PROTOTIPE METERAN AIR DIGITAL PRABAYAR BERBASIS ARDUINO UNO R3

TAUFIQ ALFARISI AKBAR
5115107222

Skripsi yang Ditulis untuk Memenuhi Sebagian Persyaratan dalam Mendapatkan Gelar Sarjana Pendidikan

PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI JAKARTA
2017

ABSTRAK

TAUFIQ ALFARISI AKBAR. **Prototipe Sistem Meteran Air Digital Prabayar Berbasis Arduino Uno R3**. Pembimbing Syufrijal dan Nur HanifahYuninda.

Penelitian ini bertujuan untuk menghasilkan meteran air dan memantau pemakaian air. Sistem ini menggunakan *keypad* 16 x 2 sebagai pengirim sinyal danArduino Uno sebagai penerimanya. Sistem ini dikendalikan oleh Arduino Uno dan dapat diakses juga dipantau melalui aplikasi komputer. Komputer akan mengirimkan data ke Arduino Uno berupa perintah menghidupkan dan mematikan alat, Arduino Uno akan memproses perintah dari komputer menjadi pengendalian hidup dan matinya *relay* distribusi listrik ke alat, sehingga meteran air dapat dikendalikan hidup dan matinya.

Metode yang digunakan dalam penelitian ini adalah metode penelitian eksperimen laboratorium, yaitu dengan membuat serta melakukan uji coba program arduino sebagai penerima dan *keypad* sebagai pengirimnya untuk diterapkan pada prototipe meteran air digital prabayar berbasis arduino uno r 3.

Hasil dari penelitian ini adalah aplikasi pengendali dan pemantau meteran air rumah dibuat untuk memonitor penggunaan air berdasarkan pengaturan pada aplikasi dikomputer.

Kesimpulan dari penilitian ini adalah untuk memudahkan proses pembacaan data hasil pemakaian air, dan mampu mengurangi *complain* para konsumen.

Kata Kunci : Meteran Air, Arduino Uno R3,dan Prabayar.

ABSTRACT

TAUFIQ ALFARISI AKBAR. **Prototype System Based Prepaid Water Meters Digital Arduino Uno R3.** InstructureSyufrijal and NurHanifahYuninda. .

This research aims to produce water meter and monitor water usage. This system uses the 16 x 2 keypad as the signal sender and the Aso Uno as the receiver. The system is controlled by Arduino Uno and can be accessed also monitored through computer applications. The computer sends data to Arduino Uno in the form of a turn on and off command, Arduino Uno will process commands from the computer to control the life and death of the electrical distribution relay to the device, so that the water meter can be controlled alive and dead.

The method used in this research is laboratory experimental research method, that is by making and testing the arduino program as receiver and keypad as the sender to be applied to prototype of digital meter of prepaid water based on arduino uno r 3.

The result of this research is the application of controller and monitoring of home water meter made to monitor water usage based on arrangement in computer application.

The conclusion of this research is to facilitate the process of reading the data of water use, and able to reduce the complain of the consumers.

Keywords : Water meter, Arduino Uno R3, and Prepaid.

HALAMAN PENGESAHAN

PROTOTYPE METERAN AIR DIGITAL PRABAYAR BERBASIS ARDUINO UNO R 3

TAUFIQ ALFARISI / 5115107222

PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
Syufrijal, ST., MT (Dosen Pembimbing 1)		16/08-17
Nur Hanifah Yuninda, ST., MT (Dosen Pembimbing 2)		14/08-17

PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
Drs. Muhammad Rif'an, MT (Ketua Penguji)		16/08-17
MassusSubekti, MT (Sekretaris)		17/08-17
Moch. Djaohar, M.Sc (Dosen Ahli)		17/08-17

Tanggal Lulus : 11 Agustus 2017

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri jakarta maupun di perguruan tinggi lain
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya sendiri dengan arahan dosen pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta,Agustus 2017

Yang membuat pernyataan

METERAI TEMPEL 6000

Taufiq Alfarisi A

5115107222

KATA PENGANTAR

Puji dan syukur saya panjatkan kehadirat Allah SWT yang telah memberikan rahmat, karunia, dan hidayahnya, sehingga saya dapat menyelesaikan skripsi dengan judul "Prototipe Meteran Air Digital Prabayar Berbasis Arduino Uno R3". Yang merupakan persyaratan untuk meraih gelar Sarjana Pendidikan Teknik Elektro pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.

Keterbatasan kemampuan saya dalam penelitian ini, menyebabkan saya sering menemukan kesulitan. Oleh karena itu, skripsi ini tidaklah dapat terwujud dengan baik tanpa adanya bimbingan, dorongan, saran-saran, dan bantuan dari berbagai pihak. Maka sehubungan dengan ini saya ingin menyampaikan ucapan terimakasih yang sebesar-besarnya kepada:

1. Massus Subekti,S.Pd., MT. selaku Ketua Program Studi Pendidikan Teknik Elektro, Fakultas Teknik, Universitas Negeri Jakarta.
2. Syufrijal, ST., MT.selaku dosen pembimbing pertama yang selalu membimbing dan memberi semangat hingga selesainya karya tulis skripsi ini.
3. Nur Hanifah Yuninda, ST., MT. selaku dosen kedua yang telah banyak membantu saya sehingga selesainya skripsi ini.

Terima kasih juga saya ucapkan kepada kedua orang tua, saudara-saudaraku, Habib Bahari, Reza Wijaya, M. Wahyudi Giyantara, Frintis Septa, Siti Hutami Indriastuti Adiningsih, serta orang yang mengasihiku yang senantiasa mendoakan dan memberikan semangat. Semoga segala kebaikan, keikhlasan, kesabaran, do'a, bantuan yang telah diberikan kepada saya sebagai peneliti akan mendapat balasan yang berlipat ganda dari Allah SWT. Amin.

Saya menyadari bahwa skripsi saya ini belum sempurna, untuk itu saya mohon maaf apabila terdapat kekurangan dan kesalahan baik dari isi maupun tulisan. Akhir kata, saya berharap semoga skripsi ini bermanfaat bagi yang membacanya.

Penulis

Taufiq Alfarisi Akbar

5115107222

DAFTAR ISI

Halaman

ABSTRAK i
ABSTRACT ii
LEMBAR PENGESAHAN.....iii
HALAMAN PERNYATAAN iv
KATA PENGANTAR.....v
DAFTAR ISI vi
DAFTAR TABEL ix
DAFTAR GAMBAR x
BIOGRAFI PENULIS.....xi
BAB I PENDAHULUAN
1.1. Latar Belakang 1
1.2. Identifikasi Masalah 4
1.3. Pembatasan Masalah 5
1.4. Perumusan Masalah.....5
1.5. Tujuan Penelitian.....6
1.6. Kegunaan Hasil Penelitian 6
BAB II KERANGKA TEORITIS DAN KERANGKA BERFIKIR
2.1. Kerangka Teoritis.....7
2.1.1. Pengertian Air.....7
2.1.2. Pengertian Meteran Air.....8
2.1.3. Alat Pencatat Penggunaan Air Saat ini 8
2.1.4. G 1/2 *waterflow sensor*.....10
2.1.5. LCD 16x2.....11
2.1.6. *Keypad* 4x4.....12
2.1.7. Mikrokontroler.....16
2.1.8. Fitur AVR Atmega328.....18
2.1.9. Konfigurasi Pin Atmega328.....21
2.1.9.1. *Port* B.....22
2.1.9.2. *Port* C.....23
2.1.9.3. *Port* D.....24

2.1.10. Arduino Uno.....26
2.1.10.1. *Power Supply*.....29
2.1.10.2.Memori.....30
2.1.10.3. *Input* dan *Output*.....30
2.1.11. Manfaat Kit Arduino Uno 31
2.1.12. Komunikasi Arduino Uno 32
2.1.13. Tutorial Bahasa Pemrograman Arduino.....32
1. Struktur.....33
2.1.14 *Solenoid Valve*……33
2.2. Kerangka Berfikir.....33
BAB III METODE PENELITIAN
3.1. Tempat dan Waktu Penelitian 37
3.2. Metode Penelitian.....37
3.3. Instrumen Penelitian.....38
3.4. Rancangan Penelitian 38
3.5. Desain Alat.....39
3.5.1. Alat 39
3.5.2. Bahan 41
3.5.3. Pembuatan Maket Alat 42
3.6. Prosedur Penelitian.....44
3.6.1. Perancangan Alat.....44
3.6.1.1 Rangkaian Catu Daya.....44
3.6.1.2. Arduino Uno Sebagai Pengontrol Rangkaian.....45
3.6.1.3. Relai Sebagai Kontak Penghubung Ke *Output*.....45
3.6.1.4. *Output*.....48
3.6.1.5. Alat-alat Pendukung.....49
3.7.1. Pemasangan dan Pengawatan.....49
3.7.1.1. Pemasangan *Keypad* pada Mikrokontroler Arduino.....49
3.7.1.2. Pemasangan LCD sebagai sistem monitoring.....50

3.7.3. Pembuatan Program.....51
3.7.3.1. *Flowchart* Alur Kerja Alat.....51
3.7.3.2. Cara Kerja Alat.....53
3.7.3.3. Program Inisialisasi Alat.....53
3.7.3.4. Konfigurasi Tabel I/O Pada Mikrokontroler.....54
3.8. Kriteria Pengujian Alat.....55
3.8.1. Pengujian LCD.....56
3.8.2. Pengujian Catu Daya.....58
3.8.3. Pengujian *Water Flow Sensor*.....59
3.8.4. Pengujian *Software*.....60
BAB IV HASIL PENELITIAN
1.1. Hasil Pengujian Sistem Meteran Air Digital Prabayar Berbasis Arduino Uno R 3.....61
4.1.1. Hasil Pengujian LCD.....61
4.1.2. Hasil Pengujian Catu Daya.....61
4.1.3. Hasil Pengujian *Water Flow* Sensor.....63
4.1.4. Hasil Pengujian *Software*.....64
4.2. Kelebihan dan Kekurangan Alat 64
4.2.1. Kelebihan Alat.....64
4.2.2. Kekurangan Alat.....65
BAB V KESIMPULAN DAN SARAN
5.1. Kesimpulan.....66
5.2. Saran 66
DAFTAR PUSTAKA 68
LAMPIRAN.....……...69

DAFTAR TABEL

Halaman

Tabel 2.1.	*Keypad* 4 x 4	13
Tabel 2.2.	Konfigurasi *Port* B	22
Tabel 2.3.	Konfigurasi *Port* C	23
Tabel 2.4.	Konfigurasi *Port* D	24
Tabel 2.5.	Spesifikasi Pin Arduino Uno	26
Tabel 3.1.	Konfigurasi Pin Mikrokontroler Atmega328P-PU dan Konfigurasi Pin Arduino Uno R3 pada Alat	52
Tabel 3.2.	Pengujian Catu Daya	56
Tabel 3.3.	PengujianDayapada*Solenoid Valve*dan*Waterflow*	57
Tabel 3.4.	Pengujian *Waterflow* Sensor	57
Tabel 3.5.	Pengujian *Software* ketika daya dimatikan/dihidupkan	61
Tabel 3.7.	Pengujian Software dengan membandingkan jumlah air per liter	61
Tabel 4.1.	Hasil Pengujian Catu Daya	62
Tabel 4.2.	Hasil Pengujian Daya pada *Solenoid Valve* dan *Waterflow*	62
Tabel 4.3.	Hasil Pengujian *Water Flow* Sensor	63
Tabel 4.4.	Hasil Pengujian *Software*	64
Tabel 4.5.	Hasil Pengujian *Software* ketika daya di matikan/di hidupkan	65
Tabel 4.6	Hasil Pengujian Software dengan membandingkan jumlah per liter	65

DAFTAR GAMBAR

Halaman

Gambar 2.1. *Water Flow* Sensor 11

Gambar 2.2. LCD 16 X 2 11

Gambar 2.3. Rangkaian Dasar *Keypad* 4 X 4 13

Gambar 2.4. Bentuk Fisik *Keypad* 16

Gambar 2.5. Arsitektur Atmega328 20

Gambar 2.6. Konfigurasi Pin Atmega328 21

Gambar 2.7. *Board* Arduino Uno 27

Gambar 2.8. Prinsip Kerja *Solenoid Valve* 34

Gambar 2.9 Diagram Blok Alat 35

Gambar 3.1. Diagram Blok Tahapan Pembuatan 37

Gambar 3.2. Sketsa Maket Alat 41

Gambar 3.3. Skematik Arduino Uno R3 44

Gambar 3.4. Skematik Sistem Meteran Air Digital Prabayar 45

Gambar 3.5. Relai Sebagai Kontak Penghubung ke *Output* 46

Gambar 3.6. *Keypad* 4 X 4 dan Arduino Uno 48

Gambar 3.7. Pemasangan LCD pada Arduino 49

Gambar 3.8. *Flow Chart* pada Sistem 50

BIOGRAFI

Taufiq Alfarisi Akbar. Lahir di Bekasi pada tanggal 19 April 1992. Anak pertama dari pasangan Pribadi dan Pondong Suparmi. Peneliti memulai riwayat pendidikan di TK Amanah pada tahun 1997 - 1998, kemudian melanjutkan ke SDI LPI At-taufiq Cempaka Putih, Jakarta, Pada tahun 1998 - 2004, selanjutnya melanjutkan pendidikan ke SMPIT Al-multazam Kuningan, Jawa Barat pada tahun 2004 – 2007, setelah lulus melanjutkan pendidikan di SMAI PB.Soedirman 2 Bekasi pada tahun 2007 – 2010. Lulus pada jenjang sekolah, peneliti melanjutkan pendidikan ke jenjang perkuliahan di Universitas Negeri Jakarta dengan mengambil Program Studi Pendidikan Teknik Elektro Program S1 Jurusan Teknik Elektro pada tahun 2010 hingga sekarang.

Peneliti memiliki pengalaman Praktik Kerja Lapangan (PKL) di Kementrian Lingkungan Hidup dan Kehutanan daerah gatot subroto Palmerah Jakarta dan Praktik Keterampilan Mengajar (PKM) di Sekolah Menengah Kejuruan (SMK) Karya Guna2 Bekasi. Peneliti menghabiskan waktu kuliahnya untuk berwirausaha dan mempelajari sistem instalasi. Peneliti mengambil judul skripsi sistem meteran air digital prabayar berbasis arduino uno r 3..

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sumber air merupakan karunia Tuhan Yang Maha Esa yang memberikan manfaat untuk mewujudkan kesejahteraan bagi seluruh rakyat Indonesia dalam segala bidang. Sejalan dengan pasal 33 ayat (3) Undang - Undang Dasar Negara Republik Indonesia tahun 1945, bahwa sumber air dikuasai oleh negara dan dipergunakan untuk sebesar-besarnya kepentingan rakyat.

Atas penguasaan sumber air tersebut, negara menjamin hak setiap orang untuk mendapatkan air bagi pemenuhan kebutuhan pokok sehari-hari. Pemerintah menyediakan PDAM (Perusahaan Daerah Air Minum) untuk memenuhi kebutuhan air dalam kehidupan rumah tangga. Penyediaan air minum PDAM menggunakan pengukuran debit aliran air untuk mengetahui jumlah laju aliran air yang melalui suatu penampang pipa.

Pengukuran debit aliran air tersebut diterapkan pada setiap rumah tangga yang menggunakan PDAM sehingga setiap rumah dipasang *water* meter atau disebut juga dengan meteran air yang mana alat tersebut digunakan untuk mengukur atau mencatat seberapa besar volume air yang telah digunakan untuk keperluan setiap rumah tangga.

Salah satu hal yang dikeluhkan masyarakat adalah monitoring pihak perusahaan terhadap alat pengukur debit air. Permasalahan yang sering

timbul adalah pencatatan debit air yang dilakukan secara manual dan biaya yang dibayar tidak sesuai dengan debit air yang didistribusikan. Oleh karena itu dibutuhkan alat yang dapat memberikan informasi data debit air kepada masyarakat dan perusahaan secara otomatis. Solusi dari *complain* konsumen tentang biaya yang tidak sesuai dengan debit air yang didistribusikan, salah satunya adalah aplikasi pembayaran dengan sistem prabayar. Sehingga masyarakat dapat mengontrol sendiri kebutuhannya terhadap air.

Sejalan dengan perkembangan teknologi masa kini yang berkembang pesat, dalam segi kualitas, dimensi, keakuratan, maupun penerapannya. Mulai yang bersifat manual menuju otomatis, dari yang analog menuju digital. Semua bidang saat ini sudah memanfaatkan teknologi modern, mulai bidang transportasi, informasi jasa, serta bidang-bidang lainnya. Berdasarkan hal tersebut, mendorong penulis untuk merancang sebuah sistem dimana dalam pengecekan meteran air untuk mengetahui jumlah pemakaian penggunaan air rumah tangga dapat dilihat langsung pada alat data *collector* yang dibawa oleh petugas PDAM tanpa harus melihat langsung nilai penggunaan air pada meteran air.

Sistem pencatat meteran air dimana sistem pencatatan masih harus melihat langsung meteran air, hanya saja petugas hanya mengidentifikasi sebuah *barcode* yang di dalamnya sudah ada *id* pelanggan dan *barcode* tersebut di foto dengan sebuah *handphone* yang memiliki program khusus, dan suatu sistem dimana pengambilan data penggunaan air PAM menggunakan infra *red*.

Era modernisasi ini ikut berimbas terhadap modernisasi alat, baik di industri kecil maupun di industri besar. Peralatan di sebuah industri yang dulunya digerakkan manual oleh manusia kini mulai terotomatisasi, yakni dikendalikan oleh mesin. Proses otomatisasi mesin dikenal dengan istilah sistem kontrol atau ada juga yang menyebut sistem pengendalian. Pentingnya mempelajari sistem pengendalian ini erat kaitannya dengan mengefisiensikan dan mengoptimalkan kerja mesin agar mampu kita atur sesuai dengan apa yang diharapkan.

Sistem pengendalian terbagi menjadi dua, yaitu sistem pengendalian manual dan sistem pengendalian otomatis. Sistem pengendalian manual dimana faktor manusia sangat dominan dalam aksi pengendalian yang dilakukan pada sistem tersebut. Sedangkan sistem pengendalian otomatis dimana faktor manusia tidak dominan dalam aksi pengendalian yang dilakukan pada sistem tersebut. Peran manusia digantikan oleh sistem kontroler yang telah diprogram secara otomatis sesuai fungsinya, sehingga bisa memerankan seperti yang dilakukan manusia.

