

**APLIKASI PENDETEKSI MASKER DENGAN METODE
FACE RECOGNITION PADA KARYAWAN
(STUDI KASUS PT KADETECH MEDIA NASIONAL)**

Diajukan Untuk Memenuhi Salah Satu Syarat Dalam Menyelesaikan
Program Srata (S-1) Program Studi Teknik Informatikan STMIK Dharma Negara Bandung



Oleh :

ALDI ABDU MALIK

NIM : 182101039

**JURUSAN TEKNIK IFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
STMIK DHARMA NEGARA BANDUNG
2020**

LEMBAR PENGESAHAN PROPOSAL SKRIPSI
APLIKASI PENDETEKSI MASKER DENGAN METODE FACE
RECOGNITION PADA KARYAWAN
(STUDI KASUS PT KADETECH MEDIA NASIONAL)

Disusun oleh :
ALDI ABDU MALIK
NIM : 182101039

Bandung, 25 Agustus 2020

Pembimbing

Ketua Jurusan Teknik Informatika

Basar M Hutauruk, S.T., M.M.

NIDN. 0403037203

Iwan Ridwan, S.T., M.Kom.

NIDN. 0422097407

Mengetahui
Ketua STMIK Dharma Negara Bandung

Dede Sulaeman, S.T., M.Kom.

NIDN. 0405107505

ABSTRAK

Aldi Abdu Malik (182101039), Aplikasi Pendeteksi Masker Dengan Metode *Face Recognition* Pada Karyawan di PT Kadetech Media Nasional.

Pada masa *New Normal*, banyak bidang usaha mulai membuka usahanya kembali namun dengan syarat menjaga aturan protokol kesehatan dalam upaya pencegahan penularan *Covid-19*. Aturan tersebut diantaranya seperti cuci tangan dengan sabun, selalu menggunakan masker dan selalu berjaga jarak dengan batas minimal lebih dari 1 meter. Begitupun dengan perusahaan yang mewajibkan karyawannya menggunakan masker. Namun dengan diberlakukannya *New Normal*, PT Kadetech Media Nasional sudah mulai beroperasi lagi. Namun kadang ada saja karyawan yang masih melalaikan kewajiban mematuhi aturan protokol kesehatan yaitu penggunaan masker salah satunya. Dan ditambah juga belum adanya bagian keamanan untuk melakukan pengecekan sebelum masuk ke dalam kantor sehingga karyawan tidak tersterilisasi sebelum masuk kedalam kantor. Sehingga penulis memiliki sebuah konsep untuk menerapkan sistem IT pada masa pandemi seperti ini dengan memanfaatkan *Face Recognition* sebagai pendeteksi penggunaan masker dan *IoT* sebagai penerapan simulasinya.

Kata kunci : *Face recognition, Skripsi Covid-19, IoT*

ABSTRACT

Aldi Abdu Malik (182101039), Mask detection application with face recognition method for employees at PT Kadetech Media Nasional.

During the New Normal period, many business sectors began to reopen their businesses, but with the requirement to maintain the rules of the health protocol in an effort to prevent Covid-19 transmission. The rules include washing hands with soap, always wearing a mask and always keeping a distance with a minimum limit of more than 1 meter. Likewise with companies that require employees to use masks. But with the enactment of New Normal, PT Kadetech Media Nasional has begun to operate again. But sometimes there are still employees who still neglect the obligation to comply with the rules of the health protocol, namely the use of masks. And also added to the lack of security to check before entering the office so that employees are not sterilized before entering the office. So that the writer has a concept for implementing IT systems during a pandemic like this by utilizing Face Recognition as a detector to use masks and IoT as the application of its simulation.

Keywords: Face recognition, Skripsi Covid-19, IoT

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah mencurahkan limpahkan nikmat, rahmat, hidayah dan inayah-Nya, sehingga penulis diberikan kemudahan dan kelancaran dalam menyelesaikan Skripsi dengan judul **“APLIKASI PENDETEKSI MASKER DENGAN METODE FACE RECOGNITION PADA KARYAWAN (STUDI KASUS PT KADETECH MEDIA NASIONAL)”**.

Dalam penyusunan Laporan Skripsi ini penulis tidak jarang selalu mendapatkan hambatan, tantangan dan kesulitan, Penulis juga sadar sepenuhnya bahwa tanpa bantuan, bimbingan, petunjuk serta dorongan dari berbagai pihak, Maka pada kesempatan ini penulis mengucapkan banyak terima kasih yang sebesar-besarnya kepada :

1. Allah SWT dan Nabi Muhammad SAW untuk semua kemudahan dan pertolongan-Nya sehingga Skripsi ini dapat terselesaikan tepat pada waktunya.
2. Kedua orang tua tercinta, Bapak dan Ibu mengucapkan beribu terimakasih atas kasih sayang, pengorbanan yang tulus, bimbingan serta didikan yang baik dan doa yang tidak pernah putus disetiap sujudnya. Dan adik, semua keluarga serta teman-teman yang telah memberikan doa dan restu, serta dukungan dan dorongannya.
3. Bapak Basar Hutaeruk, S.T., M.M. sebagai dosen pembimbing yang telah memberikan arahan, bimbingan, petunjuk dan nasehat dalam proses pembuatan Skripsi sampai dengan selesai.
4. Bapak Dede Sulaeman, S.T., M.Kom. selaku ketua STMIK Dharma Negara.
5. Bapak Iwan Ridwan, S.T., M.kom. selaku ketua jurusan Teknik Informatika
6. Seluruh Staff dan Dosen pengajar Telecom Professional Development Center yang telah memberikan bekal dan ilmu yang bermanfaat.
7. Kepada tunangan tercinta, Yusmita Dwi Anjani terima kasih banyak atas doa dan selalu memberikan dukungan moral maupun moril.

8. Serta semua pihak yang tidak bisa disebutkan satu persatu yang telah banyak membantu penulis menyelesaikan skripsi ini,

Penulis menyadari bahwa penulisan skripsi ini masih jauh dari kata sempurna, dan masih banyaknya kekurangan pada penyelesaian skripsi ini yang disebabkan oleh adanya keterbatasan data dan kemampuan yang dimiliki oleh penulis. Dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun guna menyempurnakan penyelesaian penulisan pada skripsi ini serta guna mencapai kesempurnaan penulis dimasa yang akan datang. Akhir kata semoga tugas akhir ini dapat berguna bagi penulis khususnya dan bagi pembaca pada umumnya.

Bandung, 25 Agustus 2020

Aldi Abdu Malik

NIM 182101039

DAFTAR ISI

ABSTRAK	ii
ABSTRACT	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x

BAB I PENDAHULUAN

1.1 Latar Belakang	I-1
1.2 Identifikasi Masalah	I-2
1.3 Batasan Masalah.....	I-2
1.4 Tujuan Penelitian	I-3
1.5 Manfaat Penelitian	I-3
1.6 Metodologi Penelitian	I-4
1.6.1 Metode Pengembangan Perangkat Lunak	I-4
1.6.2 Metode Analisis	I-5
1.6.2.1 Metode <i>Object Oriented Analysis and Design</i> (OOAD)	I-5
1.6.2.2 OOA (<i>Object oriented analysis</i>).....	I-5
1.6.2.3 OOD (<i>Object Oriented Design</i>).....	I-6
1.6.3 Metode Pengumpulan Data	I-6

BAB II LANDASAN TEORI

2.1 Aplikasi	II-1
2.2 <i>Face Recognition</i>	II-1
2.3 <i>Internet of Things</i> (IoT).....	II-2
2.4 <i>NodeMCU</i>	II-2
2.5 <i>NodeJS</i>	II-3
2.6 <i>Web Service</i>	II-3

2.7 <i>Application Programming Interface (API)</i>	II-4
2.8 <i>Arduino IDE (Intergrated Development Environment)</i>	II-4
2.9 <i>UML (Unified Modeling Language)</i>	II-5

BAB III OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian	III-1
3.2 Sejarah Institusi	III-1
3.2.1 Visi dan Misi Institusi	III-2
3.2.2 Struktur Organisasi dan Fungsi	III-3
3.3 Analisa Kebutuhan	
3.3.1 Kebutuhan Perangkat Keras (Hardware)	III-5
3.3.2 Kebutuhan Perangkat Lunak (Software)	III-5
3.4 Metode Penelitian.....	III-6
3.4.1 Metode Pengumpulan Data	III-6
3.4.2 Metode Perancangan Sistem	III-7

BAB IV ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem.....	IV-1
4.2 Perancangan Sistem	IV-1
4.2.1 Fase Analisis	IV-1
4.2.1.1 Analisis Sistem Berjalan	IV-1
4.2.1.2 Aktivitas Sistem	IV-2
4.2.1.2.1 <i>Description Actor</i>	IV-2
4.2.1.2.2 <i>Tabel Description Use Case</i>	IV-2
4.2.1.2.3 <i>Use Case Diagram</i>	IV-3
4.2.1.2.4 <i>Skenario Use Case Diagram</i>	IV-3
4.2.1.3 <i>Object Interaction (Sequence Diagram)</i>	IV-5
4.2.1.4 <i>Object Behavior (Activity Diagram)</i>	IV-6
4.2.2 Fase Desain	IV-6
4.2.2.1 Perancangan Antarmuka (<i>Interface</i>)	IV-7

