

BAB 2

TINJAUAN PUSTAKA

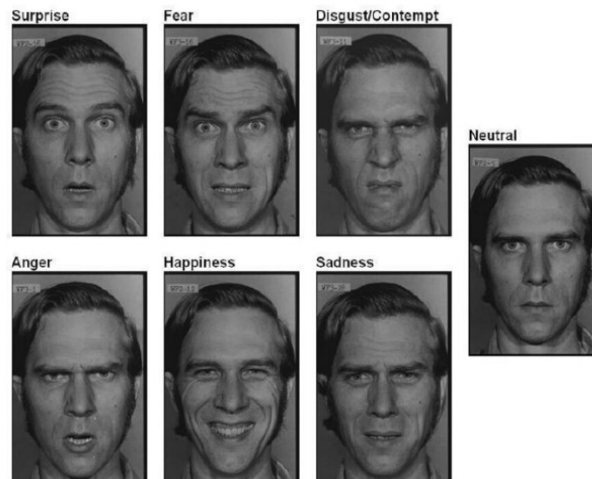
2.1 Landasan Teori

Pada bagian ini akan dibahas landasan teori yang menjadi dasar dalam pengembangan aplikasi *Moodify*, yaitu aplikasi rekomendasi musik berdasarkan emosi dengan analisis ekspresi wajah. Landasan teori ini mencakup konsep-konsep utama yang berkaitan dengan pengenalan ekspresi wajah, emosi manusia, serta teknologi kecerdasan buatan yang digunakan, khususnya *Deep Learning*.

2.1.1 Ekspresi Wajah

Emosi adalah salah satu perasaan yang menyebabkan seseorang bereaksi atau bertindak sebagai respon. Kita dapat mengenali emosi seseorang dengan memperhatikan dan mengamati ekspresi wajahnya secara spesifik. Emosi bisa dikenali dari perubahan ekspresi wajah, seperti kerutan di dahi atau kedipan mata. Penelitian ini menunjukkan bahwa ekspresi wajah adalah bentuk komunikasi *nonverbal* yang sangat penting, sering kali lebih cepat dan langsung dibandingkan kata-kata (Syaufina dkk., 2024).

Pengenalan ekspresi wajah merupakan bidang yang mempelajari bagaimana mengidentifikasi emosi manusia dari ekspresi wajah mereka. Ekspresi wajah adalah bentuk komunikasi *nonverbal* yang penting dan dapat memberikan informasi berharga tentang keadaan emosional seseorang. Dengan perkembangan teknologi kecerdasan buatan yang pesat, pengenalan ekspresi wajah secara otomatis telah dipelajari secara intensif dalam beberapa tahun terakhir. *Facial Expression Recognition* (FER) telah menerima perhatian banyak dari bidang psikologi, *computer vision*, dan *pattern recognition*. FER memiliki penerapan yang luas dalam berbagai bidang, misalnya *computer interaction*, *virtual reality*, *augmented reality*, *advanced driver assistance system*, dan pendidikan. Pendekatan untuk pengenalan ekspresi wajah dibagi menjadi dua pendekatan yaitu *Conventional FER Approaches* dan *Deep Learning-Based FER Approaches* (Kusdiananggalih dkk., 2021).

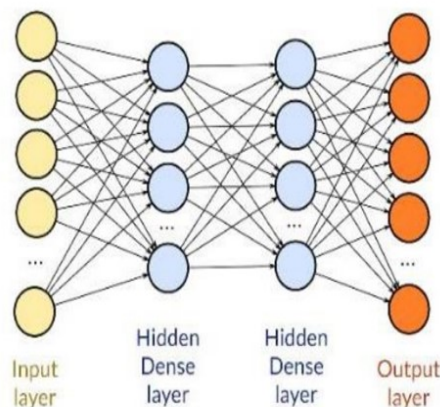


Gambar 2.1 Ekspresi Dasar Manusia (Adicius, 2022)

2.1.2 Deep Learning

Deep Learning merupakan bidang turunan dari *machine learning* (ML) yang memungkinkan komputer untuk belajar berdasarkan *experience* atau pengalaman dan mengerti dunia dipandang dari segi hierarki suatu konsep. Sehingga *deep learning* ini akan melakukan pembelajaran dengan sendiri dari pengetahuan yang kita berikan dan menghasilkan sebuah hierarki dalam bentuk lapisan yang banyak dan dalam atau *deep* (Nugroho dkk., 2020). Ada beberapa jenis algoritma dari *deep learning* yaitu:

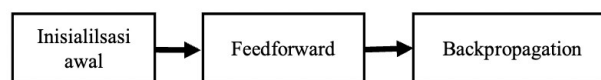
1. *Convolutional Neural Network* (CNN)
2. *Recurrent Neural Network* (RNN)
3. *Long Short Term Memory Network* (LSTM)
4. *Self Organizing Maps* (SOM)



Gambar 2.2 Konsep Pengerjaan *Deep Learning* (Syaufina dkk., 2024)

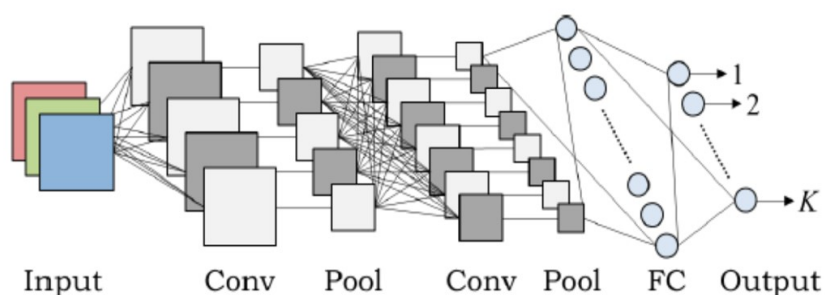
2.1.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu algoritma dari *deep learning* yang banyak digunakan pada data citra, seperti gambar atau video langsung dari kamera. CNN digunakan untuk menganalisis gambar visual, mendeteksi dan mengenali objek pada gambar, berdimensi tinggi yang akan melibatkan banyak parameter untuk mencirikan jaringan (Nugroho dkk., 2020). Tahapan-tahapan yang akan dilakukan pada proses CNN meliputi Inisialisasi Awal, *Feedforward*, dan *Backpropagation* (Reynaldo, 2020).



Gambar 2.3 Tahapan CNN (Reynaldo, 2020)

CNN memiliki variasi arsitektur yang berbeda-beda. Dapat terdiri dari *Convolutional Layer*, *Pooling Layer*, dan *Fully-Connected Layer*. Dan bisa juga terdiri dari arsitektur yang lebih detail yaitu *Input*, *Convolutional Layer*, *ReLU Layer*, *Pooling Layer*, *Flatten Layer*, *Fully Connected Layer* dan *Softmax Layer*.



Gambar 2.4 Contoh Arsitektur CNN (Kusdiananggalih, 2022)

Secara umum CNN dibagi menjadi dua tahapan yaitu:

1. Ekstraksi Fitur, didalamnya terdapat beberapa lapisan dan tiap lapisan saling terhubung. Lapisan pertama adalah *Convolution Layer* + *ReLU Layer* dan lapisan kedua adalah *Pooling Layer*. Setiap lapisan memiliki masing-masing fungsi. Tahap ini memproses *input* citra menjadi *output* berupa vektor untuk diolah pada lapisan berikutnya.
2. Klasifikasi, didalamnya terdapat *Fully connected layer*. Tahap ini menerima hasil *input*, hasil *output* pada lapisan ekstraksi fitur. Pada lapisan *Flatten layer*

terjadi proses perubahan metrik dua dimensi menjadi satu dimensi. Dan terakhir yaitu lapisan *Softmax Layer* akan mengubah hasil *neural network* menjadi probabilitas terhadap kelas yang dideteksi.

Kesimpulannya CNN merupakan metode untuk mentransformasikan nilai *pixel* pada citra menjadi nilai terhadap kelas untuk diklasifikasi.

2.1.4 ML Kit Face Detection

ML Kit adalah *Software Development Kit (SDK) machine learning* dari *Google* yang memungkinkan pengembang menerapkan fitur kecerdasan buatan pada aplikasi *mobile* secara *on-device*, yaitu pemrosesan dilakukan langsung di perangkat tanpa memerlukan koneksi ke server. Salah satu fitur yang disediakan adalah *Face Detection*, yaitu kemampuan mendeteksi keberadaan, posisi, dan kotak pembatas wajah pada sebuah gambar (*Google*, 2026).

Pada penelitian ini, *ML Kit Face Detection* digunakan untuk mendeteksi dan memotong area wajah dari foto yang diambil pengguna sebelum gambar diproses oleh model deteksi emosi. Proses *cropping* ini penting karena model dilatih menggunakan citra wajah yang sudah terpotong rapat (*dataset FER2013*), sehingga *input* berupa wajah yang sudah terpotong akan menghasilkan prediksi yang lebih akurat dibandingkan memproses keseluruhan foto yang masih memuat latar belakang.