Berdasarkan latar belakang masalah yang telah dijelaskan, penulis tertarik untuk menghasilkan sebuah proyek akhir dengan judul "Prototipe Meteran Air Digital Prabayar Berbasis Arduino Uno". Pada tugas akhir ini akan dirancang suatu teknologi monitoring distribusi air bersih untuk mengontrol pemakaian air secara digital dengan menggunakan *selenoid valve* sebagai pengganti motor DC dan saat melakukan pembayaran, pelanggan cukup memasukkan suatu kode melalui *keypad*.

Teknologi ini memonitoring debit air yang mengalir didalam pipa dengan *output* data digital. Konsumen dan pihak PDAM pun memiliki keuntungan dengan sistem prabayar ini.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka ada beberapa masalah yang dapat diidentifikasikan, yaitu :

1. Bagaimanakah mengatasi *complain* konsumen tentang biaya pembayaran air yang tidak sesuai dengan debit air yang didistribusikan?
2. Bagaimanakah menghasilkan Prototipe Meteran Air Digital Prabayar Berbasis Arduino Uno?
3. Bagaimanakah penggunaan teknologi *Solenoid Valve* sebagai sistem kontroler pada Prototipe Meteran Air Digital Prabayar Berbasis Arduino Uno?

1.3 Pembatasan Masalah

Dalam perencanaan dan pembuatan tugas akhir ini diberi batasan-batasan masalah sebagai berikut.

1. Tidak dibahas masalah penurunan rumus secara matematis dalam perancangan.
2. Debit air yang diukur adalah air yang mengalir di dalam pipa.
3. LCD hanya menampilkan debit air dan sisa pulsa pemakaian.
4. Tidak menganalisa keamanan sistem.
5. Sensor yang digunakan adalah sensor *Waterflow* .
6. Pengontrolan perangkat dilakukan oleh Arduino uno R3.
7. Catu daya berasal dari listrik PLN (220V AC).
8. Tidak ada daya tambahan ketika listrik PLN mati.
9. Pompa air tidak termasuk ke dalam perhitungan daya.

1.4 Perumusan Masalah

Berdasarkan identifikasi masalah dan batasan masalah, maka rumusan masalah yang dikemukakan adalah “Bagaimanakah menghasilkan Prototipe Meteran Air Digital Prabayar Berbasis Arduino Uno?”.

1.5 Tujuan Penelitian

Adapun tujuan penelitian ini adalah menghasilkan prototipe meteran air digital prabayar berbasis arduino uno dengan menggunakan perangkat *flow* meter digital untuk monitoring distribusi air bersih dari PDAM.

1.6. Kegunaan Penelitian

Adapun kegunaan dari penelitian ini adalah mempermudah banyak pihak, yaitu pihak konsumen dan pihak PDAM sebagai pendistributor air bersih. Di bawah ini adalah beberapa keuntungan yang didapatkan dari penelitian tentang meteran air digital prabayar berbasis arduino uno r3.

Salah satu keuntungan yang didapat oleh konsumen :

1. Dapat memonitoring debit air sesuai kebutuhan.
2. Mempermudah dalam membayar rekening air, tanpa harus membayar di kantor PDAM.
3. Konsumen dapat men-stok kode yang sudah dibeli karena kesibukan setiap konsumen berbeda.

Adapun keuntungan yang didapat oleh pihak PDAM, yaitu:

1. Tidak khawatir akan keluhan masyarakat tentang tagihan air.
2. Tidak adanya tunggakan yang dilakukan masyarakat.
3. Petugas tidak perlu lagi untuk mengecek meteran air ke setiap rumah.

BAB II

KERANGKA TEORITIS DAN KERANGKA BERPIKIR

2.1 Kerangka Teori

2.1.1 Pengertian Air

Air adalah unsur yang tidak dapat dipisahkan dari kehidupan manusia, bahkan dapat dipastikan tanpa pengembangan sumber daya air secara konsisten peradaban manusia tidak akan mencapai tingkat yang dinikmati sampai saat ini. Oleh karena itu, pengembangan dan pengolahan sumber daya air merupakan dasar peradaban manusia. Salah satu faktor penting penggunaan air dalam kehidupan sehari-hari adalah untuk kebutuhan air minum. Manusia tidak dapat hidup tanpa air, lebih dari 70% tubuh manusia terdiri dari air. Kekurangan air tubuh 1 persen dapat menimbulkan rasa haus dan gangguan *mood*; kekurangan air tubuh 2 sampai 3 persen dapat meningkatkan suhu tubuh, rasa haus, dan gangguan stamina; kekurangan air tubuh 4 persen dapat menurunkan kemampuan fisik hingga 25 persen; dan apabila kadar air dalam tubuhnya berkurang sampai 7 persen seseorang bisa pingsan hingga menyebabkan kematian. Kurang minum juga dapat mengakibatkan sejumlah penyakit, antara lain gangguan ginjal dan infeksi saluran kemih (Hardiansyah, 2010).

Sebagian besar penduduk di Indonesia masih menggunakan air sumur sebagai sumber air bersih untuk memenuhi kebutuhan hidupnya sehari-hari. Dengan bertambahnya aktivitas dan jumlah penduduk, maka jumlah air bersih yang diperlukan manusia akan semakin meningkat. Secara global kuantitas

sumber daya tanah dan air relatif tetap, sedangkan kualitasnya makin hari makin menurun.

2.1.2 Pengertian Meteran Air

Alat untuk mengukur banyaknya aliran air secara terus menerus melalui sistem kerja peralatan yang dilengkapi dengan unit sensor, unit penghitung, dan unit indikator pengukur untuk menyatakan volume air yang lewat. Sedangkan badan meter air merupakan bagian utama yang ditengahnya merupakan ruang untuk menempatkan alat hitung dan mempunyai saluran masuk dan saluran keluar pada sisi yang berlawanan. (SNI 2547:2008)

2.1.3 Alat Pencatat Penggunaan PDAM

Bagi pelanggan PDAM, istilah 'sistem tembak' pencatatan meteran sudah tidak asing lagi. Istilah ini timbul karena petugas pencatat biasanya tidak datang ke rumah pelanggan melihat langsung angka di meteran airnya. Mereka hanya 'main tembak' dari jauh. Tidak heran jika kemudian banyak konsumen yang melayangkan keluhannya ke perusahaan. Tapi itu dulu, PDAM kemudian berbenah. Dengan memanfaatkan teknologi yang ada, sejak April 2012 lalu digunakanlah sistem *barcode* dalam pencatatan meteran. Penerapan sistem baru ini bertujuan untuk meningkatkan akurasi dan efisiensi pencatatan meter air.

Dengan menggunakan sistem baru, petugas pencatat hanya tinggal memfoto *barcode* dan angka yang tertera dalam meteran air, dengan menggunakan *handphone* berprogram khusus. *Barcode* tersebut sudah langsung teridentifikasi dengan *id* pelanggan. Keuntungannya, laporan bisa dilengkapi

dengan foto. Sistem ini juga akan mempermudah petugas di lapangan, akurasi data pencatatan, lebih terpercaya, dan mengefisiensikan waktu pencatatan.

Petugas juga diwajibkan datang ke lokasi. Jika tidak datang, mereka tidak akan bisa mencatat stan meteran. Diharapkan data hasil transaksi merupakan data aktual dari lokasi pelanggan langsung. Pencatat meteran PDAM yang berada disana tidak lagi memasukkan angka meteran dengan perkiraan pemakaian per bulan.

Dengan menggunakan teknologi *barcode* maka proses *input* data hanya pada saat pencatatan di lapangan saja. Sehingga pada saat *upload* data, data tersebut bisa diakses langsung sehingga tidak ada sistem *input* yang rawan kesalahan. Hanya saja, dalam perjalanannya banyak *barcode* yang hilang, baik karena dicabut, terhapus karena hujan dan beberapa penyebab lainnya. Sehingga yang efektif memanfaatkan sistem ini sekitar 60-70 persen pelanggan.

Untuk pembacaan meteran menggunakan dua *barcode*. Ada yang ukuran sebesar tempat menulis nama di undangan pernikahan, dan ada yang kecil. *Barcode* ukuran besar yang terdapat data pelanggan biasanya ditempel di pagar bagian depan rumah. Sementara yang ukuran kecil ditempel di meteran.

Ketika petugas pembaca meteran datang, mereka langsung *scan barcode* melalui *handphone*. Kemudian muncul nomor pelanggan, Ponsel meminta untuk memfoto angka meteran. Setelah ok, langsung disimpan. Penggunaan sistem ini bukanlah tanpa cela. Sebab dengan *barcode* ada

kemungkinan terjadi pembaca meteran salah obyek. Itu karena *barcode* sudah tidak ada. Jika ini terjadi, termasuk ketika pembacaan meteran tidak bisa dilakukan karena pelanggan tidak di rumah dan pintu pagar dalam kondisi terkunci, maka sistem taksasi pun diberlakukan.

Selain *barcode*, PDAM juga menerapkan penggunaan kamera dalam pencatatan meteran. Sistem ini setidaknya dapat membantu menekan jumlah *complain* pelanggan akibat tingginya biaya yang harus dibayar setiap bulannya. Karena jika mereka datang dan mempertanyakan penyebab melonjaknya pembayaran, pihak PDAM bisa langsung menunjukkan bukti dengan memperlihatkan hasil foto meteran yang dimasukkan petugas.

2.1.4 G 1/2 *Water Flow Sensor*

Sensor adalah suatu alat yang dapat mengukur atau mendeteksi kondisi sebenarnya di dunia nyata, seperti pergerakan, panas atau cahaya dan mengubah kondisi nyata tersebut ke dalam bentuk analog atau digital.

Sensor aliran air terbuat dari plastik katup, rotor air, dan sensor aula-efek. Ketika air mengalir melalui rotor, rotor gulungan, Kecepatannya berubah dengan tingkat yang berbeda aliran. *Hall-sensor* efek *output* pulsa yang sesuai *signal*. Gambar 2.1 di bawah ini merupakan bentuk fisik dari *waterflow sensor*.

Gambar 2.1 *waterflow sensor*
Sumber: http://www.hotmcu.com

2.1.5 LCD 16 x 2

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan di berbagai bidang misalnya alal-alat elektronik seperti televisi, kalkulator, atau pun layar komputer. Pada postingan aplikasi LCD yang digunakan ialah LCD dot matrik dengan jumlah karakter 16 x 2. LCD sangat berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat. Gambar 2.2 di bawah ini merupakan bentuk fisik LCD 16 x 2.

Gambar 2.2 LCD 16 x 2
Sumber : http://www.leselektronika.com

Modul LCD memiliki karakteristik sebagai berikut:

1. Terdapat 16 x 2 karakter huruf yang bisa ditampilkan.
2. Setiap huruf terdiri dari 5x7 dot-matrix *cursor*.
3. Terdapat 192 macam karakter.
4. Terdapat 80 x 8 bit *display* RAM (maksimal 80 karakter).
5. Memiliki kemampuan penulisan dengan 8 bit maupun dengan 4 bit.
6. Dibangun dengan osilator lokal.
7. Satu sumber tegangan 5 volt.
8. Otomatis *reset* saat tegangan dihidupkan.
9. Bekerja pada suhu 0 ^{o}C sampai 55 ^{o}C.

2.1.6 *Keypad* (Modul 4x4)

4 x 4 *Keypad Module* merupakan suatu modul *keypad* berukuran 4 kolom x 4 baris. Modul ini dapat difungsikan sebagai *input* dalam aplikasi seperti pengaman digital, absensi, pengendali kecepatan motor, robotik, dan sebagainya.

Spesifikasi *Hardware* adalah sebagai berikut :

1. Memiliki 16 tombol (fungsi tombol tergantung aplikasi).
2. Memiliki konfigurasi 4 baris dan 4 kolom.
3. Kompatibel penuh dengan DT-51^{TM} *Low Cost Series* dan DT-AVR *Low Cost Series*. Mendukung DT-51^{TM} *Minimum System* (Minsys) ver 3.0, DT-51^{TM} PetraFuz, DT-*BASIC Series*, dan lain-lain.

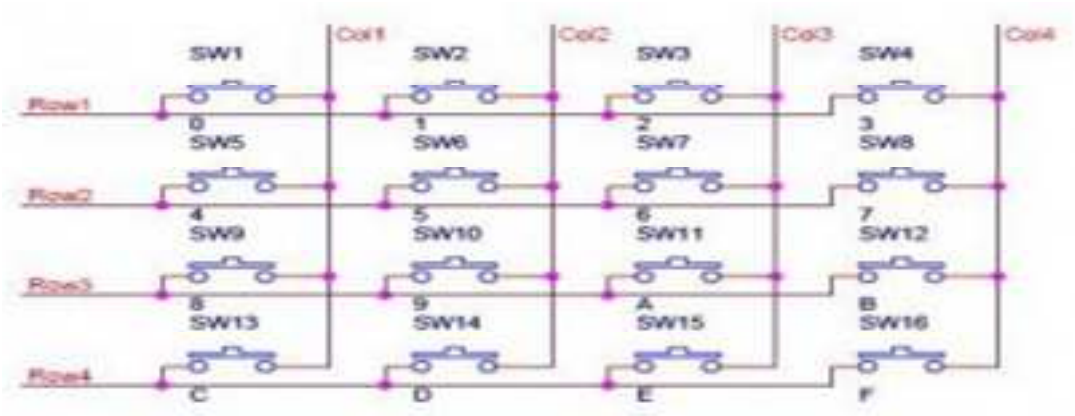
Tabel 2.1 di bawah ini merupakan bentuk kolom dan baris dari *keypad* 4x4

Tabel 2.1 *Keypad* 4X4

		Kolom			
		C1	C2	C3	C4
Baris	R1	1	2	3	A
	R2	4	5	6	B
	R3	7	8	9	C
	R4	*	0	#	D

Penggunaan *Keypad* dilakukan dengan cara menjadikan empat buah kolom sebagai *input*-an dan empat buah baris sebagai *output*-nya.

Di bawah ini adalah gambar 2.3 rangkaian dasar *keypad* yang berfungsi sebagai *input*-an.



Gambar 2.3 Rangkaian dasar *keypad* 4x4
Sumber : http://elektronika-dasar.web.id

Cara kerja rangkaian *keypad* :

a. Mengirimkan logika *Low* untuk kolom 1 (*Col*1) dan logika *HIGH* untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (*Row*1) akan *LOW* sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan *LOW* sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.
b. Mengirimkan logika *Low* untuk kolom 2 (*Col*2) dan logika *HIGH* untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (*Row*1) akan *LOW* sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan *LOW* sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.
c. Mengirimkan logika *Low* untuk kolom 3 (*Col* 3) dan logika *HIGH* untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (*Row*1) akan *LOW* sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan *LOW* sehingga data yang terbaca 1011, atau

tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.

d. Mengirimkan logika *Low* untuk kolom 4 (*Col* 4) dan logika *HIGH* untuk kolom yang lain kemudian membaca data baris, misal tombol SW1 ditekan maka data baris pertama (*Row*1) akan *LOW* sehingga data baris yang dibaca adalah 0111, atau tombol yang ditekan tombol SW5 maka data pada baris ke 2 akan *LOW* sehingga data yang terbaca 1011, atau tombol SW9 yang ditekan sehingga data yang terbaca 1101, atau tombol SW13 yang ditekan maka data yang dibaca adalah 1110 dan atau tidak ada tombol pada kolom pertama yang di tekan maka data pembacaan baris akan 1111.

Kemudian kembali ke semula, artinya program *looping* terus mendeteksi data kolom dan data baris, cara ini disebut *scanning* atau penyapuan *keypad* untuk mendapatkan saklar mana yang ditekan. Gambar 2.4 di bawah ini merupakan bentuk fisik dari *keypad* 4 x 4.

Gambar 2.4 Bentuk fisik dari *keypad* 4 x 4
Sumber : https://www.parallax.com/

2.1.7 Mikrokontroller

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah *chip*. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan *input/output*.

Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Sekedar contoh, bayangkan diri anda saat mulai belajar membaca dan menulis, ketika anda sudah bisa melakukan hal itu anda bisa membaca tulisan apapun baik buku, cerpen, artikel dan sebagainya, dan anda pun bisa pula menulis hal-hal sebaliknya. Begitu pula jika anda sudah mahir membaca dan menulis data maka anda dapat membuat program untuk membuat

suatu sistem pengaturan otomatis menggunakan mikrokontroler sesuai keinginan anda. Mikrokontroler merupakan komputer di dalam *chip* yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut "pengendali kecil" dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC, TTL, dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Mikrokonktroler digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti sistem kontrol mesin, *remote controls*, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat *input/output* yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan mikrokontroler ini maka :

1. Sistem elektronik akan menjadi lebih ringkas
2. Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi .
3. Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak

Agar sebuah mikrokontroler dapat berfungsi, maka mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimal paling tidak dibutuhkan sistem *clock* dan *reset*, walaupun pada beberapa mikrokontroler sudah menyediakan

sistem *clock internal*, sehingga tanpa rangkaian eksternal pun mikrokontroler sudah beroperasi.

Yang dimaksud dengan sistem minimal adalah sebuah rangkaian mikrokontroler yang sudah dapat digunakan untuk menjalankan sebuah aplikasi. Sebuah IC mikrokontroler tidak akan berarti bila hanya berdiri sendiri. Pada dasarnya sebuah sistem minimal mikrokontroler AVR memiliki prinsip yang sama.

2.1.8 Fitur AVR ATMega328

ATMega328 adalah mikrokontroller keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*).

