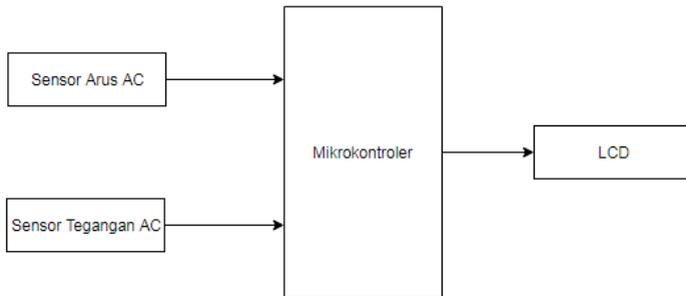


BAB III METODOLOGI PENELITIAN

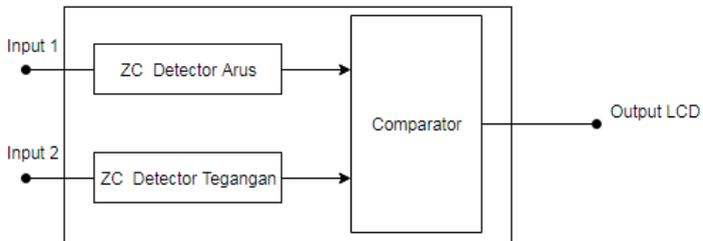
3.1 Model Rancangan Alat Ukur

3.1.1. Perancangan Hardware

Hardware atau perangkat keras merupakan salah satu bagian penting dalam pembuatan sistem ini. Perangkat yang dibuat terdiri dari satu papan dimana keseluruhan komponen dari sistem berada di papan tersebut. Di bagian papan terdapat satu lcd 20x4 yang akan memunculkan daya nilai $\cos \phi$ dan nilai kapasitor.



Gambar 3.1. Diagram Blok



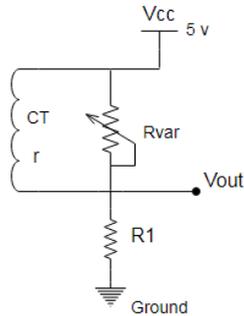
Gambar 3.2 Mikrokontroler

3.1.2. Faktor Daya

A. Daya (I , V)

B. Cos phi ($\cos\varphi$)

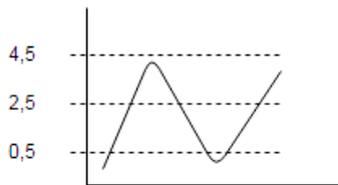
3.1.2.1. Perancangan Sensor Arus



Gambar 3.2 Rangkaian Arus

$$V = \frac{R1}{\left(\frac{r \times Rvar}{r + Rvar}\right) + R1} \times 5v$$

Tegangan pada CT dan pada Rvar keluaran Vout harus 2,5 volt karena pada gelombang sinus dimana gelombang sinus memiliki tegangan nol (0), min (-), plus (+) dan sedangkan arduino tidak bias mendeteksi tegangan min (-), maka dari itu gelombang kita geser ke atas seperti pada gambar dbawah ini.



Gambar 3.3 Gelombang Offset

a. Rumus Absolute

$$\begin{aligned} & \sqrt{(V_{out} - 2,5)^2} \\ &= \sqrt{(0,5 - 2,5)^2} \\ &= \sqrt{-2^2} \\ &= \sqrt{4} \\ &= 2 \end{aligned}$$

Atau

$$\begin{aligned} &= \sqrt{(4,5 - 2,5)^2} \\ &= \sqrt{2^2} \\ &= \sqrt{4} \\ &= 2 \end{aligned}$$

Dari rumus absolute diatas, baik itu min (-) maupun plus (+) hasilnya akan tetap plus (+), maka nilai rata – rata 2 v

b. Hukum Norton Thevenin

Thevenin adalah hambatan yang diukur atau dihitung pada terminal beban ketika beban dilepas dari rangkaian dan sumber arus dibuat menjadi nol atau dihubung singkatkan. Hambatan Norton adalah hambatan yang diukur atau dihitung ketika sumber arus dikurangi hingga nol dan hambatan beban dilepas. Hambatan Norton sama dengan hambatan thevenin.

