

Improved Sampling in Carrier-Based Discontinuous SVPWM Simulation

Peningkatan *Sampling* pada Simulasi SVPWM *Discontinuous* Berbasis Pembawa

Okik Surikno¹, Tole Sutikno²

¹ Mahasiswa Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Indonesia

² Dosen Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Indonesia

INFORMASI ARTIKEL

Riwayat Artikel:

Dikirimkan 14 Februari 2020,
Direvisi 28 September 2020,
Diterima 28 September 2020.

Kata Kunci:

SVPWM,
Discontinuous,
Sinyal referensi,
Sampling,
FPGA,
Quartus II.

Penulis Korespondensi:

Okik Surikno, Tole Sutikno
Universitas Ahmad Dahlan
Kampus 4 UAD, Jln. Ring
Road Selatan, Tamanan,
Banguntapan, D.I. Yogyakarta,
Indonesia.

Surel/Email:

okiksurikno02@gmail.com
tole@ee.uad.ac.id

ABSTRACT / ABSTRAK

In this study, a circuit was designed to improve sampling in discontinuous SVPWM simulations using Quartus II software. The v_beta_sin unit was successfully simulated using the waveform editor. The results displayed are in accordance with previous calculations, if the input is 000000001 then the output is 100000011 and so on. The v_alfa_cos unit is also successfully simulated using the waveform editor. The simulation results if the input is 000000001 then the output is 111111111 and so on, according to the previous calculation. The unit counter was successfully simulated using the block diagram in Quartus II. The output of this unit counter is in the form of 9 bits to retrieve v_beta_sin and v_alfa_cos data. The v_beta_sin, v_alfa_cos and counter units used as reference signals are successfully simulated and can be used as a supporting circuit in the simulation of the discontinuous SVPWM method. The results of the simulation show an increase in sampling or sampling by 512.

Pada penelitian ini dirancang rangkaian untuk meningkatkan *sampling* pada simulasi SVPWM *discontinuous* dengan menggunakan perangkat lunak Quartus II. Unit v_beta_sin berhasil di simulasikan menggunakan waveform editor. Hasil yang ditampilkan sesuai dengan perhitungan sebelumnya, jika inputannya 000000001 maka outputnya 100000011 dan seterusnya. Unit v_alfa_cos juga berhasil di simulasikan menggunakan waveform editor. Hasil simulasinya jika inputnya 000000001 maka outputnya 111111111 dan seterusnya, sesuai dengan perhitungan sebelumnya. Unit counter berhasil di simulasikan menggunakan diagram blok pada Quartus II. Keluaran dari unit counter ini berupa cacahan 9 bit untuk mengambil data v_beta_sin dan v_alfa_cos. Unit v_beta_sin, v_alfa_cos dan counter yang digunakan sebagai Sinyal referensi berhasil disimulasikan dan dapat digunakan sebagai rangkaian pendukung pada simulasi metode SVPWM *discontinuous*. Hasil dari simulasi menunjukkan peningkatan *sampling* atau pencuplikan sebesar 512.

This work is licensed under a [Creative Commons Attribution-Share Alike 4.0](https://creativecommons.org/licenses/by-sa/4.0/)



Sitasi Dokumen ini:

Okik Surikno and Tole Sutikno, "Improved Sampling in Carrier-Based Discontinuous SVPWM Simulation," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 2, no. 3, pp. 137-144, 2020. DOI: [10.12928/biste.v2i3.1753](https://doi.org/10.12928/biste.v2i3.1753)

1. PENDAHULUAN

Teknik modulasi SPWM adalah teknik modulasi dengan membandingkan antara gelombang sinusoida sebagai sinyal referensi dan gelombang segitiga sebagai sinyal pembawa untuk memperoleh sinyal PWM yang digunakan pada proses *switching* [1]. Dasar dari metode SVPWM (*Space Vector Pulse Width Modulation*) berbeda dengan metode SPWM (*Sinusoidal Pulse Width Modulation*). SVPWM bertujuan untuk mencapai bentuk gelombang tegangan sinus tiga fase dari tegangan dan frekuensi yang dapat disesuaikan, sementara SVPWM menggunakan inverter dan motor secara keseluruhan menggunakan delapan tegangan vektor untuk mendapatkan frekuensi, tegangan dan kecepatan [2].

