

Aplikasi Pengenalan Citra Wajah di KTP Menggunakan *Google Cloud Vision* API dan Kairos API Berbasis Android

Rangga Gelar Guntara*

Bisnis Digital UPI Tasikmalaya, Universitas Pendidikan Indonesia, Indonesia

ranggagelar@upi.edu

* penulis korespondensi

INFO ARTIKEL

Sejarah Artikel

Diterima: 3 Agustus 2022
Direvisi: 20 Agustus 2022
Diterbitkan: 22 Agustus 2022

Kata Kunci

Android
Computer Vision
Google Cloud Vision
Kairos
Pengenalan Wajah

ABSTRAK

Berbagai penelitian sudah mencoba beberapa metode dan algoritma untuk proses pendeteksian dan pengenalan citra wajah pada manusia. Pada penelitian ini akan memanfaatkan API *Google Cloud Vision* dan API Kairos untuk pendeteksian dan pengenalan wajah pada foto KTP. API *Google Cloud Vision* merupakan arsitektur machine learning yang dikembangkan oleh Google untuk pengolahan data berbasis visual (computer vision) seperti OCR, *Face detection*, *Image Retrieval*, dan sebagainya. Sedangkan API Kairos adalah sebuah layanan API yang berfokus pada pengenalan dan verifikasi citra wajah. Hasil pengujian pada penelitian ini menunjukkan tingkat akurasi hingga 100% dalam 10 kali percobaan menggunakan citra KTP yang berbeda. Sedangkan hasil pengujian kuesioner kepada 50 responden mendapatkan nilai persentase sebesar 4.4 dari 5.

PENDAHULUAN

Saat ini berbagai proses bisnis pada perusahaan-perusahaan memanfaatkan kartu identitas untuk melakukan verifikasi. Verifikasi dilakukan untuk kebutuhan validasi. Proses verifikasi secara detail melalui berbagai tahapan. Salah satu tahap proses adalah kegiatan verifikasi wajah. Namun sebagian besar proses verifikasi masih dilakukan secara manual. Yakni dengan cara membandingkan foto wajah pada kartu identitas dengan foto wajah dalam pas foto. Sehingga cara seperti ini seringkali membutuhkan waktu yang lama. Dibutuhkan suatu media yang dapat secara otomatis melakukan verifikasi pengenalan wajah tersebut.

Sudah banyak penelitian yang mengembangkan aplikasi pengenalan wajah manusia. Banyak pula metode dan algoritma yang digunakan untuk proses pengenalan tersebut. Penelitian Prathivi [1] mengembangkan sistem presensi di kelas menggunakan pengenalan wajah dengan metode *Haar Cascade Classifier* hasilnya memiliki presentasi sebesar 75%. Penelitian Al-Aidid & Pamungkas [2] lainnya menggabungkan metode *Haar Cascade Classifier* sebagai pendeteksi wajah pada citra dan algoritma *local binary pattern histogram* sebagai pengenalan wajahnya. Hasilnya sistem tersebut berhasil mengenali citra wajah di antara citra objek lainnya.

Algoritma *local binary pattern* (LBP) juga digunakan oleh penelitian (Mujib, Hidayatno, & Prakoso [3] untuk pengenalan wajah yang dikombinasikan dengan *support vector machine* (SVM). Citra wajah berhasil di klasifikasikan berdasarkan ciri fitur tertentu dengan tingkat akurasi mencapai 96%. Sedangkan pada penelitian Frenza & Mukhaiyar [4] menggunakan metode *Adaptive Resonance Theory* menghasilkan tingkat akurasi yang baik yaitu sebesar 92%. Penelitian Yulianti, Triastomoro, & Sa'idah [5] mengkombinasikan algoritma *local binary pattern* (LBP) untuk proses pendeteksian wajah dan metode K-

Nearest Neighbor (KNN) sebagai pengklasifikasian citra wajah dengan hasil akurasi rata-rata sebesar 85,32%.

