



**TUGAS AKHIR - EE 184801**

**RANCANG BANGUN SISTEM ABSENSI PERKULIAHAN  
BERBASIS INTERNET OF THINGS**

Ismail Ashari  
NRP 07111640000020

Dosen Pembimbing  
Sri Rahayu, S.T., M.Kom.

**DEPARTEMEN TEKNIK ELEKTRO**  
**Fakultas Teknologi Elektro dan Informatika Cerdas**  
**Institut Teknologi Sepuluh Nopember**

Surabaya 2020





**TUGAS AKHIR - EE 184801**

**RANCANG BANGUN SISTEM ABSENSI PERKULIAHAN  
BERBASIS INTERNET OF THINGS**

Ismail Ashari  
NRP 07111640000020

Dosen Pembimbing  
Sri Rahayu, S.T., M.Kom.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Surabaya 2020





FINAL PROJECT - EE 184801

***IoT-BASED LECTURE ATTENDANCE SYSTEM DESIGN***

Ismail Ashari  
NRP 07111640000020

Supervisor  
Sri Rahayu, S.T., M.Kom.

ELECTRICAL ENGINEERING DEPARTEMENT  
Faculty of Electrical Technology and Intelligent Information  
Institut Teknologi Sepuluh Nopember

Surabaya 2020



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Rancang Bangun Sistem Absensi Perkuliahan Berbasis *Internet of Things***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2020

Ismail Ashari  
NRP. 0711 16 4000 0020

*-Halaman ini sengaja dikosongkan-*

**RANCANG BANGUN SISTEM ABSENSI  
PERKULIAHAN BERBASIS INTERNET OF THINGS**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Telekomunikasi Multimedia  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Menyetujui:



**SURABAYA  
JULI, 2020**

*-Halaman ini sengaja dikosongkan-*

# RANCANG BANGUN SISTEM ABSENSI PERKULIAHAN BERBASIS INTERNET OF THINGS

Nama Mahasiswa : Ismail Ashari  
Dosen Pembimbing : Sri Rahayu S.T., M.Kom.

## Abstrak

Sistem pendidikan yang diterapkan di perguruan tinggi normalnya membutuhkan tatap muka, oleh sebab itu kehadiran dalam perkuliahan sangatlah penting. Karena sering dikaitkan dengan peraturan akademik yang harus mengikuti perkuliahan minimum 80% untuk dapat mengikuti ujian akhir semester, maka untuk mengecek kehadiran dalam perkuliahan dilakukan proses absensi. Hanya saja mayoritas masih banyak menggunakan absensi konvensional (tanda tangan tulis), selain mudah dimanipulasi, juga dalam pengelolaan data membutuhkan waktu dan tenaga lebih. Dengan tujuan untuk kemudahan penggunaan dan pengelolaan data seiring dengan perkembangan zaman solusinya adalah dengan memanfaatkan penggunaan teknologi. Salah satu teknologi yang dimaksud adalah dengan berbasis Internet of Things (IoT).

Untuk merealisasikan hal tersebut, maka dalam tugas akhir ini dibuat tahapan rancangan sistem absensi, dengan menggunakan sensor sidik jari (*fingerprint*) sebagai data input. Sistem yang dibuat menggunakan sensor FPM10A sebagai input, dan mikrokontroler NodeMCU ESP8266 sebagai pusat keseluruhan kinerja sistem termasuk pengolahan data input yang selanjutnya dihubungkan ke database server (*cloud*) melalui internet menggunakan modul IoT. Selanjutnya data diproses sedemikian rupa dengan menggunakan Firebase Cloud hingga dapat ditampilkan secara visual dalam suatu aplikasi *user interface* sehingga dapat diakses dengan internet darimana saja.

Hasil pengujian operasional yang dilakukan menunjukkan bahwa sistem absensi dapat berjalan baik dengan kondisi kelas serial maupun paralel, serta memiliki rata-rata kecepatan pembacaan sidik jari saat melakukan absensi selama 2,343 detik. Fitur-fitur unggulan yang tersedia pada web berupa monitoring kehadiran pada setiap mata kuliah beserta rekapitulasi dalam satu semester untuk mahasiswa, serta pengubahan jadwal mengajar untuk dosen.

**Kata Kunci:** *Internet of Things*, mikrokontroler, *cloud*, *fingerprint*, *database*, *dashboard*.

*-Halaman ini sengaja dikosongkan-*

# **IoT-BASED LECTURE ATTENDANCE SYSTEM DESIGN**

Student Name : Ismail Ashari  
Supervisor : Sri Rahayu S.T., M.Kom.

## **Abstract**

The education system applied in many institutions normally requires face to face meeting, therefore attendance at lectures is very important. Because of it is often associated with academic regulations that must take 80% of the minimum class to be able to take the final semester exams, then a presence/attendance system is conducted to check attendance at the class. It's just that the majority still use conventional attendance system (written signatures), it is not only being easily manipulated, but also in managing data requires more time and energy. With the aim of ease of use and data management, along with the times the solution is to take advantage of the use of technology. One of the technologies in question is based on the Internet of Things (IoT).

To realize this, then in this final project will be made the attendance system design stages, using the fingerprint sensor as the input data. The system is made using FPM10A sensor as input, and the NodeMCU ESP8266 microcontroller as the center of overall system performance including processing its input data which is then connected to the database server (cloud) via the internet using the IoT module. Furthermore, the data will be processed in such a way using Firebase Cloud that it can be displayed visually in a user interface application so that it can be accessed by the internet from anywhere.

The results of operational tests show that the attendance system can run well with serial and parallel class conditions, and has an average speed of fingerprint reading during the process of absentees for 2,343 seconds. The main features available on the web are monitoring attendance in each course along with a recapitulation in one semester for students, and it is able to changing the teaching schedule for lectures.

**Keywords:** *Internet of Things, mikrokontroler, cloud, fingerprint, database, dashboard.*

*-Halaman ini sengaja dikosongkan-*

## KATA PENGANTAR

Dengan menyebut Asma Allah Yang Maha Pengasih lagi Maha Penyayang, puji syukur kami panjatkan kehadiran Allah SWT atas segala Rahmat, Karunia dan Petunjuk yang telah diliimpahkan-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “**RANCANG BANGUN SISTEM ABSENSI PERKULIAHAN BERBASIS INTERNET OF THINGS**”. Tugas akhir ini disusun sebaik-baiknya sebagai salah satu persyaratan untuk menyelesaikan jenjang pendidikan sarjana pada Bidang Studi Telekomunikasi Multimedia, Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.

Pada secarik lembar ini penulis ingin mengucapkan terima kasih sebesar-besarnya kepada berbagai pihak yang bersinggungan, yang mana sangat membantu penulis dalam menyelesaikan Tugas Akhir ini, kepada:

1. Keluarga. Bapak, Ibu dan kakak yang selalu memberikan doa dan cinta tanpa henti dalam berbagai keadaan, dan selalu mendukung serta memberikan motivasi kepada penulis.
2. Ibu Sri Rahayu, S.T., M.Kom. selaku dosen pembimbing tugas akhir yang senantiasa membimbing penulis sehingga dapat menyelesaikan tugas akhir dengan sebaik-baiknya.
3. Erliana Nurul Fatihah, yang selalu menjadi motivasi dan *support system* untuk segala hal termasuk dalam pengerjaan tugas akhir.
4. Teman-teman kontrakan E-120. Chandra, Hasbi, Luthfi, Dundux, Bagas, Fachrul, Arkan, yang selalu memberikan hiburan di perantauan dan di sela-sela pengerjaan tugas akhir.
5. Teman-teman bimbingan yang selalu saling memberi motivasi dan mengingatkan satu sama lain dalam pengerjaan tugas akhir.
6. Teman-teman e56 yang selalu menjadi inspirasi sekaligus tempat menghibur diri.
7. Teman-teman MSDM Himatektro dan Pemandu Next-Gen FTE, yang menjadi tempat belajar untuk banyak hal, termasuk dalam manajemen waktu untuk menyelesaikan tugas akhir.

8. Semua pihak yang terlibat dalam penggerjaan tugas akhir ini, yang tidak dapat penulis sebutkan satu-persatu.

Besar harapan penulis bahwa Tugas Akhir ini dapat memberikan manfaat bagi pembaca pada umumnya dan mahasiswa Teknik Elektro ITS pada khususnya.

Surakarta, Juni 2020

Penulis

## DAFTAR ISI

<b>PERNYATAAN KEASLIAN .....</b>	vii
<b>KATA PENGANTAR .....</b>	xv
<b>DAFTAR ISI.....</b>	xvii
<b>DAFTAR GAMBAR.....</b>	xxiii
<b>DAFTAR TABEL.....</b>	xxvi
<b>BAB I PENDAHULUAN .....</b>	1
1.1    Latar Belakang.....	1
1.2    Rumusan Masalah.....	2
1.3    Batasan Masalah .....	2
1.4    Tujuan .....	2
1.5    Metodologi .....	3
1.6    Sistematika Penulisan .....	4
1.7    Relevansi .....	5
<b>BAB II TINJAUAN PUSTAKA .....</b>	7
2.1    Sistem Absensi .....	7
2.2    Sensor Sidik Jari .....	8
2.2.1 <i>Total Internal Reflection Method</i> .....	9
2.2.2 <i>In-finger Light Dispersion Method</i> .....	10
2.2.3 <i>Multispectral Method</i> .....	10
2.3    Mikrokontroler.....	11
2.4    Cloud .....	13

<b>2.4.1 Pengertian <i>Cloud Computing</i></b> .....	13
<b>2.4.2 Cara Kerja <i>Cloud Computing</i></b> .....	15
<b>2.4.3 Klasifikasi <i>Cloud Computing</i></b> .....	15
<b>2.5 Jaringan Nirkabel</b> .....	17
<b>2.5.1 Jaringan Sensor Nirkabel</b> .....	18
<b>2.6 <i>Internet of Things</i> (IoT)</b> .....	19
<b>2.6.1 Pengertian IoT</b> .....	19
<b>2.6.2 Cara Kerja IoT</b> .....	20
<b>2.6.3 Arsitektur IoT</b> .....	21
<b>2.7 <i>Quality of Service</i> (QoS)</b> .....	23
<b>2.7.1 Throughput</b> .....	24
<b>2.7.2 Delay</b> .....	25
<b>2.7.3 Jitter</b> .....	25
<b>2.7.4 Packet Loss</b> .....	26
<b>2.8 Wireshark</b> .....	27
<b>2.9 Bahasa Pemrograman Web</b> .....	29
<b>2.9.1 HTML (<i>Hypertext Markup Language</i>)</b> .....	29
<b>2.9.2 SQL (<i>Structured Query Language</i>)</b> .....	30
<b>2.9.3 PHP (<i>Hypertext Preprocessor</i>)</b> .....	30
<b>2.9.4 CSS (<i>Cascading Style Sheet</i>)</b> .....	30
<b>2.9.5 Java</b> .....	30
<b>2.9.6 JavaScript</b> .....	30
<b>2.10 Komponen Penyusun Website</b> .....	31

<b>2.10.1</b>	<b>Web Browser .....</b>	31
<b>2.10.2</b>	<b>Web Server .....</b>	31
<b>2.10.3</b>	<b>Database Server.....</b>	31
<b>2.10.4</b>	<b>Web Editor.....</b>	32
<b>2.11</b>	<b>Firebase .....</b>	32
<b>2.11.1</b>	<b><i>Realtime Database</i> dan <i>Cloud Firestore</i> .....</b>	33
<b>2.11.2</b>	<b><i>Cloud Functions</i> .....</b>	33
	<b>BAB III METODOLOGI PENELITIAN.....</b>	35
<b>3.1</b>	<b>Diagram Alir.....</b>	35
<b>3.2</b>	<b>Permasalahan Sistem Presensi <i>Online</i> .....</b>	36
<b>3.3</b>	<b>Desain Sistem Absensi Berbasis <i>IoT</i> .....</b>	36
<b>3.3.1</b>	<b>Konsep Sistem Absensi Berbasis IoT .....</b>	36
<b>3.3.2</b>	<b><i>Online Attendance System</i> (OASYS) .....</b>	38
<b>3.3.3</b>	<b>Skenario Pemanfaatan Sistem .....</b>	41
<b>3.4</b>	<b>Rancangan Perangkat Keras .....</b>	41
<b>3.4.1</b>	<b>Perencanaan Proses Sistem Absensi.....</b>	42
<b>3.4.2</b>	<b>Pemilihan Komponen <i>Device</i> .....</b>	43
<b>3.4.2.1</b>	<b>NodeMCU ESP8266 .....</b>	43
<b>3.4.2.2</b>	<b>Sensor Sidik Jari FPM10A .....</b>	44
<b>3.4.2.3</b>	<b>LCD Display.....</b>	46
<b>3.4.2.4</b>	<b>Lampu LED .....</b>	47
<b>3.4.2.5</b>	<b>Buzzer .....</b>	47
<b>3.4.3</b>	<b>Rangkaian <i>Device</i> Sistem Absensi .....</b>	48

<b>3.5 Rancangan Perangkat Lunak .....</b>	49
<b>3.5.1 <i>Software Pemrograman</i> .....</b>	49
<b>3.5.2 Pembuatan Pemrograman.....</b>	50
<b>3.5.2.1 Pemrograman Pada Arduino .....</b>	50
<b>3.5.2.2 Pemrograman Pada Database .....</b>	51
<b>3.5.2.3 Pemrograman Tampilan Web.....</b>	54
<b>3.6 Integrasi Sistem .....</b>	66
<b>3.7 Pengujian Sistem .....</b>	66
<b>3.7.1 Pengujian Segmen Komunikasi .....</b>	66
<b>3.7.2 Pengujian Operasional Sistem .....</b>	67
<b>3.8 Pengukuran Kualitas Layanan .....</b>	68
<b>3.8.1 <i>Tool Pengukuran</i> .....</b>	69
<b>3.8.2 Parameter Pengukuran .....</b>	69
<b>3.8.3 Skenario Pengukuran .....</b>	70
<b>BAB IV ANALISIS DATA DAN PEMBAHASAN .....</b>	73
<b>4.1 Hasil Implementasi Desain Sistem Absensi .....</b>	73
<b>4.1.1 Hasil Implementasi <i>Hardware</i> .....</b>	73
<b>4.1.2 Hasil Pemrograman Database .....</b>	75
<b>4.1.3 Hasil Pemrograman Web .....</b>	77
<b>4.1.3.1 Hasil Halaman Login .....</b>	77
<b>4.1.3.2 Hasil Tampilan Mahasiswa .....</b>	78
<b>4.1.3.3 Hasil Tampilan Dosen .....</b>	79
<b>4.2 Hasil Pengujian Integrasi Sistem.....</b>	81

<b>4.2.1 Hasil Pengujian Segmen Komunikasi dan Integrasi Sistem.....</b>	<b>81</b>
<b>4.2.2 Hasil Pengujian Operasional Sistem .....</b>	<b>84</b>
<b>    4.2.2.1 Hasil Pengujian Fitur .....</b>	<b>84</b>
<b>    4.2.2.2 Hasil Pengujian Jenis Kelas.....</b>	<b>90</b>
<b>    4.2.3 Hasil Pengujian Keandalan Sistem.....</b>	<b>94</b>
<b>        4.2.3.1 Hasil Pengujian Kecepatan Proses Pembacaan Sidik Jari</b>	<b>95</b>
<b>        4.2.3.2 Hasil Pengujian Ketepatan Penyimpanan Data.....</b>	<b>95</b>
<b>4.3     Hasil Pengukuran Kualitas Layanan .....</b>	<b>97</b>
<b>    4.3.1 Hasil Pengujian Pengaruh Jarak Ruang Kelas .....</b>	<b>97</b>
<b>    4.3.2 Hasil Pengujian Pengaruh Jumlah Ruang Kelas .....</b>	<b>98</b>
<b>    4.3.3 Hasil Pengujian Pengaruh Waktu Pengiriman Data ...</b>	<b>99</b>
<b>4.4     Analisa Hasil dan Pembahasan.....</b>	<b>100</b>
<b>    4.4.1 Hasil Pengujian Sistem .....</b>	<b>100</b>
<b>4.5     Perbandingan Performansi Sistem .....</b>	<b>101</b>
<b>    4.5.1 Perbandingan Parameter Hasil Pengukuran .....</b>	<b>102</b>
<b>    4.5.2 Klasifikasi Hasil Pengukuran Berdasarkan Standar Tiphon .....</b>	<b>106</b>
<b>4.6     Pemanfaatan Sistem OASYS .....</b>	<b>109</b>
<b>4.7     Alur Penggunaan Sistem OASYS (<i>User Manual</i>).....</b>	<b>110</b>
<b>    4.7.1 Proses Registrasi .....</b>	<b>110</b>
<b>    4.7.2 Proses Absensi .....</b>	<b>111</b>
<b>    4.7.3 Monitoring Kehadiran.....</b>	<b>111</b>

<b>4.8</b>	<b>Analisa Pengembangan Sistem .....</b>	112
<b>BAB V KESIMPULAN DAN SARAN .....</b>		115
<b>5.1</b>	<b>Kesimpulan .....</b>	115
<b>5.2</b>	<b>Saran .....</b>	116
<b>DAFTAR PUSTAKA .....</b>		117
<b>LAMPIRAN .....</b>		119
<b>BIODATA PENULIS .....</b>		195

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Permukaan Kulit .....	8
<b>Gambar 2. 2</b> Total Internal Reflection Method.....	9
<b>Gambar 2. 3</b> In-finger Light Dispersion Method .....	10
<b>Gambar 2. 4</b> Multispectral Method.....	11
<b>Gambar 2. 5</b> Komponen Mikrokontroler .....	12
<b>Gambar 2. 6</b> Cloud Computing.....	14
<b>Gambar 2. 7</b> Perbedaan Cloud Computing berdasarkan Service Layer	16
<b>Gambar 2. 8</b> Contoh Arsitektur Dasar WSN .....	18
<b>Gambar 2. 9</b> Elemen Utama IoT.....	20
<b>Gambar 2. 10</b> Layer 1: IoT Devices (sensor dan aktuator).....	21
<b>Gambar 2. 11</b> Layer 2: IoT Gateway (modul wifi).....	22
<b>Gambar 2. 12</b> Layer 3: Cloud .....	22
<b>Gambar 2. 13</b> 7 Layer IoT .....	23
<b>Gambar 2. 14</b> <i>Struktur GUI Software Wireshark</i> .....	28
<b>Gambar 2. 15</b> <i>Struktur Packet Sniffer</i> .....	29
<b>Gambar 2. 16</b> Firebase.....	32
<b>Gambar 2. 17</b> Contoh penggunaan Cloud functions.....	34
<b>Gambar 3. 1</b> Diagram Alir Penelitian .....	35
<b>Gambar 3. 2</b> Rancangan Alur Sistem .....	37
<b>Gambar 3. 3</b> Desain Sistem OASYS .....	38
<b>Gambar 3. 4</b> Diagram Alur Sistem .....	40
<b>Gambar 3. 5</b> Diagram Alur Device Absensi IoT .....	42
<b>Gambar 3. 6</b> NodeMCU ESP8266.....	43
<b>Gambar 3. 7</b> Sensor FPM10A.....	44
<b>Gambar 3. 8</b> OLED Display SSD1306 0.96” inch White .....	46
<b>Gambar 3. 9</b> Lampu LED .....	47
<b>Gambar 3. 10</b> Buzzer Aktif.....	47
<b>Gambar 3. 11</b> Rangkaian Device Absensi .....	48
<b>Gambar 3. 12</b> Arduino IDE .....	49
<b>Gambar 3. 13</b> Pemrograman Pada Mikrokontroler dan Modul IoT.....	51
<b>Gambar 3. 14</b> Diagram Alur Website .....	54
<b>Gambar 3. 15</b> Tampilan Halaman Login User.....	55
<b>Gambar 3. 16</b> Pemrograman Halaman Login .....	55

<b>Gambar 3. 17</b> Tampilan Mata Kuliah .....	56
<b>Gambar 3. 18</b> Pemrograman Halaman Mata Kuliah Mahasiswa.....	57
<b>Gambar 3. 19</b> Tampilan Halaman Laporan .....	57
<b>Gambar 3. 20</b> Pemrograman Halaman Laporan .....	58
<b>Gambar 3. 21</b> Diagram Alur Admin pada Sistem.....	59
<b>Gambar 3. 22</b> Tampilan Halaman Login .....	60
<b>Gambar 3. 23</b> Pemrograman Halaman Login .....	60
<b>Gambar 3. 24</b> Tampilan Halaman Mata Kuliah.....	61
<b>Gambar 3. 25</b> Pemrograman Halaman Mata Kuliah Admin.....	62
<b>Gambar 3. 26</b> Tampilan Halaman Daftar Mahasiswa.....	62
<b>Gambar 3. 27</b> Pemrograman Halaman Daftar Mahasiswa.....	63
<b>Gambar 3. 28</b> Diagram Alur Pendaftaran Sidik Jari .....	64
<b>Gambar 3. 29</b> Diagram Alur Penghapusan Sidik Jari .....	65
<b>Gambar 4. 1</b> Rangkaian Device Absensi .....	73
<b>Gambar 4. 2</b> Prototype Absensi .....	74
<b>Gambar 4. 3</b> Visualisasi Single-class.....	75
<b>Gambar 4. 4</b> Visualisasi Multi-class.....	75
<b>Gambar 4. 5</b> Tampilan Database Firebase Kelompok Mahasiswa .....	76
<b>Gambar 4. 6</b> Tampilan Database Firebase Kelompok MataKuliah .....	76
<b>Gambar 4. 7</b> Tampilan Halaman Login .....	77
<b>Gambar 4. 8</b> Halaman Mata Kuliah – Mahasiswa .....	78
<b>Gambar 4. 9</b> Halaman Laporan – Mahasiswa.....	79
<b>Gambar 4. 10</b> Halaman Mata Kuliah – Dosen.....	80
<b>Gambar 4. 11</b> Halaman Daftar Mahasiswa - Dosen .....	80
<b>Gambar 4. 12</b> Koneksi Sensor ke Mikrokontroler .....	81
<b>Gambar 4. 13</b> Deteksi Sensor Fingerprint .....	81
<b>Gambar 4. 14</b> Modul IoT .....	82
<b>Gambar 4. 15</b> Koneksi ke WiFi (internet) .....	82
<b>Gambar 4. 16</b> Database pada Firebase.....	83
<b>Gambar 4. 17</b> Tampilan Daftar Sidik Jari.....	85
<b>Gambar 4. 18</b> Proses scanning sidik jari.....	85
<b>Gambar 4. 19</b> Sidik jari terbaca pada scanning pertama.....	86
<b>Gambar 4. 20</b> Lepaskan jari dari sensor .....	86
<b>Gambar 4. 21</b> Sidik jari terbaca pada scanning kedua .....	86
<b>Gambar 4. 22</b> Sidik jari tersimpan.....	86
<b>Gambar 4. 23</b> Tampilan web saat menekan tombol hapus .....	87

<b>Gambar 4. 24</b> Sidik jari telah dihapus dari database.....	87
<b>Gambar 4. 25</b> Tampilan web saat ubah jadwal .....	88
<b>Gambar 4. 26</b> Posisi idle.....	89
<b>Gambar 4. 27</b> Halaman Mata Kuliah – Mahasiswa .....	89
<b>Gambar 4. 28</b> Halaman Laporan – Mahasiswa.....	90
<b>Gambar 4. 29</b> Tap Sidik Jari Paralel .....	91
<b>Gambar 4. 30</b> Halaman Mata Kuliah Mahasiswa 1 .....	91
<b>Gambar 4. 31</b> Halaman Mata Kuliah Mahasiswa 2 .....	92
<b>Gambar 4. 32</b> Tap Sidik Jari Sesi 1 .....	92
<b>Gambar 4. 33</b> Halaman Mata Kuliah Sistem Komunikasi.....	93
<b>Gambar 4. 34</b> Tap Sidik Jari Sesi 2 .....	93
<b>Gambar 4. 35</b> Halaman Mata Kuliah Standard dan Keandalan SisKom .....	94
<b>Gambar 4. 36</b> Data Tersimpan Pada Database .....	96
<b>Gambar 4. 37</b> Data Tersimpan Pada Database .....	97
<b>Gambar 4. 38</b> Grafik Perbandingan Pengukuran Throughput .....	103
<b>Gambar 4. 39</b> Grafik Perbandingan Pengukuran Delay.....	104
<b>Gambar 4. 40</b> Grafik Perbandingan Pengukuran Jitter .....	105
<b>Gambar 4. 41</b> Grafik Perbandingan Pengukuran Packet Loss .....	106
<b>Gambar 4. 42</b> Desain Pemanfaatan Sistem.....	109

## DAFTAR TABEL

<b>Tabel 2. 1</b> Persentase dan Nilai QoS .....	24
<b>Tabel 2. 2</b> Kategori Throughput .....	24
<b>Tabel 2. 3</b> Kategori Delay .....	25
<b>Tabel 2. 4</b> Kategori Jitter .....	26
<b>Tabel 2. 5</b> Kategori Packet Loss.....	26
<b>Tabel 3. 1</b> Spesifikasi NodeMCU ESP8266.....	43
<b>Tabel 3. 2</b> Spesifikasi Sensor FPM10A.....	45
<b>Tabel 3. 3</b> Spesifikasi OLED SSD1306.....	46
<b>Tabel 3. 4</b> Spesifikasi Lampu LED .....	47
<b>Tabel 3. 5</b> Spesifikasi Buzzer .....	48
<b>Tabel 3. 6</b> Database Mahasiswa .....	51
<b>Tabel 3. 7</b> Database Jadwal Mata Kuliah .....	53
<b>Tabel 4. 1</b> Hasil Uji Segmentasi .....	84
<b>Tabel 4. 2</b> Hasil Pengujian Fitur.....	90
<b>Tabel 4. 3</b> Hasil Pengujian Jenis Kelas.....	94
<b>Tabel 4. 4</b> Hasil Pengujian Kecepatan Pembacaan Sidik Jari.....	95
<b>Tabel 4. 5</b> Hasil Pengujian Ketepatan Pengiriman Data.....	96
<b>Tabel 4. 6</b> Hasil Pengujian Pengaruh Jarak Ruang Kelas .....	98
<b>Tabel 4. 7</b> Hasil Pengujian Pengaruh Jumlah Ruang Kelas.....	99
<b>Tabel 4. 8</b> Hasil Pengujian Pengaruh Waktu Pengiriman Data .....	100
<b>Tabel 4. 9</b> Deskripsi Kondisi Pengukuran .....	102
<b>Tabel 4. 10</b> Klasifikasi Hasil Pengukuran Berdasarkan Standar Tiphon .....	107
<b>Tabel 4. 11</b> Hasil Rata-rata Keseluruhan Pengukuran .....	109
<b>Tabel 4. 12</b> Perbandingan Sistem Absensi Online.....	112

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Absensi merupakan suatu aktivitas pelaporan dan pendataan kehadiran yang ada dalam suatu institusi. Absensi atau kehadiran adalah dasar dari segala bentuk suatu hubungan, dengan hadir berarti kita siap untuk menerima, mendapat ataupun menyampaikan sesuatu. Berikut juga dalam proses belajar mengajar, dengan hadir berarti kita siap untuk terlibat dalam proses pembelajaran. Dalam proses belajar mengajar, kehadiran yang dimaksud adalah sistem absensi atau presensi. Berdasarkan cara penggunaannya, sistem absensi dapat dikelompokkan menjadi manual dan digital. Sistem absensi manual masih banyak ditemukan di banyak perguruan tinggi, yaitu dengan tanda tangan pada lembar kertas yang dibagikan di dalam kelas. Sistem ini memiliki beberapa kekurangan seperti sangat rawan akan adanya manipulasi data atau titip absen kepada teman, memerlukan banyak kertas dan tinta, serta membutuhkan ruang yang banyak sebagai tempat penyimpanan sehingga menjadi tidak ramah lingkungan.

Seiring dengan kemajuan teknologi yang ada pada era digital seperti sekarang ini, berbagai permasalahan di kehidupan sehari-hari dapat diatasi menggunakan pemanfaatan teknologi, salah satunya adalah dengan menghubungkan ke internet. Semakin hari penggunaan internet semakin mudah, hampir setiap orang dengan mudah dapat mengakses internet kapanpun dan dari manapun. Hal ini menjadi dorongan banyak orang untuk berlomba-lomba menciptakan inovasi teknologi dengan internet.

Berdasarkan permasalahan yang telah diuraikan, hal ini dapat dioptimalkan dengan memadukan teknologi dan sistem absensi manual atau konvensional. Dalam tugas akhir ini akan dibuatkan perancangan sistem absensi dengan menggunakan sensor sidik jari yang dihubungkan ke internet, untuk selanjutnya dapat disebut dengan *Internet of Things*. Dengan adanya sistem absensi ini diharapkan kecurangan-kecurangan dalam kehadiran mahasiswa pada perkuliahan dapat diminimalisir, sehingga peraturan yang telah dibuat dapat dijalankan sebagaimana mestinya, dan pada akhirnya dapat menaikkan kualitas kampus.

## **1.2 Rumusan Masalah**

Beberapa permasalahan yang akan dibahas dalam tugas akhir tentang sistem absensi berbasis IoT ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun desain sistem absensi berbasis IoT dengan sensor sidik jari.
2. Bagaimana kualitas jaringan yang digunakan untuk menghubungkan *device* ke server.
3. Bagaimana menampilkan data yang diperoleh untuk *monitoring* kehadiran melalui tampilan web yang *real-time* dan dapat diakses dimana saja melalui browser.

## **1.3 Batasan Masalah**

Adapun batasan masalah dalam Tugas Akhir ini adalah sebagai berikut:

1. Menggunakan sensor sidik jari FPM10A untuk pengambilan data input absensi mahasiswa.
2. Menggunakan NodeMCU sebagai mikrokontroler.
3. Menggunakan ESP8266 sebagai modul wifi.
4. Menggunakan Cloud server.
5. Mengambil skala 1 departemen (Teknik Elektro) sebagai objek penelitian.
6. Telah dibuat terlebih dahulu database jadwal kuliah mahasiswa.
7. Pengukuran kinerja sistem menggunakan perangkat lunak wireshark.
8. Menggunakan standar pengukuran ITU-T dan TIPHON (ETSI).

## **1.4 Tujuan**

Adapun tujuan dari pembuatan tugas akhir ini adalah untuk menjawab berbagai permasalahan tentang sistem absensi berbasis IoT, antara lain:

1. Membuat rancangan dan desain sistem absensi sidik jari yang terhubung ke internet dan dapat diakses dimana saja serta tidak mudah untuk dimanipulasi, untuk memudahkan dalam monitoring kehadiran mahasiswa.

2. Membuat tampilan antarmuka web menggunakan bahasa pemrograman PHP, HTML, dan MySQL, sebagai aplikasi *monitoring* kehadiran mahasiswa.
3. Melakukan pengukuran kinerja jaringan pada sistem yang telah dibuat.
4. Mengetahui persentase kehadiran tiap mahasiswa untuk pemantauan serta sebagai syarat untuk dapat mengikuti Evaluasi Akhir Semester (EAS).

## **1.5 Metodologi**

Dalam pelaksanaan Tugas Akhir ini, menggunakan metodologi yang dapat diuraikan sebagai berikut:

1. Studi Literatur  
Pada tahap ini dimulai dengan mencari dan mempelajari beberapa referensi, baik jurnal maupun buku yang berhubungan dengan sistematika pada sidik jari (biometric). Melakukan observasi terhadap sistem absensi yang telah ada di berbagai institusi khususnya yang berhubungan dengan sistem pendidikan. Melakukan studi lebih lanjut mengenai mikrokontroler, database dan server cloud, Bahasa pemrograman web, dan Quality of Service (QOS) pada jaringan. Hal ini dilakukan untuk menambah pemahaman mengenai permasalahan yang dihadapi dan mengetahui langkah-langkah dalam menyelesaikan permasalahan tersebut.
2. Pendefinisian Permasalahan  
Pada tahap ini semua permasalahan yang akan muncul harus diidentifikasi sebaik-baiknya, untuk selanjutnya dapat dipertimbangkan dalam kebutuhan baik perangkat keras maupun perangkat lunak sebagai sistem yang akan dibuat dalam tugas akhir.
3. Perancangan Sistem  
Dalam tahap perancangan sistem dilakukan penentuan perangkat keras dan perangkat lunak apa saja yang dibutuhkan untuk membuat prototipe sistem absensi berbasis IoT dan bagaimana

cara agar alat dapat terintegrasi satu sama lain melalui koneksi internet.

4. Realisasi Sistem

Dalam merealisasikan sistem absensi berbasis IoT ini akan dibagai menjadi 2, yaitu pembuatan hardware dan software untuk dapat mengambil data dan ditampilkan pada website.

5. Pengujian Sistem

Dalam tahap pengujian sistem dilakukan implementasi terhadap alat yang dibuat yang akan dibagi menjadi 3 jenis pengujian, yaitu pengujian segmen, pengujian jaringan dan pengujian operasional.

6. Analisis Data dan Kesimpulan

Setelah melakukan beberapa skema pengujian sistem akan diperoleh hasil data pengujian, yang kemudian akan dilakukan analisis terhadap performansi sistem, sehingga dapat dihasilkan suatu kesimpulan.

7. Penulisan Buku Tugas Akhir

Dalam tahap ini dilakukan penulisan buku laporan tugas akhir yang terdiri dari beberapa bab, yaitu Bab 1 (Pendahuluan), Bab 2 (Tinjauan Pustaka), Bab 3 (Metodologi), Bab 4 (Hasil Data dan Analisa) dan Bab 5 (Penutup).

## **1.6 Sistematika Penulisan**

Laporan tugas akhir akan disusun dalam beberapa bab dengan perincian sebagai berikut:

### **BAB 1 : PENDAHULUAN**

Pada bab ini berisikan tentang latar belakang penelitian, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, sistematika penulisan dan relevensi.

### **BAB 2 : TINJAUAN PUSTAKA**

Pada bab ini membahas tentang teori penunjang yang digunakan dalam penelitian secara singkat dan jelas. Tinjauan pustaka yang dibahas adalah teori dasar terkait sistem absensi, biometrika sidik jari, mikrokontroler, sensor nirkabel, dan bahasa pemrograman web.

### **BAB 3 : METODOLOGI**

Pada bab ini membahas mengenai metode yang digunakan untuk merancang sistem absensi berbasis IoT menggunakan sidik jari, rumus-rumus yang dibutuhkan dalam perhitungan dan juga data-data pendukung perancangan sistem.

### **BAB 4 : HASIL SIMULASI DAN ANALISA DATA**

Pada bab ini berisi tentang analisis dari hasil pengujian sistem yang sudah dirancang. Hasil analisis meliputi pengujian segmen, pengujian QoS, dan pengujian operasional termasuk kinerja website.

### **BAB 5 : PENUTUP**

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran terhadap penelitian selanjutnya yang terkait dengan penelitian ini.

## **1.7 Relevansi**

Dengan dibuatnya sistem absensi berbasis IoT sebagaimana yang dikerjakan dalam tugas akhir ini dapat dimanfaatkan untuk membantu memudahkan pengelolaan administrasi dari segi kehadiran mahasiswa, terkhusus pada Departemen Teknik Elektro ITS. Dengan pengelolaan absensi yang lebih baik, maka diharapkan data tersebut bisa digunakan sebaik-baiknya agar peraturan yang berlaku dapat dijalankan sebagaimana mestinya.

*-Halaman ini sengaja dikosongkan-*

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sistem Absensi**

Absensi atau catatan kehadiran adalah dokumen yang mencatat waktu hadir setiap orang, dalam hal ini adalah mahasiswa di perguruan tinggi. Catatan kehadiran mahasiswa tersebut dapat berupa daftar hadir biasa yang diisi dengan tandatangan, ataupun yang lebih modern menggunakan teknologi sensor untuk pencatatan. Pencatatan kehadiran yang dilakukan kepada setiap mahasiswa bisa mempengaruhi nilai akademik ataupun status diperbolehkan atau tidaknya mengikuti ujian akhir dilihat dari catatan kehadiran. Pencatatan waktu hadir tersebut dimaksudkan untuk mengumpulkan data mengenai jumlah kehadiran mahasiswa per mata kuliah yang dikalkulasikan selama satu semester.

Berdasarkan cara penggunaannya, sistem absensi dapat dikelompokkan menjadi manual dan digital. Sistem absensi manual yang dilakukan di banyak perguruan tinggi mengharuskan mahasiswa mengisi formulir absensi dengan tandatangan pada saat perkuliahan berlangsung ataupun sebelum dimulai. Hal ini tentu memiliki kekurangan, selain dari segi lingkungan yang berupa banyaknya penggunaan kertas dan tinta serta ruang untuk penyimpanan, juga dari segi kejujuran masih banyak ditemui adanya penitipan tandatangan.

Seiring dengan berkembangnya zaman, pemanfaatan teknologi sangat amat membantu dalam memecahkan berbagai permasalahan di kehidupan, salah satunya dalam dunia pendidikan terkhusus dari sisi absensi siswa/mahasiswa semakin banyak dikembangkan. Kekurangan-kekurangan yang ada pada sistem absensi manual tersebut diperbaiki sedikit demi sedikit melalui inovasi pada bidang digital. Digitalisasi absensi, yaitu dimana proses absensi yang menggunakan tandatangan semakin berkurang dan diganti dengan sensor-sensor digital, yang dapat berupa sidik jari, kartu RFID, NFC, sensor pengenalan wajah, dan lain sebagainya. Dengan digunakannya sistem absensi digital ini diharapkan berbagai bentuk kekurangan ataupun kecurangan dari sistem sebelumnya dapat semakin berkurang.

Beberapa kelebihan dalam penggunaan sistem absensi digital (*online*) adalah:

1. Rekapitulasi data lebih mudah.
2. Mengurangi potensi kecurangan.
3. Pencatatan jam/waktu kehadiran lebih akurat.
4. Dapat dijadikan dasar untuk penentuan nilai akademik.
5. Pemantauan *real-time*.
6. Mengurangi penggunaan kertas (*paperless*).

## 2.2 Sensor Sidik Jari

Sidik jari (*fingerprint*) adalah gurat-gurat yang terdapat pada kulit ujung jari. Para pakar membuktikan bahwa tidak ada dua individu yang memiliki pola *ridge* yang serupa. Pola *ridge* tidak dapat diwariskan. Karakteristik sidik jari merupakan gabungan dari pola bukit (*ridge*) dan lembah (*valley*). Pola bukit pada gambar terlihat gelap, sedangkan pola lembah pada gambar terlihat terang. Bentuk dari bukit dan lembah ini merupakan kombinasi dari faktor genetik dan faktor lingkungan, dapat dilihat pada Gambar 2.1 sebagai berikut.



**Gambar 2. 1** Permukaan Kulit

DNA memberikan arah dalam pembentukan kulit pada janin, namun pembentukan sidik jari pada kulit itu sendiri merupakan suatu kejadian acak yang tidak dapat dijelaskan. Sistem biometrika sidik jari merupakan sistem autentifikasi berbasis biometrika yang paling banyak digunakan saat ini karena memiliki tingkat akurasi tinggi dan mudah diterapkan. Dari hasil penelitian, ditemukan ada tujuh macam pola utama *poppillary ridge*, antara lain: *Loop*, *Arch*, *Whorl*, *Tented Arch*, *Double*

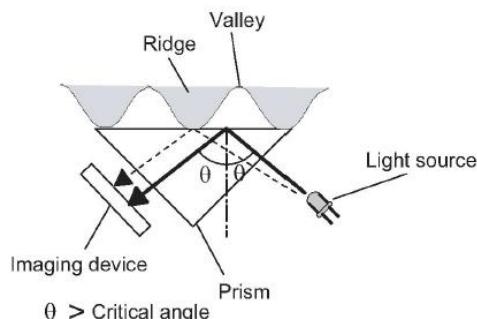
*Loop*, *Central Pocked Loop*, dan *Accidental*. Dari ketujuh pola tersebut, yang paling utama dan sering ditemukan di masyarakat adalah tipe *arch*, *loop*, dan *whorl*. Tipe *loop* merupakan pola yang paling banyak ditemukan.

Komputer dapat menganalisa garis-garis perubahan arah berbentuk *ridge*, dengan kemampuan seperti mata manusia yang terlatih. Pada tingkat lokal, sidik jari dapat terlihat lebih detail. Pada tingkat ini dapat ditemukan titik minusi. Titik minusi merupakan titik-titik informasi yang dapat mencirikan suatu sidik jari pada setiap orang.

Ada beberapa macam prinsip/metode penginderaan pola sidik jari, metode penginderaan pola sidik jari secara luas dapat dibagi menjadi sistem optik dan sistem semikonduktor (Emiko Sano, 2006). Berikut ini merupakan beberapa metode penginderaan sensor sidik jari berbasis optik.

### 2.2.1 Total Internal Reflection Method

Dalam metode *total internal reflection*, jari ditempatkan pada permukaan prisma dan permukaan yang sama diiradiasi dengan cahaya di bawah. Pola *ridge* pada sidik jari merespon kontak antara kulit dengan permukaan prisma, dan mempertahankan kondisi *total internal reflection*. Penggambaran metode ini dapat dilihat pada Gambar 2.2 sebagai berikut.

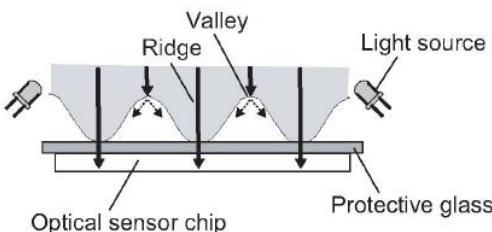


**Gambar 2. 2 Total Internal Reflection Method**

Teknik ini secara langsung menangkap pola sidik jari dengan mendeteksi kontak antara bentuk cekung/cembung (bukit/lembah) dari permukaan jari dan permukaan sensor.

### **2.2.2 In-finger Light Dispersion Method**

Jari ditempatkan langsung pada chip sensor optik yang ditutupi oleh kaca pelindung transparan khusus. Cahaya sekitar yang tersebar di bawah sidik jari akan melewati permukaan sidik jari, dan mencapai chip sensor. Pada posisi lembah, cahaya akan tersebar di udara dan menjadi lemah, menghasilkan gambar yang lebih gelap. Cahaya yang melewati posisi punggungan secara langsung terjadi pada chip sensor, menghasilkan gambar yang lebih terang. Penggambaran metode ini dapat dilihat pada Gambar 2.3 sebagai berikut.

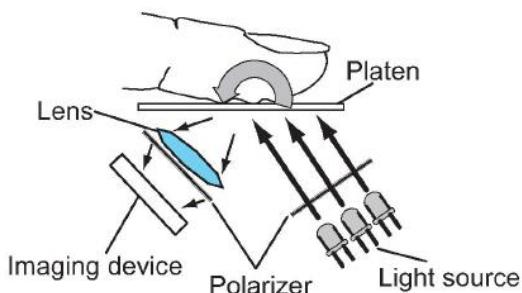


**Gambar 2.3 In-finger Light Dispersion Method**

Karena metode ini mendeteksi perbedaan kekuatan cahaya yang tersebar di bagian lembah dan langsung masuk ke chip sensor pada bagian bukit, metode ini dapat lebih sensitif untuk jari kering dan basah daripada metode konvensional lainnya. Namun, metode ini pada dasarnya tergantung pada ada atau tidaknya kontak antara jari dan permukaan sensor, dan yang tidak dapat dihindari yaitu akan ada pengaruh pada situasi ketika bukit tidak menyentuh permukaan sensor karena adanya kerutan dalam sidik jari.

### **2.2.3 Multispectral Method**

Metode ini bertujuan untuk mendeteksi gambar sidik jari pada permukaan dan dibawah permukaan jari sebagai cahaya yang dipantulkan yang memiliki banyak panjang gelombang iluminasi. Dalam metode ini, cahaya disinari dengan kondisi *total internal reflection* yang rusak. Jadi cahaya memantulkan permukaan jari dan permukaan dibawah jari. Penggambaran metode ini dapat dilihat pada Gambar 2.4 sebagai berikut.



