



PENERAPAN SERVERLESS COMPUTING DALAM MENDETEKSI PENYAKIT MULUT DENGAN METODE CNN

R. Danantyo Andaru Kusumo¹, Sri Eniyati²

¹Teknik Informatika – Unisbank Semarang, danantyo1908@gmail.com

²Teknik Informatika – Unisbank Semarang, eniyati03@gmail.com

Jalan Tri Lomba Juang Semarang, Telp (024) 8451976

ARTICLE INFO

Article history:

Received 28 Januari 2023

Received in revised form 2 Februari 2023

Accepted 4 Februari 2023

Available online Desember 2023

ABSTRACT

In addressing the problem of low public awareness in dental and oral health checks, machine learning could offered a practical solution to this problem. The solution that could been provided was in the form of an application that could classify and detect early images for oral and dental diseases. There were many types of approaches that could been used to carried out classification and detection, but the most widely researched and applied was the convolutional neural network method which was one of several types of methods from deep learning algorithms. Serverless computing could been done in an isolated environment, this could been utilized to test several hypotheses in parallel which was advantageous for developers, but challenges remain with serverless computing such as timeframe and memory capacity that needs have been addressed, but for now serverless computing seems have been a viable alternative to the training process model.

Keywords: Serverless, Cloud Computing, CNN, Detection

1. Introduction

Mulut merupakan salah satu organ terpenting pada tubuh manusia yang terdiri dari berbagai bagian seperti bibir, gigi, lidah, gusi. Mulut dan bagiannya mempunyai fungsi utama untuk mengunyah, mengecap, menjilat, menggigit dan sebagainya. Dilihat dari fungsi dan manfaatnya kesehatan mulut haruslah selalu dijaga dari berbagai macam penyakit, sebab mulut seringkali menjadi faktor utama masuknya penyakit kedalam tubuh. Ditemukan bahwa 98,7% masyarakat membutuhkan perawatan gigi, namun baru sebagian yang sudah sadar untuk melakukan perawatan. Sekitar 66,2% diantaranya disebabkan oleh gigi berlubang namun jumlah masyarakat yang datang baru sekitar 35% saja, 65% masyarakat memilih untuk merawat dan mengobatinya sendiri [1].

Mempunyai nama *deep* sebab dalam cara kerjanya menggunakan beberapa lapisan dan menyebarkan rincian lebih lanjut daro fitur data untuk dipelajari, secara total strukturnya memiliki tiga bagian yaitu *input, output*, dan *hidden layers*. Hasil output akan tergantung kepada beberapa parameter dalam arsitektur, seperti kombinasi jaringan saraf di setiap lapisan, bobot, bias, dan fungsi aktivasi. Semua langkah proses pembelajaran terjadi dalam dua tahap utama, yaitu menemukan model dengan mengubah data menjadi model linier atau nonlinier dan kemudian melakukan peningkatan model [2]. Dalam kaitannya dengan penelitian yang dilakukan deep learning mempunyai keterkaitan yang erat dengan machine learning, sebab merupakan cabang dari artificial intelligence yang secara sistematis menerapkan algoritma untuk menyintesis hubungan yang mendasari antara data dan informasi [3].

Serverless computing adalah model eksekusi kode dimana proses berada pada sisi server yang dijalankan dalam wadah komputasi singkat tanpa status, yang dipicu oleh peristiwa,

Received Januari 28 2023; Revised February 2, 2023; Accepted February 4, 2023

serta sepenuhnya dikelola oleh pihak ketiga yaitu penyedia cloud [4]. Penyedia layanan cloud memisahkan rincian infrastruktur perangkat keras dan lunak supaya memberi kemudahan kepada pengguna untuk menyesuaikan beberapa opsi tergantung kebutuhan, ini menjadi keuntungan tersendiri bagi pengembang karena mereka dapat memfokuskan lebih banyak waktu untuk fokus pada masalah utama dalam aplikasinya daripada memecahkan masalah-masalah yang berhubungan dengan *server*.

