

# Perancangan Prototype Brankas Menggunakan Sistem Pengenalan Wajah dengan Metode Convolutional Neural Network (CNN)

Andik Yulianto\*<sup>1</sup>, Willy Andreas<sup>2</sup>, Sabariman<sup>3</sup>

<sup>1,3</sup>Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, Jalan Gajah Mada Baloi Sei Ladi, Batam

<sup>2</sup>Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Internasional Batam, Jalan Gajah Mada Baloi Sei Ladi, Batam

## ARTICLE INFO

### Keywords:

Convolutional Neural Network (CNN), Face Recognition, Safe-Deposit Box

**Received:** March 11, 2023

**Revised:** June 20, 2023

**Accepted:** June 25, 2023

### \*Corresponding author:

E-mail: [andik@uib.ac.id](mailto:andik@uib.ac.id) (Andik Yulianto)

### DOI:

[http://dx.doi.org/10.37253/telcomatics.v8i1.7852](https://dx.doi.org/10.37253/telcomatics.v8i1.7852)

## ABSTRACT

A safe-deposit box is a box used for keeping precious items. Safe-deposit boxes are designed to be difficult for people to open by force. There are various security systems that may be used in it, such as mechanical key, combinational lock, PIN, etc. However, a safe-deposit box is still prone to unpermitted access because anyone who knows the PIN or possesses the key is still able to open it.

This research aims to create a safety-box prototype which has a face recognition system implemented on it to ensure no unauthorized person may access this box. Classification on safe owners is considered success if the percentage output in corresponding classes is at least 90 %. Classification on "unknown" class is considered success if the result is at least 90 % or percentage on each class is lower than 90 %. Accuracy for each class is 0 %, 71.43 %, dan 100 %.

## INTISARI

Brankas adalah sebuah kotak penyimpanan yang berfungsi untuk menyimpan benda-benda berharga. Brankas dirancang agar sulit dibuka secara paksa. Sistem keamanan yang digunakan bermacam-macam, seperti kunci mekanis, kunci kombinasi (combinational lock), kode PIN, dan sebagainya. Namun brankas masih beresiko untuk dibobol jika seseorang mengetahui kode pin atau memiliki atribut yang diperlukan untuk membuka brankas.

Penelitian ini bertujuan untuk membuat sebuah prototype brankas yang dilengkapi dengan sistem pengenalan wajah (face recognition) untuk memastikan brankas tidak diakses oleh pihak yang tidak berwenang. Klasifikasi untuk pemilik brankas dianggap berhasil jika output menghasilkan persentase minimal 90 %. Klasifikasi "unknown" dianggap berhasil jika output menghasilkan persentase di atas 90 % atau output ketiga kelas kurang dari 90 %. Tingkat akurasi dari masing-masing kelas adalah 0 %, 71.43 %, dan 100 %.

## I. PENDAHULUAN

Brankas adalah sebuah kotak penyimpanan yang berfungsi untuk menyimpan benda-benda berharga. Brankas banyak digunakan karena terbuat dari bahan yang sangat keras sehingga sulit untuk dibuka secara paksa dan beberapa jenis brankas dirancang untuk tahan terhadap api. Sistem keamanan yang digunakan brankas bermacam-macam, ada yang menggunakan kunci mekanis, kunci kombinasi (*combination lock*), kode pin (digital), dan sebagainya.

Walapun dirancang untuk mencegah pencurian, namun brankas tetap berpotensi untuk dibobol. Siapapun yang mengetahui kode pin atau memiliki atribut yang dibutuhkan bisa mengakses brankas dengan mudah. Oleh karena itu diperlukan sistem pengaman yang bisa mengenali siapa yang mengakses brankas untuk memastikan hanya orang-orang tertentu yang bisa mengakses brankas.

Berdasarkan permasalahan di atas, dirancang sebuah prototype brankas yang menggunakan sistem pengenalan wajah (*face recognition*) dengan metode CNN (Convolutional Neural Network) sehingga mampu mengurangi resiko

terjadinya pembobolan oleh pihak yang tidak bertanggung jawab. Dengan menggunakan wajah sebagai masukan (*input*), maka hanya orang-orang tertentu yang bisa mengakses brankas.

## II. KAJIAN PUSTAKA

### A. Penelitian Terdahulu

Penelitian tentang pengenalan wajah dengan menggunakan metode Eigenface telah dilakukan pada [1]. Persentase keberhasilan yang didapatkan dalam mengenali wajah adalah 94.74%. Dataset yang digunakan terdiri dari 20 gambar dari 152 orang sehingga ada 3.040 gambar yang disimpan.

Prinsip kerja dari metode Eigenface adalah menggunakan beberapa gambar wajah asli dari orang yang berbeda-beda untuk membuat gambar-gambar baru yang mengandung fitur-fitur dari setiap wajah yang digunakan. Gambar baru ini disebut dengan Eigenface. Dengan menggabungkan beberapa Eigenface dengan persentase yang berbeda-beda, maka wajah dari setiap orang dapat dibentuk

Penelitian lain mengenai pengenalan wajah juga [2] dilakukan dengan menggunakan metode Local Binary Patterns (LBP). Persentase gambar yang dikenali pada penelitian ini adalah 99% dengan gambar yang diuji berjumlah 2000.

