



**RANCANG BANGUN SISTEM
PRABAYAR PADA PDAM
BERBASIS ARDUINO UNO R3**

TUGAS AKHIR

**Program Studi
S1 Sistem Komputer**

**INSTITUT BISNIS
DAN INFORMATIKA**

**stikom
SURABAYA**

Oleh:

MOCH ADIB MUSYAFAT

09410200057

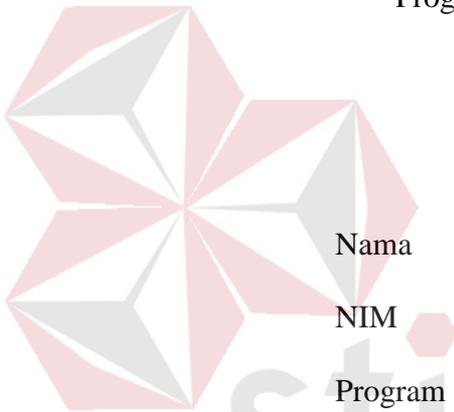
**SEKOLAH TINGGI MANAJEMEN INFORMATIKA &
TEKNIK KOMPUTER SURABAYA
2014**

RANCANG BANGUN SISTEM PRABAYAR PADA PDAM
BERBASIS ARDUINO UNO R3

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



Disusun Oleh :

Nama : Moch. Adib Musyafa'

NIM : 09.41020.0057

Program : S1 (Strata Satu)

Jurusan : SistemKomputer

SEKOLAH TINGGI

MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER

SURABAYA

2014



“A Words alone..

...can make an entire army withdraw”

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA



Kupersembahkan kepada

Bapak, Ibu dan adik-adikku tercinta

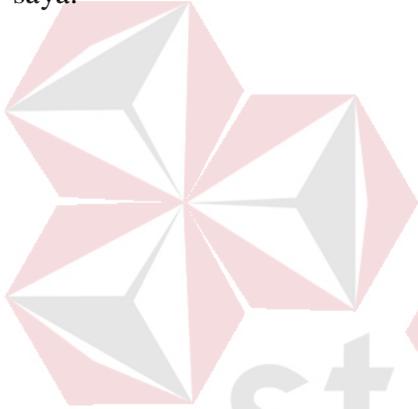
Bapak Dosen Wali yang selalu membimbingku

Beserta semua orang yang menyayangiku

INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA

PERNYATAAN

Dengan ini menyatakan dengan benar, bahwa Tugas Akhir ini adalah asli karya saya, bukan plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain yang ada dalam tugas akhir ini adalah semata hanya rujukan yang dicantumkan dalam daftar pustaka saya. Apabila dikemudian hari ditemukan adanya tindakan plagiat pada karya tugas akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.



INSTITUT BISNIS
DAN INFORMATIKA

Surabaya, 12 September 2014

Meterai
Rp. 6000,-

Moch. Adib Musyafa'
NIM. 09410200057

Tugas Akhir

RANCANG BANGUN SISTEM PRABAYAR PADA PDAM BERBASIS ARDUINO UNO R3

Dipersiapkan dan disusun oleh

Moch. Adib Musyafa'

NIM :09.41020.0057

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

pada : September 2014

Susunan Dewan Penguji

Pembimbing

I. Susijanto Tri Rasmana, S.Kom, M.T

II. Pauladie Susanto, S.Kom

Penguji

I. Dr. Jusak

II. I Dewa Gedhe Rai Mardiana, S.Kom

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Pantjawati Sudarmaningtyas, S.Kom.,M.Eng
Pembantu Ketua Bidang Akademik

ABSTRAK

Saat ini banyak PDAM yang masih menggunakan sistem pencatatan meter air secara manual. Petugas datang langsung ke rumah-rumah pelanggan untuk mencatat angka yang ada di meter air. Tapi dengan cara ini timbul banyak masalah yang merugikan pelanggan maupun pihak PDAM. Masalah yang sering terjadi adalah saat proses pencatatan meter air. Dalam proyek tugas akhir ini akan dibuat suatu *prototype* sistem pembayaran prabayar pada PDAM.

Sistem ini terbagi menjadi dua. Yang pertama adalah server, digunakan untuk *generate* pulsa prabayar yang dilengkapi dengan sistem enkripsi. Yang kedua miniatur alat. Setelah melakukan proses dekripsi pulsa kemudian sensor *flow water* melakukan counter untuk menghitung jumlah air yang keluar sesuai dengan pulsa. Ketika pulsa habis sistem akan memberi perintah pada *solenoid valve* untuk menutup keran. Sistem prabayar ini memiliki tingkat keakuratan sebesar 97.44%.

KATA PENGANTAR

Pertama-tama penulis panjatkan puji dan syukur kepada Allah SWT karena berkat, rahmat, dan karuniaNya penulis dapat menyelesaikan penulisan Tugas Akhir ini dengan sebaik-baiknya. Penulis mengambil judul “Rancang Bangun Sistem Prabayar Pada PDAM” ini sebagai salah satu syarat dalam menyelesaikan Tugas Akhir di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya.

Dalam pelaksanaan Tugas Akhir serta pembuatan laporan Tugas Akhir ini, banyak sekali pihak yang telah membantu penulis sehingga Tugas Akhir dapat terlaksana dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada :

1. Bapak, Ibuk, Alm. Ipud, dan Adek Galuh yang menjadi semangat bagi penulis untuk menyelesaikan Tugas Akhir ini.
2. Prita Isnaniyah Ramadhany, yang tanpa henti memberikan dukungan dan semangat dengan penuh kasih sayang kepada penulis.
3. Seluruh guru yang telah mendidik penulis mulai kecil hingga saat ini sehingga penulis mendapatkan ilmu pengetahuan yang dapat menjadikan penulis menjadi seperti saat ini.
4. Pimpinan STIKOM Surabaya yang telah banyak memberikan motivasi serta teladan yang dapat membantu penulis selama menempuh pembelajaran hingga saat ini.

5. Bapak Dr. Jusak, selaku Kepala Program Studi Sistem Komputer STIKOM Surabaya yang telah membantu serta mendukung setiap kegiatan sehingga pelaksanaan Tugas Akhir ini dapat berjalan dengan baik.
6. Bapak Susijanto Tri Rasmana, S.Kom., M.Eng., dan bapak Pauladie Susanto, S.Kom., selaku dosen pembina pertama dan kedua sehingga penulis dapat melaksanakan Tugas Akhir ini dengan baik.
7. Bapak Susijanto Tri Rasmana, S.Kom., M.Eng., selaku dosen wali dan para dosen lainnya yang telah membantu penulis jika mengalami berbagai macam kesulitan sehingga penulis dapat termotivasi untuk terus berusaha hingga Tugas Akhir ini terlaksana sesuai dengan harapan.
8. Teman-teman penulis yang telah mendampingi, memberi tempat saat penulis membutuhkan yang juga membantu dalam pelaksanaan Tugas Akhir ini.

Banyak hal dalam laporan Tugas Akhir ini yang masih perlu diperbaiki lagi. Oleh karena itu penulis mengharapkan saran dan kritik yang dapat membangun dari semua pihak agar dapat menyempurnakan penulisan ini kedepannya. Penulis juga memohon maaf yang sebesar-besarnya jika terdapat kata-kata yang salah serta menyinggung perasaan pembaca. Akhir kata penulis ucapkan banyak-banyak terima kasih yang sebesar-besarnya kepada para pembaca, semoga tulisan ini dapat bermanfaat bagi para pembaca.

Surabaya, September 2014

Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI.....	5
2.1 PDAM.....	5
2.2 <i>Caesar Chiper</i>	5
2.3 Arduino Uno R3	7
2.3.1 Daya (Power)	8
2.3.2 Memori.....	10
2.3.3 Input Output.....	10
2.3.4 <i>Software</i> Arduino IDE	11

2.4	<i>Water Flow Sensor</i>	12
2.5	Modul LCD 16x2.....	14
2.6	Solenoid Valve	15
2.7	Rangkaian <i>Driver Relay</i>	16
2.8	RTC	17
BAB III METODE PENELITIAN.....		20
3.1	Alat dan Bahan Penelitian	20
3.1.1	Alat Penelitian	20
3.1.2	Bahan Penelitian	21
3.2	Jalan Penelitian.....	21
3.2.1	Pengumpulan Literatur	22
3.2.2	Perancangan Perangkat Keras	22
3.2.3	Perancangan Perangkat Lunak.....	38
3.3	Langkah Pengujian	35
3.3.1	Pengujian Perangkat Keras.....	35
3.4.1	Pengujian Perangkat Lunak.....	38
BAB IV HASIL DAN PEMBAHASAN		43
4.1	Pengujian <i>Board Arduino Uno r3</i>	43
4.1.1	Hasil Pengujian <i>Board Arduino Uno R3</i>	43
4.1.2	Pembahasan	43
4.2	Pengujian LCD	44
4.2.1	Hasil Pengujian LCD	44
4.2.2	Pembahasan	45

4.3	Pengujian RTC	45
4.3.1	Hasil Pengujian RTC	45
4.3.2	Pembahasan	46
4.4	Pengujian <i>Water Flow Sensor</i>	46
4.4.1	Hasil Pengujian <i>Water Flow Sensor</i>	46
4.4.2	Pembahasan	47
4.5	Pengujian Solenoid Valve	48
4.5.1	Hasil Pengujian Solenoid Valve	48
4.5.2	Pembahasan	49
4.6	Pengujian Server.....	49
4.6.1	Hasil Pengujian Server.....	49
4.6.2	Pembahasan	53
4.7	Pengujian Algoritma Dekripsi Pada Arduino.....	54
4.7.1	Hasil Pengujian Algoritma Dekripsi Pada Arduino.....	54
4.7.2	Pembahasan	56
4.8	Pengujian Keseluruhan.....	56
4.8.1	Hasil Pengujian Keseluruhan.....	57
4.8.2	Pembahasan	57
BAB V PENUTUP.....		59
5.1	Kesimpulan.....	59
5.2	Saran.....	59
DAFTAR PUSTAKA		60

DAFTAR TABEL

	Halaman
Tabel 2.1 Deskripsi Arduino Uno R3	8
Tabel 2.2 Deskripsi <i>Water Flow Sensor</i>	13
Tabel 3.1 Pengujian Arduino Uno R3.....	36
Tabel 3.2 Pengujian LCD 16x2.....	36
Tabel 3.3 Pengujian RTC.....	37
Tabel 3.4 Pengujian <i>Water Flow Sensor</i>	37
Tabel 3.5 Pengujian Relay <i>Solenoid Valve</i>	38
Tabel 3.6 Pengujian Rangkaian <i>Power Supply</i> 12V	38
Tabel 3.7 Pengujian Program Enkripsi Pada Server	39
Tabel 3.8 Pengujian Program Dekripsi Pada Alat	39
Tabel 3.9 Pengujian Sistem Keseluruhan.....	40
Tabel 4.1 Hasil Pengujian Keluaran Pin Digital Arduino (Pin 4).....	41
Tabel 4.2 Hasil Pengujian Sensor	46
Tabel 4.2 Hasil Pengujian Sistem	54

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Arduino Uno R3 Sisi Depan (Kiri) dan Belakang(Kanan)	8
Gambar 2.2 Tampilan <i>Software</i> Arduino IDE	11
Gambar 2.3 <i>Water Flow Sensor</i>	13
Gambar 2.4 Konfigurasi pin LCD.....	14
Gambar 2.5 Solenoid Valve	16
Gambar 2.6 Layout IC DS1307	18
Gambar 3.1 Blok Diagram Sistem	22
Gambar 3.2 Rangkaian Arduino Uno R3	23
Gambar 3.3 Rangkaian LCD.....	24
Gambar 3.4 Rangkaian RTC	25
Gambar 3.5 Rangkaian Relay <i>Solenoid Valve</i>	26
Gambar 3.6 Rangkaian <i>Power Supply</i> 12V.....	27
Gambar 3.7 Perancangan Kode Pulsa	27
Gambar 3.8 Tampilan Program Enkripsi Server.....	29
Gambar 3.9 <i>Flowchart</i> Enkripsi Pulsa.....	30
Gambar 3.10 <i>Flowchart</i> Alat	32
Gambar 3.11 <i>Flowchart</i> Dekripsi	33
Gambar 3.12 <i>Flowchart</i> Counter Pulsa	35
Gambar 4.1 Hasil Pengujian LCD	42
Gambar 4.2 Tampilan Pengujian RTC.....	44
Gambar 4.3 <i>Water Flow Sensor</i>	45

Gambar 4.4 Solenoid Valve	47
Gambar 4.5 Tampilan program Server.....	48
Gambar 4.6 Hasil Enkripsi.....	49
Gambar 4.7 Hasil Dekripsi.....	51
Gambar 4.8 Input Pulsa.....	52
Gambar 4.9 Penukaran Posisi Karakter	53
Gambar 4.10 Hasil dekripsi.....	53
Gambar 4.11 Tampilan saat Pengujian Sistem	54



DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Kode Program Mikrokontroler Arduino Uno.....	61
Lampiran 2. Kode Program Server dengan Visual Basic 6.0	63
Lampiran 3. Biodata Penulis	64



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Saat ini banyak PDAM yang masih menggunakan sistem pencatatan meter air secara manual. Petugas datang langsung ke rumah-rumah pelanggan untuk mencatat angka yang ada di meter air. Tapi dengan cara ini timbul banyak masalah yang merugikan pelanggan maupun pihak PDAM.

Kerugian yang terjadi pada pelanggan dikarenakan seringnya terjadi kesalahan pencatatan meter air. Dari masalah-masalah tersebut PDAM membuat solusi dengan menciptakan sistem *barcode* dalam pencatatan meteran sejak april 2012. Dengan menggunakan sistem baru tersebut, 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 di tidak lagi memasukkan angka meteran dengan perkiraan pemakaian per bulan. (Ronalyw, 2013)

Namun sistem tersebut bisa berjalan hanya jika petugas datang langsung ke lokasi. Dengan memanfaatkan teknologi yang sudah ada bisa diciptakan sistem perhitungan dan pembayaran Prabayar yang lebih mudah dan efisien. Yaitu

dengan sebuah server yang bisa membuat deretan angka yang berisi nilai pulsa air dan ID pelanggan sehingga meter air tinggal membaca angka-angka tersebut untuk menjalankannya. Dan akan berhenti otomatis ketika pulasanya habis.

Dengan pemakaian sistem Prabayar, PDAM dan pelanggan sama-sama diuntungkan. Untuk PDAM adalah efisiensi petugas karena tidak perlu pencatatan langsung ke pelanggan dan juga tidak akan terjadi kesalahan pencatatan. Sedangkan untuk pelanggan akan lebih mudah untuk memantau pengeluaran pada penggunaan air PDAM.

1.2 Rumusan Masalah

Dari masalah di atas dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana merancang server untuk *generate* pulsa prabayar?
2. Bagaimana membuat rangkaian untuk membaca dan menghitung pulsa prabayar?

