

## **BAB 2**

### **LANDASAN TEORI**

Dalam menganalisa serta perancangan aplikasi berbasis mobile ini, diperlukan studi kepustakaan yang mendukung pembuatan skripsi. Landasan dan sumber teori dari studi kepustakaan ini sesuai dengan topik bahasan.

#### **2.1. Literatur Terkait**

##### **2.1.1. E-Voting**

###### **A. Voting**

Voting adalah pengambilan suatu keputusan dengan cara mengambil suara mayoritas.

###### **B. E-Voting**

E-voting adalah suatu sarana pemungutan suara dan perhitungan suara dalam pemilihan umum dengan menggunakan perangkat elektronik [9].

Barikut beberapa komponen *e-voting*[10] yaitu :

1. Orang yang dapat mengikuti pemilihan yaitu orang yang sah secara hukum untuk memberikan suara.
2. Pemilihan hanya dapat dilakukan 1 kali untuk masing - masing orang.
3. Pilihan orang lain tidak boleh diketahui oleh pihak lain.
4. Suara pemilih tidak boleh diduplikasikan.
5. Pilihan orang lain tidak boleh dirubah tanpa sepengetahuan orang tersebut.
6. Masing - masing orang dapat memastikan bahwa suaranya sudah masuk ke dalam tabulasi suara.
7. Masing - masing orang dapat mengetahui siapa yang sudah memilih dan tidak memilih.

## C. E-voting di Indonesia

Indonesia pernah menggunakan e-voting di Kabupaten Jembrana-Bali pada 54 kepala dusun Juli 2009 lalu. Proses e-voting bermula setelah verifikasi pemilih yang sudah terdaftar dalam daftar pemilih tetap (DPT). Pemilih akan mendatangi TPS membawa KTP untuk verifikasi kemudian jika pengguna terdaftar maka akan tampil nama dan foto pemilih. Kemudian Pemilih dapat memilih kandidat. Setelah semua pemilih memilih, maka pemenang langsung ditampilkan pada layar.

### 2.1.2. Android

Menurut Purwanto (2013), Android merupakan suatu perangkat lunak (*software*) yang digunakan pada perangkat berjalan (*mobile device*) yang meliputi sistem operasi, middleware dan aplikasi inti. Sistem ini bersifat *open source*.

#### A. Kotlin

Kotlin merupakan bahasa pemrograman yang diketik secara statis yang menargetkan *Java Virtual Machine*(JVM), Android, JavaScript, dan native[11]. Kotlin adalah bahasa pemrograman *open source* yang dikembangkan pada tahun 2010 oleh JetBrains. Kotlin memiliki beberapa fitur yaitu sebagai berikut :

##### 1. Kompatibilitas

Kotlin dapat berjalan pada perangkat Android versi lama dan kompatibel dengan JDK 6 serta Android build sistem.

##### 2. Performa

Dilengkapi dengan struktur *bytecode* yang mirip dengan Java membuat kotlin dapat berjalan dengan sama cepatnya.

##### 3. Interoperabilitas

Programmer dapat menggunakan semua *Android Library* dalam aplikasi, termasuk pengolahan anotasi, sehingga *data binding* dapat bekerja. Kotlin juga dapat beroperasi dengan Java.

##### 4. Footprint

Melalui penggunaan *ProGuard*, *runtime library* kotlin yang tersusun rapat dapat dikurangi sehingga *runtime* pada Kotlin

hanya menambahkan beberapa ratus metode dan kurang dari 100K untuk ukuran file.apk.

#### 5. Waktu kompilasi

Selain itu, kotlin mendukung kompilasi inkremental yang efisien, sehingga jika ada beberapa tambahan *overhead* untuk *clean build*, pembangunannya sama cepat atau lebih cepat daripada Java.

#### 6. *Learning curve*

Adanya fitur *Automated Java to Kotlin Converter* dan *Kotlin Koans* dalam plugin membantu pengembang Java untuk menggunakan Kotlin dengan mudah.

### **B. Android SDK (Software Development Kit)**

Android SDK adalah sebuah alat dari kode java yang dapat secara lancar dan diperbolehkan untuk bisa berjalan di perangkat android. Untuk menulis program digunakan java dan untuk menjalankan program digunakan android SDK. Penggabungan java dan Android SDK menciptakan Android Studio. Melalui aplikasi ini, programmer dapat menemukan bug pada aplikasi yang dikembangkannya.

### **C. Android JDK(Java Development Kit)**

Android JDK adalah sebuah aplikasi yang digunakan untuk mengkompilasi dari kode java menjadi bytecode. JDK yang biasanya digunakan yaitu Java SE Development Kit 8.

