

BAB II

TINJAUAN PUSTAKA

2.1 Perusahaan Daerah Air Minum (PDAM)

Air adalah salah satu kebutuhan yang sangat vital dan mutlak diperlukan oleh semua makhluk hidup terutama manusia. Dalam kehidupannya sehari-hari, manusia tidak mungkin terlepas dari kebutuhannya akan air. Karena itulah penyediaan air bersih sangat diperlukan untuk keperluan konsumsi.

Perusahaan Daerah Air Minum yang merupakan kepanjangan PDAM adalah salah satu unit usaha milik daerah, yang bergerak dalam distribusi air bersih bagi masyarakat umum. PDAM terdapat di setiap provinsi, kabupaten, dan kotamadya di seluruh Indonesia.



Gambar 2. 1 Logo PDAM Tirta Musi [1]

PDAM merupakan perusahaan daerah yang memberikan jasa penyediaan air kepada seluruh penduduk masyarakat Indonesia yang membutuhkan. PDAM mengecek jumlah penggunaan air pada masing-masing pelanggan setiap bulan dengan mengirimkan petugas ke rumah pelanggan untuk mengecek dan mencatat jumlah penggunaan air melalui meter air.

Penggunaan meteran air PDAM pada bak hampir disetiap rumah di daerah kota ataupun kabupaten di Indonesia sudah terpasang karena sesuai dengan program pemerintah daerah untuk menyalurkan air bersih kepada masyarakat. Saat ini PDAM mendistribusikan air masih menggunakan meteran air yang terpasang, met eran air yang digunakan jenis meteran air analog. Untuk setiap perhitungan pemakaian air pelanggan dapat dilihat dari besaran angka pada meteran air tersebut.

2.2 Internet of Things (IoT)

Internet of things, atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-terusan. Adapun kemampuan seperti berbagai data, *Remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata.

Pada dasarnya, *Internet of Things* mengacu pada benda yang dapat diidentifikasi secara unik sebagai representasi virtual dalam struktur berbasis internet. Istilah *Internet of Things* awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui Auto-ID center di MIT. Kini IoT menjadi salah satu tugas bagi seorang mahasiswa di sebuah perguruan tinggi.

Jadi *Internet of Thing* (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer.



Gambar 2. 2 Ilustrasi dari *Internet of Things* [2]

“A Things” pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan orang dengan monitor implant jantung, hewan peternakan dengan transponder biochip, sebuah mobil yang telah dilengkapi built-in sensor untuk memperingatkan pengemudi ketika tekanan ban rendah. Sejauh ini, IoT paling erat hubungannya dengan komunikasi *machine-to-machine* (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan

kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau “*smart*”. Sebagai contoh yaitu *smart* kabel, *smart* meter, *smart* grid sensor.

Perkembangan teknologi semakin pesat dari waktu ke waktu. Mulai dari mobil pintar (*smart car*) yang bisa berjalan sendiri ke berbagai tujuan tanpa pengemudi manusia, hingga perangkat rumah pintar (*smart home*) semacam Alexa yang bisa otomatis bersuara mengingatkan untuk melakukan aktifitas sesuai jadwal. Seluruh teknologi terbaru ini adalah bagian dari *Internet of Things*. *Internet of Things* (IoT) adalah sebuah konsep di mana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan adanya interaksi dari manusia ke manusia atau dari manusia ke komputer. *Internet of Things* (IoT) adalah struktur di mana objek, orang disediakan dengan identitas eksklusif dan kemampuan untuk pindah data melalui jaringan tanpa memerlukan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer.

IoT juga bisa mencakup teknologi-teknologi sensor lainnya, seperti teknologi nirkabel maupun kode QR yang sering kita temukan di sekitar kita, contoh penerapannya dalam benda yang ada di dunia nyata adalah untuk pengolahan bahan pangan, elektronik, dan berbagai mesin atau teknologi lainnya yang semuanya tersambung ke jaringan lokal maupun global lewat sensor yang tertanam dan selalu menyala aktif.

Tantangan terbesar yang bisa menjadi hambatan dalam mengkonfigurasi IoT adalah menjembatani kesenjangan antara dunia fisik dan dunia informasi dan bagaimana menyusun jaringan komunikasinya, dikarenakan jaringan yang dibutuhkan oleh IoT sangatlah kompleks. Selain itu, IoT juga memerlukan suatu sistem keamanan yang cukup ketat. Di samping masalah tersebut, biaya pengembangan IoT yang mahal juga sering menjadi faktor penyebab kegagalan, sehingga pembuatan dan pengembangannya bisa berakhir gagal produksi.

2.3 MIT App Inventor

MIT *App Inventor* adalah sebuah web sumber terbuka yang awal kemunculannya dikembangkan oleh Google dan saat ini dikelola oleh *Massachusetts Institute of Technology* atau lebih dikenal dengan MIT.

App Inventor ini didesain untuk para pemula dalam pemrograman untuk menciptakan aplikasi perangkat lunak khusus untuk sistem operasi *Android*. Penggunaan aplikasi pemrograman ini dilengkapi fitur *drag and drop* untuk mendesain user *interface* aplikasi *Android* yang akan dibuat. Sekilas, cara menggunakan *App Inventor* mirip dengan pemrograman *Visual Basic* dengan fitur *drag and drop* nya. Namun yang membedakan antara *App Inventor* dengan *Visual Basic* adalah dari segi koding.

Pada *App Inventor*, proses koding dilakukan dengan lebih sederhana karena hanya membutuhkan logika saja, tanpa harus mengingat bahasa pemrograman secara langsung. Dengan begitu aplikasi ini sangat cocok digunakan untuk proyek-proyek yang memanfaatkan *Android* namun tidak rumit dalam hal koding.

