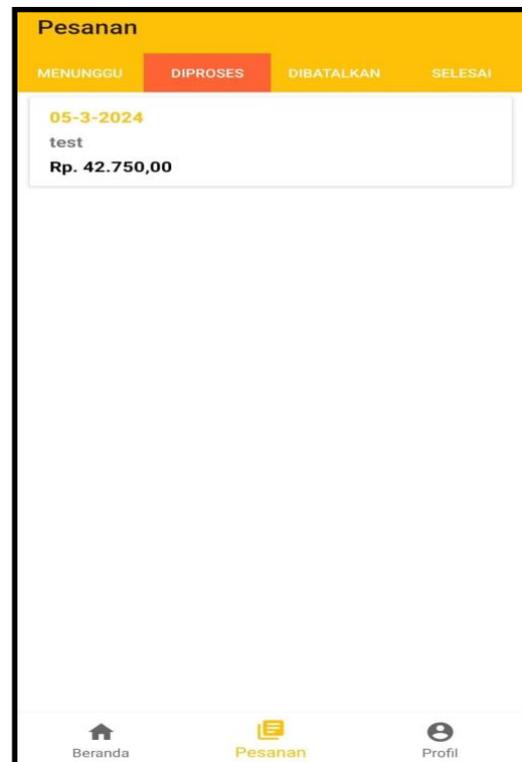
7. Halaman Pesanan (Menunggu)



Gbr 20. Halaman Pesanan (Menunggu)

Gambar 20 merupakan tampilan halaman menunggu pesanan yang sedang diproses. User dapat melihat tampilan jenis produk yang dipesan dengan rincian harga dan tanggal pemesanan.

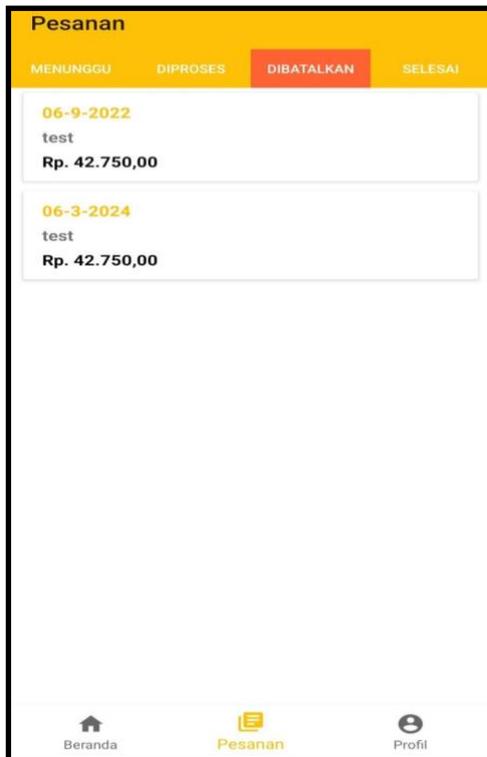
8. Halaman Pesanan (Diproses)



Gbr 21. Halaman Pesanan (Diproses)

Gambar 21 merupakan tampilan halaman ketika pesanan sudah diproses oleh penjual. Sama halnya dengan Gambar 20 dimana user dapat melihat tampilan produk serta rincian harga dan tanggal proses pemesanan.

9. Halaman Pesanan (Dibatalkan)



Gbr 22. Halaman Pesanan Dibatalkan

Gambar 22 merupakan halaman pesanan yang dibatalkan dan tidak akan dilanjutkan proses pengiriman. User dapat melihat tanggal pesanan yang dibatalkan beserta produk dan rincian harga.

10. Halaman Pesanan (Selesai)



Gbr 23. Halaman Pesanan (Selesai)

Pesanan selesai dapat dilihat pada gambar 23 dimana terdapat detail tanggal diterima barang, jenis produk dan harga produk.

IV. KESIMPULAN

Dari hasil pengujian yang telah Dilakukan, penulis mengambil beberapa kesimpulan pada penerapan metode deep learning menggunakan algoritma CNN based recommendation yaitu:

1. Hasil pengujian 8 kategori *fashion* yang terdiri dari *Training Data* sebanyak 6000 (750 data per kategori) dan *Test Data* sebanyak 1000 (125 data per kategori), menghasilkan akurasi sebesar 86,42.
2. Hasil penerapan algoritma CNN dapat diimplementasikan dalam sistem rekomendasi pada aplikasi *e-commerce* GOLS dengan menu fitur *search by image* menjadi *text*.

V. SARAN

Adapun saran untuk pengembangan pada penelitian ini supaya menjadi lebih baik lagi, terdapat beberapa saran yaitu:

1. Penambahan variasi *dataset* yang memiliki beberapa aspek. Aspek yang pertama dari penambahan kuantitas *dataset*, hal ini dapat dilakukan untuk menambah akurasi akhir pengujian. Aspek yang kedua yaitu berupa penambahan pada kategori yang bisa diimplementasikan lebih luas pada jenis *fashion*.