Pada penelitian ini akan memanfaatkan API *Google Cloud Vision* dan API Kairos untuk pendeteksian dan pengenalan wajah pada foto KTP. Penelitian (Mulfari, Celesti, Fazio, Villari, & Puliafito [6] berhasil memanfaatkan API *Google Cloud Vision* untuk ekstraksi fitur citra. Penelitian Gonçalves & Paiva lainnya [7] menggunakan API *Google Cloud Vision* untuk mengenali teks, logo, dan gedung dengan tingkat akurasi mencapai 100%. API Kairos digunakan pada penelitian Vicky & Dasawaty [8] untuk pengenalan wajah dengan tingkat akurasi sebesar 88% dalam 3 kali pengujian. Sehingga sejauh pengetahuan peneliti belum menemukan penelitian yang menggabungkan API *Google Cloud Vision* untuk proses pendeteksian wajah pada foto KTP dan API Kairos untuk proses pengenalan wajah. Diharapkan dengan adanya penelitian ini dapat memberikan kontribusi yang baru dalam metode pengenalan wajah secara cepat, akurat, dan mudah dibandingkan dengan penelitian-penelitian lainnya.

METODE

Metodologi penelitian yang akan digunakan terbagi menjadi 3 bagian yaitu metode pengumpulan data, metode pengembangan perangkat lunak, dan metode pengujian. Metode pengumpulan data dilakukan untuk mempersiapkan data apa saja yang dibutuhkan. Metode pengembangan perangkat lunak merupakan metode yang berisi langkah-langkah pembangunan aplikasi. Sedangkan metode pengujian digunakan untuk mengujikan mengukur tingkat keberhasilan penelitian.

Metode Pengumpulan Data

Metode dalam proses pengumpulan data untuk keperluan penelitian ini berdasarkan pada penelitian Andryan, Aslam, Ridha, & Yaqin [9] yang terdiri dari:

1. Studi Literatur
Studi Literatur adalah proses pengumpulan data yang dilakukan melalui penggalian data dari berbagai sumber kepustakaan seperti artikel jurnal, hasil karya ilmiah, buku, prosiding dan sumber pustaka lainnya.
2. Akuisisi Dataset
Akuisisi dataset dilakukan untuk mengumpulkan *database* citra KTP dan citra wajah sebagai dataset latih dan dataset pengujian pada penelitian ini.

Metode Pengembangan Perangkat Lunak

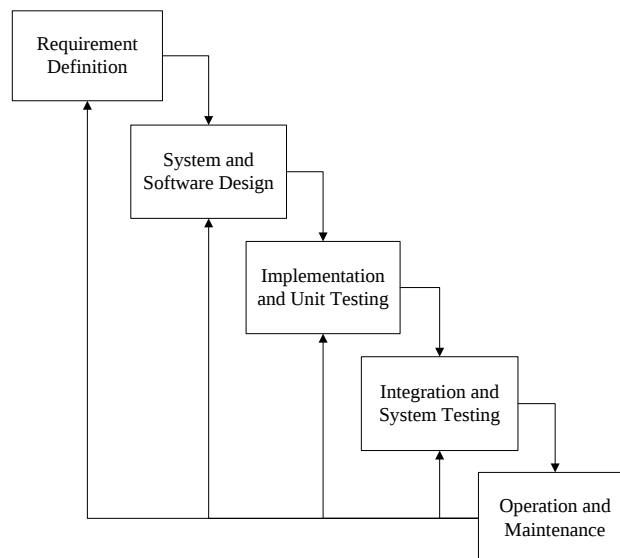
Dalam pembangunan aplikasi pada penelitian ini akan menggunakan pendekatan model *waterfall* yang merupakan model tahapan pengembangan perangkat lunak (Karim, Wijayanti, & Kurniadin [10], sebagaimana ditampilkan pada Gambar 1.