BAB V IMPLEMENTASI DAN PENGUJIAN

5.1 Spesifikasi Kebutuhan Implementasi Sistem

5.1.1 Spesifikasi Perangkat Keras (*Hardware*)..... V-1

5.1.2 Spesifikasi Perangkat Lunak (*Software*) V-2

5.2 Implementasi Antarmuka Sistem V-2

5.3 Pengujian Perangkat Lunak..... V-4

5.3.1 Black Box Testing..... V-5

5.3.2 White Box Testing V-5

BAB VI PENUTUP

6.1 Kesimpulan VI-1

6.2 Saran..... VI-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 3.1 Struktur Organisasi PT. Kadetech Media Nasional	III-3
Gambar 4.1 <i>Use case diagram</i>	IV-3
Gambar 4.2 <i>Sequence Diagram</i> Skenario sistem	IV-5
Gambar 4.3 <i>Activity Diagram</i> Skenario sistem	IV-6
Gambar 4.4 Perancangan halaman pendeteksi masker	IV-7
Gambar 4.5 Perancangan <i>Hardware</i> Simulasi Sistem	IV-8
Gambar 5.1 Antarmuka Halaman Pendeteksi Masker	V-3
Gambar 5.2 Antarmuka Simulator Palang Pintu.....	V-4
Gambar 5.3 <i>Flow Graph</i>	V-7

DAFTAR TABEL

Tabel 2.1 Simbol <i>Use Case Diagram</i>	II-5
Tabel 2.2 Simbol <i>Class Diagram</i>	II-7
Tabel 2.3 Simbol <i>Activity Diagram</i>	II-8
Tabel 2.4 Simbol <i>Sequence Diagram</i>	II-9
Tabel 3.1 Spesifikasi Perangkat Keras (<i>Hardware</i>).....	III-5
Tabel 3.2 Spesifikasi Perangkat Lunak (<i>Software</i>)	III-5
Tabel 4.1 Tabel <i>Description Use Case</i>	IV-2
Tabel 4.2 Skenario <i>Use Case</i> Menunjukkan Wajah.....	IV-3
Tabel 4.3 Skenario <i>Use Case</i> Identifikasi Wajah.....	IV-4
Tabel 4.4 Skenario <i>Use Case</i> Pintu Otomatis	IV-5
Tabel 5.1 Spesifikasi Perangkat Keras (<i>Hardware</i>).....	V-1
Tabel 5.2 Spesifikasi Perangkat Lunak (<i>Software</i>)	V-2
Tabel 5.3 <i>Black Box Testing</i>	V-3
Tabel 5.4 <i>Independen Path</i>	V-7

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada masa pandemi Covid-19 seperti sekarang ini, banyak bidang seperti bidang Pendidikan, bidang usaha dan bidang-bidang lain terkena imbasnya. Sehingga pemerintah mencari solusi untuk menghadapi masalah pandemi ini, diantaranya dengan meberlakukan PSBB (Pembatasan Sosial Bersekala Besar) dan *New Normal*.

Pada masa *New Normal*, banyak bidang usaha mulai membuka usahanya kembali namun dengan syarat menjaga aturan protokol kesehatan dalam upaya pencegahan penularan *Covid-19*. Aturan tersebut diantaranya seperti cuci tangan dengan sabun, selalu menggunakan masker dan selalu berjaga jarak dengan batas minimal lebih dari 1 meter. Begitupun dengan perusahaan yang mewajibkan karyawannya menggunakan masker.

PT Kadetech Media Nasional misalnya, sempat terkena imbas pandemi. Dengan terpaksa harus meliburkan beberapa karyawannya. Namun dengan diberlakukannya *New Normal*, PT Kadetech Media Nasional sudah mulai beroperasi lagi. Namun kadang ada saja karyawan yang masih melalaikan kewajiban mematuhi aturan protokol kesehatan yaitu penggunaan masker salah satunya. Dan ditambah juga belum adanya bagian keamanan untuk melakukan pengecekan sebelum masuk ke dalam kantor sehingga karyawan tidak tersterilisasi sebelum masuk kedalam kantor. Sehingga penulis memiliki sebuah konsep untuk menerapkan sistem IT pada

masa pandemi seperti ini dengan memanfaatkan *Face Recognition* sebagai pendeteksi penggunaan masker dan *IoT* sebagai penerapan simulasinya.

Penerapan *Face recognition* untuk mendeteksi penggunaan masker pada aplikasi ini diharapkan dapat bermanfaat pada masa pandemi seperti sekarang ini.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang penulis sudah uraikan diatas dapat disimpulkan identifikasi masalahnya yaitu :

1. Bagaimana orang atau khususnya karyawan agar untuk selalu mematuhi protokol kesehatan dengan menggunakan masker salah satunya?
2. Bagaimana melakukan pengecekan penggunaan masker pada karyawan tanpa harus menegur langsung?

1.3 Batasan Masalah

Permasalahan diatas mempunyai batasan masalah sebagai berikut :

1. Aplikasi dibuat dengan Bahasa pemrograman Node JS (Server) dan Arduino (*Robotic*).
2. Penerapan Face recognition dengan menggunakan simulasi palang pintu.
3. *Hardware* yang digunakan dalam simulasi adalah NodeMCU sebagai *mikrokontroler*.

1.4 Tujuan Penelitian

Tujuan utama dari skripsi ini adalah :

1. Membuat aplikasi untuk mendeteksi penggunaan masker pada karyawan.
2. Memastikan karyawan sudah menggunakan masker untuk mencegah bahaya penularan *Covid-19*.
3. Melihat seberapa berpengaruhnya bidang IT dalam menghadapi masa pandemi.

1.5 Manfaat Penelitian

Manfaat dari hasil penelitian ini adalah :

1. Manfaat untuk penulis

Sebagai salah satu syarat untuk kelulusan Program S1 (SI) Program Studi Teknik Informatika di Sekolah Tinggi Manajemen Informatika dan Komputer Dharma Negara Bandung.

2. Manfaat untuk objek penelitian

Memberikan konsep dan simulasi penggunaan Face recognition dalam upaya pencegahan penularan *Covid-19*.

3. Manfaat untuk pembaca

Memberikan pemahaman sehingga dapat di pelajari dan di *development* ulang di kemudian hari.

1.6 Metodologi Penelitian

Metodologi penelitian berisi langkah-langkah yang digunakan dalam penelitian ini agar terstruktur dengan baik. Dengan sistematika ini proses penelitian yang dilakukan oleh penulis dapat dipahami dan diikuti oleh pihak lain. Penelitian yang dilakukan oleh penulis diperoleh dari pengamatan data-data yang ada. Adapun metodologi penelitian yang digunakan dalam penyusunan skripsi ini adalah sebagai berikut :

1.6.1 Metode Pengembangan Perangkat Lunak

Aplikasi ini dibuat dengan menggunakan metode model air terjun (*waterfall*) dalam pelaksanaannya SDLC waterfall dibagi dalam empat tahap yaitu:

1. Analisis

Penulis menganalisa secara langsung hal-hal yang dibutuhkan dalam pembuatan aplikasi ini.

2. Desain

Penulis mendesain terlebih dahulu baik alur maupun antarmuka aplikasi sebelum di implementasikan menjadi sebuah program.

3. Pembuatan kode program (*Development*)

Penulis membuat kode program dari hasil desain yang sudah penulis buat lalu ditranslasikan ke dalam bentuk program

4. Pengujian

Penulis melakukan test/uji coba program yang sudah dibuat, untuk memastikan bahwa semua bagian sudah diuji, untuk meminimalisir kesalahan (*error*), dan untuk memberikan hasil yang sesuai dengan yang diinginkan

5. Peluncuran dan pemeliharaan (*Maintenance*)

Setelah melalui tahap pengujian dan meminimalisir kesalahan (*error*), penulis melakukan pemeliharaan baik dari sisi *software* maupun *hardware*.

1.6.2 Metode Analisis

Adapun metode yang digunakan penulis untuk analisa serta perancangan perangkat lunak adalah metode *Object Oriented Analysis Design* (OOAD).

1.6.2.1 Metode Object Oriented Analysis and Design (OOAD)

OOAD adalah metode pengembangan sistem yang lebih menekankan pada objek dibandingkan dengan data atau proses. Dalam tahapannya OOAD terbagi menjadi dua yaitu OOA (*Object oriented analysis*) dan OOD (*Object Oriented Design*).

