

## PERANCANGAN DAN IMPLEMENTASI EFEK GITAR *LOOPER* BERBASIS MIKROKONTROLER

### DESIGN AND IMPLEMENTATION OF LOOPER GUITAR EFFECT BASED ON MICROCONTROLLER

Dami Mahardiwana

Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom  
[damimusisi@gmail.com](mailto:damimusisi@gmail.com)

---

#### Abstrak

Gitar adalah alat musik yang sangat populer di seluruh dunia. Dapat kita temui berbagai kalangan mengenal dan memainkan alat musik ini. Untuk menghasilkan suara yang lebih menarik, dibuatlah efek gitar. Salah satu fungsi dari efek gitar adalah sebagai *looper*, dimana frase permainan gitar dapat disimpan dan diputarkan kembali secara berulang lalu digabungkan dengan frase permainan gitar lainnya sehingga menjadi permainan gitar yang harmonis.

Maka pada tugas akhir kali ini, dirancang sebuah efek gitar berbasis mikrokontroler yang didukung beberapa komponen, yang berfungsi sebagai pengontrol software *looper*, sehingga dapat menjalankan proses *looper* di dalam PC dalam bentuk sinyal digital. Kabel *jack* digunakan sebagai input yang berfungsi sebagai pengantar sinyal masuk dan *speaker* sebagai output yang berfungsi sebagai pengeluar suara. Proses perancangan berfokus pada pengontrolan sistem *looping* yang mampu menyimpan suara dan memutar ulang dengan baik, dalam hal ini adalah sinyal dari petikan dawai gitar, sehingga menghasilkan performansi sesuai efek gitar pada umumnya.

Dalam perancangan ini, telah terbuat alat yang mampu mengontrol sistem *looping*. Dengan sebuah *arduino* yang terhubung dengan sebuah *pedal shield*. Diharapkan dari implementasi sistem ini, akan menghasilkan efek gitar yang mampu berjalan secara *real-time* dengan *process time* dan *noise* yang kecil. Dari hasil pengujian didapat *process time* terkecil adalah 1 ms pada saat pengambilan *sample* sebanyak 64 *samples* setiap 1/11025 detik dan *process time* terbesar adalah 377 ms ketika pengambilan *sample* sebanyak 2048 *samples* setiap 1/192000 detik.

Kata Kunci : Gitar, *real-time*, *looping*, kabel *jack*, efek gitar

---

#### ABSTRACT

Guitar is a very popular music instrument throught the world. We can meet this tnstrument in every element of the society, they even know and play this instrument. To produce more attractive sound, invented a guitar effect. One of the functions of guitar effect is looper, it is a phrase where the guitar plays can be stored and played back repeatedly with another guitar plays to become a harmonic.

This final project designed a guitar effect based on microcontroller supported by several components which can be used as controller of looper software, that it can run as looper process in dicrete signal on a PC. Jack cable used as input to store input signal and speaker as an output. The design focusing on controlling looping system to store and played back well, in this case is the signal from strum of guitar strings. So it can poduce general guitar effect performance.

It has designed a device that is able to control looping system. Use an Arduino connected to pedal shield. From the implementation of the system, generated the guitar effect that is able to operate real time with good output with small process time and noise, as well. As the result, the smallest process time is 1 ms taking 64 samples every 1/11025 second, and the biggest process time is 377 ms taking 2048 samples every 1/192000 second. It is expected from the implementation of this system produces guitar effects that can run in real-time with a good sound output without process time and noise.  
Keywords: Guitar, real-time, looping, cable jack, amplifier, guitar effects

---

## 1. Pendahuluan

Musik saat ini telah mengalami pertumbuhan yang sangat pesat. Gejala ini dapat diamati dari banyaknya variasi jenis musik yang terus bermunculan baik dari media visual maupun audio. Dalam sejarah dan perkembangannya, seni musik Indonesia sangat dipengaruhi oleh seni musik Barat, sejak Indonesia mengalami penjajahan, masa kemerdekaan, bahkan sampai sekarang ini.

Hampir semua aliran musik menggunakan instrument gitar. Gitar sendiri memiliki pengertian sebagai cerminan dari jiwa. Dikatakan seperti itu karena pada saat memetik sebuah gitar, dentingan yang terdengar dari gitar dapat mengekspresikan emosi dari peminatnya. Gitar merupakan alat musik yang digemari oleh banyak kalangan, baik golongan tingkat bawah, tingkat menengah, tingkat atas, usia muda, maupun usia tua.