Mikrokontroller ini memiliki beberapa fitur antara lain :

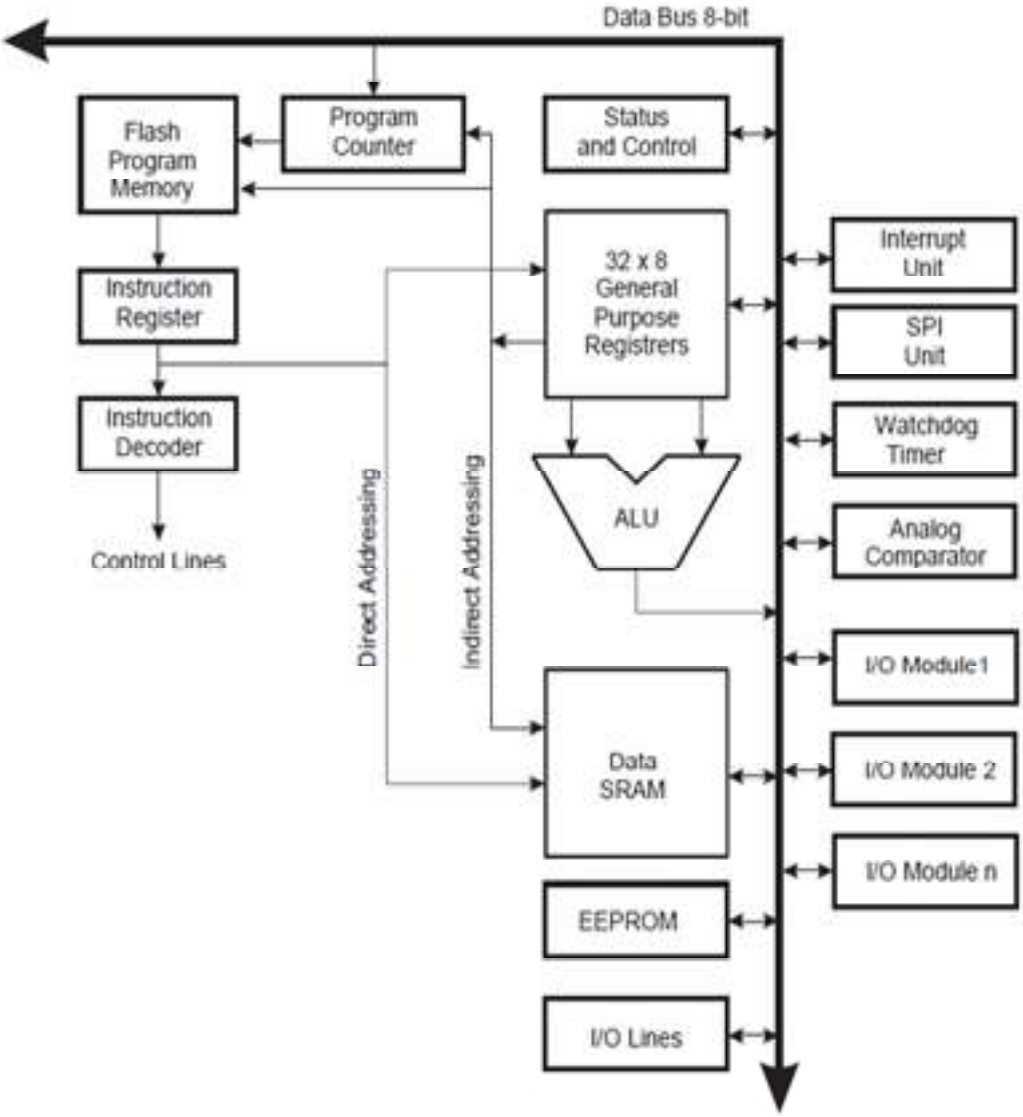
1. 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
2. 32 x 8-bit *register* serba guna.
3. Kecepatan mencapai 16 MIPS dengan *clock* 16 MHz.
4. 32 KB *Flash memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari *flash memori* sebagai *bootloader*.
5. Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1KB sebagai tempat penyimpanan data semi *permanent* karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.

6. Memiliki SRAM (*Static Random Access Memory*) sebesar 2KB.
7. Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya PWM (*Pulse Width Modulation*) *output*.
8. *Master / Slave* SPI *Serial interface*.

Mikrokontroller ATmega 328 memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism*. Instruksi - instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. Konsep inilah yang memungkinkan instruksi - instruksi dapat dieksekusi dalam setiap satu siklus *clock*. 32 x 8-bit *register* serba guna digunakan untuk mendukung operasi pada ALU (*Arithmatic Logic unit*) yang dapat dilakukan dalam satu siklus. 6 dari *register* serbaguna ini dapat digunakan sebagai 3 buah *register pointer* 16-bit pada mode pengalamatan tidak langsung untuk mengambil data pada ruang memori data.

Ketiga *register pointer* 16-bit ini disebut dengan *register* X (gabungan R26 dan R27), *register* Y (gabungan R28 dan R29), dan *register* Z (gabungan R30 dan R31). Hampir semua instruksi AVR memiliki format 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit. Selain *register* serba guna di atas, terdapat register lain yang terpetakan dengan teknik *memory mapped* I/O selebar 64 *byte*. Beberapa *register* ini digunakan untuk fungsi khusus antara lain sebagai *register control Timer*/ *Counter*, Interupsi, ADC, USART, SPI, EEPROM, dan fungsi I/O lainnya. *Register-register* ini menempati memori pada alamat 0x20h - 0x5Fh.

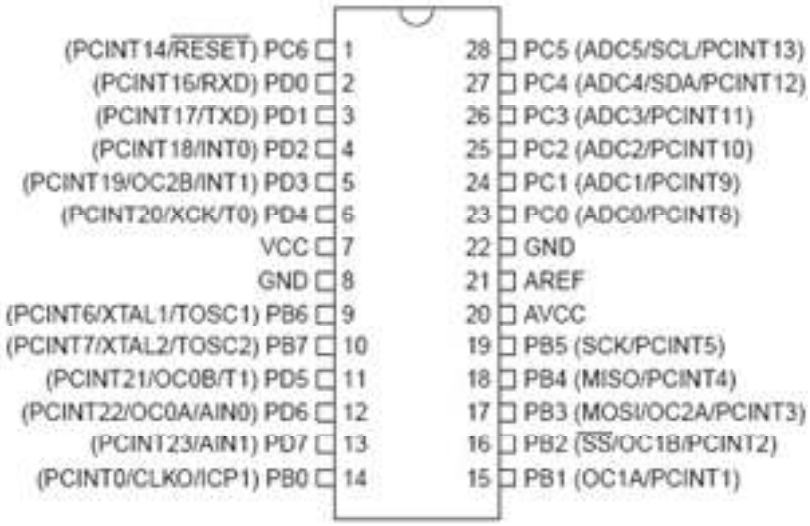
Gambar 2.5 di bawah ini adalah tampilan *architecture* ATmega 328 :



Gambar 2.5 Arsitektur Atmega328
Sumber : http://sains-mikrokontroler.blogspot.co.id/

2.1.9 Konfigurasi PIN ATMega328

ATMega328 merupakan mikrokontroler keluarga AVR 8 bit. Beberapa tipe mikrokontroler yang sama dengan ATMega328 ini antara lain ATMega8535, ATMega16, ATMega32, ATMega328, yang membedakan antara mikrokontroler antara lain adalah, ukuran memori, banyaknya GPIO (pin *input/output*), peripherial (USART, *timer*, *counter*, dll). Dari segi ukuran fisik, ATMega328 memiliki ukuran fisik lebih kecil dibandingkan dengan beberapa mikrokontroler diatas. Namun untuk segi memori dan peripherial lainnya ATMega328 tidak kalah dengan yang lainnya karena ukuran memori dan peripherialnya relatif sama dengan ATMega535, ATMega32, hanya saja jumlah GPIO lebih sedikit dibandingkan mikrokontroler diatas. Gambar 2.6 di bawah ini merupakan konfigurasi Pin ATMega328.



Gambar 2.6. Konfigurasi Pin Atmega 328
Sumber : http://trisnote.blogspot.co.id/

2.1.9.1 Port B

Port B merupakan jalur data 8 bit yang dapat difungsikan sebagai *input*/*output*. Konfigurasi *PORTB* dapat dilihat pada tabel 2.2 di bawah ini:

Tabel 2.2. Konfigurasi *Port* B

Port Pin	*Alternate Function*
PB7	XTAL2 (*Chip Clock Oscillator* pin 2) TOSC2 (*Timer Oscillator* pin 2)
PB6	XTAL1 (*Chip Clock Oscillator* pin 1 *or External clock input*). TOSC1 (*Timer Oscillator* pin 1)
PB5	SCK (SPI *Bus Master Clock Input*)
PB4	MISO (SPI *Bus Master Input/ Slave Output*)
PB3	MOSI (SPI *Bus Master Output/ Slave Input*) OC2 (*Timer/Counter* 2 *Output Compare Match Output*)
PB2	SS (SPI *Bus Master Slave select*) OC1B (*Timer/Counter* 1 *Output Compare Match* B *Output*)
PB1	OC1A (*Timer/Counter 1 Output Compare Match* A *Output*)
PB0	ICP1 (*Timer/Counter* 1 *Capture* 1 *Input Capture* Pin)

Selain itu *PORT*B juga dapat memiliki fungsi alternatif seperti di bawah ini.

a. ICP1 (PB0), berfungsi sebagai *Timer Counter* 1 *input capture* pin.
b. OC1A (PB1), OC1B (PB2) dan OC2 (PB3) dapat difungsikan sebagai keluaran PWM (*Pulse Width Modulation*).
c. MOSI (PB3), MISO (PB4), SCK (PB5), SS (PB2) merupakan jalur komunikasi SPI.

d. Selain itu pin ini juga berfungsi sebagai jalur pemograman *serial* (ISP).

e. TOSC1 (PB6) dan TOSC2 (PB7) dapat difungsikan sebagai sumber *clock external* untuk *timer*.

f. XTAL1 (PB6) dan XTAL2 (PB7) merupakan sumber *clock* utama mikrokontroler.

2.1.9.2 Port C

Port C merupakan jalur data 7 bit yang dapat difungsikan sebagai *input/output* digital. Konfigurasi *PORT*C dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2.3 Konfigurasi *Port* C

Port Pin	*Alternate Function*
PC6	*RESET* (*Reset* pin) PCINT14 (Pin *Change Interrupt* 14)
PC5	ADC5 (ADC *Input Channel* 5) SCL (2-*Wire Serial Bus Clock Line*) PCINT13 (Pin *Change Interrupt* 13)
PC4	ADC4 (ADC *Input Channel* 4) SDA (2-*Wire Serial Bus* Data *Input*/*Output Line*) PCINT12 (Pin *Change Interrupt* 12)
PC3	ADC3 (ADC *Input Channel* 3) PCINT11 (Pin *Change Interrupt* 11)
PC2	ADC2 (ADC *Input Channel* 2) PCINT10 (Pin *Change Interrupt* 10)
PC1	ADC1 (ADC *Input Channel* 1) PCINT9 (Pin *Change Interrupt* 9)
PC0	ADC0 (ADC *Input Channel* 0) PCINT8 (Pin *Change Interrupt* 8)

Fungsi alternatif *PORT*C antara lain sebagai berikut.

a. ADC6 *channel* (PC0,PC1,PC2,PC3,PC4,PC5) dengan resolusi sebesar 10 bit. ADC dapat kita gunakan untuk mengubah *input* yang berupa tegangan analog menjadi data digital.
b. I2C (SDA dan SDL) merupakan salah satu fitur yang terdapat pada *PORT*C. I2C digunakan untuk komunikasi dengan sensor atau *device* lain yang memiliki komunikasi data tipe I2C seperti sensor kompas, *accelerometer nunchuck*.

2.1.9.3 Port D

Port D merupakan jalur data 8 bit yang masing-masing pinnya juga dapat difungsikan sebagai *input*/*output*. Konfigurasi *PORT*D dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2.4 Konfigurasi *Port* D

Port Pin	*Alternate Function*
PD7	AIN1 (*Analog Comparator Negative Input*) PCINT23 (Pin *Change Interrupt* 23)
PD6	AIN0 (*Analog Comparator Positive Input*) OC0A (*Timer/Counter* 0 *Output Compare Match* A *Output*) PCINT22 (Pin *Change Interrupt* 22)
PD5	T1 (*Timer/Counter* 1 *External Counter Input*) OC0B (*Timer/Counter* 0 *Output Compare Match* B *Output*) PCINT21 (Pin *Change Interrupt* 21)
PD4	XCK (USART *External Clock Input*/*Output*) T0 (*Timer/Counter* 0 *External Counter Input*) PCINT20 (Pin *Change Interrupt* 20)
PD3	INT1 (*External Interrupt* 1 *Input*) OC2B (*Timer/Counter* 2 *Output Compare Match* B *Output*) PCINT19 (Pin *Change Interrupt* 19)

PD2	INT0 (*External Interrupt* 0 *Input*) PCINT18 (Pin *Change Interrupt* 18)
PD1	TXD (USART *Output* Pin) PCINT17 (Pin *Change Interrupt* 17)
PD0	RXD (USART *Input* Pin) PCINT16 (Pin *Change Interrupt* 16)

Sama seperti *Port* B dan *Port* C, *Port* D juga memiliki fungsi alternatif dibawah ini.

a. USART (TXD dan RXD) merupakan jalur data komunikasi serial dengan level sinyal TTL. Pin TXD berfungsi untuk mengirimkan data serial, sedangkan RXD kebalikannya yaitu sebagai pin yang berfungsi untuk menerima data serial.
b. *Interrupt* (INT0 dan INT1) merupakan pin dengan fungsi khusus sebagai interupsi *hardware*. Interupsi biasanya digunakan sebagai selaan dari program, misalkan pada saat program berjalan kemudian terjadi interupsi *hardware/software* maka program utama akan berhenti dan akan menjalankan program interupsi.
c. XCK dapat difungsikan sebagai sumber *clock external* untuk USART, namun kita juga dapat memanfaatkan *clock* dari CPU, sehingga tidak perlu membutuhkan *external clock.*
d. T0 dan T1 berfungsi sebagai masukan *counter external* untuk *timer* 1 dan *timer* 0.
e. AIN0 dan AIN1 keduanya merupakan masukan *input* untuk *analog comparator*.

2.1.10 Arduino Uno

Arduino adalah sebuah *board* mikrokontroller yang berbasis ATmega328. Arduino memiliki 14 pin *input/output* yang mana 6 pin dapat digunakan sebagai output PWM, 6 *analog input*, crystal osilator 16 MHz, koneksi USB, *jack power*, kepala ICSP, dan tombol *reset*. Arduino mampu men-*support* mikrokontroller; dapat dikoneksikan dengan komputer menggunakan kabel USB. Gambar 2.7. merupakan bentuk fisik dari Arduino uno r3 dan tabel 2.5 merupakan spesifikasi dari Arduino Uno.

Tabel 2.5. Spesifikasi Arduino Uno

Mikrokontroler	Atmega328
Operasi *Voltage*	5 V
Tegangan *input*	7 – 12 V (rekomendasi)
Batas tegangan *input*	6 – 20 V
I/O	14 pin (6 pin untuk PWM)
Arus	50 mA
Flash Memory	32 KB
Bootloader	SRAM 2 KB
EEPROM	1 KB
Kecepatan	16 MHz

Gambar 2.7 *Board* Arduino Uno R3
Sumber : http://*forefront*.io

Apakah arduino? Arduino adalah merupakan sebuah *board* minimum sistem mikrokontroler yang bersifat *open source*. Didalam rangkaian *board* arduino terdapat mikrokontroler AVR seri ATMega 328 yang merupakan produk dari Atmel.

Arduino memiliki kelebihan tersendiri dibanding *board* mikrokontroler yang lain selain bersifat *open source*, arduino juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C. Selain itu dalam *board* arduino sendiri sudah terdapat *loader* yang berupa USB sehingga memudahkan kita ketika kita memprogram mikrokontroler di dalam arduino. Sedangkan pada kebanyakan *board* mikrokontroler yang lain yang masih membutuhkan rangkaian *loader* terpisah untuk memasukkan program ketika kita memprogram mikrokontroler. *Port* USB tersebut selain untuk *loader* ketika memprogram, bisa juga difungsikan sebagai *port* komunikasi serial.

Arduino menyediakan 20 pin *I/O*, yang terdiri dari 6 pin *input* analog dan 14 pin digital *input/output*. Untuk 6 pin analog sendiri bisa juga difungsikan sebagai *output* digital jika diperlukan *output* digital tambahan selain 14 pin yang sudah tersedia. Untuk mengubah pin analog menjadi digital cukup mengubah konfigurasi pin pada program. Dalam *board* kita bisa lihat pin digital diberi keterangan 0-13, jadi untuk menggunakan pin analog menjadi *output* digital, pin analog yang pada keterangan *board* 0-5 kita ubah menjadi pin 14-19. dengan kata lain pin analog 0-5 berfungsi juga sebagai pin *output* digital 14-16.

Sifat *open source* arduino juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan *board* ini, karena dengan sifat *open source* komponen yang kita pakai tidak hanya tergantung pada satu merek, namun memungkinkan kita bisa memakai semua komponen yang ada dipasaran.

Bahasa pemrograman arduino merupakan bahasa C yang sudah disederhanakan syntax bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami mikrokontroler.

Berikut ini adalah konfigurasi dari arduino *duemilanove* 328 :

1. Mikrokontroler ATmega328.
2. Beroperasi pada tegangan 5V
3. Tegangan *input* (rekomendasi) 7 - 12V.
4. Batas tegangan *input* 6 - 20V.
5. Pin digital *input/output* 14 (6 mendukung *output* PWM).
6. Pin analog *input* 6
7. Arus pin per *input/output* 40 mA .

8. Arus untuk pin 3.3V adalah 50 mA .
9. *Flash Memory* 32 KB (ATmega328) yang mana 2 KB digunakan oleh *bootloader*
10. SRAM 2 KB (ATmega328).
11. EEPROM 1KB (ATmega328).
12. Kecepatan *clock* 16 MHz.

2.1.10.1 Power Supply

Arduino dapat diberikan *power* melalui koneksi USB atau *power supply*. *Power*-nya ditekan secara otomatis. *Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan mencolok *jack adaptor* pada koneksi *port input supply*. *Board* arduino dapat dioperasikan menggunakan *supply* dari luar sebesar 6 - 20 volt. Jika *supply* kurang dari 7V, kadangkala pin 5V akan menyuplai kurang dari 5 volt dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 V, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 sampai 12 volt. Penjelasan pada pin *power* adalah sebagai berikut :

a. Vin

   Tegangan *input* ke *board* arduino ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 volt dari koneksi USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan pin ini.

b. 5V

Regulasi *power supply* digunakan untuk *power* mikrokontroller dan komponen lainnya pada *board*. 5V dapat melalui Vin menggunakan regulator pada *board*, atau *supply* oleh USB atau *supply* regulasi 5V lainnya.

c. 3V3

Suplai 3.3 volt didapat oleh FTDI *chip* yang ada di *board*. Arus maksimumnya adalah 50mA

d. Pin *Ground*

berfungsi sebagai jalur *ground* pada arduino.

