

## **BAB II**

### **DASAR TEORI**

#### **2.1 Kajian Pustaka**

Penelitian yang dilakukan oleh Wahyu Purnama, Rista, dengan judul “*Monitoring* Pencahayaan Baterai Dan Lampu Penerangan Jalan Umum (Pju) Dengan Sistem Informasi Telegram Berbasis Mikrokontroler” Perkembangan teknologi secara otomatis telah sangat dirasakan dalam berbagai aspek kehidupan manusia. Banyak keunggulan yang ditawarkan oleh kemajuan teknologi seperti kemudahan dan keamanan yang berdampak positif, termasuk dalam bidang energi listrik. Sarana jalan yang diberikan pemerintah untuk meningkatkan atau mengoptimalkan fasilitas perlengkapan jalan guna mewujudkan keamanan, ketertiban dan kelancaran dalam berlalu lintas adalah Penerangan Jalan Umum. Lampu yang sering mati dan pengawasan yang kurang mengakibatkan ketidaknyamanan pengguna jalan umum oleh karena hal itu maka teknologi *monitoring* menggunakan pencahayaan batrei dalam lampu penerangan jalan umum dengan system informasi telegram berbasis mikrokontroler ini di rancang untuk membantu memonitoring ketika ada masalah dengan baterai dan ketika lampu mati, melalui sensor (LUX) sebagai yang saklar otomatis yang dapat menyalakan lampu pada malam hari serta mematikan lampu pada saat siang hari. Teknologi ini menngunakan sensor (PIR) sebagai saklar otomatis yang mendeteksi kendaraan untuk menyalakan lampu secara otomatis redup atau menyala. Pada saat tidak ada kendaraan yang melintasi lampu otomatis redup dan akan menyala terang ketika ada kendaraan yang terdeteksi dalam jarak 8 meter. Teknologi ini juga akan menginformasikan kepada teknisi melalui telegram apabila terjadi kerusakan lampu dan baterai pada jalan tersebut. Upaya ini dilakukan untuk memudahkan (efisiensi), mewujudkan keselamatan transportasi dan kenyamanan bagi masyarakat serta mengembangkan sistem teknologi menjadi lebih inovatif. [3].

Penelitian yang dilakukan oleh Avanbe dan Sutiknopo dengan judul “Rancang Bangun Smart Kontrol Lampu Penerangan Jalan Umum Berbasis SMS Gateway” membahas mengenai perancangan smart kontrol lampu penerangan jalan umum dengan SMS gateway dan alat smart kontrol lampu jalan bekerja secara

otomatis dan dapat membaca arus dan tegangan listrik lampu. Tujuan penelitian ini untuk mengetahui smart kontrol lampu penerangan jalan umum dengan SMS gateway dan smart kontrol lampu jalan bekerja secara otomatis dan dapat membaca arus dan tegangan listrik lampu. Dalam penelitian ini menggunakan metode percobaan (*experiment*) dengan melakukan pengujian, mengukur dan mendata serta beberapa percobaan dengan sensor LDR dan SMS gateway, prinsip kerja untuk mendeteksi tegangan, arus dengan mikrokontroler ATMEGA328 dan sensor-sensor. Komponen yang digunakan: mikrokontroler ATMEGA328, *transformator* 2amper, IC *regulator*, sensor cahaya, relay, sensor arus ACS712, sensor tegangan ZMPT101b, lampu LED, GSM SIM 800L V.2 dan SMS gateway, baterai 12Volt. Hasil pengujian penelitian ini untuk mengetahui seberapa efisien, supaya alat kontrol jarak jauh penerangan jalan dan sistem otomatis menyalakan dan mematikan lampu jalan, batas cahaya yang sudah ditentukan dan mengetahui seberapa arus dan tegangan listriknya. Pengiriman sinyal diterima modul SIM 800L V.2. Jika sinyal yang kurang baik, maka sulit untuk pengiriman data SMS ke handphone. Perlu adanya tambahan *interface* atau program berbasis website, *output* arus lebih dari 2Amper agar kinerja alat lebih maksimal dalam pengoperasian modul alat dan mode pencarian sinyal GSM ke handphone [4].

Penelitian yang dilakukan oleh Taufik Tauladan dan Fitri Latifah dengan judul “Sistem Kendali Otomatis Hemat Energi Pada Lampu Penerangan Jalan Tol Ldr, *Infrared* Dengan Mikrokontroler At89s52” membahas mengenai sistem kendali otomatis yang dapat dipasang di dalam peralatan elektronik saat ini, implementasi dari sistem kendali otomatis ini salah satunya diperuntukan bagi lampu penerangan jalan khususnya untuk jalan bebas hambatan. Lampu penerangan penerangan merupakan bagian yang tidak dapat dipisahkan dari pembangunan jalan bebas hambatan, tujuan dari penerangan adalah untuk menghasilkan kecepatan, keakuratan dan kenyamanan penglihatan di waktu malam hari, menjaga kualitas jarak pandang serta memudahkan bagi kendaraan yang melintas. Fungsi penerangan jalan bebas hambatan selain meningkatkan keselamatan dan kenyamanan pengendara, khususnya untuk mengantisipasi situasi perjalanan pada malam hari juga untuk keamanan lingkungan atau mencegah kriminalitas serta untuk memberikan kenyamanan dan keindahan lingkungan jalan. Dari hasil penelitian

yang dilakukan, Alat ini merupakan suatu inovasi baru bagi pengembangan ilmu teknologi dalam dunia usaha, Alat ini berfungsi untuk merubah sistem manual dengan cara pemasangan sensor menjadi sistem otomatis disisi bagian jalan tol, Alat ini memiliki manfaat untuk memudahkan serta mempercepat pengguna pemakaian listrik oleh pengelola jalan bebas hambatan guna mengontrol penggunaan [5].