1.3 Batasan Masalah

Batasan masalah pada semua pengaturan peralatan / parameter yang digunakan dalam Tugas Akhir ini adalah:

1. Sistem kontrol menggunakan Arduino Uno R3.
2. Menggunakan *water flow sensor*.
3. Sistem enkripsi menggunakan metode caesar chipper.
4. ID untuk *user* dibatasi 2 digit dan untuk *value* pulsa 5 digit.
5. Kran otomatis menggunakan solenoid valve DC 24 Volt.
6. Tekanan air tidak ditentukan.

1.4 Tujuan

Adapun tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Merancang sebuah server yang digunakan untuk *generate* pulsa Prabayar.
2. Membuat sebuah rangkaian minimum sistem untuk membaca dan menjalankan sistem Prabayar.

1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini secara garis besar tersusun dari 5 (lima) bab, yang sebagai berikut :

BAB I. PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang masalah, perumusan masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

BAB II. LANDASAN TEORI

Pada Bab ini akan dibahas teori penunjang dari permasalahan, yaitu mengenai PDAM, Kriptografi, Arduino Uno R3, sensor *flow water*, modul LCD, *Solenoid Valve* DC 24 Volt, , rangkaian *driver* relay, rangkaian RTC.

BAB III. METODE PENELITIAN

Pada Bab ini akan dibahas tentang alat dan bahan yang digunakan dalam penelitian. Perancangan perangkat keras meliputi rangkaian minimum sistem Arduino Uno R3, rangkaian LCD, rangkaian RTC, rangkaian relay *solenoid valve* dan rangkaian *power supply* 12V. Perancangan perangkat lunak meliputi perancangan kode pulsa, perancangan program enkripsi pada server, perancangan program dekripsi pada alat dan perancangan sistem secara keseluruhan.

BAB IV. HASIL DAN PEMBAHASAN

Pada Bab ini akan dibahas mengenai hasil pengujian dari setiap rancangan yang ada pada BAB III. Yaitu hasil pengujian Arduino Uno R3, hasil pengujian rangkaian LCD, hasil pengujian rangkaian RTC, hasil pengujian *water flow sensor*, hasil pengujian *solenoid valve*, hasil pengujian server, hasil pengujian algoritma dekripsi pada alat dan hasil pengujian sistem keseluruhan.

BAB V. PENUTUP

Berisi kesimpulan yang didapat dari hasil penelitian berdasarkan rumusan masalah serta saran untuk perkembangan penelitian selanjutnya.



BAB II

LANDASAN TEORI

2.1 PDAM

PDAM atau Perusahaan Daerah Air Minum merupakan salah satu unit usaha milik daerah, yang bergerak dalam distribusi air bersih bagi masyarakat umum. PDAM terdapat di setiap provinsi, kabupaten, dan kotamadya di seluruh Indonesia. PDAM merupakan perusahaan daerah sebagai sarana penyedia air bersih yang diawasi dan dimonitor oleh aparat aparat eksekutif maupun legislatif daerah.

Pada umumnya sistem pembayaran pada PDAM adalah pasca bayar. Pelanggan membayar sebulan sekali sesuai dengan catatan meter air. Pencatatan meter air dilakukan secara manual oleh petugas PDAM dengan mendatangi satu persatu lokasi meter air pelanggan. Namun di beberapa daerah sudah ada yang menggunakan *barcode*. Sistem tersebut digunakan untuk mencegah kesalahan pada pencatatan meter air namun tetap dilakukan secara manual oleh petugas PDAM.

2.2 Caesar Chiper

Dalam kriptografi, sandi *caesar*, atau sandi geser, kode caesar atau Geseran caesar adalah salah satu teknik enkripsi paling sederhana dan paling terkenal. Sandi ini termasuk sandi substitusi dimana setiap huruf pada teks terang (*plaintext*) digantikan oleh huruf lain yang memiliki selisih posisi tertentu dalam alfabet. Misalnya, jika menggunakan geseran 3, W akan menjadi Z, I menjadi L, dan K menjadi N sehingga teks terang "wiki" akan menjadi "ZLNL" pada teks tersandi. Nama *Caesar* diambil dari *Julius Caesar*, jenderal, konsul, dan diktator

Romawi yang menggunakan sandi ini untuk berkomunikasi dengan para panglimanya.

Langkah enkripsi oleh sandi Caesar sering dijadikan bagian dari penyandian yang lebih rumit, seperti sandi *Vigenère*, dan masih memiliki aplikasi modern pada sistem ROT13. Pada saat ini, seperti halnya sandi substitusi alfabet tunggal lainnya, sandi Caesar dapat dengan mudah dipecahkan dan praktis tidak memberikan kerahasiaan bagi pemakainya.

Cara kerja sandi ini dapat diilustrasikan dengan membariskan dua set alfabet; alfabet sandi disusun dengan cara menggeser alfabet biasa ke kanan atau ke kiri dengan angka tertentu (angka ini disebut kunci). Misalnya sandi Caesar dengan kunci 3, adalah sebagai berikut:

Alfabet Biasa: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Alfabet Sandi: DEFGHIJKLMNOPQRSTUVWXYZABC

Untuk menyandikan sebuah pesan, cukup mencari setiap huruf yang hendak disandikan di alfabet biasa, lalu tuliskan huruf yang sesuai pada alfabet sandi. Untuk memecahkan sandi tersebut gunakan cara sebaliknya. Contoh penyandian sebuah pesan adalah sebagai berikut.

teks terang: KIRIM PASUKAN KE KIRI

teks tersandi: NLULP SDVXNDQ NH NLUL

Proses penyandian (enkripsi) dapat secara matematis menggunakan operasi modulus dengan mengubah huruf-huruf menjadi angka, A = 0, B = 1, ..., Z = 25. Sandi (E_n) dari "huruf" x dengan geseran n secara matematis dituliskan dengan,

$$E_n(x) = (x + n) \pmod{26} \dots\dots\dots(2.1)$$

Sedangkan pada proses pemecahan kode (dekripsi), hasil dekripsi (D_n) adalah

$$D_n(x) = (x - n) \pmod{26} \dots\dots\dots(2.2)$$

Setiap huruf yang sama digantikan oleh huruf yang sama di sepanjang pesan, sehingga sandi Caesar digolongkan kepada, *substitusi monoalfabetik*, yang berlawanan dengan *substitusi polialfabetik*. (Miftah, Zein 2010)

2.3 Arduino Uno R3

Arduino Uno R3 adalah *board* berbasis mikrokontroler pada ATmega328. *Board* ini memiliki 14 digital input / output pin (dimana 6 pin dapat digunakan sebagai *output PWM*), 6 input *analog*, 16 MHz *osilator* kristal, koneksi USB, jack listrik tombol *reset*. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya.

Board Arduino Uno R3 memiliki fitur-fitur baru sebagai berikut : 1,0 pinout: tambah SDA dan SCL pin yang dekat ke pin aref dan dua pin baru lainnya ditempatkan dekat ke pin RESET, dengan IO REF yang memungkinkan sebagai *buffer* untuk beradaptasi dengan tegangan yang disediakan dari board sistem. Pengembangannya, sistem akan lebih kompatibel dengan *Prosesor* yang menggunakan AVR, yang beroperasi dengan 5V dan dengan Arduino Karena yang beroperasi dengan 3.3V. Yang kedua adalah pin tidak terhubung, yang disediakan untuk tujuan pengembangannya. (Banzi, 2009)



Gambar 2.1 Arduino Uno R3 Sisi Depan (Kiri) dan Belakang(Kanan)

Sumber: (arduino.cc, 2014)

Secara umum Arduino Uno R3 terdiri dari dua bagian, yaitu:

1. *Hardware*: papan *input/output* (I/O)
2. *Software*: *software* Arduino Uno R3 meliputi IDE untuk menulis program, *driver* untuk koneksi dengan komputer, contoh program dan *library* untuk pengembangan program. (Djuandi, 2011)

Tabel 2.1 Deskripsi Arduino Uno R3

Mikrokontroler	ATmega328
Tegangan pengoperasian	5V
Tegangan input yang disarankan	7-12V
Batas tegangan input	6-20V
Jumlah pin I/O digital	14 (6 di antaranya menyediakan keluaran PWM)
Jumlah pin input analog	6
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Memori Flash	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

2.3.1 Daya (Power)

Arduino Uno dapat disuplai melalui koneksi USB atau dengan sebuah power suplai eksternal. Suplai eksternal (non-USB) dapat diperoleh dari sebuah

adaptor AC ke DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan sebuah *center-positive* plug yang panjangnya 2,1 mm ke power jack dari *board*. Kabel lead dari sebuah battery dapat dimasukkan dalam *header/kepala pin Ground* (Gnd) dan pin Vin dari konektor *POWER*.

Board Arduino Uno dapat beroperasi pada sebuah suplai eksternal 6 sampai 20 Volt. Jika disuplai dengan yang lebih kecil dari 7 V, kiranya pin 5 Volt mungkin mensuplai kecil dari 5 Volt dan *board* Arduino Uno bisa menjadi tidak stabil. Jika menggunakan suplai yang lebih dari besar 12 Volt, *voltage* regulator bisa kelebihan panas dan membahayakan *board* Arduino Uno. *Range* yang direkomendasikan adalah 7 sampai 12 Volt. (arduino.cc, 2013)

Pin-pin dayanya adalah sebagai berikut:

1. VIN. Tegangan input ke Arduino *board* ketika *board* sedang menggunakan sumber suplai eksternal (seperti 5 Volt dari koneksi USB atau sumber tenaga lainnya yang diatur). Kita dapat menyuplai tegangan melalui pin ini, atau jika penyuplaian tegangan melalui *power jack*, aksesnya melalui pin ini.
2. 5V. Pin output ini merupakan tegangan 5 Volt yang diatur dari regulator pada *board*. *Board* dapat disuplai dengan salah satu suplai dari DC *power jack* (7-12V), USB *connector* (5V), atau pin VIN dari *board* (7-12). Penyuplaian tegangan melalui pin 5V atau 3,3V *membypass* regulator, dan dapat membahayakan *board*. Hal itu tidak dianjurkan.
3. 3V3. Sebuah suplai 3,3 Volt dihasilkan oleh regulator pada *board*. Arus maksimum yang dapat dilalui adalah 50 mA.
4. GND. Pin *ground*.

2.3.2 Memori

ATmega328 mempunyai 32 KB yang bersifat *non-volatile*, digunakan untuk menyimpan program yang dimuat dari komputer. (dengan 0,5 KB digunakan untuk *bootloader*). ATmega 328 juga mempunyai 2 KB SRAM yang *volatile* (hilang saat daya dimatikan), digunakan oleh variable-variabel di dalam program. dan 1 KB EEPROM (yang dapat dibaca dan ditulis (*RW/read and written*)). (arduino.cc, 2013)

2.3.3 Input dan Output

Setiap 14 pin digital pada Arduino Uno dapat digunakan sebagai input dan output. Fungsi-fungsi tersebut beroperasi di tegangan 5 Volt. Setiap pin dapat memberikan atau menerima suatu arus maksimum 40 mA dan mempunyai sebuah resistor pull-up (terputus secara default) 20-50 kOhm. Selain itu, beberapa pin mempunyai fungsi-fungsi sebagai berikut:

1. Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan memancarkan (TX) serial data TTL (*Transistor-Transistor Logic*). Kedua pin ini dihubungkan ke pin-pin yang sesuai dari chip Serial Atmega8U2 USB-ke-TTL.
2. External Interrupts: 2 dan 3. Pin-pin ini dapat dikonfigurasi untuk dipicu sebuah interrupt (gangguan) pada suatu nilai rendah, suatu kenaikan atau penurunan yang besar, atau suatu perubahan nilai.
3. PWM: 3, 5, 6, 9, 10, dan 11. Memberikan 8-bit PWM *output* dengan fungsi *analogWrite()*.
4. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin-pin ini mensupport komunikasi SPI menggunakan SPI *library*.

5. LED: 13. Ada sebuah LED yang terpasang, terhubung ke pin digital 13. Ketika pin bernilai *HIGH* LED menyala, ketika pin bernilai *LOW* LED mati.

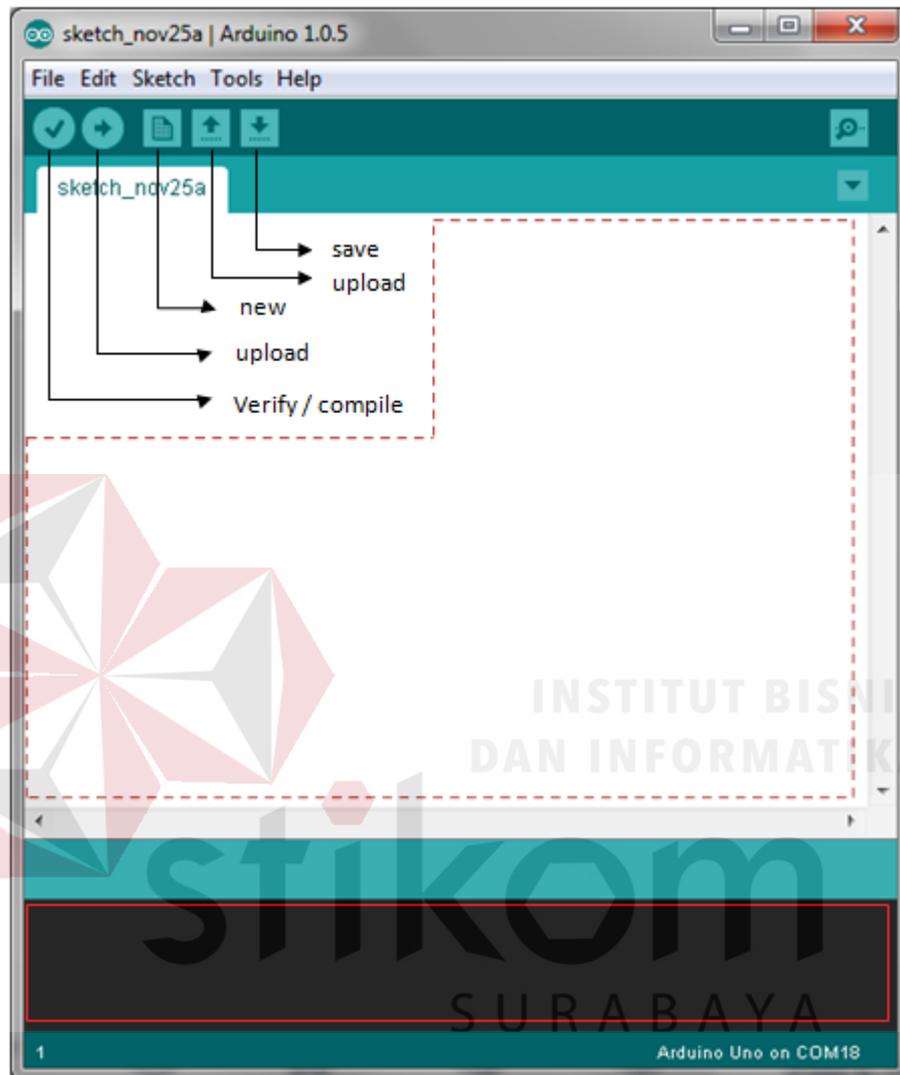
Arduino Uno mempunyai 6 input analog, diberi label A0 sampai A5, setiapnya memberikan resolusi 10 bit. Secara default, 6 input analog tersebut mengukur tegangan dari *ground* sampai tegangan 5 Volt, dengan itu memungkinkan untuk mengganti batas atas dari rangnya dengan menggunakan pin AREF dan fungsi *analogReference()*. Di sisi lainnya, beberapa pin mempunyai fungsi spesifik yaitu pin A4 atau SDA dan pin A5 atau SCL. Mendukung komunikasi TWI dengan menggunakan *Wire library*. Ada sepasang pin lainnya pada board yaitu AREF referensi tegangan untuk input analog. Digunakan dengan *analogReference()*, dan reset untuk mereset mikrokontroler. (arduino.cc, 2013)

2.3.4 Software Arduino IDE

Arduino IDE adalah *software* yang ditulis menggunakan java dan berdasarkan pengolahan seperti, avr-gcc, dan perangkat lunak *open source* lainnya (Djuandi, 2011). Arduino IDE terdiri dari:

1. *Editor program*, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa processing.
2. *Verify / Compiler*, sebuah modul yang mengubah kode program (bahasa processing) menjadi kode biner. Bagaimanapun sebuah mikrokontroller tidak akan bisa memahami bahasa processing, yang dipahami oleh mikrokontroller adalah kode biner.