Sebuah JDK berisi JVM (Java Virtual Machine) yang berfungsi untuk mengubah bahasa program java menjadi bahasa mesin, hal ini dibutuhkan karena bahasa java merupakan bahasa pemrograman tingkat tinggi atau mendekati bahasa manusia akan dibutuhkan pengkonversi ke bahasa mesin dan JRE (Java Runtime Environment) yang berfungsi menjalankan aplikasi yang dibangun menggunakan java dan terdapat aplikasi yang dibutuhkan untuk proses pembuatan yang menggunakan kode pemrograman java.

### **D. Android Studio**

Menurut Android (2020), "Android Studio adalah IDE resmi untuk pengembangan aplikasi Android, yang didasarkan pada InteliJ

IDEA.” Android Studio berfungsi sebagai editor kode dan fitur developer IntelliJ. Fitur - fitur Android Studio lainnya, yaitu :

1. Gradle Sistem Build
2. Emulator yang kaya fitur
3. Tempat mengembangkan aplikasi untuk semua perangkat Android
4. Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
5. Integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel.
6. Alat pengujian yang lengkap dan framework
7. Alat untuk merekam performa, kegunaan, kompatibilitas versi dan masalah lainnya.
8. Fitur C++ dan NDK
9. Fitur Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine.

### **2.1.3. Firebase**

Firebase Realtime Database adalah database yang di-host di cloud dimana data akan disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Aplikasi lintas-platform dengan SDK Android, iOS dan Java Script memungkinkan klien berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis.



Gambar 2.1. Logo Firebase

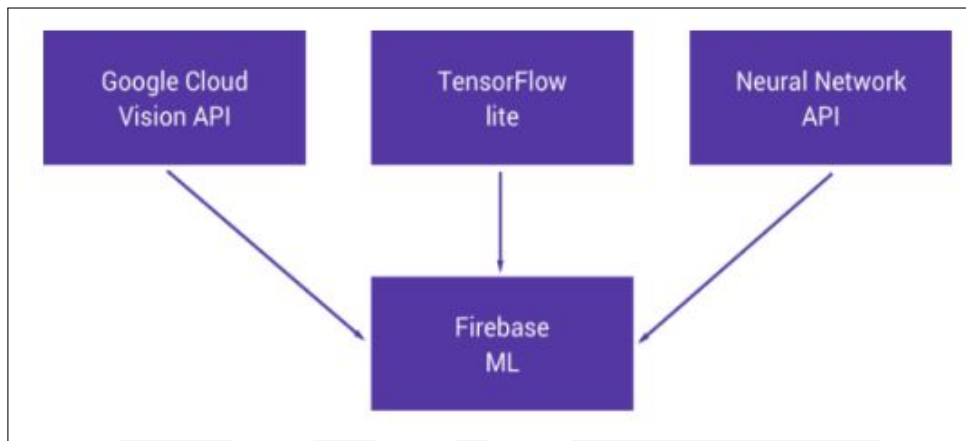
Dari gambar 2.1 yaitu Firebase memiliki fitur Firebase Realtime Database memungkinkan user untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien dan data tersimpan di lokal. Saat offline sekalipun, program bersifat responsif karena peristiwa realtime terus berlangsung. Realtime database akan menyinkronkan perubahan data lokal dengan update setiap koneksi perangkat pulih kembali sehingga setiap perbedaan akan otomatis digabungkan. Aturan keamanan Firebase Realtime Database merupakan fitur firebase yang dapat menentukan metode stukturisasi data dan kapan data dapat dibaca atau ditulis.

Firestore memiliki beberapa fitur diantaranya yaitu;

1. Analytics, berfungsi untuk mendapatkan insight tentang perilaku pengguna.
2. Authentication untuk menyiapkan alur autentikasi pengguna.
3. Cloud Messaging untuk mengirimkan notifikasi
4. Cloud Real-time Database atau Cloud Firestore berfungsi untuk menyimpan data seperti informasi pengguna.
5. Cloud Storage, berfungsi untuk menyimpan file seperti foto dan video
6. Cloud Functions yaitu untuk memicu kode backend yang berjalan di lingkungan aman.
7. Crashlytics yaitu untuk mengetahui kapan dan mengapa aplikasi mengalami error.

#### **2.1.4. ML kit Firebase**

Menurut Google Firebase, ML Kit adalah SDK Seluler yang menghadirkan keahlian machine learning Google untuk aplikasi Android dan iOS dalam paket yang andal dan mudah digunakan. ML Kit menyediakan API praktis yang dapat membantu user menggunakan model TensorFlow Lite kustom di aplikasi seluler.



Gambar 2.2. Diagram MLKit[1]

Dari gambar 2.2 terlihat beberapa fitur ML Kit yang dapat digunakan untuk mengenali teks, mendeteksi wajah, mengidentifikasi bangunan terkenal, memindai barcode, melabeli gambar dan mengidentifikasi bahasa teks. Di perangkat atau di cloud, berbagai API untuk ML Kit dapat dijalankan dan memproses data dengan cepat dan berfungsi bahkan ketika tidak ada koneksi jaringan.

