

PERANCANGAN DAN IMPLEMENTASI TUNER GITAR OTOMATIS DENGAN PENGGERAK MOTOR SERVO BERBASIS ARDUINO

DESIGN AND IMPLEMENTATION OF AUTOMATIC GUITAR TUNER WITH SERVO MOTOR BASED ON ARDUINO

Randi Yusuf Nasution¹, Hasanah Putri, ST., MT.², Yuli Sun Hariyani, ST., MT.³

^{1,2,3}Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu

Terapan, Universitas Telkom

¹randiotin19@gmail.com, ²hasanahputri@tass.telkomuniversity.ac.id,

³yulisun@telkomuniversity.ac.id

Abstrak

Gitar merupakan salah satu alat musik yang paling digemari, dan dapat dimainkan oleh semua orang. Saat ini masih banyak ditemukan pengguna gitar yang tidak bisa melakukan *tuning* senar gitar dan jika dilakukan manual juga membutuhkan waktu yang lama untuk mendapatkan hasil *tuning* yang akurat. Hal ini dapat diatasi dengan perkembangan teknologi. Pengguna gitar sekarang ini dapat melakukan *tuning* senar lebih mudah dan lebih cepat dengan tingkat toleransi kesalahan frekuensi *tuning* ± 1 Hz dengan menggunakan *tuner* gitar otomatis.

Perancangan alat *tuner* gitar otomatis ini menggunakan 5 komponen pembentuk alat yaitu *selector switch* untuk memilih frekuensi yang akan diatur, *Op amp* untuk menguatkan amplitudo gelombang suara agar diterima arduino dengan baik, Arduino untuk menginisialisasi frekuensi dan memberi sinyal ke motor servo, Motor servo untuk melakukan *tuning* senar gitar, dan LED sebagai indikator *tuning*.

Hasil dari penelitian ini adalah keakuratan *tuning* untuk setiap senar didapatkan rata-rata *tuning* paling baik yaitu senar 5 dengan rata-rata kesalahan *tuning* 0,07 Hz dan rata-rata *tuning* paling buruk yaitu senar 1 dengan rata-rata kesalahan *tuning* sebesar 0,8 Hz. Alat yang dapat membantu pengguna gitar melakukan *tuning* senar gitar secara otomatis dengan tingkat kesalahan maksimum ± 1 Hz dari standar frekuensi setiap nada senar gitar yang telah ditetapkan.

Kata kunci : arduino, motor servo, gitar listrik, op amp, Selector Switch

Abstract

Guitar is one of the most popular musical instrument, and can be played by everyone. Currently there are still many users who can't perform to tune guitar strings and if do it manually the results obtained are not very accurate. This can be overcome with the development of technology. Users can do this now guitar string tuning easily and accurately with the helping of an automatic guitar tuner

Automatic guitar tuner design uses 5 forming tool components are selector switch to select the frequency to be set, the op amp to amplify the amplitude of the sound wave that is received well arduino, Arduino to initialize the frequency and gives a signal to the servo motor, servo motor to perform tuning guitar strings, and LED as an indicator of tuning.

The results of this final project is an average tuning is most excellent strings 5 with an average error of 0.07 Hz tuning and the average worst tuning that string 1 with an average error of 0.8 tuning Hz. Automatic guitar tuner that can help users perform to tune guitar strings automatically with a maximum error rate is ± 1 Hz of each string standard frequency that has been set.

Keywords : arduino, motor servo, gitar listrik, op amp, Selector Switch

1. PENDAHULUAN

Gitar merupakan salah satu alat musik yang paling digemari, dan dapat dimainkan oleh semua orang. Akan tetapi masih banyak ditemukan orang yang hanya bisa memainkan gitar sementara dalam *tuning* nada senar gitar belum bisa. *Tuning* secara manual juga sering tidak akurat. Saat ini sudah ada beberapa alat *tuner* gitar yang sudah menggunakan teknologi dalam sistem kerjanya, contohnya *arduino guitar tuner*^[1] yang bekerja pada gitar listrik yang mentransmisikan sinyal listrik untuk menganalisis kode frekuensi yang dihasilkan senar. Alat ini dilengkapi dengan LED untuk menampilkan status senar yang sedang diatur. Akan tetapi untuk memutar *tuning peg* senarnya masih manual dengan tangan. Karena itulah dibuat Penelitian ini untuk membantu para pengguna gitar dalam *tuning* nada senar gitar secara otomatis sehingga menghasilkan nada yang sesuai dengan standar frekuensi nada senar gitar.