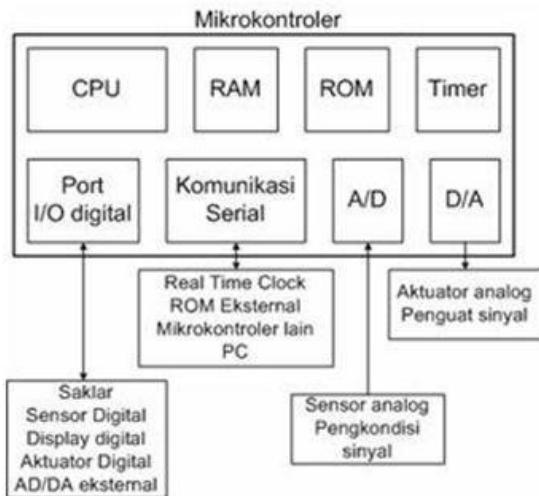
**Gambar 2. 4 Multispectral Method**

Metode ini menggunakan informasi gabungan dari permukaan dan bawah permukaan jari, oleh karena itu dapat diperkirakan kontribusi permukaan jari lebih tinggi, gambar yang diperoleh dapat dipengaruhi oleh keadaan permukaan jari.

### 2.3 Mikrokontroler

Mikrokontroler adalah sebuah komputer kecil di dalam satu *integrated circuit (IC)* yang digunakan untuk menjalankan suatu tugas dan program. Mikrokontroler sangat umum dalam sistem elektronika modern, penggunaannya sangat luas dalam kehidupan sehari-hari baik di rumah, kantor, dan sekolah. Mikrokontroler dapat digunakan untuk berbagai aplikasi misalnya untuk sistem otomasi, sistem akuisisi data, telekomunikasi, perbankan, dan telekomunikasi. Keuntungan menggunakan mikrokontroler yaitu harga yang murah, dapat di program berulang kali, dan bentuk yang simpel.

Dalam strukturnya, komponen mikrokontroler dibagi menjadi CPU, RAM, ROM, Timer, Port I/O Digital, Komunikasi Serial, A/D dan D/A. Komponen pada mikrokontroler dapat dilihat pada Gambar 2.5 sebagai berikut.



**Gambar 2. 5 Komponen Mikrokontroler**

Pada sebuah mikrokontroler terdapat komponen kerja yang saling berintegrasi untuk dapat menjalankan sebuah sistem. Komponen yang ada di dalam sebuah mikrokontroler adalah sebagai berikut:

**A. CPU (*Central Processing Unit*)**

CPU pada mikrokontroler berupa mikroprosesor yang berfungsi sebagai otak dari mikrokontroler. Mikroprosesor berfungsi untuk memproses data yang berupa fungsi logika dan aritmatika. Pekerjaan utama dari CPU adalah mengerjakan program yang dibuat oleh programmer.

**B. Memori**

Dalam suatu mikrokontroler biasanya terdapat tiga buah memori, yaitu RAM, ROM dan EEPROM. RAM digunakan sebagai penyimpanan data sementara yang berupa register-register. Register adalah tempat penyimpanan data yang berkaitan dengan variabel dalam program, keadaan input/output, ataupun pengaturan *timer/counter*. ROM digunakan sebagai tempat penyimpanan program. ROM yang banyak dipakai pada mikrokontroler saat ini adalah flash PEROM.

(*Programmable Erasable ROM*), yang mirip seperti memori pada flash disk, namun perbedaannya adalah flash PEROM hanya dapat dihapus dan ditulis sekaligus. EEPROM biasanya digunakan untuk menyimpan data yang tidak boleh hilang meski catu daya dihapus/dilepas.

#### C. Counter/timer

*Counter/timer* adalah sebuah piranti yang digunakan untuk mencacah sinyal dari suatu *clock* ataupun dari suatu perintah. Jika sinyal yang dicacah berasal dari *clock* maka piranti ini berfungsi sebagai *timer*. *Timer* bisa digunakan dalam banyak hal, seperti mengatur delay dan mengukur selang waktu tertentu dalam suatu proses.

#### D. Port

Piranti antarmuka ke input/output pada mikrokontroler disebut sebagai *port*. Pada satu *port* I/O digital terdiri beberapa pin, biasanya berjumlah 8 atau satu *byte(s)*, dengan masing-masing pin dapat mentransfer satu bit data biner (logika 0 dan 1) dari/ke mikrokontroler. Beberapa standar atau protokol yang dipakai dalam komunikasi serial adalah sebagai berikut:

1. SPI (*Serial Peripheral Interface*)
2. I2C (*Inter-Integrated Circuit*)
3. UART (*Universal Asynchronous Receiver Transmitter*)
4. USART (*Universal Synchronous-Asynchronous Receiver Transmitter*)

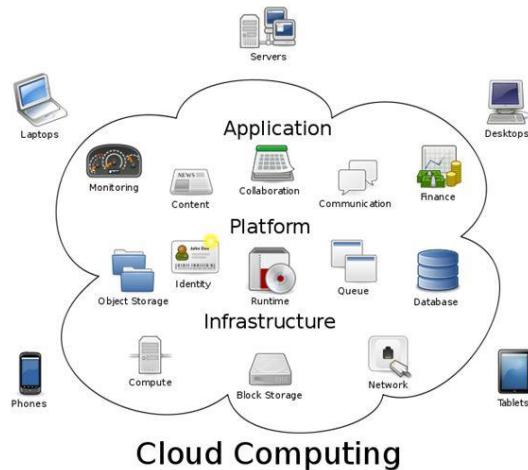
Jenis mikrokontroler berdasarkan pada tingkat kompleksitas instruksi yang diberikan, dibagi menjadi dua yaitu RISC dan CISC. RISC merupakan kependekan dari *Reduced Instruction Set Computer*. Instruksi yang dimiliki terbatas, tetapi memiliki fasilitas yang lebih banyak. CISC merupakan kependekan dari *Complex Instruction Set Computer*. Instruksi yang dimiliki lebih lengkap tapi dengan fasilitas secukupnya.

## 2.4 Cloud

### 2.4.1 Pengertian *Cloud Computing*

*Cloud computing* adalah suatu proses pengolahan daya komputasi model client-server yang melibatkan baik CPU, RAM, *Network Speed*, Software, OS ataupun *Storage* melalui jaringan (internet). Transfer data

yang dilakukan tidak secara fisik dan sumber daya komputasi yang dimiliki berada di lokasi pengguna yang memakainya, sehingga dapat diakses oleh pengguna setiap saat. Pada Gambar 2.6 berikut ini merupakan gambaran dari sistem cloud computing.



**Gambar 2. 6 Cloud Computing**

Dengan peningkatan jumlah pengguna komputer dan seluler, penyimpanan data/*data storage* telah menjadi prioritas di hampir segala bidang bisnis. Baik bisnis dengan skala besar ataupun skala kecil saat ini sangat berkembang dengan data sebagai pusatnya. Berbagai perusahaan menghabiskan biaya besar demi mempertahankan data yang berharga tersebut. Maka dari itu saat ini sangat diperlukan dukungan IT yang kuat dengan penyimpanan data yang memiliki performa tinggi. Tidak semua bisnis mampu membayar biaya tinggi untuk infrastruktur IT berjenis in-house, jadi dengan memanfaatkan *Cloud Computing* ini dapat menjadi solusi yang bagus karena lebih murah dan efisien, karena mengurangi permintaan hardware dan software dari sisi pengguna. Satu-satunya hal yang harus dijalankan oleh pengguna adalah *software interface* dari sistem *cloud computing* yang bisa jadi sesederhana web browser.

Layanan cloud popular yang telah banyak digunakan saat ini adalah semacam email tools, seperti Gmail atau Yahoo. Dalam menyimpan berkas data mereka memanfaatkan penyimpanan *Cloud*.

#### **2.4.2 Cara Kerja *Cloud Computing***

*Cloud computing* adalah seperti media penyimpanan dan media pengolahan data yang terdapat di internet, yang dapat diakses atau dilihat kapan saja dan darimana saja. *Cloud computing* tidak terikat dengan komputer ataupun gadget yang digunakan. Namun dapat diakses asalkan terhubung dengan internet.

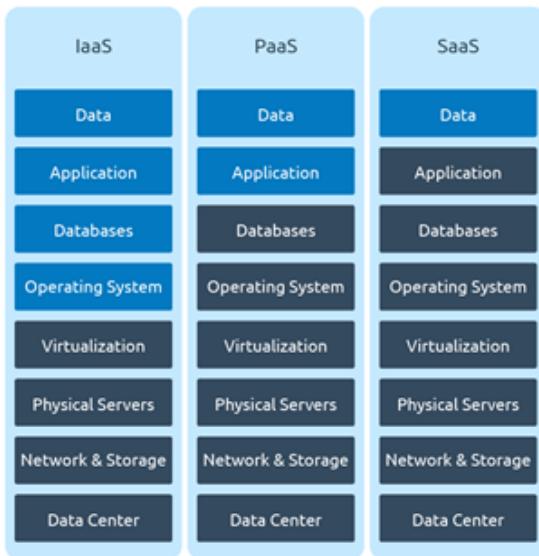
Terdapat server yang terus menerus bekerja secara online pada jaringan internet. Di server tersebut lah, semua data akan disimpan. Contohnya adalah penggunaan email pada gmail. Gmail dapat diakses asal sudah terdaftar. Saat pengguna mendaftar ataupun login, maka aktivitas pada gmail tersebut akan dikirim ke server melalui jaringan internet. Nantinya server akan memberikan respon lalu mengirim hasilnya pada pengguna.

Setelah login, server merespon dengan memberikan informasi tentang email akun pengguna. Apapun yang pengguna klik atau lakukan langsung direspon server. Data yang disimpan atau diterima juga bisa dilihat ataupun diunduh. Jika pengguna log out, pengguna masih bisa mengakses data tersebut saat nanti login kembali.

#### **2.4.3 Klasifikasi *Cloud Computing***

Terdapat berbagai klasifikasi atau pembagian pada *Cloud computing*. Menurut yang didapatkan dari beberapa referensi menyebutkan bahwa klasifikasi *Cloud computing* berbeda-beda tergantung pada parameter apa yang akan dibandingkan. Pada pembahasan ini akan dijelaskan pembagian atau klasifikasi *Cloud computing* berdasarkan *service layer* atau *delivery model* yang disediakan, serta klasifikasi *Cloud computing* berdasarkan model infrastruktur yang digunakan.

Pembagian atau klasifikasi berdasarkan service layer ini dilakukan untuk mengelompokkan fungsi-fungsi yang dilakukan pada masing-masing bagian dalam pelayanan atau penggunaan dari *Cloud computing* itu sendiri. Klasifikasi *Cloud computing* berdasarkan service layer tersebut dapat dilihat pada Gambar 2.7 sebagai berikut:



**Gambar 2. 7 Perbedaan Cloud Computing berdasarkan Service Layer**

**A. *Infrastructure-as-a-Service (IaaS)***

Meliputi seluruh penyediaan infratruktur IT seperti fasilitas data center, storage, server, grid untuk virtualized server, dan seluruh komponen networking yang ada didalam sistem cloud yang dikelola pihak ketiga.

**B. *Platform-as-a-Service (PaaS)***

Merupakan pengembangan platform dengan basis web, dimana Anda bisa menggunakan untuk membuat sebuah aplikasi web. PaaS akan membuka kesempatan bagi para developer dari berbagai tingkat pengetahuan, untuk mengembangkan aplikasi secara cepat dan murah.

**C. *Software-as-a-Service (SaaS)***

Adalah aplikasi web-based interface, yang dijalankan di sisi pihak ketiga, sehingga dapat diakses melalui jaringan oleh setiap pelanggan.

*Cloud computing* dibagi menurut model infrastruktur adalah sebagai berikut:

**A. *Private Cloud***

Layanan komputasi awan yang dioperasikan hanya untuk sebuah organisasi/perusahaan tertentu. Biasanya digunakan dalam skala yang besar. Infrastrukturnya dapat dikelola sendiri oleh organisasi atau oleh pihak ketiga. Begitu juga dengan lokasi bisa on-site atau off-site.

**B. *Community Cloud***

Infrastruktur yang dipakai bersama-sama oleh organisasi/perusahaan yang punya tujuan dan kepentingan yang sama. Jenis ini merupakan pengembangan dari private cloud. Infrastruktur dapat dikelola sebagaimana private cloud.

**C. *Public Cloud***

Jenis layanan ini disediakan untuk umum. Layanan ini disediakan oleh penyedia/provider layanan cloud.

**D. *Distributed Cloud***

Jenis layanan ini membagi perangkat keras ke beberapa lokasi tapi tetap terhubung ke jaringan yang sama atau melalui hub.

**E. *Intercloud***

Fokus dari jenis infrastruktur ini interoperabilitas langsung antara penyedia layanan *Public Cloud*, lebih daripada antara penyedia dan konsumen.

**F. *Multicloud***

Adalah penggunaan *cloud computing* dalam satu arsitektur heterogen tunggal untuk mengurangi ketergantungan pada vendor tunggal, meningkatkan fleksibilitas melalui pilihan, mitigasi terhadap bencana, dll

## 2.5 Jaringan Nirkabel

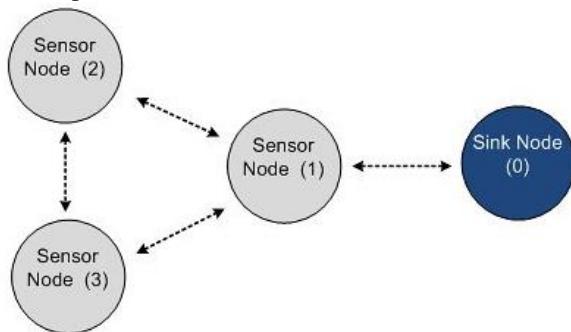
Jaringan nirkabel atau *wireless network* merupakan suatu teknologi yang memanfaatkan dua piranti untuk bertukar data tanpa perlu menggunakan kabel untuk mentransmisikan data. Jaringan nirkabel menggunakan beberapa macam media transmisi seperti gelombang radio, gelombang mikro ataupun cahaya infra merah. Ini merupakan suatu

revolusi teknologi untuk meminimalisir penggunaan kabel yang merupakan salah satu faktor penyebab adanya error atau loss pada jaringan.

Berdasarkan jarak jangkauan transmisi datanya, jaringan nirkabel dikelompokan menjadi 4, yaitu WPAN (*Wireless Personal Area Network*), WLAN (*Wireless Local Area Network*), WMAN (*Wireless Metropolitan Area Network*) dan WWAN (*Wireless Wide Area Network*).

### 2.5.1 Jaringan Sensor Nirkabel

Dalam pembahasan Tugas Akhir ini akan berfokus lebih banyak mengenai jaringan sensor nirkabel. Jaringan sensor nirkabel atau *wireless sensor network* (WSN) adalah jaringan yang menghubungkan perangkat-perangkat seperti sensor node, router dan sink node. Perangkat ini terhubung secara ad-hoc dan mendukung komunikasi multi-hop. Istilah ad-hoc merujuk pada kemampuan perangkat untuk berkomunikasi satu sama lain secara langsung tanpa memerlukan infrastruktur jaringan seperti router atau akses point. Sedangkan multi-hop yaitu istilah yang merujuk pada komunikasi beberapa perangkat yang melibatkan perangkat antara (*intermediate*), multi-hop melibatkan perangkat antara seperti router untuk meneruskan sebuah paket dari satu node ke node lain dalam jaringan. Secara sederhana pengertian WSN yang dimaksud, disajikan dalam arsitektur pada Gambar 2.8 berikut:



**Gambar 2.8** Contoh Arsitektur Dasar WSN

Pada Gambar 2.8 WSN terdiri dari tiga Sensor Node dan sebuah Sink yang terhubung dan berkomunikasi melalui gelombang radio. Sensor

Node 2 dan Sensor Node 3 dapat berkomunikasi secara langsung (ad-hoc). Sementara jika ingin berkomunikasi dengan Sink, keduanya hanya dapat mengirim paket datanya melalui Sensor Node 1, dan Sensor Node 1 yang akan meneruskan paket tersebut ke Sink Node. Pada Gambar 2.8 ini Sensor Node 1 sekaligus bertindak sebagai perantara (router) untuk menyediakan komunikasi multi-hop. Sebuah aplikasi WSN seringkali melibatkan banyak node (ratusan hingga ribuan).

Pada awal pengembangan teknologi WSN ini secara khusus digunakan dalam bidang militer, tetapi seiring perkembangan zaman, saat ini WSN telah digunakan diberbagai bidang serta untuk berbagai macam keperluan diantaranya untuk pemantauan kondisi lingkungan (kelembaban dan suhu), pemantauan aktivitas gunung berapi (deteksi gempa), perancangan *smart city*, pemantauan aktivitas di bidang pertanian, pemanfaatan teknologi sebagai sarana memajukan pendidikan, dan sebagainya.

## 2.6 *Internet of Things* (IoT)

### 2.6.1 Pengertian IoT

*Internet of Things* adalah suatu konsep teknologi modern yang memanfaatkan konektivitas jaringan internet dalam penggunaannya. *Internet of Things* atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. IoT menggambarkan sebuah dunia di mana apa saja dapat dihubungkan dan berkomunikasi dengan cara yang cerdas. Dengan kata lain, dengan *Internet of Things*, mengubah dunia fisik menjadi salah satu sistem informasi yang besar.

Dalam pengembangannya IoT sangat didukung oleh banyak pihak, meskipun masih ada pro dan kontra. Salah satu pemanfaatan IoT yang didukung oleh pemerintah adalah penerapan *Smart City*. Beberapa contoh penerapan *Smart City* di berbagai wilayah di Indonesia yaitu:

#### A. Informasi Banjir Online.

Selain mengandalkan laporan warga, sensor-sensor banjir yang dapat mengukur ketinggian air secara real-time disebarluaskan ke seluruh wilayah kota sehingga informasi dapat diteruskan langsung ke *Command Center* secara cepat dan selanjutnya langsung dapat ditangani oleh Dinas terkait.

## B. Sistem Notifikasi Gempa dan Tsunami.

Beberapa kejadian bencana alam di Indonesia memakan korban jiwa begitu banyak. Jumlah korban jiwa dapat dikurangi secara signifikan apabila *Early Warning System* diterapkan secara benar dan tepat sasaran. Sensor-sensor yang ditempatkan di daerah rawan bencana alam dapat memberikan informasi secara langsung kepada warga sekitar lokasi rawan gempa, longsor, atau tsunami dalam hitungan detik.

## C. Smart Parking.

Pada sistem ini, sensor parkir ditaruh di tempat parkir umum. Pemakaian sistem *Smart Parking* ini dapat membantu pemerintah kota memantau dan mengendalikan pendapatan daerah dari parkir. Keuntungan yang bisa dirasakan langsung oleh masyarakat berupa pemeriksaan status dari parkir yang tersedia dan sistem bayar parkir online.

### 2.6.2 Cara Kerja IoT

Cara Kerja *Internet of Things* yaitu dengan memanfaatkan argumentasi pemrograman yang dimana tiap-tiap perintah argumennya itu menghasilkan sebuah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan manusia dan dalam jarak berapa pun. Internet menjadi penghubung di antara kedua interaksi mesin tersebut, sementara manusia hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung.

Tantangan terbesar dalam mengkonfigurasi *Internet of Things* bagi seorang *engineer* adalah menyusun jaringan komunikasinya sendiri, yang dimana jaringan tersebut sangatlah kompleks, dan memerlukan sistem keamanan yang ketat. Gambaran cara kerja IoT beserta dengan elemen utamanya dapat dilihat pada Gambar 2.9 sebagai berikut.



Gambar 2.9 Elemen Utama IoT

Konsep IoT cukup sederhana dengan cara kerja mengacu pada 3 elemen utama pada arsitektur IoT, yaitu barang fisik (*things*) yang dilengkapi modul IoT, perangkat koneksi internet (modem atau router wifi), serta *cloud server* untuk menyimpan aplikasi dan database.

### 2.6.3 Arsitektur IoT

Terdapat banyak referensi terkait layer apa saja yang ada pada suatu sistem IoT, dalam hal ini disebutkan terdapat 4 layer yaitu:

#### A. Layer 1: IoT Devices

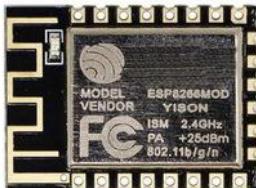
Merupakan alat/*things* yang akan dihubungkan, biasanya berupa sensor (sensor kelembaban, kedalaman, jarak, sidik jari, dll) ataupun dapat berupa aktuator (relay, kipas, lampu, dll); seperti pada Gambar 2.10 berikut ini.



**Gambar 2. 10** Layer 1: IoT Devices (sensor dan aktuator)

#### B. Layer 2: IoT Gateway

Pada layer ini adalah tempat bagi *device* yang menghubungkan antara alat/*things* ke *cloud* (internet). *Device* tersebut dapat berupa modul wifi (NodeMCU esp8266), Bluetooth, dsb. Salah satu contoh modul wifi dapat dilihat pada Gambar 2.11 berikut.



**Gambar 2. 11** Layer 2: IoT *Gateway* (modul wifi)

C. Layer 3: *Cloud*

Data yang telah didapatkan dari sensor, dikirimkan ke *cloud server* menggunakan modul wifi sebagai *gateway*. Server ini berfungsi untuk menrima, menyimpan dan memproses data untuk dianalisis dan dilakukan pengambilan keputusan.

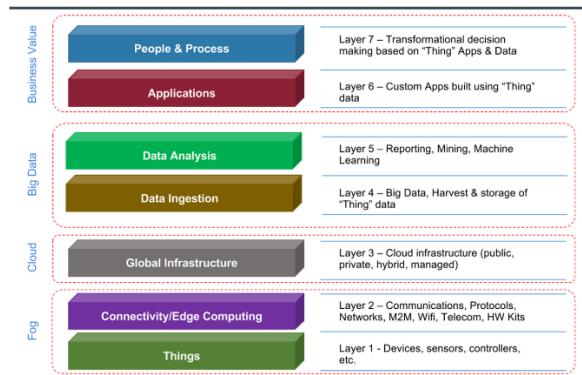


**Gambar 2. 12** Layer 3: *Cloud*

D. Layer 4: *Analytics*

Data raw yang berada di *cloud* akan dikonversi menjadi suatu tampilan/*interface*. Pada layer ini data yang telah disimpan di *cloud* akan ditampilkan, tampilan data tersebut dapat berbasis aplikasi ataupun berbasis web. Beberapa referensi menyebutkan bahwa dalam pengoperasian atau pengambilan tindakan dalam web tersebut merupakan bentuk dari layer 5 IoT.

Dalam referensi lain disebutkan bahwa terdapat 7 layer dalam arsitektur IoT yang dibagi menjadi 4 bagian, dapat dilihat pada Gambar 2.13 sebagai berikut.



Gambar 2. 13 7 Layer IoT

Sesuai pada Gambar 2.13 bahwa terdapat 7 layer IoT yang dibagi ke dalam 4 bagian diantaranya:

1. **Fog**: terdiri dari layer *Things* dan *Connectivity*.
2. **Cloud**: terdiri dari layer *Global Infrastructure*.
3. **Big Data**: terdiri dari layer *Data Analysis* dan *Data Ingestion*
4. **Bussiness Value**: terdiri dari layer *Application* dan *People & Process*

## 2.7 *Quality of Service (QoS)*

*Quality of Service* atau kualitas layanan adalah suatu metode pengukuran yang digunakan untuk menentukan seberapa baik kemampuan atau kualitas sebuah jaringan, dengan tujuan untuk memberikan pelayanan jaringan yang terbaik sehingga dapat memenuhi kebutuhan suatu layanan. QoS menawarkan kemampuan untuk mendefinisikan atribut-atribut layanan jaringan yang disediakan, baik secara kualitatif maupun kuantitatif. Pada Tabel 2.1 berikut adalah nilai persentase dari QoS.

**Tabel 2. 1** Persentase dan Nilai QoS

No	Nilai	Persentase (%)	Indeks
1.	<b>3,8 – 4</b>	95 – 100 %	Sangat Memuaskan
2.	<b>3 – 3,79</b>	75 – 94,75 %	Memuaskan
3.	<b>2 – 2,99</b>	50 – 74,75 %	Kurang Memuaskan
4.	<b>1 – 1,99</b>	25 – 49,75 %	Jelek

### 2.7.1 *Throughput*

*Throughput* adalah kecepatan (rate) transfer data yang diukur dalam satuan bps (bit per second). *Throughput* merupakan jumlah total paket datang dengan sukses yang diamati selama interval waktu tertentu. Kategori pada *throughput* dijelaskan pada Tabel 2.2 berikut.

**Tabel 2. 2** Kategori Throughput

No.	Kategori Throughput	Throughput (bps)	Indeks
1.	<b>Sangat Bagus</b>	100 ms	4
2.	<b>Bagus</b>	75 ms	3
3.	<b>Sedang</b>	50 ms	2
3.	<b>Jelek</b>	<25 ms	1

Persamaan perhitungan *Throughput*:

$$\text{Throughput} = \frac{\text{Total paket data diterima (bit)}}{\text{Lama waktu pengiriman (s)}} \quad (2.1)$$

### **2.7.2 Delay**

Delay (Latency) merupakan waktu tunda yang terjadi karena proses transmisi data. Delay dipengaruhi oleh beberapa faktor diantaranya jarak, media fisik dan waktu proses yang dibutuhkan. Satuan dari delay adalah second (s). Semakin kecil delay berarti proses transmisi data semakin baik, hal ini dijelaskan dalam kategori delay pada Tabel 2.3 berikut.

**Tabel 2. 3 Kategori Delay**

<b>No</b>	<b>Kategori Delay</b>	<b>Besar Delay (ms)</b>	<b>Indeks</b>
1.	<b>Sangat Bagus</b>	<150 ms	4
2.	<b>Bagus</b>	150 s/d 300 ms	3
3.	<b>Sedang</b>	300 s/d 450 ms	2
4.	<b>Jelek</b>	>450 ms	1

Persamaan perhitungan *Delay*:

$$\textit{Delay} = \text{Waktu diterima (s)} - \text{Waktu terkirim (s)} \quad (2.2)$$

### **2.7.3 Jitter**

Jitter adalah variasi delay antar paket yang diakibatkan oleh panjang antrian dalam suatu pengolahan data dan penghimpunan ulang paket-paket data diakhir pengiriman akibat kegagalan sebelumnya. Delay antrian pada router dan switch menyebabkan jitter. Semakin besar beban trafik atau nilai variasi delay dalam suatu jaringan akan menyebabkan semakin besar pula peluang terjadinya tumbukan antar paket, sehingga nilai jitter akan semakin besar. Kategori jitter ditunjukkan pada Tabel 2.4 berikut.

**Tabel 2. 4** Kategori Jitter

No	Kategori Jitter	Nilai Jitter (ms)	Indeks
1.	<b>Sangat Bagus</b>	0 ms	4
2.	<b>Bagus</b>	0 s/d 75 ms	3
3.	<b>Sedang</b>	75 s/d 125 ms	2
4.	<b>Jelek</b>	125 s/d 225 ms	1

Persamaan perhitungan *Jitter*:

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\text{Total paket yang diterima}} \quad (2.3)$$

#### 2.7.4 Packet Loss

Packet Loss merupakan sebuah parameter yang menunjukkan kegagalan dalam proses transmisi paket data untuk mencapai tujuan. Terdapat beberapa faktor yang menyebabkan kegagalan proses transmisi paket data tersebut, diantaranya adalah:

1. Terjadinya overload trafik dalam jaringan.
2. Tumbukan (*congestion*) dalam jaringan.
3. Error yang terjadi pada media fisik.
4. Kegagalan yang terjadi pada sisi penerima.

Pembagian kategori pada packet loss ditunjukkan dalam Tabel 2.5 berikut.

**Tabel 2. 5** Kategori Packet Loss

No	Kategori Degredasi	Packet Loss (%)	Indeks
1.	<b>Sangat Bagus</b>	0 %	4
2.	<b>Bagus</b>	3 %	3

No	Kategori Degredasi	Packet Loss (%)	Indeks
3.	Sedang	15 %	2
4.	Jelek	25 %	1

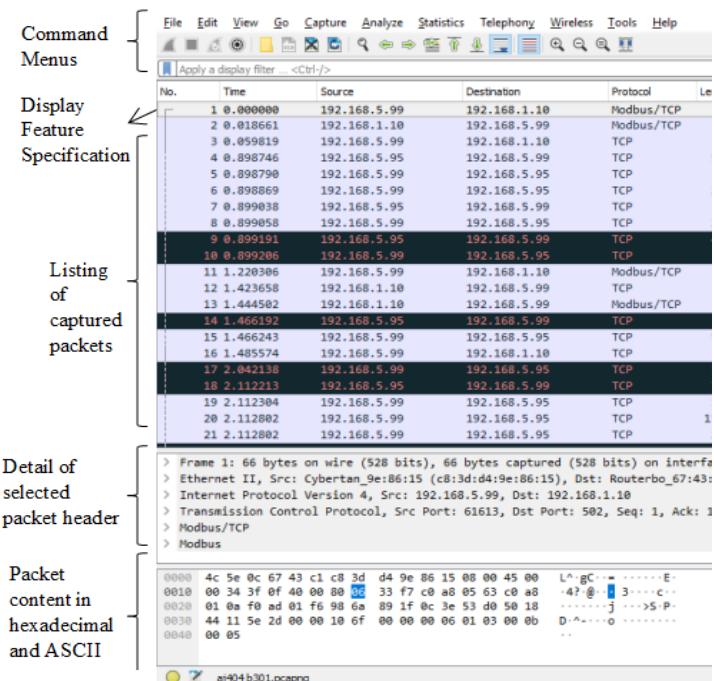
Persamaan perhitungan *Packet Loss*:

$$\text{Packet Loss} = \frac{(\text{Paket data dikirim} - \text{Paket data diterima}) \times 100\%}{\text{Paket data yang dikirim}} \quad (2.4)$$

## 2.8 Wireshark

Wireshark adalah sebuah perangkat lunak atau aplikasi *Network Protocol Analyzer* yang bersifat *free* dan *open source*. Aplikasi ini digunakan untuk melakukan *troubleshooting* jaringan yang lengkap, analisa paket data serta pengembangan protokol komunikasi dan pendidikan. Pada awal pembuatan, aplikasi ini bernama Etheral, kemudian pada Mei 2006 berganti nama menjadi Wireshark karena masalah merek dagang. Bahasa pemrograman yang digunakan pada aplikasi ini adalah bahasa C dengan lisensi public.

Wireshark banyak disukai karena interfacenya yang menggunakan Graphical User Interface (GUI) atau tampilan grafis. Tampilan ini mudah untuk dipahami bahkan untuk pemula sekalipun, karena bahasa-bahasa yang digunakan masih umum sehingga dapat dimengerti. Tampilan Wireshark dapat dilihat pada Gambar 2.14 sebagai berikut.

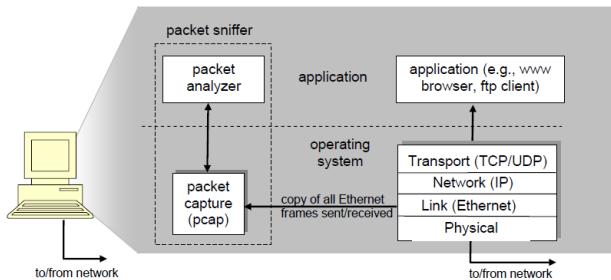


**Gambar 2. 14 Struktur GUI Software Wireshark**

Wireshark mampu menangkap paket-paket data atau informasi yang berjalan dalam suatu jaringan yang terlihat dan semua jenis informasi ini dapat dengan mudah dianalisa memakai sniffing, dengan sniffing ini dapat diperoleh informasi penting seperti password email account milik orang lain. Wireshark dapat membaca data secara langsung dari Ethernet, Token-Ring, 802.11 Wireless LAN dan koneksi ATM.

Secara garis besar cara kerja Wireshark dibagi menjadi dua tahapan, yaitu merekam semua paket yang melewati interface yang dipilih (perangkat penghubung antar jaringan, dapat berupa wifi atau Ethernet dan lain sebagainya), serta menganalisa hasil rekaman. Dalam menganalisa hasil rekaman tersebut dapat dilakukan filter atau pemilihan terhadap protokol apa yang diinginkan, seperti protokol tcp, http, udp dan

sebagainya. Struktur dari *Packet Sniffer* pada Wireshark dapat dilihat pada Gambar 2.15 sebagai berikut.



Gambar 2. 15 Struktur Packet Sniffer

Wireshark juga dikenal sebagai *packet sniffer*, yaitu sebagai sebuah program atau tool yang memiliki kemampuan untuk mencegah dan melakukan pencatatan terhadap *traffic* data dalam jaringan. Selama terjadi aliran data, *packet sniffer* dapat menangkap *protocol data unit* (PDU), melakukan decoding serta melakukan analisis terhadap isi paket berdasarkan spesifikasi RFC atau spesifikasi-spesifikasi yang lain dengan struktur dapat dilihat pada Gambar 2.15. Wireshark diprogram sedemikian rupa untuk dapat mengenali berbagai macam protokol jaringan.

## 2.9 Bahasa Pemrograman Web

Pemrograman web adalah suatu kegiatan menciptakan atau membuat program aplikasi berbasis web menggunakan bahasa pemrograman sehingga dapat memproses data dan menghasilkan informasi yang diinginkan. Dalam pemrograman web, program atau aplikasi yang telah selesai dapat ditampilkan dan diakses melalui browser. Bahasa pemrograman web terdiri dari berbagai unsur bahasa, beberapa bahasa utama yang biasa digunakan dalam membuat website adalah sebagai berikut:

### 2.9.1 HTML (*Hypertext Markup Language*)

Merupakan bahasa pemrograman web atau markup internet yang berasal dari kombinasi antara teks dan informasi berupa simbol atau kode yang akan dimasukkan ke dalam suatu file atau disebut dengan tag HTML, tag ini berfungsi untuk memberitahukan browser untuk

menampilkan sebuah desain layout dari teks tersebut. HTML merupakan bahasa markup standar untuk pembuatan halaman website.

#### **2.9.2 SQL (*Structured Query Language*)**

Merupakan suatu bahasa pemrograman yang khusus digunakan untuk mengakses data-data yang ada dalam sebuah database relasional, berfungsi sebagai pengatur pertukaran data antara aplikasi atau program dengan database tempat penyimpanan data. SQL menggunakan standar ANSI (*American National Standard Institute*).

#### **2.9.3 PHP (*Hypertext Preprocessor*)**

Merupakan bahasa pemrograman *server side scripting* yang dirancang untuk membuat dan mengembangkan web. *Server side scripting* sendiri adalah bahasa pemrograman yang berjalan di sisi server, jadi server juga ikut bekerja apabila bahasa pemrograman tersebut digunakan. PHP merupakan bahasa pemrograman yang sangat populer dan banyak digunakan oleh para developer. Contoh penggunaan PHP pada sebuah website yaitu pada form login, PHP bekerja dengan membandingkan email dan password yang dimasukkan apakah sesuai dengan data yang ada pada database.

#### **2.9.4 CSS (*Cascading Style Sheet*)**

CSS merupakan bahasa pemrograman yang berisi kumpulan kode-kode dengan tujuan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup (seperti HTML). CSS berfungsi untuk memisahkan konten dari tampilan visualnya di suatu situs. CSS dan HTML mempunyai keterkaitan yang erat, karena HTML adalah bahasa markup yang merupakan pondasi situs, sedangkan CSS memperbaiki style untuk semua aspek yang terkait dengan tampilan website.

#### **2.9.5 Java**

Merupakan sebuah bahasa pemrograman dasar dalam sebuah pembuatan software atau aplikasi. Java dapat dijalankan di berbagai komputer ataupun berbagai gadget. Salah satu penggunaan terbesar Java adalah dalam pembuatan aplikasi *native* untuk Android. Bahasa dalam Java banyak mengadopsi sintaksis yang terdapat pada bahasa C dan C++.

#### **2.9.6 JavaScript**

Merupakan bahasa pemrograman yang bersifat *Client side scripting*, yaitu tipe bahasa pemrograman yang pemrosesannya dilakukan

oleh client. Aplikasi client yang dimaksud adalah web browser seperti Google Chrome atau Mozilla Firefox. Salah satu fungsi dari JavaScript adalah membuat website lebih interaktif, serta untuk menyempurnakan tampilan dan sistem halaman web.

## **2.10 Komponen Penyusun Website**

Untuk membuat suatu website diperlukan komponen-komponen yang harus ada pada komputer (PC), diantaranya adalah sebagai berikut:

### **2.10.1 Web Browser**

Merupakan perangkat lunak atau software yang wajib ada dalam computer, karena untuk menjalankan suatu aplikasi web harus menggunakan web browser. Web browser berfungsi untuk menampilkan dokumen-dokumen web dalam format HTML. Web browser merupakan sebuah komponen klien atau disebut juga dengan *front-end*. Beberapa contoh diantaranya seperti Google Chrome, Mozilla Firefox, Internet Explorer, Opera.

### **2.10.2 Web Server**

Merupakan software yang memberikan layanan berbasis data dan berfungsi untuk menerima permintaan dengan protokol HTTP pada klien yang berupa web browser dan untuk mengirimkan kembali dalam bentuk halaman web (HTML). Fungsi dari web server adalah sebagai tempat untuk menyimpan dokumen-dokumen web. Semua dokumen web baik yang ditulis menggunakan *client side scripting* maupun *server side scripting* tersimpan dalam direktori utama (root) web server. Web server merupakan sebuah komponen server atau disebut juga dengan *back-end*.

### **2.10.3 Database Server**

Merupakan sebuah program yang digunakan untuk menyimpan data yang akan diolah di halaman web. Database disebut juga dengan Database Management System (DBMS), yaitu sebuah aplikasi yang bertugas untuk menjembatani user dan data-data di dalam database. DBMS merupakan kebutuhan opsional dalam pembuatan aplikasi web, keberadaannya sangat tergantung dengan kebutuhan web yang akan dibuat, apakah pada web tersebut memerlukan untuk menyimpan data di database atau tidak.

#### **2.10.4 Web Editor**

Merupakan sebuah program yang memiliki fungsi untuk memasukkan perintah-perintah dokumen baik yang diketik secara *client side scripting* ataupun *server side scripting*. Contoh editor paling sederhana adalah dapat menggunakan Notepad atau Notepad++.

### **2.11 Firebase**

Firebase merupakan salah satu fitur atau layanan yang disediakan oleh Google untuk memudahkan app developer mengembangkan aplikasi. Firebase memiliki cukup banyak fitur yang bisa digunakan untuk mengembangkan aplikasi baik Android, iOS, Web dan lainnya. Fasilitas yang dimiliki oleh Firebase ini diantaranya adalah Realtime Database yang pertama kali pernah dikembangkan, berbasis Cloud, layanan yang diberikan dapat diakses secara gratis dalam waktu yang tak terbatas, Firebase sudah menggunakan JSON jadi tidak perlu lagi menggunakan SQL, serta Firebase bisa digunakan untuk Android, iOS, Java, Objective-C, Swift, Node.Js, dan JavaScript.

Firebase memudahkan dalam pembuatan aplikasi, cepat dan tanpa perlu memikirkan infrastruktur karena Firebase sendiri memiliki fungsi seperti analisis, basis data, pengiriman pesan, dan pelaporan kerusakan sehingga memudahkan para pengembang dan dapat lebih terfokus pada pengguna.



**Gambar 2. 16** Firebase

Firebase memiliki keunggulan yaitu menangani detail dari *backend infrastructure* pada aplikasi yang dibuat, pemakaian pengguna, dan *monetizing* sehingga pengembang dapat lebih fokus untuk membangun aplikasinya tanpa perlu memikirkan hal-hal tersebut.

### **2.11.1 Realtime Database dan Cloud Firestore**

Firebase memiliki 2 jenis layanan database yang berbasis cloud dan keduanya memungkinkan sinkronisasi data secara real-time, yaitu *Realtime Database* dan *Cloud Firestore*. *Realtime Database* merupakan layanan database pertama milik firebase yang berbasis cloud, sedangkan *Cloud Firestore* merupakan layanan database terbaru milik Firebase yang masih dalam versi beta.

Dilihat dari data model, *Realtime Database* menyimpan data sebagai satuan JSON *tree* yang besar dan memudahkan untuk menyimpan data, tetapi sulit untuk mengatur data yang kompleks. Sedangkan pada *Cloud Firestore* data yang disimpan di dokumen telah dikelompokkan, sehingga data yang kompleks dapat diatur dengan mudah.

Data dapat dikelompokkan dan difilter dari database melalui *queries*. *Realtime Database* memiliki *queries* yang terbatas dalam penggunaannya. Dalam *sngle query* hanya dapat dilakukan salah satu diantara sortir atau filter. Sedangkan pada *Cloud Firestore* dapat melakukan keduanya sekaligus. *Cloud Firestore* memiliki fitur *query* yang lebih lengkap dan lebih cepat, serta penskalaan yang lebih mendalam dibandingkan dengan *Realtime Database*.

*Realtime Database* merupakan layanan yang lebih dulu hadir di Firebase, sehingga secara stabilitas sudah sangat teruji, *Realtime Database* merupakan solusi dari jaringan dengan latensi yang rendah dan efisien untuk aplikasi seluler yang membutuhkan status tersinkronisasi di seluruh klien secara real-time. Sedangkan *Cloud Firestore* masih dalam versi beta sehingga stabilitasnya masih dalam tahap pengujian, belum sempurna.

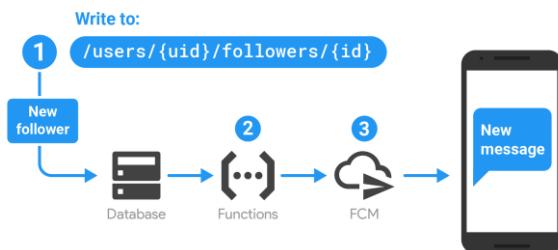
Secara keseluruhan, memungkinkan untuk menggunakan kedua database bersamaan dalam sebuah *project* Firebase. Keduanya sama-sama dapat menyimpan data yang sama. Dalam pemilihan penggunaan database disesuaikan dengan kebutuhan.

### **2.11.2 Cloud Functions**

*Cloud functions* merupakan salah satu layanan yang disediakan oleh firebase. *Cloud functions* Firebase berjalan di *environment* Node.js yang *hosted*, bersifat privasi, dan bisa diubah ukurannya, yang dapat menjalankan kode JavaScript. Dengan menggunakan *Cloud functions*, developer dapat menjalankan kode back-end secara otomatis sebagai respons terhadap berbagai perintah (peristiwa) yang dijalankan atau dipanggil oleh fitur Firebase dan permintaan HTTP(S). *Cloud functions*

memberi developer akses ke peristiwa Firebase dan Google Cloud, beserta daya komputasi yang skalabel untuk menjalankan kode sebagai respons terhadap peristiwa tersebut. Kelebihan *Cloud functions* secara sederhana adalah mengurangi kompleksitas yang terjadi saat menjalankan dan mengatur back-end dan server.

Salah satu contoh penggunaan *Cloud functions* ini adalah memberi tahu pengguna jika ada sesuatu yang menarik dalam sebuah layanan aplikasi. Developer dapat menggunakan *Cloud functions* untuk selalu berinteraksi dengan pengguna dan memberikan informasi tentang suatu aplikasi kepada pengguna. Misalnya, dalam suatu aplikasi, sebuah fungsi yang dipicu oleh peristiwa dalam *Realtime Database* untuk menyimpan pengikut baru, dapat membuat notifikasi Firebase Cloud Messaging (FCM) yang akan memberi tahu pengguna bahwa mereka memiliki pengikut baru dalam aplikasi tersebut. Contoh penggunaan dari Cloud Functions dapat dilihat pada Gambar 2.17 sebagai berikut.



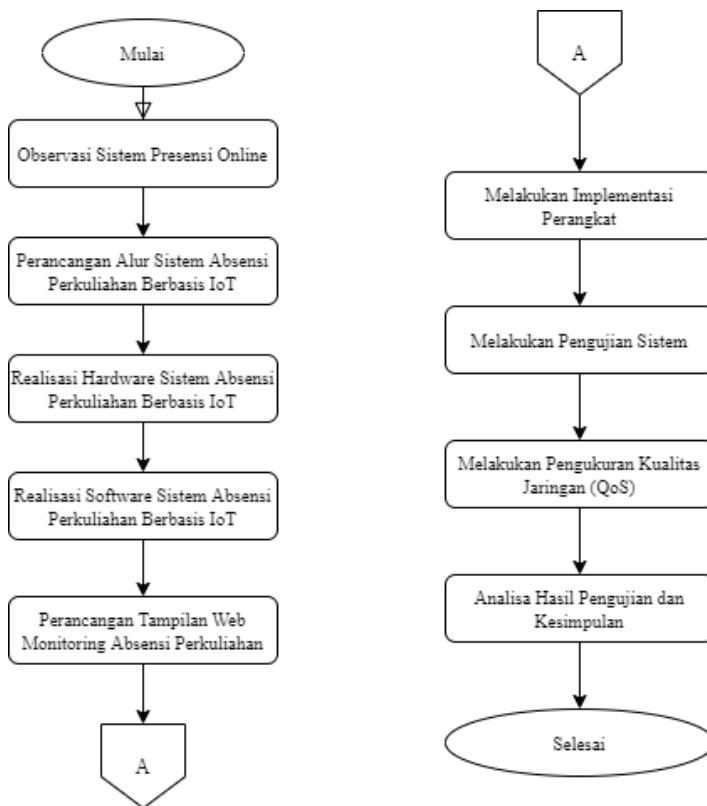
**Gambar 2. 17** Contoh penggunaan *Cloud functions*

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Diagram Alir

Dalam pelaksanaan tugas akhir ini akan dilakukan beberapa langkah yang dijelaskan oleh diagram alir. Pada Gambar 3.1 berikut adalah diagram alir dalam pembuatan sistem absensi perkuliahan berbasis *Internet of Things*.



