

## Optimasi Purwarupa Kendali Virtual Instrumen Musik Drum Berbasis Sensor Akselerometer dan LDR

Alveo Yuniar<sup>\*1</sup>, Bambang Nurcahyo Prastowo<sup>2</sup>

<sup>1</sup>Jurusan Ilmu Komputer dan Elektronika, FMIPA, UGM, Yogyakarta

<sup>2</sup>Program Studi Elektronika dan Instrumentasi, JIKE, FMIPA, UGM, Yogyakarta

e-mail: <sup>\*1</sup>[alveo\\_yuniar@mail.ugm.ac.id](mailto:alveo_yuniar@mail.ugm.ac.id), <sup>2\*</sup>[prastowo@ugm.ac.id](mailto:prastowo@ugm.ac.id)

### Abstrak

Telah berhasil dioptimasi sistem pengendali virtual instrumen musik drum berbasis sensor akselerometer dan LDR. Sistem ini merupakan sistem untuk mengendalikan virtual instrumen musik yang ada pada software DAW (software produksi musik) yang dapat digunakan untuk proses pembuatan instrumen musik drum digital dalam proses perekaman musik. Alat pengendali yang dioptimasi adalah pengendali virtual instrumen musik drum berbasis sensor akselerometer dan LDR yang telah dikembangkan sebelumnya oleh Migdal (2011). Optimasi yang dilakukan mencakup pengembangan sensor dalam menghasilkan kekerasan bunyi yang beragam, pengentasan masalah sensor yang terlalu sensitif dalam menghasilkan bunyi, dan penambahan beberapa kit drum yang dapat dibunyikan pada virtual instrumen musik.

Purwarupa sistem ini menggunakan board Arduino Mega 1280 sebagai pemroses, sensor yang digunakan dalam sistem adalah sensor akselerometer dengan tipe analog ADXL 335 dan sensor LDR. Sensor akselerometer akan mendeteksi percepatan yang terjadi. Percepatan yang terjadi akan dikirim menuju Arduino sebagai masukan analog yang akan diolah menjadi suatu data MIDI, data MIDI tersebut akan dikirim menuju virtual instrumen musik yang ada pada software DAW menggunakan kabel komunikasi MIDI to USB sehingga virtual instrumen tersebut dapat dikendalikan. Adapun sensor LDR berfungsi untuk mengganti bunyi drum yang dihasilkan sensor akselerometer sehingga bunyi yang dihasilkan oleh sistem menjadi semakin banyak.

**Kata kunci**—virtual instrumen musik, software DAW, Akselerometer, sensor, MIDI

### Abstract

Successfully optimized a prototype system that's reliable for controlling virtual musical instrumen drum based on accelerometer and LDR sensor. The purpose of this system is to controll virtual musical instruments that exist in the DAW software (music production software). it can be useful to create digital drum in music production events. The optimized system was a modification from system that had been developed previously by Migdal (2011), which has some weaknesses in functionality. Optimization included the development of sensors that can produce a diverse of variety sound level. Sensitivity problem in sensors when generate a sound, and the addition of several drum kits that can be played by system.

Prototype system uses an Arduino Mega 1280 board as the processor, the sensor that used in the system is ADXL 335 Accelerometer which is an analog accelerometer sensor and LDR sensor. When the acceleration occurs and detected by the accelerometer sensors, Sensors will sent an analog signal towards the Arduino board, and will be processed into a MIDI data, the data will be sent to the virtual instruments that exist in the software DAW using MIDI to USB. Furthermore the LDR sensor works to change the drum sounds that was generated by accelerometer so that the system can produce a lot of sounds in virtual musical instrumen.

**Keywords**—virtual musical instruments, DAW software, accelerometer, sensor, MIDI

## 1. PENDAHULUAN

*Virtual* instrumen musik adalah fitur dari *software* produksi musik yang biasa dikenal sebagai *software* DAW. *Virtual* instrumen musik mensimulasikan instrumen musik *real* kedalam bentuk digital yang dapat dimainkan melalui media komputer. Terdapat berbagai jenis *virtual* instrumen musik yang ada di dunia, salah satu yang paling sering digunakan adalah *virtual* instrumen musik drum. *Virtual* instrumen musik drum sering digunakan oleh beberapa produser musik dalam pembuatan musik drum digital dalam proses perekaman musik. Maraknya penggunaan *virtual* instrumen musik drum tersebut dikarenakan *virtual* instrumen musik drum memiliki kualitas *output* suara yang tidak jauh berbeda dengan drum *real* dan dari segi biaya, *virtual* instrumen musik drum dapat menghemat biaya dalam proses produksi musik.