Selain itu, serverless computing dapat dilakukan dalam lingkup yang terisolasi, hal ini dapat dimanfaatkan untuk menguji beberapa hipotesis secara paralel yang menguntungkan bagi pengembang, namun tantangan tetap ada pada serverless computing seperti jangka waktu dan kapasitas memori yang perlu ditangani, tetapi untuk saat ini serverless computing tampaknya menjadi alternatif yang layak untuk proses pelatihan model yang sudah ada saat ini. Penelitian oleh [5] menghasilkan akurasi sebesar 99.64% dari total 87,000 dataset penyakit bunga dengan menggunakan metode CNN berbasis serverless web. Penelitian oleh [6] menggunakan CNN dan model MobileNet untuk membandingkan client-side dengan server-side, hasilnya lebih unggul server-side dengan didapatkan learning rate sebesar 0.0001 dalam pengaturan 20 epoch. Penelitian oleh [7] mengimplementasikan CNN dan serverless computing dalam mendeteksi masker wajah yang menghasilkan akurasi sebesar 89%. Penelitian [8] membangun sistem absensi wajah berbasis browser dengan CNN yang mencocokkan gambar dengan titik acuan pada wajah partisipan. Penelitian oleh [9] menerapkan machine learning menggunakan metode CNN dengan serverless computing pada layanan AWS lambda.

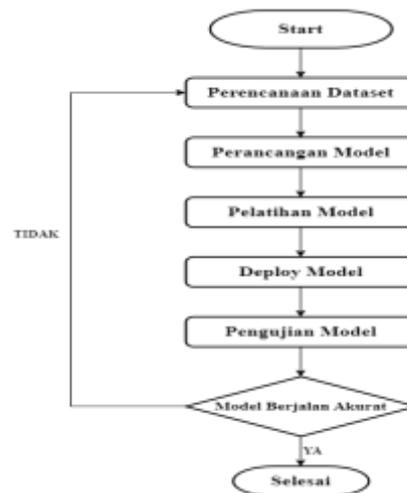
2. Metode Penelitian

Proses perancangan sistem nantinya menggunakan bahasa program python dengan environment program docker kemudian untuk sistem serverless computing menggunakan layanan dari pihak ketiga yaitu Google Cloud Platform. Secara sederhana cloud computing atau komputasi awan hanyalah cara untuk menggambarkan bagaimana organisasi dapat mengambil beberapa atau semua infrastruktur dan operasi teknologi informasi yang ada dan menyerahkannya kepada orang lain. Secara teknis dijabarkan bahwa layanan teknologi informasi tradisional sekarang divirtualisasikan dan disediakan melalui arsitektur referensi modular yang dibuat oleh penyedia atau vendor daripada dilaksanakan oleh pengguna itu sendiri [10].

Convolutional neural network atau CNN merupakan tipe dari model deep learning untuk pemrosesan data citra yang mempunyai pola grid dan dirancang secara otomatis dan adaptif untuk mempelajari hirarki fitur spasial dari level terendah hingga tertinggi suatu pola. CNN adalah konstruksi matematis yang terdiri dari tiga jenis lapisan yaitu convolution, pooling, dan fully connected layers. Dua lapisan awal, convolution dan pooling bekerja dengan cara mengekstrak fitur dari suatu citra dan lapisan terakhir, fully connected layers melakukan pemetaan fitur yang telah diekstrak menjadi hasil akhir keluaran, yang dinamakan dengan klasifikasi [11]. Secara sederhana operasi CNN bekerja dengan menggabungkan dua array dengan ukuran yang berbeda atau sama dan kemudian menghasilkan array ketiga sebagai hasil operasi atau yang disebut sebagai citra hasil. Operasi konvolusi mempunyai dua fungsi $f(x)$ dan $f(g)$ yang dapat didefinisikan sebagai berikut:

$$h(x) = f(x) * g(x) = \int_{-x}^x f(a).g(x - a)da$$

Secara garis besar proses pelaksanaan penelitian yang dilakukan dapat digambarkan pada gambar diagram flowchart berikut:



Gambar 1. Diagram *Flowchart* Penelitian

Menurut gambar diagram flowchart diatas dapat disimpulkan bahwa proses pelaksanaan penelitian memiliki enam tahapan, penjelasan tahapan-tahapan tersebut adalah sebagai berikut:

a. Perencanaan *Dataset*

Tahapan pertama yang dilakukan setelah menetapkan tujuan penelitian yang akan dicapai adalah mendapatkan data spesifik untuk proses penelitian. Sesudah mengumpulkan dataset tahapan setelahnya adalah melakukan *praprocessing* pada dataset, yaitu memisahkan data citra kedalam kategorinya masing-masing dalam hal ini adalah memberi nama penyakit mulut dan gigi, tahap selanjutnya adalah melihat ukuran pada data citra lalu mengubahnya menjadi ukuran yang seragam agar dapat memudahkan proses penelitian. Setelah melakukan *praprocessing* tahapan selanjutnya adalah membagi data dalam tiga folder dengan perbandingan data 60:20:20. Folder pertama untuk kebutuhan *training*, folder kedua untuk kebutuhan *validtaion*, dan folder terakhir untuk kebutuhan *evaluation* atau *testing*.