Metode LBP menggunakan gambar grayscale sebagai input dan gambar dibagi beberapa bagian. Cara kerja dari metode ini adalah dengan membandingkan nilai dari pixel di tengah dengan beberapa pixel di sekelilingnya. Jika nilai pixel di pinggir lebih besar dari nilai pixel di tengah, maka hasil yang didapatkan adalah "1" dan untuk kondisi sebaliknya bernilai "0". Setiap digit yang didapatkan dari hasil perbandingan digabungkan membentuk suatu bilangan biner yang baru lalu diubah ke bentuk desimal. Setiap bilangan yang didapat dari proses ini digabungkan ke dalam histogram dan dibandingkan dengan gambar yang disimpan di dalam dataset [2].

Pada [3] penelitian mengenai pengenalan wajah dengan menggunakan metode Convolutional Neural Network telah dilakukan. Penelitian ini menggunakan wajah dari 50 individu yang berbeda-beda sebagai dataset dengan masing-masing berjumlah 15 gambar per individu. Performa dari CNN diuji dengan menggunakan beberapa input dengan ukuran pixel yang berbeda-beda pada dua percobaan yang berbeda. Percobaan pertama adalah pengujian untuk melihat apakah sistem dapat mengenali wajah dengan tepat. Percobaan kedua adalah pengujian untuk melihat apakah individu yang dikenali oleh sistem sesuai dengan salah satu dari 5 prediksi yang diberikan. Hasil percobaan pertama menunjukkan tingkat keberhasilan tertinggi yang didapat adalah 94.8%, sedangkan tingkat keberhasilan tertinggi pada percobaan kedua 98.8% [3].

Penelitian [4] membahas tentang perbandingan antara performa dari beberapa metode pengenalan wajah. Metode yang dibandingkan adalah Eigenfaces, Fisherface, Local Binary Pattern (LBP), dan Convolutional Neural Network (CNN). Fisherface merupakan algoritma Eigenface yang telah dimodifikasi. Eigenface dan Fisherface tidak mengalami peningkatan yang konsisten atau bahkan mengalami penurunan pada tingkat keakuratan seiring dengan bertambahnya jumlah gambar yang digunakan untuk pelatihan. Tingkat keakuratan pada LBP relatif lebih rendah dari CNN. Selain itu, CNN memiliki performa yang baik walaupun jumlah data yang digunakan relatif sedikit. Hasil penelitian yang didapatkan adalah CNN merupakan metode yang paling efisien dan memiliki performa terbaik [4].

## B. Convolutional Neural Network (CNN)

*Convolutional Neural Network* adalah salah satu jenis jaringan saraf tiruan yang bekerja dengan menggunakan konvolusi (*convolution*) untuk mengekstrak ciri khas dari suatu *input*. CNN terinspirasi dari penelitian yang dilakukan oleh Hubel dan Wiesel terhadap korteks visual pada kucing [5]. Beberapa sel pada korteks visual sensitif terhadap objek tertentu. Objek yang berbeda akan mengaktifkan sel saraf yang berbeda.

Tahap pertama dari CNN adalah melakukan konvolusi pada input. Sebuah matriks filter yang berukuran  $n \times n$ , dimana  $n$  lebih kecil dari ukuran input, diletakkan di dalam matriks input di semua posisi yang memungkinkan di dalam matriks. Pada setiap posisi yang ditempati, perkalian dot dilakukan antara matriks kecil yang ditempati filter dengan matriks filter

tersebut. Hasil perkalian dot dimasukkan ke dalam matriks baru sesuai dengan posisi yang sesuai dengan posisi penempatan filter di matriks sebelumnya. Matriks baru ini disebut dengan *feature map*.

Di dalam CNN, fungsi aktivasi (*activation function*) digunakan untuk melakukan *mapping* terhadap nilai pada *feature map*. Fungsi aktivasi digunakan setelah konvolusi dilakukan. Ada beberapa macam fungsi aktivasi yang dikenal seperti sigmoid, tanh, ReLU (*Rectified Linear Unit*), dan sebagainya. Dibanding fungsi aktivasi yang lain, ReLU memiliki kelebihan dari segi kecepatan dan keakuratan sehingga ReLU banyak digunakan di dalam CNN, menggantikan fungsi aktivasi lainnya [6].

*Padding* adalah proses penambahan pixel bernilai "0" di sekeliling *feature map* sebelum konvolusi dilakukan. Salah satu efek yang disebabkan dari konvolusi adalah mengurangi dimensi matriks pada layer selanjutnya dan informasi di bagian pinggir *feature map* cenderung hilang [6]. *Padding* dapat digunakan untuk mempertahankan atau bahkan memperbesar dimensi pada *feature map* sehingga informasi pada gambar tidak berkurang.