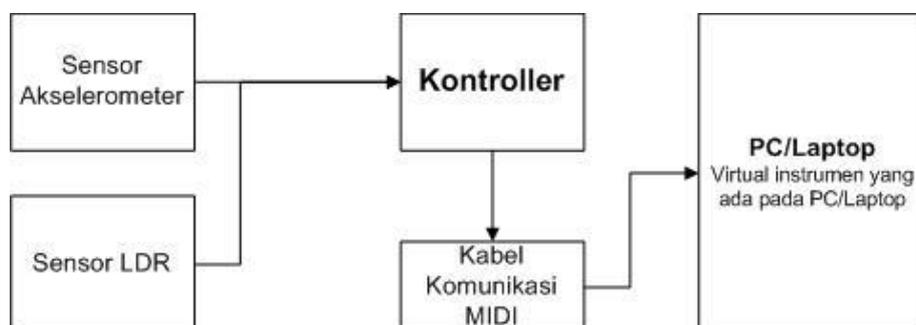
Untuk dapat memanfaatkan fitur *virtual* instrumen musik drum tersebut, awalnya digunakan keyboard *controller* dan drum *electric* sebagai media pengendali, namun kedua alat tersebut dirasa masih memiliki beberapa kelemahan baik dari sisi fungsionalitas hingga fisik, sehingga dikembangkan suatu pengendali *virtual* instrumen musik drum berbasis sensor akselerometer dan LDR yang dapat mengatasi kelemahan tersebut [1]. Dalam proses penggunaannya pengendali *virtual* instrumen musik drum berbasis sensor akselerometer tersebut mempunyai keunggulan pada bentuk fisik yang sederhana, namun masih memiliki beberapa kelemahan dari sisi fungsionalitas. Kelemahan-kelemahan tersebut adalah masalah sensor yang terlalu sensitif dalam menghasilkan bunyi, minimnya bunyi kit drum pada *virtual* instrumen musik yang dapat dibunyikan, serta masalah sistem yang tidak dapat menghasilkan kekerasan bunyi yang beragam sehingga menyulitkan proses perekaman *virtual* instrumen musik drum pada *software* DAW.

Maka dari itu dibutuhkan suatu proses optimasi untuk mengatasi kelemahan-kelemahan dari sistem yang telah dikembangkan sebelumnya seperti masalah sensor yang terlalu sensitif, minimnya bunyi kit drum yang dapat dibunyikan, serta masalah sistem yang tidak dapat menghasilkan kekerasan bunyi yang beragam, sehingga sistem dapat bekerja secara optimal dalam mengendalikan *virtual* instrumen musik drum yang ada pada *software* DAW dalam suatu proses perekaman musik.

## 2. METODE PENELITIAN

### 2.1 Analisis dan Perancangan Sistem

Optimasi purwarupa ini ditujukan untuk mengembangkan sekaligus mengatasi kelemahan sistem yang telah ada sebelumnya. Sistem dari purwarupa pengendali *virtual* instrumen musik drum ini dimodelkan sebagai sistem kalang terbuka (*open loop*). Secara umum, diagram blok sistem diperlihatkan pada Gambar 1.



Gambar 1 Diagram blok rancangan sistem pengendali *virtual* instrumen musik drum

Pada Gambar 1, Sensor akselerometer berfungsi sebagai sensor inti pada sistem. Melalui sensor inilah bunyi drum pada *virtual* instrumen musik drum dapat dibunyikan. Sensor

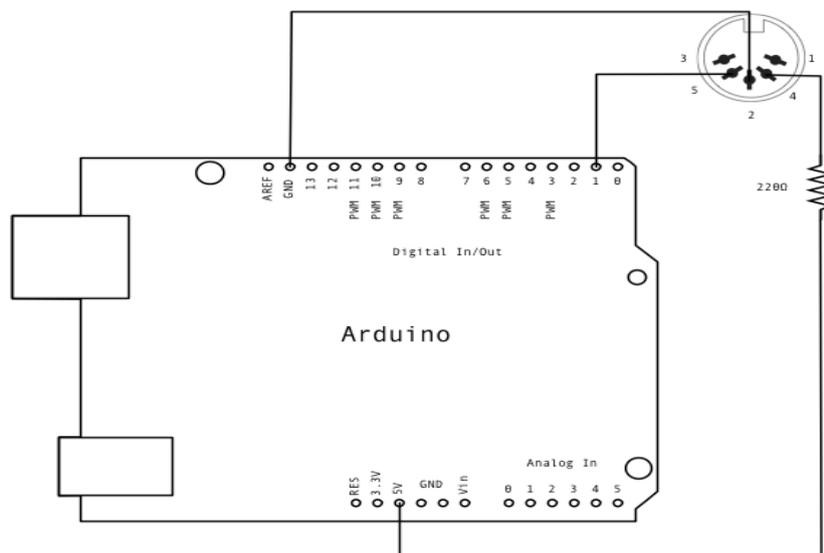
LDR digunakan sebagai *switch* yang akan mengganti jenis bunyi yang dibunyikan dari sensor akselerometer, sehingga bunyi drum yang dapat dibunyikan menjadi semakin banyak. Terdapat tiga sensor akselerometer dan empat sensor LDR dalam sistem dengan tipe analog, data-data analog sensor tersebut akan masuk menuju kontroller.

