

Image Recognition Menggunakan Metode Cosine Distance untuk Aplikasi Penanganan Food Waste

Monica Chandra¹, Edwin Pramana¹

¹Departemen Informatika, Fakultas Sains dan Teknologi, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Indonesia

Corresponding author: Penulis A. Monica Chandra (e-mail: monica1@mhs.stts.ac.id).

ABSTRACT The Food and Agriculture Organization of the United Nations (FAO) claimed that 33% - 50% of the food that has been produced is not consumed properly. Also, 11% of food products purchased are wasted, and even if they are not being opened. In 2016-2017, Indonesia itself has become the second largest country after Saudi Arabia, which produced the most food waste in the world. These can have a bad impact on our environment. Therefore, the “Jangan Dibuang” application was made with the aim of reducing the food waste produced. This app was developed for the Android platform with Flutter framework and Amazon Web Service Aurora Database. The “Jangan Dibuang” application is also equipped with image recognition that uses Tensorflow to simplify food searching with an image, which later the image’s feature is extracted into matrix and being compared with Cosine Distance method. The “Jangan Dibuang” application can be used by three types of actors, namely administrators, food providers, and buyers. The testing involved seven food providers and 20 buyers. Based on the results of the testing, 201 transactions were made, which saved 285 products. 59 of 201 transactions were aimed at donations. The functionality of the food providers’ application gets a score of 79.98% for excellent criteria. For the functionality of the buyers’ application, the value obtained is 83% for excellent criteria. For the Image Recognition itself shows the accuracy of 93% after using EfficientNetV2 Keras Application Model which helps to recognize two pictures with different lighting and angle.

KEYWORDS Amazon Web Services, Aurora, Flutter, Food Waste, Tensorflow

ABSTRAK Badan Pangan PBB (FAO) menyatakan 33% - 50% makanan yang telah diproduksi, tidak dikonsumsi dengan semestinya. Selain itu, 11% produk makanan yang dibeli terbuang bahkan tidak dibuka. Tahun 2016-2017, Indonesia sendiri telah menjadi negara terbesar kedua setelah Arab Saudi yang menghasilkan food waste terbanyak di dunia. Penumpukan limbah ini berdampak pada lingkungan. Oleh karena itu, aplikasi “Jangan Dibuang” dibuat dengan tujuan untuk mengurangi food waste yang dihasilkan. Aplikasi ini dibuat untuk platform Android dengan framework Flutter dan database Amazon Web Service Aurora. Selain itu, aplikasi ini juga dilengkapi dengan fitur image recognition yang memanfaatkan Tensorflow untuk mempermudah pencarian makanan dengan sebuah gambar yang mana gambar tersebut akan diekstrak fiturnya menjadi matriks yang kemudian dibandingkan dengan metode Cosine Distance. Aplikasi “Jangan Dibuang” dapat digunakan oleh 3 jenis aktor, yaitu administrator, penyedia makanan, dan pembeli. Uji coba dilakukan terhadap 7 penyedia makanan dan 20 pembeli. Berdasarkan hasil uji coba yang telah dilakukan, didapatkan 201 transaksi, yang mana telah menyelamatkan 285 limbah makanan. 59 dari 201 transaksi ditujukan untuk donasi. Fungsionalitas aplikasi penyedia makanan mendapatkan nilai 79,98% untuk kriteria sangat baik. Untuk fungsionalitas aplikasi pembeli, nilai yang didapatkan adalah 83% untuk kriteria sangat baik. Dari sisi Image Recognition sendiri menunjukkan akurasi 93,3% setelah menggunakan Keras Application Model EfficientNetV2 yang membantu mengenali kedua gambar walaupun dengan pencahayaan dan posisi pengambilan yang berbeda.

KATA KUNCI Amazon Web Services, Aurora, Flutter, Food Waste, Tensorflow

I. PENDAHULUAN

Salah satu permasalahan yang sedang dihadapi dunia saat ini adalah limbah. Limbah terdiri dari berbagai jenis, salah satunya limbah makanan atau yang lebih dikenal sebagai *food waste*. Badan Pangan PBB (Food and Agriculture Organization of The United Nations/FAO) menyatakan bahwa 33% hingga 50% makanan yang telah diproduksi, tidak dikonsumsi dengan semestinya. Hal ini ditunjukkan dengan adanya penelitian yang menyatakan bahwa sebanyak 11% produk makanan yang dibeli terbuang bahkan tidak dibuka [1].

Beberapa toko roti telah menerapkan langkah positif untuk mengurangi limbah pangan ini. Alih-alih membuang sisa roti yang tidak terjual pada hari tersebut, toko roti ini memiliki campaign menjual sisa roti tersebut separuh harga pada hari tertentu dan jam tertentu. Oleh karena itu, dengan campaign yang telah dilakukan, hal ini dapat mengurangi limbah makanan sekaligus menekan kerugian toko atas roti-roti yang tidak terjual. Sayangnya, pembelian tersebut hanya dapat dilakukan on the spot. Artinya, pembeli harus datang di toko dan memilih roti yang tersisa. Hal ini dirasa kurang efektif, mengingat setiap orang memiliki kesibukannya masing-masing dan belum tentu memiliki waktu untuk datang on the spot. Oleh karena itu, pembelian food waste seperti ini dapat dilakukan melalui aplikasi.

