

# Sistem Pendeteksi Suhu Tubuh Menggunakan Face Recognition dengan Web-based Data Management System untuk Memonitor Persebaran Covid-19

Felix Filipi, Febrian Nugroho, Yutaro Tanaka  
Program Studi Computer Science

Universitas Bina Nusantara Malang, Tirtomoyo, Kec. Pakis, Jawa Timur 65154  
felixfilipi4@gmail.com , fbriann@gmail.com , [yutarotanaka3@gmail.com](mailto:yutarotanaka3@gmail.com)

**Abstrak**—Pada masa Pandemi Covid-19. Alat scanner temperatur sangatlah sering kita temui di beberapa tempat seperti hotel, maupun tempat-tempat umum lainnya. Namun, alat yang beredar seringkali kurang efisien dan masih bersifat manual, serta memerlukan petugas untuk penggunaannya. Melalui lomba ini, kami mengembangkan produk *Wall Mounted Body Temperature Scanner* dengan *Web-based Data Management System* yang bertujuan untuk mempermudah pengunjung dalam memeriksa suhu tubuh mereka tanpa perlu campur tangan dari pihak ketiga. Produk *Wall Mounted Body Temperature Scanner* ini juga ditujukan untuk menjadi instrumen yang dapat memonitor persebaran Covid-19 melalui sistem *data scanning* yang dapat merekam data wajah serta suhu dari pengunjung secara real time, serta mencocokkan data wajah yang terekam dengan data wajah yang ada di dalam *database* untuk menampilkan identitas pengunjung tersebut. Pengembangan aplikasi ini menggunakan beberapa *library* Python berbasis *open source* antara lain OpenCV, Flask, paho MQTT, mySql-Python connector, dan menggunakan arduino sebagai *microcontroller* serta raspberry pi sebagai *mini-pc*. Demi menjaga data-data pribadi dari pengunjung, data yang ada hanya dapat diakses oleh pihak admin. Admin sendiri memiliki 3 pilihan menu antara lain, *Camera Stream* dimana admin dapat melihat apa

yang sedang kamera rekam, beserta *face detection* yang bekerja secara *real-time*. Admin juga dapat melihat *database* serta menambahkannya, serta *data scan* yang akan menampilkan data hasil scan berupa data wajah, suhu, serta informasi yang sesuai dengan hasil dari pencocokan wajah dalam *database*.

**Keyword**—*Temperature Scanner; Web-based Data Management System; face-recognition; Internet of Things; Covid-19*

## I. PENDAHULUAN

Total kasus terjangkit Covid-19 di Indonesia hingga tanggal 6 September 2020 telah mencapai angka 194.109 kasus dengan pasien meninggal mencapai 8025 orang. Berdasarkan data tersebut, Indonesia telah menempati posisi ke 23 untuk jumlah kasus penyakit Covid-19 terbanyak di dunia. Untuk mengatasi hal ini, berbagai upaya telah dilakukan oleh pemerintah maupun masyarakat salah satunya adalah pembatasan sosial berskala besar yang terjadi di seluruh Indonesia.

Dalam pelaksanaannya, Pembatasan Sosial Berskala Besar (PSBB) ini menjadi puncak dari usaha pencegahan penularan Covid-19 yang mana telah terlewati, dan memasuki pasca PSBB yang kita kenal dengan era *new-normal*. Hotel dan tempat umum lainnya telah dibuka, namun masih dengan protokol kesehatan serta tata cara yang berbeda dari

sebelumnya, sebagai contoh adalah warga yang diwajibkan untuk mengenakan masker saat berada diluar rumah. Juga seringkali ditemui adanya petugas yang mengecek suhu tubuh pengunjung untuk memastikan kesehatan mereka sebelum memasuki area.

Proses pemeriksaan suhu pengunjung ini sangatlah penting untuk dilakukan. Karena salah satu ciri dari orang yang terinfeksi Covid-19 adalah mengalami flu disertai dengan demam. Penanganan pengunjung dengan Suhu diatas 38 derajat celcius berbeda-beda tergantung dengan kebijakan dari setiap tempat. Ada beberapa tempat yang akan mempersilahkan pengunjung untuk meninggalkan tempat tersebut, ada juga tempat yang menyarankan pengunjung untuk melakukan rapid test untuk memastikan kondisi dari pengunjung tersebut.

Untuk lebih mengefisienkan dan memudahkan pelanggan dalam melakukan pengujian ini, penulis mengusungkan ide untuk membuat sebuah produk yang dapat melakukan scan tubuh lebih cepat dan mudah yaitu melalui *scan* tangan. *Scan* di tangan ini kami pilih dikarenakan efisiensi yang tinggi, dimana pengguna dapat dengan mudah meletakkan tangan di tempat yang disediakan dibandingkan bagian tubuh yang lain.

Secara umum, produk ini berupa aplikasi *website*. Penulis memilih aplikasi *website* dikarenakan lebih mudah diakses sehingga *client* tidak diharuskan memasang aplikasi tersebut ke perangkat yang dimiliki. Selain itu juga memudahkan admin untuk mengatur atau mengakses aplikasi ini.

Pengembangan produk ini menggunakan *Mini-PC Raspberry Pi* ditambah dengan *Arduino UNO* yang berfungsi sebagai pemroses data subjek yaitu suhu serta foto. Sedangkan untuk *server*, kami menggunakan *server HTTP open-source* yaitu *Apache HTTP Server*. Dan untuk komunikasi kedua komponen ini, kami menggunakan protokol *MQTT (Message Queuing Telemetry Transport)*.

Setelah itu untuk pembuatan *website* kami, secara desain kami menggunakan *framework*

*Bootstrap 4* sebagai kerangka utama, dan *PHP native* versi 7 sebagai kerangka pengembangan *website* dinamis, sedangkan untuk komunikasi *MQTT* dan *face recognition* kami memakai aplikasi *python*.

## II. KAJIAN PUSTAKA

### A. *Arduino*

*Arduino* adalah platform pembuatan prototipe elektronik yang bersifat open source hardware yang berdasarkan pada perangkat keras dan perangkat lunak yang fleksibel dan mudah digunakan. *Microcontroller* diprogram menggunakan bahasa pemrograman *arduino* yang memiliki kemiripan *syntax* dengan bahasa pemrograman C. Karena sifatnya yang *open source*, maka siapa saja dapat mengunduh skema *hardware arduino* dan membangunnya.