Gambar 3. 1 Diagram Alir Penelitian

### **3.2 Permasalahan Sistem Presensi *Online***

Sistem presensi di berbagai institusi memiliki metode yang berbeda-beda dan sesuai dengan kebutuhannya. Apabila dikerucutkan dalam institusi perguruan tinggi, terlebih khususnya untuk mahasiswa, sistem presensi memiliki berbagai kegunaan diantaranya dalam pengelolaan administrasi, monitoring kehadiran, ataupun sebagai syarat mengikuti ujian.

Banyak perguruan tinggi yang telah menerapkan sistem presensi dengan metode *online*, salah satunya terdapat sistem presensi yang menggunakan kartu RFID. Dari segi pengelolaan administrasi metode ini cukup baik, karena sudah memanfaatkan teknologi digital dimana data yang masuk akan langsung diolah ke database dan server secara online melalui jaringan internet. Namun dalam beberapa kasus ditemui untuk penyimpanan data masih menggunakan database dan server lokal, atau terdapat di pc yang dikelola oleh admin. Hal ini juga berlaku pada sistem presensi dengan menggunakan QR Code atau berbasis android. Metode-metode ini sangat baik dari sisi administrasi dengan memanfaatkan teknologi digital untuk mempermudah dalam pengelolaan datanya, akan tetapi masih memiliki celah kekurangan di sisi keamanan. Manipulasi data masih bisa dilakukan, karena masih memungkinkan untuk memberikan kartu RFID ke ataupun scan QR Code ke orang lain.

Berbagai kekurangan tersebut dapat diminimalisir dengan menggunakan metode presensi yang bersifat biometrik, dimana setiap orang memiliki ciri biometrik yang berbeda-beda. Dalam tugas akhir ini sistem biometrik yang digunakan adalah sidik jari. Setiap manusia memiliki sidik jari yang tidak sama satu dengan yang lain, sehingga kemungkinan untuk menitipkan atau manipulasi sangat kecil.

Dengan memanfaatkan teknologi digitalisasi dan biometrik, sistem pengelolaan administrasi akan lebih mudah dan proses pengumpulan data presensi akan lebih valid. Maka pada tugas akhir ini akan dibuat rancangan sistem presensi menggunakan sensor sidik jari dengan database yang dibuat secara *online* sehingga data akan mudah dikelola dan lebih aman dengan fasilitas layanan keamanan dari penyedia.

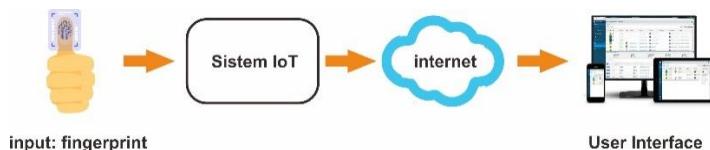
### **3.3 Desain Sistem Absensi Berbasis *IoT***

#### **3.3.1 Konsep Sistem Absensi Berbasis *IoT***

Sistem yang akan dibuat dalam tugas akhir ini adalah sebuah sistem absensi untuk perkuliahan dengan menggunakan konsep teknologi

*Internet of Things*. Komponen utama dalam pembuatan sistem ini adalah Mikrokontroler yang berfungsi sebagai pengatur kerja seluruh perangkat keras pada sistem. Kemudian data dikirimkan ke modul IoT yang memiliki peran dalam pengiriman data ke server melalui jaringan internet. Mikrokontroler membaca data yang diperoleh dari sensor sidik jari, kemudian dikirim ke database cloud melalui modul IoT, untuk menghasilkan data absensi kehadiran mahasiswa (pengguna) setiap harinya sesuai mata kuliah yang diambil. Kemudian selanjutnya pada database cloud, data akan diolah lebih lanjut sehingga menjadi tampilan antarmuka website.

Dari sisi perangkat lunak monitoring absensi, tampilan dari antarmuka aplikasi dibuat berbasis website sehingga dapat dibuka di berbagai device (pc/mobile). Dalam pembuatan database akan dilakukan melalui web, yaitu dengan mendaftarkan atau registrasi sidik jari sesuai dengan daftar nama yang telah ada dari administrasi. Adapun yang memiliki hak akses untuk dapat melakukan registrasi hanyalah admin, sedangkan mahasiswa atau pengguna hanya dapat melakukan absensi dan pengecekan atau monitoring terhadap jadwal mata kuliahnya masing-masing. Pada Gambar 3.2 berikut adalah rancangan alur sistem absensi perkuliahan berbasis IoT yang akan dibuat:

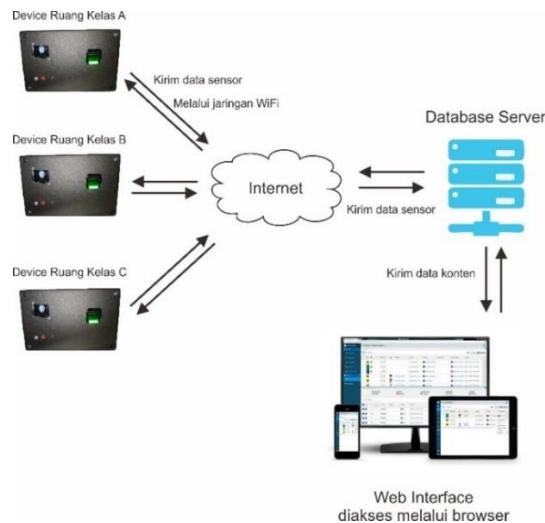


**Gambar 3.2** Rancangan Alur Sistem

Jika dilihat dari pembagian layer pada rancangan alur sistem yang dibuat, terdapat 2 gateway pada sistem absensi berbasis IoT ini. Gateway 1 terdapat pada komunikasi antara mikrokontroler dengan modul IoT. Gateway ini bersifat lokal karena dalam pengiriman datanya tidak membutuhkan koneksi internet, dapat menggunakan kabel ataupun dalam beberapa contoh telah *embedded*. Gateway 2 terdapat pada komunikasi antara modul IoT dengan *access point* atau router WiFi.

### 3.3.2 *Online Attendance System* (OASYS)

Dalam tugas akhir ini akan dibahas rancangan sistem absensi berbasis IoT yang kemudian disebut dengan OASYS. OASYS adalah kependekan dari *online attendance system*. Dalam pembuatan sistem ini, rancangan akan direalisasikan dalam bentuk *prototype device* absensi dan untuk monitoring akan ditampilkan dalam bentuk web. Komponen utama dalam pembuatan *prototype* ini adalah Mikrokontroler NodeMCU ESP8266. Mikrokontroler ini berfungsi sebagai pusat dari seluruh sistem, serta pengiriman data ke internet karena pada mikrokontroler NodeMCU ESP8266 ini telah dilengkapi dengan modul IoT. Dalam pengambilan data input digunakan sensor sidik jari FPM10A yang memiliki total sidik jari yang bisa tersimpan sebanyak 128 user. *Device* dilengkapi dengan LCD Display yang berfungsi untuk menampilkan status atau notifikasi dalam penggunaan, selain itu juga terdapat lampu LED dan buzzer yang berfungsi sebagai notifikasi dalam berbagai fitur saat digunakan. Dalam pembuatannya akan digunakan *software* Arduino IDE untuk melakukan pemrograman pada mikrokontroler sekaligus modul IoT, beserta komponen-komponen lainnya. Desain sistem yang akan digunakan pada OASYS dapat dilihat pada Gambar 3.3 berikut.



**Gambar 3. 3** Desain Sistem OASYS

Dalam menampilkan hasil absensi atau monitoring kehadiran, dibuat web yang dikostumisasi sedemikian rupa agar memiliki fitur-fitur sesuai dengan rancangan yang dibuat. Penulisan pemrograman pada web akan menggunakan bahasa JavaScript dengan bantuan *software* Notepad++. Desain tampilan web akan menggunakan Ionic Framework, dan database yang digunakan adalah layanan dari Google Cloud yaitu Firebase Cloud Firestore.

Sistem OASYS dalam penggunaan *device*-nya tidak jauh berbeda dengan absensi sidik jari pada umumnya, akan tetapi inovasi yang dilakukan dalam tugas akhir ini lebih mengedepankan pada interface pengguna, seperti kenyamanan dan kemudahan dalam penggunaan fitur-fitur yang tersedia. Dengan menggunakan Firebase yang berbasis cloud maka data yang terkirim akan disimpan pada database yang diunggah secara online sehingga data akan lebih mudah diakses untuk digunakan seakutu-waktu. Selain itu kelebihan pada Firebase adalah developer tidak perlu memikirkan bagaimana keamanan data pada back-end system sehingga developer dapat lebih berfokus untuk mengembangkan aplikasi yang dibuat.

Dari segi keamanan data, Cloud Firestore pada Firebase memiliki aturan keamanan terkait akses ke database. Dengan menerapkan *Firebase Authentication* dan *Cloud Firestore Security Rules* maka keamanan pada Firebase akan melakukan autentikasi otorisasi, serta validasi data tanpa server saat menggunakan *library* klien seluler dan web. *Firebase Authentication* memverifikasi kredensial pengguna dan memberikan landasan untuk sistem akses *user-based* dan *role-based*. Sebelum membaca atau menulis data, setiap permintaan database dari *library* klien seluler/web *Cloud Firestore* akan dievaluasi terhadap aturan keamanan yang telah dibuat. Apabila aturan tersebut menolak akses ke salah satu lokasi dokumen yang ditentukan, seluruh permintaan akan gagal. Ketika menggunakan *library* klien server, *developer* dapat mengelola akses ke *resource* dengan Identity and Access Management (IAM). IAM memungkinkan *developer* untuk mengontrol siapa (*user*) yang memiliki (*role*) izin untuk *resource* mana dengan menetapkan kebijakan IAM. Kebijakan IAM memberikan satu atau lebih peran(*role*) kepada pengguna, serta memberikan izin tertentu kepada pengguna.

Berikut ini adalah alur sistem absensi perkuliahan berbasis IoT (OASYS) yang akan dibuat, ditampilkan dalam diagram alur pada Gambar 3.4 berikut.



Gambar 3. 4 Diagram Alur Sistem

### **3.3.3 Skenario Pemanfaatan Sistem**

Perancangan sistem absensi berbasis IoT (OASYS) ini mengacu pada kondisi lingkungan yang terdapat di Departemen Teknik Elektro ITS. Rancangan sistem ditujukan kepada sejumlah mahasiswa dan dosen atau tenaga pendidikan untuk dapat menggunakannya dengan pembagian peran user-admin.

Pengguna mahasiswa (user) dirancang untuk dapat digunakan oleh seluruh mahasiswa pada Departemen Teknik Elektro ITS dengan mensinkronisasikan database yang dibuat dengan database yang telah ada di Integra. Mahasiswa dapat melihat atau monitoring kehadiran pada masing-masing mata kuliah yang diambil, selain itu juga dapat melihat jadwal perkuliahan terbaru pada setiap mahasiswa yang dilengkapi dengan keterangan waktu dimulai, waktu selesai, ruang kelas dan dosen pengajar. Sedangkan untuk pengguna dosen atau tenaga pendidikan (admin) dirancang untuk dapat digunakan oleh seluruh dosen di Departemen Teknik Elektro ITS untuk memonitoring kehadiran setiap mahasiswa yang mengikuti perkuliahannya ataupun memungkinkan dosen untuk mengubah jadwal mengajar sesuai yang dikehendaki dengan lebih mudah, selain hal tersebut sebagai tenaga pendidikan atau bagian Tata Usaha juga dapat menggunakan sistem ini untuk keperluan menambahkan nama mahasiswa ke dalam database serta mendaftarkan dan menghapus sidik jari mahasiswa yang namanya telah terdaftar sesuai pada database.

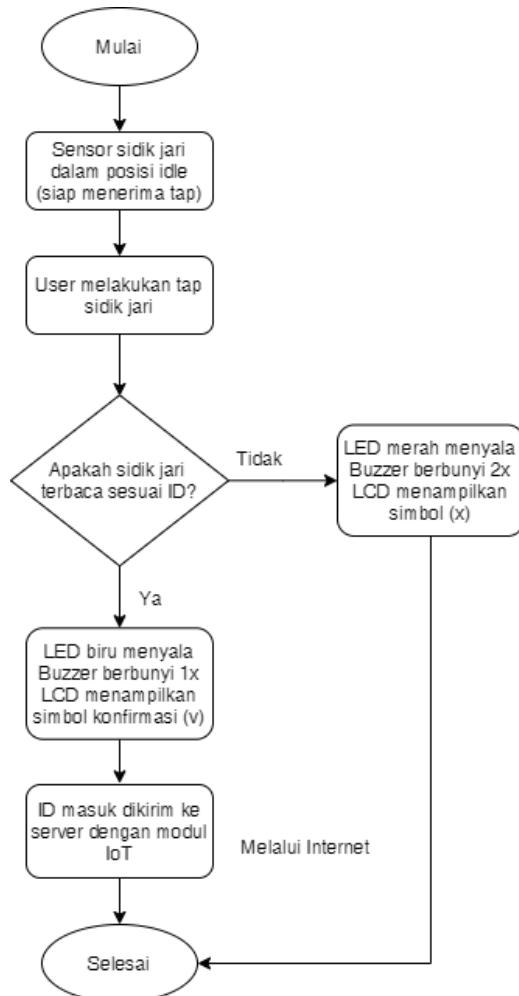
Dalam menampilkan rekapitulasi kehadiran mahasiswa pada setiap mata kuliah yang diambil, terdapat persentase kehadiran keseluruhan mata kuliah dalam satu semester, dimana persentase ini digunakan untuk mengambil keputusan apakah mahasiswa tersebut dapat mengikuti evaluasi akhir semester ataukah tidak sesuai dengan peraturan akademik yang ada di ITS yaitu dengan persentase kehadiran minimal 80% pada masing-masing mata kuliah.

## **3.4 Rancangan Perangkat Keras**

Dalam sub bab ini akan didefinisikan perangkat apa saja yang dibutuhkan dalam perancangan sistem absensi perkuliahan berbasis IoT beserta spesifikasi dari beberapa komponen yang digunakan.

### 3.4.1 Perencanaan Proses Sistem Absensi

Untuk proses atau cara kerja dari *device* absensi berbasis IoT ini dijelaskan pada diagram alur dalam Gambar 3.5 sebagai berikut.



**Gambar 3.5** Diagram Alur *Device* Absensi IoT

### **3.4.2 Pemilihan Komponen *Device***

Komponen perangkat keras yang dibutuhkan dalam membuat sistem absensi perkuliahan berbasis IoT ini adalah sebagai berikut:

#### **3.4.2.1 NodeMCU ESP8266**

Mikrokontroler yang dipakai adalah NodeMCU v3 ESP8266 CH340. Mikrokontroler merupakan komponen utama pada tugas akhir ini. Keuntungan menggunakan mikrokontroler NodeMCU ini salah satunya adalah adanya modul WiFi yang *embedded* atau terpasang menjadi satu device dengan mikrokontroler, sehingga lebih ringkas dan mudah dalam penggunaannya. Selain itu dari sisi harga juga relatif lebih murah dibanding mikrokontroler Arduino pada umumnya (Arduino Uno, Nano, Mega, dll). Karena hanya memerlukan sedikit jumlah pin yang digunakan, jadi mikrokontroler NodeMCU ESP8266 ini sudah mencukupi dalam pembuatan sistem absensi perkuliahan berbasis IoT.



**Gambar 3. 6 NodeMCU ESP8266**

Sedangkan untuk spesifikasi dari mikrokontroler ini dapat dilihat pada Tabel 3.1 sebagai berikut:

**Tabel 3. 1 Spesifikasi NodeMCU ESP8266**

No	Spesifikasi	Keterangan
1.	Mikrokontroler/Chip	ESP8266-12E
2.	Input voltage	3.3V ~ 5V
3.	GPIO	13 Pin
4.	PWM channel(s)	10 channels

No	Spesifikasi	Keterangan
5.	10 bit ADC Pin	1 Pin
6.	Flash Memory	4 MB
7.	Clock Speed	40/26/24 MHz
8.	WiFi	IEEE 802.11 b/g/n
9.	Frequency	2.4 GHz ~ 22.5 GHz
10.	USB Port	Micro USB
11.	USB Chip	CH340G

### 3.4.2.2 Sensor Sidik Jari FPM10A

FPM10A merupakan sebuah sensor sidik jari yang digunakan dalam tugas akhir ini. Komponen ini juga merupakan salah satu komponen penting dalam pembuatan sistem ini, karena sensor ini merupakan satu-satunya jalan data masuk untuk kemudian akan diolah lebih lanjut. Pada Gambar 3.7 berikut ini adalah sensor sidik jari FPM10A yang akan dipakai.



**Gambar 3. 7 Sensor FPM10A**

Sedangkan untuk spesifikasi dari sensor FPM10A dapat dilihat pada Tabel 3.2 berikut:

**Tabel 3. 2 Spesifikasi Sensor FPM10A**

No	Spesifikasi	Keterangan
1.	Power supply voltage	3.3v
2.	Supply current	< 1 seconds
3.	Working current	< 1 seconds
4.	Peak current	< 1 seconds
5.	The time of fingerprint image input	< 1 seconds
6.	The window size	14 x 18 mm
7.	Profile	256 bytes
8.	Template file	512 bytes
9.	Storage capacity	300
10.	False accept rate (FAR)	<0.001% (safety grade 3)
11.	FRR (FRR)	< 1% (safety grade 3)
12.	The search time	<1 seconds (1:500, mean)
13.	Computer interface	UART (TTL logic level)
14.	The communication baud rate (UART)	(9600 x N) BPS where N=1 ~ 12 (the default value of N=6, namely 57600bps)
15.	Working environment Temperature: Relative humidity:	-20C to +50C 40% RH to 85% RH (no treatment)
16.	Storage conditions Temperature: Relative humidity:	-40Cto +85C < 85% H (no treatment)
17.	Dimensions (L * W * H):	46.25 x 20.4 x 18.1mm

### **3.4.2.3 LCD Display**

LCD Display yang digunakan adalah OLED SSD1306 dengan ukuran 0”96 inch, resolusi 128x64 px, dan warna putih monochrome. Komponen ini berfungsi untuk menampilkan proses mulai dari pembacaan sidik jari, sampai menampilkan nama pengguna yang telah melakukan tap pada sensor. Pada Gambar 3.8 berikut ini merupakan LCD Display SSD1306 yang akan dipakai.



**Gambar 3. 8** OLED Display SSD1306 0.96” inch White

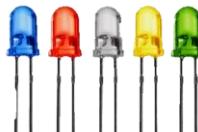
Sedangkan untuk spesifikasi dari komponen tersebut dapat dilihat pada Tabel 3.3 sebagai berikut:

**Tabel 3. 3** Spesifikasi OLED SSD1306

No	Spesifikasi	Keterangan
1.	Resolution	128 x 64
2.	Technology	OLED White Graphic
3.	Driver IC	SSD1306
4.	Wide input voltage	2.5V - 5.5V
5.	Communication	Serial I2C
6.	Wide display angle	160 degree
7.	Power consumption	0.06W
8.	PCB size	29.5x27.35mm

#### **3.4.2.4 Lampu LED**

Menggunakan 2 lampu LED berwarna biru dan merah yang memiliki fungsi atau kegunaan masing-masing. Lampu LED berwarna biru untuk menunjukkan bahwa jari yang di tap telah dikonfirmasi dan terdaftar di database (valid). Lampu LED berwarna merah menunjukkan bahwa jari tidak terbaca pada sensor (tidak valid).



**Gambar 3. 9 Lampu LED**

Spesifikasi dari komponen LED yang digunakan dapat dilihat pada Tabel 3.4 berikut:

**Tabel 3. 4 Spesifikasi Lampu LED**

No	Spesifikasi	Keterangan
1.	Color	Red, Green, Blue
2.	LED diameter	5 mm
3.	Input voltage	1.8V-2.1V DC
4.	Continous forward current	20mA
5.	Max forward current	30mA

#### **3.4.2.5 Buzzer**

Kegunaan buzzer pada rangkaian ini hampir sama seperti lampu LED, yaitu sebagai penanda apakah sidik jari sudah terkonfirmasi (terdeteksi), tidak terbaca ataupun terdapat error dalam pembacaan. Buzzer yang digunakan dapat dilihat pada Gambar 3.10 berikut.



**Gambar 3. 10 Buzzer Aktif**

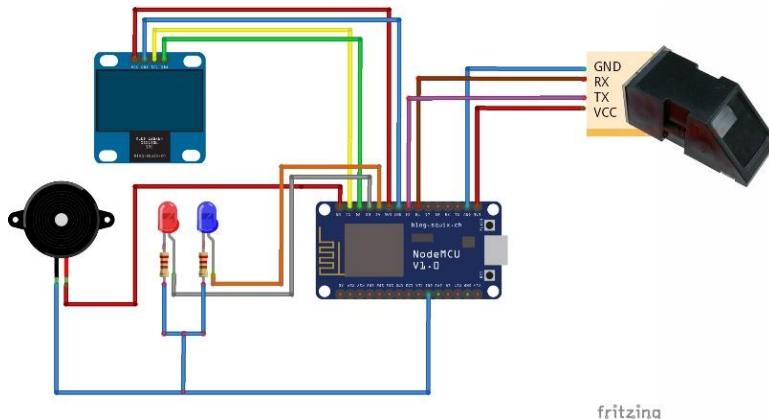
Sedangkan untuk spesifikasi pada buzzer yang digunakan dapat dilihat pada Tabel 3.5 berikut.

**Tabel 3. 5 Spesifikasi Buzzer**

No	Spesifikasi	Keterangan
1.	Rated voltage	5V DC
2.	Operating Voltage	4-8V DC
3.	Max rated current	30mA (5V DC)
4.	Min sound output	85dB (at 10 cm)
5.	Resonant frequency	2300 +/- 300Hz
6.	Operating temperature	-20C to 45C
7.	Diameter	12mm
8.	Height	9mm

### 3.4.3 Rangkaian *Device* Sistem Absensi

Setelah semua komponen perangkat keras sudah terkumpul, kemudian dirangkai sedemikian rupa untuk menjadi sebuah sistem. Rangkaian *device* yang dibuat, ditampilkan dalam Gambar 3.11 berikut:



**Gambar 3. 11 Rangkaian *Device* Absensi**

### 3.5 Rancangan Perangkat Lunak

Dalam perancangan sistem absensi berbasis IoT, selain dibutuhkan beberapa komponen perangkat keras, juga digunakan *software* atau perangkat lunak untuk memproses komponen-komponen tersebut menjadi sebuah sistem.

#### 3.5.1 *Software* Pemrograman

Beberapa *software* yang digunakan dalam pemrograman sistem ini adalah sebagai berikut:

##### A. Arduino IDE

Untuk dapat memprogram board Arduino, dibutuhkan aplikasi IDE bawaan dari Arduino. IDE merupakan kependekan dari *Integrated Development Environment*, atau lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dituliskan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang mirip dengan bahasa C. Bahasa pemrograman Arduino (Sketch) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya.

Dalam pembuatan sistem absensi IoT ini, Arduino IDE berfungsi untuk memprogram NodeMCU ESP8266, beserta berbagai komponen yang terhubung dengan menambahkan library-library yang dibutuhkan.



Gambar 3.12 Arduino IDE

## B. Notepad++

Notepad++ digunakan untuk menuliskan kode-kode pemrograman dalam bahasa yang diinginkan. Bahasa pemrograman dapat berupa C, C++, JavaScript, dan sebagainya. File kemudian disimpan dalam format .html, .ts, ataupun .css sesuai dengan jenis program yang akan dibuat. Dalam pembuatan sistem absensi berbasis IoT ini, Notepad++ digunakan untuk membuat halaman web dengan menggunakan bahasa JavaScript.

## C. Firebase

Firebase merupakan layanan dari Google yang menyediakan berbagai fitur untuk memudahkan developer dalam membuat dan mengembangkan suatu *project*. Pada tugas akhir ini memanfaatkan layanan *Cloud Firestore* milik Firebase sebagai database yang akan digunakan, serta *Cloud Functions* yang digunakan sebagai jalur komunikasi dengan *devices*.

### 3.5.2 Pembuatan Pemrograman

Dalam proses pembuatan pemrograman ini dibagi menjadi 3 bagian, yaitu pemrograman pada *hardware* yang berupa mikrokontroler menggunakan Arduino IDE, pemrograman database pada Firebase, serta pemrograman web menggunakan Notepad++.

#### 3.5.2.1 Pemrograman Pada Arduino

Dalam tugas akhir ini dilakukan pemrograman pada komponen paling utama yaitu mikrokontroler yang berupa NodeMCU ESP8266 menggunakan *software* Arduino IDE yang berbasis bahasa C. Pada pembuatan program ini, menggunakan beberapa library yang telah disediakan oleh banyak developer untuk kemudian lebih dikembangkan sesuai dengan kebutuhan. Beberapa library yang digunakan diantaranya adalah SPI.h, Wire.h, ESP8266WiFi.h, SoftwareSerial.h, ESP8266WebServer.h, ESP8266HTTPClient.h, Adafruit\_GFX.h, Adafruit\_SSD1306.h, Adafruit\_Fingerprint.h. Pada Gambar 3.12 berikut adalah pemrograman yang dibuat pada NodeMCU ESP8266 untuk mikrokontroler dan modul IoT yang digunakan. Contoh pemrograman yang dibuat pada Arduino IDE untuk proses absensi menggunakan Mikrokontroler NodeMCU ESP8266 dan modul sensor sidik jari FPM10A dapat dilihat pada Gambar 3.13 berikut.

```

// Fingerprint ID
int fingerID = 0;
String fingerName = "Unknown";

// Fingerprint ID array
String fingerNames[] = {"Ismail", "Rahman", "Unknown"};

// Fingerprint ID size
int fingerNamesSize = 3;

// Fingerprint ID index
int fingerIndex = 0;

// Fingerprint ID count
int fingerCount = 0;

// Fingerprint ID error
int fingerError = 0;

// Fingerprint ID status
String fingerStatus = "Unknown";

// Fingerprint ID message
String fingerMessage = "Unknown error";

// Fingerprint ID response
String fingerResponse = "Unknown error";

// Fingerprint ID result
String fingerResult = "Unknown error";

// Fingerprint ID result2
String fingerResult2 = "Unknown error";

```

The code handles fingerprint identification and sends the results via an HTTP POST request. It includes logic for handling errors and displaying results.

Gambar 3. 13 Pemrograman Pada Mikrokontroler dan Modul IoT

### 3.5.2.2 Pemrograman Pada Database

Pembuatan database menggunakan Firebase dari Google Cloud. Database yang dibuat dibagi menjadi 2 yaitu database mahasiswa yang berisi Nama, NRP, Mata Kuliah, Hari dan Kelas. Serta database jadwal kuliah yang berisi Mata Kuliah, Hari, Jam, Kelas, Ruang dan Dosen. Diasumsikan database telah dibuat terlebih dahulu, sehingga mahasiswa tidak dapat menambah ataupun mengurangi mata kuliah yang diambil. Isi dari database mahasiswa yang dibuat dapat dilihat pada Tabel 3.6 berikut.

Tabel 3. 6 Database Mahasiswa

No	Nama	NRP	Mata Kuliah	Hari	Kelas
1.	Ismail	0711204 0000001	1	Senin	A
			7	Selasa	A
			4	Kamis	A
			5	Jumat	A
			6	Senin	A
2.	Rahman	0711204 0000002	2	Selasa	A
			8	Rabu	A
			10	Jumat	A

No	Nama	NRP	Mata Kuliah	Hari	Kelas
3.	Fathin	0711204 0000003	6	Senin	A
			3	Rabu	A
			4	Kamis	A
			9	Kamis	A
4.	Kurnia	0711204 0000004	1	Senin	A
			7	Selasa	A
			8	Rabu	A
			5	Jumat	A
5.	Shidiq	0711204 0000005	1	Senin	B
			7	Selasa	A
			3	Rabu	A
			9	Kamis	A
6.	Arif	0711204 0000006	2	Selasa	A
			8	Rabu	A
			4	Kamis	B
			10	Jumat	A
7.	Cahyo	0711204 0000007	1	Senin	B
			3	Rabu	B
			4	Kamis	B
			10	Jumat	B
8.	Jabbar	0711204 0000008	6	Senin	A
			7	Selasa	A
			3	Rabu	B
			5	Jumat	A
9.	Fauzi	0711204 0000009	2	Selasa	A
			8	Rabu	A
			9	Kamis	A
			5	Jumat	A
10.	Arkan	0711204 0000010	6	Senin	A
			2	Selasa	A
			9	Kamis	A
			10	Jumat	B

Sedangkan isi dari database jadwal mata kuliah yang telah dibuat dapat dilihat pada Tabel 3.7 berikut.

**Tabel 3.7** Database Jadwal Mata Kuliah

No	Mata Kuliah	Hari	Jam	Kelas	Ruang	Dosen
1	Rekayasa Internet dan Web	Senin	08.00	A	C-101 A	1
			- 10.30	B	C-101 B	2
2	Elektronika Komunikasi	Selasa	08.00 - 10.30	A	C-101	3
3	Jaringan dan Rekayasa Trafik	Rabu	08.00 - 10.30	A	C-102	4
				B	C-103	5
4	Sistem Komunikasi	Kamis	08.00 - 10.30	A	C-104	6
				B	C-106	7
5	Propagasi Gelombang	Jumat	08.00 - 10.30	A	C-107	8
6	Jaringan Komunikasi Nirkabel	Senin	13.00 - 15.30	A	C-109	9
7	Jaringan Sensor Nirkabel dan IoT	Selasa	13.00 - 15.30	A	C-110	10
8	Sistem Broadcast	Rabu	13.00 - 15.30	A	C-111	11
9	Standard dan Keandalan SisKom	Kamis	13.00 - 15.30	A	B-404 A	12
10	Sekuriti dan Kriptografi	Jumat	13.00 - 15.30	A	B-404 B	13
				B	B-404 C	14

### 3.5.2.3 Pemrograman Tampilan Web

Dalam tugas akhir ini, rancangan OASYS dalam fiturnya dibagi menjadi 2 bagian yaitu tampilan untuk user dan tampilan untuk admin. User yang dimaksud dalam sistem ini adalah mahasiswa, sedangkan admin adalah petugas administrasi atau tata kelola usaha serta dosen dan tenaga pendidikan.

#### A. Tampilan User

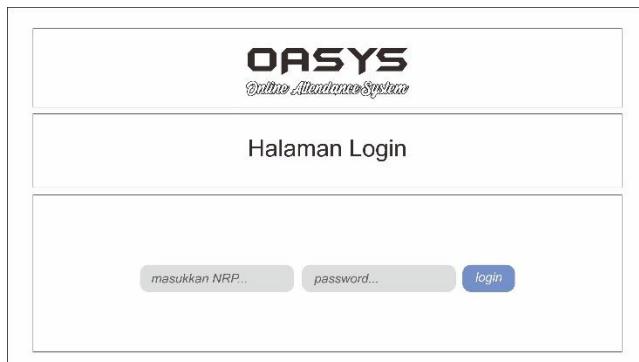
Pada tampilan website absensi berbasis IoT “OASYS” ini terdapat beberapa halaman web yang dapat diakses oleh user, yaitu halaman login, halaman daftar mata kuliah, halaman kehadiran per mata kuliah serta halaman laporan akumulasi kehadiran secara keseluruhan. Cara kerja web untuk user dapat dilihat dalam diagram alur pada Gambar 3.14 berikut.



Gambar 3. 14 Diagram Alur Website

## 1. Halaman Login

Halaman login adalah halaman awal ketika user mengakses website OASYS. Pada halaman ini, pelanggan akan memasukkan NRP sebagai username dan juga password. Setelah itu pelanggan akan diarahkan untuk ke halaman beranda. Rancangan tampilan halaman login dapat dilihat pada Gambar 3.15 sebagai berikut.



Gambar 3. 15 Tampilan Halaman Login User

Pada Gambar 3.16 berikut ini adalah pemrograman dengan notepad++ untuk halaman login user pada website.

A screenshot of the Notepad++ code editor showing the 'LoginPage.java' file. The code is a Java class named 'LoginPage' that implements the 'OnInit' interface. It contains methods for initializing components, setting up an HttpClient, and handling HTTP GET and POST requests to the 'http://192.168.100.17:8100/login' endpoint. The code uses RxJava for asynchronous operations.

Gambar 3. 16 Pemrograman Halaman Login

## 2. Halaman Mata Kuliah

Halaman ini menampilkan daftar mata kuliah yang diambil oleh user pada semester tersebut, beserta dengan hari, waktu dan ruang. Untuk melihat kehadiran user pada setiap mata kuliah, dapat dipilih mata kuliah terkait dan kemudian akan dialihkan ke halaman monitoring kehadiran per mata kuliah yang dilengkapi dengan akumulasi kehadiran sementara. Rancangan tampilan halaman mata kuliah dapat dilihat pada Gambar 3.17 sebagai berikut.

The figure consists of two vertically stacked screenshots of the OASYS system interface.

**Screenshot 1:** This screenshot shows a list of courses for the current semester. The header reads "OASYS Online Attendance System". Below it is a navigation bar with links: beranda | mata kuliah | laporan | logout. The main content is a table with the following data:

NO	MATA KULIAH	HARI	JAM	DOSEN	RUANG
1	Rukayasa Internet dan Web	Selasa	08.00-10.50		c-104
2	Jaringan Sensor Nirkabel	Rabu	11.00-13.50		c-107
3	Sistem Komunikasi	Kamis	13.00-15.50		c-109

**Screenshot 2:** This screenshot shows the attendance monitoring for the course "Rukayasa Internet dan Web". The header and navigation bar are identical. The main content is a table with the following data:

NO	TANGGAL	MINGGU	WAKTU MASUK	WAKTU ABSEN	KETERLAMBATAN	KETERANGAN	%
1	2-3-2020	1	08.00	08.15	15 MENIT	HADIR	100
2	9-3-2020	2	08.00	-	-	IJIN	50
3	16-3-2020	3	08.00	08.35	35 MENIT	ALPA	0

Below the table, there is a summary row: AKUMULASI KEHADIRAN (%) followed by a value of 80%. A yellow arrow points from the bottom right of the table towards this percentage value.

Gambar 3. 17 Tampilan Mata Kuliah

Pada Gambar 3.18 berikut ini adalah pemrograman dengan notepad++ untuk halaman mata kuliah pada website.

```

1 import | Component, Oninit | from 'sapui5/core';
2 import | DataService | from 'sapui5/service/data/service';
3 import | HubService, HubDetail | from 'sapui5/app/service/HubService';
4 import | User | from 'sapui5/app/service/User';
5 import | PopUpService | from 'sapui5/app/service/PopUpService';
6
7 /**
8  * @Component({
9  *   templateURL: './mata-kuliah.page.html',
10  *   styleClass: './mata-kuliah.page.css'
11  * })
12 */
13 export class MataKuliahPage implements Oninit {
14
15   /**
16    * @param {boolean} [oEvent]
17    * @param {Object} [oData]
18    * @param {Object} [oError]
19    * @param {Object} [oResult]
20    * @param {Object} [oUser]
21    * @param {Object} [oUserDetail]
22    * @param {Object} [oUserList]
23    * @param {Object} [oUserStatus]
24    * @param {Object} [oUserTask]
25    * @param {Object} [oUserTask2]
26    * @param {Object} [oUserTemp]
27    */
28
29   onInit(oEvent) {
30     const oRouter = Router;
31     private data: DataService;
32     private user: User;
33     private userDetail: UserDetail;
34     private popUp: PopUpService;
35   }
36
37   onInit(oEvent) {
38     const oUser = oEvent.getParameter('user');
39     const oUserDetail = oEvent.getParameter('userDetail');
40     const oUserList = oEvent.getParameter('userList');
41     const oUserStatus = oEvent.getParameter('userStatus');
42     const oUserTask = oEvent.getParameter('userTask');
43     const oUserTask2 = oEvent.getParameter('userTask2');
44     const oUserTemp = oEvent.getParameter('userTemp');
45   }
46
47   onInit() {
48     const oUser = User;
49     const oUserDetail = UserDetail;
50     const oUserList = UserList;
51     const oUserStatus = UserStatus;
52     const oUserTask = UserTask;
53     const oUserTask2 = UserTask2;
54     const oUserTemp = UserTemp;
55   }
56
57   onInit() {
58     const oUser = User;
59     const oUserDetail = UserDetail;
60     const oUserList = UserList;
61     const oUserStatus = UserStatus;
62     const oUserTask = UserTask;
63     const oUserTask2 = UserTask2;
64     const oUserTemp = UserTemp;
65   }
66
67   onInit() {
68     const oUser = User;
69     const oUserDetail = UserDetail;
70     const oUserList = UserList;
71     const oUserStatus = UserStatus;
72     const oUserTask = UserTask;
73     const oUserTask2 = UserTask2;
74     const oUserTemp = UserTemp;
75   }
76
77   onInit() {
78     const oUser = User;
79     const oUserDetail = UserDetail;
80     const oUserList = UserList;
81     const oUserStatus = UserStatus;
82     const oUserTask = UserTask;
83     const oUserTask2 = UserTask2;
84     const oUserTemp = UserTemp;
85   }
86
87   onInit() {
88     const oUser = User;
89     const oUserDetail = UserDetail;
90     const oUserList = UserList;
91     const oUserStatus = UserStatus;
92     const oUserTask = UserTask;
93     const oUserTask2 = UserTask2;
94     const oUserTemp = UserTemp;
95   }
96
97   onInit() {
98     const oUser = User;
99     const oUserDetail = UserDetail;
100    const oUserList = UserList;
101    const oUserStatus = UserStatus;
102    const oUserTask = UserTask;
103    const oUserTask2 = UserTask2;
104    const oUserTemp = UserTemp;
105  }
106}

```

Gambar 3. 18 Pemrograman Halaman Mata Kuliah Mahasiswa

### 3. Halaman Laporan

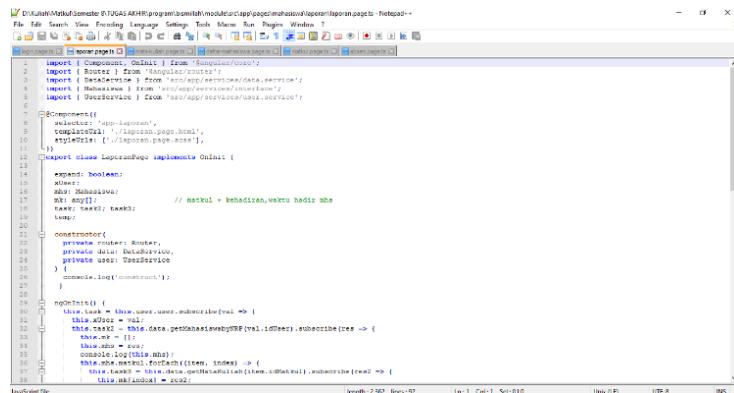
Halaman ini menampilkan laporan rekapitulasi kehadiran user di tiap mata kuliah, dengan persentase kehadiran setiap minggu yang diakumulasi dalam satu semester, serta terdapat juga keterangan apakah user tersebut dapat mengikuti UAS atau tidak, dengan asumsi untuk dapat mengikuti UAS adalah kehadiran diatas 80%. Rancangan tampilan halaman laporan dapat dilihat pada Gambar 3.19 sebagai berikut.

MATA KULIAH	KEHADIRAN									
	1	2	3	4	5	6	7	...	%	KET
Rekayasa Internet dan Jaringan Sensor Nirkabel	100%	50%	0%	100%	100%	0%	100%	...	86%	YES
Elektronika Telekomunikasi	100%	100%	100%	100%	100%	100%	100%	...	100%	NO
Sistem Komunikasi	50%	50%	100%	100%	100%	0%	0%	...	76%	NO

\*YES = boleh mengikuti EAS  
NO = tidak diperbolehkan mengikuti EAS

Gambar 3. 19 Tampilan Halaman Laporan

Pada Gambar 3.20 berikut adalah pemrograman dengan menggunakan notepad++ untuk halaman laporan pada website.

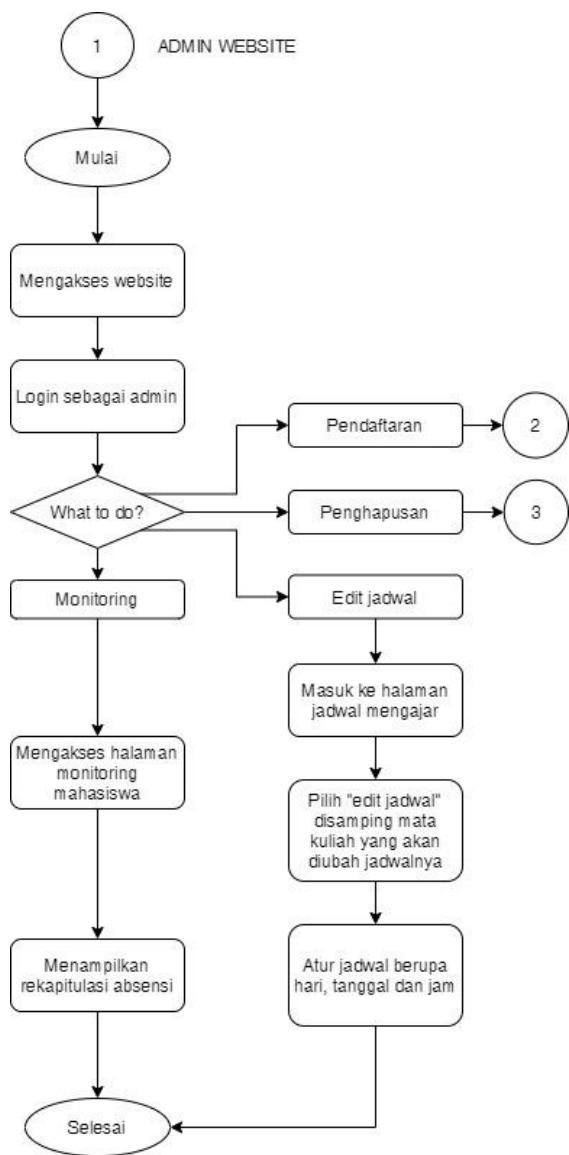


```
1 // D:\Kuliah\Matkul Semester 5\TUGAS AHIR\program absensi\module\src\app\pages\mahasiswa\laporan\laporan.page.ts - Notepad++
2 import { Component } from '@angular/core';
3 import { DataService } from 'src/app/services/data.service';
4 import { Router } from '@angular/router';
5 import { UserService } from 'src/app/services/user.service';
6
7 @Component({
8   selector: 'app-laporan',
9   templateUrl: './laporan.page.html',
10   styleUrls: ['./laporan.page.scss'],
11 })
12 export class LaporanPwpu implements OnInit {
13
14   ngOnInit(): void {
15     this.userService.getDetailMahasiswa();
16     this.userService.getKehadiran();
17     this.userService.getAbsen();
18   }
19
20   constructor(
21     private userService: UserService,
22     private detailMahasiswa: DataService,
23     private router: Router,
24     private user: UserService
25   ) {}
26   console.log('constructor');
27 }
28
29 ngOnInit(): void {
30   this.userService.user.subscribe(val => {
31     this.user = val;
32     this.detail = this.userService.getKehadiranByNRP(val.idNRP).subscribe(res => {
33       this.data = res;
34       this.userService.getAbsen();
35       this.userService.getDetailMahasiswa();
36       this.userService.getKehadiran();
37       this.userService.getAbsen();
38       this.userService.getMetakuliah();
39       this.userService.getMetakuliah([this.user.idMahasiswa]).subscribe(read => {
40         this.absen = read;
41       });
42     });
43   });
44 }
```

Gambar 3. 20 Pemrograman Halaman Laporan

## B. Tampilan Admin

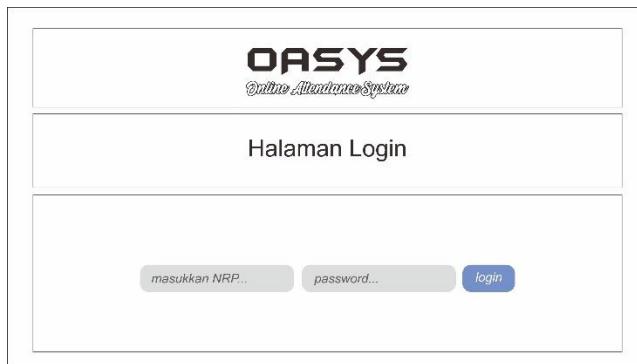
Dalam sistem ini admin adalah petugas administrasi, tata kelola usaha, dosen dan tenaga pendidikan. Admin berfungsi untuk mengatur beberapa hal yang perlu ditambahkan, dihapus ataupun diubah dari database yang sudah ada. Petugas administrasi dan tata kelola usaha lebih berfungsi kepada menambahkan dan menghapus ID sidik jari mahasiswa, sedangkan dosen dan tenaga pendidikan lebih ke monitoring pengubahan jadwal kuliah. Adapun gambaran cara kerja admin dalam sistem absensi IoT ini dijelaskan dalam diagram alur pada Gambar 3.21 berikut.