Kontroller pada sistem menggunakan Arduino Mega 1280. Sensor akselerometer yang digunakan adalah ADXL335. Sensor LDR yang digunakan adalah sensor LDR tipe PGM 12. *Software* DAW yang digunakan adalah *software* DAW keluaran Steinberg Jerman, Cubase, dengan *virtual* instrumen musik drum bernama Addictive drum. Agar sistem dapat berkomunikasi dengan *software* DAW pada PC, digunakan suatu kabel komunikasi MIDI dimana tergambar pada Gambar 2.



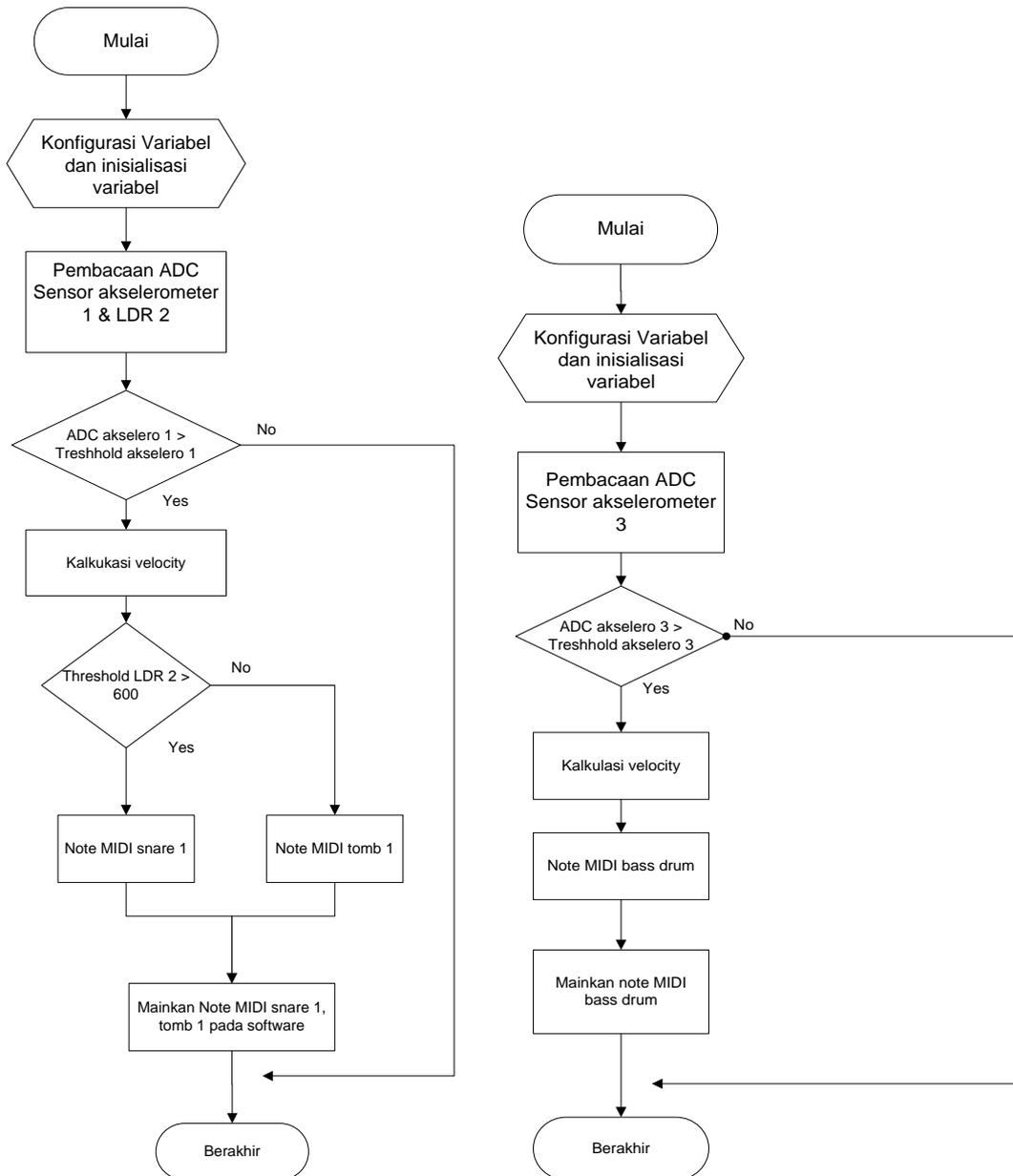
Gambar 2 Kabel Komunikasi MIDI

Agar kabel MIDI tersebut dapat terhubung dengan sistem, digunakan suatu konektor MIDI 5 din yang akan menghubungkan input kabel MIDI menuju sistem. Hubungan antara sistem dengan kabel MIDI tersebut tergambar pada Gambar3 yang diambil dari referensi [2].



