BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian ini dilakukan beberapa tinjauan pustaka penelitian terkait yang nantinya akan mendukung penelitian yang akan dilakukan, berikut adalah beberapa sumber pustaka yang digunakan dalam penelitian ini :

1. Oleh Arief Budiman dan Joko Triono (2016) Jurnal Ilmiah Ilmu-ilmu Teknik dengan judul "Sistem Informasi Parkir Kendaraan Bermotor Berbasis Android" [7]. Pada penelitian ini pengembangan aplikasi sistem informasi berbasis android akan digunakan oleh petugas parkir sebagai admin. Perancangan sistem memanfaatkan QR *code* yang merepresentasikan data kendaraan seperti nama pemilik, plat nomor kendaraan, dan foto pemilik kendaraan. Aplikasi android dijalankan oleh admin untuk memindai QR *code* yang terpasang pada setiap kendaraan, admin akan melakukan pemindaian ketika pengendara memasuki tempat parkir dan hasil pemindaian pertama akan menunjukkan keterangan parkir serta keterangan identitas pemilik kendaraan. Ketika pengendara akan keluar tempat parkir, admin akan memindai lagi QR *code* yang terpasang pada kendaraan dan akan menunjukkan keterangan parkir keluar beserta informasi kendaraan dan tanggal parkir saat itu. Pada aplikasi sistem aplikasi parkir ini memiliki fitur yang dapat menampilkan seluruh daftar pemilik kendaraan, apabila admin menekan salah satu list daftar nama pemilik kendaraan maka akan muncul menu edit dan delete, serta fitur tambah kendaraan untuk mendaftarkan kendaraan yang belum terdaftar pada database.
2. Oleh Bagus Septian Adityia Wijayanto, Fitri Utaminingrum, dan Issa Arwani (2019) Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer dengan judul "Face Recognition Untuk Sistem Pengaman Rumah Menggunakan Metode HOG dan KNN Berbasis Embedded" [8]. Pada penelitian ini mengembangkan sistem pengamanan rumah *face recognition*

berbasis *embedded* menggunakan metode *Histogram of Oriented Gradient* (HOG). Sistem ini bekerja dengan mengekstraksi fitur wajah yang terdeteksi dan akan digunakan untuk klasifikasi wajah menggunakan metode *k-nearest neighbor*. Ketika wajah dikenali oleh sistem akan membuka kunci pintu dan menyalakan LED berwarna hijau, namun ketika wajah tidak dikenali maka pintu tetap terkunci, buzzer akan berbunyi dan mengirimkan sms notifikasi kepada pemilik rumah serta LED akan menyala berwarna merah. Dari hasil pengujian didapatkan tingkat akurasi deteksi wajah sebesar 100% pada jarak 40cm dan tingkat akurasi pengenalan wajah sebesar 87,5%.

3. Oleh Adlan Bagus Pradana, Cholifah Ma'rifadiyah, Dwiantono Jatinugroho, dan Fakhrurrozi Zainal Abidin (2019) Jurnal Teknik Elektro dengan judul "Perancangan Sistem Perparkiran Rendah Biaya Berbasis Ponsel Cerdas Android" [9]. Pada penelitian ini mengembangkan aplikasi parkir menggunakan QR *code* dengan proses validasi atau pemindaian dilakukan dua kali saat masuk maupun keluar tempat parkir. Pemindaian dilakukan pada QR *code* yang berada pada kartu mahasiswa maupun kartu karyawan untuk identifikasi pengendara dan QR *code* yang ditempelkan di kendaraan untuk identifikasi kendaraan, kemudian kedua data disimpan pada database. Ketika pengendara keluar dari tempat parkir akan dilakukan pengecekan kesesuaian data pengendara dan kendaraan. Dengan waktu yang diperlukan untuk proses pemindaian dua QR *code* (pengendara dan kendaraan) adalah selama 10 detik.
4. Oleh Saeful Bahri dan Heri Kusindaryadi (2020) Jurnal Elektronika Kendali Telekomunikasi Tenaga Listrik Komputer dengan judul "Rancang Bangun Pemantauan Absensi Mahasiswa dengan Menggunakan Sidik Wajah secara Simultan Melalui CCTV Ruang Kelas" [10]. Penelitian ini bertujuan untuk membuat sistem absensi dengan *face recognition* (pengenalan wajah) menggunakan metode Histogram of Oriented Gradient (HOG). Sistem akan melakukan pendeteksian wajah mahasiswa menggunakan metode HOG dengan pengambilan gambar melalui raspberry. Sistem akan melakukan pencocokan wajah pada database dan mengirimkan hasil pengenalan wajah