Penelitian yang dilakukan oleh Dadan Somadani dan Ade Heri Ginanjar tentang judul “Prototipe Penerangan Jalan Umum (PJU) Pintar Berbasis Arduino Menggunakan Solar Panel, Sensor HC-SR04 Dan Sensor LDR”. Penerangan Jalan Umum ini menggunakan panel surya dari sumber cahaya yang berasal dari panas matahari kemudian diubah menjadi energi listrik yang disimpan pada baterai. Penggunaan energi listrik menjadi lebih hemat karena menggunakan panel surya sebagai sumber energi. Ketika kondisi malam, maka lampu akan menyala terang jika sensor ultrasonic mendeteksi adanya benda, namun ketika tidak terdeteksi adanya benda yang melewati maka lampu akan menyala redup. Proses iterative yang mengubah requirement diubah ke dalam sistem yang bekerja secara bertahap lalu diperbaiki oleh *user* dan analisis disebut dengan metode *prototyping*. Beberapa tahapan dalam metodologi *prototype* diantaranya analisis kebutuhan sistem, desain sistem, pengujian sistem serta implementasi sistem. Penerangan jalan umum pintar dalam bentuk prototipe akan membantu mengukur penerangan secara otomatis mulai dari tingkat redup atau terangnya berdasarkan situasi dan kondisi jalan. Penelitian ini menjelaskan *Prototype* penerangan jalan umum pintar menggunakan panel surya (solar panel) sebagai energi listrik. Solar panel menyimpan aliran listrik ke dalam baterai sebagai penghidup lampu penerangan jalan umum. Penggunaan Sensor HC-SR04 dan sensor LDR (*Light Dependent Resistor*) dihubungkan dengan bahasa pemrograman C sebagai integrasi dengan mikrokontroler Arduino UNO R3 untuk mendeteksi intensitas cahaya dan jarak benda terhadap sensor HC-SR04. Parameter pengaturan lampu LED berdasarkan nilai dari sensor LDR dan HC-SR04 [1].

Penelitian yang dilakukan oleh Zulfian Azmi, Saiful Nur Arif dan Elvis. T. Pasaribu dengan judul “Implementasi Kecerdasan Buatan Untuk Sistem Kendali Lampu Jalan Berbasis Mikrokontroler” membahas tentang Sistem penerangan jalan

dapat lebih efisien dengan dengan mengurangi rugi daya akibat penggunaan dari kendali lampu pintar penerangan jalan untuk deteksi intensitas cahaya, jarak dan jarak pandang objek atau kendaraan khususnya pada malam hari. Sistem ini menggunakan sensor LDR, ultrasonik, dan LED bekerja secara otomatis untuk mendeteksi intensitas cahaya dan objek. Sistem Penerangan Lampu Jalan menggunakan metode fuzzy logic berbasis mikrokontroler untuk mendeteksi intensitas cahaya dan jarak objek atau kendaraan . Sistem ini dapat mendeteksi intensitas cahaya objek dengan jarak yang sudah ditentukan dan lampu akan secara otomatis hidup tanpa adanya *operator*. Kesimpulan dari penelitian ini adalah sistem lampu penerangan jalan Cerdas menggunakan metode fuzzy berbasis mikrokontroler dapat dirancang dengan mengkombinasikan sensor LDR dan sensor ultrasonik yang digunakan untuk mendeteksi Intensitas cahaya dan objek/kendaraan sebagai *unit input* dan LED dengan cahaya sebagai *unit output*. Sensor LDR pada penerangan lampu pintar penerangan jalan untuk mendeteksi intensitas cahaya, dimana lampu akan meyal redup pada malam hari atau tidak ada cahaya matahari. Sedangkan sensor LDR untuk mendeteksi objek atau kendaraan yang melintasi jalan, sehingga ketika sensor mendeteksi objek/kendaraan maka lampu akan menyala sangat terang. Metode fuzzy logic diimplementasikan pada penerangan lampu jalan yang dapat mendeteksi Intensitas cahaya dan objek atau kendaraan untuk mengatur kecerahan cahaya pada LED [6].

Tabel 2. 1 Kajian Pustaka

<b>Nama</b>	<b>Judul</b>	<b>Kesimpulan</b>
Wahyu Purnama dan Rista	“ <i>Monitoring</i> Pencahayaan Baterai dan Lampu Penerangan Jalan Umum (PJU) Dengan Sistem Informasi Telegram Berbasis Mikrokontroler”	Penelitian ini dilakukan untuk membantu memonitoring ketika ada masalah pada baterai dan lampu yang bertujuan untuk meningkatkan kenyamanan, efisiensi, keselamatan transportasi dan juga pengembangan sistem teknologi agar lebih inovatif.
Avanbe dan Sutiknopu	“Rancang Bangun Smart Kontrol Lampu Penerangan Jalan Umum Berbasis SMS Gateway”	Dalam penelitian ini menggunakan metode percobaan ( <i>experiment</i> ) dengan melakukan

		<p>pengujian, mengukur dan mendata serta beberapa percobaan dengan sensor LDR dan SMS gateway, prinsip kerja untuk mendeteksi tegangan, arus dengan mikrokontroler ATMEGA328 dan sensor-sensor.</p>
<p>Taufik Tauladan dan Fitri Latifah</p>	<p>“Sistem Kendali Otomatis Hemat Energi Pada Lampu Penerangan Jalan Tol LDR, <i>Infrared</i> Dengan Mikrokontroler At89s52”</p>	<p>Alat ini berfungsi untuk merubah sistem manual dengan cara pemasangan sensor menjadi sistem otomatis disisi bagian jalan tol, tujuan dari penerangan adalah untuk menghasilkan kecepatan, keakuratan dan kenyamanan penglihatan di waktu malam hari, menjaga kualitas jarak pandang serta memudahkan bagi kendaraan yang melintas.</p>
<p>Dadan Somadani dan Ade Heri Ginanjar</p>	<p>“Prototipe Penerangan Jalan Umum (PJU) Pintar Berbasis Arduino Menggunakan Solar Panel, Sensor HC-SR04 Dan Sensor LDR”</p>	<p>Penerangan jalan umum pintar dalam metode prototipe akan membantu mengukur penerangan secara otomatis mulai dari tingkat redup atau terangnya berdasarkan situasi dan kondisi jalan</p>
<p>Zulfian Azmi, Saiful Nur Arif and Elvis. T.</p>	<p>“Implementasi Kecerdasan Buatan Untuk Sistem Kendali Lampu Jalan Berbasis Mikrokontroler”</p>	<p>Kesimpulan dari penelitian ini adalah sistem lampu penerangan jalan Cerdas menggunakan metode fuzzy berbasis mikrokontroler dapat dirancang dengan mengkombinasikan sensor LDR dan sensor ultrasonik yang digunakan untuk mendeteksi Intesitas cahaya dan objek/kendaraan sebagai <i>unit input</i> dan LED dengan cahaya sebagai <i>unit output</i></p>

Berdasarkan studi literatur penelitian sebelumnya diketahui jika adanya perbedaan dengan penelitian yang dilakukan saat ini. Berikut beberapa perbedaan penelitian “Rancang Bangun Smart Street *Lighting* Berbasis *Internet of Things* (IoT)” dengan penelitian terkait sebelumnya.