*Pooling* adalah proses pengurangan dimensi pada *feature map* dengan mempertahankan informasi paling penting. *Feature map* dibagi menjadi bagian-bagian kecil lalu sebuah nilai baru akan ditentukan berdasarkan nilai-nilai yang ada di bagian kecil tersebut. Ada beberapa jenis *pooling* yang dapat digunakan, seperti *max pooling*, *average pooling*, dan sebagainya. Namun *max pooling* merupakan jenis yang lebih sering digunakan dibanding jenis lainnya [6].

Pada tahap ini, *feature map* yang didapatkan dari tahap-tahap sebelumnya diubah menjadi sebuah vektor yang panjang untuk dijadikan sebagai sebuah input ke dalam *fully connected layers*, yang merupakan tahap akhir dari CNN.

*Fully connected layers* merupakan jaringan saraf yang memiliki bentuk yang mirip dengan jaringan saraf pada umumnya. Dengan memanfaatkan ciri khas dari suatu gambar yang telah diekstrak dari proses konvolusi, klasifikasi dari suatu gambar dilakukan di tahap ini. Setiap nilai dari *feature map* dimasukkan sebagai input sehingga jumlah bobot yang diperlukan untuk setiap sambungan saraf sangat banyak. Oleh karena itu sebagian besar parameter di dalam CNN digunakan pada *fully connected layers*. *Backpropagation* digunakan untuk melatih setiap bobot yang ada di dalam *fully connected layers*.

Prinsip kerja dari *backpropagation* adalah dengan menghitung gradien dari *error* atau *loss* pada output terhadap setiap bobot. Nilai bobot yang baru diatur dengan proporsi yang sesuai dengan hasil perhitungan tersebut. Pengaturan dari nilai bobot ini bertujuan untuk mengurangi nilai dari *error* yang akan dihasilkan setelah bobot diperbaharui [7].

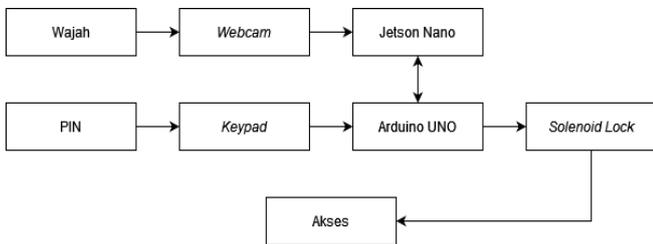
*Stochastic Gradient Descent* merupakan salah satu jenis *gradient descent* yang berfungsi untuk mengurangi *error* pada output. Algoritma *Stochastic Gradient Descent* lebih cepat, lebih dapat diandalkan, dan lebih tidak rentan untuk mencapai *local minima* yang buruk dibanding *gradient descent* standar [8]. *Stochastic Gradient Descent* dirumuskan dengan:

$$w_{t+1} = w_t - \varepsilon_t \nabla_w J(z, w_t) \quad (1)$$

di mana  $w_{t+1}$  adalah nilai bobot baru,  $w_t$  adalah nilai bobot lama,  $\mathcal{E}_t$  adalah penguatan (gain), dan  $\nabla_w J(z, w_t)$  adalah gradien *loss function* terhadap  $w$ .

### III. METODE PENELITIAN

Pada rancangan penelitian ini, ada 5 komponen utama yang digunakan, yaitu webcam, membrane keypad, Jetson Nano, Arduino UNO, dan solenoid lock. Webcam dan keypad digunakan untuk menerima input berupa gambar wajah dan nomor PIN. Webcam diakses melalui Jetson Nano, sedangkan membrane keypad diakses melalui Arduino UNO. Solenoid lock digunakan sebagai alat untuk membuka dan menutup pintu brankas. Pada keadaan normal solenoid lock berada di dalam posisi terkunci. Solenoid lock akan terbuka hanya ketika Arduino memberikan perintah melalui pin GPIO.



Gambar 1. Diagram blok dari prototype brankas

Alur kerja dari prototype brankas ini dapat dilihat pada Gambar 2. Tombol “A” pada keypad ditekan untuk menandakan bahwa ada orang yang ingin mengakses brankas. Gambar wajah kemudian ditangkap oleh webcam dan diproses dengan menggunakan metode CNN untuk mengidentifikasi siapa yang sedang mengakses brankas. Jika tingkat kemiripan wajah berada di atas 90%, pengguna diharuskan untuk memasukkan kode PIN melalui keypad yang terpasang di brankas. Jika persyaratan untuk mengakses brankas telah terpenuhi, maka solenoid lock akan aktif selama 10 detik sehingga brankas bisa dibuka. Namun jika wajah tidak dikenali atau pin yang dimasukkan salah, maka brankas tidak akan terbuka.

Di dalam program pengenalan wajah ini, dua orang dijadikan sebagai pemilik brankas, yaitu “erwin” dan “willy”. Orang-orang yang bukan merupakan pemilik brankas dimasukkan ke dalam kategori “unknown”. Secara keseluruhan, ada 3 kelas yang dilatih pada sistem CNN ini, yaitu “erwin”, “unknown”, dan “willy”.

Program pengenalan wajah dirancang memiliki 3 kali percobaan untuk mengenali wajah jika tidak ada kelas yang memiliki hasil di atas 90%. Jika ketiga percobaan gagal, maka sistem akan menganggap orang yang mengakses brankas sebagai “unknown”.

