

Deteksi Osilasi Masalah Penalaan Kontroler secara Real-time dengan Metode Deteksi Puncak Spektrum

By Bobby A. Permana

Deteksi Osilasi Masalah Penalaan Pengendali secara *Real-time* dengan Metode Deteksi Puncak Spektrum

Boby A. Permana ¹⁾ dan Awang N. I. Wardana ²⁾

14

^{1,2)} Departemen Teknik Nuklir dan Teknik Fisika, Universitas Gadjah Mada, Indonesia

e-mail korespondensi: ¹⁾boby.aoliya.p@mail.ugm.ac.id dan ²⁾awang.wardana@ugm.ac.id

Abstract/Abstrak – In process industry, most control loops suffer with disturbances indicated by existing oscillations. One of the major problems of the oscillation is controller tuning problem. Peak spectral detection method which is classic and simple way is proposed as technique for detecting oscillation. Nevertheless, the utilization and performance of peak spectral detection on process oscillation detection still cannot be found. This study develops the utilization and performance of peak spectral detection method. It implemented based on IEC 61499. To evaluate the method, test data are used to set the lag value in peak spectral detection method and industrial data for obtaining performance of peak spectral detection method. With fast Fourier transform length 512 points and lag value 180 points, the developed peak spectral detection method detected two oscillations of three different oscillating signals at 461 s and 441 s.

Keywords: control loop, tuning problem, peak spectral, process control, oscillation detection.

I. PENDAHULUAN

Operasi sistem kendali pada industri proses akan selalu mengalami gangguan yang diindikasikan dengan adanya osilasi pada sinyal kalang kendali [1]. Survey terhadap industri-industri proses menunjukkan bahwa 30% kalang kendali mengalami osilasi [2]. Osilasi dapat diakibatkan oleh stiksi pada katup, penalaan pengendali yang buruk, dan gangguan eksternal [3]. Penalaan pengendali yang buruk adalah penyebab dominan dari osilasi karena kebanyakan pengendali masih diatur menggunakan cara pengaturan lama [2] juga masih banyak pengendali dalam industri proses yang tidak diatur dengan baik [4]. Studi menunjukkan bahwa sekitar 66% sampai dengan 80% pengendali pada industri proses memiliki masalah terkait kinerja akibat adanya osilasi [4]. Penurunan kinerja kalang kendali akan mengurangi produktivitas, efisiensi, dan keuntungan.

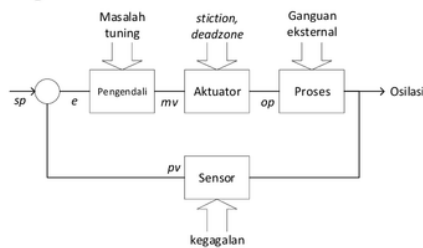
Sudah banyak metode yang dikembangkan untuk mendeteksi osilasi. Babuska dkk [5] mendeteksi osilasi ketika amplitudo maksimum dalam *Power Spectral Density* (PSD) jauh lebih besar dibanding amplitudo pada frekuensi rendah dilanjutkan dengan tahap kuantifikasi dimana model logika *fuzzy* dilatih menggunakan data pengukuran dan pengetahuan terkait proses. Zhang dkk [6] mendeteksi osilasi ketika amplitudo pada hasil *Fast Fourier Transform* (FFT) melewati nilai batas tertentu yang ditentukan berdasarkan distribusi Raleigh dan

tingkat kepercayaan yang diatur oleh pengguna. Hagglund [7] mendeteksi osilasi dari kelambanan pengendali berdasarkan pada indeks *idle* yang dipengaruhi oleh korelasi antara nilai ralat kendali dan nilai titik operasi pada daerah transisi namun sangat sensitif terhadap derau. Kuehl dkk [8] mengajukan pra-pemrosesan sebelum mendeteksi osilasi, yaitu dengan teknik *filtering*, deteksi kondisi tunak, dan kuantisasi sinyal yang memiliki beberapa parameter yang harus diatur dan tergantung dari karakteristik sinyal.

Osilasi pada kalang kendali dapat dideteksi menggunakan metode deteksi puncak spektrum [1]. Metode deteksi puncak spektrum adalah metode klasik sederhana berbasis frekuensi namun penggunaan dan dokumentasi kinerja pada kasus deteksi osilasi kalang kendali masih sangat minim. Metode yang sederhana dan tepat seringkali dipilih oleh para insinyur di industri proses.

