

**SISTEM KONTROL DAN *MONITORING* METERAN AIR
PERUSAHAAN DAERAH AIR MINUM (PDAM)
BERBASIS *IOT***

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Diusulkan oleh

Adam Ramadhan NIM : 0031801

Nila Fazila NIM : 0031821

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG**

2021

LEMBAR PENGESAHAN

**SISTEM KONTROL DAN *MONITORING* METERAN AIR
PERUSAHAAN DAERAH AIR MINUM (PDAM) BERBASIS
*IOT***

Oleh:

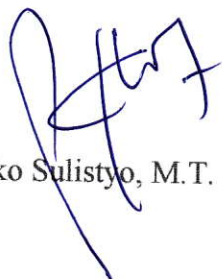
Adam Ramadhan / 0031801

Nilfa Fazila / 0031821

Laporan akhir ini disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



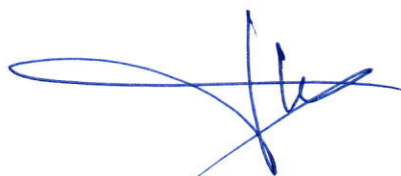
Eko Sulistyono, M.T.

Pembimbing 2



Nofriyani, M.Tr.T.

Penguji 1



Surojo, M.T.

Penguji 2



Muhammad Iqbal Nugraha, S.ST, M.Eng.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Adam Ramadhan NIM 0031801
Nama Mahasiswa 2 : Nila Fazila NIM 0031821

Dengan Judul : SISTEM KONTROL DAN *MONITORING* METERAN
AIR PERUSAHAAN DAERAH AIR MINUM (PDAM)
BERBASIS *IOT*

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 04 September 2021

Nama Mahasiswa

1. Adam Ramadhan
2. Nila Fazila

Tanda Tangan


...ADAM RAMADHAN...

...NILA FAZILA...

ABSTRAK

PDAM merupakan suatu badan usaha yang menyediakan pelayanan air bersih kepada masyarakat. Di setiap rumah pelanggan PDAM terpasang meteran air yang berfungsi sebagai pengukur volume dan debit air. Namun sistem pengukuran tersebut masih bersifat manual dan analog. Pelanggan PDAM tidak dapat melakukan monitoring serta pengontrolan terhadap penggunaan air setiap harinya. Tidak adanya suatu alat untuk me-monitoring penggunaan air ini menyebabkan kurangnya kesadaran masyarakat terhadap penghematan air dan melonjaknya tagihan bulanan. Untuk mengatasi permasalahan tersebut dibutuhkan suatu sistem pengontrol dan monitoring penggunaan air PDAM dalam bentuk digital. Tujuan yang dicapai pada proyek akhir ini yaitu mampu menampilkan data pengukuran volume air, debit air, dan tarif biaya penggunaan air melalui Smartphone sehingga dapat memudahkan dan menyadarkan pelanggan PDAM terhadap penghematan penggunaan air. Dengan demikian pada proyek akhir ini dirancang sebuah sistem kontrol dan monitoring meteran air PDAM dengan input air yang terdeteksi oleh Water Flow Sensor G1/2 yang dikontrol oleh mikrokontroler Arduino Uno sehingga diperoleh output berupa pembacaan besar volume air, debit air, dan tarif biaya yang dapat di monitoring pada smartphone. Pengujian proyek akhir ini dilakukan dengan membandingkan besar volume air yang terukur dan terbaca pada wadah, gelas ukur, LCD, dan aplikasi smartphone. Hasil pengujian didapatkan bahwa sistem pengontrolan air telah dapat dilakukan melalui aplikasi smartphone serta sistem mampu membaca besar volume air, debit air, dan tarif biaya penggunaan air dengan rata-rata persentase error 1,60% untuk kontrol otomatis dan 0,15% untuk kontrol manual. Dengan sistem ini maka pelanggan PDAM dapat dengan mudah memantau serta mengontrol penggunaan air setiap harinya secara real time melalui smartphone.

Kata kunci : Air, Pelanggan PDAM, sistem kontrol, Monitoring, Smartphone

ABSTRACT

PDAM is a business entity that provides clean water services to the community. In PDAM customer's house, a water meter is installed which functions as a measure of the volume and flow of water. However, the measurement system is still manual and analog. PDAM customers cannot monitor and control the use of water every day. The absence of a thing to monitor water use has led to a lack of public awareness of saving water and an increase in monthly bills. To overcome these problems, a system for controlling and monitoring PDAM water use is needed in digital. The goal achieved in this final project is to be able to display water volume measurement data, water discharge, and water usage fees via Smartphones so that it can facilitate and make PDAM customers aware of water usage savings. Thus in this final project a PDAM water meter control and monitoring system is designed with water input detected by the Water Flow Sensor G1/2 which is controlled by the Arduino Uno microcontroller so that the output is obtained in the form of large readings of water volume, water discharge, and cost rates that can be calculated. monitored on a smartphone. Testing of this final project is carried out by comparing the volume of water that is measured and read in containers, measuring cups, LCDs, and smartphone applications. The test results show that the water control system can be done through a smartphone application and the system is able to read the volume of water, water discharge, and water usage fees with an average percentage error of 1.60% for automatic control and 0.15% for manual control. With this system, PDAM customers can easily monitor and control daily water use in real time via smartphones.

Keywords: Water, PDAM Customers, control system, Monitoring, Smartphone

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat, ridha dan hidayah-Nya serta memberikan banyak kesempatan sehingga penulis dapat menyelesaikan karya tulis proyek akhir yang berjudul “Sistem Kontrol Dan *Monitoring* Meteran Air Perusahaan Daerah Air Minum (PDAM) Berbasis *IoT*” dengan baik.

Karya tulis proyek akhir ini disusun sebagai salah satu persyaratan akademik dan kewajiban mahasiswa pada semester 6 untuk menyelesaikan kurikulum program pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung. Selesainya penyusunan laporan ini tidak terlepas dari dukungan, semangat, serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan kali ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang ikut berperan dalam penyelesaian laporan ini, terutama kepada :

1. Keluarga yang selalu memberikan motivasi, dukungan, serta semangat sehingga penulis dapat menyelesaikan laporan ini sesuai dengan waktu yang telah ditentukan.
2. Bapak I Made Andik Setiawan, M.Eng,Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Ocsirendi, M.T. selaku Kepala Program Studi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Muhammad Iqbal Nugraha, M.Eng. selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak Eko Sulistyono, M.T, selaku pembimbing 1 dan Ibu Nofriyani, M.Tr.T. selaku pembimbing 2 dalam proyek akhir ini.
6. Seluruh staf pengajar dan karyawan Politeknik Manufaktur Negeri Bangka Belitung.
7. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu dalam proses penyelesaian Proyek Akhir ini.

8. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan Proyek Akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa penulisan Proyek Akhir ini masih jauh dari kata sempurna karena kesempurnaan hanya milik Allah SWT. Oleh sebab itu itu, penulis mengharapkan segala petunjuk, kritik dan saran yang bersifat membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan selanjutnya.

Besar harapan penulis semoga makalah tugas akhir dan alat proyek akhir yang dibuat dapat memberikan manfaat bagi pihak yang berkepentingan pada khususnya dan baik bagi perkembangan ilmu teknologi pada umumnya.

Sungailiat, 01 Juni 2021

Penulis

DAFTAR ISI

HALAMAN SAMPUL	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Proyek Akhir.....	3
BAB II DASAR TEORI	4
2.1 Perusahaan Daerah Air Minum (PDAM)	4
2.1.1 Perhitungan	4
2.2 Meter Air Analog.....	5
2.3 Sistem Kontrol	5
2.3.1 <i>Step Down DC Buck Converter</i> LM2596	6
2.3.2 Arduino Uno	7
2.3.3 <i>Water Flow Sensor</i>	9

2.3.4	NodeMCU Modul WIFI ESP8266.....	12
2.3.5	<i>Solenoid Valve</i>	13
2.3.6	<i>Internet of Things (IoT)</i>	16
2.3.7	Mit App Inventor	17
BAB III METODE PELAKSANAAN		18
3.1	Survei Data (Pengumpulan Data dan Pengolahan Data)	21
3.2	Perancangan Konstruksi Alat.....	21
3.3	Pembuatan Konstruksi Alat	21
3.4	Perakitan Konstruksi Alat	21
3.5	Perancangan <i>Hardware</i> Elektrik.....	22
3.6	Pembuatan <i>Hardware</i> Elektrik	22
3.7	Pengujian <i>Hardware</i> Elektrik	22
3.8	Perancangan <i>Software</i> Alat.....	22
3.9	Pembuatan <i>Software</i> Alat.....	22
3.10	Perakitan <i>Hardware</i> Elektrik dan <i>Software</i> Alat.....	23
3.11	Pengujian Kontrol dan <i>Monitoring</i> Alat	23
3.12	Pembuatan Laporan Proyek Akhir.....	23
BAB IV PEMBAHASAN.....		24
4.1	Deskripsi Alat	24
4.2	Diagram Blok Alat.....	24
4.2.1	Prinsip Kerja	25
4.3	Perancangan Konstruksi Alat.....	25
4.4	Pembuatan Konstruksi Alat	26
4.5	Perancangan <i>Hardware</i> Elektrik Alat.....	29
4.6	Pembuatan <i>Hardware</i> Elektrik Alat.....	29

4.7	Pengujian <i>Hardware</i> Elektrik Alat	31
4.7.1	Pengujian <i>Water Flow Sensor</i>	31
4.7.2	Pengujian <i>Solenoid Valve</i>	35
4.8	Perancangan <i>Software</i> Aplikasi	38
4.9	Uji Coba dan Pengambilan Data Sistem Kontrol dan <i>Monitoring</i> Meteran Air PDAM	43
4.9.1	Pengujian Kontrol Otomatis dan <i>Monitoring</i> Meteran Air PDAM	44
4.9.2	Pengujian Kontrol Manual dan <i>Monitoring</i> Meteran Air PDAM...	47
BAB V PENUTUP.....		52
5.1	Kesimpulan	52
5.2	Saran	52
DAFTAR PUSTAKA		53
LAMPIRAN.....		55

DAFTAR TABEL

Tabel 2. 1 Tarif Penggunaan Air PDAM 1	4
Tabel 2. 2 Spesifikasi <i>Step Down Buck Converter</i> LM2596	6
Tabel 2. 3 Tabel Spesifikasi Arduino Uno [10]	8
Tabel 2. 4 Tabel Spesifikasi <i>Water Flow Sensor</i>	10
Tabel 2. 5 Tabel Spesifikasi NodeMCU ESP8266	12
Tabel 2. 6 Tabel keterangan bagian-bagian <i>Solenoid valve</i>	14
Tabel 2. 7 Tabel Spesifikasi <i>Solenoid Valve</i>	15
Tabel 4. 1 Hasil Pengujian <i>Water Flow Sensor</i>	33
Tabel 4. 2 Hasil Pengujian <i>Solenoid Valve</i>	37
Tabel 4. 3 Hasil Pengujian Kontrol Otomatis Dan Monitoring Meteran Air PDAM	46
Tabel 4. 4 Hasil Pengujian Kontrol Manual dan <i>Monitoring</i> Meteran Air PDAM	48

DAFTAR GAMBAR

Gambar 2. 1 Meter Air	5
Gambar 2. 2 <i>Step Down Buck Converter</i> LM2596	6
Gambar 2. 3 Rangkaian <i>Equivalent Buck Converter</i>	6
Gambar 2. 4 Arduino Uno.....	8
Gambar 2. 5 <i>Water Flow Sensor</i>	9
Gambar 2. 6 Skematik Instalasi <i>Water Flow Sensor</i>	9
Gambar 2. 7 Bagian Dalam <i>Water Flow Sensor</i>	11
Gambar 2. 8 Prinsip Kerja Sensor <i>Hall Effects</i>	11
Gambar 2. 9 Fisik Modul NodeMCU	12
Gambar 2.10 Bagian-Bagian <i>Solenoid Valve</i>	14
Gambar 2. 11 Bentuk Fisik <i>Solenoid Valve</i>	15
Gambar 2. 12 Diagram Konsep <i>Internet Of Things</i>	16
Gambar 3. 1 <i>Flowchart</i> Sistem Kontrol dan <i>Monitoring</i> Meteran Air Perusahaan Daerah Air Minum (PDAM) Berbasis <i>IoT</i>	20
Gambar 4. 1 Diagram Blok Kontrol dan <i>Monitoring</i> Meteran Air Perusahaan Air Minum (PDAM) Berbasis <i>IoT</i>	24
Gambar 4. 2 Rancangan Konstruksi Alat.....	26
Gambar 4. 3Konstruksi Kontrol <i>Monitoring</i> Meteran Air PDAM Tampak Belakang.....	27
Gambar 4. 4 Konstruksi Kontrol <i>Monitoring</i> Meteran Air PDAM Tampak Atas	27
Gambar 4. 5 Konstruksi Kontrol <i>Monitoring</i> Meteran Air PDAM Tampak Depan	27
Gambar 4. 6 Konstruksi Kontrol <i>Monitoring</i> Meteran Air PDAM Tampak Samping Kanan	28
Gambar 4. 7 Konstruksi Kontrol <i>Monitoring</i> Meteran Air PDAM Tampak Samping Kiri	28

Gambar 4. 8 Konstruksi Prototipe Kontrol <i>Monitoring</i> Meteran Air PDAM Keseluruhan.....	28
Gambar 4. 9 Skematik Elektrik Sistem Kontrol dan <i>Monitoring</i>	29
Gambar 4. 10 Pemasangan <i>Hardware</i> Elektrik LCD 20 X 4 Pada Bagian Atas Konstruksi	30
Gambar 4. 11 Pemasangan Komponen Elektrik Utama Pada Bagian Tingkat 1 Konstruksi	30
Gambar 4. 12 Pemasangan <i>Hardware</i> Elektrik <i>Water Flow sensor</i> G1/2 Pada Bagian Tingkat 2 Konstruksi	30
Gambar 4. 13 Pemasangan <i>Hardware</i> Elektrik <i>Solenoid Valve</i> Pada Bagian Kiri Konstruksi	31
Gambar 4. 15 Diagram Blok Pengujian <i>Water Flow Sensor</i>	32
Gambar 4. 14 Skematik Elektrik Pengujian <i>Water Flow Sensor</i>	32
Gambar 4. 16 Ilustrasi Pengujian <i>Water Flow Sensor</i>	32
Gambar 4. 18 Diagram Blok Pengujian <i>Solenoid Valve</i>	35
Gambar 4. 17 Skematik Pengujian <i>Solenoid Valve</i>	35
Gambar 4. 19 Hasil Pengujian <i>Solenoid Valve Level Low</i>	37
Gambar 4. 20 Hasil Pengujian <i>Solenoid Valve Level Medium</i>	37
Gambar 4. 21 Hasil Pengujian <i>Solenoid Valve Level High</i>	38
Gambar 4. 22 Tampilan Awal <i>Mit App Inventor</i>	38
Gambar 4. 23 Tampilan Membuat Proyek Baru Pada <i>Mit App Inventor</i>	39
Gambar 4. 24 Tampilan Membuat Nama Proyek Pada <i>Mit App Inventor</i>	39
Gambar 4. 25 Tampilan awal membuat desain aplikasi pada <i>Mit App Inventor</i> ..	39
Gambar 4. 26 Tampilan Desain Aplikasi <i>Screen 1</i>	40
Gambar 4. 27 Tampilan <i>Blocks Program Screen 1</i>	40
Gambar 4. 30 Tampilan Desain Aplikasi <i>Screen 2</i>	41
Gambar 4. 28 Tampilan Desain Aplikasi <i>Screen 3</i>	41
Gambar 4. 29 Tampilan <i>Blocks Program Aplikasi Screen 2</i>	41
Gambar 4. 31 Tampilan <i>Blocks Program Aplikasi Screen 3</i>	42
Gambar 4. 32 Tampilan Desain Aplikasi <i>Screen 4</i>	42
Gambar 4. 33 Tampilan <i>Blocks Program Aplikasi Screen 4</i>	42

Gambar 4. 34 Skema Elektrik Pengukuran Volume dan Debit Air	43
Gambar 4. 35 Pengujian Pengukuran Volume Dan Debit Air	44
Gambar 4. 36 Proses Pengujian Kontrol Otomatis	45
Gambar 4. 37 Hasil Pengukuran Volume Air	45
Gambar 4. 38 Data pengukuran yang tersimpan	51

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup	55
Lampiran 2 Program Keseluruhan	57

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Air bersih merupakan kebutuhan utama makhluk hidup terutama manusia. Pentingnya air bagi kehidupan manusia dapat dilihat dari penggunaan air yang semakin meningkat [1]. Air digunakan setiap harinya untuk dikonsumsi, mandi, mencuci, dan lain-lain. Salah satu penyedia air bersih untuk kebutuhan masyarakat adalah Perusahaan Daerah Air Minum (PDAM) yang bertanggungjawab dengan kualitas air yang diproduksinya. *Supply* air dari PDAM dengan sistem prabayar ini memberikan kecukupan kepada masyarakat akan kebutuhan air bersih [2]. Di setiap rumah pelanggan PDAM terpasang meteran air yang berguna untuk mengukur serta mencatat besar volume dan debit air yang dikeluarkan. Namun sistem pengukuran dan pemeriksaan yang diterapkan oleh PDAM terhadap meteran air tersebut masih manual dan bersifat analog sehingga kurang efektif dan efisien [3]. Pada saat ini pelanggan PDAM tidak dapat melihat berapa banyak air yang telah digunakan serta tidak dapat melakukan pengontrolan terhadap penggunaan air tersebut per harinya. Dengan tidak adanya suatu alat *monitoring* terhadap penggunaan air ini maka kesadaran masyarakat terhadap penghematan penggunaan air masih kurang. Hal ini menjadi salah satu faktor penyebab melonjaknya tagihan biaya yang harus dikeluarkan untuk kebutuhan air ini kepada pihak PDAM setiap bulan [3]. Untuk mengatasi permasalahan tersebut dibutuhkan suatu sistem pengontrol dan *monitoring* penggunaan air PDAM dalam bentuk digital dan dapat diakses melalui *smartphone* sehingga dapat memudahkan pelanggan PDAM terhadap penghematan penggunaan air.

Pada proyek akhir ini akan dibuat sebuah sistem pengontrolan dan *monitoring* meteran air PDAM berbasis *IoT* yang dapat langsung diakses melalui aplikasi *smartphone*. Pada sistem ini, *supply* air sebagai *input* mengalir melewati *solenoid valve* sebagai keran elektrik yang dikendalikan melalui *smartphone* kemudian melewati saluran air yang telah dipasangkan *Water Flow Sensor G1/2*

yang berfungsi sebagai pendeteksi volume dan debit air, kemudian data yang terbaca oleh sensor diproses pada mikrokontroler Arduino Uno sebagai kontrol utama dalam sistem, dan NodeMCU ESP8266 sebagai sarana jaringan yang dapat mengirimkan data pengukuran volume air, debit air, serta total biaya pengeluaran air dari arduino ke *smartphone*. Data pengukuran tersebut disimpan dan dapat dilihat kembali pada aplikasi google sheet. Dengan adanya sistem ini pelanggan dapat dengan mudah mengontrol serta melakukan *monitoring* penggunaan air setiap harinya secara *real time* melalui *smartphone* sehingga dapat membangun kesadaran pelanggan PDAM terhadap penghematan penggunaan air dalam kehidupan sehari-hari.