Gambar 3 Hubungan Arduino dan Mikrokontroller

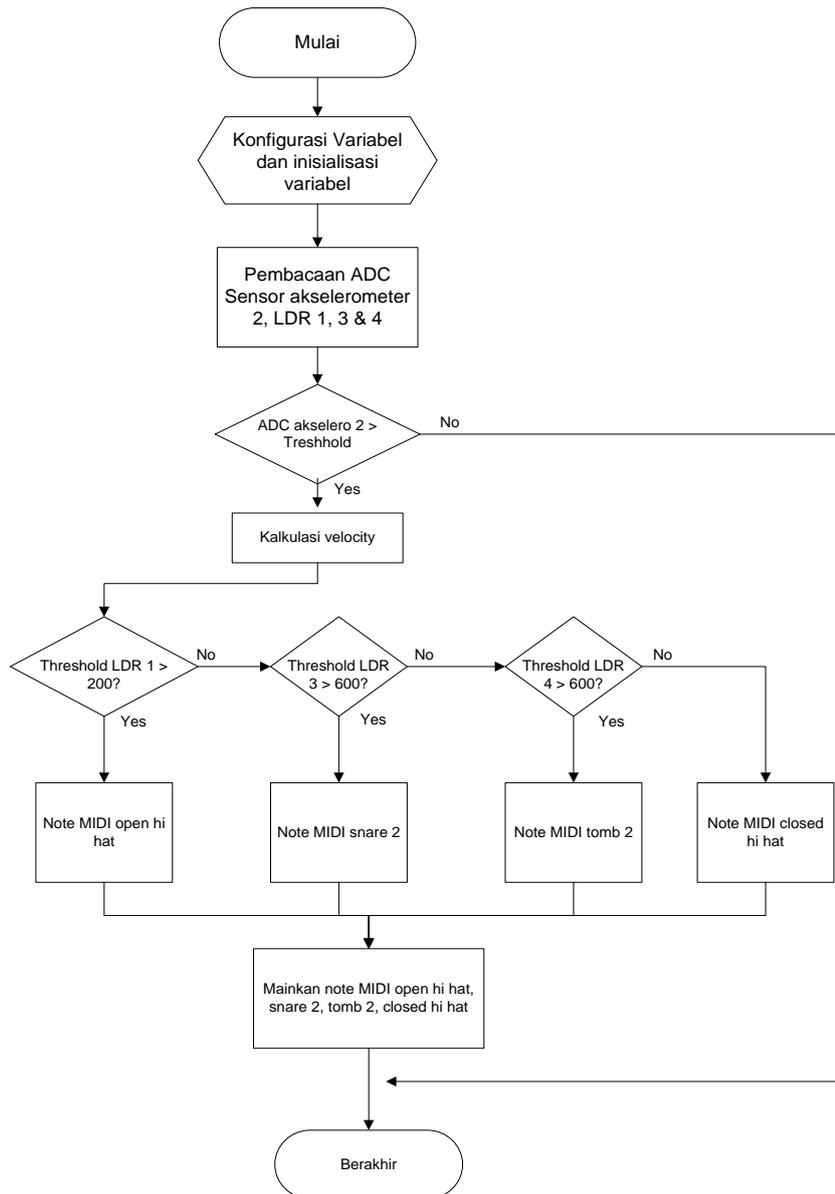
Perangkat lunak yang digunakan untuk pemrograman adalah Arduino IDE (*Integrated Development Environment*) versi 1.0. Bahasa pemrograman yang digunakan arduino adalah bahasa pemrograman turunan C++ sehingga kebanyakan fungsi C dan C++ dapat dijalankan di arduino. Pemilihan bahasa pemrograman arduino sendiri karena arduino bersifat *open source* yang membuat banyak pengembang-pengembang membuat *library* sehingga memudahkan kita. Gambar 4 dan Gambar 5 menunjukkan diagram alir cara kerja sistem secara umum.



Gambar 4 Flowchart pengendali *virtual drum* pada bagian stik A dan sandal A

besar jalannya program adalah dapat mengolah keluaran ADC dari sensor-sensor pada sistem hingga mengeleuarkan *output* berupa pesan MIDI yang akan dikirimkan dari mikrokontroler menuju *software* DAW pada PC/laptop melalui sebuah kabel MIDI to USB, sehingga sistem dapat mengendalikan *virtual* instrumen musik yang terdapat pada *software* DAW. Dua sensor akselerometer pada stik A dan B akan membunyikan empat kit drum yaitu kit *snare*, *cowbell*, *crash*, dan *hi hat* dengan memanfaatkan *axis X* dan *axis Z* pada masing-masing sensor akselerometer. Sensor akselerometer pada stik B pada perancangan program akan berhubungan dengan sensor LDR 1 pada sandal B yang berfungsi sebagai pedal *hi hat* untuk mengatur bunyi *hi hat* apakah harus berbunyi *open* atau *close*.

Selain berhubungan dengan LDR 1, sensor akselerometer pada stik B juga akan berhubungan prosesnya dengan LDR 3 dan 4, untuk mengganti bunyi kit drum *hi hat* menjadi kit drum *tomb 2* dan kit drum *snare 2*. Sensor akselerometer pada stik A sendiri juga akan berhubungan dengan LDR 2 yang akan mengganti bunyi kit *snare 1* menjadi kit drum *tomb 1*.



Gambar 5 *flowchart* pengendali *virtual drum* pada bagian stik B.