ke PC/laptop. Ketika proses pengambilan gambar terdapat wajah mahasiswa yang tidak terdeteksi, sistem akan melakukan pengambilan gambar dan pencocokan wajah pada database kembali sampai wajah mahasiswa terdeteksi sesuai dengan database wajah yang didaftarkan. Dengan hasil pengujian sistem dapat berjalan dengan baik ketika pencahayaan berada pada tingkat minimum 80 lux dan maksimum 300 lux serta jarak mahasiswa dari kamera raspberry jarak minimum 10 cm dan maksimum 2 meter.

5. Oleh Sunanto, Yoze Rizki dan Yulia Fatma (2020) *Jurnal of Information Technology and Computer Science* dengan judul "Sistem Parkir Cerdas Menggunakan Teknologi Biometrika Dan Optical Character Recognition" [11]. Pada penelitian ini mengembangkan sistem parkir dengan menggunakan 2 validasi input yaitu sidik jari pengendara dan plat nomor kendaraan. Sistem akan melakukan pemrosesan citra plat nomor kendaraan dengan menggunakan Optical Character Recognition (OCR) untuk mengubah citra *image* atau gambar menjadi teks dan menyimpannya ke dalam database bersama dengan sidik jari pengendara ketika memasuki tempat parkir. Kemudian sistem akan melakukan pengecekan apakah sidik jari dan plat nomor kendaraan sesuai dengan data saat pengendara masuk, jika data sesuai maka palang parkir akan terbuka sedangkan jika tidak sesuai maka alarm akan berbunyi dan palang parkir tetap tertutup. Dengan hasil pengujian deteksi <56% ketika plat nomor kendaraan palsu dan putih (plat dari dealer) sehingga palang parkir tidak terbuka sedangkan untuk plat nomor kendaraan asli hasil pendeteksian >75% maka palang parkir terbuka. Hasil keseluruhan akurasi validasi sidik jari dan plat nomor kendaraan sebesar >=75%.

Berikut ini diberikan tabel perbandingan dari penelitian terkait :

Tabel 2.1 Perbandingan penelitian terkait

No	1	2	3	4	5
Judul	Sistem Informasi Parkir Kendaraan Bermotor Berbasis Android	Face Recognition Untuk Sistem Pengaman Rumah Menggunakan Metode HOG dan KNN Berbasis Embedded	Perancangan Sistem Perparkiran Rendah Biaya Berbasis Ponsel Cerdas Android	Rancang Bangun Pemantauan Absensi Mahasiswa dengan Menggunakan Sidik Wajah secara Simultan Melalui CCTV Ruang Kelas	Sistem Parkir Cerdas Menggunakan Teknologi Biometrika Dan Optical Character Recognition
Peneliti	- Arief Budiman - Joko Triono	- Bagus Septian Adityia Wijayanto - Fitri Utaminingrum - Issa Arwani	- Adlan Bagus Pradana - Cholifah Ma'rifadiyah - Dwiantono Jatinugroho - Fakhrurrozi Zainal Abidin	- Saeful Bahri - Heri Kusindaryadi	- Sunanto - Yoze Rizki - Yulia Fatma
Tahun	2016	2019	2019	2020	2020
Metode / Teknologi	- QR code	- Face recognition - Histogram of oriented gradient (HOG) - K-nearest neighbour	- QR code	- Face recognition - Histogram of oriented gradient (HOG)	- Biometrik sidik jari - Optical Character Recognition (OCR)