Pada penelitian ini, metode deteksi puncak spektrum akan dikembangkan dan diimplementasikan secara *real-time*. Hasil implementasi akan diuji menggunakan data kalang kendali yang berosilasi. Sebelum diuji, dilakukan pengaturan nilai parameter dan pengukuran waktu eksekusi. Metode yang dikembangkan kemudian di implementasikan dalam IEC 61499. Selanjutnya, evaluasi dilakukan dengan menggunakan data industri.

Sistem kalang kendali terdiri dari pengendali, aktuator, proses, dan sensor (Gambar 1). Gangguan pada salah satu komponen dari sistem atau kinerja kalang kendali yang buruk dapat diindikasikan oleh adanya osilasi. Osilasi didapatkan melalui sinyal kalang kendali yang terdiri dari variabel manipulator, keluaran aktuator, variabel proses, dan ralat kendali. Satu atau lebih sinyal kalang kendali dapat digunakan untuk mendeteksi osilasi.



Gambar 1. Osilasi Kalang Kendali

Metode deteksi puncak spektrum terdiri dari jendela digital, *Autocorrelation Function* (ACF), jendela *Bartlett*, *Fast Fourier Transform* (FFT), dan algoritma deteksi puncak [9]. Metode deteksi puncak spektrum bekerja dengan mendeteksi puncak pada *Power Spectral Density* (PSD) dimana PSD adalah hasil FFT dari ACF.

Jendela digital berfungsi untuk memotong barisan data. Jendela digital menggunakan operasi perkalian dalam domain waktu sebagai:

$$v[n] = x[n]w[n] \quad (1),$$

dimana $x[n]$ adalah barisan data masukan, $w[n]$ adalah amplitudo dari jendela digital, dan $v[n]$ adalah hasil operasi jendela digital. Nilai $w[n]$ tergantung dari tipe jendela digital yang digunakan.

ACF menunjukkan karakter ketergantungan statistik dari suatu data yang berubah terhadap waktu [3]. ACF akan meredam sinyal derau. Hasil ACF dari sinyal beresiliasi adalah sinyal beresiliasi dengan derau yang teredam [2]. ACF (2) dapat dilakukan estimasi [9] dengan:

$$\hat{\varphi}_{xx}[m] = \frac{1}{Q} \sum_{n=0}^{Q-|m|-1} x[n]x[n+|m|] \quad (2),$$

dimana $\hat{\varphi}_{xx}$ adalah estimasi ACF, m adalah lag yang memiliki rentang dari $-M$ sampai dengan M , M adalah batas lag, dan Q adalah panjang konvolusi ACF yang memiliki nilai sebesar $N - M + 1$.

Jendela *Bartlett* adalah satu-satunya jendela yang dapat menjamin PSD bernilai positif. Hasil FFT dari jendela *Bartlett* selalu bernilai positif. Bentuk dari jendela *Bartlett* adalah segitiga dengan garis yang memiliki gradien tergantung panjang barisan data [9] yang dirumuskan sebagai:

$$w[n] = \begin{cases} \frac{2n}{N-1}, & 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1}, & \frac{N+1}{2} \leq n \leq N-1 \end{cases} \quad (3),$$

dimana $w[n]$ adalah amplitudo jendela dan N adalah panjang barisan data.

FFT digunakan untuk melakukan transformasi dari domain waktu ke domain frekuensi yang dirumuskan sebagai:

$$X[k] = \begin{cases} \sum_{n=0}^{N-1} x[n]W_N^{kn}, & 0 \leq k \leq N-1 \\ 0, & k < 0 \text{ atau } k > N-1 \end{cases} \quad (4),$$

dimana $x[n]$ adalah sinyal dalam domain waktu dan $X[k]$ adalah sinyal dalam domain frekuensi.

Algoritma deteksi puncak digunakan setelah dilakukan FFT terhadap hasil ACF. Algoritma deteksi puncak [3] bekerja berdasarkan perhitungan rata-rata sebagian sampel, sebagai:

$$\bar{x}[n] = \frac{x[n-5] + x[n-4] + \dots + x[n+5]}{10} \quad (5),$$

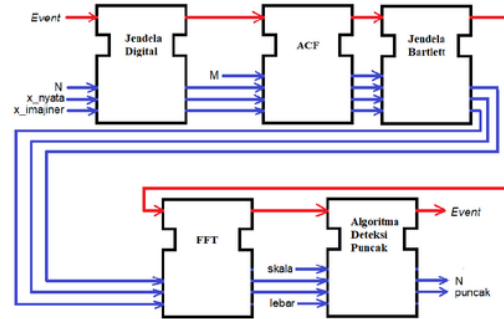
$$x_p[n] \geq 3\bar{x}[n] \quad (6),$$

dimana $\bar{x}[n]$ adalah nilai rata-rata lokal dari 10 sampel dan $x_p[n]$ adalah nilai titik puncak lokal dari 10 sampel.