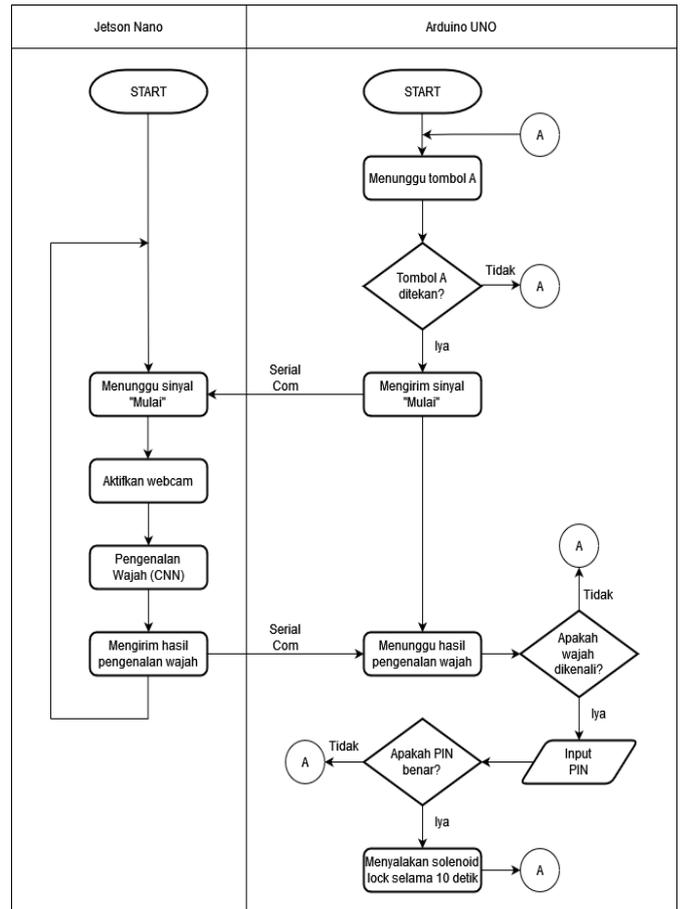
Pada proses memasukkan PIN, kesempatan yang diberikan adalah sebanyak 3 kali. Jika seseorang salah memasukkan PIN sebanyak 3 kali, maka akses akan ditolak dan proses untuk membuka brankas harus dimulai dari awal kembali.

### C. Dataset

Dataset yang digunakan untuk melatih CNN terdiri dari 700 gambar wajah 2 pemilik brankas beserta sejumlah orang

yang dipilih secara acak. Wajah yang acak ini digunakan untuk melatih CNN mengenali orang yang bukan merupakan pemilik brankas (“unknown”). Dua orang yang dijadikan sebagai pemilik brankas ini adalah “erwin” dan “willy”. Secara keseluruhan, ada 3 kelas yang dilatih pada sistem CNN ini, yaitu “erwin”, “unknown”, dan “willy”.

Sebanyak 400 gambar wajah kedua pemilik brankas diambil oleh penulis, sedangkan 300 wajah acak diambil dari dataset yang dibuat oleh Karras, dkk [9] dan Cheng, dkk [10]. Dari 700 gambar yang digunakan, 140 gambar (20%) dipilih sebagai bahan validasi hasil pelatihan CNN.



Gambar 2. Flowchart dari prototype brankas

### D. Arsitektur CNN

Desain dari arsitektur CNN ditentukan secara empiris untuk mendapatkan hasil yang relatif akurat. Input ke CNN yang digunakan adalah gambar dengan ukuran 224 x 224 pixel. Pada akhir sistem, terdapat tiga neuron yang mewakili kedua pemilik brankas dan orang yang tak diketahui. Masing-masing neuron akan menghasilkan nilai yang menunjukkan tingkat kemiripan antara wajah yang ada di input dengan wajah salah satu pemilik.

Dalam melatih CNN yang telah dirancang, *learning rate* yang digunakan adalah 0.0001 dan *loss function* yang digunakan adalah *categorical cross entropy*. Jumlah epoch ditentukan secara empiris berdasarkan *accuracy* dan *loss* yang dihasilkan selama pelatihan. Hasil dari pelatihan CNN dapat dilihat pada bab berikutnya. Tabel 1 menunjukkan arsitektur yang digunakan dalam penelitian ini.

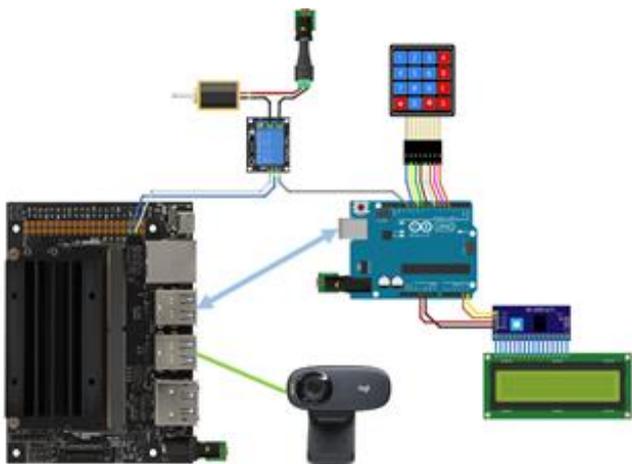
**E. Desain Brankas**

Gambar 3 menunjukkan diagram blok dari sistem yang akan dibuat. Kami menggunakan Jetson Nano sebagai prosesor utama untuk mengolah data. Jetson Nano bisa digunakan untuk menjalankan machine learning, terutama dengan memanfaatkan GPU 128-core yang terpasang di dalamnya [11].

