

SISTEM UNTUK MENKLASIFIKASIKAN EMOSI DAN MENDETEKSI WAJAH PADA PEMBELAJARAN DARING

Leonard Christopher Limanjaya¹, Handry Khoswanto¹, Indar Sugiarto¹

¹Program Studi Teknik Elektro, Universitas Kristen Petra Jl. Siwalankerto 121-131, Surabaya 60236, Indonesia

E-Mail: leonardchristopher002@gmail.com, handry@petra.ac.id ; indi@petra.ac.id

Abstrak – Sistem pembelajaran *offline* masih belum dapat tergantikan oleh pembelajaran *online*. Kualitas belajar anak dapat dilihat dari ketertarikan anak saat belajar. Penelitian ini bertujuan untuk merancang program untuk mendeteksi kehadiran siswa (*behavioral engagement*) dan mengklasifikasikan emosi siswa (*emotional engagement*) dalam pembelajaran *online*. Pendeteksian *engagement* ini dilakukan melalui kamera dan dikirimkan ke database. Sistem ini juga menyediakan halaman web bagi guru untuk melihat dan mengevaluasi data siswa.

Sistem ini dibuat terpisah dari platform pembelajaran *online*. Sistem ini akan menggunakan laptop siswa untuk mengolah gambar dari webcam. Program akan memproses gambar untuk mengambil data tentang keterlibatan siswa saat belajar. Ini menerapkan konsep *deep learning* menggunakan bahasa pemrograman Python. Sebelum tahap klasifikasi emosi, program akan mengecek kehadiran siswa dengan deteksi wajah.

Sistem dapat melakukan deteksi wajah dan klasifikasi emosi ekspresi wajah berdasarkan hasil pengujian. Saat dilakukan pengujian pada anak-anak, akurasi sistem ini untuk ekspresi wajah adalah 74,8%. Sistem juga berhasil menampilkan data bacaan bahan evaluasi guru pada website.

Kata Kunci— Artificial Intelligence, CNN, Face Detection, Website, Desktop Application, dan API Gateway

I. PENDAHULUAN

Saat ini sistem belajar secara online dapat dikatakan masih belum lebih baik dari sistem belajar yang dilakukan secara offline. Masih banyak siswa yang belum puas dengan sistem belajar secara online. Rasa gelisah, gugup, dan sebab muncul saat mereka menjalani sistem belajar tersebut. Perasaan tersebut dapat timbul akibat kurangnya interaksi secara langsung baik dengan temannya maupun dengan pengajar. Berdasarkan survei yang dilakukan Top Hat, 77% siswa mengatakan bahwa *engagement* dalam pembelajaran online ini kurang. Kurangnya interaksi dan pengalaman belajar online yang tidak menarik menyebabkan proses belajar mengajar kurang maksimal [1].

Menilik dari kebutuhan tersebut, penelitian ini diharapkan dapat membantu guru untuk menyesuaikan cara atau metode mengajar mereka sesuai kondisi kelas seperti saat pembelajaran *offline*. Ada 3 tipe dari *engagement* siswa, *emotional*, *behaviour*, dan *cognitive* [2] Dari ketiga tipe *engagement*, Ding Dong ingin membuat sistem mengukur 2 tipe *engagement* (*Emotional* dan *Behavior*) dengan sebuah kamera. *Emotional Engagement* dapat dilihat melalui apakah emosi mereka lebih cenderung positif atau negatif dalam belajar. Sedangkan untuk *behavior engagement* dapat dilihat melalui kehadiran siswa saat belajar dan tingkah lakunya. Kedua hal ini dapat dilihat dengan bantuan sebuah kamera.

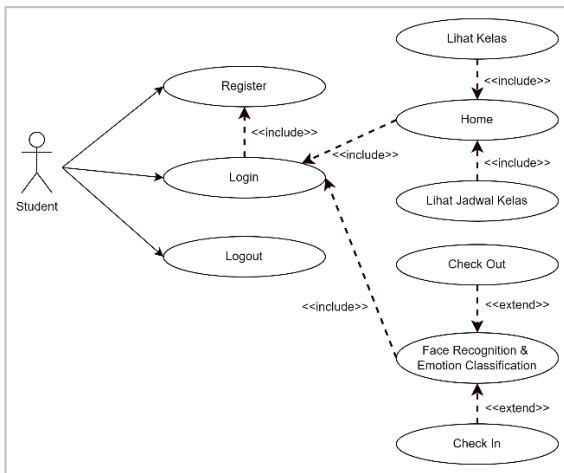
Dalam mengklasifikasi emosi siswa dalam belajar, sistem ini dapat menggunakan pengolahan gambar dengan metode *deep learning*. Menurut Paul Ekman Ada 6 emosi dasar yang secara universal dapat dialami oleh manusia. Emosi ini dapat diklasifikasi sebagai bahagia (*happiness*), sedih (*sad*), marah (*angry*), jijik (*disgust*), takut (*fear*), terkejut (*surprise*) [3]. Berbeda dengan *Behavioural Engagement*, dimana dapat dilihat dari kehadirannya saat kelas dan atensinya saat pembelajaran berlangsung [4]. Dalam kelas *online*, kehadiran seorang siswa dapat diidentifikasi secara visual melalui kamera. Gambar yang didapat oleh kamera dapat diproses untuk mendeteksi apakah siswa tersebut hadir di depan kamera. *Behaviour engagement* ini merupakan poin yang penting dalam pembelajaran *online*, tetapi tidak dapat mendefinisikan secara utuh usaha belajar siswa. Pengajar harus melihat aspek *engagement* lain untuk mendefinisikan usaha belajar seorang siswa [5].