3. *Requirement analysis and definition*
Tahapan analisis pada penelitian ini meliputi analisis arsitektur sistem, analisis pendeteksian citra wajah KTP dengan menggunakan API *Google Cloud Vision*, analisis pengenalan citra wajah dengan API Kairos.
4. *System and software design*
Perancangan terdiri dari perancangan basis data, perancangan antarmuka, perancangan API.
5. *Implementation and unit testing*
Tahap implementasi terdiri dari implementasi teknologi, implementasi *class*, implementasi antarmuka, implementasi basis data.
6. *Integration and system testing*

Tahap pengujian yang dilakukan terbagi menjadi dua yaitu pengujian alpha dan pengujian beta. Untuk pengujian alpha menggunakan metode pengujian *blackbox*. Sedangkan pengujian beta menggunakan metode kuesioner.

7. *Operation and maintenance*

Tahap menjalankan program di lingkungannya disertai dengan pengawasan dan pemeliharaan sistem. Hal ini dilakukan untuk proses penyesuaian atau adanya perubahan saat sistem dijalankan di lapangan.



Gambar 1. Model Waterfall

Metode Pengujian

Pengujian dilakukan untuk menentukan apakah aplikasi yang dibangun sudah sesuai dengan tujuan pembuatan aplikasi di awal. Pada penelitian ini tahapan pengujian aplikasi akan dilakukan secara dua tahap. Tahap pertama adalah pengujian secara internal (pengujian *alpha*) oleh pengembang aplikasi dengan metode *black-box*. Sedangkan tahap kedua (pengujian beta) akan dilakukan pengujian kepada pengguna awal (*early adopter*) melalui metode kuesioner.

Pengujian alpha adalah pengujian yang dilakukan secara internal di lingkungan pengembang aplikasi untuk menguji semua fungsionalitas dari sistem yang dibangun [11]. Sedangkan metode pengujiannya akan menggunakan pengujian *black-box*. Dimana skenario pengujiannya dilakukan dengan cara menguji keberhasilan fungsionalitas aplikasi dengan memberikan data masukan. Data masukan yang diuji terbagi menjadi dua bagian. Bagian pertama menguji dengan data masukan yang benar. Bagian kedua menguji dengan data masukan yang salah. Kesimpulan yang diambil adalah apabila hasil keluaran sesuai dengan kebutuhan fungsional maka dianggap sudah benar.

Setelah proses pengujian alpha selesai, dilanjutkan dengan pengujian beta. Pengujian beta merupakan pengujian yang melibatkan pihak eksternal dalam hal ini adalah pengguna awa. Pengguna awal biasanya adalah responden awal yang memiliki permasalahan. Sehingga diharapkan dengan adanya pengujian ke pengguna awal ini, tujuan penelitian dapat langsung terjawab dengan aplikasi yang dibangun. Untuk pengumpulan data dalam menggunakan pengujian beta ini adalah melalui instrument kuesioner.

Adapun teknik untuk mendapatkan nilai rata-rata dari setiap jawaban yang memetakan fungsionalitas aplikasi menggunakan rumus Skala *Likert* Persamaan (1) merupakan rumus skala likert untuk menghitung nilai persentase.

$$p = \frac{T}{n} \times 100 \quad (1)$$

Dimana:

p = nilai persentase yang dicari

T = jumlah total skor

n = nilai tertinggi dikalikan dengan jumlah responden

HASIL DAN PEMBAHASAN

Tahapan hasil dan pembahasan pada penelitian ini dituliskan berdasarkan tahapan pada metode pengembangan perangkat lunak menggunakan model *Waterfall*. Yakni dari mulai tahapan analisis, perancangan, implementasi, pengujian dan pemeliharaan.

Analisis Arsitektur Sistem

Pada penelitian ini akan menggunakan aplikasi berbasis android dan memanfaatkan API *Google Cloud Vision* untuk deteksi wajah pada foto. Arsitektur pada sistem android akan melibatkan *smartphone*, *server*, *web service/API*, API *Google Cloud Vision*, API *Kairos* dan database. Untuk lebih jelasnya dapat dilihat pada Gambar 2 yang menjelaskan arsitektur sistem pada penelitian ini.