No	1	2	3	4	5
Hasil	Hasil dari penelitian ini berupa aplikasi android dengan menggunakan teknologi QR *code* yang digunakan oleh petugas parkir untuk melakukan pemindaian QR *Code*.	Hasil dari penelitian ini berupa prototipe sistem keamanan rumah berbasis *embedded* dengan hasil pengujian sebesar 100% untuk tingkat akurasi deteksi wajah dan 87,5% untuk akurasi pengenalan wajah.	Hasil dari penelitian ini berupa aplikasi android yang digunakan oleh petugas parkir untuk pemindaian QR code, dengan proses validasi dilakukan dua kali yaitu pada QR code kartu mahasiswa dan QR code kendaraan.	Hasil dari penelitian ini berupa sistem absensi mahasiswa berbasis *embedded*. Dengan hasil pengujian sistem berjalan dengan baik ketika pencahayaan berada pada tingkat 80-300 lux serta jarak mahasiswa dari kamera berada pada 10 centimeter sampai 2 meter	Hasil dari penelitian ini berupa sistem palang parkir menggunakan 2 validasi input, yaitu sidik jari dan plat nomor kendaraan yang telah diubah menjadi teks menggunakan metode OCR. Dengan hasil pengujian deteksi <56% untuk plat nomor palsu dan >75% untuk plat nomor asli.
Perbedaan	Pada metode yang digunakan yaitu Histogram of oriented gradient serta penggunaan *face recognition*	Pada objek penelitian yaitu tempat parkir, dan hasil akhir produk berupa aplikasi android	Pada metode yang digunakan yaitu Histogram of oriented gradient serta penggunaan *face recognition*	Pada objek penelitian yaitu tempat parkir, dan hasil akhir produk berupa aplikasi android	Teknologi yang digunakan yaitu face recognition serta metode Histogram of oriented gradient

Dari beberapa penelitian diatas, dapat disimpulkan bahwa sistem parkir sangat dibutuhkan. Dengan adanya sistem parkir, proses keluar masuk akan tercatat pada aplikasi serta keamanan parkir menjadi lebih ketat dan aman dengan pengecekan ganda. Berdasarkan penelitian sebelumnya, penulis menambahkan maupun menggabungkan beberapa hal yang menjadi pembeda dengan penelitian sistem parkir yang telah dikembangkan sebelumnya. Perbedaan tersebut adalah sebagai berikut :

1. Sistem aplikasi parkir memiliki dua hak akses yaitu user (pengguna/pengendara) dan admin (petugas parkir).
2. QR *code* dalam bentuk stiker ditempelkan pada kendaraan dan merepresentasikan plat nomor kendaraan.
3. Pada aplikasi ini menggabungkan dua teknologi yaitu QR *code* untuk identifikasi kendaraan dan *face recognition* untuk identifikasi wajah pengendara.
4. User dapat mendaftarkan kendaraannya sendiri dan mengunduh QR *code* hasil *generate* untuk dicetak dan ditempelkan pada kendaraan.
5. Pendaftaran meliputi dua hal, yaitu pendaftaran user (pengguna) dan pendaftaran kendaraan untuk mendapatkan QR *code*.
6. User harus memindai QR *code* yang disediakan di tempat parkir ketika masuk ke tempat parkir dan ketika akan keluar maka admin akan melakukan pemindaian QR *code* yang terpasang pada kendaraan kemudian memindai wajah pengendara, jika wajah pengendara sama dengan data ketika masuk ke tempat parkir maka diperbolehkan keluar.

2.2 Landasan Teori

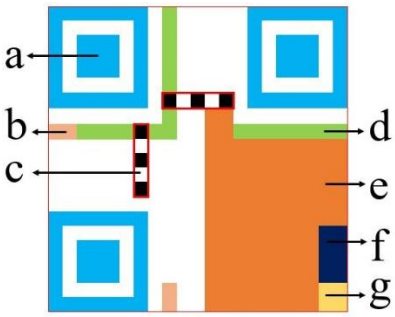
2.2.1 QR *Code* (*Quick Response Code*)

QR *code* (*Quick Response Code*) adalah teknologi untuk membuat kode dua dimensi yang dapat dicetak pada media yang lebih ringkas dari data tertulis [3]. QR *code* memiliki kemampuan untuk menyimpan semua jenis data, seperti data angka, alpanumerik, biner. Ukuran cetak yang kecil dan kapasitas data yang besar membuat QR *code* sangat populer. QR *code* dapat menyimpan kapasitas data hingga 7.089 karakter numerik, 2.953 *binary bytes*, 4.296 karakter alpanumerik [5].