Terdapat referensi penelitian yang serupa dengan penelitian ini, penelitian tersebut membangun *real-time* sistem pendeteksian ekspresi wajah. Hal ini dibuat karena melihat bahwa depresi merupakan sebuah masalah kesehatan yang tidak diperhatikan dan dapat mempengaruhi prestasi siswa dalam belajar. Dengan sistem ini guru dapat memantau suasana hati siswa melalui aktivitas kelas. Sistem ini harus cukup handal ketika dijalankan pada komputer dengan spesifikasi *mid-end*. Siswa akan diberikan sebuah kuisioner untuk mengukur *stress* mereka, yang hasilnya digunakan bagi guru untuk meminimalisir *stress* siswa. Penelitian ini dilakukan pada siswa tingkat sekolah dasar kelas 4 hingga 5. Pada penelitian ini, sistem ini berhasil membantu dalam mengurangi *stress* siswa [6].

Pada penelitian tersebut pendeteksian ekspresi wajah digunakan pada saat pembelajaran *offline* dan terbukti dapat membantu agar pengalaman belajar lebih baik. Sistem pada penelitian ini memiliki tujuan yang serupa, yaitu memantau emosi siswa melalui ekspresi wajah siswa tersebut. Guru dapat memantau hal tersebut melalui *website* yang telah disediakan. Hal ini dapat bermanfaat bagi guru agar dapat meminimalisir *stress* yang dialami oleh siswa. Sehingga dalam penelitian ini akan dibuat sebuah aplikasi untuk memudahkan guru memantau *engagement* siswa secara online. Guru juga dapat melihat hasil pembacaan emosi melalui website yang diberikan. Data tersebut dapat guru gunakan untuk mengevaluasi proses belajar mengajar.

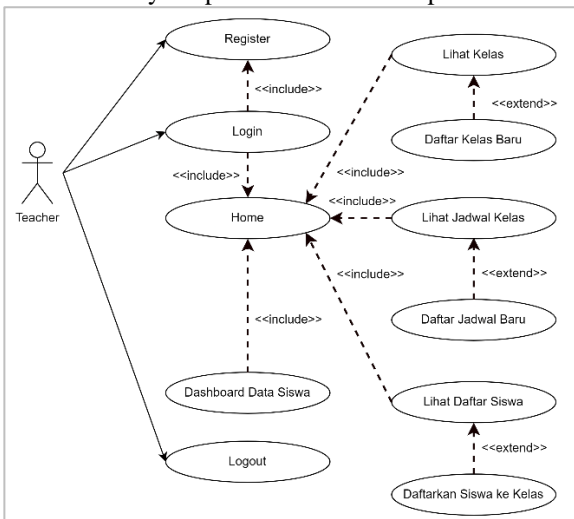
II. DESKRIPSI SISTEM

Sistem yang dirancang pada penelitian ini dipergunakan untuk dapat mengklasifikasi emosi siswa berdasarkan ekspresi wajah. Selain itu juga dapat digunakan untuk mengetahui kehadiran siswa menggunakan pendeteksian wajah dalam pembelajaran *online*. Sistem ini juga dapat dimanfaatkan bagi guru untuk melihat dan mengevaluasi performa siswanya dalam belajar. Oleh karena itu sistem ini akan digunakan oleh 2 jenis *client*, yaitu guru dan siswa. Guru dan siswa memiliki hak akses dan fitur yang berbeda-beda. Secara umum, jumlah dari *client* dari guru dan siswa yang menggunakan atau mengakses tidak dibatasi. Guru memiliki fitur untuk mengatur kelas yang mereka ajar, jadwal kelas beserta siswa yang mengikuti kelas mereka. Namun pada sisi siswa, siswa hanya dapat melihat atau memantau kelas yang mereka ikuti dan jadwal kelas tersebut.



Gambar 1. Use Case Diagram Siswa

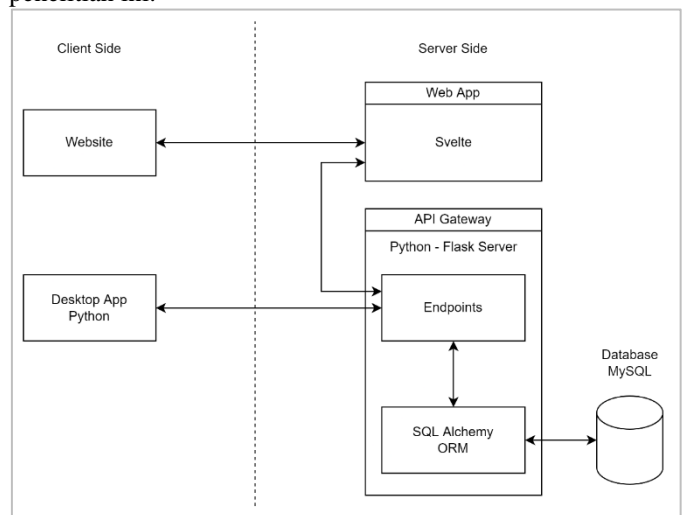
Melalui gambar diperlihatkan fitur apa saja yang dapat diakses oleh siswa. Setelah *login*, siswa dapat melakukan proses *face recognition* dan *emotion classification*. Saat kelas sudah dimulai siswa dapat menekan tombol *check in* untuk memulai pengiriman data ke *server* dan tombol *check out* saat kelas telah selesai dan menghentikan proses pengiriman data. Namun fitur-fitur tersebut terbagi dalam 2 aplikasi yang dapat diakses untuk siswa. Siswa dapat mengakses lihat kelas dan jadwal kelas tersebut hanya melalui *website*. Sedangkan fitur *face recognition* dan *emotion classification* beserta *check in* dan *check out* hanya dapat diakses melalui aplikasi desktop.