$$I = V/R$$

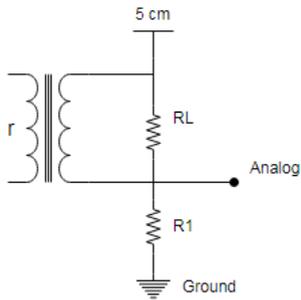
I = Arus (current), satuan dasarnya adalah ampere (A)

V = tegangan (voltage), satuan dasarnya adalah volt (V)

R = hambatan (resistance), satuan dasarnya adalah ohm (Ω)

3.1.2.2. Perancangan Sensor Tegangan

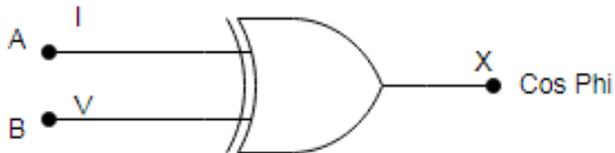
Rangkaian sensor tegangan ditunjukkan pada Gambar 3.4. Keluaran data dari sensor ini berbentuk tegangan analog dan akan diolah oleh Arduino agar tegangan tersebut dapat dibaca oleh sistem.



Gambar 3.4 Rangkaian Tegangan

Faktor daya 1 jika dihitung $1 - 1 = 0$ maka tidak ada selisih atau nilai sempurna. Jika faktor daya $0,9 = 1 - 0,9 = 0,1$ terdapat pada tiap satu gelombang, jika menggunakan 50 Hz maka periode gelombang 20ms, jadi jika perbedaan 2ms maka akan melenceng 0,1 atau 10%. Maka cara mendeteksinya.

a. XOR atau eksklusif OR

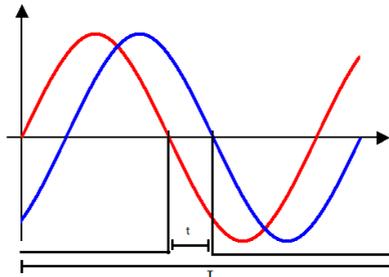


Gambar 3.5 Rangkaian XOR Atau Eksklusif OR

Berdasarkan tabel kebenarannya, bila nilainya sama maka akan jadi nol dan bila nilai tidak sama maka akan jadi satu.

I	V	X
0	0	0
0	1	1
1	0	1
1	1	0

Tabel 3.1 Logika



Gambar 3.6 Gelombang Sinus Cos Phi

Warna biru adalah tegangan

Warna merah adalah arus

Di waktu yang sama ketika arus mendapatkan fasa 0 tegangan masih 1 maka logikanya 1, dan jika tegangan sudah bergeser menjadi 0 maka arusnya 1. Dan jika dalam 1 periode 20ms berapa panjang periode arus dan tegangan tersebut :

$$\begin{aligned}
 \Delta T(\cos phi) &= \frac{t}{T} \\
 &= \frac{1 \text{ ms}}{20 \text{ ms}} \\
 &= 1 - 0,05 \\
 &= 0,95
 \end{aligned}$$

3.1.3. Perhitungan Nilai Capasitor

$$Q_c = Q_1 - Q_2$$

Q_c : Daya Reaktif Capasitor yang di butuhkan

Q_1 : Daya Reaktif sebelum perbaikan

Q_2 : Daya Reaktif yang ingin dicapai



Gambar 3.7 Segitiga Daya

1. Daya Semu
($S = V \times I$)
2. Daya Aktif
($P = V \times I \times \text{Cos}\varphi$)
3. Daya Reaktif
($Q = V \times I \times \text{Sin}\varphi$)

Jika Beban Resistif

$V = 228.60$ Volt, $I = 0.18$ A, $\text{Cos}\varphi_1 = 0,98$ ($\varphi = \cos^{-1}0.98 = 11,47^\circ$)
dan $\text{Cos}\varphi_2 = 0,99$ ($\varphi = \cos^{-1}0,99 = 8,10^\circ$)

