Implementasi *Machine Learning* pada Google Cloud Platform untuk Mengidentifikasi Tingkat Kualitas Ikan Menggunakan Aplikasi Android

Dede Chandra Wijaya, Ledya Novamizanti, Fityanul Akhyar
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
chandechan@student.telkomuniversity.ac.id, ledyaldn@telkomuniversity.ac.id, fityanul@telkomuniversity.ac.id

***Abstrak* — Seperti teknologi baru lainnya, komputasi awan menghadirkan kemudahan pengembangan aplikasi Android. Paper ini akan menjelaskan tentang penerapan komputasi awan untuk meningkatkan efisiensi penyortiran ikan berdasarkan waktu pemrosesan serta mengurangi biaya pengeluaran dengan aplikasi Android. Pemanfaatan komputasi awan pada aplikasi Android sebagai jembatan penghubung antara machine learning agar dapat saling berkomunikasi satu sama lain. Layanan yang digunakan yaitu layanan Google Cloud Platform (GCP). Di dalam GCP terdapat layanan Compute Engine untuk mengimplementasikan sistem machine learning pada *cloud* dengan menggunakan *virtual machine instance* sebagai mesin virtual dengan sistem operasi Ubuntu. Penerapan API yang digunakan yaitu menggunakan metode GET dan POST. Kedua metode tersebut diuji dengan menggunakan aplikasi postman. Metode GET digunakan terlebih dahulu untuk memastikan ketersediaan layanan. Kemudian pengujian menggunakan metode POST yaitu untuk memastikan *machine learning* yang sudah dipasang pada *virtual machine* dapat berjalan dengan baik dan dapat memberikan keluaran berupa JSON. Pengujian menggunakan 20 gambar ikan cakalang dari 2 kelas yang berbeda menunjukkan waktu pemrosesan *machine learning* di *cloud* sebesar 1,58 dan 1,34 detik. Setelah pengujian dilakukan, seluruh sistem pengolahan kualitas ikan telah berhasil diimplementasikan dengan baik. *Cloud computing* berjalan menggunakan server virtual yang dikelola melalui perintah systemctl.**

***Kata kunci*— Android, GCP, Komputasi Awan, Machine Learning**

I. PENDAHULUAN

Perkembangan teknologi yang sangat cepat memberikan banyak dampak baik dalam bisnis. Khususnya pada bidang komputasi awan, untuk pengembangan aplikasi Android, web dan yang lainnya dalam bisnis. Permasalahan saat ini yang ada di industri adalah pengecekan kualitas ikan pada pusat sortasi masih dilakukan secara manual. Ini menyebabkan pengeluaran biaya yang cukup mahal dengan membayar pekerja untuk melakukan pengecekan kualitas ikan secara manual. Untungnya, komputasi awan memiliki sumber daya untuk penyimpanan data, penyediaan infrastruktur, dan penyediaan platform sebagai layanan. Banyak sekali layanan komputasi awan yang menyediakan sumber daya tersebut seperti IBM Cloud, Google Cloud Platform (GCP), Amazon Web Services (AWS), Microsoft Azure, dan masih banyak layanan *cloud* lainnya. Penulis menggunakan layanan GCP dalam perancangan sistem ini.

Dalam penerapannya, GCP menyediakan layanan infrastruktur seperti Compute Engine yang dapat menjalankan *Virtual Machine* (VM) yang biasanya digunakan untuk sistem operasi Windows dan Linux, lalu ada juga App Engine yang merupakan platform sebagai layanan untuk menghosting dan membuat mesin virtual, khususnya aplikasi web, di pusat data yang dioperasikan oleh Google [1]. Kedua layanan tersebut sangat popular digunakan oleh para pengembang seperti pengimplementasian *machine learning*, aplikasi berbasis data, aplikasi Andoid atau iOS dan masih banyak yang lainnya [2]. Dalam paper ini, implementasi *machine learning* untuk mengidentifikasi kualitas ikan dan juga sebagai jembatan antara aplikasi Andoird agar dapat saling berkomunikasi melalui layanan *cloud*.