Gambar 2. Use Case Diagram Guru

Pada memperlihatkan *use case* diagram untuk guru. Guru memiliki hak akses lebih untuk mengatur kelas, jadwal kelas, dan siswa mereka. Berbeda dengan siswa, guru dapat mendaftarkan jadwal kelas beserta siswa mereka pada kelas-kelas tertentu. Selain itu dari data *face recognition* dan *emotion classification* yang dikirimkan siswa, guru dapat melihat data tersebut melalui *dashboard* data siswa. Data ini akan disajikan dalam bentuk grafik untuk memudahkan membaca data bagi guru. Bila siswa dapat *login* atau akses melalui *website* dan aplikasi desktop, guru hanya dapat *login* melalui *website*. Pada sistem ini guru mengatur kelasnya sendiri, baik itu menambahkan, mengurangi jumlah kelas, jadwal, beserta siswa.

Dalam memenuhi kebutuhan dari *use case* diagram, maka dalam penelitian ini akan dibuat dua macam *client*, yaitu aplikasi *website* dan aplikasi desktop. Dalam hal komunikasi, pada penelitian ini akan memanfaatkan protokol HTTP untuk berkomunikasi antara *client* dengan *server*. Komunikasi ini akan ditangani oleh *API gateway*. *API gateway* ini juga akan berhubungan dengan *database* untuk melakukan *query* baik untuk menyimpan atau mengambil data. Pada sistem ini akan memanfaatkan ORM untuk membantu melakukan *query*. Gambar di bawah ini akan menggambarkan arsitektur dari penelitian ini.



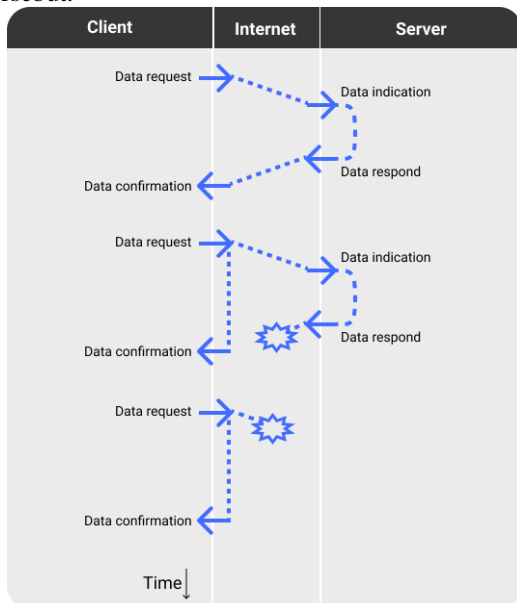
Gambar 3. Arsitektur Sistem

Pada sisi *client*, terdapat dua macam, yaitu *web client* dan *desktop app*. *Web client* akan melakukan *request* halaman pada *web app* yang terdapat pada server. Web server ini sendiri dibuat dengan memanfaatkan Svelte. Sedangkan Desktop App ini dibuat dengan menggunakan bahasa pemrograman Python. Aplikasi desktop ataupun *Web App* akan melakukan pengambilan dan memasukkan data pada *API Gateway* yang dibuat dengan Flask dengan bahasa pemrograman Python. Flask ini akan dimanfaatkan untuk membuat *endpoints* untuk meng-handle API. Dari *endpoints* tersebut, maka program akan melakukan *query* pada MySQL database dengan memanfaatkan SQL Alchemy.

A. Desain Protokol Komunikasi Data

Protokol komunikasi data antara aplikasi desktop ataupun *web app* dengan *API gateway* akan dilakukan dengan protokol HTTP. *Client* akan melakukan *request* data dengan protocol HTTP sesuai dengan *Endpoint* yang dituju. Layanan komunikasi yang digunakan pada sistem ini adalah *user-*

confirm, karena *request* data hanya dilakukan ketika terdapat interaksi oleh *user*. Contohnya saat *user* akan *login*, *user* perlu menekan tombol *login* untuk dapat mengimkan data, contoh lainnya adalah saat guru ingin melihat data kelas yang diajar, maka guru perlu untuk pindah halaman kelas untuk mendapatkan data tersebut. Ketika *client website* melakukan *request* untuk halaman *website*, komunikasi ini juga memanfaatkan *protocol* HTTP. Data yang dikirimkan baik oleh *client* menuji aplikasi *server* maupun sebaliknya, menggunakan file JSON (*JavaScript Object Notation*). File dengan ekstensi JSON ini digunakan untuk memudahkan format data karena struktur informasi pada file JSON lebih terjaga. Gambar 4 menjelaskan *time sequence diagram* dari komunikasi *client* dengan API gateway, termasuk aplikasi web dengan API gateway. Bila *client* melakukan *request* data, ketika *server* menerima maka *server* akan melakukan proses data tersebut.



Gambar 4. Time sequence diagram komunikasi dengan API Gateway

B. Desain API Gateway

Aplikasi *server* ini dibuat dengan menggunakan *text editor* VS Code. Aplikasi *server* ini digunakan untuk menerima *request* sesuai API Gateway lalu mengirimkan kembali data yang diinginkan *user* sebagai *respond*. Aplikasi ini dibuat dengan bahasa pemrograman python dengan beberapa *library* pendukung, yaitu:

- Flask
- Flask_sqlalchemy
- Werkzeug
- Flask_cors
- Json
- Datetime

Aplikasi *server* ini diimplementasikan pada *server* milik IoT Universitas Kristen Petra. Aplikasi *server* ini dijalankan pada port 5000, sehingga link untuk aplikasi *server* ini adalah: iot.petra.ac.id:5000