**Gambar 3. 21** Diagram Alur Admin pada Sistem

## 1. Halaman Login

Halaman login yang ditampilkan sama dengan halaman login pada user, yang membedakan adalah username dan password yang akan dimasukkan. Apabila username dan password sesuai untuk admin, maka akan dialihkan ke halaman beranda admin. Rancangan tampilan halaman login dapat dilihat pada Gambar 3.22 sebagai berikut.



Gambar 3. 22 Tampilan Halaman Login

Pada Gambar 3.23 berikut ini adalah pemrograman menggunakan notepad++ untuk halaman login admin pada website.

A screenshot of the Notepad++ code editor showing the Java code for the login page. The code defines a class 'LoginPage' that implements the 'OnInit' interface. It contains a constructor that initializes an 'HttpClient' object. The 'onInit()' method sends a GET request to the URL 'http://192.168.100.17:1000/login'. The response is then processed to extract the token from the JSON response. If successful, the token is stored in the session and the user is redirected to the home page ('index.jsp').

```
1 import java.io.IOException;
2 import javax.servlet.ServletException;
3 import javax.servlet.http.HttpServlet;
4 import javax.servlet.http.HttpServletRequest;
5 import javax.servlet.http.HttpServletResponse;
6
7 import org.json.JSONObject;
8
9 public class Loginpage implements OnInit {
10
11     public void onInit() throws ServletException {
12         HttpClient client = new DefaultHttpClient();
13
14         String response = client.execute(new HttpGet("http://192.168.100.17:1000/login"));
15
16         JSONObject json = new JSONObject(response.getEntity().getContent());
17
18         String token = json.getJSONObject("token");
19
20         HttpSession session = ((HttpServletRequest) request).getSession(true);
21
22         session.setAttribute("token", token);
23
24         response.sendRedirect("index.jsp");
25     }
26
27 }
```

Gambar 3. 23 Pemrograman Halaman Login

## 2. Halaman Mata Kuliah

Pada halaman mata kuliah admin terdapat opsi edit jadwal pada masing-masing mata kuliah. Opsi ini berfungsi untuk pengubahan jadwal mengajar sesuai dengan kebutuhan dosen. Kemudian ketika memilih opsi mata kuliah maka akan menampilkan daftar mahasiswa yang mengikuti mata kuliah tersebut berikut dengan keterangan kehadiran tiap minggunya serta akumulasi kehadiran dalam satu semester dengan bentuk persentase. Rancangan tampilan halaman mata kuliah dapat dilihat pada Gambar 3.24 sebagai berikut.

The image displays two screenshots of the OASYS Online Attendance System interface. Both screenshots feature a header with the OASYS logo and 'Online Attendance System' text, followed by navigation links: 'beranda | mata kuliah | daftar mahasiswa | logout'. A yellow arrow labeled 'A' points to the 'mata kuliah' link in the top header of the first screenshot.

**Screenshot 1: Course Schedules**

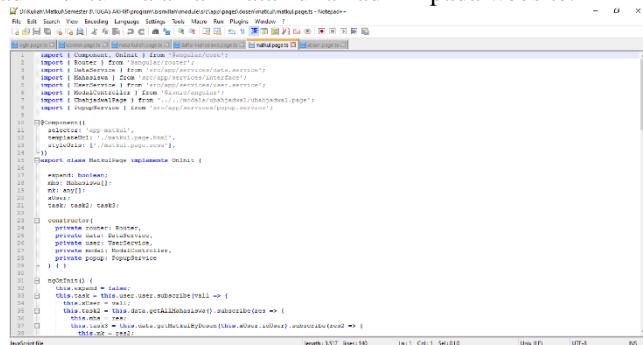
NO	MATA KULIAH	HARI, TANGGAL	JAM	RUANG	
1	<i>Rekayasa Internet dan Web</i>	Selasa,	08.00-10.50	C-104	<a href="#">edit jadwal</a>
2	<i>Jaringan Sensor Nirkabel</i>	Rabu	11.00-13.50	C-103	<a href="#">edit jadwal</a>
3	<i>Sistem Komunikasi</i>	Kamis	13.00-15.50	C-102	<a href="#">edit jadwal</a>

**Screenshot 2: Student Attendance Details**

NO	NAMA	NRP	MINGGU 1				MINGGU 2				MINGGU 3				—		AKUMULASI (%)
			HADIR	HADIR	ALPA	...	HADIR	IZIN	HADIR	...	HADIR	HADIR	...	...	...		
1	Ismail	07111640000020	HADIR	HADIR	ALPA	...	90										
2	Ashari	07111640000180	HADIR	IZIN	HADIR	...	50										
3	...	...	HADIR	HADIR	HADIR	...	86										
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	

Gambar 3. 24 Tampilan Halaman Mata Kuliah

Pada Gambar 3.25 berikut ini adalah pemrograman menggunakan notepad++ untuk halaman mata kuliah admin pada website.



```

1 import | Components, UIEvent | from "javax.faces.event";
2 import | DatabaseService | from "com.oasys.moodle/data/service";
3 import | DataModel | from "javax.faces.model";
4 import | FacesContext | from "javax.faces.context";
5 import | HttpServletRequest | from "java.servlet.http.HttpServletRequest";
6 import | HttpServletResponse | from "java.servlet.http.HttpServletResponse";
7
8 /**
9 * @author : Ismail
10 * @email : ismail.page.host@gmail.com
11 */
12 export class Mahasiswa implements Serializable {
13     private DatabaseService database;
14     private String name;
15     private int NRP;
16
17     public Mahasiswa() {
18         database = new DatabaseService();
19         name = null;
20         NRP = 0;
21     }
22
23     public Mahasiswa(String name, int NRP) {
24         database = new DatabaseService();
25         this.name = name;
26         this.NRP = NRP;
27     }
28
29     public void addMahasiswa() {
30         database.addUser(this);
31     }
32
33     public void removeMahasiswa() {
34         database.removeUser(this);
35     }
36
37     public void updateMahasiswa() {
38         database.updateUser(this);
39     }
40
41     public void generateQRCode() {
42         QRCodeUtil.generateQRCode(this.name + this.NRP + "-" + this.NRP);
43     }
44
45     public String getQRCode() {
46         return "data:image/png;base64," + QRCodeUtil.getQRCode(this.name + "-" + this.NRP);
47     }
48 }

```

Gambar 3. 25 Pemrograman Halaman Mata Kuliah Admin

### 3. Halaman Daftar Mahasiswa

Pada halaman daftar mahasiswa akan menampilkan daftar nama mahasiswa berikut dengan NRP dan keterangan semester, yang dapat disortir berdasarkan tahun angkatan. Kemudian terdapat opsi pendaftaran dan hapus yang berfungsi untuk registrasi dan pengapusan sidik jari, dimana perintah ini terhubung langsung dengan device absensi. Rancangan tampilan halaman daftar mahasiswa dapat dilihat pada Gambar 3.26 sebagai berikut.



NO	NAMA	NRP	SEMESTER	
1	Ismail	0711164000020	8	<a href="#">pendaftaran</a> <a href="#">hapus</a>
2	Ashari	0711164000180	8	<a href="#">pendaftaran</a> <a href="#">hapus</a>
3				
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Gambar 3. 26 Tampilan Halaman Daftar Mahasiswa

Pada Gambar 3.27 berikut ini adalah pemrograman yang ditulis menggunakan notepad++ untuk halaman daftar mahasiswa dan akan diterapkan pada website.

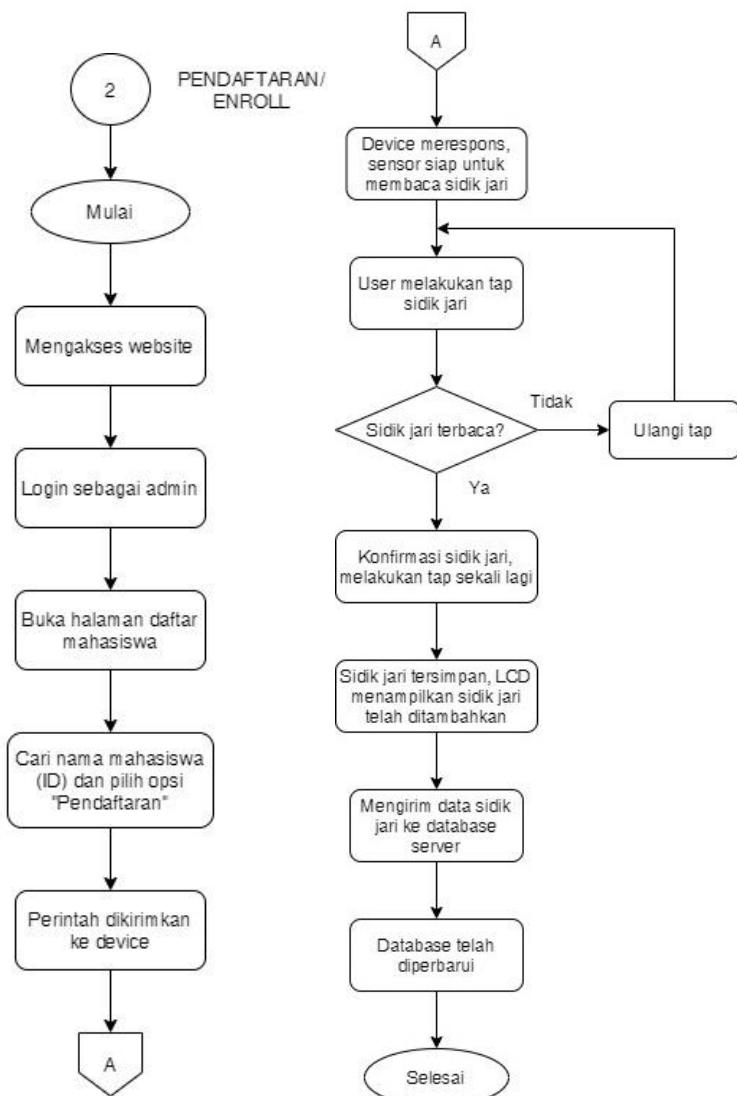
```

1 import i Components;Galaxy I from 'Components/page';
2 import i Router I from 'Components/router';
3 import i View I from 'Components/view';
4 import i Mahasiswa I from 'src/app/mahasiswa/mahasiswa';
5 import i Page I from 'src/app/_base/base';
6 import i LoadingController I from 'ionicons/ionicons';
7 import i NavParams I from 'src/app/core/navparams/navparams';
8
9 @Component({
10   selector: 'app-mahasiswa',
11   templateUrl: './mahasiswa.page.html',
12   styleUrls: ['./mahasiswa.page.scss'],
13 })
14 export class MahasiswaPage implements OnInit {
15   name: any;
16   user: any;
17   userToken: any;
18   disableAll = false;
19
20   constructor(
21     private navCtrl: NavController,
22     private route: ActivatedRoute,
23     private navParams: NavParams,
24     private loading: LoadingController
25   ) { }
26
27   ngOnInit() {
28     this.name = this.user.user.username();
29     this.route.params.subscribe(res => {
30       this.name = this.data.get("username").subscribe(res => {
31         this.name = res.name && (!res.disableAll) ? res.name : false;
32       });
33     });
34   }
35   ionDidacticOpener() {
36     if(this.disableAll) {
37       if(this.navCtrl.getState() == "selected" || person.getStatus == "selanjutnya" || person.getStatus == "Deleting")) {
38         person.disableAll = true;
39       } else {
40         person.disableAll = false;
41       }
42     }
43   }
44 }

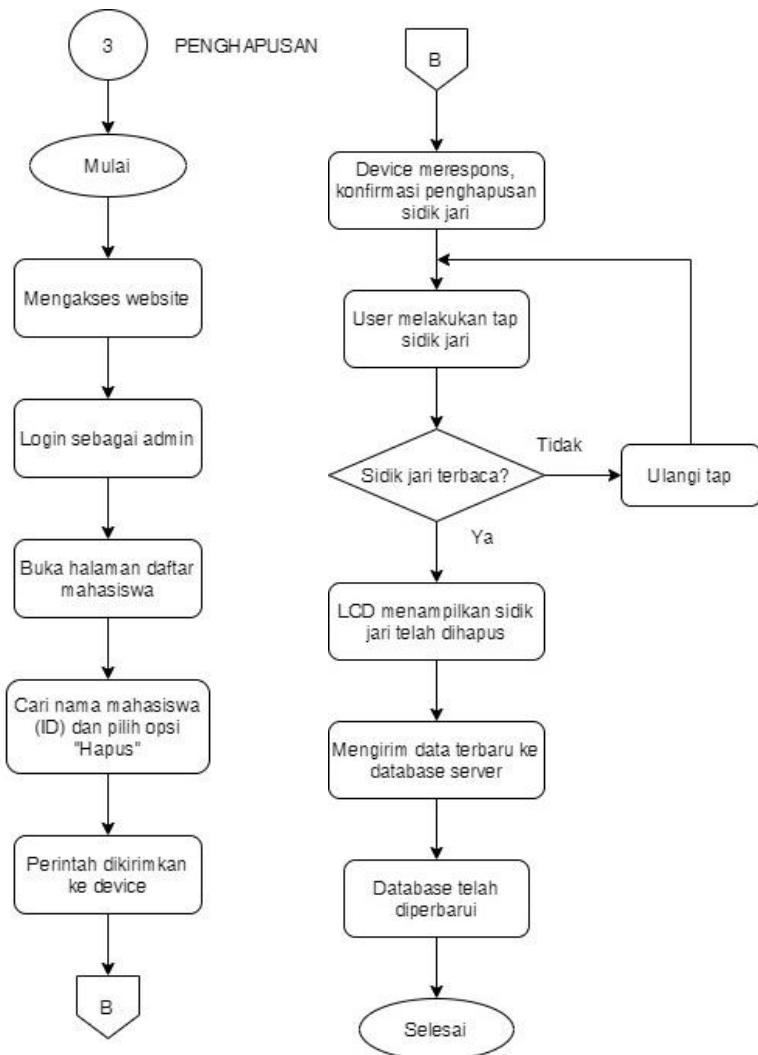
```

**Gambar 3.27** Pemrograman Halaman Daftar Mahasiswa

Selain tampilan monitoring, akan dibuat juga berbagai fitur-fitur yang dapat digunakan, diantaranya terdapat fitur registrasi atau pendaftaran sidik jari mahasiswa serta fitur penghapusan sidik jari mahasiswa. Cara kerja fitur registrasi dan fitur penghapusan sidik jari mahasiswa, dijelaskan dalam diagram alur pada Gambar 3.28 dan Gambar 3.29 sebagai berikut.



**Gambar 3. 28** Diagram Alur Pendaftaran Sidik Jari



**Gambar 3. 29** Diagram Alur Penghapusan Sidik Jari

### **3.6 Integrasi Sistem**

Dalam pembahasan integrasi sistem, *device* yang telah dibuat sesuai dengan rancangan kemudian dihubungkan ke database server melalui jaringan internet dengan menggunakan modul IoT yang terdapat pada masing-masing *device*. Untuk dapat melakukan komunikasi antara *device* dan server digunakan protokol HTTP yang menghubungkan modul IoT dengan cloud functions pada Firebase. Setelah dapat terhubung dengan baik, kemudian untuk menampilkan isi dari database digunakan Ionic Framework yang diprogram dengan bahasa JavaScript dan terhubung secara *back-end*.

*Device* yang telah selesai dibuat akan direaliasikan pada setiap kelas atau ruang yang ada di kampus. Penempatan *device* berada di dalam kelas agar setiap pengguna atau mahasiswa yang akan melakukan absensi berupa tap sidik jari benar-benar masuk ke dalam kelas. Sedangkan untuk mengakses atau monitoring website absensi dapat dilakukan melalui browser masing-masing, dapat diakses lewat PC ataupun HP.

Cara kerja sistem ini adalah dimulai dari *device* yang ada di masing-masing kelas, untuk melakukan absensi pengguna dapat langsung melakukan tap pada ruang kelas masing-masing, lalu data sidik jari yang terekam dan tersimpan kemudian dikirimkan ke server menggunakan modul IoT melalui jaringan internet. Dari server, data yang diterima kemudian ditampilkan pada website yang nantinya dapat diakses melalui browser. Adapun untuk melakukan penghapusan dan pendaftaran sidik jari, dimulai dari perintah yang ada di website yang kemudian dikirimkan menuju *device* hingga pada akhirnya sensor siap untuk membaca sidik jari.

### **3.7 Pengujian Sistem**

Dalam pengujian sistem absensi berbasis IoT ini akan dibagi menjadi 3, yaitu pengujian antar segmen dan integrasi sistem, pengujian operasional sistem, serta pengujian keandalan sistem.

#### **3.7.1 Pengujian Segmen Komunikasi**

Pada pengujian ini akan dilakukan pengecekan konektivitas antar segmen atau komponen dengan tujuan untuk mengetahui apakah antar komponen terhubung dengan baik sehingga keseluruhan sistem dapat terintegrasi dan berjalan dengan baik. Pengujian segmen ini akan dibagi menjadi berikut.

#### **A. Segmen Komunikasi Sensor - Mikrokontroler**

Pengujian pertama adalah pada jalur komunikasi yang dipakai, apakah menggunakan kabel (*wired*) atau tidak (*wireless*). Kemudian pengecekan apakah sensor telah terkoneksi ke mikrokontroler.

#### **B. Segmen Komunikasi Mikrokontroler - Modul IoT**

Pengujian dilakukan dengan melihat pada jalur komunikasi yang dipakai, apakah menggunakan kabel (*wired*) atau tidak (*wireless*). Kemudian pengecekan apakah mikrokontroler dan modul IoT telah terkoneksi.

#### **C. Segmen Komunikasi Modul IoT - Internet**

Pengujian dilakukan dengan melihat pengiriman data yang dilakukan dari modul IoT ke internet, apakah data dapat terkirim dengan baik atau tidak.

#### **D. Segmen Komunikasi Internet - Database Server**

Pengujian dilakukan dengan melihat data yang diterima pada database server, memastikan data diterima sesuai dengan data yang dikirim.

#### **F. Segmen Komunikasi Database Server - Web Interface**

Pengujian dilakukan dengan melihat perubahan pada tampilan web setelah database diperbarui, dan memastikan apakah perubahan tersebut sudah sesuai dengan yang diinginkan atau tidak.

#### **3.7.2 Pengujian Operasional Sistem**

Pada pengujian operasional sistem akan dilakukan pengujian penggunaan *device* absensi, serta penggunaan fitur-fitur yang tersedia pada website. Dengan kondisi telah dibuat terlebih dahulu dibuat database yang berisi daftar nama mahasiswa beserta dengan jadwal mata kuliah yang diambil pada semester tersebut.

Skenario yang akan diterapkan dalam pengujian operasional sistem ini adalah sebagai berikut:

1. Pengujian dengan kondisi terdapat mahasiswa yang melakukan absensi pada mata kuliah yang berbeda, namun di hari dan waktu yang bersamaan. Pengujian ini disebut juga dengan pengujian kelas paralel.

2. Pengujian dengan kondisi terdapat mahasiswa yang melakukan absensi pada mata kuliah yang berbeda, dan waktu yang berbeda, namun pada hari yang sama. Pengujian ini disebut juga dengan pengujian kelas serial.

Dalam pengujian dengan menggunakan 2 kondisi tersebut, akan dilihat hasilnya pada website, beserta penggunaan fitur-fitur yang ada. Selain itu juga memperhatikan kinerja dari *device* absensi sesuai dengan diagram alur sistem yang telah dijelaskan sebelumnya.

### **3.7.3 Pengujian Keandalan Sistem**

Pengujian keandalan sistem dilakukan dengan tujuan untuk mengetahui kelayakan fungsi pada bagian-bagian dari sistem yang diharapkan sistem memiliki kualitas yang baik sehingga dapat disebut layak untuk digunakan.

Pada pengujian keandalan sistem ini akan dibagi menjadi 2 bagian, yaitu sebagai berikut:

1. Pengujian kecepatan proses pembacaan pada sensor sidik jari. Pengujian ini dilakukan dengan melihat waktu yang diperlukan bagi sensor untuk memproses sidik jari, dimulai dari pembacaan (*scanning*) sidik jari pada saat kondisi idle sampai dengan notifikasi sidik jari tersimpan dan kembali pada posisi idle.
2. Pengujian ketepatan data yang masuk pada database. Pengujian ini dilakukan dengan melihat data masukan (*input*) yang berasal dari sensor sidik jari, yang kemudian dibandingkan dengan data yang tersimpan pada database untuk memastikan keduanya telah sesuai.

## **3.8 Pengukuran Kualitas Layanan**

Pengukuran kualitas layanan pada tugas akhir berfungsi untuk mengukur kemampuan suatu jaringan untuk menyediakan layanan yang baik. Baik atau buruknya kualitas layanan yang diberikan kepada pengguna disebut juga dengan *Quality of Service* (QoS). Pengukuran QoS memiliki tujuan untuk dapat dijadikan parameter dalam memberikan pelayanan jaringan yang terbaik sehingga dapat memenuhi kebutuhan suatu layanan

### **3.8.1 Tool Pengukuran**

Pengukuran kualitas jaringan dilakukan dengan menggunakan *software* Wireshark. Wireshark merupakan sebuah perangkat lunak yang memiliki fungsi untuk menganalisa jaringan. Pada perancangan sistem absensi IoT ini, Wireshark berfungsi untuk menampilkan statistik jaringan dalam waktu tertentu untuk kemudian diambil datanya sebagai pengukuran kualitas jaringan yang dipakai.

### **3.8.2 Parameter Pengukuran**

Dalam melakukan pengukuran kualitas jaringan ini, terdapat parameter yang akan diamati dan dianalisa, yaitu *Throughput*, *Delay*, *Jitter* dan *Packet Loss*.

#### **A. Throughput**

*Throughput* adalah kecepatan (rate) transfer data yang diukur dalam satuan bps (bit per second). Pengukuran throughput dilakukan untuk melihat trafik kecepatan pengiriman data pada masing-masing kondisi.

#### **B. Delay**

*Delay* adalah waktu tunda yang terjadi karena proses transmisi data. Pengukuran delay dilakukan untuk melihat kecenderungan waktu tunda pada masing-masing kondisi, yang bisa berpengaruh pada kecepatan pengiriman data.

#### **C. Jitter**

*Jitter* adalah variasi delay antar paket yang diakibatkan oleh panjang antrian dalam suatu pengolahan data dan penghimpunan ulang paket-paket data diakhir pengiriman akibat kegagalan sebelumnya sebelumnya. Pengukuran jitter dilakukan untuk mengetahui banyaknya variasi delay pada pengiriman data dalam masing-masing kondisi.

#### **D. Packet Loss**

*Packet loss* adalah sebuah parameter yang menunjukkan kegagalan dalam proses transmisi paket data untuk mencapai tujuan. Tujuan pengukuran packet loss adalah untuk mengetahui apabila ada paket data yang tidak terkirim.

### **3.8.3 Skenario Pengukuran**

Pengukuran dilakukan dengan skenario yang ditentukan, untuk mendapatkan data yang kemudian akan dianalisa dari beberapa parameter yang telah ditentukan. Dalam pengukuran kualitas jaringan ini, akan dilakukan skenario pengujian sebagai berikut.

#### **A. Pengujian Pengaruh Jarak Ruang Kelas**

Melihat data statistik jaringan yang terhubung dari *device* ke server, dengan *device* dalam keadaan *loop* mengirim data selama 5 menit pada satu *device*. Pengukuran ini dilakukan dalam jarak terentu, yaitu 5 meter dan 10 meter. Pengukuran ini bertujuan untuk melihat perbedaan kualitas jaringan yang digunakan oleh *device* ketika dalam jarak yang berbeda. Setelah mendapatkan data statistik tersebut kemudian dilakukan perhitungan untuk menentukan nilai masing-masing parameter pengujian, yaitu *Throughput*, *Delay*, *Jitter*, dan *Packet Loss*. Berikut adalah langkah-langkah dalam pengukuran QoS.

1. Langkah pertama yang dilakukan adalah menyiapkan *device* dan mengatur jaraknya dengan *access point* atau router WiFi sejauh 5 meter.
2. Nyalakan *device* dan pastikan pc yang akan digunakan untuk mengukur nilai QoS terhubung pada jaringan yang sama dengan *device*.
3. Buka *software wireshark* pada pc. Lalu pilih *interface* yang akan dipakai dalam pengukuran.
4. Sebelum memulai pengukuran, atur waktu lama pengukuran selama 5 menit pada tab *capture*. Lalu pilih start.
5. Setelah selesai pengukuran, untuk menghitung *throughput* tuliskan filter “tcp && ip.src==192.168.xx.xx && tcp && ip.dst==192.168.xx.xx” untuk hanya menampilkan protokol TCP dengan IP asal dan tujuan yang sama. Dalam hal ini IP yang dimaksud adalah IP *device*. Buka *capture file properties* pada tab *statistic* untuk menampilkan hasil data trafik yang terekam.
6. Untuk menghitung *delay*, tuliskan filter “tcp.analysis.ack\_rtt && ip.src==192.168.xx.xx || ip.dst==192.168.xx.xx” untuk menampilkan pengiriman data dengan IP asal dan IP tujuan yang

ditentukan. IP asal dan IP tujuan disini merupakan alamat dari *device*. Setelah dilakukan filter kemudian *export packet dissection*, as csv. File .csv dapat dibuka dengan MS. Excel. Menghitung delay dengan cara  $T-(T-1)$ , atau menghitung selisih waktu antar paket terkirim, lalu dirata-rata terhadap total paket yang dikirim.

7. Untuk menghitung *jitter* dapat dilakukan setelah didapatkan nilai delay. Pada pengukuran ini dilakukan perhitungan selisih waktu antar delay pada paket yang dikirim ( $D-(D-1)$ ), menggunakan MS. Excel. Nilai jitter diperoleh dengan total variasi *delay* atau total selisih waktu *delay* dibagi dengan jumlah total paket yang dikirim.
8. Untuk menghitung *packet loss* dilakukan dengan menuliskan filter “tcp.analysis.ack\_lost\_segment”. Kemudian amati pada transmisi data, jika tidak ada data yang tertampil berarti tidak ada *packet loss* pada paket data yang dikirim.
9. Lakukan langkah yang sama pada jarak 10 meter, untuk menghitung masing-masing nilai dari *throughput*, *delay*, *jitter* dan *packet loss*.

## B. Pengujian Pengaruh Jumlah Ruang Kelas

Pada pengujian ini ruang kelas direpresentasikan oleh *device*. Pengujian ini dilakukan dengan melihat data statistik jaringan yang terhubung dari masing-masing *device* ke server, dengan menggunakan 2 *devices* sebagai perbandingan dalam kondisi terhubung 1 *device* dan 2 *devices* yang masing-masing dalam keadaan *loop* mengirim data selama 5 menit. Pengukuran ini dilakukan dalam jarak 5 meter. Pengukuran ini bertujuan untuk melihat perbedaan kualitas jaringan yang digunakan oleh *device* ketika banyak *device* yang terhubung ke satu server yang sama. Setelah mendapatkan data statistik tersebut kemudian dilakukan perhitungan untuk menentukan nilai masing-masing parameter pengujian, yaitu *Throughput*, *Delay*, *Jitter*, dan *Packet Loss*. Berikut adalah langkah-langkah dalam pengukuran QoS.

1. Langkah pertama yang dilakukan adalah menyiapkan kedua *device* dan atur jarak ke router WiFi atau *internet gateway* masing-masing sejauh 5 meter.

2. Lakukan pengukuran dengan 1 *device* yang terhubung terlebih dahulu, untuk mengukur masing-masing parameter QoS pada *device* 1, lakukan langkah-langkah yang sama seperti pada pengujian pengaruh jarak ruang kelas, sampai pada pengukuran *packet loss*.
3. Setelah selesai, nyalakan kedua *device* agar terhubung ke jaringan yang sama secara bersamaan.
4. Hitung masing-masing parameter QoS pada *device* 1 terlebih dahulu, dengan langkah-langkah yang sama seperti pada saat pengukuran 1 *device* terhubung.
5. Setelah selesai, lakukan langkah-langkah yang sama dengan mengubah alamat IP *device* 1 menjadi alamat IP pada *device* 2.
6. Untuk menghitung nilai QoS pada saat terdapat 2 *device* terhubung yaitu dengan hitung rata-rata pada kedua device.

### C. Pengujian Pengaruh Waktu Pada Pengiriman Data

Melihat data statistik jaringan yang terhubung dari *device* ke server, dengan kondisi *device* dalam keadaan *loop* mengirim data selama 5 menit. Pengukuran ini dilakukan selama 2 hari berturut-turut. Pengukuran dilakukan sebanyak 3 waktu dalam sehari, yaitu pada jam 8 pagi, jam 1 siang, dan jam 9 malam. Setelah mendapatkan data statistik tersebut kemudian dilakukan perhitungan untuk menentukan nilai masing-masing parameter pengujian, yaitu *Throughput*, *Delay*, *Jitter*, dan *Packet Loss*. Berikut adalah langkah-langkah dalam pengukuran QoS.

1. Pengukuran pertama dilakukan pada pukul 08.00 pagi.
2. Langkah-langkah yang dilakukan pada pengujian pengaruh waktu pada pengiriman data sama seperti pada pengukuran ketika 2 *device* terhubung secara bersamaan.
3. Lakukan langkah-langkah yang sama hingga mendapatkan nilai dari *packet loss*.
4. Setelah selesai, lakukan semua langkah-langkah tersebut pada pukul 13.00 siang serta pukul 21.00 malam.

## BAB IV

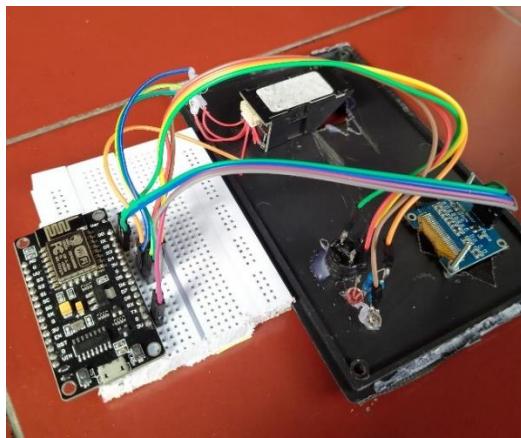
### ANALISIS DATA DAN PEMBAHASAN

Pada bagian ini akan membahas data hasil pengujian dan pengukuran beserta dengan analisisnya.

#### 4.1 Hasil Implementasi Desain Sistem Absensi

##### 4.1.1 Hasil Implementasi *Hardware*

Pada bagian ini akan menjelaskan tentang hasil purwarupa atau *prototype* dari *device* absensi yang dibuat sesuai dengan rancangan yang telah dijelaskan pada bab 3, serta mengasumsikan *prototype* pada kelas-kelas. Dari sisi *hardware* menggunakan komponen utama mikrokontroler NodeMCU ESP8266, yang berfungsi sebagai otak dari sistem. Sebuah sensor sidik jari FPM10A dihubungkan dengan mikrokontroler NodeMCU ESP8266 sebagai input. Kemudian output yang digunakan adalah LCD Display SSD1306, lampu LED 2 buah berwarna biru dan merah, serta buzzer aktif 5v yang masing-masing berfungsi untuk menampilkan proses dari *device*. Berikut pada Gambar 4.1 merupakan hasil dari rangkaian pada *device* absensi.



**Gambar 4. 1** Rangkaian *Device* Absensi

Pada sisi *software*, dalam pembuatan program pada sistem absensi menggunakan Arduino IDE. Pada pembuatan program ini, menggunakan beberapa library yang telah disediakan oleh banyak *developer* untuk kemudian lebih dikembangkan sesuai dengan kebutuhan. Beberapa library yang digunakan diantaranya adalah SPI.h, Wire.h, ESP8266WiFi.h, SoftwareSerial.h, ESP8266WebServer.h, ESP8266HTTPClient.h, Adafruit\_GFX.h, Adafruit\_SSD1306.h, Adafruit\_Fingerprint.h. Untuk keseluruhan program lebih detail akan dilampirkan.

Hasil akhir dari *prototype device* absensi yang telah dibuat ditampilkan pada Gambar 4.2 berikut.



**Gambar 4. 2 Prototype Absensi**

Setelah *prototype* selesai dibuat, kemudian diaplikasikan pada kelas-kelas. Telah dibuat visualisasi kelas untuk simulasi penggunaan sistem absensi ini. Visualisasi kelas dibagi menjadi 2, yaitu *single-class* dan *multi-class*. Berikut pada Gambar 4.3 dan Gambar 4.4 adalah merupakan hasil dari visualisasi kelas yang telah dibuat.



**Gambar 4. 3** Visualisasi *Single-class*



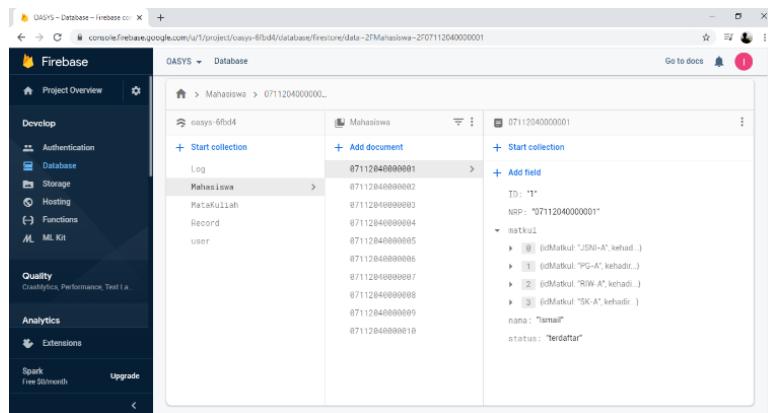
**Gambar 4. 4** Visualisasi *Multi-class*

Dalam penggunaannya, *prototype* dapat digunakan di banyak kelas, namun dalam simulasi ini hanya digunakan 2 kelas sebagai contoh.

#### **4.1.2 Hasil Pemrograman Database**

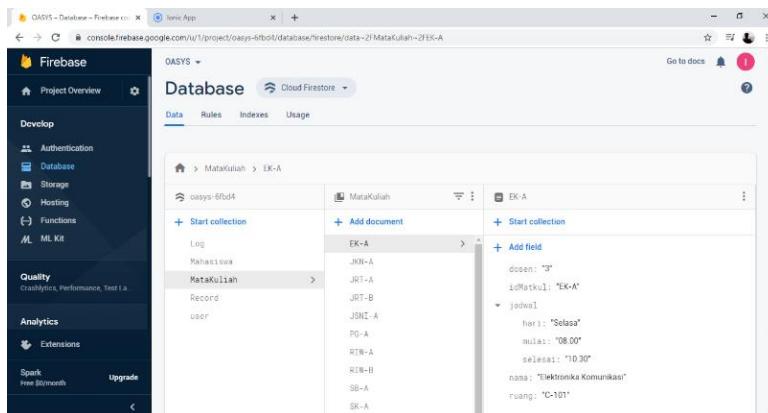
Pada bagian ini akan menjelaskan hasil database yang telah dibuat menggunakan Firebase. Database dibuat dengan memasukkan jenis-jenis data sesuai yang telah dijelaskan pada bab 3, dengan menggunakan fitur layanan *Cloud Firestore* milik Firebase. Tampilan hasil database yang

telah selesai dibuat, dapat dilihat pada Gambar 4.5 dan Gambar 4.6 berikut.



The screenshot shows the Firebase Database interface for the 'Mahasiswa' collection. The left sidebar shows the project overview and various database settings. The main area displays a list of documents under the 'Mahasiswa' collection. One document is expanded to show its fields: ID (07112040000001), NRP (07112040000001), matkul (an array containing 'JSMI-A', 'PG-A', 'IRW-A', and 'SK-A'), nana ('female'), and status ('terdaftar').

Gambar 4.5 Tampilan Database Firebase Kelompok Mahasiswa



The screenshot shows the Firebase Database interface for the 'MataKuliah' collection. The left sidebar shows the project overview and various database settings. The main area displays a list of documents under the 'MataKuliah' collection. One document is expanded to show its fields: EK-A (document ID), JMK-A, JRT-A, JRT-B, JSMI-A, PG-A, RTR-A, RTR-II, SB-A, SK-A, docan ('D'), idMatkul ('EK-A'), jadwal (an array containing 'Selasa' at 08.00 and '10.30'), nana ('Teknik Komunikasi'), and ruang ('C-101').

Gambar 4.6 Tampilan Database Firebase Kelompok MataKuliah

Pada database yang telah dibuat, pengelompokan dibagi menjadi 2, yaitu Mahasiswa dan MataKuliah. Dalam kelompok Mahasiswa berisi nomor-nomor NRP yang berfungsi sebagai ID. Di dalam ID tersebut

berisi daftar mata kuliah yang diambil, nama, dan status apakah sidik jari sudah terdaftar atau belum, sesuai pada Gambar 4.5.

Dalam kelompok MataKuliah berisi daftar mata kuliah dalam bentuk kode yang berfungsi sebagai ID mata kuliah. Di dalam ID mata kuliah tersebut berisi dosen yang mengajar, ruang kelas, serta jadwal yang meliputi hari, jam mulai dan jam selesai, sesuai pada Gambar 4.6.

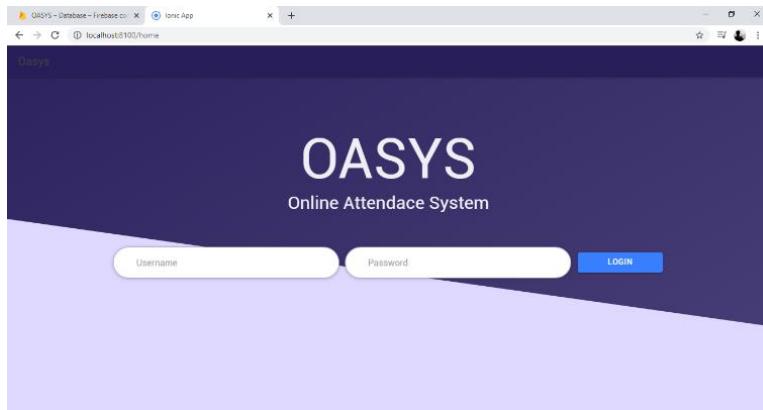
#### 4.1.3 Hasil Pemrograman Web

Pada bagian ini akan menjelaskan tentang tampilan dari website yang telah dibuat untuk sistem absensi berbasis IoT, yang diberi nama OASYS (*Online Attendance System*), serta pemrograman yang digunakan.

Dalam penggunaannya website dapat diakses dengan *host* lokal, menggunakan cmd lalu buka dengan perintah “ionic serve”. Maka website akan ditampilkan dengan url *localhost:(port)*.

##### 4.1.3.1 Hasil Halaman Login

Halaman ini berisi kolom Username dan Password yang harus diisi oleh pengguna. Fungsinya adalah untuk membedakan antara dosen dan mahasiswa, serta antar pengguna (mahasiswa satu dan mahasiswa lain). Pada Gambar 4.7 berikut ini adalah merupakan tampilan halaman *login* pada web OASYS.



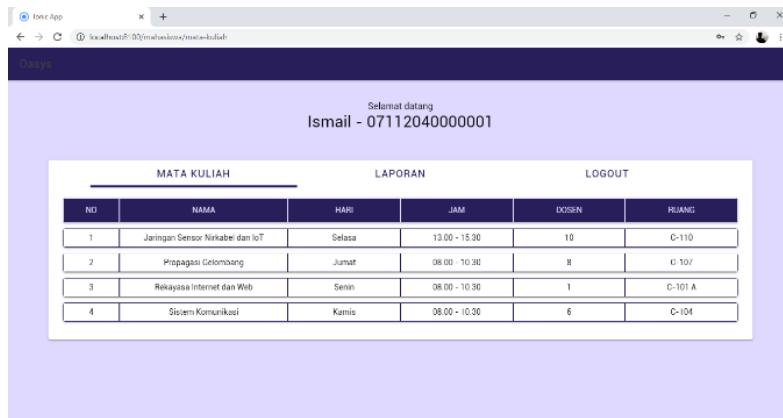
Gambar 4.7 Tampilan Halaman Login

#### **4.1.3.2 Hasil Tampilan Mahasiswa**

Pada tampilan pengguna mahasiswa akan menampilkan dua halaman untuk monitoring, yaitu halaman mata kuliah dan halaman laporan.

##### **A. Hasil Halaman Mata Kuliah**

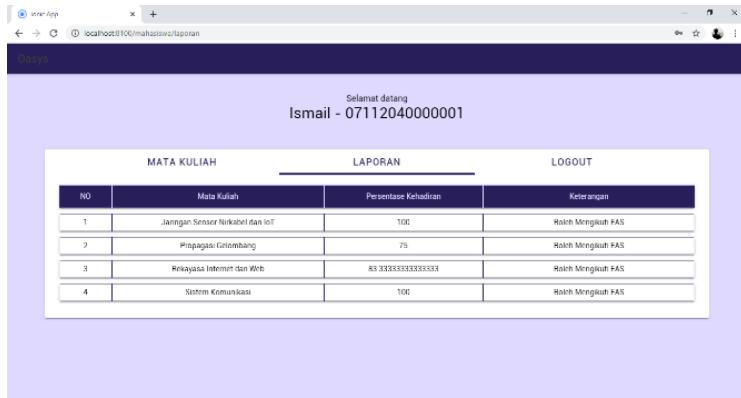
Pada halaman ini akan menampilkan daftar mata kuliah yang diambil oleh setiap mahasiswa, apabila diklik pada kolom mata kuliah, maka akan menampilkan keterangan kehadiran mahasiswa tiap minggunya. Halaman mata kuliah dapat dilihat pada Gambar 4.8 berikut.



**Gambar 4.8** Halaman Mata Kuliah – Mahasiswa

##### **B. Hasil Halaman Laporan**

Pada halaman ini menampilkan rekapitulasi absensi pada setiap mata kuliah yang diambil oleh mahasiswa, yang kemudian akan muncul keterangan apakah mahasiswa tersebut dapat mengikuti EAS atau tidak, dengan catatan kehadiran lebih dari 80%. Halaman laporan dapat dilihat pada Gambar 4.9 berikut.



**Gambar 4. 9** Halaman Laporan – Mahasiswa

#### 4.1.3.3 Hasil Tampilan Dosen

Pada tampilan admin/dosen akan menampilkan dua halaman untuk monitoring. Halaman pertama menampilkan mata kuliah apa saja yang diampu oleh dosen tersebut, disertai dengan fitur untuk mengubah jadwal. Halaman kedua menampilkan daftar nama mahasiswa, yang disertai fitur registrasi dan penghapusan sidik jari untuk pengguna.

##### A. Hasil Halaman Mata Kuliah

Halaman ini berisi daftar mata kuliah yang diampu oleh dosen, dilengkapi dengan hari, waktu dan ruang kuliah. Pada halaman ini juga terdapat fitur ubah jadwal yang berfungsi untuk mengubah jadwal suatu mata kuliah yang hanya dapat dilakukan oleh dosen/admin. Jadwal dapat diubah sesuai dengan kehendak dosen. Pengubahan jadwal berlaku untuk mata kuliah pada setiap pertemuannya, dan secara otomatis akan mengubah database yang ada. Berikut pada Gambar 4.10 adalah tampilan dari halaman mata kuliah pada dosen.