1.2 Rumusan Masalah

Berikut rumusan masalah pada proyek akhir ini:

1. Bagaimana perancangan sistem kontrol dan *monitoring* meteran air PDAM ?
2. Bagaimana sistem pengontrol meteran air PDAM sehingga dapat diakses melalui *Smartphone*?
3. Bagaimana sistem *monitoring* meteran air PDAM sehingga dapat diakses oleh pengguna melalui *smartphone*?

1.3 Batasan Masalah

Berikut batasan masalah dalam perancangan dan pembuatan proyek akhir dengan judul Sistem Kontrol Dan *Monitoring* Meteran Air PDAM Berbasis *IoT*:

1. Sistem hanya dapat bekerja ketika mendapat sumber tegangan listrik.
2. *Water Flow Sensor G1/2* yang digunakan dapat membaca pergerakan air dan angin.
3. Tekanan aliran air mempengaruhi akurasi dari pembacaan *Water Flow Sensor*.
4. Data pengukuran akan mengulang kembali ketika terjadi pemadaman listrik tiba-tiba.

1.4 Tujuan Proyek Akhir

Tujuan dalam proyek akhir ini adalah:

1. Merancang sebuah sistem kontrol dan monitoring meteran air PDAM yang dapat diakses melalui *smartphone*.
2. Dapat melakukan pengontrolan air PDAM menggunakan *Solenoid valve* melalui *smartphone*.
3. Dapat melakukan *monitoring* penggunaan air PDAM melalui *smartphone*.

BAB II DASAR TEORI

2.1 Perusahaan Daerah Air Minum (PDAM)

PDAM merupakan salah satu Badan Usaha Milik Daerah yang menyediakan pelayanan air bersih kepada masyarakat. PDAM merupakan salah satu instansi pemerintahan daerah yang mempunyai jenis pelayanan berupa pelayanan barang. Dalam hal ini pelayanan barang yang dimaksud ialah berupa penyediaan air bersih [4].

Dalam menyediakan air bersih pihak PDAM menetapkan biaya pelayanan terhadap penggunaan air yang dikeluarkan kepada pelanggan setiap bulan. Penetapan biaya tersebut diambil daribesar volume serta debit air yang dikeluarkan setiap bulan yang kemudian dikonversikan kedalam nilai rupiah [3]. Untuk data tarif penggunaan air PDAM dapat dilihat pada tabel 2. 1 berikut.

Tabel 2. 1 Tarif Penggunaan Air PDAM 1

No	Pemakaian Dalam Meter Kubik	Harga (Rp/m ³)	Jasa Admin (Rp)
1.	0-10m ³	4.420	
2.	>10m ³	6.025	4.500

2.1.1 Perhitungan

Untuk menghitung debit air dapat digunakan rumus berikut :

- Rumus menghitung debit aliran

$$\text{Debit Air} = \frac{\text{Volume air}}{\text{waktu aliran}} \dots\dots\dots(1)$$

- Rumus menghitung waktu aliran

$$\text{Waktu Aliran} = \frac{\text{Volume air}}{\text{Debit air}} \dots\dots\dots(2)$$

Apabila telah diketahui debit aliran dan waktu aliran maka didapatkan besar volume aliran dengan rumus berikut :

$$\text{Volume Air} = \text{Debit aliran} \times \text{Waktu aliran} \dots\dots\dots(3)$$

Untuk menghitung tarif biaya penggunaan air PDAM diperoleh dari hasil pengukuran banyaknya volume air yang terdeteksi oleh *flowmeter sensor* kemudian dikalikan dengan tarif per kubik [5].

$$\text{Total Biaya (Rp)} = \text{Volume air (m}^3\text{)} \times \text{Tarif (Rp)} \dots\dots\dots(4)$$

2.2 Meter Air Analog

Meter air analog merupakan suatu alat yang berfungsi sebagai pengukur aliran air berupa volume dan debit melalui sistem kerja alat yang telah dilengkapi dengan unit sensor, unit pengukur, dan unit indikator penghitung untuk menyatakan volume serta debit air yang mengalir. Pada setiap rumah pelanggan PDAM terpasang meteran air yang berfungsi untuk mengukur serta mencatat besar volume dan debit air yang distribusikan oleh pihak PDAM kepada pelanggan [4]. Besar volume dan debit air yang keluar kemudian dikonversikan kedalam nilai rupiah sehingga dapat ditentukan serta diketahui jumlah biaya yang harus dikeluarkan sesuai dengan data yang diambil [3]. Untuk gambar fisik meter air analog dapat dilihat pada gambar 2. 1 berikut.



Gambar 2. 1 Meter Air [4]

2.3 Sistem Kontrol

Sistem kontrol adalah suatu sistem yang berfungsi sebagai pengendali dan pengatur keadaan dari suatu alat sehingga dapat memudahkan manusia. Peran sistem kontrol ini ialah mengatur keluaran yang telah ditetapkan oleh masukan melalui elemen sistem pada suatu kondisi [6].

2.3.1 Step Down DC Buck Converter LM2596

Step Down DC Buck Converter LM2596 merupakan *converter* penurun tegangan yang berfungsi sebagai pengubah *input* tegangan DC menjadi *output* tegangan DC yang lebih rendah [7]. Untuk bentuk fisik dari *Step Down DC Buck Converter* LM2596 dapat dilihat pada gambar 2. 2 berikut ini.



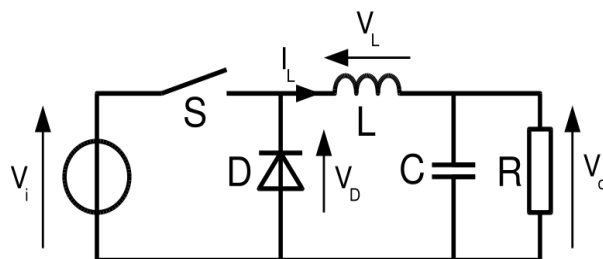
Gambar 2. 2 *Step Down Buck Converter* LM2596 [7]

Untuk tabel spesifikasi *Step Down Buck Converter* LM2596 dapat dilihat pada tabel 2. 2 dibawah ini.

Tabel 2. 2 Spesifikasi *Step Down Buck Converter* LM2596 [7]

No	Spesifikasi	Keterangan
1.	<i>Input Voltage</i>	DC 4V-40V
2.	<i>Input Voltage</i>	DC 1.3V-37V
3.	Arus Maksimum	2A
4.	<i>Board Size</i>	65 X 35 mm

Seperti namanya *Step Down Buck Converter* mampu menghasilkan tegangan yang sama atau lebih rendah dibandingkan dengan tegangan masukan. Untuk rangkaian *buck converter* dapat dilihat pada gambar 2. 3 berikut ini.



Gambar 2. 3 Rangkaian *Equivalent Buck Converter* [8]

Prinsip kerja *buck converter* yaitu menggunakan kendali saklar yang mana ketika saklar tertutup, dioda pada rangkaian bekerja pada kondisi *reverse* sehingga tegangan masukan mengalir ke induktor dan beban, sedangkan ketika saklar terbuka, dioda bekerja pada kondisi *forward*. Hal ini menyebabkan energi yang tersimpan pada induktor dapat mengalir ke beban [8].

Berikut merupakan persamaan umum tegangan keluaran yang dihasilkan dari *buck converter* [9] :

$$V_{out} = V_{in} \left(\frac{T_{ON}}{T} \right) \dots\dots\dots(5)$$

Atau

$$V_{out} = V_{in} \times D \dots\dots\dots(6)$$

Keterangan :

V_{out} = tegangan *output* (Volt)

V_{in} = tegangan *input* (Volt)

T_{ON} = waktu muncul denyut tegangan positif dalam satuan detik

D = *Duty-Cycle*

2.3.2 Arduino Uno

Arduino Uno merupakan mikrokontroler berbasis chip ATmega328. Arduino Uno ini memiliki 14 *input* dan *output* digital (6 *output* PWM), 6 analog input, dan juga memiliki tombol reset , koneksi USB, dan pin *header* ICSP. Kelebihan yang dimiliki Arduino diantaranya adalah telah tersedia berbagai *library* yang siap digunakan, tidak memerlukan perangkat chip programmer karena didalamnya telah terdapat *bootloader* yang mengatur proses *upload* program dari komputer, serta mempunyai sarana komunikasi USB. Untuk gambar fisik Arduino Uno dapat dilihat pada gambar 2. 4 berikut.



Gambar 2. 4 Arduino Uno [10]

Software arduino telah dilengkapi berbagai *library* yang cukup lengkap dan siap pakai sehingga bahasa pemrograman dari *software* ini tergolong sederhana dan lebih mudah untuk dipahami. Pada arduino terdapat modul siap pakai (*shield*) seperti *shield* GPS, *Ethernet*, *SD Card*, dll yang tertancap pada *board* arduino [10]. Untuk spesifikasi Arduino Uno dapat dilihat pada tabel 2. 3 berikut.

Tabel 2. 3 Tabel Spesifikasi Arduino Uno [10]

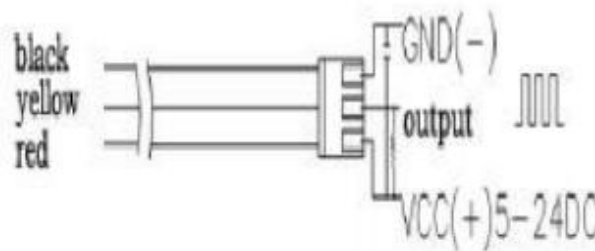
No	Spesifikasi	Keterangan
1.	Mikrokontroler	ATmega328
2.	Tegangan Operasi	5V
3.	Tegangan <i>input</i> yang disarankan	7-12V
4.	Batas tegangan <i>input</i>	6-20V
5.	Jumlah pin I/O digital	14 (dimana 6 menyediakan PWM output)
6.	Jumlah pin <i>input</i> analog	6
7.	Jumlah pin I/O PWM	6
8.	Arus DC tiap pin I/O	20 mA
9.	Arus DC untuk pin 3.3V	50 mA
10.	Memori Flash	32 KB, sekitar 0.5 KB digunakan oleh bootloader
11.	SRAM	2 KB
12.	EEPROM	1 KB
13.	Clock Speed	16 MHz
14.	LED_BUILTIN	13
15.	Panjang	68.6 mm
16.	Lebar	53.4 mm

2.3.3 Water Flow Sensor

Water flow sensor merupakan sebuah sensor yang berfungsi sebagai pengukur debit aliran air. Sensor ini terdiri dari beberapa bagian diantaranya adalah *valve body*, rotor air, dan sebuah sensor *hall effect*. Terdapat motor pada bagian dalam modul sensor yang dapat bergerak dengan kecepatan yang sesuai dengan kecepatan aliran air. Sensor *hall effect* yang terdapat pada modul sensor ini berfungsi sebagai pembaca sinyal berupa tegangan yang diubah menjadi pulsa dan kemudian di *transfer* ke mikrokontroler untuk diolah sebagai data pengukuran debit aliran air [11]. Untuk gambar fisik dari *Water flow sensor* dapat dilihat pada gambar 2. 5 berikut.



Gambar 2. 5 *Water Flow Sensor* [11]



Gambar 2. 6 Skematik Instalasi *Water Flow Sensor* [11]

Prinsip kerja *Water flow sensor* adalah ketika air mengalir melewati katup maka rotor air akan berputar dengan kecepatan tertentu sesuai dengan tingkat laju dari aliran air tersebut. Pada rotor air terdapat medan magnet yang memberikan efek terhadap sensor *hall effect* sehingga menghasilkan sinyal pulsa berupa tegangan (*Pulse Width Modulator*). Keluaran berupa pulsa tegangan tersebut memiliki tingkat tegangan yang sama dengan masukan frekuensi kecepatan aliran

air. Sinyal tersebut kemudian diproses di mikrokontroler menjadi data digital [11]. Untuk spesifikasi *Water Flow Sensor* dapat dilihat pada tabel 2. 4 berikut.

Tabel 2. 4 Tabel Spesifikasi *Water Flow Sensor* [11]

No	Spesifikasi	Keterangan
1.	Tegangan Operasional	5VDC-24VDC
2.	Arus operasional minimum	15 mA
3.	<i>Flow Rate</i>	1-30 L/min
4.	Kapasitas beban	10 mA
5.	Suhu Operasional Maks	180 derajat
6.	Suhu Air Maks	120 derajat
7.	Kelembaban	35%-90% RH
8.	Tekanan Air Maks	2.0 Mpa
9.	Operasi tekanan bawah	1,75 Mpa
10.	<i>Precision</i>	3%

Untuk menghitung nilai *pulse frequency* dalam satuan *Hertz* dapat digunakan persamaan berikut :

$$Pulse\ Frequency = 7,5 \times Q \dots\dots\dots(7)$$

Keterangan :

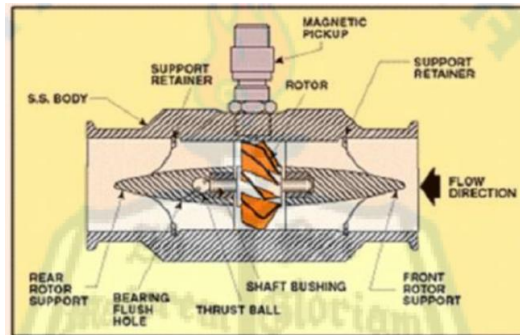
Q (*Flow rate*) = Debit air Liter/menit

Pulse Frequency = Pulsa Frekuensi (Hz)

7,5 = Pulsa per detik output dari *Water Flow Sensor*

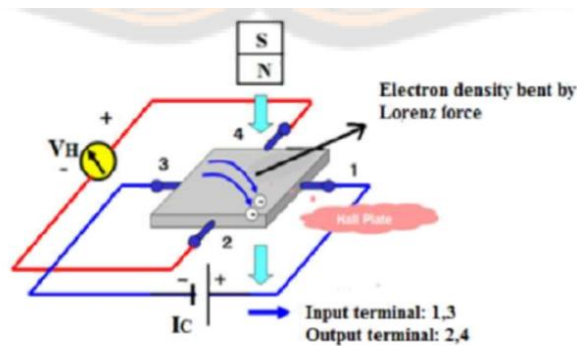
2.3.3.1 Prinsip Kerja Sensor *Hall Effect*

Sensor *Hall Effect* yang terdapat pada *Water Flow Sensor* ini merupakan *transduser* yang berfungsi sebagai pendeteksi putaran rotor dan menghasilkan sinyal pulsa berupa tegangan (*Pulse Width Modulator*) yang berperan sebagai masukan frekuensi ke mikrokontroler [12]. Untuk gambar bagian dalam *Water Flow Sensor* dapat dilihat pada gambar 2. 7 berikut.



Gambar 2. 7 Bagian Dalam *Water Flow Sensor* [12]

Sensor *Hall Effect* dibentuk dari sebuah lapisan semikonduktor yang tipis. Sensor *Hall Effect* memiliki masing-masing 2 terminal tegangan *input* dan *output*. Akibat dari gaya *Lorentz* pada Sensor *Hall Effect* fluks magnetik yang dimiliki tegak lurus terhadap lapisan semikonduktor menghasilkan *output* tegangan [12]. Untuk prinsip kerja dari *Sensor Hall Effect* dapat dilihat pada gambar 2. 8 berikut.



Gambar 2. 8 Prinsip Kerja Sensor *Hall Effects* [12]

Pada *element hall* tegangan yang diperoleh berbanding lurus dengan arus yang diperoleh serta kerapatan fluks magnetik. Maka didapatkan persamaan berikut :

$$V_H = R_H \left(\frac{I \times B}{T} \right) \dots \dots \dots (8)$$

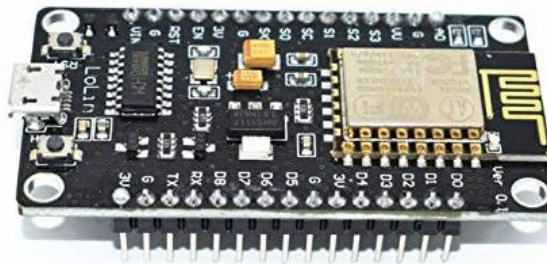
Keterangan :

- V_H = Tegangan *Hall*
- R_H = Koefisien *Hall Effects*
- I = Arus yang melalui sensor

- T = Tebal sensor (mm)
 B = Kerapatan fluks magnetik (Tesla)
 I_c = Diver Current

2.3.4 NodeMCU Modul WIFI ESP8266

NodeMCU merupakan sebuah *platform IoT* yang bersifat *Open-source firmware* yang terdiri dari perangkat keras berupa *system on chip* ESP8266 buatan *espressif system*. NodeMCU menggunakan Lua sebagai bahasa pemrograman *scripting*. Lua dan bahasa C mempunyai susunan pemrograman serta logika yang sama, perbedaannya hanya terdapat pada *syntax*. NodeMCU dapat dikatakan sebagai *board arduino* dari ESP8266. Secara fungsi modul ini hampir mirip dengan platform modul arduino [13]. Untuk bentuk fisik dari NodeMCU dapat dilihat pada gambar 2. 9 berikut.



Gambar 2. 9 Fisik Modul NodeMCU [13]

Kelebihan dari NodeMCU ini ialah mempunyai ukuran *board* yang cukup kecil dengan ukuran panjang 4.83 cm, lebar 2.54 cm, dan berat 7 gram. Selain itu modul ini dilengkapi dengan fitur *wifi* dan *firmware* yang bersifat *open-source*. NodeMCU mempunyai spesifikasi berupa board berbasis ESP8266 Serial WiFi Single on Chips (SoC) dengan onboard USB to TTL serta IEEE 802.11 b/g/n sebagai standar untuk Wireless. NodeMCU dilengkapi mikro USB port, tombol *reset*, dan *flash* [13]. Untuk spesifikasi NodeMCU dapat dilihat pada tabel 2. 5 berikut.

