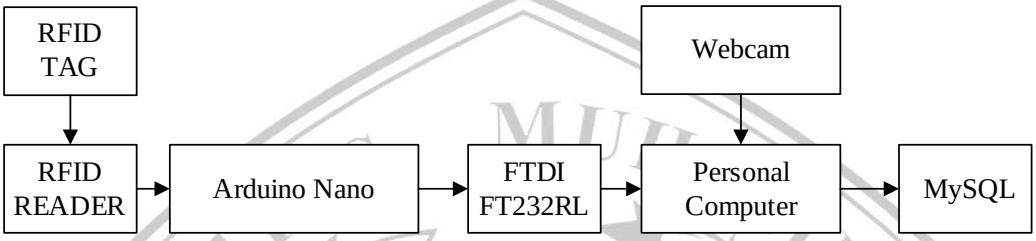
BAB III
METODE PENELITIAN

Dalam rangka menjawab permasalahan yang telah dirumuskan di atas, maka diusulkan metode yang secara umum digambarkan dengan blok diagram pada Gambar 3.1.



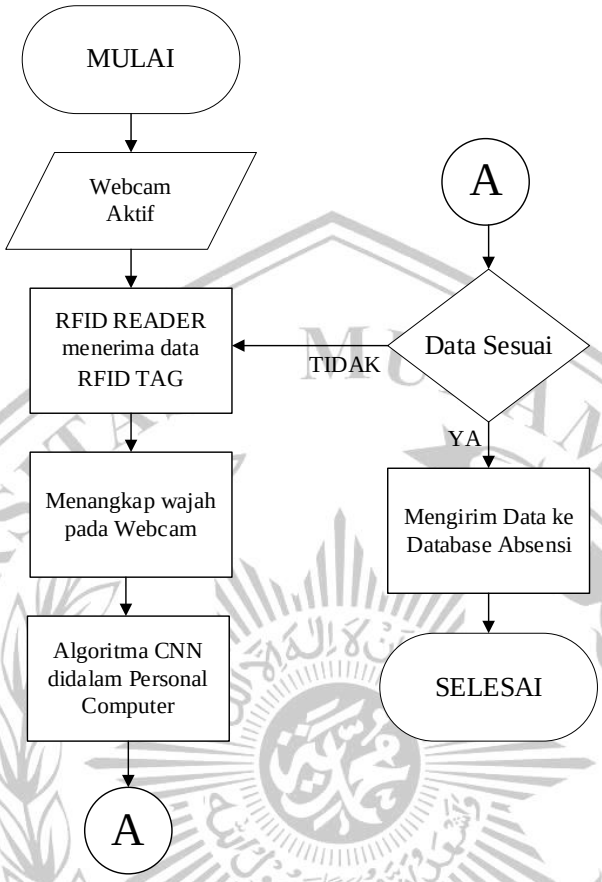
Gambar 3. 1 Diagram Blok Sistem

Dari Gambar 3.1, dapat diuraikan masing masing bagian dari sistem diatas seperti berikut.

a. RFID TAG merupakan sebuah alat yang melekat pada objek yang akan diidentifikasi oleh RFID READER.
b. RFID READER merupakan alat yang mampu mengidentifikasi RFID tag dan mampu berkomunikasi menggunakan protokol SPI.
c. Arduino Nano merupakan *mikrokontroler* yang berfungsi menerima data dari SPI dan mengubah menjadi Data Serial UART.
d. FTDI FT232RL merupakan berfungsi untuk mengkonversi komunikasi UART ke USB agar dapat dibaca oleh personal computer.
e. *Personal Computer* merupakan media yang digunakan memproses data dari *Microkontroller* dan *Webcam*, kemudian data diolah melalui *machine learning* dengan algoritma CNN untuk mengidentifikasi kecocokan data dengan pengguna.
f. Webcam merupakan perangkat kamera digital untuk dihubungkan ke komputer atau laptop.
g. MySQL merupakan *Database* yang menyimpan data presensi pengguna.

