

Rancang Bangun Generator PWM Berbasis Mikrokontroler AVR ATmega

Firdaus^{1*}, Rivanol Chadri², Nasrullah³

¹ Jurusan Teknik Elektro, ^{2,3} Jurusan Teknik Mesin, Politeknik Negeri Padang
Jurusan Teknik Elektro Politeknik Negeri Padang, Jl. Limau Manih Padang, 25164, Indonesia
*e-mail: mrdauz@yahoo.com

Abstrak— PWM banyak digunakan di bidang kendali otomatis, elektronika daya, dan komunikasi seluler. Penelitian sebelumnya merancang generator PWM memakai chip OMAP-L138 menghasilkan rangkaian sederhana, presisi tinggi, fleksibel dan portable. Generator lain menggunakan single board computer FEZ Panda III dan board Arduino untuk power inverter. Sementara FPGA Spartan 3 juga dipakai untuk menghasilkan sinyal PWM yang dapat divariasikan duty cycle-nya. Pada makalah ini, generator sinyal PWM dibuat menggunakan mikrokontroler AVR ATmega8535 dimana parameter frekuensi dan duty cycle sinyal keluarannya dapat diatur melalui keypad. Sinyal tersebut diuji menggunakan alat ukur untuk diketahui keakuratannya. Keluarga mikrokontroler AVR ATmega memiliki timer/counter dengan salah satu mode operasinya yaitu fast PWM. Agar frekuensi dan duty cycle dapat diubah sesuai keinginan maka register ICR digunakan untuk menyimpan nilai TOP dan register OCR untuk nilai MAX. Nilai OCR yang mempengaruhi duty cycle dan nilai ICR yang mempengaruhi rekuensi. Hasil yang didapatkan adalah semakin tinggi frekuensi PWM maka semakin besar nilai perbedaan frekuensi yang diukur dengan yang diinginkan, ini disebabkan oleh nilai frekuensi yang besar membuat kecilnya nilai yang ada di dalam register ICR walaupun nilai pembagi frekuensinya sudah paling minimal diberikan. Sementara selisih antara pengukuran dan perhitungan pada Duty Cycle memberikan hasil cukup kecil di bawah 1 %. Selisih ini juga terjadi akibat dari pembulatan nilai ICR dan OCR, namun di frekuensi 20 kHz dan Duty Cycle 25% dimana nilai ICR adalah 599 dan OCR adalah 116 menghasilkan selisih frekuensi dan Duty Cycle yang paling minimum. Dari percobaan yang telah dilakukan, rancang bangun PWM generator berbasis mikrokontroler AVR ATmega berhasil diwujudkan

Kata kunci PWM, mikrokontroler, AVR, timer, counter

Abstract— PWM is widely used in the fields of automatic control, power electronics and cellular communications. Previous research designed a PWM generator using the OMAP-L138 chip to produce a simple, high-precision, flexible and portable circuit. Another generator uses a single board computer FEZ Panda III and an Arduino board for power inverters. While the FPGA Spartan 3 is also used to generate PWM signals that can vary the duty cycle. In this paper, the PWM signal generator is made using the AVR ATmega8535 microcontroller where the frequency parameters and the output signal duty cycle can be adjusted via the keypad. The signal is tested using a measuring instrument for its accuracy. The ATmega AVR microcontroller family has a timer / counter with one of its operating modes, namely fast PWM. In order to change the frequency and duty cycle as desired, the ICR register is used to store the TOP value and the OCR register for the MAX value. The OCR value determine the duty cycle and the ICR value specify the frequency. The results are the higher the PWM frequency, the greater the value of the measured and desired frequency difference, this is because the large frequency makes ICR register value becomes small even though the frequency divider at minimum value. The difference between measurements and calculations on the Duty Cycle gives the result under 1%. This difference also occurs due to the rounding of the ICR and OCR values, but at a frequency of 20 kHz and a 25% Duty Cycle where the ICR value is 599 and OCR is 116 resulting in the minimum difference in frequency and Duty Cycle. From the experiments that have been carried out, the design of the PWM generator based on the AVR ATmega microcontroller has been successfully realized