Mengingat penduduk Indonesia yang mayoritas merupakan pengguna Android, yakni 41 juta pengguna atau pangsa pasarnya sebesar 94%, maka aplikasi “Jangan Dibuang” akan dibuat untuk platform Android. Banyak framework yang mendukung developer untuk mengembangkan aplikasi Android, salah satu yang sedang populer saat ini adalah Flutter yang menawarkan fitur cross-platform dan tampilan yang fleksibel dan menarik. Selain itu, tak dapat dipungkiri, bahwa kecerdasan buatan telah menjangkau hampir seluruh aspek kehidupan manusia. Oleh karena itu, untuk menerapkannya, “Jangan Dibuang” juga dilengkapi dengan fitur Image Recognition untuk memudahkan pembeli untuk mencari makanan sejenis dengan cara memfoto makanan yang dimilikinya serta memudahkan penjual agar mereka tidak perlu menginputkan makanan dari awal lagi setiap harinya, melainkan hanya mengupdate stok setelah aplikasi menemukan makanan yang sesuai dengan hasil Image Recognition. Pada kasus ini, pencarian makanan yang sesuai dilakukan dengan mengekstraksi fitur-fitur pada sebuah gambar dan membandingkannya dengan fitur yang telah tersimpan sebelumnya. Pengukuran kesamaan fitur dilakukan menggunakan Cosine Distance dikarenakan tingkat efisiensinya [2].

Aplikasi “Jangan Dibuang” dibuat agar lebih banyak restoran, bakery, dan penyedia makanan lainnya lebih *aware* dengan food waste dengan cara menjual sisa makanannya melalui aplikasi ini. Selain mengurangi food waste dan meminimalisir kerugian pemilik brand, aplikasi ini juga bermanfaat bagi calon “pahlawan penyelamat” sisa makanan

yang hampir terbuang dengan harga miring. Dengan harga miring, setidaknya beban finansial calon pembeli juga berkurang, terutama bagi anak kos dan orang-orang baik yang hendak berbagi makanan/bakti sosial.

II. TINJAUAN PUSTAKA

Flutter merupakan teknologi milik Google yang dapat digunakan untuk membangun aplikasi dengan tampilan UI yang apik. Selain itu, keunggulan Flutter adalah source code yang dibuat dapat di-compile secara native ke dalam aplikasi mobile, web, dan desktop hanya dari satu basis kode. Flutter menggunakan bahasa Dart [11], sebuah bahasa pemrograman yang dikembangkan oleh Google. Amazon Web Services (AWS) adalah platform cloud paling komprehensif dan digunakan secara luas di dunia. Hingga saat ini, AWS menawarkan lebih dari 200 layanan unggulan yang lengkap dari pusat data secara global, salah satunya adalah Aurora. Amazon Aurora adalah basis data relasional yang kompatibel dengan MySQL dan PostgreSQL [7]. Database ini dibangun AWS untuk cloud, yang menggabungkan kinerja dan ketersediaan basis data perusahaan tradisional dengan kesederhanaan dan keefektifan biaya basis data sumber terbuka.

Firestore Cloud Messaging (FCM) dulu dikenal sebagai Google Cloud Messaging (GCM). Firestore Cloud Messaging (FCM) dapat mengirim dan menerima pesan.. Selain tools yang telah disebutkan, digunakan juga Xendit dan Web Service. Xendit adalah perusahaan fintek Indonesia yang menyediakan infrastruktur pembayaran untuk Indonesia. Xendit membantu marketplace mengirimkan pembayaran, pinjaman, dan mendeteksi penipuan. Web service yang merupakan aplikasi yang berisi sekumpulan basis data dan perangkat lunak atau bagian dari program perangkat lunak yang diakses secara remote oleh piranti dengan perantara tertentu. Web service mampu menukar data tanpa memandang sumber database, bahasa yang digunakan, dan pada platform apa data tersebut dikonsumsi. Laravel menyediakan fitur dasar untuk membangun fullstack application, yang menangani request, routing, controller, service, domain/model, hingga view. Namun, Laravel bisa juga dibangun sebagai web service RESTful API.

Image Recognition adalah proses identifikasi dan pendeteksian sebuah objek atau fitur di dalam sebuah gambar digital atau video. Tensorflow adalah library perangkat lunak open source untuk Machine Learning [14]. TensorFlow dapat digunakan dalam berbagai tugas tetapi memiliki fokus utama pada training Neural Network. Tensorflow banyak digunakan dalam kasus-kasus Machine Learning, seperti face recognition untuk mengenali wajah orang pada log absen perkantoran atau dipadukan dengan Embedded System.

Selain penggunaan Tensorflow, untuk memilih fitur-fitur yang digunakan, digunakan Principal Component Analysis. Pada dasarnya analisis komponen utama (PCA) [13] bertujuan menerangkan struktur varians-kovarians melalui

kombinasi linear dari variabel-variabel. Secara umum analisis komponen utama bertujuan untuk mereduksi data dan menginterpretasikannya. Meskipun dari p buah variabel asal dapat diturunkan menjadi p buah komponen utama untuk menerangkan keragaman total sistem (p buah variabel), namun seringkali keragaman total itu dapat diterangkan secara memuaskan oleh sejumlah kecil komponen utama, misal, oleh k buah komponen utama, dimana $k < p$ (k lebih kecil dari pada p). Dalam hal ini, k buah komponen utama dapat menggantikan p buah variabel asal. Analisis komponen utama sering kali bukan merupakan akhir dari suatu pengolahan data tetapi juga merupakan tahap (langkah) antara dalam kebanyakan penelitian yang bersifat lebih luas.