Jetson Nano, Arduino UNO, dan *solenoid lock* akan dihubungkan dengan sumber tegangan dari luar dikarenakan sistem yang dirancang tidak memiliki alat peyimpan daya. Sebagai masukan digunakan *membrane keypad*, *relay switch*, dan LCD. Relay switch menggunakan sumber tegangan dari Jetson Nano dikarenakan Arduino UNO hanya memiliki 1 pin 5V, namun pin IN pada relay tetap dihubungkan pada Arduino UNO agar sinyal *high* dan *low* bisa diterima. *Webcam* dihubungkan ke Jetson Nano melalui USB port. Tidak ada komponen tambahan yang digunakan untuk menghubungkan *webcam* ke Jetson Nano.

**Tabel 1.** Arsitektur CNN yang digunakan

Layer	Activation function	Output
Convolution (1)	ReLU	(222, 222, 64)
Max pooling (1)	-	(111, 111, 64)
Convolution (2)	ReLU	(109, 109, 128)
Max pooling (2)	-	(54, 54, 128)
Flatten	-	(373248)
Fully connected layer (1)	ReLU	(128)
Fully connected layer (2)	ReLU	(64)
Fully connected layer (3)	ReLU	(32)
Fully connected layer (4)	ReLU	(16)
Fully connected layer (5)	Softmax	(3)



**Gambar 3.** Rangkaian listrik yang digunakan

Kotak brankas yang akan digunakan sebagai *prototype* terbuat dari *plywood* dengan ukuran 30 cm x 30 cm x 60 cm untuk panjang, lebar, dan tinggi kotak diukur dari luar. Plywood yang digunakan memiliki ketebalan 1 cm. Ruang setinggi 10 cm digunakan untuk menutup *adapter* dari Jetson Nano, Arduino UNO, dan *solenoid lock*. Sisa ruang yang

tersedia digunakan untuk menyimpan barang-barang milik pemilik brankas. Hasil akhir dari *prototype* brankas ini dapat dilihat pada bab berikutnya.

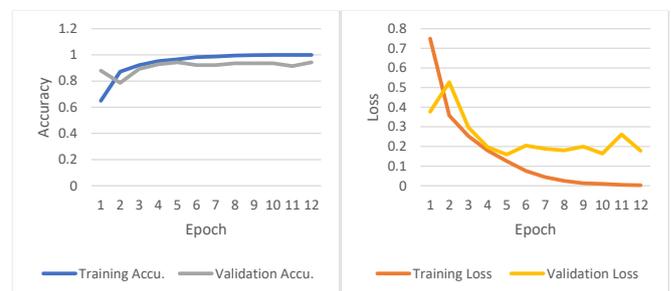


**Gambar 4.** Desain Prototipe Brankas

**IV. HASIL DAN PEMBAHASAN**

**A. Pengujian CNN**

Gambar 5 menunjukkan hasil pelatihan model CNN selama 12 *epoch*. Tingkat akurasi CNN mencapai 100% sejak *epoch* ke-9. Sistem ini hanya dilatih sebanyak 12 *epoch* karena hasil *validation loss* cenderung meningkat di beberapa *epoch* berikutnya.



**Gambar 5.** Hasil Pelatihan CNN

Hasil dari pengenalan wajah yang dilakukan CNN terhadap beberapa sampel gambar ditunjukkan pada Tabel 5. Setiap sampel diambil berasal dari gambar yang digunakan sebagai validasi dan bukan dari *dataset* yang digunakan selama *training*. *Webcam* masih belum digunakan pada proses ini dan setiap gambar dimasukkan ke CNN secara manual. Untuk setiap kategori, ada 3 sampel yang diuji. Dari kesembilan sampel yang diuji ke CNN, semuanya memiliki tingkat kemiripan di atas 90%, dengan 8 di antaranya berada di atas 99%. Berdasarkan hasil ini, pelatihan sistem CNN dianggap sudah cukup.

**B. Pengujian Perangkat Keras**

Saat menguji *membrane keypad* dengan Jetson Nano, output yang didapatkan tidak sesuai dengan yang diharapkan. Ini menyebabkan *membrane keypad* tidak bisa digunakan dengan baik. Oleh karena itu, penulis menguji GPIO Jetson Nano dengan sebuah *tactile push button* untuk memeriksa hasil pembacaan di Jetson Nano. Rangkaian yang digunakan ditunjukkan pada Gambar 6.