MATA KULIAH						DAFTAR MAHASISWA	LOGOUT
NO	Mata Kuliah	Hari, Tanggal	Jam	Ruang			
1	Elektronika Komunikasi (DK-A)	Selasa	08.00 - 10.30	C-101	<a href="#">UBAH JADWAL</a>		
2	Jaringan Komunikasi Nirkabel (JKN-A)	Senin	13.00 - 15.30	C-109	<a href="#">UBAH JADWAL</a>		
3	Jaringan dan Rekayasa Trafik (JRT-A)	Rabu	08.00 - 10.30	C-102	<a href="#">UBAH JADWAL</a>		
4	Jaringan dan Rekayasa Trafik (JRT-B)	Rabu	08.00 - 10.30	C-103	<a href="#">UBAH JADWAL</a>		
5	Jaringan Sensor Nirkabel dan IoT (JSNI-A)	Selasa	13.00 - 15.30	C-110	<a href="#">UBAH JADWAL</a>		
6	Pregabasi Gelombang (PG-A)	Jumat	08.00 - 10.30	C-107	<a href="#">UBAH JADWAL</a>		

Gambar 4. 10 Halaman Mata Kuliah – Dosen

## B. Hasil Halaman Daftar Mahasiswa

Halaman ini berisi daftar nama-nama mahasiswa beserta NRP dan keterangan semester. Pada halaman ini juga terdapat fitur untuk mendaftarkan sidik jari mahasiswa dan menghapusnya, dapat dilihat pada Gambar 4.11 berikut ini.

MATA KULIAH						DAFTAR MAHASISWA	LOGOUT
NO	Nama Mahasiswa	NRP	Semester	DAFTAR	HAPUS		
1	Ismail	0711204000001	8	<a href="#">DAFTAR</a>	<a href="#">HAPUS</a>		
2	Rahman	0711204000002	8	<a href="#">DAFTAR</a>	<a href="#">HAPUS</a>		
3	Fathis	0711204000003	8	<a href="#">DAFTAR</a>	<a href="#">HAPUS</a>		
4	Kurnia	0711204000004	8	<a href="#">DAFTAR</a>	<a href="#">HAPUS</a>		
5	Sriñiq	0711204000005	8	<a href="#">DAFTAR</a>	<a href="#">HAPUS</a>		
6	Ant	0711204000006	8	<a href="#">DAFTAR</a>	<a href="#">HAPUS</a>		

Gambar 4. 11 Halaman Daftar Mahasiswa - Dosen

## 4.2 Hasil Pengujian Integrasi Sistem

### 4.2.1 Hasil Pengujian Segmen Komunikasi dan Integrasi Sistem

Pengujian dilakukan dengan melihat bagaimana antar segmen terhubung. Segmen-segmen tersebut adalah sebagai berikut.

#### A. Hasil Segmen Komunikasi Sensor - Mikrokontroler

Sensor sidik jari FPM10A terhubung ke mikrokontroler NodeMCU ESP8266 dengan menggunakan kabel jumper, seperti pada Gambar 4.12 berikut.



Gambar 4.12 Koneksi Sensor ke Mikrokontroler

Untuk membuktikan bahwa sensor terdeteksi pada mikrokontroler, digunakan pemrograman pada Arduino IDE untuk mengonfirmasi jika sensor sidik jari benar-benar terhubung dan terdeteksi pada mikrokontroler. Hal ini dapat dilihat pada Gambar 4.13 berikut.

Jendela terminal serial yang menampilkan logika C dan hasil deteksi sensor. Logika C menunjukkan bahwa sensor berhasil terhubung dan mendekati. Hasil deteksi sensor menunjukkan bahwa sensor mendekati dan menemukan 10 template.

```
COM6
|
Connecting to aspire
.....
Connected
IP address: 192.168.137.118

Detecting fingerprint sensor...
Found fingerprint sensor!

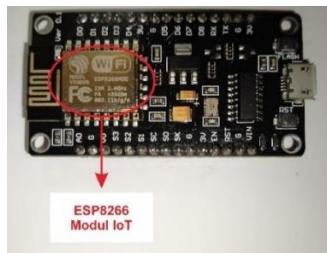
Sensor contains 10 templates
Waiting for valid finger...
200
()
FingerID=0
0
```

Gambar 4.13 Deteksi Sensor Fingerprint

Dapat disimpulkan bahwa sensor terhubung ke mikrokontroler menggunakan kabel jumper, dan dapat bekerja dengan baik.

### B. Hasil Segmen Komunikasi Mikrokontroler - Modul IoT

Salah satu kelebihan dari mikrokontroler NodeMCU ESP8266 adalah modul IoT yang sudah *embedded* ke dalam mikrokontroler, hal ini ditampilkan pada Gambar 4.14 berikut.



Gambar 4. 14 Modul IoT

Untuk menghubungkan mikrokontroler dengan modul IoT nya cukup dengan langsung menuliskan library yang diperlukan pada pemrograman Arduino IDE. Contoh library yang digunakan adalah ESP8266WiFi.h. Dapat disimpulkan bahwa mikrokontroler terhubung dengan modul IoT yang *embedded* di dalamnya, dan dapat bekerja dengan baik.

### C. Hasil Segmen Komunikasi Modul IoT - Internet

ESP8266 terhubung ke internet menggunakan WiFi (*wireless*) yang telah diatur terlebih dahulu dengan menggunakan pemrograman pada Arduino IDE. Untuk membuktikan hal tersebut, dapat dilihat pada Gambar 4.15 berikut.

```
Connecting to aspire...
.....
Connected
IP address: 192.168.137.118

Detecting fingerprint sensor...
Found fingerprint sensor!
```

Gambar 4. 15 Koneksi ke WiFi (internet)

Dapat disimpulkan bahwa modul IoT terhubung ke internet, dengan IP yang didapatkan *device* adalah 192.168.137.118, dan dapat bekerja dengan baik.

#### D. Hasil Segmen Komunikasi Internet - Database Server

Database yang digunakan adalah Firebase yang merupakan produk dari Google Cloud. Untuk mengakses database ini diharuskan untuk online atau terhubung ke internet, maka secara otomatis dari internet ke database terhubung secara wireless. Untuk mengakses firebase dapat dibuka pada alamat [www.console.firebaseio.google.com](http://www.console.firebaseio.google.com)

#### E. Hasil Segmen Komunikasi Database Server - Web Interface

Database server merupakan sumber data yang nantinya akan ditampilkan lewat web, sesuai apa yang diinginkan. Database yang digunakan adalah layanan dari Firebase yaitu *Cloud Firestore*. Sedangkan *web interface* diprogram menggunakan *software* Notepad++ dengan bahasa JavaScript. Pada Gambar 4.16 berikut adalah tampilan dari database.

The screenshot shows the Firebase Database interface. On the left, there's a sidebar with options like Project Overview, Authentication, Database, Storage, Hosting, Functions, ML Kit, Quality, Analytics, and Extensions. The main area shows a collection named 'Mahasiswa'. Inside this collection, there are several documents, each with fields such as 'ID', 'NRP', 'NIM', 'Email', 'Nama', and 'Alamat'. One document is expanded to show its internal structure, including arrays for 'kota' and 'alamat'.

Gambar 4. 16 Database pada Firebase

Dari keseluruhan pengujian segmen komunikasi sistem didapatkan hasil yang ditampilkan pada Tabel 4.1 berikut.

**Tabel 4. 1** Hasil Uji Segmentasi

No	Segmen	Konektivitas	Keterangan	Hasil
1.	Sensor – Mikrokontroler	<i>Wired</i>	Kabel Jumper	Terhubung dengan baik
2.	Mikrokontroler – Modul IoT	<i>Built-in</i>	Embedded	Terhubung dengan baik
3.	Modul IoT – Internet	<i>Wireless</i>	IEEE 802.11	Terhubung dengan baik
4.	Internet – Database Server	<i>Wireless</i>	HTTP	Terhubung dengan baik
5.	Database Server – Web Interface	<i>Wireless</i>	Back-end	Terhubung dengan baik

Dapat disimpulkan bahwa masing-masing segmen terhubung dengan baik dengan media yang berbeda-beda. Oleh karena itu, sistem dapat terintegrasi dengan baik.

## 4.2.2 Hasil Pengujian Operasional Sistem

### 4.2.2.1 Hasil Pengujian Fitur

Pada bagian ini akan menjelaskan fitur-fitur yang ada pada OASYS. Fitur-fitur tersebut diantaranya pendaftaran sidik jari, hapus sidik jari, ubah jadwal serta absensi dan monitoring.

#### A. Daftar Sidik Jari

Sebelum dapat melakukan absensi sehari-hari, terlebih dahulu diperlukan registrasi sidik jari untuk setiap mahasiswa. Sesuai pada diagram alur yang telah dijelaskan pada sub bab 3.5, proses registrasi sidik jari hanya bisa dilakukan oleh admin. Proses registrasi sidik jari dimulai dengan memilih tombol “DAFTAR” di website pada halaman

daftar mahasiswa, kemudian akan menampilkan proses pengambilan sidik jari seperti Gambar 4.17 berikut.

NO	Nama Mahasiswa	NIM	Semester	
1	Iswal	0711234000001	8	[DAFTAR] [HAPUS]
2	Ruhman	0711234000002	8	[DAFTAR] [HAPUS]
3	Fathin	0711234000003	8	[DAFTAR] [HAPUS]
4	Kurnia		8	[DAFTAR] [HAPUS]
5	Shidqy		8	[DAFTAR] [HAPUS]
6	Aidil	0711234000006	8	[DAFTAR] [HAPUS]
7	Cahyo	0711234000007	8	[DAFTAR] [HAPUS]
8	Jakber	0711234000008	8	[DAFTAR] [HAPUS]
9	Faizi	0711234000009	8	[DAFTAR] [HAPUS]
10	Ariyan	0711234000010	8	[DAFTAR] [HAPUS]

Gambar 4. 17 Tampilan Daftar Sidik Jari

Pada saat yang bersamaan ketika memilih tombol “DAFTAR” pada website, sensor sidik jari yang ada pada device sedang dalam posisi *scanning* atau pengambilan sidik jari sesuai pada Gambar 4.18 berikut.



Gambar 4. 18 Proses scanning sidik jari

Kemudian dilakukan tap pada sensor untuk merekam sidik jari yang didaftarkan, ini dilakukan 2x untuk konfirmasi. Setelah proses pengambilan sidik jari selesai, maka layar akan menampilkan bahwa sidik jari telah tersimpan, proses registrasi dijelaskan pada Gambar 4.19, Gambar 4.20, Gambar 4.21 dan Gambar 4.22 berikut.



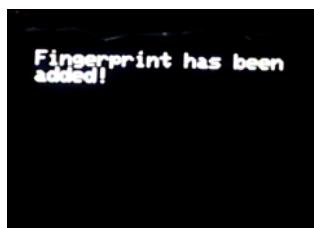
Gambar 4. 19 Sidik jari terbaca pada scanning pertama



Gambar 4. 20 Lepaskan jari dari sensor



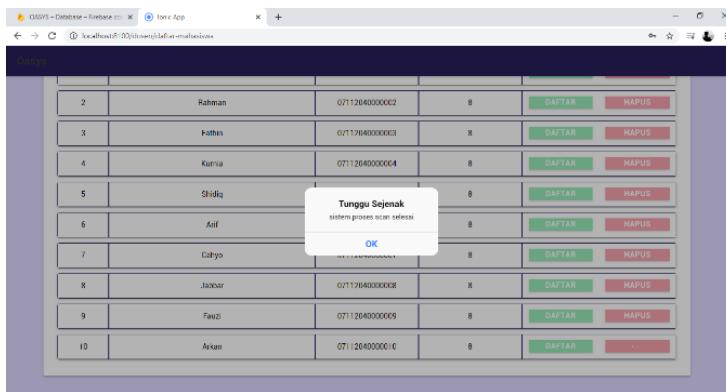
Gambar 4. 21 Sidik jari terbaca pada scanning kedua



Gambar 4. 22 Sidik jari tersimpan

## B. Hapus Sidik Jari

Fitur hapus sidik jari hanya bisa digunakan ketika pada user tersebut sudah pernah mendaftarkan sidik jarinya. Proses hapus sidik jari hanya bisa dilakukan oleh admin. Proses hapus sidik jari dimulai dengan memilih tombol “HAPUS” di website pada halaman daftar mahasiswa, kemudian akan menampilkan proses penghapusan sidik jari seperti Gambar 4.23 berikut.



Gambar 4. 23 Tampilan web saat menekan tombol hapus

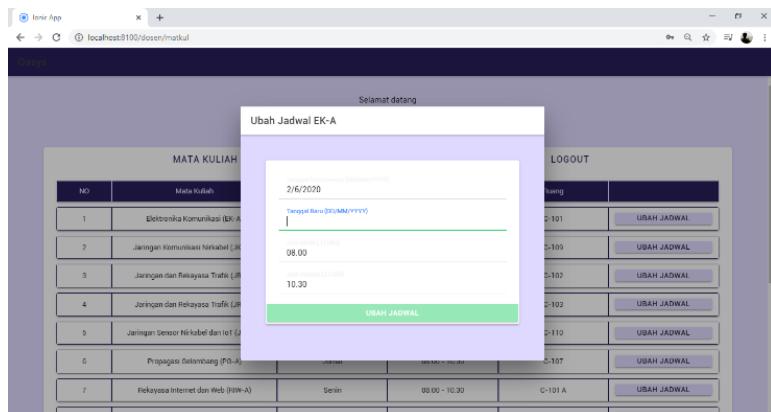
Disaat yang bersamaan, device akan memproses penghapusan sidik jari. Apabila telah selesai maka layar akan menampilkan pemberitahuan sidik jari telah dihapus seperti Gambar 4.24 berikut.



Gambar 4. 24 Sidik jari telah dihapus dari database

### C. Ubah Jadwal

Fitur ubah jadwal hanya bisa digunakan oleh dosen/admin. Fitur ini memungkinkan dosen untuk mengubah jadwal mengajar kuliah untuk setiap mata kuliahnya. Hari dan jam mengajar yang akan diubah sangat fleksibel, bisa kapan saja sesuai yang dikehendaki oleh dosen. Proses ubah jadwal dimulai dengan memilih tab MATA KULIAH pada website, kemudian klik pada tombol “UBAH JADWAL” disamping mata kuliah yang akan diubah. Lalu menuliskan tanggal mengajar sesuai jadwal sebelumnya, jika sistem mendeteksi format penulisan tanggal yang benar, maka secara otomatis akan terbuka tampilan untuk menuliskan tanggal baru beserta jam dimulai dan jam selesai, seperti yang ditampilkan pada Gambar 4.25 berikut.



Gambar 4. 25 Tampilan web saat ubah jadwal

Setelah dilakukan perubahan pada jadwal mata kuliah, maka secara otomatis database akan berubah.

### D. Absensi dan Monitoring

Fitur absensi adalah fitur utama dari sistem absensi OASYS ini. Setelah sidik jari telah tersimpan sesuai pada nama masing-masing mahasiswa, maka proses absensi baru dapat dilakukan. Dalam fitur absensi ini telah diatur untuk dapat melakukan absensi hanya pada waktu 30 menit sebelum dan 30 menit sesudah jam mata kuliah dimulai, selain itu akan dianggap tidak hadir.

Fitur absensi merekam data masuk kemudian menampilkannya pada halaman mata kuliah di web. Adapun data yang tercantum adalah nama mata kuliah, tanggal, jam masuk, jam absensi, keterlambatan dan keterangan. Jam masuk adalah jam yang sesuai pada jadwal mata kuliah, sedangkan jam absensi adalah waktu disaat mahasiswa melakukan absensi, dan secara otomatis akan menampilkan keterlambatan dalam satuan menit. Apabila mahasiswa melakukan absensi masih dalam jangka waktu 30 menit sebelum dan 30 menit sesudah jam masuk, maka pada kolom keterangan akan menampilkan hadir. Sebaliknya apabila mahasiswa melakukan absensi diluar jangka waktu tersebut maka pada kolom keterangan akan menampilkan alfa.

Pada Gambar 4.26 berikut adalah tampilan pada *device* ketika proses absensi berlangsung.



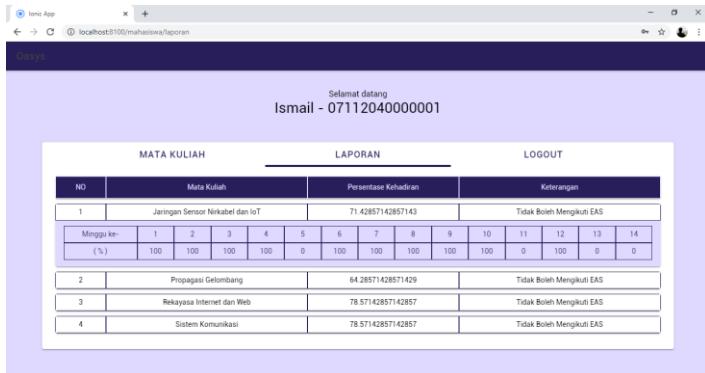
**Gambar 4. 26 Posisi *idle***

Ketika layar menampilkan Gambar 4.26 maka absensi sudah terrekam. Kemudian data dikirim ke database untuk ditampilkan pada web. Gambar 4.27 berikut adalah tampilan absensi pada satu mata kuliah.

A screenshot of a web browser window. The address bar shows 'localhost:100/mahasiswa/mata-kuliah'. The page header includes 'Salamat datang Ismail - 0711204000001', 'MATA KULIAH', 'LAPORAN', and 'LOGOUT'. Below the header is a table with columns: NO, NAMA, HARI, JAM, DOSEN, and RUANG. The first row shows 'Jaringan Sensor Nirkabel dan IoT' with 'Rabu' in the HARI column. The table body contains 8 rows of attendance data. The last row is partially visible.

**Gambar 4. 27 Halaman Mata Kuliah – Mahasiswa**

Setelah proses tersebut, disaat yang bersamaan data juga diproses untuk ditampilkan secara keseluruhan atau akumulasi dari kehadiran mahasiswa, ini ditampilkan pada halaman laporan seperti pada Gambar 4.28 berikut.



Gambar 4. 28 Halaman Laporan – Mahasiswa

Dari keseluruhan pengujian fitur-fitur sistem OASYS, didapatkan hasil masing-masing fitur yang ditampilkan pada Tabel 4.2 berikut.

Tabel 4. 2 Hasil Pengujian Fitur

No.	Fitur	Hasil
1.	Pendaftaran Sidik Jari	Bekerja dengan baik
2.	Hapus Sidik Jari	Bekerja dengan baik
3.	Ubah Jadwal Dosen	Bekerja dengan baik
4.	Absensi dan Monitoring	Bekerja dengan baik

#### 4.2.2.2 Hasil Pengujian Jenis Kelas

Seperti penjelasan pengujian yang telah dibahas pada sub bab 3.7, pada pengujian ini dibagi menjadi 2 bagian, yaitu absensi kelas paralel dan absensi kelas seri.

##### A. Absensi Kelas Paralel

Pada pengujian ini diambil kasus pada hari Kamis, 21 Mei 2020 dimana mahasiswa bernama Ismail dan Fathin mengambil mata kuliah yang sama, pada waktu yang sama namun berada pada kelas yang

berbeda. Pengujian dilakukan dengan melakukan tap sidik jari secara bersamaan pada 2 *devices* dengan memisalkan mahasiswa bernama Ismail menggunakan ibu jari kanan sedangkan mahasiswa bernama Fathin menggunakan jari tengah kiri. Pada Gambar 4.29 berikut adalah pengujian tap sidik jari paralel yang telah dilakukan.



Gambar 4. 29 Tap Sidik Jari Paralel

Tap sidik jari telah dilakukan, dan kedua *devices* mampu membaca sidik jari dengan baik dan mengirimkan data ke database. Dapat dilihat pada Gambar 4.30 dan Gambar 4.31 bahwa kedua mahasiswa tersebut telah melakukan absensi pada jam yang sama yaitu pukul 7.56.

No	Tanggal	Minggu	Waktu Masuk	Waktu Absen	Keterangan	%
1	14/02/2020	1	08.00	07.56	Normal	100%
2	15/02/2020	2	08.00	08.00	Normal	100%
3	23/02/2020	3	08.00	07.56	Normal	100%
4	24/02/2020	4	08.00	07.56	Normal	100%
5	01/03/2020	5	08.00	07.56	Normal	100%
6	14/03/2020	6	08.00	07.56	Normal	100%
7	21/03/2020	7	08.00	07.56	Normal	100%

Gambar 4. 30 Halaman Mata Kuliah Mahasiswa 1

The screenshot shows a web application window titled 'OASYS - Dekanats - Fakultas'. The main content area displays a table with two sections: 'MATA KULIAH' and 'LAPORAN'. The 'MATA KULIAH' section contains three rows of subject information:

NO	NAMA	HARI	JAM	DOSEN	RUANG
1	Jaringan Komunikasi Nirkabel	Senin	13.00 - 15.30	9	C-109
2	Jaringan dan Kelebihan Protokol	Rabu	08.00 - 10.30	4	C-102
3	Sistem Komunikasi	Kamis	08.00 - 10.30	6	C-104

The 'LAPORAN' section below contains a table with columns: No, Tanggal, Minggu, Waktu Masuk, Waktu Absen, Keterlambatan, Keteringatan, and %. The data is as follows:

No	Tanggal	Minggu	Waktu Masuk	Waktu Absen	Keterlambatan	Keteringatan	%
1	9/5/2020	1	08.00	8.00	0 menit	Hadir	100%
2	16/4/2020	2	08.00	8.03	3 menit	Hadir	100%
3	7/3/2020	3	08.00	7.59	1 menit	Hadir	100%
4	30/4/2020	4	08.00	7.57	-3 menit	Hadir	100%
5	7/5/2020	5	08.00	8.07	7 menit	Hadir	100%
6	14/5/2020	6	08.00	8.09	9 menit	Hadir	100%
7	21/5/2020	7	08.00	7.56	4 menit	Hadir	100%

**Gambar 4. 31** Halaman Mata Kuliah Mahasiswa 2

Pengujian selesai. Dapat disimpulkan bahwa sistem absensi OASYS dapat berjalan dengan baik pada kasus absensi kelas paralel.

### B. Absensi Kelas Seri

Pada pengujian ini diambil kasus pada hari Kamis, 21 Mei 2020 dimana mahasiswa bernama Fathin mengambil 2 mata kuliah yang berada pada hari yang sama namun pada jam yang berbeda (sesi 1 dan sesi 2). Pengujian dilakukan dengan melakukan tap sidik jari pada pukul 7.56 sesuai jangka waktu aman pada sesi 1. Pada Gambar 4.32 berikut adalah pengujian tap sidik jari pada sesi 1 yang telah dilakukan.



**Gambar 4. 32** Tap Sidik Jari Sesi 1

Tap sidik jari telah dilakukan, *device* dapat membaca sidik jari dengan baik dan mengirimkan data ke database untuk selanjutnya ditampilkan pada web. Dapat dilihat pada Gambar 4.33 bahwa di halaman mata kuliah pada mata kuliah Sistem Komunikasi tanggal 21/5/2020 telah dilakukan absensi pada pukul 7.56.

MATA KULIAH		LAPORAN			LOGOUT	
NO	NAMA	HARI	JAM	DOSSEN	RUANG	
1	Jaringan Komunikasi Nirkabel	Senin	13.00 - 15.30	9	C-109	
2	Jaringan dan Perekayasa Trafik	Rabu	08.00 - 10.30	4	C-102	
3	Sistem Komunikasi	Kamis	08.00 - 10.30	5	C-104	

No	Tanggal	Minggu	Waktu Masuk	Waktu Absen	Keterlambatan	Keterangan	%
1	9/4/2020	1	08.00	8.00	0 menit	Hadir	100%
2	16/4/2020	2	08.00	8.03	3 menit	Hadir	100%
3	23/4/2020	3	08.00	7.59	1 menit	Hadir	100%
4	30/4/2020	4	08.00	7.57	>3 menit	Hadir	100%
5	7/5/2020	5	08.00	8.07	7 menit	Hadir	100%
6	14/5/2020	6	08.00	8.09	9 menit	Hadir	100%
7	21/5/2020	7	08.00	7.56	4 menit	Hadir	100%

Gambar 4. 33 Halaman Mata Kuliah Sistem Komunikasi

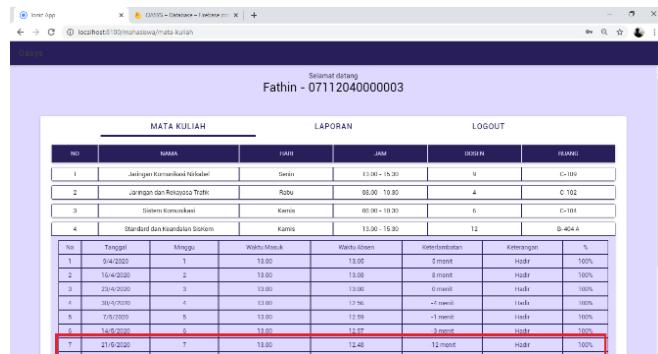
Selanjutnya dilakukan tap sidik jari yang dapat dilihat pada Gambar 4.34, pada pukul 12.48 sesuai jangka waktu aman pada sesi 2.



Gambar 4. 34 Tap Sidik Jari Sesi 2

Tap sidik jari telah dilakukan, *device* dapat membaca sidik jari dengan baik dan mengirimkan data ke database untuk selanjutnya

ditampilkan pada web. Dapat dilihat pada Gambar 4.35 bahwa di halaman mata kuliah pada mata kuliah Standard dan Keandalan SisKom tanggal 21/5/2020 telah dilakukan absensi pada pukul 12.48.



The screenshot shows a web browser window with the following details:

- Title Bar:** OASYS - Database - Firefox
- Address Bar:** localhost:100/mazabsysw/mata\_kuliah
- Page Content:** A login form with fields for Nama and Password. Below the form is a message: "Selamat datang Fathin - 0711204000003".
- Navigation:** Links for MATA KULIAH, LAPORAN, and LOGOUT.
- MATA KULIAH Table:** A table with columns NO, NAMA, HARI, JAM, OKT/N, and HIJRIAH. It lists four entries:

NO	NAMA	HARI	JAM	OKT/N	HIJRIAH
1	Jurusan Komunikasi Nirkabel	Sabtu	13:00 - 15:30	9	C-109
2	Jurusan dan Rekayasa Traffic	Rabu	08:00 - 10:30	4	C-102
3	Sistem Komputer	Kamis	08:00 - 10:30	6	C-101
4	Standard dan Keandalan Sistkom	Kamis	13:00 - 15:30	12	D-404-A

- LAPORAN Table:** A table with columns NO, Tanggal, Mingezi, Waktu-Minut, Kebut-Harren, Keterhadapan, Keteringinan, and %. It lists six entries for May 21, 2020:

NO	Tanggal	Mingezi	Waktu-Minut	Kebut-Harren	Keterhadapan	Keteringinan	%
1	01/4/2020	1	10:00	13:00	2 meute	tidak	100%
2	16/4/2020	2	10:00	13:00	2 meute	tidak	100%
3	23/4/2020	3	10:00	13:00	2 meute	tidak	100%
4	30/4/2020	4	10:00	13:00	<1 meute	tidak	100%
5	7/5/2020	5	10:00	13:00	<1 meute	tidak	100%
6	14/5/2020	6	10:00	13:00	<1 meute	tidak	100%
T	21/5/2020	T	10:00	12:48	3 meute	tidak	100%

Gambar 4. 35 Halaman Mata Kuliah Standard dan Keandalan SisKom

Pengujian selesai. Dapat disimpulkan bahwa sistem absensi OASYS dapat berjalan dengan baik dalam kasus absensi kelas serial.

Dari keseluruhan pengujian jenis kelas pada sistem OASYS, didapatkan hasil masing-masing pengujian yang ditampilkan pada Tabel 4.3 berikut.

Tabel 4. 3 Hasil Pengujian Jenis Kelas

No.	Jenis Kelas	Hasil
1.	Kelas Serial	Absensi dapat dilakukan dan bekerja dengan baik
2.	Kelas Paralel	Absensi dapat dilakukan dan bekerja dengan baik

#### 4.2.3 Hasil Pengujian Keandalan Sistem

Pengujian keandalan sistem dibagi menjadi 2 bagian. Hasil dari pengujian tersebut adalah sebagai berikut.

#### **4.2.3.1 Hasil Pengujian Kecepatan Proses Pembacaan Sidik Jari**

Pengujian dilakukan dengan mengamati lama waktu yang diperlukan sensor untuk membaca sidik jari. Hasil pengamatan dapat dilihat pada Tabel 4.4 berikut.

**Tabel 4. 4** Hasil Pengujian Kecepatan Pembacaan Sidik Jari

No.	Nama ID	Waktu (s)
1.	Ismail	2,45 detik
2.	Rahman	2,11 detik
3.	Fathin	2,38 detik
4.	Kurnia	2,27 detik
5.	Shidiq	2,78 detik
6.	Arif	2,09 detik
7.	Cahyo	2,64 detik
8.	Jabbar	2,23 detik
9.	Fauzi	2,40 detik
10.	Arkan	2,08 detik
Rata-rata		2,343 detik

Dari hasil pengujian didapatkan bahwa rata-rata waktu yang dibutuhkan oleh sensor dalam membaca sidik jari dan memprosesnya adalah 2,343 detik. Dari data tersebut dapat disimpulkan bahwa setiap mahasiswa melakukan absensi selama 2,343 detik, maka dalam rentang waktu 60 menit sesuai dengan waktu absensi yang dibuka, maksimal sebanyak 1.536 nama mahasiswa dapat melakukan absensi pada satu device (satu kelas).

#### **4.2.3.2 Hasil Pengujian Ketepatan Penyimpanan Data**

Pengujian dilakukan dengan melakukan absensi sesuai jadwal, kemudian melihat ketersesuaian data yang masuk dari sensor saat melakukan absensi dengan data yang terbarui pada database. Pada pengujian ini dilakukan absensi oleh beberapa ID mahasiswa pada mata kuliah Rekayasa Internet dan Web kelas A (RIW-A), mata kuliah Jaringan Komunikasi Nirkabel kelas A (JKN-A), mata kuliah Sistem Broadcast

kelas A (SB-A) serta mata kuliah Sistem Komunikasi kelas A (SK-A) yang dilaksanakan Hari Senin, 29 Juni 2020 sampai dengan Hari Kamis, 2 Juli 2020. Hasil pengujian dapat dilihat pada Tabel 4.5 berikut.

**Tabel 4. 5** Hasil Pengujian Ketepatan Pengiriman Data

No.	Nama ID	Mata Kuliah	Hari, Tanggal	Waktu Absensi
1.	Ismail	RIW-A	Senin, 29-6-2020	07.55 WIB
2.	Kurnia	RIW-A	Senin, 29-6-2020	07.58 WIB
3.	Rahman	JKN-A	Senin, 29-6-2020	12.51 WIB
4.	Fathin	JKN-A	Senin, 29-6-2020	12.54 WIB
5.	Rahman	SB-A	Rabu, 1-7-2020	13.02 WIB
6.	Kurnia	SB-A	Rabu, 1-7-2020	13.05 WIB
7.	Ismail	SK-A	Kamis, 2-7-2020	07.58 WIB
8.	Fathin	SK-A	Kamis, 2-7-2020	08.03 WIB

Sedangkan untuk ketersesuaian data yang terkirim pada database dapat dilihat beberapa contoh capture halaman web untuk masing-masing nama ID pada Gambar 4.36 dan Gambar 4.37 berikut

The screenshot shows a web application interface. At the top, there's a browser-like header with tabs and navigation buttons. Below this, a purple navigation bar with the word "mhsys". The main content area has a light blue background.

At the top of the content area, there is a message: "Selamat datang" followed by the name "Rahman - 07112040000002".

Below the message, there are two tables:

MATA KULIAH		LAPORAN		LOGOUT	
NO	NAMA	HARI	JAM	DOSEN	RUANG
1	Jaringan Komunikasi Nirkabel	Senin	13.00 - 15.30	9	C-109

No	Tanggal	Minggu	Waktu Masuk	Waktu Absen	Keterlambatan	Keterangan	%
1	6/4/2020	1	13.00	13.00	0 menit	Hadir	100%
2	13/4/2020	2	13.00	13.13	13 menit	Hadir	100%
3	20/4/2020	3	13.00	13.05	5 menit	Hadir	100%
9	1/6/2020	9	13.00	13.31	31 menit	Alpha	0%
10	8/6/2020	10	13.00	13.18	18 menit	Hadir	100%
11	15/6/2020	11	13.00	13.19	19 menit	Hadir	100%
12	22/6/2020	12	13.00	13.12	12 menit	Hadir	100%
12	29/6/2020	13	13.00	12.51	-9 menit	Hadir	100%
14	6/7/2020	14	13.00	unset	7 menit	Alpha	0%

**Gambar 4. 36** Data Tersimpan Pada Database

MATA KULIAH		LAPORAN			LOGOUT		
NO	NAMA	HARI	JAM	DOSEN	RUANG		
1	Jaringan Komunikasi Nirkabel	Senin	13.00 - 15.30	9	C-109		
No	Tanggal	Minggu	Waktu Masuk	Waktu Absen	Keterlambatan	Keterangan	%
1	6/4/2020	1	13.00	13.12	12 menit	Hadir	100%
2	13/4/2020	2	13.00	13.15	15 menit	Hadir	100%
3	20/4/2020	3	13.00	13.05	5 menit	Hadir	100%
9	1/6/2020	9	13.00	13.18	18 menit	Hadir	100%
10	8/6/2020	10	13.00	13.15	15 menit	Hadir	100%
11	15/6/2020	11	13.00	-	NaN menit	Alpha	0%
12	22/6/2020	12	13.00	13.15	15 menit	Hadir	100%
13	29/6/2020	13	13.00	12.54	-6 menit	Hadir	100%
14	6/7/2020	14	13.00	unset	7 menit	Alpha	0%

Gambar 4.37 Data Tersimpan Pada Database

Dari data yang diperoleh sesuai pada Tabel 4.5 serta Gambar 4.36 dan Gambar 4.37 maka dapat disimpulkan bahwa pengiriman data dapat terkirim dengan baik, data yang dikirimkan oleh sensor saat dilakukan absensi telah sesuai dengan data yang ditampilkan pada halaman web yang berasal dari database.

### 4.3 Hasil Pengukuran Kualitas Layanan

Pengukuran kualitas layanan pada sistem absensi IoT dilakukan dengan meng-*capture* jaringan pada wireshark yang ada pada setiap skenario pengukuran, seperti yang dijelaskan pada Sub Bab 3.8. Berikut hasil dari pengukuran dan analisis jaringan pada sistem absensi IoT.

#### 4.3.1 Hasil Pengujian Pengaruh Jarak Ruang Kelas

Pada pengujian ini dilakukan pengukuran trafik jaringan dari IP *client (device)* ke IP *server (pc)* yang dibagi menjadi 2, yaitu berdasarkan jarak antara device dengan router wifi, dengan jarak 5 meter dan 10 meter. Pengujian ini dilakukan dengan kondisi *device* sedang mengirim data secara terus-menerus (*loop*).

Dari keseluruhan pengukuran *throughput*, *delay*, *jitter* dan *packet loss* berdasarkan pengaruh jarak ruang kelas, dengan jarak masing-masing 5 meter dan 10 meter, diperoleh hasil yang ditampilkan pada Tabel 4.4 sebagai berikut.

**Tabel 4. 6** Hasil Pengujian Pengaruh Jarak Ruang Kelas

No	Parameter	Jarak	
		5 meter	10 meter
1.	Throughput (bps)	3029 bps	2952 bps
2.	Delay (ms)	49,38073 ms	51,18321 ms
3.	Jitter (ms)	0,024066 ms	0,03635 ms
4.	Packet Loss (%)	0 %	0 %

Dari hasil pengukuran dapat dilihat bahwa nilai *throughput* pada saat jarak *device* ke router wifi sejauh 10 meter memiliki nilai yang lebih kecil, hal ini berarti kecepatan pengiriman data saat jarak 10 meter lebih lambat daripada saat jarak 5 meter. *Delay* yang didapatkan pada saat jarak 10 meter lebih besar daripada saat 5 meter, hal ini berarti pada jarak 10 meter waktu tunda pengiriman paket data lebih besar, sehingga menyebabkan paket akan lebih lama terkirim. Ini juga berlaku pada *jitter*. Nilai *jitter* didapatkan berdasarkan banyaknya variasi *delay*, karena dalam pengiriman data ada banyak transmisi data yang dilakukan sehingga menyebabkan cukup tingginya nilai *jitter*. Dari data tersebut didapatkan nilai *packet loss* sebesar 0%, yang artinya tidak ada paket data yang gagal terkirim pada kedua jarak pengujian.

#### 4.3.2 Hasil Pengujian Pengaruh Jumlah Ruang Kelas

Pada pengujian ini dilakukan pengukuran trafik jaringan dengan menggunakan 2 *device* dan jarak yang digunakan adalah 5 meter. Pengujian dilakukan 2 kali dengan kondisi device mengirim data secara terus-menerus (*loop*) dengan kurun waktu selama 5 menit.

Dari keseluruhan pengukuran *throughput*, *delay*, *jitter* dan *packet loss* berdasarkan pengaruh jumlah ruang kelas, dengan jarak 5 meter pada kedua *devices*, diperoleh hasil sesuai pada Tabel 4.5 sebagai berikut.

**Tabel 4. 7** Hasil Pengujian Pengaruh Jumlah Ruang Kelas

No	Parameter	Jumlah Device Terhubung	
		1 buah	2 buah
1.	Throughput (bps)	3029 bps	3028 bps
2.	Delay (ms)	49,38073 ms	52,69275 ms
3.	Jitter (ms)	0,024067 ms	0,065142 ms
4.	Packet Loss (%)	0 %	0 %

Dari hasil pengukuran dapat dilihat bahwa nilai *throughput* pada saat jumlah device yang terhubung ke router wifi sebanyak 2 buah memiliki nilai yang lebih kecil, walaupun hanya sedikit. Hal ini berarti kecepatan pengiriman data saat terdapat 2 *devices* terhubung pada satu WiFi yang sama adalah lebih lambat daripada saat hanya 1 *device* yang terhubung ke WiFi. *Delay* yang didapatkan pada saat 2 *devices* terhubung lebih besar daripada saat hanya 1 *device* terhubung, hal ini berarti pada saat 2 *devices* terhubung bersamaan, waktu tunda pengiriman paket data lebih besar, sehingga menyebabkan paket akan lebih lama terkirim. Ini juga berlaku pada jitter. Sedangkan nilai *packet loss* pada kedua kondisi didapatkan sebesar 0%, yang berarti tidak ada paket yang gagal terkirim.

#### 4.3.3 Hasil Pengujian Pengaruh Waktu Pengiriman Data

Pada pengujian ini dilakukan pengukuran trafik jaringan dengan menggunakan 2 *devices* dalam kondisi *loop* mengirim data setiap 5 detik. Pengujian dilakukan pada waktu atau jam yang berbeda-beda, yaitu pada jam 8.00, jam 13.00 dan jam 21.00. Pengujian dilakukan selama 2 hari berturut-turut yaitu Hari Kamis, 28 Mei 2020 sampai dengan Hari Jumat, 29 Mei 2020, dengan masing-masing waktu pengambilan data selama 5 menit. Keseluruhan hasil pengujian pengaruh waktu pengiriman data dapat dilihat pada Tabel 4.6 berikut.

**Tabel 4. 8** Hasil Pengujian Pengaruh Waktu Pengiriman Data

No	Parameter	Waktu Pengiriman Data		
		08.00	13.00	21.00
1.	Throughput (bps)	3133,3 bps	3181,3 bps	3209 bps
2.	Delay (ms)	58,319 ms	57,981 ms	61,592 ms
3.	Jitter (ms)	0,0608 ms	0,2809 ms	0,0589 ms
4.	Packet Loss (%)	0 %	0 %	0 %

Dari hasil pengukuran dapat dilihat bahwa nilai *throughput* pada saat pengiriman data pukul 21.00 memiliki nilai terbesar, hal ini berarti rata-rata kecepatan pengiriman data saat pukul 21.00 adalah yang tercepat dibandingkan dengan pukul 08.00 dan pukul 13.00. Namun ditemui juga bahwa *delay* pada pukul 21.00 memiliki nilai terbesar, hal ini berarti pada saat pukul 21.00 terdapat rata-rata waktu tunda pengiriman paket data yang lebih besar dibandingkan dengan 2 waktu lainnya, walaupun kecepatan pengiriman data per detiknya lebih cepat. Hal ini berlaku juga pada *jitter*. Sedangkan untuk *packet loss* didapatkan semua nilai sebesar 0%, yang berarti tidak ada paket yang gagal dikirim baik pada pukul 08.00, pukul 13.00 maupun pukul 21.00.

## 4.4 Analisa Hasil dan Pembahasan

### 4.4.1 Hasil Pengujian Sistem

Sistem absensi OASYS telah selesai dibuat sesuai dengan rancangan yang ada. Dalam pembuatan *prototype device* digunakan modul IoT yang sudah terhubung ke mikrokontroler secara *embedded* yaitu dengan NodeMCU ESP8266. Kemudian ditambahkan LCD Display untuk menampilkan kondisi pada setiap penggunaan alat, dan lampu LED serta buzzer untuk membantu memberikan *feedback* notifikasi kepada pengguna saat menggunakan alat. Keseluruhan *prototype device* ini diprogram menggunakan Arduino IDE, termasuk pengiriman ke server

melalui modul IoT. Pada pembuatan *prototype* ini dibuat 2 *devices* sebagai representasi banyak ruang kelas.

Setelah selesai pembuatan *prototype device* absensi, dilakukan pembuatan web dengan menuliskan program pada Notepad++. Pembuatan web ini menggunakan bahasa JavaScript yang kemudian dihubungkan ke Firebase Cloud, yang merupakan database server. Device dihubungkan dengan web menggunakan protokol HTTP, untuk mengirim dan menerima data. Device dihubungkan dengan api server pada cloud function milik firebase agar dapat berkomunikasi.

Dalam pengujian sistem dibagi menjadi pengujian integrasi sistem dan pengujian operasional sistem. Dari pengujian integrasi sistem didapatkan hasil bahwa komponen antar sistem terhubung dengan baik, baik secara *wired* maupun secara *wireless*, serta terintegrasi untuk bisa bekerja sebagai satu sistem.

Dari pengujian operasional sistem diperoleh hasil bahwa fitur-fitur yang tersedia di website telah sesuai dengan rancangan yang dibuat dan dapat berkomunikasi dengan *device*. Dalam pengujian fitur diperoleh hasil bahwa penggunaan sistem absensi OASYS dapat dibagi menjadi 3 bagian, yaitu registrasi/penghapusan, proses absensi, serta monitoring. Sedangkan dari sisi pengguna dibagi menjadi 2, yaitu mahasiswa dan admin/dosen.

Secara keseluruhan pengujian sesuai dengan kondisi kelas, dapat disimpulkan bahwa sistem absensi OASYS dapat berjalan dengan baik pada kondisi kelas paralel, dimana banyak pengguna/mahasiswa melakukan absensi pada kelas yang berbeda-beda dalam waktu yang bersamaan. Serta sistem absensi OASYS dapat berjalan dengan baik pada kondisi kelas serial, dimana dalam satu hari, satu mahasiswa dapat melakukan absensi 2 kali atau lebih sesuai pada jadwal.

Pengukuran QoS telah dilakukan dengan menggunakan *software* wireshark dan dalam penghitungannya memerlukan bantuan dari MS. Excel. Dari hasil yang didapatkan masing-masing parameter pengukuran didapatkan hasil yang berbeda-beda sesuai pada skenario pengujian, yang akan dijelaskan lebih rinci pada subbab berikutnya.

## 4.5 Perbandingan Performansi Sistem

Pada subbab ini akan dilakukan perbandingan kinerja atau performansi dari sistem OASYS yang didapatkan dari hasil pengukuran kualitas layanan (QoS), beserta penjelasan dari masing-masing kondisi sesuai pada skenario.