API berbasis cloud juga memanfaatkan kecanggihan teknologi machine learning dari Google Cloud untuk memberikan tingkat akurasi yang tinggi. Selain API ML Kit, model TensorFlow Lite juga dapat digunakan. Model hanya perlu diupload ke Firebase dan Firebase akan menangani proses hosting dan menanyangkannya ke aplikasi sebagai lapisan API untuk model kustom, sehingga lebih mudah dijalankan dan digunakan. Teknologi ML dari google seperti Google Cloud Vision API, TensorFlow Lite, dan Android Neural Networks API yang terdapat dalam SDK ML Kit. ML Kit dapat digunakan untuk memperoleh berbagai hal seperti pemrosesan berbasis cloud, kemampuan real-time pada model di perangkat yang dioptimalkan untuk seluler, atau fleksibilitas model TensorFlow Lite kustom. Berikut merupakan fitur yang tersedia di perangkat dan di cloud yang dapat dilihat dari gambar 2.3.

Fitur	Di perangkat	Cloud
Pengenalan teks	✓	✓
Deteksi wajah	✓	
Pemindaian kode batang	✓	
Pelabelan gambar	✓	✓
Deteksi & pelacakan objek	✓	
Pengenalan bangunan terkenal		✓
Identifikasi bahasa	✓	
Terjemahan	✓	
Smart Reply	✓	
Inferensi model AutoML	✓	
Inferensi model kustom	✓	

Gambar 2.3. Perangkat vs Cloud

### 2.1.5. Alur Implementasi

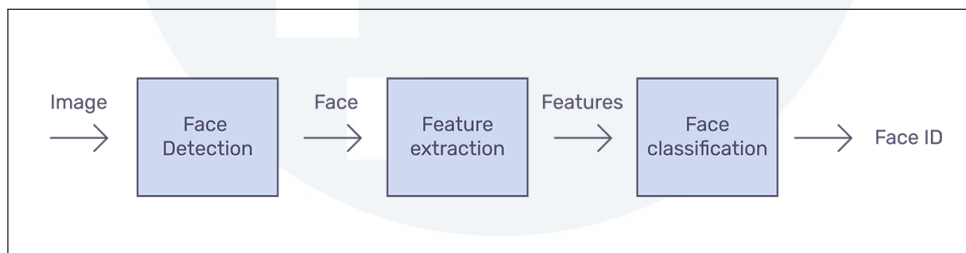
Dalam penggunaan ML Kit, kita perlu mengikuti alur implementasi sebagai berikut :

1. Mengintegrasikan SDK. Pertama kita perlu menyertakan SDK dengan menggunakan gradle.
2. Mempersiapkan data input. Jika menggunakan fitur vision. ambil gambar dari kamera dan buat metadata yang diperlukan, seperti rotasi gambar, atau minta pengguna untuk memilih foto dari galeri mereka.
3. Menerapkan model ML ke data. Penerapan model ML ke data, menghasilkan insight misalnya keadaan emosi wajah yang terdeteksi atau objek dan konsep yang dikenali dalam gambar, bergantung pada fitur yang digunakan. Insight berfungsi untuk mendukung fitur di aplikasi, seperti aksesoris foto, pembuatan metadata otomatis dan seterusnya.



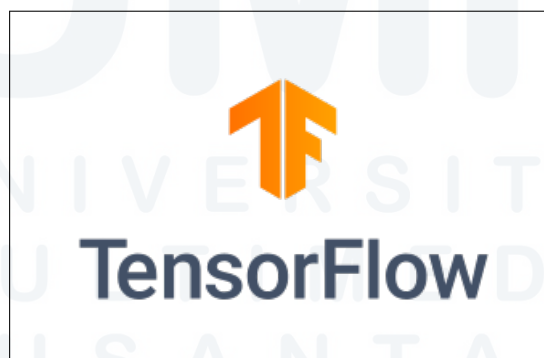
### 2.1.6. Face Detection dan Recognition

*Face Recognition* adalah sistem pengenalan wajah manusia yang dapat digunakan dalam bidang keamanan *security system* seperti akses masuk ruangan, pengawasan lokasi *surveillance*, dan pencarian identitas individu pada database kepolisian. Selain itu, face recognition juga digunakan dalam identifikasi pelaku kejahatan, pengembangan sistem keamanan, pemrosesan citra maupun film, dan interaksi manusia komputer. *Face Recognition* juga merupakan aktivitas *Computer Vision* untuk mengidentifikasi dan memverifikasi seseorang berdasarkan foto diri mereka. Pengenalan wajah terdiri atas 4 tahap yaitu deteksi, *alignment*, ekstraksi fitur dan pencocokan seperti yang terlihat pada gambar 2.4.