Maka :

- a) Daya Reaktif Q_1 sebelum perbaikan

$$Q_1 = \sqrt{S^2 - P^2}$$

$$Q_1 = \sqrt{(V \times I)^2 - (V \times I \times \text{Sin}\varphi)^2}$$

$$Q_1 = \sqrt{(228.60 \times 0.18)^2 - (228.60 \times 0.18 \times 11.47^\circ)^2}$$

$$Q_1 = \sqrt{1693,15 - 66,95}$$

$$Q_1 = \sqrt{1626,2}$$

$$Q_1 = 40,32 \text{ VAR}$$

b) Daya Reaktif Q2 dengan nilai $\cos\varphi$ yang ingin dicapai

$$Q2 = \sqrt{S^2 - P^2}$$

$$Q2 = \sqrt{(V \times I)^2 - (V \times I \times \sin\varphi)^2}$$

$$Q2 = \sqrt{(228,60 \times 0,18)^2 - (228,60 \times 0,18 \times 8,10^\circ)^2}$$

$$Q2 = \sqrt{1693,15 - 33,61}$$

$$Q2 = \sqrt{1659,54}$$

$$Q2 = 40,73 \text{ VAR}$$

c) Nilai Capacitor yang dibutuhkan

$$Qc = Q1 - Q2$$

$$Qc = 40,32 \text{ VAR} - 40,73 \text{ VAR}$$

$$Qc = -0,41 \text{ VAR}$$

Untuk merubah Farad ke μF dapat menggunakan rumus :

$$C = \frac{Qc}{-V^2 \times \omega}$$

$$C = \frac{-0,41 \text{ VAR}}{-228,60^2 \times 2\pi f}$$

$$C = \frac{-0,41 \text{ VAR}}{-52257,96 \times 2 \times 3,14 \times 50}$$

$$C = \frac{-0,41 \text{ VAR}}{-16408999,4}$$

$$C = 0,25 \times 10^{-7} \text{ Farad}$$

$$C = 0,025 \mu F$$

Beban Induktif

$V = 229,95 \text{ Volt}$, $I = 0,78 \text{ A}$, $\cos\varphi_1 = 0,98$ ($\varphi = \cos^{-1}0,98 = 11,47^\circ$)

dan $\cos\varphi_2 = 0,99$ ($\varphi = \cos^{-1}0,99 = 8,10^\circ$)

Maka :

a) Daya Reaktif Q1 sebelum perbaikan

$$Q1 = \sqrt{S^2 - P^2}$$

$$Q1 = \sqrt{(V \times I)^2 - (V \times I \times \sin\varphi)^2}$$

$$Q1 = \sqrt{(229,95 \times 0,78)^2 - (229,95 \times 0,78 \times 11,47^\circ)^2}$$

$$Q1 = \sqrt{32170,36 - 1272,12}$$

$$Q1 = \sqrt{30898,24}$$

$$Q1 = 175,77 \text{ VAR}$$

b) Daya Reaktif Q2 dengan nilai $\cos\varphi$ yang ingin dicapai

$$Q2 = \sqrt{S^2 - P^2}$$

$$Q2 = \sqrt{(V \times I)^2 - (V \times I \times \sin\varphi)^2}$$

$$Q2 = \sqrt{(229,95 \times 0,78)^2 - (229,95 \times 0,78 \times 8,10^\circ)^2}$$

$$Q2 = \sqrt{32170,36 - 638,68}$$

$$Q2 = \sqrt{31531,68}$$

$$Q2 = 177,57 \text{ VAR}$$

c) Nilai Capacitor yang dibutuhkan

$$Qc = Q1 - Q2$$

$$Qc = 175,77 \text{ VAR} - 177,57 \text{ VAR}$$

$$Qc = -1,8 \text{ VAR}$$

Untuk merubah Farad ke μF dapat menggunakan rumus :

$$C = \frac{Qc}{-V^2 \times \omega}$$

$$C = \frac{-1,8 \text{ VAR}}{-229,95^2 \times 2\pi f}$$

$$C = \frac{-1,8 \text{ VAR}}{-52877,0025 \times 2 \times 3,14 \times 50}$$

$$C = \frac{-16603378,78}{-16603378,78}$$

$$C = 1,084 \times 10^{-7} \text{ Farad}$$

$$C = 0,108 \mu F$$

Beban Kapasitif

$V = 229,95 \text{ Volt}$, $I = 0,63 \text{ A}$, $\cos\varphi_1 = 0,95$ ($\varphi = \cos^{-1}0,95 = 18,19^\circ$)

dan $\cos\varphi_2 = 0,99$ ($\varphi = \cos^{-1}0,99 = 8,10^\circ$)