Gambar 2.1 Contoh qr code

QR *code* terdiri dari tiga modul hitam (titik persegi) pada kotak persegi dengan latar belakang putih sehingga dapat dibaca dari segala arah dalam 360° walaupun terdapat distorsi. QR *code* merupakan hasil dari pengembangan dari kode batang (*barcode*). Beberapa keunggulan yang dimiliki QR *code* dibanding jenis barcode lainnya, yaitu :

1. Kapasitas yang lebih besar
2. Mampu menyimpan angka dan huruf
3. Dapat dibaca dari berbagai arah
4. Ukuran yang kecil
5. Mudah dibaca
6. Tahan terhadap kotor dan kerusakan



Gambar 2.2 Bagian anatomi qr code

Pada Gambar 2.2 merupakan gambaran dari bagian atau anatomi QR *code* dengan nama masing-masing bagian dapat dilihat sebagai berikut :

a. *Finder pattern* untuk mengetahui posisi letak QR code.
b. *Level error correction* QR *code.*
c. *Timing pattern* untuk identifikasi pusat QR *code.*
d. *Format information* yang berisi *error correction level* dan indikator *pattern.*

e. Isi data hasil *encoding*.
f. Panjang data dalam 14 *byte*.
g. Jenis data yang telah dikonversi ke dalam biner.

2.2.1.1 *Encoding* QR *Code*

Encoding merupakan proses pengubahan data tertulis ke dalam bentuk QR *code*. Pada proses encoding QR *code* memiliki beberapa tahapan yakni [12] :

a. Konversi data ke dalam bentuk biner dengan analisis model *encoding* numerik, alpanumerik, atau *byte* model.
b. Konversi biner ke dalam bentuk desimal untuk pencocokkan karakter awal dengan karakter yang dihasilkan dari tabel kode ASCII.
c. Koreksi kesalahan *coding codeword* pada setiap blok data.
d. Alokasikan data yang telah disandikan ke dalam bentuk QR *code*. Data yang dialokasikan merupakan data hasil perhitungan kesalahan dan hasil representasi biner.
e. Penempatan modul matriks yang mencakup pola fungsi utama pada QR *code* dengan aturan sebagai berikut :
    - Data yang akan dialokasikan sesuai dengan kapasitas data pada versi QR *code* ke dalam matriks.
    - Penempatan data pertama kali dimulai pada koordinat pojok kanan bawah dan dilanjutkan diletakkan diatasnya.
    - Jika kanan penuh maka dilanjutkan ke arah bawah dengan tetap memperhatikan arah penempatan yaitu dari kanan ke kiri.
f. Menentukan pola data (*masking*) untuk modifikasi QR *code* agar mudah dibaca oleh alat pemindai QR *code*.
g. Menambahkan format informasi dan versi yang digunakan untuk menampung data informasi. Isi dari format informasi merupakan koreksi kesalahan dan indikator *pattern*.

Tabel 2.2 Format informasi

No	Koreksi Kesalahan	Indikator
1	L (Low) – 7%	01
2	M (Medium) – 15%	00
3	Q (Quartile) – 25%	11
4	H (High) – 30%	10

2.2.1.2 *Decoding* QR *Code*

Decoding merupakan proses pembacaan QR *code* yang berisi suatu data untuk mengetahui isi informasi dari data yang ada pada QR *code*. Pada proses *decoding* ini memiliki beberapa tahapan yakni [13] :

a. Mengenali *finder pattern* atau pola pencari yang terletak pada kiri bawah, kiri atas, dan kanan atas.
b. Mendeteksi kemiringan dan perputaran simbol yang berpatokan pada *finder pattern*.
c. Memperkirakan versi dari simbol QR *code* yang digunakan.
d. Ekstraksi format informasi yang akan dibaca.
e. Pendeteksian *error correction* pada data.
f. Mengembalikan data asli dengan mengembalikan rangkaian data ke dalam bentuk awal sesuai dengan indikator pada format informasi data.

2.2.2 *Library* Zxing

Library Zebra crossing atau yang lebih sering disebut sebagai zxing merupakan sebuah pustaka (*library*) *open source* yang dapat melakukan pemrosesan berbagai format *barcode* 1 dimensi maupun 2 dimensi. *Library* Zxing berfokus pada penggunaan kamera yang digunakan untuk proses pemindaian pada smartphone, serta *library* ini masih dapat digunakan pada *server* dan juga *desktop*. Proses *encoding* dan *decoding* QR *code* dapat dilakukan dengan menggunakan *library* zxing ini.

2.2.3 Face Recognition

Face recognition atau pengenalan wajah merupakan bagian dari computer vision yang dapat melakukan identifikasi atau verifikasi wajah seseorang. *Face*

recognition merupakan teknologi biometrik selain pengenalan sidik jari, retina mata, dan iris mata. Dalam implementasinya, *face recognition* menggunakan kamera untuk menangkap wajah individu kemudian dibandingkan dengan wajah yang ada pada *database*. Pada *face recognition* umumnya memiliki dua tahapan dalam pemrosesannya. Tahap pertama yaitu *face detection*, pencarian atau pendeteksian wajah yang terdapat dalam sebuah citra. Tahap kedua adalah *face recognition*, pemrosesan wajah yang terdeteksi kemudian mengenali wajah dengan membandingkan wajah yang tersimpan pada *database*.