Model *machine learning* yang dipasangkan pada VM yang ada pada GCP adalah *You Only Look Once Version 7* (YOLOv7) dan EfficientnetV2S. Penelitian sebelumnya telah dilakukan menggunakan model YOLO tetapi menggunakan versi model lama. Prinsip kerja YOLO melibatkan pemisahan gambar masukan menjadi kotak-kotak berukuran S, di mana S memiliki nilai tetap yaitu 7, pada resolusi gambar masukan 488x488 piksel [3]. Sebagai pendukung, dengan memiliki total 167 lapisan pemrosesan, sistem deteksi dalam YOLOv4 membutuhkan waktu yang lebih panjang untuk melakukan proses deteksi jika dibandingkan dengan YOLOv3 [4]. Lalu pada model YOLOv5 digunakan untuk mengenali objek secara umum seperti manusia, kendaraan, hewan, dan objek lainnya dalam sebuah gambar [5]. Penggunaan model YOLOv7 tentu membawa peningkatan yang signifikan dalam aspek akurasi pengenalan objek dan efisiensi.

Aplikasi Android untuk mengidentifikasikan kualitas ikan menggunakan *deep learning* yang diusulkan menggunakan layanan GCP untuk mengimplementasikan *machine learning* di dalam virtual server yang dibuat atau dengan Compute Engine. Kontribusi penulis yaitu mengimplementasikan layanan *cloud* dengan GCP untuk mendeteksi kualitas ikan menggunakan *deep learning*. Penulis melakukan pengujian waktu pemrosesan dengan aplikasi postman dan kesiapan API yang sudah dirancang. Proses pembuatan layanan *cloud* dari awal hingga akhir akan dijelaskan juga pada pembahasan ini.

II. KAJIAN TEORI

A. *Application Programming Interface*

Application Programming Interface (API), merupakan aturan yang telah ditetapkan untuk memungkinkan komunikasi antara berbagai aplikasi [6]. API bisa diartikan juga sebagai jembatan komunikasi dan digunakan untuk pertukaran data sehingga aplikasi dapat berinteraksi satu sama lain. *Representational State Transfer* (REST) merupakan jenis API yang umum digunakan untuk berkomunikasi melalui permintaan HTTP dengan metode GET, POST, PUT, dan DELETE sesuai dengan kebutuhan [7].

B. Flask

Bahasa pemrograman yang digunakan dan juga untuk membuat API adalah dengan python. Flask merupakan modul yang ada pada python yang dapat menghasilkan aplkasi juga merupakan kerangka kerja yang dapat mengambil kendali serta mengontrol aplikasi [8]. Flask juga menjadi salah satu alat server yang terkemuka dalam ekosistem python [9]. Metode yang digunakan ada 2 yaitu POST dan GET pada perancangan sistem ini.

C. Firewall

Firewall dalam *cloud* yaitu sebagain keamanan jaringan untuk menyaring lalu lintas jaringan yang berpotensi terkena serangan dari luar yang berbahaya. Firewall bisa disebut juga sebagai penghalang dari seluruh lalu lintas masuk dan keluar [10]. Implentasi sistem pada GCP menerapkan aturan firewall yang digunakan dalam perancangan sistem. Firewall *rules* memiliki fungsi sebagai pengamanan untuk mengizinkan atau menolak koneksi ke VM dalam jaringan VPC [11].

D. *Secure Shell*

Secure Shell (SSH) adalah protokol untuk melakukan komunikasi melalui jaringan internet yang dapat menukar data dengan aman [12]. Untuk masuk kedalam jaringan ini, pengguna harus memiliki akses terlebih dahulu. Akses yang dimaksud adalah kunci publik dan kunci pribadi yang dimiliki oleh pengguna.

E. *Virtual Server*

Virtual server mirip dengan server fisik, namun virtual server tidak ada secara fisik melainkan sebagai perangkat keras terpisah dan ada sebagai entitas virtual yang dapat diakses dari jarak jauh. Dalam penggunannya *virtual server* digunakan untuk pemasangan model *machine learning* pada sistem komputasi awan.