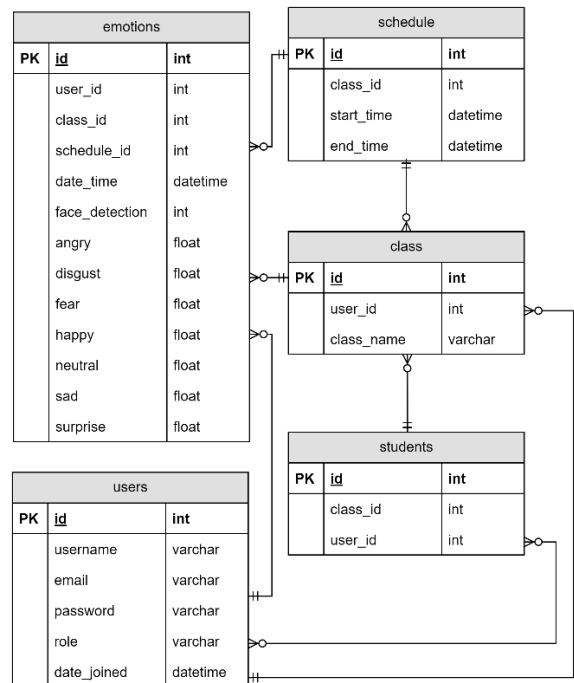
C. Database Server

Pada tugas akhir ini digunakan *database* MySQL dengan versi *server* 8.0.29 dan dijalankan pada *server* berbasis ubuntu 20.04.3. *Database* ini memiliki 5 tabel yang saling memiliki hubungan. Gambar 5 memperlihatkan isi kolom dalam masing-

masing tabel beserta tipe data dari masing-masing kolom. Serta pada gambar tersebut memperlihatkan maka kolom yang menjadi *primary key* dari masing-masing tabel.

Semua informasi mengenai *user* akan disimpan pada tabel *users* dengan *id* menjadi *primary key*. Pada tabel ini data *password* sudah disimpan dalam format yang sudah di-hash-ing. Data setiap kelas yang ditambahkan atau diajarkan oleh guru akan disimpan pada tabel *class* dengan setiap kelasnya yang memiliki *id* tertentu sebagai *primary key*. Setiap jadwal dari kelas akan disimpan pada tabel *schedule* dengan menyimpan waktu mulai dan waktu berakhirnya kelas terhadap *id* kelas. Tabel *schedule* ini akan menyimpan setiap datanya dengan memiliki *id* sebagai *primary key*. Setiap siswa yang menjadi siswa di kelas tertentu akan disimpan pada tabel *students*, tabel ini memiliki *id* sebagai *primary key*. Setiap data emosi dan *face detection* disimpan pada tabel *emotions*. Tabel *emotions* akan menyimpan data emosi dan *face detection* terhadap *id* kelas, *id* *schedule*, dan *id* *user*. Tabel *emotions* juga memiliki *id* sebagai *primary key*.

Berikut ini merupakan ERD dari database yang digunakan pada penelitian ini:



Gambar 5. Entity Relation Diagram

D. Website

Pada pembuatannya kode program *website* ini dibuat dengan menggunakan memanfaatkan *text editor* VS Code. Aplikasi *website* ini juga dijalankan dengan menggunakan npm, dimana akan membutuhkan Node.js. Aplikasi ini dijalankan pada *server* iot.petra.ac.id dengan port 2801. Node.js yang dimiliki oleh *server* ini ada pada versi 16.15.0. *Website* ini dibuat dengan memanfaatkan bahasa pemrograman JavaScript dengan didukung oleh Svelte yang merupakan salah satu *framework* dari javascript. *Library* ini dimanfaatkan untuk membuat tampilan sebuah *website* atau UI. Pada *website* ini ada beberapa halaman, berikut ini merupakan daftar dari halaman yang ada:

- Index.svelte
Halaman ini merupakan halaman utama saat *website* ini dikunjungi, namun halaman ini menampilkan informasi tambahan saat *user* telah *login*. Saat *user* sudah *login*,

maka akan menampilkan *username* beserta ID dari *user* tersebut.

- *Auth/Login.svelte*
Halaman *login* ini digunakan oleh *user* untuk bisa *login*.
- *Auth/Register.svelte*
Halaman ini digunakan agar *user* dapat mendaftarkan akun mereka baik itu untuk guru ataupun siswa.
- *Student/ClassShow.svelte*
Setiap siswa dapat melihat kelas apa saja yang mereka ikuti, mereka dapat melihat daftar kelas tersebut melalui halaman ini.
- *Student/ScheduleSetting.svelte*
Setiap siswa juga dapat melihat daftar jadwal dari setiap kelas mereka melalui halaman ini.
- *Teacher/ClassSetting.svelte*
Halaman ini hanya dapat diakses oleh guru, dan halaman ini digunakan bagi guru untuk melihat, mendaftarkan, dan menghapus kelas.
- *Teacher/Dashboard.svelte*
Halaman ini dapat digunakan guru untuk memantau atau melihat data siswa sesuai dengan kelas, jadwal dan siswa yang dipilih.
- *Teacher/ScheduleSetting.svelte*
Sedangkan halaman ini dapat digunakan oleh guru untuk dapat mendaftarkan, melihat, serta menghapus jadwal dari sebuah kelas.
- *Teacher/StudentSetting.svelte*
Daftar siswa setiap kelasnya dapat diatur melalui halaman ini. Seorang guru dapat melihat, mendaftarkan, dan menghapus siswa dari sebuah kelas.

Dalam melakukan perpindahan halaman, aplikasi web ini akan memanfaatkan *library svelte-routing*. Melalui *library* ini, web ini dibuat untuk dapat meng-*handle routing* berdasarkan direktori atau halaman yang dituju. Selain itu juga perpindahan halaman (navigasi) juga dapat dilakukan melalui *library svelte-routing*. Sedangkan halaman *website* pada tugas akhir ini dibuat dengan *web components*. Dengan *web components* ini dapat memudahkan pembuatan agar tidak perlu menuliskan kode yang sama secara berulang-ulang.