Keywords: PWM, microcontroller, AVR, timer, counter

© 2020 Elektron Jurnal Ilmiah

I. PENDAHULUAN

Modulasi lebar pulsa atau *Pulse Width Modulation* (PWM) adalah metode pengkodean digital untuk level sinyal analog, yang setara dengan bentuk gelombang yang dibutuhkan oleh modulasi serangkaian pulsa[1]. PWM adalah bagian yang sangat penting dari teknologi elektronika daya dan banyak digunakan di bidang kendali otomatis, komunikasi seluler, teknologi komputer, pengukuran dan deteksi sinyal. PWM adalah inti teknis dari keseluruhan sistem dalam sistem aplikasi praktis seperti deteksi daya[2]. Terdapat dua cara untuk membangkitkan sinyal PWM yaitu menggunakan rangkaian elektronika analog atau

rangkainan elektronika digital. Rangkaian analog terdiri dari komponen diskrit seperti kapasitor, resistor dan *chip op-amp*, tapi kekurangan menggunakan rangkaian ini adalah tidak dapat diubah dengan mudah kemampuan kerjanya karena harus mengganti komponen terlebih dahulu. Rangkaian digital terutama yang memakai perangkat sistem embedded seperti mikrokontroler AVR, ARM, *chip DSP*, *chip Field Programmable Gate Array* (FPGA)[3], dapat diprogram untuk menghasilkan bentuk gelombang PWM, keuntungan rangkaian digital untuk menghasilkan bentuk gelombang PWM adalah pada kontrolnya dan modifikasinya fleksibel dan mudah. Dengan improvisasi yang berkelanjutan pada performa

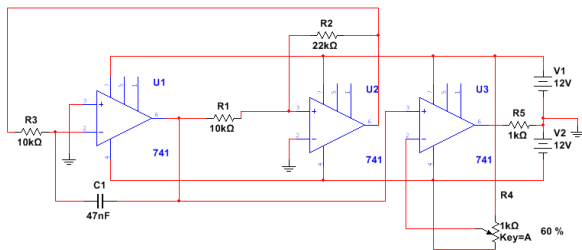
switching komponen elektronik, teknologi PWM berkembang dengan pesat. Generasi gelombang PWM telah digantikan oleh sirkuit terpadu (IC) skala besar yang membuat teknologi modulasi digital menjadi teknologi utama dari teknologi modulasi lebar pulsa[4]. Oleh karena itu, sistem pembangkitan PWM sebagian besar menggunakan IC khusus yang terintegrasi di dalamnya *modulator* PWM dan *mikroprosesor*.

Penelitian sebelumnya merancang generator PWM memakai chip OMAP-L138 menghasilkan struktur rangkaian yg sederhana, memiliki kepresisian yang tinggi fleksibel dan portable[5]. Generator yang lain menggunakan single board computer FEZ Panda III dan board Arduino untuk power inverter[6]. Sementara FPGA Spartan 3 juga dipakai untuk menghasilkan sinyal PWM yang dapat divariasikan *duty cycle*-nya[7]. Multi channel PWM generator digunakan untuk robot mainan telah diterapkan pada mikrokontroler PIC18f8720[8].

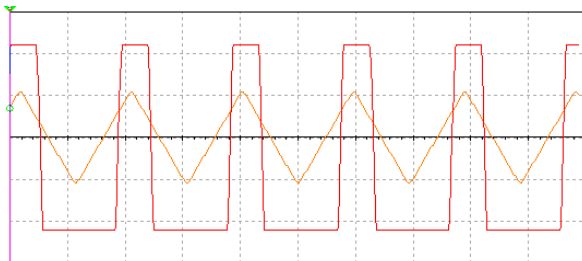
Pada makalah ini, generator sinyal PWM dibuat menggunakan mikrokontroler AVR ATmega8535 dari Microchip Technology Inc. dimana parameter frekuensi dan *duty cycle* sinyal keluarannya dapat diatur melalui keypad. Sinyal tersebut diuji menggunakan alat ukur untuk diketahui keakuratannya.