Teknologi mikroprocessor dan mikrokontroler saat ini memungkinkan manusia melakukan penelitian di bidang pengolahan sinyal. Banyak alat-alat yang telah dibuat berbasis mikrokontroler dengan tujuan mempermudah kerja manusia, salah satunya efek gitar. Efek gitar adalah alat untuk memanipulasi suara gitar. Dengan alat ini, manusia dapat memainkan gitar dengan keluaran suara sesuai keinginan. Salah satu variasi dari efek gitar adalah efek gitar looper. Alat ini menyimpan suara gitar yang manusia mainkan lalu dapat diputar kembali dan digabungkan dengan permainan gitar selanjutnya.

Pada tugas akhir ini akan dirancang efek gitar looper. Alat ini merupakan salah satu dari macam-macam jenis efek gitar. Efek gitar ini akan menyimpan frase-frase permainan gitar, kemudian menggabungkannya menjadi satu frase permainan gitar yang harmonis. Dimana frase tersebut juga dapat menjadi latar frase permainan gitar selanjutnya.

Efek gitar looper di dalam Tugas Akhir ini dirancang dengan menggunakan mikrokontroler. Untuk mengolah sinyal input yang berasal dari nada gitar, digunakan PC.

## 2. Dasar Teori

### 2.1 Pengolahan Sinyal Digital

Pada umumnya, istilah *signal processing* digunakan untuk menganalisa proses fisik *time-varying*. Pengolahan sinyal dibagi menjadi dua kategori yaitu pengolahan sinyal analog dan pengolahan sinyal digital (sinyal diskrit). Istilah analog digunakan untuk menggambarkan gelombang pada waktu yang kontinu dan memiliki *range* yang kontinu pada amplitudo. Sedangkan istilah sinyal waktu diskrit digunakan untuk menggambarkan sinyal yang memiliki variabel waktu bebas integer sebagai hasil kuantisasi.

Pengolahan Sinyal Digital berkaitan dengan representasi sinyal waktu diskrit dengan urutan angka atau simbol serta pengolahan dari sinyal-sinyal tersebut. Pengolahan sinyal digital dan pengolahan sinyal analog merupakan bagian dari pengolahan sinyal. Tujuan dari PSD umumnya adalah untuk mengukur, menyaring dan/atau mengkompres sinyal analog kontinu. Langkah pertama biasanya untuk mengkonversi sinyal dari analog ke bentuk digital, dengan mengambil sampel menggunakan konverter analog-ke-digital (ADC), yang mengubah sinyal analog menjadi aliran angka.

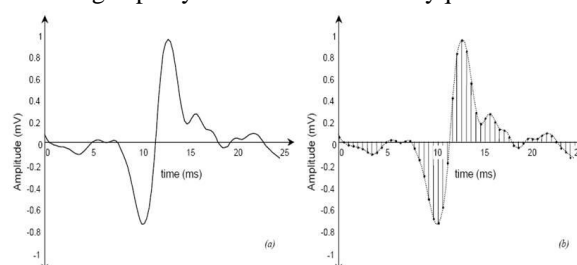


Gambar 1. Proses ADC

#### 2.1.1 Sampling

*Sampling* adalah proses pengambilan suatu data diskrit  $x(n)$  dari data kontinu dalam satu titik waktu tertentu dengan periode yang tetap. Teknik *sampling* yang digunakan berdasarkan pada teorema *sampling* Nyquist, di mana laju *sampling* minimal adalah dua kali besar frekuensi sinyal yang masuk. Laju *sampling*, atau *sampling rate*, atau frekuensi *sampling* ( $f_s$ ) mendefinisikan jumlah sampel per detik (atau per unit lainnya) diambil dari sinyal kontinu yang disampel menjadi sinyal diskrit. Semakin besar  $f_s$ , semakin banyak data diskrit yang didapat. Untuk sinyal waktu-domain, unit untuk *sampling rate* adalah hertz.

