

Penggunaan Teknologi *Face Recognition* Untuk Mengklasifikasi Emosi Anak TK Berbasis Website

*Abizer Aprido Seka¹⁾, Wahyuni²⁾, Pitrasacha Adytia³⁾

^{1,2,3)}Teknik Informatika, STMIK Widya Cipta Dharma

Jl. M. Yamin, Gn. Kelua, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur 75123

Email: Benuaquetam@gmail.com

ABSTRAK

Anak-anak Taman Kanak-kanak (TK) merupakan masa perkembangan tahap awal untuk mereka mengenal dan peka terhadap lingkungan, di mana tidak hanya belajar secara akademis tetapi juga secara emosional dan sosial. Emosi memainkan peran penting dalam pengalaman anak-anak ini. Serta peran penting teknologi face recognition telah menarik perhatian sebagai alat yang dapat mengidentifikasi dan mengklasifikasi ekspresi emosi seseorang dengan akurasi tinggi. Tujuan penelitian ini membuat *prototype* sistem klasifikasi emosi sederhana. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *prototyping*. Bahasa pemrograman atau software pendukung yang digunakan untuk membuat sistem klasifikasi emosi terdiri dari visual studio code, python, html, dan bootstrap. Menggunakan modul dari *TensorFlow* untuk membaca data berupa gambar dari *Kaggle* yang berjumlah 21.077 dan data lapangan yang berjumlah 201 yang sudah dibagi menjadi 4 parameter yaitu *Sad*, *Happy*, *Anger* dan *Neutral*. Yang kemudian akan diolah menggunakan oleh algoritma *CNN* untuk membuat model. Hasil dari Penelitian berupa model yang dibuat dari algoritma *CNN* dengan akurasi sebesar 75% serta sistem ini berhasil mendeteksi emosi dengan faktor-faktor seperti emosi yang terdeteksi masih belum akurat atau gagal mendapatkan emosi yang sesuai.

Kata Kunci: Anak, *Face Recognition*, Emosi, Website

ABSTRACT

Children's Kindergarten (TK) is an early stage of development for them to recognize and be sensitive to the environment, where they learn not only academically but also emotionally and socially. Emotions play an important role in these children's experiences. As well as the important role of face recognition technology has attracted attention as a tool that can identify and classify a person's emotional expression with high accuracy. The purpose of this research is to create a simple emotional classification system prototype. The system development method used in this research is the prototyping method. The programming language or supporting software used to create an emotional classification system consists of visual studio code, python, html, and bootstrap. Using a module from TensorFlow to read Kaggle's image data of 21.077 and field data of 201 already divided into four parameters: Sad, Happy, Anger and Neutral. Which will then be processed using the CNN algorithm to make a model. The results of the research were models made from the CNN algorithm with an accuracy of 75% and the system was able to detect emotions with factors such as emotions that were detected were still not accurate or failed to get the appropriate emotions.

Keywords: Children, *Face recognition*, Emotions, Website

1. PENDAHULUAN

Anak-anak Taman Kanak-kanak (TK) merupakan masa perkembangan tahap awal untuk mereka mengenal dan peka terhadap lingkungan, di mana tidak hanya belajar secara akademis tetapi juga secara emosional dan sosial. Emosi memainkan peran penting dalam pengalaman anak-anak ini. Bagaimana mereka bisa memengaruhi seberapa baik mereka belajar dan bahkan kesejahteraan mereka secara keseluruhan. Pentingnya memahami dan mengelola emosi pada anak TK telah menjadi fokus bagi para pendidik dan orang tua. Ini karena lingkungan emosional yang positif dapat meningkatkan motivasi belajar, meningkatkan keterampilan sosial, dan membantu anak-anak mengatasi tantangan dengan baik.

Dalam beberapa tahun terakhir, teknologi *face recognition* telah menarik perhatian sebagai alat yang dapat mengidentifikasi dan mengklasifikasi ekspresi emosi seseorang dengan akurasi tinggi. Penerapan teknologi ini telah menunjukkan potensi besar dan

inovatif dalam berbagai konteks, termasuk pendidikan anak-anak. Akan tetapi tidak banyak orang tua ataupun pendidik pada saat ini tahu bahwa memahami emosi anak-anak usia dini itu penting dan menganggap hal tersebut termasuk perihal yang tidak perlu untuk diperhatikan. Bagi anak-anak yang berada di masa usia tumbuh kembang serta sensitif terhadap kebutuhan emosional perlu adanya bimbingan atau tuntunan yang mengarahkan mereka agar dapat memahami dan paham kalau emosi itu penting serta harus diarahkan ke hal yang bersifat positif. Dengan demikian maka diperlukan untuk melakukan penelitian terkait penggunaan *face recognition* untuk mengklasifikasi emosi anak-anak TK, hal tersebut dapat dijadikan acuan pada penelitian ini ialah bagaimana cara pemanfaatan teknologi *face recognition* terhadap klasifikasi emosi anak TK guna menciptakan lingkungan yang lebih responsif dan mendukung bagi perkembangan emosi mereka serta mempermudah pendidik dan orang tua dalam memahami pentingnya tentang kebutuhan emosional anak-anak dan cara mengambil langkah-

langkah yang sesuai untuk mendukung perkembangan mereka.

Berdasarkan permasalahan diatas maka pada penelitian ini akan dirancang sistem klasifikasi emosi dengan mengimplementasikan *face recognition* yang menjadi metode pendeteksi wajah dengan dikombinasikan dengan OpenCV yang mempunyai fungsi sebagai *library computer vision*. Sistem ini akan terhubung dengan webcam untuk menangkap gambar ketika terjadi perubahan ekspresi emosi pada anak-anak TK. Selanjutnya, hasil deteksi tersebut akan diterapkan ke dalam sebuah *platform website* yang berfungsi sebagai antarmuka aplikasi bagi pengguna.