2.2.4 *Histogram of Oriented Gradient* (HOG)

Histogram of Oriented Gradient (HOG) *detection* merupakan ekstraksi fitur pada *computer vision* untuk mendeteksi suatu objek dari hasil perhitungan nilai gradien pada suatu citra yang diperoleh [14]. Hal utama pada HOG adalah tampilan lokal objek serta bentuk citra yang dijelaskan oleh arah tepi atau distribusi intensitas gradien. Proses untuk membangun HOG menggunakan deteksi tepi Sobel 1-D dengan menghitung nilai gradien horizontal dan vertikal dengan rumus sebagai berikut :

$$f_x(x,y) = I(x+1,y) - I(x-1,y) \quad (2.1)$$

$$f_y(x,y) = I(x,y+1) - I(x,y-1) \quad (2.2)$$

Keterangan :

f_x(x,y) = nilai gradien citra horizontal

f_y(x,y) = nilai gradien citra vertikal

I(x,y) = intensitas citra/nilai piksel pada baris x dan kolom y

Kemudian magnitude (m) dan orientasi gradien (θ) dikalkulasikan dari nilai gradien tersebut, dengan menggunakan rumus :

$$\mathrm{m(x,y)} = \sqrt{fx(x,y)^2 + fy(x,y)^2} \quad (2.3)$$

$$\theta = tan^{-1}\left(\frac{fy(x,y)}{fx(x,y)}\right) \quad (2.4)$$

Setelah nilai magnitude dan orientasinya didapatkan, selanjutnya menentukan orientasi bin dengan jarak merata dari 0°-180° untuk gradien

"*undsigned*" atau 0°-360° untuk gradien "*signed*". Kemudian citra dibagi menjadi daerah spasial kecil berbentuk persegi dengan ukuran 8 x 8 yang disebut *cell*. Setiap piksel pada *cell* diletakkan dalam bin berdasarkan orientasi untuk membentuk histogram pada setiap *cell*.

Histogram besar dibentuk dari seluruh histogram yang dihasilkan oleh sebuah *block* yang telah digabungkan menjadi satu histogram. Histogram hasil kombinasi dinormalisasi dengan rumus :

$$V_i = \frac{v_i}{\sqrt{\|v\|_i^2 + \varepsilon^2}} \qquad (2.5)$$

Keterangan :

V_i = normalisasi

v_i = vektor fitur histogram yang digabungkan untuk satu block

$i = 1 \leq i \leq 27$ (3 cell x 9 bin)

ε = konstanta kecil bernilai 1 untuk menghindari pembagian nol

Setelah itu membuat vektor fitur berukuran 1-D dari histogram untuk setiap block yang telah dikumpulkan. Vektor fitur kemudian diletakkan dalam algoritma *learning* untuk menentukan citra wajah tersebut dikenali atau tidak berdasarkan data *training.*

2.2.5 *Euclidean Distance*

Euclidean distance adalah perhitungan jarak 2 buah titik bilangan *cartesian* pada n-ruang dalam *Euclidean space,* yang terdiri dari bilangan real (x1,x2,…,xn). Perhitungan Euclidean ini biasanya diterapkan pada 1, 2, dan 3 dimensi yang berkaitan dengan teorema *phytagoras*. Dengan nilai *euclidean distance* didapatkan dari persamaan berikut :

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \qquad (2.6)$$

Keterangan :

d = jarak euclidean

p dan q = jarak titik yang akan dihitung

2.2.6 Penggunaan Pustaka (*Library*) Dlib

Pustaka (library) Dlib merupakan *library cross platform open source* yang berisi algoritma *machine learning* untuk pembuatan perangkat lunak dalam memecahkan masalah dalam dunia nyata. Pustaka ini dikembangkan sejak tahun 2002 oleh Davis E. King dengan menggunakan bahasa pemrograman C++ yang difokuskan untuk menyediakan dukungan pengembangan perangkat lunak machine learning [15].