1.6.2.1.1 OOA (*Object oriented analysis*)

Object oriented analysis (OOA) merupakan metode analisis yang memeriksa *requirements* (syarat/ keperluan yang harus dipenuhi oleh sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan. OOA mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut dengan menggunakan metode berorientasi

objek. Biasanya analisa sistem dimulai dengan adanya dokumen permintaan yang diperoleh dari semua pihak yang berkepentingan. Analisa ini sebaiknya dilakukan oleh orang-orang yang benar-benar memahami implementasi sistem yang berbasis atau berorientasi objek, karena tanpa pemahaman itu maka sistem yang dihasilkan bisa jadi tidak realistis jika di implementasikan dengan berbasis objek.

1.6.2.1.2 OOD (*Object Oriented Design*)

Object Oriented Design (OOD) merupakan metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem. OOD adalah sebuah metode mendesain yang mencakup proses pendekomposisi objek dan digambarkan dalam notasi sehingga bisa menggambarkan *static (class diagram)* dan *dynamic (statechart diagram)* model sistem.

1.6.3 Metode Pengumpulan Data

Metode pengumpulan data berupa suatu sifat, keadaan, kegiatan tertentu dan sejenisnya. Pengumpulan data dilakukan untuk mendapatkan suatu informasi yang dibutuhkan dalam mencapai tujuan penelitian.

Dalam pengumpulan data dalam penulisan skripsi ini, penulis menggunakan metode pengumpulan data sebagai berikut :

1. Metode Observasi

Penulis mengamati langsung kegiatan yang dilakukan oleh karyawan PT Kadetech Media Nasional.

2. Metode Wawancara

Penulis melakukan wawancara secara langsung dengan direktur dan beberapa karyawan untuk memperoleh informasi yang berhubungan dengan masalah sedang diteliti oleh penulis.

3. Studi Pustaka

Penulis mencari literatur dan buku-buku yang berkaitan dengan tema penulisan, yang digunakan untuk mendukung materi-materi yang dibutuhkan dalam penyusunan tugas skripsi ini.

1.7 Sistematika Penulisan

Terdapat ringkasan-ringkasan yang terdiri dari BAB I sampai BAB VI pada sistematika penulisan ini, dimana tiap-tiap BAB memberikan gambaran secara langsung. Berikut sistematika penulisan dari laporan skripsi yang dibuat :

BAB I PENDAHULUAN

Pada bab ini diuraikan mengenai pokok-pokok pemikiran yang medasari skripsi ini, antara lain adalah latar belakang, identifikasi masalah, batasan masalah, tujuan dan manfaat penelitian serta metodologi penelitian.

BAB II LANDASAN TEORI

Pada bab ini akan dibahas informasi umum tentang objek penelitian dan dasar-dasar teori yang berkaitan dengan skripsi ini.

BAB III OBJEK DAN METODE PENELITIAN

Pada bab ini akan dibahas mengenai gambaran umum tentang objek penelitian, teknik penelitian, teknik pengumpulan data dan teknik analisis data.

BAB IV ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai analisa sistem dan perancangan sistem beserta data-data yang diperoleh.

BAB V IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dibahas mengenai tahapan implementasi sistem yang dirancang sebagai tahapan lanjutan dari analisa dan perancangan.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari semua yang sudah dilakukan apakah sudah sesuai dengan tujuan dibuatnya sistem ini atau tidak. Saran merupakan opini dari penulis yang bertujuan untuk melengkapi dan menyempurnakan dari apa yang sudah dikerjakan.

BAB II

LANDASAN TEORI

2.1 Aplikasi

Aplikasi menurut (Dhanta dikutip dari Sanjaya, 2015) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, *Microsoft Excel*. Aplikasi berasal dari kata *application* yang artinya penerapan lamaran penggunaan. Menurut (Jogiyanto dikutip oleh Ramzi, 2013) aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, dan pekerjaan itu sendiri.

2.2 Face Recognition

Identifikasi (pengenalan) wajah atau *face recognition* adalah sebuah tugas yang dikerjakan oleh manusia secara rutin dan mudah dalam kehidupan sehari-hari. Penelitian dan pengembangan ilmu pengenalan wajah berkembang secara otomatis atas dasar ketersediaan desktop kuat dan rendah biaya serta *embedded system* yang telah menciptakan minat yang sangat besar dalam pengolahan citra digital dan video. Motivasi penelitian dan pengembangan dari pengenalan wajah termasuk dalam lingkup otentikasi *biometric*, pengawasan, interaksi manusia komputer, dan manajemen multimedia (Li dan Jain, 2005:1).

Sistem *face recognition* pada umumnya mencakup empat modul utama menurut (Li dan Jain, 2005:2), yaitu: deteksi, *alignment*, ekstraksi fitur dan pencocokan. Proses lokalisasi dan normalisasi (deteksi wajah dan *alignment*) adalah langkah-langkah sebelum proses pengenalan wajah (ekstraksi fitur wajah dan pencocokan) dilakukan.

2.3 Internet of Things (IoT)

IoT adalah sebuah konsep yang menggunakan internet untuk menjadi sarana segala aktifitas yang pelakunya saling berinteraksi. *IoT* mengacu pada miliaran perangkat yang saling terhubung atau bisa disebut dengan “Objek Cerdas” atau “Smart Things” (Cirani et al., 2015). Dengan adanya *IoT* segala kegiatan dan aktifitas dimudahkan melalui *online* dan lebih efisien (Sulaiman dan Widarma 2017). *IoT* merupakan inti dari industri teknologi informasi generasi baru.

Dampak *IoT* pada evolusi internet menuju lingkungan cerdas generasi berikutnya yang sangat bergantung pada integrasi *IoT* dengan *cloud computing*. Saat *IoT* terhubung dengan *cloud* sejumlah data besar yang telah dikumpulkan dari banyak tempat, dapat diolah dan dianalisis untuk membuat makna informasi ke *end-user* (Barcelo et al., 2016).

Penelitian ini akan memanfaatkan *IoT* sebagai alat simulasi yang terhubung dengan pendeteksi masker pada wajah menggunakan metode *Face recognition*.

2.4 NodeMCU

NodeMCU adalah sebuah platform *IoT* yang bersifat *opensource*. Terdiri dari perangkat keras berupa *System OnChip ESP8266*. dari *ESP8266* buatan *Espressif*

System, juga firmware yang digunakan, yang menggunakan Bahasa pemrograman scripting Lua. Menurut (Sumardi, 2016) Istilah NodeMCU secara *default* sebenarnya mengacu pada *firmware* yang digunakan dari pada perangkat keras *development kit NodeMCU* bisa dianalogikan sebagai *board* Arduino-nya *ESP8266*.

2.5 NodeJS

Nodejs dikembangkan dari *engine* *javaScript* yang dibuat oleh Google untuk *Browser* *Chrome* / *Chromium* (*V8*) ditambah dengan *lib* *UV* serta beberapa 6 pustaka internal lainnya. Dengan menggunakan *Nodejs* semua pengembangan akan dilakukan dengan *javascript*, baik pada sisi *client* maupun *server*. Pengembangan aplikasi dengan menggunakan *Nodejs* dapat dilakukan secara moduler yaitu dengan memisahkan berbagai komponen kedalam pustaka (*library*). Pustaka tersebut dapat dikelola dengan *npm* yang terdapat di *Nodejs*. Pada dasarnya, *Nodejs* sebuah *runtime environment* dan *script library*. Sebuah *runtime environment* adalah sebuah *software* yang berfungsi untuk mengeksekusi, menjalankan dan mengimplementasikan fungsi-fungsi serta cara kerja inti dari suatu bahasa pemrograman. Sedangkan *script library* adalah kumpulan, kompilasi atau bank data berisi skrip/kode-kode pemrograman. (Equan Pr, 2013).

2.6 Web Service

Web service adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi yang bisa beroperasi *machine to machine* diatas jaringan. *Web service*

mempunyai alat penghubung yang diuraikan di dalam format *machine-processable* (secara spesifik WSDL). Sistem lain saling berhubungan dengan *Web service* di dalam cara yang ditentukan oleh deskripsinya menggunakan pesan SOAP REST yang secara khas disampaikan menggunakan HTTP dengan serialisasi XML atau JSON bersama dengan standar lain yang terkait dengan *web* (Booth et al., 2004).

2.7 Application Programming Interface (API)

API adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API terdiri dari berbagai elemen seperti *function*, *protocols*, dan *tools* lainnya yang memungkinkan *developers* untuk membuat aplikasi. Menurut (Abdul Kadir, 2016) Tujuan penggunaan API adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur yang serupa. Penerapan API akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Terdapat berbagai jenis sistem API yang dapat digunakan, termasuk *system* operasi, *library*, dan web.

2.8 Arduino IDE (Intergrated Development Environment)

IDE merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Menurut (Muhammad Syahwil, 2015) Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk

melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman *Arduino* (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC *mikrokontroler Arduino* telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengahantara *compiler Arduino* dengan *mikrokontroler*.

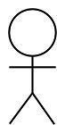
2.9 UML (*Unified Modeling Language*)









Menurut (Sri Dharwiyanti, 2003) dalam bukunya “Pengantar *Unified Modeling Language*” menuliskan bahwa UML adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

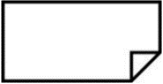
UML mendefinisikan diagram-diagram sebagai berikut :

1. *Use Case Diagram*

Tabel 2.1 Simbol *Use Case Diagram*





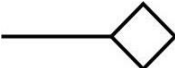
Gambar	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>Use Case</i> .