2. RUANG LINGKUP

Agar penelitian ini tidak menyimpang dan masih terarah sesuai latar belakang, maka didapatkan beberapa batasan masalah yang akan membantu penelitian ini hingga akhir. Batasan masalah tersebut dijabarkan dalam beberapa poin seperti berikut :

1. Sistem yang digunakan menggunakan Bahasa Pemrograman *Python*.
2. Menggunakan *Website* sebagai *interface* dan aplikasi konfigurasi programan *face recognition*.
3. Penelitian ini akan difokuskan pada klasifikasi emosi utama yaitu senang, marah, sedih dan netral.
4. Menggunakan dataset training yang berasal dari internet.
5. Menggunakan dataset test yang berasal dari pengumpulan langsung dari lapangan.
6. Menggunakan *webcam* sebagai media penangkap gambar atau video wajah
7. Sistem yang digunakan bersifat *real time*.
8. Sistem masih bersifat *prototype*
9. Output yang dihasilkan berupa penilaian emosi yang sesuai.
10. Sistem tidak memiliki fitur login atau akses admin.
11. Sistem ini hanya berfokus untuk mengklasifikasi emosi dan memberikan akurasi yang sesuai dan tidak memberikan suatu rekomendasi.

3. BAHAN DAN METODE

3.1 Face Recognition

Facial Recognition atau yang lebih dikenal dengan *Face Recognition* merupakan suatu teknologi sistem pengenalan atau pendeteksi wajah yang mampu memverifikasi serta mengidentifikasi wajah seseorang melalui citra digital. Menurut Aرسال, dkk (2020), adapun tahapan proses *face recognition* yang dimulai dari pengambilan gambar hingga indentifikasi gambar yaitu :

1. Pengumpulan data/akuisisi gambar

Proses Pengambilan atau penginputan citra digital maupun kamera yang dilanjutkan ketahap *preprocessing*.

2. Preprocessing

Proses penyesuaian citra input dengan cara melakukan normalisasi *size* citra, *median filtering* buat menghilangkan noise dampak pergeseran frame atau kamera, *histogram equalization* untuk memudahkan proses pengenalan citra dengan memperbaiki kualitas citra tanpa menghilangkan data utamanya, *high pass filtering* untuk membuat sisi tepi suatu citra, *background removal* buat menghilangkan bagian background

sehingga bagian wajah saja yang diproses dan *grayscaleing* buat mengkonversi citra *RGB* sebagai citra skala abu-abu. *Preprocessing* dilakukan buat menghilangkan persoalan yang akan ada pada saat proses *face recognition*.

3. Ekstrasi fitur

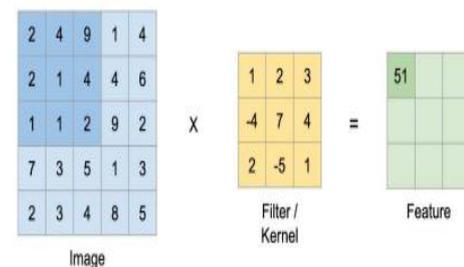
Ekstrasi fitur adalah tahapan lanjutan setelah *preprocessing* yaitu tahap membuat objek pembeda terhadap objek yang lain agar mempermudah proses pengelompokkan objek-objek yang akan berlangsung pada tahap klasifikasi.

4. Klasifikasi

Proses pengelompokkan objek yang sudah dilakukan ekstrasi fitur atau membuat objek yang berbeda yang akan digunakan untuk mempermudah dalam tahap identifikasi.

5. Identifikasi

Tahap terakhir yaitu mencocokkan objek yang mirip kedalam kelompok yang sudah diklasifikasikan ke dalam dataset yang sama.



3.2 Emosi

emosi adalah perasaan atau afeksi yang muncul. Sikap yang menunjukkan (mengekspressikan) kenyamanan atau ketidaknyamanan dari situasi atau hubungan tersebut dikenal sebagai emosi. (Solikhah, dkk, 2014)

3.3 Anak Usia Dini (Anak-anak TK)

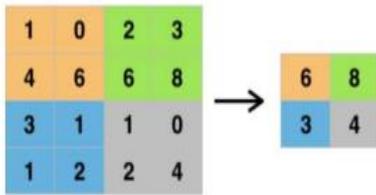
Definisi anak usia dini yang dikemukakan oleh National Assosiation Education for Young Chlidren (NAEYC) ialah sekelompok individu yang berada pada rentang usia antara 0 – 8 tahun. Anak usia dini merupakan sekelompok manusia yang berada pada proses pertumbuhan serta perkembangan. Secara tradisional pemahaman tentang anak tak jarang diidentifikasi sebagai manusia dewasa mini, masih polos serta belum bisa apa-apa atau menggunakan kata lain belum mampu berfikir. Pemahaman lain perihal anak usia dini ialah anak adalah insan kecil yang memiliki potensi yang masih wajib dikembangkan. (Priyanto, 2014)

3.4 Klasifikasi

Klasifikasi ialah sebuah proses untuk mengelompokkan barang-barang yang dianggap memiliki kecenderungan-kesamaan tertentu. Proses klasifikasi dapat dilakukan pada dua tahap, yaitu tahap belajar dari data pelatihan dan klasifikasi kasus baru. pada proses belajar, algoritma klasifikasi akan membuat data training untuk membentuk sebuah model. setelah model diuji dan dapat diterima, selanjutnya model tersebut akan dipergunakan untuk memprediksi kelas dari kasus baru untuk membantu proses pengambilan Keputusan. (Sosodoro, dkk, 2014)

3.5 Website

Website merupakan halaman informasi atau halaman URL berisi teks, suara animasi, gambar ataupun video yang dimana cara mengaksesnya harus terkoneksi dengan



meskipun objek gambar mengalami translasi (pergeseran).

Gambar 2.3 Max Pooling

internet. Sedangkan Menurut Yumarlin MZ, (2016), Website adalah suatu lokasi atau alamat di internet yang menampung halaman web. Umumnya, halaman web ini terdiri dari dokumen HTML dan bisa berisi berbagai elemen seperti foto, gambar grafis, musik, teks, dan bahkan animasi.