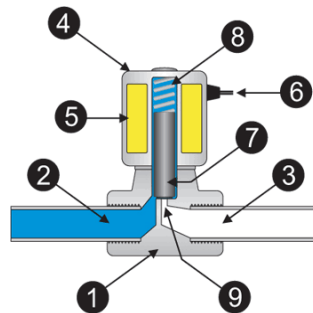
Tabel 2. 5 Tabel Spesifikasi NodeMCU ESP8266 [13]

No.	Spesifikasi	Keterangan
1.	Mikrokontroler	ESP8266

2.	Tegangan <i>input</i> yang disarankan	3.3V - 12V
3.	GPIO	13 Pin
4.	Kanal PWM	10 Kanal
5.	10 bit ADC Pin	1 Pin
6.	Memori <i>Flash</i>	4 Mb
7.	WiFi	IEEE 802.11 b/g/n
8.	Frekuensi	2.4 GHz – 22.5 GHz
9.	<i>Clock Speed</i>	40/26/24 MHz
10.	USB <i>to Serial Converter</i>	CH340G
11.	Panjang	57 mm
12.	Lebar	30 mm
13.	USB <i>Port</i>	Mikro USB

2.3.5 Solenoid Valve

Solenoid valve merupakan perangkat yang berfungsi sebagai saklar pada saluran air yang diaktifkan oleh energi listrik yang memiliki kumparan sebagai penggerakannya. Fungsi dari kumparan ini ialah sebagai penggerak piston yang dialiri oleh arus AC atau DC sebagai daya penggerak. *Solenoid valve* merupakan keran otomatis sehingga tidak perlu untuk memutar *valve* secara manual jika ingin mengubah kondisi. *Solenoid valve* dapat dihubungkan dengan perangkat sistem pengendali sehingga kondisi dari keran ini dapat dikontrol. *Solenoid valve* ini akan berada dalam kondisi terbuka (on) jika diberikan input tegangan 220V AC, dan akan tertutup (off) jika input tegangan terputus [14]. *Solenoid valve* terdiri dari 2 buah saluran yaitu saluran masuk cairan atau air (*inlet port*) dan saluran keluar cairan atau air (*outlet port*) [15]. Untuk bagian-bagian *Solenoid valve* dapat dilihat pada gambar 2. 10 berikut.



Gambar 2.10 Bagian-Bagian *Solenoid Valve* [12]

Untuk keterangan dari komponen diatas dapat dilihat pada tabel 2. 6 berikut:

Tabel 2. 6 Tabel keterangan bagian-bagian *Solenoid valve*

No	Keterangan
1.	<i>Valve Body</i>
2.	<i>Inlet Port</i>
3.	<i>Outlet Port</i>
4.	<i>Coil/Solenoid</i>
5.	<i>Coil Windings</i>
6.	<i>Lead Wires</i>
7.	<i>Plunger</i>
8.	<i>Spring</i>
9.	<i>Orifice</i>

Solenoid valve menggunakan tegangan kerja DC yaitu sebesar 12 Volt, 24 Volt, 48 Volt dan 110 Volt dalam prinsip kerja. Prinsip kerja *solenoid valve* yaitu pada saat *coil magnet* dialiri arus (aktif), maka *solenoid valve* akan tertarik menuju *coil magnet*. Hal ini menyebabkan *inlet hole* akan terbuka dan udara bertekanan akan masuk menekan batang actuator untuk menggerakkan *valve actuator on*. Namun ketika *coil magnet* tidak dialiri arus (off), *solenoid valve* akan terdorong menjauhi *coil magnet*, hal ini disebabkan adanya pegas pembalik yang mengakibatkan *outlet hole* akan terbuka dan tekanan udara masuk menekan batang actuator dan menggerakkan *valve actuator off*. *Solenoid valve* hanya mempunyai 2 kondisi, yaitu kondisi on dan kondisi off. Dalam menggerakkan

valve actuator, solenoid valve membutuhkan tekanan angin dengan besar nilai tekanannya disesuaikan dengan jenis *actuator valve* tersebut. Penerapan *solenoid valve* ini sangat cocok pada sistem kontrol yang membutuhkan kecepatan respon tinggi karena mempunyai tingkat responsibilitas yang cepat [15].

2.3.5.1 Electric Solenoid Valve G1/2

Jenis *solenoid valve* yang digunakan yaitu *Electric Solenoid Valve G1/2 normally closed*. *Electric Solenoid Valve G1/2* merupakan jenis *solenoid valve* yang mempunyai catu daya 12 VDC yang dilengkapi dengan 1 *port input* dan 1 *port output* berdiameter $\frac{1}{2}$ Inch. Untuk bentuk fisik dari *solenoid valve* dapat dilihat pada gambar 2. 11 berikut [12].



Gambar 2. 11 Bentuk Fisik *Solenoid Valve* [13]

Untuk spesifikasi *Solenoid Valve* dapat dilihat pada tabel 2. 7 berikut.

Tabel 2. 7 Tabel Spesifikasi *Solenoid Valve*

No.	Spesifikasi	Keterangan
1.	<i>Pressure</i>	Medium
2.	Material	Plastik
3.	Temperatur media	Normal
4.	Ukuran	$\frac{1}{2}$ inch
5.	Aplikasi	General
6.	Tekanan kerja	0-0.8 Mpa
7.	Tekanan air maksimum	over 2.5 Mpa
8.	Jenis koneksi	$\frac{1}{2}$ inch male thread inlet-outlet

9.	<i>Input power</i>	4.8 W (24 VDC 190mA)
10.	Tegangan kerja	12 VDC
11.	Arus Maksimum	450 Ma
12.	Mode Operasi	<i>Normally closed</i>
13.	<i>Valve responsse time</i>	Cepat
14.	<i>Inlet/Outlet ports</i>	G1/2

2.3.6 Internet of Things (IoT)

Internet of Things (IoT) merupakan sebuah konsep program yang bertugas dalam mengolah serta mentransmisikan data berupa informasi digital yang diperoleh dari komponen-komponen yang digunakan pada sistem melewati jaringan tanpa membutuhkan adanya interaksi dari manusia ke manusia. Tujuan dari *Internet of Things (IoT)* yaitu memperluas manfaat serta konektivitas internet secara terus menerus. Konsep *Internet of Things (IoT)* mengacu kepada 3 elemen utama yaitu objek/benda yang dilengkapi modul *Internet of Things (IoT)*, Perangkat yang mengkoneksikan objek ke internet seperti modem dll serta *platform clouds* untuk menyimpan aplikasi dan database [11]. Untuk diagram konsep dari *Internet Of Things (IoT)* dapat dilihat pada gambar 2. 12 berikut.



Gambar 2. 12 Diagram Konsep *Internet Of Things* [11]

Prinsip kerja dari *Internet of Things (IoT)* yaitu dengan memanfaatkan sebuah argumentasi pemrograman yang setiap perintah nya menghasilkan sebuah interaksi antara sesama mesin yang saling terkoneksi secara otomatis. Jaringan

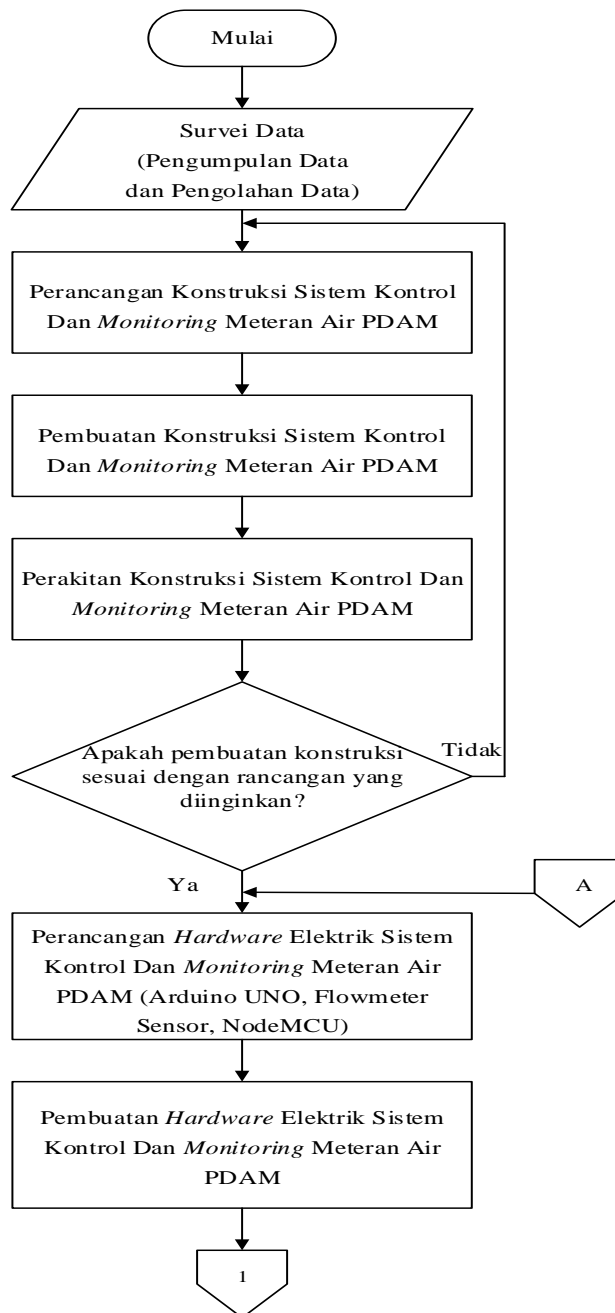
komunikasi dari *Internet of Things* sangatlah kompleks, sehingga memerlukan usaha yang cukup besar dalam melakukan proses konfigurasi *Internet of Things* tersebut [17].

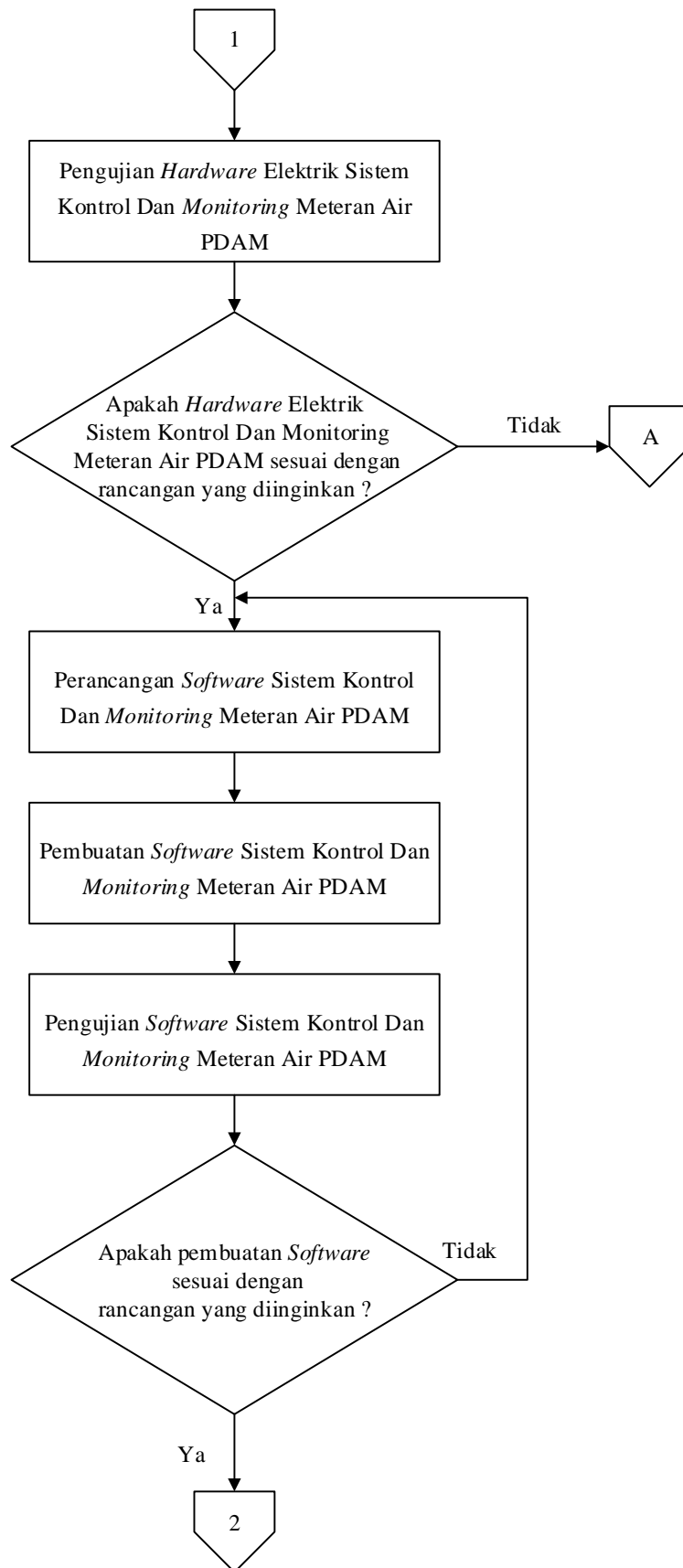
2.3.7 Mit App Inventor

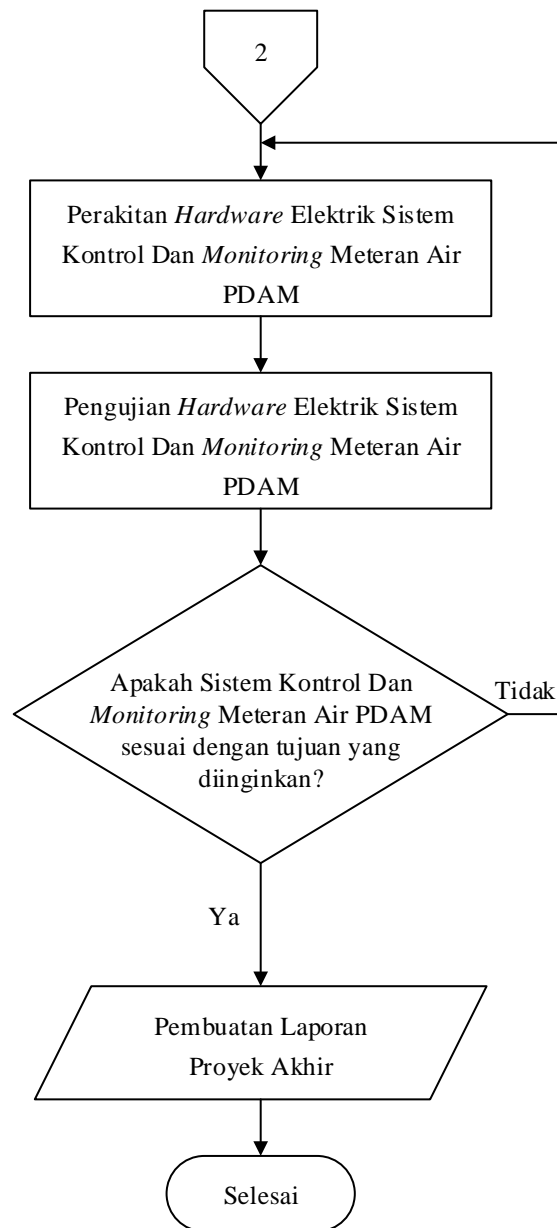
Mit App Inventor adalah *fitur mobile* yang sangat terkenal. pada *fitur* ini tersedia bermacam aplikasi mobile yang mudah untuk digunakan. MIT App Inventor mempunyai bahasa pemrograman berbasis blok visual. MIT App Inventor dirilis sebagai *software open source* oleh *Massachusetts Institute of Technology*. MIT App Inventor mempermudah pengguna dalam pembuatan sebuah aplikasi dan mengembangkan semua jenis aplikasi. MIT App Inventor memiliki dua jendela utama yaitu *Designer* komponen yang berfungsi untuk tampilan aplikasi dan *Editor Blok* yang berfungsi untuk menentukan perilaku aplikasi. Pada bagian *Designer* komponen, *user* dapat membuat tampilan aplikasi dengan memilih komponen-komponen yang merupakan bagian dari aplikasi yang kemudian ditambahkan pada tampilan aplikasi tersebut. Setelah tampilan aplikasi selesai maka pada bagian *Editor Blok*, *user* dapat membuat program menggunakan susunan kerangka *Blok* yang digabungkan untuk menentukan perilaku aplikasi sebagai respons. *User* dapat menguji aplikasi yang telah dibuat secara langsung pada *smartphone* atau emulator dari *smartphone* yang berjalan di komputer. Semua proyek yang telah dibuat pada MIT App Inventor dapat tersimpan otomatis serta memungkinkan *user* dapat masuk ke akun mereka kembali untuk melanjutkan proyek yang telah dibuat tersebut [17].

BAB III METODE PELAKSANAAN

Dalam pelaksanaan proyek akhir ini, dilakukan beberapa tahapan yang bertujuan untuk mempermudah dalam proses pengerjaan proyek akhir.







Gambar 3. 1 *Flowchart* Sistem Kontrol dan *Monitoring* Meteran Air Perusahaan Daerah Air Minum (PDAM) Berbasis *IoT*

3.1 Survei Data (Pengumpulan Data dan Pengolahan Data)

Survei data merupakan suatu proses pencarian informasi serta pengumpulan data-data yang dibutuhkan dalam proses pengerjaan proyek akhir ataupun penyusunan makalah proyek akhir yang diperoleh dari berbagai sumber referensi seperti jurnal dan lain-lain.

Pada tahap ini pengumpulan data terbagi menjadi dua aspek sumber data, yaitu pengumpulan data secara langsung dan pengumpulan data secara tidak langsung. Pengumpulan data secara langsung diperoleh dari hasil konsultasi dengan dosen pembimbing serta dari hasil survei data ke lapangan. Sedangkan pengumpulan data secara tidak langsung merupakan pengumpulan data yang diperoleh dari referensi-referensi buku dan jurnal yang berkaitan dengan proyek akhir.

Setelah proses pengumpulan data yang dibutuhkan selesai maka selanjutnya adalah tahap pengolahan data. Data-data yang telah diperoleh akan dipilih dan digunakan sebagai referensi dan acuan dalam pembuatan proyek akhir.

3.2 Perancangan Konstruksi Alat

Perancangan konstruksi alat pada proyek akhir ini dibuat menggunakan *software Solidworks* yang meliputi perancangan *box* sistem meteran air PDAM digital, saluran air PDAM, dan penentuan dimensi/ukuran material yang akan digunakan.

3.3 Pembuatan Konstruksi Alat

Pembuatan konstruksi alat ini dilakukan di luar area Politeknik Manufaktur Negeri Bangka Belitung dengan membuat konstruksi berupa *box* sistem meteran air PDAM digital, saluran air PDAM serta prototipe *supply* air PDAM.

3.4 Perakitan Konstruksi Alat

Perakitan konstruksi alat dilakukan dengan cara merakit dari setiap bagian *box* sistem menggunakan akrilik dengan ketebalan 3 mm sesuai dengan bentuk dan ukuran yang telah dirancang. Selain itu, pemasangan pipa PVC diameter ½ *Inch* sebagai saluran air dilakukan dengan memasang pipa tersebut melewati

bagian dalam *box* sistem sesuai dengan ukuran serta posisi yang telah dirancang menuju prototipe *supply* air PDAM.

3.5 Perancangan *Hardware* Elektrik

Proses perancangan *hardware* elektrik sistem kontrol *monitoring* meteran air PDAM dilakukan dengan menentukan komponen elektrik yang akan digunakan seperti Arduino Uno, *Water Flow sensor*, NodeMCU, dan *Solenoid valve* keran air otomatis.