Gambar 2.4. Face Recognition Pipeline[2]

### 2.1.7. Tensorflow



Gambar 2.5. Logo Tensorflow

Berdasarkan laman tensorflow.org[12], Tensorflow adalah sebuah framework komputasional untuk membuat model *machine learning* yang menyediakan berbagai *toolkit* yang memungkinkan pembuatan model pada



tingkat abstraksi yang pengguna kehendaki. Selain itu, Tensorflow juga membantu membuat *neural network* yaitu jaringan artifisial yang mirip dengan otak manusia dalam skala besar[13]. Ada dua komponen yang dimiliki oleh tensorflow yaitu:

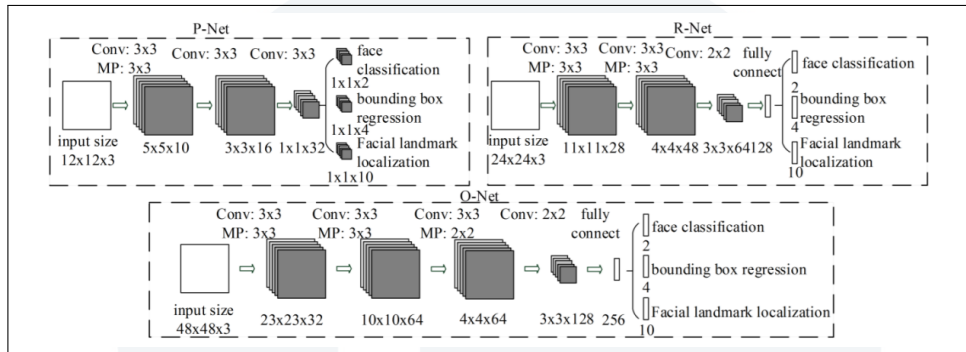
1. *Protocol Buffer*(.pb) yang berisikan grafik dan model untuk menjalankan model yang telah diuji coba.
2. Waktu proses yang menjalankan grafik tersebut.

*protocol buffer* dan waktu proses bersifat analog terhadap kode *Python* dan penerjemah *Python*. Tensorflow juga merupakan penerjemah kode pada beberapa platform hardware seperti CPU, GPU dan TPU. Maka dari itu, Tensorflow membuat pembuatan *software* dapat dilakukan dengan cepat karena adanya kemudahan dalam menggunakan dan mempelajarinya.

Tensorflow memiliki beberapa fitur[14] yang dapat diimplementasikan dalam penelitian yaitu :

1. Mengoptimalkan, mendefinisikan dan menghitung secara efisien ekspresi matematis yang melibatkan *array multidimension (tensors)*
2. Mendukung program *machine learning neural network*
3. Menggunakan GPU yang mengoptimalkan penggunaan memori. Tensorflow dapat mengenali penulisan kode yang sana dan menjalankannya di CPU atau GPU serta perhitungan mana yang perlu dipindahkan ke GPU.
4. Memiliki skalabilitas komputasi yang tinggi di seluruh mesin dan kumpulan data yang besar.

### 2.1.8. Multi-task Cascaded



Gambar 2.6. Arisektur MTCNN [3]

*Multi-task Cascaded Convolutional Neural Network (MTCNN)* yaitu salah satu dari metode *Deep Convolutional Neural Network* yang digunakan untuk mengenali atau mendeteksi wajah. Ada 3 jaringan terpisah pada MTCNN seperti terlihat pada gambar 2.6 yaitu