2.1.5 TensorFlow

TensorFlow adalah *library* yang bertujuan untuk mengerjakan pembelajaran mesin dan jaringan syaraf dalam penelitiannya. Fitur utama yang terdapat dalam *tensorflow* adalah:

1. Mendefinisikan, mengoptimalkan, dan menghitung operasi matematis menggunakan struktur data multidimensi yang disebut *tensor*.
2. Pemakaian GPU (*Graphics Processing Unit*) yang efisien, mengotomasi manajemen dan mengoptimalkan memori yang sama terhadap data yang dipakai. *Tensorflow* mampu menulis kode yang sama dan menjalankannya di CPU atau GPU. Lebih khususnya lagi *tensorflow* dapat mengetahui bagian mana yang harus dipindahkan ke GPU.

Untuk kebutuhan *deployment* pada perangkat *mobile*, model *TensorFlow* yang telah dilatih dikonversi ke format *TensorFlow Lite* (TFLite). TFLite merupakan versi ringan dari *TensorFlow* yang dioptimalkan untuk berjalan secara efisien pada perangkat dengan sumber daya terbatas seperti *smartphone*. Dengan TFLite, proses prediksi emosi dapat dilakukan secara *on-device* tanpa memerlukan koneksi internet, sehingga lebih cepat dan menjaga privasi pengguna.

2.1.6 Keras

Keras adalah sebuah pustaka *open-source* yang ditulis dalam *Python* dan dirancang untuk memudahkan pengembangan dan pelatihan model *deep learning*. *Keras* berfungsi sebagai *API* tingkat tinggi yang dapat berjalan di atas beberapa *backend* seperti *TensorFlow*, CNTK (*Microsoft Cognitive Toolkit*), atau *Theano* (Arianto dkk., 2024).

Keras menawarkan kemudahan dalam membangun, melatih, dan menyebarkan model jaringan saraf. Salah satu keunggulan utama dari *Keras* adalah kemampuannya untuk mengarahkan proses pembuatan model dengan menyediakan antarmuka yang pencitraan dan modular. Hal ini memungkinkan pengguna untuk dengan cepat mendefinisikan model arsitektur, menambahkan lapisan-lapisan yang diperlukan, serta mengkompilasi model dengan fungsi *loss* dan *optimizer* yang sesuai.

2.1.7 Confusion Matrix

Confusion matrix adalah metode untuk melakukan prediksi pada kategori yang berbeda pada suatu *classifier*. Setelah model diuji akan dihasilkan suatu persentase yang menunjukkan tingkat keakuratan. Tingkat keakuratan ini dapat diukur dengan membandingkan ekspresi wajah yang berhasil dikenali dengan benar dengan jumlah keseluruhan data pengujian. *Confusion matrix* menghitung akurasi dengan menghitung kumulatif dari jumlah prediksi yang benar dibagi dengan jumlah dari prediksi yang benar ditambah dengan prediksi yang salah (Darmawan dkk., 2022). Perhitungan dirumuskan sebagai berikut:

$$Akurasi = \frac{\sum \text{data uji yang dikenali}}{\sum \text{data uji}} * 100\% \quad (2.1)$$

Evaluation Model adalah tahap untuk mendapatkan model terbaik dari yang telah dirancang. Ada lima parameter yang digunakan pada penelitian ini, yaitu

akurasi, *loss*, presisi, *recall*, dan F1- score. Berikut ini merupakan persamaannya:

1. *Accuracy* (ketepatan) merupakan parameter yang mengukur seberapa tepat model dalam melakukan prediksi secara keseluruhan. Parameter ini menghitung perbandingan antara jumlah prediksi yang benar (baik positif maupun negatif) dengan total keseluruhan data. Semakin tinggi nilai *accuracy*, semakin baik model dalam melakukan klasifikasi. Nilai *accuracy* dihitung menggunakan persamaan.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.2)$$

2. *Precision* (presisi) fokus pada mengukur ketepatan model dalam memprediksi kasus positif. Parameter ini menghitung proporsi dari kasus yang diprediksi positif yang benar-benar merupakan kasus positif. *Precision* sangat penting ketika kita ingin menghindari *false positive*, yaitu situasi di mana model salah mengidentifikasi kasus negatif sebagai positif. Selanjutnya, tingkat *precision* (presisi) digunakan untuk melihat seberapa akurat model dalam mengidentifikasi kasus positif dari seluruh prediksi positif yang dilakukan. Nilai *Precision* dihitung menggunakan persamaan.

$$Precision = \frac{TP}{TP + FP} \quad (2.3)$$

3. Kemampuan model dalam mengenali data positif dapat diukur melalui nilai sensitivitas (*recall/sensitivity*), yang menunjukkan proporsi kasus positif yang berhasil diidentifikasi dengan benar. Nilai *Recall* dihitung menggunakan persamaan.