II. METODOLOGI

A. Implementasi Metode Deteksi Puncak Spektrum pada IEC 61499

Setiap bagian dari metode deteksi puncak spektrum (Gambar 2) seperti jendela digital, ACF, jendela *Bartlett*, Fast Fourier Transform (FFT), dan algoritma deteksi puncak diimplementasikan pada IEC 61499 menjadi sebuah blok fungsi dasar (BFD). Satu titik data akan dikirimkan terlebih dahulu ke jendela digital sebelum dioperasikan oleh ACF. Operasi ACF menghasilkan sampel estimasi ACF yang selanjutnya akan dikirimkan ke jendela *Bartlett*. Data keluaran dari jendela *Bartlett* akan dikirimkan ke FFT sehingga mendapatkan sampel estimasi PSD. Sampel estimasi PSD adalah barisan data yang menunjukkan spektrum densitas daya. Sampel estimasi PSD selanjutnya dimasukkan ke algoritma deteksi puncak untuk mendapatkan puncak spektrum dari sampel estimasi PSD.



Gambar 2. Implementasi pada IEC 61499

Operasi untuk menghitung sampel estimasi ACF (Algoritma 1) dimasukkan ke dalam blok fungsi ACF. Perhitungan sampel estimasi ACF dimulai dengan perhitungan nilai Q . Proses perhitungan dilanjutkan oleh kalang pertama yang bekerja dengan menaikkan nilai indeks lag sampai nilai indeks lag sama dengan besar lag. Di dalam kalang pertama terdapat kalang kedua. Kalang kedua melakukan perhitungan konvolusi barisan data terhadap masing-masing nilai indeks lag yang diberikan oleh kalang pertama. Panjang titik data yang akan dikonvolusi tergantung pada nilai Q dan indeks lag. Masing-masing titik data dibagi dengan nilai Q untuk mendapatkan sampel estimasi ACF setelah keluar dari kalang kedua. Kalang pertama menyimpan sampel estimasi ACF terhadap masing-masing indeks lag. Operasi ACF diakhiri dengan mengeluarkan besar lag dan panjang titik data dari blok fungsi.

Algoritma 1: ACF

Deklarasi
 N : UINT; M : UINT;
 $x[n]$: LREAL512; $acf[m]$: LREAL512;
 Q : UINT;
 $Q = N - M$;
for $i = 1 : M$ **do**
 Inisiasi
 $x_nyata = 0$;
 $x_imajiner = 0$;
for $j = 1 : Q - i$ **do**
 $x_nyata = x_nyata +$ Perkalian titik data $(i, j, x[n])$;
 $x_imajiner = x_imajiner +$ Perkalian titik data $(i, j, x[n])$;
end
 $acf_nyata[m] \leftarrow x_nyata / Q$;
 $acf_imajiner[m] \leftarrow x_imajiner / Q$;
end
 $M_out \leftarrow M_in$;
 $N_out \leftarrow N_in$;

Blok fungsi jendela *Bartlett* berisi operasi jendela *Bartlett* (Algoritma 2). Proses perhitungan dimulai dengan menghitung gradien berdasarkan nilai besar *lag*. Proses perhitungan dilanjutkan oleh kalang pertama. Kalang pertama akan melakukan perkalian titik data masukan dengan amplitudo jendela. Proses perkalian pada kalang pertama dimulai dari nilai indeks *lag* nol sampai dengan besar *lag*. Gradien pada kalang pertama bernilai negatif. Amplitudo jendela akan menurun seiring dengan meningkatnya indeks *lag*. Setelah keluar dari kalang pertama, proses perhitungan dilanjutkan oleh kalang kedua. Kalang kedua melakukan perkalian titik data masukan dengan amplitudo jendela. Amplitudo jendela akan meningkat seiring dengan meningkatnya indeks *lag* karena gradien bernilai positif. Proses perkalian terjadi dari nilai indeks sebesar panjang titik data dikurangi besar *lag* sampai dengan nilai indeks sebesar panjang titik data. Hasil dari operasi jendela *Bartlett* memiliki sebagian daerah dalam barisan data yang tidak dilakukan perkalian dengan amplitudo jendela. Sebagian daerah dalam barisan data akan bernilai nol. Panjang dari daerah yang bernilai nol akan tergantung pada besar *lag*. Nilai besar *lag* yang tinggi menyebabkan daerah yang bernilai nol semakin kecil.