#### 1. Proposal Network (P-Net)

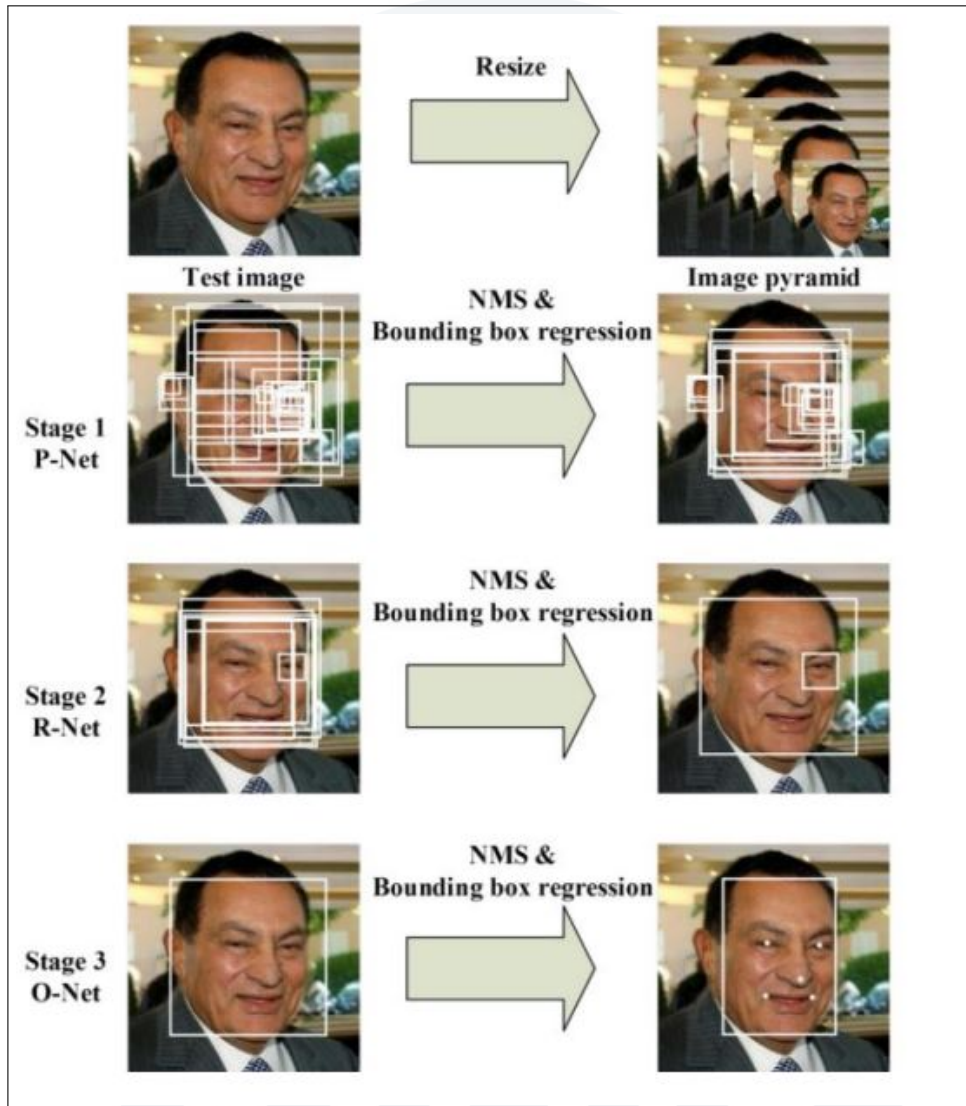
Pada jaringan ini setiap gambar akan ditelusuri oleh kernel 12x12 untuk mencari wajah secara horizontal sampai sudut bawah gambar.

#### 2. Refine Network (R-Net)

Sama halnya dengan P-Net, maka pada jaringan RNet, gambar akan ditelusuri untuk mencari wajah secara horizontal. Namun pada jaringan ini wajah akan dideteksi akan menggunakan kernel 24x24

#### 3. Output (O-Net)

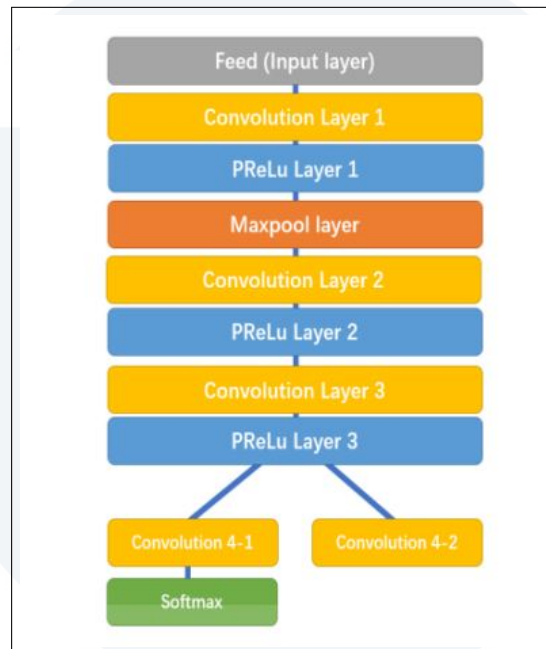
Pada jaringan ini, gambar akan ditelusuri oleh kernel 48x48 untuk mendeteksi wajah.



Gambar 2.7. Deteksi dan penyalarsan wajah MTCNN [3]

Pada gambar 2.7 terlihat proses pendeteksi dan pengenalan wajah dengan ketiga jaringan yang ada pada MTCNN. Terdapat tiga tahapan konvolusi yang dijalankan dengan 3x3 kernel dalam masing - masing 12x12 kernel pada jaringan P-Net. Selanjutnya, lapisan Parametric Rectified Linear Unit(pReLU) akan menghilangkan *vanishing gradient* dengan cara menerapkan fungsi aktivasi element. Jaringan akan terbagi menjadi dua lapisan setelah lapisan konvolusi ketiga. Kemudian, lapisan ketiga yang diaktivasi akan dibagi ke dua lapisan konvolusi yang terpisah dan lapisan *softmax* yang memeberikan probabilitas desimal untuk setiap hasil dan akan bertambah hingga satu. Probabilitas terdiri atas dua yaitu probabilitas bahwa