1. Menggunakan IC mikrokontroler ESP32 memiliki *Central Prosessing Unit* (CPU) *Xtensa dual-core*, yang telah dilengkapi dengan Wi-Fi dan Bluetooth.
2. Menggunakan 3 jenis sensor yaitu sensor LDR (*Light Dependent Resistor*), sensor *Infrared transmitter* dan sensor *Infrared Receiver*.

## 2.2 Dasar Teori

### 2.2.1 Lampu Penerangan Jalan

Lampu jalan atau dikenal juga sebagai Penerangan Jalan Umum (PJU) merupakan lampu yang digunakan untuk penerangan jalan di malam hari sehingga mempermudah pengguna jalan melihat dengan lebih jelas jalan yang akan dilalui pada malam hari, sehingga dapat meningkatkan keselamatan lalu lintas dan keamanan.

Fungsi utama lampu penerangan jalan umum (PJU) adalah memberikan pencahayaan buatan bagi pengguna jalan sehingga mereka merasa aman dalam melakukan aktivitas perjalanan di malam hari.

Lampu Penerangan Jalan Umum merupakan barang-barang elektronik yang rentan atau dapat dikatakan memiliki umur pakai yang pendek, sehingga kegiatan perbaikan dan pemeliharaan mutlak dibutuhkan. Perbaikan dapat meliputi perbaikan jaringan, penggantian lampu yang mati, atau pun pengecekan kondisi PJU [2].



Gambar 2. 1 Penerangan Lampu Jalan

### 2.3 Mikrokontroler

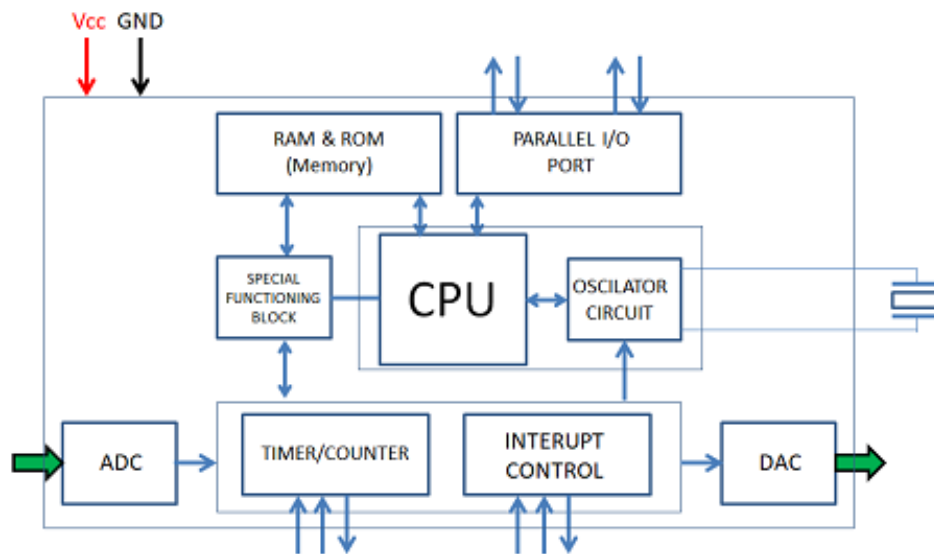
Mikrokontroler adalah sebuah computer kecil yang dikemas dalam bentuk chip *IC (Integrated Circuit)* dan dirancang untuk melakukan tugas sebagai pengontrol rangkaian elektronik. Pada dasarnya, sebuah *IC* mikrokontroler terdiri dari satu atau lebih inti yaitu *Central Processing Unit (CPU)*, memori, I/O tertentu dan *unit* pendukung seperti *Analog Digital Converter (ADC)* yang sudah terintegrasi Dan dapat menyimpan program didalamnya.

Mikrokontroler juga merupakan pengontrol utama standart industri dan riset saat ini, dikarenakan berbagai kelebihan yang dimiliki antara lain murah, dukungan *software* dan dokumentasi yang memadai, dan memerlukan komponen pendukung yang sangat sedikit. Mikrokontroler juga tidak hanya memiliki 1 tipe saja, karena berbagai produsen mikrokontroler juga menciptakan berbagai macam mikrokontroler dengan keunggulannya masing-masing seperti:

1. Intel, contoh: 8031, 89C51 dll
2. Zilog, contoh: Z8, Z8F1680 dll
3. Microchip, contoh: PIC16F84, PIC16F877 dll
4. Motorola, contoh: 68HC11, MC68HC705V12CFN dll
5. Philips Semiconductors, contoh: LPC2000, LPC900, LPC700dll
6. Atmel, contoh: Atmel AT91 series (ARM THUMB architecture), AT90, Tiny & Mega series – AVR (Atmel Norway design), Atmel AT89 series (Intel 8051/MCS51 architecture) [7].

### 2.3.1 Struktur Mikrokontroler

Struktur umum mikrokontroler adalah sebagai berikut:



Gambar 2. 2 Struktur Mikrokontroler

Fungsi setiap bagian :

#### 1. *Central Processing Unit (CPU)*

*Central Processing Unit (CPU)* merupakan pusat mikrokontroler atau biasa dikenal sebagai otak dari mikrokontroler. Tugas CPU adalah mengambil instruksi (*fetch*) lalu menerjemahkan instruksi (*decode*), hingga dieksekusi (*execute*). Menghubungkan bagian mikrokontroler yang satu dengan sistem lainnya merupakan bagian tugas dari CPU. Proses pengambilan dan mendekode instruksi yang di dapat dari memori program akan diterjemahkan (*decode*) dilakukan oleh CPU.

#### 2. *Memori Penyimpanan*

Mikrokontroler memiliki fungsi yang sama dengan mikrprosesor sebagai memori atau media penyimpanan. Mikrokontroler memiliki berbagai media penyimpanan seperti RAM, ROM (EPROM, EEPROM dan lainnya) sebagai media penyimpanan data serta *memory flash* yang diperuntukan sebagai media penyimpanan *source code* program.

#### 3. *Port INPUT / OUTPUT paralel*



Port paralel digunakan untuk menghubungkan perangkat yang satu dengan yang lainnya dengan mikrokontroler seperti LCD, printer, LED, memori serta perangkat lain sebagai *INPUT/OUTPUT* ke mikrokontroler.

#### **4. Port Serial (Serial Port)**

*Serial Port* atau *Port serial* memberikan berbagai penyediaan untuk antarmuka *serial* antara mikrokontroler dan *periferal* lain seperti *port paralel*.