	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	Generalization	Hubungan dimana objek anak berbagi perilaku dan stuktur data dari objek yang ada diatasnya objek induk.
	Include	Menspesifikasikan bahwa Use Case sumber secara eksplisit.
	Extend	Menspesifikasikan bahwa Use Case target memperluas perilaku dari Use Case sumber dari suatu titik yang diberikan.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	Use Case	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terstruktur bagi suatu aktor.
	Collaboration	Interaksi aturan – aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen






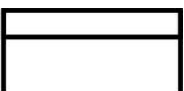


		– elemennya.
	Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. Class Diagram




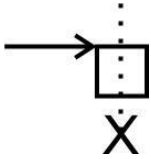
Tabel 2.2 Simbol *Class Diagram*

Gambar	Nama	Keterangan
	Association	Realisasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
	Direct Association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
	Generalization	Relasi antar kelas dengan makna generalisasi – spesialisasi.
	Dependency	Relasi antar kelas dengan makna kebergantungan antar kelas.
	Aggregation	Relasi antar kelas dengan makna semua – bagian.

3. *Activity Diagram*Tabel 2.3 Simbol *Activity Diagram*

Gambar	Nama	Keterangan
	Start	Status awal aktivitas sistem, sebuah <i>Activity Diagram</i> mempunyai sebuah status awal.
	Activity	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	End	Status akhir yang dilakukan sistem, sebuah <i>Activity Diagram</i> memiliki sebuah status akhir.
	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
	Fork	Digunakan untuk menunjukan Kegiatan yang dilakukan secara paralel.
	Join	Digunakan untuk menunjukan kegiatan yang digabungkan.

4. *Sequence Diagram*Tabel 2.4 Simbol *Sequence Diagram*

Gambar	Nama	Keterangan
	Actor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem.
	Object	Berpartisipasi secara berurutan dengan mengirimkan dan atau menerima pesan.
	Life Line	Menandakan kehidupan objek selama urutan, diakhiri tanda x pada titik dimana kelas tidak lagi berinteraksi
	Destroy	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri.

BAB III

OBJEK DAN METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian dalam skripsi ini adalah menguji bagaimana pengaruh teknologi dalam menghadapi masa pandemi *Covid-19* khususnya dalam meningkatkan upaya pencegahan dengan cara mematuhi aturan protokol kesehatan yang ada diantaranya yaitu menggunakan masker.

Penelitian ini dilakukan pada karyawan di PT Kadetech Media Nasional yang bertempat di Babakan Inggun Desa Duren, Kecamatan Klari, Karawang.

3.2 Sejarah Institusi

PT Kadetech Media Nasional merupakan perusahaan yang berfokus pada bidang jasa berupa Konsultan IT dan Multimedia yang bertempat di Babakan Inggun Desa Duren, Kecamatan Klari, Karawang.

Berdiri sejak tahun 2018 lalu, PT Kadetech Media Nasional pada saat berdirinya hanya berupa tim IT yang tidak memiliki legalitas sebagai perusahaan dengan hanya beranggotakan 4 orang. Namun dengan adanya perkembangan usaha yang cukup baik, maka pada tahun 2019 dibuatlah legalitas yang berupa Perseroan Terbatas (PT). Sampai saat ini, PT Kadetech Media Nasional sudah memiliki puluhan karyawan yang terbagi dari beberapa divisi diantaranya IT, Multimedia, Keuangan dan *Marketing*.

3.2.1 Visi dan Misi Institusi

Setiap organisasi atau perusahaan mempunyai visi dan misinya masing-masing sesuai dengan latar belakang para pemilik perusahaan dan tujuan dibentuknya perusahaan. Adapun visi dan misi dari PT. Kadetech Media Nasional yaitu :

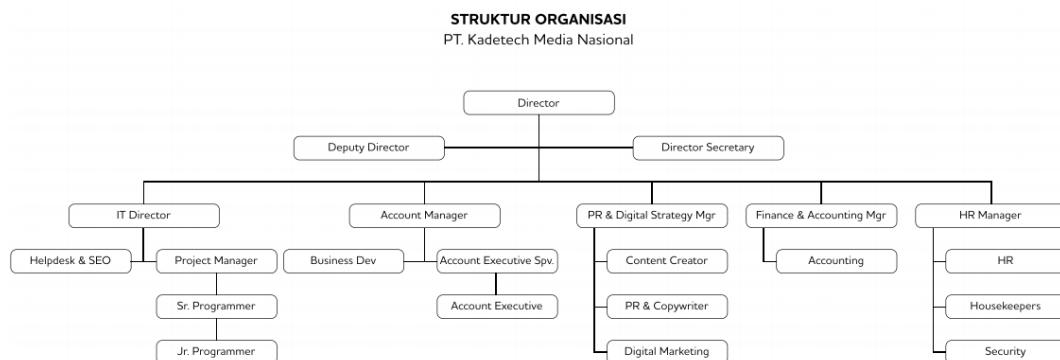
A. Visi

Menjadi perusahaan yang dapat terus berkembang dan dapat memberikan nilai kepuasan pada setiap klien yang kami tangani.

B. Misi

1. Meningkatkan kualitas sumber daya manusia yang kompeten dan berpengalaman.
2. Memberikan manfaat dan kualitas yang maksimal untuk semua partner dan klien kami.
3. Menjadi mitra terpercaya dalam memberikan dan menerapkan manfaat bagi klien kami.

3.2.2 Struktur Organisasi dan Fungsi



Gambar 3.1 Struktur Organisasi PT. Kadetech Media Nasional

Berdasarkan struktur organisasi pada gambar 3.1 diatas maka dapat diuraikan tugas dan tanggung jawab dari masing-masing divisi yang ada pada PT. Kadetech Media Nasional :

1. *Director*

Tugas utama *Director* adalah memimpin perusahaan dengan menerbitkan kebijakan-kebijakan perusahaan atau institusi. Selain itu *Director* juga memiliki wewenang dalam memilih, menetapkan, mengawasi tugas dari karyawan dan kepala bagian (manajer) atau *Deputy director*.

2. *Deputy Director*

Deputy director merupakan wakil dari *director*. *Deputy director* memiliki fungsi dan tanggung jawab yang lebih terfokus kepada internal perusahaan.

Wewenang dan tanggung jawab *Deputy director* adalah memimpin direktorat dibawahnya, baik dalam segi pengembangan, pelaksanaan dan pengendalian untuk meningkatkan kinerja perusahaan.

3. *Director Secretary*

Director secretary bertugas dalam hal mengurus pencatatan dan mengatur janji untuk *Director*.

4. *IT Director*

IT director bertanggung jawab dalam mengawasi semua fungsi TI (teknologi informatika) disuatu perusahaan.

5. *Account Manager*

Account manager bertugas untuk menyusun strategi pemasaran dan menjalin kerjasama dengan pihak klien.

6. *Public Relation & Digital Strategy Manager*

Public relation & digital strategy manager bertugas membina hubungan baik dengan publik baik internal maupun eksternal perusahaan.

7. *Finance & Accounting Manager*

Tugas utama *Finance & accounting manager* adalah bertanggung jawab untuk membantu perencanaan bisnis dan pengambilan keputusan dengan memberi nasihat keuangan yang sesuai kepada pimpinan tertinggi di perusahaan.

8. *Human Resource Manager*

Tugas utama *Human resource manager* adalah menjadi penghubung antara manajemen dan karyawan.

3.3 Analisa Kebutuhan

Dalam proses penelitian tentunya pasti ada kebutuhan perangkat keras dan perangkat lunak yang harus dipenuhi. Dalam menyelesaikan penelitian ini dibutuhkan beberapa kebutuhan diantaranya adalah :

3.3.1 Kebutuhan Perangkat Keras (*Hardware*)

Tabel 3.1 Spesifikasi Perangkat Keras (*Hardware*)

No.	Perangkat Keras	Spesifikasi
1.	Laptop	HP 14-AF115AU AMD Quad Core A6 RAM 4GB 14 Inch
2.	Mikrokontroler	NodeMCU LUA WIFI V3 4MB 32MBITS, Sensor Ultrasonik HC-SR04, Motor Servo SG90 dan Kabel jumper
3.	Kabel USB	Micro USB

3.3.2 Kebutuhan Perangkat Lunak (*Software*)

Tabel 3.2 Spesifikasi Perangkat Lunak (*Software*)

No.	Perangkat Lunak	Spesifikasi
1.	Sistem Operasi	Windows 10
2.	Web Server	Apache 2.0
3.	Code Editor	Visual Studio Code dan Arduino IDE
4.	Browser	Chrome
5.	Bahasa Pemrograman	Node JS dan C
6.	StarUML	Versi 3.0

3.4 Metode Penelitian

Metode penelitian adalah salah satu cara yang bersifat otomatis dan objektif dengan tujuan untuk memperoleh dan mengumpulkan keterangan dan informasi yang teliti secara efisien dan dapat digunakan sebagai pedoman dalam melaksanakan penelitian. Penulis telah melakukan riset guna mendapatkan data atau informasi yang akurat mengenai sistem berjalan untuk dianalisa agar penulis dapat mengetahui kelemahan atau kekurangan dengan sistem tersebut.