Dalam teknik modulasi persyaratan yang penting adalah mendapatkan output daya yang lebih tinggi dan efisien untuk berbagai kontrol tegangan keluaran inverter [3]. Metode *Space Vector Pulse Width Modulation* (SVPWM) merupakan metode yang terbaik di antara semua teknik PWM dalam penerapan drive frekuensi variabel, SVPWM lebih fleksibel serta mudah diimplementasikan dengan DSP (*Digital Signal Processing*) atau FPGA (*Field Programmable Gate Array* [4] dan [5]. Fleksibilitas ini didasari dengan SVPWM yang hanya membutuhkan tegangan referensi tunggal daripada tiga referensi tegangan yang diperlukan dalam SPWM, dan tegangan inverter *output* dalam SVPWM dapat secara bertahap dikendalikan dari PWM ke mode *six step*. Selain itu dibandingkan dengan metode SPWM, Metode SVPWM lebih unggul dalam banyak aspek yaitu indeks modulasi lebih tinggi, tegangan output 15% lebih tinggi, serta harmonik arus dan torsi yang dihasilkan jauh lebih sedikit [6].

Ada 2 Metode SVPWM yang sudah banyak diimplementasikan, yaitu metode SVPWM *continuous* dan SVPWM *discontinuous*. SVPWM *continuous* memiliki kekurangan seperti beban komputasi yang tinggi, kinerja yang lebih rendah pada indeks modulasi tinggi dan kerugian *switching* inverter yang tinggi. Jadi untuk mengurangi kerugian *switching* yang tinggi dan meningkatkan kinerja di daerah indeks modulasi tinggi, maka dalam penelitian ini akan menggunakan metode SVPWM *discontinuous*.

SVPWM *discontinuous* adalah teknik modulasi yang menawarkan kerugian *switching* yang lebih rendah dan algoritma yang lebih sederhana dibandingkan dengan SVPWM *continuous* [7]. Pada saat yang sama, frekuensi *switching* rata-rata, atau frekuensi *switching* 1,5 kali lebih tinggi daripada yang digunakan dalam SVPWM *continuous*, SVPWM *discontinuous* menghasilkan distorsi harmonik arus yang lebih rendah daripada yang diperoleh dalam SVPWM *continuous* pada indeks modulasi tinggi [8] dan [9]. Metode SVPWM *discontinuous* menggunakan enam vektor ruang aktif [10]. Keenam vektor aktif tersebut menghasilkan tegangan output sinusoidal dan harmonik arus rendah. Dalam penelitian ini akan memverifikasi metode yang disederhanakan dan mendefinisikan sektor vektor tegangan ruang. Ini akan menunjukkan bahwa pendekatan sederhana untuk merancang SVPWM *discontinuous* menghasilkan kecepatan tinggi dalam komputasi keseluruhan algoritma sehingga menghasilkan frekuensi *switching* yang tinggi dan distorsi harmonik arus yang rendah dalam sistem penggerak motor induksi.

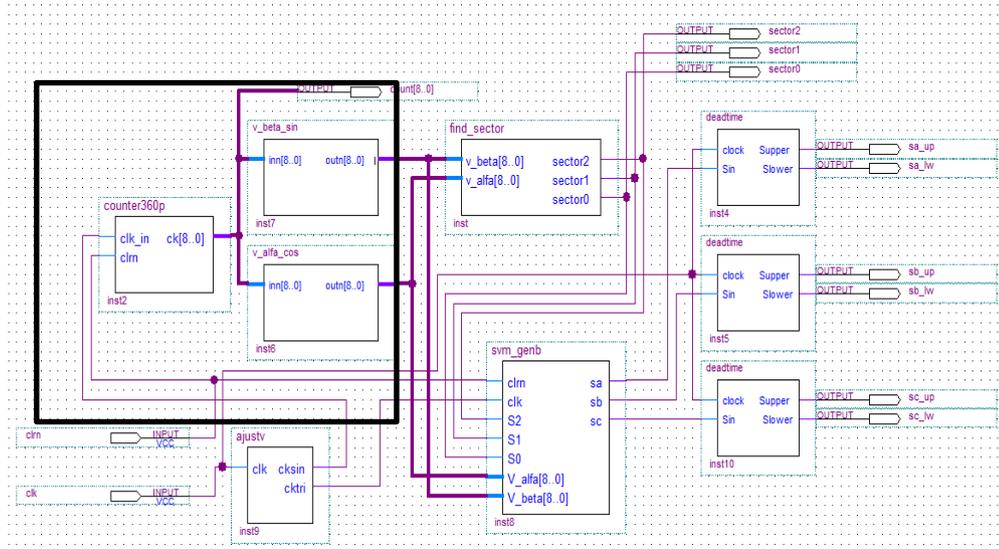
2. METODE PENELITIAN

Pada bagian ini menjelaskan mengenai peningkatan sampling pada simulasi SVPWM *discontinuous* berbasis pembawa. Setelah menentukan metode yang digunakan, maka dilanjutkan dengan perancangan untuk mensimulasikan dan mengumpulkan data. Proses perancangan dan simulasi pada penelitian ini menggunakan perangkat lunak Quartus II. Rancangan dari metode SVPWM *discontinuous* terdiri dari beberapa rangkaian proyek pendukung diantaranya adalah: *ajust*, *counter*, *v_beta_sin*, *v_alfa_cos*, *find sector*, *svm gen*, dan *deadtime*. Peneliti tidak membuat semua rangkaian tersebut, melainkan menggunakan beberapa rangkaian yang sudah ada. Rangkaian yang dibuat peneliti yaitu rangkaian sinyal referensi yang terdiri dari *v_beta_sin*, *v_alfa_cos* dan *counter*.