Distance Similarity digunakan untuk menentukan kesamaan antara dokumen atau vektor [6]. Secara matematis, hal ini digunakan untuk mengukur cosine sudut antara dua vektor yang diproyeksikan dalam ruang multi-dimensi. Hubungan antara Distance Similarity dan Cosine Distance berkebalikan. Cosine Distance akan menurun apabila Similarity tinggi. Rumus untuk Cosine Distance dapat dilihat pada formula 1 [9].

$$\text{Cosine Distance} = 1 - \frac{u \cdot v}{\|u\| \|v\|} \quad (1)$$

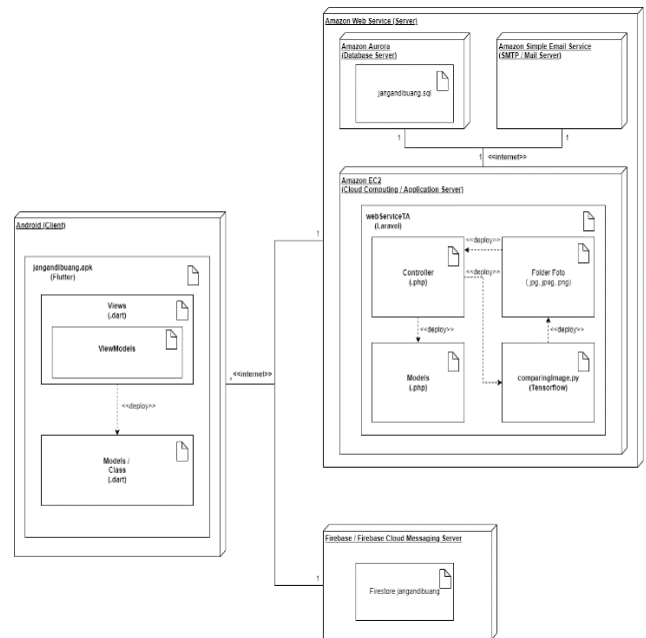
Penelitian pengenalan makanan menggunakan Image Recognition telah banyak dilakukan sebelumnya. Kong et al. [3] pernah membuat aplikasi mobile bernama DietCam. DietCam menggunakan algoritma SIFT feature descriptor dan Nearest Neighbour untuk mengenali makanan. Melalui penelitian ini, akurasi yang didapatkan sebesar 92%. Andrews Samraj et al. [4] dalam artikelnya yang berjudul "Food Genre Classification from Food Images by Deep Neural Network with Tensorflow and Keras" juga pernah melakukan proses klasifikasi jenis makanan dengan Neural Network memanfaatkan Tensorflow dan Keras. Andrews bersama timnya menggunakan dataset berisi 170 gambar makanan dalam bentuk array berdimensi (350, 350, 3). Untuk meningkatkan akurasi dan mengurangi loss pada model, tim Andrew menggunakan teknik Back Propagation Multilabel Learning (BP MLL). Dalam pengerjaannya, library Keras digunakan untuk back propagation dan Tensorflow digunakan untuk mengasosiasikan data point dengan sebuah set berisi label-label. Penelitian lainnya yang masih terkait dengan makanan dilakukan oleh Nupur Bhave dan Dipti Belsare [5] untuk mengenali makanan beserta estimasi kandungan nutrisi pada makanan tersebut. Peneliti menggunakan deep learning dengan metode Convolutional Neural Networks (CNN). Pengaplikasian CNN dilakukan dengan menggunakan Tensorflow dan Keras.

III. RANCANGAN SISTEM

Pada bagian ini akan dijelaskan mengenai rancangan sistem, yakni rancangan arsitektur mencakup alur image recognition, serta use case aktor-aktor yang terlibat.

A. ARSITEKTUR SISTEM

Bagian A menjelaskan arsitektur sistem yang digunakan dalam aplikasi. Arsitektur yang akan dibahas adalah dituangkan dalam bentuk Deployment Diagram agar setiap komponen dapat dijelaskan lebih rinci.



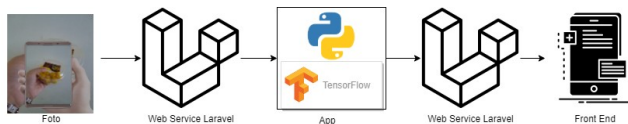
GAMBAR 1. Arsitektur Sistem Aplikasi.

Gambar 1 menunjukkan arsitektur sistem yang dibuat. Aplikasi yang dibuat berbasis Android dengan menggunakan framework Flutter. Framework Flutter memiliki arsitektur Model-View-View Model atau disingkat sebagai MVVM. Data yang ditampilkan diambil dari web service berbasis Laravel yang terhosting di Amazon Web Service (AWS). Dalam hal ini, View Model yang bertugas sebagai mediator pengambilan data. Pengambilan data memerlukan koneksi internet untuk menghubungkan node client dengan node AWS. Seperti yang digambarkan, hubungan asosiasi kedua node memiliki derajat keserbaragaman 1 untuk AWS dan banyak (*) untuk client. Kemudian data-data yang diperoleh akan ditampung dalam array list of object. Dalam hal ini, model/ class yang berperan. Terakhir, data-data tersebut ditampilkan pada view. Oleh karena itu, view memiliki dependensi ke model, sehingga dilambangkan dengan <<deploy>>.