3.6 Pembuatan *Hardware* Elektrik

Proses pembuatan *hardware* elektrik sistem kontrol *monitoring* meteran air PDAM dilakukan di luar area Politeknik Manufaktur Negeri Bangka Belitung yaitu dengan membeli serta memasang komponen elektrik yang telah jadi dan siap digunakan.

3.7 Pengujian *Hardware* Elektrik

Pengujian komponen elektrik ini dilakukan untuk mengetahui apakah komponen yang digunakan bekerja sesuai dengan fungsi yang diinginkan dan dibutuhkan. Berikut uji coba *hardware* pada komponen elektrik :

1. Uji coba Arduino Uno
2. Uji coba *Water Flow Sensor*
3. Uji coba NodeMCU ESP8266
4. Uji coba *Solenoid Valve*
5. Uji coba pengoneksian ke aplikasi *Mit App Inventor*

3.8 Perancangan *Software* Alat

Perancangan *software* ini dilakukan dengan membuat rancangan berupa pemrograman Arduino Uno sebagai pengontrol sistem. Selain itu perancangan *software* ini juga dilakukan dengan membuat rancangan *monitoring* meteran air PDAM yang akan ditampilkan pada *smartphone*. Perancangan program *monitoring* ini dibuat menggunakan aplikasi *Mit App Inventor*.

3.9 Pembuatan *Software* Alat

Pembuatan *software* yang akan dilakukan meliputi :

1. Pembuatan pemrograman pada Arduino Uno sebagai pengontrol sistem.
2. Pembuatan pemrograman tampilan monitoring pada *smartphone*. menggunakan aplikasi *Mit App Inventor*.

3.10 Perakitan *Hardware* Elektrik dan Software Alat

Perakitan *hardware* elektrik, *hardware* konstruksi, dan *software* sistem kontrol *monitoring* meteran air PDAM dilakukan dengan cara merakit secara keseluruhan setiap bagian dari *hardware* konstruksi dan *hardware* elektrik, serta pengkoneksian *software* yang telah dibuat menjadi satu kesatuan.

3.11 Pengujian Kontrol dan *Monitoring* Alat

Pengujian ini dilakukan untuk mengetahui hasil akhir secara keseluruhan dari sistem yang telah dibuat apakah telah bekerja sesuai dengan yang diinginkan atau belum. Berikut pengujian yang dilakukan :

1. Pengujian *Water Flow Sensor* dalam membaca volume air, debit air, serta tarif biaya penggunaan air PDAM yang keluar dan ditampilkan pada monitor *smartphone*.
2. Pengujian *solenoid valve* otomatis melalui *smartphone*.

3.12 Pembuatan Laporan Proyek Akhir

Pembuatan laporan merupakan tahap terakhir dalam pembuatan proyek akhir. Hal ini bertujuan untuk merangkum keseluruhan yang berkaitan dengan detail proyek akhir yang dibuat seperti latar belakang, tujuan, rumusan masalah, batasan masalah, landasan teori, metode pelaksanaan, pembahasan, serta kesimpulan dan saran.

BAB IV PEMBAHASAN

Pada bab ini akan dibahas mengenai proses serta pengujian alat yang dikembangkan dalam pembuatan Proyek Akhir dengan judul “Sistem Kontrol dan *Monitoring* Meteran Air Perusahaan Daerah Air Minum (PDAM) Berbasis *IoT*”.

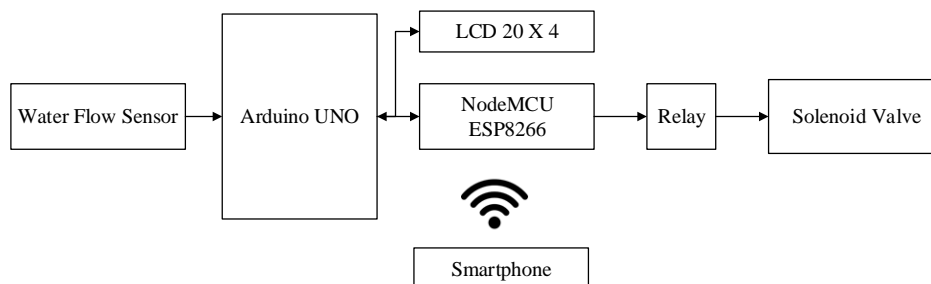
4.1 Deskripsi Alat

Kontrol dan *Monitoring* Meteran Air Perusahaan Air Minum (PDAM) Berbasis *IoT* ini merupakan sistem kontrol dan monitor penggunaan air pada meteran air yang pada awalnya parameter yang digunakan berbentuk analog berganti bentuk menjadi digital. Sistem pengontrolan dan *monitoring* dari alat ini menggunakan Arduino Uno dalam mengolah data dan NodeMCU ESP8266 sebagai pengirim data yang akan ditampilkan pada aplikasi *smartphone*. Untuk sistem kontrol dan *monitoring* dari alat ini digunakan *Water flow sensor* sebagai pengukur volume dan debit dari aliran air.

Sistem kontrol dan *monitoring* ini dapat menampilkan dan menyimpan hasil data pengukuran volume dan debit air yang terukur oleh *Water Flow sensor* dan dapat mengontrol pengeluaran air berdasarkan kendali dari *Solenoid valve* baik secara manual maupun otomatis berdasarkan *input* dari aplikasi *smartphone*.

4.2 Diagram Blok Alat

Diagram blok dari Sistem Kontrol dan *Monitoring* Meteran Air Perusahaan Air Minum (PDAM) Berbasis *IoT* ditunjukkan pada gambar 4.1 berikut.



Gambar 4. 1 Diagram Blok Kontrol dan *Monitoring* Meteran Air Perusahaan Air Minum (PDAM) Berbasis *IoT*

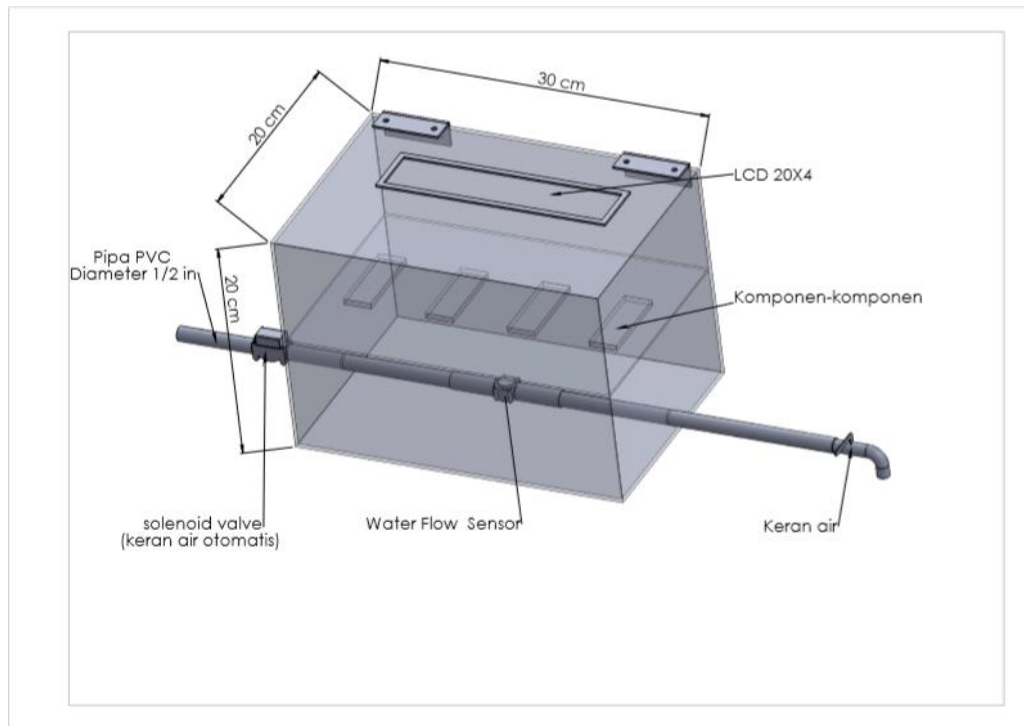
4.2.1 Prinsip Kerja

Prinsip kerja dari blok kontrol dan *monitoring* diatas adalah semua komponen elektrik pada sistem ini terhubung dengan *power supply* sehingga sistem ini hanya dapat bekerja ketika mendapatkan tegangan listrik. *Supply* air akan mengalir melalui saluran pipa PVC melewati *solenoid valve otomatis*. *Solenoid valve* ini akan berada dalam kondisi terbuka (on) jika diberikan input tegangan. Kondisi dari *solenoid valve* ini dikendalikan oleh mikrokontroler Arduino Uno yang memperoleh input dari *smartphone*. *Solenoid valve* terhubung dengan *driver relay*, dan NodeMCU. Perintah dari *smartphone* akan dikirim ke NodeMCU yang kemudian perintah tersebut akan dikirim ke *solenoid valve* melalui *driver relay* sehingga *solenoid valve* berada dalam kondisi sesuai dengan perintah yang diinginkan.

Pada saat *solenoid valve* dalam kondisi terbuka (on) maka air akan mengalir melewati *Water flow sensor*. Aliran air yang telah melewati *Water flow sensor* akan terbaca oleh sensor tersebut. Data yang terbaca oleh *Water flow sensor* adalah besar volume serta debit aliran air. Data yang terbaca akan dikirim dan diolah pada mikrokontroler Arduino Uno. Dari mikrokontroler kemudian data tersebut dapat di *monitoring* pada LCD. Selain itu, data dari mikrokontroler tersebut juga dikirim ke NodeMCU ESP8266 yang merupakan modul WiFi sehingga data tersebut dapat di *monitoring* melalui aplikasi *smartphone*.

4.3 Perancangan Konstruksi Alat

Tahap perancangan *hardware* konstruksi alat dilakukan dengan membuat rancangan *hardware* konstruksi sistem berdasarkan dengan konsep yang telah ditentukan menggunakan *software Solidworks*. Untuk desain konstruksi dapat dilihat pada gambar 4. 2 berikut.



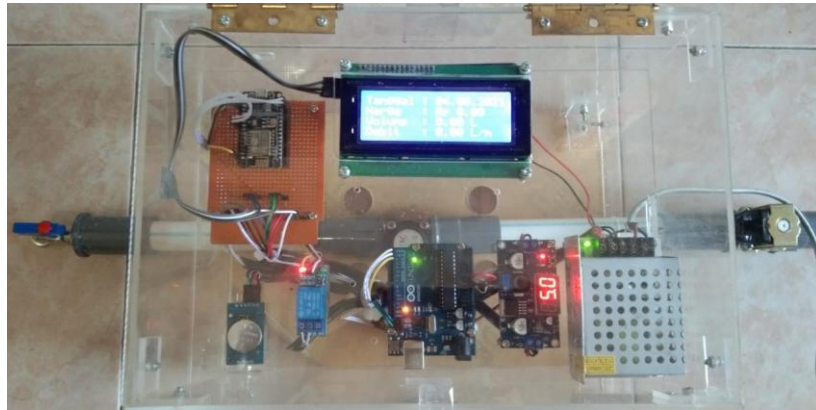
Gambar 4. 2 Rancangan Konstruksi Alat

Perancangan konstruksi *box* sistem meteran air PDAM ini direncanakan dibuat menggunakan bahan dasar akrilik dengan ketebalan 3 mm. Konstruksi ini berbentuk kotak dengan ukuran panjang 30 cm, lebar 20 cm, dan tinggi 20 cm. *Box* sistem ini disatukan dengan pipa PVC berdiameter $\frac{1}{2}$ inch sebagai saluran aliran air dengan panjang yang disesuaikan.

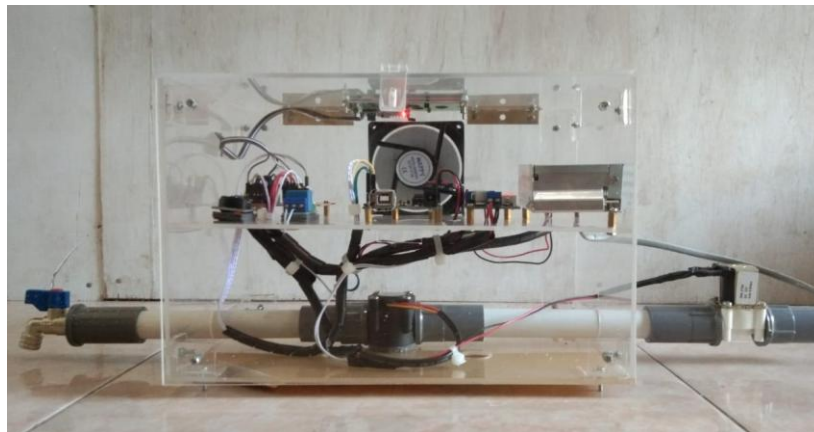
4.4 Pembuatan Konstruksi Alat

Pada tahap pembuatan konstruksi alat dilakukan perakitan serta pemasangan material-material menggunakan alat dan bahan yang telah ditentukan pada tahap perancangan. Dalam proses pembuatan konstruksi kami memilih akrilik dengan ketebalan sebesar 3 mm sebagai bahan dasar dari konstruksi yang kemudian dirakit membentuk sebuah *box* sistem dengan ukuran panjang 30 cm, lebar 20 cm, dan tinggi 20 cm. Setelah konstruksi dasar selesai dibuat, kemudian dipasang pipa PVC sebagai saluran air dengan bentuk dan ukuran yang telah ditentukan pada konsep.

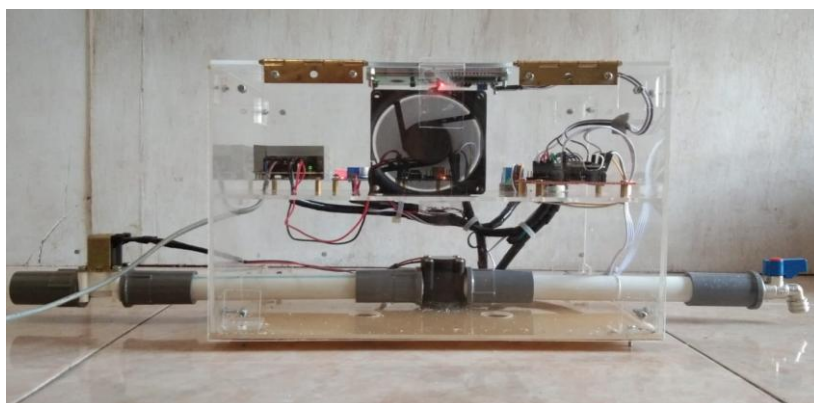
Berikut merupakan gambar hasil dari pembuatan konstruksi alat.



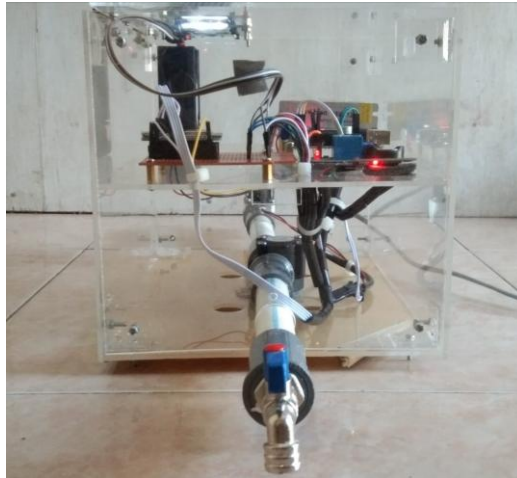
Gambar 4. 4 Konstruksi Kontrol *Monitoring* Meteran Air PDAM Tampak Atas



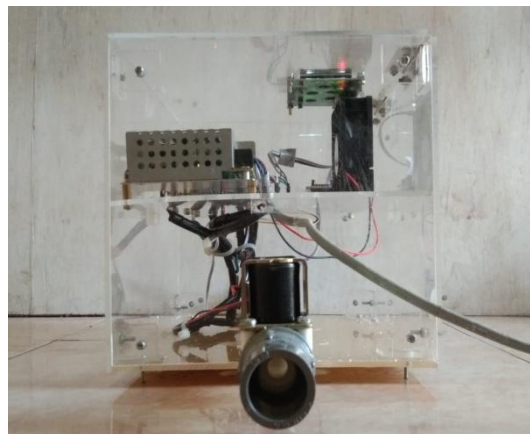
Gambar 4. 5 Konstruksi Kontrol *Monitoring* Meteran Air PDAM Tampak Depan



Gambar 4. 3 Konstruksi Kontrol *Monitoring* Meteran Air PDAM Tampak Belakang



Gambar 4. 6 Konstruksi Kontrol *Monitoring* Meteran Air PDAM Tampak Samping Kanan



Gambar 4. 7 Konstruksi Kontrol *Monitoring* Meteran Air PDAM Tampak Samping Kiri

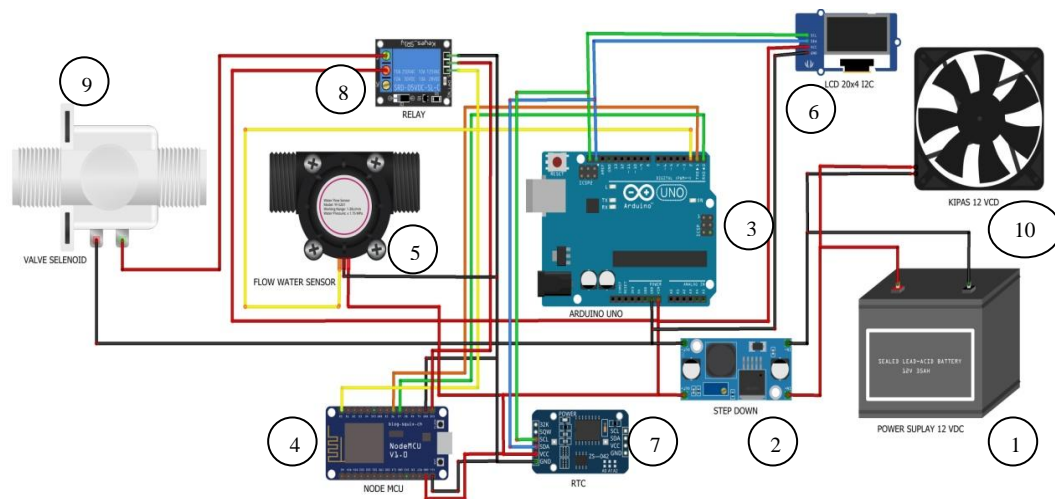


Gambar 4. 8 Konstruksi Prototipe Kontrol *Monitoring* Meteran Air PDAM Keseluruhan

4.5 Perancangan *Hardware* Elektrik Alat

Perancangan *hardware* elektrik alat dilakukan dengan merancang penempatan komponen-komponen yang digunakan pada sistem. Komponen alat meliputi Arduino Uno, Water Flow Sensor G1/2, Relay, LCD 20 X 4, NodeMCU ESP8266, RTC, *solenoid valve*, *Stepdown*, Kipas DC dan *power supply*. Perancangan *hardware* elektrik ini dibuat dengan menggunakan *software* *Fritzing*.

Untuk skematik elektrik sistem kontrol dan *monitoring* dapat dilihat pada gambar 4. 9 berikut.