A. Prinsip PWM

Bentuk gelombang PWM terdiri dari serangkaian pulsa persegi panjang dengan proporsi *amplitudo* positifnya dapat berbeda dengan *amplitude* nol atau negatifnya. Rangkaian generator PWM analog dan bentuk sinyalnya ditunjukkan pada Gambar 1 dan Gambar 2.



Gambar 1. Skema rangkaian generator PWM analog



Gambar 2 Bentuk gelombang PWM yang dihasilkan generator PWM analog

Gelombang PWM keluar dari komparator op-amp U3 yang membandingkan dua tegangan input yaitu gelombang segitiga dari output U1 dan pembagi tegangan di potensiometer R4. Selama tegangan dari

potensiometer lebih besar dari amplitudo gelombang segitiga maka output U3 bertegangan saturasi negatif sebaliknya output U3 bertegangan saturasi positif. Potensiometer inilah yang mengatur *duty cycle* gelombang PWM dimana *duty cycle* ditentukan oleh (1)

$$D = \frac{t_P}{T} * 100\% \tag{1}$$

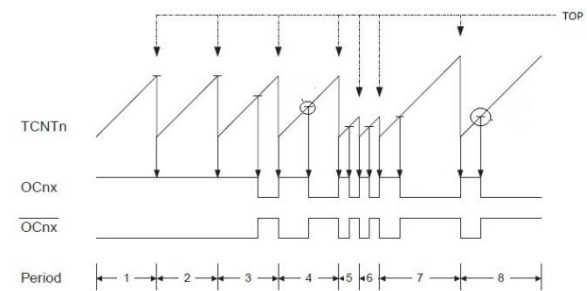
D adalah *duty cycle* (%), t_p yaitu lamanya amplitudo positif (s) dan T ialah perioda gelombang PWM (s)

B. Timer/Counter mikrokontroler AVR ATmega

Keluarga mikrokontroler AVR ATmega memiliki timer/counter 8 bit dan 16 bit, masing-masing memiliki fitur tersendiri yang dapat dimanfaatkan sesuai kebutuhan. Khusus timer/counter 16 bit beberapa kelebihan adalah memiliki beberapa unit output yang independen, register pembanding ganda, peredam noise, perioda PWM yang dapat divariasikan dan dapat digunakan untuk generator frekuensi[9].

Terdapat empat mode operasi jika ingin memakai timer/counter AVR ATmega yaitu normal, *clear timer on compare match*, *fast PWM* dan *phase correct PWM*. Agar dapat menghasilkan gelombang PWM maka timer/counter dioperasikan pada *mode fast* atau *phase correct PWM*. Salah satu perbedaan antara mode fast dan *phase correct* adalah frekuensi gelombang yang bisa dihasilkan oleh mode fast dua kali lebih besar dari mode *phase correct*[9].

Resolusi PWM untuk mode fast dapat dipilih salah satunya yaitu 8, 9, atau 10 bit atau ditentukan oleh register Input Capture Register (ICR) atau Output Compare Register (OCR). Resolusi minimum yang diperbolehkan adalah 2 bit dengan memberikan nilai 0x0003 di ICR atau OCR dan resolusi tertinggi 16 bit dengan memberikan nilai MAX atau 0xFFFF di dua register tersebut [9]. Cara kerja generator PWM mode fast diilustrasikan pada Gambar 3.



Gambar 3 Diagram timing mode fast PWM[9]

Counter TCNT akan terus menghitung naik hingga nilainya sama dengan salah satu nilai batas (TOP) yaitu 0x00FF, 0x01FF, 0x03FF, nilai di ICR atau nilai di OCR kemudian direset kembali menuju nol 0x0000. Register OCR atau ICR dapat digunakan untuk menentukan nilai TOP sehingga frekuensi PWM yang dihasilkan ditentukan oleh (2).

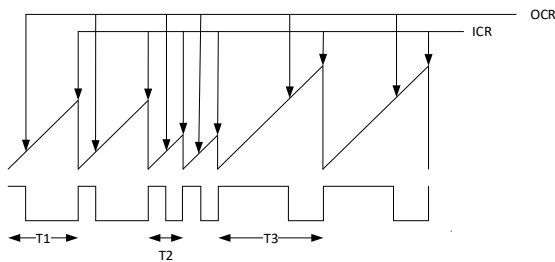
$$f_{OCPWM} = \frac{f_{clkIO}}{N(1+TOP)} \quad (2)$$

dimana f_{OCPWM} adalah frekuensi di pin OC berupa output PWM, f_{clkIO} ialah frekuensi osilator mikrokontroler, N yaitu pembagi prescaler (1, 8, 64, 256, 1024) dan TOP adalah nilai batas[9].