#### 4.5.1 Perbandingan Parameter Hasil Pengukuran

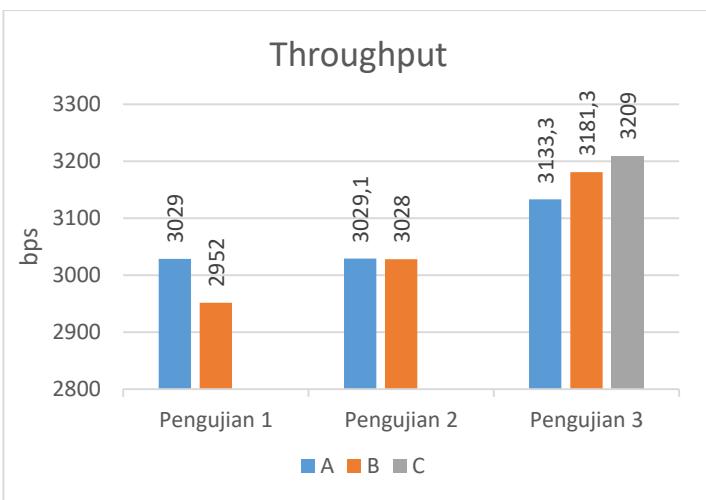
Pada uraian berikut akan dijelaskan perbandingan hasil pengukuran antar pengujian untuk parameter QoS yang sama, yang telah dijelaskan sebelumnya yaitu *throughput*, *delay*, *jitter* dan *packet loss*. Sebelum memulai pembahasan mengenai hasil pengukuran yang didapatkan, akan didefinisikan terlebih dahulu masing-masing kondisi pengukuran dalam bentuk kode yang dapat dilihat pada Tabel 4.7 berikut.

**Tabel 4.9** Deskripsi Kondisi Pengukuran

No	Skenario Pengukuran	Kondisi	Kode
1.	Pengaruh Jarak Ruang Kelas	Jarak 5 meter, 1 <i>device</i> terhubung, diukur pada siang hari selama 5 menit	1A
		Jarak 10 meter, 1 <i>device</i> terhubung, diukur pada siang hari selama 5 menit	1B
2.	Pengaruh Jumlah Ruang Kelas	1 <i>device</i> terhubung pada jarak 5 meter, diukur pada siang hari selama 5 menit	2A
		2 <i>devices</i> terhubung pada jarak 5 meter, diukur pada siang hari selama 5 menit	2B
3.	Pengaruh Waktu Pengiriman Data	Pukul 08.00, 2 <i>devices</i> terhubung pada jarak 5 meter, diukur selama 5 menit	3A
		Pukul 13.00, 2 <i>devices</i> terhubung pada jarak 5 meter, diukur selama 5 menit	3B
		Pukul 21.00, 2 <i>devices</i> terhubung pada jarak 5 meter, diukur selama 5 menit	3C

##### A. Perbandingan Parameter *Throughput*

Dalam pengukuran kualitas layanan pada ketiga pengujian dengan setiap kondisinya, didapatkan nilai *throughput* dari masing-masing kondisi seperti yang ditampilkan dalam grafik pada Gambar 4.36 berikut.



**Gambar 4. 38** Grafik Perbandingan Pengukuran Throughput

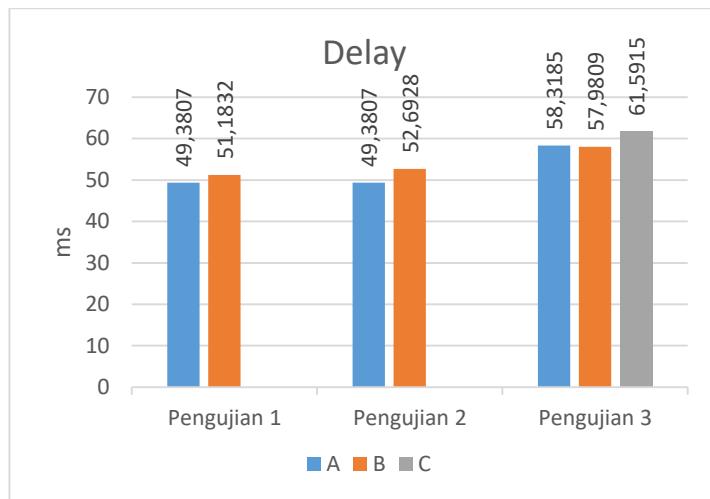
Dari grafik yang ditampilkan, dapat dilihat bahwa hasil pengukuran *throughput* memiliki nilai yang berbeda-beda pada setiap kondisi, namun masih dalam range yang kecil atau hampir sama. Dari pengujian 1 dapat disimpulkan bahwa pada saat jarak ruang kelas atau *device* dengan router WiFi sejauh 5 meter memiliki nilai *throughput* yang lebih besar daripada saat jarak 10 meter, artinya pada saat jarak 5 meter kecepatan pengiriman data lebih cepat daripada saat jarak 10 meter.

Dari pengujian 2 dapat dilihat bahwa nilai *throughput* lebih besar saat hanya ada 1 *device* yang terhubung ke WiFi, yang berarti disimpulkan bahwa banyaknya jumlah ruang kelas atau *devices* mempengaruhi kecepatan pengiriman data, walaupun hanya memiliki selisih nilai yang sangat kecil.

Dari pengujian 3 dapat dilihat bahwa nilai *throughput* tertinggi terjadi saat pengiriman data pada pukul 21.00, dimana penggunaan internet pada jaringan WiFi cenderung lebih sedikit, sehingga pengiriman data dapat lebih cepat jika dibandingkan dengan waktu yang lain yaitu pada pukul 08.00 dan 13.00.

## B. Perbandingan Parameter Delay

Dalam pengukuran kualitas jaringan pada ketiga pengujian dengan setiap kondisinya, didapatkan nilai *delay* dari masing-masing kondisi seperti yang ditampilkan dalam grafik pada Gambar 4.37 berikut.



Gambar 4.39 Grafik Perbandingan Pengukuran Delay

Dari grafik yang ditampilkan, dapat dilihat bahwa hasil pengukuran delay memiliki nilai yang berbeda-beda pada setiap kondisi walaupun selisih nilai pada masing-masing kondisi hanyalah sedikit, dengan rata-rata 54,36 ms Pada pengujian 1, delay lebih besar saat jarak 10 meter yang berarti waktu tunda pengiriman saat jarak 10 meter adalah lebih lama.

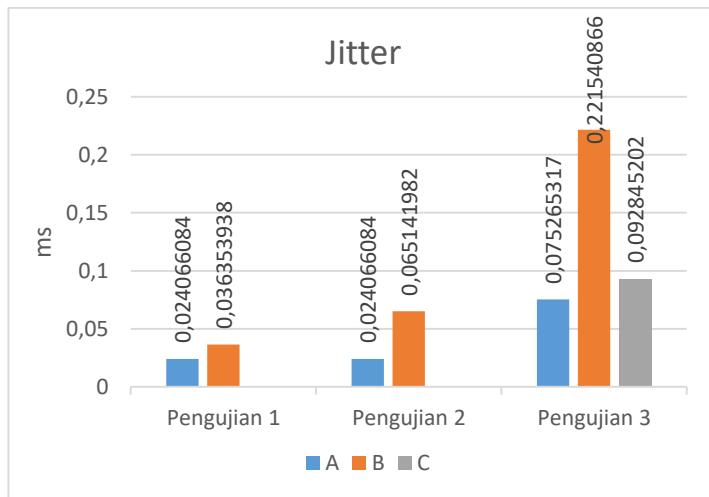
Pada pengujian 2 dapat dilihat bahwa nilai delay lebih besar saat kondisi terdapat 2 *devices* yang terhubung, ini berarti waktu tunda saat 2 *devices* terhubung lebih besar. Hal ini dikarenakan kemungkinan pengiriman paket data lebih banyak, sehingga waktu tunda dalam pengiriman juga menjadi lebih besar.

Pada pengujian 3 dapat dilihat bahwa nilai delay terbesar terjadi saat pengiriman data pada pukul 21.00. Bisa disimpulkan bahwa walaupun kecepatan pengiriman data saat pukul 21.00 adalah yang

tercepat, namun juga memiliki waktu tunda yang besar dalam penghitungan rata-rata.

### C. Jitter

Dalam pengukuran kualitas jaringan pada ketiga pengujian dengan setiap kondisinya, didapatkan nilai *delay* dari masing-masing kondisi seperti yang ditampilkan dalam grafik pada Gambar 4.38 berikut.

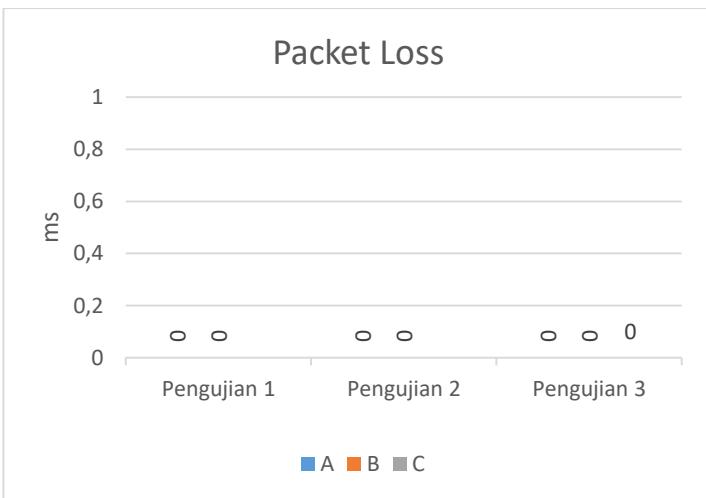


Gambar 4. 40 Grafik Perbandingan Pengukuran Jitter

Dari grafik yang ditampilkan, dapat dilihat bahwa hasil pengukuran jitter memiliki nilai yang berbeda-beda pada setiap kondisi walaupun selisih nilai pada masing-masing kondisi sangatlah kecil. Nilai jitter tertinggi terjadi saat pengiriman data pada pukul 13.00 sesuai pada pengujian 3. Pada pengujian 1 nilai jitter lebih tinggi pada saat jarak 10 meter. Sedangkan pada pengujian 2 nilai jitter lebih tinggi pada saat terdapat 2 *devices* yang terhubung ke server secara bersamaan dalam satu waktu pengiriman data.

### D. Packet Loss

Dalam pengukuran kualitas jaringan pada ketiga pengujian dengan setiap kondisinya, didapatkan nilai *delay* dari masing-masing kondisi seperti yang ditampilkan dalam grafik pada Gambar 4.39 berikut.



**Gambar 4. 41** Grafik Perbandingan Pengukuran Packet Loss

Dari grafik yang ditampilkan, dapat dilihat bahwa semua nilai *packet loss* yang didapatkan pada hasil pengukuran adalah 0%, yang berarti tidak ada satupun paket yang gagal terkirim pada setiap kondisi pengukuran.

#### 4.5.2 Klasifikasi Hasil Pengukuran Berdasarkan Standar Tiphon

Setelah masing-masing kondisi telah dianalisa dan dibahas hasilnya, kemudian pada bagian ini akan dijelaskan keseluruhan hasil pengukuran QoS pada sistem OASYS yang pada akhirnya hasilnya akan diklasifikasikan berdasarkan Standar TIPHON pada setiap parameter QoS. Hasil data yang diklasifikasikan berdasarkan standar, ditampilkan dalam Tabel 4.8 sebagai berikut.

**Tabel 4. 10** Klasifikasi Hasil Pengukuran Berdasarkan Standar Tiphon

No	Kode	Hasil Pengukuran		Standar Tiphon
		Parameter	Nilai	
1.	1A	Throughput (bps)	3029	4 (Sangat Baik)
		Delay (ms)	49,3807	4 (Sangat Baik)
		Jitter (ms)	0,02466	3 (Baik)
		Packet Loss (%)	0	4 (Sangat Baik)
2.	1B	Throughput (bps)	2952	4 (Sangat Baik)
		Delay (ms)	51,1832	4 (Sangat Baik)
		Jitter (ms)	0,03635	3 (Baik)
		Packet Loss (%)	0	4 (Sangat Baik)
3.	2A	Throughput (bps)	3029,1	4 (Sangat Baik)
		Delay (ms)	49,3807	4 (Sangat Baik)
		Jitter (ms)	0,024066	3 (Baik)
		Packet Loss (%)	0	4 (Sangat Baik)
4.	2B	Throughput (bps)	3028	4 (Sangat Baik)
		Delay (ms)	52,6927	4 (Sangat Baik)
		Jitter (ms)	0,065142	3 (Baik)
		Packet Loss (%)	0	4 (Sangat Baik)

No	Kode	Hasil Pengukuran		Standar Tiphon	
		Parameter	Nilai		
5.	3A	Throughput (bps)	3133,3	4 (Sangat Baik)	3,75 (Memuaskan)
		Delay (ms)	58,3185	4 (Sangat Baik)	
		Jitter (ms)	0,0608	3 (Baik)	
		Packet Loss (%)	0	4 (Sangat Baik)	
6.	3B	Throughput (bps)	3181,3	4 (Sangat Baik)	3,75 (Memuaskan)
		Delay (ms)	57,9809	4 (Sangat Baik)	
		Jitter (ms)	0,2809	3 (Baik)	
		Packet Loss (%)	0	4 (Sangat Baik)	
7.	3C	Throughput (bps)	3209	4 (Sangat Baik)	3,75 (Memuaskan)
		Delay (ms)	61,5915	4 (Sangat Baik)	
		Jitter (ms)	0,0589	3 (Baik)	
		Packet Loss (%)	0	4 (Sangat Baik)	

Dari keseluruhan pengelompokan hasil pengukuran berdasarkan standar Tiphon, diperoleh hasil pada setiap pengujian yang termasuk dalam kategori memuaskan. Pada setiap skenario pengujian ditemukan hasil nilai jitter yang tidak dalam kategori sangat baik, hal ini terjadi karena banyaknya variasi waktu tunda atau delay pada setiap pengiriman data sehingga menyebabkan nilai jitter dapat lebih tinggi.

Dari keseluruhan pengukuran, kemudian dilakukan penghitungan rata-rata terhadap setiap kondisi pengujian untuk mengetahui nilai keseluruhan sistem berdasarkan Standar Tiphon. Dari hasil penghitungan rata-rata diperoleh bahwa keseluruhan sistem OASYS termasuk dalam

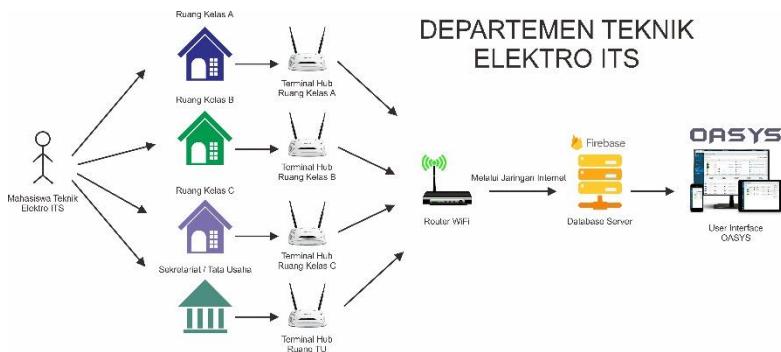
kategori Memuaskan menurut Standar Tiphon, yang dapat dilihat pada Tabel 4.9 berikut.

**Tabel 4. 11** Hasil Rata-rata Keseluruhan Pengukuran

No.	Kondisi	Standar Tiphon (Kategori)	
1.	1A	3, 75	Memuaskan
2.	1B	3, 75	Memuaskan
3.	2A	3, 75	Memuaskan
4.	2B	3, 75	Memuaskan
5.	3A	3, 75	Memuaskan
6.	3B	3, 75	Memuaskan
7.	3C	3, 75	Memuaskan
<b>Rata-rata</b>		<b>3, 75</b>	<b>Memuaskan</b>

## 4.6 Pemanfaatan Sistem OASYS

Dalam penerapannya, diharapkan sistem OASYS ini dapat dimanfaatkan pada Departemen Teknik Elektro ITS sesuai dengan pembahasan pada sub bab 3.3 tentang skenario pemanfaatan sistem. Alur dari pemanfaatan sistem yang dibuat dapat dilihat pada Gambar 4.42 berikut.



**Gambar 4. 42** Desain Pemanfaatan Sistem

Pada gambar tersebut dapat dilihat bahwa pada setiap kelas terdapat satu device yang masing-masing terhubung ke internet melalui hub yang berada di dalam kelas. Kemudian masing-masing hub di setiap kelas terhubung ke router pusat jaringan yang ada di Departemen Teknik Elektro ITS, sehingga dapat terhubungkan ke internet. Database server yang mendapat data dari device melalui jaringan internet, kemudian ditampilkan ke dalam user interface yang dapat diakses melalui browser.

Sesuai dengan hasil pengujian keandalan sistem, rata-rata waktu yang dibutuhkan masing-masing pengguna untuk dapat melakukan absensi adalah 2,343 detik. Apabila diasumsikan setiap pengguna melakukan absensi selama 3 detik, maka sebanyak 1200 pengguna dapat melakukan absensi dalam rentang waktu 60 menit, yang berarti dapat mencukupi keseluruhan mahasiswa pada Departemen Teknik Elektro ITS untuk melakukan absensi.

## **4.7 Alur Penggunaan Sistem OASYS (*User Manual*)**

Pada bagian ini akan menjelaskan penggunaan dari Sistem Absensi Berbasis IoT (OASYS) yang dibagi menjadi menjadi 3 bagian, yaitu registrasi, absensi, dan monitoring.

### **4.7.1 Proses Registrasi**

1. Proses registrasi dimulai dengan menemui admin untuk mengakses website OASYS bagian pendaftaran sidik jari. Pastikan nama mahasiswa sudah terdaftar di database.
2. Admin login ke website lalu pilih tab daftar mahasiswa, kemudian cari nama mahasiswa yang akan mendaftarkan sidik jari, dan pilih tombol "DAFTAR" disamping nama mahasiswa.
3. Jika layar pada *device* sensor sudah menampilkan logo *scanning* maka lakukan tap sidik jari pada *device* sensor, dan tunggu sampai layar menampilkan logo (v).
4. Lepaskan jari, dan ulangi sekali lagi untuk konfirmasi sidik jari yang didaftarkan.
5. Jika berhasil layar akan menampilkan "Fingerprint has been added". Jika gagal layar akan menampilkan "Error..." maka ulangi lakukan tap sekali lagi.

6. Sidik jari berhasil didaftarkan, proses registrasi telah selesai. Mahasiswa dapat melakukan absensi sesuai jadwal kuliah.

#### **4.7.2 Proses Absensi**

1. Proses absensi kuliah dapat dilakukan dalam *range* waktu 30 menit sebelum dan sesudah kelas dimulai sesuai jadwal.
2. Lakukan tap sidik jari pada *device* sensor di masing-masing kelas, dan tunggu sampai *device* merespon sidik jari.
3. Jika berhasil layar akan menampilkan logo (v) dan berbunyi “beep”. Jika gagal layar akan menampilkan logo (x) dan berbunyi “beep beep”.
4. Ulangi lakukan tap sidik jari lagi apabila *device* gagal membaca sidik jari, dan apabila berhasil maka proses absensi telah selesai dilakukan.

#### **4.7.3 Monitoring Kehadiran**

Monitoring kehadiran dilakukan melalui website OASYS. Langkah pertama yang dilakukan adalah dengan membuka website, lalu login dengan memasukkan Nama dan password.

##### **A. Mahasiswa**

1. Setelah berhasil login sebagai mahasiswa, tersedia fitur monitoring yang dibagi menjadi 2, yaitu pada tab MATA KULIAH dan tab LAPORAN.
2. Untuk melihat presensi kehadiran pada setiap mata kuliah beserta rincian waktu ketika melakukan absensi, buka tab MATA KULIAH lalu klik pada mata kuliah yang dipilih.
3. Untuk melihat persentase kehadiran secara keseluruhan minggu, pada setiap mata kuliah beserta dengan keterangan mengikuti ujian, buka tab LAPORAN dan klik pada mata kuliah yang dipilih.

## B. Dosen

1. Dosen dapat melakukan pengubahan jadwal mata kuliah sesuai yang diinginkan dengan cara membuka tab MATA KULIAH, lalu klik tombol UBAH JADWAL disamping mata kuliah yang dipilih.
2. Masukkan TANGGAL SEBELUMNYA, lalu ubah menjadi TANGGAL BARU serta masukkan WAKTU DIMULAI dan WAKTU SELESAI pada mata kuliah yang akan diubah diubah.
3. Untuk melihat presensi kehadiran mahasiswa pada setiap mata kuliah yang diampu dosen terkait, buka tab MATA KULIAH dan klik pada mata kuliah yang dipilih.

## 4.8 Analisa Pengembangan Sistem

Sistem OASYS memiliki kelebihan dan kekurangannya selayaknya suatu sistem pada umumnya, begitu juga dengan sistem absensi online dengan metode yang lain. Contoh perbandingan sistem absensi online dapat dilihat pada tabel berikut.

**Tabel 4. 12** Perbandingan Sistem Absensi Online

Para-meter	RFID	QR Code	Browser	Fingerprint
Keamanan (Validitas)	Menggunakan kartu, memungkinkan manipulasi data dengan menitipkan kartu	Memungkin kan manipulasi dengan mengirim QR Code	Memungkin kan manipulasi dengan menitipkan akun	Susah dimanipulasi karena sidik jari setiap orang berbeda
Jumlah Device	Membutuh-kan 1 <i>device</i> pada masing-masing kelas	Tidak memerlukan prototipe <i>device</i>	Tidak memerlukan prototipe <i>device</i>	Membutuh-kan minimal 1 <i>device</i> pada masing-masing kelas

<b>Par- meter</b>	<b>RFID</b>	<b>QR Code</b>	<b>Browser</b>	<b>Fingerprint</b>
Penggu- naan	Harus datang ke tempat dan menempelkan kartu	Harus datang ke tempat dan melakukan scanning QR Code	Dapat melakukan absensi dari jarak jauh	Harus datang ke tempat dan menempelkan sidik jari
Biaya	Biaya komponen cukup terjangkau, namun untuk pengembangan pada database memerlukan biaya tambahan	Biaya pengembangan web dan aplikasi cukup mahal, namun dalam penggunaan terpangkas biaya pembuatan aplikasi karena telah tersedia di beberapa device (HP)	Biaya pengembangan web cukup mahal serta perlu menyewa hosting untuk server	Biaya komponen cukup terjangkau, namun untuk pengembangan pada database dan server memerlukan biaya lebih

Sistem OASYS memiliki keunggulan dibandingkan sistem absensi *online* yang lainnya pada sisi keamanan dalam hal validitas data input, karena menggunakan sidik jari sebagai masukan yang secara teknis susah untuk dimanipulasi. Keuntungan menggunakan sistem OASYS ini adalah kemudahan dalam penggunaannya baik untuk absensi maupun monitoring melalui web, karena tampilan web yang dibuat telah dirancang untuk mudah digunakan serta memiliki fitur-fitur tambahan yang membuat web lebih dinamis. Selain itu penggunaan *cloud* sebagai database juga menjadi keunggulan karena masih sedikit ditemukan sistem absensi berbasis *cloud*.

Selain memiliki keunggulan-keunggulan tentunya sebuah sistem juga memiliki suatu kekurangan, pada sistem OASYS ini masih menggunakan *cloud* dengan fasilitas yang gratis dimana terdapat limit dalam penggunaannya, serta *interface web* yang digunakan masih belum menggunakan *hosting* sehingga untuk membuka web harus dijalankan dengan *framework* lokal.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Dari keseluruhan pembahasan tentang Rancang Bangun Sistem Absensi Perkuliahannya Berbasis *Internet of Things*, dapat disimpulkan bahwa:

1. Sistem Absensi Perkuliahannya Berbasis *Internet of Things* dapat berjalan sesuai dengan rancangan yang telah dibuat, dengan menggunakan Mikrokontroler NodeMCU ESP8266 sebagai pusat dari seluruh kinerja *device* absensi, serta menggunakan sensor sidik jari FPM10A sebagai input data absensi.
2. Website telah dibuat sesuai pada rancangan dengan menggunakan bahasa JavaScript, dan berbagai fitur-fitur didalamnya dapat bekerja dengan baik, termasuk fitur absensi jenis kelas paralel dan kelas serial, serta fitur ubah jadwal mata kuliah untuk dosen.
3. Pada pengujian keandalan sistem, data yang diterima pada database telah sesuai dengan data yang terbaca oleh sensor sidik jari, serta didapatkan rata-rata waktu yang dibutuhkan setiap pengguna untuk melakukan absensi adalah 2,343 detik.
4. Pada pengukuran QoS untuk parameter Throughput, diperoleh kesimpulan bahwa hasil pengukuran throughput dengan rata-rata 3080,243 bps termasuk dalam kategori sangat baik berdasarkan standar Tiphon dengan indeks 4. Nilai throughput akan berkurang jika lokasi atau jarak ruang kelas dengan router WiFi semakin jauh, serta jumlah ruang kelas yang semakin banyak.
5. Pada pengukuran QoS untuk parameter Delay dan Jitter, diperoleh kesimpulan bahwa nilai Delay dan Jitter dengan rata-rata masing-masing adalah 54,361 ms dan 0,0769 ms termasuk dalam kategori sangat baik dan baik menurut standar Tiphon. Nilai delay dan jitter akan semakin tinggi jika lokasi atau jarak ruang kelas dengan router WiFi semakin jauh, serta jumlah ruang kelas semakin banyak.
6. Pada pengukuran QoS untuk parameter Packet Loss, diperoleh nilai 0% yang berarti dapat diambil kesimpulan bahwa dalam berbagai kondisi pengujian, tidak ditemukan adanya paket data yang gagal terkirim selama dilakukannya pengukuran.

7. Berdasarkan keseluruhan hasil pengukuran nilai QoS, diperoleh bahwa sistem OASYS dapat bekerja dengan baik, dan mendapatkan predikat MEMUASKAN dengan indeks 3,75 sesuai pada standar Tiphon.

## **5.2 Saran**

Adapun beberapa saran yang bisa dikembangkan dari Rancang Bangun Sistem Absensi Perkuliahan Berbasis *Internet of Things* adalah sebagai berikut:

1. Menggunakan komponen dengan kualitas yang lebih bagus agar pembacaan sensor lebih cepat dan tepat, serta proses absensi dapat lebih efisien.
2. Menggunakan hosting untuk dapat mengakses website secara mudah dan dari mana saja.

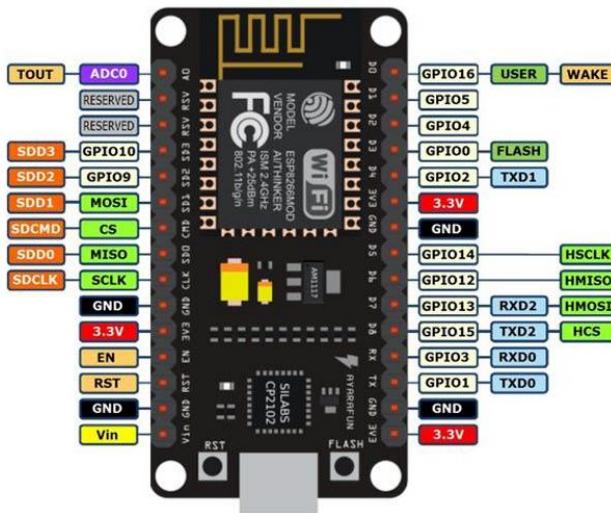
## DAFTAR PUSTAKA

- [1] L. D. Xu, W. He and S. Li, "Internet of Things in Industries: A Survey," *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 2233-2243, 2014.
- [2] Sano, E. Maeda, T. dkk. "Fingerprint Authentication Using Optical Characteristics in a Finger". Korea: SICE-ICASE International Joint Conference, pp. 1774-1777, Oktober, 2006.
- [3] R. Vijaysanthi, N. Radha, M. J. Shree and V. Sindhujaa, "Fingerprint Authentication using Raspberry Pi based on IoT," Hosur: Er. Perumal Manimekalai College of Engineering.
- [4] Hals, Electronic Tech, "Biometric Attendance System V1.0", <URL:<https://github.com/InfinityWorldHI/Biometric-attendance-system-V1.0>>, Terakhir Diperbaharui: Maret, 2020.
- [5] Grokhovskiy, I., Phillhower, E. F., Stanley, J., dkk, "HTTPS Request", <URL:<https://github.com/esp8266/Arduino/blob/master/libraries/ESP8266WiFi/examples/HTTPSSRequest/HTTPSSRequest.ino>>, Terakhir Diperbaharui: Maret, 2020.
- [6] Wisesa, L. "IoT Exprss Cloud Firestore Heroku", <URL:[https://github.com/LintangWisesa/IoT\\_Express\\_CloudFirestore\\_Heroku/blob/master/index.js](https://github.com/LintangWisesa/IoT_Express_CloudFirestore_Heroku/blob/master/index.js)>, September, 2018.
- [7] Firebase, "Angular Fire" <URL:<https://firebaseopensource.com/projects/firebase/angularfire/>>, Terakhir Diperbaharui: Februari, 2020
- [8] Firebase, "Cloud Firestore" <URL:<https://firebase.google.com/docs/firestore>>, Terakhir Diperbaharui: April, 2020

- [9] MDN, “HTML Basics” <URL:[https://developer.mozilla.org/en-US/docs/Learn/Getting started with the web/HTML basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics)>, Mei, 2019.
- [10] Sophia, A. C., “*Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)*”. Perancis: ETSI Technology. 1999.
- [11] Wulandari, R. “Analisis QoS (*Quality of Service*) Pada Jaringan Internet (Studi Kasus: Upt Loka Uji Teknik Penambangan Jampang Kulon – Lipi)” Jurnal Teknik Informatika dan Sistem Informasi, vol. 2, pp 162-172, Agustus, 2016.

## LAMPIRAN A DATASHEET

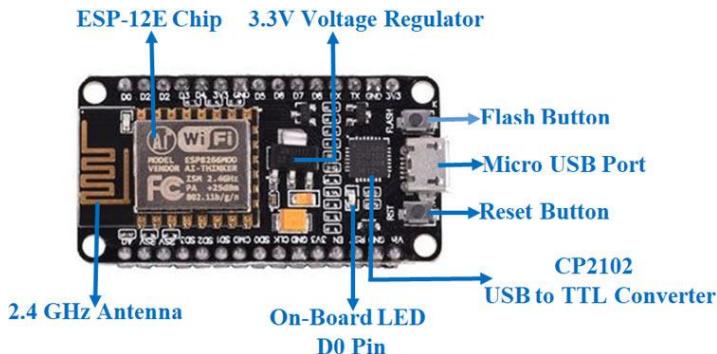
### 1. NodeMCU ESP8266



NodeMCU is an open-source Lua based firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

The **NodeMCU ESP8266 development board** comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.

NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.



### NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

### **NodeMCU Development Board Pinout Configuration**

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	<b>Micro-USB:</b> NodeMCU can be powered through the USB port <b>3.3V:</b> Regulated 3.3V can be supplied to this pin to power the board <b>GND:</b> Ground pins <b>Vin:</b> External Power Supply
Control Pins	EN, RST	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

## **2. Modul Sensor Sidik Jari FPM10A**

### **A. SPECIFICATION**

Voltage supply: DC 3.6 to 6.0V

Current supply: <120mA

Backlight color: green

Interface: UART

Bad rate: 9600

Safety level: five (from low to high: 1,2,3,4,5)

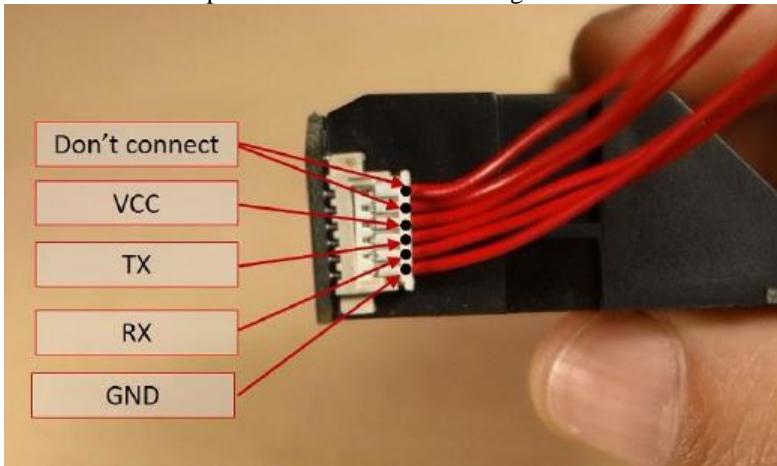
False Accept Rate (FAR): <0.001% (security level 3)

False Reject Rate (FRR): <1.0% (security level 3)

Able to store 127 different fingerprints

## B. SENSOR PINOUT

The sensor has six pins that are labeled in the figure below.



The fingerprint sensor module used in this project came with really thin wires, so soldering breadboard-friendly wires was needed.

We recommend using different colors according to the pin function. In our case:

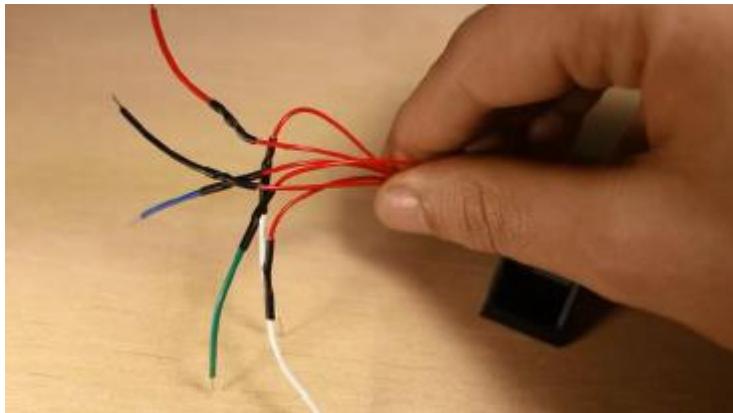
DNC – white wires

VCC – red wire

TX – blue wire

RX – green wire

GND – black wire



The following table shows how to wire the sensor to the Arduino.

#### Fingerprint Sensor

VCC

#### Arduino

5V (it also works with 3.3V)

TX

RX (digital pin 2, software serial)

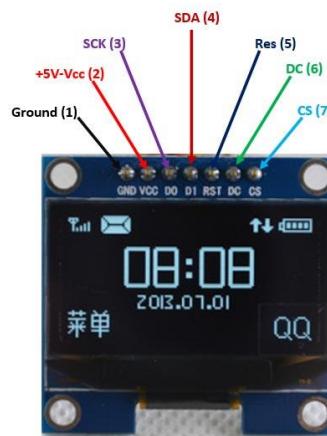
RX

TX (digital pin 3, software serial)

GND

GND

### 3. LCD Display OLED SSD1306 0.96 inch



**Pin Configuration:**

Pin No:	Pin Name:	Description
1	Ground (Gnd)	Connected to the ground of the circuit
2	Supply (Vdd,Vcc,5V)	Can be powered by either 3.3V or 5V
3	SCK (D0,SCL,CLK)	The display supports both IIC and SPI, for which clock is supplied through this pin
4	SDA (D1,MOSI)	This is the data pin of the both, it can either be used for IIC or for SPI
5	RES(RST,RESET)	When held to ground momentarily this pin resets the module
6	DC (A0)	This is command pin, can either be used for SPI or for IIC
7	Chip Select (CS)	Normally held low, used only when more than one SPI device is connected to MCU

**A. GENERAL DESCRIPTION**

SSD1306 is a single-chip CMOS OLED/PLED driver with controller for organic / polymer light emitting diode dot-matrix graphic display system. It consists of 128 segments and 64common. This IC is designed for Common Cathode type OLED panel. The SSD1306 embeds with contrast control, display RAM and oscillator, which reduces the number of external components and power consumption. It has 256-step brightness control. Data/Commands are sent from general MCU through the hardware selectable 6800/8000 series compatible Parallel Interface, I2C interface or Serial Peripheral Interface. It is suitable for many compact portable applications, such as mobile phone sub-display, MP3 player and calculator, etc.

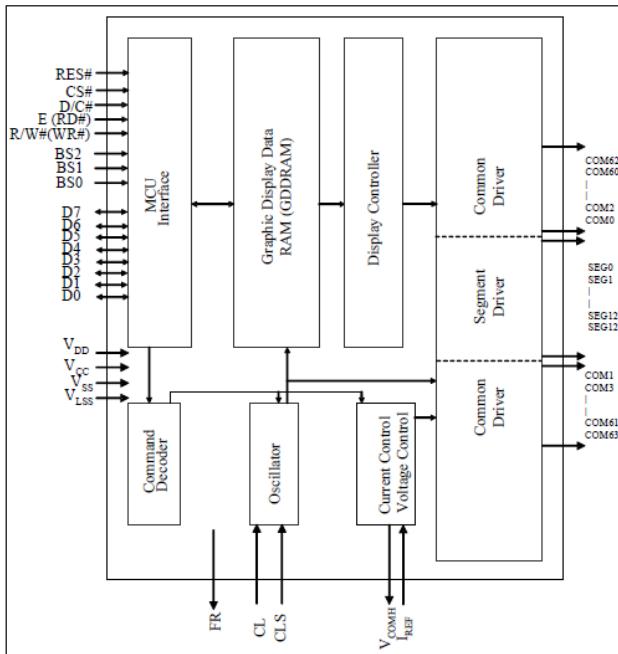
**B. FEATURES**

- Resolution: 128 x 64 dot matrix panel
- Power supply
  - VDD = 1.65V to 3.3V for IC logic
  - VCC = 7V to 15V for Panel driving
- For matrix display
  - OLED driving output voltage, 15V maximum
  - Segment maximum source current: 100uA
  - Common maximum sink current: 15mA
  - 256 step contrast brightness current control

- Embedded 128 x 64 bit SRAM display buffer
- Pin selectable MCU Interfaces:
  - 8-bit 6800/8080-series parallel interface
  - 3 / 4 wire Serial Peripheral Interface
  - I2C Interface
- Screen saving continuous scrolling function in both horizontal and vertical direction
- RAM write synchronization signal
- Programmable Frame Rate and Multiplexing Ratio
- Row Re-mapping and Column Re-mapping
- On-Chip Oscillator
- Chip layout for COG & COF
- Wide range of operating temperature: -40°C to 85°C

## C. BLOCK DIAGRAM

Figure 4-1 SSD1306 Block Diagram



## 4. Buzzer 5v

### Buzzer

pro-SIGNAL



#### Features

- Black in colour
- With internal drive circuit
- Sealed structure
- Wave solderable and washable
- Housing material: Noryl

RoHS  
Compliant

#### Applications

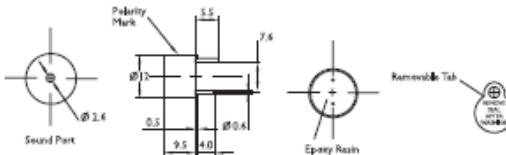
- Computer and peripherals
- Communications equipment
- Portable equipment
- Automobile electronics
- POS system
- Electronic cash register

#### Specifications:

Rated Voltage	: 6V DC
Operating Voltage	: 4 to 8V DC
Rated Current*	: <30mA
Sound Output at 10cm*	: 85dB
Resonant Frequency	: 2300 ±300Hz
Tone	: Continuous
Operating Temperature	: -25°C to +80°C
Storage Temperature	: -30°C to +85°C
Weight	: 3g

\*Value applying at rated voltage (DC)

#### Diagram



Dimensions : Millimetres

Tolerance : ±0.5mm

#### Part Number Table

Description	Part Number
Buzzer, Electromech, 6V DC	AB1-009-RC

Important Notice : This data sheet and its contents (the "Information") belong to the members of the Premier Farnell group of companies ("the Group") or are licensed to it. No licence is granted for the use of it other than for information purposes in connection with the products to which it relates. No licence of any intellectual property rights is granted. The Information is subject to change without notice and replaces all data sheets previously supplied. The Information supplied is believed to be accurate but the Group assumes no responsibility for its accuracy or completeness, any errors or omissions may be present. The Group reserves the right to correct any errors or omissions without prior notice. The Group is not liable for any typographical or other errors which may appear in this document. The Group accepts no liability for any inaccuracy in this document. The Group shall not be liable for any loss or damage arising from the use of the Information or from any reliance on the Information or use of it (including liability resulting from negligence or where the Group was aware of the possibility of such loss or damage arising). Liability for loss or damage resulting from any reliance on the Information or use of it (including liability resulting from negligence or where the Group was aware of the possibility of such loss or damage arising) is excluded. This will not operate to limit or restrict the Groups liability for death or personal injury resulting from its negligence.