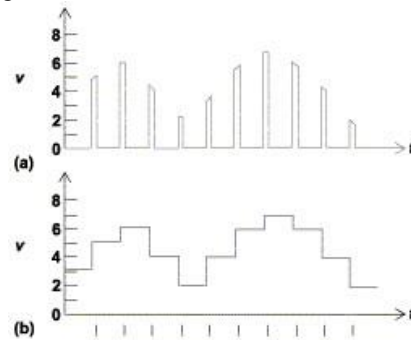
Untuk keperluan menangkap suara yang mencakup keseluruhan rentang pendengaran manusia (20 – 20.000 Hz) seperti saat merekam musik, laju *sampling* yang digunakan biasanya sebesar 44,1 kHz, 48 kHz, atau 96 kHz. Laju *sampling* ini sesuai dengan persyaratan dari teorema Nyquist.



Gambar 2. Proses *sampling*. (a) sinyal analog (b) hasil *sampling*

### 2.1.2 Kuantisasi

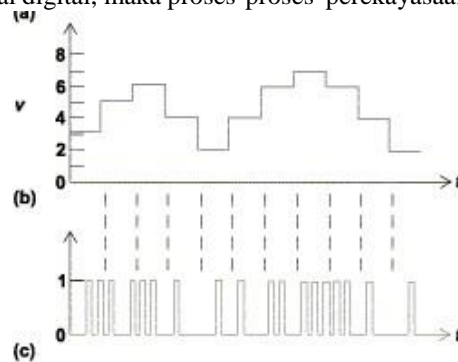
Sinyal digital merupakan sebuah deretan data diskrit (sampel) yang diwakili oleh beberapa digit dengan jumlah tertentu. Proses melakukan konversi sinyal yang telah dicuplik (sampel) menjadi sinyal digital yang diwakili oleh nilai dengan jumlah digit tertentu (level) disebut kuantisasi.



Gambar 3. Proses kuantisasi

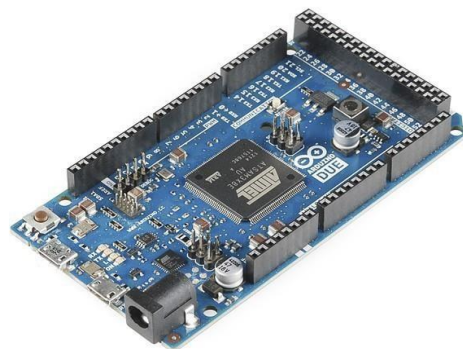
### 2.1.3 Encoding

Setelah melalui kuantisasi, proses selanjutnya ialah *encoding*. Level-level pada proses sebelumnya dapat dijadikan kombinasi bilangan biner, sehingga terbentuklah bilangan-bilangan biner yang merupakan informasi dari sinyal. Setelah menjadi sinyal digital, maka proses-proses perenkapsulasi dapat dilakukan.



Gambar 4. Proses *encoding*.

## 2.2 Arduino Due



Gambar 5. Arduino Due

Arduino Due adalah papan mikrokontroler berdasarkan CPU Atmel SAM3X8E ARM Cortex-M3. Due adalah papan Arduino pertama berdasarkan pada mikrokontroler ARM inti 32-bit. Ini memiliki 54 digital pin input / output (yang 12 dapat digunakan sebagai output PWM), 12 analog input, 4 UART (hardware port serial), clock 84 MHz, koneksi USB OTG, 2 DAC (digital ke analog), 2 TWI, jack listrik, header SPI, header JTAG, tombol reset dan tombol erase.

Papan berisi semua yang diperlukan untuk mendukung mikrokontroler; hanya perlu terhubung ke komputer dengan kabel micro-USB atau dengan adaptor AC-DC atau baterai untuk dipakai. Due kompatibel dengan semua perisai Arduino yang bekerja di 3.3V dan telah sesuai dengan 1.0 Arduino pinout.

Spesifikasi:

<i>Microcontroller</i>	<i>AT91SAM3X8E</i>
<i>Operating Voltage</i>	<i>3.3V</i>
<i>Input Voltage (recommended)</i>	<i>7-12V</i>
<i>Input Voltage (limits)</i>	<i>6-16V</i>
<i>Digital I/O Pins</i>	<i>54 (of which 12 provide PWM output)</i>
<i>Analog Input Pins</i>	<i>12</i>
<i>Analog Output Pins</i>	<i>2 (DAC)</i>
<i>Total DC Output Current on all I/O lines</i>	<i>130 mA</i>
<i>DC Current for 3.3V Pin</i>	<i>800 mA</i>
<i>DC Current for 5V Pin</i>	<i>800 mA</i>
<i>Flash Memory</i>	<i>512 KB all available for the user applications</i>
<i>SRAM</i>	<i>96 KB (two banks: 64KB and 32KB)</i>
<i>Clock Speed</i>	<i>84 MHz</i>
<i>Length</i>	<i>101.52 mm</i>
<i>Width</i>	<i>53.3 mm</i>
<i>Weight</i>	<i>36</i>