3.6 Algoritma CNN

Menurut Trisiawan and Yuliza (2022), CNN adalah satu dari beberapa macam model yang ada pada deep learning. Struktur CNN terdiri dari feature learning layer dan fully connected layer. Feature learning layer dalam CNN terdiri dari beberapa layer, yaitu layer konvolusi, fungsi aktivasi (ReLU), max pooling dan dropout regularization. CNN bekerja secara hierarki, sehingga output pada layer konvolusi pertama digunakan sebagai input pada layer konvolusi selanjutnya.

3.6.1 Layer Konvolusi

Layer konvolusi merupakan layer pertama dalam arsitektur CNN. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan gambar, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada gambar di semua bagian yang memungkinkan seperti yang ditunjukkan pada Gambar 2 Kotak biru secara keseluruhan adalah gambar input. Kernel bergerak dari sudut kiri atas ke kanan bawah sehingga hasil konvolusi dari gambar tersebut berupa feature map (kotak hijau).

Gambar 2.2 Operasi Konvolusi

$$G[m, n] = \sum_j \sum_k h[j, k] f[m - j, n - j]$$

Dimana :

f = input

h = kernel

m = baris pada feature map

n = kolom pada feature map

3.6.2 Fungsi aktivasi (ReLU)

ReLU adalah salah satu jenis fungsi aktivasi yang bersifat non-linear. Nilai output dari neuron bisa dinyatakan sebagai 0 jika inputnya adalah negatif. Jika nilai input adalah positif, maka output dari neuron adalah nilai input aktivasi itu sendiri. Adapun persamaan untuk adalah sebagai berikut.

$$F(x) = \max(0, x)$$

3.6.3 Max Pooling

Max pooling merupakan operasi yang membagi output dari layer konvolusi menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks gambar yang telah direduksi seperti pada Gambar 3, grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama

3.6.4 Dropout Regularization

Dropout merupakan sebuah teknik regulasi jaringan syaraf tiruan dengan tujuan memilih beberapa neuron secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain neuron-neuron tersebut dinonaktifkan secara acak.

3.6.5 Fully Connected layer

Fully connected layer merupakan layer dimana semua neuron aktivasi dari layer sebelumnya terhubung semua dengan neuron di layer selanjutnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu. Fungsi dari layer ini adalah untuk menyatukan semua node menjadi satu dimensi.

3.6.6 Binary Cross-Entropy

Binary Cross-Entropy merupakan salah satu jenis loss function yang dipergunakan untuk mengetahui nilai loss dari sebuah model jaringan klasifikasi, terutama untuk klasifikasi biner. Binary Cross-Entropy memberikan nilai loss untuk setiap probabilitas klasifikasi binernya. Pada Binary Cross-Entropy,

komponen vektor dari output berdiri sendiri dan tidak saling terikat satu sama lain, artinya loss yang dihitung untuk setiap komponen vektor keluaran CNN tidak dipengaruhi oleh komponen vektor lainnya. Maka dari itu Binary Cross-Entropy digunakan untuk klasifikasi multi-label, dimana nilai dari kelas tertentu tidak mempengaruhi kelas lainnya. Adapun persamaannya adalah sebagai berikut:

$$Loss = \frac{1}{N} \sum_1^N = 1y_i \cdot \log(y^{\wedge}_i) + (1 - y_i) \log(y^{\wedge}_i)$$

Dimana:

Y^{\wedge}_i = output neuron

y_i = target

3.6.7 Sigmoid

Sigmoid merupakan salah satu fungsi aktivasi non-linear dimana pada fungsi aktivasi ini akan menerima input berupa angka tunggal dan mengubah nilai tersebut menjadi sebuah nilai yang memiliki range mulai dari 0 sampai 1. Adapun persamaan untuk sigmoid adalah sebagai berikut.

$$f(x_i) = \frac{1}{1 + e^{-x_i}}$$

Dimana :

X_i = nilai input

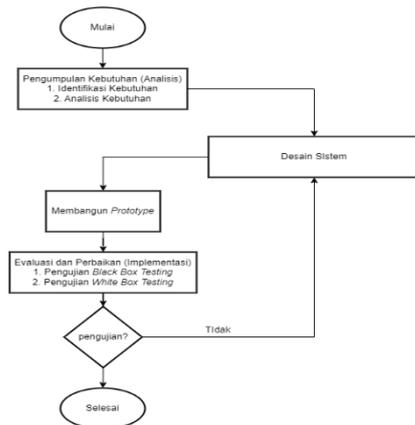
3.7 Tensorflow

Menurut Febrywinata (2024), Tensorflow Merupakan framework open-source yang dikembangkan oleh Google. Tensorflow ini digunakan untuk mengembangkan dan melatih berbagai model yang ada di machine learning, deep learning, serta pekerjaan yang berkaitan dengan analisis statistik lainnya.

3.8 OpenCV

OpenCV merupakan salah satu library yang sering dipergunakan pada pengolahan citra computer vision yang memanfaatkan sebuah Application Programming Interface (API) dimana OpenCV memungkinkan komputer bisa melihat seperti manusia berdasarkan vision

tersebut. komputer bisa membuat keputusan, melakukan



aksi serta mengenali terhadap suatu objek berdasarkan deteksi wajah. (Isum, dkk, 2019)

3.9 Prototype

Menurut Purnomo (2017), Prototyping ialah metode pengembangan perangkat lunak, yang berupa model fisik kerja sistem dan berfungsi menjadi versi awal yang berasal dari sistem. Dengan menggunakan metode prototyping ini, prototype sistem akan dibuat untuk membantu pengembang dan pengguna berinteraksi dalam proses pengembangan sistem informasi.

metode pengembangan sistem berbasis prototype yang terdapat tahapan atau Langkah-langkah dalam proses pengembangan dan akan dijelaskan sebagai berikut :

1. Pengumpulan Kebutuhan (Analisis)

Mengumpulkan kebutuhan melibatkan pertemuan antara pengembang dan pelanggan untuk menentukan keseluruhan tujuan dibuatnya perangkat lunak; mengidentifikasi kebutuhan berupa garis besar kebutuhan dasar dari sistem yang akan dibuat. Maka diperlukan penilaian kebutuhan awal dan analisa tentang ide atau gagasan untuk membangun ataupun mengembangkan sistem. Analisis dilakukan untuk mengetahui komponen apa saja pada sistem yang sedang berjalan, dapat berupa *hardware*, *software*, jaringan dan pemakai sistem sebagai *level* pengguna akhir sistem. Langkah selanjutnya adalah mengumpulkan informasi yang dibutuhkan pengguna akhir yang meliputi biaya dan manfaat sistem yang dibangun ataupun dikembangkan.