Sandal A yang sudah dipasang oleh sensor akselerometer berfungsi sebagai *bass pedal* yang nantinya akan membunyikan kit *bass drum* dengan memanfaatkan *axis Z* yang terdapat pada sensor akselerometer.

Permulaannya sistem akan menginisialisasi variable yang ada pada program seperti pin analog, nilai treshold, dll. Setelah itu sensor akselerometer 1 akan mendeteksi gerakan dan mengeluarkan nilai ADC, setelah mendapatkan nilai ADC Arduino akan mengolah data ADC tersebut dan menentukan apakah nilai ADC yang masuk lebih besar dari *threshold* akselerometer, Setelah kondisi terpenuhi maka akan dilakukan kalkulasi *velocity* (kekerasan bunyi) dan penentuan note MIDI yang akan dimainkan pada *software* DAW, penentuan note MIDI sendiri juga ditentukan oleh nilai *threshold* sensor LDR.

## 2.2 Implementasi Sistem

Rancangan purwarupa kendali *virtual* instrumen musik drum berbasis akselerometer dan LDR yang sudah diimplementasikan terlihat pada Gambar 6. Pada Gambar 6 implementasi purwarupa kendali *virtual* instrumen musik drum ini mempunyai media berupa stik drum yang terbuat dari pipa PVC, pada pipa tersebut diletakan dua sensor akselerometer dan tiga sensor

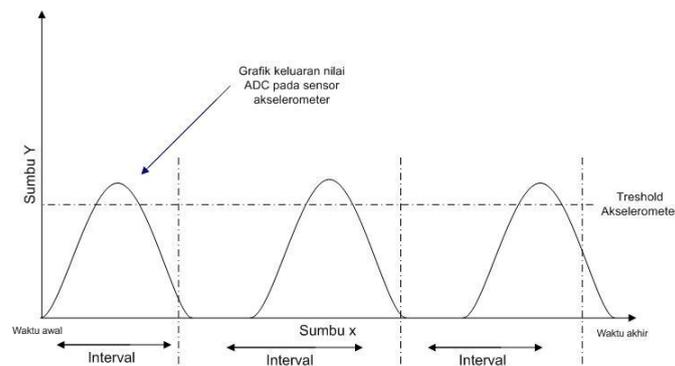
LDR, media sebagai pengganti pedal bass dan pedal hi hat adalah sandal selop berbahan gabus, pada sandal ini salah satunya terpasang sensor akselerometer dan LDR, selain itu terdapat *hardware* pemroses yang akan mengkoneksikan *output* dari sensor-sensor tersebut. Board Arduino Mega terpasang pada *hardware* pemroses ini, dimana akan terkoneksi oleh kabel komunikasi MIDI yang akan terhubung menuju Komputer yang telah terinstal *software* DAW didalamnya.



Gambar 6 Implementasi sistem

### 3. HASIL DAN PEMBAHASAN

Merujuk pada *datasheet* [3], nilai ADC sensor akselerometer dalam keadaan tidak mendeteksi percepatan ( $0g$ ) adalah sebesar 512. Seketika saat sensor pada sistem dihentikan, atau dalam kata lain mengalami suatu percepatan, maka akan terjadi perubahan nilai ADC, sehingga akan menghasilkan suatu kurva yang mempunyai nilai maksimum, hal ini ditunjukkan pada grafik kurva yang ada pada Gambar 7. Grafik tersebut adalah interpretasi keadaan sensor saat mendeteksi suatu percepatan, pada awalnya grafik ada pada rentang yang stabil dalam mengeluarkan nilai ADC, dan sesaat ketika sensor dihentikan, akan terjadi perubahan, sehingga akan menciptakan grafik dengan puncak dan lembah.



Gambar 7. Grafik ADC akselerometer saat mendeteksi percepatan

Kurva tersebut adalah interpretasi dari nilai ADC yang dihasilkan sensor akselerometer ketika mendeteksi suatu percepatan, tampak jelas, bahwa pada saat sensor akselerometer mendeteksi suatu percepatan, nilai ADC akan bertambah besar dari nilai ADC sesaat sebelum mendeteksi percepatan, dan akan kembali lagi ke nilai ADC awal saat sensor berada dalam kondisi diam, atau dalam kata lain sensor tidak mendeteksi percepatan, hal ini sesuai dengan teori yang telah ditulis pada jurnal [4]. Berdasarkan analisis kerja sensor akselerometer tersebut, agar sensor akselerometer dapat digunakan untuk mengendalikan *virtual* instrumen musik, diperlukan sebuah nilai batasan (*threshold*) yang berfungsi sebagai batas nilai, yang nantinya akan menentukan *range* nilai ADC berapa saja yang akan diproses sehingga sensor dapat bekerja sebagaimana mestinya, dengan adanya variabel nilai *threshold* ini, akan mengatur seberapa sensitif sistem dalam membunyikan *virtual* instrumen musik. Nilai batasan (*threshold*) tersebut ditentukan dengan cara mengamati besar nilai ADC dari sensor akselerometer yang nilainya akan dibandingkan dengan nilai percepatan ( $g$ ) yang ditampilkan oleh *serial monitor*

saat sensor akselerometer dihentikan. Nilai ADC yang tampil sendiri adalah cerminan nilai percepatan dari sensor akselerometer. Tabel 1 menunjukkan perbandingan range nilai ADC sensor dan akselerometer saat mendeteksi hentakan dan besar nilai percepatan yang dihasilkan dari hentakan tersebut, pada pengujian ini penulis menghentakkan sensor dengan level hentakan yang paling pelan hingga hentakan tercepat.