[www.element14.com](http://www.element14.com)  
[www.larneil.com](http://www.larneil.com)  
[www.newark.com](http://www.newark.com)  
[www.cpc.co.uk](http://www.cpc.co.uk)

pro-SIGNAL

## **LAMPIRAN B**

### **LISTING PROGRAM**

#### **1. Pemrograman Arduino IDE**

```
//*****libraries*****
#include <SPI.h>
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Fingerprint.h>

//*****
//Fingerprint scanner Pins
#define Finger_Rx D5 //D5
#define Finger_Tx D6 //D6

// Declaration for SSD1306 display connected using software I2C
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET 0 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);

//*****
SoftwareSerial mySerial(Finger_Rx, Finger_Tx);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

//*****
/* Set these to your desired credentials. */
const char* ssid = "aspire"; //ENTER YOUR WIFI SETTINGS
const char* password = "12345678";
```

```

//*****
String postData ; // post array that will be send to the website
String host = "http://us-central1-oasys-6fdb4.cloudfunctions.net/app";
//computer IP or the server domain
int FingerID = 0; // The Fingerprint ID from the scanner
uint8_t id;

//*****Biometric Icons*****
#define Wifi_start_width 54
#define Wifi_start_height 49
const uint8_t PROGMEM Wifi_start_bits[] = {
  0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x1f,0xff,0x00,0x00,0x00
,0x00,0x03,0xff,0xff,0x80,0x00,0x00
,0x00,0x1f,0xf0,0x1f,0xf0,0x00,0x00,0x00
,0x00,0x7e,0x00,0x00,0xfc,0x00,0x00
,0x01,0xfc,0x00,0x00,0x1f,0x00,0x00
,0x03,0xc0,0x00,0x00,0x07,0xc0,0x00
,0x0f,0x00,0x00,0x00,0x01,0xe0,0x00
,0x1c,0x00,0x00,0x00,0x00,0x70,0x00
,0x38,0x00,0x07,0xc0,0x00,0x38,0x00
,0x70,0x00,0xff,0xfe,0x00,0x1e,0x00
,0xe0,0x03,0xfc,0x7f,0xc0,0x0e,0x00
,0x00,0x1f,0x80,0x03,0xf0,0x00,0x00,0x00
,0x00,0x3c,0x00,0x00,0x78,0x00,0x00
,0x00,0xfc,0x00,0x00,0x1c,0x00,0x00
,0x01,0xe0,0x00,0x00,0x0c,0x00,0x00
,0x03,0x80,0x00,0x00,0x00,0x00,0x00
,0x03,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x3f,0xf8,0x07,0x1e,0x00
,0x00,0x00,0xff,0xfe,0x1f,0xbff,0x80
,0x00,0x03,0xe0,0x04,0x7f,0xff,0xc0
,0x00,0x07,0x80,0x00,0xff,0xff,0xe0
,0x00,0x0e,0x00,0x00,0xff,0xff,0xe0

```

```
,0x00,0x0c,0x00,0x00,0x7f,0xff,0xc0  
,0x00,0x00,0x00,0x00,0xfe,0x07,0xe0  
,0x00,0x00,0x00,0x03,0xf8,0x03,0xf8  
,0x00,0x00,0x07,0xe7,0xf9,0xf1,0xfc  
,0x00,0x00,0x1f,0xe7,0xf1,0xf9,0xfc  
,0x00,0x00,0x1f,0xe7,0xf3,0xf9,0xfc  
,0x00,0x00,0x3f,0xe7,0xf3,0xf9,0xfc  
,0x00,0x00,0x3f,0xe7,0xf1,0xf1,0xfc  
,0x00,0x00,0x3f,0xe3,0xf8,0xe3,0xfc  
,0x00,0x00,0x3f,0xf3,0xfc,0x07,0xf8  
,0x00,0x00,0x1f,0xf0,0x7f,0x0f,0xc0  
,0x00,0x00,0x0f,0xe0,0x7f,0xff,0xe0  
,0x00,0x00,0x07,0xc0,0xff,0xff,0xe0  
,0x00,0x00,0x00,0x00,0x7f,0xff,0xe0  
,0x00,0x00,0x00,0x00,0x3f,0xff,0x80  
,0x00,0x00,0x00,0x00,0x00,0x1f,0xbff,0x00  
,0x00,0x00,0x00,0x00,0x00,0x03,0x18,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
};  
#define Wifi_connected_width 63  
#define Wifi_connected_height 49  
const uint8_t PROGMEM Wifi_connected_bits[] = {  
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x03,0xff,0xff,0x80,0x00,0x00  
,0x00,0x00,0x3f,0xff,0xff,0xf8,0x00,0x00  
,0x00,0x01,0xff,0xff,0xff,0xff,0x00,0x00  
,0x00,0x0f,0xff,0xff,0xff,0xff,0xe0,0x00  
,0x00,0x3f,0xff,0xc0,0x07,0xff,0xf8,0x00  
,0x00,0xff,0xf8,0x00,0x00,0x3f,0xfe,0x00
```

```
,0x03,0xff,0x80,0x00,0x00,0x03,0xff,0x80  
,0x07,0xfe,0x00,0x00,0x00,0x00,0xff,0xc0  
,0x1f,0xff,0x00,0x00,0x00,0x00,0x3f,0xf0  
,0x3f,0xe0,0x01,0xff,0xff,0x00,0x0f,0xf8  
,0x7f,0x80,0x0f,0xff,0xff,0xe0,0x03,0xfc  
,0xff,0x00,0x7f,0xff,0xff,0xfc,0x01,0xfe  
,0xfc,0x01,0xff,0xff,0xff,0x00,0x7e  
,0x78,0x07,0xff,0xc0,0x07,0xff,0xc0,0x3c  
,0x00,0x0f,0xfc,0x00,0x00,0x7f,0xe0,0x00  
,0x00,0x1f,0xf0,0x00,0x00,0x1f,0xf0,0x00  
,0x00,0x3f,0xc0,0x00,0x00,0x07,0xf8,0x00  
,0x00,0x7f,0x00,0x01,0x00,0x01,0xfc,0x00  
,0x00,0x7e,0x00,0x7f,0xfc,0x00,0xfc,0x00  
,0x00,0x3c,0x03,0xff,0xff,0x80,0x78,0x00  
,0x00,0x00,0x07,0xff,0xff,0xc0,0x00,0x00  
,0x00,0x00,0x1f,0xff,0xff,0xf0,0x00,0x00  
,0x00,0x00,0x3f,0xf0,0x1f,0xf8,0x00,0x00  
,0x00,0x00,0x3f,0x80,0x03,0xf8,0x00,0x00  
,0x00,0x00,0x3f,0x00,0x01,0xf8,0x00,0x00  
,0x00,0x00,0x1c,0x00,0x00,0x70,0x00,0x00  
,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x0f,0xe0,0x00,0x00,0x00  
,0x00,0x00,0x00,0x1f,0xf0,0x00,0x00,0x00  
,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00  
,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00  
,0x00,0x00,0x00,0x3f,0xf8,0x00,0x00,0x00  
,0x00,0x00,0x00,0x1f,0xf0,0x00,0x00,0x00  
,0x00,0x00,0x00,0x0f,0xe0,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
};  
#define FinPr_start_width 64  
#define FinPr_start_height 64
```

```
const uint8_t PROGMEM FinPr_start_bits[] = {  
    0x00,0x00,0x00,0x1f,0xe0,0x00,0x00,0x00  
,0x00,0x00,0x01,0xff,0xfe,0x00,0x00,0x00  
,0x00,0x00,0x03,0xff,0xff,0x80,0x00,0x00  
,0x00,0x00,0x0f,0xc0,0x0f,0xe0,0x00,0x00  
,0x00,0x00,0x1f,0x00,0x01,0xf8,0x00,0x00  
,0x00,0x00,0x3c,0x00,0x00,0x7c,0x00,0x00  
,0x00,0x00,0x78,0x00,0x00,0x3e,0x00,0x00  
,0x00,0x00,0xf0,0x3f,0xf8,0x0f,0x00,0x00  
,0x00,0x01,0xe0,0xff,0xfe,0x07,0x80,0x00  
,0x00,0x03,0xc3,0xff,0xff,0x03,0x80,0x00  
,0x00,0x03,0x87,0xc0,0x07,0xc3,0xc0,0x00  
,0x00,0x07,0x0f,0x00,0x03,0xe1,0xc0,0x00  
,0x00,0x0f,0x0e,0x00,0x00,0xe0,0xe0,0x00  
,0x00,0x0e,0x1c,0x00,0x00,0xf0,0xe0,0x00  
,0x00,0x0c,0x3c,0x1f,0xe0,0x70,0xe0,0x00  
,0x00,0x00,0x38,0x3f,0xf0,0x38,0x70,0x00  
,0x00,0x00,0x78,0x78,0xf8,0x38,0x70,0x00  
,0x00,0x00,0x70,0x70,0x3c,0x18,0x70,0x00  
,0x00,0x00,0xe0,0xe0,0x1e,0x1c,0x70,0x00  
,0x00,0x03,0xe1,0xe0,0x0e,0x1c,0x70,0x00  
,0x00,0x0f,0xc1,0xc3,0x0e,0x1c,0x70,0x00  
,0x00,0x3f,0x03,0xc3,0x8e,0x1c,0x70,0x00  
,0x00,0x3e,0x03,0x87,0x0e,0x1c,0x70,0x00  
,0x00,0x30,0x07,0x07,0x0e,0x18,0xe0,0x00  
,0x00,0x00,0x0e,0x0e,0x0e,0x38,0xe0,0x00  
,0x00,0x00,0x3e,0x1e,0x1e,0x38,0xe0,0x00  
,0x00,0x00,0xf8,0x1c,0x1c,0x38,0xe0,0x00  
,0x00,0x03,0xf0,0x38,0x3c,0x38,0xe0,0x00  
,0x00,0x3f,0xc0,0xf8,0x78,0x38,0xe0,0x00  
,0x00,0x7f,0x01,0xf0,0x70,0x38,0xf0,0x00  
,0x00,0x78,0x03,0xe0,0xe0,0x38,0x70,0x00  
,0x00,0x00,0x0f,0x81,0xe0,0x38,0x7c,0x00  
,0x00,0x00,0x3f,0x03,0xc0,0x38,0x3e,0x00  
,0x00,0x00,0xfc,0x0f,0x80,0x38,0x1e,0x00  
,0x00,0x07,0xf0,0x1f,0x1c,0x1c,0x04,0x00  
,0x00,0x3f,0xc0,0x3e,0x3f,0x1e,0x00,0x00  
,0x00,0x7f,0x00,0xf8,0x7f,0x0f,0x00,0x00  
,0x00,0x38,0x01,0xf0,0xf7,0x07,0xc0,0x00
```

```

,0x00,0x00,0x07,0xe1,0xe3,0x83,0xf8,0x00
,0x00,0x00,0x3f,0x87,0xc3,0xc0,0xfc,0x00
,0x00,0x01,0xfe,0x0f,0x81,0xe0,0x3c,0x00
,0x00,0x0f,0xf8,0x1f,0x00,0xf0,0x00,0x00
,0x00,0x1f,0xc0,0x7c,0x00,0x7c,0x00,0x00
,0x00,0x1e,0x01,0xf8,0x00,0x3f,0x00,0x00
,0x00,0x00,0x07,0xe0,0x78,0x0f,0xc0,0x00
,0x00,0x00,0x3f,0x81,0xfe,0x07,0xf0,0x00
,0x00,0x01,0xfe,0x07,0xff,0x01,0xf0,0x00
,0x00,0x07,0xf8,0x0f,0x87,0x80,0x30,0x00
,0x00,0x07,0xc0,0x3f,0x03,0xe0,0x00,0x00
,0x00,0x06,0x00,0xfc,0x01,0xf8,0x00,0x00
,0x00,0x00,0x03,0xf0,0x00,0x7e,0x00,0x00
,0x00,0x00,0x0f,0xc0,0x00,0x3f,0x80,0x00
,0x00,0x00,0x7f,0x00,0xf8,0x0f,0x80,0x00
,0x00,0x00,0xfc,0x03,0xfe,0x01,0x80,0x00
,0x00,0x00,0xf0,0x1f,0xff,0x80,0x00,0x00
,0x00,0x00,0x00,0x7f,0x07,0xe0,0x00,0x00
,0x00,0x00,0xfc,0x03,0xf8,0x00,0x00
,0x00,0x00,0x03,0xf0,0x00,0x78,0x00,0x00
,0x00,0x00,0x0f,0xc0,0x00,0x18,0x00,0x00
,0x00,0x00,0x0f,0x01,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0x07,0xfe,0x00,0x00,0x00
,0x00,0x00,0x00,0x1f,0xfe,0x00,0x00,0x00
,0x00,0x00,0x00,0x1e,0x0e,0x00,0x00,0x00
,0x00,0x00,0x00,0x18,0x00,0x00,0x00,0x00
};

//-----
#define FinPr_valid_width 64
#define FinPr_valid_height 64
const uint8_t PROGMEM FinPr_valid_bits[] = {
  0x00,0x00,0x03,0xfe,0x00,0x00,0x00,0x00
,0x00,0x00,0x1f,0xff,0xe0,0x00,0x00,0x00
,0x00,0x00,0x7f,0xff,0xf8,0x00,0x00,0x00
,0x00,0x00,0xfc,0x00,0xfe,0x00,0x00,0x00
,0x00,0x03,0xe0,0x00,0x1f,0x00,0x00,0x00
,0x00,0x07,0xc0,0x00,0x07,0x80,0x00,0x00
,0x00,0x0f,0x80,0x00,0x03,0xe0,0x00,0x00
,0x00,0x0e,0x03,0xff,0x01,0xe0,0x00,0x00

```

,0x00,0x1c,0x1f,0xff,0xe0,0xf0,0x00,0x00  
,0x00,0x3c,0x3f,0xff,0xf0,0x78,0x00,0x00  
,0x00,0x78,0x7c,0x00,0xf8,0x3c,0x00,0x00  
,0x00,0x70,0xf0,0x00,0x3c,0x1c,0x00,0x00  
,0x00,0xe1,0xe0,0x00,0x1e,0x1c,0x00,0x00  
,0x00,0xe1,0xc0,0x00,0x0f,0x0e,0x00,0x00  
,0x00,0xc3,0x81,0xfc,0x07,0x0e,0x00,0x00  
,0x00,0x03,0x83,0xff,0x07,0x8e,0x00,0x00  
,0x00,0x07,0x07,0x8f,0x83,0x87,0x00,0x00  
,0x00,0x0f,0x0f,0x03,0xc3,0x87,0x00,0x00  
,0x00,0x1e,0x0e,0x01,0xc3,0x87,0x00,0x00  
,0x00,0x3c,0x1c,0x00,0xe1,0x87,0x00,0x00  
,0x00,0xf8,0x1c,0x30,0xe1,0x87,0x00,0x00  
,0x07,0xf0,0x38,0x70,0xe1,0x86,0x00,0x00  
,0x07,0xc0,0x78,0x70,0xe3,0x8e,0x00,0x00  
,0x02,0x00,0xf0,0xf0,0xe3,0x8e,0x00,0x00  
,0x00,0x01,0xe0,0xe0,0xe3,0x8e,0x00,0x00  
,0x00,0x03,0xc1,0xe1,0xc3,0x8e,0x00,0x00  
,0x00,0x0f,0x83,0xc3,0xc3,0x8e,0x00,0x00  
,0x00,0x7f,0x07,0x83,0x83,0x0e,0x00,0x00  
,0x07,0xfc,0x0f,0x07,0x83,0x0e,0x00,0x00  
,0x07,0xf0,0x1e,0x0f,0x03,0x0e,0x00,0x00  
,0x07,0x80,0x7c,0x1e,0x03,0x07,0x00,0x00  
,0x00,0x00,0xf8,0x3c,0x03,0x87,0x80,0x00  
,0x00,0x03,0xf0,0x78,0x03,0x83,0xc0,0x00  
,0x00,0x1f,0xc0,0xf0,0x02,0x00,0x00,0x00  
,0x00,0xff,0x01,0xe1,0xc0,0x0c,0x00,0x00  
,0x07,0xfc,0x03,0xc3,0xe1,0xff,0xc0,0x00  
,0x07,0xe0,0x0f,0x87,0xc7,0xff,0xf0,0x00  
,0x07,0x00,0x3f,0x0f,0x0f,0xff,0xfc,0x00  
,0x00,0x00,0x7c,0x3e,0x3f,0xff,0xfe,0x00  
,0x00,0x03,0xf8,0x7c,0x3f,0xff,0xff,0x00  
,0x00,0x1f,0xe0,0xf0,0x7f,0xff,0xff,0x80  
,0x00,0xff,0x83,0xe0,0xff,0xff,0x80  
,0x01,0xfc,0x07,0xc1,0xff,0xff,0xe3,0xc0  
,0x01,0xe0,0x1f,0x01,0xff,0xff,0xc3,0xc0  
,0x00,0x00,0xfe,0x01,0xff,0xff,0x87,0xe0  
,0x00,0x03,0xf8,0x13,0xff,0xff,0x0f,0xe0  
,0x00,0x1f,0xe0,0x73,0xff,0xfe,0x1f,0xe0

```

,0x00,0x7f,0x81,0xf3,0xff,0xfc,0x1f,0xe0
,0x00,0xfc,0x03,0xe3,0xef,0xf8,0x3f,0xe0
,0x00,0x60,0x0f,0xc3,0xc7,0xf0,0x7f,0xe0
,0x00,0x00,0x3f,0x03,0xc3,0xe0,0xff,0xe0
,0x00,0x00,0xfc,0x03,0xc1,0xc1,0xff,0xe0
,0x00,0x07,0xf0,0x13,0xe0,0x83,0xff,0xe0
,0x00,0x0f,0xc0,0x7b,0xf8,0x07,0xff,0xe0
,0x00,0x0f,0x01,0xf9,0xfc,0x0f,0xff,0xc0
,0x00,0x00,0x07,0xf1,0xfe,0x1f,0xff,0xc0
,0x00,0x00,0x1f,0xc0,0xff,0x3f,0xff,0x80
,0x00,0x00,0x7e,0x00,0xff,0xff,0xff,0x80
,0x00,0x00,0xfc,0x00,0x7f,0xff,0xff,0x00
,0x00,0x00,0xf0,0x1f,0x3f,0xff,0xfe,0x00
,0x00,0x00,0x00,0x7f,0x1f,0xff,0xfc,0x00
,0x00,0x00,0x01,0xff,0x8f,0xff,0xf8,0x00
,0x00,0x00,0x03,0xe0,0xe3,0xff,0xe0,0x00
,0x00,0x00,0x01,0x80,0x00,0x7f,0x00,0x00
};

//-----
#define FinPr_invalid_width 64
#define FinPr_invalid_height 64
const uint8_t PROGMEM FinPr_invalid_bits[] = {
  0x00,0x00,0x03,0xfe,0x00,0x00,0x00,0x00
,0x00,0x00,0x1f,0xff,0xe0,0x00,0x00,0x00
,0x00,0x00,0x7f,0xff,0xf8,0x00,0x00,0x00
,0x00,0x00,0xfc,0x00,0xfe,0x00,0x00,0x00
,0x00,0x03,0xe0,0x00,0x1f,0x00,0x00,0x00
,0x00,0x07,0xc0,0x00,0x07,0x80,0x00,0x00
,0x00,0x0f,0x80,0x00,0x03,0xe0,0x00,0x00
,0x00,0x0e,0x03,0xff,0x01,0xe0,0x00,0x00
,0x00,0x1c,0x1f,0xff,0xe0,0xf0,0x00,0x00
,0x00,0x3c,0x3f,0xff,0xf0,0x78,0x00,0x00
,0x00,0x78,0x7c,0x00,0xf8,0x3c,0x00,0x00
,0x00,0x70,0xf0,0x00,0x3c,0x1c,0x00,0x00
,0x00,0xe1,0xe0,0x00,0x1e,0x1c,0x00,0x00
,0x00,0xe1,0xc0,0x00,0x0f,0x0e,0x00,0x00
,0x00,0xc3,0x81,0xfc,0x07,0x0e,0x00,0x00
,0x00,0x03,0x83,0xff,0x07,0x8e,0x00,0x00
,0x00,0x07,0x07,0x8f,0x83,0x87,0x00,0x00

```

,0x00,0x0f,0x0f,0x03,0xc3,0x87,0x00,0x00  
,0x00,0x1e,0x0e,0x01,0xc3,0x87,0x00,0x00  
,0x00,0x3c,0x1c,0x00,0xe1,0x87,0x00,0x00  
,0x00,0xf8,0x1c,0x30,0xe1,0x87,0x00,0x00  
,0x07,0xf0,0x38,0x70,0xe1,0x86,0x00,0x00  
,0x07,0xc0,0x78,0x70,0xe3,0x8e,0x00,0x00  
,0x02,0x00,0xf0,0xf0,0xe3,0x8e,0x00,0x00  
,0x00,0x01,0xe0,0xe0,0xe3,0x8e,0x00,0x00  
,0x00,0x03,0xc1,0xe1,0xc3,0x8e,0x00,0x00  
,0x00,0x0f,0x83,0xc3,0xc3,0x8e,0x00,0x00  
,0x00,0x7f,0x07,0x83,0x83,0x0e,0x00,0x00  
,0x07,0xfc,0x0f,0x07,0x83,0x0e,0x00,0x00  
,0x07,0xf0,0x1e,0x0f,0x03,0x0e,0x00,0x00  
,0x07,0x80,0x7c,0x1e,0x03,0x07,0x00,0x00  
,0x00,0x00,0xff,0x3c,0x03,0x87,0x80,0x00  
,0x00,0x03,0xf0,0x78,0x03,0x83,0xc0,0x00  
,0x00,0x1f,0xc0,0xf0,0x02,0x00,0x00,0x00  
,0x00,0xff,0x01,0xe1,0xc0,0x00,0x00,0x00  
,0x07,0xfc,0x03,0xc3,0xe1,0xff,0xc0,0x00  
,0x07,0xe0,0x0f,0x87,0xc7,0xff,0xf0,0x00  
,0x07,0x00,0x3f,0x0f,0x0f,0xff,0xf8,0x00  
,0x00,0x00,0x7c,0x3e,0x1f,0xff,0xfe,0x00  
,0x00,0x03,0xff,0x7c,0x3f,0xff,0xff,0x00  
,0x00,0x1f,0xe0,0xf0,0x7f,0xff,0xff,0x00  
,0x00,0xff,0x83,0xe0,0xfe,0xff,0xbff,0x80  
,0x01,0xfc,0x07,0xc0,0xfc,0x7f,0x1f,0xc0  
,0x01,0xe0,0x1f,0x01,0xf8,0x3e,0x0f,0xc0  
,0x00,0x00,0xfe,0x01,0xf8,0x1c,0x07,0xe0  
,0x00,0x03,0xff,0x13,0xf8,0x00,0x0f,0xe0  
,0x00,0x1f,0xe0,0x73,0xfc,0x00,0x1f,0xe0  
,0x00,0x7f,0x81,0xf3,0xfe,0x00,0x3f,0xe0  
,0x00,0xfc,0x03,0xe3,0xff,0x00,0x7f,0xe0  
,0x00,0x60,0x0f,0xc3,0xff,0x80,0xff,0xe0  
,0x00,0x00,0x3f,0x03,0xff,0x00,0x7f,0xe0  
,0x00,0x00,0xfc,0x03,0xfe,0x00,0x3f,0xe0  
,0x00,0x07,0xf0,0x13,0xfc,0x00,0x1f,0xe0  
,0x00,0x0f,0xc0,0x79,0xf8,0x08,0x0f,0xe0  
,0x00,0x0f,0x01,0xf9,0xf8,0x1c,0x0f,0xc0  
,0x00,0x00,0x07,0xf1,0xfc,0x3e,0x1f,0xc0

```

,0x00,0x00,0x1f,0xc0,0xfe,0x7f,0x3f,0x80
,0x00,0x00,0x7e,0x00,0xff,0xff,0xff,0x80
,0x00,0x00,0xfc,0x00,0x7f,0xff,0xff,0x00
,0x00,0x00,0xf0,0x1f,0x3f,0xff,0xfe,0x00
,0x00,0x00,0x00,0x7f,0x1f,0xff,0xfc,0x00
,0x00,0x00,0x01,0xff,0x8f,0xff,0xf8,0x00
,0x00,0x00,0x03,0xe3,0xe3,0xff,0xe0,0x00
,0x00,0x00,0x01,0x80,0x00,0x7f,0x00,0x00
};

//-----
#define FinPr_failed_width 64
#define FinPr_failed_height 64
const uint8_t PROGMEM FinPr_failed_bits[] = {
0x00,0x00,0x3f,0xe0,0x00,0x00,0x00,0x00
,0x00,0x01,0xff,0xfe,0x00,0x00,0x00,0x00
,0x00,0x0f,0xc0,0x1f,0x80,0x00,0x00,0x00
,0x00,0x1e,0x00,0x03,0xc0,0x00,0x00,0x00
,0x00,0x78,0x00,0x00,0xf0,0x00,0x00,0x00
,0x00,0xe0,0x00,0x00,0x38,0x00,0x00,0x00
,0x01,0xc0,0x00,0x00,0x1c,0x00,0x00,0x00
,0x03,0x80,0x00,0x00,0x0e,0x00,0x00,0x00
,0x07,0x00,0x7f,0xe0,0x07,0x00,0x00,0x00
,0x06,0x01,0xff,0xf8,0x03,0x00,0x00,0x00
,0x0c,0x03,0xc0,0x3c,0x03,0x80,0x00,0x00
,0x1c,0x0f,0x00,0x0e,0x01,0x80,0x00,0x00
,0x18,0x0c,0x00,0x03,0x00,0xc0,0x00,0x00
,0x18,0x18,0x00,0x01,0x80,0xc0,0x00,0x00
,0x30,0x38,0x00,0x01,0xc0,0xe0,0x00,0x00
,0x30,0x30,0x0f,0x00,0xc0,0x60,0x00,0x00
,0x30,0x30,0x3f,0xc0,0xe0,0x60,0x00,0x00
,0x70,0x60,0x78,0xe0,0x60,0x60,0x00,0x00
,0x60,0x60,0x60,0x60,0x60,0x70,0x00,0x00
,0x60,0x60,0x60,0x60,0x60,0x30,0x00,0x00
,0x60,0x60,0x60,0x60,0x30,0x30,0x00,0x00
,0x60,0x60,0x60,0x30,0x30,0x20,0x00,0x00
,0x60,0x60,0x60,0x30,0x30,0x01,0xe0,0x00
,0x60,0x60,0x60,0x30,0x30,0x0f,0xfc,0x00
,0x60,0x60,0x60,0x30,0x30,0x3f,0xff,0x00
,0x60,0x60,0x60,0x30,0x18,0x78,0x03,0x80

```

```
,0x60,0x60,0x60,0x30,0x1c,0x60,0x01,0x80
,0x60,0x60,0x30,0x38,0x0c,0xc0,0x00,0xc0
,0x00,0x60,0x30,0x18,0x00,0xc0,0x00,0xc0
,0x00,0x60,0x30,0x18,0x00,0xc0,0x00,0xc0
,0x00,0xe0,0x30,0x0c,0x01,0xc0,0x00,0xe0
,0x00,0xc0,0x18,0x0e,0x01,0xc0,0x00,0xe0
,0x60,0xc0,0x18,0x07,0x01,0xc0,0x00,0xe0
,0x01,0xc0,0x1c,0x03,0x81,0xc0,0x00,0xe0
,0x01,0x80,0x0c,0x01,0xc1,0xc0,0x00,0xe0
,0x03,0x80,0x0e,0x00,0xf1,0xc0,0x00,0xe0
,0x0f,0x00,0x06,0x00,0x01,0xc0,0x00,0xe0
,0x3e,0x01,0x03,0x00,0x01,0xc0,0x00,0xe0
,0x30,0x03,0x83,0x80,0x1f,0xff,0xff,0xfe
,0x00,0x03,0x81,0xc0,0x3f,0xff,0xff,0xff
,0x00,0x07,0xc0,0xe0,0x30,0x00,0x00,0x03
,0x00,0x0e,0xc0,0x78,0x30,0x00,0x00,0x03
,0x00,0x3c,0x60,0x1e,0x30,0x00,0x00,0x03
,0x00,0x78,0x70,0x0f,0x30,0x00,0x00,0x03
,0x03,0xe0,0x38,0x03,0x30,0x00,0x00,0x03
,0x07,0x80,0x1c,0x00,0x30,0x00,0x00,0x03
,0xc0,0x00,0x0f,0x00,0x30,0x00,0x00,0x03
,0xc0,0x00,0x03,0x80,0x30,0x01,0xe0,0x03
,0x00,0x18,0x01,0xe0,0x30,0x03,0xf0,0x03
,0x00,0x18,0x00,0x7c,0x30,0x07,0x38,0x03
,0x00,0x0c,0x00,0x1f,0x30,0x06,0x18,0x03
,0x18,0x0e,0x00,0x07,0x30,0x06,0x18,0x03
,0x0c,0x07,0x80,0x00,0x30,0x07,0x38,0x03
,0x0e,0x03,0xc0,0x00,0x30,0x03,0x30,0x03
,0x07,0x00,0xf0,0x00,0x30,0x03,0x30,0x03
,0x03,0x00,0x7e,0x00,0x30,0x03,0x30,0x03
,0x01,0x80,0x1f,0xc0,0x30,0x03,0x30,0x03
,0x01,0xc0,0x03,0xe1,0x30,0x07,0xf8,0x03
,0x00,0xf0,0x00,0x01,0x30,0x03,0xf0,0x03
,0x00,0x38,0x00,0x00,0x30,0x00,0x00,0x03
,0x00,0x1e,0x00,0x00,0x30,0x00,0x00,0x03
,0x00,0x07,0xc0,0x00,0x30,0x00,0x00,0x03
,0x00,0x01,0xff,0x80,0x3f,0xff,0xff,0xff
,0x00,0x00,0x3f,0x80,0x1f,0xff,0xff,0xfe
};
```

```
//-----
#define FinPr_scan_width 64
#define FinPr_scan_height 64
const uint8_t PROGMEM FinPr_scan_bits[] = {
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0x7f,0xff,0x00,0x00,0x00
,0x00,0x00,0x01,0xfc,0x7f,0xc0,0x00,0x00
,0x00,0x00,0x03,0xc0,0x03,0xe0,0x00,0x00
,0x00,0x00,0x07,0x80,0x00,0xf0,0x00,0x00
,0x00,0x00,0xe,0x00,0x00,0x3c,0x00,0x00
,0x00,0x00,0x1c,0x1f,0xfc,0x1c,0x00,0x00
,0x00,0x00,0x38,0x7f,0xfe,0x0e,0x00,0x00
,0x00,0x00,0x78,0xf8,0x0f,0x87,0x00,0x00
,0x00,0x00,0x71,0xe0,0x03,0xc7,0x00,0x00
,0x00,0x00,0xe3,0x80,0x01,0xc3,0x80,0x00
,0x00,0x00,0xc3,0x83,0xc0,0xe3,0x80,0x00
,0x00,0x00,0xc7,0x0f,0xf0,0x71,0x80,0x00
,0x00,0x00,0x06,0x1f,0xf8,0x71,0xc0,0x00
,0x00,0x00,0xe,0x1c,0x3c,0x31,0xc0,0x00
,0x00,0x00,0x1c,0x38,0x1c,0x31,0xc0,0x00
,0x00,0x00,0x38,0x70,0x0e,0x39,0xc0,0x00
,0x00,0x01,0xf0,0x71,0x8e,0x39,0xc0,0x00
,0x00,0x03,0xe0,0xe1,0x86,0x31,0xc0,0x00
,0x00,0x03,0x81,0xe3,0x8e,0x31,0x80,0x00
,0x00,0x00,0x03,0xc3,0x8e,0x33,0x80,0x00
,0x00,0x00,0x07,0x87,0x0c,0x73,0x80,0x00
,0x00,0x00,0x1f,0x0e,0x1c,0x73,0x80,0x00
,0x7f,0xff,0xff,0xff,0xff,0xff,0xfe
,0xff,0xff,0xff,0xff,0xff,0xff,0xff
,0xff,0xff,0xff,0xff,0xff,0xff,0xfe
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
```

```
,0x00,0x03,0xf0,0x1e,0x3e,0x1c,0x00,0x00
,0x00,0x03,0x80,0x7c,0x77,0x0f,0x00,0x00
,0x00,0x00,0x01,0xf0,0xe3,0x07,0xc0,0x00
,0x00,0x00,0x07,0xe3,0xc3,0x81,0xf0,0x00
,0x00,0x00,0x3f,0x87,0x81,0xc0,0x60,0x00
,0x00,0x01,0xfc,0x1f,0x00,0xf0,0x00,0x00
,0x00,0x01,0xe0,0x3c,0x00,0x7c,0x00,0x00
,0x00,0x00,0x00,0xf8,0x78,0x1f,0x00,0x00
,0x00,0x00,0x07,0xe0,0xfc,0x0f,0xc0,0x00
,0x00,0x00,0x3f,0x83,0xef,0x03,0xc0,0x00
,0x00,0x00,0xfc,0x0f,0x87,0x80,0x00,0x00
,0x00,0x00,0x70,0x1f,0x03,0xe0,0x00,0x00
,0x00,0x00,0x00,0x7c,0x00,0xf8,0x00,0x00
,0x00,0x00,0x01,0xf0,0x00,0x3e,0x00,0x00
,0x00,0x00,0x0f,0xc0,0xf8,0x0f,0x00,0x00
,0x00,0x00,0x1f,0x03,0xfe,0x02,0x00,0x00
,0x00,0x00,0x0c,0x0f,0x8f,0x80,0x00,0x00
,0x00,0x00,0x00,0x3f,0x03,0xe0,0x00,0x00
,0x00,0x00,0x00,0xf8,0x00,0xf0,0x00,0x00
,0x00,0x00,0x01,0xe0,0x00,0x30,0x00,0x00
,0x00,0x00,0x01,0xc0,0xf8,0x00,0x00,0x00
,0x00,0x00,0x00,0x07,0xfe,0x00,0x00,0x00
,0x00,0x00,0x00,0x0f,0x8e,0x00,0x00,0x00
,0x00,0x00,0x00,0x06,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};

//void Beep(int beep) {
//  digitalWrite(D0, HIGH);
//  delay(beep);
//  digitalWrite(D0, LOW);
//}

void LEDOK(int ok) {
  digitalWrite(D4, HIGH);
```

```

digitalWrite(D0, HIGH);
delay(ok);
digitalWrite(D4, LOW);
digitalWrite(D0, LOW);
}

void LEDFAILED(int failed) {
    digitalWrite(D3, HIGH);
    digitalWrite(D0, HIGH);
    delay(failed);
    digitalWrite(D3, LOW);
    digitalWrite(D0, LOW);
}
//*****************************************************************************
void setup() {

    Serial.begin(115200);
    pinMode(D3, OUTPUT); //RED
    pinMode(D4, OUTPUT); //BLUE
    pinMode(D0, OUTPUT); //BUZZER

    //-----initiate OLED display-----

    // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V
    // internally
    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address
        0x3D for 128x64
        Serial.println(F("SSD1306 allocation failed"));
        for(;); // Don't proceed, loop forever
    }
    // Show initial display buffer contents on the screen --
    // the library initializes this with an Adafruit splash screen.
    // you can delet these three lines if you don't want to get the Adfruit logo
    appear
    display.display();
    delay(2000); // Pause for 2 seconds
    display.clearDisplay();

    //-----
}

```

```

connectToWiFi();

//-----
// set the data rate for the sensor serial port
finger.begin(57600);
Serial.println("\n\nDetecting fingerprint sensor...");

if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
    display.clearDisplay();
    display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width,
FinPr_valid_height, WHITE);
    display.display();
} else {
    Serial.println("Did not find fingerprint sensor :(");
    display.clearDisplay();
    display.drawBitmap( 32, 0, FinPr_failed_bits, FinPr_failed_width,
FinPr_failed_height, WHITE);
    display.display();
    while (1) { delay(1); }
}
//-----

finger.getTemplateCount();
Serial.print("\nSensor contains "); Serial.print(finger.templateCount);
Serial.println(" templates");
Serial.println("Waiting for valid finger...");

//-----*test the connection*-----
//SendFingerprintID( FingerID );

}

//*****void loop() {

//check if there's a connection to WiFi or not

```

```

if(WiFi.status() != WL_CONNECTED){
    connectToWiFi();
}
//-----
//If there no fingerprint has been scanned return -1 or -2 if there an error
or 0 if there nothing, The ID start form 1 to 127
FingerID = getFingerprintID(); // Get the Fingerprint ID from the
Scanner
delay(50); //don't need to run this at full speed.

//-----
DisplayFingerprintID();

//-----
ChecktoAddID();

//-----
ChecktoDeleteID();

//-----
SendFingerprintID( FingerID );
delay(5000);
}

//*****Display the fingerprint ID state on the OLED*****
void DisplayFingerprintID(){
//Fingerprint has been detected
if (FingerID > 0){
    display.clearDisplay();
    display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width,
FinPr_valid_height, WHITE);
    display.display();
    LEDOK(300);
}

```

```

    SendFingerprintID( FingerID ); // Send the Fingerprint ID to the
website.

}

//-----
//No finger detected
else if (FingerID == 0){
    display.clearDisplay();
    display.drawBitmap( 32, 0, FinPr_start_bits, FinPr_start_width,
FinPr_start_height, WHITE);
    display.display();
}
//-----
//Didn't find a match
else if (FingerID == -1){
    display.clearDisplay();
    display.drawBitmap( 34, 0, FinPr_invalid_bits, FinPr_invalid_width,
FinPr_invalid_height, WHITE);
    display.display();
    LEDFAILED(200);
    delay(200);
    LEDFAILED(200);
}
//-----
//Didn't find the scanner or there an error
else if (FingerID == -2){
    display.clearDisplay();
    display.drawBitmap( 32, 0, FinPr_failed_bits, FinPr_failed_width,
FinPr_failed_height, WHITE);
    display.display();
    LEDFAILED(100);
}
}

//*****send the fingerprint ID to the website*****
void SendFingerprintID( int finger ){

HTTPClient http; //Declare object of class HTTPClient
//Post Data

```

```

postData = "FingerID=" + String(finger); // Add the Fingerprint ID to
the Post array in order to send it
// Post methode

http.begin(host); //initiate HTTP request, put your Website URL or Your
Computer IP
http.addHeader("Content-Type", "application/x-www-form-
urlencoded"); //Specify content-type header

int httpCode = http.POST(postData); //Send the request
String payload = http.getString(); //Get the response payload

Serial.println(httpCode); //Print HTTP return code
Serial.println(payload); //Print request response payload
Serial.println(postData); //Post Data
Serial.println(finger); //Print fingerprint ID

if (payload.substring(0, 5) == "login") {
    String user_name = payload.substring(5);
    // Serial.println(user_name);

    display.clearDisplay();
    display.setTextSize(2); // Normal 2:2 pixel scale
    display.setTextColor(WHITE); // Draw white text
    display.setCursor(15,0); // Start at top-left corner
    display.print(F("Welcome"));
    display.setCursor(0,20);
    display.print(user_name);
    display.display();
}
delay(1000);

postData = "";
http.end(); //Close connection
}
*****Get the Fingerprint ID*****
int getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {

```

```

case FINGERPRINT_OK:
    //Serial.println("Image taken");
    break;
case FINGERPRINT_NOFINGER:
    //Serial.println("No finger detected");
    return 0;
case FINGERPRINT_PACKETRECIEVEERR:
    //Serial.println("Communication error");
    return -2;
case FINGERPRINT_IMAGEFAIL:
    //Serial.println("Imaging error");
    return -2;
default:
    //Serial.println("Unknown error");
    return -2;
}
// OK success!
p = finger.image2Tz();
switch (p) {
    case FINGERPRINT_OK:
        //Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        //Serial.println("Image too messy");
        return -1;
    case FINGERPRINT_PACKETRECIEVEERR:
        //Serial.println("Communication error");
        return -2;
    case FINGERPRINT_FEATUREFAIL:
        //Serial.println("Could not find fingerprint features");
        return -2;
    case FINGERPRINT_INVALIDIMAGE:
        //Serial.println("Could not find fingerprint features");
        return -2;
    default:
        //Serial.println("Unknown error");
        return -2;
}
// OK converted!

```

```

p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    //Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    //Serial.println("Communication error");
    return -2;
} else if (p == FINGERPRINT_NOTFOUND) {
    //Serial.println("Did not find a match");
    return -1;
} else {
    //Serial.println("Unknown error");
    return -2;
}
// found a match!
//Serial.print("Found ID #"); Serial.print(finger.fingerID);
//Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;
}
//*****Check if there a Fingerprint ID to delete*****
void ChecktoDeleteID(){

HTTPClient http; //Declare object of class HTTPClient
//Post Data
postData = "DeleteID=check"; // Add the Fingerprint ID to the Post array
in order to send it
// Post methode

http.begin(host); //initiate HTTP request, put your Website URL or Your
Computer IP
http.addHeader("Content-Type", "application/x-www-form-
urlencoded"); //Specify content-type header

int httpCode = http.POST(postData); //Send the request
String payload = http.getString(); //Get the response payload

if (payload.substring(0, 6) == "del-id") {
    String del_id = payload.substring(6);
    Serial.println(del_id);
}

```

```

    deleteFingerprint( del_id.toInt() );
}

http.end(); //Close connection
}
/****************Delete Fingerprint ID********************
uint8_t deleteFingerprint( int id) {
    uint8_t p = -1;

    p = finger.deleteModel(id);

    if (p == FINGERPRINT_OK) {
        //Serial.println("Deleted!");
        display.clearDisplay();
        display.setTextSize(2);          // Normal 2:2 pixel scale
        display.setTextColor(WHITE);      // Draw white text
        display.setCursor(0,0);          // Start at top-left corner
        display.print(F("Deleted!\n"));
        display.display();
    } else if (p == FINGERPRINT_PACKETRECEIVEERR) {
        //Serial.println("Communication error");
        display.clearDisplay();
        display.setTextSize(1);          // Normal 1:1 pixel scale
        display.setTextColor(WHITE);      // Draw white text
        display.setCursor(0,0);          // Start at top-left corner
        display.print(F("Communication error!\n"));
        display.display();
        return p;
    } else if (p == FINGERPRINT_BADLOCATION) {
        //Serial.println("Could not delete in that location");
        display.clearDisplay();
        display.setTextSize(1);          // Normal 1:1 pixel scale
        display.setTextColor(WHITE);      // Draw white text
        display.setCursor(0,0);          // Start at top-left corner
        display.print(F("Could not delete in that location!\n"));
        display.display();
        return p;
    } else if (p == FINGERPRINT_FLASHERR) {
        //Serial.println("Error writing to flash");

```

```

display.clearDisplay();
display.setTextSize(1);      // Normal 1:1 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0);     // Start at top-left corner
display.print(F("Error writing to flash!\n"));
display.display();
return p;
} else {
//Serial.print("Unknown error: 0x"); Serial.println(p, HEX);
display.clearDisplay();
display.setTextSize(2);      // Normal 2:2 pixel scale
display.setTextColor(WHITE); // Draw white text
display.setCursor(0,0);     // Start at top-left corner
display.print(F("Unknown error:\n"));
display.display();
return p;
}
//*****Check if there a Fingerprint ID to add*****
void ChecktoAddID(){

HTTPClient http; //Declare object of class HTTPClient
//Post Data
postData = "Get_Fingerid=get_id"; // Add the Fingerprint ID to the Post
array in order to send it
// Post methode

http.begin(host); //initiate HTTP request, put your Website URL or Your
Computer IP
http.addHeader("Content-Type", "application/x-www-form-
urlencoded"); //Specify content-type header

int httpCode = http.POST(postData); //Send the request
String payload = http.getString(); //Get the response payload

if (payload.substring(0, 6) == "add-id") {
String add_id = payload.substring(6);
Serial.println(add_id);
id = add_id.toInt();
}

```

```

        getFingerprintEnroll();
    }
    http.end(); //Close connection
}
//*****Enroll a Fingerprint ID*****
uint8_t getFingerprintEnroll() {

    int p = -1;
    display.clearDisplay();
    display.drawBitmap( 34, 0, FinPr_scan_bits, FinPr_scan_width,
FinPr_scan_height, WHITE);
    display.display();
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
        case FINGERPRINT_OK:
            //Serial.println("Image taken");
            display.clearDisplay();
            display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width,
FinPr_valid_height, WHITE);
            display.display();
            break;
        case FINGERPRINT_NOFINGER:
            //Serial.println(".");
            display.setTextSize(1);      // Normal 2:2 pixel scale
            display.setTextColor(WHITE); // Draw white text
            display.setCursor(0,0);     // Start at top-left corner
            display.print(F("scanning"));
            display.display();
            break;
        case FINGERPRINT_PACKETRECIEVEERR:
            display.clearDisplay();
            display.drawBitmap( 34, 0, FinPr_invalid_bits, FinPr_invalid_width,
FinPr_invalid_height, WHITE);
            display.display();
            break;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error");
            break;
    }
}

```

```

default:
    Serial.println("Unknown error");
    break;
}
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
    case FINGERPRINT_OK:
        display.clearDisplay();
        display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width,
FinPr_valid_height, WHITE);
        display.display();
        break;
    case FINGERPRINT_IMAGEMESS:
        display.clearDisplay();
        display.drawBitmap( 34, 0, FinPr_invalid_bits, FinPr_invalid_width,
FinPr_invalid_height, WHITE);
        display.display();
        return p;
    case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}
display.clearDisplay();
display.setTextSize(2);          // Normal 2:2 pixel scale
display.setTextColor(WHITE);     // Draw white text
display.setCursor(0,0);         // Start at top-left corner

```

```
display.print(F("Remove"));
display.setCursor(0,20);
display.print(F("finger"));
display.display();
//Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
display.clearDisplay();
display.drawBitmap( 34, 0, FinPr_scan_bits, FinPr_scan_width,
FinPr_scan_height, WHITE);
display.display();
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            //Serial.println("Image taken");
            display.clearDisplay();
            display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width,
FinPr_valid_height, WHITE);
            display.display();
            break;
        case FINGERPRINT_NOFINGER:
            //Serial.println(".");
            display.setTextSize(1);      // Normal 2:2 pixel scale
            display.setTextColor(WHITE); // Draw white text
            display.setCursor(0,0);     // Start at top-left corner
            display.print(F("scanning"));
            display.display();
            break;
        case FINGERPRINT_PACKETRECIEVEERR:
            Serial.println("Communication error");
            break;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error");
    }
}
```

```

        break;
    default:
        Serial.println("Unknown error");
        break;
    }
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
    case FINGERPRINT_OK:
        //Serial.println("Image converted");
        display.clearDisplay();
        display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width,
FinPr_valid_height, WHITE);
        display.display();
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();

```

```

if (p == FINGERPRINT_OK) {
    //Serial.println("Prints matched!");
    display.clearDisplay();
    display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width,
    FinPr_valid_height, WHITE);
    display.display();
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    //Serial.println("Stored!");
    display.clearDisplay();
    display.drawBitmap( 34, 0, FinPr_valid_bits, FinPr_valid_width,
    FinPr_valid_height, WHITE);
    display.display();
    confirmAdding();
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

```

```

}

//*****Check if there a Fingerprint ID to add*****
void confirmAdding(){

    HTTPClient http; //Declare object of class HTTPClient
    //Post Data
    postData = "confirm_id=" + String(id); // Add the Fingerprint ID to the
    Post array in order to send it
    // Post methode

    http.begin(host); //initiate HTTP request, put your Website URL or Your
    Computer IP
    http.addHeader("Content-Type", "application/x-www-form-
    urlencoded"); //Specify content-type header

    int httpCode = http.POST(postData); //Send the request
    String payload = http.getString(); //Get the response payload

    display.clearDisplay();
    display.setTextSize(1.5); // Normal 1:1 pixel scale
    display.setTextColor(WHITE); // Draw white text
    display.setCursor(0,0); // Start at top-left corner
    display.print(payload);
    display.display();
    delay(1000);
    Serial.println(payload);

    http.end(); //Close connection
}

//*****connect to the WiFi*****
void connectToWiFi(){
    WiFi.mode(WIFI_OFF); //Prevents reconnection issue (taking too
    long to connect)
    delay(1000);
    WiFi.mode(WIFI_STA);
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    // WiFi.config(IP, NETWORK, NETMASK, DNS);
}

```

```

display.clearDisplay();
display.setTextSize(1);          // Normal 1:1 pixel scale
display.setTextColor(WHITE);     // Draw white text
display.setCursor(0, 0);         // Start at top-left corner
display.print(F("\n\nConnecting to \n"));
display.setCursor(0, 50);
display.setTextSize(2);
display.print(ssid);
display.drawBitmap( 73, 10, Wifi_start_bits, Wifi_start_width,
Wifi_start_height, WHITE);
display.display();

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("Connected");

display.clearDisplay();
display.setTextSize(2);          // Normal 1:1 pixel scale
display.setTextColor(WHITE);     // Draw white text
display.setCursor(8, 0);         // Start at top-left corner
display.print(F("Connected \n"));
display.drawBitmap( 33, 15, Wifi_connected_bits,
Wifi_connected_width, Wifi_connected_height, WHITE);
display.display();

Serial.print("IP address: ");
Serial.println(WiFi.localIP()); //IP address assigned to your ESP
}

}