2.1.10.2 Memori

ATmega328 memiliki 32 KB *flash memori* untuk menyimpan kode, juga 2 KB yang digunakan untuk *bootloader*. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

2.1.10.3 Input dan Output

Setiap 14 pin digital pada arduino dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode*(), *digitalWrite*(), dan *digitalRead*(). *Input*/*output* dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau menerima maksimum 40 mA dan memiliki *internal pull-up resistor* (*disconnected* oleh *default*) 20-50 KOhms.

Beberapa pin memiliki fungsi sebagai berikut :

1. *Serial* : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB FTDI ke TTL *chip serial*.
2. *Interupt eksternal* : 2 dan 3. Pin ini dapat dikonfigurasikan untuk *trigger* sebuah interap pada *low value*, *rising* atau *falling edge*, atau perubahan nilai.
3. PWM : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit *output* PWM dengan fungsi *analogWrite*().
4. SPI : 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini men-*support* komunikasi SPI, yang mana masih mendukung *hardware*, yang tidak termasuk pada bahasa arduino.
5. LED : 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai *HIGH*, LED hidup, ketika pin *LOW*, LED mati.

2.1.11 Manfaat KIT Arduino Uno

Arduino Uno adalah KIT Elektronik atau papan rangkaian elektronik *open*. *Source* yang didalamnya terdapat komponen utama yaitu sebuah *Chip*. Mikrokontroller dengan jenis AVR dari Perusahaan Atmel. Arduino Uno adalah sebuah *board* Mikrokontroller yang berbaris Atmega 3288. Arduino Uno memiliki 4 PIN. *Input*/*output* yang mana 6 PIN dapat digunakan sebagai *output* PWM, 6 analog *Input*, *crystal* osilator 16 MHz, koneksi USB, *Jack Power*, Kepala ICSP, dan tombol *Reset*. Arduino Uno mampu men-*support* Mikrokontroller, dapat dikoneksikan dengan komputer menggunakan kabel USB.

2.1.12 Komunikasi Arduino Uno

Arduino uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran *board* ini komunikasi serial melalui USB dan muncul sebagai com *port virtual* untuk perangkat lunak pada komputer. *Firmware* '16U2 menggunakan USB *driver* standar COM, dan tidak ada *driver eksternal* yang dibutuhkan. Namun, pada Windows, *file*. Inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data tekstual sederhana yang akan dikirim ke dan dari papan Arduino. RX dan TX LED di papan akan berkedip ketika data sedang dikirim melalui *chip* USB-*to*-serial dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1). Sebuah perpustakaan *SoftwareSerial* memungkinkan untuk komunikasi serial pada setiap pin digital Uno itu. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI.

2.1.13 Tutorial Bahasa Pemograman Arduino

Arduino menggunakan pemrograman dengan bahasa C. Berikut ini adalah sedikit penjelasan yang ditujukan kepada anda yang hanya mempunyai sedikit pengalaman pemrograman dan membutuhkan penjelasan singkat mengenai karakter bahasa C dan *software* Arduino. Untuk penjelasan yang lebih mendalam, web Arduino.cc adalah sumber yang lengkap.

1. Struktur

Setiap program Arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada.

a. *void setup*( ) { }

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.

b. *void loop*( ) { }

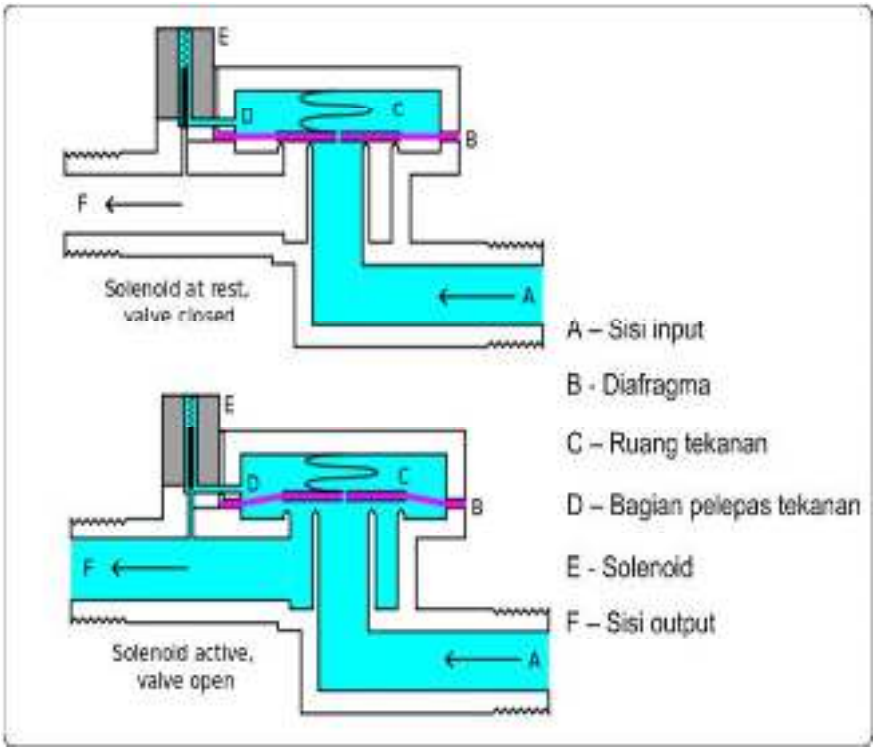
Fungsi ini akan dijalankan setelah *setup* (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.

2.1.14 Solenoid Valve

Solenoid valve merupakan katup yang dikendalikan dengan arus listrik baik AC maupun DC melalui kumparan / selenoida. *Solenoid valve* ini merupakan elemen kontrol yang paling sering digunakan dalam sistem fluida. Seperti pada sistem pneumatik, sistem hidrolik ataupun pada sistem kontrol mesin yang membutuhkan elemen kontrol otomatis. Contohnya pada sistem pneumatik, *solenoid valve* bertugas untuk mengontrol saluran udara yang bertekanan menuju aktuator pneumatik(*cylinder*). Atau pada sebuah tandon air yang membutuhkan solenoid valve sebagai pengatur pengisian air, sehingga tandon tersebut tidak sampai kosong.

2.1.14.1 Prinsip kerja solenoid valve

Solenoid valve akan bekerja bila kumparan/coil mendapatkan tegangan arus listrik yang sesuai dengan tegangan kerja(kebanyakan tegangan kerja solenoid valve adalah 100/200VAC dan kebanyakan tegangan kerja pada tegangan DC adalah 12/24VDC). Dan sebuah pin akan tertarik karena gaya magnet yang dihasilkan dari kumparan selenoida tersebut. Dan saat pin tersebut ditarik naik maka fluida akan mengalir dari ruang C menuju ke bagian D dengan cepat. Sehingga tekanan di ruang C turun dan tekanan fluida yang masuk mengangkat diafragma. Sehingga katup utama terbuka dan fluida mengalir langsung dari A ke F. Untuk melihat penggunaan solenoid valve pada sistem pneumatik. Buka artikel saya selanjutnya tentang sistem pneumatik beserta cara kerjanya.Gambar 2.8 menunjukan prinsip kerja pada *solenoid valve*



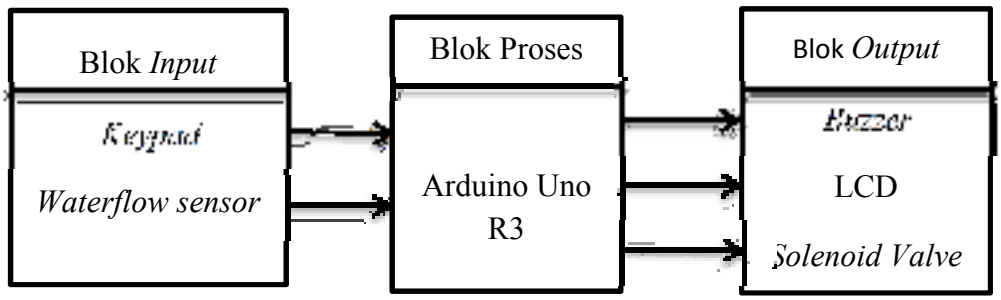
Gambar 2.8 Prinsip Kerja *Solenoid Valve*
Sumber : http://trikueni-desain-sistem.blogspot.co.id

2.2 Kerangka Bepikir

Kerangka berpikir merupakan sebuah model konseptual yang melandasi pemahaman-pemahaman yang lainnya, sebuah pemahaman yang paling

mendasar dan menjadi pondasi untuk pembuatan prototipe meteran air digital prabayar berbasis Arduino Uno.

Langkah awal untuk melakukan pembuatan alat diawali dengan memahami skema diagram blok alat. Gambar 2.9 menunjukkan skema diagram blok dari alat yang akan dibuat. Gambar 2.9 menerangkan alur kerja antar *hardware* dan juga antar blok.



Gambar 2.9 Diagram blok alat

Menurut gambar 2.9 blok *input* terdiri dari *keypad* 4 x 4, dan *waterflow* sensor. *Keypad* 4 x 4 merupakan modul yang digunakan untuk memasukkan data ke *board* arduino uno. data yang dimasukkan berupa algoritma yang telah dienkripsikan dan akan dieskripsikan oleh arduino uno sebagai blok proses. *Waterflow sensor* merupakan sensor yang memanfaatkan fenomena efek *Hall*. Efek *hall* ini didasarkan pada efek medan magnetik terhadap partikel bermuatan yang bergerak.

Blok proses terdiri dari *board* sistem arduino uno. Arduino merupakan *board* sistem pengendali yang *open source*. *Board* ini bebas untuk dikembangkan oleh siapa saja.

Arduino sebagai blok proses akan mengolah data *input*an dari *keypad*. Arduino akan mendeskripsikan data yang telah dikirimkan dan memberikan keputusan untuk *output* sesuai dengan hasil dari data yang telah diolah.

Blok *output* terdiri dari *solenoid valve, buzzer,* dan LCD. Blok ini merupakan akhir rangkaian dari sistem. Blok ini menerima data dari *board* sistem arduino uno apakah harus mengalirkan air atau tidak. Blok ini juga merupakan blok yang berfungsi untuk menghitung jumlah volume air yang mengalir, juga memonitoring sisa pulsa dan debit air.

BAB III

METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilakukan laboratorium tehnik digital dan mikrokontroler Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Jakarta (UNJ).

Waktu Penelitian dimulai pada bulan April 2016 sampai dengan September 2017 tahun akademik 2016/2017.

3.2 Metode Penelitian

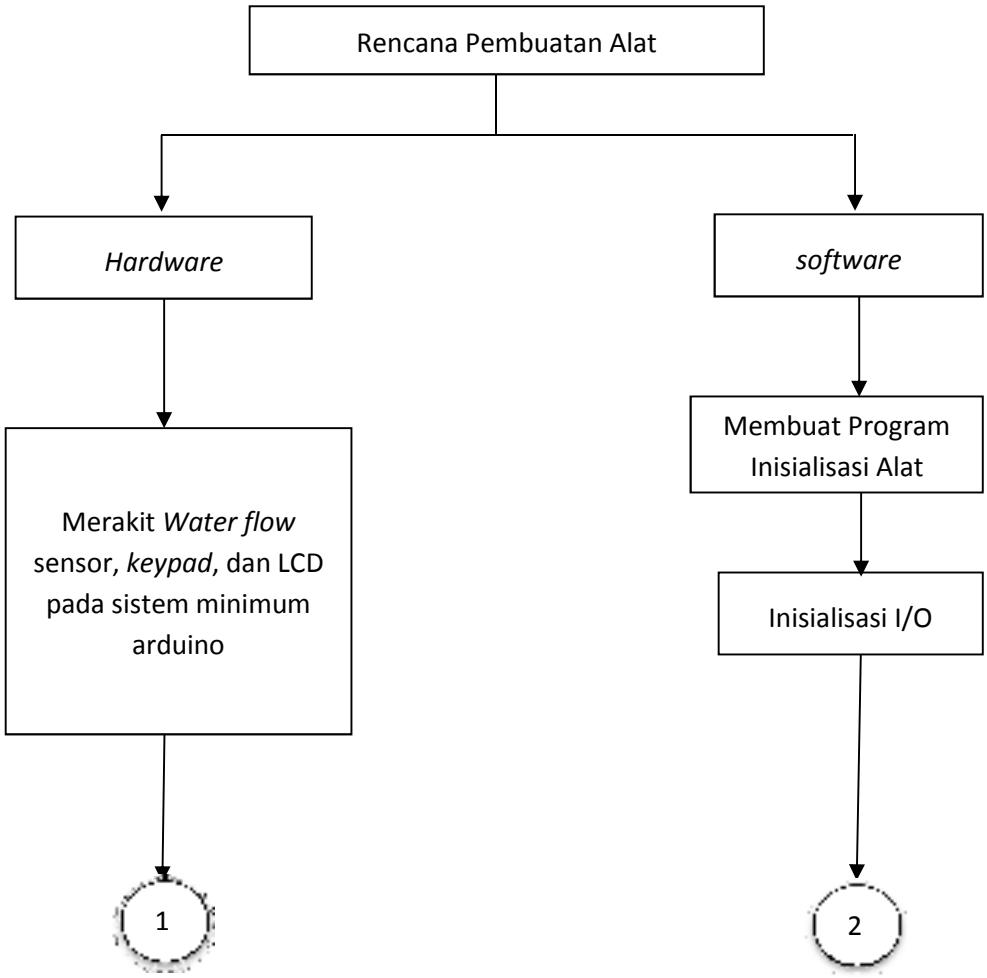
Metode penelitian yang digunakan untuk membuat prototipe meteran air prabayar digital berbasis arduino adalah metode eksperimen.

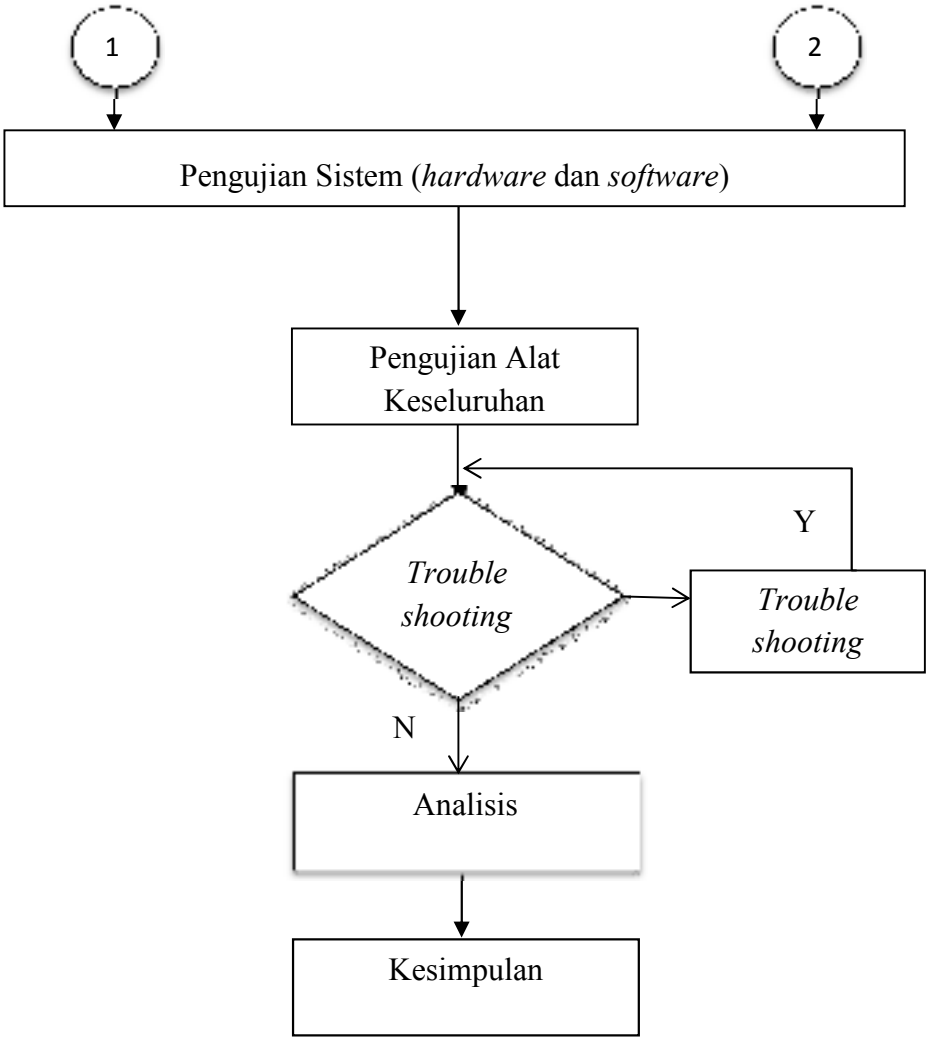
3.2.1 Instrumen Penelitian

Multimeter : digunakan untuk pengecekkan dan pengukuran komponen-komponen seperti resistor dan kapasitor, untuk mengukur tegangan masukkan dan keluaran di rangkaian penyearah serta *port-port* mikrokontroler, serta untuk pengecekkan apakah ada jalur PCB yang *short* atau terputus.

3.3 Rancangan Penelitian

Rancangan penelitian dalam membuat alat ditunjukkan pada gambar 3.1





Gambar 3.1 Diagram blok tahapan pembuatan prototipe meteran air prabayar digital berbasis arduino.

Penelitian dibagi menjadi dua tahap perancangan yaitu perancangan perangkat keras *(hardware)* dan perancangan perangkat lunak *(software).*

Perancangan perangkat keras *(hardware)* diawali dengan perakitan *keypad* 4 x 4, *water flow sensor* dan LCD pada rangkaian sistem minimum mikrokontroler. Rangkaian sistem minimum mikrokontroler kemudian diujikan terlebih dahulu, dan dilanjutkan dengan pembuatan catu daya 5 volt dan 12 volt.

Setelah perangkaian selesai dilanjutkan dengan pengujian pada setiap rangkaian dengan mengukur tegangan keluaran *(output)* menggunakan AVO

meter. Bila hasil pengukuran tidak sesuai dengan hasil yang diharapkan maka dilakukan analisa rangkaian atau lebih dikenal dengan istilah *trouble shooting.* Bila hasil sesuai dengan hasil yang diinginkan maka dilanjutkan ke tahap perancangan perangkat lunak *(software).*

Sebelum merancang program perangkat lunak *(software),* peneliti membuat diagram alur *(flowchart)* terlebih dahulu. Diagram alur *(flowchart)* berfungsi untuk menggambarkan urutan proses kerja suatu program secara terstruktur sehingga jika terjadi masalah kita dapat dengan mudah menelusuri kesalahan dalam pemrograman. Perancangan perangkat lunak *(software).*

Setelah seluruh program tersebut teruji dan berhasil, maka akan dilakukan penyatuan antara *hardware* dan *software.* Bila penyatuan antara *hardware* dengan *software* belum berhasil maka akan dilakukan analisa rancangan *(trouble shooting).* Bila *trouble shooting* telah terselesaikan dan *hardware* dengan *software* telah sejalan atau dapat bekerja dan dapat diujikan, maka tahap terakhir yaitu mengambil kesimpulan.