Gambar 4. 9 Skematik Elektrik Sistem Kontrol dan *Monitoring*

Keterangan :

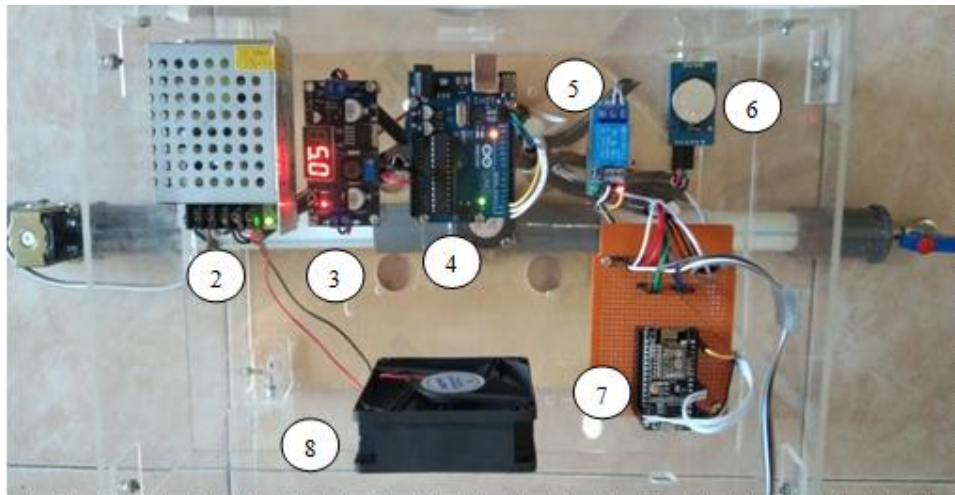
- | | |
|------------------------------|--------------------------|
| 1. <i>Power Supply</i> 12VDC | 6.LCD 20 X 4 12C |
| 2. <i>Step Down</i> | 7. RTC |
| 3. Arduino Uno | 8. Relay |
| 4. NodeMCU | 9. <i>Solenoid Valve</i> |
| 5. <i>Water Flow Sensor</i> | 10. Kipas 12 VDC |

4.6 Pembuatan *Hardware* Elektrik Alat

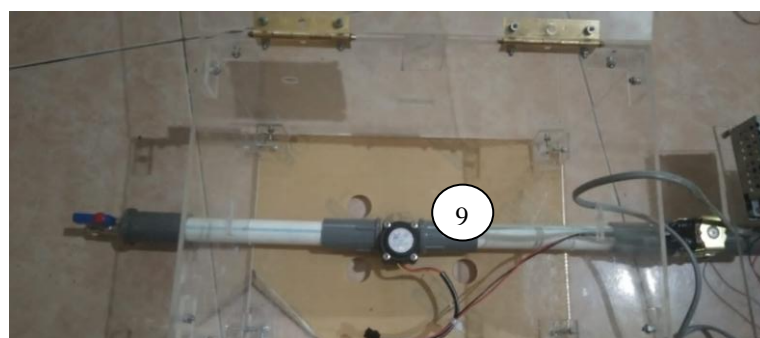
Tahap pembuatan *hardware* elektrik alat dilakukan dengan perakitan serta pemasangan komponen-komponen elektrik yang telah ditentukan pada rancangan konsep. Untuk hasil pemasangan komponen *hardware* elektrik dapat dilihat pada gambar berikut:



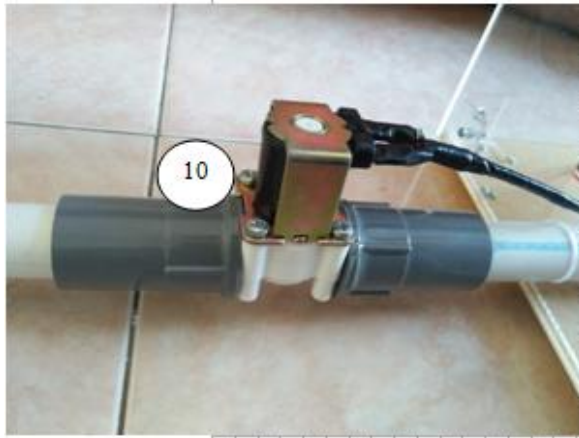
Gambar 4. 10 Pemasangan *Hardware* Elektrik LCD 20 X 4 Pada Bagian Atas Konstruksi



Gambar 4. 11 Pemasangan Komponen Elektrik Utama Pada Bagian Tingkat 1 Konstruksi



Gambar 4. 12 Pemasangan *Hardware* Elektrik *Water Flow sensor* G1/2 Pada Bagian Tingkat 2 Konstruksi



Gambar 4. 13 Pemasangan *Hardware* Elektrik *Solenoid Valve* Pada Bagian Kiri Konstruksi

Keterangan :

- | | |
|------------------------------|-----------------------------|
| 1. LCD 20 X 4 12C | 6. RTC |
| 2. <i>Power Supply</i> 12VDC | 7. NodeMCU |
| 3. <i>Step Down</i> | 8. Kipas 12 VDC |
| 4. Arduino Uno | 9. <i>Water Flow Sensor</i> |
| 5. Relay | 10. <i>Solenoid Valve</i> |

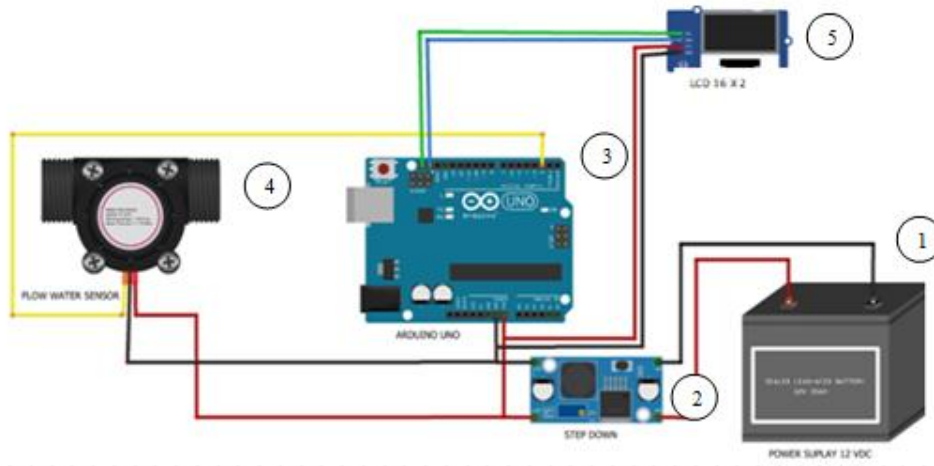
4.7 Pengujian *Hardware* Elektrik Alat

Pengujian *hardware* ini dilakukan untuk mengetahui kesesuaian fungsi dari setiap komponen-komponen yang digunakan. Berikut tahap pengujian komponen-komponen elektrik :

4.7.1 Pengujian *Water Flow Sensor*

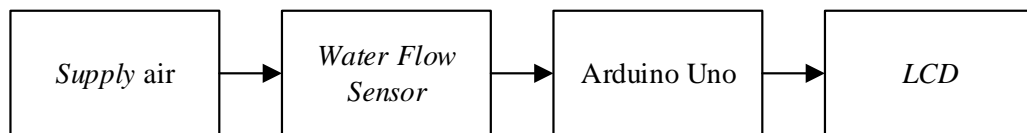
Pengujian *Water Flow Sensor* ini bertujuan untuk mengetahui tingkat keakuratan dari *Water flow sensor* apabila diberikan input berupa aliran air. Pengujian ini dilakukan dengan menghubungkan *Water Flow Sensor* ke mikrokontroler Arduino Uno menggunakan kabel jumper dimana pin-pin dari sensor ini akan dihubungkan ke mikrokontroler Arduino yang kemudian dihubungkan ke LCD 16 X 2 sebagai *monitoring output*. Pengujian ini dilakukan dengan membandingkan pengukuran volume air pada gelas ukur dengan volume

air yang terbaca pada LCD. Penyambungan pin-pin *Water Flow sensor* dapat dilihat pada gambar 4. 14 berikut.



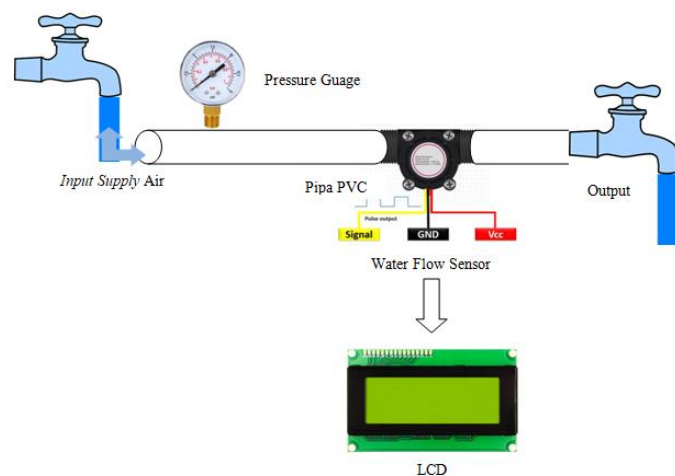
Gambar 4. 14 Skematik Elektrik Pengujian *Water Flow Sensor*

Untuk diagram blok pengujian *Water Flow sensor* dapat dilihat pada gambar 4.16 berikut:



Gambar 4. 15 Diagram Blok Pengujian *Water Flow Sensor*

Ilustrasi pengujian *Water Flow Sensor* dapat dilihat pada gambar 4. 16 berikut:



Gambar 4. 16 Ilustrasi Pengujian *Water Flow Sensor*

Berikut adalah *list* program pengujian *Water Flow sensor*.

```

bytesensorInt = 0; }
byteflowsensor = 2; } Pendefinisian tipe data byte dan pin yang digunakan
floatkonstanta = 6.4; } Nilai konstanta sensor water flow / nilai untuk pengkalibrasian
float debit; }
floatharga; } Pendefinisian tipe data desimal yang terbaca dari sensor water flow
float liter; }
float volume; }

unsigned long oldTime; } Pendefinisian tipe data untuk menyimpan nilai positif dan
unsignedintflowmlt; } nilai negatif dalam bentuk bilangan integer

volatile byte pulseCount; } Pendefinisian tipe byte qualifier yang digunakan
sebelum tipe data byte

if((millis() - oldTime) > 1000) {
detachInterrupt(sensorInt);
debit = ((1000.0/(millis()-oldTime))*pulseCount)/konstanta;
oldTime = millis();
flowmlt = (debit / 60) * 1000;
volume += flowmlt;
liter = volume/1000;
harga = liter*4,42;
}

```

Pembacaan nilai sensor *water flow* pada pin 2 dengan membaca sinyal pulse pada setiap 1000 ms dengan mengupdate nilai secara terus menerus dan nilai pembacaan volume air, debit air, dan jumlah harga yang terbaca.

Hasil pengujian *Water Flow sensor* tercatat pada tabel 4. 1 berikut.

Tabel 4. 1 Hasil Pengujian *Water Flow Sensor*

<i>Input</i>		<i>Output</i>				
No	Tekanan Air (MPa)	Pembanding Volume Air pada Gelas Ukur (Liter)	Pembacaan Volume Air pada LCD (Liter)	Debit air (Liter/menit)	Waktu (Detik)	<i>Error</i> (%)
1.	0,060	1,08	1,06	4,45	23.89	1,85

2.	0,065	1,18	1,19	10,48	11.11	0,85
3.	0,070	0,98	0,97	11,97	09.09	1,02
4.	0,075	1,00	1,00	13,07	08.22	0,00
5.	0,080	1,00	1,01	13,07	07.48	1,00

Rumus persentase *error* , dimana:

$$\text{Persentase } error = \left| \frac{\text{Alat Ukur} - \text{sensor}}{\text{Alat Ukur}} \right| \times 100\% \dots\dots\dots(8)$$

Alat Ukur = Pengukuran Gelas Ukur

Sensor = Hasil pembacaan sensor pada LCD

Perhitungan persentase *error* pada tabel 4.1 antara *input* dengan *output* yang didapat.

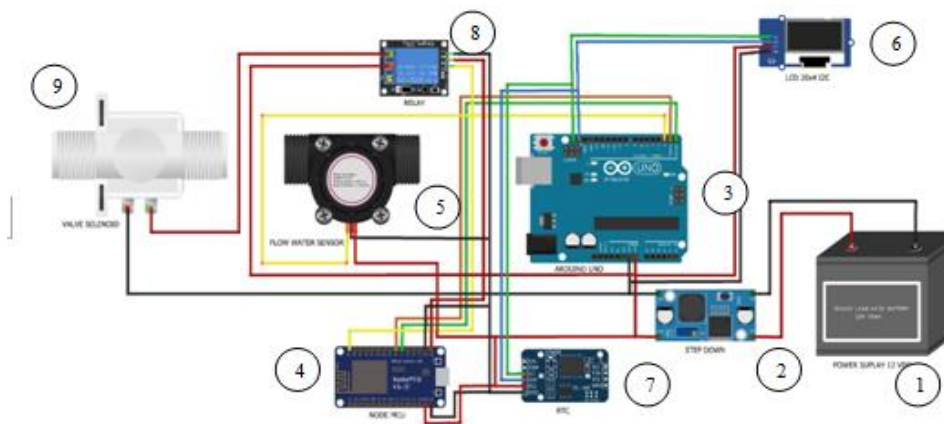
1. Persentase *error* = $\left| \frac{1,08-1,06}{1,08} \right| \times 100\% = 1,85\%$
2. Persentase *error* = $\left| \frac{1,18-1,19}{1,18} \right| \times 100\% = 0,85\%$
3. Persentase *error* = $\left| \frac{0,98-0,97}{0,98} \right| \times 100\% = 1,02\%$
4. Persentase *error* = $\left| \frac{1,00-1,00}{1,00} \right| \times 100\% = 0,00\%$
5. Persentase *error* = $\left| \frac{1,00-1,01}{1,00} \right| \times 100\% = 1,00\%$

Rata-rata *error* = 0,94 %

Berdasarkan Tabel 4.1 dapat dilihat bahwa *Water flow sensor* dapat membaca besar volume air dan debit air dengan tekanan air yang berbeda. dari pengujian yang dilakukan semakin tinggi nilai tekanan air maka semakin besar nilai debit air yang keluar serta semakin cepat pula waktu yang ditempuh dalam proses pengisian volume air. Dari volume air yang keluar diperoleh rata-rata *persentase error* keakuratan sebesar 0,94% sehingga dapat disimpulkan bahwa *Water flow sensor* berfungsi dengan baik sesuai dengan prinsip kerjanya sehingga dapat digunakan dalam pembuatan proyek akhir sebagai sensor pengukur volume serta debit air yang keluar.

4.7.2 Pengujian *Solenoid Valve*

Pengujian *Solenoid valve* ini bertujuan untuk mengetahui kondisi serta prinsip kerja dari *Solenoid valve*. Pengujian *Solenoid valve* dilakukan dengan memasang *Solenoid valve* dengan pipa kemudian pin-pin dari *Solenoid valve* ini akan dihubungkan ke relay dan NodeMCU ESP8266. Pengujian ini dilakukan dengan memberikan *input* berupa *supply* air bertekanan. Penyambungan pin-pin *Solenoid Valve* dapat dilihat pada gambar 4. 17 berikut.

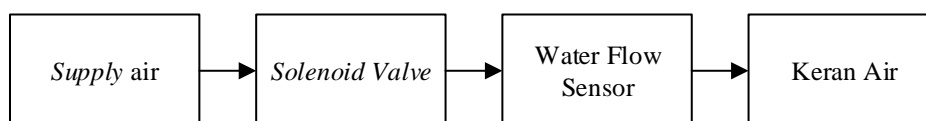


Gambar 4. 17 Skematik Pengujian *Solenoid Valve*

Keterangan :

1. *Power Supply* 12VDC
2. *Step Down*
3. *Arduino Uno*
4. *NodeMCU*
5. *Water Flow Sensor*
6. *LCD* 20 X 4 12C
7. *RTC*
8. *Relay*
9. *Solenoid Valve*

Untuk diagram blok pengujian *Solenoid valve* dapat dilihat pada gambar 4. 18 berikut:



Gambar 4. 18 Diagram Blok Pengujian *Solenoid Valve*

Berikut merupakan list program *Solenoid valve*.

```
void loop() {
    a=Firebase.getString("valve");
    Serial.println (a);
    if (a=="1"){
        Serial.println("valve ON");
        digitalWrite(valve,LOW); //Device1 is ON
    }
    else {
        Serial.println("valve OFF");
        digitalWrite(valve,HIGH); //Device1 if OFF
    }
    //counfigurasi millis
    unsigned long currentMillis = millis(); //Baca waktumillissaatini
    if(currentMillis - previousMillis>= interval){
        //Update data previousMillis
        previousMillis = currentMillis;
        //pembacaan data dari arduino (hasil kiriman data)
        //baca data serial
        String data = "";
        while(NodeMCU.available()>0)
        {
            data += char (NodeMCU.read());
        }
        //Pembuangan spasi datanya
        data.trim();
        //uji data
        if(data != "")
        {
            //parsing datanya (pecah data)
            int index = 0;
            for(int i=0; i<= data.length(); i++)
            {
                char delimiter = '#' ;
                if(data[i] != delimiter)
                    arrData[index] += data[i] ;
                else
                    index++; //variable index bertambah 1
            }
            //pemastian data yang dikirim lengkap
            if(index == 3)
            {
                //tampilkan nilai sensor ke serial monitor
                Serial.println("tanggal : " + arrData[0]); //data waktu
                Serial.println("harga :Rp " + arrData[1]); //data harga
                Serial.println("volume: " + arrData[2]); //data volume
                Serial.println("debit : " + arrData[3]); //data debit
            }
            tanggal = arrData[0].toFloat();
            harga = arrData[1].toFloat();
            volume = arrData[2].toFloat();
            debit = arrData[3].toFloat();
            arrData[0] = "";
            arrData[1] = "";
            arrData[2] = "";
            arrData[3] = "";
        }
    }
}
```

```

    }
    //minta data ke arduino
    NodeMCU.println("Ya");
  }
  Firebase.setString("tanggal", String(tanggal));
  Firebase.setString("harga", String(harga));
  Firebase.setString("volume", String(volume));
  Firebase.setString("debit", String(debit));
  delay(2000);
}

```

Hasil pengujian *Solenoid Valve* tercatat pada tabel 4. 2 berikut.

Tabel 4. 2 Hasil Pengujian *Solenoid Valve*

<i>Input</i>	<i>Level Tekanan Aliran Air</i>	<i>Output</i>
Air	<i>Low</i>	Baik
Air	<i>Medium</i>	Baik
Air	<i>High</i>	Baik

Berikut merupakan gambar hasil pengujian dari *Solenoid Valve*.