Tabel 1 Tabel nilai ADC dan nilai percepatan (g)

Level Hentakan	Rentang ADC yang dihasilkan	Rentang nilai percepatan (g)
Tidak ada hentakan	512 - 614	0,12 - 0,99
pelan	614 - 713	0,99 - 1,97
sedang	713 - 820	1,97 - 2,97
kencang	820 - 923	2,97 - 3,99
sangat kencang	923 -1023	3,99 - 5,00

Dari data yang didapat pada Tabel 1, agar sensor tidak terlalu sensitif dalam menghasilkan bunyi, kisaran nilai *threshold* ADC minimal harus berkisar pada level hentakan sedang, yaitu pada nilai ADC 713 – 820, yang pada rentang nilai ADC tersebut, besar nilai percepatan sensor ada pada rentang 1,97g – 2,97g. Apabila nilai *threshold* ada pada level dibawah itu, maka sensor akan sangat sensitif dalam menghasilkan bunyi. Hal ini dikarenakan dengan hentakan yang sangat pelan sensor dapat membunyikan bunyi drum pada *virtual* instrumen, hal ini ditunjukkan dengan nilai percepatan yang terjadi pada level ini sangatlah kecil, yaitu ada pada kisaran 0-2 g, jika nilai *threshold* ada pada kisaran itu, tentu sensor akan sangat sensitif terutama apabila terjadi senggolan, ataupun gerakan yang tak terduga sebelumnya, sehingga dibutuhkan rentang nilai g yang lebih besar, sehingga nilai batasan pada program sendiri ditetapkan sebesar 820, yang pada tabel, adalah nilai ADC pada level kencang dan sedang, dengan nilai g yang lebih besar yang ada pada kisaran 2-5 g. Penetapan nilai sebesar itu ditetapkan karena pada level nilai inilah sensor tidak terlalu sensitif dalam menghasilkan bunyi, dengan nilai batasan minimal sebesar itu, maka dibutuhkan hentakan yang cukup keras hingga sensor dapat membunyikan *virtual* instrumen musik.

Nilai kekerasan bunyi (*velocity*) memiliki rentang nilai dari 0-127, nilai kekerasan bunyi tersebut dapat diketahui dengan melakukan perekaman *note* MIDI *virtual* instrumen musik drum pada *software* DAW. Agar sistem dapat menghasilkan nilai kekerasan bunyi yang beragam pada *software*, digunakan sebuah persamaan yang akan menghitung nilai selisih dari nilai ADC maksimum pada *array*, dengan nilai sebelumnya pada *array*. Persamaan 1 adalah persamaan yang digunakan pada program untuk mencari nilai selisih tersebut.

$$dADC = ADC_{akhir} - ADC_{awal} \quad (1)$$

dimana,

$dADC$  = nilai selisih  
 $ADC_{akhir}$  = nilai ADC pada saat sensor akselerometer dihentikan  
 $ADC_{awal}$  = nilai ADC sebelum sensor akselerometer dihentikan

Dengan menggunakan persamaan tersebut didapat nilai kekerasan bunyi yang beragam. Hasil pengujian nilai kekerasan bunyi dengan menggunakan persamaan tersebut ditampilkan pada Tabel 2.

Tabel 2 Nilai *dADC* dan *velocity* dengan *smooth value* 5

Jumlah Pengujian	Analogread Akhir	Analogread Awal	dADC4	Velo4
1	852	834	18	18
2	908	782	122	122
3	897	798	94	94
4	898	823	71	71
5	903	790	108	108
6	972	816	55	55
7	899	820	75	75
8	893	824	65	65
9	899	802	95	95
10	891	819	70	70

Nilai *dADC* yang dihasilkan tersebut akan dikonversi menjadi nilai kekerasan bunyi dengan menambahkan fungsi *constrain* pada program. Karena rentang nilai kekerasan bunyi hanya ada pada rentang nilai 0-127 dengan nilai maksimum adalah 127, maka fungsi *constrain* tersebut akan membatasi nilai *dADC* yang melebihi 127 agar dianggap sebagai nilai kekerasan bunyi yang mempunyai nilai 127. Nilai *ADCawal*, *ADCakhir*, *dADC*, serta konversi nilai *dADC* menjadi nilai kekerasan bunyi (*velocity*) sendiri akan ditampilkan pada *serial monitor* pada Arduino IDE yang kemudian akan ditampilkan kembali pada Tabel 3. Data yang didapat pada tabel tersebut memiliki nilai kekerasan bunyi yang sangat bervariasi, dengan jangkauan nilai minimum dan maksimum.