F. *User*

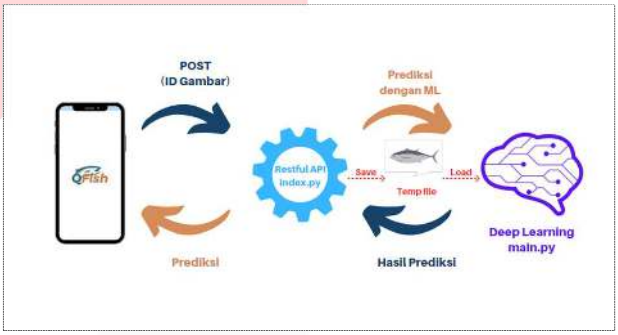
User merupakan pengguna dalam layanan komputasi awan yang dapat memanfaatkan layanan *cloud*. Layanan yang dimanfaatkan untuk mengimplementasikan sistem yaitu untuk memenuhi kebutuhan pengguna. Pengguna memiliki hak akses dan kendali atas layanan komputasi awan yang digunakan. Seperti pada pembahasan di dokumen ini yaitu menggunakan layanan GCP.

III. METODE

A. Rancangan Secara Umum

Layanan komputasi awan yang digunakan untuk mengimplementasikan model *machine learning* pada aplikasi Android adalah dengan GCP. GCP merupakan layanan yang disediakan oleh Google. Berbagai macam layanan yang ditawarkan oleh GCP yang disesuaikan dengan kebutuhan pengguna seperti penyimpanan data, alat pengembang, pembuatan jaringan dan masih banyak yang lainnya [13]. Pengguna bisa memilih dan menggunakannya, seperti pada pembahasan di dokumen ini, penulis menggunakan layanan Compute Engine.

Compute Engine yang tawarkan oleh layanan GCP menyediakan mesin virtual yang dapat digunakan untuk menjalankan aplikasi. Pada Compute Engine pengguna dapat mengatur dan memilih jenis sistem operasi yang akan digunakan seperti sistem operasi Ubuntu, Debian, dan yang lainnya. Pengguna juga dapat mengatur spesifikasi mesin yang diinginkan, seperti jumlah CPU, memori, dan kapasitas penyimpanan mesin virtual yang diperlukan.

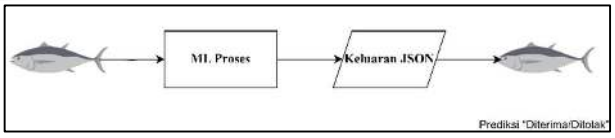


Gambar 1. Cara Kerja Sistem secara Umum

Gambar 1 merupakan cara kerja sistem keseluruhan secara umum. Aplikasi Andorid mengirimkan gambar ke API yang telah dibuat menggunakan GCP. Metode yang digunakan untuk proses mengirimkan gambar ke API adalah dengan metode POST. Kemudian gambar akan disimpan sementara kedalam mesin virtual yang sudah dibuat dan akan langsung diproses oleh deep learning. Ketika deep learning sudah selesai melakukan prosesnya, hasil prediksi berupa JSON akan muncul dan hasil tersebut akan digunakan oleh aplikasi Andoird.

B. Cara Kerja *Cloud*

Sebelum masuk kedalam proses implementasi, tentu ada cara kerja sistem *cloud*. Berikut merupakan diagram yang menggambarkan alur proses dari awal hingga selesai.

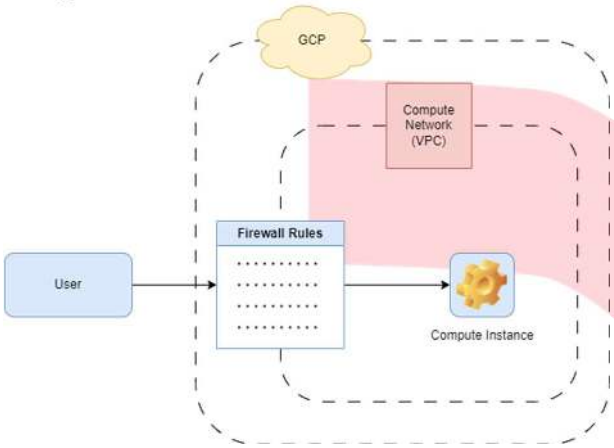


Gambar 2. Cara Kerja Cloud

Gambar 2 menjelaskan tentang langkah-langkah dalam proses kerja *cloud*. Proses dimulai dengan pengguna memasukkan gambar ikan sebagai data masukan. Ketika pengguna sudah selesai memasukan gambar ikan, gambar tersebut akan dijadikan sebagai data input untuk proses machine learning. Dalam *cloud*, terdapat algoritma dan