Dalam penelitian tugas akhir ini, pustaka Dlib diimplementasikan pada proses deteksi wajah, ekstraksi fitur, serta pengenalan wajah. Pada proses pendeteksian wajah menggunakan model *shape predictor 5 face landmarks* yang telah dilatih dan disediakan oleh Dlib. Pada proses ekstraksi fitur menggunakan metode *Histogram of Oriented Gradient* (HOG). Dan pada proses pengenalan wajah menggunakan perhitungan Euclidean Distance dari hasil ekstraksi fitur metode HOG dengan klasifikasi wajah teridentifikasi jika hasil *Euclidean distance* dengan jarak matriks vektor kurang dari 0,6, dengan implementasi pada aplikasi menggunakan model *dlib face recognition resnet model v1* yang telah dilatih dan disediakan oleh Dlib.

2.2.6.1 Tahapan Pendeteksian Wajah

Face detection atau pendeteksian wajah merupakan proses untuk mendeteksi wajah manusia pada teknologi komputer dengan menentukan ukuran dan posisi wajah manusia dalam sebuah citra digital. Teknologi ini dapat mendeteksi wajah melalui karakteristik wajah maupun sifat wajah manusia dengan mengabaikan objek lain seperti badan manusia itu sendiri [16]. Pada tahap pendeteksian wajah dengan Dlib ini menggunakan model *shape predictor 5 face landmarks* yang disediakan dan telah dilatih oleh Dlib.

2.2.6.2 Tahapan Ekstraksi Fitur dengan *Histogram of Oriented Gradient* (HOG)

Pada tahap ekstraksi fitur merupakan tahap penentuan ciri dari suatu citra yang digunakan untuk membedakan satu wajah dengan wajah lainnya. Pada proses ekstraksi fitur menggunakan *Histogram of Oriented Gradient* (HOG) ini akan dihitung gradien dan orientasi gradien kemudian dikonversi menjadi histogram yang akan menghasilkan sebuah vektor fitur citra. Dengan contoh perhitungan ekstraksi HOG sebagai berikut, dengan misal matriks A berdimensi 8 x 8 yang merepresentasikan citra wajah.

$$A = \begin{bmatrix} 49 & 148 & 172 & 61 & 104 & 39 & 70 & 83 \\ 153 & 51 & 87 & 111 & 83 & 98 & 40 & 175 \\ 56 & 122 & 59 & 76 & 83 & 55 & 100 & 91 \\ 90 & 42 & 54 & 47 & 45 & 71 & 83 & 39 \\ 77 & 101 & 71 & 153 & 102 & 165 & 43 & 102 \\ 82 & 89 & 134 & 172 & 65 & 65 & 55 & 62 \\ 114 & 74 & 48 & 53 & 87 & 56 & 101 & 160 \\ 107 & 39 & 73 & 66 & 57 & 88 & 170 & 153 \end{bmatrix}$$

Proses berikutnya yaitu menghitung gradien horizontal dan vertikal dari matriks tersebut menggunakan persamaan (2.1) untuk gradien horizontal dan persamaan (2.2) untuk gradien vertikal. Dengan contoh perhitungan menggunakan piksel 51 pada koordinat 1,1 sebagai berikut :

$$fx = 87 - 153 = -66$$

$$fy = 148 - 122 = 26$$

Sehingga didapatkan hasil perhitungan gradien horizontal (fx) dan gradien vertikal (fy) sebagai berikut :

$$fx = \begin{bmatrix} 65 & 123 & -87 & -68 & -22 & -34 & 44 & -21 \\ -124 & -66 & 60 & -4 & -13 & -43 & 77 & -113 \\ -31 & 3 & -46 & 24 & -21 & 27 & 36 & -44 \\ 3 & -36 & 5 & -9 & 24 & 38 & -32 & 7 \\ -1 & -6 & 52 & 31 & 12 & -59 & -63 & 34 \\ 27 & 52 & 83 & -69 & -107 & -10 & -3 & 27 \\ -86 & -66 & -21 & 39 & 3 & 14 & 104 & 13 \\ -114 & -34 & 27 & -16 & 22 & 113 & 65 & -63 \end{bmatrix}$$

$$fy = \begin{bmatrix} -46 & -12 & -14 & -45 & -26 & -10 & 130 & -22 \\ -7 & 26 & 113 & -15 & 21 & -16 & -30 & -8 \\ 63 & 9 & 33 & 64 & 38 & 27 & -43 & 136 \\ -21 & 21 & -12 & -77 & -19 & -110 & 67 & -11 \\ 8 & -47 & -80 & -125 & -20 & 6 & 28 & -23 \\ -37 & 28 & 23 & 100 & 15 & 109 & -58 & -58 \\ -25 & 50 & 61 & 106 & 8 & -23 & -115 & -91 \\ 65 & -74 & -124 & -8 & -17 & 17 & 31 & 77 \end{bmatrix}$$