Gambar 2. Arsitektur Sistem

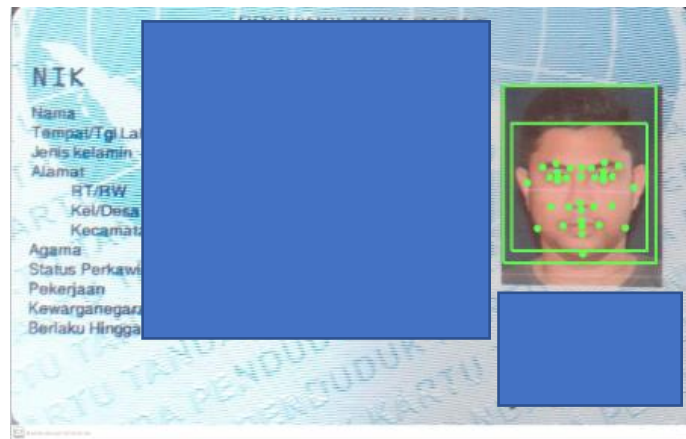
Penjelasan proses setiap tahap pada arsitektur sistem yang dibangun pada penelitian ini adalah sebagai berikut:

- 1) Aplikasi *user* membuat proses permintaan (*request*) ke API *Google Cloud Vision* untuk mendeteksi wajah pada foto.
- 2) API *Google Cloud Vision* menerima request dari perangkat aplikasi mobile pengguna.
- 3) API *Google Cloud Vision* memberikan response data hasil deteksi wajah pada foto.

- 4) Aplikasi *user* membuat proses permintaan data foto ke database yang sudah tersimpan di server.
- 5) Database memberikan response data foto.
- 6) Perangkat mobile pengguna melakukan request ke API Kairos untuk melakukan pengenalan wajah.
- 7) API Kairos memberikan response hasil pengenalan wajah ke sistem mobile pengguna.
- 8) Aplikasi *user* mendapatkan *response* dari API Kairos dengan hasil keluaran berupa JSON. Format JSON ini kemudian akan di parsing oleh aplikasi *user* untuk ditampilkan.

Analisis Pendeteksian Citra Wajah KTP

Untuk proses mendeteksi citra wajah pada KTP Indonesia akan menggunakan API *Google Cloud Vision*. Aplikasi yang dibangun akan mengirim citra KTP yang sudah di konversi dalam bentuk *String encode base64*. String tersebut sebagai parameter dalam mengakses API *Google Cloud Vision*. Gambar 3 menunjukkan hasil pendeteksian citra wajah pada KTP. Adapun Output yang dihasilkan oleh API berupa format JSON. Sedangkan pada Gambar 4 menampilkan format permintaan ke API beserta format keluaran yang dihasilkannya.



Gambar 3. Hasil Pendeteksian Citra Wajah Pada KTP



Gambar 4. Request dan Response API

Analisis Pengenalan Citra Wajah Menggunakan API Kairos

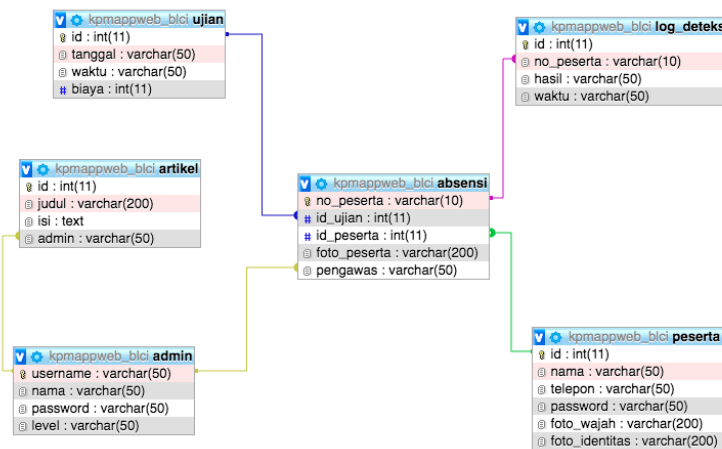
Hasil dari pendeteksian wajah pada citra KTP akan dikirim sebagai parameter ke API Kairos. Pada API ini akan dilakukan proses pengenalan wajah dengan cara membandingkan dengan citra wajah yang sudah tersimpan di *dataset*. Proses penggunaan API dengan cara mengakses url melalui protokol cURL disertai dengan parameter berupa citra wajah yang

akan dikenali. Citra yang disematkan sebagai parameter harus di konversikan terlebih dahulu ke format String encode64.