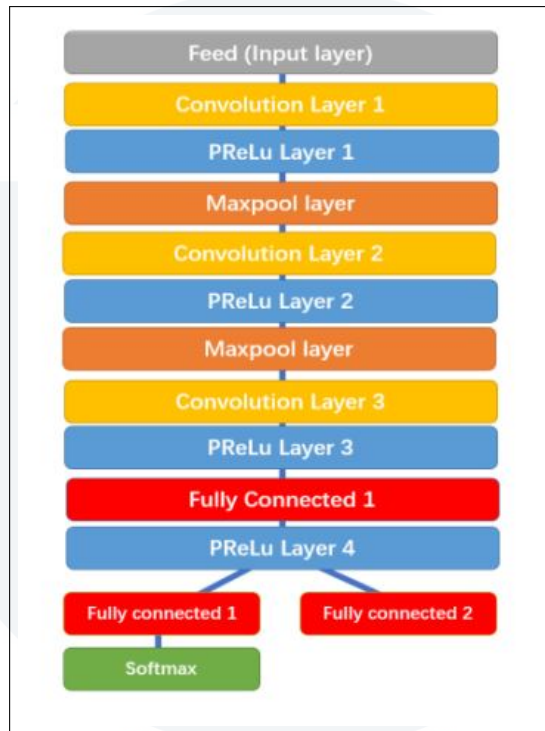
ada wajah di daerah tersebut dan probabilitas bahwa tidak ada wajah. Cara kerja MTCNN pada jaringan P-Net dapat terlihat pada gambar.



Gambar 2.8. MTCNN pada jaringan P-Net [3]

Probabilitas suatu wajah berada di dalam kotak *bounding box* yang dihasilkan oleh konvolusi 4-1 dan koordinat *bounding box* dihasilkan oleh konvolusi 4-2. Pada gambar 2.8 terlihat R-Net memiliki struktur yang serupa namun lebih banyak lapisan menggunakan vektor *bounding box* P-Net sebagai inputnya untuk menyempurnakan koordinat *bounding box*.

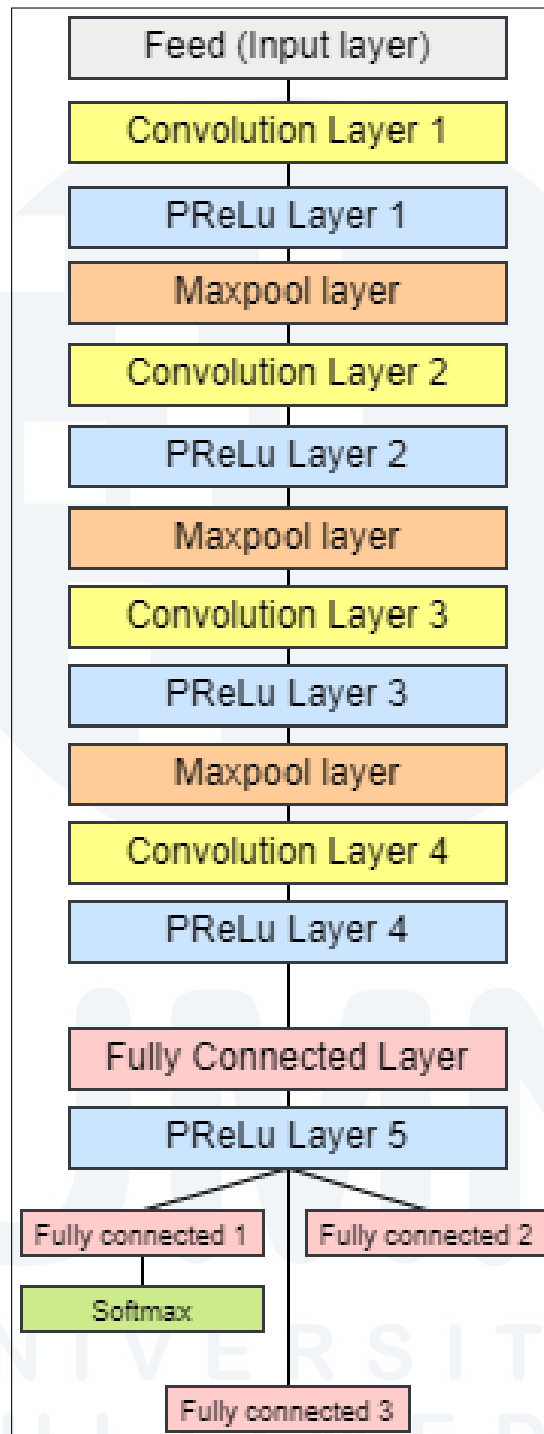
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.9. MTCNN pada jaringan R-Net [3]