Keunggulan ARM Core

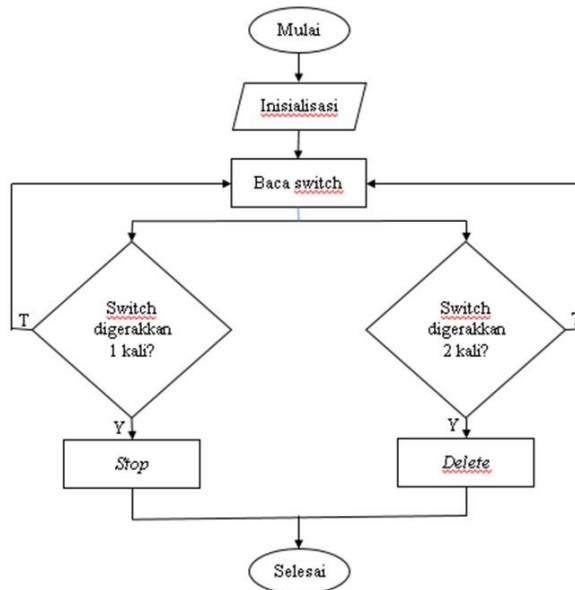
- 32-bit inti, yang memungkinkan operasi pada 4 byte data yang luas dalam clock CPU tunggal.
- clock CPU di 84Mhz.
- 96 KByte dari SRAM.
- 512 KByte memori Flash untuk kode.
- DMA controller, yang dapat meringankan CPU melakukan tugas-tugas memori **intensif**.



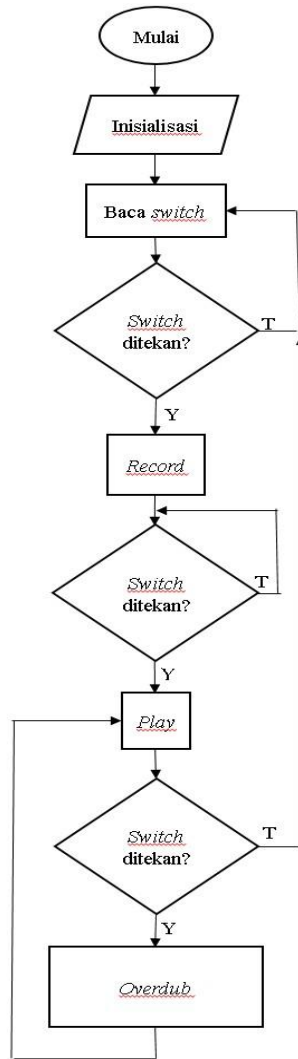
Gambar 6. Diagram blok penyimpanan suara



Gambar 7. Diagramblok pedal



Gambar 8. Diagram alir toogle switch



Gambar 9. Diagram alir *footswitch*

**3. Pembahasan**

**3.1 Pengujian Process time**

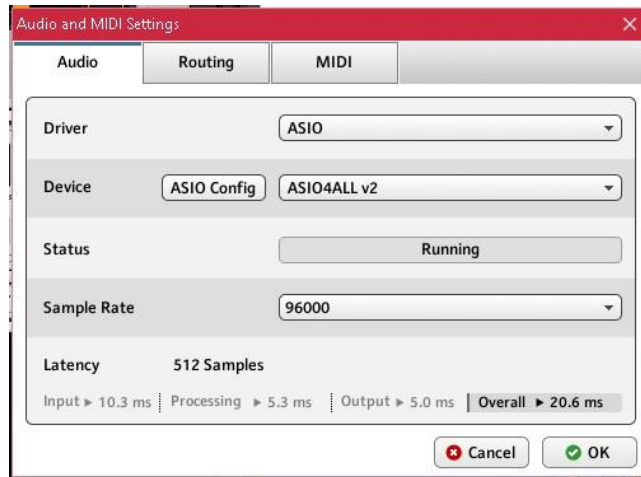
A. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui process time yang terjadi di setiap keadaan nilai sample dan sample rate yang berbeda

B. Skenario Pengujian



Gambar 10. Aplikasi ASIO4ALL



Gambar 9. Audio and MIDI Settings

Pegujian dilakukan melalui Audio and MIDI settings yang terdapat di aplikasi looper yang kita pakai , dapat dilihat tampilannya pada gambar 4.2, lalu kita pilih ASIO Config untuk mengatur nilai nilai sampelnya, setelah itu nilai nilai sample diatur di aplikasi ASIO4ALL di PC, dapat dilihat tampilannya pada gambar 4.1.