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

4. F1-Score memberikan nilai tunggal yang menyeimbangkan antara *precision* dan *recall*. Parameter ini sangat berguna ketika kita membutuhkan keseimbangan antara *precision* dan *recall*, terutama pada *dataset* yang tidak seimbang. *F1-Score* merupakan rata-rata harmonik yang memberikan bobot yang sama antara *precision* dan *recall*. Nilai F1-Score dapat dihitung menggunakan persamaan.

$$F1\ Score = 2 \times \frac{(Recall \times Presisi)}{(Recall + Presisi)} \times 100 \quad (2.5)$$

5. *Loss* adalah metrik yang digunakan untuk mengukur seberapa baik model dalam memprediksi target yang benar selama pelatihan. *Loss* berfungsi sebagai fungsi kerugian yang menghitung kesalahan antara prediksi model dan target yang sebenarnya. Semakin rendah nilai *loss*, semakin baik model dalam melakukan prediksi. *Loss* menjadi indikator kinerja model selama proses pelatihan dan pengembangan. Nilai *Loss* dapat dihitung menggunakan persamaan.

$$Loss = \sum_{i=1}^n = p(x_i) \log q(x_i) \quad (2.6)$$

2.1.8 Spotify Web API

Spotify Web API merupakan antarmuka pemrograman aplikasi berbasis *web* yang disediakan secara resmi oleh *Spotify* untuk memungkinkan pengembang mengakses data dan layanan dari katalog musik *Spotify*. Melalui API ini, aplikasi kita dapat memperoleh berbagai informasi seperti data lagu, artis, dan album beserta metadata lainnya, termasuk judul lagu, nama artis, dan gambar sampul album. *Spotify Web API* bekerja menggunakan protokol HTTP dengan format pertukaran data berupa JSON, sehingga mudah diintegrasikan ke dalam berbagai *platform*, termasuk aplikasi *Android* (Lesmana, A., & Setiawan, E. B., 2021).

Untuk dapat mengakses *Spotify Web API*, aplikasi terlebih dahulu harus melalui proses autentikasi untuk memperoleh *access token*. Pada penelitian ini digunakan metode autentikasi *Client Credentials Flow*, yaitu skema autentikasi yang memanfaatkan *Client ID* dan *Client Secret* dari aplikasi yang telah didaftarkan pada *Spotify Developer Dashboard*. Metode ini sesuai digunakan untuk mengakses data publik seperti pencarian lagu tanpa memerlukan proses login akun pengguna. Setelah memperoleh *token*, aplikasi dapat memanggil *endpoint* pencarian untuk mengambil daftar lagu berdasarkan kata kunci tertentu, yang pada aplikasi *Moodify* diperoleh dari pemetaan emosi yang terdeteksi ke genre atau suasana musik yang sesuai.

Namun, *Spotify Web API* memiliki keterbatasan, yaitu pemutaran lagu secara penuh hanya dapat dilakukan melalui aplikasi resmi *Spotify* dan memerlukan akun

Spotify Premium, sehingga pada penelitian ini pemutaran lagu dilakukan dengan mengarahkan pengguna ke aplikasi *Spotify* melalui *deep link*.

2.1.9 Purposive Sampling

Purposive sampling merupakan salah satu teknik pengambilan sampel dalam metode *non-probability sampling*, yaitu teknik pemilihan sampel yang dilakukan berdasarkan pertimbangan atau kriteria tertentu yang telah ditetapkan oleh peneliti sesuai dengan tujuan penelitian. Teknik ini digunakan ketika peneliti memerlukan responden yang memiliki karakteristik atau pengalaman tertentu sehingga mampu memberikan informasi yang relevan terhadap objek penelitian (Sarfo dkk., 2022).