2. Desain Sistem

Desain berfokus di representasi asal aspek perangkat lunak berdasarkan sudut pengguna; ini mencakup input, proses serta format output. Pada desain sistem ada perancangan relasi dan skema basis data, Sebuah relasional skema basisdata biasanya dikembangkan dari sebuah *domain class Diagram* Setiap kelas diidentifikasi secara terpisah. Desain diharapkan menggunakan tujuan bagaimana sistem akan memenuhi tujuannya dibuat atau diciptakan. Desain sistem terdiri asal kegiatan pada mendesain yang hasilnya sebuah spesifikasi dari sistem. Bagian asal desain sistem bisa berupa konsep desain interface, proses serta data dengan tujuan menghasilkan spesifikasi sistem yang sesuai dengan kebutuhan.

3. Membangun Prototype

Membangun prototyping, yang merupakan desain serba cepat yang mengarah pada pengembangan prototyping, prototyping tersebut dieplorasi oleh pengguna dan analisis proyek dan dipergunakan dalam

rangka menyamakan kebutuhan software yang akan *di-develop*. Model *prototyping* direncanakan guna memenuhi semua keperluan/kebutuhan user, dan pada ketika proses pengembangan, programmer dengan jelas dan komprehensif memahami apa yang harus dilakukan.

4. Evaluasi dan perbaikan (Implementasi)

Langkah selanjutnya berupa proses membuat atau konstruksi produk aplikasi yang sebenarnya.

Gambar 2.4 Langkah metode pengembangan sistem berbasis *Prototype*.

4. Pembahasan

4.1 Pengumpulan Kebutuhan (Analisis)

Analisis yang didapatkan bahwasan pemanfaatan sistem klasifikasi emosi dapat menjadi tujuan dalam meneliti tentang perubahan emosi anak-anak yang selalu berubah yang akan dijabarkan dalam bentuk identifikasi kebutuhan dan analisis kebutuhan.

4.1.1 Identifikasi kebutuhan

Dalam identifikasi kebutuhan data-data diperlukan dalam proses pembuatan sistem klasifikasi emosi yang menggunakan *face recognition* adalah sebagai berikut:

1. Membutuhkan data lapangan dan data dari internet sebagai dataset.
2. Website sebagai *interface* sistem
3. Program untuk mengaktifkan sistem klasifikasi emosi

4.1.1 Analisis kebutuhan

Berdasarkan identifikasi kebutuhan didapat beberapa analisis kebutuhan sistem yang akan dibuat yang terbagi dalam dua analisis kebutuhan yaitu analisis kebutuhan Fungsional dan analisis kebutuhan Non-Fungsional.

a) Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah suatu kebutuhan yang berisi proses-proses apa saja yang dapat dilakukan oleh sistem ketika sistem berjalan. Adapun kebutuhan fungsional diantaranya :

1. Sistem dapat memberikan akurasi yang sesuai
2. Fitur-fitur yang ada pada sistem dapat berjalan
3. Sistem dapat menangkap visual
4. Sistem dapat mengolah video menjadi akurasi yang akurat

b) Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan Non-Fungsional adalah kebutuhan yang menitikberatkan pada *property* perilaku yang dimiliki oleh sistem. Adapun kebutuhan non fungsional diantaranya :

1. Sistem Operasi yang digunakan ialah Open CV sebagai library sistem pengolah gambar dan visual studio code sebagai media pengerjaan program sistem.
2. Rekomendasi spek sistem yaitu Amd 5 gen 7000 atau semi medium PC.
3. Minimal RAM 4 GB
4. Google Chrome sebagai web browser.
5. Bahasa Pemrograman yang digunakan antara lain *HTML*, *PHP*, *Javascript* dan *Python*.
6. WebCam sebagai alat pengambil gambar dengan minimal resolusi 720P atau diatasnya sangat direkomendasikan untuk hasil yang lebih akurat.

Dari penjelasan tentang identifikasi kebutuhan dan analisis kebutuhan maka disesuaikan untuk pembangunan

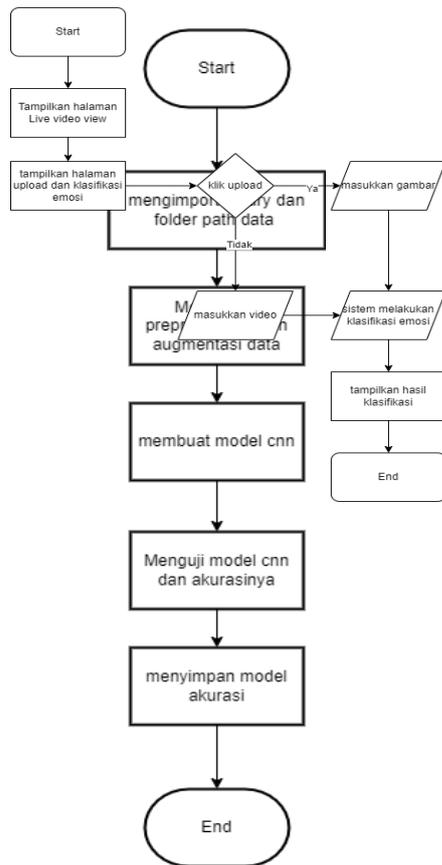
sistem klasifikasi emosi yang memanfaatkan teknologi *face recognition*.

4.2 Desain Sistem

alam Pembangunan sistem yang ada maka perlu adanya alur proses sistem yang dibangun berdasarkan kebutuhan yang ada dari setiap bagian sistem yang akan dibangun sehingga secara jalur proses yang terjadi dapat terlaksana dengan benar dan teratur sesuai dengan pengumpulan kebutuhan (analisis). Pembuatan desain sistem menggunakan media use case diagram dan flowchart sebagai alat bantu membuat sistem.