Proses terbentuknya PWM karena nilai TCNT yang terus berubah naik akan selalu dibandingkan dengan nilai tertentu yang dapat disimpan di register OCR atau ICR, jika sama maka logika di pin OC akan low dan akan menjadi high ketika TCNT bernilai nol atau melewati nilai TOP. Kombinasi register ICR sebagai penentu nilai TOP dan register OCR sebagai penyimpan nilai pembanding (MAX) atau sebaliknya dapat membuat gelombang keluaran di pin OC bervariasi frekuensi dan duty cycle-nya.

II. METODE

Penggunaan Timer/Counter dalam mode fast PWM memberikan bandwidth frekuensi output yang lebih besar. Agar frekuensi dan duty cycle dapat diubah sesuai keinginan maka register ICR digunakan untuk menyimpan nilai TOP dan register OCR untuk nilai MAX. Nilai OCR yang mempengaruhi duty cycle dan nilai ICR yang mempengaruhi perioda atau frekuensi dijelaskan dalam Gambar 4.



Gambar 3 Diagram timing pembentukan gelombang PWM dari nilai OCR dan ICR

Gunakan (2) untuk mendapatkan nilai ICR namun tentukan terlebih dahulu frekuensi sinyal PWM yang diinginkan misalkan 1 kHz dengan frekuensi osilator mikrokontrolernya 12 MHz dan N adalah 1024 maka diperoleh,

$$TOP = \frac{12000000}{1024 * 1000} - 1 = 10.718$$

Hasilnya TOP sebesar 10,718 dan dibulatkan menjadi 10 karena register ICR tidak dapat memuat nilai TOP berbentuk angka pecahan. Nilai ini digunakan juga untuk menentukan duty cycle dan disimpan di register OCR. Berdasarkan (1), rasio OCR dan ICR menentukan duty cycle (DC).

$$DC = \frac{OCR}{ICR} * 100\% \quad (3)$$

Misalkan DC ditentukan 27 %, didapatkan nilai OCR

$$OCR = \frac{10 * 27}{100} = 2.7$$

Data DC dibulatkan lagi menjadi 2 lalu disimpan di register OCR.

Terdapat beberapa pilihan konstanta N di (2) dan akan mempengaruhi nilai OCR dan ICR seperti terlihat pada Tabel 1.

Tabel 1. Nilai OCR dan ICR pada setiap N untuk frekuensi 1 kHz dan DC 27 %

N	ICR	ICR (pembulatan)	OCR pada DC 27%	OCR (pembulatan)
256	45.875	45	12.15	12
64	186.5	186	50.22	50
8	1499	1499	404.73	404
1	11999	11999	3239.73	3239

Data pada Tabel 2 menampilkan pengaruh nilai N terhadap frekuensi PWM yang akan dihasilkan.

Tabel 2. Variasi N terhadap frekuensi PWM

N	ICR	Frekuensi (Hz)
256	45	1019.02
64	186	1002.67
8	1499	1000
1	11999	1000

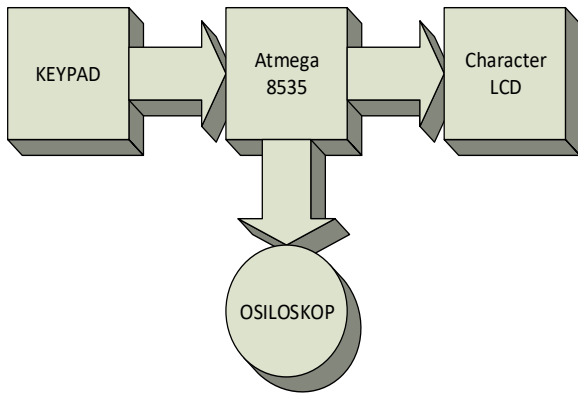
Nilai OCR hasil pembulatan diuji kembali menggunakan (3) untuk melihat perbedaan DC yang akan diperoleh di output pada setiap nilai N dan datanya tersaji pada Tabel 3.