#### **5. Pengatur Waktu dan Penghitung (Timer dan Counter)**

Pengatur dan penghitung waktu (*Timer dan Counter*) dimiliki oleh mikrokontroler sebagai salah satu fungsi. Fungsi ini sangat membantu karena mikrokontroler mungkin memiliki pengatur dan penghitung waktu lebih dari satu. Dalam pengaturan dan perhitungan waktu mikrokontroler terdapat beberapa operasi utama yang dapat dilakukan seperti modulasi, pembangkit pulsa, pengukuran jam dan frekuensi, osilasi serta dapat digunakan sebagai alat bantu perhitungan pulsa eksternal.

#### **6. Analog to Digital Converter atau Pengonversi Analog ke Digital (ADC)**

Fungsi sebagai konverter ADC ini bertujuan untuk membantu mengubah sinyal *analog* ke dalam bentuk *digital*. Sinyal *input* harus dalam bentuk *analog* yang dihasilkan dari output sensor kemudian akan diubah menjadi bentuk *digital* sehingga dapat digunakan dalam berbagai aplikasi *digital* seperti halnya layar *digital* untuk perangkat pengukuran.

#### **7. Digital to Analog Converter atau Pengonversi Digital ke Analog (DAC)**

Konversi DAC melakukan hal kebalikan dari fungsi konverter ADC yang akan mengubah sinyal *digital* menjadi format sinyal dalam bentuk *analog*. Hal ini dilakukan bertujuan untuk dapat mengendalikan perangkat-perangkat *analog* yang digunakan seperti motor DC dan lainnya.

#### **8. Kontrol Interupsi (Interrupt Control)**

*Interrupt Control* atau kontrol interupsi merupakan fungsi yang digunakan sebagai penundaan (*interrupt control*) terhadap program kerja. Penundaan atau *interrupt* dapat dilakukan dengan *internal* dan *eksternal*. *Internal* dilakukan dengan menggunakan instruksi interupsi selama pemrograman dan untuk interupsi secara *eksternal* dilakukan dengan mengaktifkan pin *interrupt* yang ada.

#### **9. Blok Fungsi Khusus (Special Functioning Block)**

*Special Functioning Block* atau blok fungsi khusus hanya dimiliki oleh beberapa mikrokontroler yang khusus diperuntukan untuk beberapa aplikasi tertentu seperti aplikasi sistem robotik dan yang lainnya. [8].

### 2.3.2 Cara Kerja Mikrokontroler

Prinsip kerja mikrokontroler adalah sebagai berikut :

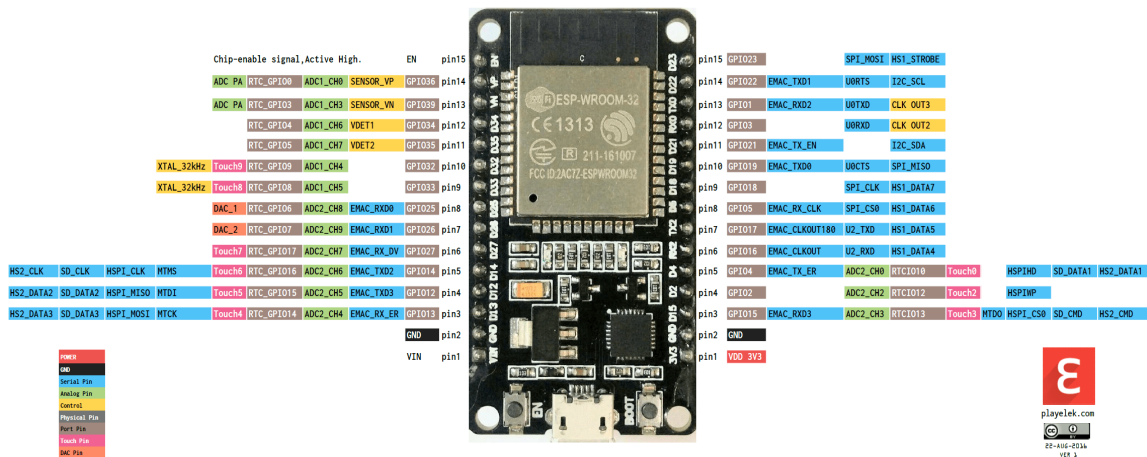
1. Berdasarkan nilai yang berada pada register Program Counter, mikrokontroler mengambil data pada read only *memory* (ROM) dengan alamat sebagaimana yang tertera pada register Program Counter. Selanjutnya isi dari register Program Counter ditambah dengan satu (Increment) secara otomatis. Data yang diambil pada read only *memory* (ROM) merupakan urutan instruksi program yang telah dibuat dan diisikan sebelumnya oleh pengguna.
2. Instruksi yang diambil tersebut diolah dan dijalankan oleh mikrokontroler. Proses pengerjaan bergantung pada jenis instruksi, bisa membaca, mengubah nilai-nilai pada *register*, RAM, isi *Port*, atau melakukan pembacaan dan dilanjutkan dengan perubahan data.
3. Program Counter telah berubah nilainya (baik karena penambahan otomatis pada langkah 1, atau karena perubahan-perubahan pada langkah 2). Selanjutnya yang dilakukan oleh mikrokontroler adalah mengulang kembali siklus ini pada langkah 1. Demikian seterusnya hingga *power* dimatikan [7].

### 2.4 Mikrokontroler ESP32

ESP32 adalah sebuah mikrokontroler yang dikenalkan oleh Espressif System penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi *Internet of Things*. terlihat pada gambar di bawah merupakan

pin out dari ESP32. Pin tersebut dapat dijadikan *input* atau *output* untuk menyalakan LCD, lampu, bahkan untuk menggerakkan motor DC.

## DOIT ESP32 DEVKIT V1 PINOUT



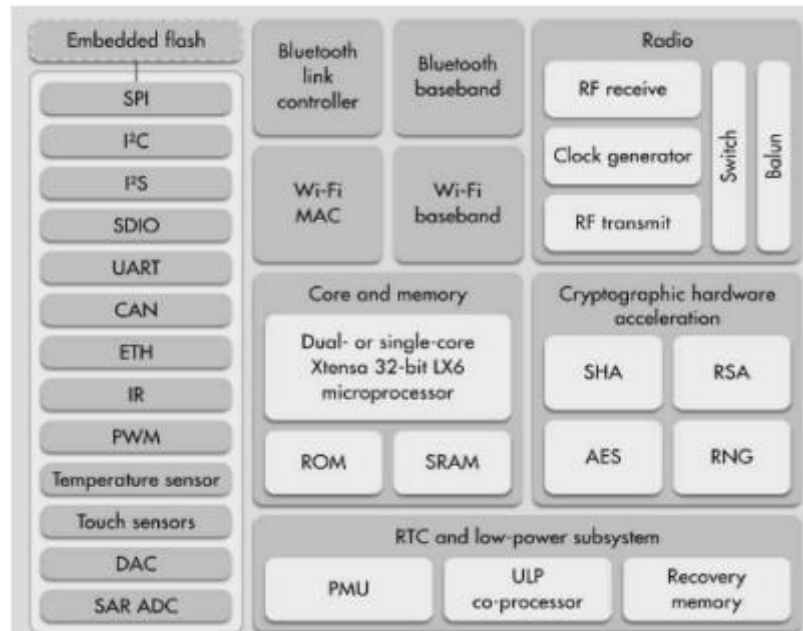
Gambar 2. 3 Mikrokontroler ESP32

Pada pin out tersebut terdiri dari :