3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memori mikrokontroler di dalam papan arduino.



Gambar 2.2 Tampilan *Software* Arduino IDE.

Pada Gambar 2.2 terdapat *menu bar*, kemudian *toolbar* dibawahnya, dan sebuah area putih untuk *editing sketch*, area hitam dapat kita sebut sebagai *progress area*, dan paling bawah dapat kita sebut sebagai "*status bar*".

2.4 *Water Flow Sensor*

Water flow sensor ini terdiri atas katup plastik, rotor air, dan sebuah sensor hall-effect. Ketika air mengalir melalui pipa dalam sensor ini, maka akan mengenai rotor, dan membuatnya berputar. Kecepatan putar rotor akan berubah

ketika kecepatan aliran air berubah pula. *Output* dari sensor *hall-effect* akan sebanding dengan pulsa yang di *generate* rotor. (Anonim, 2013)

Tabel 2.2 Deskripsi *Water Flow Sensor*

Working voltage	5V-24V
Maximum current	15 mA(DC 5V)
Weight	43 g
Flow rate range	0.5~60 L/min
Operating temperature	0°C~80°
Operating humidity	35%~90%RH
Operating pressure	under 1.75Mpa
Store temperature	-25°C~+80°
Store humidity	25%~90%RH



Gambar 2.3 *Water Flow Sensor*

Pada sistem Prabayar diperlukan perhitungan untuk mengetahui berapa banyak debit air yang keluar sesuai dengan nilai input yang dimasukkan, maka dengan memanfaatkan output sensor yang berupa *half-effect* akan diketahui *flow rate* air yang mengalir. (Anonim, 2013)

Rumus Flow Rate :

$$Flow Rate = \frac{Pulse Frequency * 60}{7.5Q} \dots\dots\dots(2.3)$$

Keterangan :

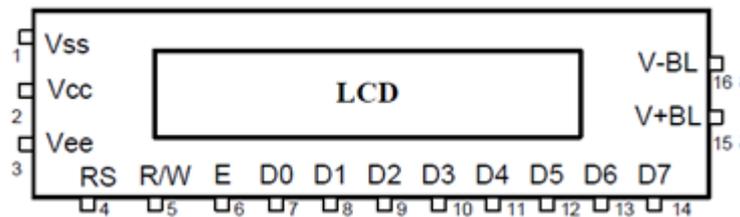
Flow rate = banyaknya air yang mengalir (liter/jam)

Pulse Frequency = Frekuensi pulsa *half-effect* dari sensor (putaran/detik)

7.5Q = Frekuensi pulsa dalam keadaan horizontal

2.5 Modul LCD 16x2

Modul LCD 16x2 merupakan suatu *display* yang digunakan untuk menampilkan suatu karakter yang diberikan oleh sistem, dalam hal ini sistem yang memberikan informasi adalah *microcontroller*. Konfigurasi pin dari LCD ditunjukkan pada Gambar 2.5.



Gambar 2.4 Konfigurasi pin LCD. (Vishay, 2002)

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°C sampai 55°C.

Tampilan karakter pada LCD diatur oleh Pin E, RS dan RW. Pin E digunakan untuk memberitahu LCD bahwa sedang mengirimkan sebuah data. Untuk mengirimkan data ke LCD, maka melalui program E harus dibuat logika *low* “0” dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, set E dengan logika “1” dan tunggu untuk sejumlah waktu

tertentu (sesuai dengan *datasheet* dari LCD tersebut) dan berikutnya set E ke logika *low* “0” lagi.

Ketika RS berlogika *low* “0”, data akan dianggap sebagai sebuah perintah atau instruksi khusus (seperti clear screen, posisi kursor dll). Ketika RS berlogika *high* “1”, data yang dikirim adalah data text yang akan ditampilkan pada *display* LCD. RW digunakan untuk menentukan mode baca dengan memberikan logika 1 atau mode tulis dengan memberikan logika 0 dari data yang terdapat pada pin DB0-DB7 (Vishay, 2002).

2.6 Solenoid Valve

Solenoid valve adalah katup yang digerakan oleh energi listrik, mempunyai kumparan sebagai penggeraknya yang berfungsi untuk menggerakkan piston yang dapat digerakan oleh arus AC maupun DC, *solenoid valve* atau katup (*valve*) solenoida mempunyai lubang keluaran, lubang masukan dan lubang exhaust, lubang masukan, berfungsi sebagai terminal / tempat cairan masuk atau *supply*, lalu lubang keluaran, berfungsi sebagai terminal atau tempat cairan keluar yang dihubungkan ke beban, sedangkan lubang *exhaust*, berfungsi sebagai saluran untuk mengeluarkan cairan yang terjebak saat piston bergerak atau pindah posisi ketika *solenoid valve* bekerja.

Prinsip kerja dari *solenoid valve*/katup (*valve*) solenoida yaitu katup listrik yang mempunyai koil sebagai penggeraknya dimana ketika koil mendapat *supply* tegangan maka koil tersebut akan berubah menjadi medan magnet sehingga menggerakkan piston pada bagian dalamnya ketika piston berpindah posisi maka pada lubang keluaran dari solenoid valve akan keluar cairan yang berasal dari

supply, pada umumnya *solenoid valve* mempunyai tegangan kerja 100/200 VAC namun ada juga yang mempunyai tegangan kerja DC. (Wardana, 2012)



Gambar 2.5 *Solenoid Valve*

2.7 Rangkaian *Driver* Relay

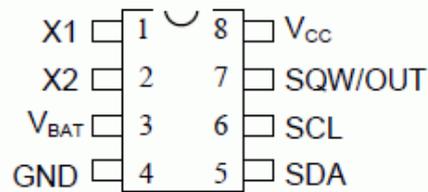
Relay adalah komponen listrik yang bekerja berdasarkan prinsip induksi medan elektromagnetis. Jika sebuah penghantar dialiri oleh arus listrik, maka di sekitar penghantar tersebut timbul medan magnet. Medan magnet yang dihasilkan oleh arus listrik tersebut selanjutnya diinduksikan ke logam ferromagnetis. Logam ferromagnetis adalah logam yang mudah terinduksi medan elektromagnetis. Ketika ada induksi magnet dari lilitan yang membelit logam, logam tersebut menjadi "magnet buatan" yang sifatnya sementara. Cara ini kerap digunakan untuk membuat magnet non permanen. Sifat kemagnetan pada logam ferromagnetis akan tetap ada selama pada kumparan yang melilitinya teraliri arus listrik. Sebaliknya, sifat kemagnetannya akan hilang jika suplai arus listrik ke lilitan diputuskan (Sant, 2013).

Rangkaian digital bekerja pada tegangan +5 volt DC sedangkan tegangan kerja relay DC antara 6V sampai 12V. Penggunaan relay sering menjadi pilihan karena relay mudah dikontrol, relay dapat diberi beban yang besar baik beban AC

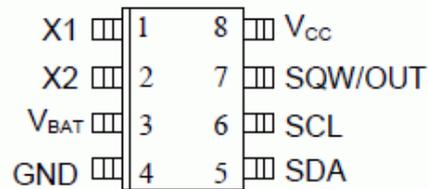
maupun DC, dan sebagai isolator yang baik antara rangkaian beban dengan rangkaian kendali. Rangkaian *interface* relay dapat dibangun menggunakan konsep transistor sebagai saklar. Transistor yang digunakan untuk *driver* relay dapat dikonfigurasi dengan *common emitor*, *emitor follower* atau transistor darlington.

2.8 RTC DS1307

Real Time Clock merupakan suatu chip (IC) yang memiliki fungsi sebagai penyimpan waktu dan tanggal. RTC DS1307 merupakan *Real-time clock* (RTC) yang dapat menyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100. 56-byte, *battery-backed*, RAM *nonvolatile* (NV) RAM untuk penyimpanan. RTC DS1307 merupakan *Real-time clock* (RTC) dengan jalur data *parallel* yang memiliki Antarmuka serial *Two-wire* (I2C), Sinyal luaran gelombang-kotak terprogram (*Programmable squarewave*), Deteksi otomatis kegagalan-daya (*power-fail*) dan rangkaian *switch*, Konsumsi daya kurang dari 500nA menggunakan mode baterai cadangan dengan operasional osilator. Tersedia fitur industri dengan ketahanan suhu: -40°C hingga +85°C. Tersedia dalam kemasan 8-pin DIP atau SOIC. (Agfi, 2009)



DS1307 8-Pin DIP (300-mil)



DS1307 8-Pin SOIC (150-mil)

Gambar 2.6 Layout IC DS1307 (Agfi, 2009)

Berikut Penjelasan Pin-Pin Pada IC RTC DS1307

1. X1 merupakan pin yang digunakan untuk dihubungkan dengan kristal sebagai pembangkit *clock*.
2. X2 berfungsi sebagai keluaran / *output* dari *crystal* yang digunakan. Trhubung juga dengan X1.
3. VBAT merupakan *backup supply* untuk RTC DS1307 dalam menjalankan fungsi waktu dan tanggal. Besarnya adalah 3V dengan menggunakan jenis *Lithium Cell* atau sumber energi lain. Jika pin ini tidak di gunakan maka harus terhubung dengan *ground*. Sumber tegangan dengan 48mAH atau lebih besar dapat digunakan sebagai cadangan energi sampai lebih dar 10 tahun, namun dengan persyaratan untuk pengoprasian dalam suhu 25°C.
4. GND berfungsi sebagai *ground*.
5. SDA berfungsi sebagai masukan / keluaran (I/O) untuk I2C *serial interface*. Pin ini bersifat *open drain*, oleh sebab itu membutuhkan eksternal *pull up* resistor.
6. SCL berfungsi sebagai *clock* untuk *input* ke I2C dan digunakan untuk mensinkronisasi pergerakan data dalam serial interface. bersifat *open drain*, oleh sebab itu membutuhkan eksternal *pull up* resistor.

7. SWQ/OUT sebagai *square wave* / *Output Driver* . jika di aktifkan, maka akan menjadi 4 frekuensi gelombang kotak yaitu 1 Hz, 4kHz, 8kHz, 32kHz sifat dari pin ini sama dengan sifat pin SDA dan SCL sehingga membutuhkan eksternal *pull up* resistor. Dapat dioprasikan dengan VCC maupun dengan VBAT.
8. VCC merupakan sumber tegangan utama. Jika sumber tegangan terhubung dengan baik, maka pengaksesan data dan pembacaan data dapat dilakukan dengan baik. Namun jika backup supply terhubung juga dengan VCC, namun besar VCC di bawah VTP, maka pengaksesan data tidak dapat dilakukan.



BAB III

METODE PENELITIAN

3.1 Alat dan Bahan Penelitian

Terdapat beberapa alat dan bahan yang digunakan dalam penyelesaian Tugas Akhir ini, berikut adalah rinciannya.

3.1.1 Alat penelitian

Dalam perancangan disetiap rangkaian yang digunakan dalam penelitian ini dibutuhkan alat-alat tertentu untuk mencapai hasil yang diharapkan.

a. AVO Meter

Alat ini digunakan untuk melihat dan mengecek arus, voltase dan hambatan yang ada pada semua rangkaian yang digunakan dalam penelitian ini.

b. Solder

Alat ini digunakan untuk membuat rangkaian-rangkaian tambahan yang diperlukan dalam penyelesaian penelitian.

c. *Power Supply* 24V DC

Digunakan sebagai sumber tenaga pada sistem yang dibuat dalam penelitian ini.

d. *Board* Arduino Uno R3

Pada penelitian ini *board* Arduino Uno R3 digunakan sebagai pemroses semua data *input* untuk menjadi *output* yang akan ditampilkan di layar LCD.

e. IC LM7812

Sebuah IC yang berfungsi sebagai pengubah tegangan menjadi 12V. Dalam penelitian ini sumber tenaganya adalah dari *power supply* yang mempunyai

tegangan 24V, sedangkan *board* Arduino Uno R3 memiliki *range power input* sebesar 6V sampai 12V.

f. DS1307

Merupakan IC *Real-time clock* (RTC) yang dapat menyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100.

g. Relay

Rangkaian ini berfungsi untuk mengontrol kinerja dari aktuator yang digunakan dalam penelitian ini.

h. *Water Flow Sensor*

Merupakan sensor yang akan melakukan proses *counter* pada pulsa dalam penelitian ini.

i. *Solenoid valve*

Merupakan aktuator yang berfungsi sebagai keran otomatis.

3.1.2 Bahan Penelitian

Adapun bahan atau objek yang ada pada penelitian ini adalah aliran air. Karena tujuan dari penelitian ini adalah menciptakan sebuah sistem penghitung meter air secara otomatis.