App Inventor juga mendukung penggunaan ekstensi, yang memungkinkan Anda untuk mengimpor komponen tambahan atau pustaka yang dikembangkan oleh komunitas. Ekstensi ini memberikan akses ke fitur dan layanan lanjutan di luar komponen bawaan. Salah satu kelebihan dari *MIT App Inventor* adalah kemampuan untuk menguji aplikasi secara langsung pada perangkat *Android* yang terhubung atau menggunakan emulator.

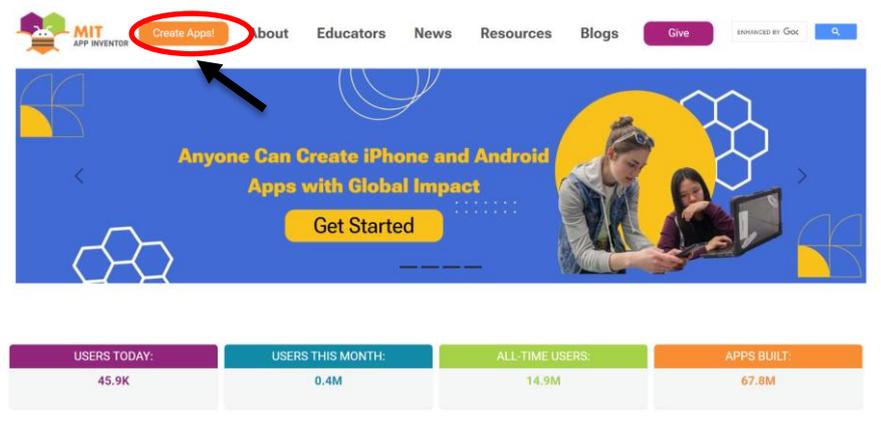


Gambar 2. 3 *MIT App Inventor* [3]

2.3.1 Cara Menggunakan *MIT App Inventor*

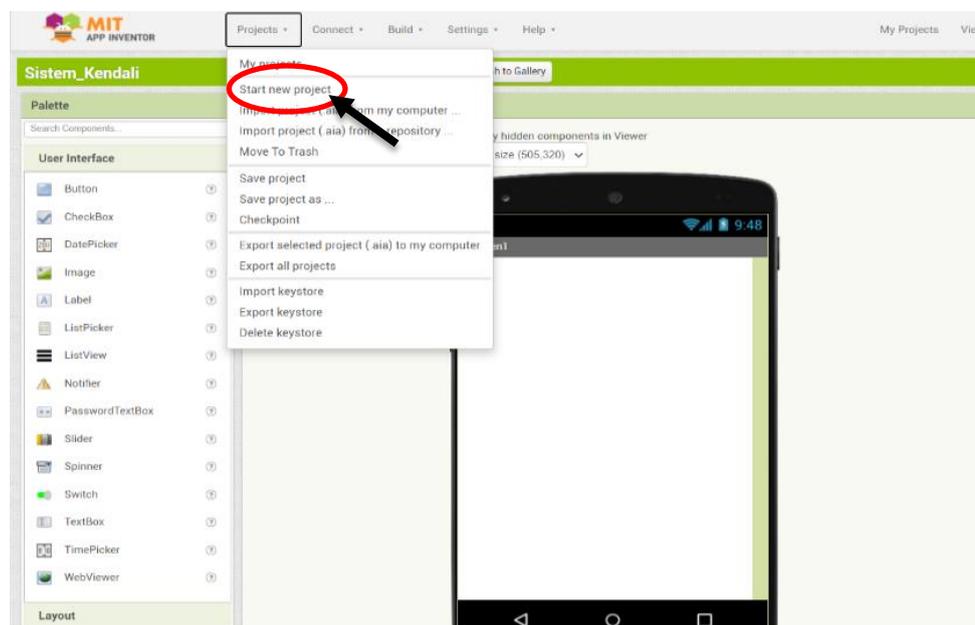
Berikut ini cara-cara dalam menggunakan *MIT App Inventor*

- Pertama Anda dapat mengakses *homepage MIT App Inventor online* dengan cara mengunjungi <https://appinventor.mit.edu>



Gambar 2. 4 Tampilan Halaman *MIT App Inventor*

- Selanjutnya, pilih menu *Create Apps*.
- Setelah itu, lakukan *login* menggunakan akun Google dan klik setuju mengenai *terms of service*.
- Pada halaman selanjutnya, Anda dapat membuat aplikasi baru dengan cara klik tombol *Project* lalu pilih opsi *Start New Project*. Di halaman ini, Anda dapat memasukkan berbagai komponen dan objek visual sesuai dengan keinginan.



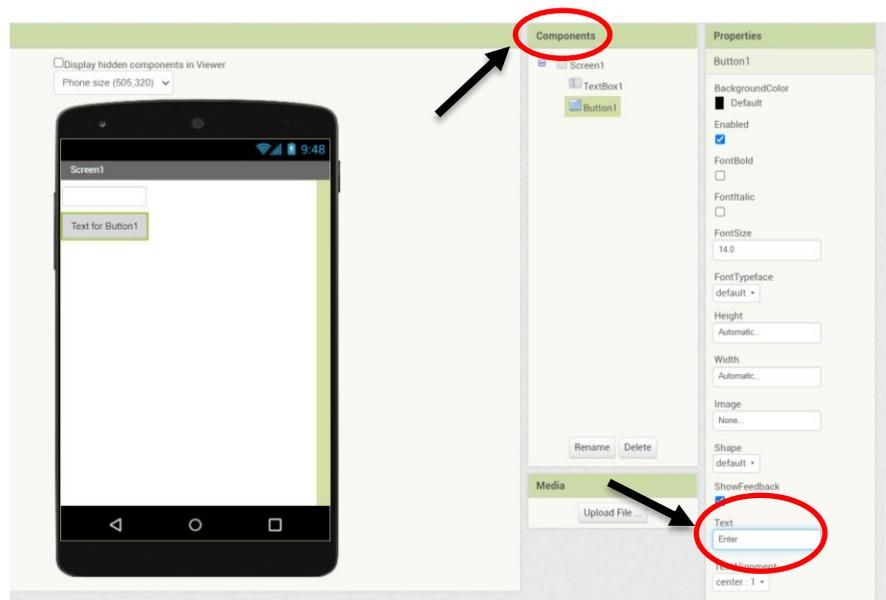
Gambar 2. 5 Tampilan Halaman Membuat *Project*

- Sebagai contoh cara membuat fungsi *TextBox* dan *Button*, dengan cara klik lalu *drag* ke dalam *display*.