Algoritma 2: Jendela Bartlett

Deklarasi
 M : UINT; N : UINT;
 $grad$: UINT; w : LREAL;
 $s[m]$: LREAL512; $x[n]$: LREAL512;
for $i = 1 : M$ **do**
 $w = 1 - (i * grad)$;
 $s_nyata[m] \leftarrow w * x_nyata$;
 $s_imajiner[m] \leftarrow w * x_imajiner$;
end

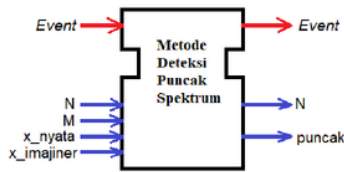
$j = M$;
for $i = (N - M) : N$ **do**
 $w = 1 * (i - (N - M)) * grad$;
 $s_nyata[m] \leftarrow w * x_nyata$;
 $s_imajiner[m] \leftarrow w * x_imajiner$;
 $j = j - 1$;
end
 $N_out = N_in$;

Blok fungsi FFT menggunakan *event-driven* FFT [10]. Blok fungsi untuk deteksi puncak berisi operasi deteksi puncak (Algoritma 3). Proses perhitungan dilakukan oleh satu kalang besar. Kalang besar bekerja dengan prinsip menggerakkan titik pusat perhitungan rata-rata dalam barisan data. Di dalam kalang besar terdapat kalang kecil. Kalang kecil digunakan untuk melakukan penjumlahan titik-titik data. Penjumlahan titik-titik data dilakukan sesuai dengan lebar sampel. Proses perhitungan dilanjutkan dengan membagi hasil penjumlahan dengan lebar sampel sehingga mendapatkan nilai rata-rata. Titik pusat akan diuji dalam percabangan apakah telah memenuhi kriteria sebagai titik puncak atau tidak. Nilai pada titik pusat yang lebih besar dari nilai skala dikalikan nilai rata-rata ditentukan sebagai titik puncak.

Algoritma 3: Algoritma deteksi puncak

Deklarasi
 $lebar\ sampel$: UINT; N : UINT;
 $x[n]$: LREAL512; $puncakPSD[k]$: LREAL512;
for $j = (lebar\ sampel / 2) : (N - (lebar\ sampel / 2))$ **do**
 $jumlah = 0$;
 $i = j$;
 $ia = i - (lebar\ sampel / 2)$;
 $ib = i + (lebar\ sampel / 2)$;
for $i = ia : ib$ **do**
 $jumlah = jumlah + x$;
end
 $rata-rata = jumlah / lebar\ sampel$;
if $x_pusat > skala * rata-rata$ **then**
 $puncakPSD[k] \leftarrow 1$;
end
end
 $N_out \leftarrow N_in$;

Keseluruhan dari operasi yang telah diimplementasikan pada sebuah BFD yang saling terhubung. Untuk menghasilkan *library* yang dapat dipakai maka keseluruhan operasi tersebut disatukan dalam satu blok fungsi komposit (BFK). Metode deteksi puncak spektrum diimplementasikan menjadi satu buah BFK (Gambar 3). Input dari blok fungsi adalah panjang barisan data (N), *lag* (M) dan hasil FFT real (x_nyata) maupun imajiner ($x_imajiner$) dari variable proses. Keluaran dari meliputi barisan data (N) dan hasil deteksi (*puncak*).



Gambar 3. Komposit blok metode deteksi puncak spektrum pada IEC 61499

B. Data uji

Data uji yang digunakan didapatkan dari satu *database* yang sama [3]. Data tersebut terdiri dari dua macam data yaitu data osilasi buatan dan data osilasi dari industri. Pada studi ini, data buatan digunakan untuk mengatur nilai batas *lag* sedangkan data osilasi dari industri digunakan untuk menguji metode deteksi puncak spektrum dengan batas *lag* yang sudah ditetapkan.

1) Data buatan 1 (TESTDATA 1)

Data berupa sinyal dengan distribusi frekuensi seragam yang direpresentasikan dengan:

$$x_1(t) = \left(\frac{1-0.2q^{-1}}{1-0.1q^{-1}+0.8q^{-2}} v(t) \right) \quad (6),$$

dimana $v(t)$ adalah rerata nol derau putih dengan variansi 0.2. Data ini adalah sinyal yang tidak beresilasi.

2) Data buatan 2 (TESTDATA 2)

Data berupa sinyal dengan satu frekuensi dominan ditambah dengan derau yang dideskripsikan dengan:

$$x_2(t) = 2.5 \sin(0.1t) + x_1(t) \quad (7),$$

dimana $x_1(t)$ adalah derau pengukuran dengan distribusi frekuensi seragam.