b. Perancangan Model

Sebelum melakukan proses perancangan model, peneliti harus mempertimbangkan salah satu algoritma *machine learning* yang digunakan dalam proses penelitian. Dalam penelitian mendeteksi penyakit mulut dan gigi, peneliti memutuskan untuk menggunakan metode *CNN*. *CNN* merupakan metode yang cukup mumpuni dalam menjawab tantangan pada bidang *computer vision* dan telah banyak penelitian yang menggunakan *CNN* sebagai metode utama sebab cara kerja pembelajaran oleh *CNN* terbilang cukup ringkas, sebab hanya terdapat dua tahapan yaitu *feature extraction* dan *classification*.

c. Pelatihan Model

Tahap ini merupakan implementasi dari aplikasi yang dibuat dengan menggunakan pendekatan *CNN* yang dirancang dan dibangun pada tahap sebelumnya. Pelatihan ini dapat diulang dan berbagai variabel dapat diubah untuk mendapatkan persentase hasil pelatihan yang maksimal.

d. *Deploy* Model

Pada tahapan ini dapat dilakukan setelah proses pelatihan model dirasa sudah mendapatkan hasil yang maksimal, sebab nantinya model yang telah dihasilkan akan di-*deploy* atau disebarkekan kedalam sistem *cloud* dari pihak ketiga yaitu *GCP*. Model yang telah jadi akan diubah menjadi website sederhana dengan menggunakan *framework flask* yang memungkinkan semua pengguna untuk mengakses secara cepat dan mudah pada *browser*.

e. Pengujian Model

Tahapan terakhir dalam metode penelitian adalah proses pengujian model, pada tahap ini website yang telah di-*deploy* kedalam sistem *cloud* dicoba dengan user mengakses link website melalui *browser*, apabila sistem berhasil dibuat dengan baik maka tampilan sebuah website akan muncul.



Gambar 2. Digram *Flowchart* Uji Model

Saat website bisa diakses maka akan muncul segmen upload file gambar penyakit dan terdapat pilihan tombol untuk memilih file dan upload file. Pada bagian memilih file, user akan diberikan pilihan untuk memilih sebuah gambar yang ada pada folder komputer user. Setelah user menentukan gambar yang dipilih maka proses yang dilakukan berikutnya adalah mengklik tombol upload file.

Proses upload file jika berhasil akan memunculkan segmen halaman website berikutnya yaitu jalankan deteksi yang menampilkan gambar file yang telah berhasil di-upload dan pilihan tombol deteksi, klik tombol deteksi untuk melakukan proses deteksi. Pada proses deteksi, sistem akan berjalan dengan otomatis setelah user meng-upload gambar dan menekan tombol deteksi, pada proses inilah yang dinamakan dengan serverless computing sebab server cloud run hanya bekerja saat ada pemicunya, dalam hal ini pemicunya adalah proses upload file gambar, dan juga pengenaan biaya dalam pengoperasian server yang ditagih hanya saat server bekerja saja.

Hasil deteksi akan muncul setelah beberapa saat server cloud run bekerja dengan mencocokkan file gambar yang telah di-upload pada rancangan proses training, evaluation, dan validation. Setelah beberapa saat hasil deteksi akan muncul dengan segmen hasil deteksi yang menunjukkan file gambar, nama jenis kelas dari file gambar, dan hasil probabilitas kelas dari file gambar. Hasil deteksi yang telah dilakukan akan menunjukkan hasil angka desimal dalam bentuk 0.XXXXXXX, X disini merupakan bentuk angka bilangan asli dari nol sampai sembilan, apabila hasil deteksi menunjukkan bentuk 0.8093428 maka ketepatan probabilitas hasil deteksi sebanyak 80%.

Dalam menentukan keakuratan proses deteksi dapat dilihat dari probabilitas hasil deteksi, apabila sistem menghasilkan nilai persentase dibawah 60% artinya persentase deteksi rendah dan perlu untuk melakukan ulang proses deteksi dengan meng-upload file gambar yang lebih jelas. Namun, jika nilai persentasenya diatas 60% maka sistem deteksi dapat dikatakan berjalan dengan akurat.