2.1. Desain sistem

Rancangan dari metode SVPWM *discontinuous* terdiri dari beberapa rangkaian proyek pendukung diantaranya adalah: *ajust*, *counter*, *v_beta_sin*, *v_alfa_cos*, *find sector*, *svm gen*, dan *deadtime*. Peneliti tidak membuat semua rangkaian tersebut, melainkan menggunakan beberapa rangkaian yang sudah ada. Rangkaian yang dibuat peneliti yaitu rangkaian sinyal referensi yang terdiri dari *v_beta_sin*, *v_alfa_cos* dan *counter* yang berada dalam kotak hitam. Skema keseluruhan dari desain SVPWM *discontinuous* ditunjukkan berdasarkan Gambar 1.

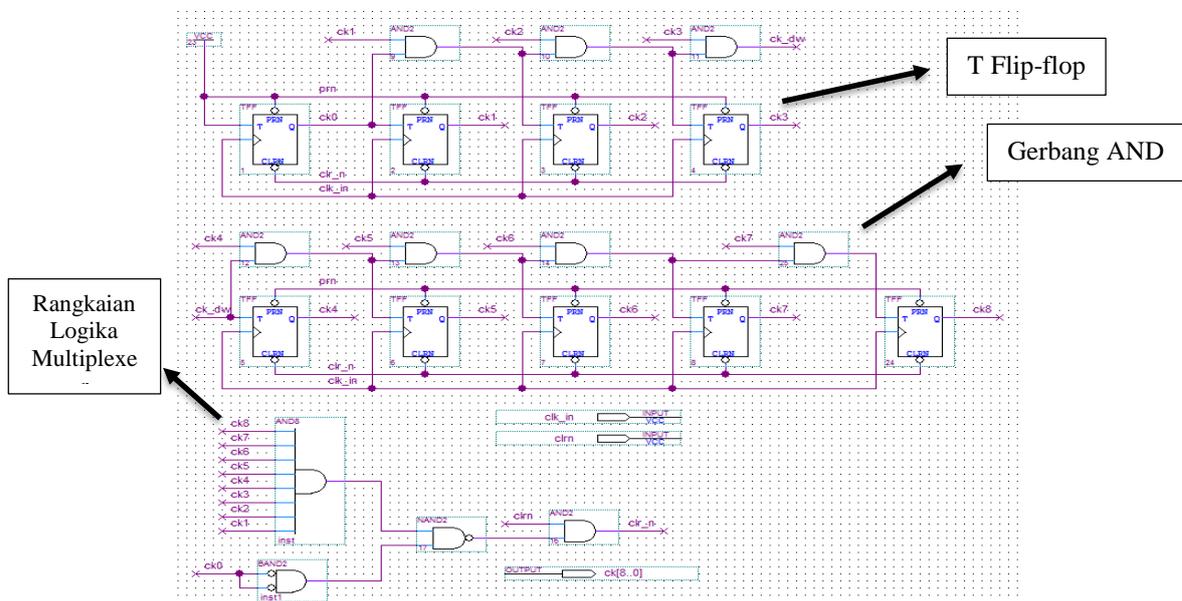
Skema rangkaian yang dibuat oleh peneliti adalah *counter*, *v_alfa_cos*, dan *v_beta_sin*. Rangkaian *counter* yang sebelumnya 8 bit dibuat menjadi 9 bit, untuk rangkaian *v_alfa_cos* dan *v_beta_sin* juga dibuat menjadi 9 bit dengan mengambil nilai pencuplikan 512 yang sebelumnya nilai pencuplikan 360.



Gambar 1. Skema Keseluruhan dari Desain SVPWM Discontinuous

2.2. Rangkaian Counter

Rangkaian counter 9 bit ini tersusun dari 9 buah T Flip-flop. T Flip-flop merupakan rangkaian flip-flop yang telah dibuat dengan menggunakan flip-flop J-K yang kedua inputnya dihubungkan menjadi satu maka akan diperoleh flip-flop yang memiliki watak membalik *output* sebelumnya jika *input* tinggi dan *output* akan tetap jika *input* rendah. Dalam satu periode dibutuhkan cuplikan sebanyak 512 bagian, sehingga menggunakan cacahan dari 0-511 dan dibutuhkan pencacah mod-512. Rangkaian counter ini memperoleh dua masukan atau input, yaitu sinyal clock dan sinyal clear untuk mereset ulang cacahan setelah mencapai 511 untuk kembali ke 0. Keluaran dari rangkaian counter/ pencacah ini berupa cacahan 9 bit untuk mengambil data *v_alfa_cos* dan *v_beta_sin*. Rangkaian counter atau pencacah 9 bit ditunjukkan berdasarkan Gambar 2.