Metode *supervised learning* pada *machine learning* yang digunakan akan mempermudah dalam mencocokkan pola pengenalan wajah dan kondisi pengguna

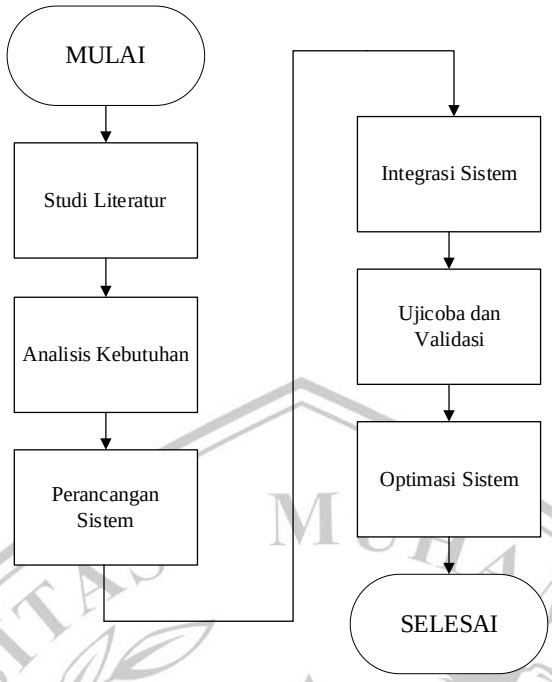
secara real menggunakan algoritma CNN. Untuk alur kerja dari metode ini secara sederhana dapat digambarkan pada Gambar 3.2.



Gambar 3. 2 *Flowchart* penggunaan Algoritma *Convolution Neural Network*

Dari *flowchart* yang terdapat pada Gambar 3.2, alur kerja dari sistem ini ketika *webcam* aktif, RFID TAG akan membaca data melalui RFID READER kemudian data dikirim ke *Mikrokontroller* Arduino Nano melalui komunikasi SPI, melalui modul FTDI data diterima oleh PC untuk menangkap gambar pada wajah dan diolah menggunakan algoritma *Convolutional Neural Network*. Kemudian data RFID dan gambar wajah akan dicocokkan, apabila data sesuai maka data akan dikirim ke database presensi, namun ketika *database* tidak cocok akan mengulangi kembali proses penerimaan data ke Mikrokontroller Arduino Nano.

Kemudian, secara garis besar penelitian ini dilakukan dengan tahapan sebagaimana digambarkan dengan *flowchart* pada Gambar 3.3 berikut.



Gambar 3. 3 *Flowchart* tahapan penelitian

3.1 Analisis Kebutuhan

Dalam mengembangkan sistem keamanan akses terkini, analisis kebutuhan dilakukan untuk mengidentifikasi persyaratan utama dalam menggabungkan teknologi RFID dan pengenalan wajah. Penggunaan teknologi RFID bertujuan untuk memastikan identifikasi cepat dan akurat melalui kartu atau tag RFID, sementara pengenalan wajah diintegrasikan untuk meningkatkan keamanan dengan verifikasi identitas biometrik. Kebutuhan utama mencakup pengembangan antarmuka yang intuitif untuk registrasi pengguna, integrasi yang lancar antara sistem RFID dan pengenalan wajah, serta keamanan tingkat tinggi untuk melindungi data biometrik yang disimpan. Selain itu, analisis mencakup pemahaman mendalam terhadap skenario penggunaan, kebutuhan skalabilitas sistem untuk mendukung jumlah pengguna yang meningkat, dan persyaratan kepatuhan privasi dan keamanan data yang berlaku

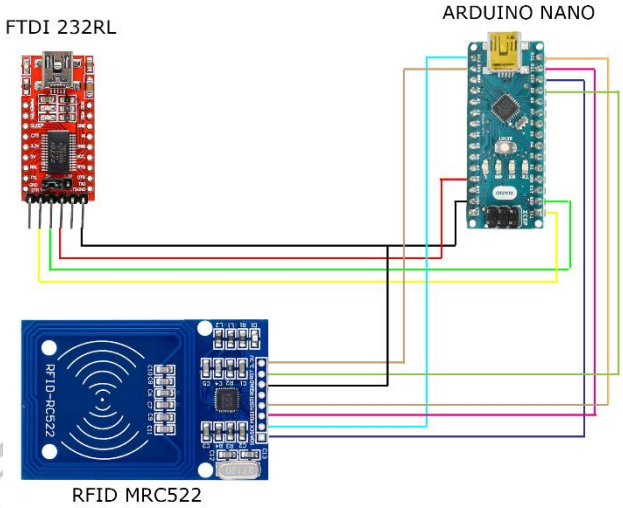
.