Alat *tuner* gitar otomatis ini bekerja pada gitar listrik yang mentransmisikan sinyal listrik untuk menganalisis kode frekuensi yang dihasilkan senar. Gitar dihubungkan ke alat *tuner* gitar otomatis dengan menggunakan kabel *audio*. Kode frekuensi untuk setiap senar diatur oleh *selector switch*. Rangkaian penguat menguatkan amplitudo gelombang suara hasil petikan agar dapat dibaca arduino dengan mudah. Arduino melakukan inisialisasi frekuensi senar dan kemudian mengirimkan perintah kepada motor servo. Motor servo akan memutar *tuning peg* gitar secara otomatis sesuai dengan informasi frekuensi dari arduino. Jika nada senar tidak selaras maka motor servo akan terus berputar sampai nada senar selaras. Jika nada sudah selaras maka motor servo secara otomatis berhenti.

Penelitian sebelumnya berjudul “Implementasi Kendali PID Dalam Penala Nada Gitar Otomatis” dengan luaran sistem kendali ini berupa arah dan kecepatan putar *motor stepper* pada sistem mekanik yang dihubungkan dan memutar knop senar gitar yang dipilih sampai nilai selisih tersebut hilang. Uji coba sistem dilakukan dengan melihat pengaruh penempatan sensor terhadap kinerja sistem dan uji coba terhadap karakteristik kendali PID. Hasil terbaik kinerja sensor dicapai dengan penempatan sensor menempel pada tabung gitar yang memiliki keakuratan 99,74%. Dari hasil uji coba sistem kendali diketahui bahwa nilai gain optimal pada sistem ini yaitu $K_p = 3$, $K_i = 1,5$, $K_d = 2,5$. Selain itu, sistem dalam penelitian ini memiliki kemampuan untuk menala senar gitar dalam waktu rata-rata 11,2 detik^[1]. Dalam Penelitian ini dibuat alat yang dapat membantu pengguna gitar melakukan *tuning* senar gitar secara otomatis dengan tingkat kesalahan maksimum ± 1 Hz dari standar frekuensi setiap nada senar gitar yang telah ditetapkan. Motor servo sebagai penggerak *tuning peg* menjadi pembeda alat yang dibuat dalam Penelitian ini dengan *arduino guitar tuner* yang masih manual dalam memutar *tuning peg*.

Dalam perancangan alat *tuner* gitar otomatis ini menggunakan 5 komponen pembentuk alat yaitu *selector switch* untuk memilih frekuensi yang akan diatur, *Op amp* untuk menguatkan amplitudo gelombang suara agar diterima arduino dengan baik, Arduino untuk menginisialisasi frekuensi dan memberi sinyal ke motor servo, Motor servo untuk melakukan *tuning* senar gitar, dan LED sebagai indikator *tuning*. Implementasi dari *tuner* gitar otomatis ini adalah motor servo akan bergerak melakukan *tuning* senar secara otomatis ketika ada petikan senar gitar. Motor servo akan memutar *tuning* untuk mencari posisi *tuning* senar yang benar yang dipilih terlebih dahulu melalui *selector switch*. Motor akan terus berputar jika nada belum selaras dan jika nada sudah selaras motor secara otomatis berhenti. Alat ini dihubungkan ke gitar menggunakan kabel *audio*.

2. METODOLOGI PERANCANGAN

2.1 Arduino Uno^[2]

Arduino Uno adalah *board* mikrokontroler berbasis ATmega328. Memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin

input analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *Board* Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya.

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal (otomatis). Eksternal (non-USB) daya dapat berasal baik dari AC-ke adaptor-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *plug jack* pusat-positif ukuran 2.1mm konektor POWER. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin *header* dari konektor POWER. Kisaran kebutuhan daya yang disarankan untuk *board* Uno adalah 7 sampai dengan 12 volt, jika diberi daya kurang dari 7 volt kemungkinan pin 5v Uno dapat beroperasi tetapi tidak stabil kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak board Uno.