Gambar 4.1 Use case diagram

Pada gambar 4.1 diatas merupakan Gambaran tentang interaksi alur kerja dan fungsionalitas antara sistem



klasifikasi emosi dengan user yang dimana alur kerjanya dimulai dari user dapat melakukan klasifikasi emosinya secara *real time* pada saat melakukan *live video viewing*, selain itu terdapat alur kerja yang Dimana user yang ingin mengupload video atau gambar yang ingin diklasifikasikan emosinya dalam bentuk akurasi yang sesuai.

Gambar 4.2 flowchart pembuatan model

Pada gambar 4.2 merupakan alur pembuatan model yang akan digunakan untuk sistem klasifikasi emosi yang dimulai dari melakukan pengimportan library dan data train yang sudah digabungkan dengan data lapangan dan data dari internet. Setelah data tersebut upload maka dilakukan preprocessing data atau menyiapkan data yang berisi normalisasi gambar. Kemudian dataset yang sudah dinormalisasi dibagi menjadi data train dan data test, setelah itu membuat model cnn yang berfungsi sebagai media klasifikasi emosi. Setelah selesai tahap tersebut kita masuk ke tahap pengujian model dan melihat akurasi yang didapatkan. Setelah akurasi yang didapatkan memenuhi

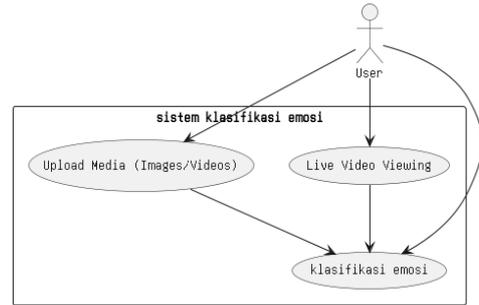
dan siap digunakan, maka akan disimpan kedalam bentuk file h5.

Gambar 4.3 flowchart sistem klasifikasi emosi

Pada gambar 4.3 merupakan alur kerja sistem klasifikasi emosi berjalan yang Dimana akan dimulai dari sistem akan menampilkan live video view yang langsung dapat mendeteksi emosi secara real time dan terdapat fitur upload media baik gambar atau video. Disamping fitur upload terdapat button yang berfungsi untuk melakukan klasifikasi emosi pada media yang sudah diupload.

4.3 Membangun Prototype

Setelah proses penentuan kebutuhan dan desain sistem



telah selesai dilakukan, maka selanjutnya adalah proses membangun prototype yang terdiri dari pembuatan model dan sistem klasifikasi emosi berdasarkan dari alur kerja desain sistem.

4.3.1 Pembuatan model

4.3.1.1 Import library

```

Initialize Reactive Jupyter | Sync all State code
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
[1] ✓ 3.8s
    
```

Gambar 4.4 import library

Pada gambar 4.5 merupakan skrip untuk memanggil library yang memiliki fungsi-fungsi untuk mengolah data seperti untuk membacanya, pembuatan modelnya dan penyimpanan hasil modelnya.

4.3.1.2 Upload data

```

# Path ke dataset
data_dir = './emosi/train2'
[2] ✓ 0.0s
    
```

Gambar 4.5 skrip upload data

Pada gambar 4.5 merupakan tampilan skrip untuk mengupload data dari folder path yang ada di windows explore yang dimasukkan kedalam data_dir sebagai variable atau folder di python yang akan memudahkan programan python untuk membaca data yang akan diolah.

4.3.1.3 Melakukan Preprocessing

Tahap selanjutnya adalah preprocessing atau persiapan data sebelum dibuat ke dalam model CNN.

Gambar 4.6 Membuat parameter

Pada gambar 4.6 merupakan skrip yang digunakan untuk membuat parameter yang akan mempermudah dalam pembuatan data training dan data validation serta dapat epochs yang memiliki fungsi untuk melihat berapa banyak model yang akan melihat data yang sedang dilatih.

4.3.1.4 Melakukan augmentasi data dan pembuatan data training serta data validation

```
# Preprocessing dan Augmentasi Data
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.3) # 70% training, 30% validation
```

Gambar 4.7 tahapan augmentasi data dan persiapan data

Pada gambar 4.7 sebuah objek yang bertugas untuk mengubah dan membagi data gambar guna melatih model pembelajaran mesin. Pertama, setiap gambar diskalakan ulang sehingga nilai pikselnya berada di antara 0 dan 1. Kemudian, gambar-gambar tersebut dimodifikasi secara acak dengan cara dishear (digeser) dan di-zoom (diperbesar atau diperkecil), serta dibalik secara horizontal untuk menciptakan variasi. Selain itu, data tersebut juga dibagi menjadi dua bagian, yaitu 70% untuk melatih model dan 30% untuk menguji kinerja model. Semua langkah ini bertujuan untuk meningkatkan kemampuan model dalam mengenali pola pada gambar yang bervariasi dan mengurangi kemungkinan model terlalu bergantung pada data pelatihan tertentu.

```
train_generator = train_datagen.flow_from_directory(
    data_dir,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical',
    color_mode='grayscale',
    subset='training') # Set as training data
```

Gambar 4.8 skrip untuk data train

```
validation_generator = train_datagen.flow_from_directory(
    data_dir,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical',
    color_mode='grayscale',
    subset='validation') # Set as validation data
```

Gambar 4.9 Skrip untuk data validation

Pada gambar 4.8 dan 4.9, terdapat skrip yang digunakan untuk mempersiapkan data yang akan dipakai dalam pembuatan model Convolutional Neural Network (CNN). Skrip ini membagi data menjadi dua bagian: data pelatihan (train) dan data validasi (validation). Pembagian ini penting agar model CNN dapat belajar dari data pelatihan dan kemudian diuji pada data validasi untuk melihat seberapa baik model tersebut bekerja pada data yang belum pernah dilihat sebelumnya.