1. 18 ADC (*Analog Digital Converter*, berfungsi untuk merubah sinyal *analog* ke *digital*)
2. 2 DAC (*Digital Analog Converter*, kebalikan dari ADC)
3. 16 PWM (*Pulse Width Modulation*)
4. 10 Sensor sentuh
5. 2 jalur antarmuka UART
6. Pin antarmuka I2C, I2S, dan SPI [9].

### 2.4.1 Block diagram Mikrokontroler ESP32

Setiap mikrokontroler memiliki spesifik yang berbeda. Salah satu jenis mikrokontroler saat ini adalah ESP32. Mikrokontroler ini memiliki beberapa komponen dan fungsi yang dapat diketahui melalui gambar berikut.



Gambar 2. 4 Blok Diagram Mikrokontroler ESP32

Mikrokontroler ESP32 memiliki beberapa fungsi seperti IR, I2C, Bluetooth, FR, WiFi, PWM, dan lain-lain. Fungsi-fungsi yang tersedia pada mikrokontroler sangat mendukung kebutuhan IoT saat ini. Jenis prosesor yang digunakan untuk kebanyakan tipe mikrokontroler ESP32 adalah dual-core, namun untuk tipe ESP32-S0WD hanya menggunakan jenis prosesor single-core saja. Frekuensi clock yang digunakan pada ESP32 tergolong tinggi hingga dapat mencapai 240MHz berbeda dengan mikrokontroler ATMEGA8 yang hanya mencapai 12 MHz. Kelebihan lainnya dari prosesor ESP32 sudah memiliki modul WiFi dan Bluetooth yang sangat jarang dimiliki oleh mikrokontroler lainnya. Bluetooth yang tersedia pada mikrokontroler ini merupakan bluetooth yang dapat menghemat daya karena berdaya rendah yang biasa dikenal dengan nama *Bluetooth Low Energy* (BLE) dan bluetooth ini mampu melakukan transfer data tertentu dalam *sleep mode*.

Mikrokontroler ESP32 memiliki memori internal dan dapat juga diekstensi menggunakan external flash dan SRAM. Penggunaan memori internal pada ESP 32 terdiri dari 3 memori yang terdiri dari ROM yang digunakan dalam proses booting

dan besarnya adalah 448 Kb, SRAM digunakan sebagai tempat penyimpanan data serta instruksi yang memiliki ukuran sebesar 520 KB dan Embedded flash yang terhubung via SC\_CMD, IO16, IO17, SD\_DATA\_0 dan SD\_DATA\_1. Besaran memori ini sekitar 2-4 MB yang dapat memperbesar kapasitas penyimpanan data. Mikrokontroler ESP32 juga dapat diekstensi menggunakan external flash dan SRAM [10].

Tegangan operasi pada ESP32 adalah maksimal 3.3V sehingga *suplay* listrik pada rangkaian diharapkan tidak melebihi tegangan maksimal tersebut. ESP32 memiliki beberapa perbedaan dengan mikrokontroler lainnya seperti mikrokontroler Atmega pada Arduino Uno. Berikut ini perbandingan beberapa jenis mikrontroller ESP32 dengan mikrokontroler yang lainnya [11].

Tabel 2. 2 Perbandingan Jenis Mikrokontroler

	<b>Arduino Uno</b>	<b>Node MCU (ESP8266)</b>	<b>ESP32</b>
<b>Tegangan</b>	5 Volt	3.3 Volt	3.3 Volt
<b>CPU</b>	Atmega328-16Mhz	<i>Xtensa single core L106-60 Mhz</i>	<i>Xtensa dual core LX6-160MHz</i>
<b>Asitektur</b>	8 bit	32 bit	32 bit
<b>Flash Memory</b>	32 Kb	16 Mb	16 Mb
<b>SRAM</b>	2 Kb	160 Kb	512 Kb
<b>GPIO Pin (ADC/DAC)</b>	14 (6/-)	17 (1/-)	36 (18/-2)
<b>Bluetooth</b>	Tidak ada	Tidak ada	Ada
<b>WiFi</b>	Tidak ada	Ada	Ada
<b>SPI/I2C/UART</b>	1/1/1	2/1/2	4/2/2

## 2.5 Light Emitting Diode (LED)

LED (*Light Emitting Diode*) adalah *diode* yang digunakan untuk dapat memancarkan cahaya dengan tegangan catu minimal 1,8 V (sebuah baterai 1,5V tidak bisa digunakan untuk menghidupkan LED). Selain itu berfungsi sebagai penyearah arus. Pada Gambar 2.5 tampak simbol dan bentuk fisik LED. LED digunakan di berbagai rangkaian elektronika, sebagai indikator hingga lampu penerangan dengan efisiensi yang tinggi dan ketahanan jauh lebih baik

dibandingkan lampu bohlam maupun neon PL. LED juga bisa digunakan sebagai pembatas tegangan dari tegangan catu yang tidak terlalu besar (3V-5V), jika dicatu melebihi 5V akan merusak LED. *Diode* pancar cahaya (*Light-emiting diode=LED*) adalah *diode* semikonduktor khusus yang dirancang untuk memancarkan cahaya apabila arus melaluinya. Apabila diberi bias maju, energi elektron yang mengalir melewati tahanan sambungan diubah langsung menjadi energi cahaya. Karena LED adalah *diode*, maka harus dioperasikan di dalam ukuran kerja tegangan dan arus yan tertentu untuk mencegah kerusakan yang tidak dapat diubah lagi. Sebagian besar LED membutuhkan 1,5 sampai 2,2 V untuk member bias maju dan dapat mengatasi dengan aman arus sebesar 20 sampai 30 mA. LED biasanya dihubungkan seri dengan tahanan yang membatasi tegangan dan arus pada nilai yang di kehendaki [2].