Aplikasi memanfaatkan beberapa service dari Amazon Web Service (AWS). Beberapa service tersebut di antaranya Amazon Aurora sebagai database server, Amazon Simple Mail Service sebagai mail server, dan Amazon EC2 sebagai application server. Amazon EC2 memiliki asosiasi dengan Amazon Aurora dan Amazon Simple Mail Service, masing-masing dengan derajat keserbaragaman 1 dengan 1. Amazon Aurora merupakan Relational Database Service yang mendukung penggunaan PostgreSQL dan MySQL. Dalam aplikasi ini digunakan MySQL, yakni database jangandibuang.sql.

Aplikasi menggunakan web service berbasis Laravel. Laravel memiliki arsitektur Model-View-Controller (MVC)[10]. Namun, karena hanya digunakan sebagai web service, komponen yang digunakan hanya model dan controller. Beberapa fungsi pada controller memiliki kemampuan untuk menyimpan gambar ke dalam folder foto. Oleh karena itu, folder foto memiliki dependensi ke controller.

Aplikasi dilengkapi dengan fitur image recognition yang digunakan untuk mencari produk sejenis bagi pembeli dan memberikan rekomendasi daily waste yang akan ditambahkan bagi penyedia makanan. Fitur image recognition ini memanfaatkan Tensorflow. Dalam hal ini, file yang akan dijalankan adalah comparingImage.py. Secara garis besar, proses dimulai ketika ada makanan yang difoto. Foto tersebut kemudian dibandingkan dengan foto-foto yang tersimpan. Sebagai hasil, dikembalikan rekomendasi list makanan yang sesuai dengan makanan yang difoto oleh user.



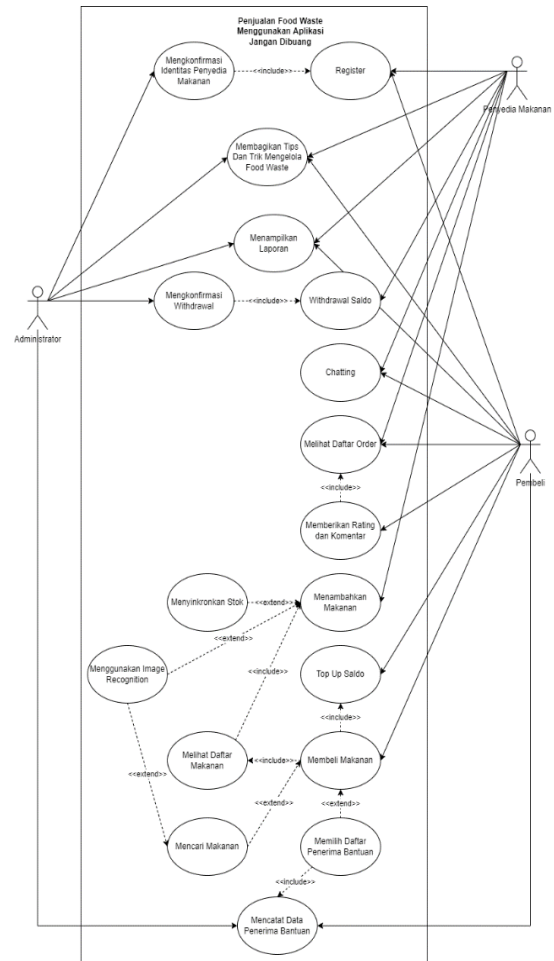
GAMBAR 2. Alur Image Recognition.

Gambar 2 menunjukkan alur image recognition. Menurut alurnya, proses dimulai dengan user memfoto makanan menggunakan smartphonenya. Kemudian gambar ini akan diterima oleh web service Laravel. Laravel akan menerima foto tersebut, menyimpannya, mengirimkan nama file, kemudian mengeksekusi file python. File python akan mengambil seluruh foto yang tersimpan dan mencari foto yang memiliki kemiripan dengan foto yang baru saja dikirimkan oleh user. Setelah proses selesai, hasilnya akan dikembalikan ke Laravel dan Laravel mengembalikan rekomendasi list makanan-makanan terkait ke front end. Oleh karena itu, file comparingImage.py memiliki dependensi terhadap folder foto.

Aplikasi juga dilengkapi dengan fitur chatting antara penyedia makanan dengan pembeli. Sesuai dengan gambar 1, fitur chatting ini memanfaatkan fitur dari Firebase, yakni Firebase Cloud Messaging. Daftar chatting dan user tersimpan di dalam firestore jangandibuang. Node Android (client) dan FCM server dihubungkan dengan asosiasi yang memiliki derajat keserbaragaman 1 (FCM server) dengan banyak (*) untuk client.

B. USE CASE DIAGRAM

Aplikasi memiliki 3 (tiga) aktor yang terlibat. 3 aktor ini meliputi administrator, penyedia makanan, dan pembeli. Setiap jenis aktor memiliki hak akses yang berbeda-beda. Berikut merupakan use case untuk menjelaskan hubungan antara aktor dan sistem.



GAMBAR 3. Alur Image Recognition.