Tabel 3 Nilai *dADC* dan *velocity*

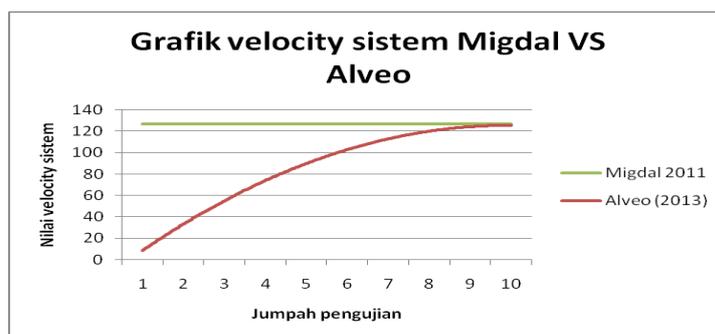
Nilai selisih ( <i>dADC</i> ) tiap kit drum	Data pengujian ke									
	1	2	3	4	5	6	7	8	9	10
<i>Kit snare 1</i>	54	63	64	73	80	103	110	125	125	140
<i>Kit hihat</i>	35	54	61	66	74	92	94	98	103	112
<i>Kit snare 2</i>	30	62	65	73	80	80	110	127	127	128
<i>Kit bass drum</i>	29	30	44	75	78	106	111	118	121	126
<i>kit crash</i>	14	35	42	62	103	112	112	118	119	129
<i>Kit cowbell</i>	20	20	40	79	83	89	102	112	118	134
<i>Kit tomb1</i>	36	68	68	78	102	116	118	120	121	127
<i>Kit tomb 2</i>	23	68	69	77	82	85	106	116	132	147
Kekerasan bunyi ( <i>velocity</i> ) tiap kit drum	Data pengujian ke									
	1	2	3	4	5	6	7	8	9	10
<i>Kit snare 1</i>	54	63	64	73	80	103	110	125	125	127
<i>Kit hihat</i>	35	54	61	66	74	92	94	98	103	112
<i>Kit snare 2</i>	30	62	65	73	80	80	110	127	127	127
<i>Kit bass drum</i>	29	30	44	75	78	106	111	118	121	126
<i>kit crash</i>	14	35	42	62	103	112	112	118	119	127
<i>Kit cowbell</i>	20	20	40	79	83	89	102	112	118	127
<i>Kit tomb1</i>	36	68	68	78	102	116	118	120	121	127
<i>Kit tomb 2</i>	23	68	69	77	82	85	106	116	127	127

Alat kendali ini sendiri juga dibandingkan performanya dengan alat kendali yang telah dibuat sebelumnya oleh Migdal dan *keyboard controller* Pabrikan merk M-Audio. sehingga pada Tabel 4 ditampilkan perbedaan nilai kekerasan bunyi dari ketiga sistem

Tabel 4 Perbandingan nilai *velocity* kendali *virtual* instrumen musik drum buatan Migdal, M-Audio, dan Alveo

Pengujian	Migdal (2011)	M - Audio	Alveo (2013)
1	127	26	14
2	127	30	35
3	127	64	42
4	127	67	62
5	127	79	112
6	127	86	118
7	127	90	103
8	127	97	112
9	127	100	119
10	127	107	127
<b>Rata-rata velocity</b>	<b>127</b>	<b>74.6</b>	<b>84.4</b>
<b>Jangkauan</b>	<b>0.00</b>	<b>81</b>	<b>113</b>
<b>Standar deviasi</b>	<b>0.00</b>	<b>28.10</b>	<b>41.78</b>

Hasil yang didapat menunjukkan perbedaan performa yang mencolok dari ketiga sistem. Perbedaan mencolok tersebut ditampilkan kembali dalam sebuah grafik yang ditunjukkan pada gambar 8 dan 9. Gambar 8, dan 9 menunjukkan tampilan hasil perekaman note MIDI *virtual* instrumen musik drum dengan alat buatan migdal (2011) yang merupakan versi terdahulu dari alat kendali yang dikembangkan pada penelitian ini oleh penulis, dan tampilan hasil perekaman note MIDI dengan *keyboard controller* pabrikan M-Audio.