Gambar 4. 19 Hasil Pengujian *Solenoid Valve Level Low*



Gambar 4. 20 Hasil Pengujian *Solenoid Valve Level Medium*

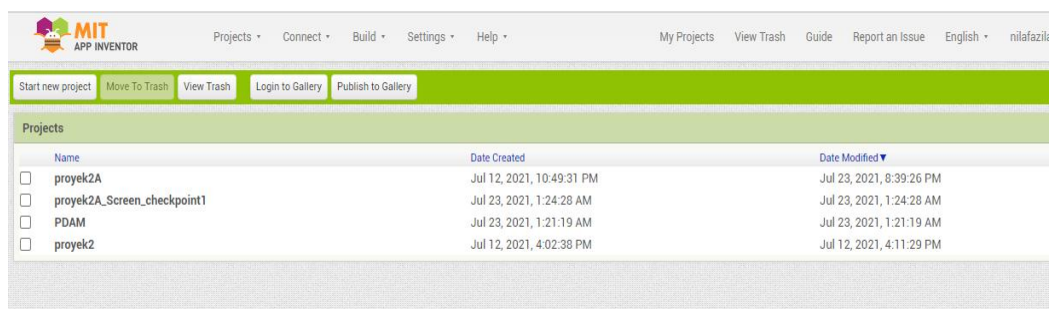


Gambar 4. 21 Hasil Pengujian *Solenoid Valve Level High*

Berdasarkan hasil dari data pengujian diatas, diketahui bahwa pada saat *solenoid valve* dialiri air bertekanan rendah, *solenoid valve* masih bisa mengalir air dengan lancar sehingga dapat disimpulkan bahwa *solenoid valve* dapat bekerja dengan baik di segala *level* tekanan.

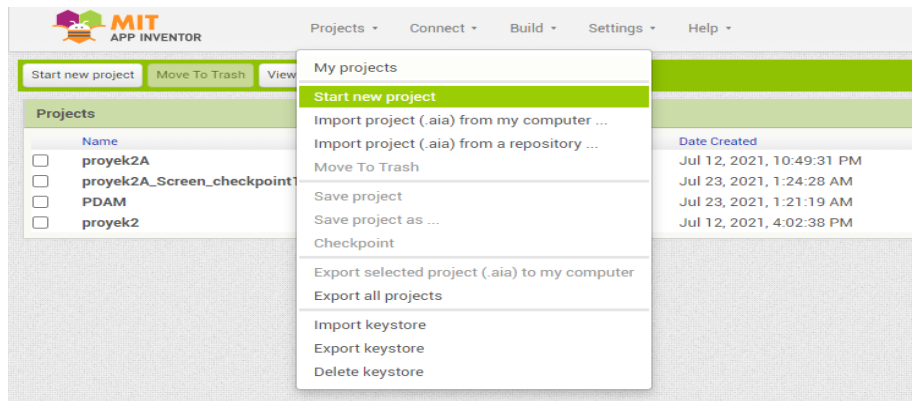
4.8 Perancangan *Software* Aplikasi

Perancangan *software* aplikasi pada *smartphone* dibuat menggunakan aplikasi Mit App Inventor. Untuk masuk ke aplikasi Mit App Inventor *input*/pilih email terlebih dahulu. Setelah masuk ke aplikasi Mit App Inventor tampilan utama yang dapat dilihat adalah seperti pada gambar 4. 22 berikut.

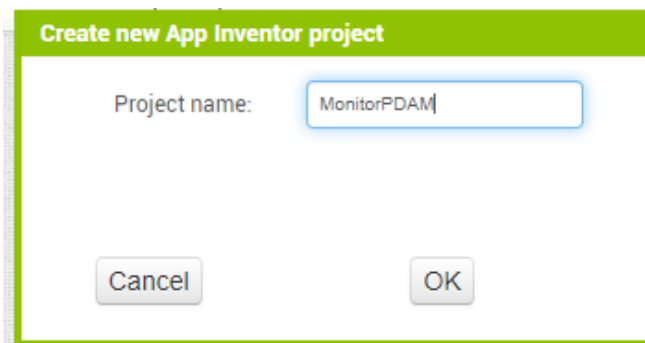


Gambar 4. 22 Tampilan Awal *Mit App Inventor*

Setelah masuk ke aplikasi *Mit App Inventor*, pilih *Projects* lalu pilih *Start new projects* kemudian tulis nama proyek yang akan dibuat pada kolom *Project Name*. Selanjutnya mulai desain aplikasi dengan memilih perangkat yang akan digunakan. Berikut merupakan proses pengaturan pembuatan proyek baru :

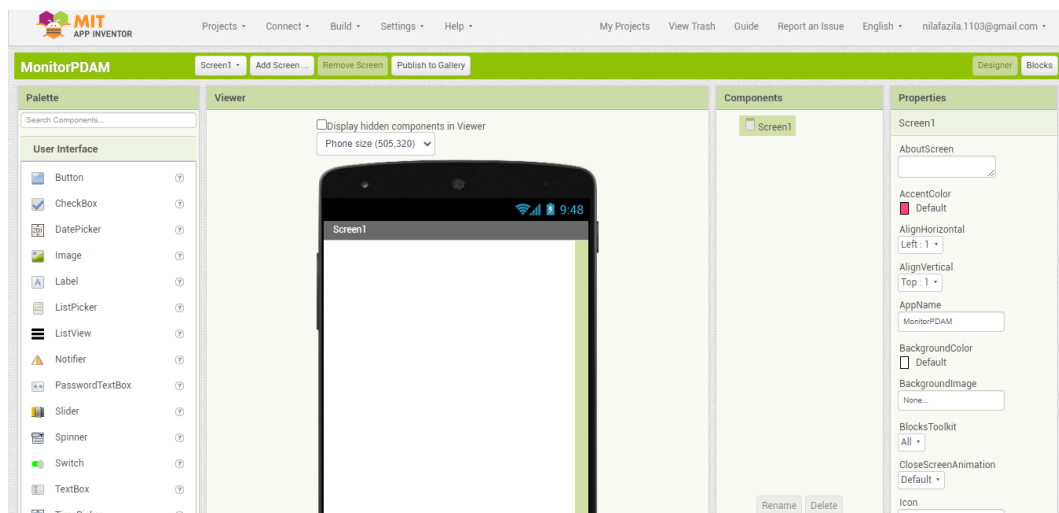


Gambar 4. 23 Tampilan Membuat Proyek Baru Pada Mit App Inventor



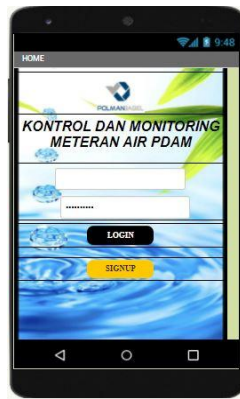
Gambar 4. 24 Tampilan Membuat Nama Proyek Pada Mit App Inventor

Untuk tampilan awal membuat desain aplikasi dapat dilihat pada gambar 4. 25 berikut.

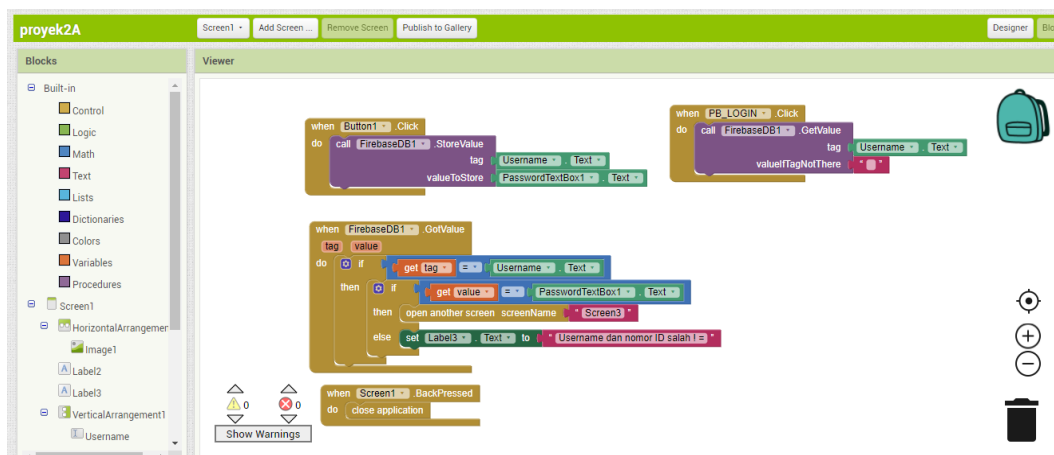


Gambar 4. 25 Tampilan awal membuat desain aplikasi pada Mit App Inventor

Membuat desain aplikasi di Mit App Inventor yaitu pada bagian *designer* desain dengan memilih perangkat-perangkat yang berkaitan dengan aplikasi yang akan dibuat pada menu *user interface* yang terletak pada bagian sebelah kiri tampilan Mit App Inventor yang kemudian ditarik masuk dan di atur sedemikian rupa di dalam *display phone* seperti pada gambar diatas. Untuk menu *propertise* disebelah kanan tampilan Mit App Inventor berfungsi untuk mengatur perangkat-perangkat yang digunakan. Setelah desain dibuat maka selanjutnya adalah membuat program atau *coding* dari setiap screen pada bagian *blocks*. Contoh gambar desain aplikasi yang telah dibuat dan siap untuk digunakan dapat dilihat pada gambar berikut.



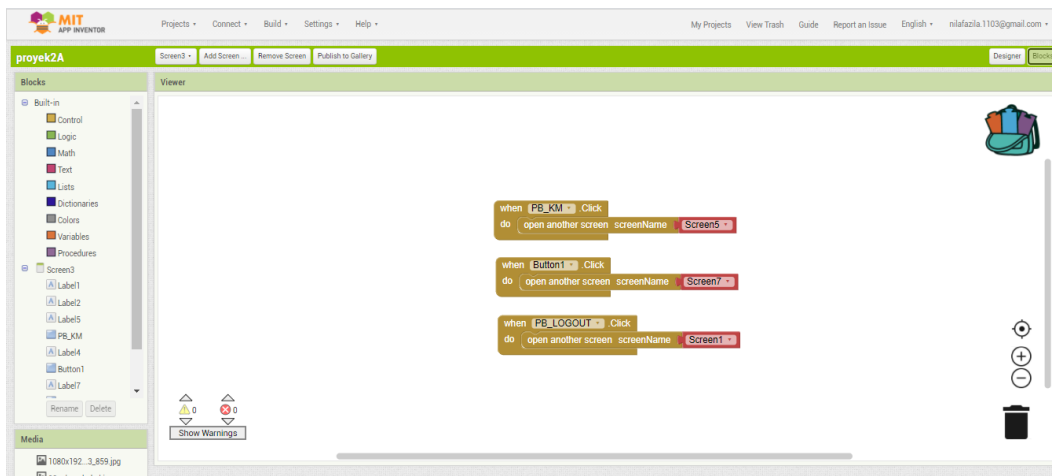
Gambar 4. 26 Tampilan Desain Aplikasi Screen 1



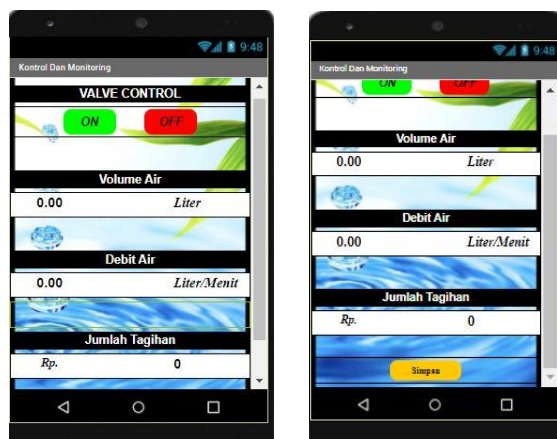
Gambar 4. 27 Tampilan Blocks Program Screen 1



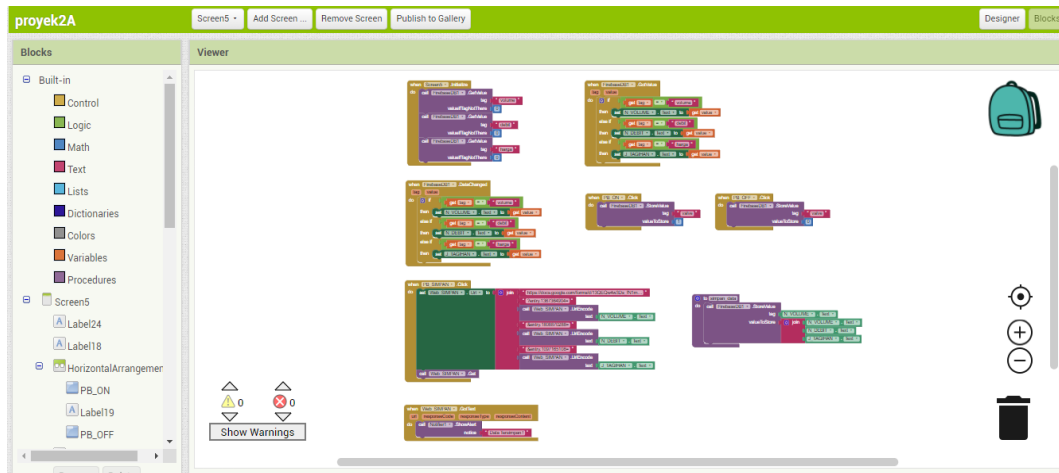
Gambar 4. 30 Tampilan Desain Aplikasi Screen 2



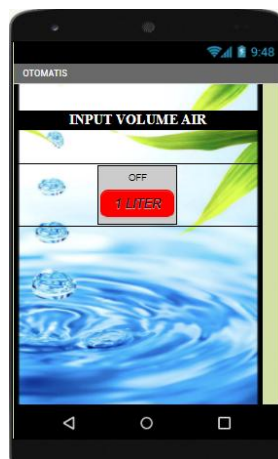
Gambar 4. 29 Tampilan Blocks Program Aplikasi Screen 2



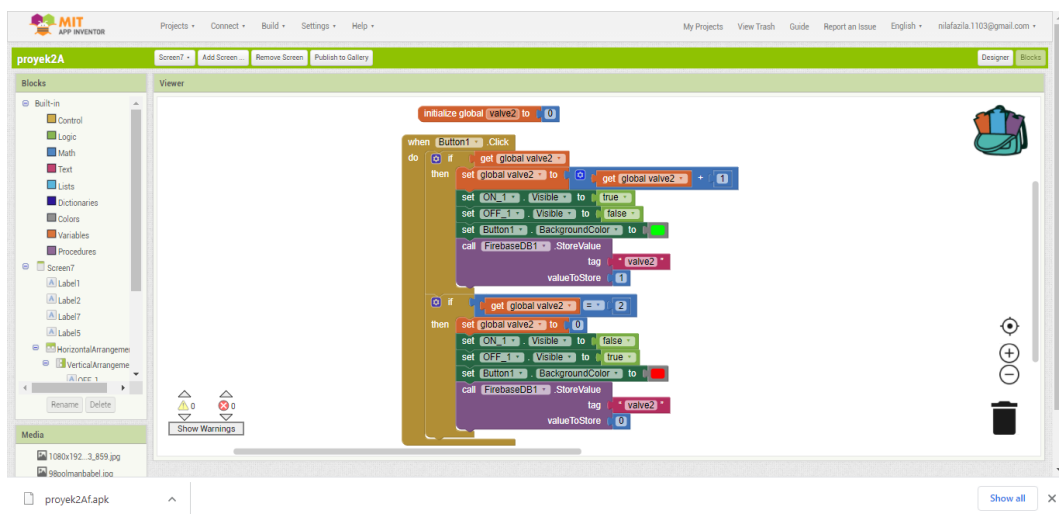
Gambar 4. 28 Tampilan Desain Aplikasi Screen 3



Gambar 4. 31 Tampilan Blocks Program Aplikasi *Screen 3*



Gambar 4. 32 Tampilan Desain Aplikasi *Screen 4*

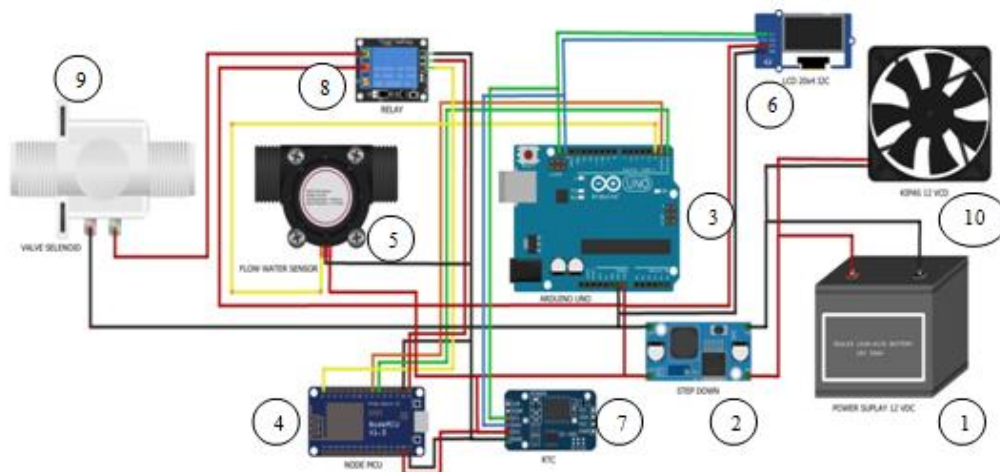


Gambar 4. 33 Tampilan Blocks Program Aplikasi *Screen 4*

Setelah pengaturan tampilan aplikasi selesai, maka untuk mengaktifkan aplikasi ini pada *smartphone* pilih *build* pada menu dan pilih (*app provide QR code for.apk*). Untuk menghubungkan alat dengan aplikasi ini maka salin alamat *firebaseURL* dan *firebase Token* ke menu *proportise* pada perangkat *firebaseDB1*. Data yang dikirimkan oleh NodeMCU ke *firebase* akan ditampilkan pada aplikasi ini sesuai dengan pin virtual yang telah ditentukan sebelumnya sehingga aplikasi pada *smartphone* bekerja sesuai dengan yang diinginkan.

4.9 Uji Coba dan Pengambilan Data Sistem Kontrol dan *Monitoring* Meteran Air PDAM

Pengujian alat dapat dilakukan setelah semua komponen sistem kontrol dan *monitoring* selesai dan terpasang pada konstruksi. Pengujian dilakukan dengan pengambilan data pada sistem. Pengambilan data Sistem Kontrol dan *Monitoring* dilakukan untuk mengetahui besar volume serta debit air yang keluar. Pengujian dilakukan untuk melihat keakuratan dari sistem meteran air PDAM yang telah dibuat. Pengambilan data ini dilakukan yaitu dengan cara pengambilan data pengukuran volume air yang terukur pada gelas ukur dan yang ditampilkan pada aplikasi yang telah dibuat pada *smartphone* dan data yang ditampilkan pada LCD kemudian dilakukan perbandingan. Rangkaian elektrik keseluruhan sistem dalam pengambilan data pengukuran volume air dapat dilihat pada gambar 4. 34 berikut.