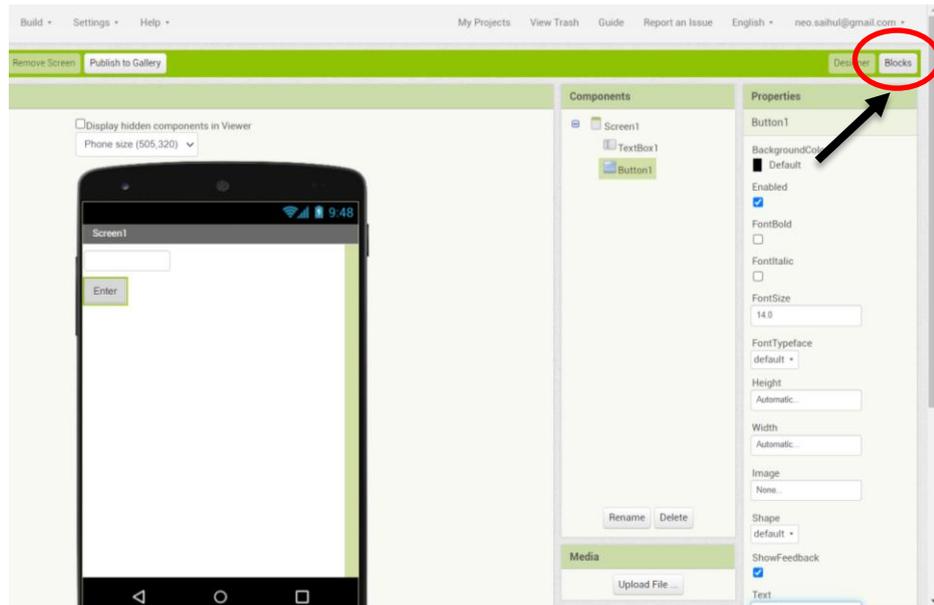
Gambar 2. 6 Tampilan Cara Membuat *Textbox* dan *Button*

- Ganti *Button* 1 dengan “Enter”. Dengan cara pilih *Button* 1, pada *Components*, lalu ganti pada *Text*.



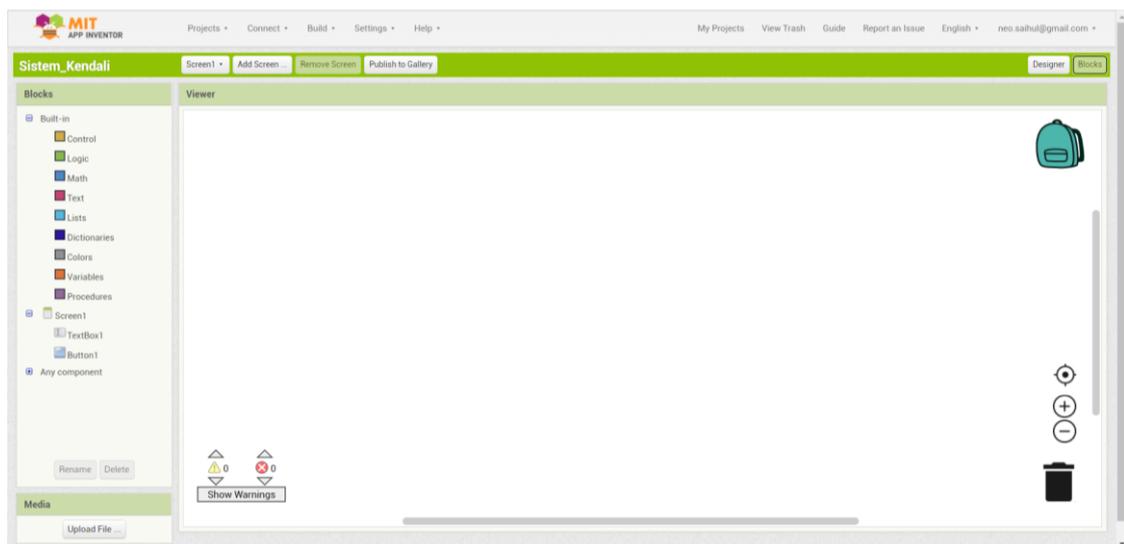
Gambar 2. 7 Tampilan Cara Ganti *Text* Pada *Button*

- Setelah selesai mendesain pada *display*, lalu beri perintah untuk menjalankan program dengan meng-klik *Blocks*.



Gambar 2. 8 Tampilan Menu *Block*

- Tampilan akan berubah untuk bahasa program berupa *block*.



Gambar 2. 9 Tampilan Mengubah *Scripting* bahasa program

- Lalu masukan bahasa program berupa *block*, sesuai dengan fungsi yang akan dibuat.



Gambar 2. 10 Tampilan Masukan Bahasa Program berupa Blok

Dalam contoh ini program yang dibuat adalah jika *button* di *click* maka script akan memanggil fungsi *connectivity web*, sebagai pilihan akses menuju server dengan alamat URL dan *Write API Keys* untuk penulisan update data *field* yang dipilih. Adapun nilai bilangan yang akan dikirim menuju server adalah nilai yang di *input* melalui *textbook*.