3) Data buatan 3 (TESTDATA 3)

Data berupa sinyal teredam terhadap waktu dengan satu frekuensi dominan ditambah dengan derau.

$$x_3(t) = 10e^{-0.005t} \sin(0.1t) + x_1(t) \quad (8),$$

dimana $e^{-0.005t}$ adalah suku peredam dan $x_1(t)$ adalah derau pengukuran dengan distribusi frekuensi seragam.

4) Data kalang kendali 1 (BAS 6)

Data kalang kendali 1 merupakan data variable proses temperatur pada bangunan dengan periode cuplikan 1 detik. Data tersebut adalah data yang beresilasi yang diakibatkan oleh akibat stiksi dan masalah penalaan pengendali.

5) Data kalang kendali 2 (CHEM 4)

Data kalang kendali 2 adalah berupa ketinggian fluida

pada industri petrokimia dengan periode cuplikan 1 detik. Data tersebut adalah data yang beresilasi akibat masalah penalaan pengendali.

6) Data kalang kendali 3 (PAP 4)

Data kalang kendali 3 adalah konsentrasi pada industri kertas dengan periode cuplikan 1 detik. Data tersebut adalah data yang beresilasi akibat *deadzone* dan masalah penalaan pengendali.

C. Tahapan Evaluasi

Metode deteksi puncak spektrum yang telah diimplementasikan menjadi blok fungsi harus dievaluasi. Tahapan evaluasi terdiri dari pengaturan batas *lag*, pengukuran waktu eksekusi, dan pengujian metode deteksi (Gambar 3). Pengaturan batas *lag* menggunakan data buatan. Sedangkan pada pengukuran waktu eksekusi dan pengujian menggunakan data kalang kendali



Gambar 3. Alur evaluasi

III. HASIL DAN PEMBAHASAN

Tahapan pertama studi ini adalah untuk mengatur batas *lag*. Nilai batas *lag* yang diatur tidak boleh lebih besar dari separuh panjang FFT. Panjang FFT yang digunakan adalah 512 titik.

Pada studi ini nilai batas *lag* divariasikan dari 50 sampai dengan 250 dengan selisih setiap percobaan sebesar 10 titik untuk mencari nilai *lag* minimum yang dibutuhkan (Tabel 1). Perubahan nilai batas *lag* akan mengubah besar variansi dalam perhitungan estimasi PSD. Variansi yang semakin besar akan menyebabkan bentuk spektrum semakin kasar. Kekasaran bentuk spektrum tersebut akan berpengaruh pada kinerja dari metode deteksi puncak spektrum. Terlihat pada kolom kedua Tabel 1, metode deteksi puncak spektrum dapat mendeteksi dengan benar dimana pada kondisi tersebut TESTDATA 1 hanya merupakan sinyal derau warna. Metode deteksi puncak spektrum telah menentukan dengan tepat bahwa sinyal derau warna tidak termasuk sinyal yang beresilasi.

Pada kolom ketiga Tabel 1, metode deteksi puncak spektrum mendeteksi osilasi dari nilai batas *lag* sebesar 160 sampai dengan 250. Hasil perobaan dari TESTDATA 2 menunjukkan bahwa bentuk spektrum dari PSD harus cukup kasar agar dapat menentukan frekuensi dominan dari data uji sebagai frekuensi osilasi.

Tabel 1. Nilai Batas Lag untuk data buatan

Batas Lag	TESTDATA 1	TESTDATA 2	TESTDATA 3
50	-	-	-
60	-	-	-
70	-	-	-
80	-	-	-
90	-	-	-
100	-	-	-
110	-	-	-
120	-	-	-
130	-	-	-
140	-	-	-
150	-	-	-
160	-	V	-
170	-	V	-
180	-	V	V
190	-	V	V
200	-	V	V
210	-	V	V
220	-	V	V
230	-	V	V
240	-	V	V
250	-	V	V

Untuk TESTDATA 3, metode deteksi puncak spektrum mulai dapat mendeteksi osilasi dengan nilai batas lag sebesar 180 sampai dengan 250. Perbedaan dengan TESTDATA 2 terjadi karena TESTDATA 3 adalah sinyal yang amplitudonya teredam terhadap waktu.