Kemudian setelah didapatkan gradiennya, langkah berikutnya adalah menghitung magnitude (m) dengan persamaan (2.3) dan nilai *angle* (θ) dengan persamaan (2.4) seperti berikut :

$$\mathrm{m}(x,y) = \sqrt{(-66)^2 + (26)^2} = \sqrt{5032} = 70{,}93659$$

$$\theta = tan^{-1}\left(\frac{26}{66}\right) = 21{,}50143$$

Sehingga didapatkan magnitude (m) dan nilai *angle* (θ) dari hasil perhitungan yang akan digunakan untuk menentukan arah orientasi bin sebagai berikut :

$m =$

79,6304	124,1974	70,21396	21,2132	8,062258	45,80393	89,56004	131,2288
123,584	70,93659	9,486833	41,67733	47,38143	59,05929	82,80097	81,43709
88,11924	127,9414	56,61272	13	95,41488	86,12781	64,51356	126,9055
81,5414	15,52417	68,35203	77,52419	128,7866	121,4949	112,9469	17,88854
34,05877	24,69818	43,41659	30,61046	23,32381	108,0463	8,544004	27,80288
35,44009	45,88028	38,18377	116,3787	59,3043	109,4578	26,92582	114,2716
137,2443	82,63776	56,0803	74,24958	68,942	58,07753	155,0516	72,01389
30,41381	113,2828	142,9405	13,0384	41,04875	63,97656	91,92388	99,48869

$\theta =$

35,28675	3,23101	63,79989	81,8699	82,87498	53,88066	16,20902	29,69073
5,572198	21,50143	71,56505	30,25644	82,725	28,30076	37,14669	65,32314
9,14164	62,03288	35,65533	67,38014	56,97613	15,4885	71,00335	77,71604
33,49518	75,06858	69,44395	33,33334	76,07166	55,39432	69,80019	26,56505
49,76364	58,24052	61,07357	38,36749	59,03624	7,980114	69,44395	37,69424
16,38954	20,40988	45	70,9423	5,806727	84,75818	58,67131	8,555558
71,30102	21,28641	50,06362	64,4703	23,96249	87,03906	47,87545	25,49755
46,33222	4,049582	72,07208	57,52881	34,0772	65,03721	81,8699	50,71059

Magnitude				43,41659					
Bin	0	20	40	60	80	100	120	140	160

Dengan hasil orientasi bin yang telah ditentukan dari magnitude (m) dan nilai *angle* (θ) sebagai berikut :

Magnitude	94,19336	61,8945	62,5782	70,53195	43,63153	43,64399	142,9405	137,2443	86,48639
Bin	0	20	40	60	80	100	120	140	160

Dari hasil orientasi bin tersebut kemudian akan dilakukan normalisasi histogram menggunakan persamaan (2.5).

$$V_i = \frac{94{,}19336}{\sqrt{72150{,}51055}} = \frac{94{,}19336}{268{,}6084707} = 0{,}350671593$$

Sehingga didapatkan vektor hasil normalisasi yang telah dilakukan sebagai berikut :

V = [(0,350671593), (0,230426481), (0,232971813), (0,262582757), (0,162435414), (0,162481815), (0,532152045), (0,510945568), (0,321979385)]

Vektor histogram diatas dapat digunakan untuk klasifikasi dengan menghitung *euclidean distance* dari 2 vektor yang dihasilkan oleh wajah target dan wajah pengendara saat pemindaian wajah.

2.2.6.3 Tahapan Klasifikasi Pengenalan Wajah

Tahapan setelah ekstraksi fitur adalah klasifikasi pengenalan wajah dengan menghitung *Euclidean Distance* dari vektor yang dihasilkan dari ekstraksi fitur. Pada tahapan ini hanya akan membandingkan 2 wajah yaitu wajah target yang ada pada database dan wajah pengendara yang diambil saat pemindaian wajah. Dengan klasifikasi wajah yang dikategorikan sama adalah jika *Euclidean Distance* yang dihasilkan kurang dari 0,6 dan jika lebih dari 0,6 maka wajah tersebut dikategorikan tidak sama. Dengan implementasi pada aplikasi menggunakan model *dlib face recognition resnet model v1* yang telah dilatih dan disediakan oleh Dlib. Penentuan jarak *euclidean* 0,6 disini berdasarkan dari model yang digunakan dalam proses pengenalan wajah. Berikut ini diberikan 3 contoh vektor histogram, dengan vektor 1 sebagai wajah yang akan digunakan sebagai data wajah pembanding.