Gambar 2. Rangkaian Counter 9 bit Modulus 512

2.3. Rangkaian v_beta_sin

Rangkaian *v_beta_sin* dihasilkan melalui fungsi sinus dengan pemetaan memori 512 entri, karena dalam desain ini menggunakan 512 alamat pemetaan memori, maka akan menggunakan counter mod-512. Sebelum membuat rangkaian *v_beta_sin* pada text editor Quartus, terlebih dahulu dibuat nilai alamat sinus dan nilai keluaran memori sin. Nilai alamat sinus dibuat dari nilai 0 sampai 511 dalam angka biner karena untuk lebar pencuplikan 512. Nilai keluaran memori sinus diperoleh dengan suatu persamaan (1):

$$\text{Memori sinus}(y) = \sin\left(\frac{\text{alamat}(x) * \text{satu periode sinus}}{\text{lebar pencuplikan}}\right)$$

dengan, x bernilai 0 sampai 511, satu periode sinus adalah 360 dan lebar pencuplikan adalah 512.

Berdasarkan persamaan tersebut, dibuat nilai perhitungan keluaran memori sin. Hasil perhitungan nilai alamat dan nilai memori sin yang menggunakan persamaan (1) ditunjukkan berdasarkan Tabel 1. Nilai alamat dan nilai memori sin dibuat dalam bentuk biner agar lebih memudahkan untuk pengkodean pada *text editor* Quartus. Nilai alamat biner dibuat sebanyak 512 cuplikan mulai dari angka 0 hingga 511 yang diubah dalam bentuk biner sebesar 9 bit. Sedangkan nilai keluaran memori sinus dilakukan perhitungan menggunakan persamaan (1) diperoleh hasil berupa nilai positif dan negatif. Sehingga setelah nilai *memori sinus* (y) diubah ke bentuk biner maka ditambahkan satu bit depan sebagai bit tanda untuk bilangan tersebut bernilai positif atau negatif. Memori sinus (y) dibuat besarnya 9 bit, dengan bit depan sebagai bit *signed* dan 8 bit selanjutnya sebagai bit *value*. Setelah nilai alamat dan nilai memori sinus (y) diperoleh, maka dibuat program di *text editor* Quartus.

Tabel 1. Perhitungan Nilai Alamat Sinus(X) dan Nilai Memori Sinus(Y)

Nilai alamat atau x (desimal)	nilai alamat atau x (biner)	y atau $\frac{\text{Alamat}(x) * 360}{512}$	nilai memori sinus(y) (derajat)	Nilai memori sinus(y) (biner)
0	000000000	0	0	100000000
1	000000001	0,703125	0,012271538	100000011
2	000000010	1,40625	0,024541229	100000110
3	000000011	2,109375	0,036807223	100001001
4	000000100	2,8125	0,049067674	100001100
5	000000101	3,515625	0,061320736	100001111
6	000000110	4,21875	0,073564564	100010010
7	000000111	4,921875	0,085797312	100010101
8	000001000	5,625	0,09801714	100011001
9	000001001	6,328125	0,110222207	100011100
10	000001010	7,03125	0,122410675	100011111
...
511	111111111	359,296875	0,012271538	000000011

2.4. Rangkaian v_alfa_cos

Rangkaian v_alfa_cos dihasilkan melalui fungsi cosinus dengan pemetaan memori 512 entri. Nilai alamat cosinus dibuat dari nilai 0 sampai 511 dalam angka biner karena untuk lebar pencuplikan 512. Nilai keluaran memori cosinus diperoleh dengan suatu persamaan sebagai berikut:

$$\text{Memori sinus}(y) = \sin\left(\frac{\text{alamat}(x) * \text{satu periode cosinus}}{\text{lebar pencuplikan}}\right)$$

dengan x bernilai 0 sampai 511, satu periode sinus adalah 360 dan lebar pencuplikan sebsar 512. Berdasarkan persamaan tersebut, dibuat nilai perhitungan keluaran memori cosinus. Hasil perhitungan nilai alamat dan nilai memori cosinus yang menggunakan persamaan (2) ditunjukkan berdasarkan Tabel 2.