3.2 Jalan Penelitian

Langkah-langkah yang dilakukan dalam melakukan penelitian ini terbagi menjadi beberapa bagian sebagai berikut:

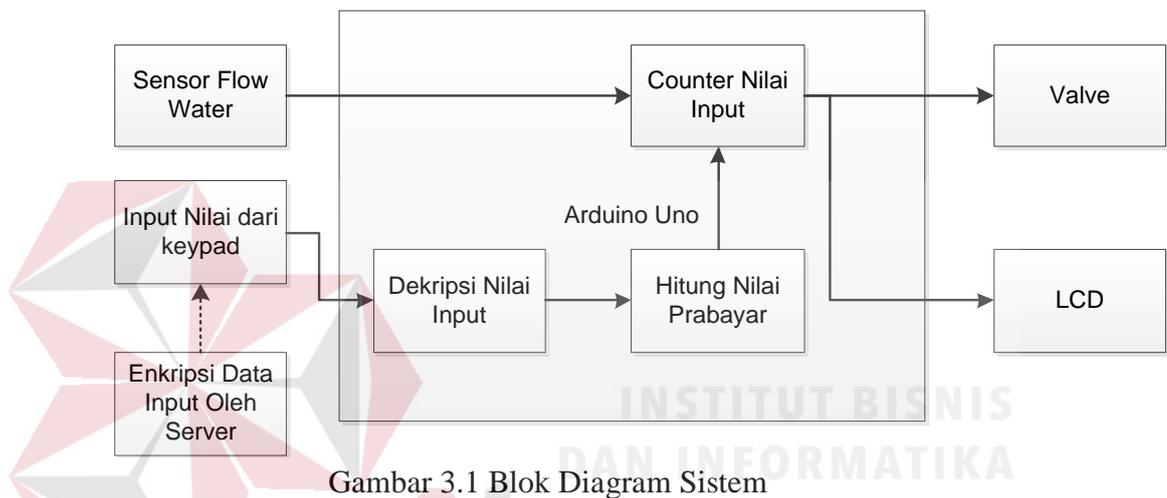
3.2.1 Pengumpulan Literatur

Mengumpulkan semua referensi yang berhubungan dengan semua alat dan bahan yang dipakai dalam penelitian ini.

3.2.2 Perancangan Perangkat Keras

Komponen-komponen yang digunakan dalam sistem ini meliputi *board* Arduino Uno R3, *power supply* 12V, rangkaian RTC dan rangkaian relay untuk *solenoid valve*.

Perancangan perangkat keras pada sistem secara keseluruhan dilakukan berdasarkan blok diagram yang terdapat pada Gambar 3.1.



Gambar 3.1 Blok Diagram Sistem

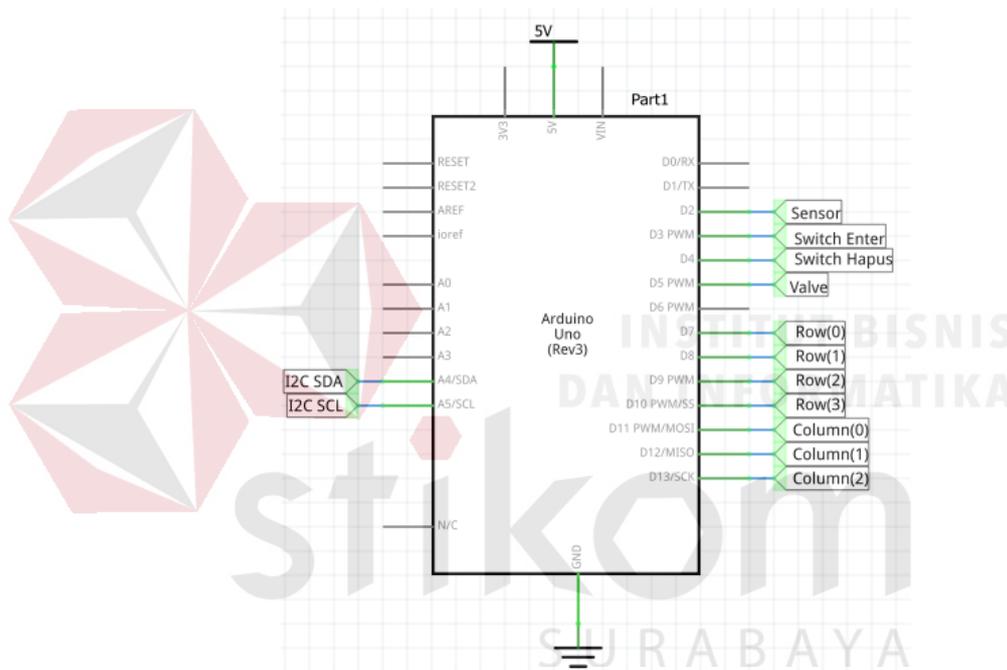
Pada Gambar 3.1 terdapat tiga bagian utama, yaitu *input*, proses, dan *output*.

1. Bagian *input* merupakan nilai pulsa yang didapatkan dari proses *generate* oleh server.
2. Bagian proses merupakan bagian yang ada didalam Arduino Uno R3 terdiri atas 3 bagian:
 - a. Proses dekripsi *input* pulsa untuk mendapatkan *value* dari pulsa tersebut.
 - b. Hitung nilai prabayar merupakan proses perhitungan *value* pulsa untuk mendapatkan berapa liter air dalam *value* tersebut.
 - c. Proses *counter* dilakukan setelah mendapatkan hasil dari perhitungan *value* hingga nilainya mencapai nol.

3. Bagian *output* terdiri dari LCD sebagai penampil nilai pulsa dan aktuator berupa *solenoid valve* untuk membuka dan menutup aliran air.

A. Rangkaian Minimum Sistem Arduino Uno R3

Arduino Uno R3 merupakan *board* berbasis *microcontroller* Atmega328. Pada penelitian ini *board* arduino digunakan untuk pemrosesan data-data dari input pulsa dan sensor flow water, membuka dan menutup *solenoid valve*, serta menampilkan data pada LCD.



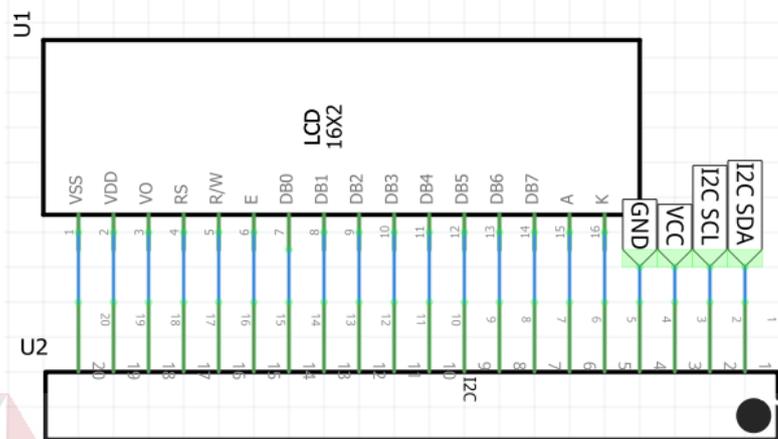
Gambar 3.2 Rangkaian Arduino Uno R3

Pada Gambar 3.2 terdapat beberapa sambungan pin, berikut penjelasannya:

1. Pin A4/SDA digunakan sebagai koneksi i2c SDA pada LCD dan RTC.
2. Pin A5/SCL digunakan sebagai koneksi i2c SCL pada LCD dan RTC.
3. Pin D2 digunakan sebagai inputan dari *water flow sensor*.
4. Pin D3 digunakan untuk mendeteksi penekanan tombol *enter*.
5. Pin D4 digunakan untuk mendeteksi penekanan tombol hapus.

6. Pin D5 digunakan untuk mengontrol *solenoid valve*.
7. Pin D7 sampai D13 digunakan sebagai inputan dari *keypad*.

B. Rangkaian LCD 16x2



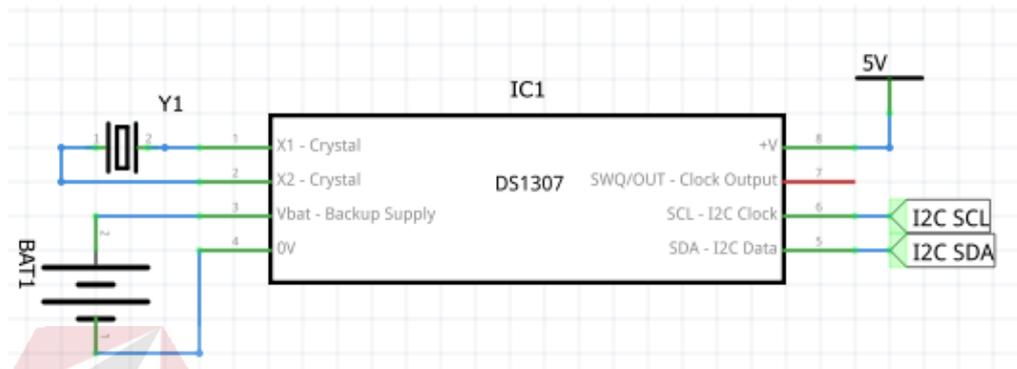
Gambar 3.3 Rangkaian LCD

Rangkaian LCD yang digunakan dalam penelitian ini menggunakan modul I2C, sehingga mampu menghemat penggunaan pin pada arduino. Potongan program untuk menampilkan data pada LCD adalah sebagai berikut:

```
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR    0x27
#define BACKLIGHT_PIN    3
#define En_pin    2
#define Rw_pin    1
#define Rs_pin    0
#define D4_pin    4
#define D5_pin    5
#define D6_pin    6
#define D7_pin    7
LiquidCrystal_I2C
  lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pi
n);
void setup()
{
  lcd.begin (16,2);
  lcd.setBacklightPin(BACKLIGHT_PIN, POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.home ();
  lcd.print("TES LCD OK");
}
```

C. Rangkaian RTC

Pada perancangan pulsa terdapat 6 digit angka yang merupakan deret tanggal, bulan dan tahun. Deret itu digunakan untuk membantu menambah sistem keamanan pada pulsa agar tidak terjadi input dua kali dengan pulsa yang sama.



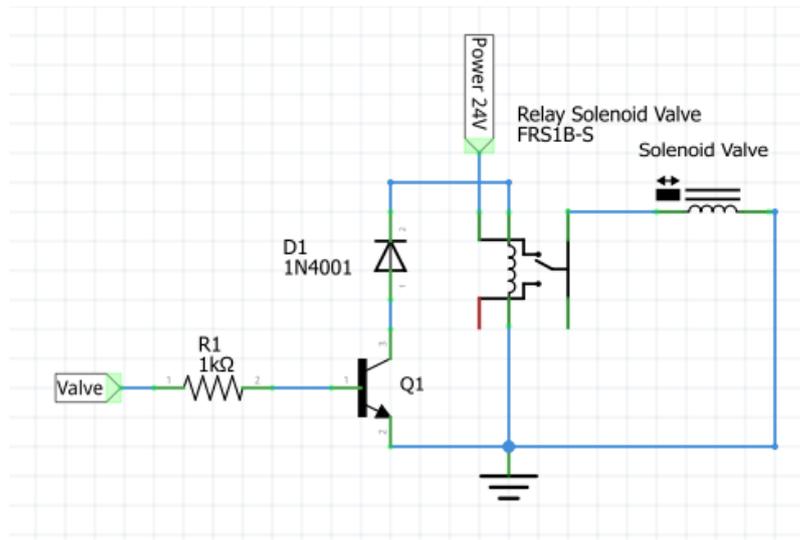
Gambar 3.4 Rangkaian RTC

Rangkaian ini digunakan sebagai perbandingan tanggal, bulan dan tahun yang ada pada pulsa dengan tanggal, bulan dan tahun secara real.

Pada gambar 3.4 terdapat beberapa sambungan pin, berikut penjelasannya:

1. X1 adalah pembangkit *clock* yang dihubungkan dengan *crystal*.
2. X2 berfungsi sebagai keluaran sinyal *clock*, yang juga dihubungkan dengan *crystal*.
3. Vbat merupakan sambungan ke baterai sebagai tenaga cadangan untuk rangkaian RTC.
4. GND berfungsi sebagai *ground*.
5. Vcc Sebagai sambungan untuk power RTC.
6. SDA berfungsi sebagai I/O untuk *serial I2C interface*.
7. SCL berfungsi sebagai *clock* untuk input ke I2C dan digunakan untuk mensinkronisasi pergerakan data dalam *serial interface*.
8. SQW/OUT Sebagai *square wave / Output Driver*.

D. Rangkaian Relay Solenoid Valve

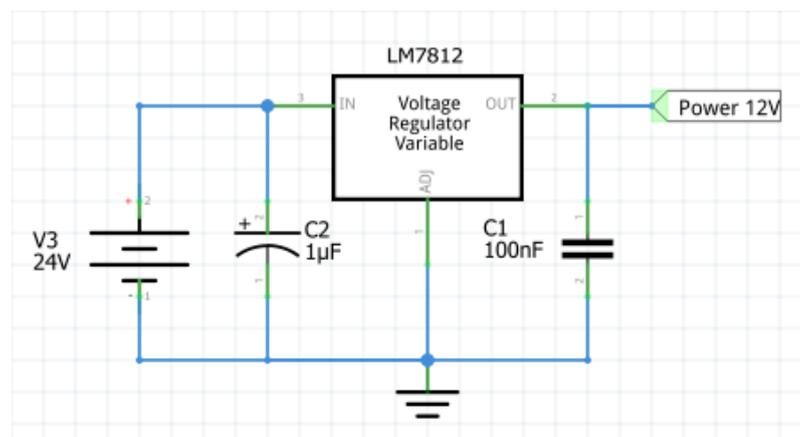


Gambar 3.5 Rangkaian Relay Solenoid Valve

Rangkaian ini digunakan untuk menyalakan *solenoid valve*, karena *output* dari pin arduino hanya sekitar 5V maka dibutuhkan sumber tegangan dari luar. Untuk mengoperasikan rangkaian ini dibutuhkan *relay* untuk menyalakan dan mematikan *solenoid valve*.

Solenoid valve mempunyai dua buah kaki positif dan negatif. Kaki positif *solenoid valve* disambungkan pada kaki *normally open* (NO) pada *relay*, sehingga pada defaultnya *solenoid valve* akan tertutup, dan akan terbuka saat *relay* aktif.

E. Rangkaian Power Supply 12 V



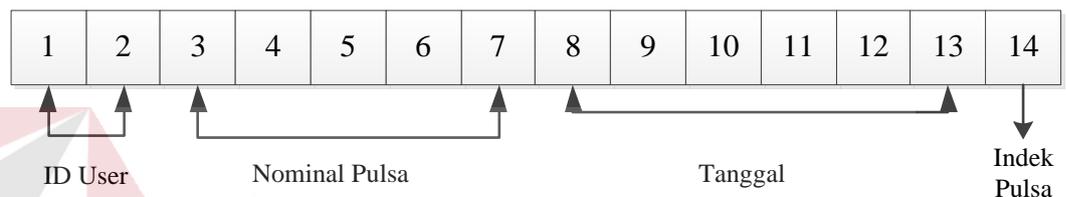
Gambar 3.6 Rangkaian *power supply* 12 V

Rangkaian *power supply* ini digunakan sebagai sumber tegangan pada *board* arduino. Rangkaian ini menggunakan IC *regulator* LM7812 yang mempunyai tegangan *output* 12 Volt.

3.2.3 Perancangan Perangkat Lunak

A. Perancangan Kode Pulsa

Pada sistem ini kode pulsa terdiri dari 14 digit angka, lebih jelasnya bisa dilihat pada gambar 3.7.