Mula-mula, penyedia makanan dan pembeli melakukan registrasi. Khusus untuk penyedia makanan, akunnya perlu dikonfirmasi oleh administrator sehingga jelas dan dapat dipertanggungjawabkan. Oleh karena itu, pada use case diagram, dihubungkan dengan <<include>>. Setelah melalui proses konfirmasi, barulah penyedia makanan dapat menjual makanannya. Saat pertama kali menginputkan makanan, penyedia makanan harus menginputkan semua data yang diperlukan, termasuk nama, foto, dan harga makanan. Namun, apabila makanan telah diinputkan sebelumnya dan besoknya masih ada sisa makanan dengan jenis yang sama, penyedia makanan tidak perlu memasukkan data makanan dari awal, melainkan menggunakan fitur image recognition untuk mengenali makanan dan memberikan rekomendasi jenis makanan tersebut, sehingga penyedia makanan hanya perlu menginputkan jumlah makanannya saja. Hubungan ini digambarkan dengan <<extend>>.

Pembeli dan penyedia makanan dapat melakukan chatting untuk mempermudah komunikasi. Keduanya juga dapat melihat daftar order masing-masing. Khusus untuk pembeli, setelah order tersebut terselesaikan, pembeli dapat memberikan rating dan komentar. Dalam hal ini, pada use case diagram ditunjukkan dengan hubungan memberikan rating

dan komentar <<include>> melihat daftar order. Pembelian pada aplikasi menggunakan e-wallet/saldo dalam aplikasi. Oleh karena itu, pembeli dapat melakukan top up saldo, sedangkan penyedia makanan dapat melakukan withdrawal saldo. Withdrawal saldo memerlukan konfirmasi dari administrator. Oleh karena itu pada use case diagram, digambarkan mengkonfirmasi withdrawal oleh administrator <<include>> withdrawal saldo.

IV. AMAZON WEB SERVICE AURORA

Secara garis besar, seluruh penyimpanan dilakukan pada AWS Aurora. AWS Aurora merupakan basis data relasional yang kompatibel dengan MySQL dan PostgreSQL yang dibangun untuk cloud, yang menggabungkan kinerja dan ketersediaan basis data perusahaan tradisional dengan kesederhanaan dan keefektifan biaya. Amazon Aurora sepenuhnya dikelola oleh Amazon Relational Database Service (RDS), yang mengotomatiskan tugas administrasi yang memerlukan banyak waktu seperti penyediaan perangkat keras, penyiapan basis data, pengelolaan patch, dan pencadangan.

Database AWS Aurora digunakan untuk kemudahan penggunaan, yakni segala administrasi hingga backup yang dilakukan secara otomatis oleh Amazon sehingga memanfaatkan teknologi serverless dan tidak perlu melakukan administrasi awal secara manual. Alasan kedua, aplikasi ini berjalan di mobile dengan bahasa pemrograman Dart. Di satu sisi, untuk Image Recognition, perlu menggunakan bantuan library Tensorflow yang berjalan di atas bahasa pemrograman Python. Amazon memudahkan keseluruhan rangkaian system, karena menyediakan fitur Virtual Private Cloud (VPC) yang dapat menyimpan keseluruhan sistem, yakni aplikasi, web service, dan lainnya. Selain itu, Amazon juga dapat menjalankan Secure Shell (SSH). Hal ini tentu lebih efisien jika dibandingkan ketika web service dan database dihosting secara terpisah-pisah.

Selain penggunaan AWS, dalam aplikasi ini juga memanfaatkan Firebase. Firebase dalam kasus ini digunakan untuk membantu proses chatting antara pembeli dan penyedia makanan. Fitur yang dimanfaatkan adalah Firestore. Collection di Firestore menyimpan data berupa daftar chat dan pesan-pesan yang dikirimkan.

V. IMAGE RECOGNITION MENGGUNAKAN TENSORFLOW

Bagian ini menjelaskan konsep yang dimiliki oleh Tensorflow. Tak hanya itu, pada bagian ini dijabarkan pula model yang digunakan dalam pembuatan modul Image Recognition pada aplikasi. Selain itu, dijelaskan cara penggunaan Tensorflow untuk aplikasi. Berikut merupakan penjabaran dari Tensorflow.

A. KERAS

Keras merupakan library Machine Learning open source berbasis Python. Keras dikembangkan untuk membuat penerapan model pembelajaran yang mendalam secepat dan semudah mungkin untuk penelitian serta pengembangan yang dirilis berdasarkan lisensi MIT. Library jaringan saraf yang bersifat open source ini dirancang untuk memberikan eksperimen cepat dengan jaringan saraf yang dalam, dan dapat berjalan di atas CNTK, Tensorflow, dan Theano. Keras berfokus untuk menjadi modular, user friendly, dan bersifat extensible. Kerangka kerja ini tidak menangani komputasi tingkat rendah namun sebaliknya, akan menyerahkan tugas tersebut ke library lain yang disebut backend.

