



Click here and write your Article Category

# Sistem Pendeteksi Gejala Awal Tantrum Pada Anak Autisme Melalui Ekspresi Wajah Dengan *Convolutional Neural Network*

Dini Ramadhani Salsabila <sup>1</sup>, Ratna Aisuwarya <sup>2</sup> Nefy Puteri Novani <sup>3</sup>

Jurusan Teknik Komputer, FTI Universitas Andalas Limau Manis Kec. Pauh, Kota Padang, Sumatera Barat, 25163, Indonesia

## ARTICLE INFORMATION

Received: February 00, 00  
Revised: March 00, 00  
Available online: April 00, 00

## KEYWORDS

Emosi Wajah, Autisme, CNN, Keras, Image Processing

## CORRESPONDENCE

Phone: +62 81365385473  
E-mail: [diniramadhani553@gmail.com](mailto:diniramadhani553@gmail.com)

## ABSTRACT

Tantrums are outbursts of anger and they can occur at any age. An attitude tantrum or what is commonly referred to as a temper tantrum is a child's outburst of anger that often occurs when a child shows negative behavior. Emotional outbursts of tantrums that occur in children with autism are not only to seek the attention of adults, but also as an outlet for a child's feelings for parents and those around him on a whim or feeling he is feeling, but the child cannot convey it. For this reason, researchers propose a system for detecting early symptoms of tantrums in children with autism through facial expressions with CNN. The CNN method is one of the deep learning methods that can be used to recognize and classify an object in a digital image. Then the preprocessing process is carried out using labeling on the data. Then the CNN architecture is designed with input containing 48x48x1 neurons. The data was then trained using 357 epochs with an accuracy rate of 72.67%. Then tested using test data for children with autism to get an average accuracy value of 72.67%.

## PENDAHULUAN

Data dari *Center for Disease Control and Prevention* menyebutkan bahwa prevalensi kejadian penderita autisme meningkat dari 1 per 150 populasi pada tahun 2000 menjadi sebesar 1 per 54 pada tahun 2016[1]. Autisme terjadi karena gangguan pada perkembangan sistem saraf pusat, menyebabkan terganggunya komunikasi verbal dan non verbal, interaksi sosial dan perilaku terbatas dan berulang[2]. Kemarahan ataupun ketidakstabilan emosi yang terjadi pada anak autisme kerap disebut dengan tantrum. Tantrum merupakan ledakan amarah serta ledakan itu bisa terjadi pada semua tahapan umur. Sikap tantrum ataupun yang biasa disebut dengan sebutan temper tantrum ialah suatu letupan kemarahan anak yang kerap terjadi pada dikala anak menunjukkan perilaku negatif[3]. Luapan emosi tantrum yang terjadi pada anak autisme tidak cuma untuk mencari perhatian orang dewasa saja, namun sebagai pelampiasan perasaan seorang anak kepada orang tua maupun orang yang terletak disekitarnya atas kemauan ataupun perasaan yang lagi dirasakannya, tetapi anak tidak dapat menyampaikannya. Ada dua jenis tantrum yang berbeda, yakni: tantrum yang berawal dari kesedihan dan amarah; dan tantrum yang berakar pada kebingungan serta ketakutan[4]

Ekspresi merupakan salah satu bentuk komunikasi nonverbal yang merupakan hasil dari satu atau lebih gerakan pada posisi otot pada wajah dan dapat menyampaikan keadaan emosi seseorang kepada pengamat. komunikasi nonverbal yang menggunakan gerakan tubuh terdiri dari tiga komponen utama: pesan fasial, pesan gestural, dan pesan postural[5]. Ekspresi wajah mengungkapkan pikiran yang melintas pada seseorang. Hal ini merupakan salah satu bentuk pengungkapan emosi melalui ekspresi wajah. Ekspresi wajah bisa dideteksi lewat serangkaian gerakan otot wajah yang unik misalnya senyuman. Ekspresi wajah adalah kunci dalam pemahaman dan mendeteksi emosi seseorang. Sebagian besar manusia zaman modern menggunakan wajah untuk memberikan sinyal emosi yang dirasakan[6]. Membaca ekspresi wajah seseorang, bukan hanya dilakukan untuk membongkar rahasia atau menemukan kebohongan dalam diri seseorang. Membaca ekspresi dapat dilakukan agar lebih berempati terhadap kondisi emosional yang sedang dialami oleh orang tersebut. Ada tujuh emosi dasar manusia, yaitu netral, marah, jijik, takut, senang, sedih, dan terkejut, dan emosi dasar ini bisa dikenali dari ekspresi wajah manusia.

Dengan tersedianya teknologi pencitraan berbiaya rendah dan perangkat komputasi, sistem pengenalan ekspresi wajah sekarang berpotensi berguna di kehidupan sehari-hari. Penelitian yang

berkaitan dengan pendeteksian emosi wajah telah banyak dilakukan, seperti *Facial Emotion Detection using Neural Network*[7] yang menerapkan *convolutional neural network* (CNN) untuk mengenali ekspresi emosi wajah, *Real Time Face Expression Recognition of Children with Autism*[8] yang mengenali ekspresi wajah pada anak-anak dengan gangguan spektrum autisme (GSA) selama waktu bermain, dan Sistem Identifikasi dan Monitoring Emosi Dasar Manusia Melalui Ekspresi Wajah dengan Metode *Deep Learning*[9] yang melakukan identifikasi dan monitoring emosi dasar manusia melalui ekspresi wajah dengan metode *deep Learning*. Hasilnya, ada banyak pekerjaan yang dilakukan untuk mengenali emosi pada ekspresi wajah dan penerapannya pada penelitian ini akan bermanfaat dalam mendeteksi gejala awal perilaku tantrum pada anak autisme. Dengan adanya pemantauan dan deteksi emosi sangat diperlukan untuk menghindari terjadinya perilaku tantrum. Berdasarkan pada permasalahan tersebut, dapat disimpulkan bahwa perilaku tantrum dapat dipicu oleh emosi negatif dan emosi ini dapat dikenali melalui ekspresi wajah. Oleh karena itu, dibutuhkan suatu alat yang dapat mendeteksi emosi berdasarkan ekspresi wajah sebagai identifikasi gejala awal perilaku tantrum yang dapat membantu pekerjaan dalam mengawasi anak dengan gangguan spektrum autisme. Oleh karena itu, di sini penulis akan membuat sistem pendeteksi gejala awal tantrum pada anak autisme melalui ekspresi wajah dengan *convolutional neural network*.

Sistem ini akan diberikan algoritma dengan metode *Convolutional Neural Network* (CNN). CNN merupakan salah satu metode perhitungan *neural network* yang sering digunakan pada pengolahan citra untuk mengklasifikasi serta mengidentifikasi objek pada suatu *image*. CNN terdiri dari neuron yang memiliki *weight*, *bias* dan *activation function*. CNN bekerja dengan cara menerima data citra latih dan menghasilkan *pre-trained* model yang akan digunakan untuk mendeteksi citra uji[10]. Pada penelitian ini CNN digunakan sebagai proses klasifikasi emosi melalui ekspresi wajah pada anak autisme.

## LANDASAN TEORI

### 1. Ekspresi Wajah

Ekspresi wajah adalah hasil dari satu atau lebih gerakan atau posisi otot pada wajah yang biasanya tidak disadari. Ekspresi wajah termasuk salah satu aspek terpenting dalam komunikasi manusia. Wajah bertanggung jawab untuk berkomunikasi tidak hanya pikiran atau ide, tetapi juga emosi[11]. Apa yang membuat komunikasi emosi menarik adalah bahwa ekspresi emosi tertanam secara biologis, dan diungkapkan dengan cara yang sama oleh semua orang. Emosi pada wajah mengomunikasikan keadaan emosional dan niat perilaku seseorang. Ekspresi emosi wajah adalah isyarat sosial yang menonjol dalam interaksi sehari-hari. Ekspresi emosi wajah digunakan dalam berbagai bidang penelitian, misalnya kognisi sosial, psikologi, dan ilmu saraf. Banyak pekerjaan telah dilakukan dalam memproses ekspresi wajah ini terutama untuk mengenali kondisi emosional.

### 2. Pengertian Anak Autism

Gangguan spektrum autisme (ASD) adalah gangguan ketidaknormalan perkembangan yang dapat menyebabkan sikap,

kemampuan komunikasi, dan interaksi sosial yang tidak normal. Beberapa anak dengan gangguan spektrum autisme membutuhkan banyak bantuan dalam kehidupan sehari-hari mereka. Anak autis memiliki beberapa kondisi: gangguan autistik, gangguan perkembangan *pervasive*, dan sindrom *asperger*. Anak dengan kondisi ini disebut gangguan spektrum autisme. Autism adalah gangguan perkembangan yang mempengaruhi komunikasi dan perilaku. Meskipun autisme dapat didiagnosis pada usia berapa pun, autisme disebut sebagai gangguan perkembangan karena gejala umumnya muncul dalam dua tahun pertama kehidupan.