Menurut Sarfo dkk. (2022), *purposive sampling* memungkinkan peneliti memilih responden yang dianggap paling sesuai untuk memberikan data yang dibutuhkan. Oleh karena itu, teknik ini banyak digunakan pada penelitian pengembangan sistem, penelitian terapan, maupun penelitian yang berfokus pada evaluasi suatu produk atau aplikasi, karena responden dipilih berdasarkan kesesuaiannya dengan tujuan penelitian, bukan secara acak.

Pada penelitian ini, *purposive sampling* digunakan dalam pengujian *usability* aplikasi *Moodify*. Responden dipilih berdasarkan beberapa kriteria yang telah ditentukan, yaitu memiliki *smartphone* berbasis *Android*, bersedia mencoba seluruh fitur aplikasi, serta bersedia mengisi kuesioner setelah menggunakan aplikasi. Dengan menggunakan teknik *purposive sampling*, diharapkan data yang diperoleh dapat memberikan gambaran yang lebih relevan mengenai tingkat kemudahan penggunaan aplikasi sesuai dengan tujuan penelitian.

2.2 Penelitian Terdahulu

Penelitian mengenai deteksi ekspresi wajah sudah pernah dibuat sebelumnya. Pertama dilakukan oleh Ihsan dkk. (2021). Penelitian ini bertujuan untuk mendeteksi ekspresi wajah menggunakan CNN dengan *library TensorFlow*. Model ini dirancang untuk mengenali tujuh ekspresi dasar wajah manusia, yaitu *Angry*, *Disgust*, *Fear*, *Happy*, *Sad*, *Surprise*, dan *Neutral*. *Dataset* yang digunakan adalah FER-2013 yang terdiri dari 35.887 citra *grayscale* berukuran 48x48 piksel. Data dibagi menjadi 28.704 citra untuk pelatihan dan 7.184 citra untuk validasi. Hasil penelitian menunjukkan akurasi pelatihan sebesar 62,24% dan akurasi validasi

sebesar 62,44%. Pada pengujian dengan data asing sebanyak 3.589 citra, model mencapai akurasi sebesar 77%.

Penelitian kedua dilakukan oleh Ghaffar dkk. (2022). Penelitian ini bertujuan untuk mengembangkan sistem deteksi emosi berbasis ekspresi wajah pada aplikasi *Android* menggunakan metode CNN. *Dataset* yang digunakan terdiri dari beberapa ribu citra wajah dengan berbagai ekspresi, meskipun jumlah spesifik tidak disebutkan. Model ini berhasil mendeteksi emosi dengan akurasi yang baik untuk aplikasi *real-time*, tetapi nilai akurasi spesifik tidak dijelaskan dalam sumber.

Penelitian ketiga dilakukan oleh Muttaqin dkk. (2021). Penelitian ini menggunakan CNN untuk mengenali ekspresi wajah manusia dari citra yang beragam. *Dataset* yang digunakan mencakup citra wajah dengan tujuh kategori ekspresi, namun jumlah data spesifik tidak disebutkan. Hasil penelitian menunjukkan bahwa model dapat mengenali ekspresi wajah dengan akurasi yang cukup tinggi dalam pengujian *real-time*.

Penelitian keempat dilakukan oleh Sihombing dkk. (2023). Penelitian ini bertujuan untuk mendeteksi ekspresi wajah menggunakan arsitektur CNN *MobileNetV2*. *Dataset* yang digunakan berasal dari *Kaggle*, terdiri dari berbagai ekspresi wajah seperti *Happy*, *Sad*, *Angry*, *Fear*, *Surprise*, dan *Neutral*. Meskipun jumlah data spesifik tidak disebutkan, hasil penelitian menunjukkan akurasi yang baik dalam pengujian model, meskipun nilai akurasi tidak dijelaskan lebih lanjut.

Penelitian kelima dilakukan oleh Kusdiananggalih dkk. (2022). Penelitian ini bertujuan untuk mengenali ekspresi wajah dari *cross-dataset* menggunakan CNN. *Dataset* yang digunakan mencakup citra dari ras *non-Indonesia* sebagai data latih dan citra dari ras Indonesia sebagai data uji. Hasil penelitian ini menunjukkan akurasi prediksi sebesar 91,29% pada data uji, dengan sensitivitas dan presisi yang juga mencapai 91,29%.