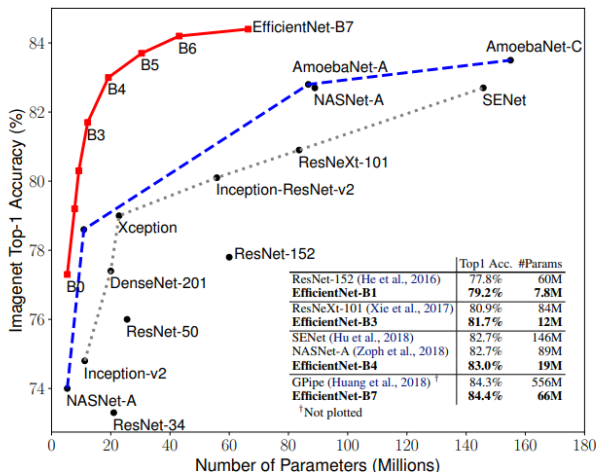
Backend adalah istilah dalam Keras yang melakukan semua perhitungan tingkat rendah. Perhitungan ini seperti produk tensor, konvolusi, dan banyak hal lain dengan bantuan library lain, seperti Tensorflow atau Theano. Jadi, backend akan melakukan perhitungan dan pengembangan model. Tensorflow adalah backend default tetapi pengguna dapat mengubahnya dalam konfigurasi.

Keras API menangani cara membuat model, mendefinisikan layer, atau mengatur beberapa model input-output. Pada level ini, Keras juga mengkompilasi model dan mengoptimalkan, serta mendukung proses pelatihan dengan fungsi fit. Keras diadopsi dan diintegrasikan ke dalam Tensorflow pada pertengahan 2017. Pengguna dapat mengaksesnya melalui modul tf.keras. Namun, library Keras masih dapat beroperasi secara terpisah dan mandiri.

B. KERAS APPLICATIONS

Keras Applications mencakup model yang berasal dari domain deep learning dan memiliki skenario bobot yang telah dilatih sebelumnya. Model aplikasi Keras memiliki penggunaannya dalam spektrum yang luas termasuk fine-tuning, prediksi dan ekstraksi fitur. Ketika model digunakan, maka secara otomatis semua bobot yang diperlukan untuk model secara otomatis diunduh dan disimpan dalam folder `./keras/models`. Ketika Keras digunakan secara instan maka sesuai format data gambar, model Keras dibangun dengan mempertimbangkan semua konfigurasi yang disebutkan di dalam file Keras yang terletak di `./keras/Keras.JSON`.

Arsitektur model Keras sepenuhnya kompatibel dengan backend CNTK, Theano, dan Tensorflow. Beberapa model yang disediakan di antaranya VGG16, VGG19, Xception, InceptionV3, ResNet50, MobileNet, InceptionResNetV2, EfficientNet, EfficientNetV2, DenseNet, NASNet-A, AmoebaNet-A, SENet, dan lain-lain. Masing-masing model memiliki ukuran dan akurasi yang berbeda-beda. Untuk penggunaan setiap application sangat mudah dan mirip untuk semua application. Pengguna cukup memuat salah satu model aplikasi Keras yang diperlukan dengan mengimpor Keras dan model yang diperlukan dari Keras. Alih-alih berupaya untuk melatih model yang telah dibuat sendiri, pengguna dapat menggunakan model yang tersedia yang telah ditentukan sebelumnya dalam aplikasi Keras.



GAMBAR 4. Ukuran Model vs Akurasi ImageNet

Gambar 4 menunjukkan grafik yang membandingkan ukuran model dari Keras Applications dengan akurasi ImageNet. Sumbu x melambangkan jumlah parameter, sedangkan sumbu y melambangkan tingkat akurasi ImageNet. Semakin kiri dari sumbu X, berarti jumlah parameter semakin sedikit dan berarti ukuran model juga semakin kecil. Pada aplikasi “Jangan Dibuang”, model yang digunakan adalah EfficientNetV2.

C. KERAS APPLICATIONS EFFICIENTNETV2

Keras Applications mencakup model yang berasal dari domain deep learning dan memiliki skenario bobot yang telah dilatih sebelumnya [12]. Model aplikasi Keras memiliki penggunaannya dalam spektrum yang luas termasuk fine-tuning, prediksi dan ekstraksi fitur. Ketika model digunakan, maka secara otomatis semua bobot yang diperlukan untuk model secara otomatis diunduh dan disimpan dalam folder `./keras/models`. Ketika Keras digunakan secara instan maka sesuai format data gambar, model Keras dibangun dengan mempertimbangkan semua konfigurasi yang disebutkan di dalam file Keras yang terletak di `./keras/Keras.JSON`.

D. PICKLE

Modul Pickle [8] merupakan library Python yang mengimplementasikan protokol biner untuk membuat serialisasi dan de-serialisasi struktur objek. “Pickling” adalah proses di mana hierarki objek Python diubah menjadi aliran byte, dan “unpickling” adalah operasi terbalik, di mana aliran byte (dari file biner atau objek seperti byte) diubah kembali menjadi hierarki objek. Pickling (dan unpickling) juga dikenal sebagai serialisasi, marshalling, atau flattening. Namun, untuk menghindari kebingungan, istilah yang digunakan di sini adalah “pickling” dan “unpickling”.

Format data yang digunakan oleh Pickle adalah format khusus Python. Hal ini memiliki keuntungan bahwa tidak ada batasan yang diberlakukan oleh standar eksternal seperti

JSON atau XDR (yang tidak dapat mewakili berbagi pointer). Namun, hal ini berarti bahwa program non-Python mungkin tidak dapat merekonstruksi objek Python menggunakan Pickle. Secara default, format data Pickle menggunakan representasi biner.