### B. *Raspberry Pi*

*Raspberry Pi* (juga dikenal sebagai *RasPi*) adalah sebuah *SBC (Single Board Computer)* seukuran kartu kredit yang dikembangkan oleh Yayasan *Raspberry Pi* di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer dasar disekolah-sekolah. *Raspberry* sendiri memiliki beberapa fitur yang dimiliki oleh komputer seperti *RAM, SoC, PMIC, USB port, Radio Module, Ethernet port, dsb*. *Raspberry Pi* sendiri bisa disebut sebagai komputer papan tunggal dan dapat menjalankan beberapa program seperti program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi

### C. *Arduino UNO*

*Arduino UNO* adalah sebuah board *microcontroller* yang didasarkan pada *ATmega328*. *Arduino UNO* mempunyai 14 pin digital *input/output* (6 di antaranya dapat digunakan sebagai *output PWM*), 6 input analog, sebuah osilator kristal 16 MHz, sebuah koneksi *USB*, sebuah *power jack*, sebuah *ICSP header*, dan sebuah tombol *reset*. *Arduino UNO* memuat semua yang dibutuhkan untuk menunjang *microcontroller*, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel *USB* atau

mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.

#### D. *Liquid Crystal Display (LCD)*

*Liquid Crystal Display (LCD)* merupakan suatu alat yang dapat menampilkan karakter ASCII sehingga kita bisa menampilkan campuran huruf dan angka sekaligus. LCD dalam hal ini digunakan sebagai tampilan dari instruksi instruksi dari *microcontroller*. LCD (*Liquid Crystal Display*) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka maupun grafik.

#### E. *Inter Integrated Circuit (I2C)*

I2C singkatan dari *Inter Integrated Circuit*, adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I2C sendiri terdiri dari saluran *Serial Clock (SCL)* dan *Serial Data (SDA)* yang membawa informasi data antara I2C dengan pengontrolnya.

#### F. *MLX90614*

MLX90614 adalah modul untuk mengukur temperatur objek dan sekitar dengan memakai sensor inframerah (*infrared*) yang dibuat oleh Melexis, suatu perusahaan penyedia perangkat elektronik *semi-konduktor*. Jenis temperatur modul ini adalah bersifat *non-contact*, artinya tidak memerlukan kontak langsung antara pengguna dan modul. Perangkat ini memakai komunikasi I2C untuk menerima/mengirim data, dan memerlukan tegangan 5V supaya bekerja dengan stabil.

#### G. *Internet of Things (IoT)*

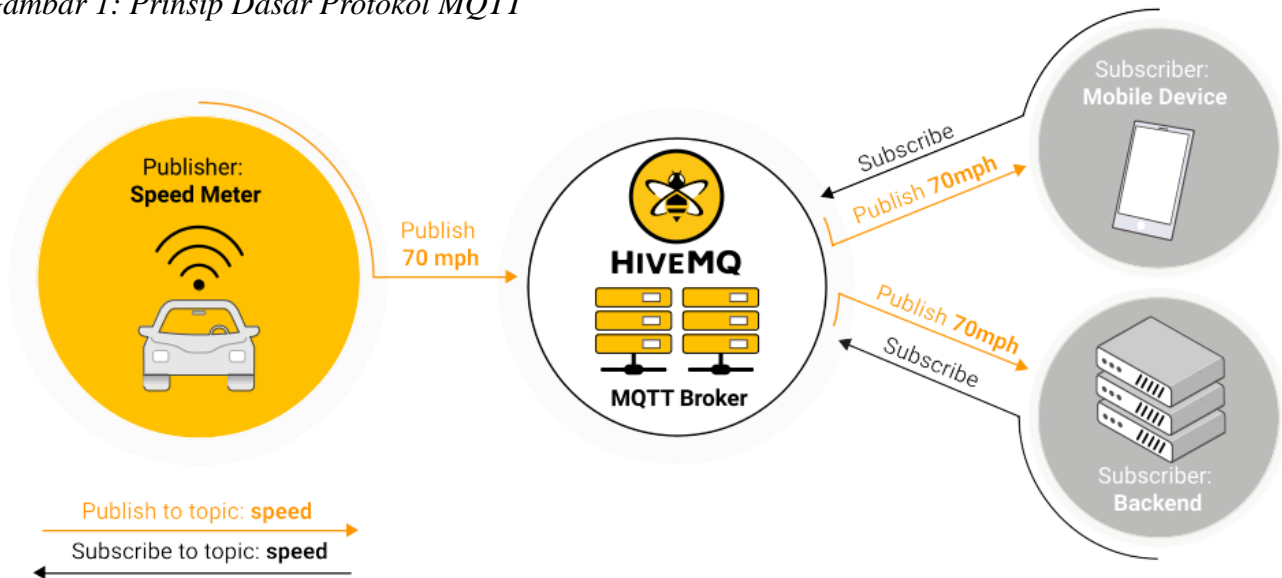
Secara harafiah, *Internet of Things (IoT)* adalah benda/objek (*Things*) yang terkoneksi dan

saling berkomunikasi (*Internet*). Dengan kata lain, *Internet of Things* meliputi komponen atau objek yang saling berkomunikasi dan terkoneksi sehingga menjadi sebuah sistem atau *network*. Objek/komponen itu tidak lain seperti; sensor, *smartwatch*, handphone, komputer, dll. Jangkauan *Internet of Things* ini sangatlah luas, dalam artian hampir semua benda dapat kita buat sedemikian rupa sehingga membentuk sebuah sistem yang saling terhubung dan berkomunikasi. *Internet of Things* ini juga termasuk salah satu subjek dalam bangkitnya industri 4.0 menjadikan IoT sebagai prospek yang menguntukan untuk masa depan.

#### H. MQTT (Message Queuing Telemetry Transport)

MQTT adalah salah satu protokol *networking* yang ringan yang berfungsi sebagai komunikasi 2 arah antar sistem, *microcontroller*, ataupun komputer. Protokol ini sering digunakan pada alat *Internet of Things* dikarenakan efektifitas dan durasi yang terbilang cepat dan stabil, terutama untuk sistem *network* yang jauh dan sulit dijangkau. Prinsip protokol ini menggunakan komunikasi *Publish* dan *Subscribe*. Seorang *Publisher* berfungsi mengirim pesan/data, sedangkan *Subscriber* akan menerima pesan dari *Publisher*. Sebelum itu, *Publisher* dan *Subscriber* harus terhubung dengan *Broker* yang berfungsi sebagai server dan juga mem-filter pesan/data yang masuk. Pemfilteran yang dilakukan *Broker* ini berkaitan dengan *Topics* yang dibuat oleh *Publisher*. Dengan kata lain, setiap pesan yang datang dan dikirim oleh *Publisher* dan *Subscriber* akan difilter oleh *Broker* terhadap *Topic – topic* yang ada. Untuk project ini kami memakai salah satu penyedia broker yang bersifat *open source*, yaitu *Eclipse Mosquitto*. Sedangkan untuk mengatur komunikasi *Publisher* dan *Subscriber* kami menggunakan *Eclipse Paho MQTT*. Kedua modul ini dibuat oleh organisasi *non-profit* dan *open-source* yang sama yaitu *Eclipse Foundation*.