3.4 Desain Alat

3.4.1 Alat

Perangkat atau alat yang digunakan untuk membuat maket ini meliputi :

1. Perangkat lunak yang digunakan:

   a.Arduino IDE

2. Perangkat keras yang digunakan:

   a. *Hacksaw* (gergaji tangan).
   b. Tang buaya.

c. Tang potong.

d. *Screw drivers* (macam-macam obeng).

e. *Soldering iron* (solder listrik).

3.6.2. Bahan

Bahan-bahan yang digunakan dalam membangun sistem ini antara lain :

1. PCB polos.
2. Arduino dengan IC mikrokontroler Atmega 328.
3. *Keypad.*
4. LCD.
5. *Buzzer.*
6. *Solenoid valve*.
7. *Waterflow sensor.*
8. Pipa paralon.
9. Relai.
10. Papan triplek.

3.6.3 Pembuatan Maket Alat

Maket merupakan representasi dalam bentuk tiga dimensi yang meniru sebuah benda atau objek. Fungsi maket, biasanya digunakan untuk mendiskripsikan sebuah keadaan. Jadi maket digunakan sebagai representasi dari keadaan yang sebenarnya menuju keadaan yang akan diciptakan.

Sebuah maket tidak lebih dan tidak kurang adalah sesuatu yang abstrak, gambar miniatur dari sesuatu yang sesungguhnya yang dipertaruhkan bukan

penggambaran yang tepat dari suatu realitas, tetapi proses dari penyederhanaan untuk mendapatkan bentuk absiran yang telah ditentukan.

Pembuatan maket alat terbuat dari bahan triplek yang di cat berwarna hitam berukuran 15 cm x 10 cm berjumlah 1 lembar yang dipotong menjadi beberapa bagian. Maket ini terdiri dari 2 sekat yang menggambarkan kondisi alat meteran air. Dalam maket ini terdapat 1 buah *power supply* untuk mengalirkan tegangan, 1 buah *solenoid valve*, 1 buah *waterflow* sensor, 1 buah *keypad* 4x4 sebagai *input* ke arduino, 1 buah LCD sebagai sistem *monitoring*, 1 buah *buzzer* sebagai sistem *monitoring* membantu LCD, 1 buah pipa dengan diameter 10mm yang telah dipotong menjadi beberapa bagian yang digunakan sebagai penghubung antara *waterflow* ke *solenoid valve.*

Gambar 3.2 Sketsa Alat

3.7 Prosedur Penelitian

Prosedur penelitian dimulai dari pembuatan rencana alat dan juga proses sampai pelaksanaan pembuatan alat. Pembuatan alat dimulai dari perancangan masing-masing blok rangkaian berdasarkan tingkat kesulitannya. Kemudian masing-masing blok diujikan sehingga sesuai dengan rencana yang telah dibuat, kemudian disatukan semua blok rangkaian, yang kemudian dijadikan bahan acuan untuk tahap perancangan perangkat lunak *(software),* yang berbasis bahasa C. Setelah blok *hardware* telah selesai dibuat dan juga perangkat lunak *(software)* telah dibuat, semua blok disatukan dan kemudian diujikan.

3.7.1 Perancangan Alat

3.7.1.1 Rangkaian Catu Daya

Catu daya menggunakan trafo *Step down* yang digunakan untuk menurunkan tegangan dari 220 Volt AC PLN menjadi 12 Volt AC. Tegangan yang dihasilkan trafo masih AC, kemudian masuk ke dioda *bridge* sehingga menjadi DC tetapi DC yang dihasilkan masih belum sempurna karena masih *ripple* atau masih berfrekuensi sama dengan frekuensi AC PLN yaitu 50 Hz.

Cara untuk menjadikan DC yang sempurna yaitu dengan mengalirkannya masuk ke kapasitor yang berfungsi sebagai filter. Kapasitor bersifat menyimpan muatan listrik sehingga aliran listrik 12 Volt DC tersebut mengalir mengisi kapasitor sampai penuh lalu setelah penuh kapasitor akan membuang isi muatannya ke komponen selanjutnya, sehingga tegangan DC tersebut sudah tidak lagi berfrekuensi. Tegangan yang dibutuhkan oleh rangkaian di atas adalah 12 Volt sehingga digunakan Regulator untuk menghasilkan tegangan 12 Volt yaitu

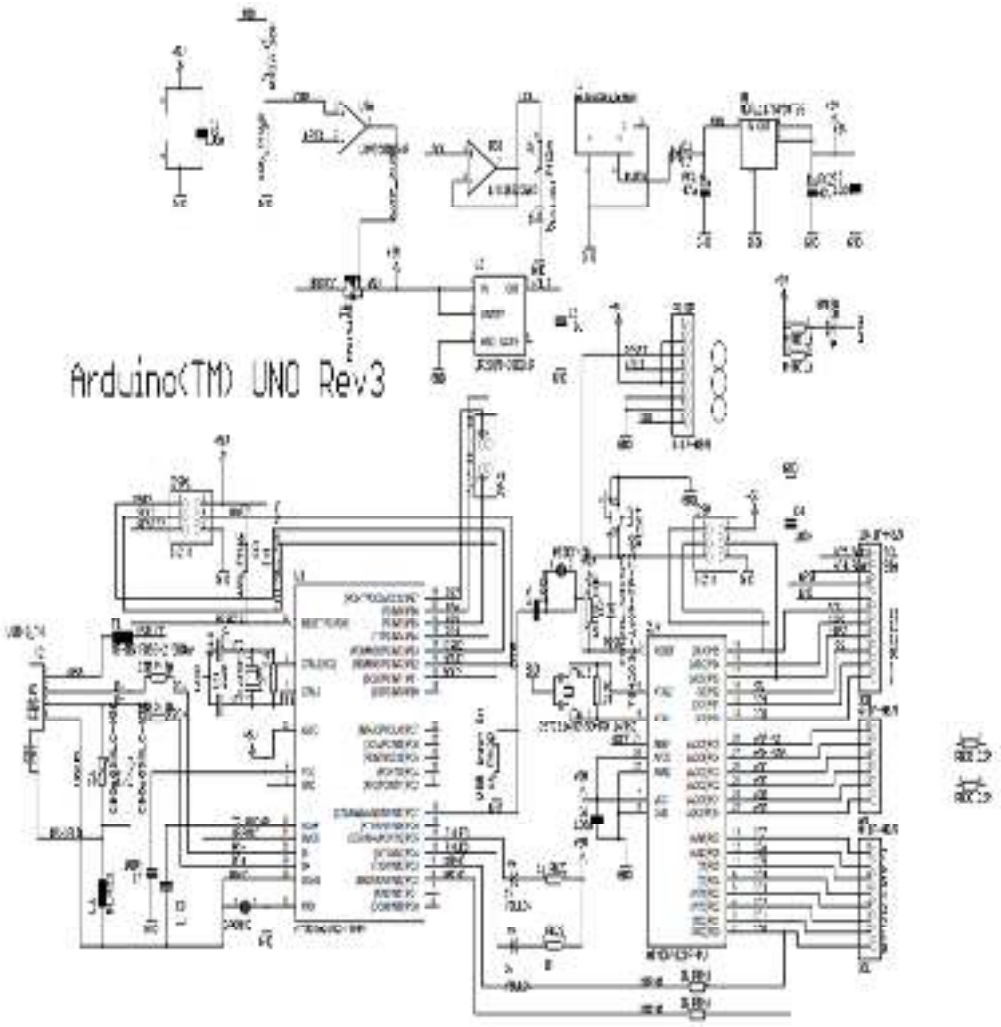
7805. Regulator adalah pembatas arus yang memiliki fungsi hampir mirip dengan dioda *zener*. Berapapun *input* tegangan yang masuk, *output*-nya tetap sesuai dengan karakteristiknya dan akan membuang sisanya ke *ground*.

3.7.1.2 Arduino Uno sebagai Pengontrol Rangkaian

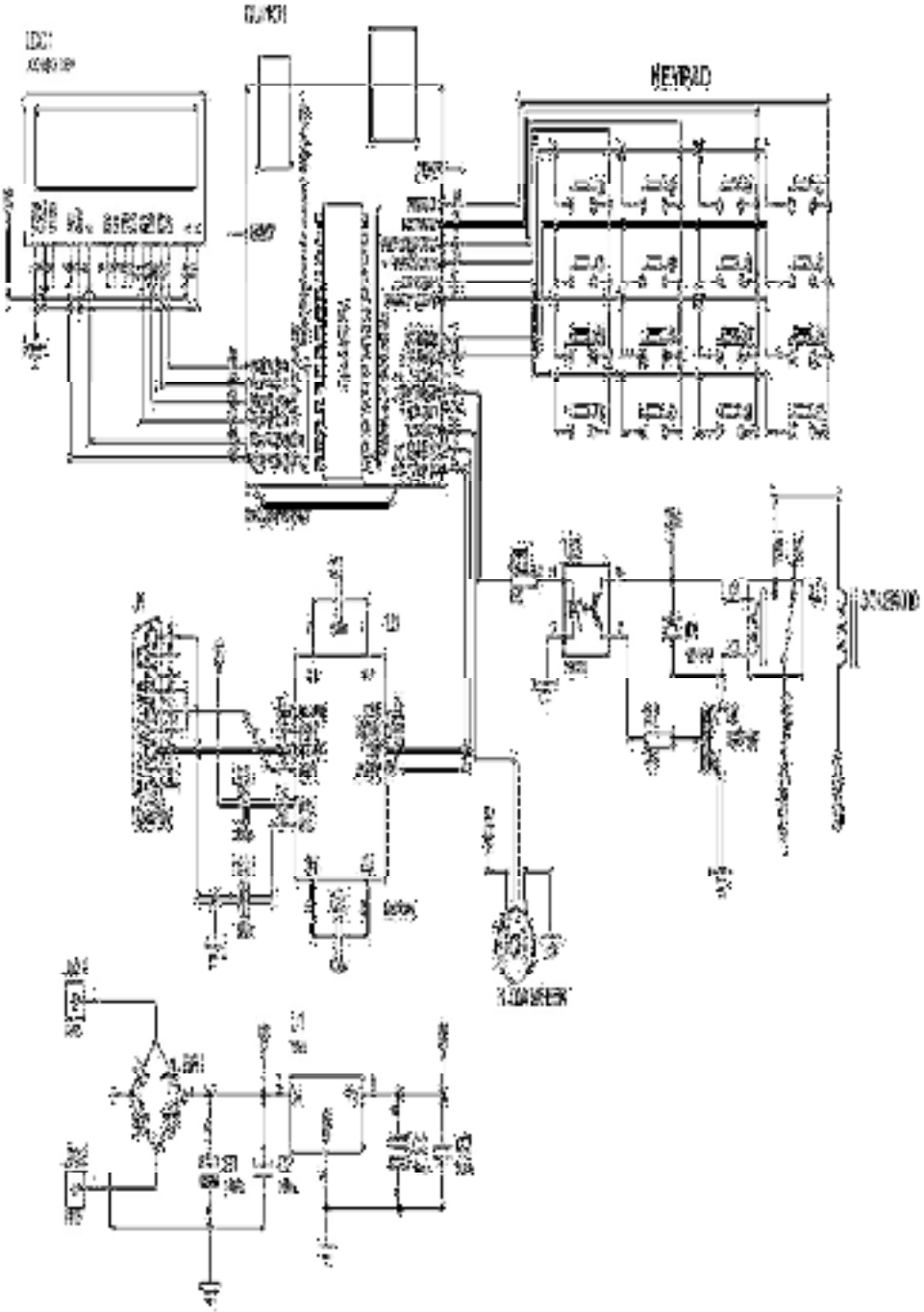
Arduino adalah rangkaian sistem minimum AVR yang ditanamkan *bootloader* ke IC berisi program *downloader* stk 500 sehingga kita tidak membutuhkan *downloader* terpisah untuk *flashing* AVR. Arduino memiliki *software* sendiri khusus untuk memprogram arduino *board*, kita bisa menggunakan ISIS *simulator* sebagai *board* sekedar untuk latihan memprogram arduino menggunakan arduino *software*. Gambar 3.3 menunjukan skematik diagram Arduino uno r3.

3.7.1.3 Relai Sebagai Kontak Penghubung Ke *Output*

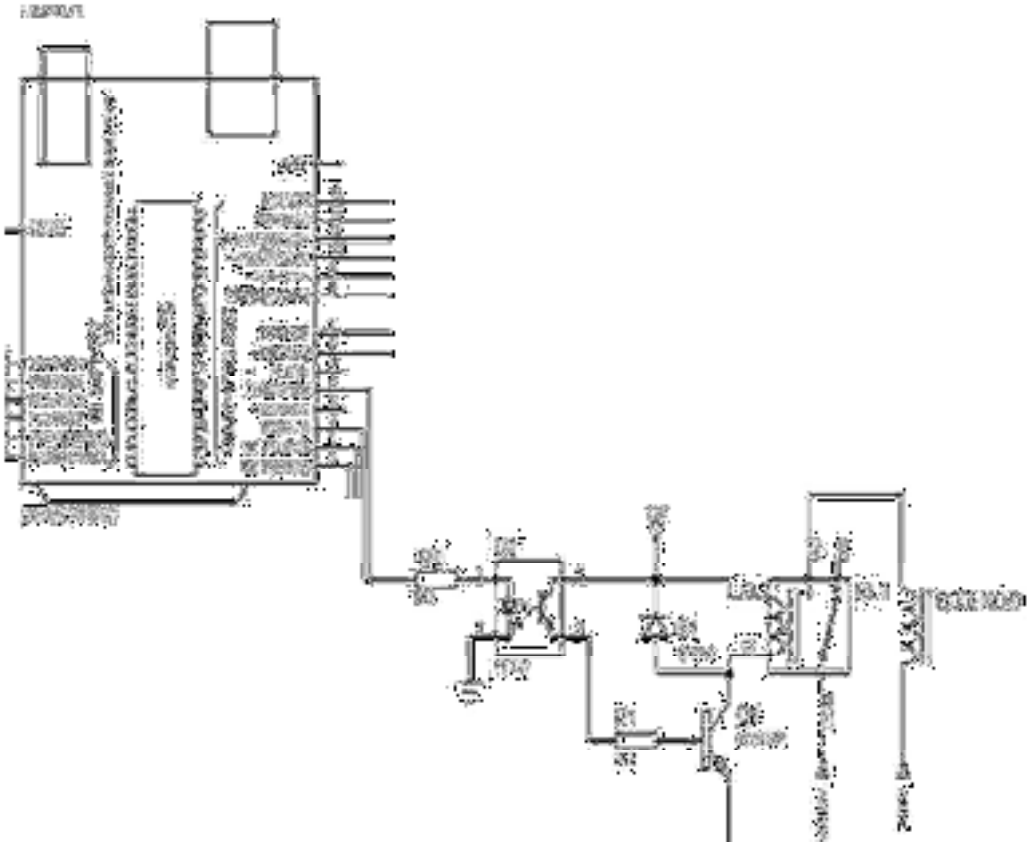
Sinyal dari mikrokontroler yang berbentuk perintah, tentu saja kita tidak bisa langsung dikontakan ke *output* yang telah kita tetapkan, sebelum itu sinyal tersebut harus melalui opto*coupl*er terlebih dahulu agar mikrokontroler dapat aman apabila terjadi loncatan tegangan. Loncatan tegangan tersebut diakibatkan karena adanya pertemuan antara arus AC dan DC, hal ini dapat merusak mikrokontroler. Relai berfungsi sebagai pengontak hasil dari peng-*couple*-an tegangan menuju ke *output*, hal ini berpacu pada fungsi relai yang sebagai saklar otomatis yang apabila diberi tegangan maka *switch* yang ada direlai akan bekerja. Skematik alat dan relai dapat dilihat pada gambar 3.4 dan 3.5



Gambar 3.3 Skematik Arduino Uno R3
Sumber : http://pratito-zicho.blogspot.co.id



Gambar 3.4 Skematik dari sistem meteran air digital prabayar berbasis arduino uno



Gambar 3.5 Relai sebagai kontak penghubung ke *output*

3.7.1.4 *Output*

Output adalah pengeluaran yang sebelumnya ada proses pengolahan data, *output* bekerja sebagai objek untuk suatu rancangan atau perencanaan, dengan kata lain *output* bisa didefinisikan sebagai hasil dari pengolahan proses. *Output* yang saya gunakan dalam perancangan alat meteran air digital prabayar berbasis mikrokontroler Arduino ini adalah *solenoid valve*, *buzzer,* lcd dimana semua alat tersebut mempunyai fungsi yang berbeda-beda.

3.7.1.5 Alat-Alat Pendukung

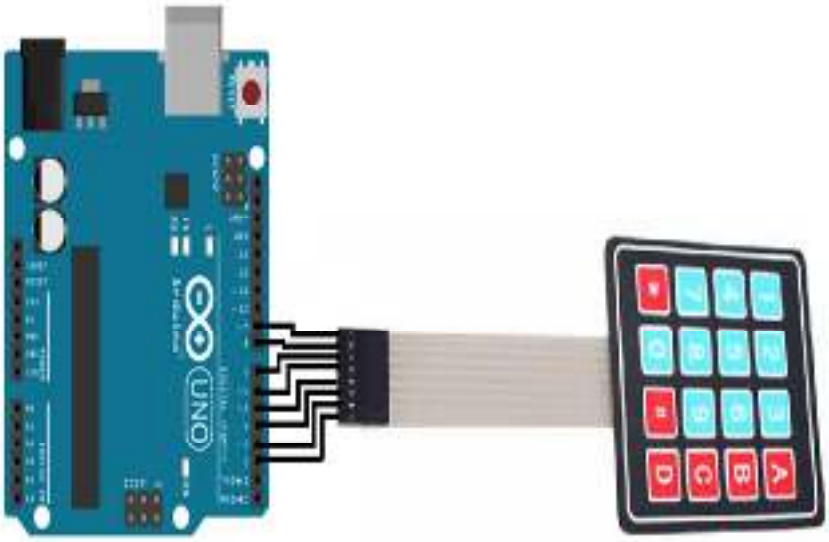
Bagian alat pendukung tentu saja perancangan alat ini tidak akan berhasil tanpa adanya *suplay* tegangan, *suplay* tegangan berfungsi sebagai penghidup mikrokontroler, opto*coupler*, dan relai, ketika mikrokontroler diberi tegangan 5 volt maka mikrokontroler akan berfungsi dan siap untuk mengontrol alat-alat yang akan dikontrol sesuai dengan program yang sudah ditetapkan, tidak hanya *suplay* tegangan, papan miniatur pun sangat penting untuk perancangan ini, karena papan miniatur berfungsi sebagai gambaran pengaplikasian ketika alat bekerja, dimulai alat dalam keadaan mati sampai alat dalam keadaan menyala.