### 3. Pengertian Tantrum

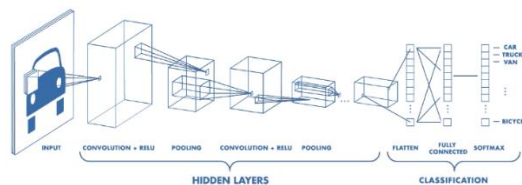
Tantrum (atau temper tantrum) adalah ledakan emosi, biasanya dikaitkan dengan anak-anak atau orang-orang dalam tekanan emosional, yang biasanya ditandai dengan keras kepala, menangis, berteriak, menjerit, membangkang, mengomel, marah-marah, penolakan terhadap upaya untuk menenangkan, dan dalam beberapa kasus, kekerasan. Kontrol fisik bisa hilang, orang tersebut mungkin tidak bisa diam, dan bahkan jika tujuan orang tersebut terpenuhi, dia mungkin tetap tidak nyaman. Tantrum adalah suatu perilaku yang masih tergolong normal yang merupakan bagian dari proses perkembangan, suatu masa perkembangan fisik, kognitif, dan emosional. Sebagai masa perkembangan, amukan pasti akan berakhir. Berdasarkan teori-teori di atas, dapat disimpulkan bahwa temper tantrum adalah semburan emosi yang meledak-ledak akibat suasana tidak menyenangkan yang dirasakan anak. Ledakan emosi bisa berupa tangisan, teriakan, lempar barang, berguling-guling, memukul ibunya atau aktivitas-aktivitas besar lainnya. Amukan juga merupakan cara normal untuk melepaskan semua perasaan yang menumpuk.

### 4. Computer Vision

*Computer vision* adalah ilmu dan teknologi mesin yang dapat melihat. *Computer vision* adalah bidang ilmu komputer yang berfokus pada mereplikasi bagian kompleksitas sistem penglihatan manusia dan memungkinkan komputer untuk mengidentifikasi dan memproses objek dalam gambar dan video dengan cara yang sama seperti yang dilakukan manusia[12]. Sebagai disiplin ilmu, *computer vision* berkaitan dengan teori dan teknologi untuk membangun sistem buatan yang memperoleh informasi dari gambar atau data *multi*-dimensi. Pemrosesan pada *computer vision* membutuhkan data masukan yang disediakan oleh sistem *computer vision*, yang bertindak sebagai sensor penglihatan dan menyediakan informasi tingkat tinggi tentang lingkungan. Sampai saat ini, *computer vision* hanya bekerja dalam kapasitas terbatas. *Computer vision* dimulai pada tahun 1950-an dan pertama kali digunakan secara komersial untuk membedakan antara teks yang diketik dan tulisan tangan pada 1970-an, saat ini aplikasi untuk *computer vision* telah berkembang secara eksponensial. Sebelum munculnya *deep learning*, tugas yang dapat dilakukan oleh *computer vision* sangat terbatas dan membutuhkan banyak pengkodean manual oleh pengembang dan operator manusia. *Computer Vision* menjadikan komputer “*acts like human sight*”, sehingga mendekati kemampuan manusia dalam menangkap informasi *visual*.

## 5. Convolutional Neural Network

*Convolutional Neural Network* merupakan metode pengenalan yang menerima citra latih dan menghasilkan *pre-trained* model yang akan digunakan untuk mendeteksi citra uji. *Convolutional neural network*, juga dikenal sebagai CNN adalah bagian dari *neural network* yang berspesialisasi dalam memproses data yang memiliki topologi *grid*, seperti gambar. Gambar digital adalah representasi biner dari data *visual*. Pada CNN tiap neuron dipresentasikan dalam wujud 2 dimensi, sehingga CNN sesuai untuk pemrosesan dengan input berupa citra. CNN terdiri dari tumpukan filter konvolusi yang dipelajari yang mengekstrak fitur gambar kontekstual hierarkis, dan merupakan bentuk populer dari jaringan pembelajaran mendalam[13]. Sama seperti setiap *neuron* dalam sistem penglihatan biologis, setiap *neuron* dalam CNN memproses data hanya di bidang reseptifnya juga. Lapisan-lapisan tersebut disusun sedemikian rupa sehingga mereka mendeteksi pola yang lebih sederhana terlebih dahulu (garis, kurva, dll.) dan pola yang lebih kompleks (wajah, objek, dll). CNN memiliki tiga layer: *convolutional layer*, *pooling layer*, dan *fully connected layer*[10].



Gambar 2.1. Contoh Proses Convolutional Neural Network[10]

## 6. Artificial Intelligence (Kecerdasan Buatan)

*Artificial Intelligence* (AI) adalah studi dan penciptaan sistem komputer yang dapat memahami, menalar, dan bertindak. Tujuan utama AI adalah menghasilkan mesin cerdas. Kecerdasan harus ditunjukkan dengan berpikir, mengambil keputusan, memecahkan masalah, lebih penting lagi dengan belajar. AI merupakan bidang interdisipliner yang membutuhkan pengetahuan dalam ilmu komputer, linguistik, psikologi, biologi, filsafat dan lain sebagainya untuk penelitian yang serius. Kecerdasan Buatan adalah cara membuat komputer, robot yang dikendalikan komputer, atau perangkat lunak berpikir cerdas, dengan cara yang serupa dengan cara berpikir manusia cerdas. AI juga dapat didefinisikan sebagai bidang ilmu komputer yang berhubungan dengan cara-cara di mana komputer dapat dibuat untuk melakukan fungsi kognitif yang dianggap berasal dari manusia. AI yang kuat percaya bahwa mesin dapat dibuat hidup atau sadar diri. Ada dua jenis AI yang kuat: AI yang mirip manusia, di mana program komputer berpikir dan beralasan sampai ke tingkat manusia. AI non-manusia, di mana program komputer mengembangkan cara berpikir dan penalaran non-manusia. AI yang lemah tidak bisa menciptakan kecerdasan tingkat manusia dalam mesin, tetapi teknik AI dapat dikembangkan untuk memecahkan banyak masalah kehidupan nyata. Saat ini pendekatan AI yang terinspirasi secara biologis seperti jaringan saraf dan algoritma genetika sudah ada. AI adalah bidang terbaik untuk dimainkan saat ini. AI berevolusi dari pemikiran bahwa membuat mesin manusia itu memungkinkan.

## 7. Deep Learning

*Deep learning* dianggap sebagai evolusi dari *machine learning*. *Deep learning* menggunakan jaringan saraf yang dapat diprogram yang memungkinkan mesin membuat keputusan akurat tanpa bantuan dari manusia. *Deep learning* secara teknis adalah *machine learning* juga dan berfungsi dengan cara yang serupa. Namun, kemampuannya berbeda. *Deep learning* menyusun algoritma berlapis-lapis untuk membuat jaringan saraf tiruan yang dapat mempelajari dan membuat keputusan cerdas sendiri. *Deep learning* adalah subbidang dari *machine learning*. Meskipun keduanya termasuk dalam kategori kecerdasan buatan, *deep learning* lebih kuat ke kecerdasan buatan yang paling mirip manusia.

## 8. Image Processing (Pengolahan Citra)

Pengolahan citra digital ialah disiplin ilmu yang memanipulasi citra dengan menggunakan komputer digital. Pengolahan citra adalah metode untuk melakukan beberapa operasi pada citra. Citra bisa berupa foto atau video digital yang bisa langsung disimpan di media penyimpanan[14]. Pengolahan citra bertujuan untuk mengubah citra menjadi bentuk digital dan melakukan beberapa proses pada citra untuk mendapatkan citra yang lebih baik atau mengambil beberapa informasi yang bisa dimanfaatkan dari citra itu. Input dari pengolahan citra ini adalah video atau gambar, seperti foto. Outputnya sesuai dengan bagian gambar yang diinginkan.

## 9. OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah sebuah *library* perangkat lunak yang berfungsi untuk memfasilitasi operasi data gambar dasar seperti memuat gambar dari *disk*, menampilkannya ke layar, dan beberapa operasi dasar lainnya[15]. Dalam OpenCV, gambar diubah menjadi *array* multi-dimensi agar manipulasinya menjadi sederhana. Misalnya, gambar *grayscale* diinterpretasikan sebagai *array* 2D dengan *pixel* yang bervariasi dari 0 hingga 255. Setiap *pixel* dalam gambar *grayscale* memiliki nilai yang mewakili bayangan abu-abu. Di OpenCV, ada 256 warna abu-abu - dari 0 hingga 255. Jadi, gambar *grayscale* akan memiliki nilai *grayscale* yang terkait dengan setiap *pixel*. Di citra warna OpenCV dalam ruang warna RGB (Merah, Hijau, Biru) memiliki 3-tupel yang terkait dengan setiap piksel : (B, G, R). Urutannya adalah BGR bukan RGB. Ini karena OpenCV pertama kali dikembangkan beberapa tahun yang lalu dan standarnya adalah BGR. Selama bertahun-tahun, standarnya sekarang telah menjadi RGB tetapi OpenCV masih mempertahankan BGR, ini untuk memastikan tidak ada pemutusan kode yang ada. Setiap nilai dalam BGR 3-tuple memiliki kisaran [0, 255]. Kemungkinan warna yang ada untuk setiap piksel dalam gambar RGB di OpenCV ada :  $256 * 256 * 256 = 16777216$ .