Kemudian API akan mencari citra wajah di dataset untuk dibandingkan dengan citra input. Hasil pemrosesan akan dikembalikan oleh API berupa format array JSON. Aplikasi akan melakukan parsing array JSON tersebut, sehingga dapat ditampilkan ke pengguna. Proses pengenalan ini membutuhkan koneksi internet karena pemanggilan API dilakukan melalui format cURL.

Perancangan Basis Data

Perancangan basis data dilakukan untuk menghindari redundansi dan duplikasi data pada sistem [12]. Adapun perancangan basis data pada penelitian ini dibuat berdasarkan kebutuhan fungsional perangkat lunak.



Gambar 5. Skema Relasi Basis Data

Perancangan Antarmuka

Perancangan antarmuka dilakukan agar perangkat lunak yang dibuat mudah digunakan oleh pengguna akhir. Dalam merancang perangkat lunak harus memperhatikan pengalaman pengguna dan kebutuhan pengguna [13].



Gambar 6. Perancangan Antarmuka

Implementasi Teknologi

Implementasi teknologi merupakan tahap untuk menerapkan kode dari teknologi yang digunakan pada perangkat lunak yang dibangun [14]. Adapun implementasi teknologi pada penelitian ini terdiri dari teknologi API *Google Cloud Vision* dan API *Kairos*.

a. Teknologi Face Detection

```

private void deteksiWajah() {
    Paint myRectPaint = new Paint();
    myRectPaint.setStrokeWidth(5);
    myRectPaint.setColor(Color.RED);
    myRectPaint.setStyle(Paint.Style.STROKE);

    Bitmap tempBitmap =
    Bitmap.createBitmap(image_to_upload.getWidth(),
    image_to_upload.getHeight(), Bitmap.Config.RGB_565);
    Canvas tempCanvas = new Canvas(tempBitmap);
    tempCanvas.drawBitmap(image_to_upload, 0, 0, null);
    FaceDetector faceDetector = new

FaceDetector.Builder(getApplicationContext()).setTrackingEnabled(false
)
    .build();
    if (!faceDetector.isOperational()) {
        new
AlertDialog.Builder(AbsenActivity.this).setMessage("Could not set up
the face detector!").show();
        return;
    }
    ByteArrayOutputStream baol = new ByteArrayOutputStream();
    newBitmap.compress(Bitmap.CompressFormat.JPEG, 100, baol);
    byte[] baPlace1 = baol.toByteArray();
    foto_wajah = Base64.encodeToString(baPlace1, Base64.DEFAULT);
    deteksi();
}

```

b. Teknologi Face Recognition

```

public void deteksi() {
    String url = AppConstans.deteksi;
    Map<String, String> params = new HashMap<>();
    params.put("foto_wajah", foto_wajah);
    APIManager.DataDeteksi(this, new OnApiListener() {
        @Override
        public void onApiStartLoad(String url) {
            showProgressDialog();
        }
        @Override
        public void onApiSuccessLoad(String url, String message,
Object object, int resultNumber) {
            hideProgressDialog();
            //      hasilItems = (ArrayList<HasilItem>) object;
            //showToast(object.toString());
            try {
                JSONArray array = new
JSONArray(object.toString());
                for (int i = 0; i < array.length(); i++) {
                    JSONObject jsonObject =
array.getJSONObject(i);
                    name = jsonObject.getString("name");
                }
                if (name.equals("nihil")) {
                    layout2.removeAllViews();
                    for (int i = 0; i < jml_wajah; i++) {
                        ImageView image2 = new
ImageView(AbsenActivity.this);