3.7.2 Pemasangan dan pengawatan

Pemasangan dan pengawatan adalah menyusun atau memasangkan semua komponen dalam sebuah perancangan sehingga menjadi suatu alat yang direncanakan sebelumnya, pada pemasangan alat meteran air digital prabayar berbasis mikrokontroler ini tentunya harus mengetahui dahulu *input*, *output*, positif, negative, *ground*, dan basis, kolektor, emitter (apabila ada transistor), tidak hanya itu, kaki-kaki yang terdapat pada IC dan relai pun tentunya harus memahaminya.

3.7.2.1 Pemasangan *Keypad* pada Mikrokontroler Arduino

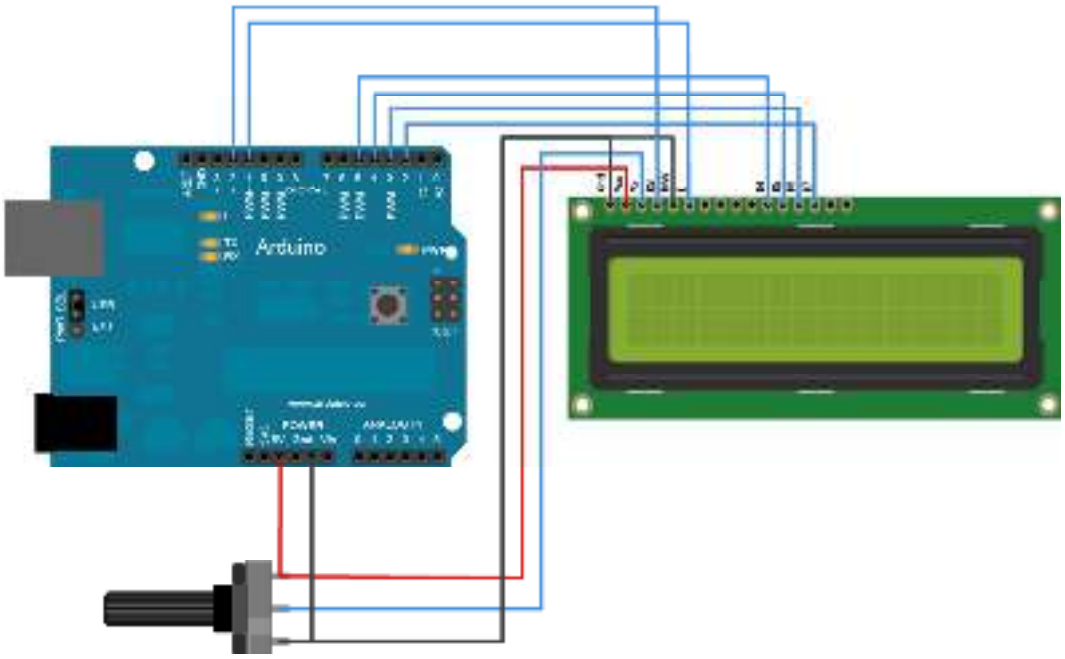
Keypad yang dipakai pada prototipe ini adalah *keypad* 4x4 merupakan *keypad* yang dibuat untuk dapat dipasangkan langsung pada sistem minimum Arduino. Gambar 3.6 merupakan gambar *keypad* yang langsung dapat dipasangkan dengan sistem minimum Arduino.



Gambar 3.6 *Keypad* 4x4 dan Arduino Uno
Sumber : http://www.boarduino.web.id

3.7.2.2 Pemasangan LCD (*Liquid Crystal Display*) sebagai sistem *monitoring*

Liquid Crystal Display (LCD) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan di berbagai bidang misalnya alal-alat elektronik seperti televisi, kalkulator, atau pun layar komputer. Pada postingan aplikasi LCD yang digunakan ialah LCD dot matrik dengan jumlah karakter 16 x 2. LCD sangat berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat. Gambar 3.7 merupakan sistem *monitoring* yang langsung dipasangkan pada sistem mikrokontroler arduino.

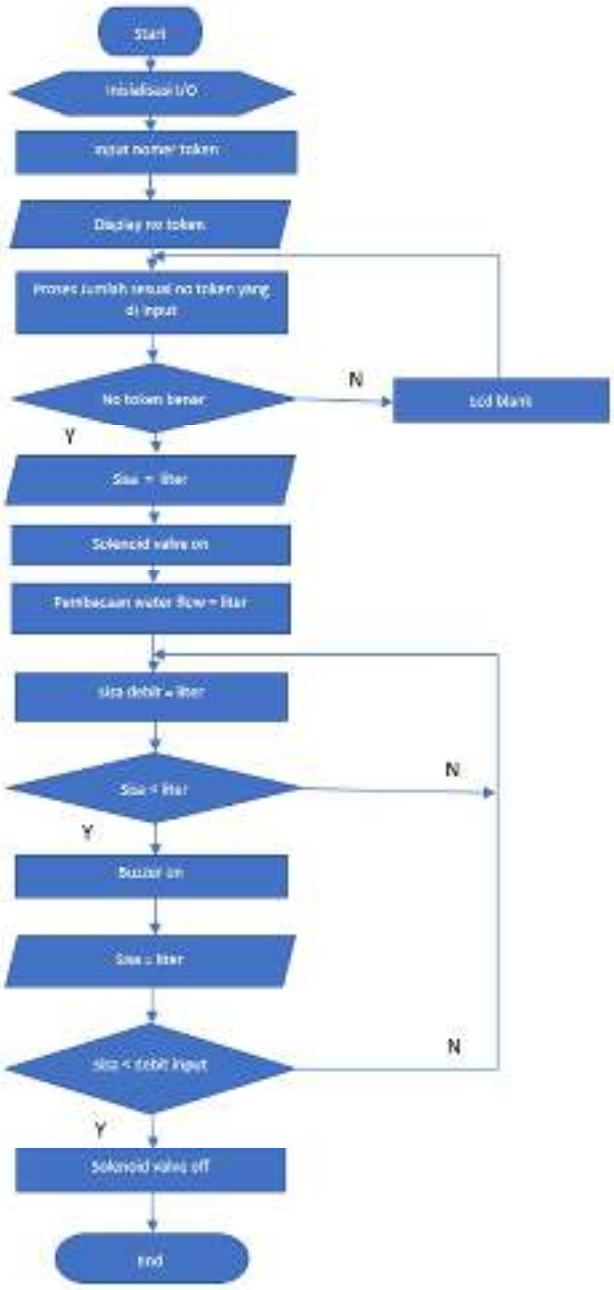


Gambar 3.7 Pemasangan LCD pada Arduino
Sumber : http://www.boarduino.web.id

3.7.3 Pembuatan Program

3.7.3.1 *Flowchart* Alur kerja Alat

Sebelum merancang program perangkat lunak (*software*) peneliti membuat diagram alur (*flowchart*) terlebih dahulu. Gambar 3.8 merupakan *flowchart* dari sistem alat.



Gambar 3.8 *Flowchart* sistem meteran air digital prabayar berbasis arduino

3.7.3.2 Cara Kerja Alat

Cara Kerja Alat

1. Nyalakan alat dengan menghubungkan *adapter* (DC 12 *volt*) ke *slot power* atau dengan menggunakan kabel *USB.*
2. Jika menggunakan *adapter* (*DC* 12 *volt*) pastikan program telah ter-*upload* ke sistem arduino atau dengan meng-*upload*-nya dari *PC/Laptop* dengan menggunakan kabel *USB*.
3. Kemudian I/O akan terinisialisasi dan dapat bekerja.
4. *Input* nomor token sesuai dengan token yang tersedia lalu nomor tersebut akan diproses.
5. Setelah nomor token sudah di-*input*, maka pada layar lcd akan muncul nomor token yang di *input* tersebut
6. Jumlah meteran yang diproses akan sesuai dengan nomor token yang di*input*.
7. Setelah diproses maka akan muncul tampilan sisa debit air pada LCD.
8. Kemudian *solenoid valve* akan *on* dan mengeluarkan air yang ditentukan.
9. *Waterflow* akan membaca debit air yang mengalir pada pipa.
10. Kemudian jika debit air lebih kecil dari *limit* yang ditentukan maka *buzzer* akan *on*.
11. Setelah proses tersebut *solenoid valve* akan *off* dan alat selesai bekerja.

3.7.3.3 Program Inisialisasi Alat

Program inisialisasi alat adalah program yang dilaksanakan setelah diaktifkan, atau di-*reset* secara *hardware* yang berfungsi untuk persiapan, yaitu

penginisialisasian I/O yang digunakan dalam alat meteran air digital prabayar berbasis arduino uno r3.

3.7.3.4 Konfigurasi Tabel I/O Pada Mikrokontroler

Tabel 3.1 di bawah ini menunjukan konfigurasi pin mikrokontroler ATMega328 dan konfigurasi pin Arduino Uno agar memudahkan dalam pembacaan *input/output* pada sistem skematik alat meteran air digital prabayar.

Tabel 3.1 Konfigurasi *Pin* Mikrokontroler Atmega328P-PU dan Konfigurasi Pin Arduino Uno R3 pada Alat

Pin Arduino	*Pin Mikrokontroler*	Nama	Fungsi
VCC	VCC	----	Sumber tegangan
GND	GND	----	*Grounding*
8	PB0	*INPUT 1*	Sebagai kolom 1 pada *keypad*
9	PB1	*INPUT 2*	Sebagai kolom 2 pada *keypad*
10	PB2	*INPUT 3*	Sebagai baris 4 pada *keypad*
11	PB3	*INPUT 4*	Sebagai baris 3 pada *keypad*
12	PB4	*INPUT 5*	Sebagai baris 2 pada *keypad*
13	PB5	*INPUT 8*	Sebagai baris 1 pada *keypad*
2	PD2	*SENSOR FLOW*	Sebagai penghitung debit air
4	PD4	*OUTPUT RELAY*	Sebagai penghubung dari arduino ke selenoid

Pin Arduino	*Pin Mikrokontroler*	Nama	Fungsi
6	PD6	*INPUT 6*	Sebagai kolom 4 pada *keypad*
7	PD7	*INPUT 7*	Sebagai kolom 3 pada *keypad*
0	PD0	*MODULE SMS*	Sebagai sistem pengamanan
A0	PC0	KE D7 LCD	Sebagai *output* mikrokontroler ke LCD
A1	PC1	KE D6 LCD	Sebagai *output* mikrokontroler ke LCD
A2	PC2	KE D5 LCD	Sebagai *output* mikrokontroler ke LCD
A3	PC3	KE D4 LCD	Sebagai *output* mikrokontroler ke LCD
A4	PC4	KE E LCD	Sebagai *output* mikrokontroler ke LCD
A5	PC5	KE RS LCD	Sebagai *output* mikrokontroler ke LCD

Dari semua pengawatan pada perancangan alat meteran air digital prabayar berbasis mikrokontroler ini tentunya sudah disusun rancangan terlebih dahulu dan diprogram pada mikrokontroler agar *output* bisa bekerja sesuai yang diinginkan.

3.8 Kriteria Pengujian Alat

Proses pengujian alat dilakukan terhadap catu daya, Pengujian Sistem *monitoring* Menggunakan *liqiud crystal display* (LCD)**,** arduino pin *output*, pengukuran , dan pengujian *waterflow sensor*. Untuk memberikan hasil penelitian yang akurat dan dapat dipertanggungjawabkan, maka diperlukan ketentuan

langkah-langkah yang harus dilakukan untuk pengujian dan analisis data terhadap sistem.

3.8.1 Pengujian LCD

Pengujian dilakukan dengan menggunakan bantuan sistem minimum arduino. Pengujian dilakukan dengan memasukkan contoh *coding* program untuk LCD pada sistem minimum arduino dan melihat apakah LCD mencetak *coding* program sesuai yang di-*upload* pada sistem minimum arduino. Berikut ini adalah contoh *coding* program yang dapat digunakan untuk menguji LCD

```
lcd.clear();
    //bagian menu
    //00D Tampilkan sisa token
    //11D Tampilkan id token
    //22D Tampilkan batas minimal untuk alaram
    //33D Tampilkan debit air liter/detik
    //44D Tampilan isi ulang terakhir



    if (isi == "00D") {
      lcd.setCursor(0, 0);
      lcd.print("Sisa Token");
      lcd.setCursor(0, 1);
      lcd.print(token);
      lcd.print(" ltr");
```

```
  standby = true;
}
if (isi == "11D") {
  lcd.setCursor(0, 0);
  lcd.print("NoId");
  lcd.setCursor(0, 1);
  lcd.print(Noid);
}
if (isi == "22D") {
  lcd.setCursor(0, 0);
  lcd.print("Batas alaram");
  lcd.setCursor(0, 1);
  lcd.print(limit);
}

if (isi == "33D") {
  lcd.setCursor(0, 0);
  lcd.print("Debit air");
  lcd.setCursor(0, 1);
  lcd.print(debit);
  lcd.print(" l/min  ");
  showdebit = true;
} else showdebit = false;
if (isi == "44D") {
```

```
    lcd.setCursor(0, 0);
    lcd.print("Isi terakhir");
    lcd.setCursor(0, 1);
    lcd.print(reload);
    lcd.print(" ltr");
  }
```

3.8.2 Pengujian Catu Daya

Catu daya / PSU (*Power Supply Unit*) merupakan sumber daya dari seluruh sistem pada prototipe sistem meteran air digital prabayar berbasis arduino uno. Rangkaian catu daya memberikan sumber tenaga yang akan digunakan oleh rangkaian kendali utama pada prototipe ini. Tujuan pengujian ini agar memastikan bahwa catu daya berfungsi dengan baik. Pengujian PSU, *solenoid valve*, dan *waterflow* dapat dilihat pada tabel 3.2, dan tabel 3.3.

Tabel 3.2. Pengujian Catu daya 12VDC.

NO	Pengujian	Kriteria Pengujian	Hasil Pengujian
1.	*Input* PSU 12 VDC	220 VAC	.....VAC
2.	*Output* PSU 12 VDC	12 VDC	.....VDC

Tabel 3.3 Pengujian Daya pada *Solenoid Valve* dan *Waterflow*

No.	*Output* Alat	Keadaan Saat Aktif (*High*)	Keadaan Saat Aktif (*Low*)
1.	*Solenoid Valve*	V	V
2.	*Waterflow Sensor*	V	V

3.8.3 Pengujian *Water Flow Sensor*

Water flow sensor merupakan sensor yang mendeteksi aliran air. Umumnya besaran yang dibaca/diukur merupakan besaran debit air yang kemudian diubah menjadi besaran listrik. Pengujian dilakukan untuk mengetahui nilai tegangan yang digunakan sensor untuk beroperasi. Tabel 3.4 di bawah ini merupakan tabel dari pengujian *water flow sensor*.

Tabel 3.4 Pengujian *Waterflow sensor*

No.	Tinggi Penampung (h)		Waktu (t)	Jumlah Air yang diperoleh (Perhitungan)	Volume Air (LCD)	Selisih	*Error (%)*
	Tinggi Awal	Tinggi Akhir					

1.	cm	cm	menit	l	l	l	%
2.	cm	cm	menit	l	l	l	%
3.	cm	cm	menit	l	l	l	%
4.	cm	cm	menit	l	l	l	%
						Total *Error(%)*	%

3.8.4 Pengujian *Software*

Pengujian dilakukan untuk mengetahui apakah *software* yang dibuat sudah benar-benar bekerja atau belum. Karena dengan adanya pengujian pada *software* ini, mempermudah monitoring pada bagian mana saja yang masih *error* atau *troubleshooting*. Tabel 3.5 dan Tabel 3.6 di bawah ini menunjukan pengujian yang akan diuji pada *software.*

Tabel 3.5 Pengujian *Software*

No.	*Input* no token	Pembacaan Sensor (LCD)	Debit Aktual
1.	Rp. 20.000,00.	l	l
2.	Rp. 50.000,00.	l	l
3.	Rp. 100.000,00.	l	l
4.	Rp. 200.000,00.	l	l

Tabel 3.6 Pengujian *Software* ketika daya di matikan/di hidupkan

Menit Ke	Kondisi	*Input* Token	Token Terpakai (Perhitungan)	Sisa (LCD)	Keterangan
0 menit	*Standby*				
5 menit	*Standby*				
10 menit	Sistem *Off*				
12 menit	*Standby*				

Tabel 3.7 Pengujian Software dengan membandingkan jumlah air per liter

No.	Liter	Sisa (LCD)	Debit Aktual	Keterangan
1	0			
2	1			
3	2			
4	5			

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengujian Sistem Meteran Air Digital Prabayar Berbasis Arduino

Hasil penelitian prototipe sistem meteran air digital prabayar berbasis arduino dilakukan untuk mengetahui seberapa besar nilai keberhasilan dalam percobaan pembuatan sistem, selain itu membuktikan apakah kenyataan sesuai dengan program atau sistem yang telah dibuat, berupa pengujian catu daya, pengujian *waterflow sensor*, pengujian sistem *monitoring* menggunakan *liquid crystal display* (LCD), dan pengujian *solenoid valve*.

4.1.1 Hasil Pengujian LCD (*Liquid Crystal Display*)

Hasil pengujian dari LCD dibuktikan ketika keypad ditekan dan dimasukkan angka yang sudah ditentukan dengan sistem minimum arduino, maka lcd akan menampilkan sisa token, no *id* pelanggan, batas *alarm* dan debit air.