3.4.1 Metode Pengumpulan Data

Berikut adalah beberapa metode yang penulis gunakan guna memperoleh data diantaranya adalah :

1. Observasi

Observasi merupakan salah satu metode pengumpulan data dengan mengadakan pengamatan langsung ke lokasi penelitian. Hal yang diobservasi yaitu keadaan lapangan dan aktivitas-aktivitas yang dilakukan dan sistem sosial yang terdapat didalamnya.

2. Wawancara

Metode ini dilakukan dengan cara mengadakan tanya jawab secara langsung dengan pihak-pihak yang berhubungan dengan observasi penelitian yaitu kepala bagian dan direktur PT. Kadetech Media Nasional.

3.4.2 Metode Perancangan Sistem

Metode perancangan sistem adalah sebuah cara yang tersistem atau teratur yang bertujuan untuk melakukan analisa pengembangan suatu sistem agar sistem tersebut dapat memenuhi kebutuhan dari *user*.

1. Analisis Sistem

Analisis sistem yang penulis gunakan untuk membangun aplikasi ini yaitu dengan cara menganalisis kebutuhan system yang berguna dan berkaitan dengan masa pandemi *Covid-19*.

2. Perancangan Model

Disini penulis merancang model berupa desain tampilan aplikasi dan juga desain simulasi pintu otomatis yang akan penulis buat.

3. Perancangan *Input*

Input yang penulis gunakan disini berupa foto karyawan PT Kadetech Media Nasional yang menggunakan masker dan tidak menggunakan masker.

4. Perancangan *Output*

Penulis merancang sebuah *output* berupa simulasi pintu otomatis.

5. Perancangan Aplikasi

Pada tahap ini penulis melakukan pengkodean aplikasi dengan beberapa jenis Bahasa pemrograman yang penulis gunakan yaitu Node JS sebagai server dan C sebagai *robotic*.

6. *Testing*

Tahap *testing* dilakukan dengan menguji bagaimana kosep tujuan dengan hasil yang ada.

BAB IV

ANALISA DAN PERANCANGAN

4.1 Analisis Sistem

Analisis terhadap suatu sistem sangat diperlukan untuk mengikuti kegiatan-kegiatan yang sedang berjalan di dalam suatu sistem. Tujuan mengetahui kegiatan-kegiatan tersebut adalah untuk memahami dan mengerti alur sistem serta hambatan yang terdapat di dalam sistem tersebut. Pada tahap ini yang perlu dilakukan adalah analisis terhadap sistem yang sedang berjalan dan bagaimana aliran dokumen sistem yang berjalan.

4.2 Perancangan Sistem

Sesuai dengan metode pengembangan sistem yang sedang digunakan yaitu menggunakan model *waterfall*, maka tahapan perancangan sistem sebagai berikut :

4.2.1 Fase Analisis

Proses ini dilakukan dengan cara mengumpulkan informasi kebutuhan sistem atau perangkat lunak melalui konsultasi dengan *user*. Proses ini mendefinisikan secara rinci mengenai fungsi-fungsi, batasan dan tujuan dari perangkat lunak sebagai spesifikasi sistem yang akan dibuat.

4.2.1.1 Analisis Sistem Berjalan

Analisa sistem yang berjalan pada PT. Kadetech Media Nasional dapat dijabarkan sebagai berikut :

1. Karyawan dapat bebas masuk tanpa adanya pengecekan terlebih dahulu
2. Antar karyawan kadang segan untuk menyuruh karyawan yang lain menggunakan masker.

4.2.1.2 Aktivitas Sistem

Pada aktivitas sistem ini akan dipaparkan mengenai *description actor*, *scenario*, dan *use case diagram*.

4.2.1.2.1 Description Actor

Perangkat lunak ini hanya menggunakan 1 aktor yaitu karyawan saja.

4.2.1.2.2 Tabel Description Use Case

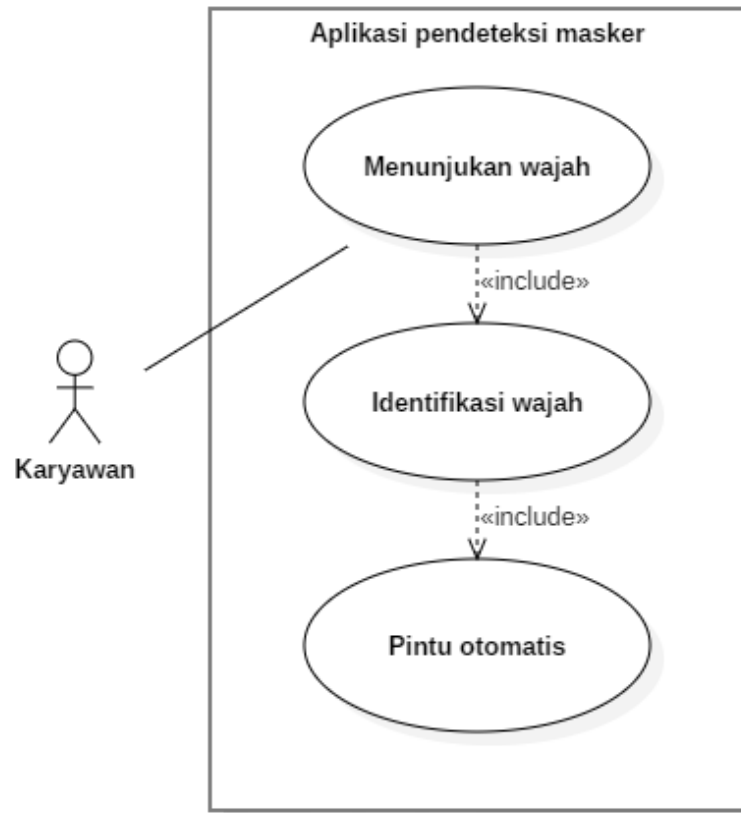
Use case yang digunakan dalam pemodelan sistem menggunakan 3 *use case* utama yaitu :

Tabel 4.1 Tabel *Description Use Case*

No	Use Case	Deskripsi
1.	Menunjukan wajah pada kamera	Merupakan hal yang harus dilakukan oleh karyawan sebelum memasuki kantor.
2.	Identifikasi wajah	Merupakan proses pengecekan terhadap karyawan untuk mengidentifikasi suatu objek pada wajah yaitu masker.
3.	Pintu otomatis	Setelah system mengidentifikasi wajah dan di temukannya suatu objek yaitu masker, maka pintu akan terbuka secara otomatis.

4.2.1.2.3 Use Case Diagram

Use case diagram menggambarkan tindakan apa saja yang dapat *actor* lakukan terhadap fungsi perangkat lunak.



Gambar 4.1 *Use case diagram*

4.2.1.2.4 Skenario Use Case Diagram

1. Menunjukkan Wajah

Tabel 4.2 Skenario *Use Case* Menunjukkan Wajah

Nama Use case diagram	Menunjukkan wajah
Deskripsi Singkat	Sebelum masuk kedalam kantor, <i>actor</i> diwajibkan untuk menunjukkan wajanya pada sistem.
Actor	Karyawan

Kondisi Sebelum	-	
Kondisi Sesudah	Sistem dapat memproses objek yang ada pada wajah yaitu masker	
Aliran Aktivitas	Actor	Sistem
	Menunjukkan wajahnya pada kamera	Mencari objek pada wajah
Penangkapan Kondisi Kesalahan	-	

2. Identifikasi Wajah

Tabel 4.3 Skenario *Use Case* Identifikasi Wajah

Nama <i>Use case</i> diagram	Identifikasi Wajah	
Deskripsi Singkat	Proses identifikasi wajah karyawan dengan melakukan pencarian objek pada wajah yaitu masker.	
Actor	-	
Kondisi Sebelum	Karyawan sudah menunjukkan wajahnya pada kamera	
Kondisi Sesudah	Sistem memberikan respons pada pintu otomatis	
Aliran Aktivitas	Actor	Sistem
	-	Menghasilkan respons wajah yang sudah diidentifikasi menggunakan masker atau tidak
Penangkapan Kondisi Kesalahan	-	

