

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Pengerjaan alat dan penelitian dilaksanakan mulai 1 April 2017 hingga selesai di laboratorium Instrumentasi dan ruang dosen pembimbing I gedung Biomol lantai 3 Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.

3.2 Alat dan Bahan

Bahan yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Timah
2. PCB
3. IC LM339
4. IC LM358N
5. Transformator CT dan Transformator *non* CT
6. *Regulator Voltage* 7805,7815
7. *Regulator Voltage* 7905,7915
8. Dioda Rectifier 3 Ampere
9. Dioda 1N4007
10. Dioda Zener $\frac{1}{4}$ Watt (15 Volt dan 2,2 Volt)
11. LED
12. Kabel *Jumper*
13. Kapasitor 2200 μ F 220 Volt
14. Kapasitor 100 nF 16 Volt
15. *Header*
16. Resistor $\frac{1}{4}$ Watt (1k Ω , 10k Ω , 33k Ω , 100k Ω)

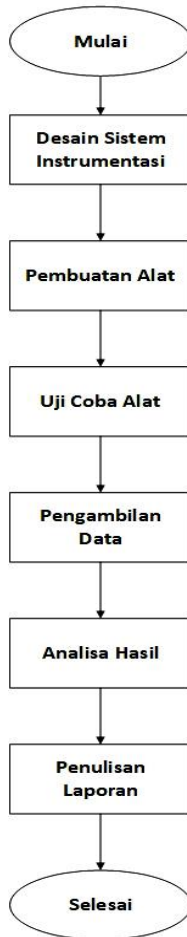
Alat yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Mikrokontroler Arduino Mega
2. *Power Supply*
3. *Signal Generator* Analog Griffin & George LTD Seri 644
4. Multimeter Heles UX866TR
5. Osiloskop Digital Tektronix TDS 1001B
6. Laptop *HP*

7. Solder

3.3 Alur Penelitian

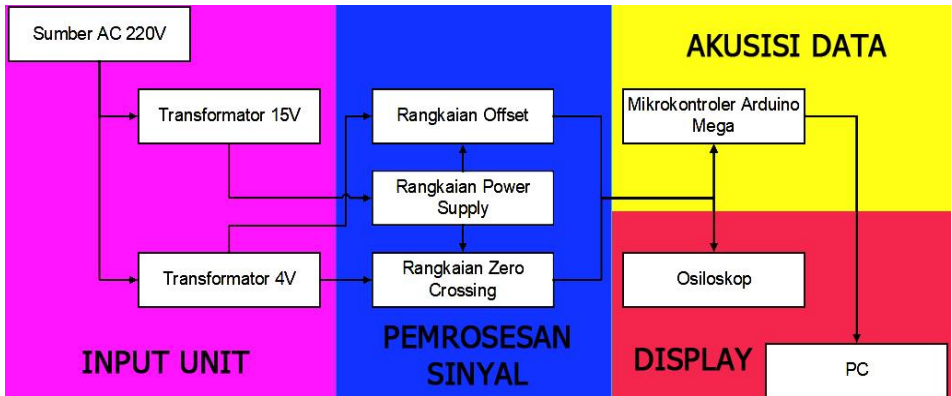
Penelitian ini terdiri dari beberapa tahapan yang dikerjakan seperti yang ditunjukkan pada Gambar 3.1. Alur penelitian dibagi menjadi beberapa bagian yang akan dijelaskan pada sub bab selanjutnya.



Gambar 3.1 Alur penelitian

3.3.1 Desain Sistem Instrumentasi

Tahap awal dalam mendesain sebuah alat adalah membuat diagram *plan* sistem instrumentasi. Diagram *plan* sistem berfungsi untuk mengetahui alur sistem kerja alat yang akan dibuat seperti yang terlihat pada Gambar 3.2 merupakan alur sistem kerja dari alat pengukur *total harmonic distortion* tegangan AC PLN.

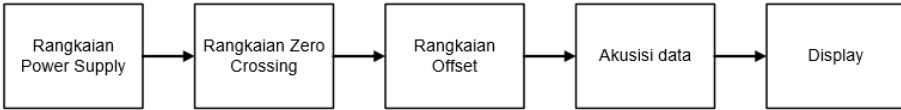


Gambar 3.2 Alur kerja sistem

Alur kerja sistem mempunyai 4 bagian yang penting, seperti yang terlihat pada Gambar 3.2 bagian tersebut dibedakan dengan warna yang berbeda. Pada blok warna ungu merupakan blok *Input*, dimana bagian tersebut penentuan objek yang akan diinputkan ke blok proses. Objek yang digunakan yaitu tegangan AC 220V yang diturunkan tegangan nya sebesar 4V AC dan 15V AC dengan menggunakan transformator. Kemudian blok warna biru merupakan blok Proses, pada bagian ini objek akan di proses agar didapatkan keluaran sesuai dengan yang diharapkan dengan menggunakan rangkaian *zero crossing*, rangkaian *offset*, dan rangkaian *power supply* sebagai catu daya. Pada blok warna kuning merupakan blok Akusisi Data, dimana keluaran dari blok proses akan di kondisikan dan dikendalikan dengan program yang ditanamkan pada mikrokontroler arduino mega sehingga didapatkan *output* yang diharapkan. Bagian terakhir yaitu blok merah merupakan blok *Display*, hasil dari pengukuran dapat ditampilkan pada PC dan osiloskop sehingga memudahkan untuk analisa hasil pengukuran.

3.3.2 Pembuatan Alat

Pembuatan alat pengukur *total harmonic distortion* tegangan AC PLN dibagi menjadi beberapa tahap. Tahap pembuatan alat dalam penelitian ini dapat dilihat pada Gambar 3.3.



Gambar 3.3 Tahap pembuatan alat

Proses pembuatan sistem instrumentasi pengukur *total harmonic distortion* tegangan AC PLN ini dimulai dari mendesain rangkaian *power supply*. Rangkaian *power supply* didesain untuk menghasilkan beberapa keluaran yang berbeda – beda sesuai dengan kebutuhan penelitian. Tahapan selanjutnya peneliti mendesain rangkaian *zero crossing* dan rangkaian *offset*, kedua rangkaian tersebut akan diberi sinyal AC sebesar 4V dari transformator sebagai *input* dari rangkaian dan diberi sumber tegangan 5V DC dari rangkaian *power supply*. Rangkaian *zero crossing* mengeluarkan *output* berupa sinyal kotak atau sinyal *digital* yang kemudian akan di *input* kan pada mikrokontroler arduino mega. Rangkaian *offset* juga diberi sinyal AC sebesar 4V dari transformator sebagai *input* dari rangkaian dan diberi sumber tegangan 5V DC sebagai sumber tegangan dari rangkaian *power supply*. Keluaran dari rangkaian *offset* berupa sinyal *analog* berjenis sinusoidal yang kemudian dijadikan *input* ke pin ADC mikrokontroler arduino mega. Pada penelitian ini mikrokontroler yang digunakan yaitu arduino mega, mikrokontroler arduino mega merupakan mikrokontroler yang banyak digunakan dalam pembelajaran karena sudah berbentuk modul dan mudah untuk digunakan. Hasil dari kedua rangkaian akan di tampilkan sinyalnya pada osiloskop dan hasil dari akusisi data akan ditampilkan pada display PC.

A. Rangkaian *Power Supply*

Pada penelitian kali ini rangkaian *power supply* dirancang untuk menghasilkan nilai tegangan 5V dan 2V. Tegangan tersebut dijadikan sebagai sumber tegangan untuk rangkaian *zero crossing* dan *offset*. Gambar 3.4 merupakan gambar skematik dari rangkaian *power supply* dan Gambar 3.5 merupakan gambar rangkaian *power supply*. rangkaian *power supply* mendapatkan sumber tegangan dari transformator CT 1A sebesar 15V AC seperti yang terlihat pada Gambar 3.6 yang dihubungkan ke sumber tegangan PLN. Transformator digunakan untuk menurunkan tegangan PLN sebagai supply tegangan masukan rangkaian *power supply*, tegangan *input* yang dipakai sebesar +15V dan -15V AC. Tegangan tersebut diubah menjadi tegangan DC dengan komponen dioda *rectifier* (penyearah gelombang penuh). Transformator yang digunakan dalam penelitian merupakan jenis *step down* yang dapat mengubah tegangan listrik AC besar menjadi tegangan listrik AC yang lebih rendah. Pada penelitian kali ini rangkaian *power supply* memiliki tegangan keluaran sebesar 15V dan 5V positif negatif.



Gambar 3.5 Rangkaian *power supply*



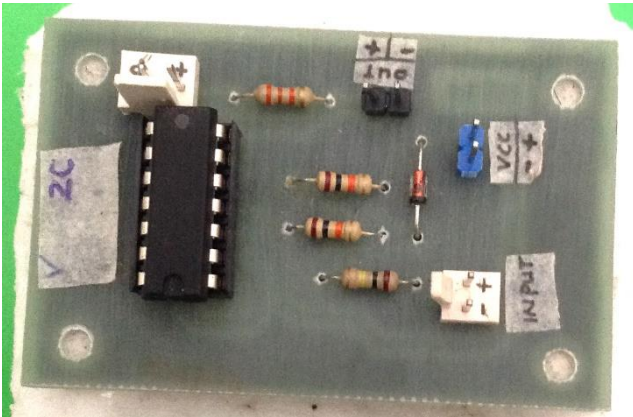
Gambar 3.6 Transformator CT

Peneliti menggunakan IC *regulator* untuk mempertahankan atau memastikan nilai tegangan pada level tertentu, dengan arti lain tegangan output DC pada *voltage regulator* tidak dipengaruhi oleh perubahan tegangan *input* maupun beban lainnya sehingga tegangan akan stabil. IC *regulator* yang digunakan yaitu LM7815 untuk mendapatkan tegangan sebesar 15V, LM7915 untuk mendapatkan tegangan sebesar -15V, LM7805 untuk mendapatkan tegangan 5V dan LM7905 untuk mendapatkan nilai tegangan sebesar 15V. Peneliti sengaja menambahkan IC regulator LM7915 dan LM7815 dikarenakan jika tidak diberikan IC tersebut *transformator* menjadi bergetar dan IC LM7905 dan LM7805 mengalami kenaikan suhu. Pada penelitian kali

ini dibutuhkan juga nilai tegangan sebesar 2V peneliti menggunakan *trimpot* 10K yang fungsinya sama seperti *potensiometer* untuk mengubah-ubah nilai tegangan sesuai dengan yang dibutuhkan. Selain itu peneliti menggunakan *reistor fix* yang diberi dioda zener 2.2V yang mengeluarkan *output* tegangan 2V.

B. Rangkaian Zero Crossing

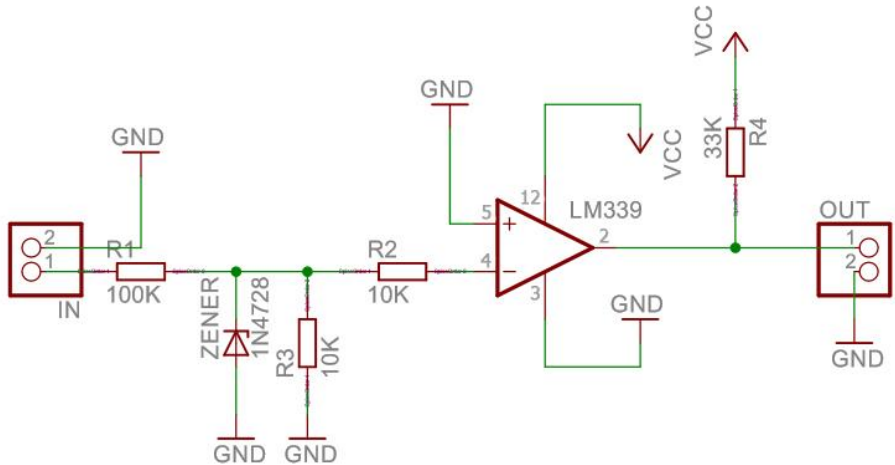
Rangkaian *zero crossing* pada penelitian ini digunakan untuk mendapatkan sinyal persilangan antara titik nol dengan keluaran sinyal kotak. Gambar 3.7 merupakan gambar dari rangkaian *zero crossing* dan Gambar 3.8 merupakan skematik dari rangkaian *zero crossing*.



Gambar 3.7 Rangkaian *zero crossing*

Rangkaian *zero crossing* terdiri dari satu masukan berupa sinyal AC PLN yang di turunkan tegangannya dengan transformator *non CT* 1A sebesar 4V AC. Gambar 3.9 merupakan gambar dari transformator *non CT* yang digunakan peneliti. Rangkaian *offset* terdiri dari beberapa komponen yaitu 4 buah *resistor* dengan nilai 100k Ω , 10 k Ω , 10k Ω , 33 k Ω , dioda zener dan sebuah IC LM 339. Dioda zener dipasang dengan bias balik yang tujuannya ketika input tegangan masuk tegangan akan turun sesuai dengan tegangan batasnya saat melewati dioda zener yang dipasang secara bias balik (*reverse bias*). IC LM339 merupakan sebuah komparator yang digunakan untuk membandingkan tegangan *input* dan tegangan referensi. Pembuatan rangkaian ini peneliti menggunakan satu

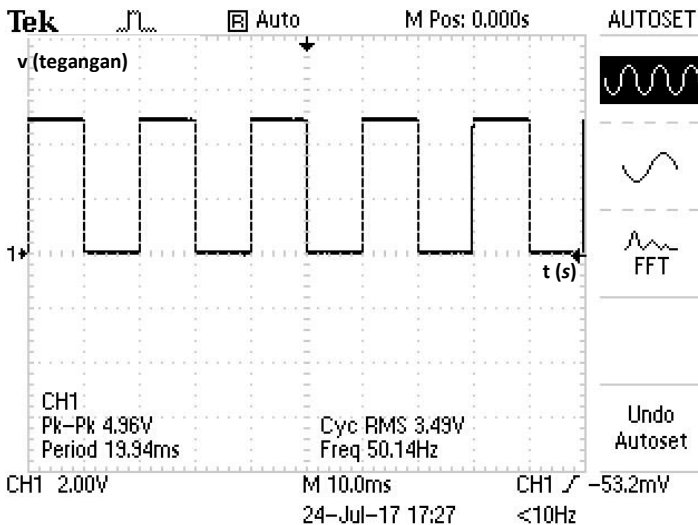
op-amp pada IC LM339 dengan masukan sinyal di *inverting* dan *ground* di *non inverting*. *Output* dari rangkaian *Zero Crossing* mengeluarkan data *digital* sehingga *output* dari rangkaian diberi *resistor pull up* yang berfungsi membuat nilai *output* berlogika 1 (*HIGH*) dengan kata lain, sejumlah arus yang mengalir antara *VCC* dan input (tidak ke *ground*), sehingga *output* dibaca mendekati *VCC*. *Output* dari rangkaian akan berupa sinyal kotak seperti pada Gambar 3.10 sinyal keluaran ini akan dijadikan *input* pada pin *digital* arduino mega.



Gambar 3.8 Skematik *zero crossing*



Gambar 3.9 Transformator non CT

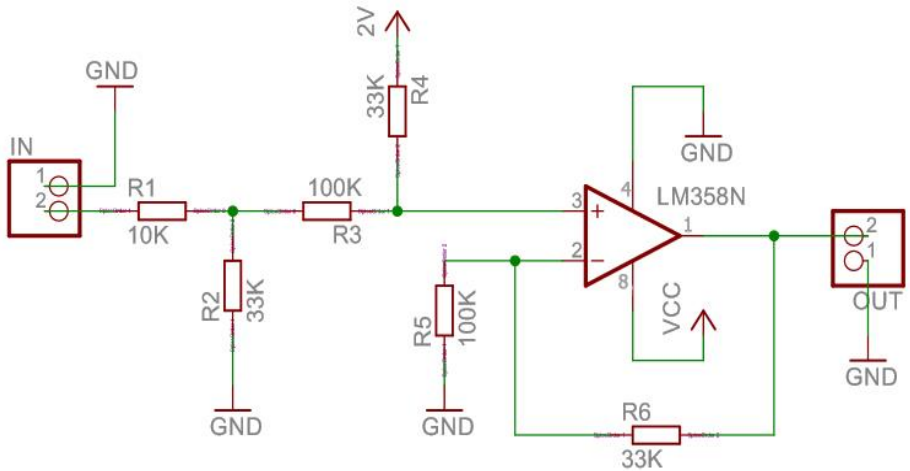


Gambar 3.10 Bentuk sinyal *output zero crossing*

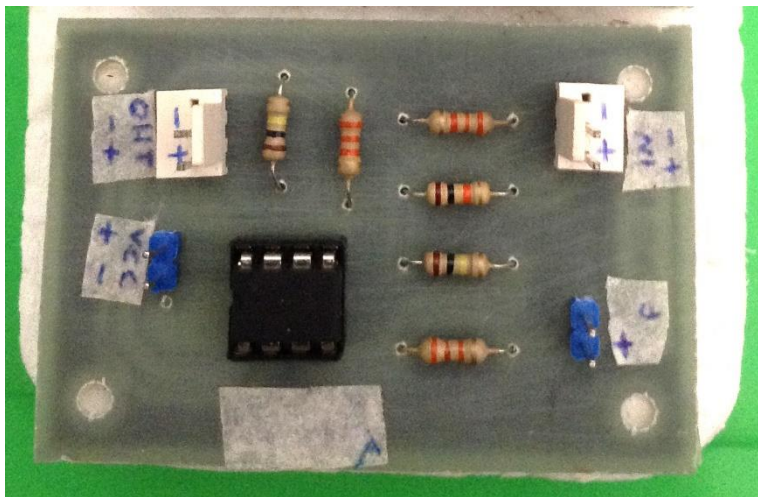
C. Rangkaian Offset

Rangkaian *offset* pada penelitian kali ini digunakan untuk mengkondisikan sinyal yang akan dideteksi distorsinya. Tujuan dari pembuatan rangkaian *offset* yaitu untuk menaikkan bentuk sinyal sebesar nilai dari tegangan *offset* yang diberikan sehingga tidak memiliki polaritas negatif. Pada Gambar 3.11 menunjukkan rangkaian didesain seperti penguat *non inverting* dimana input sinyal berada pada (+) *op-amp* yang bertujuan untuk mendapatkan sinyal dengan karakteristik dasar sinyal *output* yang dikuatkan memiliki fasa yang sama dengan sinyal *input*, dan (-) *op-amp* di *ground* kan dengan modus *loop* tertutup yang digunakan untuk memberikan umpan balik pada keluaran. Modus *loop* tertutup juga akan menstabilkan rangkaian dan mengurangi derau.

Dalam pembuatan rangkain *offset* peneliti menggunakan 6 buah *resistor* yang bernilai 10K, 33K, 100K, 33K, 100K, 33K, IC LM358N dan transformator *non CT* dengan keluaran 4V AC. *Input* pada rangkaian akan diberi sinyal dengan nilai 4V AC dari transformator *step down non CT* yang dihubungkan langsung dengan sumber tegangan PLN 220V. Gambar 3.12 merupakan bentuk jadi rangkaian *offset*.



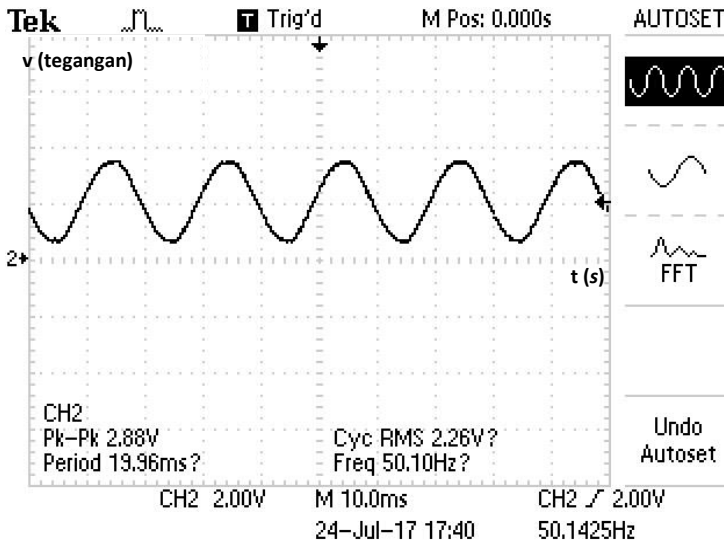
Gambar 3.11 Skematik rangkaian *offset*



Gambar 3.12 Rangkaian *offset*

Fungsi *resistor* yaitu digunakan untuk menghambat arus yang melewati rangkaian dua diantaranya digunakan untuk membagi tegangan dan satu diantaranya digunakan untuk menurunkan tegangan *offset* yang diberikan peneliti yaitu sebesar 2V DC. Tegangan *offset* masukan idealnya *op-amp* akan mengeluarkan tegangan nol manakala

masuk bernilai nol. Tetapi dengan memberikan tegangan *offset* pada masukan rangkaian, tegangan keluaran dapat dinolkan kembali. Pada pembuatan rangkaian ini peneliti menggunakan IC LM358N yang merupakan sebuah komparator yang digunakan untuk membandingkan tegangan input dengan tegangan referensi. Rangkaian ini membutuhkan tegangan sebesar 5V DC untuk menyuplai *op-amp* pada rangkaian. Keluaran dari rangkaian akan membentuk sinyal sinusoidal yang disertai dengan *offset* DC seperti pada Gambar 3.13.

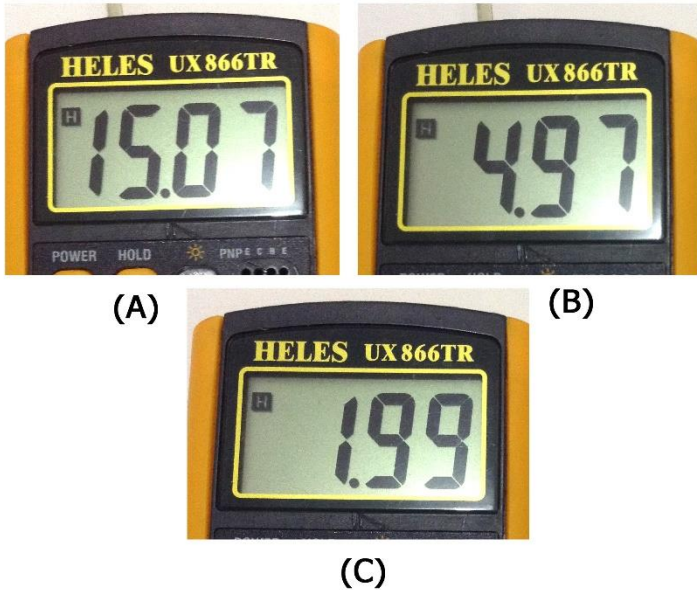


Gambar 3.13 Bentuk sinyal keluaran rangkaian *offset*

3.3.3 Uji Coba Alat

A. Pengujian Rangkaian *Power Supply*

Proses pengujian alat dimulai dari menguji rangkaian *power supply*. Pengujian *power supply* dilakukan dengan menggunakan *multimeter*, dimana untuk memastikan hasil keluaran dari *power supply* pada OUTPUT15, OUTPUT5, dan R.OUT sesuai dengan yang diharapkan yaitu bernilai 15V DC, 5V DC, dan 2V DC. Hasil dari pengujian tersebut dapat dilihat pada Gambar 3.14.

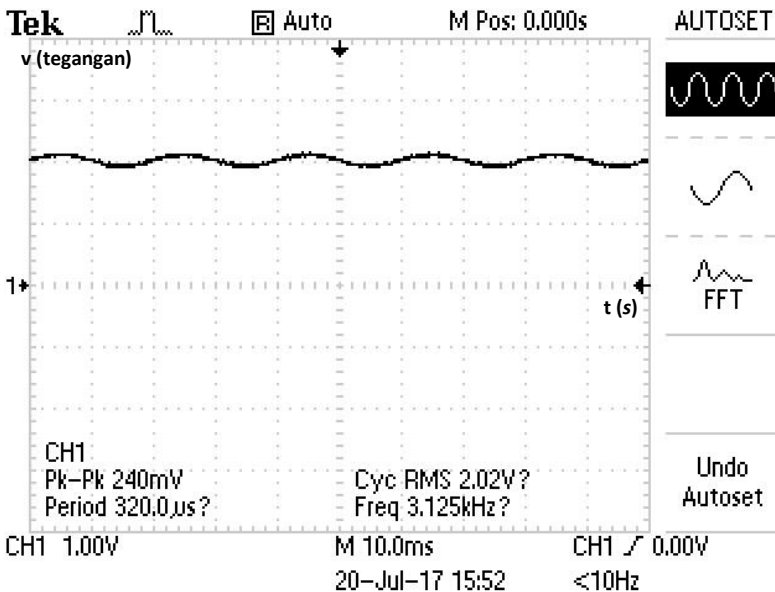


Gambar 3.14 Hasil pengujian dengan multimeter (A= 15,07V DC ; B= 4,97 V DC ; C= 1,99 V DC)

Hasil dari pengujian didapatkan nilai yang hampir mendekati nilai yang diharapkan yaitu bernilai 15,07V DC, 4,97V DC, pada *output trimpot* nilai tegangan dapat diubah-ubah sesuai dengan yang diinginkan tetapi nilai maksimal hanya sebesar 5V DC, dan pada keluaran dari *resistor fix* didapatkan tegangan sebesar 1,99V DC. Peneliti menggunakan *trimpot* dan *resistor fix* yang bertujuan untuk mendapatkan nilai tegangan sebesar 2V DC untuk *supply* pada rangkain *offset*. Pengujian *output* dari *trimpot* dan *resistor fix* dilakukan menggunakan osiloskop digital seperti yang terlihat pada Gambar 3.15 merupakan osiloskop *digital* Tekronix TDS 1001B yang dimiliki oleh laboratorium Instrumentasi Jurusan Fisika. Ketika diuji menggunakan multimeter tegangan *output* yang didapatkan dari *trimpot* sebesar 2V DC akan tetapi setelah di uji dengan osiloskop gelombang yang didapatkan tidak stabil seperti yang terlihat pada Gambar 3.16.



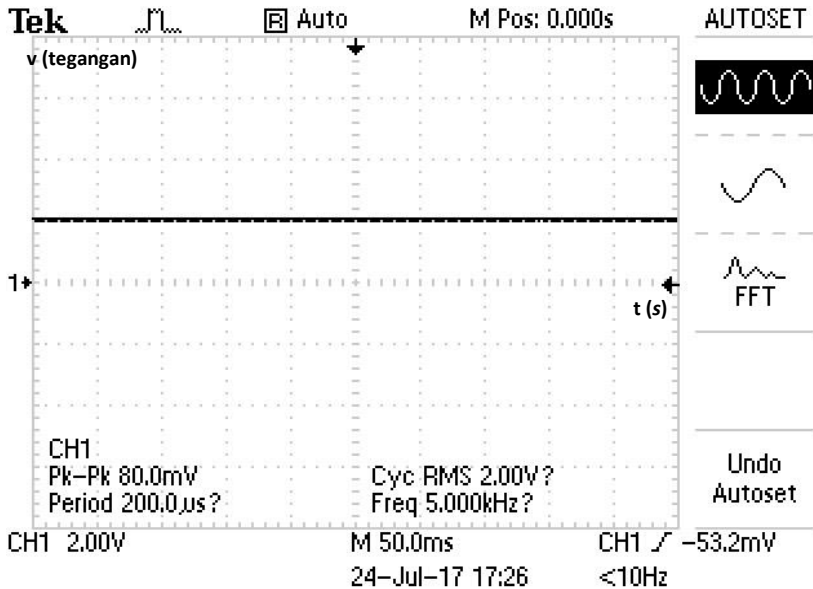
Gambar 3.15 Osiloskop digital *Tekronix TDS 1001B*



Gambar 3.16 Bentuk sinyal sebelum diberi dioda *zener*

Gambar 3.16 menunjukkan gelombang berbentuk sinusoidal. Setelah itu untuk mendapatkan tegangan 2V yang stabil peneliti menggunakan *resistor fix* dengan nilai resistor 560 Ω , 10k Ω dan

menambahkan dioda zener 2,2V yang dipasang bias balik. Tegangan yang didapatkan dari *resistor fix* yaitu sebesar 4,71V DC dan setelah ditambahkan dioda zener 2,2V tegangan berubah menjadi 1,99V DC. Ketika diuji menggunakan *osiloscope* gelombang yang didapatkan stabil terlihat seperti pada Gambar 3.17. Tegangan inilah yang nantinya akan di *input* kan pada rangkaian *offset*.



Gambar 3.17 Bentuk sinyal setelah diberi dioda *zener*

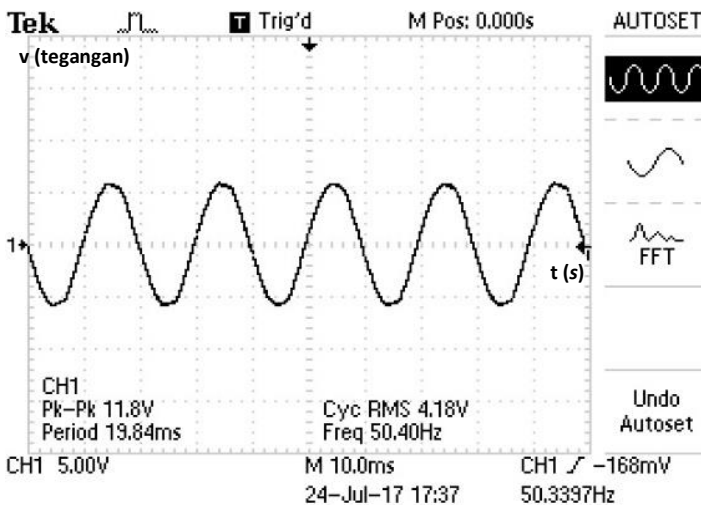
B. Pengujian Rangkaian *Zero Crossing*

Proses pengujian rangkaian *zero crossing* dimulai dengan memberi *input* tegangan AC sebesar 4V dan memberi *suppy* tegangan pada rangkaian sebesar 5V DC serta tegangan *pull up* sebesar 5V DC. Setelah itu digunakan multimeter untuk mengetahui nilai *output* pada rangkaian yaitu sebesar 2,51 V DC terlihat seperti pada Gambar 3.18. *Output* pada rangkaian *zero crossing* berupa tegangan DC dan berbentuk sinyal *digital*. Kemudian peneliti menggunakan *osiloscope* yang telah di *setting* V/DIV dengan nilai 2V/DIV untuk menguji bentuk sinyal *input* dan *output* pada rangkaian. Pertama – tama penguji menggunakan 1 buah *probe* pada *osiloscope* untuk mengetahui bentuk

sinyal *input* pada keluaran transformator. *Probe* osiloskop dihubungkan pada *ground* dan 4V, hasil yang didapatkan dapat dilihat pada Gambar 3.19.



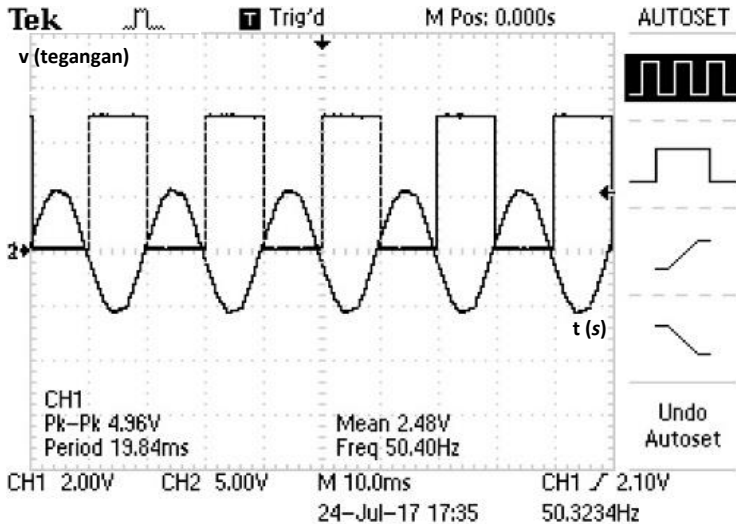
Gambar 3.18 Hasil *output* rangkaian *zero crossing* dengan multimeter



Gambar 3.19 Bentuk sinyal input

Bentuk sinyal yang didapatkan yaitu berupa sinyal AC sinusoidal yang mempunyai polaritas (-) dan (+) dengan nilai PP (*peak to peak*) sebesar 11,8 V. Kemudian digunakan 2 *probe* untuk melihat hasil dari *input* dan *output* pada rangkaian, *probe* 1 dihubungkan pada *input* di transformator dan *probe* 2 dibungkan pada *output* rangkaian *zero*

crossing. Hasil yang didapatkan yaitu berupa sinyal kotak dengan PP (*peak to peak*) sebesar 4,96 V, dari hasil yang didapatkan sesuai dengan yang diharapkan peneliti yaitu sinyal kotak yang merupakan sinyal digital yang nantinya akan dihubungkan pada *board* mikrokontroler arduino mega pada pin digital. Gambar 3.20 merupakan sinyal *input* dan *output* yang didapatkan dari rangkaian *zero crossing* .



Gambar 3.20 Bentuk sinyal *input* (sinus) dan *output* rangkaian *zero crossing* (kotak)

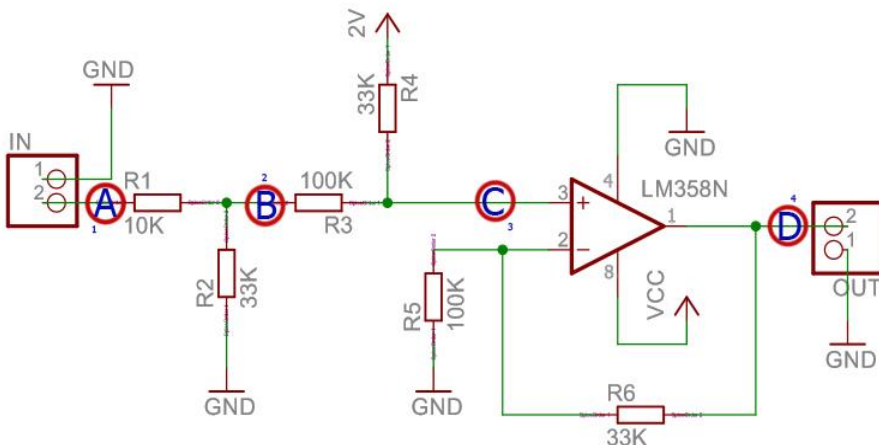
C. Pengujian Rangkaian *Offset*

Pengujian rangkaian *offset* dilakukan dengan menggunakan alat yang sama yaitu multimeter dan osiloskop. Uji coba rangkaian *offset* diawali dengan memberikan sinyal input dari transformator sebesar 4V AC, memberikan *supply* tegangan sebesar 5V DC dan tegangan *offset* sebesar 2V DC. Setelah itu dilanjutkan mengukur nilai *output* pada rangkaian, dimana nilai tegangan yang didapatkan dengan menggunakan multimeter sebesar 2,03V DC. Gambar 3.21 menunjukkan hasil pengukuran tegangan pada *output* rangkaian *offset*. Setelah itu dilakukan pengujian menggunakan osiloskop untuk mengetahui sinyal *input* dan *output*. Peneliti menguji rangkaian pada 4 titik yang berbeda

yaitu seperti yang terlihat pada Gambar 3.22 yang menunjukkan 3 titik yang akan dilihat bentuk sinyalnya.



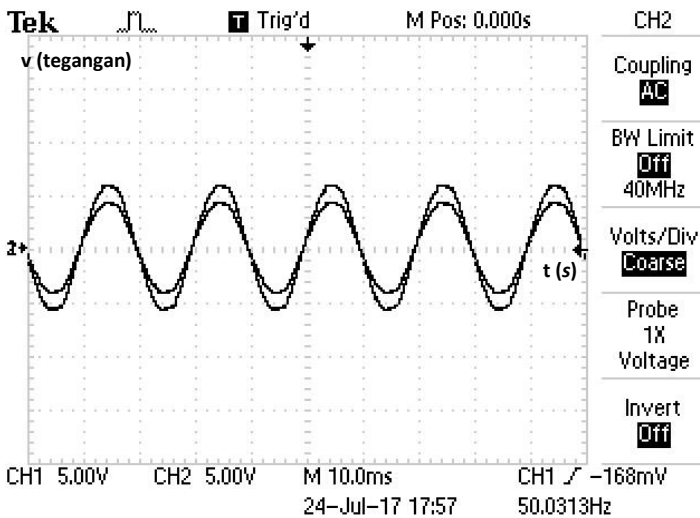
Gambar 3.21 Hasil *output* rangkaian *offset* dengan multimeter



Gambar 3.22 Konfigurasi pengujian rangkaian *offset*

Konfigurasi pengujian rangkaian *offset* yaitu menggunakan 2 *probe* yang terhubung pada osiloskop dengan *setting* V/DIV sebesar 2V/DIV, kemudian *probe* 1 dihubungkan pada titik A yaitu *input* dari transformator dan *probe* 2 dihubungkan pada titik B. Hasil yang didapatkan dari titik A dan B seperti pada Gambar 3.23 dimana titik A melihat sinyal *input* berbentuk sinusoidal dengan polaritas (-) dan (+) dengan nilai PP (*peak to peak*) sebesar 11,8 V sama seperti *input* pada rangkaian *zero crossing*. Pada titik B menunjukkan sinyal

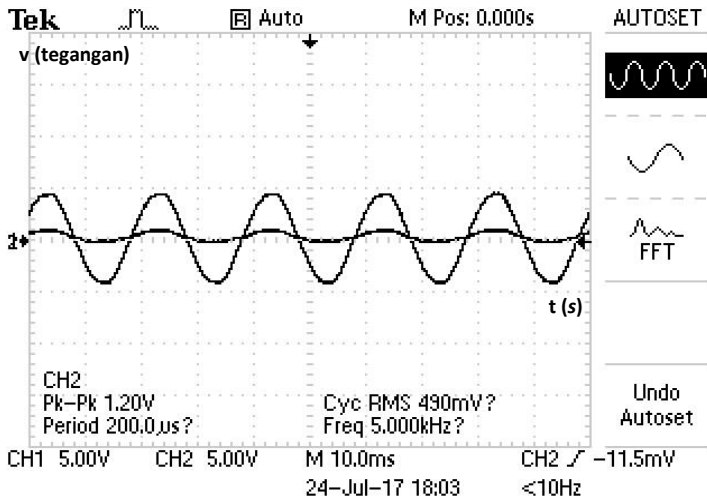
berbentuk sinusoidal yang mengalami penurunan sebesar 2 V pada polaritas (-) dan (+) dengan nilai PP (*peak to peak*) 8 V, dikarenakan dihambat dengan 2 resistor sehingga nilai tegangan berkurang.



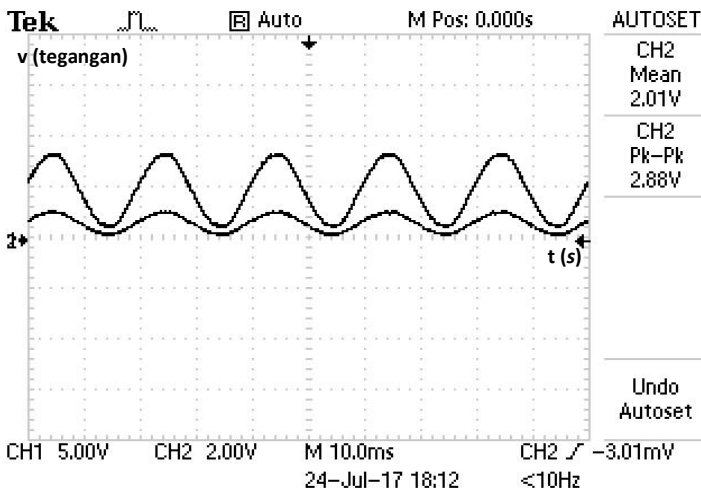
Gambar 3.23 Bentuk sinyal A (besar) dan B (kecil)

Setelah itu pengujian dilakukan pada titik B dan C, dimana titik B sudah diketahui. Pengujian tetap menggunakan 2 buah *probe*, *probe* 1 dihubungkan pada titik B dan *probe* 2 dihubungkan pada titik C. Sehingga didapatkan hasil seperti pada Gambar 3.24 yang menunjukkan pada titik B tetap bernilai sama, tetapi pada titik C mengalami perubahan bentuk sinyal yang awalnya mempunyai 2 polaritas (-) dan (+) menjadi satu polaritas (+) yang disebabkan karena pemberian tegangan *offset*. Bentuk gelombang pada titik C yaitu sinusoidal dengan PP (*peak to peak*) sebesar 1,02 V. Pengujian selanjutnya dilakukan pada titik C dan D dengan menggunakan 2 *probe*. Pada titik C dihubungkan dengan *probe* 1 dan titik D dihubungkan dengan *probe* 2. Hasil yang didapatkan tampak pada Gambar 3.25 dimana pada titik C tetap sama hasilnya tetapi pada titik D mengalami perubahan pada bentuk gelombang yang mengalami kenaikan tinggi gelombang dengan PP (*peak to peak*) sebesar 2,88 V. Dikarenakan penguatan dari *op-amp* yang digunakan serta rangkaian yang dibuat, nilai penguatan sebesar 1 ½ kalinya. Hasil yang didapatkan pada

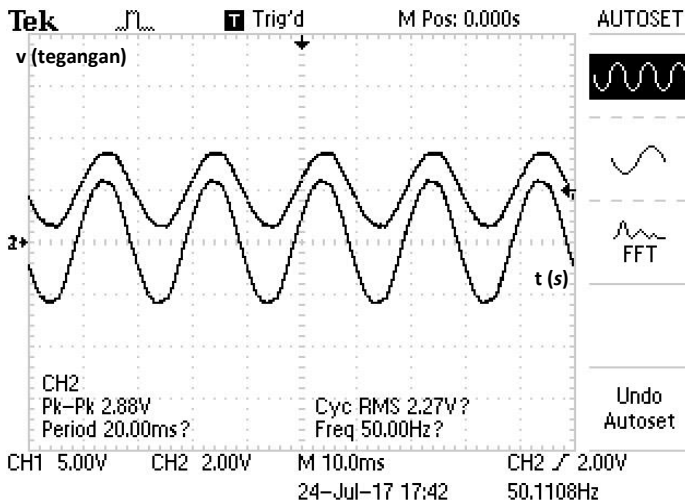
rangkaian *offset* yaitu berupa sinyal analog sinusoidal yang nantinya akan dihubungkan ke *board* mikrokontroler arduino mega pada pin analog. Gambar 3.26 menunjukkan hubungan antara sinyal *input* dengan sinyal *output* dari rangkaian *offset*.



Gambar 3.24 Bentuk sinyal B (besar) dan C (kecil)



Gambar 3.25 Bentuk sinyal C (kecil) dan D (besar)

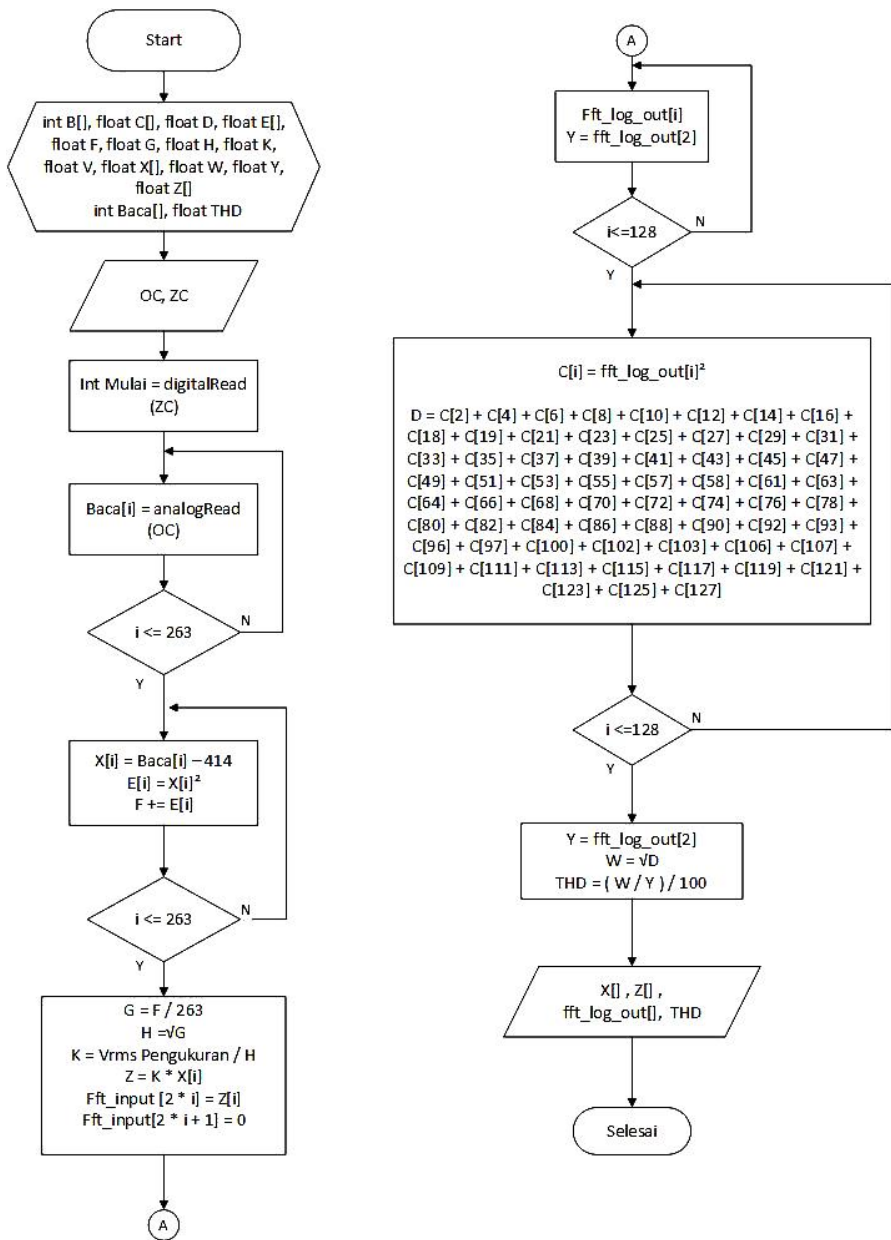


Gambar 3.26 Bentuk sinyal *input* (besar) dan *output* rangkaian *offset* (kecil)

3.3.4 Akusisi Data

Pada pemrosesan data penelitian kali ini, penulis menggunakan mikrokontroler arduino mega, dikarenakan pada penelitian ini menggunakan memori yang besar. Mikrokontroler arduino mega terhubung dengan rangkaian *power supply*, rangkaian *offset*, dan rangkaian *zero crossing*. Sinyal yang akan diproses yaitu berupa tegangan AC yang akan diinputkan pada rangkaian *offset* dan rangkaian *zero crossing*. Sinyal keluaran dari kedua rangkaian tersebut akan diproses pada mikrokontroler arduino mega.

Mikrokontroler arduino mega yang digunakan untuk memproses data sudah dalam bentuk modul, sehingga peneliti cukup membuat program dan memasukkan pada *software* arduino IDE. Gambar 3.27 merupakan diagram alir (*flowchart*) program mikrokontroler arduino mega. Untuk membuat program langkah pertama yang dilakukan yaitu menyiapkan laptop dan menginstal *software* arduino IDE yang nantinya digunakan untuk wadah membuat program serta *interface* hasil data. Setelah itu *software* dijalankan, pada halaman pertama sudah terdapat *void setup ()* dan *void loop ()*.



Gambar 3.27 Flowchart program

Sebelum membuat *source code* terlebih dahulu peneliti menginstal *library* yang digunakan untuk memproses data. Pada penelitian kali ini peneliti menggunakan *library* *fft.h* arduino, dimana *library* tersebut digunakan untuk mengetahui nilai frekuensi yang terdapat pada sinyal yang akan diukur. *Library* *fft.h* arduino merupakan implementasi cepat dari algoritma *fft* standar yang dapat digunakan untuk data nyata, *fft.h* hanya dapat menyimpan data maksimal 256 data frekuensi. Setelah itu langkah selanjutnya yang dilakukan mendeklarasikan *library* *fft* serta pin yang digunakan pada *board* mikrokontroler arduino mega. Selanjutnya menginialisasi variabel dan menjelaskan mode pin yang digunakan pada fungsi *void setup* (). Pin yang digunakan yaitu pin analog A0 sebagai sinyal yang akan diukur distorsinya dan pin digital 3 digunakan untuk *zero crossing* sebagai acuan untuk memulai pengambilan data. Sinyal *zero crossing* akan memberikan nilai LOW (0) dan HIGH (1) , peneliti dapat memulai pengambilan data sesuai dengan nilai tersebut. Pentingnya sinyal *zero crossing* yaitu agar peneliti dapat mengambil data dimulai dari 0 sampai nilai yang diharapkan.

Fungsi selanjutnya yaitu *void loop* () yang digunakan untuk menuliskan *source code* berupa perintah – perintah dan persamaan untuk memproses data. Perintah yang dimasukkan dimulai dari memindahkan pin analog ADC (OC) ke variabel *baca[i]* dimana pin ADC akan membaca data dan menyimpan nya pada variabel *baca[i]*. Selanjutnya memindahkan pin digital (ZC) dengan variabel *mulai* yang bertujuan untuk membaca data dari pin digital kemudian disimpan pada variabel *mulai*. Setelah itu dilanjutkan dengan perintah fungsi perhitungan data seperti yang terlihat pada *flowchart* pada Gambar 3.14. Pada *flowchat* terdapat beberapa persamaan untuk memproses data, dan proses tersebut akan dibahas di bab selanjutnya.

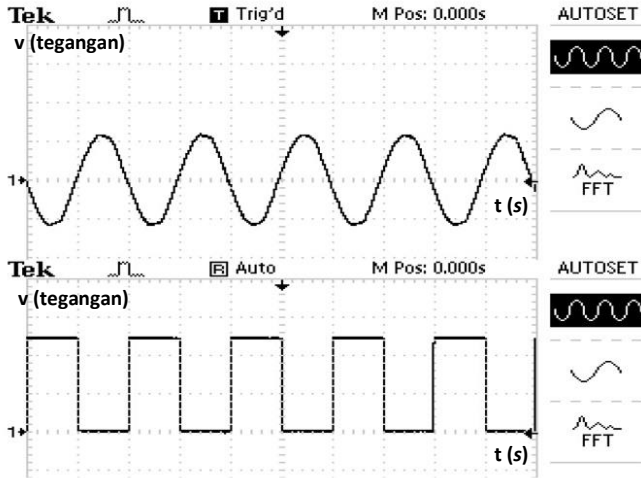
Didalam program terdapat fungsi *fft* yang digunakan untuk mengubah data tegangan ke *domain* frekuensi. Peneliti menggunakan *library* *fft.h* pada arduino untuk mengubah data ke *domain* frekuensi, yang bertujuan untuk mendapatkan *index* harmonik. Proses perhitungan *fft* dilakukan pada fungsi *fft_input*, dimana peneliti dimudahkan dengan hanya mengubah nilai dari fungsi tersebut dan hasil perhitungan *fft* disimpan pada *fft_log_out* seperti yang pada *flowchart* diatas. Setelah *index* harmonik didapatkan data diproses dengan persamaan THD untuk diukur distorsinya, pada bab selanjutnya proses perhitungan THD akan dibahas.

3.3.5 Proses Pengambilan data

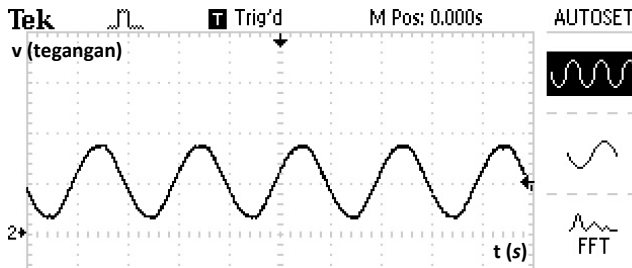
A. Pengukuran dengan *Input* Sinyal Tegangan AC PLN

Pengukuran kali ini digunakan rangkaian *zero crossing*, rangkaian *offset*, multimeter, osiloskop, dan mikrokontroler arduino mega. Sebelum melakukan pengukuran rangkaian *zero crossing* diberikan *supply* tegangan 5V DC, tegangan *pull up* 5V DC dan diberi sinyal *input* tegangan AC sebesar 4V. Rangkaian *offset* juga diberi *supply* tegangan 5V DC, tegangan *offset* 2V DC, dan sinyal *input* tegangan 4V AC. Sinyal yang digunakan peneliti berasal dari tegangan PLN 220V AC yang diturunkan dengan menggunakan *transformator non CT* sehingga menjadi 4V AC. Sinyal tersebutlah yang nantinya akan digunakan sebagai *input* untuk mengukur *total harmonic distortion* dari tegangan AC PLN. Langkah awal peneliti mengukur *output* dari rangkaian *zero crossing* didapatkan nilai tegangan sebesar 2,51 V DC, kemudian digunakan osiloskop untuk melihat bentuk sinyal awal dan sinyal keluaran dari rangkaian *zero crossing*. Seperti yang terlihat pada Gambar 3.28 sinyal PLN berbentuk sinus dan sinyal keluaran berbentuk kotak. Nilai V_{rms} yang didapatkan yaitu untuk sinyal PLN sebesar 4,18V dan *zero crossing* sebesar 3,49V. Tujuan peneliti menggunakan rangkaian *zero crossing* yaitu untuk mengubah sinyal sinus menjadi kotak, dapat diartikan bahwa *output* dari rangkaian telah benar. Peneliti sengaja membuat *output* dari rangkaian *zero crossing* agar memiliki tegangan dibawah 5V DC dikarenakan pada mikrokontroler arduino mega memberikan batasan *input* tidak lebih dari 5V DC.

Setelah itu peneliti mengukur *output* rangkaian *offset*, didapatkan nilai tegangan sebesar 2,03V DC dengan bentuk sinyal sinus seperti yang terlihat pada Gambar 3.29. Nilai tersebut didapat melalui pengukuran menggunakan multimeter. Setelah diketahui bentuk sinyal dengan menggunakan osiloskop sinyal yang diperoleh sesuai dengan yang diharapkan dengan nilai V_{rms} 2,26V. Peneliti sengaja membuat nilai *output* rangkaian *offset* tidak mempunyai polaritas negatif (-) dengan nilai tegangan sebesar 2V dikarenakan nantinya *output* tersebut di *inputkan* ke mikrokontroler arduino mega.



Gambar 3.28 Bentuk sinyal PLN (sinus) dan bentuk sinyal *zero crossing* (kotak)



Gambar 3.29 Bentuk sinyal rangkaian *offset*

Output dari rangkaian *offset* dimasukkan pada pin analog dan *output zero crossing* dimasukkan pada pin digital mikrokontroler arduino mega. Sinyal *output* rangkaian *offset* merupakan sinyal analog yang nantinya akan diubah menjadi data digital oleh ADC pada pin analog mikrokontroler arduino mega yang mempunyai resolusi sebesar 10bit dengan nilai maksimum 1023 dan nilai minimum 0, yang berarti mampu memetakan data hingga 1024 *data discrete analog level*. Sedangkan sinyal *output* dari rangkaian *zero crossing* merupakan sinyal digital yang mempunyai nilai 0 (*low*) dan 1 (*high*). Fungsi dari *pull up* resistor pada rangkaian *zero crossing* yaitu untuk mencegah *output* dari

keadaan *float state* (mengambang). Bentuk sinyal *output* rangkaian *zero crossing* yaitu sinyal kotak seperti pada Gambar 3.28 dimana hasil sinyal tersebut sesuai dengan tujuan awal dibuat. Fungsi dari *zero crossing* yaitu untuk mendeteksi titik persilangan nol pada suatu gelombang listrik. Dimana titik nol tersebut dijadikan acuan untuk mengatur mulai nya pengambilan data pada mikrokontroler.

| No | Source Code |
|----|--------------------------------------|
| 1 | #define LOG_OUT 1 |
| 2 | #define FFT_N 256 |
| 3 | #include <FFT.h> |
| 4 | int baca[263]; |
| 5 | int OC = A0 ; |
| 6 | int ZC = 3 ; |
| 7 | int B [130] ; |
| 8 | float C[130]; |
| 9 | float D ; |
| 10 | float E [263] ; |
| 11 | float F ; |
| 12 | float G ; |
| 13 | float H ; |
| 14 | float K ; |
| 15 | float X[263] ; |
| 16 | float V ; |
| 17 | float W; |
| 18 | float Y; |
| 19 | float Z[263] ; |
| 20 | float THD; |
| 21 | |
| 22 | void setup() { |
| 23 | Serial.begin (115200); |
| 24 | pinMode (OC, INPUT); |
| 25 | pinMode (ZC, INPUT); |
| 26 | while (digitalRead(ZC) == HIGH) {}; |
| 27 | while (digitalRead(ZC) == LOW) {}; |
| 28 | while (digitalRead(ZC) == HIGH) {}; |
| 29 | Serial.println ("*****START*****"); |
| 30 | Serial.println ("\n"); |
| 31 | } |
| 32 | void loop() { |
| 33 | int mulai = digitalRead(ZC); |
| 34 | Serial.print ("Dimulai saat ZC = "); |
| 35 | Serial.println(mulai); |
| 36 | Serial.print ("\n"); |
| 37 | while (digitalRead(ZC) == HIGH) {}; |
| 38 | while (digitalRead(ZC) == LOW) {}; |
| 39 | for (int i = 0 ; i <= 262 ; i++) { |
| 40 | baca[i] = analogRead(OC); |
| 41 | delayMicroseconds(45); |
| 42 | } |
| 43 | Serial.print ("\n"); |
| 44 | Serial.print (".HASIL X= "); |

```

45     Serial.print("\n");
46     F = 0;
47     for (int i = 0 ; i <= 262 ; i++) {
48         X[i] = baca[i] - 414 ;
49         E[i] = pow (X[i], 2);
50         F += E[i];
51         Serial.println(X[i]);
52     }
53     Serial.print("\n");
54     Serial.print (" .HASIL Z= ");
55     Serial.print ("\n");
56     for (int i = 0 ; i <= 262 ; i++) {
57         G = F / 263 ;
58         H = sqrt(G) ;
59         K = 226 / H ;
60         Z[i] = K * X[i] ;
61         fft_input[2*i] = Z[i] ;
62         fft_input[2*i + 1] = 0 ;
63         Serial.println(Z[i]);
64     }
65     Serial.print ("\n");
66     Serial.print ("HASIL PENJUMLAHAN (E) = ");
67     Serial.println(F);
68     Serial.print ("HASIL RATA-RATA = ");
69     Serial.println(G);
70     Serial.print ("HASIL AKAR (G) = ");
71     Serial.println(H);
72     Serial.print ("KONSTANTA = ");
73     Serial.println(K);
74     Serial.print ("\n");
75     Serial.print (" .HASIL FFT = ");
76     Serial.print ("\n");
77     fft_reorder ();
78     fft_run();
79     fft_mag_log();
80     for (int i = 0; i <= 127 ; i++) {
81         Serial.println(fft_log_out[i]);
82         Y = fft_log_out[2] ;
83     }
84     Serial.print ("\n");
85     Serial.print (" .HASIL PANGKAT = ");
86     Serial.print ("\n");
87     D = 0;
88     for (int i = 0; i <= 127 ; i++) {
89         C[i] = pow (fft_log_out[i], 2);
90         D = C[2] + C[4] + C[6] + C[8] + C[10] + C[12] + C[14] + C[16] + C[18] +
100 C[19] + C[21] + C[23] + C[25] + C[27] + C[29] + C[31] + C[33] + C[35] + C[37] +
101 C[39] + C[41] + C[43] + C[45] + C[47] + C[49] + C[51] + C[53] + C[55] + C[57] +
102 C[58] + C[61] + C[63] + C[64] + C[66] + C[68] + C[70] + C[72] + C[74] + C[76] +
103 C[78] + C[80] + C[82] + C[84] + C[86] + C[88] + C[90] + C[92] + C[93] + C[96] +
104 C[97] + C[100] + C[102] + C[103] + C[106] + C[107] + C[109] + C[111] + C[113]
105 + C[115] + C[117] + C[119] + C[121] + C[123] + C[125] + C[127];
106         Serial.println(C[i]);
107     }

```

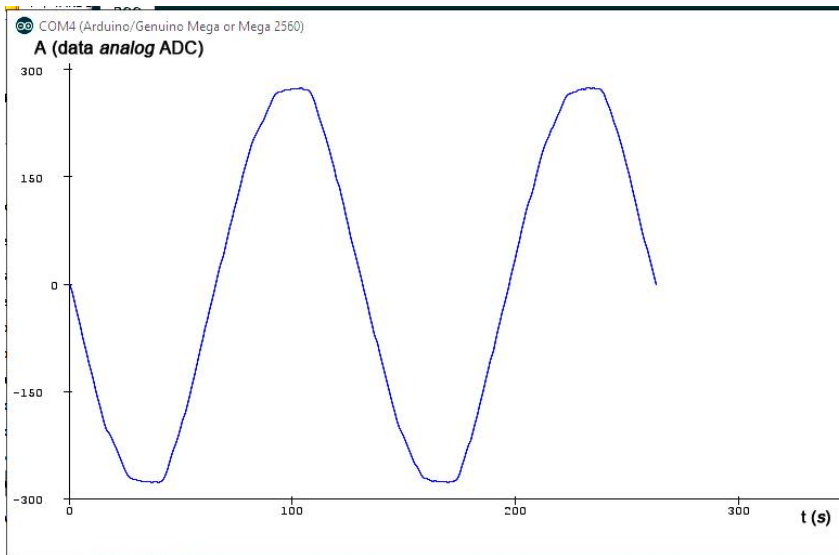
| | |
|-----|--|
| 108 | Y = fft_log_out[2]; |
| 109 | W = sqrt(D); |
| 110 | THD = (W / Y) * 100; |
| 111 | Serial.print ("\n"); |
| 112 | Serial.print ("*** HASIL PENJUMLAHAN = "); |
| 113 | Serial.println(D); |
| 114 | Serial.print ("*** HASIL AKAR = "); |
| 115 | Serial.println(W); |
| 116 | Serial.print ("*** FUNDAMENTAL = "); |
| 117 | Serial.println(Y); |
| 118 | Serial.print ("*** HASIL THD = "); |
| 119 | Serial.println(THD); |
| 120 | Serial.print ("\n"); |
| 121 | delay(1000); |
| 122 | } |

Gambar 3.30 Kode program

Setelah kedua sinyal *input* terhubung dengan mikrokontroler arduino mega, kemudian diproses dengan kode program seperti yang terlihat pada Gambar 3.30 yang merupakan bentuk kode program untuk memproses *output* dari rangkaian *offset* dan *zero crossing*. Didalam kode program terdapat beberapa fungsi untuk memproses data. Untuk memproses sinyal *input* rangkaian *zero crossing* dituliskan program seperti pada Gambar 3.30 yang diblok warna kuning. Pada blok warna kuning dituliskan program untuk membaca *input* sinyal *digital* pada rangkaian *zero crossing*, dituliskan juga perintah *while* HIGH dan LOW. Perintah tersebut dituliskan dengan tujuan jika sinyal *digital* terbaca HIGH/LOW maka proses pencuplikan data akan dilakukan. Setelah itu dilanjutkan memproses sinyal *analog* dari *input* rangkaian *offset*. Peneliti mencuplik data sepanjang 2 periode saja, maka seperti pada Gambar 3.30 blok warna abu-abu dituliskan kode program pembacaan sinyal *analog* dan pencuplikan data sebanyak 263 data dalam 2 periode digunakan fungsi *for* dengan *delayMicroseconds(45)*. Data yang didapatkan berupa sinyal *analog* yang belum di *offset*kan. Kemudian agar data tersebut dimulai dari nol dan mempunyai 2 polaritas positif dan negatif maka diberikan nilai *offset* sebesar 414. Untuk mendapatkan nilai *offset* dilakukan dengan cara memutuskan sinyal *input* pada rangkaian *offset* dan menggabungkan *ground* dengan (+) sehingga didapatkan nilai *offset* tersebut. Bentuk sinyal yang didapatkan seperti yang terlihat pada Gambar 3.31 pada sumbu (A) merupakan nilai data *analog* ADC dan sumbu (t) merupakan waktu .

Sinyal yang didapatkan masih berupa bentuk nilai *analog* ADC, untuk merubah sinyal tersebut menjadi tegangan maka digunakan persamaan ($K \times$ Nilai *analog* ADC) seperti yang dituliskan pada Gambar 3.30 blok warna coklat. (K) merupakan konstanta, dimana untuk mendapatkan konstanta dibutuhkan nilai V_{rms} tegangan PLN dibagi dengan V_{rms} nilai *analog* ADC. Untuk mendapatkan V_{rms} tegangan PLN peneliti cukup mengukur tegangan PLN dengan multimeter. Untuk mencari V_{rms} nilai *analog* ADC digunakan persamaan 3.1. Simbol (A) yang tertulis pada persamaan merupakan nilai data *analog* ADC.

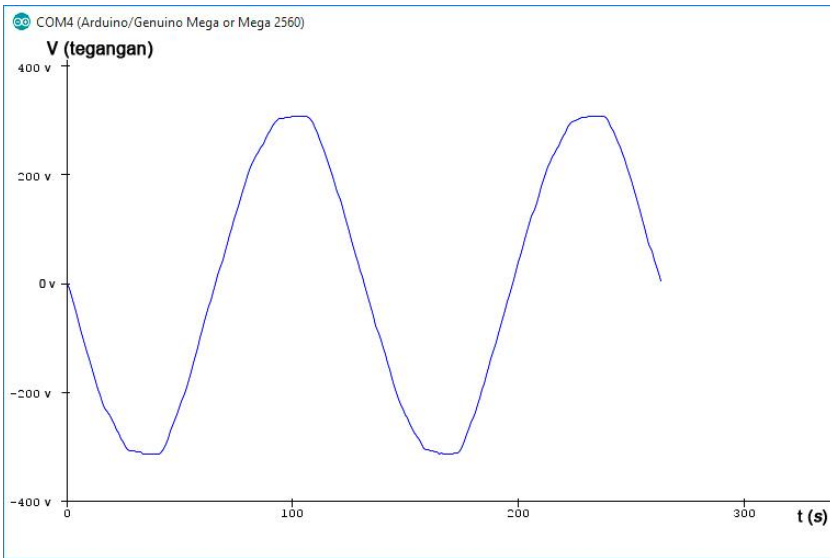
$$\sqrt{\frac{1}{n} \sum_{i=0}^{n-1} A_i^2} \quad (3.1)$$



Gambar 3.31 Bentuk sinyal data *analog* ADC

Setelah didapatkan nilai konstanta selanjutnya nilai tersebut dikalikan dengan data *analog* ADC sehingga didapatkan sinyal dalam bentuk tegangan seperti yang terlihat pada Gambar 3.32 pada sumbu (V) merupakan nilai tegangan dan sumbu (t) merupakan waktu. Gambar tersebut menunjukkan nilai tegangan hasil ukur dari sumber PLN

dengan nilai tegangan maksimum ± 300 V, nilai tersebut dapat berubah – ubah sesuai dengan kondisi pemakaian listrik PLN.

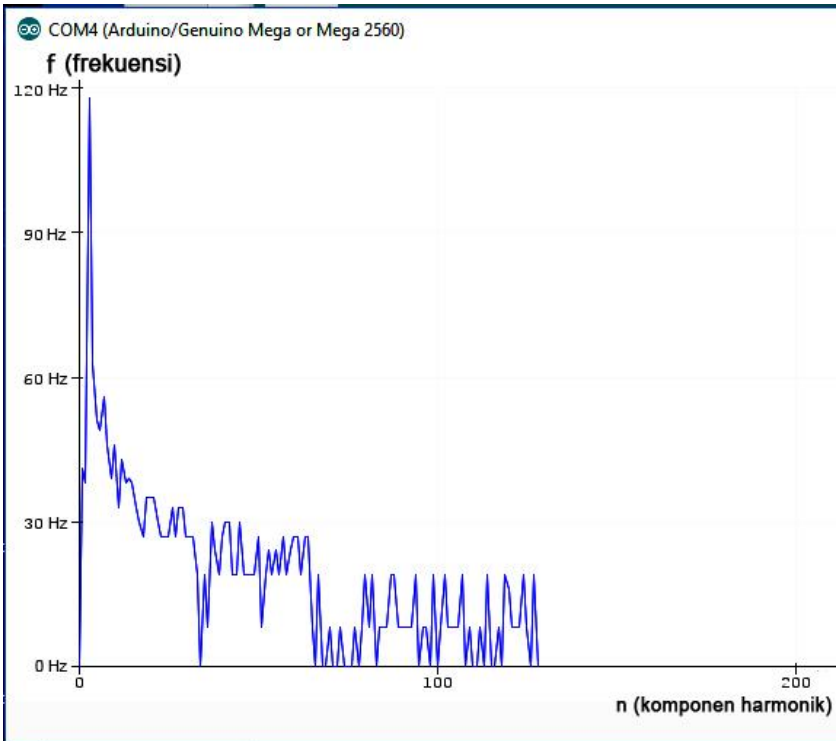


Gambar 3.32 Bentuk sinyal tegangan

Nilai tegangan yang didapatkan merupakan hasil rekam sinyal yang didapatkan dari sumber PLN. Setelah didapatkan data nilai tegangan, kemudian data tersebut di masukkan dalam program fft seperti yang terlihat pada Gambar 3.30 blok warna biru. Didalam blok biru dituliskan kode program untuk mengubah dari domain waktu ke domain frekuensi menggunakan fungsi fft pada *library* arduino IDE. Data tegangan yang didapatkan diinputkan pada fungsi fft sehingga diperoleh nilai *index* frekuensi dari data tegangan. Gambar 3.33 menunjukkan bentuk dari *plot* frekuensi yang didapatkan pada sumbu (f) merupakan frekuensi dan sumbu (n) merupakan komponen data harmonik, dimana nilai tertinggi menunjukkan nilai tertinggi dari data tegangan yang didapatkan dari sumber PLN. Nilai tertinggi dari *plot* frekuensi merupakan nilai *fundamental* dari frekuensi tersebut yaitu bernilai 118, pada *library* fft.h arduino IDE tidak dapat menampilkan data dalam bentuk *float* dan hanya dapat menampilkan data dalam bentuk *interger*. Data tersebut kemudian dijadikan *input* untuk mengetahui nilai distorsi yang terjadi pada tegangan sumber listrik PLN

seperti pada Gambar 3.30 blok warna hijau, dengan menggunakan persamaan *total harmonic distortion* (THD) untuk tegangan seperti pada persamaan 2.3.

Untuk mengetahui data mana yang akan dijumlahkan seperti pada persamaan THD untuk tegangan, dilakukan pengujian menggunakan *signal generator* dengan tujuan untuk mengetahui *index harmonic* yang akan dijumlahkan. Dengan cara mengubah nilai frekuensi pada *signal generator* dari rentang 50Hz sampai dengan 3300Hz. Setelah diketahui nilai *index harmonic* nya kemudian data dilakukan perhitungan sesuai dengan urutan *index harmonic* yang didapatkan dan dibagi dengan nilai frekuensi *fundamental* dari sumber listrik PLN yang didapatkan sebelumnya. Sehingga didapatkan nilai *total harmonic distortion* untuk tegangan dalam bentuk persen.



Gambar 3.33 Bentuk *plot* frekuensi

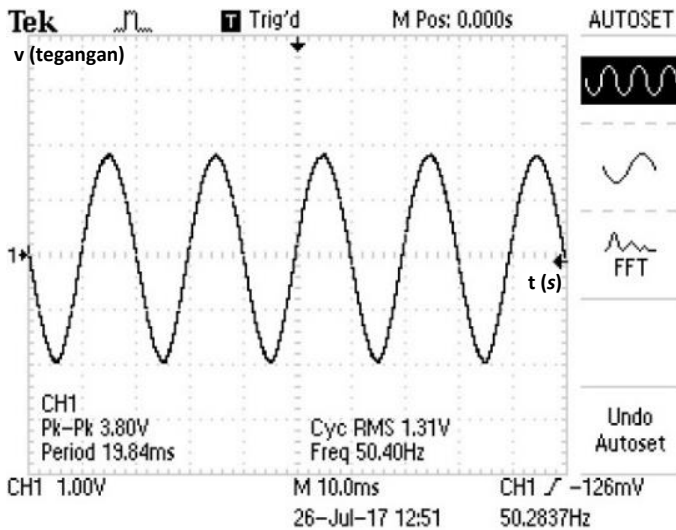
B. Pengukuran dengan Input Signal Generator

Pengukuran *selanjutnya* dilakukan dengan menggunakan *input* dari *signal generator*. Dengan tujuan untuk mengetahui nilai *index harmonic* dari frekuensi. Nilai *index harmonic* dibutuhkan untuk mengetahui urutan dari proses perhitungan THD untuk tegangan. Peneliti menggunakan *signal generator* analog Griffin & George LTD seri 644 milik laboratorium Instrumentasi Jurusan Fisika seperti pada Gambar 3.34.

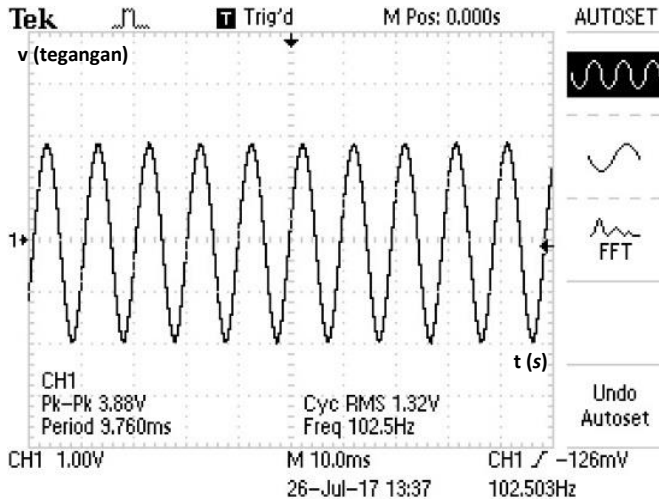


Gambar 3.34 *Signal generator* analog

Peneliti mengambil data *index harmonic* dari frekuensi 50Hz – 3300Hz dengan bentuk gelombang sinusoidal dengan nilai V_{rms} dari sumber PLN. Peneliti cukup memutar kontrol frekuensi untuk merubah nilai frekuensi tersebut. Kendala yang dialami peneliti yaitu kesulitan dalam memutar kontrol frekuensi agar nilai frekuensi sesuai dengan yang diharapkan. Dari hasil yang didapatkan, dari total data *plot* frekuensi yaitu 128 data didapatkan 66 nilai *index harmonic*. Nilai *index harmonic* inilah yang menjadi acuan peneliti untuk melakukan perhitungan sesuai dengan persamaan THD untuk tegangan. Setelah itu peneliti melakukan uji mengukur *total harmonic distortion* tegangan dengan menggunakan *input* dari sumber *signal generator*. Pengujian dilakukan dengan memvariasi nilai frekuensi dan bentuk gelombang. Pertama peneliti mengeset nilai frekuensi pada posisi 50Hz, 100Hz, dan 150Hz dengan bentuk gelombang sinusoidal. Nilai V_{rms} yang akan dituliskan pada program yaitu seperti yang Gambar 3.35.

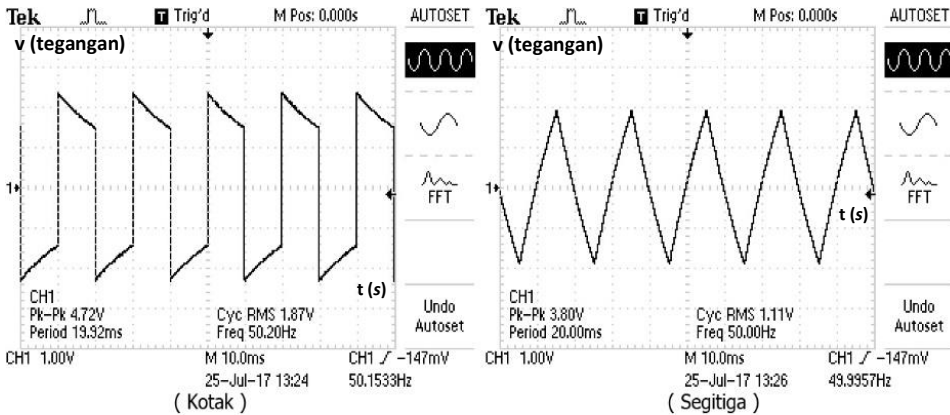


Gambar 3.35 Bentuk sinyal dari *signal generator* dengan frekuensi 50Hz



Gambar 3.36 Bentuk sinyal dari *signal generator* yang di *seting* 100hz

Pada gambar 3.35 menunjukkan bentuk sinyal dengan frekuensi 50Hz. Nilai frekuensi *fundamental* yang didapatkan dari sinyal *input* 50Hz dengan bentuk sinusoidal dari *signal generator*, berada pada urutan *index harmonic* ke-2 dari jumlah data sebanyak 128. Ketika frekuensi dinaikkan menjadi 100 Hz maka bentuk gelombang akan menjadi lebih rapat seperti yang terlihat pada Gambar 3.36. Dapat diartikan bahwa semakin besar frekuensinya maka dalam kurun waktu 1 detik akan terbentuk banyak gelombang(getaran). Setelah didapatkan nilai frekuensinya yang diproses dengan kode program arduino IDE. Maka dilakukan perhitungan menggunakan persamaan THD untuk tegangan seperti pada percobaan sebelumnya, sehingga didapatkan nilai *total harmonic distortion* untuk tegangan yang berasal dari *signal generator*. Selanjutnya peneliti mengubah sinyal *input* dengan bentuk gelombang menjadi segitiga dan kotak pada frekuensi 50Hz dari *signal generator*. Setelah diubah didapatkan bentuk sinyal seperti pada Gambar 3.37 yang merupakan bentuk gelombang segitiga dan kotak dengan frekuensi *fundamental* sama seperti bentuk gelombang sinusoidal.



Gambar 3.37 Bentuk sinyal kotak dan segitiga dari *input signal generator*