Berdasarkan tiga percobaan, nilai batas lag yang ditentukan yaitu sebesar 180. Karena dengan nilai batas lag 180, metode deteksi puncak spektrum mampu mendeteksi osilasi dari dua sinyal uji. Nilai batas lag 250 juga mampu mendeteksi osilasi dari dua sinyal namun akan membuat metode deteksi menjadi cenderung menghasilkan *false positive*. Kecendrungan *false positive* akan menurunkan kinerja dari metode deteksi puncak spektrum.

Tabel 2. Waktu eksekusi metode deteksi

Batas Lag	Rata-rata (ms)	Standar Deviasi (ms)	Maksimum (ms)
50	58,5	21,9	145
180	87,6	27,3	179
250	73,9	25,3	158

Selanjutnya, pada studi ini pengukuran waktu eksekusi dilakukan dengan menggunakan FFT 512 titik dan batas lag 50, 180, dan 250. Pengukuran waktu eksekusi (Tabel 2) penting dilakukan karena metode deteksi akan diuji secara *real-time*. Waktu eksekusi harus lebih kecil dibanding dengan waktu cuplikan. Data uji maupun data kalang kendali memiliki waktu cuplikan 1 detik sehingga

waktu eksekusi harus lebih kecil dari 1 detik agar dapat bekerja secara *real-time*.

Waktu eksekusi yang diukur adalah waktu yang dibutuhkan perangkat keras untuk melakukan metode deteksi puncak spektrum pada arsitektur IEC 61499. Perangkat lunak yang digunakan untuk memodelkan dan melakukan *export* blok fungsi adalah 4DIAC-IDE [11]. FORTE [12] digunakan sebagai lingkungan *runtime* untuk mengeksekusi blok fungsi. Waktu eksekusi juga sangat tergantung pada spesifikasi perangkat keras. Perangkat keras yang digunakan adalah komputer dengan prosesor Intel i3-6006U dual-core, *clock rate* 2 GHz, dan RAM 4GB.

Hasil menunjukkan bahwa batas lag 180 memiliki waktu eksekusi yang lebih besar. Hal tersebut dapat ditunjukkan oleh rata-rata atau maksimum. Dalam hal praktis, nilai maksimum yang dijadikan acuan. Nilai maksimum waktu eksekusi tidak boleh lebih besar dari periode cuplikan sinyal. Metode deteksi puncak spektrum dapat diimplementasikan secara *realtime* pada IEC 61499 karena nilai maksimum waktu eksekusi dari batas lag 50 sampai dengan 250 masih lebih kecil dibanding dengan periode cuplikan. Nilai maksimum pada batas lag 180 adalah 179 ms sedangkan periode cuplikan yaitu 1 s. Dengan melihat rata-rata waktu eksekusi dan standar deviasi, dapat diprediksi bahwa waktu eksekusi akan jarang sekali mendekati nilai maksimumnya.

Setelah diketahui nilai batas lag yang digunakan, maka kemudian dapat dilakukan pengujian dengan menggunakan tiga buah data kalang kendali. Hasil dari pengujian (Tabel 3) menunjukan tiga data kalang kendali, dua data terdeteksi beresilasi dan satu data terdeteksi tidak beresilasi. Terdeteksi osilasi artinya terdeteksi masalah penalaan pengendali. Tiga data yang digunakan sudah dipastikan berupa sinyal beresilasi yang diakibatkan oleh penalaan pengendali yang buruk.

Tabel 3. Pengujian Metode Deteksi

Data	Frekuensi (rad/s)	Waktu (s)	Penyebab Osilasi	Hasil Deteksi
BAS 6	-	-	Stiksi & Penalaan	Tidak Terdeteksi
CHEM 4	0,3804	461	Penalaan	Terdeteksi
PAP 6	0,1473	441	Deadzone & Penalaan	Terdeteksi

Untuk data BAS 6, osilasi tidak dapat dideteksi oleh metode deteksi puncak spektrum. Karakter osilasi tidak reguler baik dari aspek frekuensi maupun amplitudo. BAS 6 adalah sinyal osilasi yang diakibatkan masalah penalaan pengendali dan stiksi secara sekaligus [3]. Sifat tidak reguler dapat diakibatkan oleh stiksi karena stiksi bersifat nonlinier dan memiliki spektrum frekuensi yang melebar.

Pada baris kedua Tabel 1 terlihat osilasi pada data CHEM 4 dapat dideteksi oleh metode deteksi puncak spektrum yang dikembangkan. Osilasi dari CHEM 4 bersifat reguler secara aspek frekuensi. Namun, amplitudo tidak begitu reguler. Jika diperhatikan, titik tengah osilasi selalu berubah terhadap waktu. Titik tengah osilasi dari CHEM 4 lebih mudah diamati pergerakannya dibanding dengan titik tengah osilasi pada BAS 6.