Pertama, skrip ini menggunakan objek bernama ImageDataGenerator dari library Keras. Fungsi utama

```
# Hyperparameters
batch_size = 32
img_height = 48
img_width = 48
epochs = 40
```

dari objek ini adalah untuk mengubah dan memperbanyak data gambar yang ada. Misalnya, setiap gambar diubah skalanya agar nilai pikselnya berada dalam rentang 0 hingga 1, yang membantu dalam mempercepat proses pelatihan model. Selain itu, gambar-gambar tersebut diubah secara acak dengan beberapa cara: digeser (shear), diperbesar atau diperkecil (zoom), dan dibalik secara horizontal. Transformasi ini membuat model lebih robust atau tahan terhadap variasi dalam gambar-gambar yang mungkin dihadapinya.

Kemudian, skrip ini juga mengatur pembagian data menjadi dua bagian, yaitu 70% dari data digunakan untuk pelatihan dan 30% sisanya digunakan untuk validasi. Langkah ini penting karena model perlu diuji pada data yang tidak digunakan selama pelatihan untuk mengukur kinerjanya dengan lebih akurat. Dengan demikian, kita dapat memastikan bahwa model tidak hanya menghafal data pelatihan tetapi juga dapat bekerja baik pada data baru.

4.3.1.5 Pembuatan model CNN

```
# Membangun Model CNN
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(img_height, img_width, 1)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(512, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax') # 4 classes: happy, angry, sad, neutral
])
```

Gambar 4.10 pembuatan model CNN

```
# Kompilasi Model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Gambar 4.10 menjelaskan proses pembuatan model CNN. Alur ini dimulai dengan pengambilan gambar berukuran 48x48 piksel dalam skala abu-abu. Gambar tersebut kemudian diproses melalui beberapa lapisan konvolusi dan pooling. Setelah itu, hasilnya diratakan dan diteruskan ke lapisan fully connected untuk menghasilkan prediksi kelas akhir. Di beberapa titik, lapisan dropout

digunakan untuk mencegah overfitting dan membantu model agar dapat melakukan generalisasi dengan lebih baik.

Gambar 4.11 model diolah sebelum diuji

Pada gambar 4.11 tahap berikutnya setelah membuat model CNN adalah mengonfigurasinya untuk pelatihan dan pengujian model dengan menggunakan skrip `model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])`. Proses ini mencakup tiga komponen utama :

1. Optimizer 'adam': Algoritma optimasi yang menggabungkan keunggulan dari AdaGrad dan RMSProp, memungkinkan pelatihan yang cepat dan stabil.
2. Fungsi kerugian sparse categorical crossentropy: Digunakan untuk masalah klasifikasi dengan lebih dari dua kelas. Ini mengukur seberapa baik prediksi model dibandingkan dengan label sebenarnya dan memberikan umpan balik untuk mengurangi kesalahan.
3. Metrik akurasi: Menunjukkan persentase prediksi yang benar dibandingkan dengan total prediksi, membantu mengevaluasi kinerja model selama pelatihan dan pengujian.

Setelah konfigurasi ini, model dilatih menggunakan data pelatihan untuk menyesuaikan bobot dan bias, meminimalkan kerugian yang dihitung oleh fungsi loss. Setelah pelatihan selesai, model diuji dengan data validasi atau pengujian untuk memastikan bahwa model bekerja baik pada data yang belum pernah dilihat sebelumnya.

```

> In[4]: train_generator(train_data_loader)
# Build model
model = model_fn()
history = model.fit()
The model.compile() function in this code snippet is used to compile the neural network model. It configures the model for training by specifying the optimizer, loss function, and metrics.
train_generator,
steps_per_epoch=train_generator.samples // batch_size,
validation_steps=validation_generator.samples // batch_size,
epochs=epochs,
validation_data=validation_generator,
)
[11] ✓ 0.0s
... WARNING:absl:You are saving your model as an HDF5 file

Epoch 1/40
1/40 [====] -- 6.51s/step -- accuracy: 0.2882 -- loss: 1.3142
...
Final validation accuracy: 75.00%

```

Gambar 4.12 tahap pengujian model

Pada gambar 4.12, ditampilkan proses pengujian model sebanyak 40 kali untuk memperoleh tingkat akurasi yang diinginkan. Hal ini dilakukan untuk memastikan bahwa model yang dikembangkan memiliki performa yang konsisten dan dapat diandalkan. Dengan melakukan uji coba berulang kali, para peneliti dapat mengidentifikasi kelemahan model dan melakukan penyesuaian yang diperlukan guna meningkatkan akurasi prediksi model tersebut.

```

# Menampilkan akurasi validasi terakhir
val_accuracy = history.history['val_accuracy'][-1]
print(f"Final validation accuracy: {val_accuracy * 100:.2f}%")
[10] ✓ 0.0s
... Final validation accuracy: 75.00%

```

Gambar 4.13 hasil akurasi model

Gambar 4.14 menunjukkan hasil akurasi sistem yang berhasil mencapai angka sebesar 75.00%. Pencapaian ini menandakan bahwa sistem telah mampu mengidentifikasi dan memproses data dengan tingkat keakuratan yang cukup tinggi, meskipun masih terdapat ruang untuk perbaikan dan peningkatan lebih lanjut.

4.3.1.6 Menyimpan model akurasi

```

# Menyimpan model
model.save('final_emotion_model3.h5')
[11] ✓ 0.0s
... WARNING:absl:You are saving your model as an HDF5 file