2. Memberian variasi tambahan pada aplikasi e-commerce GOLS dengan menu mengambil foto secara langsung (tidak import picture).

UCAPAN TERIMA KASIH

Peneliti senantiasa mengucapkan rasa syukur yang sangat besar kepada Tuhan Yang Maha Esa atas segala berkah dan juga pertolonganNya, sehingga peneliti mampu menyelesaikan proyek dan artikel ilmiah ini dengan baik. Terima kasih penulis ucapkan juga kepada Dosen Pembimbing yang telah memberi arahan dan masukan yang sangat bermanfaat buat peneliti. Kedua orang tua peneliti yang tak henti-hentinya mendoakan dan membantu dalam segala hal. Keluarga Besar Dinas KBPPPA Kab. Gresik Terutama Sub. Bidang Keuangan yang memberikan *support*. Semua saudara-saudara Jurusan Teknik Informatika terutama kelas TI 2017 B yang selalu memberikan dukungan sehingga peneliti dapat menyelesaikan skripsi ini dengan lancar.

REFERENSI

- [1] Fathurrahman, Mohammad Iqbal, Dade Nurjanah & Rita Rismala. (2017). "Sistem Rekomendasi Pada Buku Dengan Menggunakan Metode *Trust-Aware Recommendation*." *E-Proceeding of Engineering*. Vol. 4, No. 3, 4966–77.
- [2] Zhang, Shuai, Lina Yao, Aixin Sun & Yi Tay. (2019). "Deep Learning Based Recommender System: A Survey and New Perspectives." *ACM Computing Surveys*. Vol. 52, No. 1, :1–35. doi: 10.1145/3285029.
- [3] B.Thorat, Poonam, R. M. Goudar & Sunita Barve. (2015). "Survey on Collaborative Filtering, Content-Based Filtering and Hybrid Recommendation System." *International Journal of Computer Applications*. Vol. 110, No. 4, 31–36. doi: 10.5120/19308-0760.
- [4] Liu, Yao, Hongbin Pu & Da Wen Sun. (2021). "Efficient Extraction of Deep Image Features Using Convolutional Neural Network (CNN) for Applications in Detecting and Analysing Complex Food Matrices." *Trends in Food Science and Technology*. Vol. 113, 193–204. doi: 10.1016/j.tifs.2021.04.042.
- [5] C. Imam, Sutrisno, A. S. Arief, H. Uswatun & I. F. Yessica. (2020). "Ai, Machine Learning & Deep Learning (Teori & Implementasi)". 1-455.
- [6] Teng, Jianing, Dong Zhang, Dah Jye Lee & Yao Chou. (2019). "Recognition of Chinese Food Using Convolutional Neural Network." *Multimedia Tools and Applications*. Vol. 78, No. 9, 11155–72. doi: 10.1007/s11042-018-6695-9.
- [7] Informatika, Prodi Teknik. (2021). "Klasifikasi Keluhan Pelanggan Menggunakan Metode Convolutional Neural Network (Cnn) Studi Kasus Pt Telkom Akses Abstrak".
- [8] Hu, Tianqing, Mohammad Khishe, M. Mohammadi, Gholam Reza P., Sarkhel H. Taher Karim & Tarik A. Rashid. (2021). "Real-time COVID-19 Diagnosis from X-Ray Images Using Deep CNN and Extreme Learning Machines Stabilized by Chimp Optimization Algorithm." *Biomedical Signal Processing and Control* 68 (January) : 102764. doi: 10.1016/j.bspc.2021.102764.
- [9] Lu Li & Yan Liu. (2022). "Mapping Modern JVM Language Code to Analysis-Friendly Graphs: A Study with Kotlin". *International Journal of Software Engineering and Knowledge Engineering*. Vol. 32, No. 11, pp. 1667-1688.
- [10] Liang Yu, Binbin Li & Bin Jiao (2019). "Research and Implementation of CNN Based on TensorFlow". *IOP Conf. Series: Materials Science and Engineering*. IOP Publishing. doi:10.1088/1757-899X/490/4/042022.
- [11] J. Lu, L. Tan & H. Jiang. (2021). "Review on convolutional neural network (CNN) applied to plant leaf disease classification". *Agriculture*. Vol. 11, No. 8, 1-18.
- [12] Zufar, M. (1998). "Introductory Computer Vision and Image Processing". *Sens. Rev*, Vol. 18, No. 3, 2-4.
- [13] Graciela Fausten N. & Pratyaksa Ocsa N. S. (2022). "Implementasi Flask Pada Sistem Penentuan Minimal Order Untuk Tiap Item Barang Di Distribution Center Pada Pt Xyz Berbasis Website". *Jurnal Mnemonic*. Vol 5, No.2.
- [14] M. Rifai Sali & Lanto Ningrayati Amali. (2019). "Aplikasi Kajian Islam Dengan Teknologi Firebase Realtime Database". *Jambura (Journal of Informatics)*.