2.4 Firebase

Firebase merupakan suatu layanan yang dimiliki oleh Google dan digunakan untuk mempermudah para developer aplikasi dalam mengembangkan aplikasinya. *Firebase* memiliki banyak SDK yang memungkinkan untuk mengintegrasikan layanan ini dengan beberapa platform seperti *Android*, *iOS*, *JavaScript*, *App Inventor*, hingga *Unity*. *Firebase* memiliki layanan utama, yaitu menyediakan database realtime dan backend sebagai layanan (*Backend as a Service*). Backend sendiri adalah sebuah bagian dalam code aplikasi yang berhubungan langsung dengan isi database.

Layanan *Firebase* ini menyediakan pengembangan aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasikan di klien dan disimpan di cloud *Firebase* ini. Beberapa fitur yang dimiliki oleh *Firebase* adalah sebagai berikut :

1. *Firebase Analytics*.
2. *Firebase Cloud Messaging dan Notifications*.
3. *Firebase Authentication*.
4. *Firebase Remote Config*.

5. *Firestore* Real Time Database.
6. *Firestore* Crash Reporting.

Dua fitur yang menarik adalah *Firestore Remote Config* dan *Firestore Real Time Database*. Secara sederhananya, *Remote Config* adalah fitur yang memungkinkan developer mengganti / mengubah beberapa konfigurasi aplikasi *Android* / *iOS* tanpa harus memberikan update aplikasi via *Play Store* / *App Store*. Salah satu konfigurasi yang bisa dimanipulasi adalah seperti warna / tema aplikasi.

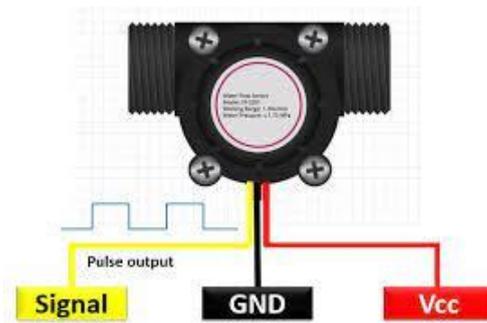


Gambar 2. 11 Firebase [4]

2.5 Water Flowmeter YF - S201

Water flowmeter merupakan suatu alat yang digunakan untuk mengukur aliran suatu zat baik itu berupa zat cair ataupun gas yang melalui suatu luas penampang tertentu. *Flowmeter* dalam proses kerjanya saat mengukur aliran akan menghasilkan suatu nilai keluaran yakni *flowrate* atau yang biasa kita dengar dalam kehidupan sehari-hari sebagai debit aliran. Debit aliran dari keluaran *flowmeter* dinyatakan dalam liter/hour, satuan tersebut dapat diperkecil lagi menjadi liter/menit atau liter/detik sesuai dengan kebutuhan [5].

Gambar 2.12 memperlihatkan salah satu bentuk *water flowmeter* YF - S201:



Gambar 2. 12 Water Flowmeter YF - S201 [5]

Volume dari aliran dapat diketahui dengan cara mengalikan debit aliran yang terbaca dengan waktu pengukuran, seperti yang telah disebutkan sebelumnya pada persamaan diatas. Pada perancangan yang akan dilaksanakan oleh penulis menggunakan *flowmeter* dengan tipe *YF-S201* G1/2 dimana tipe ini mampu mengukur debit air 1 sampai 30 liter/menit, spesifikasi dari sensor tersebut adalah sebagai berikut:

Tegangan input	: 5-24V DC
Tekanan maksimal	: 1.75 Mpa
Kapasitas	: 1-30 L/min
Panjang kabel	: 10.5 cm
Diameter luar	: 1.9 cm
Diameter dalam	: 1.1 cm
Maksimal arus kerja	: 15 Ma

Prinsip kerja dari *flowmeter* jenis ini ialah dengan menghitung putaran kincir yang terletak didalam sensor yang otomatis berputar ketika air melewatinya. Dalam kincir tersebut diletakan sebuah rotor yang mempunyai magnet, saat kincir berputar karena terpaan aliran air maka akan dihasilkan medan magnet berdasarkan prinsip *effect hall*, berdasarkan ada dan tidaknya medan magnet saat rotor berputar maka akan dihasilkan sinyal keluaran berupa gelombang kotak, sinyal keluaran inilah yang nantinya digunakan untuk menghitung volume dan debit air yang melewati sensor tersebut.

Tabel 2. 1 Komponen *Water Flow Sensor*

No.	Name	Quantity	Material
1	Valve Body	1	PA66+33% glass fiber
2	Stainless Steel Bead	1	Stainless Steel SUS304
3	Axis	1	Stainless Steel SUS304
4	Impeller	1	POM
5	Ring Magnet	1	Ferrite
6	Middle Ring	1	PA66+33% glass fiber
7	O-Seal Ring	1	Rubber
8	Electronic Seal Ring	1	Rubber
9	Cover	1	PA66+33% glass fiber
10	Screw	1	Stainless Steel SUS304

Water Flow Sensor ini terdiri atas katup plastik, *rotor* air, dan sebuah sensor *hall effect*. Ketika air mengalir melalui pipa dalam sensor ini, maka akan mengenai *rotor*, dan membuatnya berputar. Dimana kecepatan putar dari pada *rotor* tersebut akan berubah ketika kecepatan aliran air berubah pula. *Output* dari sensor *hall effect* akan sebanding dengan pulsa yang dihasilkan *rotor*. Selanjutnya nilai pulsa yang dihasilkan oleh sensor tersebut diteruskan kepada bagian mikrokontroler yang akan diolah sehingga akan menghasilkan nilai angka dalam meter kubik dan akan dikalkulasikan dengan harga yang berlaku [6]

Proses kalkulasi dilakukan dengan rumus sebagai berikut :

- Rumus penghitung Flow rate:

$$\text{Flow rate} = \text{pulse count} / 7.5 \dots \dots \dots (1)$$

- Rumus penghitung Flow milliliter:

$$\text{Flow milliliter} = (\text{Flow rate} / 60) * 1000 \dots \dots \dots (2)$$