**Tabel 2.** Hasil Pelatihan Wajah dengan CNN

Input	Target	Output
	Erwin	<b>Erwin: 99.94439482688904 %</b> Unknown: 0.054253096459433436 % Willy: 0.0013522620974981692 %
	Erwin	<b>Erwin: 99.82607960700989 %</b> Unknown: 0.17228185897693038 % Willy: 0.0016376641724491492 %
	Erwin	<b>Erwin: 99.98804330825806 %</b> Unknown: 0.010740784637164325 % Willy: 0.0012114028322685044 %
	Unknown	Erwin: 0.0006433198450395139 % <b>Unknown: 99.99874830245972 %</b> Willy: 0.0006107261469878722 %
	Unknown	Erwin: 0.05995528772473335 % <b>Unknown: 99.90395903587341 %</b> Willy: 0.036085222382098436 %
	Unknown	Erwin: 0.4503358155488968 % <b>Unknown: 99.54525828361511 %</b> Willy: 0.004419003744260408 %
	Willy	Erwin: 0.00032775722047517775 % Unknown: 0.0006428396318369778 % <b>Willy: 99.99903440475464 %</b>
	Willy	Erwin: 0.10223528370261192 % Unknown: 0.6973585113883018 % <b>Willy: 99.20040965080261 %</b>
	Willy	Erwin: 7.561878114938736 % Unknown: 1.57986618578434 % <b>Willy: 90.85825085639954 %</b>



**Gambar 6.** Skema pengujian tactile push button di Jetson Nano

Tabel 3 menunjukkan hasil pembacaan dari GPIO dengan kombinasi antara nilai resistor dan sumber tegangan yang digunakan. Dengan konfigurasi yang menggunakan resistor pull-down, pembacaan GPIO hanya bekerja dengan baik ketika pin 3.3 V digunakan. Menggunakan membrane keypad memerlukan pin GPIO sebagai sumber tegangan karena setiap baris harus dialiri tegangan secara bergantian untuk mendeteksi tombol yang ditekan. Berdasarkan hasil yang didapat, dapat disimpulkan bahwa membrane keypad tidak bisa digunakan di Jetson Nano. Membrane keypad dipasang di Arduino UNO karena terdapat resistor pull-down internal dan

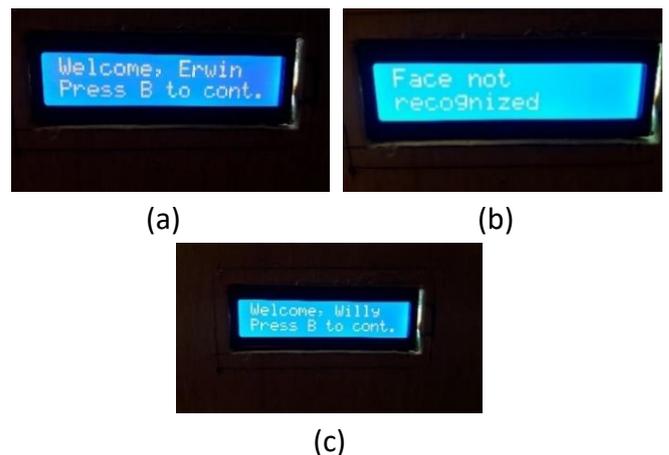
daya yang cukup untuk melakukan pembacaan pada membrane keypad. Solenoid lock bekerja dengan baik ketika diberi tegangan input dari pin GPIO. Solenoid lock memiliki delay yang cukup singkat dan dapat diabaikan. LCD tidak bermasalah ketika diuji menggunakan Arduino UNO. Selama percobaan, LCD dapat digunakan dengan baik. Selama melakukan percobaan terhadap webcam, tidak ada masalah yang ditemukan. Webcam dapat berfungsi dengan baik.

**Tabel 3.** Hasil Percobaan Tactile Push Button di Jetson Nano

Sumber tegangan	Pull-down resistor	Hasil pembacaan
Pin 3.3 V	Tanpa pull-down	Latch ketika input high
Pin 3.3 V	220 Ω	Normal
Pin 3.3 V	1 kΩ	Normal
Pin 3.3 V	4.7 kΩ	Latch ketika input high
Pin GPIO HIGH (3.3 V)	Tanpa pull-down	Latch ketika input high
Pin GPIO HIGH (3.3 V)	220 Ω	Floating ketika input high
Pin GPIO HIGH (3.3 V)	1 kΩ	Floating ketika input high
Pin GPIO HIGH (3.3 V)	4.7 kΩ	Latch ketika input high

**C. Kinerja Sistem Secara Keseluruhan**

Dari segi perangkat keras, tidak ada masalah yang terjadi selama pengguna mengakses brankas. Semua komponen dapat berfungsi dengan baik. Ketika Jetson Nano sudah selesai melakukan pengenalan wajah, nama dari salah satu kelas akan dikirim ke Arduino UNO. Jika pemilik brankas terdeteksi, maka LCD akan menampilkan nama pemilik seperti yang ditunjukkan di Gambar 7 (a) dan (c). Namun jika wajah tidak dikenali, maka LCD akan menampilkan hasil seperti yang ditunjukkan pada Gambar 7 (b). Pesan ini akan hilang secara otomatis dan brankas di-reset ke semula.