V_1 = [(0,045418263), (0,032463761), (0,097873388), (0,38916587), (0,053491575), (0,012272767), (0,03528628), (0,209349295), (0,124678802)]

V_2 = [(0,350671593), (0,230426481), (0,232971813), (0,262582757), (0,162435414), (0,162481815), (0,532152045), (0,510945568), (0,321979385)]

V_3 = [(0,008509616), (0,015356646), (0,277300638), (0,545070513), (0,071915353), (0,006340938), (0,005221022), (0,04533381), (0,024951463)]

Dengan menggunakan persamaan (2.6) dapat diketahui *euclidean distance* antara vektor 1 dengan vektor 2 sebagai berikut :

$$d(v_1, v_2) = \sqrt{0{,}577838641} = 0{,}760156985$$

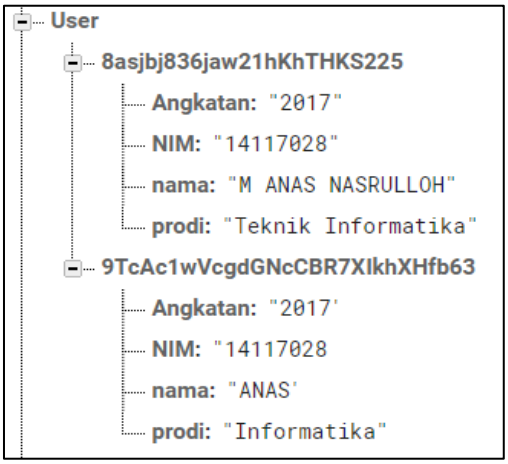
Dengan menggunakan persamaan (2.6) dapat diketahui *euclidean distance* antara vektor 1 dengan vektor 3 sebagai berikut :

$$d(v_1, v_3) = \sqrt{0{,}096280461} = 0{,}310290929$$

Dari hasil perhitungan *euclidean distance* yang telah dilakukan dapat disimpulkan bahwa vektor 3 dikategorikan sama dengan vektor 1 karena hasil *euclidean distance* < 0,6. Sedangkan vektor 2 dikategorikan tidak sama atau berbeda karena hasil *euclidean distance* > 0,6.

2.2.7 Firebase Realtime Database

Database atau basis data merupakan kumpulan berbagai data yang tersusun secara sistematik yang tersimpan di awan (*cloud*) dan dapat diolah, diperiksa atau dimanipulasi, untuk mendapatkan informasi dari basis data tersebut menggunakan bantuan program komputer. *Firebase Realtime Database* adalah basis data online sebagai media penyimpanan data dari aplikasi yang dapat digunakan untuk menyimpan data-data aplikasi dalam bentuk JSON dan dapat melakukan sinkronisasi secara *realtime* ke setiap *client* yang terhubung [17].



Gambar 2.3 Contoh firebase

Firebase Database merupakan basis data yang bersifat *non-relational* atau NoSQL yang memungkinkan untuk menyimpan berbagai tipe data seperti String, Long, dan Boolean [18]. Berbeda dengan basis data jenis SQL, pada basis data NoSQL tidak menggunakan sistem tabel yang berisi baris dan kolom dalam implementasinya serta data tidak disimpan secara lokal pada perangkat melainkan disimpan pada awan (*cloud*).

2.2.8 Android Studio

Android Studio merupakan Integrated Development Environment (IDE) resmi dari Google yang menggantikan Eclipse Android Development Tools (ADT) sebagai IDE untuk pengembangan aplikasi Android yang bersifat open source atau gratis [19]. Android studio diperkenalkan pada bulan Mei tahun 2013 yang dikembangkan berdasarkan Intellij IDEA yang memiliki kemiripan dengan Eclipse dan didesain khusus untuk pengembangan aplikasi Android. Android Studio dikembangkan menjadi lebih kompleks dan profesional, sebagai pengembangan dari Eclipse. Bahasa pemrograman Java adalah bahasa utama yang digunakan oleh Android Studio, sedangkan bahasa XML digunakan untuk merancang *layout* atau tampilan aplikasi yang akan dikembangkan.