Gambar 3.7 Perancangan Kode Pulsa

Berikut adalah penjelasan mengenai tiap bagian pada perancangan kode pulsa yang digunakan dalam penelitian ini.

1. ID User

Merupakan kode unik yang digunakan untuk membedakan tiap user. Dibatasi 2 digit angka karena dalam sistem ini menggunakan kriptografi *caesar chipper* yang kurang mendukung dengan panjang data yang dinamis.

2. Nominal Pulsa

Merupakan *value* dari pembelian air di mana dalam sistem ini harga air per liter ditetapkan Rp 2000. Di bagian nominal pulsa ini dibatasi dengan 5 digit angka karena menggunakan metode *caesar chipper* yang kurang mendukung dengan panjang data yang dinamis. Dan nilai pembelian juga sudah di rubah menjadi putaran sensor. Dari hasil percobaan yang dilakukan menggunakan rumus 2.3 didapatkan banyak *record* dari *pulse*

frequency. Dan dalam perhitungan ini diambil satu sampel dari *record* tersebut, sebagai berikut:

Contoh: *Pulse Frequency* = 100 putaran/detik

$$\text{Flow Rate} = \frac{100 * 60}{7.5}$$

Flow Rate = 800 liter/jam

Dari hasil perhitungan tersebut kemudian dicari putaran per liter, seperti berikut:

$$\frac{100 \text{ putaran}}{\text{detik}} = \frac{800 \text{ liter}}{\text{jam}}$$

$$\frac{100 \text{ putaran}}{\text{detik}} = \frac{800 \text{ liter}}{3600 \text{ detik}}$$

$$800 \text{ liter} = 360000 \text{ putaran}$$

$$\text{liter} = 450 \text{ putaran}$$

Dari perhitungan diatas diperoleh dibutuhkan 450 putaran sensor untuk menghitung setiap liter air yang mengalir.

3. Tanggal

Pada bagian tanggal terdapat 6 digit angka yang terdiri dari 2 digit hari, 2 digit bulan dan 2 digit tahun. Tanggal digunakan untuk membantu sistem keamanan pada enkripsi agar tidak terjadi input 2 kali oleh pulsa yang sama.

4. Indek Pulsa

Merupakan 1 digit angka untuk menghitung pembelian pulsa pada hari yang sama, digunakan untuk mencegah input pulsa 2 kali pada hari yang sama.

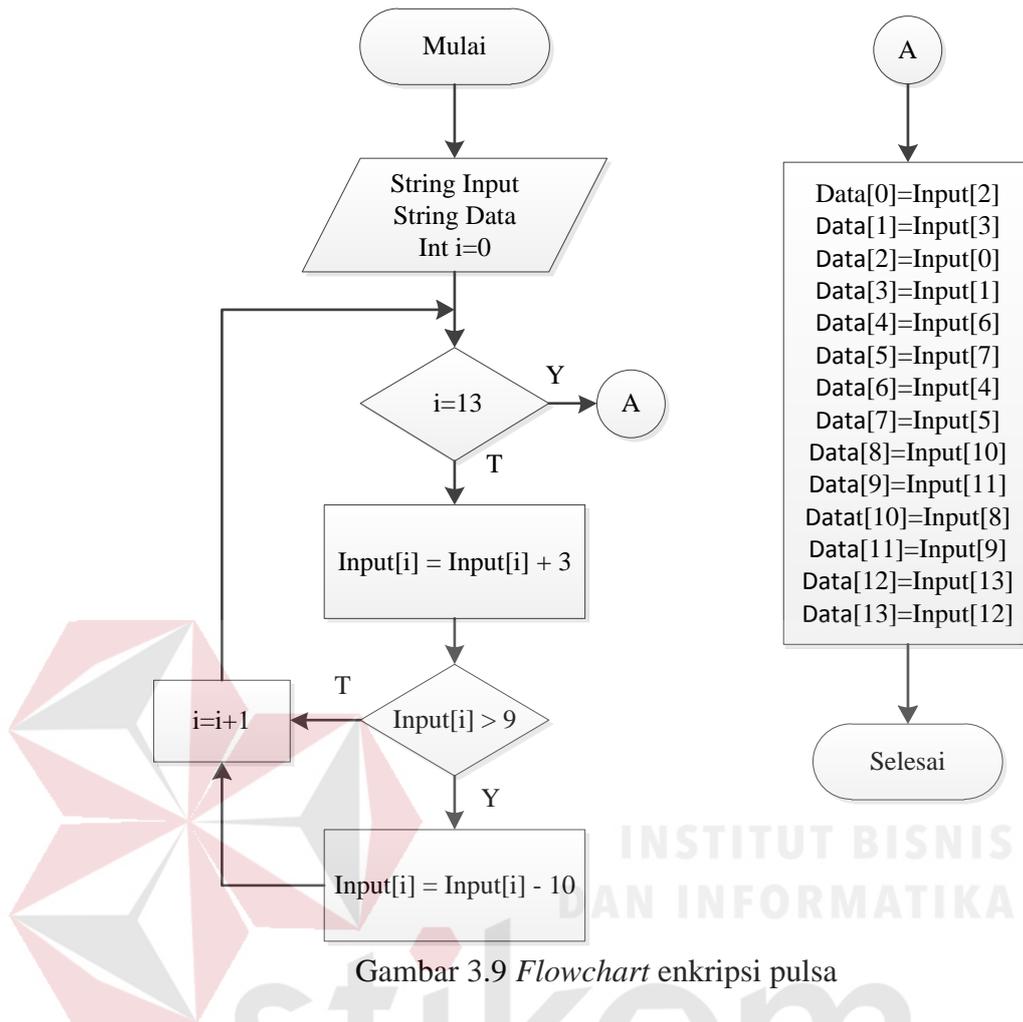
B. Perancangan Enkripsi Program Pada Server

Blok ini berisi tentang proses enkripsi nilai pulsa yang dilakukan oleh server dan nantinya akan diinputkan ke dalam alat.



Gambar 3.8 Tampilan Program Enkripsi Server

Tampilan awal program server bisa dilihat pada gambar 3.8. Metode dasar yang digunakan pada enkripsi pulsa adalah *caesar chiper*, yaitu dengan menjumlahkan tiap karakter dengan 3. Namun metode tersebut terlalu sederhana dibanding metode enkripsi lainnya sehingga ditambahkan dengan menukar posisi dari tiap karakter, seperti *flowchart* berikut:



Gambar 3.9 Flowchart enkripsi pulsa

Gambar 3.9 merupakan proses enkripsi data dengan metode *caesar chiper* yang dilakukan oleh server. Dari rumus 2.1 didapatkan perhitungan sebagai berikut:

$$E_n(x) = (x + n) \text{ mod } 26.$$

Karena pada sistem ini menggunakan angka sebagai data yang diolah dengan *range* 0 sampai 9, maka terdapat perubahan pada rumus sebagai berikut:

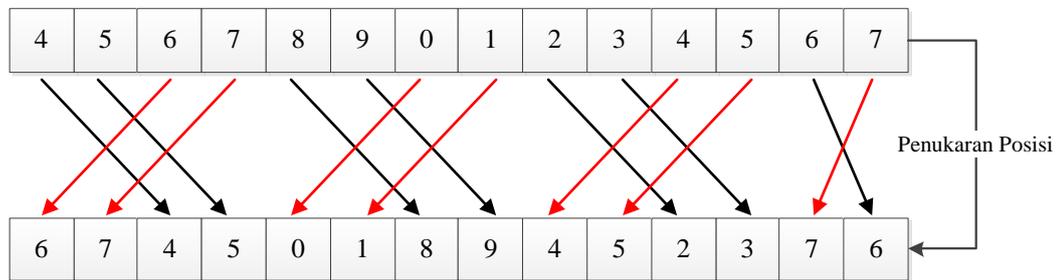
$$E_n(x) = (x + n) \text{ mod } 10$$

Di dalam sistem ini ditentukan menggunakan metode geser 3 ($n=3$). Dari rumus tersebut kemudian dimasukkan data yang akan diolah. Berikut contoh perhitungannya:

Input = 1 2 3 4 5 6 7 8 9 0 1 2 3 4

Caesar chiper = 4 5 6 7 8 9 0 1 2 3 4 5 6 7 (Setelah ditambah dengan 3)

Setelah proses penjumlahan kemudian dilakukan proses punukuran posisi tiap karakter dengan rule seperti berikut:



Gambar 3.10 Proses Penukaran

Gambar 3.10 menunjukkan alur penukaran tiap karakter pada pulsa.

Berikut contoh perhitungannya:

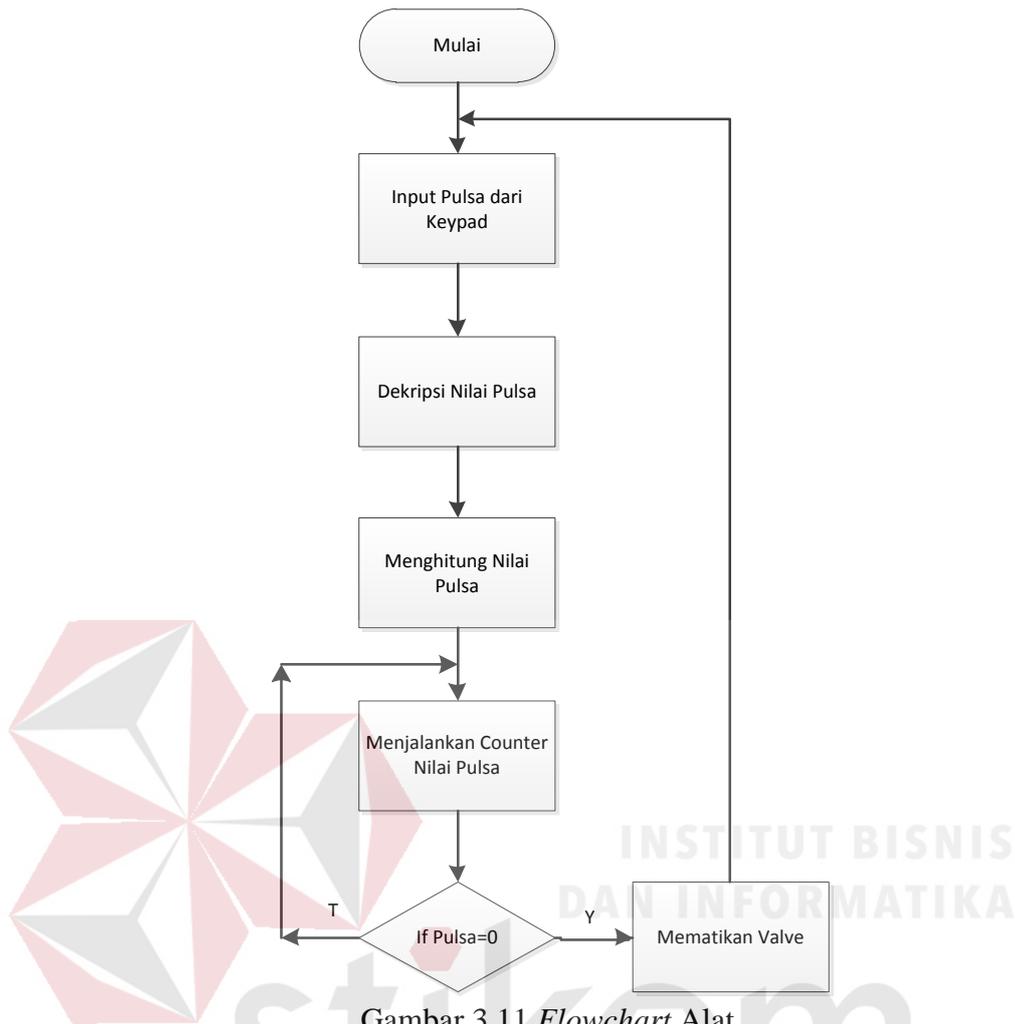
Caesar chiper = 4 5 6 7 8 9 0 1 2 3 4 5 6 7 (Setelah ditambah dengan 3)

Penukaran = 6 7 4 5 0 1 8 9 4 5 2 3 7 6 (Setelah ditukar posisi karakter)

Jadi hasil enkripsi yang nantinya akan diberikan pada pelanggan adalah hasil terakhir setelah penukaran posisi tiap karakter.

C. Perancangan Dekripsi Program Pada Alat

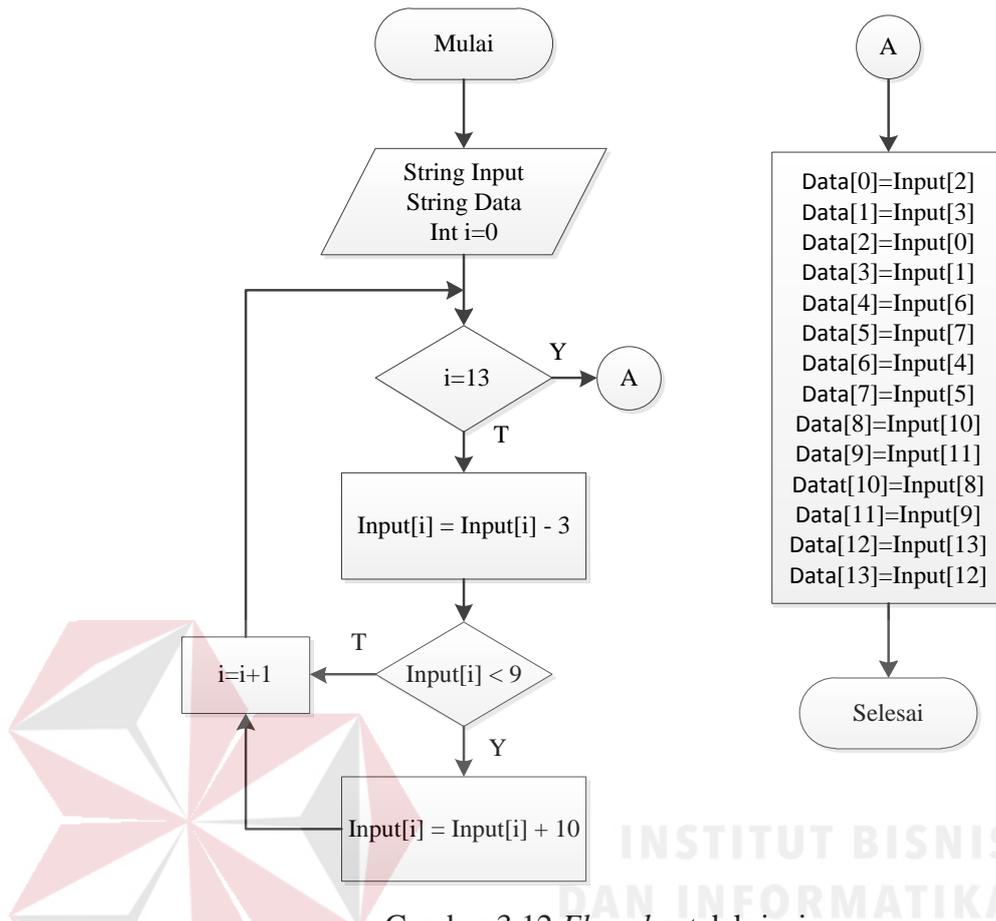
Program utama pada alat sistem ini bisa dilihat pada gambar flowchart berikut ini:



Gambar 3.11 *Flowchart* Alat

Gambar 3.11 menjelaskan bagaimana alur program yang terdapat pada alat. Proses pertama yang dilakukan adalah memasukkan pulsa melalui *keypad* kemudian alat akan melakukan proses dekripsi pulsa untuk mendapatkan nilai yang ada pada pulsa tersebut.