model *machine learning* yang telah diimplementasikan. Gambar ikan akan diproses menggunakan algoritma dan model *machine learning*. Setelah proses tersebut selesai, *cloud* akan mengeluarkan keluaran dalam format JSON. Keluaran tersebut berisi informasi atau prediksi angka yang dapat dibaca oleh manusia. Hasil prediksi angka ini akan digunakan oleh aplikasi Andorid yang telah terhubung ke *cloud*. Aplikasi Android akan menggunakan angka prediksi ini untuk menentukan tingkat kualitas ikan. Dengan demikian, melalui proses tersebut *machine learning* dan aplikasi Andoird dapat saling berkomunikasi satu sama lain.

C. Lapisan *Cloud*



Gambar 3. Lapisan pada *Cloud* (sumber : www.pulumi.com)

Aplikasi deteksi kualitas ikan menggunakan GCP sebagai layanan untuk implementasi dan *deploy* API. Pada Gambar 3 dapat dilihat bahwa jika pengguna ingin masuk ke dalam Compute Instance harus melewati firewall *rules* terlebih dahulu. Firewall dan Compute Instance berada didalam jaringan yang sama yaitu didalam Compute Network yang dapat digunakan secara *private*.

Proses perancangan sistem *cloud* terbagi menjadi dua bagian utama yaitu pembuatan VPC *network* dan Compute Engine yang akan dibahas penulis pada point selanjutnya. Untuk seorang pengguna dapat terkoneksi dengan Compute Instances, pengguna harus terkoneksi dengan GCP dan memenuhi firewall *rules* pada VPC *network*.

D. *Virtual Private Cloud Network*

Virtual Pivate Cloud (VPC) network merupakan layanan yang disediakan oleh Google. Didalam VPC *network* pengguna dapat mengatur, membuat, dan mengendalikan jaringan yang berjalan pada GCP. Seperti pada Gambar 4 penulis membuat subnets terlebih dahulu agar dapat terhubung dengan wilayah yang ditentukan.

Gambar 4. Subnets

Pada Gambar 4 dapat dilihat bahwa nama subnets yang dibuat yaitu qfish-jkt untuk region menggunakan asia-southeast2 karena itu merupakan region yang paling dekat. Selanjutnya untuk tipe IP yaitu IPv4 karena dengan IPv4 kebutuhan sudah terpenuhi. Internal IP yang digunakan dan sudah dibuat juga yaitu "10.70.69.0/24" dan gateaway yang didapatkan adalah "10.70.69.1" sebagai gerbang penghubung.

Gambar 5. Eksternal dan Internal Alamat IP

Didalam VPC *network* terdapat 2 jenis alamat IP yang digunakan yaitu internal dan eksternal. IP internal digunakan di dalam VPC *network* sedangkan IP eksternal digunakan dari luar jaringan, seperti internet. Gambar 5 merupakan alamat IP yang telah dibuat untuk mengakses server atau mesin virtual dari jarak jauh. Untuk mengakses mesin virtual penulis mememakai IP eksternal yaitu "34.101.154.74".

E. Firewall

Gambar 6. *Allow* Flask

Pada Gambar 6, terdapat tiga firewall yang dibuat untuk diimplementasikan pada sistem. Firewall tersebut bernama "allow-flask", "qfish-icmp", dan "qfish-ssh", dari ketiga firewall tersebut memakai jenis yang sama yaitu ingress dan untuk protokol serta port yang dibuka adalah tcp:5000, icmp, dan tcp:22.