Osilasi pada data PAP 6 dapat dideteksi oleh metode deteksi puncak spektrum. PAP 6 lebih reguler dibanding

BAS 6 dan CHEM 4, baik dari segi amplitudo maupun frekuensi. Meskipun, PAP 6 diakibatkan oleh *deadzone* dan masalah penalaan pengendali. *Deadzone* bersifat nonlinier sama seperti stiksi. Dampak dari *deadzone* tidak begitu signifikan terhadap karakter osilasi. Metode deteksi puncak spektrum hanya mendapatkan satu titik frekuensi osilasi. Osilasi yang berhasil dideteksi adalah osilasi yang disebabkan oleh masalah penalaan pengendali bukan *deadzone*.

Metode deteksi puncak spektrum mendeteksi osilasi pada sampel ke-461 untuk CHEM 4 dan sampel ke-441 untuk PAP 6. Panjang FFT dan panjang jendela yang digunakan adalah 512 titik. Kedua hasil tersebut menunjukkan jumlah titik sampel harus cukup penuh terlebih dahulu agar metode deteksi puncak spektrum dapat mendeteksi osilasi. Jumlah titik sampel ini tergantung dari panjang FFT, panjang jendela dan nilai batas *lag*.

Metode deteksi puncak spektrum yang diajukan memiliki beberapa kelebihan. Pertama, untuk menggunakannya tidak memerlukan pengetahuan proses. Kedua, tahan terhadap derau, khususnya adalah derau warna. Ketiga, tidak memerlukan tahapan pra-pemrosesan sinyal. Keempat, memiliki beban perhitungan yang ringan sehingga memungkinkan aplikasi metode secara realtime. Namun, metode deteksi puncak spektrum memiliki satu parameter yang harus diatur agar metode deteksi bekerja dengan optimal.

IV. KESIMPULAN

Metode deteksi puncak spektrum dapat digunakan sebagai teknik untuk mendeteksi osilasi pada kalang kendali yang diakibatkan oleh buruknya penalaan pengendali. Metode deteksi puncak spektrum memerlukan pengaturan nilai batas *lag* yang sesuai. Dengan nilai batas *lag* sebesar 180 dan panjang FFT sebesar 512 titik, metode deteksi puncak spektrum dapat mendeteksi dua osilasi dari tiga sinyal berosilasi dengan karakter yang berbeda.

Untuk meningkatkan kinerja dari metode yang telah dikembangkan, terutama untuk dapat mendeteksi osilasi non-linier yang dimungkinkan terjadi karena adanya stiksi pada katup maka metode tersebut perlu diimplementasikan dengan menggabungkan metode lain seperti yang dikembangkan dengan menggunakan *variational mode decomposition* (VMD) [13, 14].

DAFTAR PUSTAKA

- [1] N. F. Thornhill dan A. Horch, "Advances and New Directions in Plant-Wide Disturbance Detection and Diagnosis," *Control Engineering Practice*, vol. 15, hal. 117-1206, 2007.
- [2] J. W. V. Dambros, J. O. Trierweiler, dan M. Farenzena, "Oscillation detection in process industries – Part I: Review of the detection methods," *Journal of Process Control*, vol. 78, hal. 108-123, 2019.
- [3] M. Jelali dan B. Huang, "Detection of Oscillating Control Loops," in *Detection and Diagnosis of Stiction in Control Loop: State of the Art and Advanced Methods*, London, Springer, 2010, hal. 81-94, 2010.
- [4] K. Ghosh, U. Nallasivam, dan N. Kubal, "A Method for Automatic Detection of Controller Tuning Issues," *IFAC Papers Online*, vol. 48, hal. 765-770, 2015.
- [5] J. Babuska, J. van Ast, dan S. Mesic, "A fuzzy-logic system for detecting oscillations in control loops," *IEEE on International Conference on Fuzzy System*, 2006.