Gambar 4. 34 Skema Elektrik Pengukuran Volume dan Debit Air

Keterangan :

1. *Power Supply* 12VDC
2. *Step Down*
3. Arduino Uno
4. NodeMCU
5. *Water Flow Sensor*
6. LCD 20 X 4 12C
7. RTC
8. Relay
9. *Solenoid Valve*
10. Kipas 12 VDC



Gambar 4. 35 Pengujian Pengukuran Volume Dan Debit Air

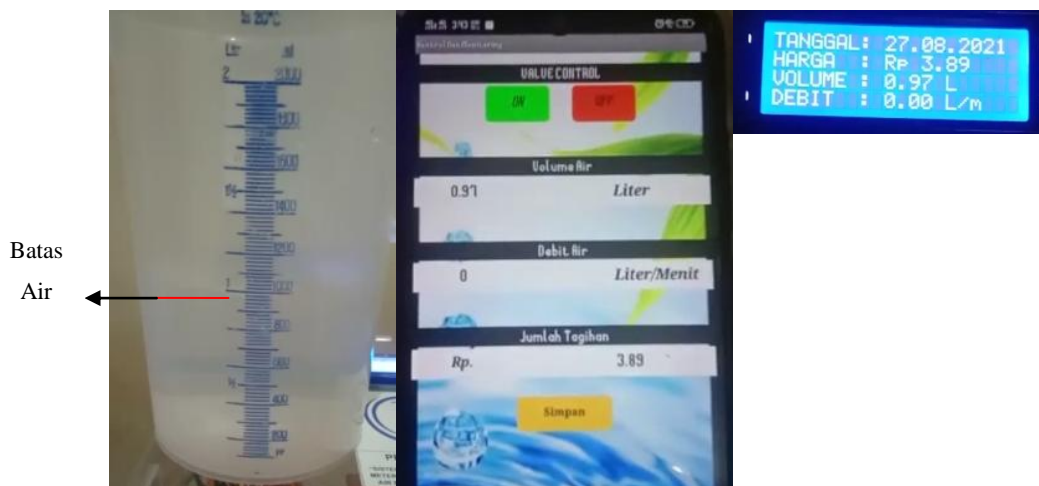
4.9.1 Pengujian Kontrol Otomatis dan *Monitoring* Meteran Air PDAM

Kontrol otomatis volume air merupakan suatu proses pengontrolan pengeluaran air berdasarkan *input* yang diberikan melalui aplikasi *smartphone*. Yang mana, jika volume air yang keluar telah sesuai dengan besar volume air yang di *input* melalui aplikasi *smartphone* maka secara otomatis keran air akan mati. Dengan kontrol ini maka pelanggan dapat menetapkan target pengeluaran volume air sesuai dengan keinginan. Apabila target volume air yang diinginkan telah terpenuhi maka keran air secara otomatis akan terkunci. Untuk gambar proses pengujian dapat dilihat pada gambar 4. 36 dibawah ini.



Gambar 4. 36 Proses Pengujian Kontrol Otomatis

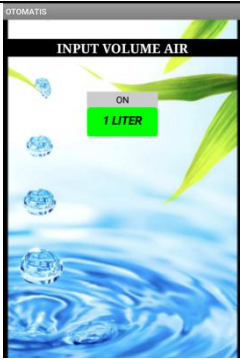
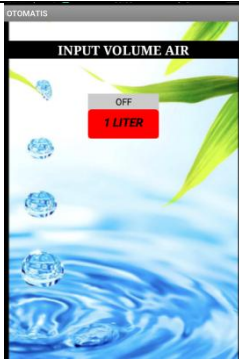
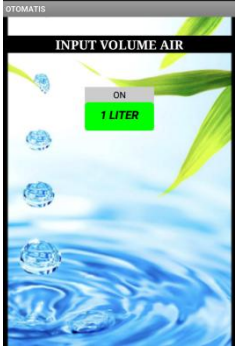
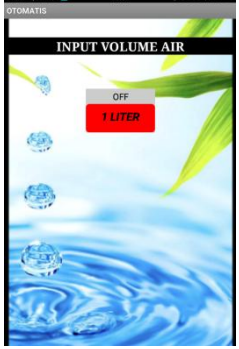
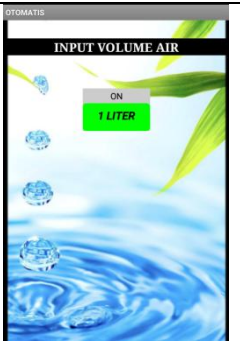

Pengujian ini bertujuan untuk mengetahui tingkat keakuratan nilai hasil pengukuran volume air ketika menggunakan kontrol otomatis apabila diberikan *input* melalui aplikasi *smartphone*. Untuk salah satu hasil pengukuran volume air dapat dilihat pada gambar 4. 37 berikut ini.



Gambar 4. 37 Hasil Pengukuran Volume Air

Hasil pengujian tercatat pada tabel 4. 3 berikut.

Tabel 4. 3 Hasil Pengujian Kontrol Otomatis Dan Monitoring Meteran Air PDAM

<i>Input</i> (<i>Smartphone</i>)	Kondisi Ketika	<i>Output</i>		
Slider <i>Input</i> Volume Air	Volume Air Telah Terpenuhi	Pembandi ng Gelas Ukur (Liter)	Pembacaan Volume Air pada LCD (Liter)	Pembacaan Volume Air pada <i>Smartphone</i> (Liter)
		0,99	0,97	0,97
		1,08	1,04	1,04
		1,10	1,11	1,11

Rumus persentase *error* , dimana:

$$\text{Persentase } error = \left| \frac{\text{Alat Ukur} - \text{sensor}}{\text{Alat Ukur}} \right| \times 100\% \dots\dots\dots(9)$$

Alat Ukur = Pengukuran Gelas Ukur

Sensor = Hasil pembacaan sensor pada LCD/*Smartphone*

Perhitungan persentase *error* pada tabel 4.6 antara *input* dengan *output* yang didapat.

1. Persentase *error* = $\left| \frac{0,99-0,97}{0,99} \right| \times 100\% = 2,02 \%$

2. Persentase *error* = $\left| \frac{1,06-1,04}{1,06} \right| \times 100\% = 1,88 \%$

3. Persentase *error* = $\left| \frac{1,10-1,11}{1,10} \right| \times 100\% = 0,90 \%$









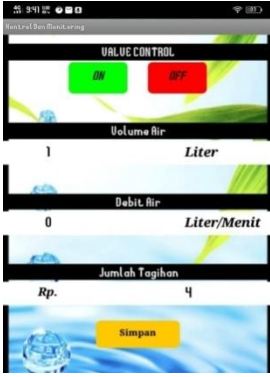
Rata-rata persentase *error* = 1,60 %

Dari hasil pengujian kontrol otomatis volume air dengan *input* volume air sebesar 1 liter melalui aplikasi *smartphone* diperoleh *output* hasil pengukuran volume air pada gelas ukur dan yang terbaca pada LCD/Aplikasi dengan rata-rata *persentase error* sebesar 1,60% untuk 3 kali percobaan yang mana pada saat volume air yang keluar telah terpenuhi sesuai dengan besar volume air yang di *input* melalui aplikasi *smartphone* maka secara otomatis keran air akan mati (off).

4.9.2 Pengujian Kontrol Manual dan *Monitoring* Meteran Air PDAM

Kontrol manual volume air merupakan suatu proses pengontrolan pengeluaran air berdasarkan *input* yang diberikan melalui aplikasi *smartphone*. Yang mana, pelanggan dapat mengaktifkan *solenoid valve* secara manual sesuai dengan keinginan. Pengujian ini bertujuan untuk mengetahui tingkat keakuratan nilai hasil pengukuran volume air ketika menggunakan kontrol manual apabila diberikan *input* perintah melalui aplikasi *smartphone*. Hasil pengujian ini tercatat pada tabel 4. 4 berikut:

Tabel 4. 4 Hasil Pengujian Kontrol Manual dan *Monitoring* Meteran Air PDAM

No	Input		Output	
	Volume Air dengan tekanan	Pembanding Volume Air Pada Gelas Ukur (Liter)	Pembacaan Volume Air Pada LCD (Liter)	Pembacaan Volume Air Pada Smartphone (Liter)
1.		Volume = 0,50		Volume = 0,50 
2.		Volume = 0,50		Volume = 0,50 
3.		Volume = 1,00		Volume = 1,00 

4.

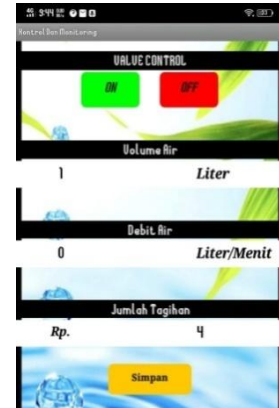
10,0 Liter
dengan
tekanan 0,05
Mpa



Volume = 1,00



Volume = 1,00



5.



Volume = 1,50



Volume = 1,51



6.



Volume = 1,50



Volume = 1,50



7.



Volume = 2,00



Volume = 2,00



8.



Volume = 2,00

Volume = 2,01



Jumlah	10,00	10.02	10,02
--------	-------	-------	-------

Rumus persentase *error* , dimana:

$$\text{Persentase } error = \left| \frac{\text{Alat Ukur} - \text{sensor}}{\text{Alat Ukur}} \right| \times 100\% \dots\dots\dots(10)$$

Alat Ukur = Pengukuran Gelas Ukur

Sensor = Hasil pembacaan sensor pada LCD/*Smartphone*

Perhitungan persentase *error* pada tabel 4.6 antara *input* dengan *output* yang didapat.

1. Persentase *error* = $\left| \frac{0,50-0,50}{0,50} \right| \times 100\% = 0,00\%$
2. Persentase *error* = $\left| \frac{0,50-0,50}{0,50} \right| \times 100\% = 0,00\%$
3. Persentase *error* = $\left| \frac{1,00-1,00}{1,00} \right| \times 100\% = 0,00\%$
4. Persentase *error* = $\left| \frac{1,00-1,00}{1,00} \right| \times 100\% = 0,00\%$
5. Persentase *error* = $\left| \frac{1,50-1,51}{1,50} \right| \times 100\% = 0,67 \%$
6. Persentase *error* = $\left| \frac{1,50-1,50}{1,50} \right| \times 100\% = 0,00\%$
7. Persentase *error* = $\left| \frac{2,00-2,00}{2,00} \right| \times 100\% = 0,00\%$
8. Persentase *error* = $\left| \frac{2,00-2,01}{2,00} \right| \times 100\% = 0,50\%$

Rata-rata persentase *error* = 0,15%

Dari hasil pengujian pengukuran volume air dengan input volume air sebesar 10 liter pada wadah diperoleh hasil pengukuran volume air (output) pada gelas ukur dan yang terbaca pada LCD/Aplikasi dengan jumlah output volume air keseluruhan sebesar 10,02 liter untuk 8 kali percobaan maka didapatkan selisih antara input dan output tersebut yaitu sebesar 0,02 liter dengan rata-rata *persentase error* sebesar 0,15% sehingga dipastikan sistem sudah berjalan dengan baik.

Data pengukuran yang tersimpan di *google sheet* :

	A	B	C	D
1	Timestamp	Volume Air (Liter)	Debit Air (Liter/Menit)	Jumlah Tagihan (Rp.)
2	01/08/2021 8:20:15	1.53		0 0.01
3	02/08/2021 8:27:45	1.32		0 5.27
4	02/08/2021 9:39:10	0	0	0
5	02/08/2021 9:39:10	0	0	0
6	02/08/2021 9:39:10	0	0	0
7	02/08/2021 9:39:10	0	0	0
8	02/08/2021 9:39:11	0	0	0
9	02/08/2021 9:39:11	0	0	0
10	02/08/2021 9:39:11	0	0	0
11	02/08/2021 9:39:11	0	0	0
12	02/08/2021 9:39:11	0	0	0
13	02/08/2021 9:39:15	0	0	0
14	02/08/2021 9:39:16	0	0	0
15	02/08/2021 9:39:16	0	0	0
16	02/08/2021 9:39:16	0	0	0
17	03/08/2021 4:04:27	1.08		0 4.33
18	03/08/2021 4:06:49	2.14		0 8.55
19	03/08/2021 4:09:46	1.16		0 4.65
20	03/08/2021 4:09:46	1.16		0 4.65
21	03/08/2021 4:09:47	1.16		0 4.65
22	03/08/2021 4:11:55	2.31		0 9.22
23	03/08/2021 4:13:05	2.31		0 9.22
24	03/08/2021 4:16:16	1.25		0 4.99
25	03/08/2021 4:16:16	1.25		0 4.99
26	03/08/2021 4:16:16	1.25		0 4.99
27	03/08/2021 4:17:25	2.42		0 9.68
28	03/08/2021 4:19:27	1.58		0 6.3
29	03/08/2021 4:20:09	1.58		0 6.3
30	03/08/2021 4:21:23	0		0
31	03/08/2021 4:22:25	2.32		0 9.28
32	03/08/2021 4:22:26	2.32		0 9.28
33	03/08/2021 4:26:09	1.1		0 4.41
34	03/08/2021 4:28:26	1.21		0 4.86
35	03/08/2021 7:07:19	0		0
36				

Gambar 4. 38 Data pengukuran yang tersimpan

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian alat yang berjudul "Sistem Kontrol Dan *Monitoring* Meteran Air Perusahaan Air Minum (PDAM) Berbasis *IoT*" diperoleh kesimpulan sebagai berikut:

1. Dari hasil pengujian kontrol otomatis dan *monitoring* meteran air PDAM berbasis *IoT* dengan input volume air sebesar 1 liter berdasarkan *input* yang diberikan melalui aplikasi *smartphone* diperoleh *output* hasil pengukuran volume air pada gelas ukur dan yang terbaca pada LCD/Aplikasi dengan rata-rata *persentase error* sebesar 1,60% untuk 3 kali percobaan yang mana pada saat volume air yang keluar telah terpenuhi sesuai dengan besar volume air yang di *input* melalui aplikasi *smartphone* maka secara otomatis keran air akan mati (off).
2. Dari hasil pengujian kontrol manual dan *monitoring* meteran air PDAM berbasis *IoT* dengan input volume air sebesar 10 liter pada wadah diperoleh hasil pengukuran volume air (*output*) pada gelas ukur dan yang terbaca pada LCD/Aplikasi dengan jumlah output volume air keseluruhan sebesar 10,02 liter untuk 8 kali percobaan maka didapatkan selisih antara *input* dan *output* tersebut yaitu sebesar 0,02 liter dengan rata-rata *persentase error* sebesar 0,15%.

5.2 Saran

Apabila alat ini akan dikembangkan lebih lanjut, fungsi yang perlu diperbaiki dan ditambahkan antara lain :

1. Pengembangan aplikasi sehingga dapat menyimpan data pengukuran langsung didalam aplikasi tersebut.
2. Menambahkan sistem simpan data pengukuran otomatis apabila terjadi pemadaman listrik secara tiba-tiba.

DAFTAR PUSTAKA

- [1] Pradipta, Ika W., & Harsanto.S. 2020. Statistik Air Bersih 2014-2019. Jakarta. Badan Pusat Statistik.
- [2] Herdianto, Sebayang, A. N., & Nasution, D. 2017. SISTEM MONITORING DATA METERAN AIR PELANGGAN PDAM MENGGUNAKAN MIKROKONTROLER ATMEGA 16 . Jurnal *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)* , 35-39.
- [3] Purwanti, H.D.,& Martono & Afrizal, F. 2020. Prototype Sistem Monitoring Penggunaan Air Berbasis Internet Of Things Pada PDAM Tirta Benteng Kota Tangerang. Jurnal Ilmiah ICIT (Innovative Creative and. Information Technology), 82-93.
- [4] Naim, N.N.,& Taufiqurrahman, I. 2020. Sistem Monitoring Penggunaan Debit Air Konsumen Di Perusahaan Daerah Air Minum Secara Real Time Berbasis Arduino Uno.Jurnal Of Energy And Electrical Engineering(JEEE), 31-39.
- [5] Diharja, R.,&Setiawan, B.,&Handini, W. 2021. Rancang Bangun Sistem Dan Kontrol Penggunaan Air PDAM Secara Realtime Berbasis Wemos Dan IoT. Jurnal Teknik Komputer AMIK BSI, 11-18.
- [6] Arif, Y.C., & Suhariningsih., & Hakim, L. 2011.Rancang Bangun Sistem Pengaturan Penerangan.Makalah.
- [7] Hanur,M.F.A. 2016. Rancang Bangun Alat Pemutus KWH Meter Sebagai Proteksi Berbasis Arduino. Skripsi. Fakultas Teknik Universitas Jember.
- [8] I. Puji, Desember 2014. [Online]. Available: https://www.academia.edu/29519499/Laporan_Pratikum_Buck_Converter. [Accessed 10 Juni 2021].
- [9] Available: <https://www.sandielektronik.com/2016/01/buck-converter.html>. [Accessed 10 Juni 2021].

- [10] Ardiansyah. 2016. Sistem Monitoring Air Layak Konsumsi Berbasis Arduino (Studi Kasus PDAM Patalassang). Skripsi. Fakultas sains dan Teknologi Universitas Islam Negeri Alauddin Makassar.
- [11] Aryanto, F.A. 2019. Rancang Bangun Kontrol Dan Monitoring Meteran Air PDAM Berbasis *Internet of Things*. Skripsi. Fakultas Teknologi dan Informatika Institut Bisnis dan Informatika STIKOM Surabaya.
- [12] Suyatno. 2015. Alat Penakar Volume Air Berbasis Mikrokontroler. Skripsi. Fakultas Sains Dan Teknologi Universitas Sanata Dharma Yogyakarta.
- [13] Ghivena,I., &Munadi,R., &Sussi. 2020. Rancang Bangun Mikrokontroler Monitoring Penggunaan Air PDAM Berbasis IOTJurnal *EPROC*, 991-998.
- [14] Ghia, 2018, "BAB II LANDASAN TEORI". [Online]. Available:https://Repository.bsi.ac.id/index.php/unduh/item/1598/File_10-BAB-II-Landasan-Teori.pdf. [Accessed 10 Juni 2021].
- [15] Triady & Rocky. 2015. Prototipe Sistem Keran Air Otomatis Berbasis Sensor Flowmeter Pada Gedung Bertingkat. Jurnal Coding Sistem Komputer Untan, 25-34.
- [16] C. N. Tanuwijaya, 2018, "Binus University". [Online]. Available:<https://sis.binus.ac.id/2018/03/08/apakah-itu-IoT-internet-ofthings/>. [Accessed 14 Juni 2021].
- [17] Hasan, M.A. & Nasution, N., &Setiawan, D. 2017. Game Bola Tangkis Berbasis Android Menggunakan App Inventor. Jurnal Teknologi Informasi dan Komunikasi Digital Zone,161-162.

LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama : Adam Ramadhan
Tempat, Tanggal lahir : Lubuk Besar, 7 Januari 1999
Alamat Rumah : JL. Baru SD 3 Lubuk Besar
Telp :-
Hp : 082281977797
Email : adamramadhan903@gmail.com
Jenis Kelamin : Laki-Laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 3 Lubuk Besar	Lulus 2012
MTS Nurul Falah	Lulus 2015
SMA Negeri 1 Lubuk Besar	Lulus 2018
DIII Politeknik Manufaktur Negeri Bangka Belitung	Lulus 2021

3. Pendidikan Non Formal

-

Sungailiat, 04 September 2021

Adam Ramadhan

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama : Nila Fazila
Tempat, Tanggal lahir : Belinyu, 11 Maret 2000
Alamat Rumah : JL. Kartini Belinyu
Telp :-
Hp : 085758022733
Email : nilafazila.1103@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

MI AL-HUDA Belinyu	Lulus 2012
SMP Negeri 1 Belinyu	Lulus 2015
SMA Negeri 1 Belinyu	Lulus 2018
DIII Politeknik Manufaktur Negeri Bangka Belitung	Lulus 2021

3. Pendidikan Non Formal

-

Sungailiat, 04 September 2021



Nila Fazila

Lampiran 2 Program Keseluruhan

```
➤ PROGRAM FLOW WATER SENSOR
#include <Wire.h>
bytesensorInt = 0; // deklarasi flow water
byteflowsensor = 2; // deklarasi pin flow sensor
floatkonstanta = 6.4; // konstanta flow sensor
float debit; // inisialisasi debit
floatharga; // inisialisasiharga
float liter; // inisialisasi liter
float volume; // inisialisasi volume
unsigned long oldTime; // inisialisasioldTime
unsignedintflowmlt; // inisialisasiflowmlt
volatile byte pulseCount; // inisialisasipulseCount
//program flow sensor
if((millis() - oldTime) > 1000) {
detachInterrupt(sensorInt);
debit = ((1000.0 / (millis() - oldTime)) * pulseCount) /
konstanta;
oldTime = millis();
flowmlt = (debit / 60) * 1000;
volume += flowmlt;
liter = volume/1000;
harga = liter*4,42;
unsignedintfrac;

➤ PROGRAM RTC
#include <DS3231.h>
// setting pertama program RTC
//rtc.setDate(12, 7, 2021); //mensettingtanggal 12 juli
2021
//rtc.setDOW(MONDAY); //mensethari "Senin"
//rtc.setTime(12, 32,30); //menset jam 12:32:30
rtc.begin();
// program RTC
hari = rtc.getDOWStr();
waktu = rtc.getTimeStr();
tanggal = rtc.getDateStr();

➤ PROGRAM MENAMPILKAN NILAI DI LCD
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
// Menampilkan monitoring pada LCD
lcd.clear();
lcd.setCursor(0,0);
lcd.print("TANGGAL: ");
lcd.print(tanggal);
lcd.setCursor(0,1);
lcd.print("HARGA : Rp ");
lcd.print(harga);
lcd.setCursor(0,2);
lcd.print("VOLUME : ");
lcd.print(liter);
lcd.print(" L");
lcd.setCursor(0,3);
lcd.print("DEBIT : ");
```



```

lcd.print(debit);
lcd.print(" L/m");
delay (100);

```

➤ PROGRAM KOMUNIKASI ARDUINO KE NODEMCU

```

//bacapermintaandariNodeMCU
String minta = "";
//bacapermintaanNodeMCU
while(Serial.available(>0)
{
minta += char(Serial.read());
}
//pembuanganspasi data yang diterima
minta.trim();
//uji variable minta
if(minta == "Ya"){
//kirimdatanya
 kirimdata();
}
//kosongkan variable minta
minta = "";
delay(1000);
}
//Variable penampungan data yang akandikirimkeNodeMCU
String datakirim = String(tanggal) + "#" + String(harga) + "#"
+
String(liter) + "#" + String(debit);
//kirim data keNodeMCU
Serial.println(datakirim);
delay(1000);

```

➤ PROGRAM KOMUNIKASI NODEMCU KE ARDUINO

```

#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
// pin serial komunikasipadaNodeMCU
SoftwareSerialNodeMCU(12, 13); //deklarasi pin D6=12=rxdan
D7=13=txpada node mcu
unsigned long previousMillis = 0;
const long interval = 3000;
//variable array untuk data pemisah
String arrData[4];
//counfigurasimillis
unsigned long currentMillis = millis(); //Baca waktumillissaatini
if(currentMillis - previousMillis>= interval){
//Update data previousMillis
previousMillis = currentMillis;
//pembacaan data dariarduinouno (hasilkiriman data)
//baca data serial
String data = "";
while(NodeMCU.available(>0)
{
data += char (NodeMCU.read());
}
//Pembuanganspasidatanya
data.trim();

```

```

        //uji data
    if(data != "")
    {
        //parsing datanya(pecah data)
        int index = 0;
        for(inti=0; i<= data.length(); i++)
        {
            char delimiter = '#' ;
            if(data[i] != delimiter)
                arrData[index] += data[i] ;
            else
                index++; //variable index bertambah 1
        }
        //pemastian data yagdikirimlengkap
        if(index == 3)
        {
            ➤ PROGRAM MENAMPILKAN NILAI DI SERIAL MONITOR NODE MCU
                //tampilkan nilai sensor ke serial monitor
                Serial.println("tanggal : " + arrData[0]); //data waktu
                Serial.println("harga :Rp " + arrData[1]); //data harga
                Serial.println("volume: " + arrData[2]); //data volume
                Serial.println("debit : " + arrData[3]); //data debit
            }
            arrData[0] = "";
            arrData[1] = "";
            arrData[2] = "";
            arrData[3] = "";
        }
        //minta data ke arduino
        NodeMCU.println("Ya");
    }
}

➤ PROGRAM MENAMPILKAN NILAI KE FIREBASE
tanggal = arrData[0].toFloat();
harga = arrData[1].toFloat();
volume =arrData[2].toFloat();
debit = arrData[3].toFloat();
arrData[0] = "";
arrData[1] = "";
arrData[2] = "";
arrData[3] = "";
}
//minta data ke arduino
NodeMCU.println("Ya");
}
Firebase.setString("tanggal", String(tanggal));
Firebase.setString("harga", String(harga)) ;
Firebase.setString("volume", String(volume)) ;
Firebase.setString("debit", String(debit)) ;
delay(2000);\
}

➤ PROGRAM KONTROL RELAY FIREBASE
a=Firebase.getString("valve");

```

```

Serial.println (a);
if (a=="1"){
Serial.println("valve ON");
digitalWrite(valve,LOW); //Device1 is ON
//          return;
delay(5000);
}
else {
Serial.println("valve OFF");
digitalWrite(valve,HIGH); //Device1 if OFF
//          delay(2000);
}

    ➤ PROGRAM UTAMA ARDUINO UNO
// kepala library
#include <DS3231.h>
#include <Wire.h>
// library LCD
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
// prosedurpembacaan RTC
DS3231 rtc(SDA,SCL);
bytesensorInt = 0; // deklarasi flow water
byteflowsensor = 2; // deklarasi pin flow sensor
floatkonstanta = 6.4; // konstanta flow sensor
float debit; // inisialisasi debit
floatharga; // inisialisasiharga
float liter; // inisialisasi liter
float volume; // inisialisasi volume
unsigned long oldTime; // inisialisasioldTime
unsignedintflowmlt; // inisialisasiflowmlt
volatile byte pulseCount; // inisialisasipulseCount
// inisialisasi RTC
String hari;
String waktu;
String tanggal;

void setup() {
Serial.begin(9600);
rtc.begin();
// Menampilkanawal LCD 20x4
lcd.init();
lcd.backlight();
lcd.clear();
lcd.setCursor(1,0);
lcd.print("SISTEM KONTROL DAN");
lcd.setCursor(1,1);
lcd.print("MONITORING METERAN");
lcd.setCursor(1,2);
lcd.print("AIR(PDAM) BERBASIS");
lcd.setCursor(8,3);
lcd.print("IOT");
delay(7000);pinMode(flowsensor, INPUT);
digitalWrite(flowsensor, HIGH);
// setting pertama program RTC

```

```

        //rtc.setDate(12, 7, 2021);      //mensettingtanggal 12 juli
2021
        //rtc.setDOW(MONDAY);           //mensethari "Senin"
        //rtc.setTime(12, 32,30);       //menseset jam 12:32:30

pulseCount = 0;
debit      = 0.0;
flowmlt    = 0;
volume     = 0.0;
oldTime    = 0;
attachInterrupt(sensorInt, pulseCounter, FALLING);
}

void loop() {
//program flow sensor
if((millis() - oldTime) > 1000) {
detachInterrupt(sensorInt);
debit      = ((1000.0 / (millis() - oldTime)) * pulseCount) /
konstanta;
oldTime = millis();
flowmlt = (debit / 60) * 1000;
volume += flowmlt;
liter   = volume/1000;
harga   = liter*4,42;
unsignedintfrac;
// program RTC
hari     = rtc.getDOWStr();
waktu    = rtc.getTimeStr();
tanggal  = rtc.getDateStr();
// Menampilkan monitoring pada LCD
lcd.clear();
lcd.setCursor(0,0);
lcd.print("TANGGAL: ");
lcd.print(tanggal);

lcd.setCursor(0,1);
lcd.print("HARGA : Rp ");
lcd.print(harga);

lcd.setCursor(0,2);
lcd.print("VOLUME : ");
lcd.print(liter);
lcd.print(" L");

lcd.setCursor(0,3);
lcd.print("DEBIT : ");
lcd.print(debit);
lcd.print(" L/m");
delay (100);
pulseCount = 0;
attachInterrupt(sensorInt, pulseCounter, FALLING);
}
delay(2000);
}

void pulseCounter() {

```

```

// Increment the pulse counter
pulseCount++;
}

    ➤ PROGRAM KOMUNIKASI ARDUINO KE NODE
// kepala library
#include <DS3231.h>
#include <Wire.h>
// library LCD
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
// prosedurpembacaan RTC
DS3231 rtc(SDA,SCL);
bytesensorInt = 0; // deklarasi flow water
byteflowsensor = 2; // input pin flow sensor
floatkonstanta = 6.4; //konstanta flow sensor
float debit; // inisialisasi debit
floatharga; // inisialisasiharga
float liter; // inisialisasi liter
float volume; // inisialisasi volume
unsigned long oldTime; // inisialisasioldTime
unsignedintflowmlt; // inisialisasiflowmlt
volatile byte pulseCount; // inisialisasipulseCount
// inisialisasi RTC
String hari;
String waktu;
String tanggal;

void setup() {
Serial.begin(9600);
rtc.begin();
// Menampilkanawal LCD 20x4
lcd.init();
lcd.backlight();
lcd.clear();
lcd.setCursor(1,0);
lcd.print("SISTEM KONTROL DAN");
lcd.setCursor(1,1);
lcd.print("MONITORING METERAN");
lcd.setCursor(1,2);
lcd.print("AIR(PDAM) BERBASIS");
lcd.setCursor(8,3);
lcd.print("IOT");
delay(7000); delay(2000);
pinMode(flowsensor, INPUT);
digitalWrite(flowsensor, HIGH);

// setting pertama program RTC
//rtc.setDate(12, 7, 2021); //mensettingtanggal 12 juli
2021
//rtc.setDOW(MONDAY); //mensethari "Senin"
//rtc.setTime(12, 32,30); //menseset jam 12:32:30

pulseCount = 0;
debit = 0.0;
flowmlt = 0;

```

```

volume      = 0.0;
oldTime     = 0;
attachInterrupt(sensorInt, pulseCounter, FALLING);
}

void loop(){
LCD();
//bacapermintaandariNodeMCU
    String minta = "";

//bacapermintaanNodeMCU
while(Serial.available()>0)
    {
minta += char(Serial.read());
    }
//pembuanganspasi data yang diterima
minta.trim();

//uji variable minta
if(minta == "Ya"){
//kirimatanya
kirimdata();
    }
//kosongkan variable minta
minta = "";
delay(1000);
}
voidkirimdata() {
//program flow sensor
if((millis() - oldTime) > 1000) {
detachInterrupt(sensorInt);
debit      = ((1000.0 / (millis() - oldTime)) * pulseCount) /
konstanta;
oldTime = millis();
flowmlt = (debit / 60) * 1000;
volume += flowmlt;
liter   = volume/1000;
harga   = liter*4,42;
unsignedintfrac;
// program RTC
hari     = rtc.getDOWStr();
waktu    = rtc.getTimeStr();
tanggal  = rtc.getDateStr();

pulseCount = 0;
attachInterrupt(sensorInt, pulseCounter, FALLING);
    }
//Variable penampungan data yang akandikirimkeNodeMCU
    String datakirim = String(tanggal) + "#" + String(harga) + "#"
+
String(liter) + "#" + String(debit);

    //kirim data keNodeMCU
Serial.println(datakirim);
delay(1000);
}

```

```

void pulseCounter(){
// Increment the pulse counter
pulseCount++;
}

void LCD(){
//program flow sensor
if((millis() - oldTime) > 1000) {
detachInterrupt(sensorInt);
debit      = ((1000.0 / (millis() - oldTime)) * pulseCount) /
konstanta;
oldTime = millis();
flowmlt = (debit / 60) * 1000;
volume += flowmlt;
liter   = volume/1000;
harga   = liter*4,42;
unsignedintfrac;
// program RTC
hari     = rtc.getDOWStr();
waktu    = rtc.getTimeStr();
tanggal  = rtc.getDateStr();

pulseCount = 0;
attachInterrupt(sensorInt, pulseCounter, FALLING);
}
lcd.clear();
lcd.setCursor(0,0);
lcd.print("TANGGAL: ");
lcd.print(tanggal);

lcd.setCursor(0,1);
lcd.print("HARGA : Rp ");
lcd.print(harga);

lcd.setCursor(0,2);
lcd.print("VOLUME : ");
lcd.print(liter);
lcd.print(" L");

lcd.setCursor(0,3);
lcd.print("DEBIT : ");
lcd.print(debit);
lcd.print(" L/m");
delay (100);
}

```

➤ PROGRAM KOMUNIKASI NODEMCU KE ARDUINO

```

// library NodeMCU
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
// pin serial komunikasipadaNodeMCU
SoftwareSerialNodeMCU(12, 13); //deklarasi pin D6=12=rxdan
D7=13=txpada node mcu
unsigned long previousMillis = 0;

```

```

const long    interval      = 3000;
//variable array untuk data pemisah
String arrData[4];

void setup() {
Serial.begin(9600);
NodeMCU.begin(9600);
}

void loop(){
//counfigurasi millis
unsigned long currentMillis = millis(); //Baca waktumillissaatini
if(currentMillis - previousMillis>= interval){
//Update data previousMillis
previousMillis = currentMillis;

//pembacaan data dariarduinouno (hasiliriman data)
//baca data serial
    String data = "";
while(NodeMCU.available(>0)
    {
data += char (NodeMCU.read());
    }
    //Pembuanganspasidatanya
data.trim();

    //uji data
if(data != "")
    {
        //parsing datanya(pecah data)
int index = 0;
for(inti=0; i<= data.length(); i++)
    {
char delimiter = '#' ;
if(data[i] != delimiter)
arrData[index] += data[i] ;
else
index++; //variable index bertambah 1
    }

        //pemastian data yagdikirimlengkap
if(index == 3)
    {
        //tampilkanilai sensor ke serial monitor
Serial.println("tanggal : " + arrData[0]); //data waktu
Serial.println("harga :Rp " + arrData[1]); //data harga
Serial.println("volume: " + arrData[2]); //data volume
Serial.println("debit : " + arrData[3]); //data debit
    }
arrData[0] = "";
arrData[1] = "";
arrData[2] = "";
arrData[3] = "";
    }
    //minta data keardduinouno
NodeMCU.println("Ya");
}

```



```
}  
}
```

➤ PROGRAM NODE MCU KE FIREBASE

```
#include <SoftwareSerial.h>  
#include <ESP8266WiFi.h>  
#include <FirebaseArduino.h>  
SoftwareSerialNodeMCU(12, 13); //deklarasi pin D7=12=rx dan  
D6=13=tx pada node mcu  
// Set these to run example.  
#define FIREBASE_HOST "monitor-dcabb-default-rtdb.firebaseio.com"  
#define FIREBASE_AUTH "C48BVY4E27UaZq7QGSWmqGMAHJCe9BIBbMh0jL90"  
#define WIFI_SSID "Nilai"  
#define WIFI_PASSWORD "qwerty11"  
  
unsigned long previousMillis = 0;  
const long interval = 3000;  
//variable array untuk data pemisah  
String arrData[4];  
float tanggal, harga, volume, debit;  
int valve = D0;  
String a;  
  
void setup() {  
  Serial.begin(9600);  
  NodeMCU.begin(9600);  
  pinMode(D0, OUTPUT);  
  // connect to wifi.  
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);  
  Serial.print("connecting");  
  while (WiFi.status() != WL_CONNECTED)  
  {  
    Serial.print(".");  
    delay(500);  
  }  
  Serial.println("");  
  Serial.print("WiFi connected! ");  
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);  
}
```

```
void loop(){  
  a=Firebase.getString("valve");  
  Serial.println(a);  
  if (a=="1"){  
    Serial.println("valve ON");  
    digitalWrite(valve, LOW); //Device1 is ON  
  }  
  else {  
    Serial.println("valve OFF");  
    digitalWrite(valve, HIGH); //Device1 if OFF  
  }  
  //counfigurasi millis  
  unsigned long currentMillis = millis(); //Baca waktu millis saat ini  
  if(currentMillis - previousMillis >= interval){  
    //Update data previousMillis  
    previousMillis = currentMillis;  
    //pembacaan data dari arduino (hasil kiriman data)
```

```

//baca data serial
String data = "";
while(NodeMCU.available()>0)
{
data += char (NodeMCU.read());
}
//Pembuanganspasidatanya
data.trim();

//uji data
if(data != "")
{
//parsing datanya(pecah data)
int index = 0;
for(inti=0; i<= data.length(); i++)
{
char delimiter = '#' ;
if(data[i] != delimiter)
arrData[index] += data[i] ;
else
index++; //variable index bertambah 1
}

//pemastian data yagdikirimlengkap
if(index == 3)
{
//tampilkanilai sensor ke serial monitor
Serial.println("tanggal : " + arrData[0]); //data waktu
Serial.println("harga :Rp " + arrData[1]); //data harga
Serial.println("volume: " + arrData[2]); //data volume
Serial.println("debit : " + arrData[3]); //data debit
}
tanggal = arrData[0].toFloat();
harga = arrData[1].toFloat();
volume =arrData[2].toFloat();
debit = arrData[3].toFloat();
arrData[0] = "";
arrData[1] = "";
arrData[2] = "";
arrData[3] = "";
}
//minta data keardduinouno
NodeMCU.println("Ya");
}
Firebase.setString("tanggal", String(tanggal));
Firebase.setString("harga", String(harga)) ;
Firebase.setString("volume", String(volume)) ;
Firebase.setString("debit", String(debit)) ;
delay(2000);
}

```