```

```
image2.setLayoutParams(new  
android.view.ViewGroup.LayoutParams(150, 150));  
image2.setMaxHeight(20);  
image2.setMaxWidth(20);  
}
```

Pengujian *Black-box*

Pengujian blackbox dilakukan pada penelitian ini untuk menguji fungsionalitas perangkat lunak yang telah dibuat [15]. Pengujian berdasarkan data input dan hasil yang ditampilkan. Semua fungsionalitas aplikasi yang dibangun dilakukan pengujian *black-box*. Tabel 1 menunjukkan hasil pengujian fungsionalitas pengenalan wajah.

Tabel 1. Hasil Pengujian Pengenalan Wajah

Kasus dan Hasil Uji (Data Benar)			
Data Input	Hasil yang Diharapkan	Pengamatan	Kesimpulan
Ktp.jpg	Sistem menampilkan hasil pengenalan wajah	Tampil informasi pengenalan wajah	Diterima
Kasus dan Hasil Uji (Data Salah)			
Ktp_salah.jpg	Sistem menampilkan pesan kesalahan hasil pengenalan wajah	Tampil informasi pesan kesalahan pengenalan wajah	Diterima

Berdasarkan hasil pengujian *Black-Box* aplikasi yang dibangun berhasil melakukan pengenalan citra wajah dari KTP dengan baik. Sebanyak 10 kali percobaan hasil akurasi mencapai 100%. Hal ini dapat terjadi karena citra wajah pada KTP memiliki karakteristik yang jelas, tanpa ekspresi, dan berformat yang seragam.

Hasil Pengujian Beta

Dalam setiap jawaban kuesioner akan diberikan skor sesuai dengan Tabel 2 skor jawaban kuesioner berikut.

Tabel 2. Skor Jawaban Kuesioner

Skala Jawaban	Keterangan	Skor
SS	Sangat Setuju	5
S	Setuju	4
RG	Ragu	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

Kuesioner dilakukan kepada 50 responden untuk melakukan pengujian aplikasi pengenalan wajah dari KTP ini. Adapun hasil persentase dapat dilihat pada Tabel 3 hasil pengujian kuesioner berikut.

Tabel 3. Hasil Jawaban Kuesioner

No	Keterangan	Jawaban Responden	Skor
1	Sangat Setuju	27	135
2	Setuju	16	64
3	Ragu	7	21
4	Tidak Setuju	0	0

5	Sangat Tidak Setuju	0	0
Jumlah		50	220

Kemudian tahap berikutnya adalah menghitung nilai persentase berdasarkan hasil jawaban dari kuesioner sebagai berikut.

$$\text{Rata - rata} = \frac{\text{jumlah skor}}{\text{jumlah responden}}$$

$$\text{Rata - rata} = \frac{220}{50}$$

$$\text{Rata - rata} = 4.4$$

Sehingga, berdasarkan hasil perhitungan nilai persentase yang didapat yaitu sebesar rata-rata 4.4 dari 5. Hasil ini masih berada pada area jawaban setuju dari 50 responden yang menjawab.

KESIMPULAN

Berdasarkan hasil pengujian maka penelitian membangun aplikasi pengenalan wajah dengan menggunakan API *Google Cloud Vision* dan API Kaios berhasil dilakukan. API *Google Cloud Vision* berhasil melakukan proses pendeteksian letak citra wajah pada dokumen KTP dengan baik. Kemudian citra wajah hasil pendeteksian tersebut dikirim ke API Kaios untuk dilakukan proses pengenalan wajah berdasarkan dataset yang sudah dipersiapkan sebelumnya. Hasilnya memiliki tingkat akurasi 100% dengan percobaan sebanyak 10 kali untuk citra KTP yang berbeda-beda.