Tabel 3. Pengaruh pembulatan OCR terhadap nilai DC

N	OCR	ICR	DC (%)
256	45	12	26.667
64	186	50	26.88
8	1499	404	26.95
1	11999	3239	26.993

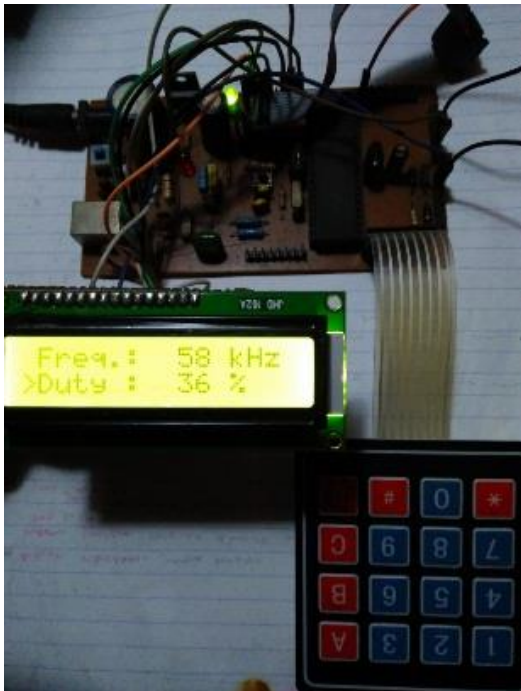
Hasil pada Tabel 2 dan Tabel 3 menunjukkan bahwa nilai N yang terkecil yaitu 1 memberikan frekuensi dan DC PWM mendekati dengan yang diinginkan yaitu 1 kHz dan 27 % sehingga nilai OCR yang digunakan adalah 11999 dan OCR adalah 3239. Catatan penting memilih nilai N yang optimal adalah yang terkecil pada saat nilai ICR dan OCR tidak melebihi 65535 (0xFFFF).

Salah satu varian mikrokontroler ATmega yaitu ATmega8535 digunakan untuk membangun generator PWM ini dimana diagram bloknya ditampilkan oleh Gambar 5 sedangkan perangkat kerasnya di Gambar 6.



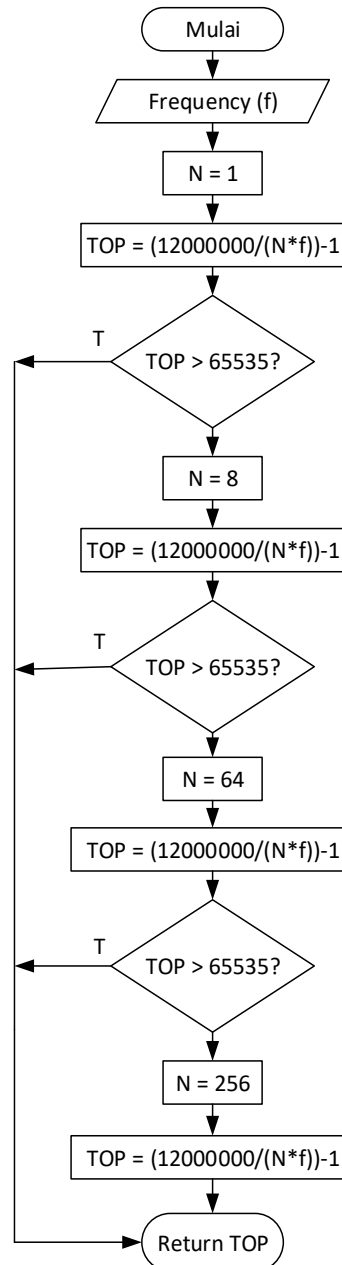
Gambar 5 Diagram blok generator PWM berbasis Mikrokontroler AVR ATmega

Keypad berfungsi untuk memasukkan nilai frekuensi dan DC yang diinginkan sedangkan display menampilkan data yang sedang dimasukkan. Output sinyal PWM ditampilkan di osiloskop.