- Rumus penghitung Flow liter:

$$\text{Flow liter} = (\text{Total milliliter} / 1000) \dots \dots \dots (3)$$

- Rumus penghitung Flow kubik:

$$\text{Meter kubik} = (\text{Flow liter} / 1000) \dots \dots \dots (4)$$

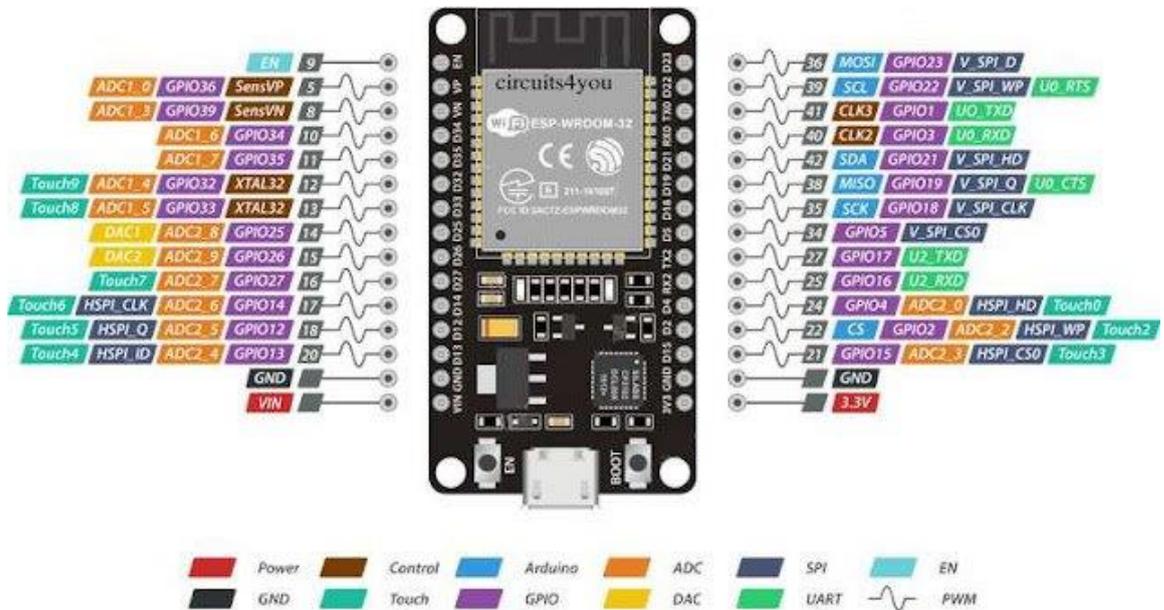
Dari rumusan di atas untuk mendapatkan nilai satu liter air diperoleh dari:

Kecepatan alir dibagi dengan 7.5. Yang mana laju alir ini didapat dari *output* sensor yang kemudian dibagi dengan 7.5. Sedangkan nilai 7.5 adalah nilai proses *kalibrasi* yang ditentukan oleh *Data Sheet*. Hasil kecepatan alir dibagi 60 dan dikali dengan 1000, yang tujuannya untuk mendapatkan nilai dalam mililiter.

Selanjutnya harga mililiter ini akan selalu menjumlahkan secara *continue*. Dan terakhir total mililiter yang dijumlahkan secara *continue* dibagi dengan 1000 yang bertujuan untuk mengkonversi nilai debit air dalam Liter. Setelah angka dan rumusan tersebut dimasukkan kedalam sebuah program, dengan format yang berlaku, maka ESP32 akan mengeluarkan *output* berupa nilai harga dan total jumlah pemakaian air yang akan ditampilkan di OLED dan *Android*.

2.6 NodeMCU ESP32

ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi Internet of Things. Module ESP32 merupakan penerus dari module ESP8266 yang cukup populer untuk Aplikasi IoT. Pada ESP32 terdapat inti CPU serta Wi-Fi yang lebih cepat, GPIO yang lebih, dan mendukung Bluetooth Low Energy.



Gambar 2. 13 ESP32 [7]

Terlihat pada gambar 2.13 merupakan pin out dari ESP32. Pin tersebut dapat dijadikan input atau output untuk menyalakan LCD, lampu, bahkan untuk menggerakkan motor DC.

Pada pin out tersebut terdiri dari :

- 18 ADC (Analog Digital Converter, berfungsi untuk merubah sinyal analog ke digital).
- 2 DAC (Digital Analog Converter, kebalikan dari ADC).
- 16 PWM (Pulse Width Modulation).
- 10 Sensor sentuh.
- 2 jalur antarmuka UART.
- pin antarmuka I2C, I2S, dan SPI.

Tabel 2. 2 Pin Input dan Output ESP32

GPIO	Input	Output	Catatan
0	Pulled Up	OK	Output sinyal PWM saat boot
1	TX Pin	OK	Output debug saat boot
2	OK	OK	Terhubung ke LED on board
3	OK	TX Pin	HIGH saat boot
4	OK	OK	Output sinyal PWM saat boot
5	OK	OK	Terhubung dengan SPI Flash terintegrasi
6	x	x	Terhubung dengan SPI Flash terintegrasi
7	x	x	Terhubung dengan SPI Flash terintegrasi
8	x	x	Terhubung dengan SPI Flash terintegrasi
9	x	x	Terhubung dengan SPI Flash terintegrasi
10	x	x	Terhubung dengan SPI Flash terintegrasi
11	x	x	Terhubung dengan SPI Flash terintegrasi
12	OK	OK	Boot gagal saat mendapatkan input high
13	OK	OK	
14	OK	OK	Output sinyal PWM saat boot
15	OK	OK	Output sinyal PWM saat boot
16	OK	OK	
17	OK	OK	
18	OK	OK	
19	OK	OK	
20	OK	OK	

2.6.1 Memori

Terdapat tiga jenis memori yang terdapat pada NodeMCU ESP32 yaitu :

1. Flash Memory, memori yang digunakan untuk menyimpan sketch/ program NodeMCU ESP32. Flash Memory adalah media penyimpanan yang berjenis “*non-volatile*” yang berarti tidak memerlukan power untuk menjaga keberadaan data. Kapasitas memorinya pun beragam, mempunyai kemampuan transfer data untuk penulisan mencapai 88 Mbps sedangkan untuk pembacaan mencapai 5 Mbps.