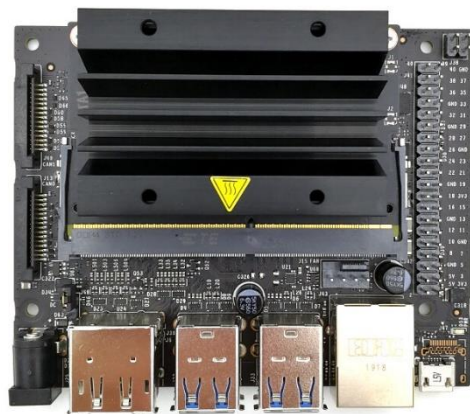
## 10. Python

Python adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dengan semantik dinamis dari jenis bahasa interpreter karena dijalankan langsung oleh interpreter tanpa melalui tahap kompilasi. Python dipilih sebagai bahasa

pemrograman dalam penelitian ini, karena bahasa ini memiliki banyak library yang memudahkan dalam hal pembuatan program yang melibatkan banyak manipulasi vektor dan matrik (library keras[16], tensorflow, numpy pandas). Python dikembangkan oleh Guido van Rossum dan secara luas dianggap sebagai bahasa penulisan dan aplikasi Web. Python mendukung banyak paradigma pemrograman, khususnya pada pemrograman berorientasi objek, pemrograman imperatif, dan pada pemrograman fungsional[17]. Python dapat digunakan untuk berbagai tujuan pengembangan perangkat lunak dan dapat dijalankan di berbagai platform sistem operasi[18]. Python mendukung modul dan paket yang memfasilitasi modularitas program dan penggunaan kode.

### 11. Nvidia Jetson Nano

NVIDIA Jetson Nano merupakan perangkat pengembangan Artificial Intelligence (AI) yang dapat digunakan untuk menjalankan berbagai beban AI modern dengan performa yang mengagumkan[19]. NVIDIA memungkinkan para pengembang dapat menjalankan *framework* dan model AI untuk aplikasi pengenalan gambar, pendeteksian objek, segmentasi, pengenalan suara dan sebagainya. NVIDIA Jetson Nano ditenagai oleh micro USB serta dibekali banyak pin I/O, mulai dari GPIO sampai CSI. Pin I/O yang berlimpah ini mempermudah pengembang dalam menghubungkan bermacam sensor untuk keperluan pengembangan aplikasi AI[20]. NVIDIA Jetson Nano juga didukung oleh NVIDIA JetPack, yang mencakup *Board Support Package* (BSP), Linux OS, NVIDIA CUDA, cuDNN, dan TensorRT untuk keperluan *deep learning*, *computer vision*, GPU computing, *multimedia processing* dan masih banyak lagi. Pada penelitian[21] jetson nano memiliki konsumsi daya yang lebih besar, tetapi performanya lebih bagus dalam waktu yang lebih singkat, akurasi yang lebih tinggi, dan kumpulan data yang besar daripada Raspberry pi.



Gambar 2.2. Nvidia Jetson Nano [19]

### 12. Modul Kamera Raspberry Pi V2

Raspberry Pi Camera Module V2 adalah salah satu kamera CSI-2 yang paling populer. Modul Kamera V2 Raspberry Pi kompatibel dengan Nvidia Jetson Nano, tidak seperti V1-nya. Modul kamera ini biasa digunakan untuk merekam video dengan *high definition*. Modul Kamera Raspberry Pi v2 terdiri dari Sony IMX219 8 megapiksel berkualitas tinggi yang dirancang khusus untuk *add-on board mini pc* seperti Nvidia Jetson Nano, yang

menampilkan lensa fokus tetap. Kamera ini dapat mengambil gambar dengan ukuran 3280 x 2464 piksel, dan juga mendukung video 1080p30, 720p60 dan 640x480p90. Modul kamera terhubung ke Nvidia Jetson Nano menggunakan antarmuka CSI khusus, yang dirancang khusus untuk antarmuka ke kamera[22].



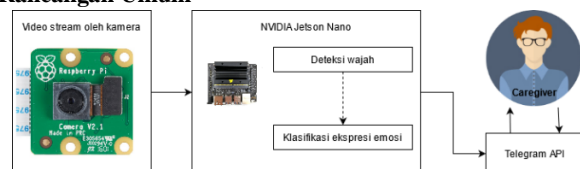
Gambar 2.3. Modul Kamera Raspberry Pi V2[22]

### 13. Aplikasi Telegram

Telegram adalah layanan perpesanan *multi-platform* yang didirikan oleh pengusaha Rusia Pavel Durov[23]. Telegram pertama kali diluncurkan di iOS dan Android pada akhir 2013. Yang membuat Telegram unik adalah fokusnya pada privasi, enkripsi, dan API *open-source*. Telegram juga memungkinkan beberapa perangkat untuk menggunakan akun yang sama (diverifikasi melalui SMS), dan beberapa akun di perangkat yang sama. Fungsionalitas inti Telegram sama seperti kebanyakan aplikasi perpesanan lainnya: dapat mengirim pesan kepada pengguna Telegram lainnya, membuat percakapan grup, menelepon kontak, dan mengirim *file* dan stiker. Telegram memungkinkan penerapan *filter* pesan terperinci sambil mencari pesan dalam obrolan. Telegram memfilter pesan berdasarkan kueri teks, dan bahkan pada tipenya, dan fitur ini sering digunakan oleh klien grafis untuk mengimplementasikan fitur seperti galeri obrolan, gambar profil obrolan, dan lainnya. Tidak seperti aplikasi obrolan lainnya, Telegram berbasis *cloud*, yang berarti bisa memindahkan riwayat percakapan antara *smartphone*, Tablet, dan bahkan laptop.

## METODOLOGI PENELITIAN

### Rancangan Umum

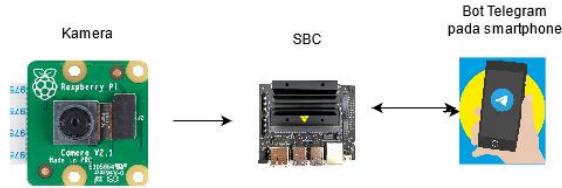


Gambar 3. 1 Diagram Blok Rancangan Umum Sistem

Berdasarkan Gambar 3.2 di atas, alur kerja sistem diawali dari kamera yang berfungsi untuk melakukan *video stream*, kemudian pendeteksian wajah yang dilanjutkan dengan pengklasifikasian ekspresi emosi dengan metode *Convolutional Neural Network* (CNN). Disaat terdeteksi ekspresi emosi negatif yaitu: marah, sedih, atau takut, maka sistem akan otomatis mengirim notifikasi melalui Telegram *messenger* melalui Telegram bot API.

### Rancangan Perangkat Keras

Bentuk perancangan perangkat keras dari sistem ini adalah :

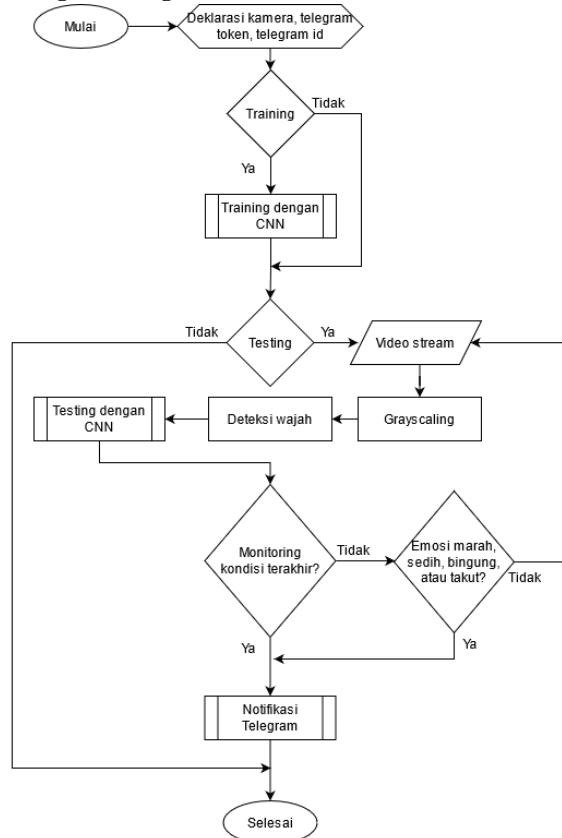


Gambar 3.2 Perancangan Perangkat Keras

Berdasarkan gambar 3.4 berikut prinsip kerja masing-masing komponen sebagai berikut :

1. Kamera, berfungsi untuk menangkap citra gambar dari wajah.
2. SBC, berperan sebagai inti pengendali sistem yang akan melakukan klasifikasi emosi menggunakan algoritma *Convolutional Neural Network*, serta mengirimkan notifikasi ke aplikasi telegram bot.
3. *Smartphone* dengan aplikasi Telegram, berperan sebagai notifikasi ke *smartphone* melalui bot Telegram.

### Rancangan Perangkat Lunak



Gambar 3.3 Flowchart Rancangan Umum Proses

Dalam penelitian ini model dilatih untuk mengenali wajah manusia dari inputan *video stream* dan memprediksi emosi dari wajah yang terdeteksi. Ketika video stream diberikan sebagai input ke sistem, pra-pemrosesan citra akan dilakukan oleh OpenCV. Citra diubah menjadi citra *grayscale* kemudian wajah dideteksi dengan Haar Cascade. Wajah yang terdeteksi kemudian di potong untuk melakukan operasi lebih lanjut.