Proses pendekripsian dilakukan seperti pada proses enkripsi, hanya saja setelah penukaran posisi tiap karakter akan dikurangi 3. Jika hasil kurang ternyata minus maka akan ditambah dengan 10, seperti *flowchart* berikut:



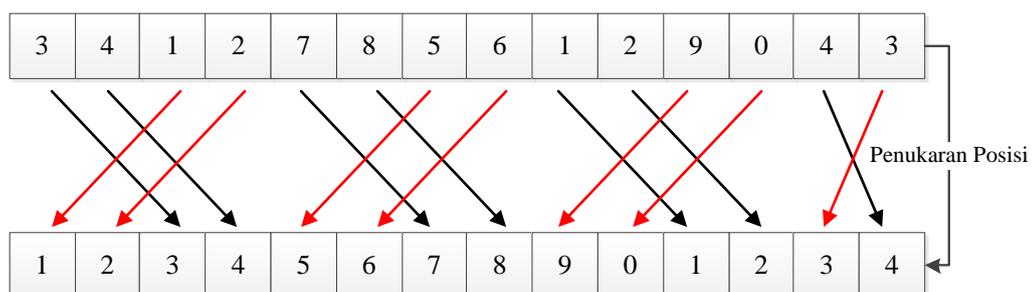
Gambar 3.12 Flowchart dekripsi

Gambar 3.12 adalah proses dekripsi yang dilakukan pada alat, berikut contoh perhitungannya:

Input = 6 7 4 5 0 1 8 9 4 5 2 3 7 6

Caesar chiper = 3 4 1 2 7 8 5 6 1 2 9 0 4 3 (Setelah dikurangi dengan 3)

Setelah proses penjumlahan kemudian dilakukan proses punukuran posisi tiap karakter dengan rule seperti berikut:



Gambar 3.13 Proses Penukaran Kembali Karakter

Pada Gambar 3.13 menunjukkan proses penukaran posisi tiap karakter pulsa saat dekripsi. Berikut contoh perhitungannya:

Caesar cipher = 3 4 1 2 7 8 5 6 1 2 9 0 4 3 (Setelah dikurangi dengan 3)

Penukaran = 1 2 3 4 5 6 7 8 9 0 1 2 3 4 (Setelah penukaran)

Jadi hasil dekripsi dari pulsa tersebut diperoleh data:

ID User = 12

Value = 34567

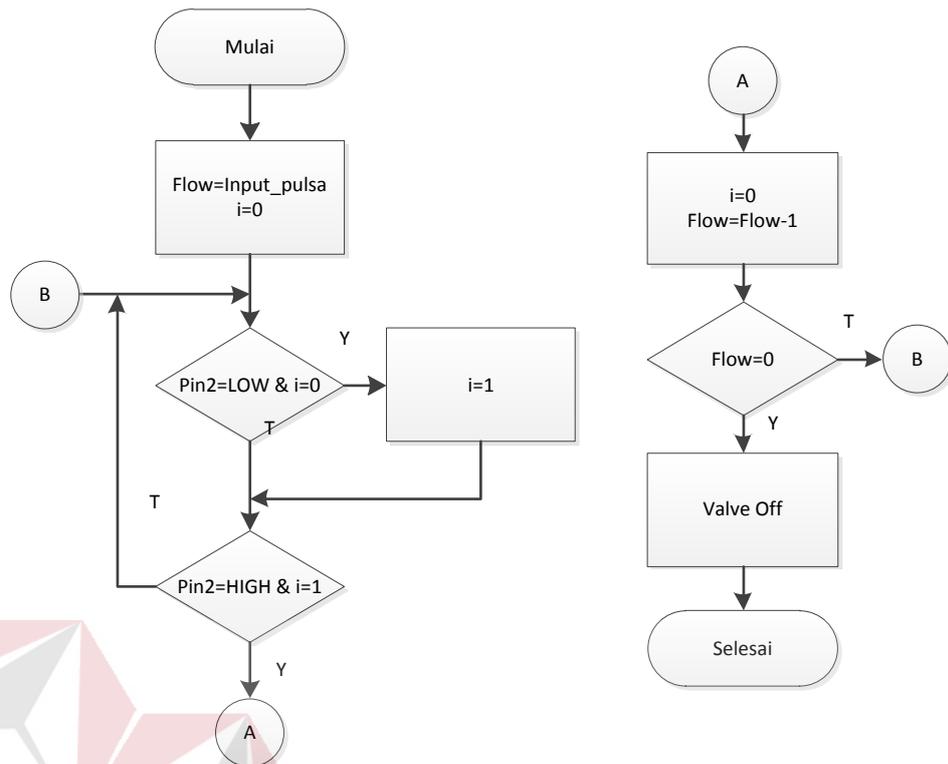
Tanggal = 890123

Indek Pulsa = 4

D. Perancangan Program Aktuator

Aktuator pada penelitian ini menggunakan *solenid valve* yang mempunyai *default normaly close*. Jadi aktuator ini mempunyai dua kondisi dimana aktuator tersebut akan tertutup didalam sistem ini, yang pertama adalah ketika tidak dialiri *power* dan yang kedua adalah ketika pulsa air habis.

Ketika pulsa dimasukkan kedalam sistem maka aktuator akan terbuka dan sensor *flow water* akan menjalan counter sesuai dengan nilai tersebut, seperti *flowchart* sebagai berikut:



Gambar 3.14 *Flowchart counter pulsa*

Gambar 3.14 menjelaskan tentang alur dari proses *counter* pulsa yang dilakukan oleh *water flow sensor* terhadap berapa banyaknya pulsa yang dimasukkan hingga pulsa mencapai nilai nol dan *solenoid valve* akan tertutup.

3.3 Langkah Pengujian

Pengujian ini adalah pengujian yang dilakukan terhadap semua rancangan yang telah dibuat. Dari semua perancangan perangkat keras dan perancangan perangkat lunak.

3.3.1 Pengujian Perangkat Keras

A. Pengujian Arduino Uno R3

Pengujian sistem Arduino Uno R3 ini dilakukan untuk untuk mengetahui keluaran yang dihasilkan apakah sesuai dengan program. Pengujian ini dilakukan dengan cara memprogram sistem Arduino Uno R3 untuk membuat pin 4 menjadi

HIGH dan diukur secara berulang-ulang. Berikut adalah potongan program pada pengujian Arduino Uno R3:

```
constint PIN_OUTPUT=4;
void setup()
{
pinMode (PIN_OUTPUT,OUTPUT);
}
void loop()
{
digitalWrite(PIN_OUTPUT,HIGH);
}
```

Pengujian berhasil jika keluaran pin 4 bernilai *LOW* dan *HIGH* pada waktu yang telah ditentukan.

Tabel 3.1 Pengujian Arduino Uno R3

Tujuan	Alat dan Bahan	Input	Output yang diharapkan	Indikator Keberhasilan
Untuk mengetahui keluaran yang dihasilkan apakah sesuai dengan program.	<ul style="list-style-type: none"> • <i>Board</i> Arduino Uno R3 • Avometer • <i>Power Supply</i> 	5V	5V	Berhasil jika <i>output</i> sesuai dengan <i>Input</i>
		5V	5V	
		5V	5V	
		5V	5V	

B. Pengujian LCD 16x2

Pengujian ini dilakukan untuk mengetahui hasil tampilan dari LCD 16x2. Pengujian dilakukan dengan menggunakan Arduino Uno R3 sebagai alat untuk memerintahkan LCD menampilkan beberapa karakter. Pada pengujian LCD ini Arduino Uno R3 diberi program untuk menampilkan karakter pada tiap baris. Berikut adalah potongan program pada pengujian LCD:

```
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>

#define I2C_ADDR    0x27
#define BACKLIGHT_PIN    3
#define En_pin    2
#define Rw_pin    1
#define Rs_pin    0
#define D4_pin    4
#define D5_pin    5
#define D6_pin    6
```

```

#define D7_pin 7

LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

void setup()
{
  Serial.begin(9600);
  lcd.begin(16,2);
  lcd.setBacklightPin(BACKLIGHT_PIN, POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.home();
}
void loop()
{
  lcd.clear();
  lcd.print("Test LCD OK..");
  delay(100);
}

```

Pengujian berhasil jika LCD dapat menampilkan karakter sesuai yang ada di dalam program.

Tabel 3.2 Pengujian LCD 16x2

Tujuan	Alat dan Bahan	Input	Output yang diharapkan	Indikator Keberhasilan
Untuk mengetahui hasil tampilan dari LCD 16x2.	<ul style="list-style-type: none"> • <i>Board</i> Arduino Uno R3 • LCD 16x2 • <i>Power Supply</i> 	“Test LCD OK”	“Test LCD OK”	Berhasil jika <i>output</i> sesuai dengan <i>Input</i>

C. Pengujian RTC

Pengujian ini dilakukan untuk mengetahui tingkat keakuratan waktu yang ada di dalam rangkaian RTC. Pengujian dilakukan dengan cara memberi program Arduino Uno R3 untuk menampilkan jam, menit, detik dan juga hari, bulan dan tahun ke dalam serial monitor. Berikut adalah potongan program pada pengujian RTC:

```

#include <Wire.h>
#include <RTClib.h>

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

```

```

while (!Serial);
Wire.begin();
RTC.begin();
if (!RTC.isrunning())
{
  RTC.adjust(DateTime(__DATE__, __TIME__));
}
}
void loop()
{
  DateTime now = RTC.now();
  Serial.print(now.day(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.year(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
  delay(2000);
}

```

Pengujian akan dianggap berhasil jika waktu yang ditampilkan sesuai dengan waktu pada kenyataan.

Tabel 3.3 Pengujian RTC

Tujuan	Alat dan Bahan	Input	Output yang diharapkan	Indikator Keberhasilan
Untuk mengetahui tingkat keakuratan waktu pada RTC	<ul style="list-style-type: none"> • Board Arduino Uno R3 • RTC • Power Supply 	Program tampilan waktu	Tampilan waktu	Berhasil jika <i>output</i> sesuai dengan <i>Input</i>

D. Pengujian Water Flow Sensor

Pengujian ini dilakukan untuk mengetahui kinerja sensor ketika dialiri air. Pengujian ini dilakukan dengan menggunakan *hyper terminal* pada arduino untuk *monitoring* perubahan putaran sensor ketika dialiri air. Berikut adalah potongan program pada pengujian *water flow sensor*:

```

#include<Wire.h>
#include<LCD.h>

```

```

#include<LiquidCrystal_I2C.h>
#include<OneWire.h>
#include<LiquidCrystal.h>
#define I2C_ADDR    0x27
#define BACKLIGHT_PIN    3
#define En_pin    2
#define Rw_pin    1
#define Rs_pin    0
#define D4_pin    4
#define D5_pin    5
#define D6_pin    6
#define D7_pin    7
int Pin_Suhu = 8;
LiquidCrystal_I2C
  lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

void setup()
{
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.setBacklightPin(BACKLIGHT_PIN, POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.home ();
}
void loop()
{
  if(digitalRead(sensor)==LOW && cek==false)
  {
    cek=true;
  }
  if(digitalRead(sensor)==HIGH && cek==true)
  {
    putaran=putaran+1;
    lcd.print("putaran= ");
    lcd.print(sisa);
    cek=false;
  }
}

```

Pengujian ini berhasil jika sensor dapat berputar ketika dialiri air.

Tabel 3.4 Pengujian Water Flow Sensor

Tujuan	Alat dan Bahan	Input (air)	Output yang diharapkan	Indikator Keberhasilan
Untuk mengetahui output dari water flow sensor	<ul style="list-style-type: none"> Water Flow Sensor Board Arduino Uno R3 Laptop 	Aliran air	Putaran Sensor	Berhasil jika sensor dapat berputar ketika dialiri air.

E. Pengujian Relay *Solenoid Valve*

Pengujian ini dilakukan untuk mengetahui kinerja *solenoid valve* ketika disambungkan dengan tegangan 24V. Pengujian ini dilakukan dengan memberikan catu daya dari power *supply* sebesar 24 volt DC untuk mengaktifkan *solenoid valve* melalui relay. Pengujian berhasil apabila ada perubahan ketika disambungkan dengan tegangan 24V.

Tabel 3.5 Pengujian Relay *Solenoid Valve*

Tujuan	Alat dan Bahan	Input	Output yang diharapkan	Indikator Keberhasilan
Untuk mengetahui kinerja dari <i>solenoid valve</i>	<ul style="list-style-type: none">• <i>Solenoid Valve</i>• Rangkaian Relay• <i>Power Supply</i> 24V	Sambungkan Relay dengan power supply	<i>Solenoid Valve</i> terbuka	Berhasil jika <i>Solenoid Valve</i> terbuka ketika dialiri tegangan 24V

F. Pengujian Rangkaian *Power Supply* 12V

Pengujian ini dilakukan untuk mengetahui kebenaran tentang *output* dari rangkaian *power supply* 12V. Pengujian dilakukan dengan cara menyambungkan tegangan sebesar 24V ke dalam rangkaian *power supply* 12V. Kemudian dilihat hasilnya apakah sudah sesuai dengan yang diharapkan. Pengujian berhasil jika *output* dari rangkaian tersebut adalah 12V.

Tabel 3.6 Pengujian *Power Supply* 12V

Tujuan	Alat dan Bahan	Input	Output yang diharapkan	Indikator Keberhasilan
Untuk mengetahui <i>output</i> dari rangkaian <i>power supply</i> 12V	<ul style="list-style-type: none">• Avometer• Rangkaian <i>power supply</i> 12V• <i>Power Supply</i> 24V	24V	12V	Berhasil jika <i>output</i> sesuai dengan rancangan
		24V	12V	

3.3.2 Pengujian Perangkat Lunak

A. Pengujian Program Enkripsi Pada Server

Pengujian server dilakukan untuk mengetahui kebenaran tentang sistem enkripsi pulsa. Pengujian dilakukan dengan cara menjalankan program yang telah dibuat dengan *visual basic 6.0* dan menjalankan simulasi pembelian pulsa. Pengujian server ini dilakukan untuk memastikan bahwa pulsa keluaran dari program berisi sesuai dengan nominal yang di inputkan.