Gambar 6. Perangkat keras generator PWM berbasis Mikrokontroler AVR ATmega

Mikrokontroler ATmega8535 diprogram dengan urutan instruksi mengambil data dari keypad dan menampilkan di display lalu mengeluarkan sinyal di pin OC. Sedangkan algoritma penentuan nilai OCR dan ICR serta N ditunjukkan pada Gambar 7.

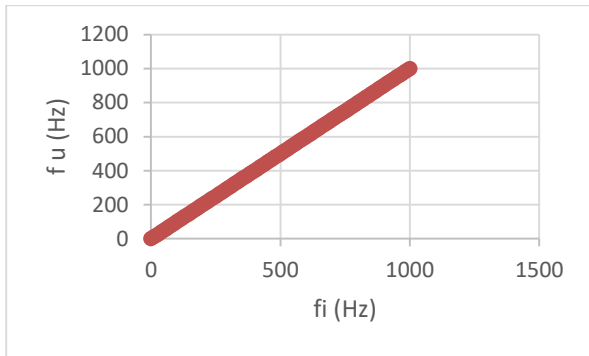


Gambar 7. Algoritma penentu nilai OCR, ICR dan N

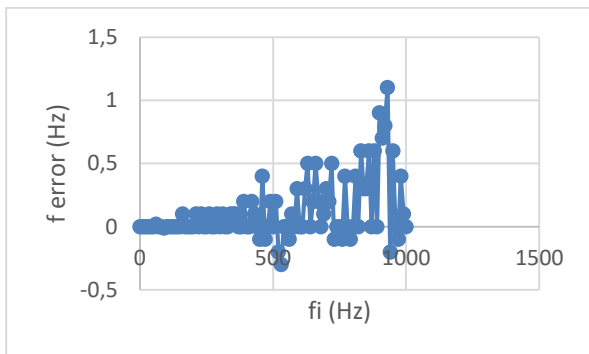
Nilai frekuensi yang dimasukkan dari keypad menjadi variable f yang dikalkulasi dengan N terkecil dahulu untuk mendapatkan nilai TOP, jika melampaui 65535 kalkulasi diulang lagi dengan N yang lebih besar hingga diperoleh TOP yang lebih kecil dari batas yang diinginkan.

III. HASIL DAN PEMBAHASAN

Pengukuran menggunakan osiloskop dilakukan untuk menguji seberapa dekat frekuensi yang diinginkan (fdes) dengan frekuensi terukur (fact). Gambar 8 menunjukkan pengujian dalam rentang 0 s.d. 1 kHz, terlihat perbandingan yang linier, hal ini karena selisih nilai yang kecil antara keduanya. Perbedaan (error) secara detail tertera pada Gambar 9.

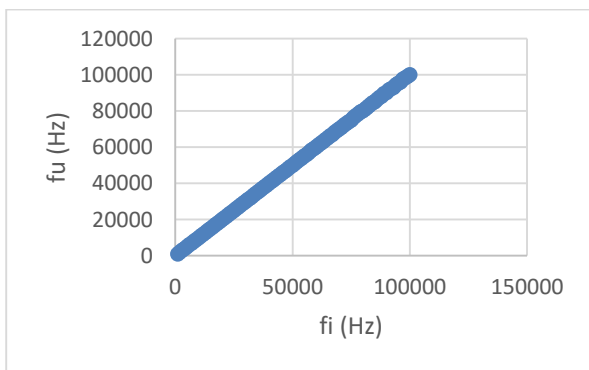


Gambar 8. Perbandingan frekuensi yang diinginkan (f_i) dengan frekuensi yang diukur (f_u) pada rentang 0 s.d. 1 kHz