Tabel 2. Perhitungan Nilai Alamat Cosinus(X) dan Nilai Memori Cosinus(Y)

Nilai alamat atau x (desimal)	nilai alamat atau x (biner)	y atau $\frac{\text{Alamat}(x) * 360}{512}$	nilai memori cosinus(y) (derajat)	Nilai memori cosinus(y) (biner)
0	000000000	0	1	100000000
1	000000001	0,703125	0,999924702	111111111
2	000000010	1,40625	0,999698819	111111111

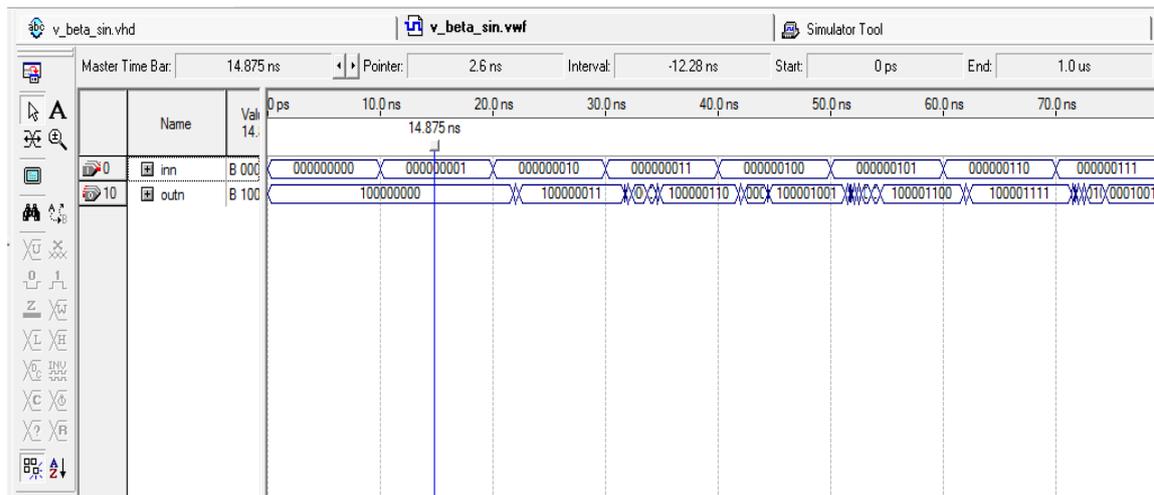
Nilai alamat atau x (desimal)	nilai alamat atau x (biner)	y atau $\frac{Alamat(x) * 360}{512}$	nilai memori cosinus(y) (derajat)	Nilai memori cosinus(y) (biner)
3	00000011	2,109375	0,999322385	111111111
4	00000100	2,8125	0,998795456	111111111
5	00000101	3,515625	0,998118113	111111111
6	00000110	4,21875	0,997290457	111111111
7	00000111	4,921875	0,996312612	111111111
8	00001000	5,625	0,995184727	111111110
9	00001001	6,328125	0,99390697	111111110
10	00001010	7,03125	0,992479535	111111110
...
511	111111111	359,296875	0,999924702	111111111

3. HASIL DAN PEMBAHASAN

Setelah melakukan perancangan rangkaian menggunakan perangkat lunak Quartus II 8.0, diperoleh hasil pengamatan berupa sinyal keluaran yang dihasilkan dari simulasi rangkaian pada perangkat lunak. Simulasi pada perangkat lunak Quartus II dilakukan pada setiap unit pembangun rangkaian SVPWM dan juga simulasi untuk keseluruhan rangkaian.

3.1. Simulasi Unit V_Beta_Sin

Simulasi dari unit *v_beta_sin* telah berhasil dilakukan dengan menggunakan perangkat lunak Quartus II dan tidak terdapat *error*. Total *logic elements* sebesar 168 dari 33.216 atau kurang dari 1%, total pin 18 dari 475 atau 4%, total bit memori 0% dan total PLLs 0%. Karena ukuran bit memori dan PLLs dari proyek Rangkaian *v_beta_sin* yang sangat kecil sehingga nilainya 0%. Hasil dari simulasi unit *v_beta_sin* seperti yang ditunjukkan pada Gambar 3.