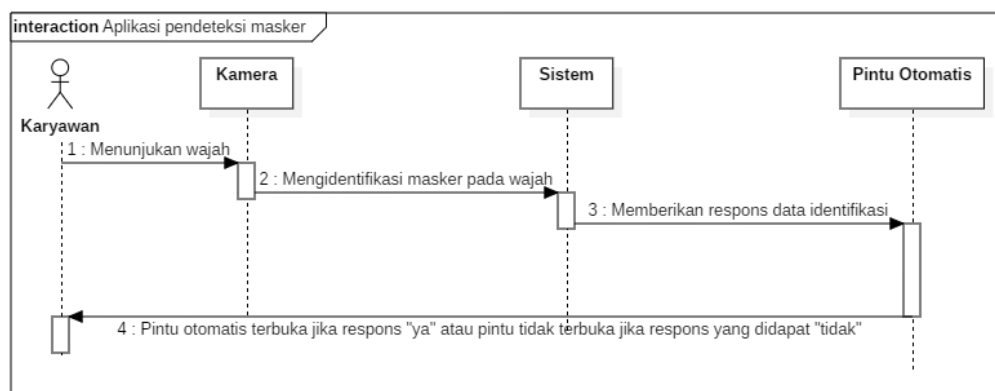
3. Pintu Otomatis

Tabel 4.4 Skenario *Use Case* Pintu Otomatis

Nama <i>Use case</i> diagram	Pintu Otomatis	
Deskripsi Singkat	Setelah sistem mengidentifikasi masker dan melakukan respons pintu otomatis akan terbuka jika tidak pintu tidak akan terbuka	
Actor	-	
Kondisi Sebelum	Sistem memberikan respons menggunakan atau tidak menggunakan masker	
Kondisi Sesudah	Pintu akan otomatis terbuka	
Aliran Aktivitas	Actor	Sistem
	-	Pintu otomatis terbuka jika respons yang didapat itu menggunakan masker
Penangkapan Kondisi Kesalahan	Pintu tidak mau terbuka jika respons yang didapat dari sistem bahwa karyawan tidak menggunakan masker	

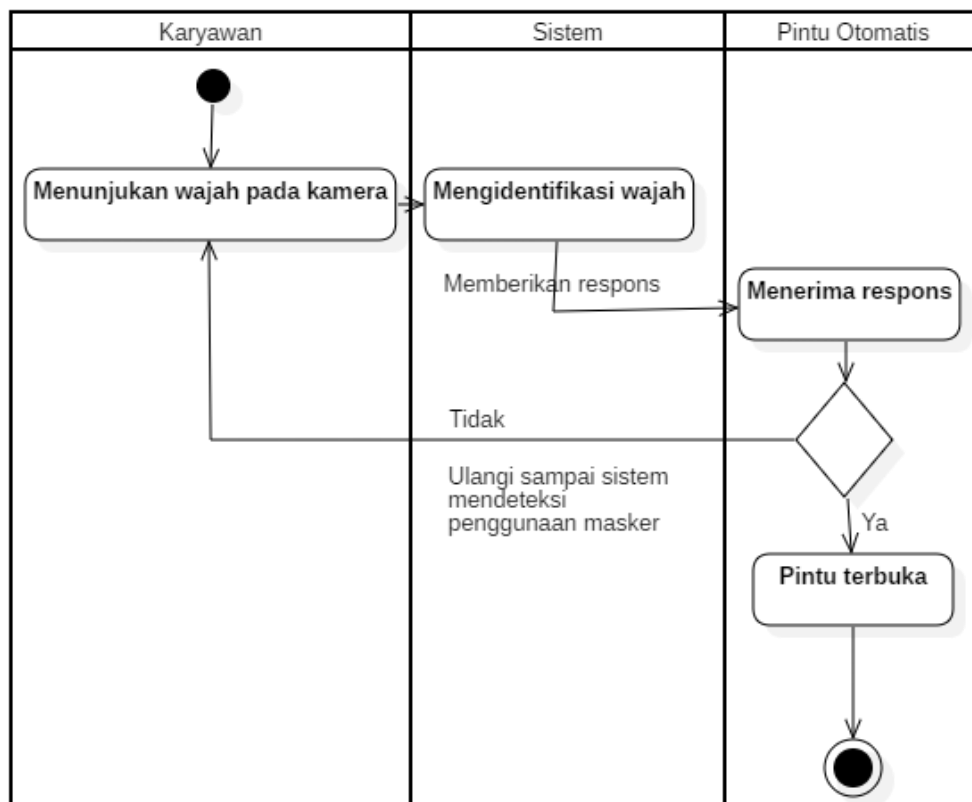
4.2.1.3 *Object Interaction (Sequence Diagram)*

Sequence diagram digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Pada gambar berikut dapat dilihat skenario sistem yang dibuat.

Gambar 4.2 *Sequence* Diagram Skenario sistem

4.2.1.4 Object Behavior (Activity Diagram)

Diagram aktivitas adalah salah satu cara untuk menggambarkan *event* yang terjadi di dalam suatu *use case* dalam bentuk sebuah model. Berikut diagram aktivitas :



Gambar 4.3 Activity Diagram Skenario sistem

4.2.2 Fase Desain

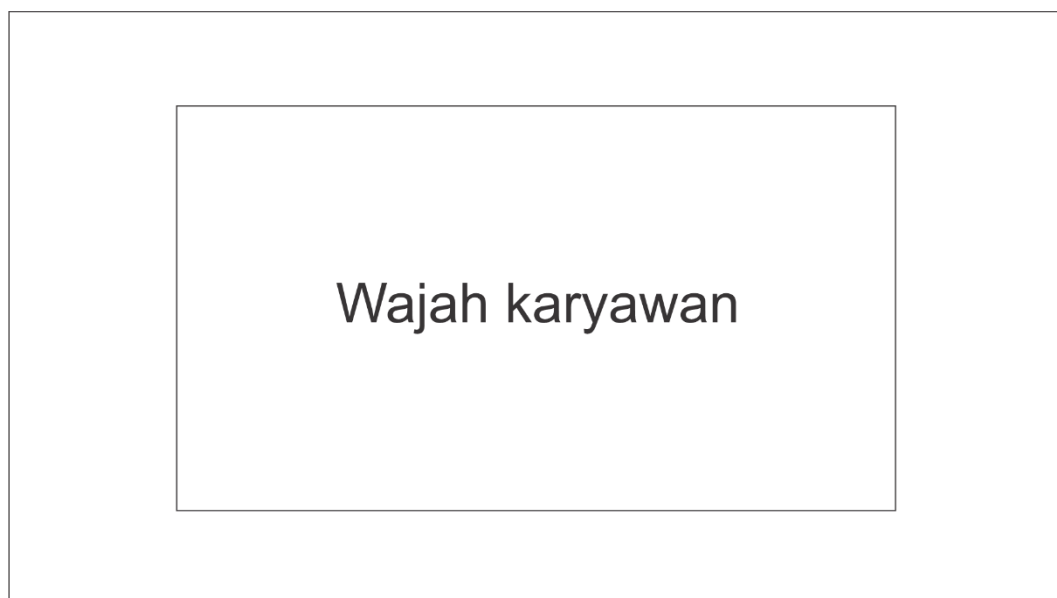
Pada fase ini berisi pemrosesan data dengan cara membawa user pada perancangan desain antarmuka (*interface*).

4.2.2.1 Perancangan Antarmuka (*interface*)

Berikut ini merupakan rancangan antarmuka dari perangkat lunak dan perangkat keras pada Aplikasi pendeteksi masker :

1. Rancangan Halaman Pendeteksi Masker

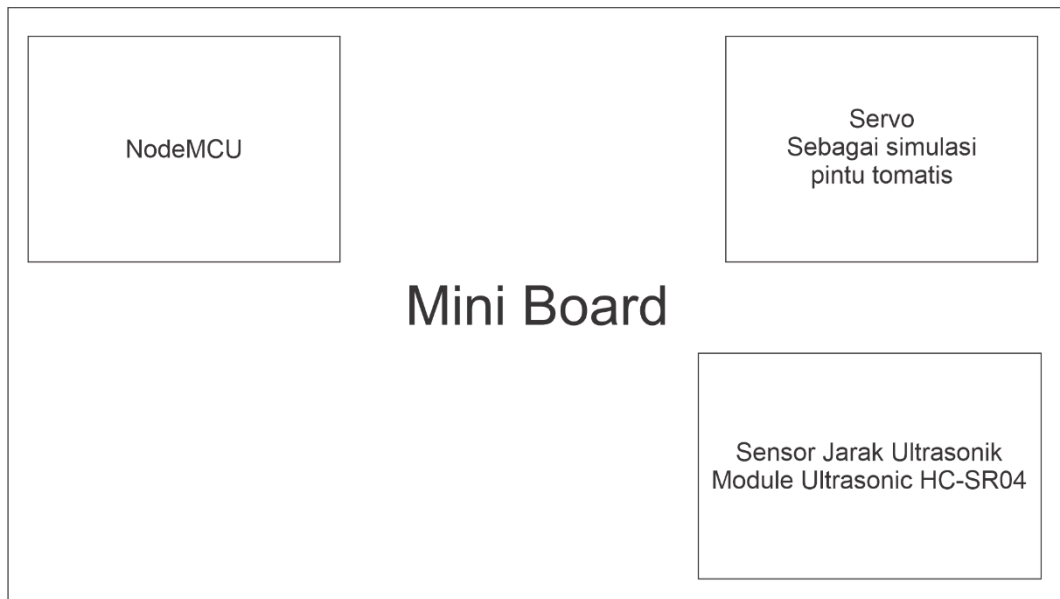
Halaman pendeteksi masker adalah halaman yang digunakan karyawan untuk menunjukkan wajahnya pada kamera, lalu pada halaman ini karyawan melihat wajahnya sendiri.