Gambar 1. Arduino Uno

Masing-masing dari 14 pin digital di Uno dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode* (), *digitalWrite* (), dan *digitalRead* (), beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki *internal pull-up resistor* (secara *default* terputus) dari 20-50 kOhms.

2.2 Motor Servo^[3]

Motor servo adalah sebuah motor dengan sistem *closed feedback* dimana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian *gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada gambar dengan pulsa 1.5 ms pada periode selebar 2 ms maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa *OFF* maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa *OFF* maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.

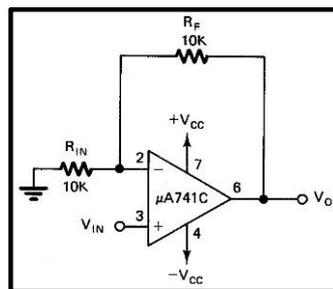
Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor *stepper*. Walau demikian, untuk beberapa keperluan tertentu, motor servo dapat dimodifikasi agar bergerak kontinyu. Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau bagian-bagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar. Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya.



Gambar 2. Motor Servo

2.3 Karakteristik Penguat Tidak Membalik (*Non-Inverting Amplifier*)^[4]

Penguat Tak-Membalik (*Non-Inverting Amplifier*) merupakan penguat sinyal dengan karakteristik dasar sinyal *output* yang dikuatkan memiliki fasa yang sama dengan sinyal *input*. Penguat tak-membalik (*non-inverting amplifier*) dapat dibangun menggunakan penguat operational, karena penguat operational memang dirancang untuk penguat sinyal baik membalik ataupun tak membalik. Rangkain penguat tak-membalik ini dapat digunakan untuk memperkuat isyarat AC maupun DC dengan keluaran yang tetap sefasa dengan sinyal masukan. Impedansi masukan dari rangkaian penguat tak-membalik (*non-inverting amplifier*) berharga sangat tinggi dengan nilai impedansi sekitar 100 MOhm. Contoh rangkaian dasar penguat tak-membalik menggunakan *operational amplifier* (Op-Amp) dapat dilihat pada gambar 4.



Gambar 4. Rangkaian *Non-Inverting Amplifier*

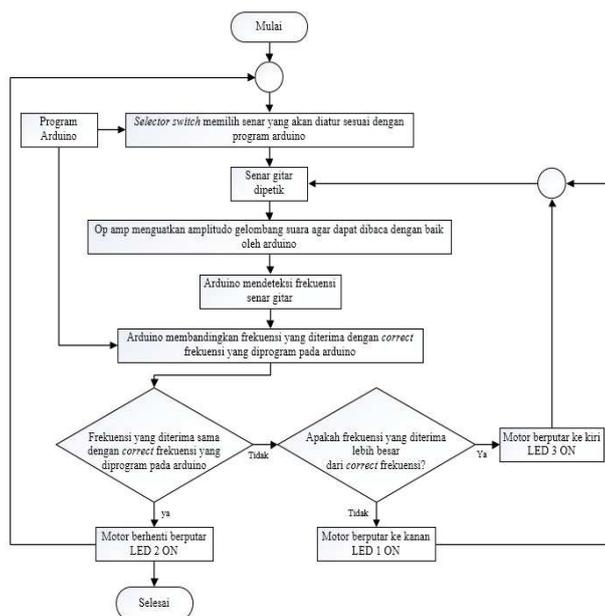
2.4 Operational Amplifier^[5]

Operational Amplifier atau di singkat op-amp merupakan salah satu komponen analog yang sering digunakan dalam berbagai aplikasi rangkaian elektronika. Aplikasi op-amp yang paling sering dipakai antara lain adalah rangkaian *inverter*, *non-inverter*, *integrator* dan *differensiator*. Pada Op-Amp memiliki 2 rangkaian *feedback* (umpan balik) yaitu *feedback* negatif dan *feedback* positif dimana *feedback* negatif pada op-amp memegang peranan penting. Secara umum, umpan balik positif akan menghasilkan osilasi sedangkan umpanbalik negatif menghasilkan penguatan yang dapat terukur.

3. PEMBAHASAN

3.1 Flowchart Cara Kerja Sistem

Flowchart perancangan sistem ini diawali dengan memilih program arduino sesuai dengan senar yang ingin diatur dengan mengatur *switch selector*. Kemudian senar dipetik, nada suara senar yang dipetik akan dikuatkan amplitudonya oleh *Op amp* sebelum diterima oleh arduino.



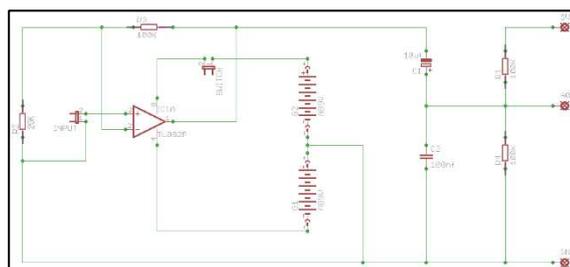
Gambar 5. flowchart Cara Kerja Sistem

Arduino akan mendeteksi frekuensi dari senar gitar yang diterima dan menyesuaikannya dengan frekuensi yang telah diatur pada program arduino. Apabila frekuensi yang diterima lebih rendah dari frekuensi yang telah diprogram pada arduino maka arduino akan memberikan perintah untuk menggerakkan motor servo ke kanan dan memberikan perintah untuk menghidupkan LED indikator 1. Apabila frekuensi yang diterima lebih besar dari frekuensi yang telah diprogram pada arduino maka arduino akan memberikan perintah untuk menggerakkan motor servo ke kiri dan memberikan perintah untuk menghidupkan LED indikator 3. Jika frekuensi belum sesuai terus lakukan petikan sampai didapatkan frekuensi yang diterima sama dengan frekuensi yang telah diprogram pada arduino. Apabila frekuensi yang diterima sama dengan frekuensi yang telah diprogram pada arduino maka motor akan berhenti berputar dan memberikan perintah untuk menghidupkan LED indikator 2 dan proses *tuning* selesai. Pilih *selector switch* untuk melakukan *tuning* senar lainnya.

3.2 Perancangan Hardware

a. Perancangan Rangkaian Op Amp

Pada pembuatan alat *tuner* gitar otomatis dengan penggerak motor servo ini dibutuhkan suatu rangkaian penguat agar arduino dapat membaca sinyal dengan baik sehingga didapatkan frekuensi yang akurat. Jenis rangkaian amplifier yang digunakan pada alat ini adalah rangkaian *op amp inverting amplifier* yang dapat dilihat pada gambar 6.



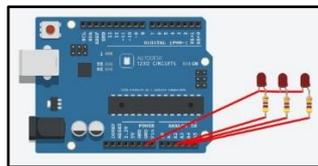
Gambar 6. Rangkaian Op amp Non-Inverting Amplifier

Dari gambar rangkaian *op amp Non-inverting amplifier* diatas dapat dilihat nilai resistansi *feedback* (R3) adalah 100k dan nilai resistansi *input* (R2) adalah 20k, sehingga kita bisa menghitung faktor penguatan rangkaian di atas yaitu :

$$A_v = \left(\frac{R_f}{R_{in}} \right) + 1 = \left(\frac{R_3}{R_2} \right) + 1 = \left(\frac{100}{20} \right) + 1 = 6 \text{ kali} \quad (1)$$

a. Perancangan Rangkaian LED Indikator

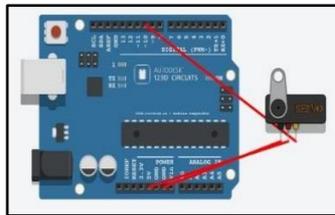
Alat *tuner* gitar otomatis dengan penggerak motor servo ini menggunakan 3 buah LED indikator 3mm yang berfungsi sebagai indikator perbandingan frekuensi yang diterima dengan frekuensi setiap senar yang telah diprogram pada arduino. Ketiga LED indikator ini terhubung ke arduino, dimana LED 1 terhubung ke *port* A2 arduino, LED 2 terhubung ke *port* A3 arduino, dan LED 3 terhubung ke *port* A4 arduino seperti terlihat pada gambar 7.



Gambar 7. Menghubungkan LED indikator ke Arduino

b. Perancangan Motor Servo

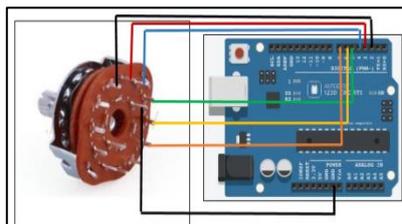
Motor servo sebagai penggerak *tuning peg* dirancang untuk mengencangkan atau mengendurkan *tuning peg* saat proses *tuning* berlangsung sesuai dengan perintah arduino yang membandingkan frekuensi yang diterima dengan frekuensi nada senar gitar yang telah diprogram pada arduino.



Gambar 8. Menghubungkan Servo ke Arduino

c. Perancangan *Selector Switch*

Selector switch pada alat ini digunakan untuk memilih atau memindahkan program arduino dari setiap senar gitar yang akan diatur.

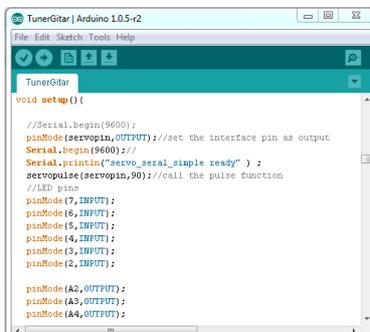


Gambar 9. Menghubungkan *Switch Selector* ke Arduino

3.3 Perancangan Software

a. Setup Input dan Output Arduino

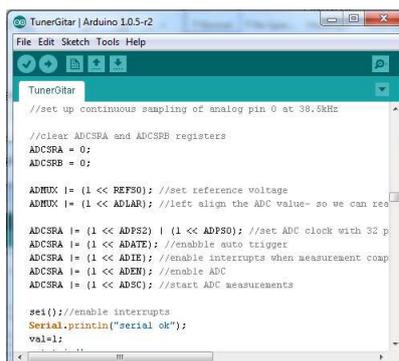
Pada pemrograman arduino ini dilakukan pendefinisian *port* arduino sebagai *input* dan *output*. Pendefinisian ini diprogram pada *void setup* arduino



Gambar 10. Setup Input dan Output Port Arduino

b. Setup ADC Arduino

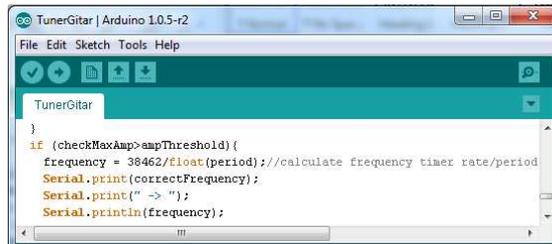
Pada dasarnya ADC arduino bernilai 10 bit (0-1023) tetapi pada program ini diatur agar bit 8 (0-255) menjadi bit tertinggi pada program ini untuk mengurangi sensitivitas lonjakan sinyal yang diterima arduino dan untuk mendapatkan frekuensi yang akurat. *Clock* ADC arduino bekerja pada frekuensi 16Mhz dengan 13 *clock cycles*, tetapi pada program ini diinginkan *sampling rate* yang mendekati standar *sampling rate audio* yaitu 40Khz. Dengan demikian harus memperkecil frekuensi *clock* ADC arduino dengan menggunakan *prescaler* untuk mengatur *clock* ADC arduino. *Prescaler* yang digunakan pada program ini yaitu *prescaler 32*, dengan *prescaler 32* nilai *clock* ADC yang awalnya bernilai 16MHz dibagi dengan 32 sehingga *clock* arduino menjadi 500KHz. Untuk mendapatkan nilai *sampling rate* nilai *clock* ADC dibagi dengan *clock cycles*, sehingga didapatkan nilai *sampling rate* mendekati nilai standar *sampling rate audio* yaitu $500\text{KHz}/13 = 38.461\text{ KHz}$. Untuk mendapatkan 1 *sample* membutuhkan waktu $13/500 = 0.026\text{ ms}$.



Gambar 11. Program Setup ADC Arduino

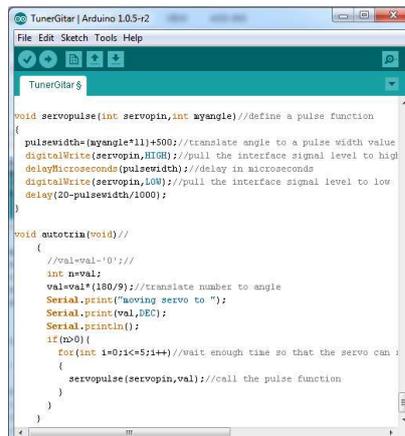
c. Program Perhitungan Frekuensi

Untuk mendapatkan nilai frekuensi yang diterima dari petikan senar gitar, arduino membagi *sampling rate* 38.461 KHz dengan periode.



Gambar 12. Program Perhitungan Frekuensi

d. Program Menggerakkan Motor Servo



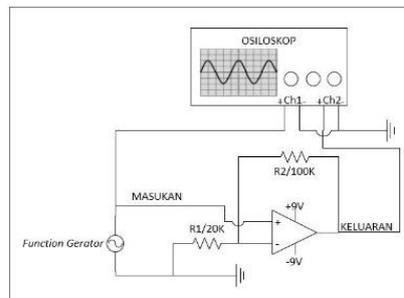
Gambar 13. Program Menggerakkan Servo

Servo dirancang untuk dapat berputar ke kiri dan ke kanan, servo berputar ke kiri saat frekuensi yang diterima lebih besar dari *correct* frekuensi dan LED indikator ON dan servo berputar ke kanan saat frekuensi yang diterima lebih rendah dari *correct* frekuensi dan LED 1 ON. Saat frekuensi yang diterima sama dengan *correct* frekuensi, maka servo berhenti berputar dan LED 2 ON.

4. HASIL DAN PEMBAHASAN

4.1 Pengujian Rangkaian Penguat Amplifier

Pada penguat *op amp* dirancang dengan penguat *non-inverting amplifier* dengan nilai penguatan sebesar 6 kali guna mendapatkan hasil pengujian yang akurat. Berikut adalah gambar prosedur pengukuran.



Gambar 14. Prosedur Pengukuran Rangkaian Penguat *Op Amp*

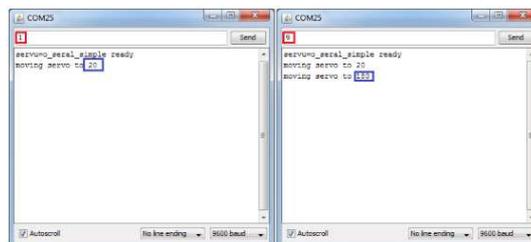
Dari pengukuran penguat berdasarkan gambar prosedur di atas dilakukan pengujian sebanyak 5 kali dengan hasil pengukuran seperti terlihat pada tabel 1. Hasil pengujian penguat *op amp* didapatkan rata-rata nilai penguatan *op amp* = 5,564 kali. Hasil pengujian tidak jauh berbeda dengan perancangan yaitu sebesar 6 kali.

Tabel 1. Hasil Pengujian Penguat *Op Amp*

Pengujian	V _{in} (mV)	V _{out} (mV)	Penguatan
1	108	616	5,7
2	232	1270	5,5
3	340	1860	5,5
4	460	2560	5,6
5	560	3140	5,6
6	638	3636	5,7
7	743	4309	5,8
8	840	4623	5,5
9	908	5084	5,6
10	1023	5527	5,4
11	1160	6612	5,7
12	1235	6916	5,6
13	1327	7564	5,7
14	1430	7865	5,5
15	1545	8807	5,7
16	1610	9016	5,6
17	1724	9482	5,5
18	1840	10304	5,6
19	1920	10944	5,7
20	2020	11514	5,7
Rata-rata penguatan			5,6

4.2 Pengujian Putaran Motor Servo

Pengujian ini dilakukan untuk memastikan motor dapat berputar ke kiri dan ke kanan. Pada perancangan alat ini motor dirancang untuk berputar ke kiri saat frekuensi yang diterima arduino lebih besar dari *correct* frekuensi yang diprogram pada arduino dan motor berputar ke kanan saat frekuensi yang diterima arduino lebih rendah dari *correct* frekuensi yang diprogram pada arduino. Pengujian dilakukan melalui serial monitor arduino dengan mengirim nilai val 1 dan val 9 yang telah diprogram pada arduino



Gambar 15. Pengujian Servo Berputar ke Kiri dan ke Kanan

Dari hasil pengujian motor servo melalui serial monitor arduino dapat dilihat servo berputar ke kiri saat diberikan nilai val 1 karena pada program untuk mendapatkan nilai pulsa

dilakukan perhitungan $val = val \times (180/9)$ untuk *translate number to angle* dan $pulsewidth = (angle \times 11) + 500$ sehingga :

1. Saat diberikan nilai val 1
 $Val = 1 \times (180/9) = 20$
 $Pulsewidth = (20 \times 11) + 500 = 720 \mu s = 0,72ms$
 Maka servo berputar ke kiri karena pulsa yang diberikan kurang dari 1,5 ms
2. Saat diberikan nilai val 9
 $Val = 9 \times (180/9) = 180$
 $Pulsewidth = (180 \times 11) + 500 = 2480 \mu s = 2,48ms$
 Maka servo berputar ke kanan karena pulsa yang diberikan lebih dari 1,5 ms

4.1 Pengujian Keakuratan *Tuning* dan Waktu yang Dibutuhkan

Pengujian ini dilakukan untuk mengetahui perbandingan antara frekuensi yang diterima arduino dengan *correct* frekuensi setiap senar yang telah diprogram pada arduino. Perbandingan ini akan menentukan tingkat keakuratan *tuning* alat. Alat ini dirancang dengan tingkat akurasi *tuning* 2Hz dari *correct* frekuensi. Pengujian ini dilakukan sebanyak 10 kali untuk setiap senar dengan kondisi awal 5 kali pengujian saat frekuensi senar lebih rendah dari *correct* frekuensi dan 5 kali pengujian saat frekuensi senar lebih besar dari *correct* frekuensi.

Tabel 2. Hasil Pengujian *Tuning* dan Lama Waktu *Tuning*

Pengujian	senar 1	senar 2	senar 3	senar 4	senar 5	senar 6
Rata-rata Frekuensi Saat <i>Tuning</i> Selesai (Hz)	328,74	246,55	195,735	146,245	109,739	82,187
Rata-rata Waktu <i>Tuning</i> Kondisi Awal FA < CF (detik)	38,722	13,558	15,186	9,752	28,734	7,464
Rata-rata Waktu <i>Tuning</i> Kondisi Awal FA > CF (detik)	20,348	20,48	20,48	11,982	8,042	12,644

Dari hasil pengujian pada tabel 2 diketahui bahwa hasil perbandingan antara frekuensi yang diterima arduino dengan *correct* frekuensi yang diprogram pada arduino sesuai dengan perancangan yang diharapkan yaitu dengan tingkat toleransi kesalahan frekuensi maksimal sebesar 2Hz dari *correct* frekuensi yang telah ditentukan. Tetapi dalam proses untuk mencapai perbandingan frekuensi yang diharapkan yaitu toleransi kesalahan frekuensi sebesar 2Hz masih ditemukan beberapa kali lonjakan frekuensi yang disebabkan oleh sensitivitas akibat bentuk sinyal gitar yang tidak beraturan atau banyak ditemukan *ripple* pada amplitudo sinyal hasil petikan gitar. Lonjakan frekuensi juga disebabkan karena beberapa faktor seperti senar lain ikut bergetar saat *tuning* senar, bentuk sinyal analog suara petikan senar gitar yang tidak beraturan, dan tidak adanya rangkaian filter pada alat ini. Cara untuk meminimalkan lonjakan frekuensi tersebut yaitu dengan teknik dalam pemetikan senar. Senar dipetik berkali-kali dengan cara setelah dipetik senar langsung ditahan agar berhenti (senar jangan dibiarkan terus bergetar).

Lama waktu yang dibutuhkan alat untuk melakukan *tuning* keseluruhan senar dapat dihitung dengan perhitungan :

Rata-rata Waktu *Tuning* Keseluruhan Senar Kondisi Awal FA < CF

$$\begin{aligned}
 &= \text{hasil penjumlahan dari hasil pengujian rata – rata waktu } tuning \text{ semua senar} \\
 &= 32,76+13,60+12,69+9,77+27,28+6,67 \\
 &= 102,77 \text{ detik} \\
 &= 1 \text{ menit } 42,77 \text{ detik}
 \end{aligned}$$

Rata-rata Waktu *Tuning* Keseluruhan Senar Kondisi Awal FA > CF

$$\begin{aligned}
 &= \text{hasil penjumlahan dari hasil pengujian rata – rata waktu } tuning \text{ semua senar}
 \end{aligned}$$

$$\begin{aligned} &= 15,58+20,11+17,90+12,46+7,86+13,03 \\ &= 86,94 \text{ detik} \\ &= 1 \text{ menit } 26,94 \text{ detik} \end{aligned}$$

Lama waktu yang dibutuhkan alat untuk melakukan *tuning* dipengaruhi oleh berapa besar beda frekuensi senar pada kondisi awal sebelum diatur dengan *correct* frekuensi yang diprogram pada arduino. Semakin besar beda frekuensi awal senar gitar saat awal *tuning* dengan *correct* frekuensi yang diprogram pada arduino maka akan semakin lama waktu *tuning* yang dibutuhkan sampai selesai melakukan *tuning*.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Setelah melakukan proses perancangan, pengukuran, dan pengujian maka diperoleh kesimpulan sebagai berikut.

1. Hasil pengujian penguat op amp didapatkan rata-rata nilai penguatan *op amp* = 5,6 kali. Dari hasil pengujian penguat *op amp* didapatkan persentase kesalahan sebesar 6,6%.
2. Hasil Pengujian keakuratan *tuning* yang dilakukan sebanyak 20 kali untuk setiap senar didapatkan rata-rata *tuning* paling baik yaitu senar 5 dengan rata-rata kesalahan *tuning* 0,07 Hz dan rata-rata *tuning* paling buruk yaitu senar 1 dengan rata-rata kesalahan *tuning* sebesar 0,8 Hz.
3. Hasil Pengujian keakuratan *tuning* yang dilakukan sesuai dengan perancangan yang diharapkan yaitu dengan toleransi kesalahan maksimum ± 1 Hz untuk setiap senar.
4. Hasil pengujian lama waktu *tuning* yang dibutuhkan setiap senar didapatkan rata-rata waktu *tuning* paling baik yaitu senar 6 dengan rata-rata waktu *tuning* 9,85 detik dan rata-rata *tuning* paling buruk yaitu senar 1 dengan rata-rata waktu *tuning* 24,17 detik.
5. Rata-rata waktu *tuning* keseluruhan senar dengan kondisi frekuensi awal < *correct* frekuensi adalah 1 menit 42,77 detik.
6. Rata-rata waktu *tuning* keseluruhan senar dengan kondisi frekuensi awal > *correct* frekuensi adalah 1 menit 26,94 detik.
7. Lama waktu yang dibutuhkan alat untuk melakukan *tuning* hingga didapatkan perbandingan frekuensi yang diharapkan dipengaruhi oleh besar beda frekuensi senar gitar pada kondisi awal *tuning* dengan *correct* frekuensi setiap senar pada program arduino.
8. Dalam proses *tuning* masih ditemukan lonjakan frekuensi saat senar gitar terus bergetar. Solusi untuk meminimalkan lonjakan frekuensi tersebut dengan cara memetik senar dan langsung menahan senar agar tidak bergetar terus.

5.2 Saran

Pengembangan yang dapat dilakukan dari Penelitian ini adalah sebagai berikut.

1. Ditambahkan rangkaian filter untuk menghilangkan lonjakan-lonjakan frekuensi untuk mempercepat proses *tuning* mendapatkan perbandingan frekuensi yang benar.
2. Ditambahkan LCD agar dapat melihat langsung perbandingan frekuensi pada saat proses *tuning* tanpa harus melalui serial monitor arduino.

Daftar Pustaka

- [1] Maulana, Reza. 2013. Implementasi Kendali PID Dalam Penala Nada Gitar Otomatis. Universitas Gadjah Mada
- [2] <http://www.instructables.com/id/Arduino-Guitar-Tuner/?ALLSTEPS> [accessed Maret 2013]
- [3] Artanto, Dian. 2012. *Interaksi Arduino dan LabView*. Jakarta: PT Elex Media. Komputindo.
- [4] Iswanto. 2012. *Aplikasi Motor Servo Dengan Mikrokontroler*. Yogyakarta : UMY

- [5] <http://elektronika-dasar.web.id/percobaan/penguat-tak-membalik-non-inverting-amplifier>
[accessed Maret 2013]
- [6] ISO 16:1975 Acoustics -- *Standard tuning frequency (Standard musical pitch)*. International Organization for Standardization. 1975.