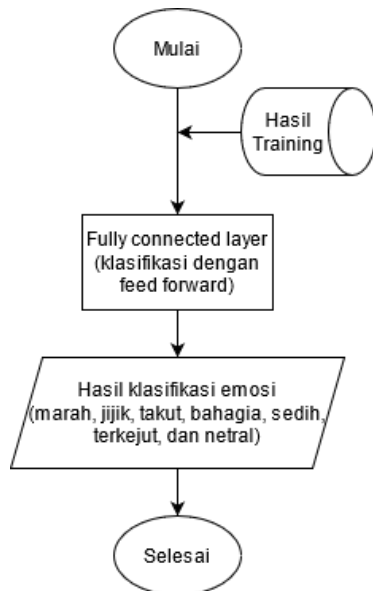
Berdasarkan Gambar 3.3 di atas, sistem terbagi atas beberapa proses yaitu: proses *training* dengan CNN, proses pendeteksian wajah, proses pendeteksian emosi oleh CNN, dan proses pemberian notifikasi pada aplikasi Telegram.



Gambar 3.4 Flowchart training

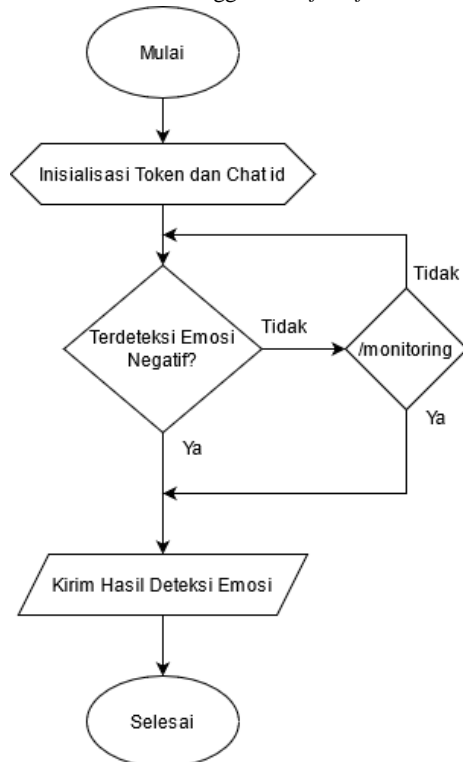
Berdasarkan gambar 3.5 diatas dipaparkan ada 3 sesi dalam arsitektur CNN, yaitu *preprocessing*, ekstraksi fitur dan klasifikasi. *Preprocessing* merupakan tahap dimana dilakukan pengolahan citra pada gambar. Ekstraksi fitur merupakan teknik yang membolehkan suatu *system* berjalan secara otomatis untuk memastikan representasi dari suatu *image* menjadi *features* yang berbentuk angka-angka yang merepresentasikan image tersebut. Klasifikasi merupakan suatu tahap dimana hasil dari ekstraksi fitur akan digunakan untuk proses klasifikasi.





Gambar 3.5 Flowchart Testing

Proses *testing* tidak jauh berbeda dengan proses *training*, yang membedakannya yaitu bobot serta bias yang digunakan pada langkah ini menggunakan bobot serta bias dari hasil *training*, dan tidak adanya proses *backpropagation* setelah proses *feed forward*. Pada proses *testing*, hasil dari proses *flatten* citra *testing* akan dicoba proses klasifikasi. Hasil dari proses *flatten* citra *testing* akan diklasifikasi menggunakan *feed forward*.



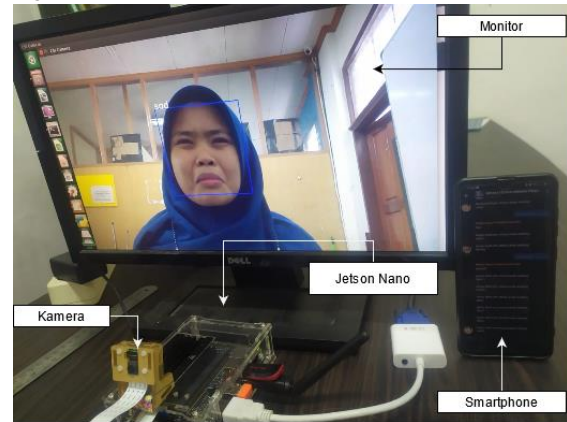
Gambar 3.6 Flowchart Rancangan Telegram Bot

Berdasarkan Gambar 3.7, program pengiriman notifikasi dimulai dengan menginisialisasi token yang merupakan API (*Application Programming Interface*) key dari bot telegram yang dibuat, serta *chat id* dari telegram user. Lalu pengiriman hasil deteksi emosi dilakukan hanya pada 2 keadaan yaitu disaat

terdeteksi emosi *negative* dan saat *user* ingin memonitoring kondisi terkini.

### Implementasi Perangkat Keras

Perancangan sistem pendeteksi gejala awal tantrum melalui ekspresi wajah dengan *convolutional neural network* dapat dilihat pada gambar 4.1.



Gambar 3.7. Implementasi Perangkat Keras

Komponen-komponen yang digunakan dalam implementasi perangkat keras pada sistem sebagai berikut :

1. Nvidia Jetson Nano A02, berfungsi sebagai sistem utama yang mengolah data dan menjalankan program deteksi emosi melalui ekspresi wajah sehingga gejala awal tantrum dapat dideteksi. Agar jetson nano dapat terhubung dengan telegram bot, jetson nano dihubungkan dengan koneksi internet. Pada sistem ini koneksi internet dihubungkan secara *wireless* (WIFI).
2. *Smartphone*, berfungsi untuk media *output* hasil deteksi emosi melalui ekspresi wajah jika terdeteksi emosi *negative* dan *monitoring* emosi yang akan dikirimkan menggunakan bot Telegram yang sebelumnya telah diinstal pada *smartphone*. Agar dapat terhubung dengan sistem, *smartphone* harus terkoneksi dengan internet. *Smartphone* yang digunakan pada sistem ini memiliki spesifikasi sebagai berikut :
  - Chipset : Exynos 9611
  - RAM & Memori : 8 GB & 128 GB
  - OS : Android 11
3. CSI Camera Raspberry Pi V2, berfungsi untuk menangkap tampilan *frame* video secara *realtime* dan mengirimkan datanya ke Jetson Nano untuk diolah dan dilakukan deteksi wajah menggunakan library *opencv*. Kamera yang digunakan yaitu kamera Raspberry Pi V2 dengan resolusi 8.0 *MegaPixel*(MP).
4. *Power*, berfungsi sebagai sumber daya kepada jetson nano
5. *Monitor*, berfungsi untuk memonitoring kerja sistem dan menampilkan hasil *video stream*

### Implementasi Perangkat Lunak

Bahasa pemrograman yang akan digunakan adalah bahasa pemrograman python. Program ini akan dibuat sesuai flowchart rancangan umum proses dengan bantuan library Tensorflow, Keras, dan Numpy, os, dan matplotlib. *Source code import library* dapat dilihat pada gambar 4.3 berikut.

```
#import library
import numpy
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.layers import BatchNormalization, Activation, MaxPooling2D, \
Dense, Input, Dropout, Flatten, Conv2D
from tensorflow.keras.optimizers import Adam
import os
import matplotlib.pyplot as plt
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau

print(f"Tensorflow version: {tf.__version__}")

Tensorflow version: 2.3.0
```

Gambar 3.8. Source Code Import Library

Pada penelitian ini, sistem pengenalan emosi dilakukan dengan menggunakan *convolutional neural network* dengan jumlah data latih 3589 citra wajah dengan emosi yang berbeda-beda. Metode klasifikasi CNN dilakukan dengan cara membagi dataset menjadi data *train* dan data *validation*, dengan persentase pembagian 80% data *train* dan 20 % data *validation*. *Source code convolutional neural network* dapat dilihat pada Gambar 4.4 berikut.

```
#model cnn
model = Sequential()

#1-Conv Layer
model.add(Conv2D(64, (3,3), padding = 'same', input_shape = (48, 48, 1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#2-Conv Layer
model.add(Conv2D(128, (3,3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#3-Conv Layer
model.add(Conv2D(512, (3,3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#4-Conv Layer
model.add(Conv2D(512, (3,3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(7, activation = 'softmax'))
```

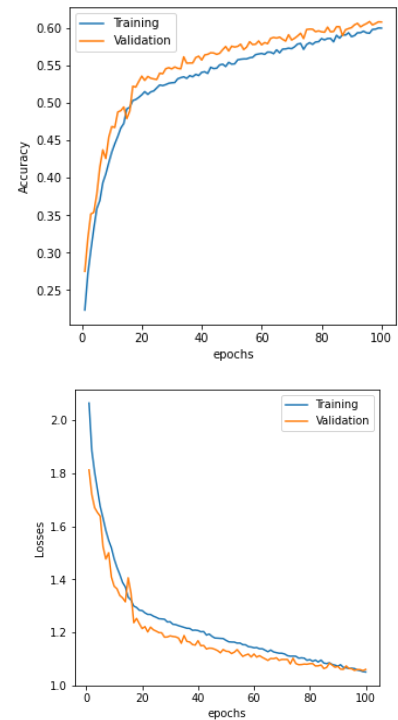
Gambar 3.9. Source Code Model CNN

Langkah selanjutnya, dilakukan *training* sistem dengan inputan data latih dan menggunakan model CNN dengan inisialisasi epochs = 357. Proses *training* dapat dilihat pada gambar berikut.

```
Epoch 0/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5925 - accuracy: 0.7799 - val_loss: 1.0345 - val_
Epoch 1/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5908 - accuracy: 0.7779
Epoch 2/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5908 - accuracy: 0.7779 - val_loss: 1.0302 - val_
Epoch 3/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5810 - accuracy: 0.7838
Epoch 4/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5803 - accuracy: 0.7838 - val_loss: 1.0340 - val_
Epoch 5/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5645 - accuracy: 0.7824
Epoch 6/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5847 - accuracy: 0.7804 - val_loss: 1.0994 - val_
Epoch 7/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5817 - accuracy: 0.7857
Epoch 8/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5817 - accuracy: 0.7857 - val_loss: 1.0313 - val_
Epoch 9/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5896 - accuracy: 0.7806
Epoch 10/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5896 - accuracy: 0.7806 - val_loss: 1.0205 - val_
Epoch 11/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5854 - accuracy: 0.7832
Epoch 12/357: saving model to model_weights.h5
357/357 [100%] - ETA: 0s - loss: 0.5854 - accuracy: 0.7832 - val_loss: 1.0218 - val_
```

Gambar 3.10 Proses Training Model

Untuk mempermudah pengamatan nilai *accuracy* dan nilai *loss*, berikut ditampilkan dalam bentuk grafik *line* seperti pada gambar 4.6 berikut.