**Gambar 7.** Tampilan LCD ketika pengenalan wajah selesai

Jika wajah pemilik dikenali, maka pemilik diharuskan untuk memasukkan PIN untuk membuka brankas. Jika PIN yang dimasukkan tepat, maka LCD akan menampilkan hasil seperti pada Gambar 8. Pemilik diberi waktu 10 detik untuk membuka pintu brankas sebelum solenoid lock terkunci kembali. Sebaliknya jika PIN tidak tepat, maka akses ditolak dan proses mengakses brankas harus dimulai dari awal.



**Gambar 8.** Tampilan LCD ketika PIN (a) benar & (b) salah

Jika wajah dikenali dan PIN dari pemilik tepat, maka akses akan diberikan. Gambar 9 (b) menunjukkan kondisi ketika akses diberikan, yaitu *solenoid lock* berada dalam kondisi membuka. Namun jika akses tidak diberikan, *solenoid lock* akan selalu berada di kondisi mengunci seperti pada Gambar 9 (a).



**Gambar 9.** *Solenoid lock* dalam kondisi (a) mengunci & (b) membuka

Tabel 4 menunjukkan hasil pengenalan wajah ketika brankas diakses oleh seseorang. Setiap gambar di Tabel IV diambil melalui *webcam* yang sudah terpasang di brankas. Brankas dirancang untuk membuka pintu ketika *output* pada kategori “erwin” atau “willy” berada di atas 90%. Brankas tidak akan terbuka jika kategori “*unknown*” mendapatkan hasil di atas 90%. Selain itu, jika *output* dari ketiga kategori kurang dari 90%, maka hasilnya juga akan dianggap sebagai “*unknown*”. Angka yang diberi cetak tebal pada tabel ini menunjukkan nilai yang sama atau melebihi 90%.

Berdasarkan hasil dari Tabel V, hanya kategori “willy” saja yang memiliki akurasi di atas 90%. Berdasarkan hasil pelatihan di Gambar 4 dan hasil percobaan di Tabel II, akurasi pada saat *training* dan *validation* sudah melebihi 90%. Ini menunjukkan bahwa CNN mungkin mengalami *overfitting*. Oleh karena itu, sistem pada CNN masih perlu diperbaiki dan dilatih ulang.

Kemungkinan penyebab lainnya adalah terletak pada *webcam*. Gambar yang ditangkap *webcam* tidak fokus pada bagian wajah saja, melainkan gambar latar belakang juga ikut ditangkap seperti yang terlihat pada Tabel IV. Semua *dataset* yang digunakan hanya fokus pada bagian wajah. Algoritma *face detection* dibutuhkan untuk mengatasi masalah ini. Tujuannya adalah untuk mendeteksi bagian yang hanya memiliki wajah dan meneruskannya ke CNN.

Proses komunikasi antara Jetson Nano dan Arduino UNO melalui USB dapat berjalan dengan baik, namun terdapat *delay* sekitar 10 detik selama komunikasi berlangsung. Ini menyebabkan Arduino UNO tidak boleh mengirim dua pesan yang berbeda dalam waktu yang berdekatan. Akibat yang mungkin akan terjadi jika pesan kedua dikirim terlalu cepat adalah Jetson Nano gagal mendapatkan pesan dan terjebak di *infinite loop* karena Jetson terus menunggu pesan dari Arduino UNO. Oleh karena itu, pengguna tidak boleh membuat Arduino UNO mengirim 2 pesan yang berbeda dalam waktu singkat seperti langsung mencoba membuka ulang brankas ketika hasil pengenalan wajah baru diterima. Salah satu cara untuk mengatasi masalah *delay* ini adalah dengan

menggunakan komputer yang mendukung CNN dan mampu menggunakan GPIO dengan baik sehingga tidak diperlukan komunikasi antara 2 perangkat yang berbeda.