```

## **2. Pemrograman Web**

### **A. Halaman Login**

#### **1. login.page.html**

```
<ion-header>
<ion-toolbar>
  <ion-title>login</ion-title>
</ion-toolbar>
</ion-header>

<ion-content>
<ion-buttons>
  <ion-button (click)="httpGet()">
    GET
  </ion-button>
  <ion-button (click)="httpPost()">
    POST
  </ion-button>
</ion-buttons>

</ion-content>
```

#### **2. login.page.ts**

```
import { Component, OnInit } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-login',
  templateUrl: './login.page.html',
  styleUrls: ['./login.page.scss'],
})
export class LoginPage implements OnInit {

  constructor(
    private http: HttpClient
  ) { }
```

```

ngOnInit() {
}

httpGet() {
    this.http.get('http://192.168.100.17:8100/login').subscribe((res) => {
        console.log(res);
    });
    console.log('get');
}

httpPost() {
    this.http.post('http://192.168.100.17:8100/login',
        'aaanjaaayy').subscribe((res) => {
        console.log(res);
    });
    console.log('post');
}
}

```

## B. Halaman Mata Kuliah – Mahasiswa

### 1. mata-kuliah.page.html

```

<ion-header>
    <ion-toolbar>
        <ion-title>Oasys</ion-title>
    </ion-toolbar>
</ion-header>

<ion-content>
    <div *ngIf="mk && mhs">
        <ion-item lines="none" class="greeting">
            <ion-label>
                <span>Selamat datang</span> <br>
                <span *ngIf="xUser.access.length === 2" style="font-size: 28px;">ADMIN</span>

```

```
<span *ngIf="xUser.access.length === 1" style="font-size: 28px;">{{ mhs.nama }} - {{ mhs.NRP }}</span>
</ion-label>
</ion-item>
<ion-card class="mainContent">
<ion-card-header>
<ion-segment (ionChange)="segmentChanged($event)" value="mata-kuliah">
<ion-segment-button expand="full" value="mata-kuliah">
<ion-label>Mata Kuliah</ion-label>
</ion-segment-button>
<ion-segment-button value="laporan">
<ion-label>Laporan</ion-label>
</ion-segment-button>
<ion-segment-button value="logout">
<ion-label>LogOut</ion-label>
</ion-segment-button>
</ion-segment>
</ion-card-header>
<ion-card-content>
<ion-list class="daftar">
<ion-card class="title">
<ion-grid style="padding: 0;">
<ion-row>
<ion-col size="1">
<ion-label>NO</ion-label>
</ion-col>
<ion-col size="3">
<ion-label>NAMA</ion-label>
</ion-col>
<ion-col size="2">
<ion-label>HARI</ion-label>
</ion-col>
<ion-col size="2">
<ion-label>JAM</ion-label>
</ion-col>
<ion-col size="2">
<ion-label>DOSEN</ion-label>
</ion-col>
```

```
<ion-col size="2">
  <ion-label>RUANG</ion-label>
</ion-col>
</ion-row>
</ion-grid>
</ion-card>

<div class="isi">
<ion-card *ngFor="let item of mk; let i = index">
  <ion-card-header style="color: black;">
    <ion-grid style="padding: 0;">
      <ion-row>
        <ion-col size="1">
          <ion-label>{ {i+1} }</ion-label>
        </ion-col>
        <ion-col size="3" (click)="expandDetails(item)">
          <ion-label>{ {item.nama} }</ion-label>
        </ion-col>
        <ion-col size="2">
          <ion-label>{ {item.jadwal[0].hari} }</ion-label>
        </ion-col>
        <ion-col size="2">
          <ion-label>{ {item.jadwal[0].mulai} }{ {item.jadwal[0].selesai} }</ion-label>
        </ion-col>
        <ion-col size="2">
          <ion-label>{ {item.dosen} }</ion-label>
        </ion-col>
        <ion-col size="2">
          <ion-label>{ {item.ruang} }</ion-label>
        </ion-col>
      </ion-row>
    </ion-grid>
  </ion-card-header>
  <ion-card-content *ngIf="item.expand" style="padding-top: 10px; background-color: #dfd8ff;">
    <ion-grid style="padding: 0;">
      <ion-row class="expandTitle">
        <ion-col size="0.5">
```

```

<ion-label>No</ion-label>
</ion-col>
<ion-col size="1.5">
  <ion-label>Tanggal</ion-label>
</ion-col>
<ion-col size="1.5">
  <ion-label>Minggu</ion-label>
</ion-col>
<ion-col size="2">
  <ion-label>Waktu Masuk</ion-label>
</ion-col>
<ion-col size="2">
  <ion-label>Waktu Absen</ion-label>
</ion-col>
<ion-col size="1.75">
  <ion-label>Keterlambatan</ion-label>
</ion-col>
<ion-col size="1.75">
  <ion-label>Keterangan</ion-label>
</ion-col>
<ion-col size="1">
  <ion-label>%</ion-label>
</ion-col>
</ion-row>
</ion-grid>

<ion-grid *ngFor="let detail of item.jadwal; let j = index"
style="padding: 0;">
  <ion-row class="expandIsi">
    <ion-col size="0.5">
      <ion-label>{ {j+1} }</ion-label>
    </ion-col>
    <ion-col size="1.5">
      <ion-label>{ {detail.tanggal} }</ion-label>
    </ion-col>
    <ion-col size="1.5">
      <ion-label>{ {j+1} }</ion-label>
    </ion-col>
    <ion-col size="2">

```

```

<ion-label>{ {detail.mulai} }</ion-label>
</ion-col>
<ion-col size="2">
  <ion-label>{ {detail.waktu_hadir} }</ion-label>
</ion-col>
<ion-col size="1.75">
  <ion-label>{ {hitungKeterlambatan(detail.mulai,
detail.waktu_hadir)} } menit</ion-label>
</ion-col>
<ion-col size="1.75">
  <ion-label *ngIf="detail.kehadiran === 100" >Hadir</ion-
label>
  <ion-label *ngIf="detail.kehadiran === 50" >Ijin</ion-
label>
  <ion-label *ngIf="detail.kehadiran === 0" >Alpha</ion-
label>
</ion-col>
<ion-col size="1">
  <ion-label>{ {detail.kehadiran} }%</ion-label>
</ion-col>
</ion-row>
</ion-grid>
</ion-card-content>
</ion-card>
</div>
</ion-list>
</ion-card-content>
</ion-card>
</div>
</ion-content>

```

## 2. mata-kuliah.page.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { DataService } from 'src/app/services/data.service';
import { Mahasiswa, MataKuliah } from 'src/app/services/interface';
import { map } from 'rxjs/operators';

```

```

import { UserService } from 'src/app/services/user.service';
import { PopupService } from 'src/app/services/popup.service';

@Component({
  selector: 'app-mata-kuliah',
  templateUrl: './mata-kuliah.page.html',
  styleUrls: ['./mata-kuliah.page.scss'],
})
export class MataKuliahPage implements OnInit {

  expand: boolean;
  segmentValue;
  xUser;
  mhs: Mahasiswa;
  mk: any[]; // matkul + kehadiran,waktu hadir mhs
  task; task2; task3;
  temp;

  constructor(
    private router: Router,
    private data: DataService,
    private user: UserService,
    private popup: PopupService
  ) { }

  ngOnInit() {
    this.expand = false;
    this.segmentValue = 'mata-kuliah';
    this.task = this.user.user.subscribe(val => {
      this.xUser = val;
      this.task2 = this.data.getMahasiswaByNRP(val.idUser).subscribe(res
        => {
          this.mk = [];
          this.mhs = res;
          console.log(this.mhs);
          this.mhs.matkul.forEach((item, index) => {
            this.task3 =
              this.data.getMataKuliah(item.idMatkul).subscribe(res2 => {
                =
              })
          })
        })
    })
  }
}

```

```

this.mk[index] = res2;
this.mk[index].expand = false;
this.mk[index].jadwal.forEach((x, i) => {
  x.kehadiran = this.mhs.matkul[index].kehadiran[i];
  x.waktu_hadir = this.mhs.matkul[index].waktu_hadir[i];
});
// this.temp = res2;
// console.log(this.mhs.matkul[index].kehadiran);
// console.log('test');
// console.log(this.mhs.matkul[index].waktu_hadir);
console.log(this.mk);
});
});
});
});
}
}

expandDetails(item) {
  this.mk.forEach(val => {
    if (val !== item) {
      val.expand = false;
    } else {
      val.expand = !val.expand;
    }
  });
}

hitungKeterlambatan(jadwal: string, absen: string) {
  if (absen !== 'unset') {
    const jamJ = parseInt(jadwal.split('.')[0], 10);
    const menitJ = parseInt(jadwal.split('.')[1], 10);
    const jamA = parseInt(absen.split('.')[0], 10);
    const menitA = parseInt(absen.split('.')[1], 10);
    return ((jamA - jamJ) * 60) + (menitA - menitJ);
  } else {
    return '?';
  }
}

```

```

segmentChanged(event) {
  this.segmentValue = 'laporan';
  console.log(event.detail.value);
  if (event.detail.value !== 'logout') {
    this.router.navigate(['mahasiswa/${event.detail.value}']);
  } else {
    this.user.logout();
  }
}

```

## C. Halaman Laporan – Mahasiswa

### 1. laporan.page.html

```

<ion-header>
  <ion-toolbar>
    <ion-title>Oasys</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div *ngIf="mk">
    <ion-item lines="none" class="greeting">
      <ion-label>
        <span>Selamat datang</span> <br>
        <span *ngIf="xUser.access.length === 2" style="font-size: 28px;">ADMIN</span>
        <span *ngIf="xUser.access.length === 1" style="font-size: 28px;">{ { mhs.nama } } - { { mhs.NRP } }</span>
      </ion-label>
    </ion-item>
    <ion-card class="mainContent">
      <ion-card-header>
        <ion-segment (ionChange)="segmentChanged($event)" value="laporan">
          <ion-segment-button expand="full" value="mata-kuliah">

```

```

<ion-label>Mata Kuliah</ion-label>
</ion-segment-button>
<ion-segment-button value="laporan">
  <ion-label>Laporan</ion-label>
</ion-segment-button>
<ion-segment-button value="logout">
  <ion-label>LogOut</ion-label>
</ion-segment-button>
</ion-segment>
</ion-card-header>
<ion-card-content>
<ion-list class="daftar">
<ion-card class="title">
  <ion-grid style="padding: 0;">
    <ion-row>
      <ion-col size="1">
        <ion-label>NO</ion-label>
      </ion-col>
      <ion-col size="4">
        <ion-label>Mata Kuliah</ion-label>
      </ion-col>
      <ion-col size="3">
        <ion-label>Persentase Kehadiran</ion-label>
      </ion-col>
      <ion-col size="4">
        <ion-label>Keterangan</ion-label>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-card>
</ion-card>

<div class="isi">
<ion-card *ngFor="let item of mk; let i = index">
  <ion-card-header style="color: black;">
    (click)="expandDetails(item)">
      <ion-ripple-effect></ion-ripple-effect>
      <ion-grid style="padding: 0;">
        <ion-row>
          <ion-col size="1">

```

```

<ion-label>{ {i+1} }</ion-label>
</ion-col>
<ion-col size="4">
  <ion-label>{ {item.nama} }</ion-label>
</ion-col>
<ion-col size="3">
  <ion-label>{ {calcKehadiran(item)} }</ion-label>
</ion-col>
<ion-col size="4">
  <ion-label>{ {keterangan(item)} }</ion-label>
  <!-- <ion-label *ngIf="79% > calcKehadiran(item)">Tidak
Boleh Mengikuti EAS</ion-label>
    <ion-label *ngIf="calcKehadiran(item) > 80">Boleh
Mengikuti EAS</ion-label> -->
  </ion-col>
</ion-row>
</ion-grid>
</ion-card-header>
<ion-card-content *ngIf="item.expand" style="padding-top:
10px; background-color: #dfd8ff;">
  <ion-grid style="padding: 0;">
    <ion-row class="expandTitle">
      <ion-col size="1.5">
        <ion-label>Minggu ke-</ion-label>
      </ion-col>
      <ion-col *ngFor="let j of [1,2,3,4,5,6,7,8,9,10,11,12,13,14]">
        <ion-label>{ {j} }</ion-label>
      </ion-col>
    </ion-row>
  </ion-grid>

  <ion-grid style="padding: 0;">
    <ion-row class="expandIsi">
      <ion-col size="1.5">
        <ion-label>( % )</ion-label>
      </ion-col>
      <ion-col *ngFor="let j of [0,1,2,3,4,5,6,7,8,9,10,11,12,13]">
        <ion-label>{ {j} }</ion-label>
      </ion-col>
    </ion-row>
  </ion-grid>

```

```

<ion-label>{ {item.kehadiran[j]} }</ion-label>
</ion-col>
</ion-row>
</ion-grid>
</ion-card-content>
</ion-card>
</div>
</ion-list>
</ion-card-content>
</ion-card>
</div>
</ion-content>

```

## 2. laporan.page.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { DataService } from 'src/app/services/data.service';
import { Mahasiswa } from 'src/app/services/interface';
import { UserService } from 'src/app/services/user.service';

@Component({
  selector: 'app-laporan',
  templateUrl: './laporan.page.html',
  styleUrls: ['./laporan.page.scss'],
})
export class LaporanPage implements OnInit {

  expand: boolean;
  xUser;
  mhs: Mahasiswa;
  mk: any[]; // matkul + kehadiran,waktu hadir mhs
  task; task2; task3;
  temp;

  constructor(
    private router: Router,
    private data: DataService,
  )
}

```

```

private user: UserService
) {
  console.log('construct');
}

ngOnInit() {
  this.task = this.user.user.subscribe(val => {
    this.xUser = val;
    this.task2 = this.data.getMahasiswaByNRP(val.idUser).subscribe(res
      => {
      this.mk = [];
      this.mhs = res;
      console.log(this.mhs);
      this.mhs.matkul.forEach((item, index) => {
        this.task3 =
          this.data.getMataKuliah(item.idMatkul).subscribe(res2 => {
            this.mk[index] = res2;
            this.mk[index].expand = false;
            this.mk[index].kehadiran = this.mhs.matkul[index].kehadiran;
            this.mk[index].waktu_hadir =
              this.mhs.matkul[index].waktu_hadir;
            // this.temp = res2;
            console.log(this.mhs.matkul[index].kehadiran);
            console.log('test');
            console.log(this.mhs.matkul[index].waktu_hadir);
          });
        });
      });
    });
}

expandDetails(item) {
  this.mk.forEach(val => {
    if (val !== item) {
      val.expand = false;
    } else {
      val.expand = !val.expand;
    }
  });
}

```

```
}

calcKehadiran(item) {
  let sum: number;
  sum = 0;
  item.kehadiran.forEach(val => {
    if (val) {
      sum += val;
      console.log(item.nama, val);
    }
  });
  return sum / item.kehadiran.length;
}

keterangan(item) {
  if (this.calcKehadiran(item) >= 80) {
    return 'Boleh Mengikuti EAS';
  } else {
    return 'Tidak Boleh Mengikuti EAS';
  }
}

segmentChanged(event) {
  console.log(event.detail.value);
  if (event.detail.value !== 'logout') {
    this.router.navigate(['mahasiswa/${event.detail.value}']);
  } else {
    this.user.logout();
  }
}

}
```

## D. Halaman Mata Kuliah – Dosen

### 1. matkul.page.html

```
<ion-header>
<ion-toolbar>
  <ion-title>Oasys</ion-title>
</ion-toolbar>
</ion-header>

<ion-content>
<div *ngIf="mk">
  <ion-item lines="none" class="greeting">
    <ion-label>
      <span>Selamat datang</span> <br>
      <span *ngIf="xUser.access.length === 2" style="font-size: 28px;">ADMIN</span>
      <span *ngIf="xUser.access.length === 1" style="font-size: 28px;">Dosen - {{ xUser.idUser }}</span>
    </ion-label>
  </ion-item>
  <ion-card class="mainContent">
    <ion-card-header>
      <ion-segment (ionChange)="segmentChanged($event)" value="matkul">
        <ion-segment-button expand="full" value="matkul">
          <ion-label>Mata Kuliah</ion-label>
        </ion-segment-button>
        <ion-segment-button value="daftar-mahasiswa">
          <ion-label>Daftar Mahasiswa</ion-label>
        </ion-segment-button>
        <ion-segment-button value="logout">
          <ion-label>LogOut</ion-label>
        </ion-segment-button>
      </ion-segment>
    </ion-card-header>
    <ion-card-content>
      <ion-list class="daftar">
        <ion-card class="title">
```

```
<ion-grid style="padding: 0;">
<ion-row>
<ion-col size="1">
<ion-label>NO</ion-label>
</ion-col>
<ion-col size="3">
<ion-label>Mata Kuliah</ion-label>
</ion-col>
<ion-col size="2">
<ion-label>Hari, Tanggal</ion-label>
</ion-col>
<ion-col size="2">
<ion-label>Jam</ion-label>
</ion-col>
<ion-col size="2">
<ion-label>Ruang</ion-label>
</ion-col>
<ion-col size="2">
<ion-label></ion-label>
</ion-col>
</ion-row>
</ion-grid>
</ion-card>

<div class="isi">
<ion-card *ngFor="let item of mk; let i = index">
<ion-card-header style="color: black;">
<ion-grid style="padding: 0;">
<ion-row>
<ion-col size="1">
<ion-label>{ {i+1} }</ion-label>
</ion-col>
<ion-col size="3" (click)="expandDetails(item)">
<ion-label>{ {item.nama} } ({ {item.idMatkul} }}</ion-label>
</ion-col>
<ion-col size="2">
<ion-label>{ {item.jadwal[0].hari} }</ion-label>
</ion-col>
```

```

<ion-col size="2">
  <ion-label>{ {item.jadwal[0].mulai} } - 
  { {item.jadwal[0].selesai} }</ion-label>
</ion-col>
<ion-col size="2">
  <ion-label>{ {item.ruang} }</ion-label>
</ion-col>
<ion-col size="2" style="padding: 7px;">
  <ion-button expand="block" class="changeJadwal"
(click)="ubahJadwal(item.idMatkul)">
    <ion-label>Ubah Jadwal</ion-label>
  </ion-button>
  <!-- <ion-button expand="block" class="changeJadwal"
(click)="addTgl(item)">
    <ion-label>Add Tanggal</ion-label>
  </ion-button> -->
</ion-col>
</ion-row>
</ion-grid>
</ion-card-header>
<ion-card-content *ngIf="item.expand" style="padding-top:
10px; background-color: #dfd8ff;">
  <ion-grid style="padding: 0;">
    <ion-row class="expandTitle">
      <ion-col size="0.5">
        <ion-label>NO</ion-label>
      </ion-col>
      <ion-col size="1.75">
        <ion-label>Nama Mahasiswa</ion-label>
      </ion-col>
      <ion-col size="1.75">
        <ion-label>NRP</ion-label>
      </ion-col>
      <ion-col *ngFor="let i of [1,2,3,4,5,6,7,8,9,10,11,12,13,14]" size="0.5">
        <ion-label>{ {i} }</ion-label>
      </ion-col>
      <ion-col size="1">
        <ion-label>Total</ion-label>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-card-content>

```

```

</ion-col>
</ion-row>
<ion-row class="expandIsi" *ngFor="let person of item.peserta; let j = index">
  <ion-col size="0.5">
    <ion-label>{ {j+1} }</ion-label>
  </ion-col>
  <ion-col size="1.75">
    <ion-label>{ {person.nama} }</ion-label>
  </ion-col>
  <ion-col size="1.75">
    <ion-label>{ {person.NRP} }</ion-label>
  </ion-col>
  <ion-col *ngFor="let x of item.kehadiranPeserta[j]" size="0.5">
    <ion-label>{ {x} }</ion-label>
  </ion-col>
  <ion-col *ngFor="let x of kosongan(item.kehadiranPeserta[j])" size="0.5">
    <ion-label> </ion-label>
  </ion-col>
  <ion-col size="1">
    <ion-label>{ {calcKehadiran(item.kehadiranPeserta[j])} }</ion-label>
  </ion-col>
</ion-row>
</ion-grid>
</ion-card-content>
</ion-card>
</div>
</ion-list>
</ion-card-content>
</ion-card>
</div>
</ion-content>

```

## 2. matkul.page.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { DataService } from 'src/app/services/data.service';
import { Mahasiswa } from 'src/app/services/interface';
import { UserService } from 'src/app/services/user.service';
import { ModalController } from '@ionic/angular';
import { UbahjadwalPage } from './modals/ubahjadwal/ubahjadwal.page';
import { PopupService } from 'src/app/services/popup.service';

@Component({
  selector: 'app-matkul',
  templateUrl: './matkul.page.html',
  styleUrls: ['./matkul.page.scss'],
})
export class MatkulPage implements OnInit {

  expand: boolean;
  mhs: Mahasiswa[];
  mk: any[];
  xUser;
  task; task2; task3;

  constructor(
    private router: Router,
    private data: DataService,
    private user: UserService,
    private modal: ModalController,
    private popup: PopupService
  ) { }

  ngOnInit() {
    this.expand = false;
    this.task = this.user.user.subscribe(val1 => {
      this.xUser = val1;
    });
    this.task2 = this.data.getAllMahasiswa().subscribe(res => {
      this.mhs = res;
    });
  }
}
```

```

this.task3
  this.data.getMatkulByDosen(this.xUser.idUser).subscribe(res2
    => {
      this.mk = res2;
      this.mk.forEach(item => {
        item.peserta = [];
        item.kehadiranPeserta = [];
        item.expand = false;
        this.mhs.forEach(person => {
          person.matkul.forEach(val => {
            if (val.idMatkul === item.idMatkul) {
              // console.log(person);
              item.peserta.push(person);
              item.kehadiranPeserta.push(val.kehadiran);
            }
          });
        });
      });
    });
  });
}

segmentChanged(event) {
  if (event.detail.value !== 'logout') {
    this.router.navigate(['dosen/${event.detail.value}']);
  } else {
    this.user.logout();
  }
}

expandDetails(item) {
  this.mk.forEach(val => {
    if (val !== item) {
      val.expand = false;
    } else {
      val.expand = !val.expand;
    }
  });
}

```

```
        console.log(item.peserta);
    }

kosongan(param) {
    const x = 14 - param.length;
    // console.log('x', x);
    const y = [];
    for (let i = 0; i < x; i++) {
        y.push(i);
    }
    // console.log('y', y);
    return y;
}

calcKehadiran(param) {
    let sum: number;
    sum = 0;
    param.forEach(val => {
        if (val) {
            sum += val;
        }
    });
    return sum / param.length;
}

async ubahJadwal(idmat) {
    const modal = await this.modal.create({
        component: UbahjadwalPage,
        componentProps: {
            idmat
        }
    });
    return await modal.present();
}

addTgl(mkul) {
    // const tgl = [6, 13, 20, 27, 4, 11, 18, 25, 1, 8, 15, 22, 29, 6];
    // if (mkul.jadwal.hari === 'Selasa') {
    //     tgl.forEach((x, i) => {
```

```

//   tgl[i] = x + 2;
// });
// } else if (mkul.jadwal.hari === 'Rabu') {
// tgl.forEach((x, i) => {
//   tgl[i] = x + 2;
// });
// } else if (mkul.jadwal.hari === 'Kamis') {
// tgl.forEach((x, i) => {
//   tgl[i] = x + 2;
// });
// } else if (mkul.jadwal.hari === 'Jumat') {
// tgl.forEach((x, i) => {
//   tgl[i] = x + 2;
// });
// }
// console.log(tgl);
this.data.addTanggal(mkul).then(
() => {
  this.popup.showAlert('SUCCESS', 'add tanggal Success');
},
(err) => {
  this.popup.showAlert('ERROR', err);
}
);
}

}

```

## E. Halaman Daftar Mahasiswa – Dosen

### 1. daftar-mahasiswa.page.html

```

<ion-header>
<ion-toolbar>
  <ion-title>Oasys</ion-title>
</ion-toolbar>
</ion-header>

```

```
<ion-content>
<div *ngIf="mhs">
<ion-item lines="none" class="greeting">
<ion-label>
<span *ngIf="xUser.access.length === 2" style="font-size: 28px;">ADMIN</span>
<span *ngIf="xUser.access.length === 1" style="font-size: 28px;">Dosen - {{xUser.idUser}}</span>
</ion-label>
</ion-item>
<ion-card class="mainContent">
<ion-card-header>
<ion-segment (ionChange)="segmentChanged($event)"
  value="daftar-mahasiswa">
  <ion-segment-button expand="full" value="matkul">
    <ion-label>Mata Kuliah</ion-label>
  </ion-segment-button>
  <ion-segment-button value="daftar-mahasiswa">
    <ion-label>Daftar Mahasiswa</ion-label>
  </ion-segment-button>
  <ion-segment-button value="logout">
    <ion-label>LogOut</ion-label>
  </ion-segment-button>
</ion-segment>
</ion-card-header>
<ion-card-content>
<ion-list class="daftar">
<ion-card class="title">
<ion-grid style="padding: 0;">
<ion-row>
<ion-col size="1">
<ion-label>NO</ion-label>
</ion-col>
<ion-col size="4">
<ion-label>Nama Mahasiswa</ion-label>
</ion-col>
<ion-col size="2">
<ion-label>NRP</ion-label>
</ion-col>
```

```

<ion-col size="2">
  <ion-label>Semester</ion-label>
</ion-col>
<ion-col size="3">
  <ion-label></ion-label>
</ion-col>
</ion-row>
</ion-grid>
</ion-card>

<div class="isi">
<ion-card *ngFor="let item of mhs; let i = index">
  <ion-card-header style="color: black;">
    <ion-grid style="padding: 0;">
      <ion-row>
        <ion-col size="1">
          <ion-label>{ {i+1} }</ion-label>
        </ion-col>
        <ion-col size="4">
          <ion-label>{ {item.nama} }</ion-label>
        </ion-col>
        <ion-col size="2">
          <ion-label>{ {item.NRP} }</ion-label>
        </ion-col>
        <ion-col size="2">
          <ion-label>8</ion-label>
        </ion-col>
        <ion-col size="1.5" style="padding: 7px; border-right: 0;">
          <ion-button [disabled]="item.disableDaftar || disableAll"
expand="full" color="success" (click)="daftar(item)">
            <ion-label *ngIf="item.status === 'mendaftar'">Daftar</ion-label>
            <ion-spinner *ngIf="item.status === 'mendaftar'">
              name="dots" color="dark"</ion-spinner>
            </ion-button>
          </ion-col>
        <ion-col size="1.5" style="padding: 7px; border-left: 0;">
          <ion-button [disabled]="item.disableHapus || disableAll"
expand="full" color="danger" (click)="hapus(item)">

```

```

<ion-label *ngIf="item.status !== 'deleting'">Hapus</ion-label>
    <ion-spinner *ngIf="item.status === 'deleting'" name="dots" color="light"></ion-spinner>
        </ion-button>
    </ion-col>
</ion-row>
</ion-grid>
</ion-card-header>
</ion-card>
</div>
</ion-list>
</ion-card-content>
</ion-card>
</div>
</ion-content>

```

## 2. daftar-mahasiswa.page.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { DataService } from 'src/app/services/data.service';
import { Mahasiswa } from 'src/app/services/interface';
import { UserService } from 'src/app/services/user.service';
import { LoadingController } from '@ionic/angular';
import { PopupService } from 'src/app/services/popup.service';

@Component({
  selector: 'app-daftar-mahasiswa',
  templateUrl: './daftar-mahasiswa.page.html',
  styleUrls: ['./daftar-mahasiswa.page.scss'],
})
export class DaftarMahasiswaPage implements OnInit {

  mhs: any[];
  xUser;
  task; task2;
  disableAll = false;
}

```

```

constructor(
  private router: Router,
  private data: DataService,
  private user: UserService,
  private popup: PopupService,
  public loading: LoadingController
) { }

ngOnInit() {
  this.task = this.user.user.subscribe(val => {
    this.xUser = val;
    this.task2 = this.data.getAllMahasiswa().subscribe(res => {
      this.mhs = res.map(x => ({...x, disableDaftar: false, disableHapus:
        false}));
      let xx = 0;
      this.mhs.forEach(person => {
        person.disableHapus = (person.status === 'deleted' || person.status
          === 'belum' || person.status === 'deleting');
        person.disableDaftar = (person.status === 'terdaftar' || person.status
          === 'mendaftar');
        if (person.status === 'mendaftar' || person.status === 'deleting') {
          xx = 1;
        }
      });
      if (xx) {
        this.disableAll = true;
      } else {
        this.disableAll = false;
      }
      console.log(this.disableAll, xx);
    });
  });
}

segmentChanged(event) {
  if (event.detail.value !== 'logout') {
    console.log(event.detail.value);
    this.router.navigate(['dosen/${event.detail.value}`]);
  }
}

```

```

    } else {
        this.user.logout();
    }
}

daftar(person) {
    // console.log(person);
    this.data.setMendaftar(person.NRP).then(
        () => {
            this.popup.showAlert('Tunggu Sejenak', 'sistem proses scan
                selesai');
        },
        (err) => {
            this.popup.showAlert('ERROR', err);
        }
    );
}

hapus(person) {
    this.data.setDeleting(person.NRP).then(
        () => {
            this.popup.showAlert('Tunggu Sejenak', 'sistem proses scan
                selesai');
        },
        (err) => {
            this.popup.showAlert('ERROR', err);
        }
    );
}

showDetailMhs(mhs: Mahasiswa) {
    const defKehadiran = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
    // tslint:disable-next-line: max-line-length
    const defWktHadir = ['unset', 'unset', 'unset', 'unset', 'unset', 'unset',
        'unset', 'unset', 'unset', 'unset', 'unset', 'unset', 'unset'];
    mhs.matkul.forEach(x => {
        x.kehadiran = defKehadiran;
        x.waktu_hadir = defWktHadir;
    });
}

```

```

        console.log(mhs.matkul);
        this.data.resetKehadiran(mhs.NRP, mhs.matkul).then(
            () => {
                this.popup.showAlert('SUCCESS', 'udah gitu ae aowkowkow');
            },
            (err) => {
                this.popup.showAlert('ERROR', err);
            }
        );
    }

}

```

## F. Fitur Ubah Jadwal – Dosen

### 1. ubahjadwal.page.html

```

<ion-header>
    <ion-toolbar>
        <ion-title>Ubah Jadwal {{xMatkul.idMatkul}}</ion-title>
    </ion-toolbar>
</ion-header>

<ion-content>
    <div *ngIf="xMatkul">
        <ion-card>
            <div class="input">
                <ion-item>
                    <ion-label position="floating" color="light">Tanggal Sebelumnya<br/>(DD/MM/YYYY)</ion-label>
                    <ion-input [(ngModel)]="tglLama" (keyup)="showIsian(tglLama)"></ion-input>
                </ion-item>
                <ion-item *ngIf="!showIsi" lines="none" style="color: red;">
                    <ion-label>maaf, format tanggal salah atau tidak ada jadwal di tanggal ini</ion-label>
                </ion-item>
            <div *ngIf="showIsi">

```

```

<ion-item>
  <ion-label position="floating" color="light">Tanggal Baru
  (DD/MM/YYYY)</ion-label>
  <ion-input [(ngModel)]="tglBaru"
  (keyup)="checkDisability()"></ion-input>
</ion-item>
<ion-item>
  <ion-label position="floating" color="light">Jam Mulai
  (JJ.MM)</ion-label>
  <ion-input [(ngModel)]="mulai"
  (keyup)="checkDisability()"></ion-input>
</ion-item>
<ion-item>
  <ion-label position="floating" color="light">Jam Selesai
  (JJ.MM)</ion-label>
  <ion-input [(ngModel)]="selesai"
  (keyup)="checkDisability()"></ion-input>
</ion-item>
</div>
</div>

<ion-button [disabled]="btnDisable" expand="full" color="success"
  (click)="ubahJadwal()">
  <ion-label>Ubah Jadwal</ion-label>
</ion-button>
</ion-card>
</div>
</ion-content>

```

## 2. ubahjadwal.page.ts

```

import { Component, OnInit } from '@angular/core';
import { DataService } from 'src/app/services/data.service';
import { MataKuliah } from 'src/app/services/interface';
import { ModalController } from '@ionic/angular';
import { PopupService } from 'src/app/services/popup.service';

```

```
@Component({
```

```

    selector: 'app-ubahjadwal',
    templateUrl: './ubahjadwal.page.html',
    styleUrls: ['./ubahjadwal.page.scss'],
  })
export class UbahjadwalPage implements OnInit {

  btnDisable = true;
  showIsi = false;
  allMatkul: MataKuliah[];
  xMatkul: MataKuliah;
  idmat;
  task;
  hari: string;
  mulai: string;
  selesai: string;
  tglLama: string;
  tglBaru: string;

  constructor(
    private data: DataService,
    private modal: ModalController,
    private popup: PopupService
  ) { }

  ngOnInit() {
    if (this.tglLama && this.tglBaru && this.hari && this.mulai &&
      this.selesai) {
      this.btnDisable = false;
    }
    this.task = this.data.getAllMatkul().subscribe(res => {
      console.log('mk', res);
      this.allMatkul = res;
      this.allMatkul.forEach(item => {
        if (item.idMatkul === this.idmat) {
          this.xMatkul = item;
        }
      });
    });
  }
}

```

```

showIsian(tgl) {
    this.xMatkul.jadwal.forEach((x, i) => {
        if (x.tanggal === tgl) {
            this.showIsi = true;
            this.hari = this.xMatkul.jadwal[i].hari;
            this.mulai = this.xMatkul.jadwal[i].mulai;
            this.selesai = this.xMatkul.jadwal[i].selesai;
        }
    });
}

checkDisability() {
    if (this.tglLama && this.tglBaru && this.hari && this.mulai &&
        this.selesai) {
        this.btnDisable = false;
    }
}

ubahJadwal() {
    const allow = true;
    // this.allMatkul.forEach(mkul => {
    //     if (this.hari === mkul.jadwal.hari) {
    //         if (this.mulai === mkul.jadwal.mulai) {
    //             allow = false;
    //         }
    //     }
    // });
    if (allow) {
        let idx = 0;
        this.xMatkul.jadwal.forEach((x, i) => {
            if (x.tanggal === this.tglLama) {
                idx = i;
            }
        });
        this.data.updateJadwalbyID(this.xMatkul, this.tglBaru, this.hari,
            this.mulai, this.selesai, idx).then(
            (res) => {
                this.allMatkul = [];
            }
        );
    }
}

```

```

this.tglBaru = "";
this.tglLama = "";
this.hari = "";
this.idmat = "";
this.mulai = "";
this.selesai = "";
this.xMatkul = { } as MataKuliah;
this.popup.showAlert('SUCCESS', 'Ubah jadwal sukses').then(
  () => {
    this.modal.dismiss();
  }
);
},
(err) => {
  this.allMatkul = [];
  this.tglBaru = "";
  this.tglLama = "";
  this.hari = "";
  this.idmat = "";
  this.mulai = "";
  this.selesai = "";
  this.xMatkul = { } as MataKuliah;
  this.popup.showAlert('ERROR', err).then(
    () => { this.modal.dismiss(); }
  );
}
);
} else {
  this.popup.showAlert('GAGAL', 'maaf, mungkin jadwal yang dipilih
    sudah terisi atau jadwal tidak mengalami perubahan');
}
}
}

```

## G. Fitur Absen – Mahasiswa

### 1. absen.page.html

```
<ion-header>
<ion-toolbar>
  <ion-title>OASYS</ion-title>
</ion-toolbar>
</ion-header>

<ion-content>
<div *ngIf="!isloading">
  <ion-item lines="none" class="greeting">
    <ion-label>
      <span style="font-size: 28px;">HALAMAN      RECORD
          ABSEN</span>
    </ion-label>
  </ion-item>
  <ion-card>
    <div class="input">
      <ion-grid>
        <ion-row>
          <ion-col size="6">
            <ion-item>
              <ion-label position="floating" color="light">NRP</ion-label>
              <ion-input [(ngModel)]="nrp"></ion-input>
            </ion-item>
          </ion-col>
          <ion-col size="3">
            <ion-item>
              <ion-label position="floating" color="light">Hari</ion-label>
              <ion-input [(ngModel)]="hari"></ion-input>
            </ion-item>
          </ion-col>
          <ion-col size="3">
            <ion-item>
              <ion-label position="floating" color="light">Jam    Absen
(JJ.MM)</ion-label>
              <ion-input [(ngModel)]="waktuAbsen"></ion-input>
            </ion-item>
          </ion-col>
        </ion-row>
      </ion-grid>
    </div>
  </ion-card>
</div>
```

```

        </ion-item>
    </ion-col>
</ion-row>
</ion-grid>
<ion-button    expand="block"    slots="end"    color="success"
    (click)="absen()">
    <ion-label>Absen</ion-label>
</ion-button>
</div>
</ion-card>
<!-- <ion-card>
<ion-card-header>
    <ion-label>RECORD ABSEN</ion-label>
</ion-card-header>
<ion-card-content></ion-card-content>
</ion-card> -->
</div>

</ion-content>

```

## 2. absen.page.ts

```

import { Component, OnInit } from '@angular/core';
import { DataService } from 'src/app/services/data.service';
import { Mahasiswa } from 'src/app/services/interface';
import { PopupService } from 'src/app/services/popup.service';
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-absen',
  templateUrl: './absen.page.html',
  styleUrls: ['./absen.page.scss'],
})
export class AbsenPage implements OnInit {

  mhs: any[];
  allMhs: Mahasiswa[];
  nrp;

```

```

waktuAbsen: string;
hari;
task; task2; task3;
isLoading; isLoading2;

weekday = ['Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu'];
constructor(
    private data: DataService,
    private popup: PopupService,
    private http: HttpClient
) { }

ngOnInit() {
    this.isLoading = true;
    this.task3 = this.data.getAllMahasiswa().subscribe(res => {
        this.allMhs = res;
        // console.log(res);
    });
    this.task = this.data.getAllMahasiswa().subscribe(res => {
        this.mhs = res;
        // console.log('test');
        // console.log(this.mhs);
        this.mhs.forEach(person => {
            person.mk = [];
            person.matkul.forEach((item, index) => {
                this.task = this.data.getMataKuliah(item.idMatkul).subscribe(res2
                    => {
                        person.mk[index] = res2;
                        person.mk[index].expand = false;
                        person.mk[index].kehadiran = person.matkul[index].kehadiran;
                        person.mk[index].waktu_hadir = person.matkul[index].waktu_hadir;
                        this.isLoading = false;
                    });
            });
        });
    });
}

```

```
absen() {
let xMhs: any;
let xMk: any;
let x: number;
console.log(this.nrp, this.hari, this.waktuAbsen);
// const today = this.getDateToday();
this.mhs.forEach(person => {
if (person.NRP === this.nrp) {
xMhs = person;
person.mk.forEach(item => {
if (item.jadwal.hari === this.hari) { // forDummyDate
x = this.hitungKeterlambatan(item.jadwal.mulai);
if ( x <= 60 && x >= -60) {
xMk = item;
// console.log(item);
}
});
}
});
// console.log(this.allMhs);
this.allMhs.forEach(person => {
if (person.NRP === this.nrp) {
person.matkul.forEach(item => {
// console.log(today);
if (item.idMatkul === xMk.idMatkul ) {
if (item.kehadiran.length) {
item.waktu_hadir.push(this.waktuAbsen);
if ( x <= 30) {
item.kehadiran.push(100);
}
} else {
item.waktu_hadir[0] = this.waktuAbsen;
if ( x <= 30) {
item.kehadiran[0] = 100;
}
}
}
});
}
});
```

```

this.data.updateMahasiswaByID(person).then(
() => {
  this.popup.showAlert('SUCCESS', `Absen      atasnama
${person.nama} untuk matkul ${xMk.nama} berhasil`);
  this.nrp = null;
  this.hari = null;
  this.waktuAbsen = null;
},
(err) => {
  this.popup.showAlert('GAGAL', err);
}
);
}
);
}

hitungKeterlambatan(jadwal: string) {
  const jamJ = parseInt(jadwal.split('.')[0], 10);
  const menitJ = parseInt(jadwal.split('.')[1], 10);
  const jamA = parseInt(this.waktuAbsen.split('.')[0], 10);
  const menitA = parseInt(this.waktuAbsen.split('.')[1], 10);
  return ((jamA - jamJ) * 60) + (menitA - menitJ);
}

getDateToday() {
  const d = new Date();
  return {
    hari: this.weekday[d.getDay()],
    jam: d.getHours(),
    menit: d.getMinutes()
  };
}
}

```

## LAMPIRAN C PROPOSAL TA

Departemen Teknik Elektro  
Fakultas Teknologi Elektro – ITS

### EE184801 TUGAS AKHIR – 6 SKS

Nama Mahasiswa : Ismail Ashari  
Nomor Pokok : 0711164000020  
Bidang Studi : Telekomunikasi Multimedia (B-301)  
Tugas Diberikan : Semester Genap Tahun Ajaran 2019/2020  
Dosen Pembimbing : Sri Rahayu, S.T., M.Kom.  
Judul Tugas Akhir : **Rancang Bangun Sistem Absensi Perkuliahan Berbasis Internet of Things**

#### Uraian Singkat:

Sistem pendidikan yang diterapkan di perguruan tinggi sejak dulu hingga saat ini membutuhkan tatacara. Karena sering dilakukan dengan peraturan akademik yang harus mengikuti perkuliahan minimum 80% untuk dapat mengikuti ujian akhir semester, maka untuk mengecek kehadiran dalam perkuliahan dilakukan proses absensi. Hanya saja mayoritas masih banyak menggunakan kertas. Dengan tujuan untuk mengurangi penggunaan kertas, seiring dengan perkembangan zaman solusinya adalah dengan memanfaatkan penggunaan teknologi. Salah satu teknologi yang dimaksud adalah dengan berbasis *Internet of Things* (IoT). Teknologi berbasis IoT ini menggunakan sensor sidik jari (*fingerprint*) sebagai data input/masukan. Teknologi berbasis IoT ini mempunyai beberapa keunggulan diantaranya *paperless*, data dapat terdokumentasikan dengan baik, selain itu dengan data input berupa *fingerprint*, maka data tidak akan mudah untuk dipalsukan.

Dalam tugas akhir ini akan dibuat sistem absensi berbasis teknologi IoT dengan input *fingerprint*, yang dalam pengolahan data inputnya menggunakan mikrokontroler. Untuk menghubungkan ke database server (cloud) melalui internet digunakan modul IoT. Selanjutnya data akan diproses sedemikian rupa hingga dapat ditampilkan secara visual dalam suatu aplikasi *user interface* sehingga dapat diakses dengan internet darimana saja. Untuk merealisasikan hal tersebut, maka dalam tugas akhir ini akan dibuat tahapan rancangan sistem absensi, merealisasikan hardware dan software sistem serta mengintegrasikan hardware dan software sehingga dapat membentuk sistem absensi berbasis IoT.

Kata kunci: *Internet of Things*, mikrokontroler, *cloud*, *fingerprint*, database, *dashboard*.

Dosen Pembimbing

  
Sri Rahayu, S.T., M.Kom.  
NIP : 196802281997022001

Mengetahui,  
Ketua Program Studi S1

Dedet C. Riwaw, S.T., M.Eng., Ph.D.  
NIP : 197311192000031001

Menyetujui,  
Kepala Lab. Jurusan Telekomunikasi

  
Dr. Ir. Achmad Affandi, DEA.  
NIP : 196510141990021001



## BIODATA PENULIS



**Ismail Ashari**, dilahirkan di salah satu rumah sakit umum daerah Kota Bengawan, Surakarta Hadiningrat. Namun secara sah tertulis dilahirkan di Sukoharjo pada Hari Kamis Legi, 1 April 1999, dari orangtua Drs. Sukirno, MM dan Dwi Astuti, SE. Penulis menempuh pendidikan pertamanya di MI Negeri Jetis Sukoharjo selama 6 tahun, dan lulus pada tahun 2011 silam. Melanjutkan pendidikan tingkat menengah pertamanya di SMP Al-Islam 1 Surakarta selama 3 tahun, lalu menyelesaikan pendidikan terakhirnya di SMA Negeri 3 Surakarta yang berhasil ditempuh selama 2 tahun dan dinyatakan lulus pada tahun 2016. Saat ini penulis sedang menyelesaikan pendidikan sarjana di salah satu perguruan tinggi negeri terbaik di Indonesia, yaitu Institut Teknologi Sepuluh Nopember Surabaya dengan mengambil Bidang Studi Telekomunikasi Multimedia pada Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas.