



UNIVERSITAS INDONESIA

PENINGKATAN PERFORMANSI ALGORITMA DIGITAL
PHASE LOCKED LOOP UNTUK SINKRONISASI GELOMBANG
PADA SISTEM GRID CONNECTED PHOTOVOLTAIC

SKRIPSI

YUDDY SYAIFUDIN

0706268120

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO

DEPOK

JUNI 2011



UNIVERSITAS INDONESIA

PENINGKATAN PERFORMANSI ALGORITMA DIGITAL
PHASE LOCKED LOOP UNTUK SINKRONISASI GELOMBANG
PADA SISTEM GRID CONNECTED PHOTOVOLTAIC

SKRIPSI

Skripsi ini diajukan untuk melengkapi sebagian persyaratan menjadi
Sarjana Teknik

YUDDY SYAIFUDIN

0706268120

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
PEMINATAN KENDALI
DEPOK
JUNI 2011

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Yuddy Syaifudin

NPM : 0706268120

Tanda Tangan :

Tanggal :

HALAMAN PENGESAHAN

SKRIPSI INI DIAJUKAN OLEH :

NAMA : Yuddy Syaifudin
NPM : 0706268120
PROGRAM STUDI : Teknik Elektro
JUDUL SKRIPSI : PENINGKATAN PERFORMANSI
ALGORITMA DIGITAL PHASE LOCKED LOOP
UNTUK SINKRONISASI GELOMBANG PADA
SISTEM GRID CONNECTED PHOTOVOLTAIC

**TELAH DIPRESENTASIKAN DAN DITERIMA SEBAGAI BAGIAN
PERSYARATAN YANG DIPERLUKAN UNTUK MEMPEROLEH
GELAR SARJANA TEKNIK PADA PROGRAM STUDI TEKNIK
ELEKTRO, FAKULTAS TEKNIK, UNIVERSITAS INDONESIA**

Pembimbing : Dr. Ir. Feri Yusivar M. Eng (.....)

Penguji : Ir. Aries Subiantoro M.Sc (.....)

Penguji : Dr. Ir. Ridwan Gunawan M.T (.....)

Ditetapkan di :

Tanggal :

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, proses penulisan skripsi ini dapat terselesaikan. Penulisan skripsi ini dilakukan dalam rangka memenuhi persyaratan dari mata kuliah Skripsi yang terdapat dalam kurikulum program studi Teknik Elektro Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan skripsi, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1). Dr. Ir. Feri Yusivar M. Eng, selaku dosen pembimbing, serta dosen-dosen lainnya, yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2). Orang tua dan keluarga saya yang telah memberikan bantuan berupa dukungan material dan moral;
- (3). Teman-teman, terutama Rian Suryadiningrat, Muhamad Yasil Farabi, dan Wayan Wicak Ananduta, selaku rekan sekerja saya, dan pihak-pihak lainnya yang telah membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap agar Tuhan Yang Maha Esa berkenan membalas segala kebaikan dari semua pihak yang telah membantu. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu.

Depok, 9 Juni 2011

Yuddy Syaifudin

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini :

Nama : Yuddy Syaifudin
NPM : 0706268120
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

, demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

**PENINGKATAN PERFORMANSI ALGORITMA DIGITAL PHASE
LOCKED LOOP UNTUK SINKRONISASI GELOMBANG PADA SISTEM
GRID CONNECTED PHOTOVOLTAIC**

, beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis / pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di :,

Pada tanggal :,

Yang menyatakan,

(Yuddy Syaifudin)

ABSTRAK

Nama : Yuddy Syaifudin
Program Studi : Teknik Elektro
Judul : PENINGKATAN PERFORMANSI ALGORITMA DIGITAL PHASE LOCKED LOOP UNTUK SINKRONISASI GELOMBANG PADA SISTEM GRID CONNECTED PHOTOVOLTAIC

Phase Locked Loop (PLL) merupakan suatu blok yang dapat berupa algoritma maupun rangkaian elektronika, dengan fungsi sebagai pembentuk sinyal yang sinkron dengan suatu sinyal referensi tertentu. Dalam skripsi ini, algoritma PLL (digital) akan diaplikasikan untuk melakukan proses sinkronisasi terhadap suatu sinyal tegangan tertentu yang berasal dari *grid*. Proses ini dilakukan dengan tujuan akhir untuk melakukan sinkronisasi antara sinyal tegangan yang diproses dari output suatu *photovoltaic* (PV) dengan sinyal tegangan dari *grid* yang bersangkutan, untuk melakukan operasi paralel. Kemudian, PLL digital ini akan dikembangkan dengan *low pass filter* untuk menghilangkan osilasi yang terdapat pada output PLL tersebut. Dan akhirnya, *all pass filter* (APF) akan digunakan bersama dengan PLL dan *low pass filter* yang telah dibuat sebelumnya untuk semakin memperbaiki respon output yang dihasilkan. Kesemua hasil simulasi yang diperoleh akan ditampilkan dalam bentuk grafik dalam skripsi ini untuk melakukan perbandingan dan analisis. Kemudian, PLL yang telah disimulasikan akan digunakan bersama algoritma *current control* untuk mensimulasikan aplikasi mereka dalam mensuplai *grid* dengan faktor daya yang optimal.

Kata Kunci : PLL, sinkronisasi, PV, *low pass filter*, APF, *current control*

ABSTRACT

Name : Yuddy Syaifudin
Study Program : Electrical Engineering
Title : PERFORMANCE ENHANCEMENT OF DIGITAL PHASE LOCKED LOOP ALGORITHM IN SYNCHRONIZING THE WAVEFORM ON GRID CONNECTED PHOTOVOLTAIC SYSTEM

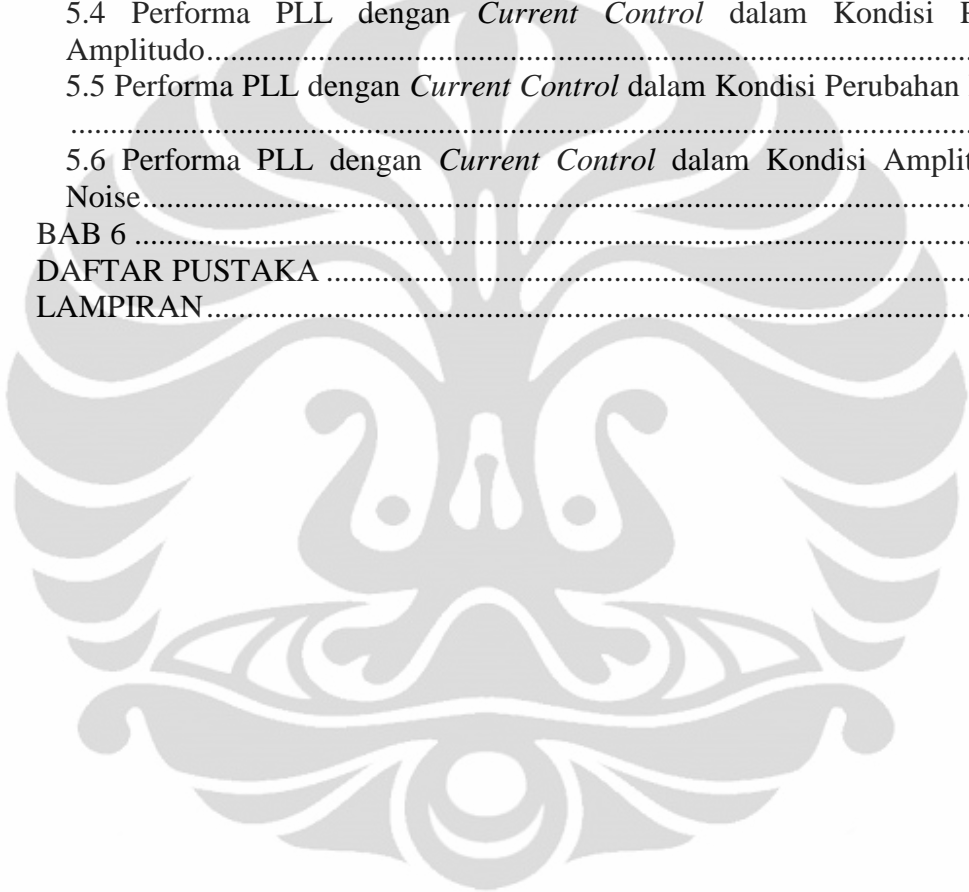
Phase Locked Loop (PLL) is a system that can be realized in an algorithm or an electronic circuit, which is used to reconstruct a signal which is synchronized to a reference signal. In this paper, a PLL algorithm (digital) will be applied to synchronize a voltage signal with a certain grid voltage signal. The final purpose of this synchronization process is to synchronize a voltage signal which is processed from the output of a photovoltaic (PV) with that grid voltage signal to perform a parallel operation. Then, this digital PLL will be developed with a low pass filter to reduce the oscillation observed in the PLL output. Finally, an all pass filter (APF) will be used together with the previous PLL and low pass filter to improve the output response. All the simulation results will be shown to be compared and analyzed. Then, the simulated PLL will be used together with current control algorithm to simulate their application in supplying grid with optimal power factor value.

Keywords : PLL, synchronization, PV, low pass filter, APF, current control

DAFTAR ISI

SKRIPSI.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
BAB 1	1
1.1 Latar Belakang	1
1.2 Tujuan.....	2
1.3 Waktu dan Tempat Pelaksanaan.....	2
1.4 Pembatasan Pembahasan Masalah	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan.....	3
BAB 2	5
2.1 Dasar Mengenai PLL dengan Perhitungan Berbasis Tegangan	5
2.2 Dasar Mengenai PLL dengan Perhitungan Berbasis Daya.....	8
2.3 Perbandingan Tiga Metode Sinkronisasi Lainnya dengan PLL.....	11
BAB 3	14
3.1 Blok Simulasi Awal dan Penjelasannya.....	14
3.2 Parameter PLL Tegangan dan PLL Daya yang Digunakan	15
3.3 <i>Low Pass Filter</i> untuk Pengembangan PLL Tegangan	15
3.4 <i>All Pass Filter</i> untuk Pengembangan PLL Tegangan dengan <i>Low Pass Filter</i>	16
3.5 Penentuan <i>Load</i> , Induktor Konektor <i>Photovoltaic-Grid</i> (L_{grid}), dan Parameter-Parameter pada <i>Current Control</i>	17
BAB 4	18
4.1 Hasil Simulasi Awal Kedua PLL dalam Kondisi Ideal	18
4.2 Hasil Simulasi Awal Kedua PLL dalam Kondisi Perubahan Amplitudo....	19
4.3 Hasil Simulasi Awal Kedua PLL dalam Kondisi Perubahan Frekuensi	21
4.4 Hasil Simulasi Awal Kedua PLL dalam Kondisi Amplitudo Ber- <i>noise</i>	23
4.5 Analisis Lanjutan Hasil Simulasi Awal dan Pengembangan Algoritma....	25
4.6 Algoritma PLL Tegangan dengan <i>Low Pass Filter</i>	27
4.7 Performa Algoritma PLL Tegangan dengan Filter dalam Kondisi Ideal	28
4.8 Performa Algoritma PLL Tegangan dengan Filter dalam Kondisi Perubahan Amplitudo.....	29
4.9 Performa Algoritma PLL Tegangan dengan Filter dalam Kondisi Perubahan Frekuensi	31
4.10 Performa Algoritma PLL Tegangan dengan Filter dalam Kondisi Amplitudo Ber- <i>noise</i>	33
4.11 Pengembangan Algoritma PLL dengan Algoritma <i>All Pass Filter</i> (APF) untuk Penggeser Fasa Sebesar Sembilan Puluh Derajat, serta Hasil Simulasinya dalam Kondisi Ideal.....	36

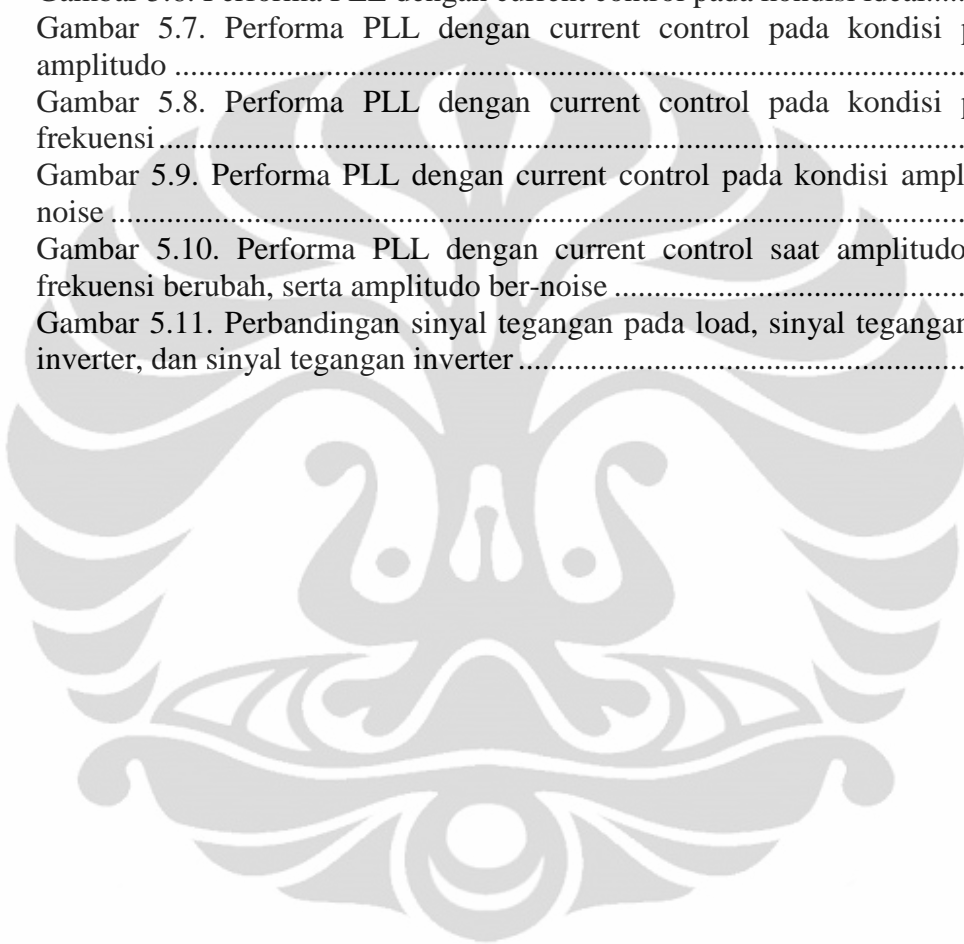
4.12 Performa Algoritma PLL dengan Algoritma APF dalam Kondisi Perubahan Amplitudo.....	38
4.13 Performa Algoritma PLL dengan Algoritma APF dalam Kondisi Perubahan Frekuensi	40
4.14 Performa Algoritma PLL dengan Algoritma APF dalam Kondisi Perubahan Amplitudo Ber-noise	42
BAB 5	45
5.1 Dasar Aplikasi PLL dalam <i>Grid Connected PV</i>	45
5.2 Skema PLL dengan <i>Current Control</i>	47
5.3 Performa PLL dengan <i>Current Control</i> dalam Kondisi Ideal	49
5.4 Performa PLL dengan <i>Current Control</i> dalam Kondisi Perubahan Amplitudo.....	52
5.5 Performa PLL dengan <i>Current Control</i> dalam Kondisi Perubahan Frekuensi	53
5.6 Performa PLL dengan <i>Current Control</i> dalam Kondisi Amplitudo Ber-Noise.....	53
BAB 6	57
DAFTAR PUSTAKA	59
LAMPIRAN.....	60



DAFTAR GAMBAR

Gambar 2.1. Blok diagram PLL dengan perhitungan berbasis tegangan.....	5
Gambar 2.2. Blok diagram PLL dengan perhitungan berbasis daya.....	8
Gambar 3.1. Blok diagram PLL awal	14
Gambar 4.1. θ estimasi untuk kedua jenis PLL	18
Gambar 4.2. ω estimasi untuk kedua jenis PLL.....	19
Gambar 4.3. Sinyal tegangan PLN saat amplitudo berubah	20
Gambar 4.4. θ estimasi kedua jenis PLL saat amplitudo berubah	20
Gambar 4.5. ω estimasi kedua jenis PLL saat amplitudo berubah.....	21
Gambar 4.6. Sinyal tegangan PLN saat frekuensi berubah.....	22
Gambar 4.7. θ estimasi kedua jenis PLL saat frekuensi berubah.....	22
Gambar 4.8. ω estimasi kedua jenis PLL saat frekuensi berubah.....	23
Gambar 4.9. Sinyal tegangan PLN dengan noise.....	24
Gambar 4.10. θ estimasi kedua jenis PLL saat amplitudo ber-noise	24
Gambar 4.11. ω estimasi kedua jenis PLL saat amplitudo ber-noise.....	25
Gambar 4.12. Blok diagram PLL tegangan yang dilengkapi dengan low pass filter	27
Gambar 4.13. θ estimasi PLL tegangan dengan filter	28
Gambar 4.14. ω estimasi PLL tegangan dengan filter	28
Gambar 4.15. v_q estimasi PLL tegangan dengan filter.....	29
Gambar 4.16. θ estimasi PLL tegangan dengan filter saat amplitudo berubah.....	30
Gambar 4.17. ω estimasi PLL tegangan dengan filter saat amplitudo berubah....	30
Gambar 4.18. v_q estimasi PLL tegangan dengan filter saat amplitudo berubah...	31
Gambar 4.19. θ estimasi PLL tegangan dengan filter saat frekuensi berubah.....	31
Gambar 4.20. ω estimasi PLL tegangan dengan filter saat frekuensi berubah	32
Gambar 4.21. v_q estimasi PLL tegangan dengan filter saat frekuensi berubah....	32
Gambar 4.22. θ estimasi PLL tegangan dengan filter saat amplitudo ber-noise...	33
Gambar 4.23. ω estimasi PLL tegangan dengan filter saat amplitudo ber-noise..	34
Gambar 4.24. v_q estimasi PLL tegangan dengan filter saat amplitudo ber-noise.	34
Gambar 4.25. Performa PLL saat frekuensi dan tegangan berubah, serta ber-noise	35
Gambar 4.26. Blok diagram PLL tegangan yang dilengkapi dengan low pass filter dan algoritma APF	37
Gambar 4.27. v_q estimasi PLL yang menggunakan algoritma APF.....	37
Gambar 4.28. θ estimasi PLL yang menggunakan algoritma APF.....	38
Gambar 4.29. ω estimasi PLL dengan menggunakan algoritma APF	38
Gambar 4.30. v_q estimasi PLL dengan APF saat amplitudo berubah	39
Gambar 4.31. θ estimasi PLL dengan APF saat amplitudo berubah	39
Gambar 4.32. ω estimasi PLL dengan APF saat amplitudo berubah.....	40
Gambar 4.33. v_q estimasi PLL dengan APF saat frekuensi berubah.....	41
Gambar 4.34. θ estimasi PLL dengan APF saat frekuensi berubah.....	41
Gambar 4.35. ω estimasi PLL dengan APF saat frekuensi berubah	42
Gambar 4.36. v_q estimasi PLL dengan APF saat amplitudo ber-noise	42
Gambar 4.37. θ estimasi PLL dengan APF saat amplitudo ber-noise	43
Gambar 4.38. ω estimasi PLL dengan APF saat amplitudo ber-noise.....	43
Gambar 4.39. Performa PLL dengan APF saat frekuensi dan tegangan berubah, serta ber-noise	44

Gambar 5.1. Skema grid dan beban sederhana	45
Gambar 5.2. Skema operasi paralel sederhana antara grid dengan PV.....	46
Gambar 5.3. Skema suplai load oleh grid dan PV secara bersamaan dengan kondisi tegangan yang tersinkronisasi	46
Gambar 5.4. Skema suplai load oleh grid dan PV secara bersamaan dengan kondisi tegangan dan arus yang tersinkronisasi.....	47
Gambar 5.5. Skema PLL tegangan dengan low pass filter dan APF dengan current control	48
Gambar 5.6. Performa PLL dengan current control pada kondisi ideal.....	50
Gambar 5.7. Performa PLL dengan current control pada kondisi perubahan amplitudo	52
Gambar 5.8. Performa PLL dengan current control pada kondisi perubahan frekuensi.....	53
Gambar 5.9. Performa PLL dengan current control pada kondisi amplitudo ber-noise	54
Gambar 5.10. Performa PLL dengan current control saat amplitudo berubah, frekuensi berubah, serta amplitudo ber-noise	55
Gambar 5.11. Perbandingan sinyal tegangan pada load, sinyal tegangan referensi inverter, dan sinyal tegangan inverter.....	55



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dalam kondisi-kondisi tertentu, pemanfaatan daya yang hanya berasal dari PLN (penyedia listrik utama) tidak dapat sepenuhnya memenuhi tuntutan keperluan konsumen. Di sini, salah satu solusi nyata yang dapat diterapkan adalah dengan memanfaatkan sumber energi listrik lainnya yang dapat diandalkan, yang sifatnya substitusional dengan sumber energi listrik dari PLN. Secara khusus, dapat digunakan suatu PV (*photovoltaic*) atau yang sering disebut juga sebagai *solar cell*.

Namun demikian, output dari PV yang berupa tegangan listrik DC tidak dapat secara langsung digunakan untuk mensuplai peralatan-peralatan listrik yang biasanya memanfaatkan tegangan listrik AC dari PLN. Akibatnya, diperlukan suatu *inverter* yang digunakan untuk mengubah tegangan listrik DC hasil output dari PV menjadi tegangan listrik AC yang sesuai dengan karakteristik tegangan listrik AC dari PLN, sehingga akhirnya peralatan-peralatan listrik tersebut dapat disuplai sesuai dengan karakteristik kebutuhannya agar dapat bekerja secara normal.

Agar dapat menghasilkan tegangan listrik AC yang karakteristiknya sesuai dengan tegangan listrik AC dari PLN, *inverter* yang bersangkutan perlu dikontrol dengan menggunakan sinyal-sinyal kendali tertentu, yang mana bentuk sinyal-sinyal pengendali ini akan sangat tergantung dari jenis *inverter* serta skema pengendalian *inverter* yang digunakan. Dalam rangka pengendalian output dari *inverter*, agar dapat disesuaikan dengan pola karakteristik tegangan listrik dari PLN, akan dilakukan proses estimasi dari sudut fasa dan frekuensi tegangan listrik AC PLN. Dua metode estimasi yang digunakan dan dibandingkan adalah dengan menggunakan algoritma PLL yang perhitungannya didasarkan atas perhitungan tegangan listrik dan daya listrik.

Dari hasil perbandingan yang dilakukan, dapat dilihat bahwa kedua jenis PLL tersebut memberikan hasil yang relatif sama, sehingga untuk pengembangan performa PLL berikutnya, ditambah juga dengan beberapa alasan yang akan

dijelaskan kemudian, simulasi hanya akan dilakukan dengan menggunakan PLL yang perhitungannya didasarkan atas tegangan saja.

1.2 Tujuan

Pembahasan dan proses simulasi yang dilaporkan dalam skripsi ini bertujuan untuk :

1. Membuktikan bahwa metode sinkronisasi karakteristik sinyal tegangan listrik (khususnya yang berupa amplitudo, frekuensi, dan sudut fasa) dengan menggunakan metode PLL yang dimodifikasi merupakan metode yang sederhana, namun memberikan hasil yang baik.
2. Memperlihatkan bahwa metode PLL yang dimodifikasi dan dilengkapi dengan blok *current control* dapat diaplikasikan dalam proses sinkronisasi sinyal tegangan PV dengan sinyal tegangan *grid* dalam rangka operasi *grid connected* PV.

1.3 Waktu dan Tempat Pelaksanaan

Aktivitas skripsi ini dilakukan oleh penulis dengan dibimbing oleh dosen pembimbing, yang mana aktivitas mandiri (simulasi dengan menggunakan komputer) oleh penulis biasanya dilakukan di rumah maupun di Laboratorium Teknik Kendali Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sedangkan aktivitas bimbingan rutin biasanya dilakukan di gedung Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Waktu pelaksanaan aktivitas mandiri biasanya dilakukan pada waktu-waktu yang disediakan oleh penulis untuk mengerjakan skripsi, sedangkan aktivitas bimbingan bersama dengan rekan kerja penulis dan dosen pembimbing yang bersangkutan biasanya dilakukan secara rutin setiap hari Selasa pada pukul 13.00.

1.4 Pembatasan Pembahasan Masalah

Dalam skripsi ini, permasalahan yang dibahas dibatasi pada simulasi pengembangan algoritma PLL yang dimodifikasi untuk dapat melakukan sinkronisasi (memberikan output yang tepat), terhadap karakteristik tegangan

listrik yang berasal dari jaringan PLN. Output yang dimaksudkan kali ini adalah amplitudo tegangan listrik, frekuensi tegangan listrik, dan sudut fasa tegangan listrik PLN. Kemudian, output dari PLL yang bersangkutan akan diaplikasikan bersama dengan blok *current control* untuk memperlihatkan (secara simulasi) bahwa PLL dengan *current control* dapat digunakan untuk melakukan sinkronisasi antara sinyal tegangan PV dengan sinyal tegangan PLN (*grid*) dalam rangka operasi *grid connected PV*.

1.5 Metodologi Penelitian

Dalam penulisan dan aktivitas pengerjaan skripsi ini, penulis mengadakan simulasi secara mandiri, bimbingan secara berkala dengan dosen pembimbing, studi literatur dari berbagai sumber yang relevan, serta diskusi-diskusi yang dilakukan bersama-sama dengan rekan kerja penulis.

1.6 Sistematika Penulisan

Secara umum, skripsi ini dituliskan dalam lima bab. Bab yang pertama berjudul pendahuluan dan berisi latar belakang pembahasan skripsi, tujuan pembahasan skripsi, waktu dan tempat pelaksanaan skripsi, metodologi penulisan skripsi, serta sistematika penulisan skripsi. Kemudian bab yang kedua berjudul dasar teori dan secara garis besar berisi teori-teori yang terutama berkaitan dengan judul dari skripsi ini, di antaranya adalah penjelasan tentang PLL dan penjelasan singkat mengenai metode sinkronisasi lainnya yang diketahui.

Lalu pada bab ketiga yang berjudul perancangan simulasi PLL, akan dijelaskan mengenai proses perancangan simulasi secara tahap demi tahap, serta proses dan alasan penentuan parameter-parameter yang digunakan dalam proses simulasi tersebut. Berikutnya, bab keempat yang berjudul hasil simulasi dan analisis PLL, akan dijelaskan tentang langkah-langkah dan penjelasan hasil simulasi yang diperoleh, serta logika-logika berpikir yang digunakan untuk pengembangan metode PLL yang bersangkutan agar dapat memberikan hasil yang lebih baik lagi.

Kemudian pada bab kelima yang berjudul aplikasi PLL dalam *grid connected PV*, akan diperlihatkan hasil simulasi dari PLL (yang telah

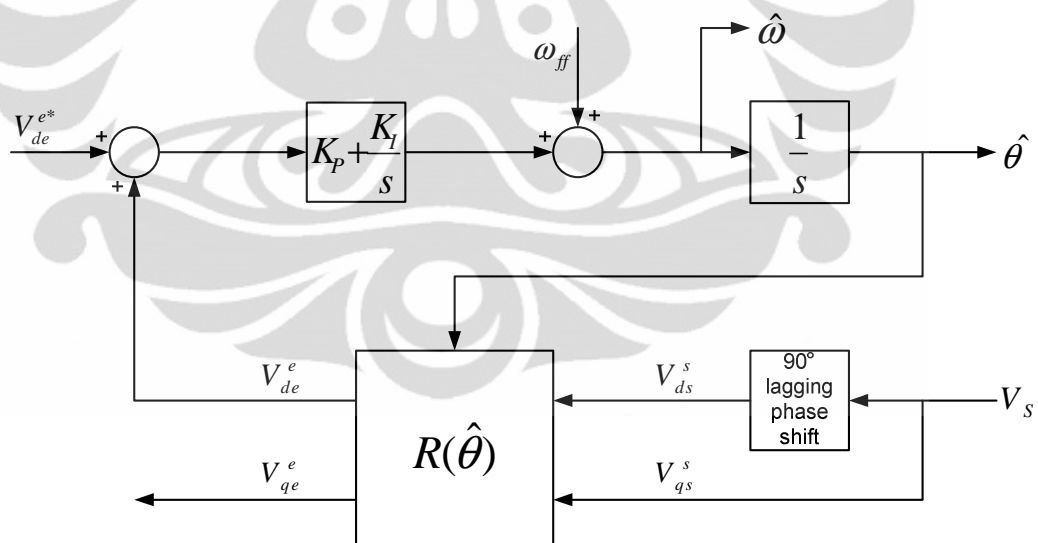
dikembangkan pada bab ketiga dan ditunjukkan performanya pada bab keempat) dengan *current control* untuk diaplikasikan dalam proses sinkronisasi antara sinyal tegangan PV dengan sinyal tegangan *grid*. Dan akhirnya pada bab keenam yang berjudul kesimpulan, akan disimpulkan hal-hal penting yang berasal dari aktivitas pengerjaan skripsi ini.



BAB 2 DASAR TEORI

2.1 Dasar Mengenai PLL dengan Perhitungan Berbasis Tegangan

PLL merupakan suatu blok (rangkaiannya maupun algoritma) yang digunakan untuk menghasilkan sinyal keluaran yang sudut fasa dan frekuensinya sama dengan suatu sinyal referensi tertentu (Langton, 1998). Secara umum, terdapat PLL analog yang terdiri atas blok-blok komponen tertentu dan PLL digital yang dapat diimplementasikan dengan baris-baris program. Dalam skripsi ini, akan digunakan PLL digital yang diimplementasikan menjadi baris-baris program dalam bahasa C yang disimulasikan dengan menggunakan SIMULINK (berupa *s-function*), baik untuk PLL yang perhitungannya didasarkan atas tegangan maupun untuk PLL yang perhitungannya didasarkan atas daya. Blok diagram untuk PLL yang didasarkan atas perhitungan tegangan dapat dilihat dalam Gambar 2.1 (Mok, Choe, Kim, Lee, & Suh, 2008).



Gambar 2.1. Blok diagram PLL dengan perhitungan berbasis tegangan

Keterangan Gambar 2.1 adalah sebagai berikut :

V_s = nilai tegangan sinusoidal sumber

V_{ds}^s = nilai tegangan sinusoidal sumber yang fasanya digeser agar tertinggal sebesar 90° dari tegangan sinusoidal sumber

$V_{qs}^s = V_s$

V_{de}^e = komponen sumbu d dari tegangan sinusoidal sumber yang ditransformasi dengan Transformasi DQ

V_{qe}^e = komponen sumbu q dari tegangan sinusoidal sumber yang ditransformasi dengan Transformasi DQ

V_{de}^{e*} = nilai V_{de}^e target (yang diinginkan)

K_p = nilai konstanta proporsional pengendali PI

K_I = nilai konstanta integral pengendali PI

ω_{ff} = asumsi awal dari nilai frekuensi sudut tegangan sinusoidal sumber

$\hat{\omega}$ = nilai estimasi frekuensi sudut tegangan sinusoidal sumber

$\hat{\theta}$ = nilai estimasi sudut fasa tegangan sinusoidal sumber

$\mathbf{R}(\hat{\theta})$ = operator Transformasi DQ dengan parameter $\hat{\theta}$

Ide dari PLL tegangan ini adalah dengan memproses input berupa tegangan sinusoidal aktual dari PLN, yang akan disimulasikan dengan fungsi kosinus, serta input berupa tegangan sinusoidal aktual dari PLN yang telah digeser fasanya agar tertinggal sebesar 90° dari tegangan sinusoidal aktual dari PLN tersebut, sehingga menjadi berupa fungsi sinus (untuk melakukan pergeseran fasa sebesar 90° , pada awalnya akan digunakan prinsip / karakteristik pergeseran fasa secara manual dengan menggunakan *state* dalam *s-function*). Tujuan dari dilakukannya pergeseran ini adalah agar Transformasi DQ dapat digunakan terhadap kedua input tegangan tersebut. Jadi secara rumusan, dapat dituliskan sebagai berikut :

$$V_{qs}^s = V_s = V_m \cos(\theta) \quad (2.1)$$

$$V_{ds}^s = V_m \cos(\theta - 90^\circ) = V_m \sin(\theta) \quad (2.2)$$

Kemudian, dengan melakukan Transformasi DQ, akan diperoleh nilai tegangan hasil transformasi sebagai berikut :

$$\begin{bmatrix} V_{qe}^e \\ V_{de}^e \end{bmatrix} = \mathbf{R}(\hat{\theta}) \begin{bmatrix} V_{qs}^s \\ V_{ds}^s \end{bmatrix} \quad (2.3)$$

$$\begin{bmatrix} V_{qe}^e \\ V_{de}^e \end{bmatrix} = \begin{bmatrix} \cos(\hat{\theta}) & \sin(\hat{\theta}) \\ -\sin(\hat{\theta}) & \cos(\hat{\theta}) \end{bmatrix} \begin{bmatrix} V_m \cos(\theta) \\ V_m \sin(\theta) \end{bmatrix} \quad (2.4)$$

$$\begin{bmatrix} V_{qe}^e \\ V_{de}^e \end{bmatrix} = \begin{bmatrix} V_m \cos(\theta) \cos(\hat{\theta}) + V_m \sin(\theta) \sin(\hat{\theta}) \\ -V_m \cos(\theta) \sin(\hat{\theta}) + V_m \sin(\theta) \cos(\hat{\theta}) \end{bmatrix} \quad (2.5)$$

$$\begin{bmatrix} V_{qe}^e \\ V_{de}^e \end{bmatrix} = V_m \begin{bmatrix} \cos(\theta) \cos(\hat{\theta}) + \sin(\theta) \sin(\hat{\theta}) \\ \sin(\theta) \cos(\hat{\theta}) - \cos(\theta) \sin(\hat{\theta}) \end{bmatrix} \quad (2.6)$$

$$\begin{bmatrix} V_{qe}^e \\ V_{de}^e \end{bmatrix} = V_m \begin{bmatrix} \cos(\theta - \hat{\theta}) \\ \sin(\theta - \hat{\theta}) \end{bmatrix} \quad (2.7)$$

Di sini dapat dilihat bahwa hasil Transformasi DQ yang dilakukan merupakan fungsi trigonometri (sinus dan kosinus) dari perbedaan fasa yang terjadi antara fasa tegangan sinusoidal aktual dari PLN dengan fasa dari hasil estimasi yang dilakukan. Oleh karena tujuan yang ingin dicapai adalah menyamakan fasa antara tegangan sinusoidal aktual PLN dengan fasa hasil estimasi, maka ide berikutnya adalah dengan membuat nilai θ menjadi sama dengan $\hat{\theta}$, atau dengan kata lain, nilai $(\theta - \hat{\theta})$ hendak dibuat menjadi sama dengan nol. Sehingga persamaan sebelumnya menjadi :

$$\begin{bmatrix} V_{qe}^e \\ V_{de}^e \end{bmatrix} = V_m \begin{bmatrix} \cos(0) \\ \sin(0) \end{bmatrix} \quad (2.8)$$

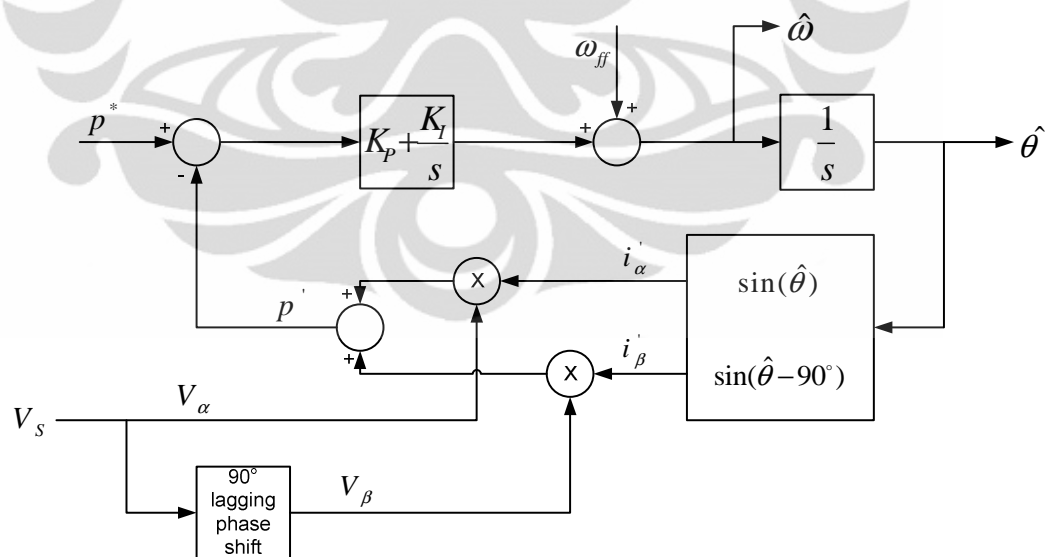
$$\begin{bmatrix} V_{qe}^e \\ V_{de}^e \end{bmatrix} = \begin{bmatrix} V_m \\ 0 \end{bmatrix} \quad (2.9)$$

Di sini didapatkan suatu syarat yang akan digunakan sebagai algoritma pemrograman PLL tegangan yang akan dirancang, yaitu saat fasa dari tegangan sinusoidal PLN sama dengan fasa hasil estimasi, maka nilai V_{qe}^e akan sama dengan nilai amplitudo tegangan sinusoidal PLN dan nilai V_{de}^e akan sama dengan nol. Untuk perancangan PLL tegangan ini, akan digunakan nilai V_{de}^e untuk terus-menerus dibandingkan dengan nilai V_{de}^{e*} , yang ditentukan sama dengan nol. Lalu, selama fasa tegangan sinusoidal PLN belum sama dengan fasa estimasi, nilai V_{de}^e akan berbeda dengan nilai V_{de}^{e*} . Nilai *error* yang timbul dari hasil penjumlahan antara V_{de}^e dan V_{de}^{e*} ini akan diproses oleh pengendali PI yang kemudian akan menghasilkan sinyal pengendali berupa frekuensi sudut kompensasi yang perlu

ditambahkan dengan nilai ω_{ff} agar diperoleh nilai $\hat{\omega}$, yang jika diintegrasikan terhadap waktu, akan menghasilkan nilai fasa estimasi yang sesuai dengan fasa tegangan sinusoidal aktual PLN. Demikianlah algoritma ini akan terus diulangi hingga diperoleh nilai fasa estimasi yang tepat.

2.2 Dasar Mengenai PLL dengan Perhitungan Berbasis Daya

Secara keseluruhan, PLL yang menggunakan dasar perhitungan terhadap daya terlihat mirip dengan PLL yang menggunakan tegangan sebagai dasar perhitungannya. Kesimpulan kemiripan ini terutama terlihat dari blok diagramnya. Akan tetapi yang perlu diperhatikan adalah bahwa PLL yang menggunakan dasar perhitungan terhadap daya ini memiliki prinsip perhitungan yang berbeda dengan PLL yang menggunakan dasar perhitungan terhadap tegangan. Hal ini terutama terlihat dari tidak digunakannya Transformasi DQ dalam PLL yang menggunakan dasar perhitungan terhadap daya. Gambar 2.2 menampilkan blok diagram / skema dari PLL yang menggunakan daya sebagai basis perhitungannya (Silva, Novochadlo, & Modesto, 2008).



Gambar 2.2. Blok diagram PLL dengan perhitungan berbasis daya

Keterangan untuk Gambar 2.2 adalah sebagai berikut :

V_s = nilai tegangan sinusoidal sumber

V_α = V_s

- V_β = nilai tegangan sinusoidal sumber yang fasanya digeser agar tertinggal sebesar 90° dari tegangan sinusoidal sumber
 K_p = nilai konstanta proporsional pengendali PI
 K_I = nilai konstanta integral pengendali PI
 ω_{ff} = asumsi awal dari nilai frekuensi sudut tegangan sinusoidal sumber
 $\hat{\omega}$ = nilai estimasi frekuensi sudut tegangan sinusoidal sumber
 $\hat{\theta}$ = nilai estimasi sudut fasa tegangan sinusoidal sumber
 i'_α = nilai estimasi arus sinusoidal imajiner yang fasanya digeser agar tertinggal sebesar 90° dari sudut fasa tegangan sinusoidal sumber
 i'_β = nilai estimasi arus sinusoidal imajiner yang fasanya tertinggal sebesar 90° dari sudut fasa i'_α
 p' = nilai estimasi daya imajiner total
 p^* = nilai p' target (yang diinginkan)

Untuk PLL daya, akan dilakukan proses awal yang sama dengan yang dilakukan pada PLL tegangan, yaitu mendefinisikan input berupa tegangan aktual PLN (berupa fungsi kosinus) dan tegangan aktual PLN yang fasanya digeser agar tertinggal sebesar 90° dari tegangan aktual PLN yang bersangkutan (berupa fungsi sinus). Kemudian, diasumsikan dua buah nilai arus sinusoidal imajiner, yang mana arus sinusoidal imajiner yang pertama merupakan fungsi sinus beramplitudo satu dari nilai fasa hasil estimasi, sedangkan arus sinusoidal imajiner yang kedua merupakan fungsi sinus beramplitudo satu dari nilai fasa estimasi yang fasanya digeser 90° agar tertinggal dari fasa arus sinusoidal imajiner yang pertama (berupa fungsi minus kosinus). Secara rumusan, keempat besaran ini dituliskan sebagai berikut :

$$V_\alpha = V_s = V_m \cos(\theta) \quad (2.10)$$

$$V_\beta = V_m \cos(\theta - 90^\circ) = V_m \sin(\theta) \quad (2.11)$$

$$i'_\alpha = \sin(\hat{\theta}) \quad (2.12)$$

$$i'_\beta = \sin(\hat{\theta} - 90^\circ) = -\cos(\hat{\theta}) \quad (2.13)$$

Kemudian, hal berikutnya yang dilakukan adalah menjumlahkan hasil perkalian antara V_α dan i'_α dengan hasil perkalian antara V_β dan i'_β , sehingga akhirnya diperoleh suatu nilai estimasi daya imajiner total (yang dilambangkan dengan p'). Jadi nilai p' ini secara rumusan adalah sebagai berikut :

$$p' = V_\alpha \cdot i'_\alpha + V_\beta \cdot i'_\beta \quad (2.14)$$

$$p' = V_m \cos(\theta) \cdot \sin(\hat{\theta}) + V_m \sin(\theta) \cdot [-\cos(\hat{\theta})] \quad (2.15)$$

$$p' = -V_m [\sin(\theta) \cdot \cos(\hat{\theta}) - \cos(\theta) \cdot \sin(\hat{\theta})] \quad (2.16)$$

$$p' = -V_m \sin(\theta - \hat{\theta}) \quad (2.17)$$

Dari persamaan yang terakhir, jika nilai fasa estimasi sama dengan fasa tegangan sinusoidal aktual PLN (nilai θ sama dengan nilai $\hat{\theta}$), maka :

$$p' = -V_m \sin(0) \quad (2.18)$$

$$p' = 0 \quad (2.19)$$

Dengan demikian, diperoleh suatu syarat yang kemudian akan diterjemahkan menjadi algoritma pemrograman PLL daya yang akan dirancang. Yaitu, nilai dari p' akan sama dengan nol saat fasa estimasi sama dengan fasa tegangan sinusoidal aktual PLN. Jadi, seperti yang dapat terlihat pada blok diagram (Gambar 2.2), nilai dari p' yang diperoleh akan terus-menerus dibandingkan dengan nilai p^* yang sama dengan nol. Tujuan akhirnya adalah agar nilai p' menjadi nol juga (sebab hal ini menunjukkan kesesuaian / kesamaan fasa).

Kemudian, langkah perhitungan berikutnya pada PLL daya ini serupa dengan langkah perhitungan pada PLL tegangan. Blok diagram PLL daya memperlihatkan bahwa *error* hasil pengurangan antara nilai p^* dengan p' akan diproses oleh pengendali PI, yang mana sinyal pengendali dari PI ini akan berupa frekuensi sudut kompensasi yang perlu ditambahkan dengan ω_{ff} agar diperoleh nilai $\hat{\omega}$, yang jika diintegrasikan terhadap waktu, akan menghasilkan nilai fasa

estimasi yang sesuai dengan fasa tegangan sinusoidal aktual PLN. Demikianlah algoritma ini akan terus diulangi hingga diperoleh nilai fasa estimasi yang tepat.

2.3 Perbandingan Tiga Metode Sinkronisasi Lainnya dengan PLL

Pada bagian ini, akan dilihat dan dianalisis tiga metode sinkronisasi lainnya jika dibandingkan dengan metode PLL. Ketiga metode yang dimaksudkan adalah metode *zero-crossing*, metode *utility observer*, dan metode *harmonic compensator*. Namun yang perlu diperhatikan adalah bahwa ketiga metode tersebut tidak akan dibahas secara mendalam, akan tetapi hanya akan dilihat mengenai deskripsi umum ketiga metode tersebut untuk dibandingkan dengan metode PLL.

Untuk metode *zero-crossing*, ide utamanya adalah dengan membangkitkan pulsa-pulsa impuls dengan frekuensi tinggi yang kemudian disesuaikan dengan kondisi sinyal sinusoidal tegangan listrik AC dari PLN, yaitu dengan membandingkan sinyal tegangan PLN dengan nilai / kondisi nol. Hasil penyesuaian ini lalu dicocokkan dengan kondisi periode (atau dengan kata lain dicocokkan dengan kondisi frekuensi) sebenarnya dari sinyal tegangan PLN. Akhirnya, dengan data-data sinkronisasi inilah, sinyal tegangan PLN dapat diketahui karakteristiknya pada suatu waktu tertentu untuk kemudian disinkronisasikan dengan sinyal tegangan AC yang akan dibangkitkan kemudian (Syamsuddin, Rahim, Krismadinata, & Selvaraj). Kelemahan dari metode ini adalah bahwa untuk bentuk sinyal tegangan yang terdistorsi, metode ini dapat memberikan kesalahan, sebab dalam keadaan terdistorsi, sinyal tegangan PLN dapat menyentuh nilai nol lebih dari dua kali dalam satu periode, sehingga pendeteksian periode (atau frekuensi) sinyal tegangan PLN dapat menjadi salah.

Metode lainnya yang dikenal adalah metode *harmonic compensator*, yang seperti namanya, cara kerjanya dapat dijelaskan sebagai berikut : *harmonic compensator* ini akan mendeteksi sinyal keluaran dari *inverter* (yang akan disinkronisasikan dengan sinyal tegangan PLN), kemudian *harmonic compensator* akan memberikan sinyal 'suntikan' / sinyal kompensasi untuk mengubah kinerja dan output dari *inverter*, sehingga dapat disinkronkan dengan sinyal tegangan dari PLN. Walaupun metode ini cukup baik karena sifatnya yang memperbaiki *error*

yang terdeteksi, namun metode ini cukup rumit untuk direalisasikan, oleh karena keperluan untuk mengenal sistem *inverter*, sistem dari *harmonic compensator* itu sendiri, serta karakteristik dan analisis superposisi antara sinyal tegangan PLN dengan sinyal yang diberikan oleh *harmonic compensator* yang bersangkutan (Castilla, Miret, Matas, Vicuña, & Guerrero, 2008).

Dan metode terakhir yang akan dijelaskan adalah metode *utility observer*, yang mana metode ini melakukan estimasi terhadap kondisi sinyal tegangan PLN yang sebenarnya berdasarkan karakteristik-karakteristik lainnya yang dapat diukur secara lebih mudah daripada karakteristik sinyal tegangan yang hendak diketahui tersebut (kinerja ini sesuai dengan fungsi yang dilakukan oleh perangkat *observer*). Akan tetapi, kesulitan yang dihadapi jika metode *utility observer* ini digunakan adalah serupa dengan kesulitan yang dihadapi jika metode *harmonic compensator* digunakan, yaitu metode ini memerlukan analisis prinsip dan matematis yang relatif kompleks dalam mengaitkan antara suatu besaran tertentu yang diukur dengan besaran yang ingin diketahui dari sinyal tegangan PLN yang bersangkutan (Rey, Pablo, Ruiz, & Ruz, 2005).

Berdasarkan atas penjelasan tersebut di atas, metode PLL memiliki kelebihan dari sisi kesederhanaannya, akan tetapi output yang diberikannya relatif baik (sesuai dengan yang dikehendaki), yaitu dapat memberikan data output berupa frekuensi, fasa, dan amplitudo dari sinyal tegangan PLN yang ingin disinkronisasikan, atau dalam kasus ini, direkonstruksi ulang secara sesempurna mungkin. Hal ini karena metode PLL tidak melibatkan hubungan-hubungan matematis yang kompleks antara besaran-besaran tertentu, akan tetapi berdasarkan atas hubungan-hubungan dasar prinsip-prinsip yang logis dan relevan terhadap besaran-besaran tersebut. Kemudian, dengan metode-metode dan prinsip-prinsip dasar tersebut, dikembangkanlah perlakuan-perlakuan lanjutan lainnya yang dapat semakin memperbaiki dan menyempurnakan output dari PLL dasar / awal yang direalisasikan sebelumnya.

Selain itu, PLL merupakan metode yang secara umum dapat dikembangkan lagi lebih lanjut dengan berbagai jenis metode tambahan, terutama untuk PLL digital yang direalisasikan dengan menggunakan baris-baris program, sehingga lebih fleksibel jika dibandingkan dengan PLL analog. Jadi hal-hal yang

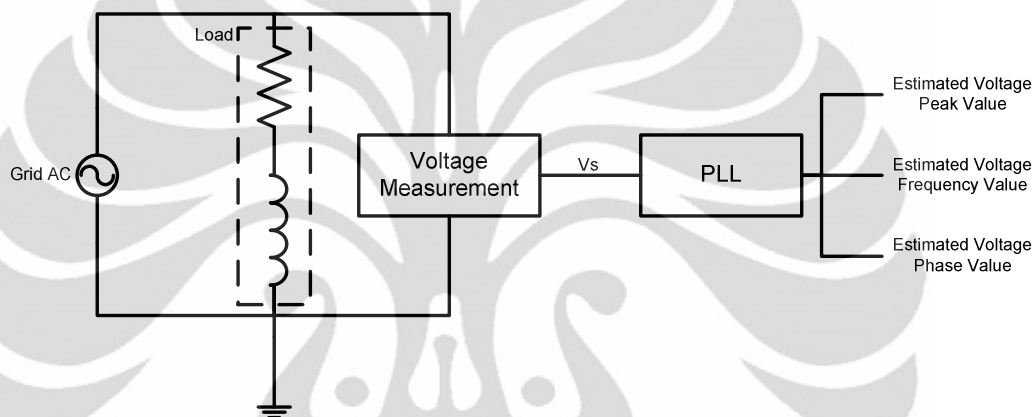
akan dilaporkan berikut ini dapat menjadi bukti-bukti nyata bahwa PLL tidak dapat secara universal dikatakan memberikan hasil yang relatif kurang begitu baik jika dibandingkan dengan metode sinkronisasi lainnya (Sujanarko, 2010).



BAB 3 PERANCANGAN SIMULASI PLL

3.1 Blok Simulasi Awal dan Penjelasan

Gambar 3.1 menunjukkan blok diagram awal yang direalisasikan untuk mensimulasikan PLL yang diperoleh dari teori awal yang telah dibahas pada bagian dasar teori, dengan menggunakan program SIMULINK yang terdapat pada MATLAB versi R2009a.



Gambar 3.1. Blok diagram PLL awal

Gambar 3.1 menjelaskan alur penggunaan blok algoritma PLL yang akan didesain. Pertama-tama, terdapat sebuah blok yang mewakili (merepresentasikan) tegangan yang terdapat pada *grid* (tegangan PLN). Blok ini dibentuk dalam program SIMULINK dengan menggunakan *controlled voltage source*. Kemudian blok *Load* digunakan untuk merepresentasikan beban yang dicatu oleh tegangan PLN yang bersangkutan. Bagian *Load* ini dibentuk dalam program SIMULINK dengan menggunakan komponen resistor dan induktor (karena beban-beban yang biasanya dicatu oleh PLN adalah beban yang bersifat induktif) yang disediakan dalam *library* SIMULINK. Lalu, terdapat blok *Voltage Measurement* yang digunakan untuk mengukur kondisi tegangan yang dicatu oleh PLN, untuk kemudian, sinyal tegangan yang diukur (dinotasikan sebagai V_s) akan diproses oleh algoritma PLL yang akan dibuat. Blok *Voltage Measurement* ini dibentuk dalam program SIMULINK dengan menggunakan *voltage measurement* yang juga terdapat dalam *library* SIMULINK. Dan akhirnya, nilai V_s yang terukur akan

diproses oleh blok algoritma PLL (sesuai dengan alur yang telah digambarkan pada Gambar 2.1 dan Gambar 2.2), yang mana algoritma PLL ini dibentuk dengan menggunakan fasilitas *s-function* yang disediakan oleh MATLAB.

3.2 Parameter PLL Tegangan dan PLL Daya yang Digunakan

Phase Locked Loop dengan perhitungan berbasis tegangan (Gambar 2.1) yang akan digunakan dalam proses estimasi amplitudo, frekuensi, serta sudut fasa sinyal tegangan listrik *grid* menggunakan parameter K_P senilai 1, K_I senilai 100, ω_{ff} senilai $2\pi \cdot 50$ rad/s, serta V_{de}^{e*} senilai nol sebagai parameter sinkronisasi (nilai K_P dan K_I ini ditentukan secara *trial and error*, karena dinilai lebih efektif jika dibandingkan dengan melakukan penentuan secara matematis terhadap sistem yang relatif kompleks). Kemudian, yang perlu diperhatikan adalah bahwa operator *feedback* yang digunakan, seperti yang terlihat pada Gambar 2.1, menggunakan tanda positif, dan bukan tanda negatif seperti yang umumnya ditemukan dalam sistem-sistem lain. Hal ini disebabkan karena nilai referensi yang digunakan, yaitu nilai V_{de}^{e*} , adalah nol, sehingga jika digunakan tanda negatif, sinyal *error* yang memasuki blok pengendali akan bernilai negatif juga. Dengan demikian, nilai frekuensi estimasi, sudut fasa estimasi, serta amplitudo estimasi akan menjadi bernilai negatif, atau dengan kata lain menghasilkan nilai estimasi yang salah.

Lalu untuk PLL yang berbasis atas perhitungan daya (Gambar 2.2), parameter yang digunakan memiliki kemiripan nilai dengan parameter yang digunakan pada PLL yang menggunakan perhitungan berbasis tegangan, yaitu K_P senilai 1, K_I senilai 100, ω_{ff} senilai $2\pi \cdot 50$ rad/s, serta p^* senilai nol sebagai parameter sinkronisasi (nilai K_P dan K_I ini, sama seperti dalam kasus PLL dengan perhitungan berbasis tegangan, ditentukan secara *trial and error*, karena dinilai lebih efektif jika dibandingkan dengan melakukan penentuan secara matematis terhadap sistem yang relatif kompleks).

3.3 Low Pass Filter untuk Pengembangan PLL Tegangan

Seperti yang nanti akan dilihat pada bab keempat, permasalahan yang ditemukan pada PLL tegangan adalah osilasi pada sinyal output, terutama pada output yang berupa estimasi frekuensi dan estimasi amplitudo. Masalah ini diatasi

dengan menambahkan *low pass filter* untuk meredam osilasi tersebut. *Low pass filter* yang digunakan adalah *low pass filter* berorde satu dengan bentuk umum fungsi alih sebagai berikut :

$$\frac{Y(s)}{U(s)} = \frac{1}{\tau \cdot s + 1} \quad (3.1)$$

Yang mana $U(s)$ merupakan input *low pass filter* yang berupa sinyal yang ingin diredam osilasinya, $Y(s)$ merupakan output *low pass filter* yang berupa sinyal yang telah diredam osilasinya, serta τ merupakan nilai kebalikan dari frekuensi *cut-off* (frekuensi minimal dari sinyal yang akan diredam) yang diaplikasikan pada *low pass filter*.

Untuk kepentingan perancangan ini, digunakan frekuensi *cut-off* sebesar 25 hertz yang berasal dari analisis terhadap osilasi sinyal output yang hendak diredam, yang mana frekuensi osilasinya adalah sebesar sekitar 25 hertz. Dengan demikian, *low pass filter* yang dirancang akan mampu meredam osilasi yang terjadi pada sinyal output, karena *low pass filter* tersebut dirancang untuk meredam osilasi dengan frekuensi 25 hertz ke atas.

3.4 All Pass Filter untuk Pengembangan PLL Tegangan dengan Low Pass Filter

Akan terlihat pada bab keempat bahwa PLL tegangan dengan *low pass filter* memiliki kekurangan berupa lambatnya respon. Analisis yang akan dilakukan nanti menunjukkan bahwa penyebabnya kemungkinan besar adalah algoritma pergeseran fasa sebesar 90° yang kurang efektif (yaitu dengan menggunakan algoritma pergeseran *state* untuk menampung nilai yang sebelumnya, sehingga seolah-olah terjadi operasi pergeseran fasa (*lagging*)). Hal ini diatasi dengan menggunakan algoritma *all pass filter* yang memiliki bentuk umum fungsi alih sebagai berikut :

$$\frac{Y(s)}{U(s)} = \frac{\omega_c - s}{\omega_c + s} \quad (3.2)$$

Yang mana $U(s)$ merupakan input yang berupa sinyal yang hendak digeser fasanya sebesar 90° , $Y(s)$ merupakan output yang berupa sinyal yang telah digeser fasanya sebesar 90° , serta ω_c merupakan nilai frekuensi dari sinyal yang hendak digeser fasanya sebesar 90° . Dengan demikian, untuk perancangan *all pass filter* ini, digunakan nilai ω_c sebesar 50 hertz. Hal ini karena sinyal input yang hendak digeser fasanya sebesar 90° adalah sinyal tegangan *grid* (PLN) yang memiliki nilai frekuensi efektif sebesar 50 hertz.

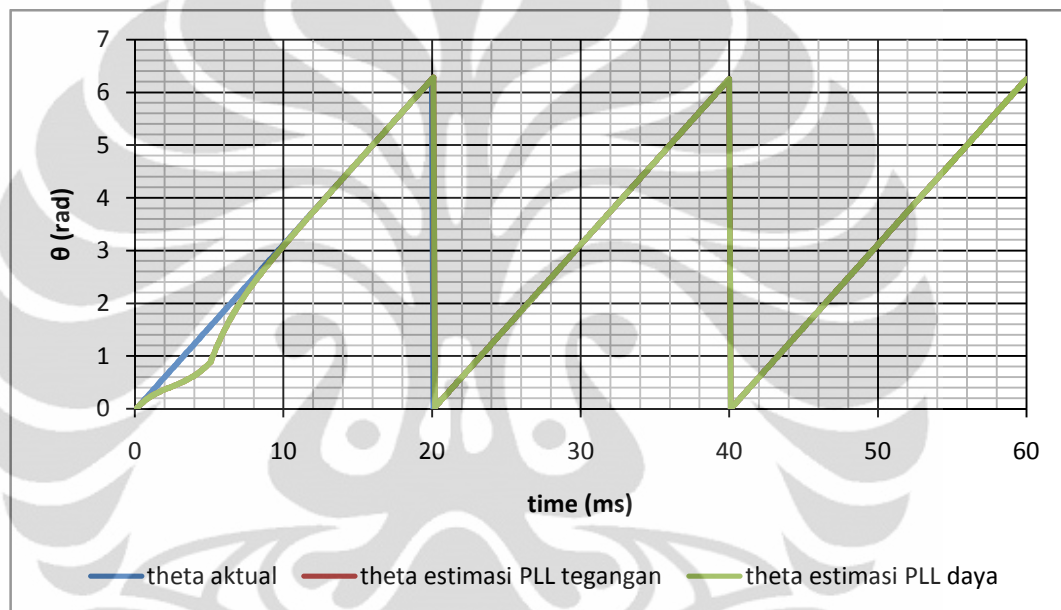
3.5 Penentuan *Load*, Induktor Konektor *Photovoltaic-Grid* (L_{grid}), dan Parameter-Parameter pada *Current Control*

Pada bab kelima akan ditampilkan performa dari PLL tegangan yang telah dimodifikasi dengan *low pass filter* dan *all pass filter*, dalam operasinya dengan blok / algoritma *current control*. Pada bagian tersebut akan disajikan simulasi secara keseluruhan dengan lengkap, dan hal itulah yang menjadi inti dari skripsi ini. Pada bagian tersebut, nilai-nilai komponen simulasi yang digunakan adalah *load* yang direpresentasikan oleh resistor dengan nilai resistansi 12 ohm dan induktor dengan nilai induktansi 51 milihenry, serta induktor sebagai konektor antara *photovoltaic* dengan *grid* (L_{grid}) yang bernilai 32 milihenry. Penentuan nilai komponen tersebut dilakukan dengan tujuan untuk dapat memperoleh nilai tegangan dan arus yang sesuai dengan kenyataan yang terdapat di lapangan (yang dijelaskan secara lebih rinci pada bab kelima). Kemudian, secara khusus untuk bagian *current control* (Gambar 5.5), nilai parameter yang digunakan adalah K_{pd} dan K_{pq} senilai 0.7, K_{id} dan K_{iq} senilai 70 (nilai K_p dan K_i ini ditentukan secara *trial and error*, karena dinilai lebih efektif jika dibandingkan dengan melakukan penentuan secara matematis terhadap sistem yang relatif kompleks), I_{qref} senilai 15 ampere (d disesuaikan dengan kebutuhan umum daya listrik rumah tangga, yang akan dijelaskan lebih lanjut di bab kelima), serta I_{dref} senilai nol sebagai parameter sinkronisasi.

BAB 4 HASIL SIMULASI DAN ANALISIS PLL

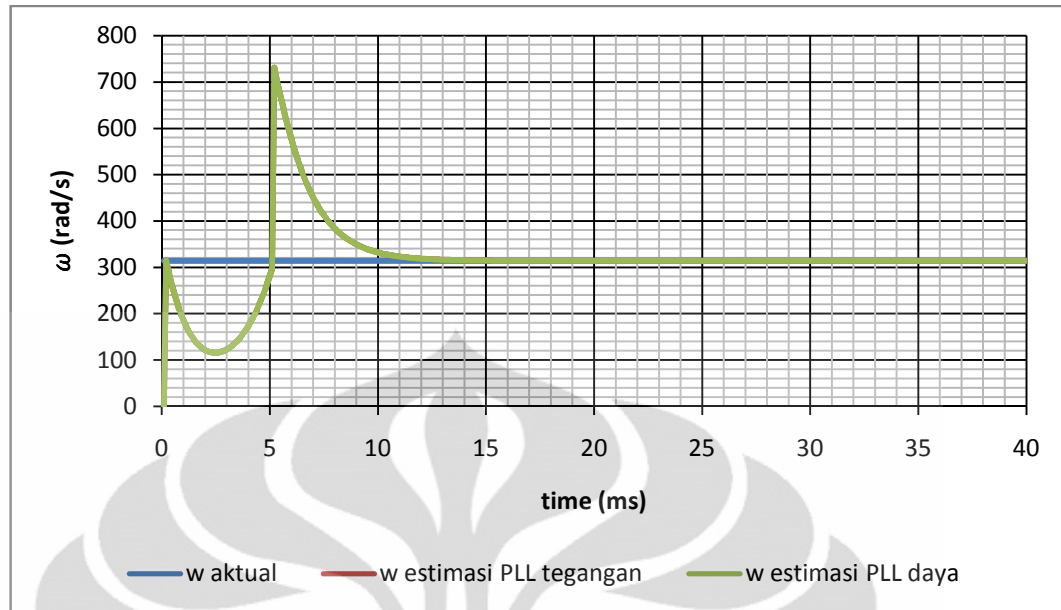
4.1 Hasil Simulasi Awal Kedua PLL dalam Kondisi Ideal

Berikut ini akan ditampilkan hasil simulasi dari kedua jenis PLL (PLL tegangan dan PLL daya) yang telah dibuat algoritmanya pada kondisi ideal, yaitu saat amplitudo dan frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 220 volt dan 50 hertz.



Gambar 4.1. θ estimasi untuk kedua jenis PLL

Dari Gambar 4.1 dan Gambar 4.2, dapat disimpulkan bahwa PLL yang berbasis perhitungan tegangan maupun daya memiliki persamaan performa (yang terlihat dari berhimpitnya garis merah, yang merupakan hasil estimasi dari PLL tegangan, dengan garis hijau, yang merupakan hasil estimasi dari PLL daya). Selain itu, perhitungan yang dilakukan dalam algoritma PLL tegangan terlihat lebih sedikit daripada algoritma PLL daya. Oleh karena itu, dalam pengembangan



Gambar 4.2. ω estimasi untuk kedua jenis PLL

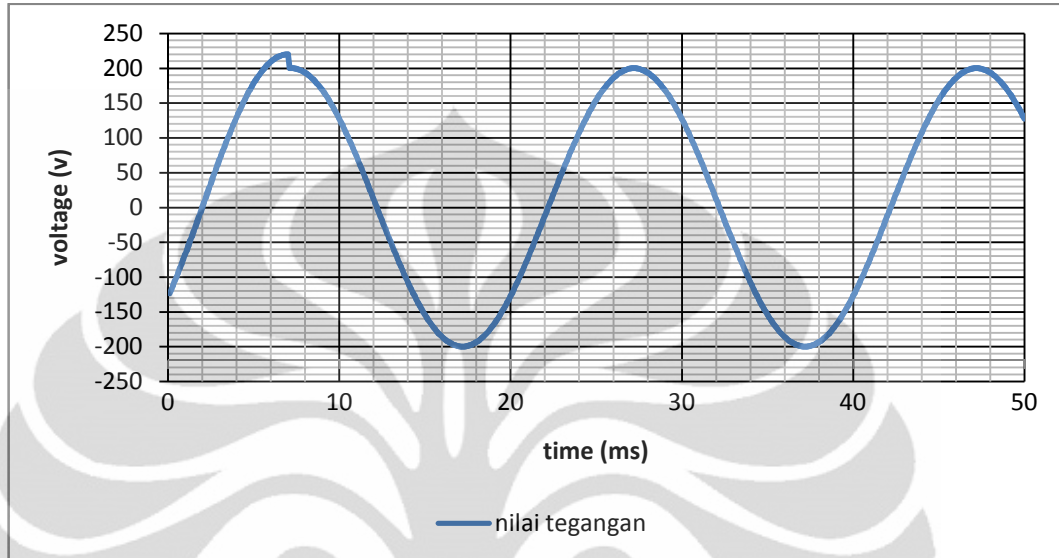
berikutnya, hanya akan dikonsentrasikan pada algoritma PLL tegangan saja, dengan catatan bahwa pengembangan-pengembangan algoritma yang diaplikasikan pada algoritma PLL tegangan juga dapat diaplikasikan pada PLL daya dengan hasil yang sama, sebab terdapat kesamaan prinsip perhitungan pada algoritma utama PLL tegangan dan PLL daya, seperti yang terlihat pada bab dasar teori, dan performanya pun membuktikan kesamaan tersebut.

4.2 Hasil Simulasi Awal Kedua PLL dalam Kondisi Perubahan Amplitudo

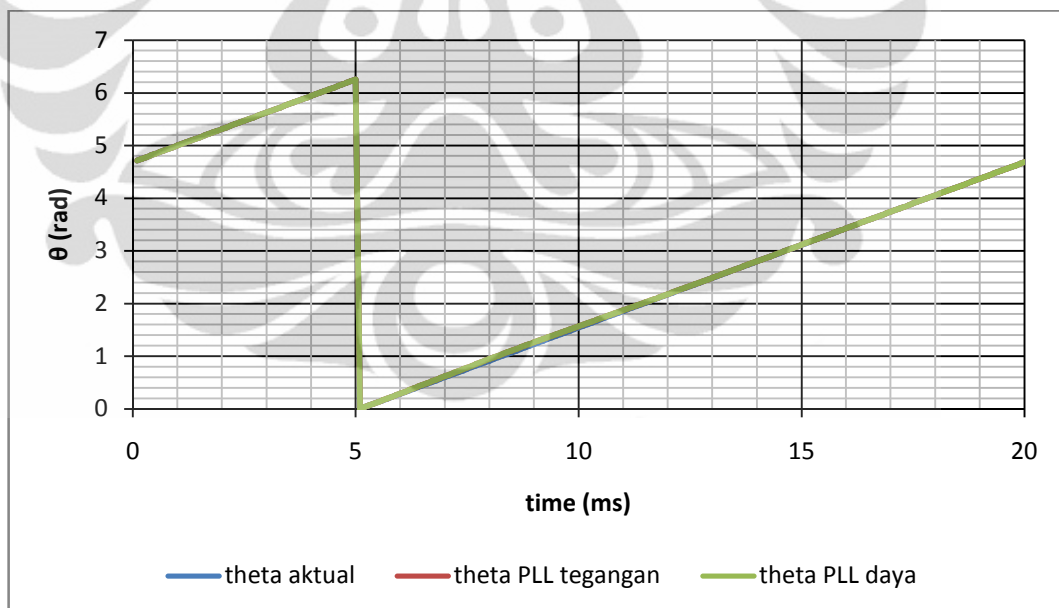
Berikut ini akan ditampilkan hasil simulasi dari kedua jenis PLL yang telah dibuat algoritmanya pada kondisi perubahan amplitudo, yaitu saat amplitudo sinyal tegangan listrik berubah dari 220 volt menjadi 200 volt pada waktu 5 milisekon, serta frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 50 hertz.

Terlihat dari Gambar 4.4 dan Gambar 4.5 bahwa kedua jenis algoritma PLL yang disimulasikan memberikan performa yang serupa, dan selain itu, juga

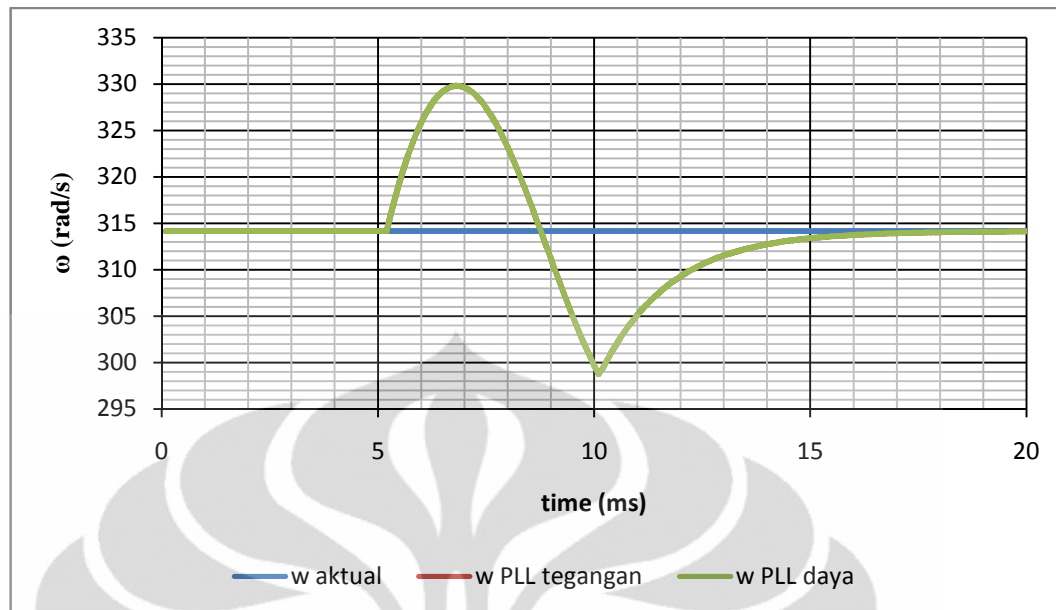
dapat dilihat bahwa perubahan amplitudo yang terjadi pada PLN dapat dideteksi oleh PLL dengan baik. Hal ini terlihat dari grafik θ yang terlihat tidak berubah secara signifikan, serta grafik ω yang mengalami sedikit distorsi sebelum akhirnya



Gambar 4.3. Sinyal tegangan PLN saat amplitudo berubah



Gambar 4.4. θ estimasi kedua jenis PLL saat amplitudo berubah



Gambar 4.5. ω estimasi kedua jenis PLL saat amplitudo berubah

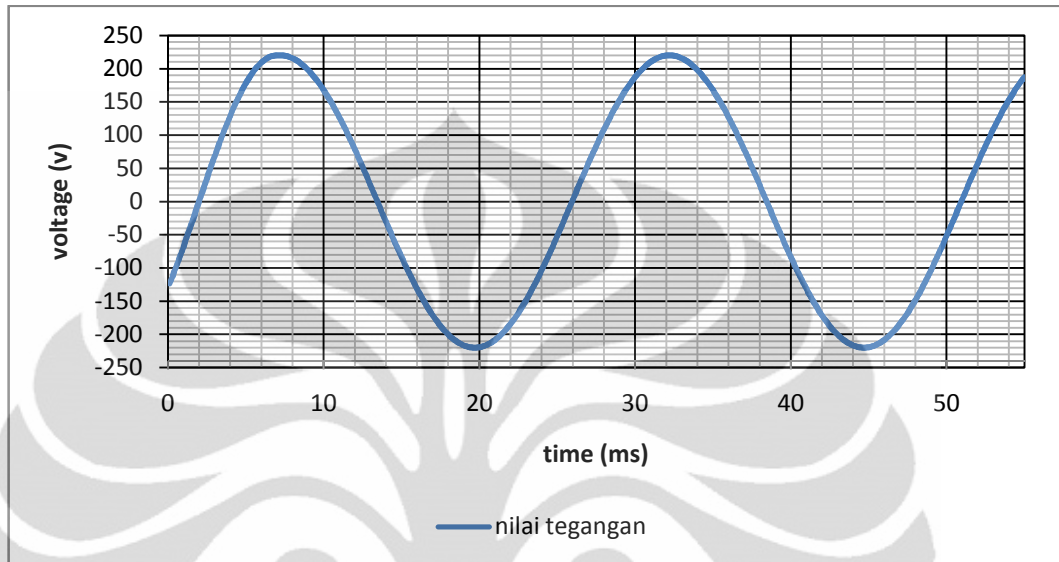
kembali ke titik kerja semula, dengan waktu transien sekitar 10 milisekon (hal ini juga terbukti dari Gambar 4.2 yang menunjukkan bahwa nilai transien untuk ω adalah sekitar 10 milisekon).

4.3 Hasil Simulasi Awal Kedua PLL dalam Kondisi Perubahan Frekuensi

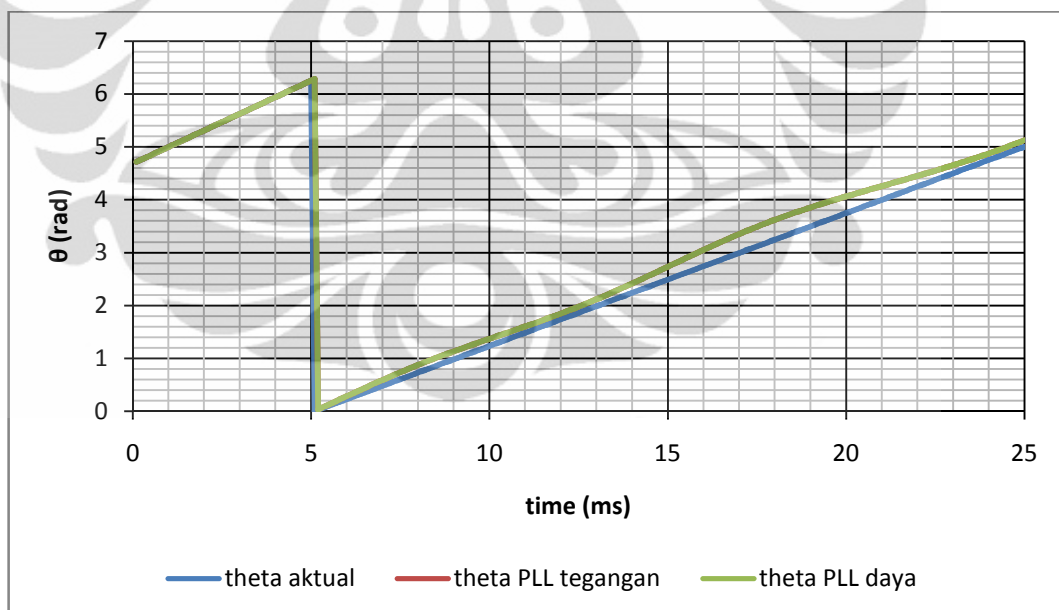
Berikut ini akan ditampilkan hasil simulasi dari kedua jenis PLL yang telah dibuat algoritmanya pada kondisi perubahan frekuensi, yaitu saat amplitudo sinyal tegangan listrik bersifat konstan pada nilai 220 volt, serta frekuensi dari sinyal tegangan listrik PLN berubah dari 50 hertz menjadi 40 hertz pada waktu 5 milisekon.

Kedua hasil pada Gambar 4.7 dan Gambar 4.8, pertama-tama menunjukkan kesamaan performa kedua jenis PLL, kemudian menunjukkan pula bahwa saat terjadi perubahan frekuensi, algoritma PLL yang dibuat kurang begitu mampu memberikan hasil pembacaan fasa dan frekuensi yang begitu tepat. Namun demikian, besaran yang paling terganggu hasil pengukurannya adalah

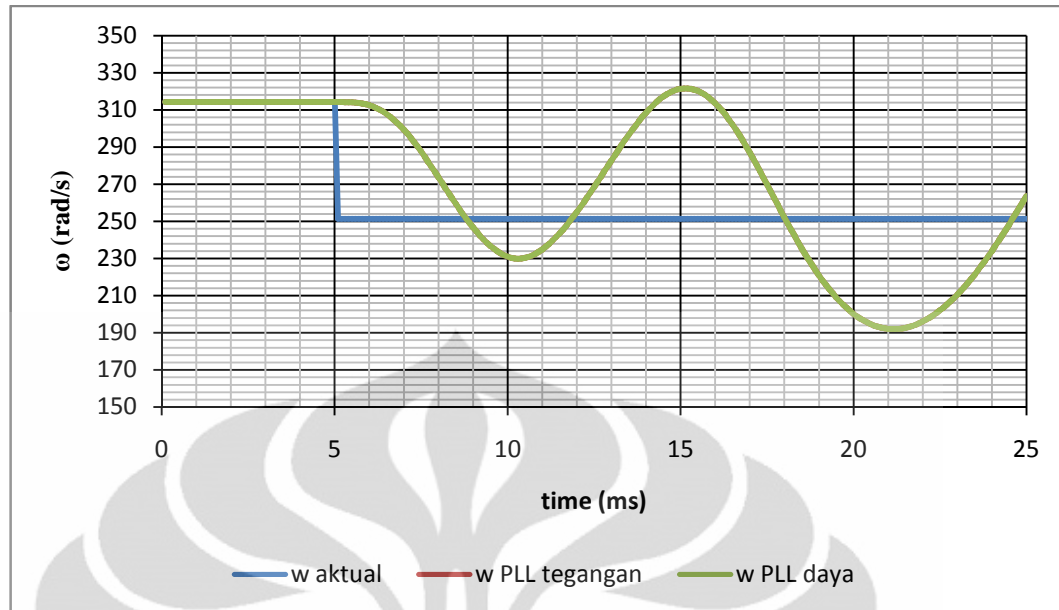
frekuensi estimasi dari PLL. Distorsi yang terjadi pada output dari PLL saat frekuensi sinyal tegangan listrik PLN berubah terjadi karena dua hal, yaitu : karena karakteristik transien PLL yang berlangsung ketika terjadi perubahan



Gambar 4.6. Sinyal tegangan PLN saat frekuensi berubah



Gambar 4.7. θ estimasi kedua jenis PLL saat frekuensi berubah



Gambar 4.8. ω estimasi kedua jenis PLL saat frekuensi berubah

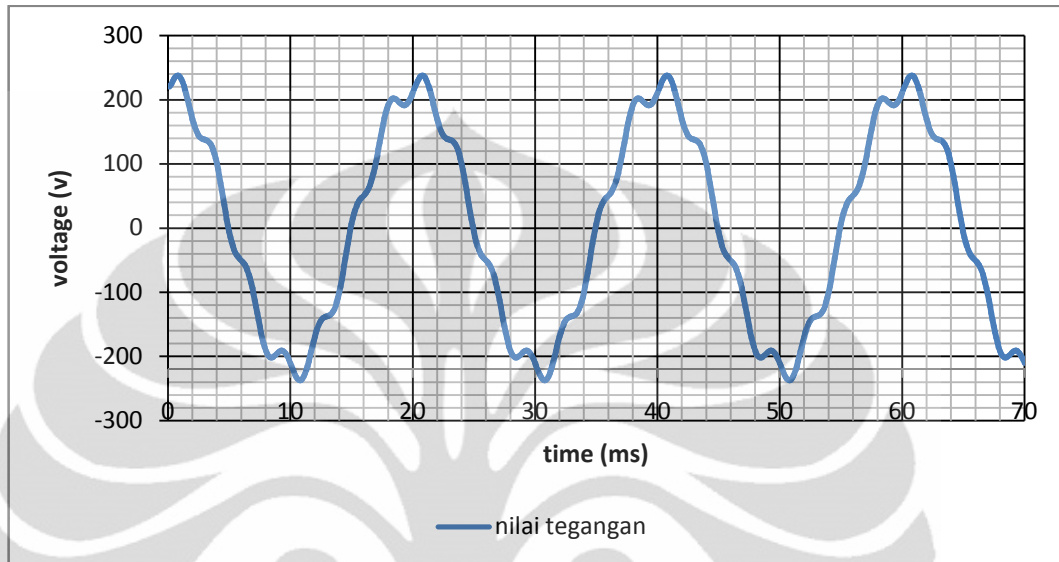
frekuensi itu sendiri, serta karena kurang tepatnya algoritma penggeser fasa (yang seharusnya menggeser fasa sinyal tegangan listrik PLN sebesar tepat 90 derajat) yang memanfaatkan proses pergeseran *state* dalam *s-function* dengan berdasarkan atas asumsi frekuensi sinyal tegangan listrik PLN yang bernilai faktual 50 hertz.

4.4 Hasil Simulasi Awal Kedua PLL dalam Kondisi Amplitudo Ber-noise

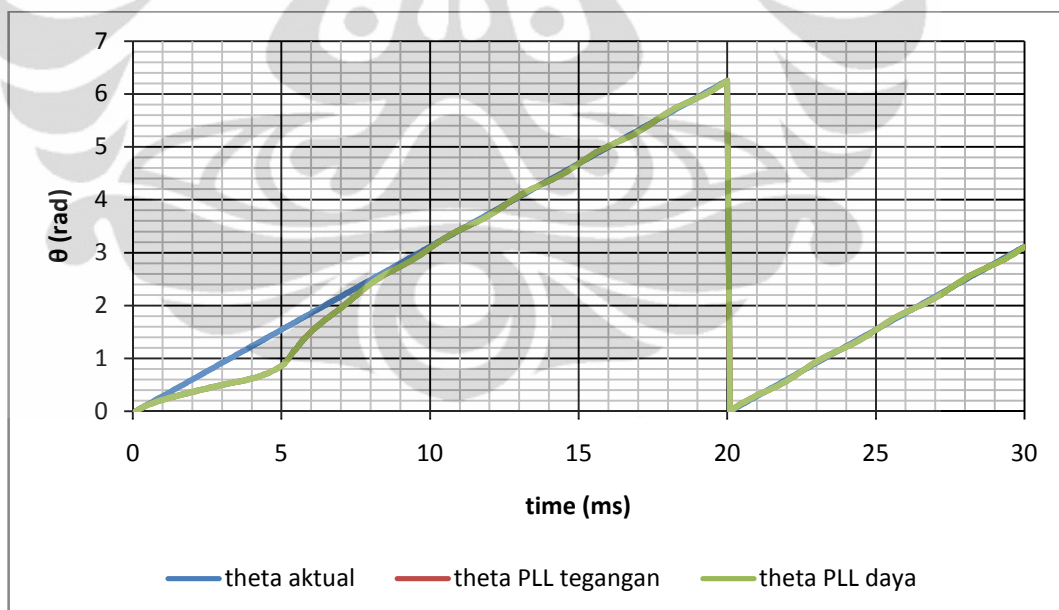
Berikut ini akan ditampilkan hasil simulasi dari kedua jenis PLL yang telah dibuat algoritmanya pada kondisi amplitudo memiliki *noise*, yaitu saat amplitudo sinyal tegangan listrik bersifat konstan pada nilai 220 volt, namun berinterferensi dengan harmonik ketujuhnya (frekuensinya $7 \times 50 \text{ hertz} = 350 \text{ hertz}$) yang memiliki amplitudo sepuluh persen dari amplitudo normalnya (amplitudonya $10\% \times 220 \text{ volt} = 22 \text{ volt}$), serta frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 50 hertz.

Sekali lagi, hasil pada Gambar 4.10 dan Gambar 4.11 menegaskan bahwa PLL tegangan dan PLL daya sama-sama memberikan respon yang sama, sebab

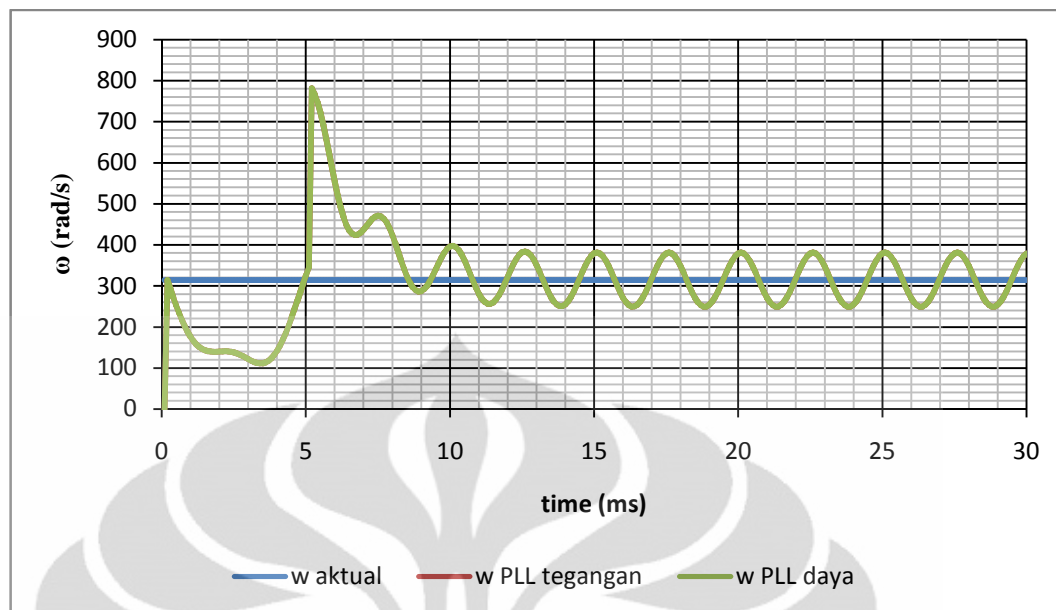
mereka memiliki bentuk rumus perhitungan yang sama walaupun diturunkan dari prinsip yang berbeda. Selain itu, dalam hasil yang diperoleh kali ini, terlihat bahwa output dari PLL yang paling sensitif terhadap gangguan yang terjadi adalah



Gambar 4.9. Sinyal tegangan PLN dengan noise



Gambar 4.10. θ estimasi kedua jenis PLL saat amplitudo ber-noise



Gambar 4.11. ω estimasi kedua jenis PLL saat amplitudo ber-noise

frekuensi estimasi. Hal ini disebabkan karena waktu transien untuk kondisi perubahan frekuensi yang terjadi adalah sekitar 10 milisekon, sedangkan faktanya, algoritma PLL yang dirancang mendeteksi frekuensi dari amplitudo yang disimpan ke dalam *state*. Akibatnya, saat terjadi distorsi berupa gangguan / *noise* pada amplitudo, nilai estimasi frekuensi akan menjadi terganggu pula. Dan oleh karena *noise* yang diberikan adalah berupa fungsi sinusoidal, maka bentuk distorsi pada frekuensi hasil estimasi akan berupa fungsi sinusoidal juga, sesuai dengan bentuk *noise* pada amplitudonya.

4.5 Analisis Lanjutan Hasil Simulasi Awal dan Pengembangan Algoritma

Telah dilihat di bagian sebelumnya bahwa permasalahan ketidaktepatan estimasi yang terjadi disebabkan terutama karena ketidaktepatan pergeseran fasa yang dilakukan oleh algoritma PLL yang dibuat. Dari titik ini, terdapat dua buah solusi yang dapat direalisasikan, yaitu membuat suatu algoritma tambahan yang berfungsi untuk mengadaptasi perubahan frekuensi yang terjadi. Dengan demikian, jika terjadi perubahan frekuensi, algoritma ini akan mengubah jumlah pergeseran *state* pada algoritma utama, sehingga algoritma utama dapat selalu menggeser fasa sinyal tegangan PLN sebesar tepat 90 derajat. Akan tetapi ada dua kesulitan dalam merealisasikan algoritma ini. Kesulitan yang pertama adalah jika dibuat suatu algoritma sederhana yang prinsipnya selalu menggeser fasa saat

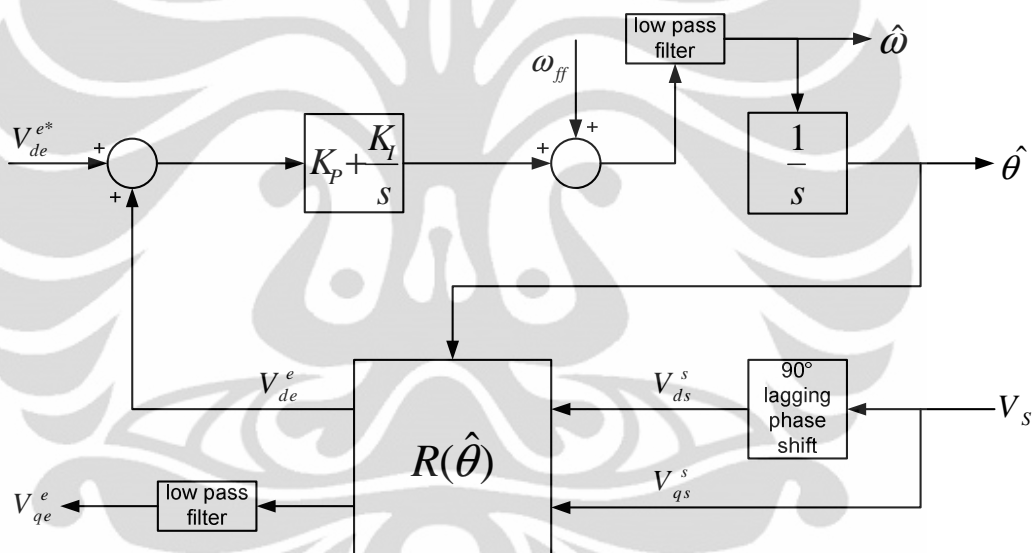
frekuensi estimasi masih belum sama dengan frekuensi sinyal tegangan PLN, hanya akan terbentuk suatu algoritma yang memperburuk keadaan frekuensi estimasi. Hal ini karena masa transien dari frekuensi estimasi tersebut tidak diperhitungkan, yang kemudian akan menyebabkan frekuensi estimasi selalu berubah-ubah karena masa transien dari frekuensi estimasi dianggap sebagai kesalahan pergeseran *state*, padahal saat pergeseran *state* sudah tepat pun, akan tetap ada masa transien yang dilewati oleh frekuensi estimasi.

Kemudian, kesulitan yang kedua adalah jika dibuat suatu algoritma ‘cerdas’ yang melakukan penyesuaian pergeseran *state* dengan memperhitungkan kondisi bahwa masa transien frekuensi estimasi telah dilalui. Algoritma ini sebenarnya menawarkan hasil akhir yang lebih baik daripada algoritma sederhana yang telah dijelaskan sebelumnya. Akan tetapi, dengan memperhitungkan masa transien ini, akan diperoleh suatu algoritma yang relatif kompleks (sulit direalisasikan, sebab berkaitan dengan waktu eksekusi program), serta memiliki waktu sinkronisasi yang relatif lambat juga (lambat karena pergeseran *state* baru dilakukan setelah masa transien dilalui, dan faktanya, jumlah pergeseran *state* ditentukan dengan cara mencoba-coba nilai-nilai tertentu satu per satu).

Akhirnya, solusi kedua yang dinilai dapat direalisasikan dengan waktu respon yang cepat dan memberikan output yang baik adalah dengan menempatkan algoritma filter pada algoritma PLL utama yang dimiliki. Algoritma filter tersebut ditempatkan untuk memproses variabel-variabel yang bersifat sensitif terhadap *noise* maupun kesalahan parameter dalam batasan tertentu, sehingga nantinya, output yang dihasilkan akan relatif lebih stabil (tidak terlalu berosilasi) daripada output yang diperoleh pada bagian sebelumnya (algoritma PLL awal). Salah satu kelemahan dari solusi ini adalah algoritma filter yang sebenarnya akan memperlambat kinerja sistem secara keseluruhan. Namun demikian, hal ini dapat diatasi dengan menentukan parameter pengendali yang baru (diperkecil daripada nilai yang digunakan sebelumnya), sampai akhirnya diperoleh suatu nilai parameter pengendali yang terbaik (cepat, namun tidak berosilasi) untuk algoritma PLL tegangan yang ditambahkan dengan filter ini.

4.6 Algoritma PLL Tegangan dengan *Low Pass Filter*

Besaran yang sangat sensitif dalam algoritma PLL tegangan biasa, namun sangat diperlukan sebagai output utama dari PLL tersebut adalah frekuensi estimasi dan tegangan proyeksi ke sumbu Q (salah satu hasil dari algoritma Transformasi DQ dalam algoritma utama PLL tegangan), sedangkan output berupa fasa sinyal tegangan listrik PLN terlihat tidak begitu sensitif terhadap perubahan maupun distorsi yang terjadi. Oleh sebab itulah, algoritma filter hanya akan diberikan pada kedua besaran awal yang bersangkutan. Berikut ini akan disajikan blok diagram dari PLL tegangan yang telah dimodifikasi dengan menambahkan *low pass filter* untuk menyaring osilasi pada besaran V_{qe}^e dan $\hat{\omega}$:



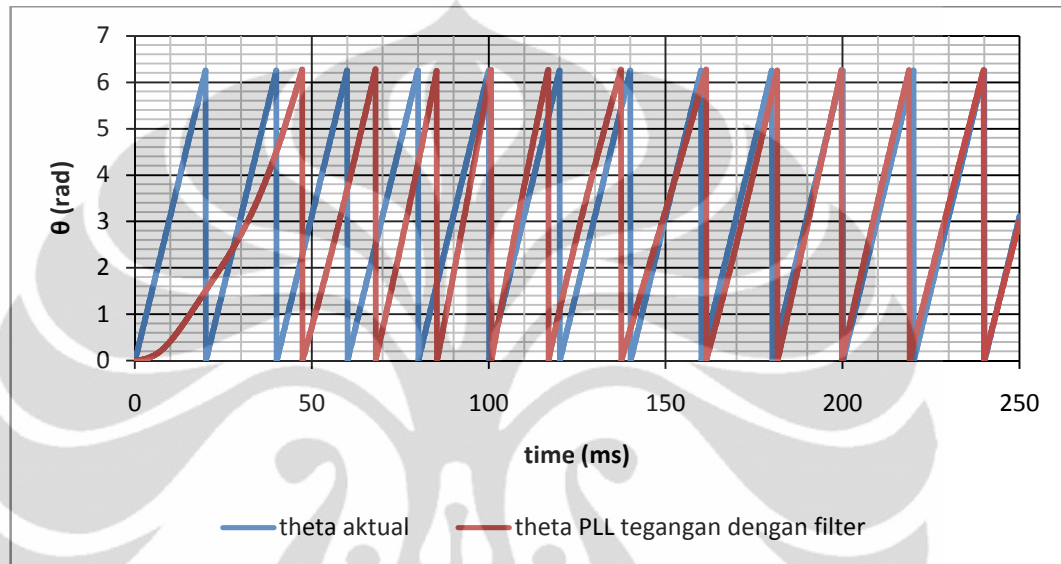
Gambar 4.12. Blok diagram PLL tegangan yang dilengkapi dengan low pass filter

Hal yang perlu diperhatikan dari algoritma PLL yang dimodifikasi di atas adalah bahwa algoritma filter yang ditambahkan memiliki frekuensi *cutoff* sebesar 25 hertz, yang berarti bahwa frekuensi sinyal yang lebih tinggi daripada 25 hertz akan diredam. Demikianlah algoritma filter tersebut bekerja, sehingga akhirnya sinyal distorsi yang berfrekuensi tinggi dapat dihilangkan dari sinyal respon output PLL tegangan tersebut, sehingga dapat diperoleh suatu respon yang baik (sesuai dengan persyaratan kerja).

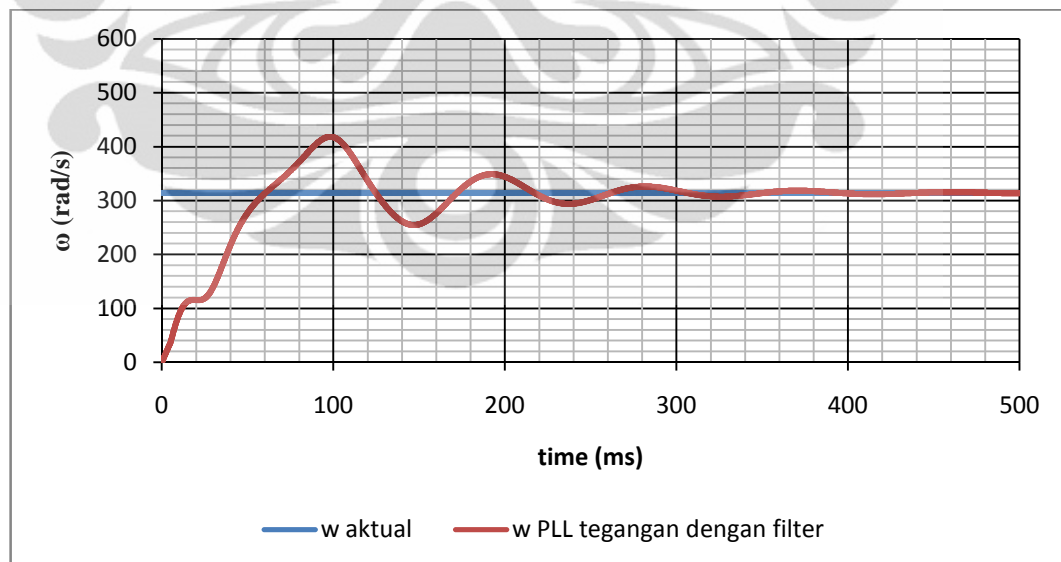
Kemudian, seperti yang telah dijelaskan sebelumnya, untuk mengimbangi perubahan yang terjadi karena adanya algoritma filter, nilai konstanta pengendali

yang digunakan perlu diganti lagi. Oleh karena gangguan yang terjadi berupa osilasi, maka nilai konstanta pengendali, secara logis, perlu diturunkan ke suatu nilai tertentu agar respon yang dihasilkan tidak berosilasi lagi, namun tetap dijaga kecepatan responnya.

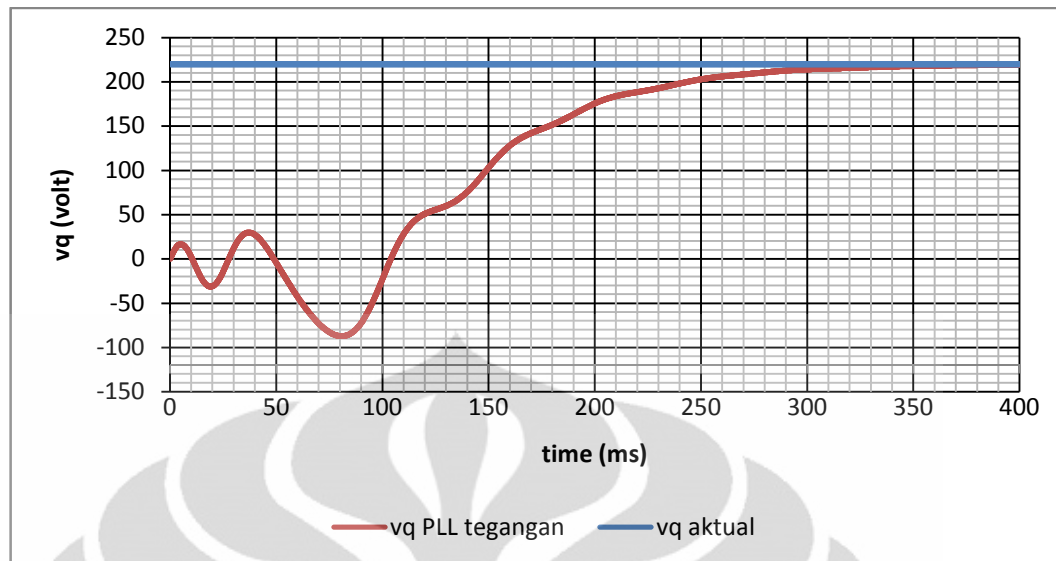
4.7 Performa Algoritma PLL Tegangan dengan Filter dalam Kondisi Ideal



Gambar 4.13. θ estimasi PLL tegangan dengan filter



Gambar 4.14. ω estimasi PLL tegangan dengan filter



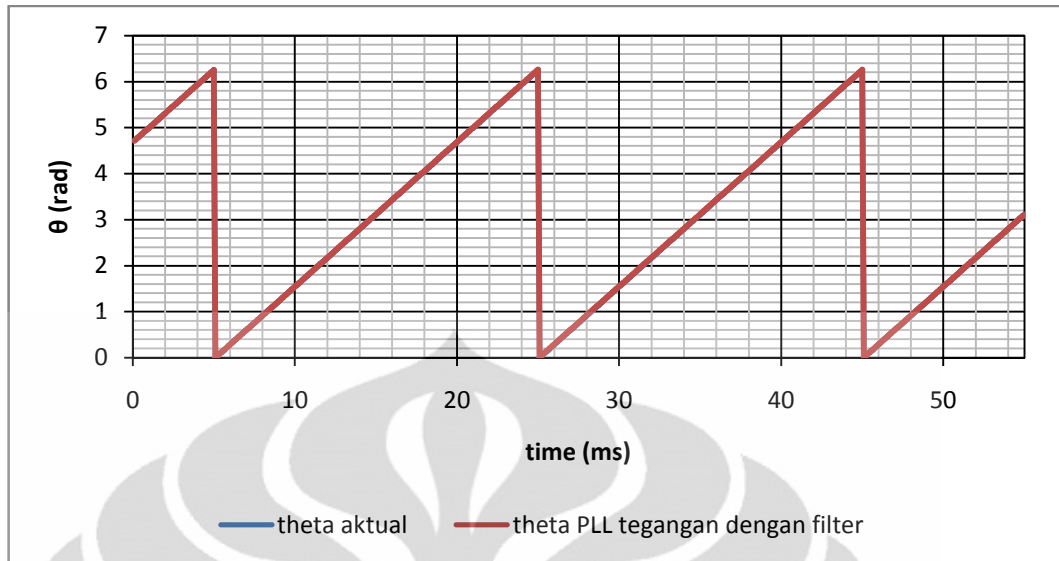
Gambar 4.15. vq estimasi PLL tegangan dengan filter

Berikut ini akan ditampilkan hasil simulasi dari PLL tegangan yang telah dimodifikasi dengan penambahan algoritma filter, pada kondisi ideal, yaitu saat amplitudo dan frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 220 volt dan 50 hertz.

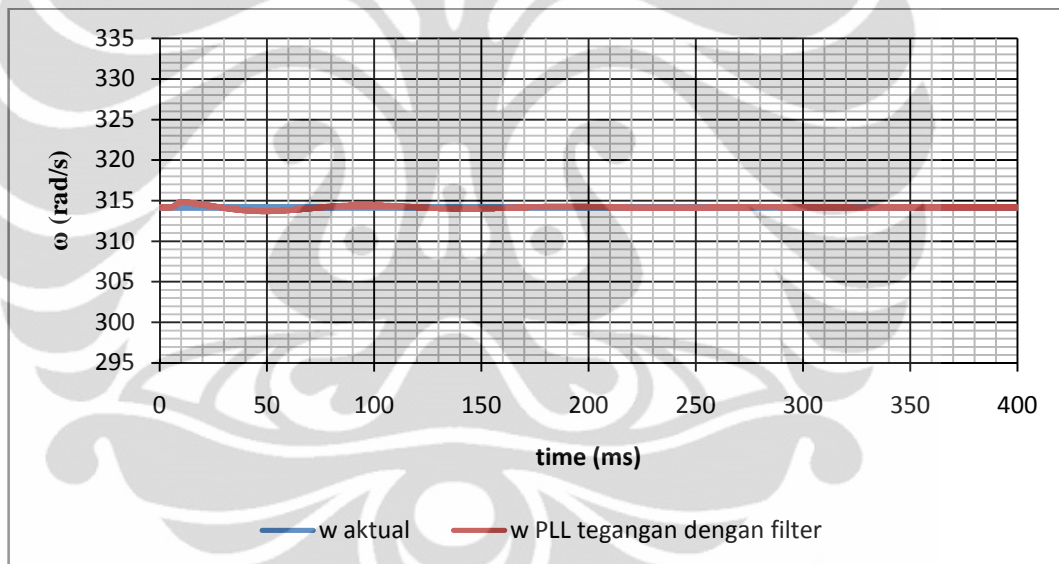
Terlihat dari Gambar 4.13, Gambar 4.14, serta Gambar 4.15, bahwa respon yang diperoleh dengan algoritma PLL tegangan yang menggunakan algoritma filter memang lebih lambat jika dibandingkan dengan algoritma PLL tegangan biasa (perbandingan dengan Gambar 4.1 dan Gambar 4.2).

4.8 Performa Algoritma PLL Tegangan dengan Filter dalam Kondisi Perubahan Amplitudo

Berikut ini akan ditampilkan hasil simulasi dari PLL tegangan yang telah ditambahkan dengan algoritma filter pada kondisi perubahan amplitudo, yaitu saat amplitudo sinyal tegangan listrik berubah dari 220 volt menjadi 200 volt pada waktu 5 milisekon, serta frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 50 hertz.

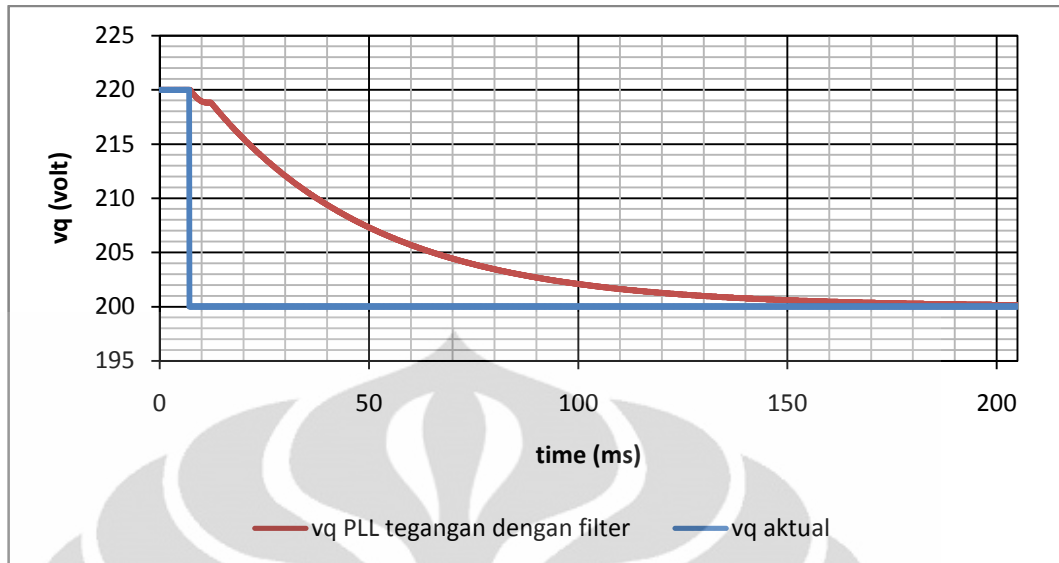


Gambar 4.16. θ estimasi PLL tegangan dengan filter saat amplitudo berubah



Gambar 4.17. ω estimasi PLL tegangan dengan filter saat amplitudo berubah

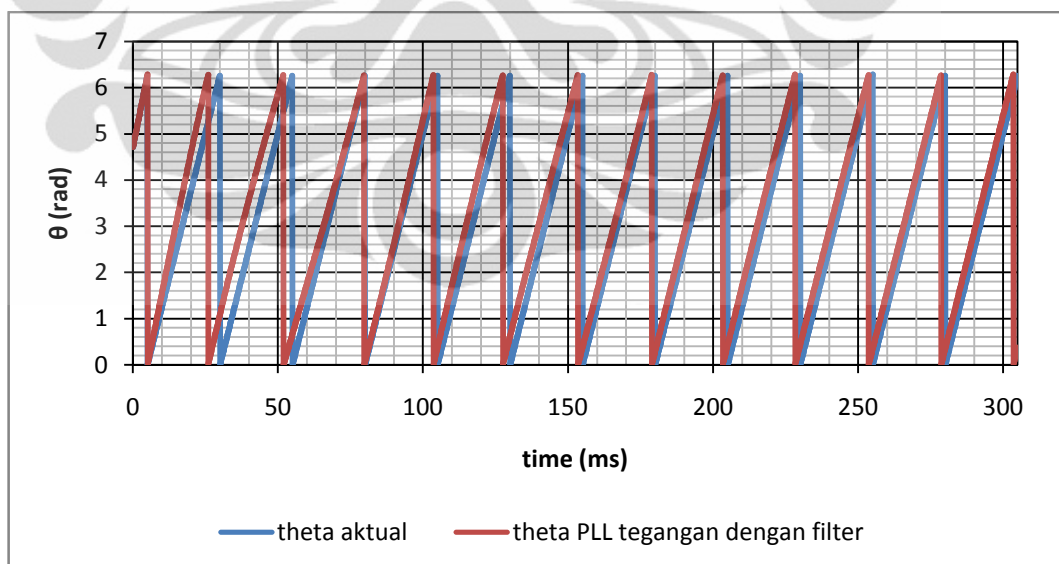
Pada Gambar 4.16, Gambar 4.17, dan Gambar 4.18 yang diperoleh kali ini pun, dapat dilihat bahwa respon yang ditunjukkan oleh PLL dalam memproses perubahan amplitudo sinyal tegangan listrik PLN adalah lebih lambat daripada algoritma PLL sebelumnya. Akan tetapi, kelebihan yang dapat diperhatikan dari



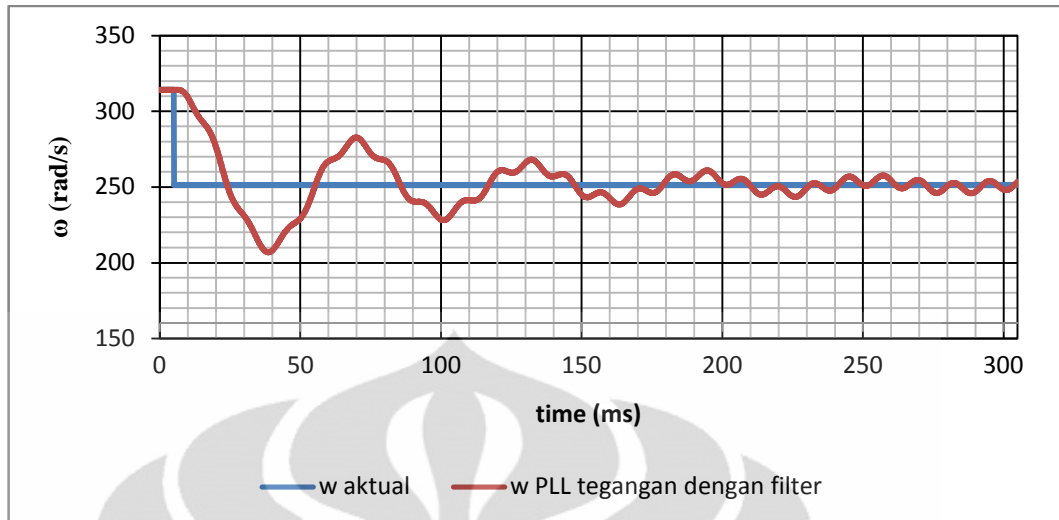
Gambar 4.18. v_q estimasi PLL tegangan dengan filter saat amplitudo berubah

PLL dengan filter ini adalah, bahwa algoritma ini memberikan respon yang lebih stabil, dengan lebih sedikit osilasi daripada algoritma PLL tanpa menggunakan filter.

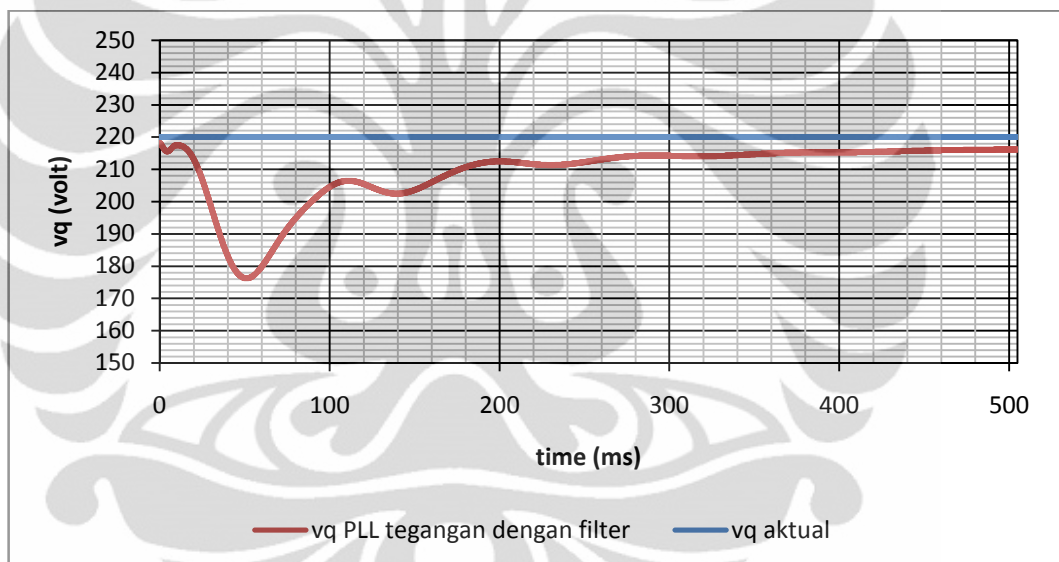
4.9 Performa Algoritma PLL Tegangan dengan Filter dalam Kondisi Perubahan Frekuensi



Gambar 4.19. θ estimasi PLL tegangan dengan filter saat frekuensi berubah



Gambar 4.20. ω estimasi PLL tegangan dengan filter saat frekuensi berubah



Gambar 4.21. v_q estimasi PLL tegangan dengan filter saat frekuensi berubah

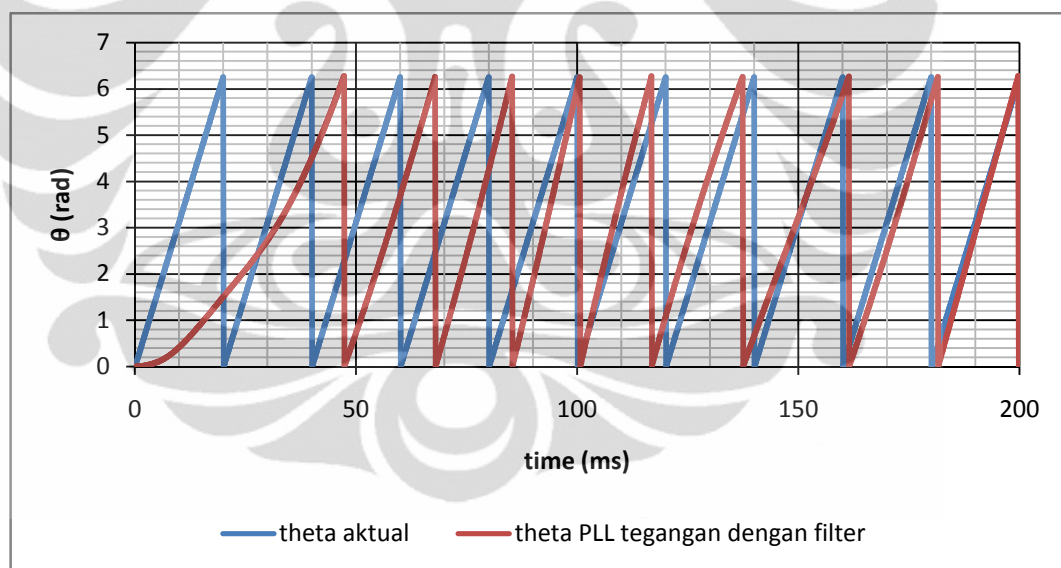
Berikut ini akan ditampilkan hasil simulasi dari PLL tegangan dengan algoritma filter pada kondisi perubahan frekuensi, yaitu saat amplitudo sinyal tegangan listrik bersifat konstan pada nilai 220 volt, serta frekuensi dari sinyal tegangan listrik PLN berubah dari 50 hertz menjadi 40 hertz pada waktu 5 milisekon.

Kali ini, Gambar 4.19, Gambar 4.20, serta Gambar 4.21 memperlihatkan salah satu manfaat nyata dari algoritma filter yang dimasukkan ke dalam algoritma utama PLL tegangan yang telah dibuat sebelumnya. Terlihat dari Gambar 4.19 dan Gambar 4.20 jika dibandingkan dengan Gambar 4.7 dan

Gambar 4.8, algoritma PLL dengan filter memberikan efek penstabil dan efek memperbaiki respon, sehingga output menjadi semakin dekat dengan respon yang diharapkan.

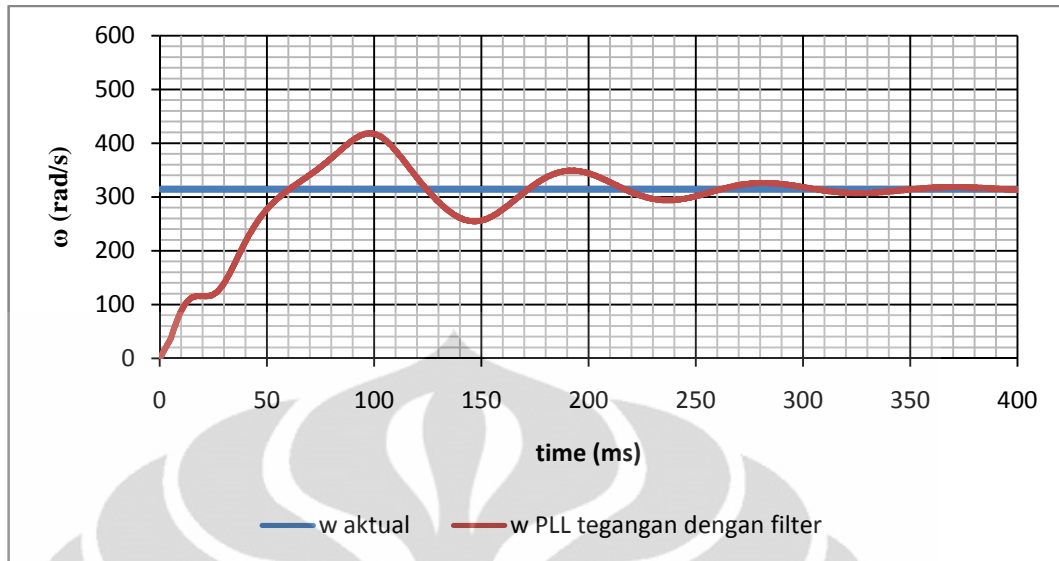
4.10 Performa Algoritma PLL Tegangan dengan Filter dalam Kondisi Amplitudo Ber-noise

Berikut ini akan ditampilkan hasil simulasi dari PLL tegangan dengan algoritma filter pada kondisi amplitudo memiliki *noise*, yaitu saat amplitudo sinyal tegangan listrik bersifat konstan pada nilai 220 volt, namun berinterferensi dengan harmonik ketujuhnya (frekuensinya $7 \times 50 \text{ hertz} = 350 \text{ hertz}$) yang memiliki amplitudo sepuluh persen dari amplitudo normalnya (amplitudonya $10\% \times 220 \text{ volt} = 22 \text{ volt}$), serta frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 50 hertz.

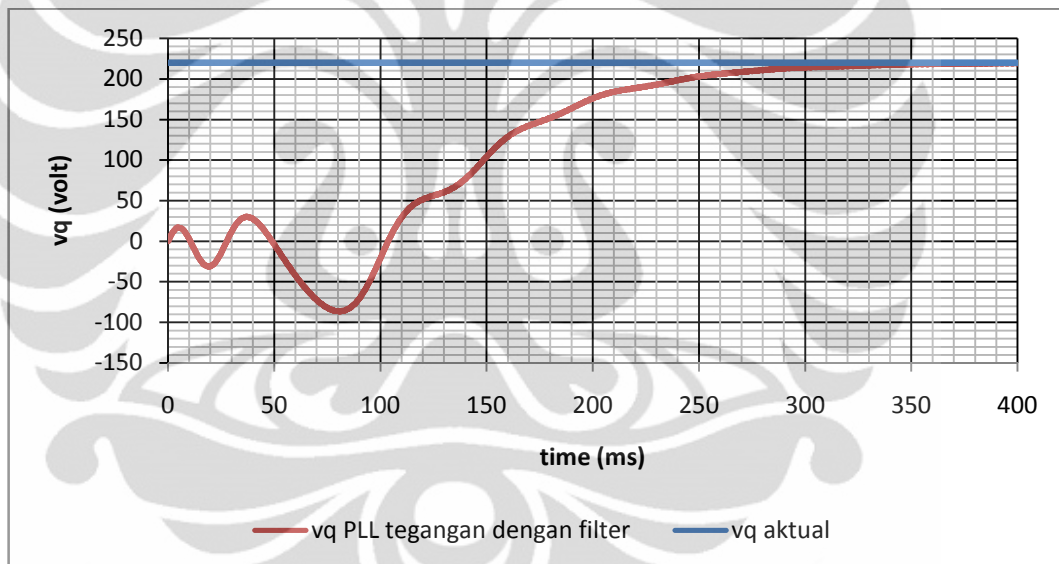


Gambar 4.22. θ estimasi PLL tegangan dengan filter saat amplitudo ber-noise

Pada Gambar 4.22, Gambar 4.23, dan Gambar 4.24 yang diperoleh kali ini pun, terlihat bahwa PLL dengan algoritma filter ternyata mampu memberikan respon yang lebih baik dan lebih stabil daripada PLL tegangan biasa yang telah



Gambar 4.23. ω estimasi PLL tegangan dengan filter saat amplitudo ber-noise

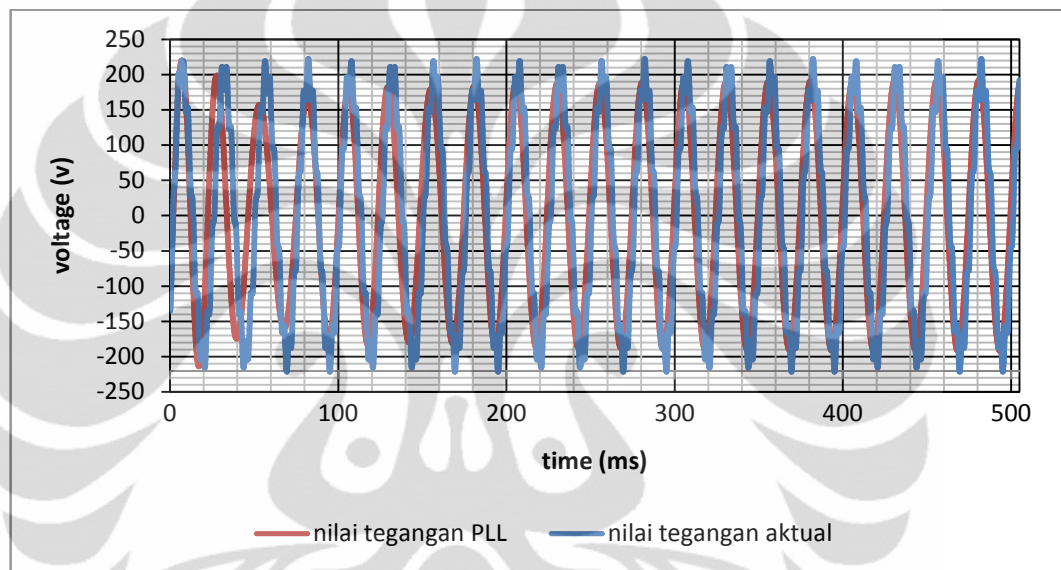


Gambar 4.24. v_q estimasi PLL tegangan dengan filter saat amplitudo ber-noise

dijelaskan sebelumnya. Dengan demikian, dapat disimpulkan secara umum bahwa algoritma PLL yang dilengkapi dengan algoritma tambahan berupa algoritma filter (yang digunakan untuk menghilangkan sinyal berfrekuensi tinggi yang dapat menyebabkan ketidakstabilan dan osilasi) pada besaran-besaran tertentu, dapat memberikan respon umum yang lebih baik dan lebih sesuai dengan apa yang diharapkan. Walaupun secara umum respon yang terjadi setelah pemberian algoritma filter adalah lebih lambat daripada algoritma PLL yang tidak menggunakan filter, akan tetapi hal ini dapat diatasi dengan pemilihan kembali

nilai-nilai parameter pengendali yang digunakan dalam algoritma PLL yang bersangkutan.

Selanjutnya, jika sinyal tegangan sinusoidal output dari data θ yang dihasilkan oleh algoritma PLL yang dilengkapi dengan algoritma filter ini ditampilkan, akan diperoleh hasil berikut ini, dengan catatan bahwa frekuensi dan amplitudo sinyal tegangan PLN berubah, serta nilai amplitudo sinyal tegangan PLN ber-*noise*.



Gambar 4.25. Performa PLL saat frekuensi dan tegangan berubah, serta ber-*noise*

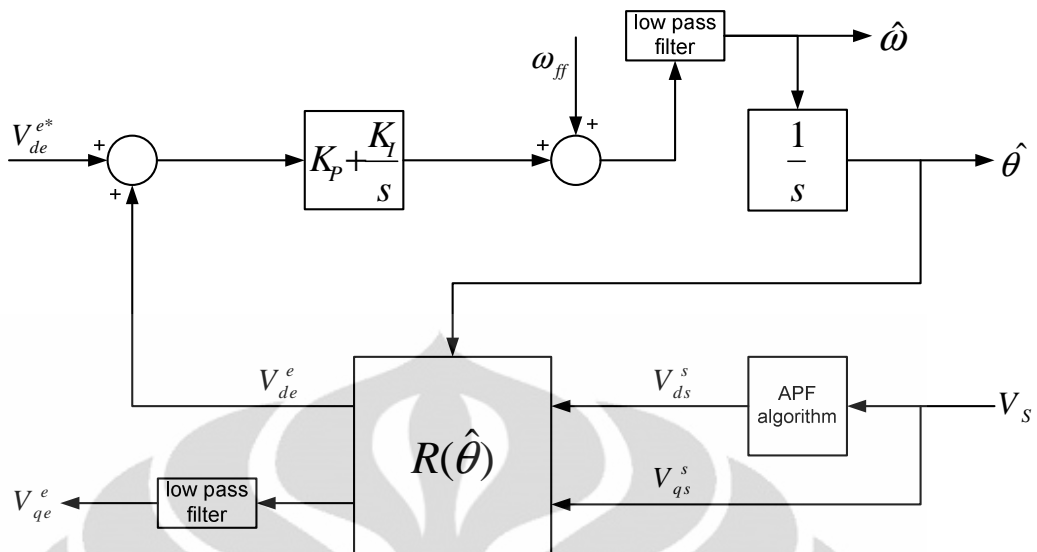
Gambar 4.25 menunjukkan bahwa PLL yang didesain dapat mengadaptasi perubahan (gangguan) yang terjadi pada sinyal tegangan PLN, yang mana gangguan yang dimaksudkan adalah perubahan amplitudo sinyal tegangan, perubahan frekuensi sinyal tegangan, serta amplitudo sinyal tegangan yang ber-*noise*. Lebih jauh lagi, dapat dilihat bahwa metode PLL yang dikembangkan terakhir ini (dengan ditambahkan algoritma filter) memberikan respon adaptasi yang relatif cepat, yaitu sekitar 0,2 sekon.

4.11 Pengembangan Algoritma PLL dengan Algoritma *All Pass Filter* (APF) untuk Penggeser Fasa Sebesar Sembilan Puluh Derajat, serta Hasil Simulasinya dalam Kondisi Ideal

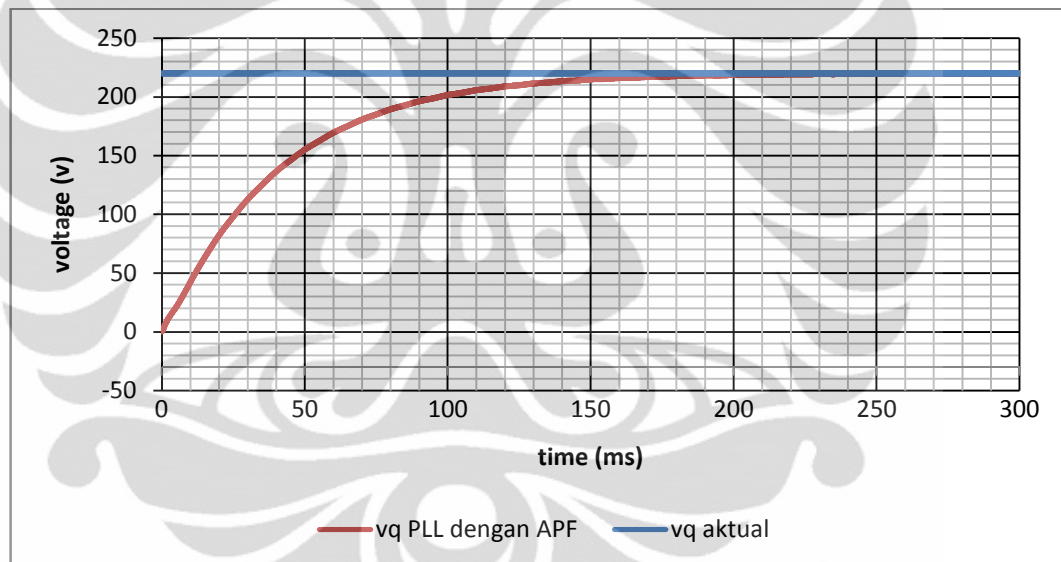
Salah satu permasalahan dari algoritma PLL yang dimiliki saat ini adalah bahwa algoritma tersebut tidak mampu melakukan pergeseran fasa sinyal sinusoidal input (berupa tegangan) secara tepat sebesar 90° untuk semua jangkauan frekuensi sinyal sinusoidal tegangan PLN yang kemungkinan memasuki PLL. Akibatnya, terjadi osilasi dalam output yang disajikan oleh PLL yang bersangkutan, yang berupa osilasi pada tegangan hasil estimasi, frekuensi hasil estimasi, dan sudut fasa hasil estimasi PLL yang bersangkutan. Walaupun osilasi yang terjadi telah diperkecil efeknya dengan menggunakan algoritma filter, seperti yang telah dijelaskan pada bagian sebelumnya, respon yang dihasilkan tetap agak berosilasi dan memiliki karakteristik transien yang agak tinggi.

Untuk semakin memperbaiki respon PLL yang dimiliki, akan diuji coba menggunakan algoritma APF untuk melakukan pergeseran fasa sinyal tegangan input sebesar 90° (Cha & Lee, 2008). Kemudian akan dibandingkan output yang diperoleh saat algoritma APF digunakan dengan saat algoritma pergeseran *state* dalam *s-function* digunakan, dengan harapan bahwa penggunaan algoritma APF akan semakin memperbaiki respon output dari PLL. Hal ini terlepas dari kenyataan bahwa algoritma APF ternyata menggunakan parameter berupa frekuensi kerja, yang sifatnya membatasi kondisi kerja dari algoritma APF yang bersangkutan (algoritma APF ternyata juga menggeser fasa secara tepat sebesar 90° saat frekuensi sinyal input yang hendak digeser sesuai dengan parameter frekuensi kerja APF tertentu, yang ditetapkan sebelumnya). Gambar 4.26 menampilkan blok diagram algoritma PLL yang telah dilengkapi dengan algoritma APF (yang sangat menyerupai Gambar 2.1).

Kemudian, berikut ini akan ditampilkan hasil simulasi dari PLL tegangan yang telah dimodifikasi dengan penambahan algoritma APF, pada kondisi ideal,



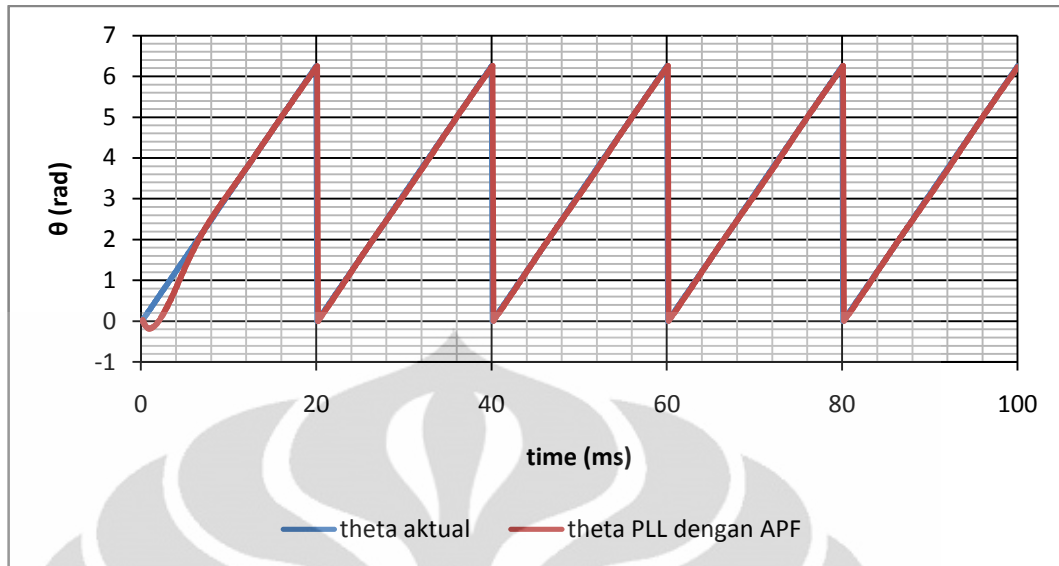
Gambar 4.26. Blok diagram PLL tegangan yang dilengkapi dengan low pass filter dan algoritma APF



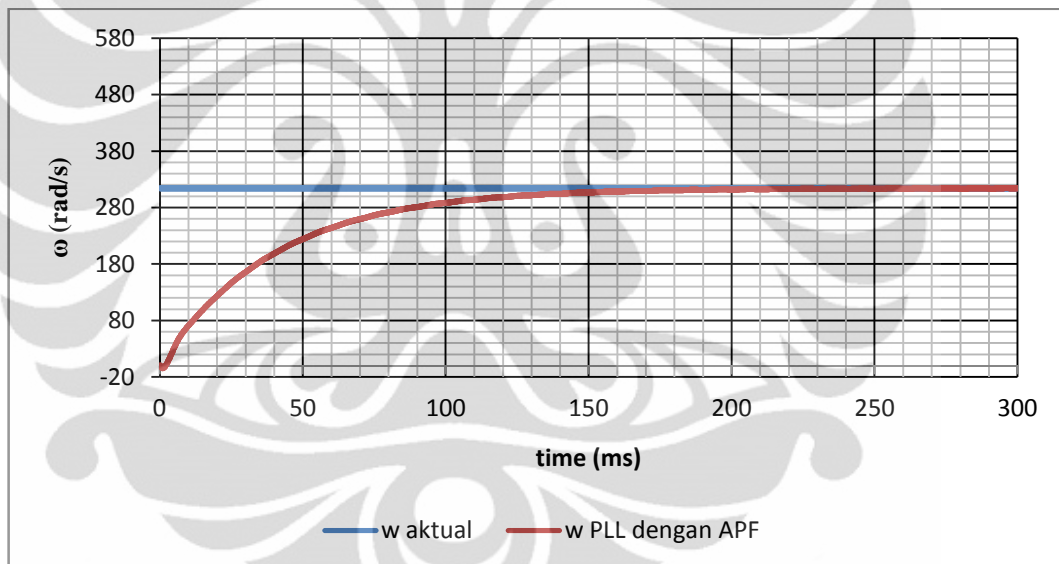
Gambar 4.27. vq estimasi PLL yang menggunakan algoritma APF

yaitu saat amplitudo dan frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 220 volt dan 50 hertz.

Dari Gambar 4.27, Gambar 4.28, dan Gambar 4.29, dapat dilihat bahwa algoritma PLL dengan filter yang dilengkapi dengan algoritma APF memberikan



Gambar 4.28. θ estimasi PLL yang menggunakan algoritma APF



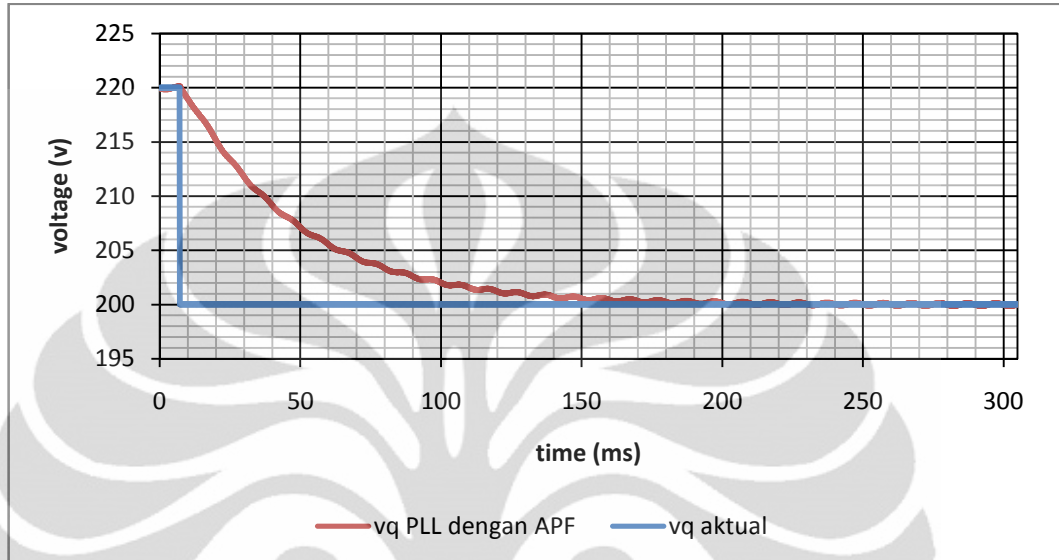
Gambar 4.29. ω estimasi PLL dengan menggunakan algoritma APF

respon yang relatif stabil serta memberikan respon yang relatif lebih cepat juga jika dibandingkan dengan algoritma PLL dengan filter biasa. Hal ini terlihat dari waktu transien yang bernilai sekitar 0,2 sekon, jika dibandingkan dengan waktu transien algoritma PLL dengan filter biasa yang bernilai sekitar 0,3 sekon.

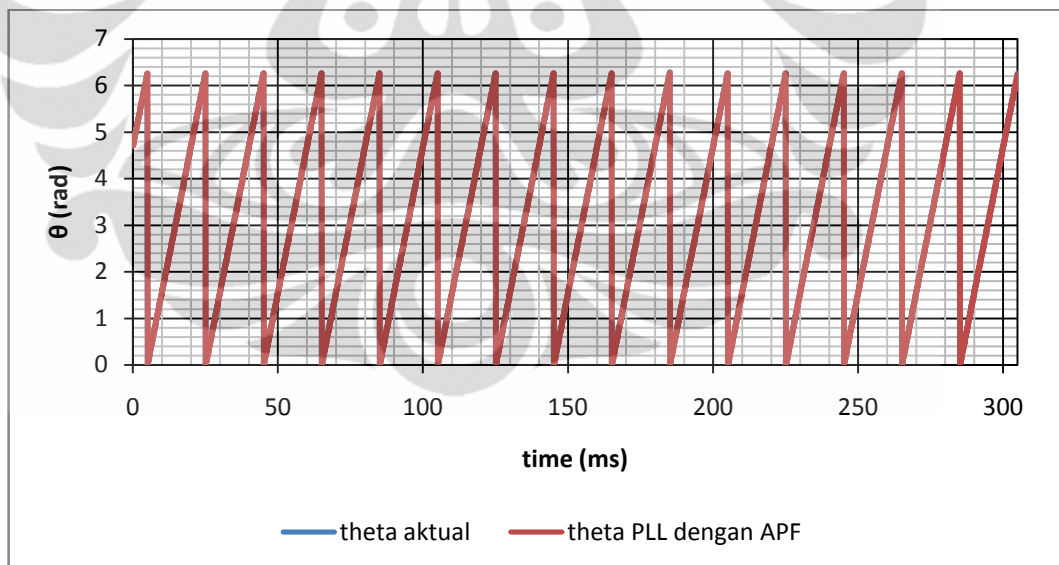
4.12 Performa Algoritma PLL dengan Algoritma APF dalam Kondisi Perubahan Amplitudo

Berikut ini akan ditampilkan hasil simulasi dari PLL tegangan yang telah ditambahkan dengan algoritma filter pada kondisi perubahan amplitudo, yaitu saat

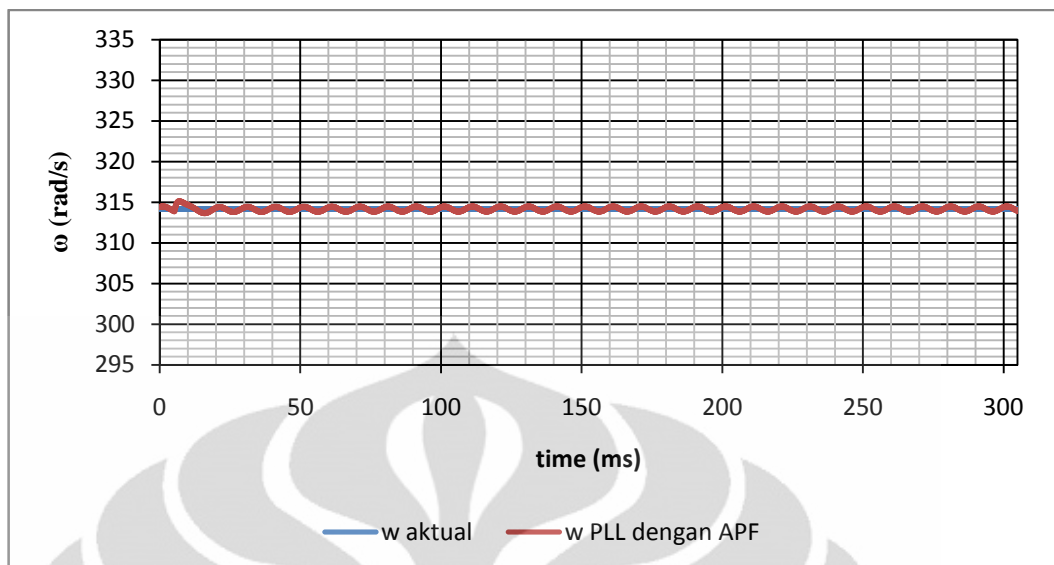
amplitudo sinyal tegangan listrik berubah dari 220 volt menjadi 200 volt pada waktu 5 milisekon, serta frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 50 hertz.



Gambar 4.30. v_q estimasi PLL dengan APF saat amplitudo berubah



Gambar 4.31. θ estimasi PLL dengan APF saat amplitudo berubah

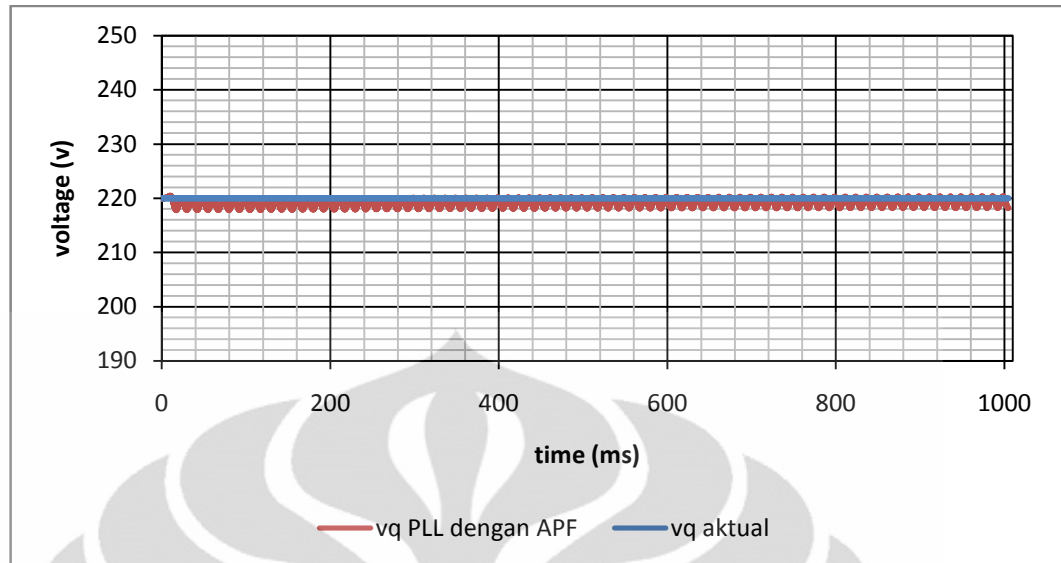
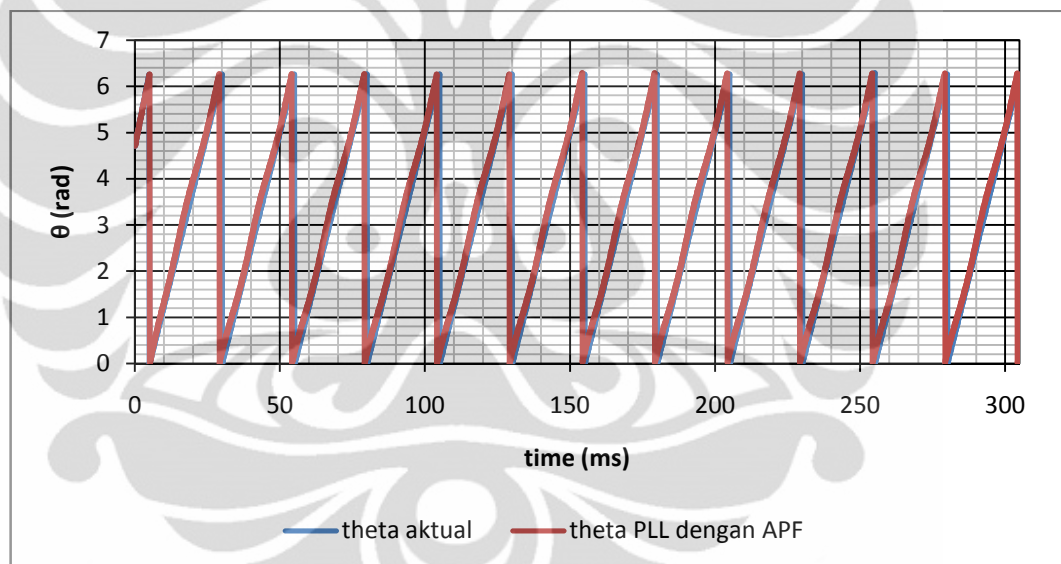


Gambar 4.32. ω estimasi PLL dengan APF saat amplitudo berubah

Dari Gambar 4.30, Gambar 4.31, dan Gambar 4.32, dapat dilihat bahwa secara keseluruhan, performa yang diberikan oleh PLL dengan APF pada saat kondisi amplitudo tegangan berubah adalah mirip dengan performa dari PLL tanpa menggunakan APF. Dengan waktu transien yang berdekatan antara algoritma PLL biasa dan algoritma PLL dengan APF, yaitu sekitar 0,2 sekon, serta perbandingan grafik yang menunjukkan bentuk respon yang memiliki kemiripan, dapat semakin memperkuat pernyataan bahwa PLL dengan APF pun memberikan respon yang baik dalam menghadapi perubahan kondisi kerja.

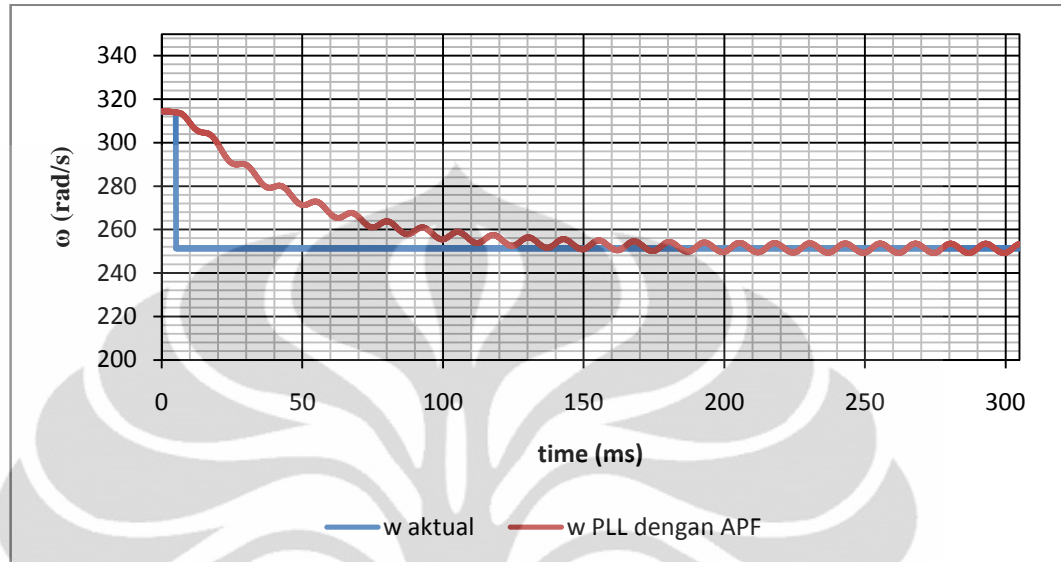
4.13 Performa Algoritma PLL dengan Algoritma APF dalam Kondisi Perubahan Frekuensi

Berikut ini akan ditampilkan hasil simulasi dari PLL tegangan dengan algoritma filter pada kondisi perubahan frekuensi, yaitu saat amplitudo sinyal tegangan listrik bersifat konstan pada nilai 220 volt, serta frekuensi dari sinyal tegangan listrik PLN berubah dari 50 hertz menjadi 40 hertz pada waktu 5 milisekon.

Gambar 4.33. v_q estimasi PLL dengan APF saat frekuensi berubahGambar 4.34. θ estimasi PLL dengan APF saat frekuensi berubah

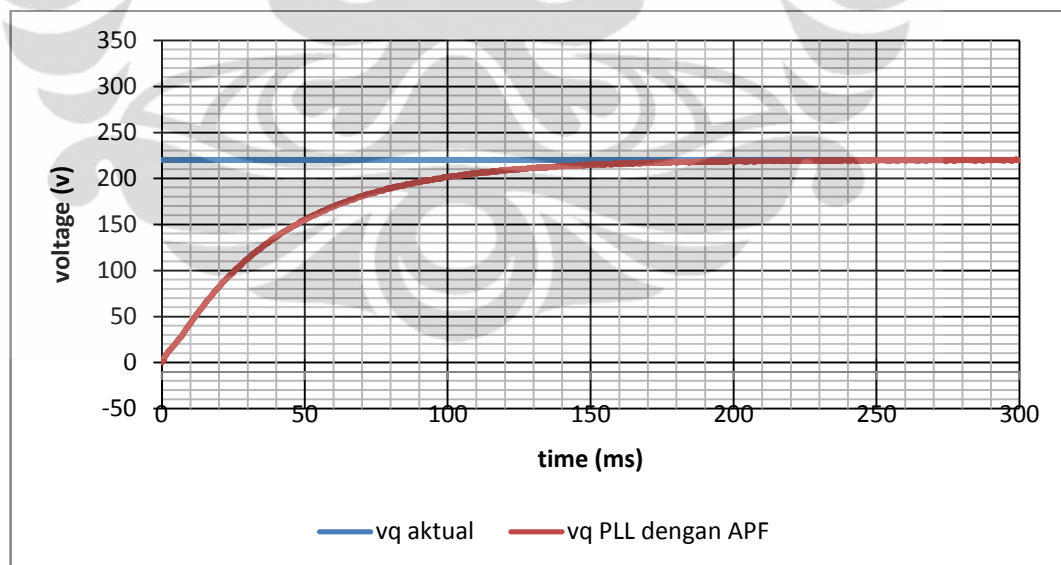
Dari Gambar 4.33, Gambar 4.34, dan Gambar 4.35, dapat dilihat bahwa walaupun respon output algoritma PLL dengan APF nampak berosilasi, akan tetapi osilasi tersebut berada dalam jangkauan yang tidak terlalu besar. Lebih jauh lagi, kelebihan dari algoritma PLL dengan APF yang berupa respon yang lebih

cepat daripada algoritma PLL biasa, merupakan kelebihan yang dapat diterima dalam kondisi osilasi yang tidak terlalu besar ini.

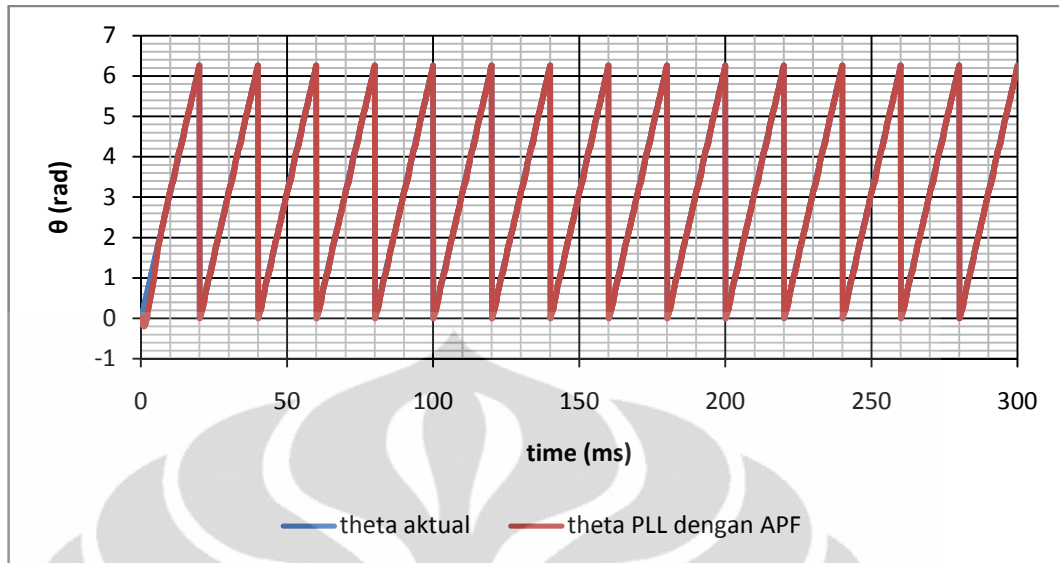
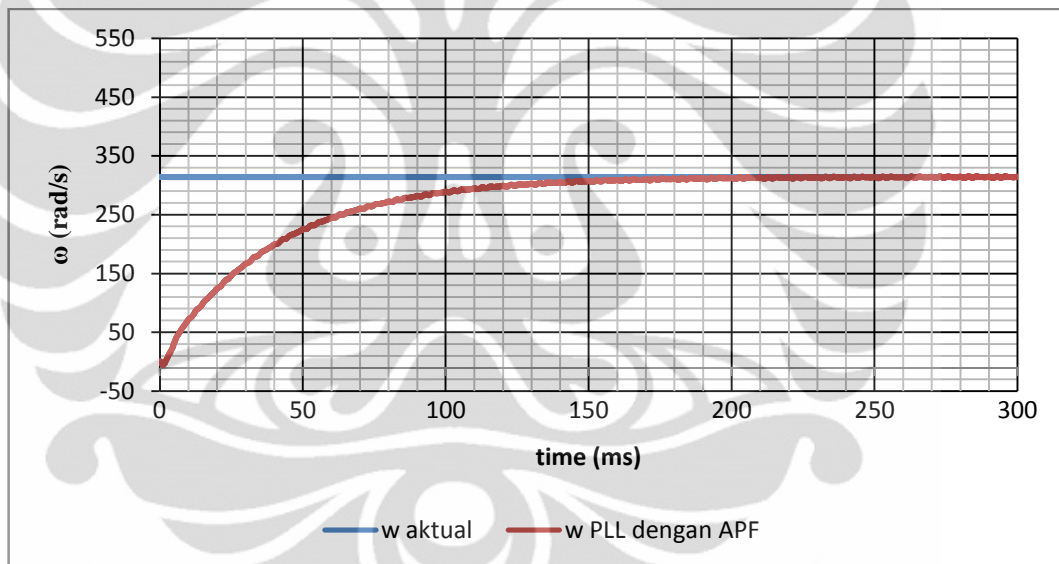


Gambar 4.35. ω estimasi PLL dengan APF saat frekuensi berubah

4.14 Performa Algoritma PLL dengan Algoritma APF dalam Kondisi Perubahan Amplitudo Ber-noise



Gambar 4.36. vq estimasi PLL dengan APF saat amplitudo ber-noise

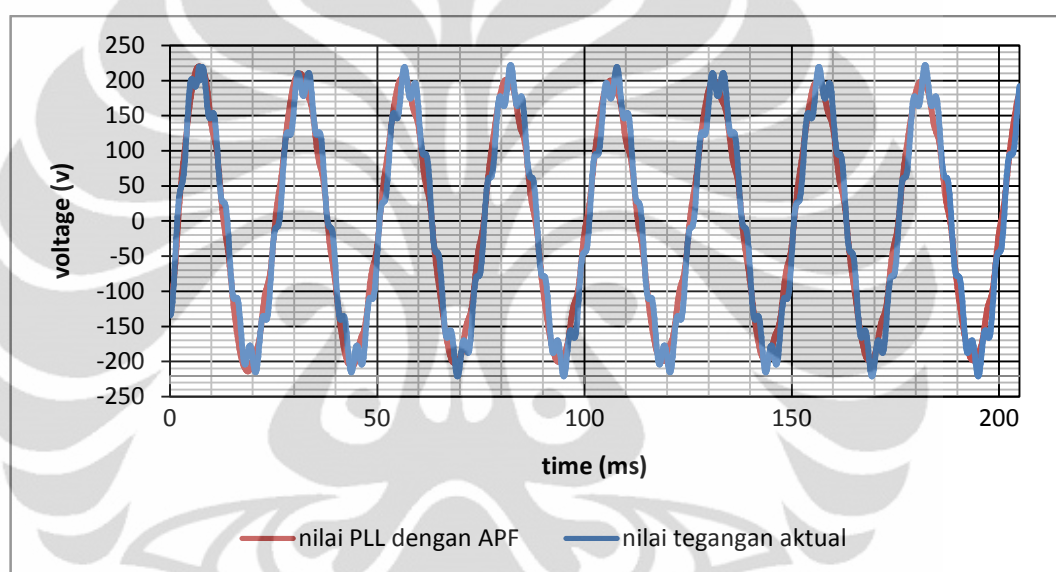
Gambar 4.37. θ estimasi PLL dengan APF saat amplitudo ber-noiseGambar 4.38. ω estimasi PLL dengan APF saat amplitudo ber-noise

Berikut ini akan ditampilkan hasil simulasi dari PLL tegangan dengan algoritma filter pada kondisi amplitudo memiliki *noise*, yaitu saat amplitudo sinyal tegangan listrik bersifat konstan pada nilai 220 volt, namun berinterferensi dengan harmonik ketujuhnya (frekuensinya $7 \times 50 \text{ hertz} = 350 \text{ hertz}$) yang memiliki amplitudo sepuluh persen dari amplitudo normalnya (amplitudonya $10\% \times 220 \text{ volt} = 22 \text{ volt}$), serta frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 50 hertz.

Melalui Gambar 4.36, Gambar 4.37, dan Gambar 4.38, dapat disimpulkan bahwa waktu transien rata-rata yang dimiliki oleh algoritma PLL dengan APF

adalah sekitar 0,2 sekon. Selain itu, walaupun respon algoritma PLL dengan APF menunjukkan osilasi, akan tetapi osilasi ini relatif kecil dan dapat diterima jika dibandingkan dengan kelebihan algoritma PLL dengan APF ini yang berupa peningkatan / semakin cepatnya waktu transien yang teramati.

Selanjutnya, jika sinyal tegangan sinusoidal output dari data θ yang dihasilkan oleh algoritma PLL dengan APF ini dikeluarkan, akan diperoleh hasil berikut ini, dengan catatan bahwa frekuensi dan amplitudo sinyal tegangan PLN berubah, serta nilai amplitudo sinyal tegangan PLN ber-noise.



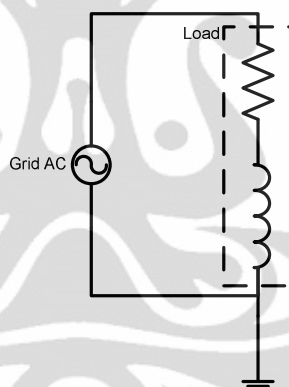
Gambar 4.39. Performa PLL dengan APF saat frekuensi dan tegangan berubah, serta ber-noise

Dari Gambar 4.39, dapat dilihat bahwa respon output yang dihasilkan oleh algoritma PLL dengan APF ini tergolong cukup baik dan dapat mengikuti / menghadapi perubahan yang terjadi dalam sinyal input ke dalam PLL yang bersangkutan. Dari gambar tersebut juga, dapat dilihat kecepatan respon algoritma PLL dengan APF yang merupakan kelebihan dari algoritma tersebut.

BAB 5 APLIKASI PLL DALAM GRID CONNECTED PV

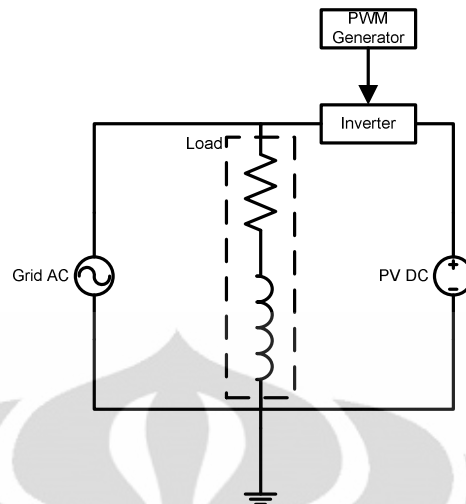
5.1 Dasar Aplikasi PLL dalam *Grid Connected PV*

Telah dijelaskan sebelumnya mengenai performa PLL digital yang telah dilengkapi dengan *low pass filter* dan APF. Terbukti bahwa PLL tersebut memberikan performa yang stabil dan waktu transien yang relatif baik (sekitar 150 milisekon) dalam hal mengestimasi nilai tegangan *grid* efektif, nilai frekuensi tegangan *grid* efektif, serta nilai sudut fasa tegangan *grid* efektif. Berikutnya, nilai-nilai ini akan digunakan untuk melakukan sinkronisasi antara output suatu PV dengan kondisi *grid* yang akan diparalelkan dengan PV tersebut. Gambar 5.1 menunjukkan suatu skema sederhana yang menunjukkan *grid* yang mensuplai beban tertentu.



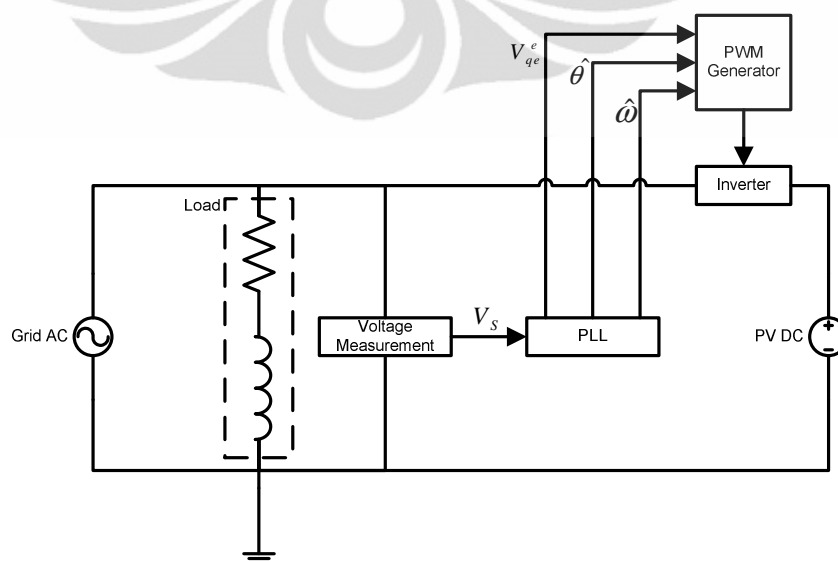
Gambar 5.1. Skema grid dan beban sederhana

Untuk memparalelkan beban yang disuplai oleh *grid* dengan output PV yang menghasilkan sinyal tegangan DC, diperlukan suatu perangkat *inverter* yang berfungsi untuk mengubah sinyal tegangan DC menjadi AC. Biasanya output dari *inverter* ini dikendalikan oleh sinyal PWM (*Pulse Width Modulation*) untuk menghasilkan suatu nilai tegangan AC dengan suatu karakteristik tertentu.



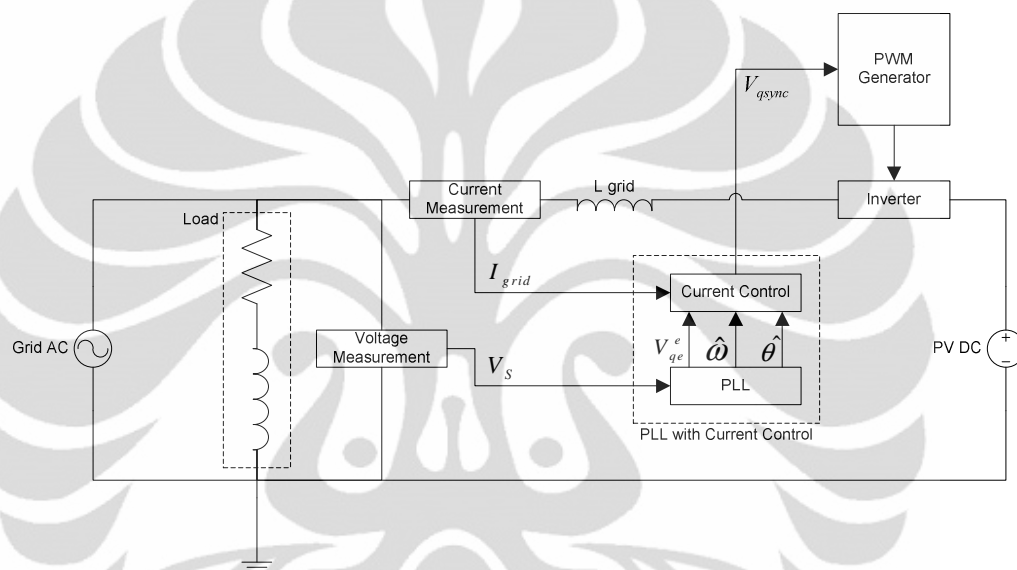
Gambar 5.2. Skema operasi paralel sederhana antara grid dengan PV

Akan tetapi, dengan skema paralel sederhana yang ditunjukkan oleh Gambar 5.2 saja, syarat sinkronisasi tidak dapat secara otomatis diperoleh, sehingga diperlukan suatu perangkat PLL yang telah dijelaskan pada bab sebelumnya, yang berfungsi untuk mengidentifikasi kondisi tegangan pada *grid* dan akhirnya merekonstruksi suatu sinyal tegangan dari perangkat *inverter* yang tersinkronisasikan dengan kondisi tegangan pada *grid* (Gambar 5.3). Hal ini akan menjamin bahwa suplai yang diberikan oleh PV lewat *inverter* dapat secara kontinu dipertahankan untuk mensuplai secara bersamaan dengan suplai yang berasal dari *grid*.



Gambar 5.3. Skema suplai load oleh grid dan PV secara bersamaan dengan kondisi tegangan yang tersinkronisasi

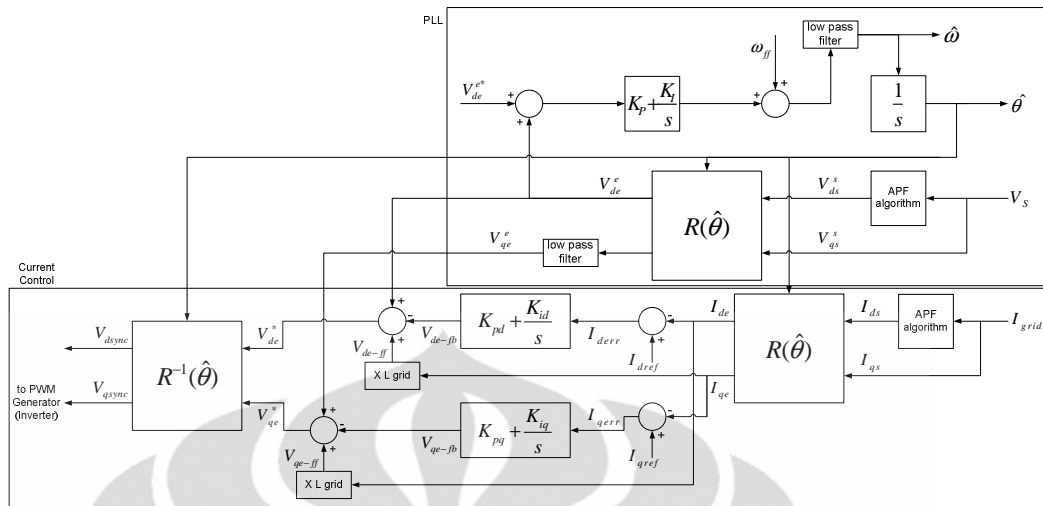
Kemudian, untuk memaksimalkan performa dari suplai *load* secara paralel antara *grid* dengan PV ini, akan dilakukan sinkronisasi antara tegangan listrik yang terukur pada *load* dengan arus listrik yang disuplai ke *load* tersebut (Gambar 5.4). Hal ini dilakukan untuk menjamin penyaluran daya secara efektif dan efisien dengan nilai faktor daya mendekati satu (yang ditandai oleh sinyal tegangan dan sinyal arus yang sefasa).



Gambar 5.4. Skema suplai load oleh grid dan PV secara bersamaan dengan kondisi tegangan dan arus yang tersinkronisasi

5.2 Skema PLL dengan *Current Control*

Berikut ini akan disajikan blok diagram yang menunjukkan skema PLL yang telah dijelaskan sebelumnya (PLL tegangan dengan *low pass filter* dan APF), yang dilengkapi dengan blok *current control* untuk aplikasi penyaluran daya secara optimal, yaitu kondisi penyaluran daya dengan nilai faktor daya sebesar satu, atau dengan kata lain sinyal tegangan sefasa dengan sinyal arus sehingga semua daya muncul sebagai daya aktif (Cha & Lee, 2008).



Gambar 5.5. Skema PLL tegangan dengan low pass filter dan APF dengan current control

Gambar 5.5 dapat dilihat dengan lebih jelas pada bagian lampiran. Ide dari PLL dengan *current control* ini sebenarnya adalah memproses output dari PLL yang berupa hasil estimasi amplitudo tegangan *grid*, frekuensi tegangan *grid*, dan sudut fasa tegangan *grid*. Pertama-tama, blok *current control* akan mengambil input berupa sinyal arus *grid* / aktual ($I_{grid} = I_{qs}$), kemudian dengan algoritma APF, sinyal arus ini akan digeser fasanya agar tertinggal 90° (I_{ds}) terhadap sinyal arus aktual. Kedua sinyal arus ini akan ditransformasi dengan operator Transformasi DQ dengan parameter sudut fasa tegangan *grid* hasil estimasi PLL ($R(\hat{\theta})$). Tujuannya mirip seperti tujuan Transformasi DQ untuk tegangan *grid*, yaitu agar Transformasi DQ untuk arus *grid* ini, dalam kondisi sinkron dengan tegangan *grid* (karena penggunaan parameter sudut fasa tegangan *grid* hasil estimasi), menghasilkan nilai amplitudo sinyal arus *grid* (I_{qe}) dan nilai nol (I_{de}). Nilai I_{qe} dan I_{de} kemudian akan dibandingkan dengan nilai referensi untuk kedua besaran tersebut. Nilai I_{qe} akan dibandingkan dengan nilai arus yang hendak disuplai ke dalam *grid* (I_{qref}), sedangkan Nilai I_{de} akan dibandingkan dengan I_{dref} yang ditentukan bernilai nol untuk menjamin kondisi sinkron (pembuktian secara matematisnya mirip dengan perhitungan syarat kondisi sinkron untuk PLL tegangan yang telah dijabarkan pada bab 2).

Nilai *error* yang terjadi, yaitu I_{derr} dan I_{qerr} akan diproses oleh pengendali PI untuk setiap sinyal *error* (dengan parameter K_{pd} dan K_{id} untuk

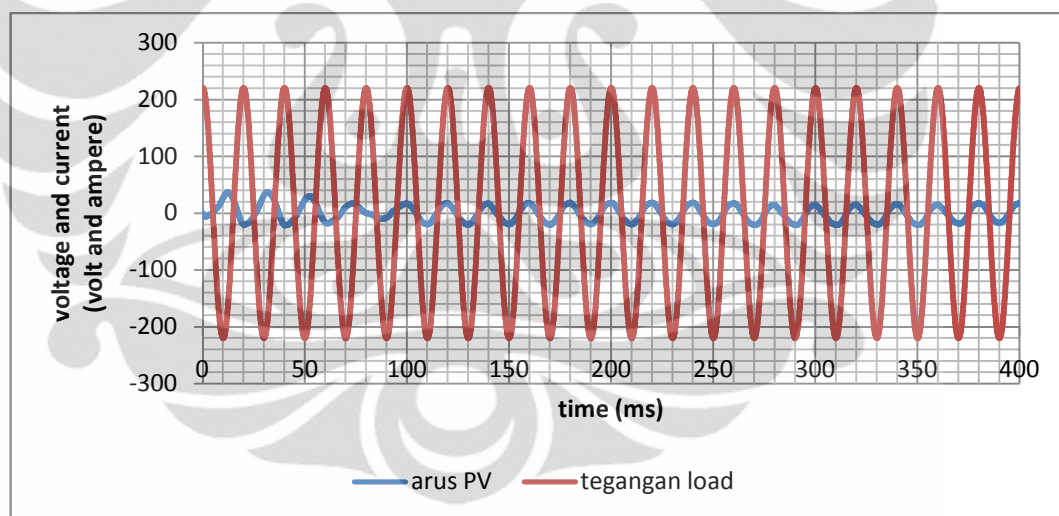
memproses I_{derr} , serta K_{pq} dan K_{iq} untuk memproses I_{qerr}) dan menghasilkan sinyal keluaran pengendali PI berupa V_{de-fb} dan V_{qe-fb} . Berikutnya, sinyal ini dijumlahkan dengan V_{de-ff} dan V_{qe-ff} yang berasal dari hasil *coupling* (perkalian menyilang) antara I_{de} dan I_{qe} dengan nilai induktansi dari L_{grid} . Kemudian, hasilnya dijumlahkan lagi dengan V_{de}^e dan V_{qe}^e yang dihasilkan oleh PLL tegangan yang disimulasikan pada bab 4. Sinyal hasil penjumlahan ketiga sinyal sebelumnya tersebut, yang disimbolkan sebagai V_{de}^* dan V_{qe}^* , kemudian akan ditransformasi balik dengan Invers Transformasi DQ yang masih menggunakan parameter $\hat{\theta}$ (sudut fasa hasil estimasi dari sinyal tegangan *grid*), untuk menghasilkan V_{dsync} dan V_{qsync} . Nilai V_{qsync} merepresentasikan sinyal tegangan yang perlu disuplai ke *load* untuk menghasilkan sinyal arus yang sinkron dengan sinyal tegangan yang terdapat / terukur pada *load*. Sehingga akhirnya, kondisi penyaluran arus dan tegangan pada *load* dapat mencapai nilai *power factor* (PF) sekitar satu. Atau dengan kata lain, sinyal tegangan yang terukur pada *load* dan sinyal arus yang terukur pada *load* adalah sefasa, sehingga penyaluran daya ke *load* ini dapat mencapai performa terbaik (yang berarti bahwa daya yang muncul seluruhnya hanyalah daya aktif saja).

5.3 Performa PLL dengan *Current Control* dalam Kondisi Ideal

Simulasi performa PLL yang dilengkapi dengan *current control*, mirip seperti simulasi yang dilakukan pada simulasi performa PLL yang dilakukan pada bab 4, akan dilakukan pada empat keadaan, yaitu performa PLL dengan *current control* dalam kondisi ideal, dalam kondisi perubahan amplitudo, dalam kondisi perubahan frekuensi, dan dalam kondisi amplitudo ber-*noise*. Selain itu, hal lainnya yang perlu diperhatikan adalah bahwa simulasi ini dilakukan dengan menggunakan waktu pencuplikan (*sampling time*) sebesar 1 milisekon. Dengan demikian, perangkat mikrokontroler dapat digunakan untuk keperluan realisasi baris-baris program, karena *sampling time* yang digunakan tergolong tidak terlalu cepat untuk sebuah mikrokontroler.

Simulasi dilakukan dengan perintah untuk mensuplai arus sebesar 15 ampere ke dalam *grid*, atau dengan kata lain, nilai I_{qref} ditentukan sebesar 15 ampere. Asumsi ini dilakukan berdasarkan kenyataan bahwa *grid* yang di

Indonesia diwakili oleh PLN memiliki kapasitas suplai umum sebesar 10 ampere untuk mensuplai beban rumah tangga. Selain itu, hal lainnya yang perlu diperhatikan adalah bahwa *inverter* yang digunakan akan direalisasikan dengan menggunakan struktur rangkaian dengan komponen IGBT (*Insulated-Gate Bipolar Transistor*) yang disuplai oleh input DC sebesar 250 volt untuk diubah menjadi output AC. Kemudian, untuk mengendalikan output dari *inverter* tersebut, digunakan PWM (*Pulse Width Modulation*) yang dibentuk dengan metode / algoritma SPWM (*Sinusoidal PWM*) dengan *carrier* berupa sinyal segitiga berfrekuensi sebesar 10000 hertz. Berikut ini akan ditampilkan hasil simulasi (yang berupa grafik perbandingan antara tegangan pada *load* dan arus AC yang disuplai oleh PV setelah melalui *inverter*) PLL dengan *current control* dalam kondisi ideal, yaitu saat amplitudo dan frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 220 volt dan 50 hertz.



Gambar 5.6. Performa PLL dengan current control pada kondisi ideal

Gambar 5.6 menunjukkan bahwa PLL yang telah dilengkapi dengan blok *current control* mampu melakukan proses sinkronisasi antara arus yang disuplai oleh PV lewat *inverter* dengan tegangan pada *load*. Terlihat bahwa PLL dengan *current control* ini pada awalnya melakukan pencarian dan pencocokan sudut fasa antara arus PV dengan tegangan *load*, sampai akhirnya mampu mensinkronisasikan keduanya secara terus menerus.

Jika diperhatikan secara lebih lanjut, terlihat bahwa waktu transien yang diperlukan oleh PLL dengan *current control* ini adalah sekitar 100 milisekon, terhitung sejak awal simulasi dijalankan sampai akhirnya terlihat bahwa sudut fasa antara arus PV dan sudut fasa tegangan *load* menjadi sama (kondisi sinkron). Hal ini dapat dijelaskan antara lain karena waktu transien dari PLL tegangan dengan *low pass filter* dan APF adalah sekitar 150 hingga 200 milisekon, yang akhirnya mempengaruhi kinerja blok *current control* yang bersangkutan, oleh karena PLL dan *current control* bekerja secara bersama-sama (saling mengumpan balik nilai-nilai parameter).

Kemudian, penjelasan yang memungkinkan untuk waktu transien PLL dengan *current control* yang terbukti lebih cepat daripada waktu transien untuk PLL tanpa *current control* adalah karena grafik respon PLL dengan *current control* tidak terlihat begitu mendetil seperti grafik respon PLL tanpa *current control* yang diperlihatkan pada bab 4, sehingga waktu transien yang diperoleh hanyalah waktu transien yang diperoleh dari pengamatan secara langsung saja (perkiraan secara logis). Hal ini menyebabkan muncul dugaan bahwa sebenarnya waktu transien dari PLL dengan maupun tanpa *current control* sebenarnya adalah hampir sama. Alasan ini didukung oleh fakta bahwa waktu transien yang teramati pada PLL dengan *current control* adalah sebesar 100 milisekon, sedangkan waktu transien yang diamati pada PLL tanpa *current control* adalah sebesar 150 hingga 200 milisekon, yang secara grafis, terlihat tidak berbeda secara signifikan.

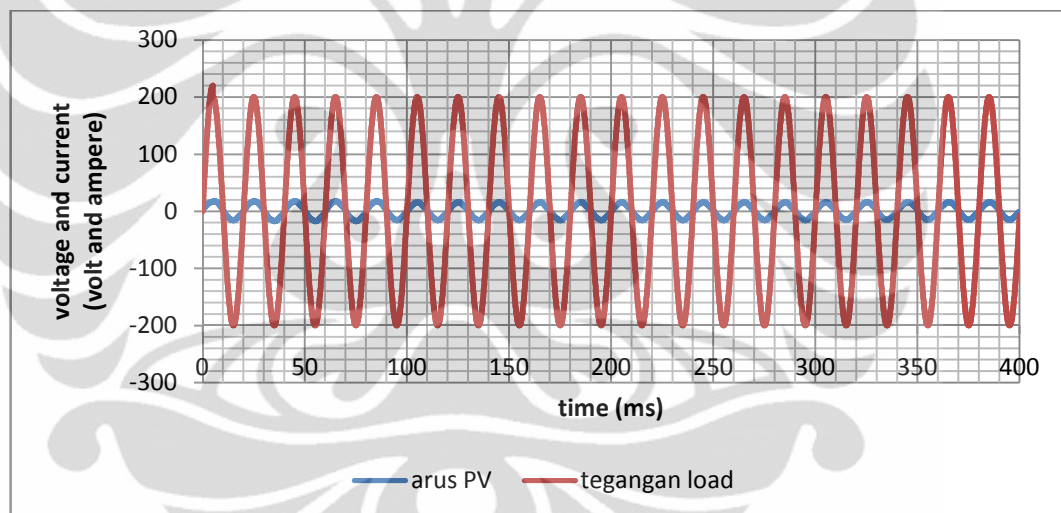
Sementara itu, fenomena lainnya yang terjadi adalah bahwa terdapat fluktuasi pada amplitudo arus PV yang terukur (terlihat pada Gambar 5.6), padahal seharusnya nilai amplitudo arus PV tersebut sesuai dengan nilai I_{qref} yang ditentukan (yaitu senilai 15 ampere). Namun demikian, fluktuasi amplitudo arus PV yang terjadi ternyata masih terdapat di sekitar nilai 15 ampere yang ditentukan sebelumnya, sehingga fluktuasi tersebut dapat diterima secara logis dan relevan.

Untuk menghasilkan sinyal PWM yang baik untuk mengendalikan *inverter*, diperlukan sinyal segitiga *carrier* yang amplitudonya sama dengan amplitudo sinyal sinusoidal yang hendak diwakili oleh sinyal PWM yang bersangkutan. Akan tetapi kondisi dari tegangan *grid* yang diengaruhi oleh banyak

faktor, hasil estimasi V_{qe}^e dan V_{qsync} yang tidak terlalu akurat, serta output *inverter* yang tidak berupa sinyal sinusoidal aktual (namun berupa sinyal sinusoidal yang direpresentasikan oleh sinyal PWM yang dimodifikasi), mengakibatkan arus PV yang terukur akhirnya tidak menjadi sempurna.

5.4 Performa PLL dengan *Current Control* dalam Kondisi Perubahan Amplitudo

Berikut ini akan ditampilkan hasil simulasi dari PLL dengan *current control* pada kondisi perubahan amplitudo, yaitu saat amplitudo sinyal tegangan listrik berubah dari 220 volt menjadi 200 volt pada waktu 5 milisekon, serta frekuensi dari sinyal tegangan listrik *grid* bersifat konstan pada nilai 50 hertz.

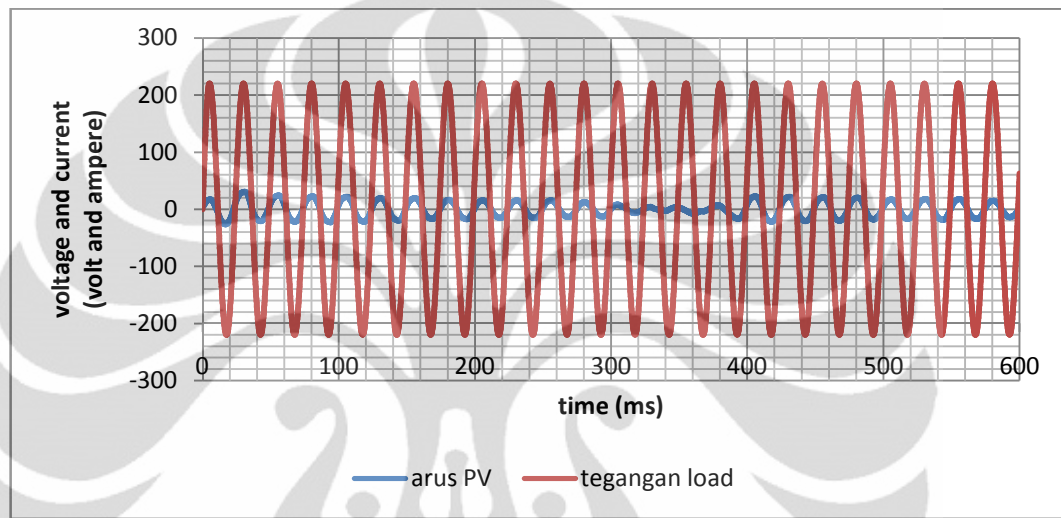


Gambar 5.7. Performa PLL dengan current control pada kondisi perubahan amplitudo

Gambar 5.7 menunjukkan bahwa dalam kondisi perubahan amplitudo sinyal tegangan listrik pada *grid*, arus PV yang terukur masih dapat melakukan sinkronisasi dengan sinyal tegangan *grid* tersebut. Sementara itu, fluktuasi yang terjadi pada arus PV yang terlihat pada grafik telah dijelaskan pada bagian sebelumnya.

5.5 Performa PLL dengan *Current Control* dalam Kondisi Perubahan Frekuensi

Berikut ini akan ditampilkan hasil simulasi dari PLL dengan *current control* pada kondisi perubahan frekuensi, yaitu saat amplitudo sinyal tegangan listrik bersifat konstan pada nilai 220 volt, serta frekuensi dari sinyal tegangan listrik PLN berubah dari 50 hertz menjadi 40 hertz pada waktu 5 milisekon.



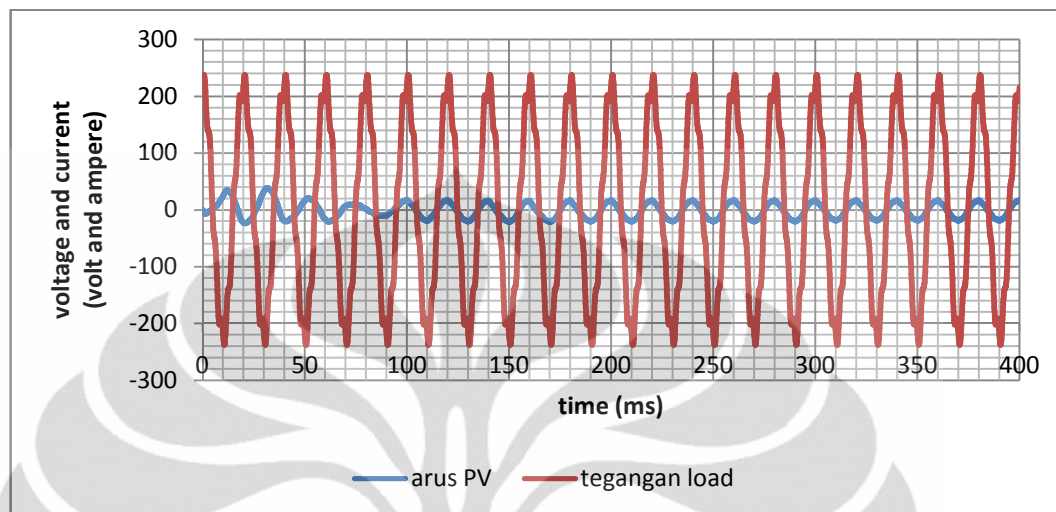
Gambar 5.8. Performa PLL dengan *current control* pada kondisi perubahan frekuensi

Gambar 5.8 menunjukkan bahwa dalam kondisi perubahan frekuensi sinyal tegangan listrik pada *grid*, arus PV yang terukur masih dapat melakukan sinkronisasi dengan sinyal tegangan *grid* tersebut, walaupun arus PV yang terukur, secara grafis, lebih terdistorsi daripada performa PLL dengan *current control* pada kondisi perubahan amplitudo. Sementara itu, fluktuasi yang terjadi pada arus PV yang terlihat pada grafik telah dijelaskan pada bagian sebelumnya.

5.6 Performa PLL dengan *Current Control* dalam Kondisi Amplitudo Ber-Noise

Berikut ini akan ditampilkan hasil simulasi dari PLL dengan *current control* pada kondisi amplitudo memiliki *noise*, yaitu saat amplitudo sinyal tegangan listrik bersifat konstan pada nilai 220 volt, namun berinterferensi dengan harmonik ketujuhnya ($7 \times 50 \text{ hertz} = 350 \text{ hertz}$) yang memiliki amplitudo sepuluh persen dari amplitudo normalnya ($10\% \times$

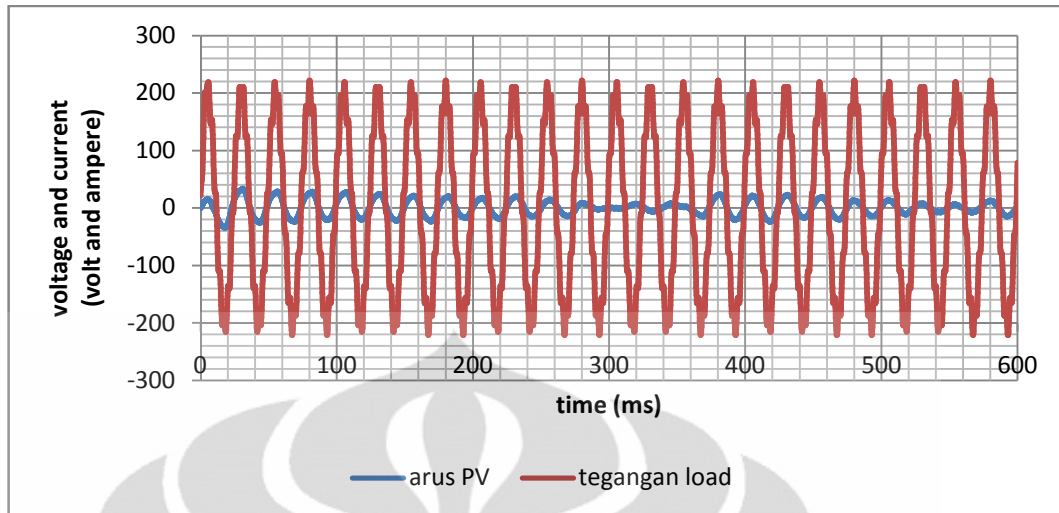
220 volt = 22 volt), serta frekuensi dari sinyal tegangan listrik PLN bersifat konstan pada nilai 50 hertz.



Gambar 5.9. Performa PLL dengan current control pada kondisi amplitudo ber-noise

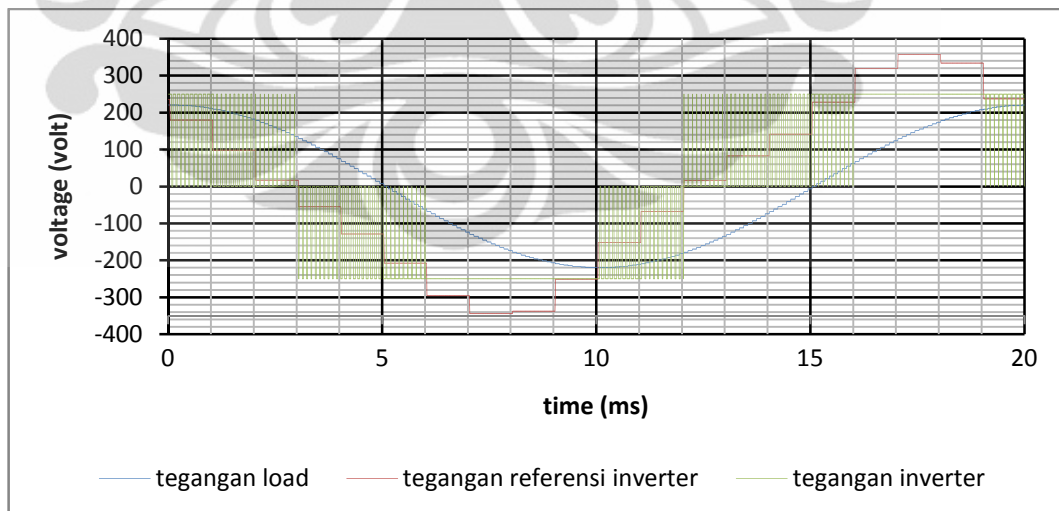
Gambar 5.9 menunjukkan bahwa dalam kondisi amplitudo sinyal tegangan listrik pada *grid* yang ber-*noise*, arus PV yang terukur masih dapat melakukan sinkronisasi dengan sinyal tegangan *grid* tersebut. Sementara itu, fluktuasi yang terjadi pada arus PV yang terlihat pada grafik telah dijelaskan pada bagian sebelumnya. Di samping itu, Gambar 5.6 sampai dengan Gambar 5.9 semakin menegaskan bahwa kondisi perubahan frekuensi merupakan kondisi yang sangat potensial mempengaruhi / mengganggu performa PLL, baik dengan maupun tanpa *current control*, jika dibandingkan dengan jenis kondisi gangguan / anomali lainnya.

Kemudian, untuk memperlihatkan bahwa PLL dengan *current control* ini mampu mengkompensasi gangguan yang terjadi secara sekaligus, berikut ini akan ditampilkan gambar yang menunjukkan performa PLL dengan *current control* pada kondisi perubahan amplitudo pada waktu 5 milisekon, perubahan frekuensi pada waktu 5 milisekon, serta amplitudo sinyal tegangan listrik berinterferensi dengan *noise*.



Gambar 5.10. Performa PLL dengan current control saat amplitudo berubah, frekuensi berubah, serta amplitudo ber-noise

Demikianlah Gambar 5.10 menunjukkan bahwa PLL dengan *current control* dapat diaplikasikan dalam kondisi gangguan yang relatif banyak, yang dalam hal ini dicontohkan oleh fluktuasi amplitudo, fluktuasi frekuensi, serta *noise* pada amplitudo. Selain itu, PLL dengan *current control* ini masih memberikan respon yang baik, bahkan pada saat ketiga jenis gangguan yang disebutkan sebelumnya terjadi pada saat yang bersamaan.



Gambar 5.11. Perbandingan sinyal tegangan pada load, sinyal tegangan referensi inverter, dan sinyal tegangan inverter

Kemudian, berikut ini juga akan ditampilkan sinyal tegangan yang terukur pada *load*, sinyal tegangan referensi *inverter*, dan sinyal tegangan *inverter*

(gambar yang lebih jelas dapat dilihat pada bagian lampiran). Gambar 5.11 ini ditunjukkan untuk memperlihatkan performa dari *inverter* yang digunakan dalam simulasi. Selain itu, gambar berikut juga menunjukkan bahwa sinyal tegangan berupa sinyal PWM yang dihasilkan oleh *inverter* ternyata dapat digunakan untuk merepresentasikan sinyal tegangan sinusoidal (AC) yang seharusnya disuplai ke dalam *grid*.



BAB 6 KESIMPULAN

Dalam skripsi ini, beberapa hal yang dapat disimpulkan antara lain adalah sebagai berikut :

1. Fleksibilitas metode PLL terlihat dari sisi keunggulannya untuk direalisasikan secara analog dengan menggunakan komponen elektronika tertentu, maupun secara digital dalam bentuk baris-baris program.
2. Metode PLL digital memiliki fleksibilitas yang lebih tinggi dan lebih mudah untuk disimulasikan terlebih dahulu daripada rangkaian PLL analog.
3. Performa PLL digital yang menggunakan perhitungan berbasis besaran tegangan listrik ternyata memberikan performa yang sama dengan PLL digital yang menggunakan perhitungan berbasis daya, karena bentuk persamaan matematis akhir mereka yang sama walaupun diturunkan dari prinsip elektronika yang berbeda.
4. Salah satu metode yang dapat meningkatkan performa PLL digital adalah dengan menggunakan algoritma filter yang berfungsi untuk menghilangkan sinyal-sinyal berfrekuensi tinggi yang berpotensi menyebabkan osilasi dan ketidakstabilan.
5. Metode lainnya yang dapat meningkatkan performa PLL digital adalah dengan menggunakan algoritma APF yang berfungsi untuk melakukan pergeseran fasa sebesar 90° terhadap suatu sinyal input tertentu yang memasuki blok PLL.
6. *Phase Locked Loop* yang telah dilengkapi dengan *low pass filter* dan APF dapat diaplikasikan bersama dengan blok *current control* untuk melakukan operasi suplai paralel antara *grid* dengan PV yang output DC-nya telah diproses oleh *inverter* menjadi sinyal tegangan AC.
7. Bentuk gangguan berupa fluktuasi frekuensi sinyal tegangan yang terjadi pada *grid* merupakan bentuk kondisi yang paling mendistorsi performa PLL dengan *current control* jika dibandingkan dengan gangguan berupa fluktuasi amplitudo maupun amplitudo yang ber-*noise*.
8. *Phase Locked Loop* dengan *current control* dapat memberikan performa yang baik dengan waktu pencuplikan sebesar 1 milisekon. Selain itu, PLL dengan *current control* inipun dapat memberikan performa yang baik dalam kondisi

gangguan fluktuasi amplitudo, fluktuasi frekuensi, serta amplitudo ber-*noise*, baik ketiga gangguan tersebut muncul secara masing-masing maupun secara bersamaan.



DAFTAR PUSTAKA

Castilla, M., Miret, J., Matas, J., Vicuña, L. G., & Guerrero, J. M. (2008). Linear Current Control Scheme With Series Resonant Harmonic Compensator for Single-Phase Grid-Connected Photovoltaic Inverters. *IEEE Transactions on Industrial Electronics* , 2724-2728.

Cha, H., & Lee, S. (2008). Design and Implementation of Photovoltaic Power Conditioning System using Current based Maximum Power Point Tracking. 1-5.

Langton, C. (1998). *Intuitive Guide to Principles of Communication*. Retrieved November 20, 2010, from complextoreal: <http://www.complextoreal.com>

Mok, H. S., Choe, G. H., Kim, S. H., Lee, J. M., & Suh, I. Y. (2008). Current THD Reduction and Anti-islanding Detection in Distributed Generation with Grid Voltage Distortion. *IEEE International Conference on Sustainable Energy Technologies, 2008. (ICSET 2008)* , 990.

Rey, A. B., Pablo, S. d., Ruiz, J., & Ruz, F. d. (2005). Two Novel Current Source Sliding Mode Control Strategies for a 3-Phase Grid-Connected Inverter. *Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. (ISIE 2005). Volume: 2* , 763-767.

Silva, S. A., Novochadlo, R., & Modesto, R. A. (2008). Single-Phase PLL Structure Using Modified p-q Theory for Utility Connected Systems. 4707.

Sujanarko, B. (2010). Metode Sinkronisasi Inverter Satu Fase dengan Jaringan Listrik yang Terdistorsi. *TELKOMNIKA* , 49-56.

Syamsuddin, S., Rahim, N., Krismadinata, & Selvaraj, J. (n.d.). Implementation of TMS320F2812 in Islanding Detection for Photovoltaic Grid Connected Inverter. *International Conference for Technical Postgraduates (TECHPOS), 2009* , 1-3.

LAMPIRAN