Gambar 2. 5 Lampu LED

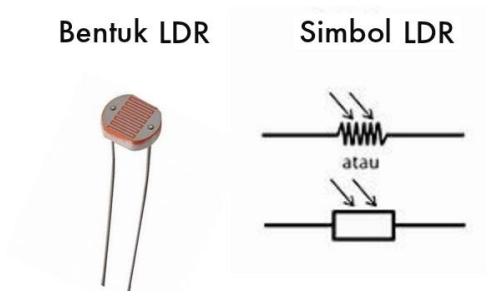
Spesifikasi Lampu LED dapat dilihat pada tabel 2.3 :

Tabel 2. 3 Spesifikasi Lampu Epistar LED

<b>Spesific Epistar LED</b>	<b>Minimal</b>	<b>Maximal</b>	<b>Unit</b>
<b>Power</b>	1	-	W
<b>DC Forward Current</b>	350	-	mA
<b>Storage Temperature</b>	-35	-85	°C
<b>PCB Material</b>	-	-	Aluminium
<b>LED Junction Temperature</b>	120°	-	°C
<b>Emitting Color</b>	-	-	Cool White
<b>Optical Attenuation (%)</b>	1	-	%
<b>Viewing Angel (°)</b>	140°	-	°
<b>Operating Temperature</b>	-35	-85	°C
<b>Color Temperature</b>	6000	7000	k
<b>Forward Voltage</b>	3.0	3.4	V

## 2.6 LDR (*Light Dependent Resistor*)

LDR merupakan salah satu jenis *resistor* yang disebut sebagai *fotoresistor*. Nilai hambatan LDR dipengaruhi oleh cahaya yang diterima dari lingkungan sekitar. Resistansi LDR dapat berubah-ubah tergantung pada intensitas cahaya yang diterima oleh LDR itu sendiri [12].



Gambar 2. 6 Bentuk Sensor *LDR* dan Simbol

Spesifikasi dari sensor LDR dapat dilihat pada tabel 2.4 :

Tabel 2. 4 Spesifikasi Sensor LDR

<i>Parameter</i>	<i>Conditions</i>	<i>Minimal</i>	<i>Type</i>	<i>Maximal</i>	<i>Unit</i>
<i>Cell Resistance</i>	1000 LUX	-	400	-	Ohm
	10 LUX	-	9	-	K Ohm
<i>Dark Resistance</i>	-	-	1	-	M Ohm
<i>Dark Capacitance</i>	-	-	3.5	-	pF
<i>Rise Time</i>	1000 LUX	-	2.8	-	ms
	10 LUX	-	18	-	ms
<i>Fall Time</i>	1000 LUX	-	48	-	ms
	10 LUX	-	120	-	ms
<i>Voltage AC/DC Peak</i>		-	-	320	V max
<i>Current</i>		-	-	75	mA max
<i>Power Dissipation</i>		-	-	100	mW max
<i>Operating Temperature</i>		-60	-	+75	Deg. C

## 2.7 Sensor *Infrared Transmitter* (IR)

Pemancar Inframerah adalah *light emitting diode* (LED) yang memancarkan radiasi inframerah yang disebut sebagai IR LED. Meskipun LED IR terlihat seperti LED biasa, radiasi yang dipancarkannya tidak terlihat oleh mata manusia [13].

Gambar LED dan Spesifikasi sensor Inframerah ditunjukkan di bawah ini.



Gambar 2. 7 Sensor *Infrared Transmitter (IR)*

Tabel 2. 5 Spesifikasi Sensor *Infrared Transmitter (IR)*

<b>Specifics Emitter</b>	<b>Minimal</b>	<b>Maximal</b>	<b>Unit</b>
<b>Reverse Voltage</b>	5	-	V
<b>Cont. Forward Current</b>	150	-	mA
<b>Forward Voltage</b>	1.3	1.7	V
<b>Wavelength</b>	950	-	nm

## 2.8 Sensor *Infrared Receiver (IR)*

Penerima inframerah atau sensor inframerah mendeteksi radiasi dari pemancar IR. Penerima IR datang dalam bentuk fotodiode dan *fototransistor*. Fotodiode Inframerah berbeda dari diode foto biasa karena hanya mendeteksi radiasi inframerah. Gambar di bawah ini menunjukkan gambar penerima IR atau fotodiode, Emitornya adalah LED IR dan detektornya adalah fotodiode IR. Fotodiode IR sensitif terhadap cahaya IR yang dipancarkan oleh LED IR. Resistansi foto-diode dan tegangan keluaran berubah sebanding dengan cahaya IR yang diterima. Ini adalah prinsip kerja yang mendasari sensor IR. [13]



Gambar 2. 8 Sensor *Infrared Receiver*



Spesifikasi dari sensor *Infrared receiver* dapat dilihat pada tabel 2.6 :

Tabel 2. 6 Spesifikasi Sensor *Infrared Receiver* (IR)

<b>Specifics Detector</b>	<b>Minimal</b>	<b>Maximal</b>	<b>Unit</b>
<b>VCEO (Collector to Emitter)</b>	70	-	V
<b>VECo (Emitter to Collector)</b>	5	-	V
<b>IC (Collector Current)</b>	50	-	mA
<b>Total Power Dissipation</b>	150	-	mW
<b>Peak Sensitivity Wavelength</b>	850	-	nm
<b>Spectral Bandwidth Length</b>	620	980	nm
<b>Angle of half sensitivity</b>	-20	+60	degree

## 2.9 Software Arduino Integrated Development Environment (IDE)

Salah satu pengembangan *prototype* yang paling banyak digunakan adalah Arduino. Arduino IDE bermula dari dikembangkannya *software* Processing yang diubah menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Arduino merupakan *physical computing* yang bersifat *open source*. Saat ini Arduino juga tidak hanya sebagai alat pengembangan namun sebagai kombinasi dari *hardware*, bahasa pemrograman serta *Integrated Development Environment* (IDE) yang canggih. Arduino IDE adalah *software* yang berperan untuk menulis, mengcompile program menjadi kode biner dan mampu meng-*upload* ke dalam *memory microcontroller*. Saat ini banyak projek dan alat-alat yang dikembangkan untuk dapat disambungkan dengan Arduino seperti sensor, tampilan, penggerak dan lainnya[14].

Bahasa pemrograman yang digunakan untuk membuat Arduino adalah java namun *software* ini juga diperlengkapi dengan *library C/C++* yang biasa disebut Wiring yang membuat operasi *input* dan *output* menjadi lebih mudah. Program yang ditulis pada Arduino *Software* (IDE) disebut sebagai *sketch*. *Sketch* yang telah ditulis dalam suatu editor teks dan akan disimpan dengan format ekstensi.ino. kelebihan *software* ini juga memiliki fitur-fitur seperti *cutting*, *paste*, *searching*, *replacing* yang akan membantu memudahkan dalam menulis kode program yang akan diinputkan[2].