Pada gambar 2.9 terlihat jaringan R-net yang terbagi menjadi 2 lapisan pada ujungnya dengan output *Fully Connected 1* berisi probabilitas suatu wajah di *bounding box* dan *Fully Connected 2* berisi koordinat *bounding box* yang baru dan menambah tingkat kepastian deteksi pada setiap *bounding box*.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.10. MTCNN pada jaringan O-Net [3]

Selanjutnya, jaringan O-Net mengambil *bounding box* R-Net sebagai input dan bertugas menandai koordinat landmark wajah seperti yang terlihat pada gambar 2.10. Terdapat tiga lapisan diujung jaringan O-Net dengan 3 output yang berbeda yaitu *Fully Connected 1* yaitu probabilitas wajah di *bounding box*, *Fully Connected 2* yaitu koordinat *Bounding Box* dan *Fully*

*Connected 3* yaitu koordinat landmark wajah yang berisikan lokasi mata, hidung dan mulut.

Dalam proses MTCNN, pertama-tama gambar akan diubah ukurannya berkali-kali untuk mendeteksi wajah dengan ukuran yang berbeda-beda. Kemudian P-Net akan memindai gambar untuk melakukan deteksi wajah yang pertama, Proses ini memiliki *threshold* yang rendah untuk pendeteksian dan mendeteksi banyak *false positives*, walaupun setelah dilakukan NMS(Non-Maximum Supression) yaitu algoritma yang dikembangkan untuk memilih *bounding box* terbaik ketika hasil objek deteksi mengalami *overlapping*[15].

P-Net yang memiliki banyak *false positives* akan menjadi *input* untuk jaringan kedua yaitu R-net. Pada lapisan ini, R-net akan melakukan filtrasi deteksi dengan NMS untuk menghasilkan *bounding boxes* yang akurat. Lapisan terakhir yaitu O-Net akan melakukan perbaikan terakhir *bounding box* ini. Sehingga hasilnya akan menghasilkan *bounding box* yang akurat dalam mendeteksi wajah.

Menurut Medium.com[16], sistem pendeteksi wajah yang baik harus memiliki karakteristik berikut :

1. Cepat. Waktu idealnya secara *real-time* yaitu minimal diatas 1 FPS
2. Akurat. Artinya sistem hanya boleh mendeteksi wajah (tidak ada *false positives*) dan mendeteksi semua muka (*no false negatives*)
3. Kokoh. Wajah harus terdeteksi dengan pose- pose, rotasi dan pencahayaan yang berbeda.
4. Memanfaatkan semua *resources* yang ada seperti menggunakan GPU jika memungkinkan atau warna RGB.

Fitur tambahan dari MTCNN yaitu pendeteksian *facial landmark* seperti mata, hidung, dan ujung mulut. MTCNN juga sangat akurat dan kokoh karena dapat mendeteksi wajah dengan baik dalam skala, penerangan dan rotasi yang berbeda. Waktu yang dibutuhkan juga terbilang cepat dengan menggunakan CPU. Selain itu MTCNN juga memiliki fitur informasi warna karena CNN menerima inputan gambar RGB. Maka dari itu dalam penelitian ini akan digunakan MTCNN untuk pendeteksian wajah.



### 2.1.9. FaceNet

FaceNet merupakan sistem *face recognition* yang dikembangkan oleh peneliti google dengan hasil akurasi yang cukup tinggi. Pada implementasinya, *Deep Convolutional Neural Network* akan mengekstraksi fitur wajah menjadi vektor. Vektor ini juga disebut dengan *Vector Embedding* yang dihasilkan dapat memetakan kemiripan wajah yang memiliki kedekatan posisi pada *embedding space*. FaceNet menggunakan model Deep CNN berupa ZF-et atau Inception[17].

FaceNet akan diimplementasikan pada aplikasi E-Voting berbasis *mobile device* pada tahapan *feature learning* . Vektor yang dihasilkan FaceNet yaitu sebanyak 128 elemen atau *Face embedding* diklasifikasikan menggunakan Linear Support Vector Machine (SVM) yang akan mengklasifikasikan identitas wajah dari *vector embedding*.

*Face Embedding* adalah vektor yang merepresentasikan fitur yang diekstraksi dari wajah. Selanjutnya, fitur ini akan dibandingkan dengan vektor yang dihasilkan untuk wajah lain. Vektor dengan ukuran yang dekat mungkin orang yang sama, sedangkan vektor lain yang jauh mungkin orang yang berbeda. Model klasifikasi yang dikembangkan akan mengambil input *face embedding* dan memprediksi identitas wajah. Model FaceNet akan menghasilkan embedding untuk gambar wajah tertentu dan digunakan sebagai bagian dari klasifikasi itu sendiri. Selain itu, model FaceNet untuk melakukan pra-proses wajah untuk membuat *embedding* wajah yang dapat disimpan dan digunakan sebagai masukan untuk model pengklasifikasian. Metode terakhir ini lebih direkomendasikan karena model FaceNet besar dan lambat untuk membuat embedding wajah.