3.2 Perancangan Sistem

Perancangan sistem yang melibatkan RFID dan pengenalan pola wajah mencakup perancangan perangkat keras (hardware) dan perangkat lunak (software) agar saling dapat terintegrasi. Berikut adalah rinciannya:

3.2.1 Perancangan Hardware

Pada modul pembaca RFID dibuat melalui skema seperti pada Gambar 3.4.



Gambar 3. 4 Skema *Hardware*

Pada Gambar 3.7. MRC 522 sebagai RFID Reader mampu bekerja pada frekuensi radio 13.56 MHz. Jarak baca pada pembaca RFID MRC 522 pada jarak sekitar 1-2 cm. Data pada RFID Reader dikirim ke Arduino Nano untuk dikirim ke modul FTDI melalui komunikasi UART lalu diterima melalui *Serial Data* pada PC

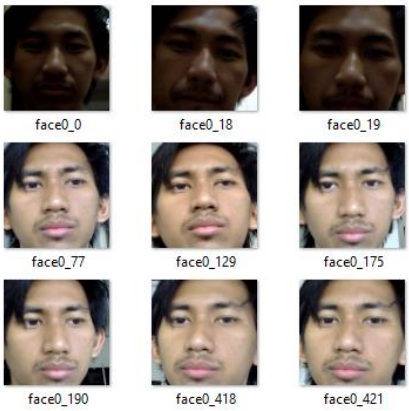
3.2.2 Perancangan Software

Pada Firmware Pembaca RFID, digunakan Arduino IDE sebagai code editor untuk memfasilitasi pengembangan perangkat lunak pada tingkat firmware. Arduino IDE menyediakan lingkungan pengembangan yang user-friendly dan populer dalam proyek-proyek berbasis mikrokontroler seperti Arduino. Sedangkan pada Perancangan algoritma pengenalan wajah melibatkan serangkaian tahapan yang terstruktur untuk memastikan kinerja yang optimal. Berikut adalah tahapan umum dalam perancangan algoritma pengenalan wajah

a. Pengumpulan Data

Tahap awal dalam pengembangan model pengenalan wajah dengan menggunakan Convolutional Neural Network (CNN) melibatkan pengumpulan dataset wajah yang mencakup berbagai karakteristik yang mungkin terjadi dalam situasi nyata. Dataset ini harus terdiri dari gambar-gambar wajah yang representatif

dan bervariasi, mencakup variasi dalam pose, ekspresi wajah, kondisi pencahayaan, dan latar belakang seperti digambarkan pada Gambar 3.5.



Gambar 3. 5 Dataset wajah

Penting untuk memastikan bahwa dataset mencerminkan keragaman yang dihadapi oleh sistem pengenalan wajah dalam kondisi penggunaan sehari-hari. Misalnya, dataset harus mencakup gambar wajah dengan pose yang berbeda seperti miring, menengadah, atau menunduk. Ekspresi wajah juga harus bervariasi, dari senyum hingga wajah serius atau sedih. Variasi pencahayaan mencakup kondisi pencahayaan yang baik, rendah, atau cahaya datar, sedangkan variasi latar belakang dapat mencakup berbagai konteks visual.