3. Hasil dan Pembahasan

3.1. Perencanaan Dataset

Bentuk dari data yang akan diolah adalah gambar visual. Data gambar tersebut diperoleh dari google image dengan cara crawling, data kemudian dibagi menjadi 20 kelas dengan total data sebanyak 700 data. Data tersebut kemudian dibagi kedalam skema 60:20:20 untuk training, validation, dan evaluation.



Gambar 3. Contoh Gambar *Dataset*

3.2. Perancangan Model

Saat data-data telah terkumpul proses selanjutnya adalah preprocessing atau perancangan model, tahapan ini dilakukan agar data dapat diketahui ukuran mean ataupun median data serta kemudian mengelompokkan data berdasarkan nama jenis penyakit.

a. *Import Libraries*

```
import glob
import shutil
import os
import random
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
```

b. *Daftar Nama Kelas*

```
penyakit_classes = ['amalgam_tattoo', 'angular_cheilitis', 'bibir_pecah-pecah', 'bibir_sumbing',
'chronic_cheek_chewing', 'cold_sores', 'gigi_berlubang', 'gigi_erosi', 'gigi-patah', 'gingivitis',
'gusi_abrasi', 'gusi_abses', 'hypodontia', 'lingua_vilosa_nigra', 'lip_pits', 'mandibular_tori',
'oral_lichen', 'periodontal', 'sariawan', 'sisa_akar']
```

c. *Split Data*

```
def split_data_into_class_folders(path_to_data, class_id):
    imgs_paths = glob.glob(path_to_data + '*.jpg')
    for path in imgs_paths:
        basename = os.path.basename(path)
        if basename.startswith(str(class_id) + '_'):
            path_to_save = os.path.join(path_to_data, penyakit_classes[class_id])
            if not os.path.isdir(path_to_save):
                os.makedirs(path_to_save)
            shutil.move(path, path_to_save)
```

Gambar 4. Setelah *Split Data*

3.3. Pelatihan Model

Rancangan model telah selesai dirancang tahapan selanjutnya adalah melakukan implementasi pelatihan menggunakan pendekatan metode Convolutional Neural Network. Tahapan ini sangatlah penting sebab seluruh keberhasilan jalannya program harus melalui tahapan pelatihan model.

a. *Import Libraries*

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.layers import Dropout, Flatten, Dense, Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
```

b. *Build Model*

```
def build_model(nbr_classes):
    base_model = InceptionV3(weights="imagenet", include_top=False,
input_tensor=Input(shape=(229, 229, 3)))
```

```

head_model = base_model.output
head_model = Flatten()(head_model)
head_model = Dense(512, activation='relu')(head_model)
head_model = Dropout(0.5)(head_model)
head_model = Dense(nbr_classes,
activation="softmax")(head_model)
model = Model(inputs=base_model.input, outputs=head_model)
for layer in base_model.layers:
layer.trainable = False
return model

```

c. Training

```

def train(path_to_data, batch_size, epochs)
    path_train_data = os.path.join(path_to_data, 'training')
    path_val_data = os.path.join(path_to_data, 'validation')
    path_eval_data = os.path.join(path_to_data, 'evaluation')
    train_generator, val_generator, eval_generator = build_data_pipelines(
        batch_size=batch_size,
        train_data_path=path_train_data,
        val_data_path=path_val_data,
        eval_data_path=path_eval_data )
    model = build_model(nbr_classes= 20)
    optimizer = Adam(lr=1e-5)
    model.compile(loss="categorical_crossentropy", optimizer=optimizer, metrics=["accuracy"])
    model.fit_generator(
        train_generator,
        steps_per_epoch=420 // batch_size,
        validation_data=val_generator,
        validation_steps=140 // batch_size,
        epochs=epochs, )

```

	precision	recall	f1-score	support
amalgam tattoo	0.56	0.57	0.44	7
angular cheilitis	0.50	0.14	0.22	7
bibir pecah-pecah	0.60	0.86	0.71	7
bibir sumbing	1.00	1.00	1.00	7
chronic_cheek_chewing	0.80	0.57	0.67	7
cold sores	0.33	0.14	0.20	7
gigi berlubang	0.38	0.43	0.40	7
gigi erosi	0.50	0.43	0.46	7
gigi patah	0.71	0.71	0.71	7
gingivitis	0.38	0.43	0.40	7
gusi abrasi	0.75	0.43	0.55	7
gusi abses	0.00	0.00	0.00	7
hypodontia	0.60	0.86	0.71	7
lingua villosa nigra	1.00	0.29	0.44	7
lip pits	0.55	0.86	0.67	7
mandibular tori	0.50	0.86	0.63	7
oral lichen	0.84	1.00	0.78	7
periodontal	0.75	0.43	0.55	7
sariawan	0.29	0.71	0.42	7
sisir akar	0.50	0.14	0.22	7
accuracy			0.54	140
macro avg	0.56	0.54	0.51	140
weighted avg	0.56	0.54	0.51	140

Gambar 5. Hasil *Training Report*

3.4. Deploy Model

Proses selanjutnya setelah model berhasil dibuat adalah adalah men-*deploy* model ke *platform google cloud*.

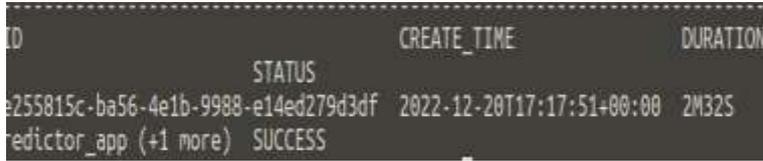
a. Membuat *dockerfile*