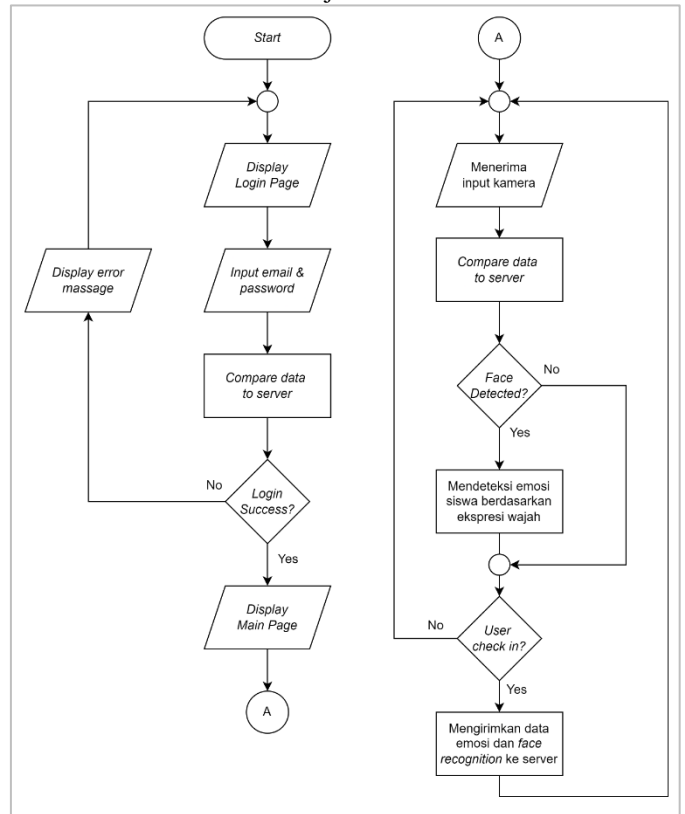
E. Aplikasi Desktop

Saat *user* menjalankan aplikasi, maka *user* akan dihadapkan dengan halaman *login*. Pada halaman ini *user* harus *login* terlebih dahulu dengan *email* dan *password* yang mereka miliki. Saat mereka menekan tombol *login*, maka aplikasi akan mengirimkan data tersebut ke *server API Gateway* untuk melakukan komparasi. Bila *server* memberikan respon '*success*' maka halaman dapat berpindah pada halaman utama, namun bila tidak, maka aplikasi akan memberikan *error message* dan *user* dapat memasukkan *email* dan *password* kembali.

Pada halaman utama, program akan mulai membaca gambar dari kamera. Kamera yang digunakan dapat dipilih oleh *user*. Program akan menganalisis gambar pada kamera, bila terdeteksi adanya wajah, maka program akan memprediksi wajah tersebut menunjukan klasifikasi ekspresi wajah apa. Ekspresi wajah yang dapat dibaca ada 7 macam ekspresi yaitu: terkejut, senang, netral, sedih, takut, marah, dan jijik. Setelah mendapat emosi atau ekspresi wajah tersebut, maka program akan melakukan pengecekan apakah *user* telah *check in*. Bila belum maka program tidak akan mengirimkan data yang dibaca

ke *server*. Bila sudah maka program akan mengirimkan data tersebut ke *server*. Sebelum melakukan *check in*, *user* harus memilih kelas dan jadwal yang sesuai, bila waktu yang dipilih tidak sesuai maka *check in* akan gagal.

Bila program mendeteksi bahwa *user* belum *check in*, maka program tetap akan melanjutkan pembacaan data, namun tidak mengirimkannya. Pembacaan data ini tetap dilakukan agar pengguna dapat mengetahui apakah sistem dapat bekerja dan memilih kamera yang digunakan. *Check in* hanya digunakan untuk mengirimkan data pada server. Meskipun wajah tidak terdeteksi dan sudah *check in*, pengiriman data tetap dilakukan ketika terdapat perubahan. Hal ini tetap dilakukan untuk memberi informasi bahwa wajah tidak terdeteksi.



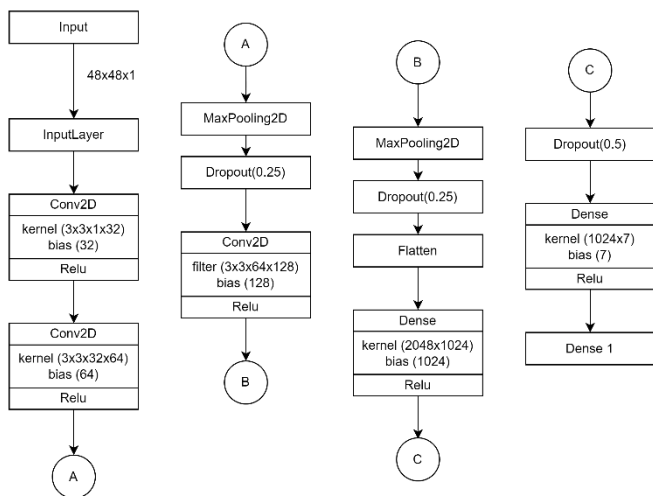
Gambar 6. Flow chart aplikasi desktop

Agar program ini dapat digunakan pada laptop atau device lain, maka program python akan dijadikan menjadi extensi '.exe'. File program '.exe' akan dapat dijalankan pada device dengan OS windows lainnya. Program ini akan dikonversi menjadi '.exe' dengan bantuan library python auto-py-to-exe. Saat sudah dikonversi menjadi '.exe', ukuran program akan menjadi sekitar 2.7 GB.