Maka :

a) Daya Reaktif Q1 sebelum perbaikan

$$Q1 = \sqrt{S^2 - P^2}$$

$$Q1 = \sqrt{(V \times I)^2 - (V \times I \times \sin\varphi)^2}$$

$$Q1 = \sqrt{(229,95 \times 0,63)^2 - (229,95 \times 0,63 \times 18,19^\circ)^2}$$

$$Q1 = \sqrt{20986,88 - 2045,16}$$

$$Q1 = \sqrt{18941,72}$$

$$Q1 = 137,62 \text{ VAR}$$

b) Daya Reaktif Q2 dengan nilai $\cos\phi$ yang ingin dicapai

$$Q2 = \sqrt{S^2 - P^2}$$

$$Q2 = \sqrt{(V \times I)^2 - (V \times I \times \sin\phi)^2}$$

$$Q2 = \sqrt{(229,95 \times 0,63)^2 - (229,95 \times 0,63 \times 8,10^\circ)^2}$$

$$Q2 = \sqrt{20986,88 - 416,65}$$

$$Q2 = \sqrt{20570,23}$$

$$Q2 = 143,42 \text{ VAR}$$

c) Nilai Capacitor yang dibutuhkan

$$Qc = Q1 - Q2$$

$$Qc = 137,62 \text{ VAR} - 143,42 \text{ VAR}$$

$$Qc = -5,8 \text{ VAR}$$

Untuk merubah Farad ke μF dapat menggunakan rumus :

$$C = \frac{Qc}{-V^2 \times \omega}$$

$$C = \frac{-5,8 \text{ VAR}}{-229,95^2 \times 2\pi f}$$

$$C = \frac{-5,8 \text{ VAR}}{-52877,0025 \times 2 \times 3,14 \times 50}$$

$$C = \frac{-16603378,78}{-16603378,78}$$

$$C = 34 \times 10^{-7} \text{ Farad}$$

$$C = 0,34 \mu F$$

3.1.4. Perancangan Software

Perancangan software pada penelitian ini adalah menuliskan algoritma pemrograman pada Arduino agar data-data sensor dapat dibaca. Mengingat data-data dari sensor bila tidak diolah nilainya maka tidak akan bisa dipakai sebagai penghitung kapasitor. Perancangan software juga bertujuan agar nilai-nilai hasil olah data Arduino dapat ditampilkan ke LCD display agar pengguna dapat mengetahui kondisi dan hasil perhitungan secara real-time.

Output sensor tegangan akan dimasukkan ke Arduino dengan cara disambungkan pada pin analog. Cara ini digunakan karena output dari sensor tegangan berbentuk tegangan analog yang masih berubah-ubah. Langkah pemrogramannya diawali dengan mengambil data secara terus menerus sebanyak 157 kali pengambilan. Cara ini disebut dengan

sampling data tegangan. Sampling diperlukan karena Arduino tidak dapat mengambil data analog dalam satu siklus secara pasti. Total data pengambilan sebanyak 157 kali tersebut akan dijumlahkan dan kemudian akan diambil rata-ratanya. Rata-rata dari proses sampling inilah yang akan dianggap Arduino sebagai nilai output dari sensor. Namun nilai ini masih belum dapat digunakan karena masih berbeda dengan nilai tegangan sebenarnya. Langkah selanjutnya adalah membuat suatu konstanta agar nilai dari sensor tegangan tersebut dapat sesuai dengan nilai tegangan PLN. Penentuan konstanta didasarkan pada nilai referensi yang dipakai yaitu dari alat ukur voltmeter. Cara ini dilakukan agar nilai tegangan yang semula kecil, dapat mendekati nilai yang sebenarnya.

Pengolahan data pada sensor arus oleh Arduino tidak jauh beda dengan pengolahan data sensor tegangan. Perbedaan terletak pada algoritma penentuan titik tengah tegangan 5V power supply terlebih dahulu. Algoritma ini dijalankan pada Arduino saat pertama kali diaktifkan. Program ini dibuat berdasarkan datasheet sensor yang mengatakan jika sensor tidak dialiri arus, maka tegangan output dari sensor adalah 2,5 V. Algoritma sampling pada sensor arus hampir sama dengan algoritma sampling sensor tegangan. Hanya saja saat melakukan proses sampling, data tiap sampling dikurangi dengan nilai titik tengah tegangan 5V. Karena yang dicari adalah arus RMS (Root Mean Square) maka total data sampling sensor arus berasal dari hasil perkalian pangkat 2 dari tiap data sampling yang sudah dikurangi dengan nilai titik tengah 5V. Hasil akhir dari algoritma ini dihitung berdasarkan akar kuadrat dari total data sampling dibagi dengan jumlah sampling dan dikali dengan resolusi ADC (Analog to Digital Converter). Hasil akhir ini nantinya yang akan digunakan sebagai data arus pada algoritma perhitungan nilai kapasitor. Sinyal output dari rangkaian beda fasa berbentuk sinyal kotak yang logikanya menggambarkan sudut beda fasa antara tegangan dan arus. Tugas dari Arduino adalah mengkonversi sinyal tersebut yang semula dalam kawasan waktu ke dalam kawasan sudut agar dapat dicari nilainya. Algoritma ini juga diperlukan untuk Arduino karena dalam perhitungan kapasitor membutuhkan nilai beda fasa dalam kawasan sudut.

Perancangan software selanjutnya dan yang paling utama adalah membuat algoritma perhitungan kapasitor. Algoritma ini akan memproses seluruh parameter sensor yang telah diolah yaitu sensor tegangan, arus, dan beda fasa. Algoritma ini dibangun berdasarkan rumus perhitungan

kapasitor. secara garis besar, langkah-langkah algoritma penghitung kapasitor adalah sebagai berikut:

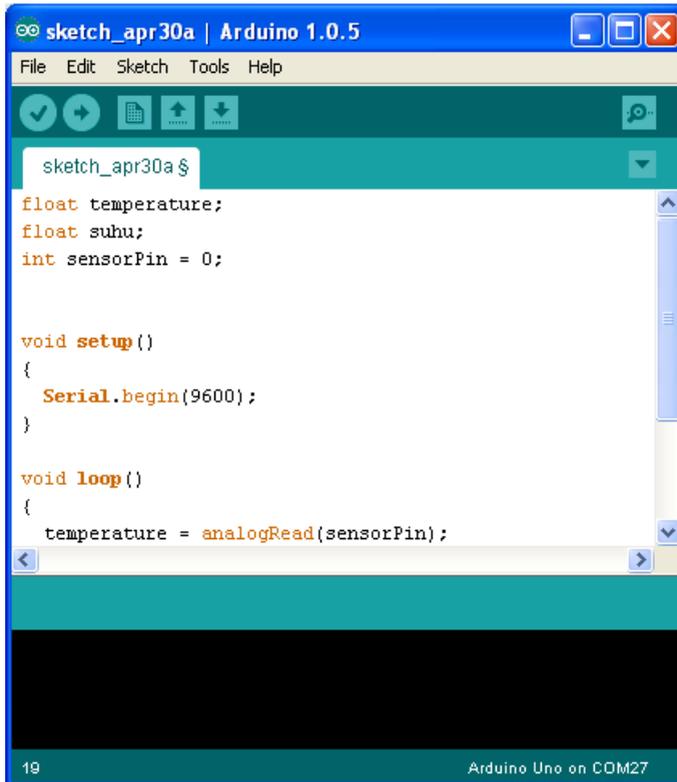
- a. Ukur tegangan, arus, faktor daya melalui sensor
- b. Hitung daya aktif dengan menggunakan persamaan 2
- c. Simpan nilai sudut dari faktor daya awal, hitung nilai sudut dari faktor daya harapan
- d. Hitung nilai \sin dari faktor daya awal dan faktor daya harapan, simpan dengan variabel yang berbeda
- e. Hitung arus reaktif awal
- f. Hitung arus total harapan jika faktor daya harapan terpenuhi berdasarkan persamaan 2
- g. Hitung arus reaktif harapan
- h. Hitung selisih antara arus reaktif awal dan arus reaktif harapan
- i. Hitung impedansi kapasitor
- j. Hitung nilai kapasitansi kapasitor menggunakan persamaan 4 dan ubah nilai akhirnya ke dalam satuan μF (micro Farad)

Pada perancangan software, proses pemrograman Arduino menggunakan 2 buah Arduino. Arduino Nano berfungsi sebagai pemroses utama yang didalamnya menangani pengolahan data sensor tegangan dan arus, pemrograman display lcd, penerima data dari Arduino Uno, dan pemrograman algoritma perhitungan kapasitor. Arduino yang kedua yaitu Arduino Uno yang berfungsi sebagai pemroses data sensor beda fasa dan mengirimkan hasil pemrosesan tersebut ke Arduino Nano.

Pemrograman menggunakan dua buah Arduino dibangun atas keterbatasan kemampuan peneliti dalam membuat program. Hal ini didasarkan atas percobaan yang telah dilakukan selama penelitian ini berlangsung yang mengalami kesulitan pembuatan program bila hanya menggunakan satu buah Arduino sebagai pemroses utama. Kesulitan ini muncul karena pada saat seluruh pemrograman dicoba dijalankan dalam satu Arduino, aliran program menjadi kacau dan mengakibatkan tidak akuratnya data-data hasil perhitungan. Peneliti mengalami kesulitan menemukan algoritma pemrograman yang efektif yang bisa dijalankan dalam satu Arduino. Faktor inilah yang menyebabkan peneliti memilih menggunakan dua buah Arduino sebagai pemroses yang hal ini bertujuan untuk meningkatkan akurasi data-data perhitungan pada alat yang dibuat pada penelitian ini. Pada penelitian selanjutnya, semoga hal ini dapat diperbaiki yaitu bisa menggunakan satu buah Arduino sebagai pemroses utama.

3.1.5. Menggunakan arduino IDE

Software Arduino IDE (Integrated Development Environment)



Gambar 3.8 Tampilan Software Editor Arduino

Arduino diciptakan untuk pemula, bahkan yang tidak memiliki basic bahasa pemrograman sama sekali karena menggunakan bahasa C++ yang dipermudah melalui library. Arduino menggunakan software processing yang digunakan untuk menulis program kedalam Arduino. Processing merupakan penggabungan antara bahasa C++ dan Java. Software Arduino ini dapat di-install di berbagai OS, seperti: LINUX, Mac OS dan Windows. Seperti pada Gambar 2.11 di atas, merupakan tampilan umum

software IDE Arduino. Software IDE Arduino terdiri dari 3 bagian, antara lain sebagai berikut :

- a. Editor Program, berfungsi untuk menulis dan mengedit program dalam bahasa processing. Listing program pada arduino disebut sketch.
- b. Compiler, merupakan modul yang berfungsi mengubah bahasa processing (kode program) ke dalam kode biner. Karena kode biner adalah satu-satunya bahasa program yang dipahami oleh mikrokontrol.
- c. Uploader, merupakan modul yang berfungsi untuk memasukkkan kode biner ke dalam memori mikrokontrol.

Struktur perintah pada arduino secara garis besar terdiri dari 2 bagian, yaitu:

1. void setup berisi perintah atau fungsi yang akan dieksekusi hanya satu kali pada awal sebuah program sejak arduino dihidupkan dan dapat menginisialisasi pengaturan.
2. void loop berisi perintah yang akan dieksekusi berulang-ulang selama daya pada arduino dinyalakan.