```

FROM tensorflow/tensorflow:latest
WORKDIR /usr/src/app
COPY . .
RUN mkdir /usr/src/app/static
RUN pip install --upgrade pip && pip install flask gunicorn opencv-python
RUN apt-get update && apt-get install -y libg1-mesa-dev libglib2.0-0 libsm6 libxrender1
libxext6

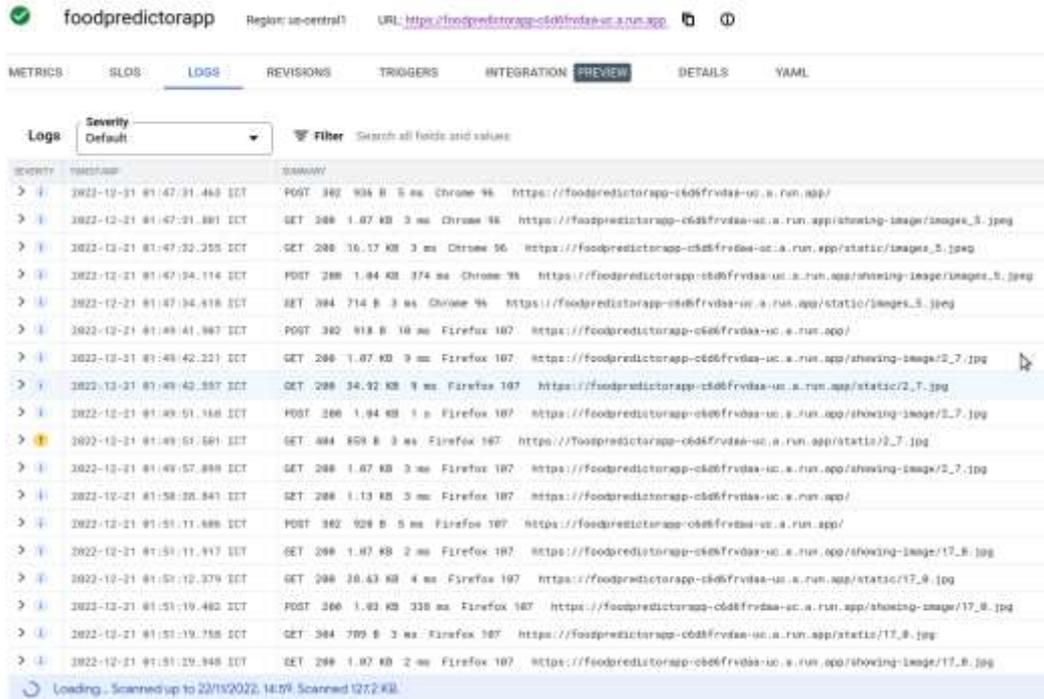
```

- CMD exec `gunicorn --bind :8080 --workers 1 --threads 8 --timeout 0 predictor:app`
- b. Mem-push docker image
`gcloud run deploy --image $IMAGE_URI --platform manage`



Gambar 6. Dockerfile berhasil di-push

- c. mengecek platform Google Cloud



Gambar 7. Website sukses berjalan

3.5. Pengujian Model

Langkah terakhir setelah semua tahapan telah selesai , dibuat adalah pengujian model. Proses pengujian model sangatlah penting sebab menjadi tolak ukur dalam keberhasilan model yang telah dibuat dan menjadi proses control kualitas sebelum digunakan oleh para pengguna.

- a. Mengakses Website
<https://foodpredictorapp-c6d6frvdaa-uc.a.run.app/>



Gambar 8. Halaman Website Muncul

- b. Mengunggah Gambar
 Pilih salah satu gambar penyakit mulut atau gigi. Setelah klik upload, gambar yang berhasil diunggah akan muncul.



Gambar 10. Gambar Berhasil Diunggah

c. Hasil Deteksi

Setelah gambar berhasil diunggah klik deteksi untuk melakukan proses deteksi nama kelas penyakit.



Gambar 11. Gambar Berhasil Dideteksi

d. Persentase Deteksi

Dari pengujian deteksi yang telah dilakukan didapatkan nama kelas deteksi adalah "bibir_sumbing" dengan nilai probabilitas 0.9999957 atau dalam bentuk persen adalah 99%. Hal ini dapat dikatakan hasil proses deteksi berjalan dengan akurat, sebab nilai hasil deteksi diatas 60%.

4. Kesimpulan

Penerapan serverless computing dapat digunakan untuk implementasi deteksi penyakit mulut dan gigi menggunakan metode CNN yang mempunyai tujuan dalam mengatasi kurangnya kesadaran masyarakat untuk memeriksakan kesehatan mulut. Dalam penelitian yang telah dilakukan menggunakan rancangan dataset penyakit mulut dan gigi terdiri dari 20 kelas yaitu Amalgam Tatto, Angular Cheilitis, Bibir pecah-pecah, Bibir Sumbing, Chronic Cheek Chewing, Cold Sores, Gigi Berlubang, Gigi Erosi, Gigi Patah, Gingivitis, Gusi Abrasi, Gusi Abses, Hypodontia, Lingua Vilosa Nigra, Lip Pits, Mandibular Tori, Oral Lichen, Periodontal, Sariawan, dan