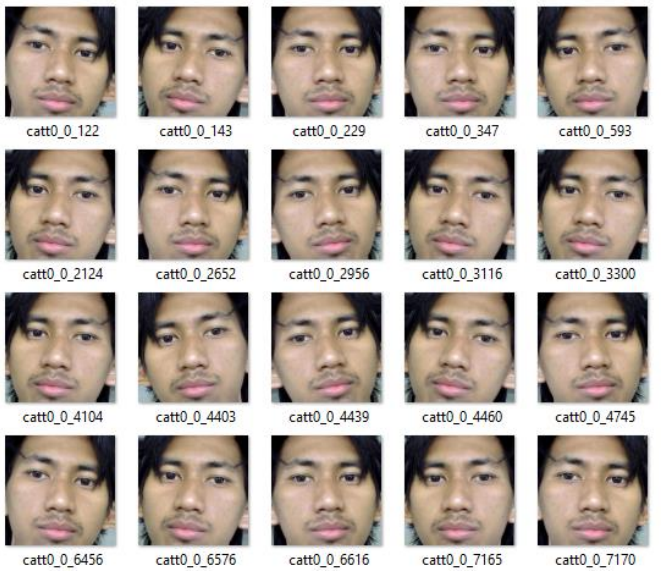
b. Pre-Processing

Pada tahapan ini, raw data wajah yang mengandung beberapa kekurangan antara lain *missing value* atau data yang tidak memiliki atribut, data memiliki nilai yang salah, dan data berisi catatan yang tidak konsisten, maka perlu diilakukan proses data cleansing agar didapatkan output yang sesuai. Proses *data cleansing* dilakukan dengan merubah data kotor menjadi data yang berkualitas agar dapat menghasilkan informasi yang akurat [20]. Proses data cleansing ini digambarkan pada Gambar 3.6.

Gambar 3. 6 Cleansing Dataset

c. Data Processing

Pada tahapan ini, gambar pada webcam akan di seleksi untuk mengambil daerah wajah menggunakan Metode *Haar-like feature* untuk memproses gambar dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa pixel [21]. Per kotak itu pun kemudian di-proses dan didapatkan perbedaan nilai *(threshold*) yang menandakan daerah gelap dan terang. Dataset yang telah melewati *cleansing* akan dilakukan juga proses augmentasi data seperti pada Gambar 3.7 agar dataset lebih beragam. Nilai – nilai inilah yang nantinya data tersebut akan masuk ke pemrosesan data.



Gambar 3. 7 Data Processing

d. Model Training dan Validasi

Pada tahap Model Training, data yang telah dikumpulkan kemudian di training menggunakan arsitektur CNN yang telah dibuat. Untuk mencapai tujuan

pembelajaran mesin, model diperoleh setelah pembelajaran harus bekerja dengan baik pada sampel baru, tidak hanya pada sampel yang digunakan untuk pelatihan. Kemampuan dari menerapkan model ke sampel baru disebut generalisasi atau ketahanan. Beberapa faktor yang menentukan model *machine learning* adalah *error*, *training error*, *underfitting*, dan *overfitting* [22].

3.3 Integrasi Sistem

Setelah model machine learning yang telah dibuat, kemudian sistem di kombinasikan dengan Database MySQL dan hasil baca Modul MRC 522 yang dikirim ke Arduino Nano kemudian diteruskan ke modul FTDI lalu diterima melalui *Serial Data* pada PC . QT Creator berfungsi membangun aplikasi GUI yang dibuat menggunakan Bahasa C++ untuk memudahkan pengguna dalam membaca hasil sistem yang dibuat.

3.4 Uji Coba dan Validasi

Pada Tahap ini, kemampuan model untuk menyesuaikan fungsi, yang juga disebut kompleksitas model. Ketika kapasitas sesuai dengan kompleksitas tugas dan jumlah data pelatihan yang disediakan, algoritma efek biasanya optimal. Model dengan kapasitas yang tidak mencukupi tidak dapat menyelesaikan tugas yang kompleks dan dapat terjadi *underfitting*. Model berkapasitas tinggi dapat menyelesaikan tugas yang rumit, tetapi *overfitting* dapat terjadi jika kapasitasnya lebih tinggi dari yang dibutuhkan oleh suatu tugas maka perlunya evaluasi.

3.5 Optimasi Sistem

Untuk menentukan model yang paling optimal, kita dapat menggunakan beberapa parameter yang bisa memberikan gambaran objektif tentang kinerja model. Beberapa ukuran yang sering digunakan adalah akurasi, presisi, recall, spesifisitas, dan F1-score [23]. Metode confusion matrix dipilih karena mampu memvisualisasikan dan menganalisis hasil prediksi yang dibuat oleh model. Namun pada model yang digunakan berjenis *Multi-class Classification* sehingga *confusion matrix* diubah menjadi Tabel 3.1