Gambar 1: Prinsip Dasar Protokol MQTT



## I. Apache

Seperti nama lengkapnya, yaitu *Apache HTTP Server*, adalah penyedia server *HTTP* untuk pembuatan dan pembangunan *website*. Software ini dipakai hampir 40% dari semua *website* di seluruh internet. *Apache* memungkinkan penggunaannya untuk menyajikan *website private* maupun publik kepada *client*-nya. Cara kerja software ini adalah sebagai penyedia komunikasi *HTTP* antar server dan *client*. Sebagai contoh kasus; Saat *client* ingin masuk *website* kita, *browser client* akan melakukan *request* pada server *apache* kita. Setelah itu server akan mengirimkan *file* yang diperlukan *client* untuk melihat *website* kita. Komunikasi ini disebut protokol *HTTP* yang disediakan oleh software *Apache* tadi.

## J. Flask

Secara definisi, *Flask* adalah sebuah *framework website*. Yang artinya *Flask* menyediakan semua *tools*, *library*, dan *plugin* yang diperlukan untuk pembuatan aplikasi *website*. Contoh aplikasi *website* beragam contohnya; *blog*, laman *website*, *wikipedia*, sampai *website* komersial. *Flask* termasuk *micro-framework*, yang artinya sedikit atau tidak sama

sekali diperlukannya *dependencies* pada *library* luar. Karena itu kelebihan dari *Flask* adalah keringanan beban dan performa saat dijalankan. Tetapi dengan konsekuensi saat membutuhkan *library* tertentu, harus mencari dan menginstall *library* yang dibutuhkan. Dalam kasus project ini, kami gunakan *Flask* untuk membuat platform untuk bisa melihat *streaming* dari *webcam* melalui *website*.

## K. OpenCV

*OpenCV*, sebagai singkatan dari *Open Source Computer Vision*, menyediakan *library* yang dibutuhkan untuk pemrosesan dan fungsi penglihatan komputer. Diciptakan sebagai proyek *researching* oleh Intel pada tahun 1998, *OpenCV* berfungsi untuk menyediakan alat alat dan modul yang dibutuhkan untuk menyelesaikan masalah – masalah penglihatan komputer, penglihatan (*vision*) komputer disini contohnya adalah pemrosesan video, foto, ataupun *camera stream*. Di dalam *OpenCV* terdapat banyak ragam pemrosesan foto dan video dari yang sederhana, misal; *image-filter*, *gray-filter*, *brightness*, sampai algoritma tingkat tinggi, contohnya; *face-detection*, *pedestrian detection*, dan *feature*

*matching*. *OpenCV* sangat diperlukan untuk proyek kami karena dibutuhkan pemrosesan *video stream*, *face-detection*, dan *face-recognition* dalam pengerjaannya. Untuk fungsi *face-detection* kami memakai algoritma *Haar-Cascades*, sedangkan fungsi *face-recognition* kami gunakan algoritma *training LBPH Face Recognition*.

#### K.1. *Haar-Cascades Face Detection Algorithm*

*Haar-Cascades* adalah salah satu algoritma *Machine Learning* yang berfungsi untuk mendeteksi objek dalam suatu video atau foto. Algoritma ini di proposalkan oleh Paul Viola dan Michael Jones pada tahun 2001 dengan *paper* berjudul “*Rapid Object Detection using a Boosted Cascade of Simple Features*”. Singkatnya, algoritma ini memakai banyak foto positif dan negatif untuk dijadikan *training data*. Ada 4 tahap dalam algoritma ini :

1. *Haar Feature Selection*
2. *Creating Integral Images*
3. *Adaboost Training*
4. *Cascading Classifiers*

#### K.2 *LBPH-Cascades Face Recognition Algorithm*

*LBPH (Local Binary Pattern Histogram)* adalah algoritma pencocokan wajah (*Face Recognition*) yang paling banyak dipakai. *LBPH* ditemukan pada tahun 1994 dan sejak saat itu menjadi algoritma utama untuk pencocokan wajah. Cara kerja algoritma *LBPH* adalah foto/video di *convert* menjadi histogram, setelah itu diubah menjadi data vektor dan dibandingkan dengan foto yang ingin dicocokkan. Untuk langkah langkah detailnya bisa dilihat dibawah ini :

1. *Parameters*
2. *Training Algorithm*
3. *Applying LBPH operation*
4. *Extracting Hisograms*

#### 5. *Perform Face Recognition*

##### L. *Base64 Encoding/Decoding*

Untuk pengiriman data berupa foto pada proyek kami menggunakan *Base64 Encoding/Decoding*. *Base64* berfungsi untuk mengubah, dalam kasus ini adalah foto, menjadi karakter (*ASCII*, *UTF-8*, *UTF-16*). Kami memakai data *Base64* ini dikarenakan kecepatan dan stabilitas yang terbilang stabil untuk dikirim memakai protokol *MQTT*.

##### M. *Use Case*

Singkatnya, *Use Case Diagram* adalah diagram yang mendefinisikan interaksi contoh kasus, aktor, dan sistem aplikasi/data. Fungsi dari *Use Case* ini antara lain dapat menjelaskan perilaku sistem, relasi sistem dengan aktor, lingkungan sistem, dsb. Dalam *Use Case* terdapat beberapa komponen, antara lain:

1. *Actor*: Menggambarkan subjek yang akan berinteraksi dengan sistem
2. *Use Case*: Mendefinisikan fungsi yang akan dilakukan oleh sistem
3. *Relationship*: Menjelaskan hubungan/relasi antar *use case* maupun *actor*
  - 3.1. *Association*: Menghubungkan antar *Use Case* dan/atau *Actor*
  - 3.2. *Generalizaion*: Sebagai elemen yang menjadi spesialisasi elemen lain
  - 3.3. *Dependency*: Elemen yang bergantung beberapa cara kepada elemen lainnya
  - 3.4. *Aggregation*: Relasi berupa elemen yang berisi elemen lainnya.

##### N. *Activity Diagram*

Adalah bagan alur kerja yang lebih kompleks dari *use case* dan berisi aktivitas dan tindakan, pilihan, dan juga perulangan. Jika *Use Case* menggambarkan aktivitas yang dilakukan aktor kepada sistem. *Activity Diagram* menggambarkan akitivitas yang dilakukan

didalam sistem. Dengan kata lain, *Activity Diagram* menggambarkan proses dan aktivitas secara mendetail dari sebuah sistem/aplikasi yang dibangun. *Activity Diagram* juga dipakai untuk memahami sebuah sistem sehingga lebih mudah untuk didefinisikan.