REFERENSI

- [1] R. Prathivi, "SISTEM PRESENSI KELAS MENGGUNAKAN PENGENALAN WAJAH DENGAN METODE HAAR CASCADE CLASSIFIER," *Jurnal SIMETRIS*, vol. 11, no. 1, pp. 135-142, 2020.
- [2] S. Al-Aidid and D. S. Pamungkas, "Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram," *Jurnal Rekayasa Elektrika*, vol. 14, no. 1, pp. 62-67, 2018.
- [3] K. Mujib, A. Hidayatno and T. Prakoso, "PENGENALAN WAJAH MENGGUNAKAN LOCAL BINARY PATTERN (LBP) DAN SUPPORT VECTOR MACHINE (SVM)," *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 7, no. 1, pp. 123-130, 2018.
- [4] D. Frenza and R. Mukhaiyar, "Aplikasi Pengenalan Wajah dengan Metode Adaptive Resonance Theory (ART)," *Ranah Research : Journal of Multidisciplinary Research and Development*, vol. 3, no. 3, pp. 35-42, 2021.
- [5] D. R. YULIANTI, I. I. TRIASTOMORO and S. SA'IDAH, "IDENTIFIKASI PENGENALAN WAJAH UNTUK SISTEM PRESENSI MENGGUNAKAN METODE KNN (K-NEAREST NEIGHBOR)," *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, vol. 5, no. 1, pp. 1-10, 2022.
- [6] D. Mulfari, A. Celesti, M. Fazio, M. Villari and A. Puliafito, "Using Google Cloud Vision in assistive technology scenarios," *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 214-219, 2016.
- [7] J. Gonçalves and S. Paiva, "Inclusive Mobility Solution for Visually Impaired People using Google Cloud Vision," *2021 IEEE International Smart Cities Conference (ISC2)*, pp. 1-7, 2021.
- [8] V. and E. S. Dasawaty, "APLIKASI SISTEM PARKIR MENGGUNAKAN FACE RECOGNITION BERBASIS ANDROID DENGAN FRAMEWORK REACT NATIVE," *Program Studi Teknik Informatika*, 2019.

- [9] M. Andryan, M. N. Aslam, M. Ridha and M. A. Yaqin, "Sistem Pendukung Keputusan Untuk Menentukan Manajer Proyek Menggunakan Metode Simple Additive Weighting (SAW)," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 3, no. 3, pp. 373-386, 2021.
- [10] S. Karim, P. Wijayanti and N. Kurniadin, "Sistem Informasi Geografis Tanah Bersertifikat di Perumahan Samarinda Hills Menggunakan LeafletJS Berbasis Web," *ILKOMNIKA : Journal of Computer Science and Applied Informatics*, vol. 3, no. 3, pp. 304-313, 2021.
- [11] R. G. Guntara, "Firebase Realtime Database Untuk Aplikasi Point of Sales UMKM Berbasis Cloud Computing Pada Smartphone Android," *Impression : Jurnal Teknologi dan Informasi*, vol. 1, no. 2, pp. 1-10, 2022.
- [12] Y. Indarta, D. Irfan, M. Muksir, W. Simatupang and F. Ranuharja, "Analisis dan Perancangan Database Menggunakan Model Konseptual Data Warehouse Sistem Manajemen Transaksi Toko Online Haransaf," *Edukatif : Jurnal Ilmu Pendidikan*, vol. 3, no. 6, 2021.
- [13] F. Effendi, K. Andreas, C. B. Valentino, D. Johan, J. Jonathan, L. R. Wijaya and M. R. Pribadi, "Perancangan Antarmuka Aplikasi Monteer dengan Metode Design Thinking," *MDP STUDENT CONFERENCE (MSC)*, vol. 1, no. 2, 2022.
- [14] "Pemaparan Teknologi Biometrika Dan Google Cloud Vision Api Di SMK Dwi Tunggal 2 Tanjung," *Jurnal TUNAS*, vol. 1, no. 2, 2020.
- [15] P. Saman and C. I. Ratnasari, "Pengujian Black Box Pada Aplikasi Pembelajaran Bahasa Mandarin Berbasis Android," *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, vol. 4, no. 1, 2022.