Tabel 2.1 Penelitian Terdahulu

Kategori	Ihsan dkk. (2021)	Ghaffar dkk. (2022)	Muttaqin dkk. (2021)	Sihombing dkk. (2023)	Kusdiananggalih dkk. (2022)
Judul	Deteksi Ekspresi Wajah Menggunakan <i>TensorFlow</i>	Implementasi Algoritma <i>Convolutional Neural Network</i> untuk Pengenalan	A Robust System for Facial Emotions Recognition Using <i>Convolutional</i>	Pengenalan Ekspresi Wajah dari <i>Cross Dataset</i> Menggunakan <i>Convolutional</i>	Pengenalan Ekspresi Wajah menggunakan <i>Convolutional Neural Network</i> (CNN)

Kategori	Ihsan dkk. (2021)	Ghaffar dkk. (2022)	Muttaqiin dkk. (2021)	Sihombing dkk. (2023)	Kusdiananggalih dkk. (2022)
		Ekspresi Wajah 2	<i>Neural Network</i>	<i>Neural Network</i>	
Metode	<i>Convolutional Neural Network</i> (CNN) dengan <i>TensorFlow</i>	CNN dengan tiga lapisan konvolusi, optimasi <i>ADAM</i> , batch size 32, dan augmentasi data.	<i>Convolutional Neural Network</i> (CNN) untuk aplikasi <i>Android</i> .	<i>Convolutional Neural Network</i> (CNN)	<i>MobileNet V2</i> (varian CNN)
Dataset	The Facial Expression Recognition 2013 (FER-2013).	Dataset dikumpulkan sendiri, mencakup tujuh macam ekspresi wajah manusia dari individu Indonesia.	Kombinasi dari JAFFE dan KDEF.	Menggunakan <i>dataset</i> public yang sudah ada.	The Facial Expression Recognition 2013 (FER-2013).
Jumlah Data	28.704 data untuk proses <i>training</i> dan 7.184 data untuk <i>testing</i> . Pengujian dilakukan pada 3.589 data asing.	Tidak disebutkan secara spesifik.	Tidak disebutkan secara spesifik, namun <i>dataset</i> JAFFE dan KDEF masing-masing berisi 213 dan 4.900 citra.	Tidak disebutkan secara spesifik.	Tidak disebutkan secara spesifik.
Hasil	Akurasi pengujian 77%.	Akurasi data training 70,16%, Akurasi data testing 64,43%, Akurasi tertinggi pada data uji 67% setelah 100 <i>epoch</i> .	Mencapai akurasi maksimum sebesar 78,1% pada data uji.	Akurasi sebesar 91,29% pada data uji.	Akurasi pengenalan ekspresi wajah sebesar 78% pada data uji.

Berdasarkan kajian terhadap beberapa penelitian terdahulu pada Tabel 2.1, diketahui bahwa berbagai penelitian telah berhasil mengembangkan model deteksi ekspresi wajah menggunakan metode CNN dengan pendekatan yang beragam. Salah satu pendekatan utama adalah implementasi CNN menggunakan *library TensorFlow*, seperti yang dilakukan oleh Ihsan dkk. (2021), yang berhasil mencapai akurasi pengujian sebesar 77% pada data asing menggunakan *dataset*

FER-2013. Penelitian lain, seperti yang dilakukan oleh Kusdiananggalih dkk. (2022), berhasil meningkatkan akurasi hingga 91,29% dengan menggunakan metode *cross-dataset*. Pada penelitian yang dilakukan oleh Muttaqin dkk. (2021) terdapat akurasi data *testing* 64,43% yang bisa disebabkan oleh *dataset* yang dikumpulkan sendiri dan juga model CNN yang hanya memiliki tiga *layer* konvolusi. Dari hasil penelitian yang ada pada tabel diatas, rentang akurasi yang dihasilkan pada data uji antara 67% hingga 91,29%.

Dengan mempertimbangkan data tersebut, akurasi yang layak untuk *deployment* adalah diatas 75%. Namun, untuk memastikan kinerja yang optimal, akurasi diatas 80% akan lebih baik. Pada penelitian ini, model yang dikembangkan akan menggunakan CNN dan *library TensorFlow* untuk mendeteksi ekspresi wajah. Berbeda dengan penelitian yang tercantum dalam tabel, penelitian ini tidak hanya berfokus pada akurasi deteksi ekspresi wajah tetapi juga mengintegrasikan hasil deteksi tersebut dengan *API Spotify*. Sistem ini akan menampilkan rekomendasi musik dari *Spotify* yang sesuai dengan ekspresi wajah pengguna, lalu mengarahkan pengguna untuk memutarinya melalui aplikasi *Spotify*.