F. Model Machine Learning

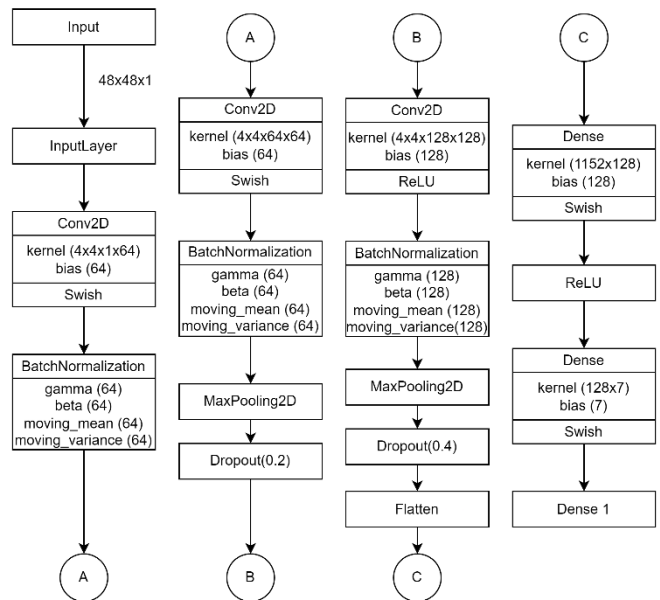
Pada tugas akhir ini ada 2 macam model yang akan dibahas dan diujikan. Pada sub bab di bawah ini akan membahas mengenai ketiga model tersebut. Setiap model ini di-*training* oleh *dataset* dari FER-2013. *Dataset* ini sendiri memiliki 35887 gambar *grayscale* dengan ukuran 48x48 *pixel*. *Dataset* ini memiliki 7 emosi yang berbeda yaitu: marah, jijik, sedih, takut, netral, senang, dan terkejut. *Dataset* ini dibuat oleh Pierre-Luc Carrier dan Aaron Courville sebagai proyek penelitian. *Dataset* ini dapat diakses pada <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>.

Model ini menggunakan arsitektur *convolutional neural network* (CNN). Pada model ini memanfaatkan *convolutional layer*, *max pooling layer*, *drop out layer*, *flatten layer* dan *dense layer*. Input layer akan menerima gambar berukuran 48x48 pixel yang kemudian diteruskan pada *convolutional layer*, *max pooling layer*, dan *drop out layer*. Layer *max pooling* ini digunakan untuk melakukan reduksi dimensi dari *features map*. Hal ini berguna untuk mengurangi jumlah parameter untuk dipelajari dan komputasi dari jaringan. Sedangkan layer *dropout* digunakan untuk mencegah *overfitting*. Model ini di-*training* dengan *dataset* dari FER-2013 dan dapat mencapai tingkat akurasi 63,2% dalam 50 *epochs* pada *test set*. Model A yang digunakan pada tugas akhir ini didapat dari <https://github.com/liviaellen/engagementdetector> [7]. Model ini memiliki akurasi yang baik sehingga dipilih penulis sebagai model dalam sistem ini.



Gambar 7. Arsitektur Model A

Model B merupakan hasil modifikasi dari referensi yang didapat dari kaggle dengan *link*: <https://www.kaggle.com/code/sankalpsrivastava26/face-emotion-recognition-using-tensorflow> [8]. Sebelum dimodifikasi penulis model ini memiliki akurasi sebesar 63.19%. Arsitektur dari Model B ini lebih kompleks dari Model A. Arsitektur dari model ini dapat dilihat pada Gambar 8.



Gambar 8. Arsitektur Model B

III. PENGUJIAN SISTEM DAN ANALISA

A. Pengujian dan Analisa Performa Aplikasi Desktop

Pengujian ini dilakukan dengan cara menjalankan aplikasi desktop ke 2 jenis laptop dengan spesifikasi yang berbeda. Berikut ini spesifikasi laptop yang digunakan:

1. Laptop A
CPU : AMD Ryzen 7 4800H
RAM : 16 GB
GPU : Nvidia GeForce GTX 1650 Ti
2. Laptop B
CPU : Intel i3-4010U
RAM : 4 GB
GPU : Nvidia GeForce GT 740M

Berikut adalah hasil percobaan dari pengujian tersebut:

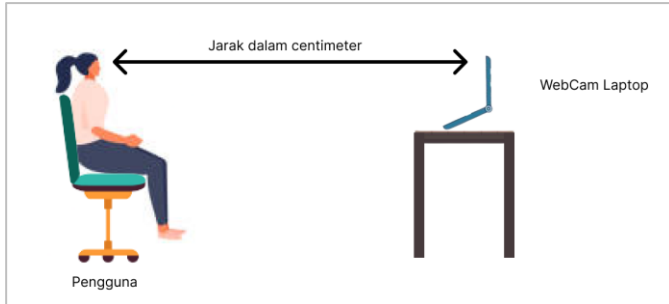
Tabel 1. Hasil Pengujian Performa Aplikasi

	Laptop A	Laptop B
CPU Usage	20.6%	52.5%
RAM Usage	352 MB	329.9 MB
GPU Usage	2.3%	1.7%
Average FPS	16.09	7.7

Setelah dilakukan pengujian, hasil pengujian tersebut menunjukkan aplikasi dapat berjalan dengan sangat baik saat dijalankan pada komputer spesifikasi tinggi. Namun saat dijalankan pada komputer dengan spesifikasi rendah, terdapat penurunan FPS dan penggunaan CPU yang jauh lebih besar. Hal ini dapat disebabkan karena komputasi dari *machine learning* yang berat. Berdasarkan rata-rata FPS saat diuji pada kedua laptop, program ini dapat berjalan dengan baik. Hasil pembacaan klasifikasi wajah dan pendeteksian wajah dapat berjalan dengan 15 FPS pada komputer spesifikasi tinggi dan 7.7 FPS pada laptop B (dengan spesifikasi yang lebih rendah). Performa dari aplikasi yang dibuat pada tugas akhir ini dapat dilihat dari FPS pembacaan klasifikasi wajah dan pendeteksian wajah.