F. Compute Engine

Setelah melalui beberapa proses, selanjutnya pembuatan dan implementasi layanan Compute Engine. Di dalam layanan tersebut terdapat mesin virtual yang dapat dioperasiakan menggunakan SSH. Mesin virtual yang disediakan oleh GCP, dapat digunakan untuk menjalankan aplikasi serta pemilihan sistem operasi.

Tabel 1. Spesifikasi Mesin Virtual

Specifications	Detail
Machine Type	E2-medium(2 vCPU, 4 GB memory)
Operating System	Ubuntu
Ubuntu Version	20.04 LTS
Size (GB)	30

Tabel 1 merupakan spesfikasi dari mesin virtual yang telah dibuat. Spesifikasi yang digunakan yaitu menggunakan tipe mesin E2-medium dengan sistem operasi Ubuntu versi 20.04 LTS dan ukurannya 30 GB. Spesifikasi tersebut, sudah memenuhi kebutuhan untuk menerapkan *machine learning* didalam mesin virtual.

IV. HASIL DAN PEMBAHASAN

Bagian ini merupakan hasil yang diperoleh dari pengujian yang dilakukan. Proses pengujian yang dilakukan juga akan dibahas pada bagian ini. Pengujian yang dilakukan mencakup pengujian metode GET dan POST serta pengujian waktu pemrosesan.

A. Pengujian Metode GET dan POST

Pengujian pada server virtual dilakukan dengan menggunakan aplikasi postman untuk mencoba apakah API dan seluruh *endpoint* yang telah dibuat bekerja dengan baik.

Gambar 7. Uji Kesehatan API

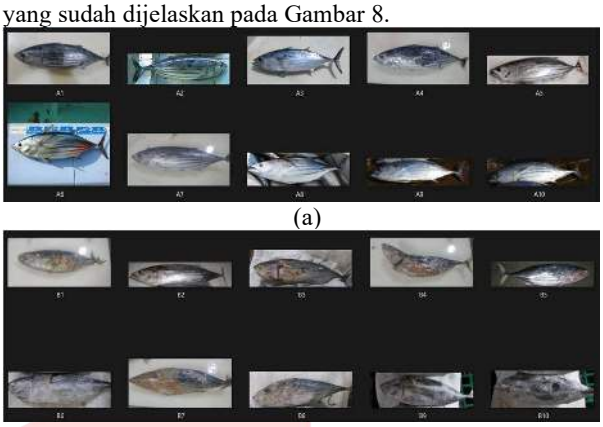
Gambar 7 merupakan pengujian dengan metode GET yaitu untuk memeriksa status kesehatan dan ketersediaan layanan API. Dengan adanya mekanisme *health check*, sistem dapat secara periodik diperiksa apakah API dapat berjalan dengan baik dan siap menerima *request* dari *user*. Dengan demikian virtual server dan API yang telah dibuat berjalan dengan baik.

Gambar 8. Metode POST

Gambar 8 merupakan pengujian dengan menggunakan metode POST dimana dalam pengujian ini, IP eksternal yang digunakan adalah 34.101.153.74 dengan menggunakan port 5000. Gambar dengan nama c17.jpeg dikirim ke *endpoint* '/classify' untuk diproses oleh machine learning. Hasil keluaran dalam bentuk JSON menunjukkan nilai "result" sebesar 0.010210683569312096, yang menandakan bahwa ikan tersebut "DITERIMA". Semakin mendekati nilai 0, semakin tinggi tingkat kualitas ikan.

B. Pengujian Waktu Pemrosesan

Gambar 9 merupakan data yang digunakan dalam pengujian kedua untuk menguji kecepatan waktu pemrosesan. Pengujian ini melibatkan total 20 gambar dalam percobaan tersebut. Aplikasi yang digunakan untuk pengujian ini sama dengan yang ditunjukkan pada Gambar 8, yaitu menggunakan aplikasi postman. Tujuan dari pengujian ini adalah untuk mengukur kecepatan waktu proses dalam pemrosesan gambar menggunakan metode yang sama seperti yang sudah dijelaskan pada Gambar 8.