Ada beberapa komponen yang ada dalam *Activity Diagram*:

1. *Start Point*: Menandakan titik awal dimulainya sebuah proses
2. *Activity*: Menunjukan aktivitas yang sedang terjadi di dalam sistem
3. *Action Flow*: Menunjukan aktivitas selanjutnya atau transisi dari aktivitas ke aktivitas yang dituju
4. *Decision*: Menunjukan adanya kondisi atau pilihan dalam sebuah transisi aktivitas
5. *Synchronization*:
  - 5.1. *Fork*: Berfungsi untuk memecah aktivitas menjadi paralel atau dikerjakan secara bersamaan
  - 5.2. *Join*: Untuk menggabungkan kembali aktivitas yang bersifat paralel
6. *Merge Event*: Berfungsi untuk menggabungkan kembali flow yang telah dipecah oleh *Decision*
7. *Swimlanes*: Berfungsi untuk memecah diagram aktivitas menjadi beberapa baris dan kolom.
8. *Final State*: Menunjukkan bagian akhir dari aktivitas.

## O. Bahasa Pemrograman

Bahasa pemrograman (*programming language*) adalah sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Bahasa ini memungkinkan seseorang dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan atau diteruskan dan jenis langkah apa secara persis yang

akan diambil dalam berbagai situasi. Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program atau aplikasi. Berikut ini adalah jenis – jenis bahasa pemrograman yang kami pakai dalam proyek kami:

### O.1. *HTML (Hypertext Markup Language)*

Salah satu bahasa pemrograman yang paling banyak dipelajari, digunakan, dan diimplementasikan, tidak bisa dipungkiri, adalah bahasa *HTML*. *HTML (Hypertext Markup Language)* adalah bahasa *mark-up* yang secara definisi, dapat meng-*tag* struktur yang ada sedemikian rupa sehingga membentuk desain yang diinginkan. Saat ini *HTML* banyak digunakan sebagai *Blockbuilding* hampir semua *website* di Internet. *HTML* dapat diedit dari aplikasi sederhana *Notepad*, sampai *IDE* yang kompleks seperti *Eclipse* ataupun *Atom*

### O.2. *PHP (Hypertext Processor)*

*PHP* adalah bahasa *scripting server-side*, Bahasa pemrograman yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi Web. *PHP* singkatan dari *Hypertext Preprocessor*, yang sebelumnya disebut *Personal Home Pages*. *PHP* adalah bahasa pemrograman umum yang berarti *PHP* dapat disematkan ke dalam kode *HTML*, atau dapat digunakan dalam kombinasi dengan berbagai sistem template web, sistem manajemen konten web, dan kerangka kerja web.

### O.3. *Python*

*Python* adalah salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung (interpretatif) dengan metode orientasi objek (*Object Oriented Programming*) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan *syntax*. Sebagian lain mengartikan *Python* sebagai bahasa yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi

dengan fungsionalitas pustaka standar yang besar serta komprehensif. Walaupun *Python* tergolong bahasa pemrograman dengan level tinggi, nyatanya *Python* dirancang sedemikian rupa agar mudah dipelajari dan dipahami.

#### O.4. CSS

*CSS (Cascading Style Sheet)* adalah salah satu bahasa desain web (*style sheet language*) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda (markup language). Biasanya *CSS* digunakan untuk mendesain sebuah halaman *HTML*, *XHTML*, dokumen *XML*, *SVG*, *XUL*, dan *ANDROID*. *CSS* merupakan bahasa pemrograman yang dipakai untuk mendesain halaman depan atau tampilan website (*front end*).

#### O.5. C++

Bahasa *C++ (C Plus Plus/CPP)* adalah bahasa pemrograman yang berorientasi objek, dimana untuk menyelesaikan sebuah masalah *C++* melakukannya dengan menjelaskan *class-class* yang merupakan anak class sebagai abstraksi dari objek-objek fisik. *Class-class* pada *C++* berisi keadaan objek, anggota objek, dan kemampuan objek. Bahasa ini juga merupakan bahasa turunan dari bahasa *C*.

#### O.6. SQL

*SQL (Structured Query Language)* bahasa pemrograman yang dipakai untuk pengaksesan dan pemanipulasian data, terutama pada *DBMS* atau *Database Management System* seperti *MySql*, *Microsoft Access*, dll. Ada 3 jenis perintah yang ada dalam bahasa *SQL* antara lain:

1. *DDL (Data Definition Language)*: Berkaitan dengan pendefinisian struktur database.
  - 1.1. *CREATE*: Membuat Database dan/atau table
  - 1.2. *ALTER*: Mengubah struktur Database dan/atau tabel
  - 1.3. *RENAME*: Mengganti/mengubah nama tabel

1.4. *DROP*: Menghapus struktur Database dan/atau tabel

2. *DML (Data Manipulation Language)* : Untuk memanipulasi atau mengolah data/query dalam table.

2.1. *SELECT*: Menampilkan data – data dalam tabel

2.2. *INSERT*: Menambahkan data dalam tabel

2.3. *UPDATE*: Mengubah atau memodifikasi data dalam tabel

2.4. *DELETE*: Menghapus data dalam tabel

3. *DCL (Data Control Language)*: Berkaitan dengan pemanipulasian user dan hak akses.

3.1. *GRANT*: Pemberian hak akses pada user tertentu, perintah ini juga dapat dipakai untuk penambahan user yang dapat mengakses database

3.2. *REVOKE*: Untuk mencabut hak akses user tertentu

Untuk proyek kami, kami menggunakan *library Python* untuk pemanipulasian database, *library* tersebut bernama *MySQL-Python Connector*. Dengan *library* ini, dimungkinkan dapat mengakses database *mysql* dengan hanya menjalankan *syntax Python* tertentu.

### III. IMPLEMENTASI PROGRAM

#### A. Tahap Penelitian

Untuk proyek ini, pertama – tama kami adakan penelitian yang bertujuan untuk mencari sebanyak – banyaknya informasi dan referensi sehingga mendukung pengerjaan proyek kami. Pada tahap penelitian ini kami bagi 3 tahap, antara lain :

##### A.1. Perencanaan

Tahap pertama dalam penelitian kami adalah menentukan tema, topik, dan kasus masalah yang dapat kita selesaikan melalui solusi yang terstruktur. Tahap ini meliputi; menentukan topik, perumusan

masalah, menentukan solusi berdasarkan masalah yang ada, menyusun rencana pengerjaan proyek berdasarkan solusi, serta menyusun manfaat dan tujuan penelitian.

### A.2. Pelaksanaan

Tahap kedua adalah pelaksanaan proyek kami, langkah – langkah pelaksanaan yang kami lakukan adalah antara lain; melakukan *research* tentang komponen – komponen, aplikasi, library, algoritma pemrograman, serta bahasa pemrograman apa saja yang diperlukan untuk menunjang kesuksesan proyek kami. Kami juga merancang sistem dan desain sistem sebaik – baiknya sehingga mempermudah pemakaian oleh administrator serta pelanggan/*customer*. Setelah pembuatan proyek, kami melakukan uji coba, tidak peduli itu uji coba fungsi, performa, dll,. Dan terakhir kami melakukan pemeliharaan sistem, terutama menjaga kelancaran sistem dan perangkat elektronik.