E. PENERAPAN TENSORFLOW UNTUK IMAGE RECOGNITION

Pada subbab ini dijelaskan mengenai implementasi dari hal-hal yang berkaitan dengan image recognition. Aplikasi Jangan Dibuang menyediakan fitur image recognition untuk membantu pembeli dalam mencari produk sejenis dan penyedia makanan dalam menambahkan daily waste. Image Recognition dijalankan dengan mengeksekusi file python pada web service Laravel. Image Recognition ini memanfaatkan Tensorflow dan menggunakan algoritma Cosine Distance.

Dalam proses image recognition yang dilakukan pada aplikasi “Jangan Dibuang”, mula-mula gambar diekstraksi fiturnya menjadi numpy array yang diserialisasi menggunakan Pickle. File Pickle disimpan dengan ekstensi `.pkl` yang bisa dibaca dan ditulis setiap ada gambar baru yang diinputkan ke dalam folder makanan.. Terdapat 2 (dua) file Pickle. File pertama digunakan untuk menyimpan fitur-fitur, sedangkan file kedua menyimpan nama file gambar-gambar yang nantinya dikembalikan untuk diolah kembali pada Laravel.

Alur Image Recognition dimulai ketika pengguna memfoto sebuah makanan yang ada didepannya melalui kamera yang disediakan pada aplikasi. Setelah gambar diambil, aplikasi akan *trigger* web service Laravel untuk dijalankan. Web service Laravel akan mengeksekusi file Python telah terlebih dahulu diletakkan pada folder public dengan perintah `exec("python 3.9 ./compareImage.py {$path_foto}", $output)`. Beberapa library yang dibutuhkan di antaranya pickle, keras.preprocessing, numpy, matplotlib.pyplot, scipy.spatial, os, dan sys. Agar dapat menggunakan library cosine distance, maka numpy array features harus dijadikan 1(satu) dimensi dengan menggunakan perintah `ravel`.

Mula-mula, gambar yang dikirimkan dari aplikasi akan diekstraksi fiturnya menjadi numpy array. Nilai pada numpy array ini kemudian dibandingkan dengan numpy array lainnya yang telah ditampung pada file pickle. Perhitungan perbandingan ini menggunakan metode Cosine Distance dan mengembalikan gambar-gambar dengan nilai Cosine Distance yang kecil. Hasil dari proses ini berupa nama-nama file gambar serupa yang ditampung ke dalam array output. Array output ini yang kemudian dikembalikan oleh web service ke aplikasi.

VI. HASIL UJI COBA

Bagian ini akan menjelaskan mengenai uji coba yang digunakan pada pembuatan Aplikasi Marketplace Food Waste Berbasis Android Menggunakan Flutter dan Database Amazon Aurora Dilengkapi Image Recognition Memanfaatkan Tensorflow. Uji coba ini mencakup uji coba

yang dilakukan secara black box testing dan kuesioner. Kuesioner dibagikan ke para pengguna untuk menilai kelayakan dan tanggapan aplikasi yang dibuat.

Uji coba dilakukan terhadap 7 penyedia makanan dan 20 pembeli. Berdasarkan hasil uji coba yang telah dilakukan, didapatkan 201 transaksi penjualan food waste, yang mana telah menyelamatkan 285 limbah makanan. 59 dari 201 transaksi ditujukan untuk donasi. Fungsionalitas aplikasi penyedia makanan mendapatkan nilai 79,98% untuk kriteria sangat baik. Untuk fungsionalitas aplikasi pembeli, nilai yang didapatkan adalah 83% untuk kriteria sangat baik. 7 dari 7 penyedia makanan yang mengikuti uji coba bersedia bergabung. Begitu pula dengan pembeli, 20 dari 20 pengujian bersedia menggunakan aplikasi ketika aplikasi dirilis.



GAMBAR 5. Hasil Image Recognition

Tak hanya fitur transaksi penjualan food waste secara keseluruhan, namun Gambar 5 juga menunjukkan hasil uji coba Image Recognition pada aplikasi yang dibuat. Gambar sebelah kiri merupakan foto yang diambil pada saat program dijalankan. Setelah program dijalankan, program mampu mengembalikan data makanan sejenis, yakni makanan dengan gambar sebelah kanan yang telah tersimpan pada database sebelumnya. Dari gambar di atas, dapat dibuktikan bahwa program telah berhasil mengenali makanan sejenis walaupun kedua gambar memiliki pencahayaan ataupun posisi pengambilan yang berbeda.

Dari 60 (enam puluh) percobaan yang dilakukan, 56 percobaan menunjukkan hasil yang benar. Hal ini berarti, penelitian ini memberikan akurasi senilai 93,3%. Dengan menggunakan Tensorflow, Cosine Distance, dan Keras Applications EfficientNetV2, akurasi dapat ditingkatkan. Hal ini dibuktikan dengan dapat dikenalnya suatu makanan walaupun pencahayaan dan posisi pengambilannya berbeda. Sebagai informasi, percobaan yang dilakukan telah melewati proses perbandingan kedua gambar yang memang betul-betul sama, kedua gambar dengan pencahayaan berbeda, kedua gambar dengan posisi pengambilan berbeda, dan kedua gambar yang memang sangat berbeda.