IDE Arduino terdiri dari beberapa bagian, seperti:

1. Editor program

Window yang digunakan untuk menulis dan mengedit program dalam Bahasa *processing*.

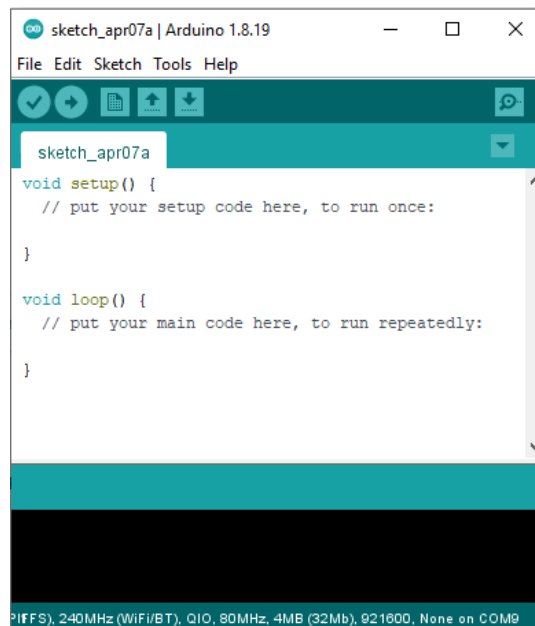
2. *Compiler*

Modul yang akan mengubah kode program (Bahasa *processing*) menjadi kode biner. sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. *Microcontroller* hanya memahami kode biner dan bukan Bahasa *processing*.

3. *Uploader*

Modul yang akan memuat kode biner dari computer ke dalam *memory* yang tersimpan pada papan Arduino.

Berikut tampilan *sketch* pada *software* Arduino[14].



```
sketch_apr07a | Arduino 1.8.19
File Edit Sketch Tools Help
sketch_apr07a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
PIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None on COM9
```

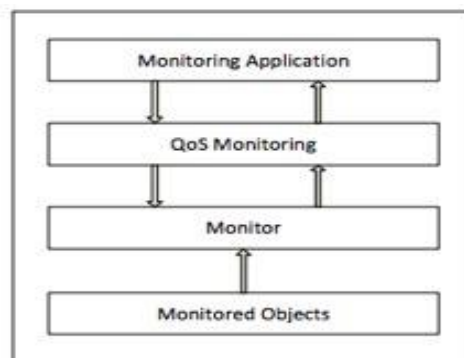
Gambar 2. 9 *Sketch* Program

## 2.10 QoS (*Quality of Services*)

*Quality of service* (QoS) adalah teknologi yang diperuntukan sebagai pengelola lalu lintas jaringan pada saat pengiriman (*transmitter*) dan penerimaan data (*receiver*) untuk mengukur *throughput*, mengurangi adanya kehilangan paket (*packet loss*), *delay* (*latency*) dan *jitter*. QoS akan mengontrol pengelolaan sumber daya jaringan untuk membuat prioritas tipe data khusus pada suatu jaringan. QoS sangat diperlukan dan memiliki arti yang besar bagi perusahaan karena termasuk penyedia layanan jaringan yang dapat menjamin kinerja yang tinggi seperti dalam *service-level agreement* (SLA). QoS dapat mengukur dan memprediksi jaringan dalam berbagai bentuk aplikasi seperti suara, data *real-time* dan video. QoS membantu mencegah adanya penundaan dalam pengiriman data (*delay*), *jitter* dan *packet loss* yang akan menyebabkan penurunan kualitas data. Penggunaan QoS dapat dilakukan menggunakan alat dan teknik khusus seperti adanya *traffic shapping* dan *jitter buffer*.

### 2.10.1 Model *monitoring* QoS

Pembagian model *monitoring* QoS yang dibedakan dalam 4 pembagian. Pembagian ini terbagi dalam mengumpulkan, merekam dan menganalisis layanan jaringan.



Gambar 2. 10 Model *Monitoring* QoS

Beberapa komponen dalam model *monitoring* Qos yaitu:

1. *Monitoring Application*

*Monitoring Application* adalah sebagai sistem antar muka (*interface*) pada *administrator* jaringan. *Monitor Application* akan mengambil data atau informasi dari *monitor* kemudian mendapatkan lalu lintas paket yang akan

dilewati, menganalisa data dan lanjut mengirimkan hasil analisis kepada *user* (pengguna).

## 2. QoS *Monitoring*

*QoS Monitoring* menyediakan cara kerja yang berdasarkan informasi dari keadaan lintas paket atau kondisi lalu lintas pada saat pengiriman dan penerimaan data dengan menerapkan nilai-nilai dari parameter QoS.

## 3. *Monitor*

Proses pengumpulan, pengukuran serta perekaman informasi dalam aliran paket data suatu lalu lintas merupakan bentuk kegiatan *monitor*. Hasil *monitor* yang dilakukan akan dilanjutkan ke komponen selanjutnya yaitu *monitoring application*.

## 4. *Monitored Objects*

5. Kumpulan informasi mengenai aktivitas dan atribut yang akan dimonitor pada suatu jaringan disebut *Monitored Objects*. Informasi dapat berbentuk aliran paket data secara waktu nyata, tipe aliran paket data berdasarkan alamat dalam hal ini alamat sebagai sumber (*source*) dan alamat sebagai tujuan (*destination*) yang dilihat di layer IP, informasi mengenai *port* apa yang sedang digunakan UDP atau TCP dan informasi parameter yang terdapat pada paket RTP[15].

### 2.10.2 Parameter Quality of Service (QoS)

Beberapa parameter *Quality of Service* yaitu:

#### 1. *Throughput*

*Throughput* merupakan pengukuran rate atau kecepatan dalam transfer data diukur dalam *bit per second* (bps). *Throughput* mampu menggambarkan penggunaan *bandwidth* pada jaringan untuk mengunduh *file* pada size atau ukuran tertentu. Pengukuran *Throughput* dihitung dengan mengamati jumlah total paket yang sampai atau yang sukses diamati kemudian dibagi dengan durasi interval waktu tertentu.