### 2.1.10. Euclidian Distance

Metode *Euclidian Distance* merupakan salah satu teknik yang digunakan dalam menentukan tingkat kesamaan (*similarity degree*) atau ketidaksamaan (*disimilarity degree*) menggunakan metode jarak antar dua vektor. Berikut merupakan rumus Euclidian yaitu akar dari kuadrat perbedaan 2 vektor.

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.1)$$

$d$  = jarak Euclidian

$p, q$  = dua titik pada  $n$ -space Euclidian

$q_i, p_i$  = Vektor Euclidian, titik awal vektor

$n$  =  $n$ -space

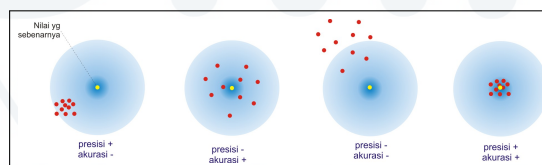
Jarak Euclidian selalu lebih besar atau sama dengan nol. Jika hasil pengukuran adalah nol maka kedua vektor identik. Namun jika hasil pengukuran tinggi maka kedua vektor tidak indentik[18].

### 2.1.11. *Optical Character Recognition*

*Optical Character Recognition*(OCR) yaitu pengenalan karakter alphanumeric dari karakter tulisan tangan atau file maupun citra menjadi teks yang dapat diedit. OCR menurut Yadav(2013) adalah proses untuk mengkonversi dokumen cetak atau tulisan tangan hasil scan menjadi karakter ASCII yaitu karakter yang dapat dibaca oleh mesin[19].

### 2.1.12. *Precision, Recall dan Accuracy*

*Precision* merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Sedangkan *recall* merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Selain itu, dalam pengenalan pola juga dikenal istilah *accuracy* yaitu tingkat kedekatan antara nilai prediksi dengan nilai aktual.



Gambar 2.11. *Akurasi vs Presisi*

Dari gambar 2.11 [20] terlihat gambaran perbedaan antara *accuracy* dan *precision*.

Penggunaanya adalah sebagai berikut. Jika kita ingin menguji akurasi dari sebuah mesin pengenalan wajah yang bertugas memisahkan orang A dari semua wajah orang yang telah didapat. Kemudian, untuk pengujian kita akan memasukkan 1 wajah orang A dan 9 wajah orang lain.

Lalu hasilnya mesin mendeteksi 1 yang mengenali sebagai orang A.

$$precision = \frac{jumlah\ orang\ terdeteksi\ benar}{jumlah\ orang\ yang\ dipisahkan} = \frac{1}{1} = 100\%$$

$$recall = \frac{jumlah\ orang\ terdeteksi\ benar}{jumlah\ orang\ yang\ terdeteksi\ sebenarnya} = \frac{1}{1} = 100\%$$

$$precision = \frac{jumlah\ yang\ terdeteksi\ orang\ A + Jumlah\ yang\ terdeteksi\ selain\ A}{jumlah\ total\ orang\ yang\ didata} = \frac{1 + 10}{11} = 100\%$$

Untuk melakukan perhitungan akurasi pada konsep data mining maka digunakan metode *confusion matrix*. Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada pengukuran kinerja menggunakan *confusion matrix*

		Nilai sebenarnya	
		TRUE	FALSE
Nilai prediksi	TRUE	TP (True Positive) Correct result	FP (False Positive) Unexpected result
	FALSE	FN (False Negative) Missing result	TN (True Negative) Correct absence of result

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Gambar 2.12. Rumus akurasi, presisi dan recall

Pada kasus pendeteksian wajah berlaku keterangan seperti pada gambar 2.12 yaitu :

- True Positif (TP) adalah gambar wajah yang benar terdeteksi sebagai wajah.
- False Positif (FP) adalah gambar bukan wajah tetapi dideteksi sebagai wajah.
- False Negatif (FN) adalah gambar wajah yang tidak terdeteksi sebagai wajah.
- True Negatif (TN) adalah gambar bukan wajah dan tidak terdeteksi sebagai wajah.

$$Precision = \frac{TP}{TP + FP} \quad (2.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.3)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

		Nilai Sebenarnya	
		TRUE	FALSE
Nilai Prediksi	TRUE	1	0
	FALSE	0	11

Gambar 2.13. Menentukan nilai precision, recall, accuracy

$$precision = \frac{TP}{TP + FP} = \frac{1}{1 + 0} = 100\%$$

$$recall = \frac{TP}{TP + FN} = \frac{1}{1 + 0} = 100\%$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 10}{11} = 100\%$$