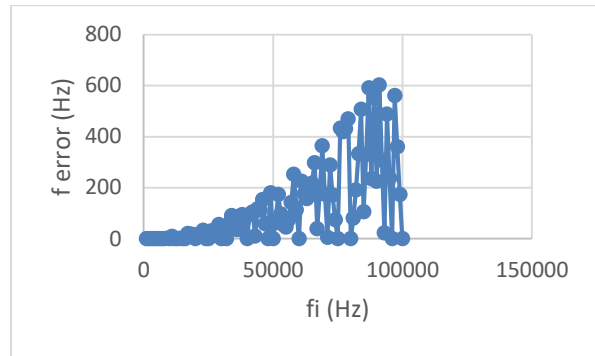


Gambar 9. Nilai error frekuensi yang diinginkan (f_i) dengan frekuensi yang diukur pada rentang 0 s.d. 1 kHz

Perbedaan yang signifikan terjadi ketika frekuensi berada dalam orde ribuan Hz. Grafik Gambar 10 terlihat linier karena rasio error dengan skala grafik jauh perbedaannya oleh karena itu Gambar 11 memperjelas besarnya error yang terjadi di setiap nilai frekuensi.



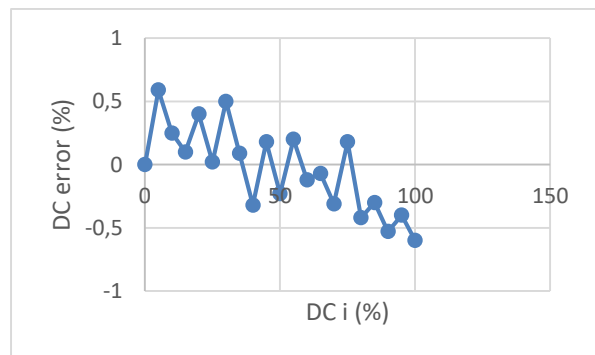
Gambar 10. Perbandingan frekuensi yang diinginkan (f_i) dengan frekuensi yang diukur (f_u) pada rentang 0 s.d. 100 kHz



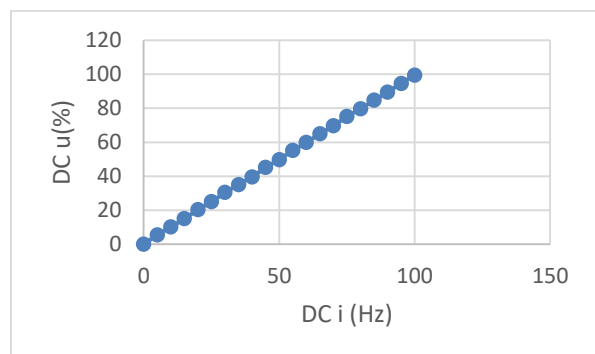
Gambar 11. Nilai error frekuensi yang diinginkan (f_i) dengan frekuensi yang diukur pada rentang 0 s.d. 100 kHz

Kecenderungan semakin besar frekuensi mengakibatkan semakin tinggi error disebabkan oleh nilai frekuensi yang besar membuat kecilnya nilai ICR walaupun nilai N sudah paling minimal diberikan. Ada juga beberapa frekuensi tertentu yg tidak memiliki error, ini karena nilai ICR yang didapatkan dari kalkulasi adalah bilangan bulat.

Hasil pengukuran DC menunjukkan error yang cukup kecil untuk semua frekuensi berkisar tidak lebih dari 1 % seperti disajikan di Gambar 12 sehingga secara keseluruhan DC sudah cukup linier yang terlihat pada Gambar 13.

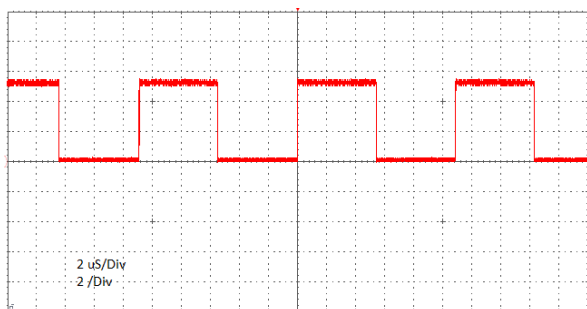


Gambar 12. Nilai error DCi yang diinginkan (DCi) dengan DC yang diukur pada rentang 0 s.d. 100 kHz