Berikut kategori *Indeks Throughput* dapat dilihat pada table 2.7 :

Tabel 2. 7 Kategori *Throughput*

Kategori <i>Throughput</i>	<i>Throughput</i>	<i>Indeks</i>
Sangat Baik	> 2,1 Mbps	4
Baik	1200 kbps – 2,1 Mbps	3
Cukup	700-1200 kbps	2
Kurang Baik	338-700 kbps	1
Buruk	0-338 kbps	0

Persamaan perhitungan *Throughput* :

$$\textit{Throughput} = \frac{\textit{Jumlah data yang dikirim (kb)}}{\textit{Waktu pengiriman data (s)}}$$

## 2. *Packet Loss*

*Packet Loss* adalah kegiatan pengukuran untuk mengetahui berapa jumlah paket (*persentase*) paket yang hilang pada saat pengiriman data. Parameter yang memperlihatkan mengenai persentase paket yang hilang diketahui karena adanya *congestion* dan *collision* dalam suatu jaringan. Faktor lain yang mengakibatkan adanya paket yang hilang (*packet loss*) adalah kesalahan perangkat yang digunakan, adanya penurunan sinyal pada jaringan, dan radiasi yang terdapat di lingkungan sekitar pada saat melakukan pengiriman data.

Berikut kategori *Indeks* Paket Loss dapat dilihat pada table 2.8 :

Tabel 2. 8 Kategori *Packet Loss*

Kategori Degradasi	<i>Packet Loss</i>	<i>Indeks</i>
Sangat Baik	0-2%	4
Baik	3-14%	3
Cukup	15-24%	2
Buruk	>25%	1

Persamaan perhitungan *Packet Loss*: [16]

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

## 3. *Delay (Latency)*

*Delay (Latency)* adalah ukuran waktu yang dibutuhkan untuk pengiriman sebuah paket data dari komputer asal dapat sampai menuju komputer tujuan data. *Delay* ditentukan oleh adanya antrian yang panjang, mengambil route lain untuk menghindari kemacetan pada *routing*, jarak data pengiriman menuju tempat tujuan, media fisik, *congesti* atau juga waktu proses yang lama [17]

Berikut kategori *Indeks Delay* dapat dilihat pada table 2.9 :

Tabel 2. 9 Kategori *Latency*

Kategori Latensi	Besar Delay	Indeks
Sangat Baik	<150 ms	4
Baik	150 – 300 ms	3
Cukup	300 – 450 ms	2
Buruk	>450 ms	1

Persamaan perhitungan *Delay (Latency)*:

$$\text{Rata - rata Delay} = \frac{\text{Total Delay}}{\text{Total Paket Yang Diterima}}$$

## 2.11 *Firestore*

*Firestore* merupakan *database* yang digunakan dalam mengolah data. *Firestore* merupakan database dengan basis NoSQL (bukan sql). Salah satu fitur dari *database* ini dapat digunakan secara *realtime*. Pengolahan data pada mikrokontroler dirancang menggunakan Arduino IDE akan dilanjutkan secara *realtime* ke *database firestore*. Pengolahan data menggunakan Bahasa pemrograman JSON (*JavaScript Object Notation*) dan akan disinkronkan secara *realtime* ke setiap klien yang terhubung. *Firestore* juga merupakan sebuah *database* yang memiliki *host* atau *cloud*. *Firestore* akan mensinkronisasi data setiap kali data berubah maka setiap perangkat yang terhubung akan menerima pembaruan data dalam bentuk waktu milidetik[18]. *Firestore* memungkinkan digunakan pada platform iOS, Android dan Web. *Firestore* memiliki beberapa Fitur *Firestore* sebagai berikut:

1. *Firestore Analytics* adalah Fitur yang menunjukkan *reporting* dan koleksi data pada aplikasi iOS ataupun Android. Segmentasi *user* dapat dibuat berdasarkan atribut pengguna (*user* atribut). Data yang telah dikoleksi dapat digunakan beranekaragam disesuaikan dengan lokasi maupun persentas berdasarkan koleksi data yang paling sering dilihat atau digunakan. *User* atribut merupakan parameter yang dapat digunakan sebagai filter dalam *reporting* atau notifikasi.
2. *Firestore Cloud Messaging and Notifications* merupakan fitur yang akan memberikan koneksi yang cepat (handal), tidak dikenakan biaya dan bisa hemat baterai dalam penggunaannya. Penggunaan fitur ini dapat dilakukan

antar perangkat (*device*) atau antar *server* yang digunakan. Target segment dilakukan dengan ditentukan berdasarkan demografi dan *behavior*/perilaku.

3. *Firebase Authentication* merupakan fitur yang saling berhubungan dengan fitur lainnya pada *firebase*. Fitur ini mengenai layanan dalam *back-end*, pada iOS dan Android yang termasuk SDK yang dapat digunakan secara mudah, bentuk *interfaces* yang sudah siap untuk diautentikasi pengguna dengan menggunakan beberapa cara anatara lain dengan menggunakan sandi, nomor telepon dan layanan atau sistem penyedia autentifikasi yang terkenal saat ini seperti menggunakan facebook, google dan juga tampilan *interfaces* menggunakan *full drop-in authentication* pada *firebase* UI.
4. *Firebase Cloud Firestore* adalah salah satu fitur yang ditujukan sebagai pengembangan perangkat yang bersifat fleksibel dan terukur seperti *server firebase*, *google cloud perform*, seluler dan web. Fitur ini akan membantu menawarkan layanan koneksi data tetap berjalan pada aplikasi *user* secara *powerfull* namun dalam keadaan *offline* dan *responsive* digunakan oleh website maupun aplikasi seluler.
5. *Firebase Realtime Database* merupakan *database* yang bertugas menyimpan data dan mengeksekusi data dalam bentuk JSON kemudian disamakan (sinkron) kepada *user* yang terhubung secara langsung (*real-time*). Fitur *database* ini menggunakan *cloud* sebagai host. Pengolahan *database* dalam skala yang lumayan besar sangat dibantu oleh fitur ini. Fitur *Firebase Realtime Database* walau digunakan secara *offline* tetap responsif dikarenakan SDK dari fitur *Firebase Realtime Database* langsung melakukan penyimpanan data ke memori lokal (*disk device*). Perubahan atau pemberitahuan data akan langsung diterima oleh perangkat *user* pada saat *online*.
6. *Firebase Hosting* adalah layanan konten web untuk hosting yang memiliki satu instruksi implementasi yang juga memberikan konten dalam bentuk dinamis dan statis ke jaringan penayangan konten (CDN) yang cepat secara global. *Firebase Hosting* yang melakukan penayangan konten yang dapat digunakan untuk semua jenis konten seperti CSS, API, HTML dan mikro Express.Js dan yang lain serta sangat aman untuk digunakan [19].