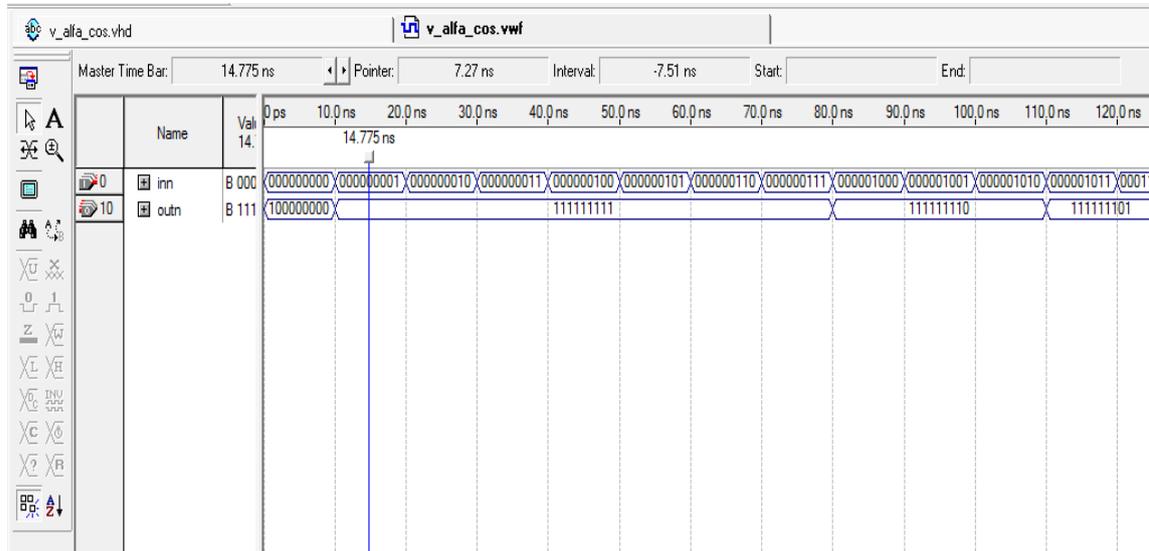


Gambar 3. Hasil Kompilasi Unit *v_beta_sin*

Hasil simulasi unit *v_beta_sin* yang ditampilkan telah sesuai dengan rancangan program yang telah dibuat. Hasil *input* dan *output* ditampilkan dalam bilangan biner, jika *input* 00000000 maka *output* 10000000, jika *input* 00000001 *output* 10000011 dan seterusnya sampai alamat 511. Hasil simulasi yang ditunjukkan sesuai dengan perhitungan nilai alamat sinus dan nilai memori sinus yang berada dalam tabel sebelumnya. Dari hasil simulasi tersebut unit *v_beta_sin* dapat digunakan sebagai rangkain pendukung untuk peningkatan *sampling* pada simulasi SVPWM *discontinuous* berbasis pembawa.

3.2. Simulasi Unit V_Alfa_Cos

Proses simulasi unit *v_alfa_cos* sama dengan proses kompilasi unit *v_beta_sin*. Simulasi dari unit *v_alfa_cos* telah berhasil dilakukan dengan menggunakan perangkat lunak Quartus II dan tidak terdapat *error*. Total *logic elements* sebesar 171 dari 33.216 atau kurang dari 1%, total pin 18 dari 475 atau 4%, total bit memori 0% dan total PLLs 0%. Hasil simulasi seperti yang ditunjukkan oleh Gambar 4.

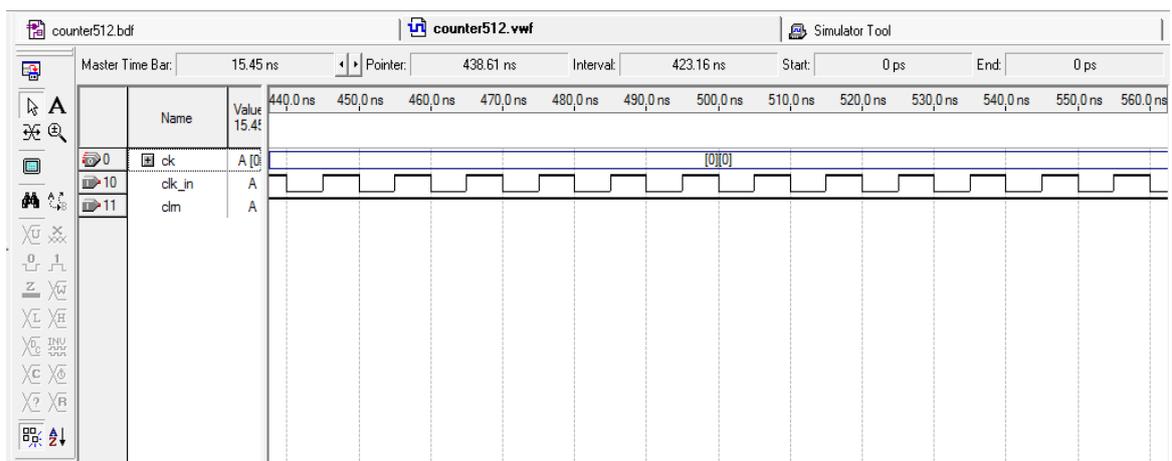


Gambar 4. Hasil Kompilasi Unit *v_alfa_cos*

Hasil *input* dan *output* ditampilkan dalam bilangan biner, ketika *input* 00000000 maka *output* 10000000, ketika *input* 00000001 *output* 11111111 dan seterusnya sampai alamat 511. Hasil simulasi yang ditunjukkan sesuai dengan perhitungan nilai alamat cosinus dan nilai memori cosinus yang berada dalam tabel sebelumnya. Dari hasil simulasi tersebut unit *v_beta_sin* dapat digunakan sebagai rangkain pendukung untuk peningkatan *sampling* pada simulasi *SVPWM discontinuous* berbasis pembawa.