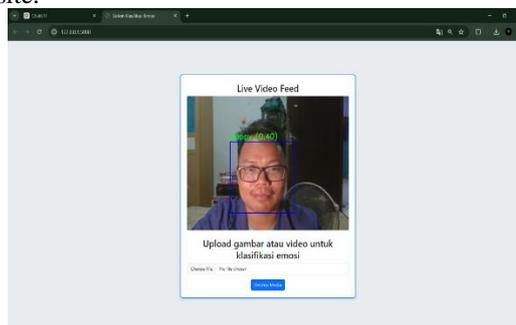
```

Gambar 4.14 menyimpan model

Pada gambar 4.15 menampilkan skrip yang digunakan untuk menyimpan file dalam format .h5, Skrip ini bertujuan untuk memastikan bahwa data yang diolah dapat disimpan dengan aman dan mudah diakses kembali di kemudian hari. Dalam proses penyimpanan ini, langkah-langkah penting meliputi pengaturan nama file, menentukan lokasi penyimpanan, dan memastikan bahwa data tersimpan dalam struktur yang benar.

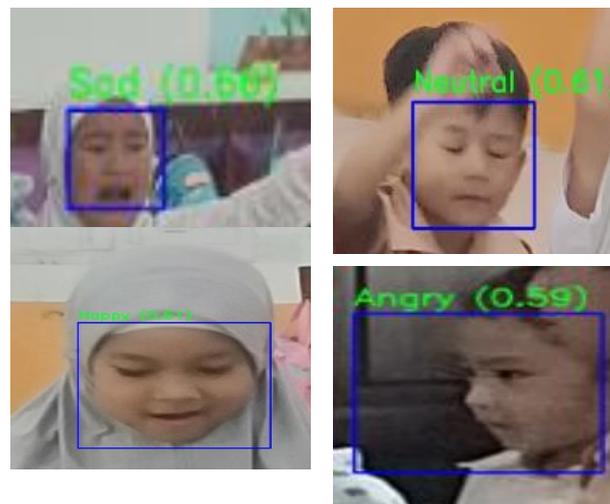
4.3.2 Tampilan sistem klasifikasi emosi

Pada tahapan ini akan ditampilkan hasil jadi dari pembuatan model yang sudah dibuat kedalam bentuk website.



Gambar 4.14 tampilan sistem klasifikasi emosi

Gambar 4.14 merupakan tampilan sistem klasifikasi emosi yang Dimana sistem setelah dirun akan mengaktifkan kamera dan mulai mendeteksi emosi secara Real Time dengan melakukan perhitungan dari model CNN yang sudah dibuat.



Gambar 4.15 fitur deteksi media

Gambar 4.15 menampilkan fitur yang dirancang untuk mendeteksi emosi pada file media berupa emosi senang, sedih, marah dan netral. Baik itu video maupun gambar, yang telah dipilih sebelumnya oleh pengguna. Proses dimulai dengan pengguna memilih file yang diinginkan melalui fitur "choose file". Setelah file dipilih, pengguna dapat melanjutkan dengan menekan tombol "deteksi media". Tombol ini berfungsi untuk memulai proses analisis media yang telah terhubung dengan model Convolutional Neural Network (CNN) yang sudah dikembangkan secara khusus untuk tugas ini.

4.4 Perbaikan dagtn evaluasi (Implementasi)

Tahap selanjutnya ialah implementasi atau pengujian dimana sistem klasifikasi emosi akan diuji menggunakan metode pengujian sistem yaitu black-box testing dan white-box testing yang berfungsi untuk melihat kesalahan sistem prototype ini dapat berjalan dengan lancar.

4.4.1 Black-Box Testing

No.	Input	Yang diharapkan	Hasil
-----	-------	-----------------	-------

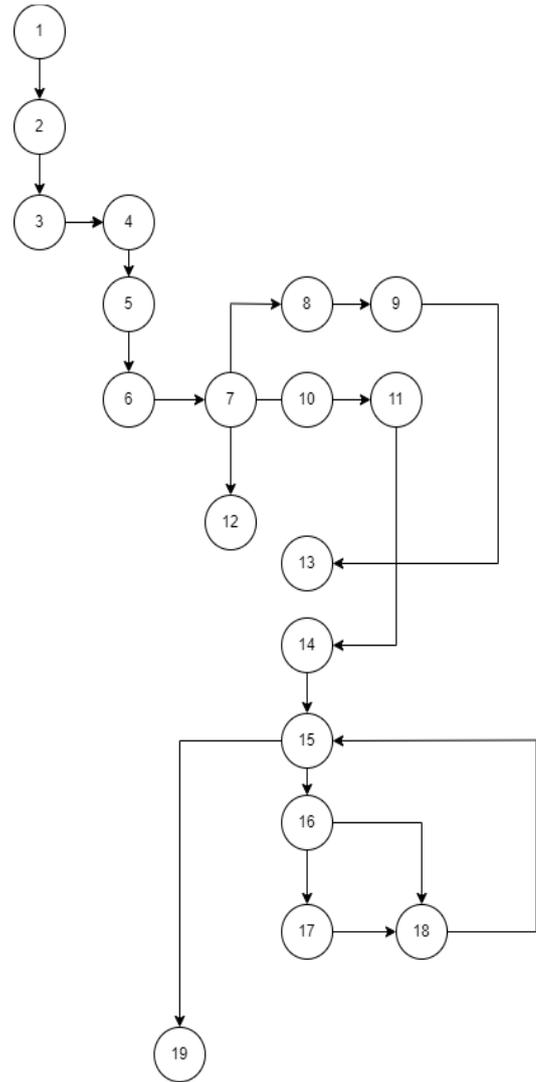
```

77 @app.route('/upload', methods=['POST'])
78 def upload():
79     if 'file' not in request.files:
80         return 'No file part'
81     file = request.files['file']
82     if file.filename == '':
83         return 'No selected file'
84     if file:
85         filename = file.filename
86         file_path = os.path.join('uploads', filename)
87         file.save(file_path)
88
89         if filename.lower().endswith(('.png', '.jpg', '.jpeg')):
90             return detect_emotion_image(file_path)
91         elif filename.lower().endswith(('.mp4', '.avi', '.mov')):
92             return Response(gen_video_frames(file_path), mimetype='multipart/x-mixed-replace; boundary=frame')
93         else:
94             return 'Unsupported file type'
95     return 'file upload failed'
96
97 def detect_emotion_image(file_path):
98     frame = cv2.imread(file_path)
99     frame = detect_emotion(frame)
100     img_encoded = cv2.imencode('.jpg', frame)
101     return Response(img_encoded.tobytes(), mimetype='image/jpeg')
102
103 def gen_video_frames(file_path):
104     cap = cv2.VideoCapture(file_path)
105     while cap.isOpened():
106         ret, frame = cap.read()
107         if not ret:
108             break
109         frame = detect_emotion(frame)
110         ret, buffer = cv2.imencode('.jpg', frame)
111         frame = buffer.tobytes()
112         yield (b'--frame\r\n'
113               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
114     cap.release()
    