Setelah mengatur nilai sample, kita masukan nilai sample ratenya pada Audio and MIDI Settings lalu akan terlihat seberapa besar process time yang terjadi pada bagian bawah. Di pengujian ini, nilai sample di atur mulai dari 64-2048 samples dan sample rate dari 11025-192000 Hz.

C. Hasil Pengujian

Hasil dari pengujian process time tercantum pada tabel di bawah ini.

Nilai sample (samples)	Sample Rate (Hz)	Process time (ms)
64	11025	17,4
	22050	8,8
	24000	8
	44100	4,4
	48000	4,0
	88200	2,2
	96000	2
	176400	1,1
128	11025	29
	22050	14,6
	24000	13,3
	44100	7,2
	48000	6,7
	88200	3,7
	96000	3,3
	176400	1,8
192000	1,7	

256	11025	52,2
	22050	26,2
	24000	24
	44100	13
	48000	12
	88200	6,6
	96000	6
	176400	3,3
	192000	3
512	11025	98,6
	22050	49,4
	24000	45,3
	44100	24,6
	48000	22,7
	88200	12,4
	96000	11,3
	176400	6,2
	192000	5,7
1024	11025	192
	22050	95,8
	24000	88
	44100	47,8
	48000	44
	88200	24
	96000	22
	176400	12
	192000	11
2048	11025	377
	22050	189
	24000	173
	44100	94,3
	48000	86,7
	88200	47,2
	96000	43,3
	176400	23,6
	192000	21,7

Tabel 1. Hasil Pengujian Process time

Dengan melihat hasil pada setiap tabel dan grafik di atas, dapat dilihat *process time* yang terus meningkat ketika semakin banyak *sample* yang diambil setiap periodenya. Tetapi *process time* akan semakin cepat setiap *sample rate* dinaikkan, ini dikarenakan nilai *sample rate* berbanding terbalik dengan waktu, sehingga semakin besar nilai *sample rate*, waktu akan semakin kecil atau cepat. Jadi didapatkan kesimpulan, bahwa semakin tinggi kualitas suara yang dibutuhkan (nilai *sample* semakin besar), maka *sample rate* yang dibutuhkan juga akan semakin besar untuk mendapatkan *process time* yang minimal. Namun nilai *sample rate* yang besar membutuhkan penggunaan *RAM* lebih besar pula, karena semakin banyak yang harus diproses di *PC* setiap waktunya.

Selain itu, dapat kita lihat nilai *process time* terkecil adalah 1 ms yang terjadi ketika nilai *sample di set* ke 64 dan *sample rate* pada 192000 Hz. Lalu nilai terbesar *process time*-nya adalah 377 ms yang terjadi saat nilai *sample* di 2048 dan *sample rate* pada 11025 Hz. Dengan demikian kita mendapatkan batas bawah *process time* sebesar 1 ms batas atas *process time* sebesar 377 ms.

### Kesimpulan

Rangkaian bekerja dengan baik untuk menjalankan setiap fungsi dari efek gitar sehingga sistem secara keseluruhan dapat bekerja. Alat tidak memakan daya yang terlalu besar untuk dapat berfungsi, hanya sekitar 0,735-0,768 watt. Semakin besar nilai *sample*, kualitas suara akan semakin baik, namun *process time*, *process time* akan semakin besar juga. Semakin besar *sample rate*, *process time* akan semakin kecil, namun penggunaan *RAM* akan lebih besar.

### Daftar Pustaka:

- [1] Proakis, J.G. and Manolakis, D.G. (2007) Digital Signal Processing: Principles, Algorithms, and Applications. Pearson Education Ltd., New Jersey.
- [2] Pedal Shield, <http://www.electrosmash.com/pedalshield>
- [3] Arduino Due, <https://www.arduino.cc/en/Main/ArduinoBoardDue>
- [4] M. Banzhi, M. Shiloh. (2014). *Getting Started with Arduino 3<sup>rd</sup> Edition*. California: Maker Media