Tabel 3.7 Pengujian Program Enkripsi Pada Server

Tujuan	Alat dan Bahan	Input	Output yang diharapkan	Indikator Keberhasilan
Untuk mengetahui sistem enkripsi pulsa	<ul style="list-style-type: none">• Laptop• Program Visual Basic 6.0	Data pulsa sebelum di enkripsi	Data pulsa setelah di enkripsi	Berhasil jika perhitungan program sesuai dengan perhitungan manual

B. Pengujian Program Dekripsi Pada Alat

Pengujian algoritma dekripsi ini dilakukan untuk mengetahui kebenaran dari sistem dekripsi pulsa pada alat. Pengujian dilakukan dengan cara *step by step*. Yaitu dengan cara menguji tiap langkah yang ada, dari proses *caesar chiper*(dikurangi dengan 3) dan juga proses penukaran posisi tiap karakter. Pengujian berhasil jika *output* sesuai dengan perhitungan manual.

Tabel 3.8 Pengujian Program Dekripsi Pada Alat

Tujuan	Alat dan Bahan	Input	Output yang diharapkan	Indikator Keberhasilan
Untuk mengetahui sistem dekripsi pulsa	<ul style="list-style-type: none">• Board Arduino Uno R3• Laptop• LCD 16x2	Pulsa	Data pulsa setelah di dekripsi	Berhasil jika perhitungan program sesuai dengan perhitungan manual

C. Pengujian Sistem Secara Keseluruhan

Pengujian sistem dilakukan untuk mengetahui tingkat keakuratan dari sistem prabayar yang dirancang. Pengujian dilakukan dengan cara memasukkan jumlah pulsa tertentu dan mengalirkan air ke pipa yang telah terpasang. Air tidak akan bisa mengalir kecuali terdapat isi pulsa di dalamnya. Setelah dimasukkan nilai pulsa maka aliran air akan menggerakkan rotor sensor sehingga counter berjalan sampai nilai pulsa tersebut habis dan mati. Pengujian berhasil jika output air sesuai dengan data yang ada di dalam pulsa.

Tabel 3.9 Pengujian Sistem Keseluruhan

Tujuan	Alat dan Bahan	<i>Input</i> (pulsa)	<i>Output</i> yang diharapkan	Indikator Keberhasilan
Untuk mengetahui kinerja sistem prabayar	<ul style="list-style-type: none">• Laptop• Rangkaian Sistem Keseluruhan• Air	2000	1 liter	Berhasil jika <i>output</i> sesuai dengan <i>value</i> pulsa
		2000	1 liter	
		4000	2 liter	
		4000	2 liter	
		5000	2,5 liter	

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian *Board* Arduino Uno R3

Pengujian *board* Arduino Uno R3 ini dilakukan dengan cara memasukkan program pada arduino, kemudian menguji keluarannya pada pin digital Arduino Uno R3.

4.1.1 Hasil Pengujian *Board* Arduino Uno R3

Hasil percobaan *board* Arduino Uno R3 dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil Pengujian Keluaran Pin Digital Arduino (Pin 4)

Input (V)	Pengukuran Output Pin Digital (V)	Nilai yang Diharapkan (V)	Error(%)
5	4,99	5	0,2
5	5,02	5	0,39
5	4,99	5	0,2
5	4,98	5	0,4
5	5,02	5	0,39
Jumlah			1,58 %
Rata-rata <i>error</i> (%)			0,31 %

4.1.2 Pembahasan

Berdasarkan Tabel 4.1 dapat disimpulkan bahwa *board* Arduino Uno R3 dapat bekerja dengan baik dan bisa digunakan untuk menyelesaikan penelitian ini, dengan rata-rata tingkat *error* 0,31 %. *Error* ini disebabkan oleh tidak stabilnya input yang digunakan pada proses pengujian.

4.2 Pengujian LCD

Pengujian LCD menggunakan arduino uno r3 sebagai alat untuk memerintahkan LCD menampilkan beberapa karakter. Pada pengujian LCD ini arduino uno r3 diberi program sederhana tes LCD.

4.2.1 Hasil Pengujian LCD

Hasil pengujian LCD dapat dilihat pada gambar-gambar berikut ini.



Gambar 4.1 Hasil Pengujian LCD 1

Gambar 4.1 menunjukkan hasil tampilan LCD pada pengujian LCD yang pertama.



Gambar 4.2 Hasil Pengujian LCD 2

Gambar 4.2 menunjukkan hasil tampilan LCD pada pengujian LCD yang kedua.



Gambar 4.3 Hasil Pengujian LCD 3

Gambar 4.3 menunjukkan hasil tampilan LCD pada pengujian LCD yang ketiga.

4.2.2 Pembahasan

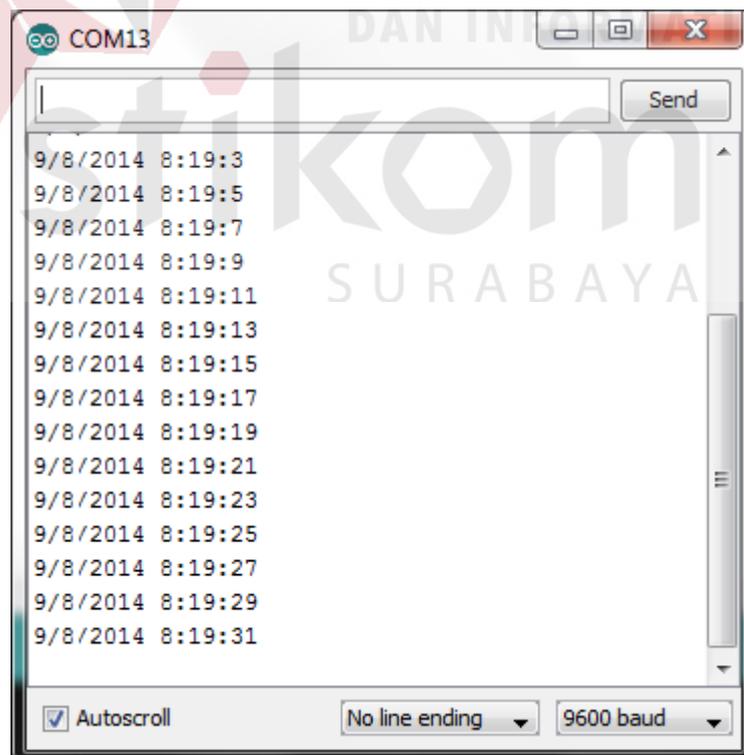
Hasil pengujian LCD menunjukkan bahwa tampilan LCD sudah sesuai dengan program yang dimasukkan dan LCD layak untuk digunakan.

4.3 Pengujian RTC

Pengujian RTC dilakukan dengan menggunakan cara memberi program Arduino Uno R3 untuk menampilkan jam, menit, detik dan juga hari, bulan dan tahun ke dalam serial monitor.

4.3.1 Hasil Pengujian RTC

Hasil pengujian RTC dapat dilihat pada gambar tampilan serial monitor berikut ini.



Gambar 4.4 Tampilan pengujian RTC

Gambar 4.4 menunjukkan hasil tampilan waktu pada *serial monitor* Arduino Uno R3 saat pengujian RTC.

4.3.2 Pembahasan

Dari hasil yang terlihat pada gambar 4.4 dapat diketahui rangkaian RTC berjalan dengan baik dan dapat menampilkan waktu sesuai dengan indikator keberhasilan yang tertera pada Tabel 3.3.

4.4 Pengujian *Water Flow Sensor*

Pengujian sensor menggunakan *hyper terminal* pada arduino untuk monitoring perubahan putaran sensor ketika dialiri air sesuai dengan ketentuan yang ada di rumus. Pengujian ini berhasil jika data yang keluar sesuai dengan hasil perhitungan rumus.

4.4.1 Hasil Pengujian *Water Flow Sensor*

Pengujian *water flow sensor* diperlukan untuk pengkalibrasian sensor, selain itu pengujian ini sangat bermanfaat agar data yang diolah lebih stabil, sehingga dapat mengambil keputusan yang tepat. Air yang digunakan sebagai sampel adalah 1 dan 2 liter. Tujuan utama dari pengujian sensor ini adalah untuk melihat tingkat akurasi *water flow sensor*.



Gambar 4.5 *Water flow sensor*

Gambar 4.5 adalah tampilan sensor saat dilakukan pengujian. Untuk perhitungan berpacu pada 1 liter air sama dengan 450 putaran *half-effect*. Dan dari pengujian diatas dihasilkan data seperti pada tabel berikut:

Tabel 4.2 Hasil pengujian sensor

No	Input (liter)	Half Effect Sensor (Putaran)	Half Effect Seharusnya (Putaran)	Error	Persen Error (%)
1	1	448	450	2	0,44
2	1	449	450	1	0,22
3	1	448	450	2	0,44
4	1	451	450	1	0,22
5	1	453	450	3	0,66
6	1	450	450	0	0
7	1	452	450	2	0,44
8	1	449	450	1	0,22
9	1	451	450	1	0,22
10	1	454	450	4	0,88
11	2	901	900	1	0,11
12	2	901	900	1	0,11
13	2	903	900	3	0,33
14	2	898	900	2	0,22
15	2	902	900	2	0,22
16	2	903	900	3	0,33
17	2	901	900	1	0,11
18	2	899	900	1	0,11
19	2	902	900	2	0,22
20	2	902	900	2	0,22
Rata-rata Error				1,75	0,29

4.4.2 Pembahasan

Berdasarkan tabel 4.2 didapatkan bahwa rata-rata kesalahan kalibrasi sensor adalah sebesar 0.29%. *Error* terjadi karena sifat dari rotor yang ada di dalam sensor. Ketika tekanan aliran air sangat rendah rotor sensor tidak berputar karena aliran air tidak cukup kuat untuk memutar rotor. Sedangkan ketika tekanan aliran air tinggi dan saat aliran itu dimatikan terdapat sisa-sisa tenaga yang memutar rotor tersebut. Dari data tersebut dapat disimpulkan presentase errornya cenderung mengecil jika inputnya besar karena tingkatan errornya tetap.

4.5 Pengujian Solenoid Valve

Pengujian blower dilakukan dengan memberikan catu daya dari power *supply* sebesar 24 volt DC.

4.5.1 Hasil Pengujian *Solenoid Valve*

Solenoid valve yang digunakan pada penelitian ini katupnya terbuka jika diberi tegangan 24 volt dan tertutup ketika tidak di aliri tegangan 24 volt. Hal ini menandakan valve dapat berfungsi dengan baik pada tegangan 24 volt sesuai dengan spesifikasi dari *solenoid valve*.



Gambar 4.6 Solenoid Valve

Berikut adalah tabel dari hasil pengujian *solenoid valve*.

Tabel 4.3 Hasil Pengujian *Solenoid Valve*

Input (V)	Katup <i>Solenoid Valve</i>	<i>Output</i> yang diharapkan
24	Terbuka	Terbuka
0	Tertutup	Tertutup
12	Tertutup	Tertutup
24	Terbuka	Terbuka
5	Tertutup	Tertutup

4.5.2 Pembahasan

Dari hasil yang terdapat pada Tabel 4.3, diperoleh data katup *solenoid valve* hanya terbuka jika disambungkan dengan sumber tegangan 24V. Dari hasil tersebut dapat disimpulkan bahwa *solenoid valve* bisa berfungsi dengan baik sesuai dengan indikator keberhasilan yang tercantum pada Tabel 3.5.

4.6 Pengujian Server

Pengujian server dilakukan dengan cara menjalankan program yang telah dibuat dengan *visual basic 6.0* dan menjalankan simulasi pembelian pulsa.

4.6.1 Hasil Pengujian Server

Pengujian ini dilakukan untuk mengetahui pulsa hasil enkripsi oleh program server apakah sesuai dengan nominal harga pembelian oleh pelanggan.

Berikut tampilan program awal:

The screenshot shows a Windows-style application window titled "Form1". It features a "ENKRIPSI" section with four input fields: "ID User", "Harga", "Indek Pulsa" (a dropdown menu currently set to "1"), and "Tanggal" (a date picker showing "11/07/2014"). A "CLEAR" button is positioned to the right of the date field. Below the input fields is an "ENKRIP" button and an empty text box for the result. The "DEKRIPSI" section consists of two empty text boxes and a "DEKRIP" button at the bottom. A large, semi-transparent watermark "Srikom SURABAYA" is visible across the center of the window.

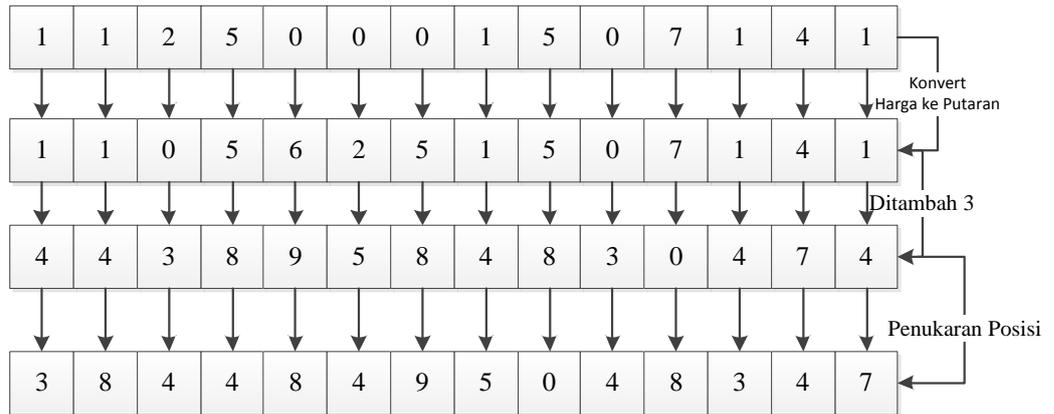
Gambar 4.7 Tampilan program server

Pengujian dilakukan dengan memasukkan ID user dan juga nominal pembelian oleh user. Sistem enkripsi ini terdapat 14 karakter. Dengan 2 karakter untuk User ID, 5 karakter untuk pulsa, 6 karakter untuk tanggal, dan 1 karakter untuk indek pulsa. penambahan tanggal bertujuan untuk menambah sistem keamanan enkripsi agar satu pulsa tidak bisa dimasukkan berkali-kali. Sedangkan indek pulsa digunakan untuk memberi indek pada pulsa untuk pembelian dalam satu hari. Enkripsi dilakukan seperti gambar berikut:

Field	Value
ID User	11
Harga	25000
Indek Pulsa	1
Tanggal	15/07/2014
ENKRIP Result	38448495048347
Tampilan Awal Sebelum Enkripsi	11 25000 1507141
Harga dikonvert menjadi putaran	11056251507141

Gambar 4.8 Hasil Enkripsi

Pada gambar 4.8 terdapat 3 lingkaran merah, itu adalah hasil dari pengujian tiap langkah yang dilakukan. Secara rinci akan dijelaskan pada perhitungan berikut:



Gambar 4.9 Alur Proses Enkripsi

Pada Gambar 4.9 menunjukkan alur dari proses enkripsi yang dilakukan oleh server. Langkah yang pertama adalah memasukkan data-data yang diperlukan seperti ID *User*, Nominal pulsa, tanggal dan juga index pulsa. Setelah dimasukkan kemudian nominal pulsa akan dirubah menjadi putaran sensor, dengan perhitungan sebagai berikut:

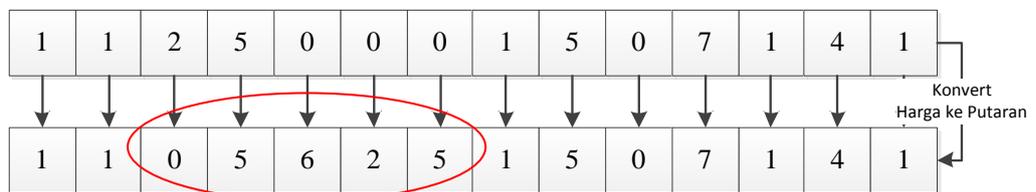
Pembelian= Rp 25.000,-

Harga perliter= Rp 2000,-

Putaran sensor perliter= 450

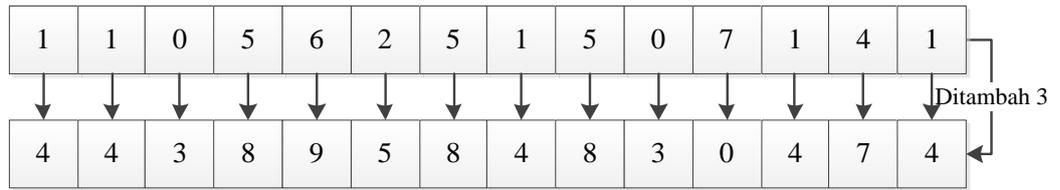
Jadi putaran yang didapat dari pembelian Rp 25.000 adalah:

$$\text{Putaran} = (25000/2000) \times 450 = 5625$$



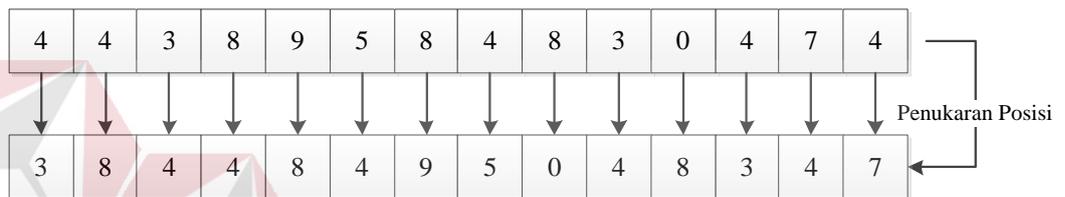
Gambar 4.10 Perubahan *Value* Pulsa Menjadi Putaran Sensor

Pada Gambar 4.10 dijelaskan proses perubahan *value* pulsa menjadi putaran sensor. Setelah didapatkan nilai putaran langkah selanjutnya adalah proses enkripsi menggunakan metode *caesar chiper* yaitu dengan menjumlahkan 3 pada tiap digit angka. Berikut penjelasannya:



Gambar 4.11 Proses Caesar Chipper

Pada Gambar 4.11 ditunjukkan tentang proses penjumlahan 3 dengan menggunakan metode *caesar chipper*. Dari hasil penjumlahan tersebut, kemudian dilakukan proses penukaran tiap karakter untuk meningkatkan keamanan enkripsi data. Berikut penjelasannya:



Gambar 4.12 Proses Penukaran posisi Karakter

Gambar 4.12 menunjukkan proses penukaran posisi tiap karakter pada perancangan pulsa. Hasil diatas adalah hasil akhir dari proses enkripsi data. Dari inputan yang berupa:

ID User = 11

Nominal Pulsa = 25000

Tanggal = 15/07/2014

Indek Pulsa = 1

Diperoleh hasil enkripsi “ 3 8 4 4 8 4 9 5 0 4 8 3 4 7” yang kemudian akan dimasukkan ke dalam alat.

Proses pengujian server berikutnya adalah pengujian enkripsi data. Pengujian ini dilakukan untuk mengetahui kebenaran dari hasil proses enkripsi.

Berikut gambar dari hasil pengujian dekripsi pada server:

Gambar 4.13 Hasil dekripsi

Pada Gambar 4.13, lingkaran merah menunjukkan hasil dari pengujian dekripsi yang dilakukan pada server.

4.6.2 Pembahasan

Dari pengujian yang dilakukan pada program enkripsi pada server di dapatkan nilai:

ID User= 11

Putaran= 05625

Tanggal pembelian= 15/07/14

Indek pulsa= 1

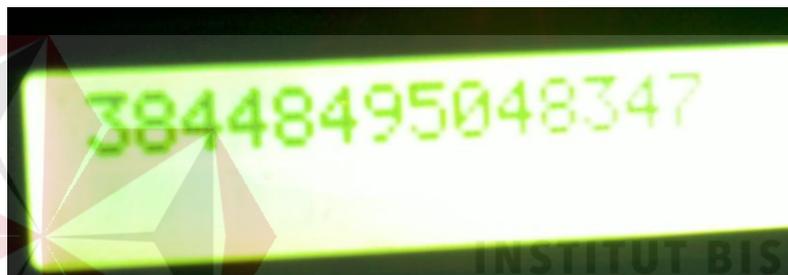
Data hasil pengujian tersebut sudah sesuai dengan input pulsa yang dimasukkan. Dengan demikian program server berjalan dengan baik sesuai dengan indikator keberhasilan yang tercantum dalam Tabel 3.7.

4.7 Pengujian Algoritma Dekripsi Pada Arduino

Pengujian algoritma dekripsi ini dilakukan dengan cara *step by step*. Yaitu dengan cara menguji tiap langkah yang ada, dari proses *caesar chiper* (dikurangi dengan 3) dan juga proses penukaran posisi tiap karakter. Pengujian ini dilakukan untuk mengetahui kebenaran dari program dekripsi pada arduino.

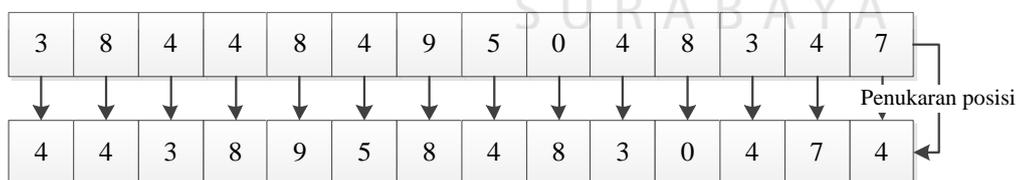
4.7.1 Hasil Pengujian Algoritma Dekripsi Pada Arduino

Gambar dibawah ini menunjukkan input pulsa yang dimasukkan melalui *keypad*.



Gambar 4.14 Input pulsa

Gambar 4.14 adalah tampilan dari proses input pulsa pada alat melalui *keypad*. Kemudian setelah di inputkan, sistem akan melakukan proses selanjutnya yaitu penukaran posisi tiap karakter. Berikut penjelasannya:



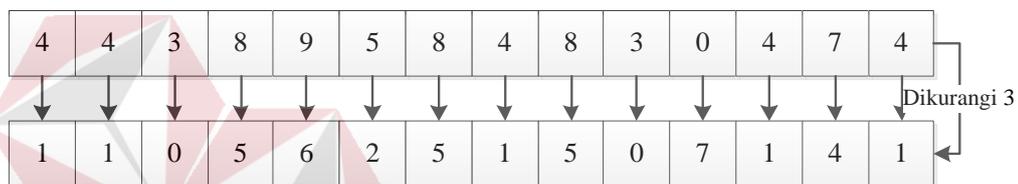
Gambar 4.15 Penukaran Posisi Karakter Saat Dekripsi

Proses pada Gambar 4.15 diatas adalah langkah pertama yang dilakukan sistem ketika pulsa berhasil di inputkan. Posisi karakter ditukar sesuai dengan nilainya masing-masing.



Gambar 4.16 Penukaran posisi karakter

Gambar 4.16 adalah hasil tampilan saat proses penukaran selesai. Setelah proses penukaran posisi karakter selesai kemudian sistem akan melanjutkan ke proses berikutnya yaitu melakukan dekripsi dengan metode *caesar chiper* seperti berikut:



Gambar 4.17 Proses Dekripsi Menggunakan *Caesar Chipper*

Perhitungan pada Gambar 4.17 di atas adalah proses dekripsi yang dilakukan oleh sistem. Dekripsi dilakukan dengan menggunakan metode *caesar chiper* geser 3. Yaitu dengan cara tiap karakter dikurangi dengan 3.



Gambar 4.18 Hasil Dekripsi

Gambar 4.18 menunjukkan tampilan dari hasil proses dekripsi yang dilakukan oleh sistem. Dari gambar 4.18 dapat dilihat hasil dari serangkaian proses dekripsi yang dilakukan oleh server, yaitu:

ID User = 11

Putaran Sensor= 05625

Tanggal = 150714

Indek Pulsa = 1

Dan halis tersebut sudah sama dengan data yang di inputkan ke dalam server.

4.7.2 Pembahasan

Dari pengujian yang dilakukan pada program dekripsi pada alat di dapatkan nilai:

ID User= 11

Putaran= 05625

Tanggal pembelian= 150714

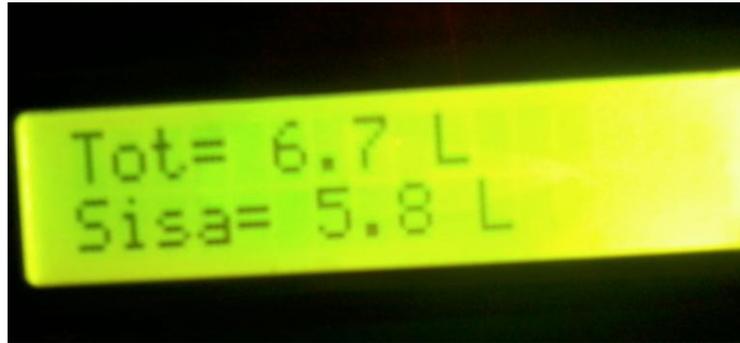
Indek pulsa= 1

Data hasil pengujian tersebut sudah sesuai dengan input pulsa yang dimasukkan. Dengan demikian program alat berjalan dengan baik sesuai dengan indikator keberhasilan yang tercantum dalam Tabel 3.8.

4.8 Pengujian Sistem Keseluruhan

Pengujian sistem dilakukan dengan mensimulasikan proses dari pembelian pulsa sampai proses counter oleh alat hingga pulsanya habis. Pengujian ini bertujuan untuk mengetahui kinerja sistem prabayar dan tingkat keakuratan outputnya.

4.8.1 Hasil Pengujian Keseluruhan



Gambar 4.19 Tampilan saat pengujian sistem

Pada gambar 4.19 menunjukkan tampilan saat proses pengujian sistem secara keseluruhan.

Tabel 4.4 Hasil pengujian sistem

No	Input Pulsa	Output air sebenarnya (liter)	Output air seharusnya (liter)	Error	Persen Error (%)
1	5000	2,6	2,5	0,1	3,8
2	5000	2,6	2,5	0,1	3,8
3	7000	3,4	3,5	0,1	2,9
4	7000	3,6	3,5	0,1	2,7
5	8000	3,9	4	0,1	2,6
6	8000	3,9	4	0,1	2,6
7	10000	4,9	5	0,1	2,1
8	10000	5,1	5	0,1	1,9
9	12000	6,1	6	0,1	1,6
10	12000	6,1	6	0,1	1,6
Rata-rata Error				0,1	2,56

4.8.2 Pembahasan

Berdasarkan tabel 4.4 didapatkan bahwa rata-rata kesalahan sistem ini adalah sebesar 2.56%. *Error* terjadi karena sifat dari rotor yang ada di dalam sensor. Ketika tekanan aliran air sangat rendah rotor sensor tidak berputar karena aliran air tidak cukup kuat untuk memutar rotor. Sedangkan ketika tekanan aliran air tinggi dan saat aliran itu dimatikan terdapat sisa-sisa tenaga yang memutar

rotor tersebut. Dari data tersebut dapat dikatakan pengujian ini berhasil jika dibandingkan dengan indikator keberhasilan yang tercentum dalam Tabel 3.9.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasar hasil pengujian didapatkan kesimpulan sebagai berikut:

1. Tingkat akurasi *flow water sensor* sebesar 97,44%. Data tersebut diperoleh dari hasil pengujian sensor dan dibandingkan dengan gelas ukur.
2. Proses *generate* pulsa oleh server berjalan dengan baik. Hal tersebut diketahui dengan membandingkan dengan perhitungan secara manual.
3. Penerapan sistem prabayar berjalan dengan baik sesuai dengan indikator keberhasilan.

5.2 Saran

Berikut ini terdapat beberapa saran yang penulis berikan untuk peneliti berikutnya apabila ingin mengembangkan sistem yang telah dibuat agar menjadi lebih baik adalah sebagai berikut:

1. Menggunakan metode lain supaya panjang kode pulsa tetap sama atau lebih kecil tapi mampu memberikan digit value pulsa yang lebih banyak lagi.
2. Menambahkan sistem enkripsi dekripsi yang lebih komplek untuk keamanan pulsa prabayar.

DAFTAR PUSTAKA

- Agfi. 2009. *Tutorial RTC*, (<http://agfi.staff.ugm.ac.id/blog/index.php/2009/>, di akses tanggal 21 April 2014)
- Anonim. 2013. *Water Flow Sensor*, (http://www.seeedstudio.com/wiki/G1/2_Water_Flow_sensor, di akses tanggal 25 April 2014)
- Arduino.cc. 2013. *Software Arduino IDE*, (<http://arduino.cc/en/main/software>, diakses tanggal 22 April 2014)
- Banzi, M. 2009. *Getting Started with Arduino*. America: O'Reilly.
- Ronalyw. 2013. *Sistem Barcode PDAM*, (<http://beritakotamakassar.com/index.php/metro-makassar/6996--inovasi-pencatatan-meteran-pdam.html>, diakses tanggal 17 Maret 2014)
- Sant, Hary. 2013. *Cara Kerja Relay*. (online). (<http://www.elangsakti.com/2013/03/pengertian-fungsi-prinsip-dan-cara.html>, diakses 12 April 2014).
- Vishay. 2002. *16 x 2 LCD Character*. (online). (<http://www.engineersgarage.com/sites/default/files/LCD%2016x2.pdf>, diakses 12 desember 2013).
- Wardana, Meri. 2012. *Cara Kerja Solenoid Valve*, (<http://www.guntursanjaya.com/2011/11/solenoid-valve.html>, di akses tanggal 16 April 2014)
- Zein, Miftah. 2010, *Caesar Chiper*, (<http://www.zenshifu.com/caesar-cipher/>, di akses tanggal 16 April 2014)