Gambar 3.11. Grafik Training Accuracy dan Loss

Langkah selanjutnya, dilakukan pengidentifikasian emosi melalui ekspresi wajah dengan *video stream* dengan menerapkan hasil *training* yang telah dilatih sebelumnya. *Source Codenya* dapat dilihat pada gambar berikut :

```
def gstreamer_pipeline(
    capture_width=1280,
    capture_height=720,
    display_width=1280,
    display_height=720,
    framerate=21,
    flip_method=0,
):
    return (
        "nvarguscamerasrc ! "
        "queue max-size-buffers=1 leaky=downstream ! "
        "video/x-raw(memory:NVMM), "
        "width=(int)%d, height=(int)%d, "
        "format=(string)NV12, framerate=(fraction)%d/1 ! "
        "nvvidconv flip-method=%d ! "
        "video/x-raw, width=(int)%d, height=(int)%d, format=(string)BGRx ! "
        "videoconvert ! "
        "video/x-raw, format=(string)BGR ! appsink max-buffers=1 drop=True"
    )

# load haarcascade untuk deteksi wajah dengan opencv
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# flip image
cap = cv2.VideoCapture(gstreamer_pipeline(flip_method=2), cv2.CAP_GSTREAMER)

#inisialisasi model
from keras.models import model_from_json
model = model_from_json(open("model.json", "r").read())
model.load_weights("model_weights.h5")

#-----
emotions = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')
```

```

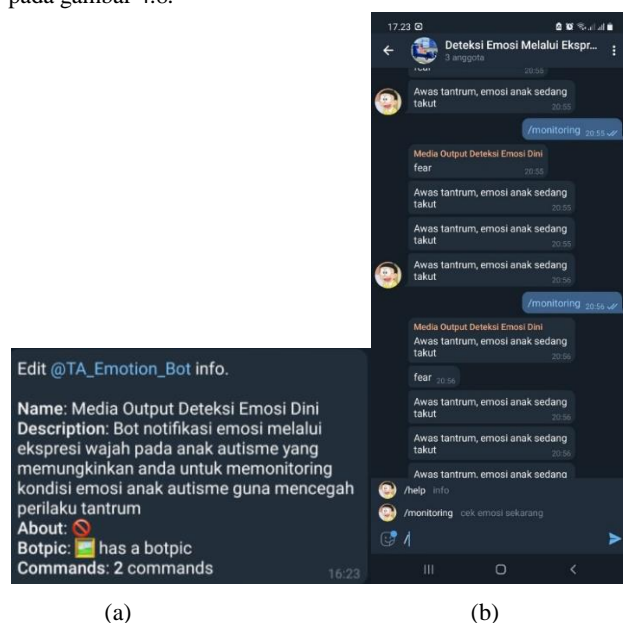
def show_camera():
    if cap.isOpened():
        window_handle = cv2.namedWindow("CSI Camera", cv2.WINDOW_AUTOSIZE)
        # Window
        while cv2.getWindowProperty("CSI Camera", 0) >= 0:
            ret_val, img = cap.read() # capture frame video stream
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, 1.3, 5)
            for (x,y,w,h) in faces:
                cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
            detected_face = img[int(y):int(y+h), int(x):int(x+w)]
            detected_face = cv2.cvtColor(detected_face, cv2.COLOR_BGR2GRAY) #ubah ke gray scale
            detected_face = cv2.resize(detected_face, (48, 48))
            img_pixels = image.img_to_array(detected_face)
            img_pixels = np.expand_dims(img_pixels, axis = 0)
            img_pixels /= 255
            predictions = model.predict(img_pixels)
            max_index = np.argmax(predictions[0])
            emotion = emotions[max_index]
            cv2.putText(img, emotion, (int(x), int(y)), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255), 2)
            cv2.imshow("CSI Camera", img) #show frame
            keyCode = cv2.waitKey(30) & 0xFF
            # tekan ESC key untuk stop
            if keyCode == 27:
                break
        cap.release()
        cv2.destroyAllWindows()
    else:
        print("Unable to open camera")

show_camera()

```

Gambar 3.12 Source Code Testing dari Video Stream

Awalnya bot telegram akan didaftarkan pada aplikasi telegram dengan memakai BotFather. BotFather sendiri merupakan bot yang telah tersedia pada telegram untuk membantu pengguna untuk membuat bot baru dan melakukan beberapa *setting* dasar seperti mendaftarkan *command* baru, mengganti nama bot, memberikan deskripsi pada bot dan lain-lain[24]. Setelah bot didaftarkan bot siap untuk diprogram agar dapat menjalankan perintah sesuai dengan tujuan dari sistem. Pada bot telegram yang dirancang pada sistem didaftarkan dengan memakai *username* @TA\_emotion\_bot dengan deskripsi bot dapat dilihat pada gambar 4.8.



(a)

(b)

Gambar 3.13 (a)Informasi Bot dan (b)Tampilan Notifikasi dari Bot

Dari gambar 4.8 terdapat dua *command* yang telah didaftarkan pada BotFather yakni */monitoring* dan */help*. Command */monitoring* digunakan untuk menampilkan kondisi emosi anak yang sedang terjadi saat itu, sedangkan *command* */help* digunakan untuk menampilkan info dari bot telegram. Agar dapat menampilkan *output* tersebut maka bot telegram diprogram pada sistem dengan menggunakan bahasa python. Pada gambar 4.9

berikut menampilkan *source code* dari bot telegram yang telah dirancang.

```

#mengaktifkan telegram bot
bot = telepot.Bot('1861704341:AAE06xT3Q9xAAtnsPP5nFPycgIwQVdRGDU')
chat_id = -46454498
print (bot.getMe())

#mengirimkan hasil deteksi ke telegram bot
def handle(msg):
    command = msg['text']
    print ('Got command: %s' % command)
    # Menanggapi pesan yang masuk untuk mengirim balasan yg sesuai
    if command == '/help':
        bot.sendMessage(chat_id, 'ini adalah notifikasi dari monitoring emosi negatif pada anak autisme')
    elif command == '/monitoring':
        bot.sendMessage(chat_id, emotion)
    # setiap kali pesan diterima, fungsi akan dipanggil
    MessageLoop(bot, handle).run_as_thread()
    print ('listening...')
    if emotion == 'angry':
        bot.sendMessage(chat_id, 'Awat tantrum, emosi anak sedang marah')
        sleep(2)
    elif emotion == 'sad':
        bot.sendMessage(chat_id, 'Awat tantrum, emosi anak sedang sedih')
        sleep(2)
    elif emotion == 'fear':
        bot.sendMessage(chat_id, 'Awat tantrum, emosi anak sedang takut')
        sleep(2)
    break

```

Gambar 3.14. Source Code Notifikasi ke Bot

Dari gambar 4.9. pemrograman bot menggunakan *library* telepot yang merupakan *library* untuk pemrograman bot. Selain itu pada pemrograman bot, untuk menghubungkan sistem dengan bot diperlukan token yang diberikan BotFather. Token merupakan serangkaian kode yang terdiri dari angka dan huruf dan bersifat rahasia. Pada bot yang dirancang secara umum dapat menerima 2 command yakni */monitoring* dan */help* yang telah didaftarkan pada BotFather. Pada perintah */monitoring*, bot akan mengirimkan hasil monitoring emosi yang terjadi saat itu.

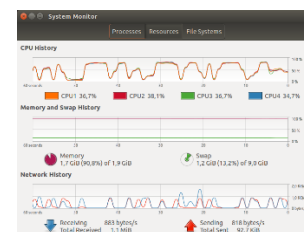
## Implementasi Sistem

Sistem ini dapat melakukan *monitoring* emosi *negative* melalui Telegram bot yang telah rancang. Setelah sistem dihidupkan *video stream* akan memulai menangkap citra wajah dan diteruskan ke sistem untuk dideteksi emosi yang terkandung dalam citra wajah tersebut. Jika hasilnya emosi *negative* (marah, sedih, atau takut) maka hasil deteksi ini diteruskan ke bot telegram agar dapat dilihat oleh pengawas. Jika pengawas ingin mengetahui emosi anak saat itu, pengawas bisa mengirimkan *command* */monitoring* ke bot Telegram dan Telegram akan meneruskan ke sistem dan sistem akan mengirim hasil *monitoring* yang terjadi saat itu ke bot Telegram.