B. Pengujian dan Analisa Fitur Face Detection Berdasarkan Jarak

Pengujian ini dilakukan untuk mengetahui *range* efektif dari seseorang agar wajah tersebut bisa dideteksi. Proses pengujian ini dilakukan dengan mengukur jarak antara tempat pengguna duduk hingga ke kamera laptop. Kondisi saat pengujian ini adalah menggunakan kamera *webcam* bawaan penulis, dengan spesifikasi dengan resolusi 720p. Pengujian ini dilakukan pada ruangan berukuran 3.5x3.5meter dengan pencahayaan menggunakan lampu 12.5W. Ilustrasi metode pengujian dapat dilihat pada Gambar 9.



Gambar 9. Ilustrasi pengujian face detection berdasarkan jarak.

Dalam melakukan pengujiannya, dilakukan secara bertahap dengan pengguna maju setiap 10 cm. Jika program mengeluarkan data berupa wajah terdeteksi maka pada jarak tersebut dinyatakan terdeteksi. Namun bila tidak terbaca maka dinyatakan tidak terdeteksi. Pengujian ini dilakukan oleh pengguna sebanyak 2 orang. Wajah yang diuji bisa dilihat pada Gambar 10.



Gambar 10. Gambar wajah yang digunakan saat pengujian (Leonard – kiri dan Wilson – kanan)

Tabel 2. Tabel Pengujian Fitur Pendeteksian Wajah Berdasarkan Jarak

Jarak (cm)	Pengguna 1	Pengguna 2
10	Tidak Terdeteksi	Tidak Terdeteksi
20	Tidak Terdeteksi	Tidak Terdeteksi
30	Tidak Terdeteksi	Terdeteksi
40	Terdeteksi	Terdeteksi
50	Terdeteksi	Terdeteksi
60	Terdeteksi	Terdeteksi
70	Terdeteksi	Terdeteksi
80	Terdeteksi	Terdeteksi
90	Terdeteksi	Terdeteksi
100	Terdeteksi	Terdeteksi
110	Terdeteksi	Terdeteksi
120	Terdeteksi	Terdeteksi
130	Terdeteksi	Terdeteksi
140	Terdeteksi	Terdeteksi
150	Terdeteksi	Terdeteksi

Berdasarkan data pengujian *face detection*, bisa diketahui bahwa kemampuan untuk mendeteksi wajah tidak dapat dilakukan karena terlalu dekat. Pada saat wajah terlalu dekat ada fitur-fitur atau titik referensi dari wajah yang tidak masuk

ke dalam kamera. Sehingga hal tersebut dapat menyebabkan program atau *haar cascade* tidak dapat mendeteksi adanya wajah. Sedangkan hasil pengujian terjauh ada pada jarak 300 cm. Program masih dapat berfungsi dengan baik. Jarak efektif dari pembacaan kamera ini dapat disimpulkan ada pada jarak 40 cm hingga 150 cm. Saat program ini digunakan pada siswa yang sedang mengalami proses belajar mengajar secara *online*, program akan dapat mendeteksi wajah siswa tersebut dengan baik. Sehingga saat siswa dekat dengan kamera (di atas 40 cm) ataupun berada jauh dari kamera, hingga 150 cm, wajah masih bisa terdeteksi.

C. Pengujian Model Pada Subjek Siswa

Pengujian ini dilakukan untuk mengetahui bagaimana performa model *machine learning* saat diuji dengan wajah anak-anak. Proses ini dilakukan dengan melakukan klasifikasi pada video yang diberikan. Video tersebut akan menampilkan ke-7 emosi dengan masing-masing berdurasi 5 detik. Subjek pada video tersebut berupa siswa kelas 6 SD, gambar di bawah ini merupakan wajah dari siswa tersebut. Pada durasi video tersebut, model dari *machine learning* akan dinilai berapa akurasi dari model tersebut. Sebagai subjek dari pengujian ini terdapat Vicco Darmawan (kelas 6 SD) sebagai subjek 1, Christian Emanuelle Khoswanto (kelas 2 SMP) sebagai subjek 2, dan Christopher Valentino Khoswanto sebagai subjek 3 (kelas 5 SD).



Gambar 11. Subjek 1, Subjek 2, dan Subjek 3.

Tabel 3. Contoh Hasil

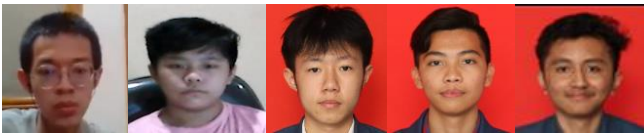
	Model A	Model B
Subjek 1	86,6%	31,4%
Subjek 2	61,6%	36,7%
Subjek 3	76,8%	28,6%

Berdasarkan Tabel 3, Model A memiliki rata-rata akurasi sebesar 86,6%, sedangkan Model B memiliki rata-rata akurasi sebesar 31,4% saat diuji pada Subjek 1. Saat diuji pada Subjek 2, Model A hanya mencapai akurasi sebesar 61,1% dan Model B sebesar 36,7%. Namun saat diuji pada Subjek 3, Model A dapat mencapai rata-rata akurasi sebesar 76,8% dan Model B mencapai 28,6%. Dari nilai rata-rata akurasi tersebut, Model A jauh lebih baik daripada Model B saat diuji secara *real*.

Model A memiliki kelemahan dalam mendeteksi ekspresi wajah sedih, jijik dan marah. Namun bila dilihat pada emosi dominan yang dapat dideteksi, sebagian masih dapat memberikan hasil yang benar. Pada Model B hanya dapat mendeteksi emosi netral, senang dan kaget. Model B tidak dapat mendeteksi ekspresi wajah marah, sedih, takut, dan jijik dengan akurasi 0%. Hal ini dapat dikarenakan Model B terlalu *overfit* dengan *dataset* sehingga memiliki ketidakmampuan saat mengklasifikasikan wajah di luar *dataset*. Berdasarkan hasil pengujian tersebut aplikasi desktop akan menggunakan Model A karena performanya yang lebih baik. Kedua model tersebut tidak dapat mendeteksi ekspresi wajah jijik dengan baik.