Gambar 4.4 Perancangan halaman pendeteksi masker

2. Perancangan *Hardware* Simulasi Sistem

Merupakan rancangan sistem *hardware* sebagai simulasi pintu otomatis.



Gambar 4.5 Perancangan *Hardware* Simulasi Sistem

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Spesifikasi Kebutuhan Implementasi Sistem

Tahap implementasi dan pengujian sistem adalah tahap dimana sistem siap diuji dan dipakai. Tahap ini merupakan tahap lanjutan dari tahap perancangan, analisis dan desain sistem yang bertujuan untuk menguji coba sistem yang telah di buat apakah sesuai dengan tujuan penelitian ini. Dalam tahap ini terdapat pendefinisian spesifikasi terhadap apa saja yang di butuhkan agar sistem dapat berjalan.

Implementasi sistem meliputi :

1. Perangkat Keras (*Hardware*)
2. Perangkat Lunak (*Software*)
3. Pengguna (User)

5.1.1 Spesifikasi Perangkat Keras (*Hardware*)

Berikut tabel perangkat keras yang memenuhi spesifikasi :

Tabel 5.1 Spesifikasi Perangkat Keras (*Hardware*)

No.	Perangkat Keras	Spesifikasi
1.	Laptop	HP 14-AF115AU AMD Quad Core A6 RAM 4GB 14 Inch

2.	Mikrokontroler	NodeMCU LUA WIFI V3 4MB 32MBITS, Sensor Ultrasonik HC-SR04, Motor Servo SG90 dan Kabel jumper
3.	Kabel USB	Micro USB

5.1.2 Spesifikasi Perangkat Lunak (*Software*)

Perangkat lunak yang memenuhi spesifikasi tertera dalam tabel berikut :

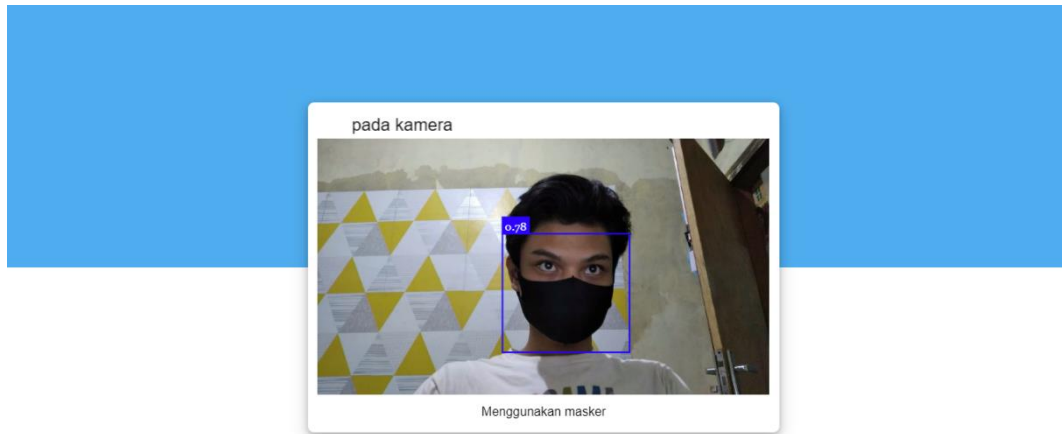
Tabel 5.2 Spesifikasi Perangkat Lunak (*Software*)

No.	Perangkat Keras	Spesifikasi
1.	Sistem Operasi	Windows 10
2.	Web Server	Apache 2.0
3.	Code Editor	Visual Studio Code dan Arduino IDE
4.	Browser	Chrome
5.	Bahasa Pemrograman	Node JS dan C
6.	StarUML	Versi 3.0

5.2 Implementasi Antarmuka Sistem

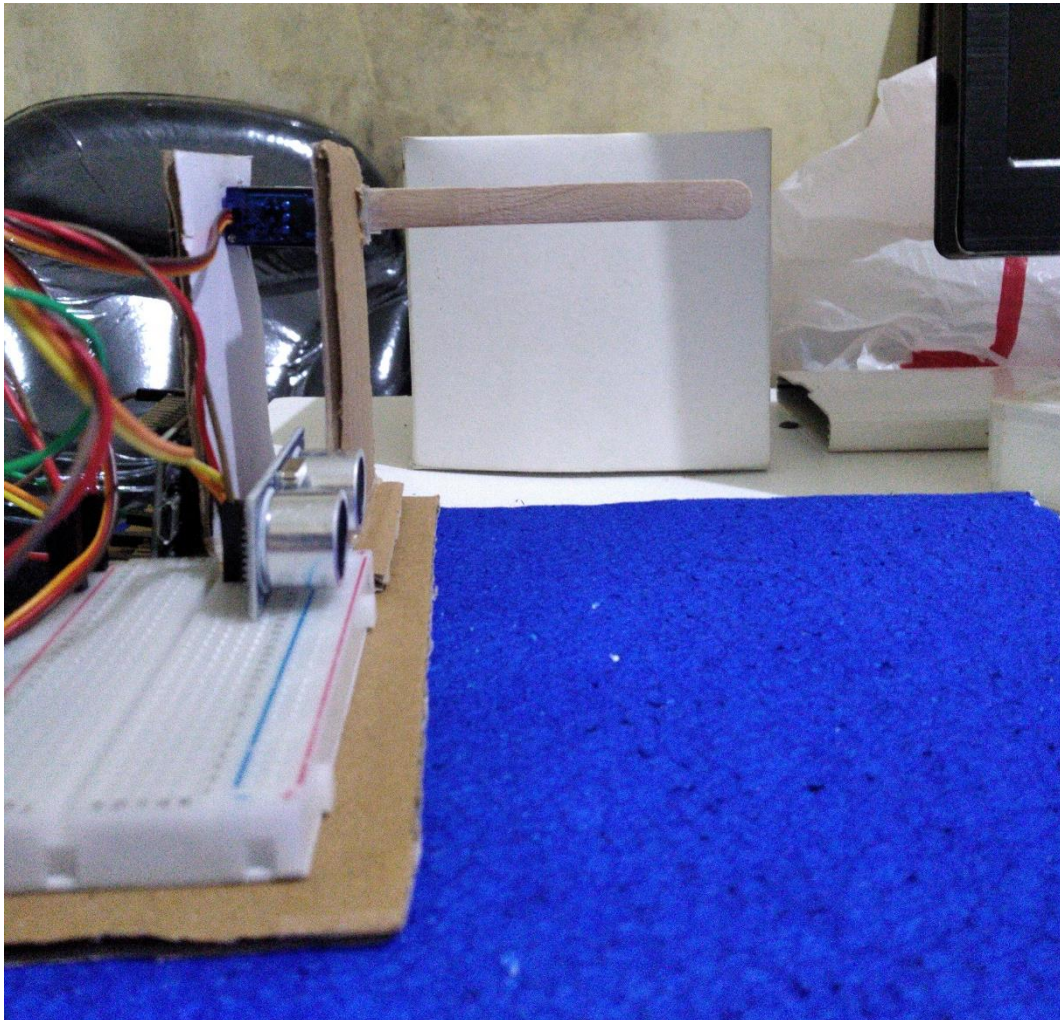
Implementasi Antarmuka merupakan pemaparan mengenai tampilan aplikasi. Untuk memperjelas bentuk dari implementasi antarmuka, berikut pemaparan dan fungsi dari setiap tampilan yang telah dibuat.

1. Halaman Pendeteksi Masker



Gambar 5.1 Antarmuka Halaman Pendeteksi Masker

2. Simulator Palang Pintu



Gambar 5.2 Antarmuka Simulator Palang Pintu


5.3 Pengujian perangkat lunak

Pengujian dilakukan untuk menjamin bahwa perangkat lunak telah berfungsi dengan baik dan sesuai dengan tujuan dari penelitian ini. Pada tahap pengujian penulis menggunakan metode *black box testing*.

5.3.1 Black Box Testing

Black box testing berfokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi *input* dan melakukan pengujian pada fungsional program (Mustaqbal, 2015:34).

Tabel 5.3 *Black Box Testing*

No.	Nama Fungsi	Kriteria	Hasil Test	Keterangan
1.	Mendeteksi masker pada wajah	Berhasil mendeteksi	Berhasil	
2.	Membuka palang pintu otomatis	Berhasil membuka	Berhasil	

5.3.2 White Box Testing

White Box Testing yaitu menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan (Rosa dan Shalahuddin, 2013:276).