## PENGUJIAN DAN ANALISA

Pengujian dan analisa dilakukan untuk menguji dan menganalisa kerja dari sistem setelah implementasi pada perangkat keras, implementasi pada perangkat lunak, serta implementasi pada sistem secara keseluruhan. Terdapat 3 bagian pengujian dan analisa yang akan dilakukan pada penelitian ini, yaitu pengujian dan analisa perangkat keras, pengujian dan analisa perangkat lunak, dan pengujian dan Analisa sistem secara keseluruhan.

### Pengujian dan Analisa Nvidia Jetson Nano



Gambar 4.1. Penggunaan CPU dan Memory pada Nvidia Jetson Nano Model A02



Dari gambar 4.10 dapat diketahui bahwa saat program dijalankan, program tersebut menggunakan 36.55% dari penggunaan CPU jetson nano secara keseluruhan, penggunaan memori sebesar 90,8% dari keseluruhan memori, dan penggunaan *swap memory* sebesar 13.2%. Kesimpulan dari pengujian ini adalah sistem dapat berjalan dengan baik dengan menggunakan Nvidia Jetson Nano RAM 2GB. Rata-rata *frame* yang dihasilkan dalam pengujian adalah 6 *Frame per Second* (FPS). Hal ini menunjukkan bahwa Nvidia Jetson Nano dan kameranya mampu bekerja dengan baik.

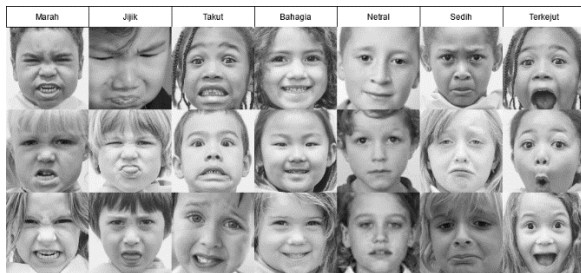
#### Pengujian dan Analisa Model CNN dengan Data Validasi

	Classification Report			
	precision	recall	f1-score	support
angry	0.47	0.62	0.53	958
disgust	0.76	0.23	0.36	111
fear	0.49	0.36	0.41	1024
happy	0.84	0.84	0.84	1774
neutral	0.56	0.63	0.59	1233
sad	0.50	0.48	0.49	1247
surprise	0.76	0.72	0.74	831
accuracy			0.62	7178
macro avg	0.63	0.55	0.57	7178
weighted avg	0.62	0.62	0.62	7178

Gambar 4.2. Classification Report

Hasil yang didapat dari pengujian model CNN dengan data validasi mencapai akurasi 62% dan sudah layak untuk diterapkan dalam sistem.

#### Pengujian dan Analisa model CNN dengan Data Inferensi



Gambar 4.3. Ekspresi Wajah Anak Untuk Setiap Emosi

Hasil dari pengujian bisa dilihat pada Tabel dibawah ini.

Tabel 4.1 Hasil Pengujian Emosi

		PREDIKSI						
		Terk ejut	Sed ih	Tak ut	Mar ah	Jij ik	Net ral	Baha gia
A K T U A L	Terk ejut	25	0	5	0	0	1	4
	Sedih	2	14	3	2	0	13	1
	Taku	9	3	8	1	0	10	4
	Mara h	3	0	5	22	1	0	4
	Jijik	1	3	4	18	2	2	5
	Netra l	1	0	0	0	0	33	1
	Baha gia	2	0	1	0	0	0	32

Dengan melakukan perihal yang sama seperti pengujian diatas, yaitu dengan melakukan prediksi terhadap gambar-gambar

ekspresi untuk masing-masing emosi terdapat 35 gambar dengan total gambar 245, maka didapatkan hasil pengujian sebagai berikut :

Tabel 4.2 Confusion Matrix hasil pengujian emosi terkejut

Total Jumlah Data = 245		Kelas Prediksi (observation)	
		Positif	Negatif
Kelas (Expection)	Aktual	Positif	TP = 25
		Negatif	FN = 10
			FP = 18
			TN = 192
Sensitivity = $TP / (TP + FN)$			71.42%
Accuracy = $(TN + TP) / (TN + TP + FN + FP)$			88.57%

Berdasarkan Tabel 4.2, hasil evaluasi performa model CNN yang digunakan untuk klasifikasi emosi terkejut diperoleh nilai akurasi 88.57%.

Tabel 4.3 Confusion Matrix hasil pengujian emosi sedih

Total Jumlah Data = 245		Kelas Prediksi (observation)	
		Positif	Negatif
Kelas (Expection)	Aktual	Positif	TP = 14
		Negatif	FN = 21
			FP = 6
			TN = 204
Sensitivity = $TP / (TP + FN)$			40%
Accuracy = $(TN + TP) / (TN + TP + FN + FP)$			88.97%

Berdasarkan Tabel 4.3, hasil evaluasi performa data validasi yang digunakan untuk klasifikasi emosi sedih diperoleh nilai akurasi 88.97%.

Tabel 4.4 Confusion Matrix Hasil Pengujian Emosi Takut

Total Jumlah Data = 245		Kelas Prediksi (observation)	
		Positif	Negatif
Kelas (Expection)	Aktual	Positif	TP = 8
		Negatif	FN = 27
			FP = 18
			TN = 192
Sensitivity = $TP / (TP + FN)$			22%
Accuracy = $(TN + TP) / (TN + TP + FN + FP)$			81.63%

Berdasarkan Tabel 4.4, hasil evaluasi performa data *testing* yang digunakan untuk klasifikasi emosi takut diperoleh nilai akurasi 81.63%.

Tabel 4.5 Confusion Matrix Hasil Pengujian Emosi Marah

Total Jumlah Data = 245		Kelas Prediksi (observation)	
		Positif	Negatif
Kelas (Expection)	Aktual	Positif	TP = 22
		Negatif	FN = 13
			FP = 21
			TN = 189
Sensitivity = $TP / (TP + FN)$			62.85%
Accuracy = $(TN + TP) / (TN + TP + FN + FP)$			86.12%

Berdasarkan Tabel 4.5, hasil evaluasi performa data *testing* yang digunakan untuk klasifikasi emosi marah diperoleh nilai akurasi 86.12%.

Tabel 4.6 Confusion Matrix Hasil Pengujian Emosi Jijik

Total Jumlah Data = 245		Kelas Prediksi ( <i>observation</i> )	
		Positif	Negatif
Kelas Aktual ( <i>Expection</i> )	Positif	TP = 2	FN = 33
	Negatif	FP = 1	TN = 209
$Sensitivity = TP / (TP + FN)$		5.71%	
$Accuracy = (TN + TP) / (TN + TP + FN + FP)$		86.12%	

Berdasarkan Tabel 4.6, hasil evaluasi performa data validasi yang digunakan untuk klasifikasi emosi jijik diperoleh nilai akurasi 86.12%.

Tabel 4.7 Confusion Matrix Hasil Pengujian Emosi Netral

Total Jumlah Data = 245		Kelas Prediksi ( <i>observation</i> )	
		Positif	Negatif
Kelas Aktual ( <i>Expection</i> )	Positif	TP = 33	FN = 2
	Negatif	FP = 26	TN = 184
$Sensitivity = TP / (TP + FN)$		94.28%	
$Accuracy = (TN + TP) / (TN + TP + FN + FP)$		88.57%	

Berdasarkan Tabel 4.7, hasil evaluasi performa data *testing* yang digunakan untuk klasifikasi emosi netral diperoleh nilai akurasi 88.57%.

Tabel 4.8 Confusion Matrix Hasil Pengujian Emosi Bahagia

Total Jumlah Data = 245		Kelas Prediksi ( <i>observation</i> )	
		Positif	Negatif
Kelas Aktual ( <i>Expection</i> )	Positif	TP = 32	FN = 3
	Negatif	FP = 19	TN = 191
$Sensitivity = TP / (TP + FN)$		91.42%	
$Accuracy = (TN + TP) / (TN + TP + FN + FP)$		91.02%	

Berdasarkan Tabel 4.8, hasil evaluasi performa data *testing* yang digunakan untuk klasifikasi emosi bahagia diperoleh nilai akurasi 91.02%.

Pengujian model mendapatkan hasil yang maksimal pada emosi bahagia. Hasil kesimpulan pengujian ini dapat dikatakan bahwa klasifikasi emosi wajah menggunakan metode *convolutional neural network* pada data inferensi wajah anak cukup baik.

## Pengujian dan Analisa Jarak dan Posisi Efektif Deteksi Wajah



Gambar 4.4 Hasil Pengujian Jarak dan Posisi Deteksi Wajah

Dari pengujian jarak yang dilakukan, maka didapatkan hasil pengujian jarak efektif deteksi sebagai berikut :

Tabel 4.9 Hasil Pengujian Jarak Efektif Deteksi Wajah

Jarak	Keterangan
60 cm	Terdeteksi
90 cm	Terdeteksi
120 cm	Terdeteksi
150 cm	Terdeteksi
180 cm	Terdeteksi
210 cm	Terdeteksi
240 cm	Terdeteksi
270 cm	Terdeteksi
300 cm	Terdeteksi

Pada Tabel 4.9 dapat dilihat bahwa pada jarak 60cm hingga 300cm, sistem mampu mendeteksi wajah, sehingga didapatkan analisa jarak efektif deteksi yaitu dari 60 cm – 300 cm dengan posisi wajah lurus ke depan.