2. SRAM (*Static Random Access Memory*), memori yang digunakan untuk menyimpan data variabel sementara. Berbeda dengan ROM yang menyimpan program, memori bertipe RAM ini digunakan untuk menyimpan data. Data dalam memori ini akan hilang ketika daya ke mikrokontroler ditiadakan (*volatile*). Kita tidak boleh mendeklarasikan variabel sebanyak-banyaknya tanpa memperhatikan pertimbangan memori yang tersedia. Sebagai contoh, jika kita ingin membuat program logger yang melibatkan banyak data, kita disarankan menggunakan memori eksternal, misalnya SD card.
3. EEPROM (*Electrically Erasable Programmable Read Only Memory*), memori yang menyimpan data variabel dalam jangka waktu yang lama. EEPROM merupakan salah satu jenis memori yang memiliki alamat (*address*) yang didalam terdapat data (*value*). Pada NodeMCU ESP32 yang digunakan memiliki memori EEPROM sebesar 512 *kBytes*. EEPROM memiliki alamat sebanyak 1024 atau mulai dari 0 – 1023 dimana setiap alamat memiliki data sebesar 8 bit atau bernilai 0 – 255. Memori EEPROM tidak terhapus walaupun tanpa dialiri listrik.

2.6.2 Komunikasi

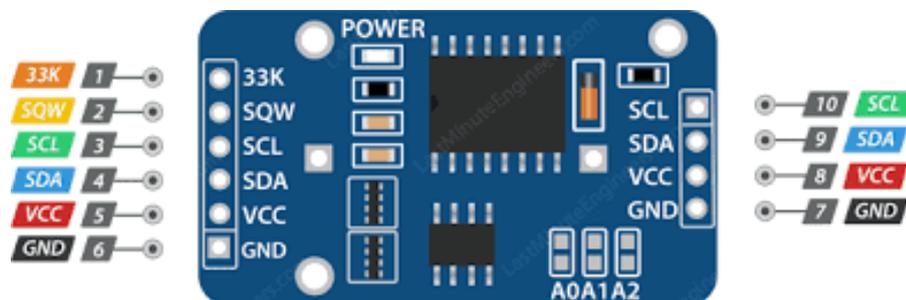
NodeMCU ESP32 telah melengkapi komunikasi serial dengan *port library* yang memudahkan untuk memprogram, yaitu :

1. *Serial Available*, digunakan untuk menyatakan angka, *bytes* atau karakter yang sudah siap dibaca dari *serial port*. Data ini adalah data yang telah diterima dan disimpan dalam *serial receive buffer*. *Serial receive buffer* dapat menampung 64*bytes* data.
2. *Serial Begin*, digunakan untuk mengatur *baudrate*/ kecepatan transmisi data. Beberapa pilihan kecepatan komunikasi data yang dapat digunakan pada board NodeMCU ESP32 adalah 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 atau 115200. Pengaturan *baudrate* dilakukan pada bagian *setup*.
3. *Serial End*, digunakan untuk menutup komunikasi *serial port*.

4. *Serial Find*, digunakan untuk membaca data dari *serial port buffer* hingga target yang ditentukan dalam perintah.
5. *Serial Print*, digunakan untuk menampilkan data ke serial monitor. Data yang ditampilkan dapat berupa karakter, *bytes*, atau angka.
6. *Serial Read*, digunakan untuk membaca data dari *serial port*.
7. *Serial Write*, digunakan untuk membaca data biner dari *serial port*. Data ini dikirim dalam bentuk *byte* atau deretan data *byte*.

2.7 Real Time Clock (RTC) DS3231

RTC (Real Time Clock) DS3231 adalah sebuah IC yang dapat digunakan sebagai pengatur waktu yang meliputi detik, menit, jam, hari, tanggal, bulan dan tahun. Pengaksesan data dilakukan dengan sistem serial sehingga hanya membutuhkan dua jalur untuk berkomunikasi yaitu jalur clock untuk membawa informasi data clock dan jalur data yang membawa data atau yang sering disebut dengan I2C (Inter-integrated Circuit).



Gambar 2. 14 Real Time Clock (RTC) DS 3231 [8]

- **Pin 33K:** Ini adalah pin output daya yang menyediakan suhu terkompensasi & referensi CLK yang akurat.
- **Pin SQW:** Pin ini pada frekuensi yang berbeda memberikan gelombang persegi dan juga sinyal interupsi.
- **Pin SCL:** Pin ini adalah pin CLK serial yang terutama digunakan dalam komunikasi I2C.
- **Pin SDA:** Ini adalah pin data serial, yang digunakan dalam komunikasi I2C.
- **Pin VCC:** Ini adalah pin catu daya modul, jadi sambungkan melalui 3.3V/5V
- **Pin GND:** Pin GND ini terutama digunakan untuk menyediakan GND umum.

2.8 Meteran Air

Meteran air merupakan salah satu alat untuk mengukur volume dari air pada jaringan perpipaan guna melayani pemakai, baik perorangan maupun kelompok dengan mencermati aspek teknis serta non teknis, sehingga masyarakat bisa dengan mudah mendapatkan air dengan jumlah tertentu, mutu yang sesuai persyaratan air minum untuk kesehatan. Aspek non teknis dibutuhkan dalam operasi serta perawatan meter air ialah dapat dibaca petugas maupun masyarakat sendiri, sehingga mudah mengetahui konsumsi air dengan pasti. Sejalan dengan kebutuhan maka sistem pelayanan air minum akan terus berubah dan dikembangkan.