### A.3. Penulisan Paper

Penulis membuat laporan penelitian berdasarkan pedoman penulisan dengan jenis penelitian aplikasi produk.

## B. *Requirement Analysis*

Tahap *Requirement Analysis* ini adalah penentuan berhubungan dengan kebutuhan dan tujuan sistem IoT ini dibangun. Tujuan utama sistem yang kami buat adalah untuk mengukur suhu tubuh pelanggan dan secara bersamaan mengirim data pelanggan beserta foto, dan temperature. Terdapat dua alat yang menjadi komponen utama dalam sistem ini, yang pertama adalah alat scanner, yang dilengkapi teknologi webcam, sensor infrared, dan lcd. Alat scanner ini akan menangkap foto wajah dan suhu tubuh pengguna, dan memproses serta mengirimkan pada alat yang kedua, yaitu server utama. Server utama akan menerima foto dan suhu tubuh dan membandingkan wajah yang, sebelumnya difoto dalam alat scanner, dengan data pendaftaran subjek.

Jika wajah yang dikirim oleh scanner cocok dengan database subjek, maka akan dicopy data dalam subjek pada wajah dan suhu yang ada. Sebagai fitur

tambahan, Server maupun client dapat melihat secara real-time webcam yang terdapat pada alat scanner.

Secara fungsional, diharapkan sistem ini dapat melakukan hal – hal berikut :

- Pelanggan dapat melakukan pengecekan suhu secara *real time*
- Administrator dan/atau server dapat melihat hasil stream webcam yang terdapat dalam menu camera stream pada website.
- Memberikan indikasi berupa lampu hijau untuk suhu dibawah 38° dan merah untuk suhu di atas 38 °.
- System dapat membandingkan hasil foto dari camera stream dengan sampel foto yang ada di database menggunakan *face recognition*.
- Administrator dan/server dapat mengakses data-data mengenai pengunjung berdasarkan hasil dari *face recognition*.
- Administrator dan/server dapat mengubah dan menambahkan data-data pengunjung pada Database.

## C. *Desain Sistem*

Pada desain sistem penulis akan mendeskripsikan mengenai kebutuhan sistem yang telah diterjemahkan dalam bentuk *Use Case Diagram*, *Class Diagram*, *Activity Diagram*.

### 1. *Use Case Diagram*

Pada *Use Case Diagram* ini, penulis akan menggambarkan aktor-aktor beserta perannya pada sistem yang sedang dikembangkan ini. Pada sistem ini, terdapat 2 aktor yaitu Admin dan Pengunjung. Terdapat juga 2 sistem antara lain Sistem Website, dan Sistem Scanner. Aktor Admin mempunyai hak akses lebih superior dibandingkan Pengunjung, Admin mempunyai akses untuk melihat data sampel, data pengunjung, stream camera. Sedangkan pengunjung hanya dapat melihat suhu dan indikator LED saja. Terdapat juga hubungan Sistem Website dengan Sistem Scanner, yang lain adalah Sistem Website



dapat meminta akses untuk pengiriman data webcam streaming. Dari sini, terdapat jelas bahwa Admin hanya akan mengakses Sistem Website, sedangkan pengunjung mengakses Sistem Scanner.

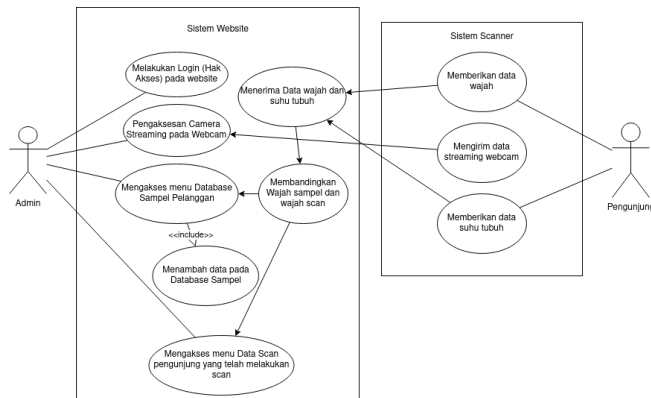
Hal yang dapat dilakukan oleh Admin pada sistem Website antara lain:

- Admin dapat melakukan Log in pada website.
- Admin dapat mengakses menu Camera stream untuk melihat hasil stream webcam secara *real time*.
- Admin dapat mengakses menu Database untuk menambahkan dan/atau mengubah data-data dari pengunjung.
- Admin dapat mengakses menu Data scan untuk melihat data-data pengunjung yang telah melakukan scan.

Hal yang dapat dilakukan oleh Pengunjung pada sistem Scanner adalah:

- Pengunjung dapat memberikan data wajah
- Pengunjung dapat memberikan data Suhu tubuh

Pada Gambar 3 terdapat visualisasi use case proyek kami.



## 2. Activity Diagram

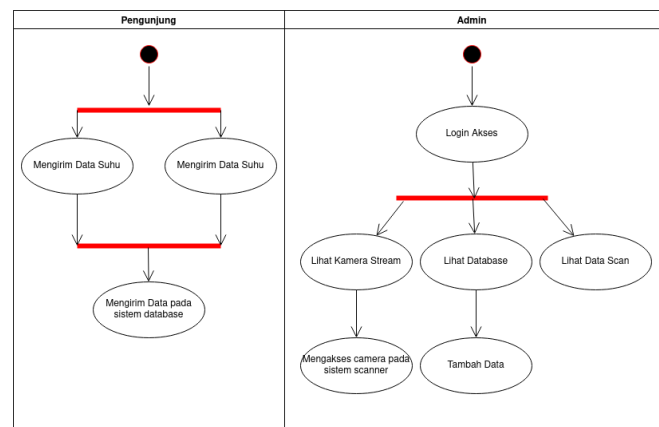
Sebagai *Starting Point*, pengunjung ingin mengecek suhu tubuhnya, maka, ia akan berhadapan dengan camera yang berteknologi face detection.

Setelah scan, data yang didapat adalah wajah dan suhu tubuh. Data wajah dan suhu tubuh akan dikirim ke sistem website untuk dibandingkan dengan sampel wajah.

Setelah pengiriman data, data wajah akan dibandingkan dengan data sampel dalam database. Jika wajah dengan sampel cocok, maka akan menyalin data dalam sampel kedalam data wajah yang baru masuk. Untuk data suhu, juga akan mengupdate suhu dalam data sampel.