(a)

(b)

Gambar 9. Dataset Ikan (a) Kelas A (b) Kelas B

Pengujian kecepatan dilakukan dengan menggunakan 20 gambar ikan. Kelas A mengacu pada ikan dengan kualitas tinggi, sedangkan kelas B mengacu pada ikan dengan kualitas rendah. Sebelum pembuatan *cloud*, penguji memiliki target waktu pemrosesan pada *cloud*. Target yang penguji harapkan untuk kecepatan waktu pemrosesan pada *cloud* adalah dibawah 10 detik. Berikut merupakan tabel pengujian kecepatan.

Tabel 2. Pengujian Waktu Pemrosesan Kelas A

No	Gambar	Status	Waktu
1	A1	Done	1,44
2	A2	Done	1,52
3	A3	Done	2,99
4	A4	Done	1,58
5	A5	Done	1,34
6	A6	Done	2,18
7	A7	Done	1,57
8	A8	Done	1,55
9	A9	Done	1,10
10	A10	Done	1,02
Rata-rata			**1,58**

Tabel 3. Pengujian Waktu Pemrosesan Kelas B

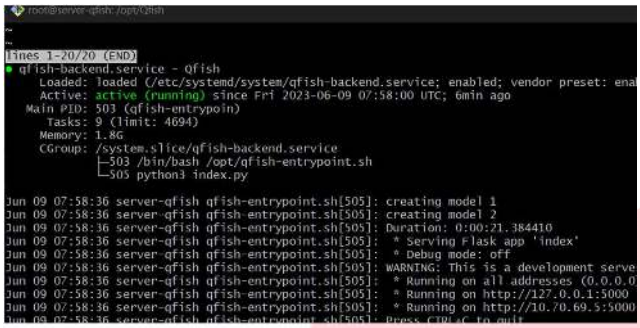
No	Gambar	Status	Waktu
1	B1	Done	1,58
2	B2	Done	1,13
3	B3	Done	1,36
4	B4	Done	1,57
5	B5	Done	1,06
6	B6	Done	1,38
7	B7	Done	1,58
8	B8	Done	1,21
9	B9	Done	1,35
10	B10	Done	1,36
Rata-rata			**1,34**

Tabel 2 dan Tabel 3 merupakan hasil pengujian kecepatan untuk gambar kelas A dan B. Rata-rata kecepatan pemrosesan gambar yaitu 1,58 dan 1,34 detik. Hasil menunjukan bahwa

sudah mencapai target yang ditentukan dari awal yaitu kurang dari 10 detik.

C. Server Virtula Berjalan dibalik Layar

Hasil pada *cloud computing* adalah server virtual, server virtual sudah bisa berjalan dibelakang layar menggunakan perintah systemctl. Utilitas yang telah tersedia dalam sistem operasi Ubuntu adalah systemctl. Fungsinya untuk mengelola dan mengendalikan unit.



Gambar 10. Systemctl *Running on Background*

Berdasarkan Gambar 10, server virtual berjalan dibalik layar menggunakan perintah systemctl. Jadi server sudah bisa dimonitor sesuai dengan kebutuhan pengguna. Tampilan pada Gambar 10, merupakan terminal mesin virtual yang sudah dipasangkan *machine learning* pada GCP. Sistem yang berjalan secara otomatis memungkinkan pengguna untuk mengirimkan gambar melalui aplikasi Android. Dalam waktu singkat, machine learning akan memproses gambar tersebut untuk mengidentifikasi jenis ikan dan menghasilkan prediksi secara instan.

V. KESIMPULAN

Implementasi layanan GCP pada aplikasi Android untuk mendeteksi kualitas ikan menggunakan deep learning telah berhasil dilakukan dan memiliki fungsi sesuai dengan spesifikasi yang telah dirancang. Pengujian API dilakukan melalui metode GET untuk memeriksa status kesehatan dan ketersediaan layanan API. Dengan adanya mekanisme *health check*, sistem secara periodik diperiksa apakah API dapat berjalan dengan baik dan siap menerima permintaan dari pengguna. Virtual server dan API yang telah dibuat dapat berjalan dengan baik. Selain itu, metode POST dengan menggunakan *endpoint* /classify untuk mengirimkan gambar telah berfungsi dengan baik dan dapat digunakan oleh aplikasi Android.