Meter air sangat penting bagi perusahaan air minum untuk memonitor secara terus menerus pemakaian air pelanggan sehingga didapat rekening tagihan bulanan yang akurat, selain itu juga berfungsi untuk mengontrol dan mengendalikan pemakaian air pelanggan sesuai dengan kebutuhan.



Gambar 2. 15 Meteran Air [9]

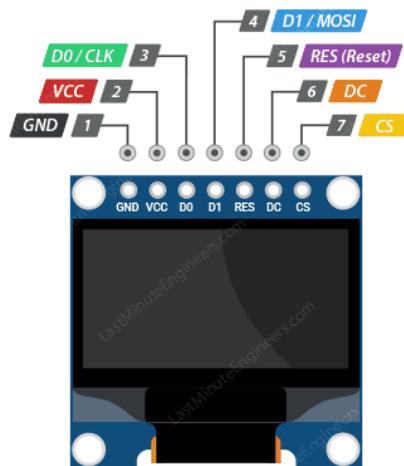
2.9 OLED display

OLED (Organic Light-Emitting Diode) adalah suatu display grafik berukuran 0.96 inci dan mempunyai resolusi 128 x 64 pixel . OLED biasanya terbuat dari karbon dan hidrogen. Pemrograman modul OLED menggunakan mikrokontroler *Arduino* yang berkomunikasi I2C, menggunakan 2 pin yaitu pin SDA dan Pin SCK sehingga dapat menghemat pin.

OLED banyak digunakan sebagai komponen penyusun layar monitor, baik televisi hingga *smartphone*. Berbeda dengan teknologi lain yang masih

menggunakan lampu latar, OLED dapat memberikan tampilan cahaya yang memancar sendiri tanpa bantuan lampu latar.

Selain mampu menampilkan cahaya sendiri, OLED juga memiliki beberapa keunggulan lain yang tidak kalah menarik. Mulai dari tampilan gambar yang lebih superior, rasio kontras yang tidak terbatas, hingga hemat daya atau energi. Tak heran, kini banyak teknologi yang menggunakan teknologi OLED untuk memberikan pengalaman yang menarik bagi pengguna.



Gambar 2. 16 OLED Display[10]

1. Nomor Pin 1 sebagai Ground pin modul
2. Nomor Pin 2 Vdd Vcc, 5V Power pin (dapat ditoleransi 3-5V)
3. Nomor Pin 3 SCK D0, SCL, CLK Bertindak sebagai pin jam. Digunakan untuk I2C dan SPI
4. Nomor Pin 4 SDA D1, MOSI Data pin modul. Digunakan untuk IIC dan SPI
5. Nomor pin 5 RES RST, RESET Mereset modul (berguna selama SPI)
6. Nomor Pin 6 Pin Perintah Data DC A0. Digunakan untuk protokol SPI
7. Nomor Pin 7 CS Chip Select Berguna bila lebih dari satu modul digunakan di bawah protokol SPI

2.10 Relay

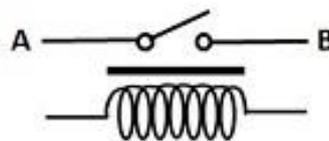
Relay adalah sebuah komponen atau perangkat saklar menjalankannya memakai listrik. Relay terdiri dari dua bagian utama yaitu coil dan kontak saklar atau mekanikal.

Fungsi relay saat di aplikasikan dalam sebuah komponen elektronika antara lain adalah :

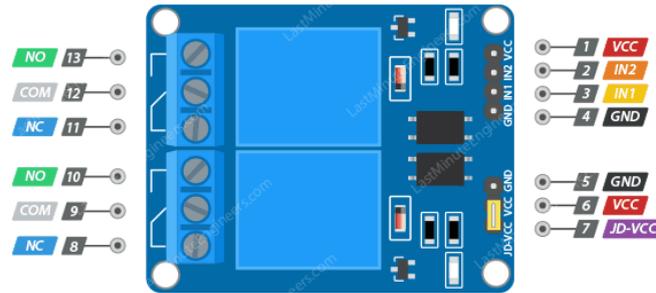
1. Mengatur sebuah rangkaian elektronika tegangan tinggi dengan menggunakan bantuan signal tegangan rendah
2. Menjalankan fungsi gerbang logika yaitu gerbang logika NOT.
3. Mengatur fungsi penundaan waktu.
4. Melindungi motor atau komponen lainnya dari kelebihan tegangan atau konsleting.

Di dalam relay terdapat kumparan elektromagnetik, jika kumparan tersebut dialiri sumber listrik maka akan menimbulkan medan magnet kemudian akan menarik tuas sehingga mengubah posisi dari kontak switch yang ada, yaitu dari yang sebelumnya NO (Normaly Open – Kontak Terbuka) yaitu saat relay tidak di beri tegangan menjadi NC (Normaly Close – Kontak Terhubung) yaitu saat relay di beri tegangan

Simbol Relay



Gambar 2. 17 Simbol Relay



Gambar 2. 18 Modul Relay 2 Channel [11]

2.11 Pompa air DC 12V

Pompa Air DC merupakan jenis pompa yang menggunakan motor dc dan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor, sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor. Pompa Air DC memiliki 3 bagian dasar :

1. Bagian yang tetap/stasioner yang disebut stator. Stator ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen.
2. Bagian yang berputar disebut rotor. Rotor ini berupa sebuah koil dimana arus listrik mengalir.
3. Gear Box yang dipasang pada pompa. Gear box ini didalamnya terdapat gear yang dipasang pada ujung rotor untuk menghisap air. Gaya elektromagnet pada motor DC timbul saat ada arus yang mengalir pada penghantar yang berada dalam medan magnet. Medan magnet itu sendiri ditimbulkan oleh megnet permanen. Garis-garis gaya magnet mengalir diantara dua kutub magnet dari kutub utara ke kutub selatan.