Untuk seorang Administrator mempunyai akses untuk melihat data sampel dalam database, stream webcam, dan data scan yang telah masuk. Admin juga dapat menambah data sampel dengan acuan foto wajah yang jelas sehingga dapat dicocokkan dengan wajah pengunjung yang baru masuk.

Pada gambar 4 adalah *Activity Diagram* untuk proyek kami.



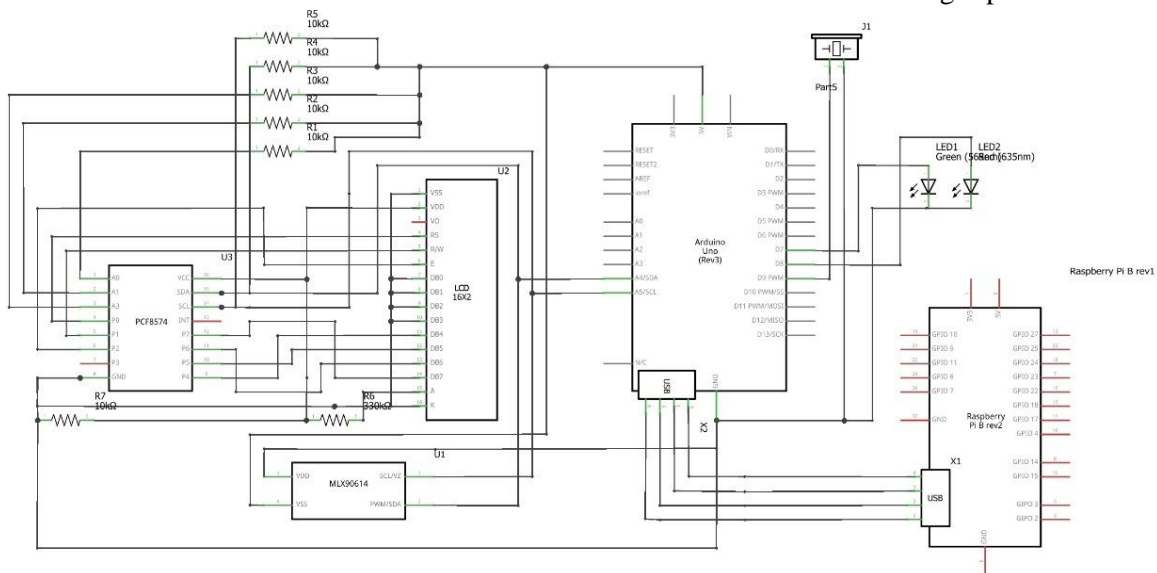
## D. Tahap Penelitian

Tahap ini sebagai klimaks dalam pengerjaan proyek kami, yaitu perakitan alat dan komponen, pengetikan bahasa program, pembuatan server dan aplikasi, dan sebagainya. Lebih jelasnya, dapat dilihat langkah – langkah atau tahap – tahap yang dikerjakan dibawah ini.

## 1. Perakitan dan pemasangan komponen – komponen

Untuk perakitan dan pemasangan, terutama komponen – komponen elektronik untuk mendukung pengerjaan ini, kami menggunakan *Mini – PC* bermerek *Raspberry Pi* dan *Arduino UNO* untuk komponen utama. Untuk komponen pendukung, kita membutuhkan *MLX90614 IR Temperature Scanner* yang berfungsi untuk mengukur suhu pengguna, *LCD I2C 16x2* untuk menampilkan jumlah suhu dalam *Celcius*. Setelah itu terdapat buzzer sebagai alarm yang akan mengindikasikan bahwa webcam telah mendeteksi adanya wajah. Terakhir, diberikan 2 *LED 5mm* berwarna merah dan hijau untuk indikator merah akan menyala jika suhu tubuh diatas 38 *Celcius*, sementara indikator hijau menyala jika tubuh dibawah 38 *Celcius*.

Dibawah ini adalah skema elektronik komponen – komponen yang diperlukan



## 2. Pengetikan dan Memasukkan bahasa pemrograman pada *Arduino UNO*

Tahap kedua adalah kita dapat memulai pengetikan bahasa pemrograman pada *Arduino UNO*, bahasa pemrograman yang kita pakai adalah C++, sementara itu *IDE (Integrated Development Environment)* yang kita pakai adalah dari produk arduino itu sendiri yaitu *Arduino IDE*. Fungsi dari

*Arduino UNO* ini akan kita program sedemikian rupa sehingga menerima pesan dari *Raspberry Pi* melalui komunikasi Serial. Setelah menerima pesan, *Arduino UNO* akan menyalakan sensor suhu tubuh. Secara otomatis akan mendapatkan suhu tubuh subjek. Setelah itu, akan dicek apakah suhu tubuh bernilai lebih dari 38 ataukah kurang dari 38. Jika suhu bernilai lebih dari 38, Indikator LED berwarna merah akan menyala. Sebaliknya jika bernilai kurang dari 38, indikator LED berwarna hijau akan menyala. Nilai suhu tubuh subjek akan di tampilkan pada layar LCD 16x2 I2C tadi.

## 3. Membangun Broker MQTT pada server

Setelah memprogram *Arduino UNO* kita bisa mulai membuat broker MQTT untuk komunikasi antara alat scanner dengan server. Kita telah mengetahui cara kerja protokol MQTT yaitu memakai fungsi *Publish* dan *Subscribe*. Dalam kasus proyek ini, kita membuat *Broker* sebagai pemfilteran komunikasi

kedua entitas ini. Karena server harus menerima data yang datang dari alat scanner, secara otomatis menjadikan server bersifat *Broker* sekaligus *Subscriber*. Bahasa yang digunakan dalam fungsi ini kami memakai bahasa *Python* dengan *Library MQTT* tambahan bernama *Paho.MQTT*. *Paho.MQTT* berfungsi sebagai *library* untuk memanipulasi komunikasi *MQTT* dengan memakai bahasa *Python*.

#### 4. Pemrograman *Raspberry Pi*

Bahasa pemrograman yang dipakai dalam tahap ini adalah *Python*. Ada beberapa fungsi dalam pemrograman *Raspberry Pi* ini. Yang pertama adalah pemrograman *face-detection* dengan memakai *library* yang telah disediakan oleh *OpenCV*, algoritma yang dipakai dalam *face-detection* ini adalah menggunakan algoritma *Haar-Cascades*. Setelah itu, kita memprogram *Raspberry Pi* sehingga dapat mengirim data pada *Arduino UNO* dengan komunikasi Serial. Singkatnya, *Raspberry Pi* akan mengirim data pada *Arduino UNO* jika ditemukan wajah pada *Haar-Cascades*. Setelah pengiriman data, akan dilakukan fungsi pemfotoan sekaligus menunggu data suhu yang akan dikirim balik oleh *Arduino UNO*. Setelah mendapatkan foto dan suhu, foto akan *diconvert* menjadi *base64* dan keduanya akan dikirim ke *subscriber* melalui protokol MQTT. Kedua pengiriman ini dilakukan secara hampir bersamaan dengan *topics* yang berbeda.