## Pengujian dan Analisa Sudut Efektif Deteksi Wajah



Gambar 4.14. Cara Pengujian Sudut Efektif

Dari masing-masing sudut, diukur kemampuan deteksi wajah. Berikut beberapa hasil pengujian yang digunakan untuk pengujian sudut efektif deteksi wajah dapat dilihat pada tabel 4.10 dibawah.

Tabel 4.10 Pengujian Deteksi Wajah pada Beberapa Sudut

Posisi Sudut	Hasil Deteksi	Keterangan Deteksi
90° hadap kiri		Tidak terdeteksi
60° hadap kiri		Tidak terdeteksi

30° hadap kiri		terdeteksi
0° (hadap depan)		Terdeteksi
30° hadap kanan		Terdeteksi
60° hadap kanan		Tidak terdeteksi
90° hadap kanan		Tidak terdeteksi

Pada Tabel 4.9 dapat dilihat bahwa pada sudut 0-30° hadap kanan dan kiri, sistem mampu mendeteksi wajah. Namun pada sudut 60-90° hadap kanan dan kiri sistem tidak bisa mendeteksi wajah karena Sebagian wajah tidak kelihatan. Sehingga didapatkan analisa sudut efektif deteksi yaitu dari 0-30° hadap kanan dan kiri.

#### Pengujian dan Analisa Deteksi Multi Wajah

Pengujian dilakukan untuk menguji kinerja sistem dalam mendeteksi lebih dari satu wajah. Hasil pengujian dapat dilihat pada tabel 4.10 dibawah.

Tabel 4.10 Pengujian Deteksi Wajah pada Beberapa Wajah

Jumlah Wajah	Hasil Deteksi	Keterangan Deteksi
1		Terdeteksi
2		Terdeteksi
3		Terdeteksi

4		Terdeteksi
---	--	------------

Dari hasil pengujian kinerja sistem dalam melakukan deteksi banyak wajah, dapat dikatakan kualitas deteksi untuk banyak wajah akurat. Dalam melakukan pengujian dengan beberapa wajah, didapatkan kesimpulan bahwa sistem mampu mendeteksi semua wajah yang tertangkap kamera dengan cukup baik.

#### Pengujian dan Analisa Bot Telegram



Gambar 4.5. Hasil Pengujian Bot Telegram

Tabel 4.11. Hasil Pengujian Telegram

Pengujian Ke-	Pengiriman Notifikasi	Respon Waktu (detik)	Keterangan
1	Diterima	2.52	Berhasil
2	Diterima	1.38	Berhasil
3	Diterima	1.38	Berhasil
4	Diterima	1.25	Berhasil
5	Diterima	1.65	Berhasil
6	Diterima	1.25	Berhasil
7	Diterima	1.25	Berhasil
8	Diterima	1.24	Berhasil
9	Diterima	1.18	Berhasil
10	Diterima	1.26	Berhasil
Respon Waktu Rata-rata		1.436 detik	

Pengujian dikatakan berhasil apabila notifikasi yang dikirimkan ke telegram bot tampil pada interface telegram dengan chat id yang telah didaftarkan pada sistem. Pada Tabel 4.11 dapat dilihat bahwa sistem mampu mencapai tingkat keberhasilan 100%, dengan respon waktu rata-rata adalah 1.436 detik.

Pengujian yang kedua dilakukan dengan cara mengirimkan perintah */monitoring* kepada bot telegram untuk menguji apakah bot telegram dapat menjalankan perintah dengan baik serta menampilkan hasil yang sesuai dengan yang diberikan oleh sistem. Pada tabel 4.11 merupakan tabel yang menampilkan hasil pengujian sebanyak 7 kali percobaan dengan perintah */monitoring*.



Gambar 4.6. Hasil Pengujian Command */monitoring*

Tabel 4.12. Pengujian Command */monitoring*

Pengujian Ke-	Kondisi	Keluaran pada Bot Telegram	Keterangan
1	Menerima <i>/monitoring</i>	<i>Surprise</i>	Berhasil
2	Menerima <i>/monitoring</i>	<i>Angry</i>	Berhasil
3	Menerima <i>/monitoring</i>	<i>Neutral</i>	Berhasil
4	Menerima <i>/monitoring</i>	<i>Angry</i>	Berhasil
5	Menerima <i>/monitoring</i>	<i>Fear</i>	Berhasil

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah Keberhasilan}}{\text{Jumlah Percobaan}} \times 100\%$$

$$\frac{5}{5} \times 100\% = 100\%$$

Dari hasil pengujian pada tabel 4.12 dapat diketahui bahwa perintah */monitoring* yang diberikan kepada bot dapat memberikan keluaran yang sesuai seperti yang telah dirancang sebelumnya. Selain itu bot juga dapat menampilkan *output* dari sistem kepada pengguna melalui bot sesuai dengan hasil pemrosesan pada sistem.

#### Pengujian dan Analisa Sistem Secara Keseluruhan

Pengujian sistem ini dilakukan dengan cara menampilkan data uji di layar dan kemudian ditangkap oleh kamera Raspberry Pi untuk diolah di Jetson Nano. Bahan uji disini adalah video anak autisme berjumlah 35 video dimana video tersebut hasil rekaman anak autisme di sekolah SLB Autisma Yppa Padang. Hasil dari pengujian sistem secara keseluruhan dengan data inferensi video anak autisme bisa dilihat pada tabel 4.13. dibawah ini.

Tabel 4.13. Hasil Pengujian Sistem Secara Keseluruhan

Video	Emosi yang terkandung	Emosi yang terdeteksi	Notifikasi Telegram	Keterangan
-------	-----------------------	-----------------------	---------------------	------------

1.mp4	Netral	Netral	Tidak ada	Berhasil
2.mp4	Netral	Netral	Tidak ada	Berhasil
		Takut	Ada	
3.mp4	Netral	Bahagia	Tidak ada	Tidak berhasil
		Bahagia	Tidak ada	
4.mp4	Netral	Sedih	Ada	Berhasil
		Netral	Tidak ada	
5.mp4	Netral	Bahagia	Tidak ada	Berhasil
		Bahagia	Tidak ada	
6.mp4	Netral	Netral	Tidak ada	Berhasil
		Sedih	Ada	
7.mp4	Netral	Takut	Ada	Berhasil
		Takut	Ada	
8.mp4	Netral	Netral	Tidak ada	Berhasil
		Sedih	Ada	
9.mp4	Netral	Bahagia	Tidak ada	Tidak berhasil
		Takut	Ada	
10.mp4	Netral	Netral	Tidak ada	Berhasil
		Sedih	Ada	
11.mp4	Netral	Bahagia	Tidak ada	Tidak berhasil
		Netral	Tidak ada	
12.mp4	Netral	Netral	Tidak ada	Berhasil
		Sedih	Ada	
13.mp4	Netral	Bahagia	Tidak ada	Tidak berhasil
		Sedih	Ada	
14.mp4	Netral	Bahagia	Tidak ada	Berhasil
		Netral	Tidak ada	
15.mp4	Netral	Bahagia	Tidak ada	Berhasil
		Sedih	Ada	
16.mp4	Netral	Bahagia	Tidak ada	Berhasil
		Netral	Tidak ada	
17.mp4	Netral	Bahagia	Tidak ada	Berhasil
		Netral	Tidak ada	
18.mp4	Netral	Bahagia	Tidak ada	Berhasil
		Netral	Tidak ada	



19.mp4	Netral	Netral	Tidak ada	Berhasil
	Sedih	Sedih	Ada	Tidak berhasil
		Takut	Ada	
	Marah	Bahagia	Tidak ada	Berhasil
Marah		Ada		
20.mp4	Netral	Netral	Tidak ada	Berhasil
		Takut	ada	
21.mp4	Bahagia	Bahagia	Tidak ada	Berhasil
	Netral	Netral	Tidak ada	Berhasil
		Sedih	Ada	
22.mp4	Netral	Netral	Tidak ada	Tidak berhasil
		Terkejut	Tidak ada	
		Takut	Ada	
23.mp4	Netral	Netral	Tidak ada	Tidak berhasil
		Bahagia	Tidak ada	
		Sedih	Ada	
	Bahagia	Bahagia	Tidak ada	Berhasil
Netral	Tidak ada			
24.mp4	Sedih	Takut	Ada	Berhasil
		Sedih	Ada	
	Netral	Netral	Tidak ada	Berhasil
25.mp4	Netral	Netral	Tidak ada	Berhasil
		Terkejut	Tidak ada	
	Terkejut	Terkejut	Tidak ada	Berhasil
		Netral	Tidak ada	
	Bahagia	Bahagia	Tidak ada	Berhasil
Marah	Marah	Ada	Berhasil	
26.mp4	Bahagia	Netral	Tidak ada	Tidak berhasil
	Netral	Netral	Tidak ada	Berhasil
		Sedih	Ada	
27.mp4	Netral	Netral	Tidak ada	Berhasil
		Bahagia	Tidak ada	
	Marah	Takut	Ada	Berhasil
		Marah	Ada	
	Sedih	Sedih	Ada	Berhasil
Netral		Tidak ada		
28.mp4	Bahagia	Bahagia	Tidak ada	Berhasil
		Netral	Tidak ada	
	Netral	Netral	Tidak ada	Berhasil
		Takut	ada	
29.mp4	Netral	Terkejut	Tidak ada	Tidak berhasil
		Netral	Tidak ada	
		Sedih	Ada	
30.mp4	Netral	Netral	Tidak ada	Berhasil
		Bahagia	Tidak ada	
	Bahagia	Terkejut	Tidak ada	Tidak berhasil
		Bahagia	Tidak ada	
Takut		Ada		
31.mp4	Netral	Bahagia	Tidak ada	Berhasil
		Netral	Tidak ada	
32.mp4	Marah	Bahagia	Tidak ada	Tidak berhasil
		Sedih	Ada	
	Netral	Netral	Tidak ada	Berhasil
		Bahagia	Netral	
33.mp4	Bahagia	Bahagia	Tidak ada	Tidak berhasil
		Takut	Ada	
	Netral	Terkejut	Tidak ada	Berhasil
		Netral	Tidak ada	
		Jijik	Marah	
34.mp4	Netral	Netral	Tidak ada	Berhasil
		Sedih	Ada	