Gambar 2. 19 Pompa Air [12]

2.12 Integrated Development Environment (IDE) Arduino

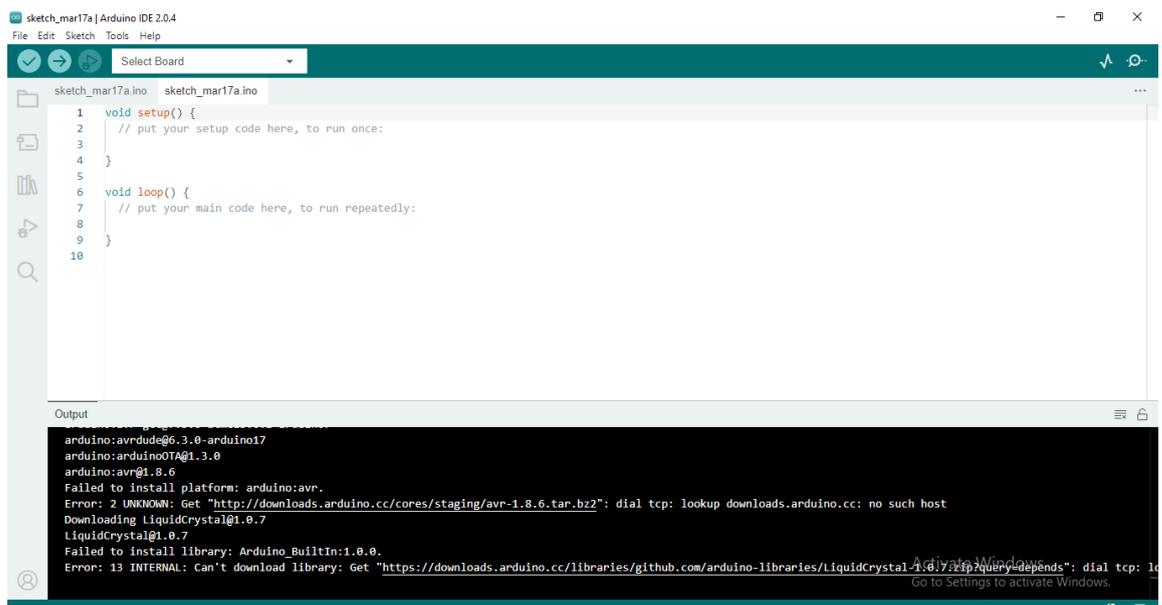
Arduino IDE dibuat dari bahasa pemrograman JAVA. *Arduino* IDE ini dikembangkan dari software Processing yang dirombak menjadi *Arduino* IDE khusus untuk pemrograman dengan *Arduino*.

Program yang ditulis dengan menggunakan *Arduino Software* (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi .ino. Teks *editor* pada *Arduino* Software memiliki fitur” seperti *cutting/paste* dan *searching/replacing* sehingga memudahkan kamu dalam menulis kode program. Pada *Software Arduino* IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan *Software Arduino* IDE, menunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan.

1. *Verify* berfungsi untuk melakukan checking kode yang kamu buat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum
2. *Upload* Berfungsi untuk melakukan kompilasi program atau kode yang kamu buat menjadi bahasa yang dapat dipahami oleh mesin alias si *Arduino*.
3. *New* berfungsi untuk membuat *Sketch* baru
4. *Open* Berfungsi untuk membuka *sketch* yang pernah kamu buat dan membuka kembali untuk dilakukan *editing* atau sekedar *upload* ulang ke *Arduino*.

5. *Save* Berfungsi untuk menyimpan *Sketch* yang telah kamu buat.

6. *Serial Monitor* Berfungsi untuk membuka serial monitor. *Serial monitor* disini merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara *Arduino* dengan *sketch* pada port serialnya. *Serial Monitor* ini sangat berguna sekali ketika kamu ingin membuat program atau melakukan debugging tanpa menggunakan LCD pada *Arduino*. *Serial monitor* ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan *error*.



Gambar 2. 20 Tampilan Arduino IDE

2.13 Rumus Perhitungan

2.13.1 Konversi Harga

Konversi harga adalah nilai harga yang akan dikeluarkan pada tampilan OLED dan android dengan acuan pada harga PDAM Tirta Musi.

1. Untuk Pemakaian Air <10 m³

$$\text{Total Harga} = \frac{\text{Nilai Volume di Sensor} \times 1595}{1000000} \dots\dots\dots(6)$$

2. Untuk Pemakaian Air 11-20 m³

$$\text{Total Harga} = \frac{(\text{Nilai Volume di Sensor} - 1000000) \times 2725}{1000000} + 18450 \dots\dots\dots(7)$$

3. Untuk Pemakaian Air 21-30 m³

$$\text{Total Harga} = \frac{(\text{Nilai Volume di Sensor} - 2000000) \times 3320}{1000000} + 45700 \dots \dots \dots (8)$$

4. Untuk Pemakaian >30 m³

$$\text{Total Harga} = \frac{(\text{Nilai Volume di Sensor} - 3000000) \times 4235}{1000000} + 78900 \dots \dots \dots (9)$$

2.10.2 Rata – rata

Rata-rata adalah nilai atau hasil pembagian dari jumlah data yang diambil atau diukur dengan banyaknya pengambilan data atau banyaknya pengukuran.

$$\text{Rata – Rata (X)} = \frac{\sum X_i}{n} \quad (10)$$

Dimana: X = rata-rata

$\sum X_i$ = Jumlah nilai data

n = Banyak data (1,2,3,...,n)

2.10.3 Error (%)

Error (kesalahan) adalah selisih antar a mean terhadap masing-masing data.

$$\% \text{error} = \frac{\text{Volume Sensor} - \text{Volume Meteran Air}}{\text{Volume meteran air}} \times 100\% \quad (11)$$