#### 5. Pembangunan *Server HTTP, Website, dan Database*

Untuk pembangunan server, kami menggunakan penyedia server *HTTP* bersifat *open-source* yang terkenal yaitu *Apache*. Pemrograman website dinamis kami menggunakan *PHP Native*, dan terakhir untuk database kami menggunakan *MySQL ber-engine MariaDB*. Dibawah ini adalah struktur database dari proyek ini.

Kami juga menggunakan *CSS Framework* bernama *Bootstrap* untuk pendesainan website kami dikarenakan kemudahan dan kecepatan yang terbilang baik. Dibawah ini adalah *screenshot* dari website yang kami buat.

#### 6. Pembuatan aplikasi *Face-Recognition* pada server

Bahasa pemrograman yang dipakai adalah *Python*, aplikasi ini kami program sedemikian rupa sehingga terhubung dengan database yang telah dibuat tadi. Untuk *library* kami menggunakan *mysql.connector* untuk pengaksesan database website, dan *OpenCV* sebagai penyedia algoritma *Face-*

*Recognition*. Algoritma aplikasi ini adalah, pertama-tama, kami buat 2 file, yang satu untuk menampung data scan yang dikirim oleh alat scanner, yang satu lagi untuk menampung data sampel algoritma *Face-Recognition*. Algoritma *Face-Recognition* yang kami pakai adalah *LBPH Cascades*.

Saat dimulai, aplikasi akan memulai *training* data dalam sampel beserta nama – nama yang diambil dalam *database*. Setelah *men-training*, aplikasi akan menunggu adanya pesan yang datang dari alat scanner. Jika terdapat data masuk, yaitu data foto dan suhu, maka akan memulai proses *decoding Base64* menjadi file .jpg, sementara suhu akan disimpan dalam sebuah *variable*. Setelah *Decoding*, hasil gambar akan dibandingkan oleh gambar dalam sampel database menggunakan *LBPH Recognizer*.

Jika terdapat gambar yang cocok, maka data dalam sampel akan disalin pada data scan pengguna. Suhu tubuh juga akan disalin pada data sampel untuk memberitahukan suhu tubuh terakhir. Setelah disalin, maka data – data akan dimasukkan kedalam database melalui fungsi – fungsi pada *library Mysql.Connector*

### IV. PERCOBAAN DAN HASIL PERCOBAAN

#### A. PERCOBAAN

Pada tahap percobaan ini penulis akan melakukan uji coba untuk memastikan bahwa produk serta aplikasi yang telah dikembangkan dapat berfungsi dengan baik. Langkah-langkah yang akan dilakukan adalah sebagai berikut:

- Mengaktifkan sistem scanner beserta server.
- Admin melakukan Log in ke aplikasi web
- Admin akan membuka menu Camera Stream untuk memeriksa bahwa stream webcam dapat berfungsi dengan baik
- Pengunjung akan melakukan scan pada alat scanner dengan meletakkan tangan di alat dan memposisikan wajah di depan webcam

- Saat scanner mendeteksi adanya wajah dari pengunjung, scanner akan memotret melalui webcam serta mengukur suhu pada sensor.
- Scanner akan menampilkan hasil dari pengukuran suhu pada layar *LCD*
- Scanner akan menyalakan lampu berwarna hijau disaat suhu yang telah diukur berada dibawah 38 derajat celcius.
- Sedangkan saat suhu berada diatas 38 derajat celcius, maka lampu berwarna merah akan menyala.
- Scanner akan mengirimkan data hasil scan berupa foto serta suhu menuju server.
- Server akan menjalankan proses *face recognition* dengan mencocokkan foto hasil scan dengan foto yang ada pada *database*.
- Admin dapat membuka menu *data scan* untuk memeriksa data dari pengunjung yakni suhu dan foto beserta data-data yang ada di dalam database yang telah dicocokkan oleh sistem.
- Admin juga dapat mengecek serta menambahkan data yang ada pada *database*.
- Admin melakukan logout untuk menghindari terjadinya pencurian data yang tidak disengaja.

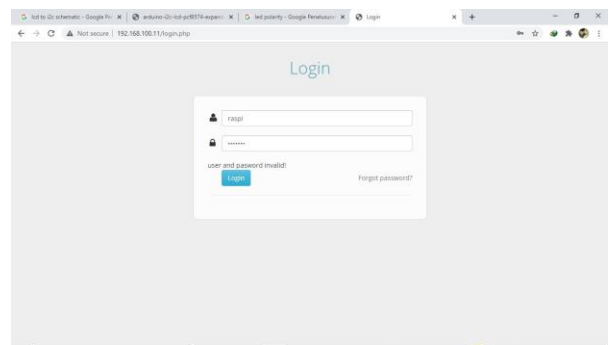
Hal yang ingin kami uji pada sistem kami adalah:

- Keakuratan pengukuran suhu pada sensor *infrared*.
- Kecepatan dalam melakukan pendeteksian wajah.
- Keakuratan dalam pencocokkan wajah pada server.
- Kesesuaian data yang dimunculkan pada *data scan* dengan data yang berada pada *database*.
- Kelancaran pada proses stream kamera.

## 2. Hasil Percobaan

Berikut ini adalah hasil dari uji coba yang telah penulis lakukan:

1. Pada scanner, alat dapat menyala dengan baik setelah dihubungkan dengan *power supply*, serta memiliki daya yang memadai.
2. Aplikasi website ini telah terenkripsi dengan baik.



**Gambar 1:** *Login Invalid*

3. Stream kamera telah teruji dapat bekerja



secara *real time*.

**Gambar 2:** *Stream Camera*

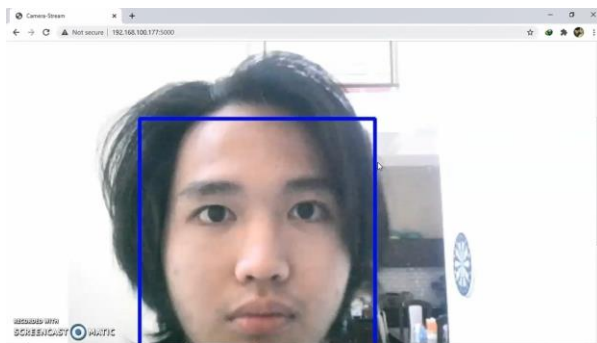
4. Indikator lampu pada *scanner* telah teruji mampu bekerja dengan baik.