1. Halaman pendeteksi masker

a. *Pseudocode*

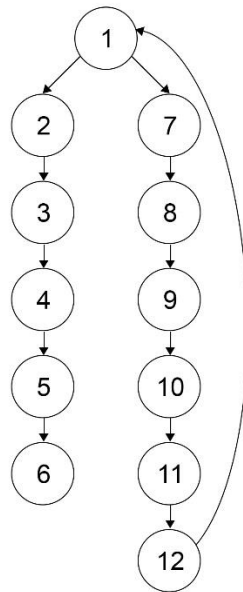
// Jika terdeteksi masker

1. If (


```
2. const maskImages = document.querySelectorAll('.mask-img');
3. maskImages.forEach(img => {
4.   const tfImg = tf.browser.fromPixels(img);
5.   const logits = mobilenetModule.infer(tfImg, 'conv_preds');
6.   classifier.addExample(logits, 0); });
7. Else (
8.   const noMaskImages = document.querySelectorAll('.no-mask-img');
9.   noMaskImages.forEach(img => {
10.    const tfImg = tf.browser.fromPixels(img);
11.    const logits = mobilenetModule.infer(tfImg, 'conv_preds');
12.    classifier.addExample(logits, 1); });
```

// Hasil deteksi

```
1. return classifier;
```

b. *Flow Graph*Gambar 5.3 *Flow Graph*c. *Cyclomatic Complexity V(G)*

$$V(G) = E - N + 2$$

$$= 12 - 12 + 2$$

$$= 2$$

d. *Independen Path*Tabel 5.4 *Independen Path*

No.	Path
1.	1, 2, 3, 4, 5, 6
2	1, 7, 8, 9, 10, 11, 12, 1

BAB VI

PENUTUP

6.1 Kesimpulan

Kesimpulan yang dapat diperoleh berdasarkan dari hasil analisis dan penerapan Aplikasi pendeteksi masker dengan metode *face recognition* yang telah dibahas pada bab-bab sebelumnya mencapai hasil dan tujuan yang diinginkan oleh perancang sistem.

Adapun beberapa kesimpulan terhadap rumusan masalah pada Sistem informasi tersebut ialah sebagai berikut :

1. Belum adanya bagian keamanan membuat prosedur pengecekan terhadap penerapan protokol kesehatan dimasa *Covid-19* pada karyawan di PT Kadetech Media Nasional menjadi tidak terkontrol.
2. Aplikasi ini dibuat sebagai alat mempermudah pengecekan masker pada karyawan di PT Kadetech Media Nasional di masa pandemi *Covid-19*.
3. Dengan dibuatnya aplikasi ini menjadi sebuah inovasi baru kegunaan *face recognition* dalam bidang industri.

6.2 Saran

Adapun saran-saran yang dikemukakan sehubungan dengan sistem baru yang dirancang adalah sebagai berikut :

1. Aplikasi dibuat hanya berbentuk simulasi, sehingga perlu diimplementasikan dengan *software* dan *hardware* yang dapat berkerja sebagaimana mestinya.

2. Aplikasi masih sangat sederhana dan mudah-mudahan dapat diperbaharui sesuai dengan kebutuhan dan manfaat yang ada.

DAFTAR PUSTAKA

Anggraini, N., Rozy, N. F., dan Lazuardy, R. A., 2013. Facial Recognition For Fatigue Detection Using Intel Realsense Technology Departement of Informatics Engineering Faculty of Science and Technology State Islamic University of Syarif Hidayatulloh Jakarta.

Cirani, S., Picone, M., Gonizzi, P., Veltri, L., & Ferrari, G. (2015). IoT-OAS: An oauth-based authorization service architecture for secure services in IoT scenarios. <https://doi.org/10.1109/JSEN.2014.2361406>

Cohen dan Birdner, 2012. Definisi Masker.

Kadir, Abdul. 2016, Arduino dan Sensor. Yogyakarta.

Mustafa, B. H. (2017). Pengembangan Perangkat Gateway untuk Sensor IoT Menggunakan Board Raspberry Pi Berbasis Protokol MQTT.

Sumardi. 2016, Belajar Mikrokontroller. Jakarta

Toko Tronik, 2018. Tutorial Pemrograman Node MCU ESP 8266 dengan Arduino. Yogyakarta.

Usman, 2002. Pengertian Penerapan.

LAMPIRAN

1. *Source Code* Halaman Pendeteksi

```
window.onload = async () => {
  const video = document.getElementById('video');
  const maskImageCount = 5;
  const noMaskImageCount = 8;
  const trainImagesContainer = document.querySelector('.train-images');
  for (let i = 1; i <= maskImageCount; i++) {
    const newImage = document.createElement('IMG');
    newImage.crossOrigin = "anonymous";
    newImage.setAttribute('src', `/public/img/mask/${i}.jpg`);
    newImage.classList.add('mask-img');
    trainImagesContainer.appendChild(newImage);
  }
  for (let i = 1; i <= noMaskImageCount; i++) {
    const newImage = document.createElement('IMG');
    newImage.crossOrigin = "anonymous";
    newImage.setAttribute('src', `/public/img/no_mask/${i}.jpg`);
    newImage.classList.add('no-mask-img');
    trainImagesContainer.appendChild(newImage);
  }
  const mobilenetModule = await mobilenet.load({ version: 2, alpha: 1 });
  const classifier = await trainClassifier(mobilenetModule);
  Promise.all([
    faceapi.nets.tinyFaceDetector.loadFromUri('/public/models'),
    faceapi.nets.faceLandmark68Net.loadFromUri('/public/models'),
    faceapi.nets.faceRecognitionNet.loadFromUri('/public/models'),
    faceapi.nets.faceExpressionNet.loadFromUri('/public/models')
  ]).then(startVideo)
  function startVideo() {
    navigator.getUserMedia(
      {
        video: {}
      },
      stream => video.srcObject = stream,
      err => console.error(err)
    )
  };
};
```

```

video.addEventListener('play', () => {
  const canvas = document.getElementById('mycanvas');
  const displaySize = { width: video.width, height: video.height }
  faceapi.matchDimensions(canvas, displaySize)
  setInterval(async () => {
    const detections = await faceapi.detectAllFaces(video, new
    faceapi.TinyFaceDetectorOptions()).withFaceLandmarks().withFaceExpressio
    ns()
    const resizedDetections = faceapi.resizeResults(detections, displaySize)
    canvas.getContext('2d').clearRect(0, 0, canvas.width, canvas.height)
    faceapi.draw.drawDetections(canvas, resizedDetections)
    const tfTestImage = tf.browser.fromPixels(video);
    const logits = mobilenetModule.infer(tfTestImage, 'conv_preds');
    const prediction = await classifier.predictClass(logits);
    if (prediction.label == 1) {
      document.getElementById('hasil').innerHTML = "Tidak menggunakan
      masker";
    } else {
      document.getElementById('hasil').innerHTML = "Menggunakan masker";
    }
  }, 100)
});
});
async function trainClassifier(mobilenetModule) {
  const classifier = knnClassifier.create();
  const maskImages = document.querySelectorAll('.mask-img');
  maskImages.forEach(img => {
    const tfImg = tf.browser.fromPixels(img);
    const logits = mobilenetModule.infer(tfImg, 'conv_preds');
    classifier.addExample(logits, 0);
  });
  const noMaskImages = document.querySelectorAll('.no-mask-img');
  noMaskImages.forEach(img => {
    const tfImg = tf.browser.fromPixels(img);
    const logits = mobilenetModule.infer(tfImg, 'conv_preds');
    classifier.addExample(logits, 1);
  });
  return classifier;
}

```

2. *Source Code* Palang Pintu

```
#include <Arduino.h>
#include <Servo.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#define USE_SERIAL Serial
ESP8266WiFiMulti WiFiMulti;
Servo servo;
// Pin Input & Output
int trigPin = D5;
int echoPin = D6;
// Variable Sensor
long duration;
int distance;
void setup() {
  USE_SERIAL.begin(9600);
  USE_SERIAL.println();
  USE_SERIAL.println();
  USE_SERIAL.println();
  for(uint8_t t = 4; t > 0; t--) {
    USE_SERIAL.flush();
    delay(1000);
  }
  WiFi.mode(WIFI_STA);
  WiFiMulti.addAP("KADETECH", "D164NT1!");
  servo.attach(D3);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {
  if((WiFiMulti.run() == WL_CONNECTED)) {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance= duration*0.034/2;
```



```

HTTPClient http;
USE_SERIAL.println("Sending Get Request to Server.....");
http.begin("http://192.168.100.64:5000/");
int httpCode = http.GET();
if(httpCode > 0) {
if(httpCode == HTTP_CODE_OK) {
// HTTP_CODE_OK == 200
String payload = http.getString();
USE_SERIAL.println(payload);
// Mulai
if(distance<20){
if(servo.read() == 150){
if(payload == "true"){
servo.write(90);
}
}else{
if(payload == "false"){
servo.write(150);
}
}
}else{
servo.write(150);
HTTPClient http;
http.begin("http://192.168.100.64:5000/dari-arduino/false/");
int httpCode = http.GET();
USE_SERIAL.println(httpCode);
}
}
}else{
// Jika Httpcode error
USE_SERIAL.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
}
http.end();
}
delay(1000);
}

```