4.1.2 Hasil Pengujian Catu Daya

Catu daya / PSU (*Power Supply Unit*) merupakan sumber daya dari seluruh sistem pada prototipe sistem kendali peralatan listrik berbasis arduino uno. Rangkaian catu daya memberikan sumber tenaga yang akan digunakan oleh rangkaian kendali utama pada prototipe.

Proses pengujian dilakukan dengan cara memasangkan kabel *power* utama ke tegangan sumber PLN lalu mulai dilakukan pengukuran tegangan menggunakan AVO meter, diawali dengan mengukur tegangan masuk yang bersumber dari PLN, lalu dilanjutkan dengan mengukur tegangan keluaran modul catudaya *switching* 12VDC. Hasil pengujian dapat dilihat pada tabel 4.1

Tabel 4.1. Hasil Pengujian Catu Daya.

NO	Pengujian	Kriteria Pengujian	Hasil Pengujian
1	Input PSU 12VDC	220 VAC	218 VAC
2	Output PSU 12VDC	12 VDC	11,94 VDC

Tabel 4.2 Hasil daya pada *Solenoid Valve* dan *Waterflow Sensor*

No.	*Output* Alat	Keadaan Saat Aktif (*High*)	Keadaan Saat Aktif (*Low*)
1.	*Solenoid Valve*	12 V	0 V
2.	*Waterflow Sensor*	11,8 V	0 V

Hasil pengujian catu daya dapat dilihat pada tabel 4.1 dan tabel 4.2 hasil pengujian menunjukkan bahwa terjadi penurunan tegangan dari sumber listrik PLN, namun penurunan itu tidak mengganggu keluaran dari PSU 12V karena hasil pengujian mendekati dengan nilai kriteria pengujian. Dengan ini dapat dikatakan bahwa pengujian catu daya sesuai dengan kriteria pengujian.

4.1.3 Hasil Pengujian *Waterflow Sensor*

Water flow sensor merupakan sensor yang mendeteksi aliran air. Umumnya besaran yang dibaca/diukur merupakan besaran debit air yang kemudian diubah menjadi besaran listrik. Pengujian dilakukan untuk mengetahui nilai tegangan yang digunakan sensor untuk beroperasi. Tabel 4.3 menunjukan hasil pengujian *waterflow sensor.*

Tabel 4.3 Pengujian *Waterflow Sensor*

No.	Tinggi Penampung (h)		Waktu (t)	Jumlah Air yang diperoleh (Perhitungan)	Debit Air (LCD)	Selisih	*Error (%)*
	Tinggi Awal	Tinggi Akhir					
1.	0 cm	18 cm	1 Menit	17,8 liter	18 liter	0,2 liter	1,11%
2.	0 cm	49 cm	3 Menit	47,6 liter	48 liter	0,4 liter	0,84%
3.	0 cm	100 cm	6 Menit	97,9 liter	98 liter	0,1 liter	0,102 %
4.	0 cm	202 cm	12 Menit	196,9 liter	198 liter	1,1 liter	0,56%
						Total *Error (%)*	0,653 %

Contoh :

Alat saya membaca volume air 18 liter, sedangkan menurut perhitungan diatas sebesar 17,8 liter, jadi persentase *error*nya sebesar 1,11%, ditambah penampungan tidak 100% berbentuk balok. Sedangkan untuk pembayaran yg dikenakan ke pelanggan (nilai yang dibaca LCD x 1000)+ Rp 2000, Rp 2000 ini adalah biaya admin. Jadi cara perhitungannya seperti itu.

4.1.4 Pengujian *Software*

Pengujian dilakukan untuk mengetahui apakah *software* yang dibuat sudah benar-benar bekerja atau belum. Karena dengan adanya pengujian pada *software* ini, mempermudah *monitoring* pada bagian mana saja yang masih *error* atau *troubleshooting*. Tabel 4.4 menunjukan pengujian apa saja yang akan diuji pada *software.*

Tabel 4.4 Pengujian *Software*

No.	*Input* no token	Pembacaan Sensor (LCD)	Debit Aktual
1.	Rp. 20.000,00.	18 l	18 l
2.	Rp. 50.000,00.	48 l	48 l
3.	Rp. 100.000,00.	98 l	98 l
4.	Rp. 200.000,00.	198 l	198 l

Rumus liter = (UANG-ADMIN)/TARIF

= (Rp 20.000-Rp 2000)/1000

= Rp 18.000/1000

= 18 liter

Rumus Rp 1 = …. Liter

$$=\frac{18}{18.000} = 0{,}001 \text{ liter}$$

Tabel 4.5 Pengujian *Software* ketika daya di matikan/di hidupkan

Menit Ke	Kondisi	*Input* Token	Token Terpakai (Perhitungan)	Sisa (LCD)	Keterangan
0 menit	*Standby*	Rp 100.000,00.	0 liter	98 liter	Sesuai
5 menit	*Standby*	Rp 100.000,00.	78 liter	20 liter	Sesuai
10 menit	Sistem *Off*	Sistem *Off*	Sistem *Off*	Sistem *Off*	Sistem *Off*
12 menit	*Standby*	Rp 100.000,00.	78 liter	20 liter	Sistem *On* setelah 2 menit *Off*

Tabel 4.6 Pengujian Software dengan membandingkan jumlah air per liter

No.	Liter	Sisa (LCD)	Debit Aktual	Keterangan
1	0	18 liter	18 liter	Sesuai
2	1	17 liter	17 liter	Sesuai
3	2	15 liter	15 liter	Sesuai
4	5	10 liter	10 liter	Sesuai

4.2 Kelebihan dan Kekurangan Alat

4.2.1 Kelebihan Alat

Kelebihan alat meteran air dengan Arduino Uno ini adalah sebagai berikut:

1. Dapat *memonitoring* penggunaan air.
2. Meminimalisir keluhan-keluhan masyarakat.
3. Masyarakat dapat men-stok kode yang sudah dibeli, tanpa harus mengantri di loket pembayaran PDAM.

4.2.2 Kekurangan Alat

Kekurangan alat meteran air dengan Arduino Uno ini adalah sebagai berikut:

1. Pada Arduino jika ada komponen elektronika yang rusak maka harus mengganti *board* komponennya.
2. Sistem tidak memiliki sumber tenaga cadangan sehingga jika sumber dari jala-jala PLN dimatikan maka seluruh sistem akan mati.
3. Meteran air digital prabayar berbasis arduino lebih mahal dari segi biaya dibandingkan dengan sistem konvensional, karena :
   a. Membutuhkan arduino untuk membuat sistem meteran air digital.
   b. LCD untuk memonitor penggunaan air.
   c. *Keypad* untuk memasukan no token.
   d. *Buzzer* untuk memberitahukan token habis dan alat dibobol.
   e. *Waterflow* dan *Selenoid Valve* sebagai *input* dan *output* alat.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah dilakukan penelitian prototipe sistem meteran air digital prabayar berbasis Arduino Uno r3 dapat ditarik kesimpulan sebagai berikut:

1. Meteran air digital yang dirancang dan dihasilkan dapat digunakan pada pipa berdiameter 25 mm.
2. Meteran air digital prabayar berbasis arduino uno R3 yang menggunakan LCD 16 x 2 sebagai tampilannya.
3. Komputer pada meteran air digital prabayar berbasis arduino uno R3 hanya sebatas sebagai media simulasi pengiriman data.
4. Meteran air digital prabayar berbasis arduino uno R3 dapat mempermudah proses pembacaan data hasil pemakaian air.

5.2 Saran

Saran yang diberikan oleh penulis untuk mengembangkan dan meningkatkan kinerja dari penelitian adalah sebagai berikut :

1. Pemilihan komponen-komponen dengan kualitas yang lebih baik sehingga sistem dapat berjalan lebih baik dan mempunyai usia pakai yang relatif lebih lama.
2. Sistem tidak mempunyai jala-jala listrik cadangan sehingga disarankan untuk mempunyai jala-jala listrik cadangan, seperti menggunakan *solar cell.*

3. Untuk pengembangan lebih lanjut untuk mencegah kebocoran pada sambungan pipa sehingga pengukuran dapat lebih presisi.

DAFTAR PUSTAKA

Dinata, Yuwono Marta. 2015. *Arduino itu Mudah*. Jakarta: PT Elex Media Komputindo.

Fakultas Teknik. 2012. *Buku Pedoman Skripsi/Karya Inofatif/Komprehensif*. Jakarta: Fakultas Teknik Universitas Negeri Jakarta.

http://repository.usu.ac.id/bitstream/123456789/35001/4/Chapter%20II.pdf., diakses pada 12 Febuari 2017.

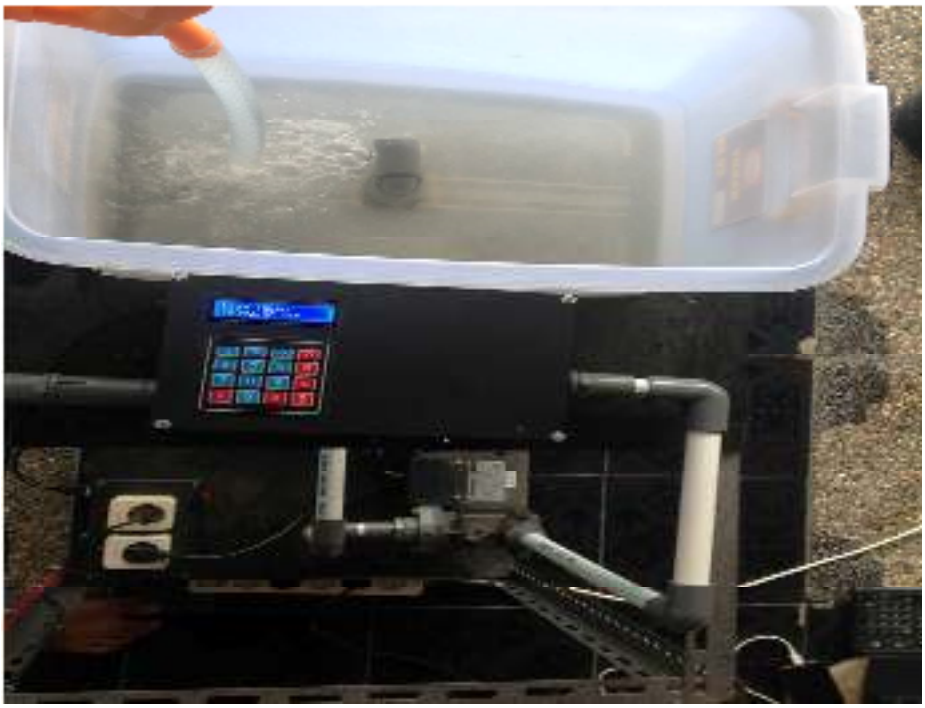
Syahwil, Muhammad. 2013. *Panduan Mudah Simulasi & Praktek Mikrokontroler Arduino*. Jakarta: Andi.

http://ecadio.com/mengenal-dan-belajar-arduino-uno-r3, diakses pada 23 Februari 2017.

http://pratito-zicho.blogspot.co.id diakses pada 23 Februari 2017.

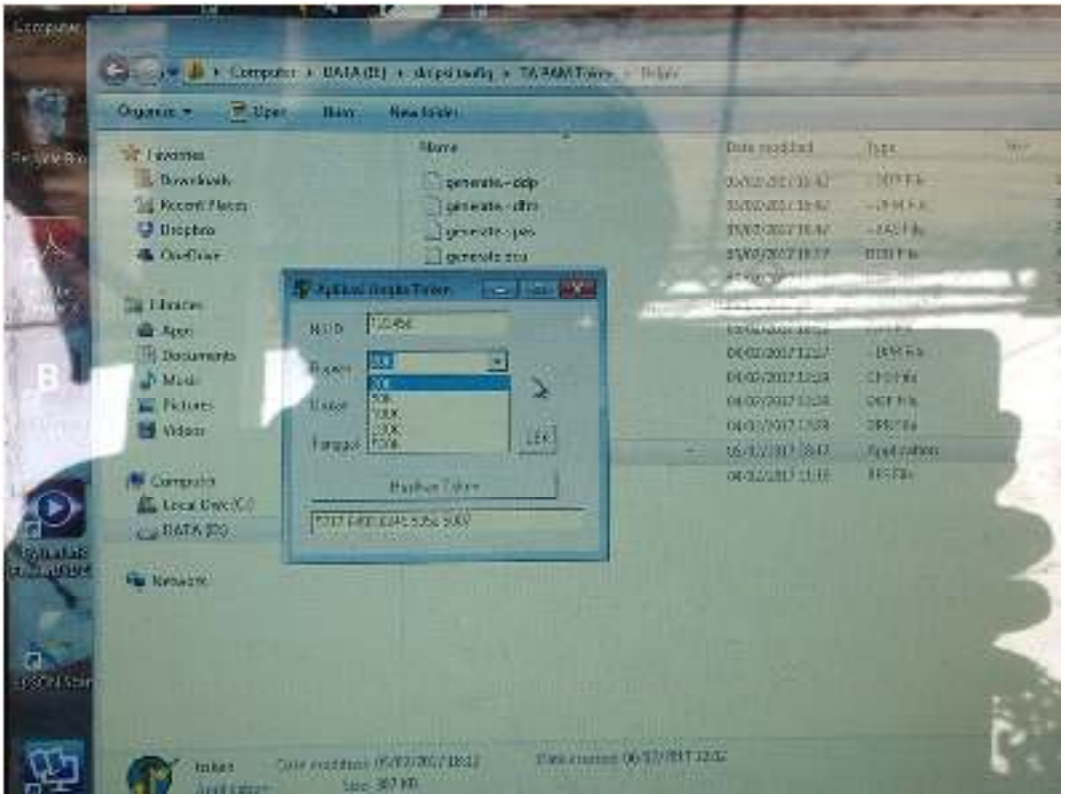
Lampiran 1

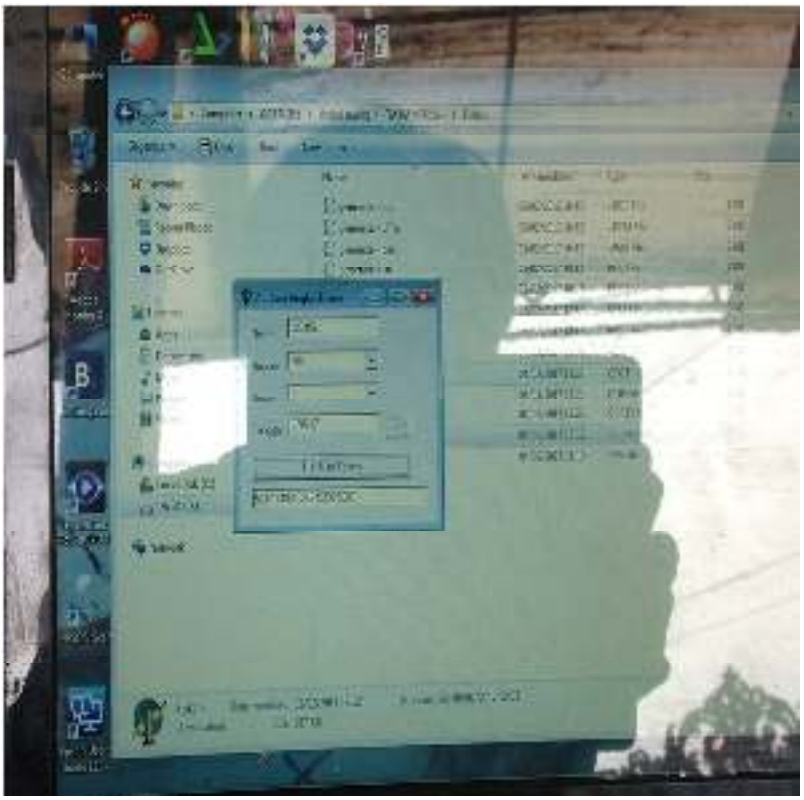
Lampiran 2



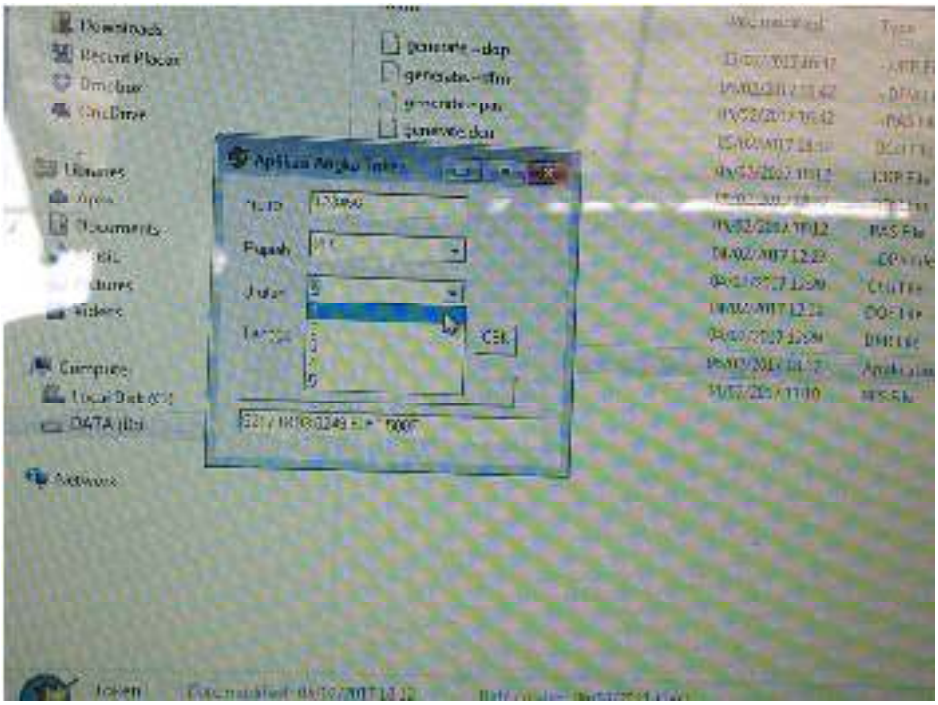
Lampiran 3

Debit air
17.17 l/min
1 2 3 A
4 5 6 B
7 8 9 C
* 0 # D

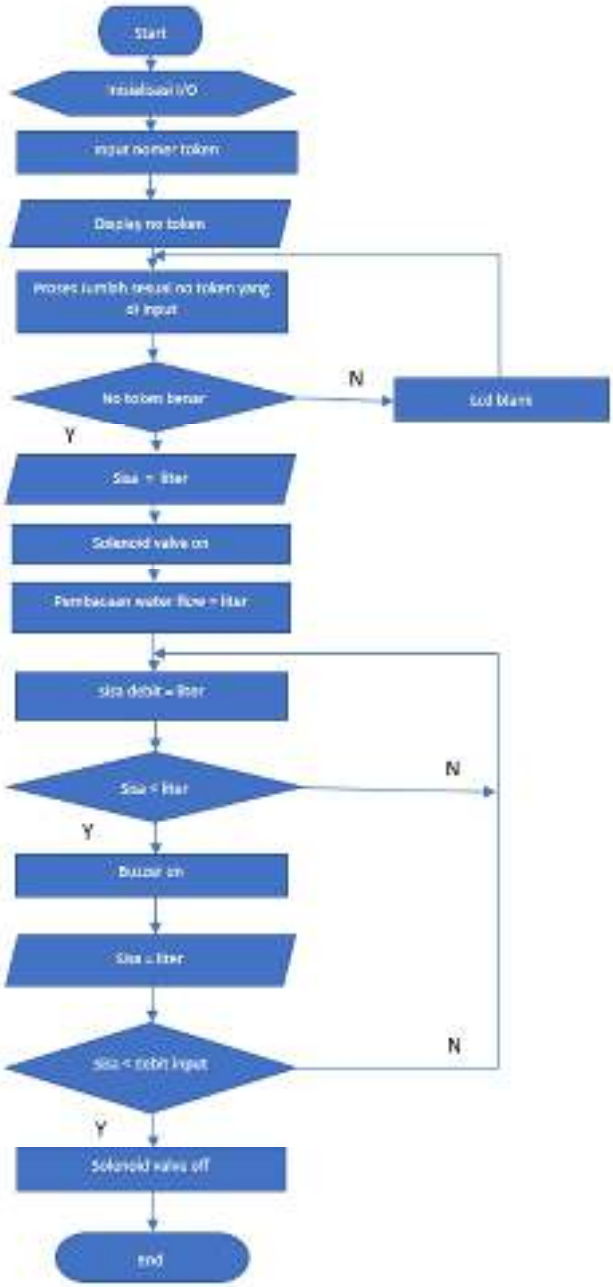




Lampiran 4



Lampiran 5



***Flowchart* Pada Alat**

Lampiran 6

Program Arduino

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <EEPROM.h>
#define buzzer 5
#define solenoid 4
#define sensor_limit 6
#define com 7
#define Modem Serial1

LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);

const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {39, 37, 35, 33}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {31, 29, 27, 25}; //connect to the column pinouts of the keypad
bool buzzeron;
```

```
long Counter_FLow = 0;
long limit;
float debit, token;
long tokenx;//sisa token
long reload;//isi ulang terakhir
long ram;
bool standby = true;
bool showdebit;
bool beep;
bool iniserver;
bool aktif;
bool tokenvalid;
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS
);

String isi, isi2;
String Noid = "123456";
String Noserver = "+6285810060206";
String PesanModem, Nohp, IsiSms, TanggalSms;
int pos1, pos2, pos3, pos4, pos5, PanjangPesan, autoset;
char Chr;
bool PesanBaru;
bool dibobol = false;
const long interval = 1000;
unsigned long previousMillis = 0;
unsigned long currentMillis;
```

```
double jumlah_Air;
String instr, Noid2, Tanggal, RamString, Datatoken;
long Datatoken2, Datavalid;
int x;
char id1, id2, id3, id4, id5, id6, Nourut, KodeRp, key;
int poslcd;
byte posisivalid;
//////////////////////////////////////////////////////
long token20 = 18;
long token50 = 48;
long token100 = 98;
long token200 = 198;
long token500 = 498;
/////////////////////////////////////////////////////

void setup() {
  lcd.begin(16, 2);
  //lcd.print("OKE");
  Serial.begin(9600);
  Modem.begin(9600);
  pinMode(buzzer, OUTPUT);
  pinMode(solenoid, OUTPUT);
  pinMode(sensor_limit, INPUT_PULLUP);
  pinMode(com, OUTPUT);
  attachInterrupt(0, SFlow1, FALLING );
```

```
  //seting Modem
  Modem.print( "ATE0\r" );//no echo
  if (Modem.find("OK"))Serial.println("NO Echo OK");
  delay(500);
  Modem.print( "AT+CMGF=1\r" );//mode text
  if (Modem.find("OK"))Serial.println("Sms Mode Text OK");
  delay(500);
  Modem.print( "AT+CNMI=2,2,0,0,0\r" );//auto info
  if (Modem.find("OK"))Serial.println("Auto send OK");
  delay(500);
  Modem.print( "AT+CMGD=1.4\r" );//kosongkan memory sms
  delay(2000);
  if (Modem.find("OK"))Serial.println("Memory Sms dihapus");
  PesanModem = "";

  digitalWrite(solenoid, LOW);
  reload = EEPROMReadlong(0);
  tokenx = EEPROMReadlong(4);
  token = float(tokenx);
  dibobol = EEPROM.read(8);
  //kirimsms();
  //reseteeprom();
}

void loop() {
  baca_keypad();
```

```
  sensor();
  Modem_konek();

}

void update_data() {
  if (dibobol == false) {
    if (token <= limit)buzzeron = true; else buzzeron = false;
    if (token == 0)digitalWrite(solenoid, LOW); else digitalWrite(solenoid, HIGH);
    if (standby == true) {
      lcd.setCursor(0, 0);
      lcd.print("Sisa Token");
      lcd.setCursor(0, 1);
      lcd.print(token);
      lcd.print(" ltr");
    }

    if (showdebit == true) {
      lcd.setCursor(0, 0);
      lcd.print("Debit air");
      lcd.setCursor(0, 1);
      lcd.print(debit);
      lcd.print(" l/min  ");
    }

    if (buzzeron == true) {
```

```
      if (beep == LOW) {
        beep = HIGH;
      } else {
        beep = LOW;
      }
      digitalWrite(buzzer, beep);
    }
    else digitalWrite(buzzer, LOW);
  }

}

void sensor () {
  if (digitalRead(sensor_limit) == 0) {
    if (aktif == false) {
      aktif = true;
      //kirimsms;
      Serial.println("Alat dibobol");
      digitalWrite(buzzer, HIGH);
      digitalWrite(solenoid, LOW);
      dibobol = true;
      EEPROM.write(8, 1);
    }
  } else aktif = false;
  currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
```

```
    previousMillis = currentMillis;
    jumlah_Air = ((Counter_FLow * 2.7) / 1000); //1 pulse 2.25ml
    debit = jumlah_Air * 60;
    Counter_FLow = 0;
    if (token > 0)token -= jumlah_Air;
    if (token <= 0)token = 0;
    tokenx = round(token);
    update_data();
    autoset++;
    if (autoset == 10) {
      if (tokenx != EEPROMReadlong(4)) {
        EEPROMWritelong(4, tokenx);
        Serial.println("save");
        autoset = 0;
      }
    }
  }
}


void Modem_konek() {
  if (Modem.available() > 0) {
    Chr = Modem.read();
    PesanModem += Chr;

    if (Chr == '\n') {
```

```
PanjangPesan = PesanModem.length();
if (PesanBaru == true) {
  IsiSms = PesanModem;
  PesanBaru = false;
  PesanModem = "";
  Serial.print("Isi SMS= ");
  Serial.println(IsiSms);
  Modem.print( "AT+CMGd=1,4\r" );//kosongkan memory sms
  delay(3000);
  if (iniserver == true) {
    pos3 = IsiSms.indexOf(Noid);
    pos5 = IsiSms.indexOf("Res");
    String topup = IsiSms.substring(pos3 + 7, IsiSms.length());
    Serial.print("toup= ");
    Serial.println(topup);
    Serial.print("pos5= ");
    Serial.println(pos5);
    //reload = topup.toInt();
    //tokenx += reload;
    //tokenx = tokenx;
    //EEPROMWritelong(0, reload);
    //EEPROMWritelong(4, tokenx);

    iniserver = false;

    if (IsiSms.indexOf("Reset") > 0) {
```

```
      dibobol = false;
      digitalWrite(buzzer, LOW);
      EEPROM.write(8, 0);
      reseteeprom();
      Serial.println("sudah direset");
      lcd.clear();
      lcd.print("Sistem direset");
      delay(2000);
      reload = EEPROMReadlong(0);
      tokenx = EEPROMReadlong(4);
      token = float(tokenx);
      dibobol = EEPROM.read(8);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Sisa Token");
      lcd.setCursor(0, 1);
      lcd.print(token);
      lcd.print(" ltr");
      standby = true;
    }
  }

}
if (PesanModem.indexOf("+CMT:") > 0) {
  PesanBaru = true;
  pos1 = PesanModem.indexOf("+6");
```

```
        pos2 = PesanModem.indexOf(",,");
        String Nohp = PesanModem.substring(pos1, pos2 - 1);
        Serial.print("Nohp= ");
        Serial.println(Nohp);
        if (Nohp == Noserver)iniserver = true; else iniserver = false;
        String TanggalSms = PesanModem.substring(pos2 + 3, pos2 + 20);
        Serial.print("Waktu SMS= ");
        Serial.println(TanggalSms);
        PesanModem = "";
      }
    }
  }
}

void SFlow1()
{
  Counter_FLow ++;
}

void kirimsms() {
  Modem.flush();
  Modem.println("AT+CMGS=085810060206");
  delay(500);
  Modem.println("Terjadi Perusakan Modul");
  Modem.write(26);
  Modem.println();
```

```
  delay(1000);

  Serial.println("sms send");
}

void baca_keypad() {
  key = keypad.getKey();
  if (key) {
    standby = false;
    poslcd++;
    if (key == 'A') limit = isi.toInt();
    isi += key;
    lcd.clear();
    lcd.print(isi);
    if (poslcd > 16) {
      isi2 += key;
      lcd.setCursor(0, 1);
      lcd.print(isi2);
    }
    if (key == 'A') {
      isi = "22D";
      key = 'D';
      poslcd = 0;
    }
    if (key == 'D') {
```

```
lcd.clear();
//bagian menu
//00D Tampilkan sisa token
//11D Tampilkan id token
//22D Tampilkan batas minimal untuk alaram
//33D Tampilkan debit air liter/detik
//44D Tampilan  isi ulang terakhir

if (isi == "00D") {
  lcd.setCursor(0, 0);
  lcd.print("Sisa Token");
  lcd.setCursor(0, 1);
  lcd.print(token);
  lcd.print(" ltr");
  standby = true;
}
if (isi == "11D") {
  lcd.setCursor(0, 0);
  lcd.print("NoId");
  lcd.setCursor(0, 1);
  lcd.print(Noid);
}
if (isi == "22D") {
  lcd.setCursor(0, 0);
  lcd.print("Batas alaram");
  lcd.setCursor(0, 1);
```

```
  lcd.print(limit);
}

if (isi == "33D") {
  lcd.setCursor(0, 0);
  lcd.print("Debit air");
  lcd.setCursor(0, 1);
  lcd.print(debit);
  lcd.print(" l/min  ");
  showdebit = true;
} else showdebit = false;

if (isi == "44D") {
  lcd.setCursor(0, 0);
  lcd.print("Isi terakhir");
  lcd.setCursor(0, 1);
  lcd.print(reload);
  lcd.print(" ltr");
}
if (poslcd == 21) {
  instr = isi + isi2;
  extracttoken();
  if (Noid == Noid2) {

    tokenvalid = true;
    //cek status token masih valid tidak
```

```
for (int aa = 10; aa <= 200; aa += 4) {
  Datavalid = EEPROMReadlong(aa);
  if (Datavalid == Datatoken2)tokenvalid = false;
}
if (tokenvalid == true) {
  Serial.print("Berhasil=");
  posisivalid = EEPROM.read(9);
  if (posisivalid == 0 || posisivalid == 255) EEPROM.write(9, 10); else
EEPROM.write(9, posisivalid + 4);
  posisivalid = EEPROM.read(9);
  EEPROMWritelong(posisivalid, Datatoken2);
  Serial.println(posisivalid);
  //prosedure isi ulang
  if (KodeRp == '1')reload = token20; else if (KodeRp == '2')reload =
token50; else if (KodeRp == '3')reload = token100; else if (KodeRp == '4')reload
= token200; else if (KodeRp == '5')reload = token500;
  token += reload;
  tokenx = token;
  iniserver = false;
  EEPROMWritelong(0, reload);
  EEPROMWritelong(4, tokenx);
  lcd.clear();
  lcd.print("Top up Berhasil");
  lcd.setCursor(0, 1);
  lcd.print(reload);
  delay(2000);
  lcd.clear();
```

```
        lcd.setCursor(0, 0);
        lcd.print("Sisa Token");
        lcd.setCursor(0, 1);
        lcd.print(token);
        lcd.print(" ltr");
        standby = true;
      } else {
        lcd.clear();
        lcd.print("Salah");
        delay(2000);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Sisa Token");
        lcd.setCursor(0, 1);
        lcd.print(token);
        lcd.print(" ltr");
        standby = true;
      }
      //
    }
  }
  isi = ""; isi2 = ""; poslcd = 0;
}
delay(500);


}
```

```
}



//tulis data long ke eeprom
void EEPROMWritelong(int address, long value)
{
  //Decomposition from a long to 4 bytes by using bitshift.
  //One = Most significant -> Four = Least significant byte
  byte four = (value & 0xFF);
  byte three = ((value >> 8) & 0xFF);
  byte two = ((value >> 16) & 0xFF);
  byte one = ((value >> 24) & 0xFF);

  //Write the 4 bytes into the eeprom memory.
  EEPROM.write(address, four);
  EEPROM.write(address + 1, three);
  EEPROM.write(address + 2, two);
  EEPROM.write(address + 3, one);
}
//baca data long ke eeprom
long EEPROMReadlong(long address)
{
  //Read the 4 bytes from the eeprom memory.
  long four = EEPROM.read(address);
  long three = EEPROM.read(address + 1);
```

```
  long two = EEPROM.read(address + 2);
  long one = EEPROM.read(address + 3);

  //Return the recomposed long by using bitshift.
  return ((four << 0) & 0xFF) + ((three << 8) & 0xFFFF) + ((two << 16) &
0xFFFFFF) + ((one << 24) & 0xFFFFFFFF);
}

void extracttoken() {
  //dekript id 1
  RamString = instr.substring(10, 12);
  x = RamString.toInt();
  id1 = char(x);

  //dekript id 2
  RamString = instr.substring(16, 18);
  x = RamString.toInt();
  id2 = char(x);
  //dekript id 3
  id3 = instr.charAt(7);
  //dekript id 4
  RamString = instr.substring(0, 2);
  x = RamString.toInt();
  id4 = char(x);
  //dekript id 5
  RamString = instr.substring(12, 14);
```

```
  x = RamString.toInt();

  id5 = char(x);

  //dekript id 6

  id6 = instr.charAt(4);

  Noid2 = id1; Noid2 += id2; Noid2 += id3; Noid2 += id4; Noid2 += id5; Noid2
+= id6;

  //Serial.print("Noid2=");
Serial.print(id1);Serial.print(id2);Serial.print(id3);Serial.print(id4);Serial.print(id5
);Serial.print(id6);


  //dekript tanggal

  Tanggal = instr.substring(18, 20);

  Tanggal += instr.substring(8, 10);

  Tanggal += instr.substring(2, 4);

  //dekript nomor urut

  RamString = instr.substring(14, 16);

  x = RamString.toInt();

  Nourut = char(x);

  //dekript kode rupiah

  RamString = instr.substring(5, 7);

  x = RamString.toInt();

  KodeRp = char(x);


  // Serial.print("Noid2="); Serial.println(Noid2);

  // Serial.print("Tanggal="); Serial.println(Tanggal);

  //Serial.print("Isike="); Serial.println(Nourut);

  // Serial.print("Kode="); Serial.println(KodeRp);
```

```
  instr = "";
  Datatoken = Tanggal; Datatoken += KodeRp; Datatoken += Nourut;
  Datatoken2 = Datatoken.toInt();
}

void reseteeprom() {
  for (int ab = 0; ab <= EEPROM.length(); ab ++) {
    EEPROM.write(ab, 0);
  }
}
```

Lampiran 7

Program Sistem Token

```
unit generate;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type
  TFORM1 = class(TForm)
    eid: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    cbrp: TComboBox;
    cbu: TComboBox;
    Button1: TButton;
    etanggal: TEdit;
    Label4: TLabel;
    Button2: TButton;
    etoken: TEdit;
    LX: TLabel;
    procedure Button1Click(Sender: TObject);
```

```
    procedure Button2Click(Sender: TObject);
    procedure cbrpSelect(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FORM1: TFORM1;

implementation

{$R *.dfm}

procedure TFORM1.Button1Click(Sender: TObject);
begin
//FORMAT ENKRIPSI
etoken.Text:='';
//1-4
etoken.Text:=etoken.Text+inttostr(ord(eid.Text[4]));  //id4
etoken.Text:=etoken.Text+copy(etanggal.Text, 5, 2)+' ';  //tahun
//5-8
etoken.Text:=etoken.Text+copy(eid.Text, 6, 1); //id6
etoken.Text:=etoken.Text+inttostr(ord(LX.caption[1]));//nilai token
```

```
etoken.Text:=etoken.Text+copy(eid.Text, 3, 1)+' '; //id3
//9-12
etoken.Text:=etoken.Text+copy(etanggal.Text, 3, 2);  //bulan
etoken.Text:=etoken.Text+inttostr(ord(eid.Text[1]))+' '; //id1
//13-16
etoken.Text:=etoken.Text+inttostr(ord(eid.Text[5])); //id5
etoken.Text:=etoken.Text+inttostr(ord(cbu.Text[1]))+' '; //no urut
//17-20
etoken.Text:=etoken.Text+inttostr(ord(eid.Text[2])); //id2
etoken.Text:=etoken.Text+copy(etanggal.Text, 1, 2);  //tanggal
end;

procedure TFORM1.Button2Click(Sender: TObject);
begin
etanggal.Text:=formatdatetime('ddmmyy',date);
end;

procedure TFORM1.cbrpSelect(Sender: TObject);
begin
if cbrp.ItemIndex=0 then lx.Caption:='1';
if cbrp.ItemIndex=1 then lx.Caption:='2';
if cbrp.ItemIndex=2 then lx.Caption:='3';
if cbrp.ItemIndex=3 then lx.Caption:='4';
if cbrp.ItemIndex=4 then lx.Caption:='5';
end;
```

```
procedure TFORM1.FormShow(Sender: TObject);
begin
button2.Click;
end;

end.
```