P

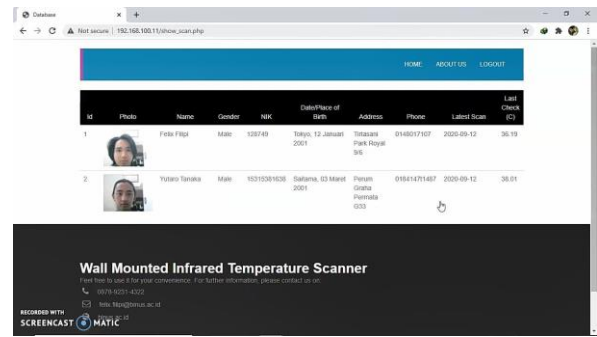


5. Sistem pendeteksian wajah telah teruji bekerja secara cepat.



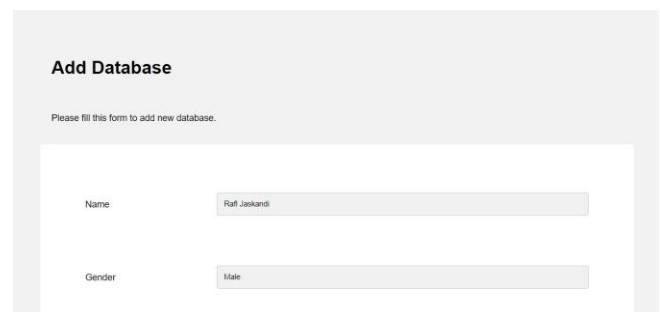
**Gambar 4:** *Face Detection*

6. Sistem pencocokkan wajah telah teruji bekerja secara akurat.
7. Data dari database yang dimunculkan pada scan data telah terbukti sama.

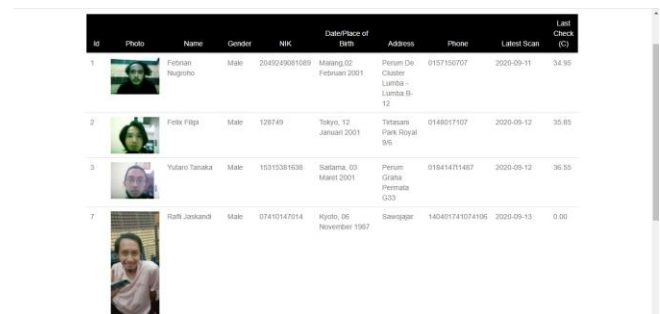


**Gambar 5:** *Face recognition dan penyamaan data*

8. Database dapat menambahkan data baru.



**Gambar 6:** Menambahkan data



**Gambar 7:** Tampilan database

## V. KESIMPULAN DAN SARAN

### Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi serta pengujian sistem yang telah dilakukan bisa disimpulkan beberapa hal sebagai berikut:

- Aplikasi *Wall Mounted Infrared Temperature Scanner* ini dapat menampilkan data sesuai



dengan analisis kebutuhan, dimana tampilan aplikasi ini telah *responsive* atau dapat menyesuaikan dengan tampilan gawai yang tengah digunakan, sehingga dapat diakses dari berbagai macam perangkat.

- Produk Wall Mounted Infrared Temperature Scanner telah diimplementasikan sesuai dengan analisis kebutuhan dan dapat bekerja dengan baik berdasarkan hasil percobaan yang telah dilakukan. Pengguna dapat menguji suhu tubuh mereka dengan tepat. Sedangkan admin dapat melihat semua yang terjadi pada kamera secara *real time*, dapat melihat serta menambahkan database, melihat hasil dari uji suhu tubuh subjek yang ada.
- Dengan adanya aplikasi Wall Mounted Infrared Temperature Scanner ini pengunjung dapat menjadi lebih mudah dan leluasa untuk mengukur suhu tubuh mereka.
- Dengan adanya aplikasi Wall Mounted Infrared Temperature Scanner ini admin dapat mengecek atau mengabsen penggunanya. Sebagai contoh pengabsenan pekerja pada perusahaan, pengabsenan anggota perpustakaan, dan banyak lagi
- Dengan adanya aplikasi Wall Mounted Infrared Temperature Scanner ini admin dapat mencegah persebaran Covid-19 pada tempat tersebut.

## SARAN

Untuk mengembangkan produk ini menjadi lebih baik, penulis telah menyiapkan beberapa saran :

- Meningkatkan lebih lanjut keakuratan dari sistem *face recognition* pada produk.
- Meningkatkan lebih lanjut keakuratan dari sensor pengukur suhu tubuh pada produk.

- Meningkatkan sistem keamanan dari aplikasi untuk menghindari adanya tindak kecurangan dalam melakukan pengujian.
- Merubah desain produk menjadi lebih fungsional serta memiliki daya estetik lebih, untuk meningkatkan perasaan nyaman dari pengguna

## VI. Daftar Pustaka

1. A. (2018a, June 2). *Pengertian Use Case - arifwicaksanaa*. Medium. <https://medium.com/@arifwicaksanaa/pengertian-use-case-a7e576e1b6bf>
2. Brisevac, M. (2019, September 20). *what is base64, benefits*. Inchoo. <https://inchoo.net/magento/what-is-base64-encoding-and-how-can-we-benefit-from-it/>
3. G., D. (2020, June 17). *What is Apache? An In-Depth Overview of Apache Web Server*. Hostinger Tutorials. <https://www.hostinger.com/tutorials/what-is-apache>
4. Prado, K. S. D. (2018, June 19). *Face Recognition: Understanding LBPH Algorithm - Towards Data Science*. Understanding LBPH Algorithm. <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>
5. *Introduction to Flask — Python for you and me 0.4.beta1 documentation*. (n.d.). <https://pymbook.readthedocs.io>. Retrieved 12 September 2020, from <https://pymbook.readthedocs.io/en/latest/flask.html>
6. Pulli, K., Baksheev, A., Korniyakov, K., & Eruhimov, V. (2012). Realtime Computer Vision with OpenCV. *Queue*, 10(4), 40–56. <https://doi.org/10.1145/2181796.2206309>

7. *Structured Query Language (SQL)*. (2019, November 27). IDCloudHost. <https://idcloudhost.com/kamus-hosting/structured-query-language/>
8. Team, T. H. (2015, January 19). *Publish & Subscribe - MQTT Essentials: Part 2*. Publish & Subscribe - MQTT Essentials: Part 2. <https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/>
9. V. (2018b, February 5). *Apa yang dimaksud dengan Diagram aktivitas atau Activity Diagram ?* Dictio Community. <https://www.dictio.id/t/apa-yang-dimaksud-dengan-diagram-aktivitas-atau-activity-diagram/15129/2>
10. *What is Use Case Diagram?* (n.d.). Visual Paradigm. Retrieved 12 September 2020, from <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
11. Widyana, W. I. (2020, February 14). *Apa itu HTML? Bagaimana Cara Menggunakannya?* PinDexain. <https://www.pindexain.com/apa-itu-html/>