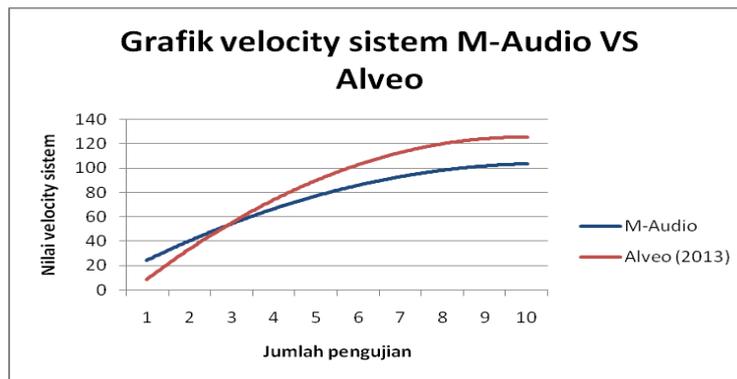


Gambar 8. Grafik perbandingan sistem Migdal dan Veo

Gambar 8 menunjukkan grafik perbandingan performa dari sistem pengendali *virtual* instrumen musik drum pengembangan penulis dan pengendali berbasis akselerometer sebelumnya. Grafik 8 menunjukkan bahwa alat pengendali berbasis akselerometer dan LDR yang dikembangkan sebelumnya oleh Migdal tidak mampu menghasilkan kekerasan bunyi yang beragam pada *virtual* instrumen drum yang ada pada *software* DAW, hal ini mengacu pada nilai kekerasan bunyi yang konstan di nilai 127, sehingga *virtual* instrumen musik drum tersebut tidak dapat mengimplementasikan bunyi drum pada *virtual* instrumen musik dengan baik, sehingga proses perekaman akan menjadi lebih rumit. Hal ini dikarenakan nilai kekerasan bunyi dari note MIDI hasil perekaman *virtual* instrumen musik drum pada *software* DAW tersebut harus di *edit* kembali agar mendapatkan bunyi drum yang natural dan baik.

Berdasarkan hasil pengujian, didapat nilai kekerasan bunyi rata-rata yang dapat dihasilkan oleh sistem adalah sebesar, 84,4. Nilai ini didapat dari sepuluh kali pengujian sistem pada saat membunyikan *virtual* instrumen musik drum. Nilai standar deviasi yang dihasilkan oleh sistem adalah sebesar 41,78, dengan standar deviasi sebesar itu menunjukkan nilai yang menyimpang sangat besar, hal ini berarti kekerasan bunyi yang dihasilkan memiliki jangkauan nilai yang sangat besar. Berdasarkan data tersebut dapat disimpulkan bahwa sistem sangat variatif dalam menghasilkan kekerasan bunyi

Selain perbandingan dengan alat pengembangan dari Migdal, sistem yang telah dioptimasi dibandingkan juga dengan pengendali *virtual* instrumen musik M-Audio, Gambar 9 menunjukkan grafik perbandingan sistem yang dioptimasi penulis dengan sistem dari M-Audio.



Gambar 9. Grafik perbandingan sistem M-Audio dan Veo

Pada Gambar 9 dapat dilihat bahwa sistem yang telah dikembangkan penulis dan sistem pengendali pabrikan buatan M-Audio mempunyai performa yang tidak terlalu jauh berbeda. Berdasarkan data pengujian, didapat nilai kekerasan bunyi rata-rata pada pengendali buatan M-Audio sebesar 74,6 dan standar deviasi yang besar, yaitu 28,10. Hal ini menunjukkan sistem juga dapat menghasilkan nilai kekerasan bunyi yang bervariasi, hanya saja perbedaan yang mencolok sistem ini dengan sistem yang telah dikembangkan oleh penulis adalah pada level kekerasan bunyi terkeras yang dapat dihasilkan. Sistem yang telah dikembangkan penulis dapat mencapai level kekerasan bunyi maksimum yaitu 127, sedangkan pada sistem M-Audio level kekerasan bunyi maksimal adalah pada level 107.

#### 4. KESIMPULAN

Dari hasil pengamatan, pengujian, dan analisis dalam perancangan sistem yang telah dilakukan, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Purwarupa kendali *virtual* instrumen musik drum pada *software* DAW berbasis sensor akselerometer dan LDR berhasil dibuat dan dioptimasi. Optimasi meliputi pengaturan masalah sensor yang terlalu sensitif, sistem dapat menghasilkan kekerasan bunyi yang beragam, dan bunyi drum yang dapat dibunyikan pada *virtual* instrumen lebih banyak.
2. Nilai kekerasan bunyi (*velocity*) dapat diimplementasikan dengan baik oleh sistem, mencakup level kekerasan bunyi terendah hingga terkeras yaitu pada kisaran 14-127
3. Pemanfaatan dua *axis* pada tiap sensor akselerometer untuk membunyikan dua kit drum, ternyata tidak sepenuhnya berjalan dengan baik, karena terkadang bunyi yang dihasilkan saling mengganggu satu sama lain
4. Sensor LDR dalam sistem bekerja sangat baik sebagai *switch* yang mengganti bunyi tiap kit drum yang dibunyikan oleh sensor akselerometer

#### DAFTAR PUSTAKA

- [1] Migdal, M., 2011, Arduino Air Drum, <http://blimp12.blogspot.com/2012/01/arduino-air-drums.html>, diakses tanggal 10 September 2012
- [2] Kent, M., 1999, *Universal Serial Bus Device Class Definition for MIDI Devices*, USB Implementers Forum, USA.
- [3] Analog Devices, 2009, *Datasheet ADXL335*, Analog Devices, Norwood.
- [4] Setiawan, I., 2009, *Hasil Uji Kalibrasi Sensor ADXL 335*, Jurnal Teknik Elektro UNDIP, Semarang.