**Tabel 4.** Hasil Pengenalan Wajah oleh Sistem

Input	Target	Output
	Erwin	Erwin: 0.00902733881957829 % <i>Unknown</i> : 0.241751573048532 % <b>Willy: 99.74921941757202 %</b>
	Erwin	Erwin: 14.983123540878296 % <i>Unknown</i> : 0.09496074635535479 % Willy: 84.92191433906555 %
	Erwin	Erwin: 0.0003469993089311174 % <i>Unknown</i> : 0.005591857916442677 % <b>Willy: 99.99406337738037 %</b>
	Erwin	Erwin: 0.08613921236246824 % <i>Unknown</i> : 0.3606669371947646 % <b>Willy: 99.55319166183472 %</b>
	Erwin	Erwin: 84.75480675697327 % <i>Unknown</i> : 0.2148329745978117 % Willy: 15.030355751514435 %
	<i>Unknown</i>	Erwin: 0.005587459600649035 % <i>Unknown</i> : 1.427378784865141 % <b>Willy: 98.56703877449036 %</b>
	<i>Unknown</i>	Erwin: 0.19154474139213562 % <b><i>Unknown</i>: 97.486013174057 %</b> Willy: 2.322445437312126 %
	<i>Unknown</i>	Erwin: 0.042931747157126665 % <b><i>Unknown</i>: 99.38710927963257 %</b> Willy: 0.5699540488421917 %
	<i>Unknown</i>	Erwin: 53.7652850151062 % <i>Unknown</i> : 3.974035382270813 % Willy: 42.26068556308746 %
	<i>Unknown</i>	Erwin: 0.07679680711589754 % <b><i>Unknown</i>: 96.2475597858429 %</b> Willy: 3.6756522953510284 %
	<i>Unknown</i>	Erwin: 0.754660414531827 % <i>Unknown</i> : 7.5556255877017975 % <b>Willy: 91.6897177696228 %</b>
	<i>Unknown</i>	Erwin: 1.387524325400591 % <b><i>Unknown</i>: 96.329265832901 %</b> Willy: 2.2832119837403297 %
	Willy	Erwin: 5.029168193004807e-07 % <i>Unknown</i> : 9.189001559661847e-07 % <b>Willy: 100.0 %</b>
	Willy	Erwin: 9.991341975990053e-07 % <i>Unknown</i> : 3.213412114178027e-06 % <b>Willy: 100.0 %</b>
	Willy	Erwin: 0.000482893528896966 % <i>Unknown</i> : 0.0010179641321883537 % <b>Willy: 99.99849796295166 %</b>
	Willy	Erwin: 0.000482893528896966 % <i>Unknown</i> : 0.0010179641321883537 % <b>Willy: 99.99849796295166 %</b>
	willy	Erwin: 4.1648520188175553e-07 % <i>Unknown</i> : 4.445479717674061e-06 % <b>Willy: 100.0 %</b>

## V. KESIMPULAN

Perangkat keras yang digunakan pada percobaan ini berjalan dengan baik. Tidak ada masalah yang terjadi selama percobaan sehingga dapat disimpulkan bahwa perangkat keras dapat diimplementasikan pada *prototype* brankas.

Model CNN masih belum bisa diimplementasikan dengan baik ke dalam sistem keamanan brankas. Berdasarkan hasil percobaan, hanya kategori “willy” saja yang memiliki akurasi di atas 90 %. Berdasarkan hasil pengujian brankas, kategori “willy” memiliki akurasi 100 %, kategori “erwin” memiliki akurasi 0 %, dan kategori “*unknown*” memiliki akurasi 71.43 %. Kemungkinan penyebab masalah ini adalah sistem CNN mengalami *overfitting* dan gambar yang ditangkap webcam tidak fokus pada bagian wajah saja.

Komunikasi antara Jetson Nano dan Arduino UNO memiliki *delay* sekitar 10 detik sehingga pengguna tidak boleh membuat Arduino UNO mengirim 2 pesan yang berbeda dalam waktu singkat seperti langsung mencoba membuka ulang brankas ketika hasil pengenalan wajah baru diterima.

## VI. DAFTAR PUSTAKA

- [1] M. üge Çarıkçı and F. Özen, “A Face Recognition System Based on Eigenfaces Method,” *Procedia Technology*, vol. 1, pp. 118–123, Jan. 2012, doi: 10.1016/J.PROTCY.2012.02.023.
- [2] B. Md Abdur Rahim, M. Najmul Hossain, T. Wahid, M. Shafiul Azam, and M. Abdur Rahim, “Face Recognition using Local Binary Patterns (LBP),” *Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc*, vol. 13, 2013.
- [3] M. Coskun, A. Ucar, O. Yildirim, and Y. Demir, “Face recognition based on convolutional neural network,” *Proceedings of the International Conference on Modern Electrical and Energy Systems, MEES 2017*, vol. 2018-January, pp. 376–379, Jun. 2017, doi: 10.1109/MEES.2017.8248937.
- [4] N. Delbiaggio, “A comparison of facial recognition’s algorithms,” 2017.
- [5] C. C. Aggarwal, “Neural Networks and Deep Learning,” *Neural Networks and Deep Learning*, 2018, doi: 10.1007/978-3-319-94463-0.
- [6] C. C. Aggarwal, *Neural Networks and Deep Learning*. New York: Springer, 2018. doi: 10.1007/978-3-319-94463-0.
- [7] R. Venkatesan and B. Li, *Convolutional Neural Networks in Visual Computing : A Concise Guide*. Boca Raton, Florida: CRC Press, 2018.
- [8] L. Bottou, “Stochastic Gradient Learning in Neural Networks,” *Proceedings of Neuro-Nîmes 91*, vol. 91, no. 8, p. 12, 1991.
- [9] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4217–4228, 2021, doi: 10.1109/TPAMI.2020.2970919.
- [10] J. Cheng, Y. Li, J. Wang, L. Yu, and S. Wang, “Exploiting effective facial patches for robust gender recognition,” *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 333–345, 2019, doi: 10.26599/TST.2018.9010090.
- [11] “IoT Projects with NVIDIA Jetson Nano: AI-Enabled Internet of Things Projects for Beginners Book - EVERYONE - Skillsoft.” <https://www.skillsoft.com/book/iot-projects-with-nvidia-jetson-nano-ai-enabled-internet-of-things-projects-for-beginners-a9706f67-be21-43ab-8c88-afedaf1a7a0> (accessed Jul. 28, 2023).