35.mp4	Marah	Bahagia	Tidak ada	Tidak berhasil
		Takut	Ada	
	Bahagia	Bahagia	Tidak ada	Berhasil
		Netral	Tidak ada	
	Netral	Netral	Tidak ada	Berhasil
		Takut	Ada	

Presentase keberhasilan :

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah Keberhasilan}}{\text{Jumlah Percobaan}} \times 100\%$$

$$\frac{49}{68} \times 100\% = 72.05\%$$

Berdasarkan pada pengujian sistem secara keseluruhan yang telah dilakukan pada data inferensi anak autisme didapatkan hasil bahwa sistem bisa mendeteksi ekspresi wajah yang tertangkap kamera dengan presentase keberhasilan dalam mendeteksi emosi dengan benar sebesar 72.05%, yang mana nilai ini cukup baik dalam mendeteksi emosi wajah. Sehingga dapat disimpulkan bahwa sistem deteksi emosi wajah dengan metode CNN yang telah dibuat dapat diimplementasikan kepada anak autisme.

## KESIMPULAN

Dari hasil implemmentasi sistem dan pengujian sistem, maka didapatkan kesimpulan :

1. Sistem dapat mendeteksi emosi pada banyak wajah yang tertangkap kamera dengan jarak ideal untuk melakukan pendeteksian adalah 60cm hingga 300cm dengan posisi wajah menghadap ke kamera.
2. Sistem dapat menangkap dan mendeteksi ekspresi wajah pada anak autisme dengan akurasi sebesar 66.66% dari 30 video data uji.
3. Dalam evaluasi performa CNN dengan data inferensi wajah anak dengan total data sebanyak 245 data dengan 35 data untuk masing-masing emosi, didapatkan akurasi tertinggi pada emosi bahagia dan diikuti dengan emosi sedih, terkejut, netral, marah, jijik, dan takut dengan akurasi masing-masing emosi berurut sebesar 91.02%, 88.97%, 88.57%, 88.57%, 82.16%, 82.16%, dan 81.63%.
4. Sistem dapat mengirim notifikasi ke bot telegram dengan tingkat keberhasilan 100% dan waktu respon rata-rata adalah 1.436 detik.
5. Sistem dapat menampilkan hasil deteksi emosi melalui *video stream* dan notifikasi telegram sehingga bisa diterapkan sebagai pendeteksi gejala awal tantrum pada anak autisme melalui ekspresi wajah.

## DAFTAR PUSTAKA

- [1] "Data & Statistics on Autism Spectrum Disorder." *Center for Disease Control and Prevention*, 2020. <https://www.cdc.gov/ncbddd/autism/data.html> (diakses Jan 29, 2021).
- [2] G. Stubbs, K. Henley, dan J. Green, "Autism: Will vitamin D supplementation during pregnancy and early

- childhood reduce the recurrence rate of autism in newborn siblings?," *Med. Hypotheses*, vol. 88, hal. 74–78, 2016, doi: 10.1016/j.mehy.2016.01.015.
- [3] Y. Astuti, "Perilaku Tantrum Anak Usia 5-6 Tahun," *J. Chem. Inf. Model.*, 2016.
- [4] A. Nadhiroh, "Strategi penanganan anak pada Fase Tantrum," *Univ. Islam Negeri Sunan Ampel Surabaya Fak. Tarb. Dan Kegur. Progr. Stud. Paud*, vol. 01, no. 03, hal. 1–72, 2018.
- [5] T. I. Kusumawati, "Komunikasi Verbal Dan Nonverbal," *J. Pendidik. dan Konseling*, vol. 6, no. 2, hal. 83–98, 2016.
- [6] "Ekspresi Wajah Manusia," *lampost*, 2019. <https://www.lampost.co/berita-ekspresi-wajah-manusia.html> (diakses Okt 13, 2020).
- [7] N. A. T. Md. Forhad Ali, Mehenag Khatun, "Facial Emotion Detection using Neural Network," no. August, hal. 1037–1041, 2020, doi: 10.1109/icces48766.2020.9137919.
- [8] S. Anwar, "Real Time Face Expression Recognition of Children with Autism Real Time Face Expression Recognition of Children with Autism," no. November, 2016.
- [9] N. FADHILAH, "Sistem identifikasi dan monitoring emosi dasar manusia melalui ekspresi wajah dengan metode," Universitas Andalas, 2019.
- [10] J. Le, "Convolutional Neural Networks: The Biologically-Inspired Model," *Apr 19, 2018*. [https://www.codementor.io/@james\\_aka\\_yale/convolutional-neural-networks-the-biologically-inspired-model-iq6s48zms](https://www.codementor.io/@james_aka_yale/convolutional-neural-networks-the-biologically-inspired-model-iq6s48zms) (diakses Feb 04, 2021).
- [11] V. Shuman, E. Clark-Polner, B. Meuleman, D. Sander, dan K. R. Scherer, "Emotion perception from a componential perspective," *Cogn. Emot.*, vol. 31, no. 1, hal. 47–56, 2017, doi: 10.1080/02699931.2015.1075964.
- [12] Ben Dickson, "What is computer vision?," *January 14, 2019*. <https://bdtectalks.com/2019/01/14/what-is-computer-vision/>.
- [13] E. Maggiori, S. Member, Y. Tarabalka, G. Charpiat, dan P. Alliez, "Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification," hal. 1–13, 2016.
- [14] T. Andono, P. N., & Sutojo, *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi, 2017.
- [15] Muhammad Raid Naufal dan Rahmi Eka Putri, "Sistem Klasifikasi Penumpang Bus Trans Padang Berdasarkan Pakaian Menggunakan Metode Image Processing," *Chipset*, vol. 1, no. 02, hal. 79–90, 2020, doi: 10.25077/chipset.1.02.79-90.2020.
- [16] R. Conlin, K. Erickson, J. Abbate, dan E. Kolen, "Keras2c: A library for converting Keras neural networks to real-time compatible C," *Eng. Appl. Artif. Intell.*, vol. 100, 2021, doi: 10.1016/j.engappai.2021.104182.
- [17] I. R. Eisa, "Implementasi Voice Recognition Dan Sensor Ultrasonik Pada Televisi," *Chipset*, vol. 02, hal. 1–6, 2021, doi: 10.25077/chipset.2.02.1-6.2021.
- [18] R. E. Putri, "Rancang Bangun Sistem Deteksi Kecepatan Kendaraan di Wilayah Zona Selamat Sekolah (ZoSS) Berbasis Mini PC," *JITCE (Journal Inf. Technol. Comput. Eng.)*, vol. 5, no. 01, hal. 41–51, 2021, doi: 10.25077/jitce.5.01.41-51.2021.
- [19] N. Developer, "Jetson Nano Developer Kit." <https://developer.nvidia.com/embedded/jetson-nano-developer-kit> (diakses Feb 03, 2021).
- [20] A. R. Fredianto, "RANCANG BANGUN MODUL KAMERA DETEKSI KECEPATAN KENDARAAN MENGGUNAKAN NVIDIA JETSON NANO DENGAN METODE IMAGE PIXEL MANIPULATION AND CALCULATION," 2018.
- [21] A. A. Suzen, B. Duman, dan B. Sen, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," *HORA 2020 - 2nd Int. Congr. Human-Computer Interact. Optim. Robot. Appl. Proc.*, hal. 3–7, 2020, doi: 10.1109/HORA49412.2020.9152915.
- [22] "Camera Module V2," *Raspberry Pi Foundation*. <https://www.raspberrypi.org/products/camera-module-v2/> (diakses Apr 20, 2021).
- [23] "Telegram FAQ," <https://telegram.org/faq#q-what-is-telegram-what%02do-i-do-here> (diakses Jan 20, 2021).
- [24] N. Hema dan J. Yadav, "Secure Home Entry Using Raspberry Pi with Notification via Telegram," *2020 6th Int. Conf. Signal Process. Commun. ICSC 2020*, hal. 211–215, 2020, doi: 10.1109/ICSC48311.2020.9182778.



## BIOGRAPHY

### Dini Ramadhani Salsabila

Lahir di Jakarta pada tanggal 06 Januari 2000. Menempuh pendidikan di SDN 117876 Cikampak, MTSS Ppm Ar-Rasyid Pinang Awan, dan SMAIT Al-Fityan Medan. Kemudian menempuh pendidikan di Jurusan Teknik Komputer, Fakultas Teknologi Informasi, Universitas Andalas.