3.3. Simulasi Unit Pencacah atau Counter

Unit pencacah atau counter yaitu unit yang berfungsi sebagai penghasil cacahan untuk mengambil data. Dalam satu periode pengambilan data diperlukan 512 cacahan, setelah selesai membuat projek atau program maka dilakukan kompilasi dan dilanjutkan dengan melakukan simulasi untuk mengetahui hasil dari program yang telah dibuat pada perangkat lunak Quartus. Hasil simulasi unit pencacah atau counter ditunjukkan pada Gambar 5.



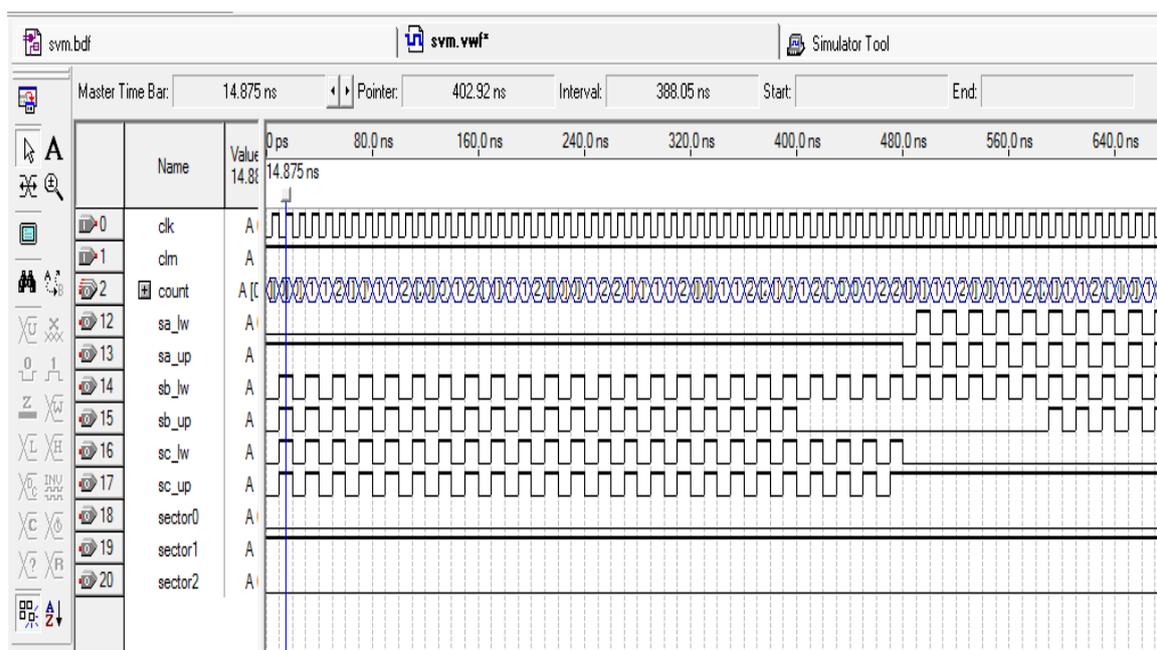
Gambar 5. Hasil Simulasi Unit Pencacah atau Counter

Hasil simulasi unit *counter* yang ditampilkan telah sesuai dengan rancangan program yang telah dibuat. Rangkaian counter ini terdapat 2 masukan atau *input* yaitu berupa sinyal clock dan sinyal clear untuk mereset

ulang *cacahan* setelah mencapai 511 untuk kembali ke 0. Dari hasil simulasi tersebut unit counter dapat digunakan sebagai rangkaian pendukung untuk peningkatan *sampling* pada simulasi SVPWM *discontinuous* berbasis pembawa.

3.4. Simulasi Metode SVPWM Secara Keseluruhan

Setelah semua projek pendukung dirangkai, kemudian dilakukan proses kompilasi. Proses kompilasi telah berhasil dilakukan dengan menggunakan total *logic elements* (Les) sebesar 1635 dari 33216 atau sebesar 5%. Total register yang digunakan sebesar 163 dan total pin terpakai 23 dari 475. Hasil simulasi peningkatan *sampling* SVPWM *discontinuous* telah berhasil seperti yang ditunjukkan pada Gambar 6.



Gambar 6. Hasil Simulasi Peningkatan Sampling SVPWM Discontinuous

Sinyal keluaran dari simulasi metode SVPWM secara keseluruhan telah berhasil disimulasikan menggunakan perangkat lunak Quartus II, dan berhasil meningkatkan *sampling* dari penelitian sebelumnya yang 360 menjadi 512.