Sisa Akar. Sukses dibangun untuk deteksi dan prediksi. Proses push sukses dilakukan dengan berjalannya akses website antarmuka pengujian secara serverless computing dengan dilakukannya proses upload gambar. Sistem deteksi penyakit mulut dan gigi menggunakan metode CNN dapat bekerja dengan cukup maksimal, dari hasil rata-rata nilai probabilitas deteksi yang menghasilkan nilai diatas 60% dan nilai tertinggi yang dapat diraih adalah 99%. Hasil nilai probabilitas deteksi yang kecil disebabkan dari beberapa gambar penyakit yang memiliki nilai piksel yang sedikit dan ukuran dataset yang kecil.

PENERAPAN SERVERLESS COMPUTING DALAM MENDETEKSI PENYAKIT MULUT DENGAN METODE CNN (R. Danantyo Andaru Kusumo)

Daftar Pustaka

- [1] S. Susilawati, Sociodental Integrated Approach for Oral Health Statis Improvement, Bandung:Unpad.ac.id, 2014.
- [2] M. Ghayoumi, Deep Learning In Practice, Los Angeles: CRC Press, 2022: 1-2.
- [3] M. Awad, R. Khanna, Efficient Learning Machine, New York: Apress, 2015: 1.
- [4] S. Rosenbaum, Serverless Computing in Azure with .NET, Birmingham: Packt Publishing, 2017: 7-8.
- [5] G. A. P. Mahayadnya, Rancang Bangun Website Berbasis Cloud Computing Untuk Mendeteksi Penyakit Tanaman Menggunakan Machine Learning, Surabaya: Dinamika.ac.id, 2021: 100-104.
- [6] F. E. Ramadhan, Penerapan Image Classification Dengan Pre-trained Model Mobilenet Dalam Client-side Machine Learning, Jakarta: uinjkt.ac.id, 2020.
- [7] Z. Wang, P. Wang, P. C. Louis, L. E. Wheless, and Y. Huo. "Wearmask: Fast In-Browser Face Mask Detection with Serverless Edge Computing for Covid-19," arXiv preprint arXiv:2101.00784, vol. 1, no. 1, pp.1-8, 2021.
- [8] D. Yadav, S. Maniar, K. Sukhani and K. Devadkar, "In-Browser Attendance System using Face Recognition and Serverless Edge Computing," International Conference on Computing Communication and Networking Technologies (ICCCNT), vol. 12, no. 1, pp. 01-06, 2021.
- [9] V. Naik, Machine Learning Using Serverless Computing, San Jose: Sjsu.edu, 2021: 31-39.
- [10] C. Poelker, Defining Cloud Computing, New York: Computerworld, 2012
- [11] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional Neural Networks: An Overview and Aplication in Radiology," Insights Imaging, vol. 9, no. 1, pp. 611-629, 2018.