

# Regulator *Capacitor Bank* Satu Fasa dengan Daya hingga 3500VA Berbasis Arduino

David Imani Nugroho, Handry Khoswanto, Hanny Hosiana Tumbelaka  
Program Studi Teknik Elektro, Universitas Kristen Petra  
Jl.Siwalankerto 121-131, Surabaya 60236, Indonesia  
E-Mail: schumi.idn@gmail.com ; handry@petra.ac.id ; tumbekh@petra.ac.id

**Abstrak--** Walau tidak diberlakukan biaya kelebihan pemakaian daya reaktif, *home industry* harus tetap memperhatikan pentingnya faktor daya untuk dapat mengoptimalkan penggunaan daya yang terpasang.

Desain regulator *capacitor bank* ini dikhususkan untuk *home industry* dengan daya hingga 3500VA dengan menggunakan Arduino sebagai otak dari rangkaian kontrol. Regulator ini mengukur faktor daya dengan metode *zero crossing detector* dan menggunakan kombinasi *relay* dan *Thyristor-Switched Capacitor* yang dimodifikasi dengan MOSFET sebagai saklar untuk kapasitor.

Pengukuran faktor daya dengan metode *zero crossing detector* bekerja dengan baik pada beban linear induktif. Kompensasi pada beban linear memiliki kesalahan rata-rata sebesar 2.4% dan untuk beban tidak linear memiliki kesalahan rata-rata sebesar 33.96%.

**Kata Kunci--** Arduino, Zero Crossing Detector, Faktor Daya, Capacitor Bank

## I. PENDAHULUAN

Peralatan listrik tidak dapat dipisahkan dari kebutuhan industri yang terus berkembang. Peralatan seperti *blower*, pendingin, pompa, dan lain-lain merupakan beberapa aplikasi motor listrik yang menjadi salah satu beban terbesar yang sering digunakan. Peralatan-peralatan tersebut termasuk beban induktif yang dapat menurunkan faktor daya sehingga menyebabkan kapasitas daya listrik yang telah terpasang tidak dapat dimanfaatkan secara optimal.

Pada beberapa kasus, pelaku industri melakukan penambahan kapasitas daya yang terpasang untuk menambah daya aktif. Padahal, penambahan kapasitas daya terpasang juga dapat menaikkan biaya penggunaan listrik tiap kWh sehingga tagihan penggunaan listrik menjadi lebih boros. Oleh karena itu, faktor daya yang baik dapat menghemat biaya yang dikeluarkan [1].

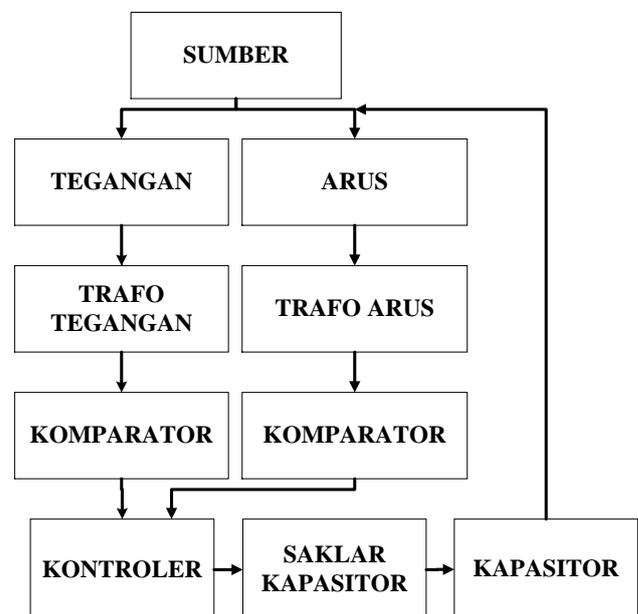
Pelaku industri tidak lagi hanya beraktifitas di sebuah pabrik yang besar tetapi juga di *home industry*. Di industri besar, kompensasi faktor daya merupakan suatu hal yang wajar dilakukan karena adanya biaya kelebihan pemakaian yang akan dikenakan apabila faktor daya di industri tersebut berada di bawah standar yang sudah ditetapkan. Akan tetapi peraturan ini tidak berlaku di skala rumah atau *home industry*. Untuk menaikkan faktor daya di industri, digunakan *capacitor bank* dengan regulator seperti “Varlogic”, “RVC”, dan “RVT”.

Sebenarnya, penggunaan *capacitor bank* skala rumah sudah ada. Regulator yang digunakan adalah jenis *on-off* dimana pada skala rumahan, alat ini justru menimbulkan masalah baru seperti faktor daya yang berlebih atau alat ini tidak bekerja sama sekali karena kapasitor yang digunakan memiliki nilai konstan sehingga dapat terjadi *over compensation* atau *under compensation*. Oleh karena itu

dibutuhkan *capacitor bank* skala rumah yang dapat melakukan kompensasi secara variabel.

## II. PERANCANGAN RANGKAIAN KONTROL

Pada bagian ini akan dijelaskan desain rangkaian kontrol regulator *capacitor bank* secara garis besar. Gambar 1 menunjukkan blok diagram rangkaian kontrol.



Gambar 1. Blok Diagram Rangkaian Kontrol

Rangkaian kontrol ini melakukan dua hal utama yaitu mengukur faktor daya dan mengkompensasi. Pengukuran dilakukan dengan metode *zero crossing detector* yang menghitung selisih waktu antara tegangan dan arus ketika melewati titik nol. Tegangan sumber ditransformasi dengan menggunakan trafo tegangan sehingga tegangan dapat dibaca oleh rangkaian kontrol. Arus ditransformasi dengan menggunakan trafo arus dan dikonversi menjadi tegangan sehingga dapat dibaca oleh rangkaian kontrol. Dengan menggunakan komparator, gelombang sinus tegangan dan arus diubah menjadi sinyal kotak sehingga kontroler dapat membaca waktu tegangan dan arus ketika melewati titik nol. Pengkompensasian dilakukan oleh kontroler dengan mengaktifkan *driver/saklar* kapasitor hingga rangkaian kontrol dapat mencapai faktor daya yang telah ditentukan.

### A. Desain Hardware

Rangkaian kontrol ini terbagi menjadi tiga bagian yaitu *input*, kontroler, dan *output*. Bagian *input* terdiri dari trafo tegangan untuk menurunkan tegangan, trafo arus untuk

mengubah arus menjadi tegangan, komparator untuk mengubah gelombang sinus menjadi sinyal kotak, dan potensiometer sebagai pengatur faktor daya acuan. Kontroler yang digunakan adalah Arduino Uno. Bagian *output* terdiri dari *driver* kapasitor untuk mengaktifkan kapasitor dan LCD untuk menampilkan faktor daya acuan dan faktor daya terukur.

Kapasitor yang digunakan harus dapat mengakomodasi kebutuhan VAR dengan daya terpasang sebesar 3500VA dengan faktor daya terendah sebesar 0.7 dan tertinggi sebesar 0.95. Perhitungan kebutuhan kapasitor menggunakan rumus segitiga daya sebagai berikut :

$$S^2 = P^2 + Q^2 \tag{1}$$

$$P = S \cos \varphi \tag{2}$$

Dimana, S = Daya Semu (VA),  
 P = Daya Aktif (W),  
 Q = Daya Reaktif (VAR),  
 cos φ = Faktor Daya.

Dengan menggunakan rumus 1 dan 2, maka didapatkan nilai kebutuhan kapasitor sebesar 2299,304VAR atau dengan kapasitansi sebesar 138.4239μF. Berdasarkan cara kerja rangkaian kontrol dalam mengaktifkan kapasitor yang dilakukan dengan metode biner, maka kapasitor yang digunakan pada tingkat tertentu memiliki kapasitansi sekitar dua kali dari kapasitor yang ada di tingkat bawahnya hingga nilai kapasitansi total tercapai. Untuk lebih jelas dapat dilihat kapasitor yang digunakan pada rangkaian kontrol pada Tabel 1.

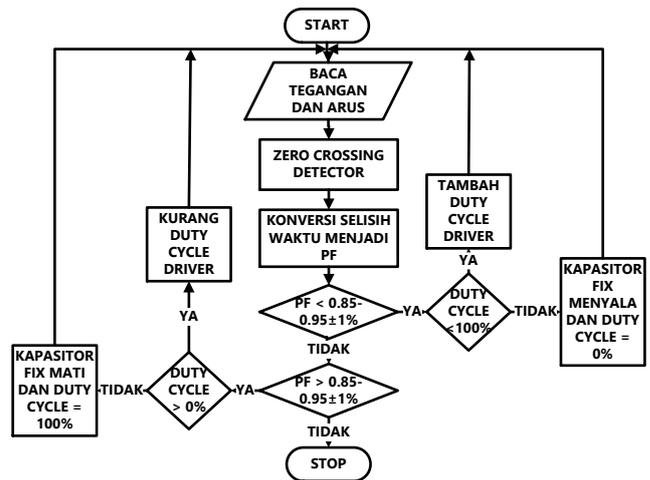
Tabel 1. Tabel Penggunaan Kapasitor

Kapasitansi	Jumlah
60 μF	1 buah
35 μF	1 buah
20 μF	1 buah
10 μF	1 buah
5 μF	2 buah
3 μF	1 buah
2 μF	1 buah
<b>Total</b>	<b>8 buah</b>

*Driver* kapasitor yang digunakan terdiri dari dua jenis yaitu dengan menggunakan *relay* dan metode TSC (*Thyristor-Switched Capacitor*). Pada rangkaian kontrol ini terdapat 7 buah *driver relay* yang digunakan pada kapasitor dengan kapasitansi 60μF, 35μF, 20μF, 10μF, 5μF 5μF, dan 3μF. Metode TSC pada umumnya digunakan sebuah kapasitor, sebuah *bidirectional thyristor*, dan sebuah *current limiting reactor* [2]. Pada rangkaian kontrol ini dilakukan modifikasi dengan mengganti *thyristor* dan *current limiting reactor* dengan MOSFET yang didukung dengan *diode bridge* sehingga dapat dioperasikan pada besaran DC dan mudah dikontrol oleh Arduino. Metode TSC digunakan pada kapasitor 2μF dengan tujuan untuk *tuning* kebutuhan kapasitor sehingga nilai faktor daya acuan dapat tercapai.

**B. Desain Software**

Secara garis besar desain *software* dibagi menjadi tiga bagian yaitu pembacaan *input*, perhitungan, dan pengaturan *output*. Gambar 2 menunjukkan alur kerja program Arduino.



Gambar 2. Alur Kerja Program Arduino

Pada tahap pembacaan *input*, Arduino membaca data yang dikirimkan oleh komparator dan pengatur faktor daya acuan. Pembacaan data yang dikirimkan oleh komparator kemudian dihitung sehingga diketahui faktor daya terukur dengan menggunakan rumus sebagai berikut :

$$\cos \varphi = \cos(2 \pi f \Delta t) \tag{3}$$

$$f = \frac{1}{T} \tag{4}$$

Dimana, cos φ = Faktor Daya,  
 f = frekuensi jaringan (Hz),  
 Δt = selisih waktu tegangan dan arus (s),  
 T = periode 1 gelombang tegangan (s).

Nilai faktor daya yang telah dihitung kemudian dibandingkan dengan nilai faktor daya acuan. Nilai faktor daya acuan berkisar antara 0.85 hingga 0.95. Rangkaian kontrol akan mengkompensasi jaringan hingga nilai faktor daya terukur ±1% nilai faktor daya acuan.

*Driver relay* dijalankan dengan menggunakan metode biner sebanyak 7 bit dimana akan menghasilkan 128 buah kombinasi. Urutan pengontrolan kapasitor dari MSB ke LSB adalah 60μF, 35μF, 20μF, 10μF, 5μF 5μF, dan 3μF. *Driver* dengan metode TSC dikontrol dengan melakukan perubahan pada *duty cycle* sebanyak 20%. Apabila nilai faktor daya terukur kurang dari nilai faktor daya acuan maka *duty cycle* akan ditambah atau ketika nilai faktor daya terukur lebih besar dari nilai faktor daya acuan maka *duty cycle* akan dikurang. Apabila *duty cycle* ditambah dan mencapai 100% maka bit LSB akan ditambah dan *duty cycle* akan direset menjadi 0%. Sedangkan ketika *duty cycle* dikurang dan mencapai 0% maka bit LSB akan dikurang dan *duty cycle* akan direset menjadi 100%.

**III. PENGUJIAN RANGKAIAN KONTROL**

**A. Pengujian Program Arduino untuk Ketepatan Pengukuran Faktor Daya**

Pengujian ini dilakukan untuk mengetahui ketepatan pengukuran *cos phi* yang dilakukan oleh Arduino. Pengukuran difokuskan pada angka 0.5 hingga 0.99 *lagging* karena disesuaikan dengan rentang kerja rangkaian kontrol. Pengujian dilakukan dengan membandingkan nilai faktor daya yang diukur oleh *Power Harmonic Analyzer* Fluke 41B dengan

Arduino. Data diambil sebanyak 70 pasang yang dapat dilihat pada Tabel 2.

Tabel 2. Tabel Perbandingan Pengukuran Fluke 41B dan Arduino

Fluke 41B	Arduino	Fluke 41B	Arduino
0.99	0.98	0.5	0.5
0.98	0.97	0.52	0.52
0.97	0.96	0.54	0.54
0.96	0.95	0.56	0.56
0.95	0.94	0.58	0.58
0.94	0.93	0.6	0.6
0.93	0.92	0.62	0.62
0.92	0.91	0.64	0.64
0.91	0.9	0.66	0.66
0.9	0.89	0.68	0.68
0.89	0.88	0.7	0.7
0.88	0.87	0.72	0.72
0.87	0.87	0.74	0.74
0.86	0.86	0.76	0.76
0.85	0.85	0.78	0.78
0.84	0.83	0.8	0.79
0.83	0.82	0.81	0.8
0.82	0.82	0.82	0.81
0.81	0.81	0.83	0.82
0.8	0.8	0.84	0.83
0.78	0.78	0.85	0.83
0.76	0.76	0.86	0.84
0.74	0.74	0.87	0.85
0.72	0.72	0.88	0.86
0.7	0.7	0.89	0.87
0.68	0.68	0.9	0.88
0.66	0.66	0.91	0.89
0.64	0.64	0.92	0.9
0.62	0.62	0.93	0.91
0.6	0.61	0.94	0.93
0.58	0.59	0.95	0.94
0.56	0.57	0.96	0.95
0.54	0.55	0.97	0.96
0.52	0.52	0.98	0.97
0.5	0.5	0.99	0.98

Pada Tabel 2 dapat dilihat ada sedikit perbedaan hasil pengukuran Fluke 41B dan Arduino dengan kesalahan rata-rata sebesar 0.78%. Hasil pengukuran yang telah dilakukan kemudian dikalibrasi sehingga didapatkan hasil yang lebih maksimal. Pengalibrasian dilakukan dengan menggunakan *Microsoft Excel* dan menggabungkan kedua data tersebut dalam grafik *scatter* dengan pengukuran Arduino sebagai nilai x dan pengukuran Fluke 41B sebagai nilai y. Setelah grafik terbentuk, ditambahkan *trendline* dengan jenis persamaan polinomial orde 2 sehingga dihasilkan persamaan sebagai berikut :

$$y = 0.0195x^2 + 1.0109x - 0.0152 \quad (5)$$

Dimana, y = Nilai Arduino Terkalibrasi  
 x = Nilai Arduino Awal

Setelah Arduino dikalibrasi, dilakukan pengujian ulang dengan metode yang sama yaitu dibandingkan dengan Fluke 41B sebanyak 70 buah data. Hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Tabel Perbandingan Pengukuran Fluke 41B dan Arduino Terkalibrasi

Fluke 41B	Arduino	Fluke 41B	Arduino
0.99	0.99	0.5	0.5
0.98	0.98	0.52	0.52
0.97	0.97	0.54	0.54
0.96	0.96	0.56	0.56
0.95	0.95	0.58	0.58
0.94	0.94	0.6	0.6
0.93	0.93	0.62	0.62
0.92	0.92	0.64	0.64
0.91	0.91	0.66	0.66
0.9	0.9	0.68	0.68
0.89	0.89	0.7	0.7
0.88	0.88	0.72	0.72
0.87	0.87	0.74	0.74
0.86	0.86	0.76	0.76
0.85	0.85	0.78	0.78
0.84	0.84	0.8	0.8
0.83	0.83	0.81	0.81
0.82	0.82	0.82	0.82
0.81	0.81	0.83	0.83
0.8	0.8	0.84	0.84
0.78	0.78	0.85	0.85
0.76	0.76	0.86	0.86
0.74	0.74	0.87	0.87
0.72	0.72	0.88	0.88
0.7	0.7	0.89	0.89
0.68	0.68	0.9	0.9
0.66	0.66	0.91	0.91
0.64	0.64	0.92	0.92
0.62	0.62	0.93	0.93
0.6	0.6	0.94	0.94
0.58	0.59	0.95	0.95
0.56	0.57	0.96	0.96
0.54	0.55	0.97	0.97
0.52	0.52	0.98	0.98
0.5	0.5	0.99	0.99

Hasil pengujian program perhitungan Arduino terkalibrasi memiliki ketepatan baca dengan persentase kesalahan sebesar 0.02%. Kesalahan pengukuran masih terjadi karena adanya ketidaklinearan transformasi pada yang dilakukan oleh trafo arus. Trafo arus akan menghasilkan tegangan yang berbeda sudut ketika arus yang lewat berubah sehingga mengganggu proses pengukuran faktor daya. Akan tetapi, angka ini lebih baik khususnya pada nilai kritis seperti 0.85-0.95 yang digunakan sebagai nilai acuan PF sehingga pengukuran Arduino layak digunakan.

#### B. Pengujian Program Arduino untuk Driver MOSFET

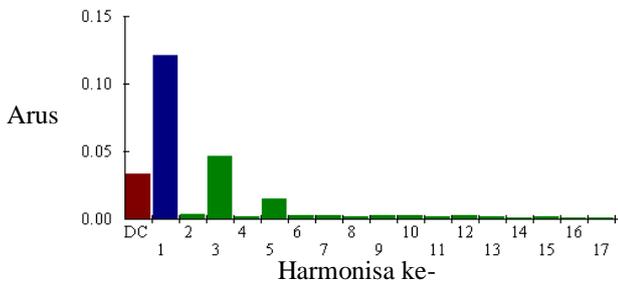
Pengujian ini dilakukan dilakukan dalam dua tahap. Tahap pertama dilakukan untuk mengetahui daya reaktif yang dihasilkan oleh sebuah kapasitor 2μF 450VAC dengan *duty cycle* yang berbeda. Tahap kedua dilakukan untuk mengetahui efek yang ditimbulkan akibat frekuensi *sampling* untuk membentuk *duty cycle*. Pengujian pertama dilakukan dengan mengukur daya reaktif yang dihasilkan pada tiap-tiap *duty cycle* dengan menggunakan Fluke 41B. Data diambil sebanyak 11 pasang yang dapat dilihat pada Tabel 4.

Tabel 4. Tabel Perbandingan *Duty Cycle* dan Daya Reaktif

Duty Cycle	Daya Reaktif
0	0
10	8
20	12
30	15
40	15
50	22
60	34
70	34
80	37
90	37
100	38

Hasil pengukuran daya reaktif adalah nilai rata-rata yang dapat diambil karena nilai daya reaktif beresilasi ketika dilakukan pengukuran. Hal ini disebabkan adanya pergeseran fasa pada trafo tegangan yang digunakan sebagai patokan dalam menjalankan driver MOSFET.

Pengujian kedua dilakukan dengan mengukur arus kapasitor ketika duty cycle diatur sebesar 80%. Hasil pengujian dapat dilihat dalam bentuk grafik harmonisa pada Gambar 3.



Gambar 3. Harmonisa Arus Penggunaan Driver MOSFET pada Duty Cycle 80%

Penggunaan metode TSC juga memiliki kelemahan karena merekayasa tegangan yang seharusnya memiliki sinyal sinus murni sehingga menimbulkan harmonisa pada urutan ke-3 dan ke-5. Pada dasarnya, penggunaan MOSFET untuk metode TSC dapat dilakukan dan rangkaian kontrol bekerja dengan normal.

C. Pengujian Program Arduino untuk Ketepatan Penggunaan Kapasitor

Pengujian ini dilakukan untuk mengetahui ketepatan penggunaan kapasitor yang dilakukan oleh Arduino. Ada dua jenis pengujian ketepatan penggunaan kapasitor yang spengukurannya lebih difokuskan pada angka 0.85 hingga 0.95 lagging karena disesuaikan dengan nilai faktor daya acuan yang menjadi patokan pengukuran dari rangkaian kontrol ini. Pertama, dilakukan pengambilan data sebanyak 22 buah dengan beban linear sebesar 700VA dengan faktor daya 0.75. Hasil pengukuran dapat dilihat pada Tabel 5.

Tabel 5. Tabel Perbandingan Faktor Daya Acuan dengan Cos Phi Terkalibrasi dan Fluke 41B pada Pengukuran Beban Linear

SetPoint	Arduino	Fluke 41B	Waktu Respon (s)
0.85	0.85	0.84	17
0.86	0.86	0.84	19
0.87	0.88	0.87	21
0.88	0.87	0.87	22
0.89	0.88	0.87	25
0.9	0.89	0.88	29

0.91	0.91	0.88	33
0.92	0.93	0.89	36
0.93	0.93	0.89	38
0.94	0.93	0.89	43
0.95	0.95	0.92	43
0.95	0.95	0.92	42
0.94	0.93	0.89	43
0.93	0.93	0.89	39
0.92	0.92	0.89	36
0.91	0.91	0.88	33
0.9	0.89	0.88	30
0.89	0.89	0.88	25
0.88	0.88	0.87	22
0.87	0.88	0.87	21
0.86	0.86	0.84	20
0.85	0.85	0.84	16

Dari hasil pengukuran di atas, kompensasi yang dilakukan sudah memenuhi syarat yaitu nilai faktor daya terukur (Arduino) memiliki kesalahan sebesar 0.14% dari nilai faktor daya acuan atau SetPoint. Selain itu, nilai pengukuran Arduino menunjukkan nilai yang berbeda dari nilai pengukuran Fluke 41B dengan persentase kesalahan sebesar 2.4%. Kesalahan pengukuran yang terjadi disebabkan oleh adanya ketidaklinearan transformasi pada arus yang dilakukan oleh trafo arus dan berdasarkan hasil pengujian pada Tabel 6, pengaruh dari harmonisa yang ditimbulkan oleh driver MOSFET seperti pada Gambar 3 juga menyebabkan kesalahan pengukuran. Selain itu, waktu respon berbanding lurus dengan nilai SetPoint yang digunakan. Hal ini disebabkan penggunaan metode pengkompensasian secara biner sehingga semakin besar beban atau SetPoint maka semakin lama regulator mencapai target.

Kedua, dilakukan pengujian terhadap beban tidak linear sebesar 300VA dengan faktor daya sebesar 0.22. Pengujian dilakukan dengan mengambil 22 buah data yang dapat dilihat pada Tabel 6.

Tabel 6. Tabel Perbandingan Faktor Daya Acuan dengan Cos Phi Terkalibrasi dan Fluke 41B pada Pengukuran Beban Tidak Linear

SetPoint	Arduino	Fluke 41B	Waktu Respon (s)
0.85	-	-	∞
0.86	-	-	∞
0.87	-	-	∞
0.88	-	-	∞
0.89	-	-	∞
0.9	0.91	0.67	32
0.91	0.91	0.67	32
0.92	0.91	0.67	32
0.93	0.92	0.68	33
0.94	0.95	0.73	36
0.95	0.95	0.73	36
0.95	0.95	0.73	36
0.94	0.95	0.73	37
0.93	0.93	0.68	33
0.92	0.91	0.67	32
0.91	0.91	0.67	31
0.9	0.91	0.67	31
0.89	-	-	∞
0.88	-	-	∞
0.87	-	-	∞

0.86	-	-	$\infty$
0.85	-	-	$\infty$

Hasil pengujian menunjukkan perbedaan pengukuran yang cukup jauh antara Arduino dan pengukuran Fluke 41B dengan persentase kesalahan sebesar 33.96%. Ketika nilai faktor daya acuan diatur pada angka 0.85-0.89, pengkompensasian mengalami osilasi lebih dari 1% sehingga tidak dapat dilakukan pengukuran. Dari pengujian di atas, dapat disimpulkan bahwa pengukuran beban tidak linear atau memiliki harmonisa dapat menyebabkan kesalahan yang cukup besar pada Arduino.

Dibandingkan dengan persentase kesalahan *Varlogic* milik Schneider sebesar 2% [3], rangkaian kontrol ini memiliki nilai kesalahan yang hampir sama pada beban linear yaitu sebesar 2.4%. Sedangkan untuk beban tidak linear dengan kesalahan sebesar 33.96%, rangkaian kontrol masih belum mampu mencapai hasil yang sama dengan *Varlogic*.

#### D. Pengujian Arus Maksimum Kapasitor untuk Menentukan Proteksi Arus

Pengujian ini dilakukan untuk mengetahui arus maksimum dari masing-masing kapasitor yang digunakan sehingga dapat ditentukan penggunaan proteksi arus yang dibutuhkan. Pengujian dilakukan dengan melakukan pengukuran arus menggunakan Fluke 337 *True RMS Clamp Meter* pada masing-masing kapasitor dengan nilai kapasitif yang berbeda. Pengujian dilakukan dengan cara mengambil data pengukuran sebanyak 10 kali pada masing-masing kapasitor. Hasil pengukuran dapat dilihat pada Tabel 7.

Tabel 7. Tabel Pengukuran Arus Maksimum Kapasitor

Percobaan ke-	Kapasitansi						
	2	3	5	10	20	35	60
1	0.2 A	0.3 A	0.4 A	0.8 A	1.5 A	2.7 A	4.7 A
2	0.2	0.3	0.4	0.8	1.5	2.7	4.7
3	0.2	0.3	0.4	0.8	1.5	2.7	4.7
4	0.2	0.3	0.4	0.8	1.5	2.7	4.7
5	0.2	0.3	0.4	0.8	1.5	2.7	4.7
6	0.2	0.3	0.4	0.8	1.5	2.7	4.7
7	0.2	0.3	0.4	0.8	1.5	2.7	4.7
8	0.2	0.3	0.4	0.8	1.5	2.7	4.7
9	0.2	0.3	0.4	0.8	1.5	2.7	4.7
10	0.2	0.3	0.4	0.8	1.5	2.7	4.7

Dari hasil pengukuran di atas, maka ditentukan proteksi arus yang digunakan terdiri dari 3 buah MCB 6A dengan rincian MCB pertama untuk kapasitor 60 $\mu$ F dengan total arus nominal sebesar 4.7A, MCB kedua untuk kapasitor 35 $\mu$ F dan 20 $\mu$ F dengan total arus nominal sebesar 4.2A, serta MCB ketiga untuk kapasitor 10 $\mu$ F, 5 $\mu$ F, 5 $\mu$ F, 3 $\mu$ F, dan 2 $\mu$ F dengan total arus nominal sebesar 2.1A.

## IV. KESIMPULAN

Kesimpulan yang dapat diambil dari perancangan, pembuatan desain dan pengujian rangkaian kontrol ini dapat disimpulkan dalam beberapa poin berikut:

- Dibutuhkan kapasitor sebesar 138.4239 $\mu$ F untuk melakukan kompensasi faktor daya 0.7 ke faktor daya

0.95 dengan daya nyata tetap pada rangkaian kontrol 3500VA.

- Penggunaan MOSFET dan *diode bridge* dapat menggantikan fungsi *Thyristor-Switched Capacitor*.
- Penggunaan *driver* MOSFET hanya dilakukan untuk *tuning* kebutuhan kompensasi.
- Pengukuran faktor daya yang baik dengan metode *zero crossing detector* hanya dapat dilakukan pada beban linear.
- Kesalahan pengukuran faktor daya pada beban linear sebesar 0.02%.
- Kesalahan pengkompensasian pada beban linear sebesar 2.4%.
- Kesalahan pengkompensasian pada beban tidak linear sebesar 33.96%.
- Waktu respon pengkompensasian berbanding lurus dengan besar beban dan nilai faktor daya acuan.

## DAFTAR PUSTAKA

- [1] Wibowo, P. S. (2010). Rancang Bangun Automatic Power Factor Correction Berbasis *Microcontroller* AVR Menggunakan Metode Jaringan Syaraf Tiruan, 1–7.
- [2] Narain G. Hingorani, L. G. (2000). *Understanding FACTS. Journal of Chemical Information and Modeling* (Vol. 53). Institute of Electrical and Electronics Engineers, Inc.
- [3] Schneider. (2009). *Varlogic NR6, NR12 Power Factor Controller User Manual*.