- [6] K. Zhang, B. Huang, dan G. Ji, "Multiple oscillations detection in control loops by using DFT and Raleigh distribution," *IFAC Papers Online*, vol. 48, hal. 529-534, 2015.
- [7] T. Hagglund, "Automatic detection of sluggish control loops," *Control Engineering Practice*, vol. 7, hal. 1505-1511, 1999.
- [8] P. Kuehl dan A. Horch, "Detection of sluggish control loops – Experiences and improvements," *Control Engineering Practice*, vol. 13, hal. 1019-1025, 2005.
- [9] A. V. Oppenheim dan R. W. Schaffer, *Discrete-Time Signal Processing*, Upper Saddle River: Pearson, 2015.
- [10] B. A. Permana, A. N. I. Wardana, N. Effendy, "Implementation of Event-Driven Fast Fourier Transform based on IEC 61499," *2019 International Conference on Science and Technology*, 2019.
- [11] A. Zoitl, T. Strasser, dan A. Valentini, "Open source initiatives as basis for the establishment of new technologies in industrial automation: 4DIAC a case study," *2010 IEEE International Symposium on Industrial Electronics*, 2010.
- [12] T. Strasser, M. Rooker, G. Ebenhofer, A. Zoitl, C. Sunder, A. Valentini, dan A. Martel, "Framework for distributed industrial automation and control (4DIAC)," *6th IEEE International Conference on Industrial Informatics*, hal. 3-288, 2008.
- [13] A. N. I. Wardana, "A Method for Detecting the Oscillation in Control Loops based on Variational Mode Decomposition", *2015 International Conference on Computer, Control, Informatics and Its Applications*, 2015.
- [14] G.R.R. Dewa, A.N. I. Wardana, S. Hawibowo, "Linear Oscillation Diagnosis of Process Variable in Control Loop based on Variational Mode Decomposition", *2018 International Conference on Science and Technology*, 2018.

Deteksi Osilasi Masalah Penalaan Kontroler secara Real-time dengan Metode Deteksi Puncak Spektrum

ORIGINALITY REPORT

9%

SIMILARITY INDEX

PRIMARY SOURCES

- 1 Milan Vathoopan, Benjamin Brandenbourger, Amil George, Alois Zoitl. "Towards an integrated plant engineering process using a data conversion tool for AutomationML", 2017 IEEE International Conference on Industrial Technology (ICIT), 2017
Crossref 32 words — 1%
- 2 "Verifying IEC 61499 Applications", Industrial Information Technology, 2016.
Crossref 29 words — 1%
- 3 Awang N. I. Wardana. "A comparative study of EMD, EWT and VMD for detecting the oscillation in control loop", 2016 International Seminar on Application for Technology of Information and Communication (ISemantic), 2016
Crossref 28 words — 1%
- 4 research.aalto.fi
Internet 26 words — 1%
- 5 Wenbing Zhu, Zijiang Yang, Meishuang Tang, Sun Zhou, Bin Yao, Guoli Ji. "Integrating principal component analysis and optimal histogram estimation for Bayesian control loop diagnosis", 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2016
Crossref 24 words — 1%
- 6 es.scribd.com
Internet 24 words — 1%
- 7 Elham Naghoosi, Biao Huang. "Wavelet Transform Based Methodology for Detection and Characterization 22 words — 1%

of Multiple Oscillations in Nonstationary Variables", Industrial & Engineering Chemistry Research, 2017

Crossref

-
- 8 Gilang R.R. Dewa, Awang N.I. Wardana, Singgih Hawibowo. "Linear Oscillation Diagnosis of Process Variable in Control Loop Based on Variational Mode Decomposition", 2018 4th International Conference on Science and Technology (ICST), 2018 20 words — 1%
- Crossref
-
- 9 www.depts.ttu.edu 20 words — 1%
- Internet
-
- 10 Palma, L. Brito, F. Vieira Coito, and P. Sousa Gil. "Real-time detection of oscillations in control loops", 2011 International Conference on Power Engineering Energy and Electrical Drives, 2011. 18 words — < 1%
- Crossref
-
- 11 www.iea.lth.se 15 words — < 1%
- Internet
-
- 12 opus-hshn.bsz-bw.de 14 words — < 1%
- Internet
-
- 13 S. Narasimhan, Chen Shan Kao, R. S. H. Mah. "Detecting changes of steady states using the mathematical theory of evidence", AIChE Journal, 1987 12 words — < 1%
- Crossref
-
- 14 id.123dok.com 10 words — < 1%
- Internet
-
- 15 pub-et.tuwien.ac.at 8 words — < 1%
- Internet
-
- 16 Lei Fang, Jiandong Wang, Xiaobo Tan. "An incremental harmonic balance-based approach for harmonic analysis of closed-loop systems with Prandtl–Ishlinskii operator", Automatica, 2018 8 words — < 1%
- Crossref
-

17 oparu.uni-ulm.de
Internet

8 words — < 1%

18 zh.m.wikipedia.org
Internet

8 words — < 1%

19 dspace5.zcu.cz
Internet

8 words — < 1%

EXCLUDE QUOTES OFF
EXCLUDE
BIBLIOGRAPHY ON

EXCLUDE MATCHES OFF