VII. KESIMPULAN

Bagian ini akan membahas mengenai kesimpulan dan saran mengenai aplikasi ini. Berikut merupakan kesimpulan yang didapatkan:

1. Aplikasi Jangan Dibuang telah menjadi sebuah aplikasi mobile yang mendukung orang-orang untuk saling berbagi makanan kepada mereka yang membutuhkan. Hal ini dibuktikan dari hasil uji coba, 59 transaksi ditujukan untuk tujuan donasi ke beberapa Panti Asuhan yang telah disediakan daftarnya oleh administrator. Selain itu, dalam pembuatannya, aplikasi mengimplementasikan framework Flutter dan database Amazon Aurora. Penggunaan framework Flutter memberikan kemudahan dalam pengaturan tampilan. Salah satunya, apabila sebelumnya pembuatan List View pada proses development *Android Native* menggunakan Java harus dibuat dalam class terpisah, dengan adanya framework Flutter, pembuatan List View lebih mudah karena komponennya telah tersedia dengan bantuan library-library yang dimiliki oleh Flutter. Untuk database Amazon Aurora, Aurora memiliki keunggulan dari sisi ketersediaan dan reliabilitasnya. Hal ini terbukti dengan adanya fitur replikasi otomatis yang telah disediakan dan waktu pemulihan kurang dari satu menit.
2. Aplikasi telah membantu mengurangi food waste yang dihasilkan penyedia makanan dan menekan beban finansial calon pembeli dengan penjualan harga miring. Dengan adanya penjualan harga miring, lebih banyak makanan yang terselamatkan. Apabila sebelumnya penyedia makanan memilih untuk membuang sisa makanan, melalui aplikasi, sisa makanan yang masih layak konsumsi dapat diperjualbelikan. Sebagai contoh, 285 produk telah terselamatkan dalam kurun waktu uji coba dengan harga hampir dari setengah harga asli. Selain itu, 83% pembeli juga puas terhadap fungsionalitas dari aplikasi, serta 20 dari 20 pembeli ingin menggunakan aplikasi ketika nantinya aplikasi dirilis. Selain itu, dari sisi Image Recognition menggunakan Keras Application EfficientNetV2 memiliki tingkat akurasi lebih tinggi dibandingkan dengan penggunaan source code dan algoritma yang sama namun memanfaatkan model VGG16. Cosine Distance juga meningkatkan efisiensi perhitungan karena hasilnya selalu ternormalisasi.

PERAN PENULIS

Monica Chandra: Tinjauan Pustaka, Rancangan dan Implementasi Sistem, Ujicoba.

Edwin Pramana: Konseptual, Penyuntingan

COPYRIGHT



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

DAFTAR PUSTAKA

- [1] Wansink, B., "Abandoned Products and Consumer Waste: How Did That Get Into the Pantry?," *Choices*, 16(316-2016-6512), 2001.
- [2] Baoli Li and Li Ping Han, "Distance Weighted Cosine Similarity Measure for Text Classification," *Henan University of Technology*, 2013, DOI: 10.1007/978-3-642-41278-3_74
- [3] Kong F and Tan J, "Dietcam: Automatic dietary assessment with mobile camera phones," *Pervasive Mob Comput* 8(1), pp 147–163, 2012.
- [4] Andrews Samraj et al., "Food Genre Classification from Food Images by Deep Neural Network with Tensorflow and Keras," 2020.
- [5] Nupur Bhawe and Dipti Belsare, "A Study on Food Recognition & Nutrition Estimation," *International Peer Reviewed Journal*, ISSN 0973-2861 Vol XVI, Issue IV, Jan-June 2022.
- [6] Anjani Kumar. 2020. Cosine Similarity & Cosine Distance.[Online]. Available: <https://medium.datadriveninvestor.com/cosine-similarity-cosine-distance-6571387f9bf8>.
- [7] Anonim. Amazon Aurora.[Online]. Available: <https://aws.amazon.com/id/rds/aurora/>.
- [8] Anonim. Pickle — Python Object Serialization [Online]. Available at : <https://docs.python.org/3/library/pickle.html>.
- [9] Anonim. Scipy Spatial Cosine Distance API Reference.[Online]. Available: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cosine.html>.
- [10] Kevin NFA. 2020. Laravel-Pengertian, Kelebihan, Kekurangan dan Cara Install Laravel. [Online]. Available: <https://medium.com/@kevinnfa0107/laravel-pengertian-kelebihan-kekurangan-dan-cara-install-laravel-224a79550a91>.
- [11] Muhammad Amirul Ihsan. 2020. Apa Itu Dart?.[Online]. Available: <https://www.kawankoding.id/apa-itu-dart/>.
- [12] Mostafa Ibrahim. 2021. Google releases EfficientNetV2 — a smaller, faster, and better EfficientNet. [Online]. Available at: <https://towardsdatascience.com/google-releases-efficientnetv2-a-smaller-faster-and-better-efficientnet673a77bdd43c#:~:text=EfficientNetV2%20uses%20the%20concept%20of%20progressive%20learning%20which,speeds%20start%20to%20suffer%20on%20high%20image%20sizes>.
- [13] Richie. 2022. Principal Component Analysis (PCA).[Online]. Available: <https://www.mobilestatistik.com/principal-component-analysis-pca/>.
- [14] Wede. 2020. Belajar Data Science : Apa yang dimaksud dengan Tensorflow dan Bagaimana Penggunaannya?.[Online]. Available: <https://dqqlab.id/belajar-data-science-pahami-tensflow>.