Hasil pengujian menggunakan 20 gambar ikan cakalang kelas A dan B menunjukkan waktu pemrosesan machine learning di cloud sebesar 1,58 dan 1,34 detik. Hal ini disebabkan oleh adanya kemiripan data antara data pelatihan machine learning dan data pengujian. Sistem yang diimplementasikan pada *cloud computing* telah berjalan dengan baik. Seluruh proses, mulai dari mengunggah gambar dari aplikasi Android, pemrosesan gambar di *cloud*, hingga penggunaan model *machine learning* yang telah dipasang dan menghasilkan keluaran JSON, telah berhasil dilakukan. Kesimpulannya, seluruh sistem pengolahan kualitas ikan telah berhasil diimplementasikan dengan baik. *Cloud* berjalan menggunakan server virtual yang dikelola melalui perintah systemctl.

Saran untuk pengembangan selanjutnya adalah skalabilitas perlu dipertimbangkan untuk menangani peningkatan jumlah pengguna dan permintaan. Lalu menggunakan penyimpanan data ketika gambar dikirm ke *cloud*, untuk saat ini gambar langsung dihapus dan tidak disimpan. Gambar ikan tersebut dapat digunakan untuk meningkatkan model machine learning. Selain itu mengoptimalkan biaya, lakukan analisis atau perbandingan biaya penggunaan layanan *cloud.*

REFERENSI

[1] Dayananda Sagar College of Engineering, Institute of Electrical and Electronics Engineers. Bangalore Section, and Institute of Electrical and Electronics Engineers, *2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020) : conference proceedings : 5-7 March, 2020*. 2020.

[2] D. Desmulyati and M. R. Perdana Putra, "Load Balance Design of Google Cloud Compute Engine VPS with Round Robin Method in PT. Lintas Data Indonesia," *SinkrOn*, vol. 3, no. 2, p. 147, Mar. 2019, doi: 10.33395/sinkron.v3i2.10064.

[3] H. M. Lathifah, L. Novamizanti, and S. Rizal, "Fast and Accurate Fish Classification from Underwater Video using You only Look Once," in IOP Conference Series: Materials Science and Engineering, IOP Publishing Ltd, Dec. 2020. doi: 10.1088/1757- 899X/982/1/012003.

[4] F. Akhyar, L. Novamizanti, T. Putra, E. N. Furqon, M. C. Chang & C. Y. Lin, Lightning YOLOv4 for a surface defect detection system for sawn lumber, In 2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 184-189, 2022.

[5] F. Akhyar, L. Novamizanti & T. Riantiarni, "Sistem Inspeksi Cacat pada Permukaan Kayu menggunakan Model Deteksi Obyek YOLOv5", ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, vol. 10, no. 4, pp. 990, 2022.

[7] IBM, "What is an API?" https://www.ibm.com/topics/api (accessed Jul. 07, 2023).

[8] IBM, "What is a REST API?" https://www.ibm.com/topics/rest-apis (accessed Jul. 07, 2023).

[9] M. Grinberg, *Flask web development : developing web applications with Python*. 2018.

[10] M. S. Bonney *et al.*, "Development of a digital twin operational platform using Python Flask," *Data-Centric Engineering*, vol. 3, no. 1, Jan. 2022, doi: 10.1017/dce.2022.1.

[11] H. Shah, A. Ud Din, and A. Khan, "Enhancing the Quality of Service of Cloud Computing in Big Data using Virtual Private Network and Firewall in Dense Mode," 2020. [Online]. Available: www.ijacsa.thesai.org

[12] Google Cloud, "VPC firewall rules." https://cloud.google.com/firewall/docs/firewalls (accessed Jul. 07, 2023).

[13] R. Y. Pratama, M. Orisa, and F. X. Ariwibisono, "Aplikasi Monitoring dan Controlling Server Menggunakan Protocol ICMP (Internet Control Message Protocol) dan SSH (Secure Shell) Berbasis Website," 2020.

[14] B. Gupta, P. Mittal, and T. Mufti, "A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services," European Alliance for Innovation n.o., Mar. 2021. doi: 10.4108/eai.27-2-2020.2303255.