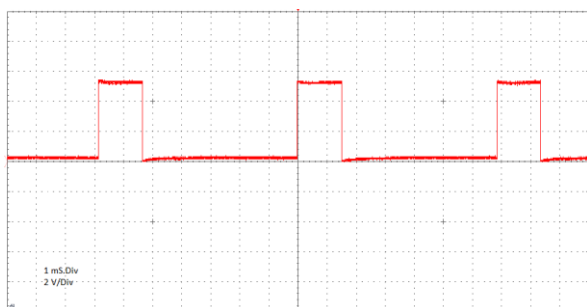


Gambar 13. Perbandingan DC (DCi) yang diinginkan dengan DC yang diukur (DCu)

Gambar 14 dan Gambar 15 menampilkan dua sampel pengukuran frekuensi dan DC yang terukur di osiloskop yaitu untuk frekuensi 91 kHz dan 146 Hz dalam DC yang berbeda.



Gambar 14. Hasil pengukuran frekuensi 91 kHz dan DC 49.54 %



Gambar 15. hasil pengukuran frekuensi 146 Hz dan DC 22.4 %

IV. KESIMPULAN

Semakin tinggi frekuensi PWM maka semakin besar nilai perbedaan frekuensi yang diukur dengan yang diinginkan, ini disebabkan oleh nilai frekuensi yang besar membuat kecilnya nilai yang ada di dalam register ICR walaupun nilai pembagi frekuensinya sudah paling minimal diberikan. Sementara selisih antara pengukuran dan perhitungan pada Duty Cycle memberikan hasil cukup kecil di bawah 1 %. Selisih ini juga terjadi akibat dari pembulatan nilai ICR dan OCR, namun di frekuensi 20 kHz dan Duty Cycle 25% dimana nilai ICR adalah 599 dan OCR adalah 116 menghasilkan selisih frekuensi dan Duty Cycle yang paling minimum. Dari percobaan yang telah dilakukan, rancang bangun PWM generator berbasis mikrokontroler AVR ATmega berhasil diwujudkan.

REFERENSI

- [1] L. C.-H. 5. Lin Bor-Ren, Analysis, design and implementation of an interleaved three-level PWM DC/DC ZVS converter, *Int. J. Electron.*, vol. 2, p. 103, 2016.
- [2] S. Bhattacharya, P. Deb, S. K. Biswas, and S. Karchowdhury, Performance and design of an open-delta connected grid tied bidirectional PWM converter, *Int. J. Electr. Power Energy Syst.*, vol. 78, no. 78, pp. 183–193, 2016, doi: 10.1016/j.ijepes.2015.11.079.
- [3] W. C. 7. ZHA Guo-xiang, MA Yong, Design of PWM generator based on FPGA, vol. 8, no. 48, p. 135, 2016.
- [4] Z. Q. 8. WU Cai-yuan, ZHOU Hua, Design of Multichannel PWM to DAC Module Based on Dual Communication, *Mech. Electr. Eng. Technol.*, vol. 6, no. 46, p. 7, 2017.
- [5] X. Zhang, T. Qu, D. Yin, and Y. Liu, Design and Implementation of PWM Generator Based on OMAP-L138, *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 381, no. 1, 2018, doi: 10.1088/1757-899X/381/1/012137.
- [6] S. Rao, S. R. Mn, S. N. Syed, and S. K. Tummala, Gate driver circuit design, PWM signal generation using FEZ Panda III and Arduino for inverter, *E3S Web Conf.*, vol. 87, no. 201 9, 2019, doi: 10.1051/e3sconf/20198701003.
- [7] S. Suneeta, R. Srinivasan, and R. sagar, Generation of Variable Duty Cycle PWM using FPGA, *IOSR J. VLSI Signal Process.*, vol. 4, no. 6, pp. 01–03, 2014, doi: 10.9790/4200-04620103.
- [8] C. Hung, W. Liu, H. Su, J. Chen, and B. Chiu, PIC-Based Multi-Channel PWM Signal Generation Method and Application to Motion Control of Six Feet Robot Toy, *Int. J. Circuits, Syst. Signal Process.*, vol. 3, no. 2, pp. 73–81, 2009.
- [9] Atmel Corporation, ATmega8535 8-bit Microcontroller with 8K Bytes In-System Programmable Flash, *Datasheet*. p. 87, 2003.