```

4.4.2 White-Box Testing

Gambar 4.16 sintaks upload dan deteksi emosi pada media

1 = baris 77-78	9 = 90	17 = 108
2 = baris 79	10 = 91	18 = 109 -113
3 = baris 80-81	11 = 92	19 = 114
4 = baris 82	12 = 93 -95	
5 = baris 83	13 = 97-101	
6 = baris 84	14 = 103-104	
7 = baris 85-87	15 = 105	
8 = baris 89	16 = 108	



Gambar 4.16 flowgraph upload dan deteksi media

Pada gambar 4.16 diatas merupakan sintaks yang akan diujikan menggunakan *white-box testing* untuk mengevaluasi alur program. Berdasarkan *flowgraph* pada gambar 4.20, diketahui bahwa terdapat 19 *node* (n) dan 20 *edge* (€). Sehingga nilai CC adalah $20 - 19 + 2 = 3$. Berikut merupakan jalur independent yang diperoleh:

- Jalur 1 : 1-2-3-4-5-6-7-8-9-13
Keterangan : berhasil mengupload dan mendeteksi emosi bertipe gambar.
- Jalur 2 : 1-2-3-4-5-6-7-10-11-14-15-16-17-18-19
Keterangan : berhasil mengupload dan mendeteksi emosi bertipe video
- Jalur 3 : 1-2-3-4-5-6-7-12
Keterangan : Tipe file tidak sesuai

5. KESIMPULAN

Berdasarkan uraian-uraian dari masing-masing bab dan hasil pembahasan maka dapat disimpulkan bahwa penggunaan teknologi Face Recognition untuk mengklasifikasi emosi anak TK berbasis website sebagai berikut :

- Akurasi pembuatan model mendapatkan sebesar 75 %
- Sistem berhasil mendeteksi emosi yang sudah dibuat dalam model.
- kamera web berhasil mendeteksi visual

4. fitur-fitur yang sudah dibuat berhasil diimplementasi seperti fitur *choose file* dan deteksi media juga dapat mendeteksi emosi dari media.

Adapun juga sistem ini banyak kekurangan seperti pendeteksian emosi pada media bertipe video yang akan membuat server local flask mengalami delay dan beberapa objek yang akan langsung terdeteksi emosinya yang merupakan benda mati dikarenakan membutuhkan kamera yang resolusinya jernih dan pencahayaan yang terang.

6. SARAN

Setelah melakukan penelitian ini maka terdapat saran-saran yang akan membantu peneliti yang ingin mengembangkan sistem ini :

1. Sistem ini diharapkan kalau bisa dikembangkan kedalam bentuk *mobile*
2. Sistem klasifikasi emosi ini memerlukan sistem admin untuk mempermudah dalam membuat model baru dan menambah data.
3. Sistem ini juga belum bisa membedakan objek yang *blur*
4. Kamera harus beresolusi tinggi agar dapat hasil deteksi yang lebih akurat
5. Membuat batasan atau parameter antar emosinya agar mudah membedakannya, seperti marah atau senang yang hampir ada kesamaan diraut wajah.
6. Memakai *mediapipe* untuk mempermudah dalam mendeteksi emosi dengan menggunakan titik koordinat raut wajah.
7. Pengumpulan sample gambar harus jernih dan terlihat jelas emosinya.
8. Penelitian selanjutnya silahkan menggunakan metode lainnya supaya bisa mendapatkan hasil yang maksimal.
9. Penelitian mungkin bisa dicoba ke anak SMP dan SMA sebagai sample supaya bisa melihat perbandingan emosinya.
10. Perbaiki nilai emosi dalam bentuk persentase
11. Perbaiki fitur deteksi agar bisa akurat

7. DAFTAR PUSTAKA

- Arsal, Muhammad, Wardijono Bheta Agus, and Dina Anggraini. 2020. "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN." *Jurnal Nasional Teknologi Dan Sistem Informasi* 6(1):55–63. doi: 10.25077/teknosi.v6i1.2020.55-63.
- Febrywinata, Edwin. 2024. "Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab." *Merkurius: Jurnal Riset Sistem Informasi Dan Teknik Informatika* 2(4):185–93. doi: 10.61132/mercurius.v2i4.162.
- Isum, Raden, Suryani Maryati, and Burhanuddin Tryatmojo. 2019. "Raden Isum Suryani Maryati Akurasi Sistem Face Recognition Akurasi Sistem Face Recognition OpenCV Menggunakan Raspberry Pi Dengan Metode Haar Cascade KATA KUNCI Akurasi Face Recognition Raspberry Pi OpenCV Haar Cascade." *Jurnal Ilmiah Informatika (JIF)* 07(02):12790.

- Priyanto, A. 2014. "PENGEMBANGAN KREATIVITAS PADA ANAK USIA DINI MELALUI AKTIVITAS BERMAIN." *Jurnal Ilmiah Guru "Cope"* (02):1–7.
- Purnomo, Dwi. 2017. "Model Prototyping Pada Pengembangan Sistem Informasi." *JIMP-Jurnal Informatika Merdeka Pasuruan* 2(2).
- Solikhah, Linda Dwi, Wiyanti Hidayat, and Mudaris Muslim. 2014. *CONSILIUUM Psikodrama Untuk Meningkatkan Kestabilan Emosi Siswa SMK*. Vol. 2.
- Sosodoro, Bambang, Dosen Jurusan, Karawitan Fakultas, and Seni Pertunjukan. 2014. "KLASIFIKASI KARAKTER REBABAN GAYA SURAKARTA." 14(1).
- Trisiawan, Inggis Kurnia, and Yuliza Yuliza. 2022. "Penerapan Multi-Label Image Classification Menggunakan Metode Convolutional Neural Network (CNN) Untuk Sortir Botol Minuman." *Jurnal Teknologi Elektro* 13(1):48. doi: 10.22441/jte.2022.v13i1.009.
- Yumarlin MZ. 2016. "Evaluasi Penggunaan Website Universitas Janabadra Dengan Menggunakan Metode Usability Testing." *Informasi Interaktif* 1(1):34–43.