4. KESIMPULAN

Rangkaian unit *v_beta_sin*, dan unit *v_alfa_cos* sebagai masukan sinyal referensi telah berhasil disimulasikan dan dapat digunakan sebagai rangkaian pendukung pada simulasi metode SVPWM *discontinuous*. Peningkatan *sampling* yang diusulkan pada simulasi SVPWM *discontinuous* adalah sebesar 512.

UCAPAN TERIMA KASIH

Sampaikan ucapan terima kasih kepada editor dan reviewer atas saran, masukan serta bantuan dalam penerbitan naskah. Ucapan terima kasih juga ditunjukkan kepada pihak-pihak yang telah mendukung penelitian.

REFERENSI

- [1] B. A. Yomahudaya and T. Sutikno, "Pembangkit Sinyal SPWM untuk Multilevel Inverter Satu Fasa Lima Tingkat Berbasis Mikrokontroler At-Mega32," *Jurnal Ilmu Teknik Elektro Komputer dan Informatika*, vol. 3-2, pp. 73-81, 2017. DOI: [10.26555/jiteki.v3i2.5927](https://doi.org/10.26555/jiteki.v3i2.5927)
- [2] Z. G. Wang, *et al.*, "SVPWM Techniques and Applications in HTS PMSM Machines Control," *Journal Of Electronic Science And Technology Of China*, vol. 6-2, pp. 191-197, 2008.
- [3] T. Sutikno, A. Jidin, N. R. N. Idris, "New approach FPGA-based implementation of discontinuous SVPWM," *Turk J Elec Eng & Comp Sci*, vol. 18, no. 4, 2010. DOI: [10.3906/elk-0906-21](https://doi.org/10.3906/elk-0906-21)
- [4] A. Belkheiri, S. Aoughellanet and M. Belkheiri, "FPGA Implementation of a Space Vector Pulse Width Modulation Technique for a Two-Level Inverter," *Elektrotehniški Vestnik*, vol. 85, no. 3, pp. 1-7, 2018. [Online](#)
- [5] T. Sutikno, W. J. Hwa, A. Jidin and N. R. N. Idris "A Simple Approach of Space-vector Pulse Width Modulation Realization Based on Field Programmable Gate Array," *Electric Power Components and Systems*, vol. 38, pp. 1546-1557, 2010. DOI: [10.1080/15325008.2010.482092](https://doi.org/10.1080/15325008.2010.482092)

- [6] B. Rashidi and M. Sabahi, "High Performance FPGA Based Digital Space Vector PWM Three Phase Voltage Source Inverter," *I.J.Modern Education and Computer Science*, vol. 1, pp. 62-71, 2013. DOI: [10.5815/ijmecs.2013.01.08](https://doi.org/10.5815/ijmecs.2013.01.08)
- [7] T. Sutikno, A. Jidin, and M. F. Basar, "Simple Realization of 5-Segment Discontinuous SVPWM Based on FPGA," *International Journal of Computer and Electrical Engineering*, vol. 2-1, pp. 1793-8163, 2010. [Online](#)
- [8] T. Sutikno, N. R. N. Idris, A. Jidin, M. H. Jopri, "FPGA Based Optimized Discontinuous SVPWM Algorithm for Three Phase VSI in AC Drives," *International Journal of Power Electronics and Drive System*, vol. 3-2, pp. 228-240, 2013. DOI: [10.11591/ijpeds.v3i1.735](https://doi.org/10.11591/ijpeds.v3i1.735)
- [9] M. A. Khan and A. Iqbal, A. R. Haitham "Investigation of Discontinuous Space Vector PWM Techniques of a Three-phase Voltage Source Inverter," *i-manager's Journal on Electrical Engineering*, vol. 2-3, pp. 60-71, 2009. [Online](#)
- [10] M. A. Khan, A. Iqbal, Sk M. Ahmad, Z. husain "Analysis of Discontinuous Space Vector PWM Techniques for a Seven-Phase Voltage Source Inverter," *International Journal of Power Electronics and Drive System*, vol. 2-2, pp. 203-218, 2012. [Online](#)

BIOGRAFI PENULIS



Okik Surikno

Lahir di Lamongan, 02 juli 1994, telah menyelesaikan program studi S1 Teknik Elektro di Universitas Ahmad Dahlan angkatan tahun 2013 dan telah menyelesaikan pendidikan tersebut pada tahun 2019 bidang peminatannya adalah Otomasi Industri.



Tole Sutikno

Lahir di Lamongan, 12 juli 1975, beliau telah menyelesaikan kuliah S1 Teknik Elektro di UNDIP Semarang, S2 Teknik Elektro di UGM Yogyakarta dan S3 Teknik Elektro di UTM Malaysia. Beliau adalah Dosen di Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Ahmad Dahlan Yogyakarta.