D. Pengujian Analisa Keseluruhan Sistem

Pengujian ini bertujuan untuk mengetahui apakah semua komponen dari sistem tugas akhir ini dapat digunakan saat pembelajaran *online*. Pada pengujian ini hanya menguji apakah aplikasi dapat mendeteksi wajah, mengklasifikasikan wajah, dan mengirimkan hasil pembacaan pada server. Selain itu juga akan menguji apakah website dapat menampilkan hasil data pembacaan tersebut. Pada pengujian ini program pada tugas akhir ini juga dijalankan bersamaan dengan *video conference* seperti zoom. Pengujian ini tidak berkaitan dengan akurasi pembacaan emosi beserta kemampuan pendeteksian wajah. Metode pengujian ini adalah menjalankan aplikasi desktop dan *check in* selama 30 menit. Hal ini disimulasikan bahwa subjek akan mengikuti kelas *online* selama 30 menit di depan komputer atau laptop. Bila data dari emosi dan kehadiran telah muncul pada *dashboard* guru, maka dapat dikatakan bahwa sistem ini telah berhasil. Pengujian ini dilakukan pada laptop pribadi milik subjek. Subjek dari pengujian ini dapat dilihat pada Gambar 12. Hasil tampilan dashboard akan dimasukkan dalam lampiran 4. Pada penelitian ini terdapat Bonaventura Jontahan T sebagai subjek A, VIcco Darmawan sebagai subjek B, Ivan Sanjaya sebagai subjek C, Lido Sanmartya sebagai subjek D, dan Miguel L. J. Luhulima sebagai subjek E.



Gambar 12. Subjek Percobaan Keseluruhan Sistem.

Tabel 4. Hasil Pengujian Keseluruhan Sistem

Subjek	Keterangan
Subjek A	Berhasil
Subjek B	Berhasil
Subjek C	Berhasil
Subjek D	Berhasil
Subjek E	Berhasil

Berdasarkan hasil pengujian pada Tabel 4, sistem ini berhasil membaca ekspresi wajah dan *face detection* dari setiap subjek. Aplikasi desktop juga dapat berjalan dengan baik pada berbagai variasi laptop atau komputer. Dengan demikian sistem ini dapat diimplementasikan saat proses belajar mengajar secara *real*. Sebagai hasil pada *dashboard* guru juga dapat digunakan oleh guru sebagai bahan evaluasi saat pembelajaran.

IV. KESIMPULAN

Dari penelitian sisten untuk menganalisa engagement siswa pada proses pembelajaran daring terdapat beberapa kesimpulan:

1. Sistem untuk mendeteksi wajah memiliki kekurangan saat wajah berada dekat pada kamera (<40 cm). Hal ini dapat dikarenakan program tidak dapat membaca fitur-fitur yang terdapat pada wajah.
2. Sistem ini juga telah dapat mengklasifikasikan emosi siswa berdasarkan ekspresi wajah siswa dengan akurasi sebesar 74,8% persen saat diuji pada siswa dengan menggunakan Model A.
3. Aplikasi desktop dapat berfungsi untuk mendeteksi wajah, mengklasifikasikan emosi siswa berdasarkan ekspresi

wajah, serta mengirimkan data hasil pembacaan pada *server*. Selain itu juga aplikasi desktop juga dapat digunakan untuk mengganti kamera dan melakukan *check in* pada kelas dan jadwal yang sesuai. Aplikasi ini juga dapat berjalan saat kelas bersamaan dengan aplikasi *video conference*.

4. Website dapat digunakan untuk menampilkan data *face detection* dan emosi siswa yang terdeteksi. Tampilan *dashboard* dapat digunakan guru sebagai bahan evaluasi pembelajaran

DAFTAR PUSTAKA

- [1] K. Kelly, "Results from Top Hat's COVID-19 Student Survey about Online Learning," May 01, 2020. <https://philonedtech.com/results-from-top-hats-covid-19-student-survey-about-online-learning/> (accessed Nov. 26, 2021).
- [2] L. A. Schindler, G. J. Burkholder, O. A. Morad, and C. Marsh, "Computer-based technology and student engagement: a critical review of the literature," *International Journal of Educational Technology in Higher Education*, vol. 14, no. 1, p. 25, Dec. 2017, doi: 10.1186/s41239-017-0063-0.
- [3] K. Cherry, "The 6 Types of Basic Emotions," *Verywell Mind*, Apr. 05, 2021. <https://www.verywellmind.com/an-overview-of-the-types-of-emotions-4163976> (accessed Nov. 15, 2021).
- [4] J. A. Fredricks, P. C. Blumenfeld, and A. H. Paris, "School Engagement: Potential of the Concept, State of the Evidence," *Review of Educational Research*, vol. 74, no. 1, pp. 59–109, Mar. 2004, doi: 10.3102/00346543074001059.
- [5] M. Hu and H. Li, "Student Engagement in Online Learning: A Review," in *2017 International Symposium on Educational Technology (ISET)*, Jun. 2017, pp. 39–43. doi: 10.1109/ISET.2017.17.
- [6] W. B. Putra and F. Arifin, "Real-Time Emotion Recognition System to Monitor Student's Mood in a Classroom," *Journal of Physics: Conference Series*, vol. 1413, no. 1, p. 012021, Nov. 2019, doi: 10.1088/1742-6596/1413/1/012021.
- [7] L. Ellen, "Projects Student Engagement Detection System in E-Learning Environment using OpenCV and CNN." <https://github.com/liviaellen/engagementdetector> (accessed Jun. 02, 2022).
- [8] S. Srivastava, "Face Emotion Recognition Using Tensorflow." <https://www.kaggle.com/code/sankalpsrivastava26/face-emotion-recognition-using-tensorflow> (accessed Jun. 03, 2022).