Tabel 3. 1 Confusion matrix 5-class

Confusion Matrix		Nilai Prediksi				
		M1	M2	M3	M4	M5
Nilai Sebenarnya	M1	Cl	C2	C3	C4	C5
	M2	C6	C7	C8	C9	C10
	M3	C11	C12	C13	C14	C15
	M4	C16	C17	C18	C19	C20
	M5	C21	C22	C23	C24	C25

Dimana M1-M5 mewakili nama Mahasiswa yang berjumlah 5 orang, lalu C1-C25 mewakili nomor pada cell. Untuk mengetahui nilai TP,TN,FP,dan FN pada confusion matrix 5-class, dapat dicari menggunakan cara berikut,

TP didapatkan ketika nilai sebenarnya dan nilai prediksi sama. Jadi untuk kelas M1, nilai C1 adalah nilai TP.

FN didapatkan dari jumlah nilai baris yang bersesuaian kecuali nilai TP. Jadi untuk kelas M1, penjunlahan dari C2,C3,C4,C5 adalah nilai FN.

FP didapatkan dari jumlah nilai kolom terkait kecuali nilai TP. Jadi untuk kelas M1, penjunlahan dari C6,C11,C16,C21 adalah nilai FP.

TN didapatkan dari jumlah nilai semua kolom dan baris kecuali nilai kelas yang kita hitung nilainya. Jadi untuk kelas M1, penjunlahan dari C7,C8,C9,C10, C12, C13, C14, C15, C17, C18, C19, C20, C22, C23, C24, dan C25 adalah nilai TN.

Parameter akurasi digunakan untuk mengukur seberapa baik model membuat prediksi yang benar dari total prediksi yang dilakukan. Dalam konteks klasifikasi, akurasi memberikan gambaran mengenai seberapa sering model memprediksi kelas yang benar, baik itu kelas positif maupun negatif. Nilai akurasi akan menjawab pertanyaan sejauh mana model yang digunakan mampu memprediksi sentimen wajah secara keseluruhan dengan benar. Untuk menghitung nilai akurasi, kita dapat menggunakan persamaan 3.1.

$$\text{Akurasi} = \frac{\text{TP+TN}}{T\text{P+TN+FP+FN}} \tag{3.1}$$

Parameter presisi digunakan untuk mengukur seberapa baik model membuat prediksi yang benar untuk kelas positif dari total prediksi positif yang dilakukan.

Dalam konteks klasifikasi, presisi memberikan gambaran mengenai seberapa sering model memprediksi kelas positif dengan benar, di antara semua prediksi positif yang dibuat oleh model. Nilai presisi akan menjawab pertanyaan sejauh mana model yang digunakan mampu memprediksi sentimen wajah yang sebenarnya positif dengan benar. Untuk menghitung nilai presisi, kita dapat menggunakan persamaan 3.2.

$$\text{Presisi} = \frac{\text{TP}}{TP+\text{FP}} \tag{3.2}$$

Parameter sensitivitas (*recall*) digunakan untuk menggambarkan seberapa baik suatu model dalam mengidentifikasi kelas positif dengan benar. Recall akan menunjukkan kemampuan model kita untuk mengenali review positif yang sebenarnya. Untuk menghitung nilai sensitivitas (*recall*), kita dapat menggunakan persamaan 3.3.

$$\text{Sensitivitas} = \frac{\text{TP}}{TP+\text{FN}} \tag{3.3}$$

Parameter spesifisitas (*specificity*) digunakan untuk menunjukkan seberapa efektif suatu model dalam mengklasifikasikan kelas negatif secara akurat. *Specificity* menggambarkan kemampuan model kita untuk membedakan review negatif yang asli. Untuk menghitung nilai spesifisitas (specificity), kita dapat menggunakan persamaan 3.4.

$$\text{Spesifisitas} = \frac{\text{T}N}{TN+\text{F}P} \tag{3.4}$$

Parameter F1 score digunakan untuk mencerminkan keseimbangan antara presisi (*precision*) dan sensitivitas (*recall*). F1 Score memberikan gambaran mengenai seberapa baik model kita dalam mengklasifikasikan baik review positif maupun negatif secara akurat.. Untuk menghitung nilai F1 Score, kita dapat menggunakan persamaan 3.5.

$$\text{F1 } Score = 2 \text{ x } \frac{\text{R}ecall\ x\ Presisi}{Recall+Presisi} \tag{3.5}$$