

Pengembangan Blok Fungsi Kendali PI-Fuzi pada IEC 61499

Dirga Eka Putra Lebukana¹, Awang Noor Indra Wardana², dan Nazrul Effendy³

Departemen Teknik Nuklir dan Teknik Fisika, Fakultas Teknik, Universitas Gadjah Mada
Jl. Grafika 2 Yogyakarta, 55281, Indonesia
dirga.e.p@mail.ugm.ac.id¹, awang.wardana@ugm.ac.id², nazrul@ugm.ac.id³

Abstract— Automation system in the form of automatic control is needed in one or several operating units in the process industry. Several control system algorithms have been widely implemented in the process industry, one of them is PI (Proportional and Integral) controller. This PI controller has a simple structure, but the adaptive ability of the controller is still not better for the controlled process. Therefore, the development of PI controller is needed to have a better adaptive ability to the process, in order to produce the responses that are also better and more robust. This research develops PI controller by adding the Fuzzy controller algorithm or also known as PI-Fuzzy controller. The PI-Fuzzy controller in this study is applied to the IEC 61499 Programmable Logic Controller (PLC) standard and to see its performance, the PI-Fuzzy controller based on IEC 61499 is tested and validated on an industrial scale process, namely coal mill. The test was carried out for three hours in real-time on the 4DIAC-IDE software, then the response results were compared with the response results of the PI controller. The PI-Fuzzy controller function block based on IEC 61499 made in this study showed good performance in controlling industrial-scale processes. This is demonstrated in testing and validation using a coal mill, with able to achieve the working range of each parameter. The mean value of coal flow parameter is 13.584 kg/s, coal mass accumulation is 2,196 kg, coal output temperature is 83.296 °C, coal moisture is 0.021, and coal fineness is 75.338 %.

Keywords— automation system, process industry, PI-Fuzzy controller, IEC 61499, coal mill

Abstrak— Sistem otomasi yang berupa kendali otomatis sangat diperlukan pada satu atau beberapa unit operasi pada industri proses. Beberapa algoritma sistem kendali telah banyak diimplementasikan pada industri proses, salah satunya adalah kendali Proporsional dan Integral (PI). Kendali PI ini mempunyai struktur yang sederhana, namun kemampuan adaptif yang dimiliki pengendali ini masih kurang baik terhadap proses yang dikendalikan. Oleh karena itu, pengembangan kendali PI dibutuhkan agar mempunyai kemampuan adaptif yang lebih baik terhadap proses, sehingga menghasilkan tanggapan respons yang juga lebih baik dan kokoh. Penelitian ini mengembangkan kendali PI dengan menambahkan algoritma kendali Fuzi atau dikenal juga dengan sebutan kendali PI-Fuzi. Kendali PI-Fuzi pada penelitian ini diterapkan pada standar *Programmable Logic Controller* (PLC) IEC 61499 dan untuk melihat kinerjanya, kendali PI-Fuzi berbasis IEC 61499 diujikan dan divalidasi pada proses berskala industri, yaitu *coal mill*. Pengujian dilakukan selama tiga jam secara waktu nyata pada perangkat lunak 4DIAC-IDE, lalu hasil responsnya dibandingkan dengan hasil respons pengendali PI. Blok fungsi kendali PI-Fuzi berbasis IEC 61499 yang dibuat pada penelitian ini menunjukkan kinerja yang baik dalam mengendalikan proses yang berskala industri. Hal ini ditunjukkan pada pengujian dan validasinya menggunakan *coal mill*, dengan mampu mencapai rentang kerja dari setiap parameter. Nilai rerata parameter aliran batu bara mencapai sebesar 13,584 kg/s, akumulasi massa batu bara sebesar 2.196 kg, suhu keluaran batu bara sebesar 83,296 °C, kelembaban batu bara sebesar 0,021, dan kehalusan batu bara sebesar 75,338 %.

Kata kunci— sistem otomasi, industri proses, kendali PI-Fuzi, IEC 61499, *coal mill*

I. PENDAHULUAN

Di era industri 4.0 ini, sistem otomasi merupakan hal yang krusial dan menonjol untuk dilakukan pengembangan lebih lanjut terlebih khususnya pada industri proses [1]. Industri proses biasanya terdiri atas beberapa unit operasi yang hasil akhirnya merupakan suatu produk jadi yang siap digunakan lebih lanjut [2]. Dalam menghasilkan produk, industri proses tak lepas dengan adanya proses kimiawi pada setiap unitnya,

sehingga dapat mempengaruhi nilai parameter-parameter pada unit operasi tersebut [3]. Nilai parameter ini hendaknya tetap terjaga pada batas ambang, agar keamanan operasi dan kualitas produk akhirnya dapat terjamin. Oleh karena itu, sistem otomasi yang berupa kendali otomatis sangat diperlukan pada satu atau beberapa unit operasi pada industri proses [3], sehingga terdiri dari atas satu atau beberapa kalang kendali. Hal ini juga disebut dengan *plant-wide control* [3].

Beberapa algoritma sistem kendali telah banyak digunakan pada industri proses, salah satunya adalah kendali PI (Proporsional dan Integral). Kendali PI ini mempunyai struktur yang sederhana, namun kemampuan adaptif yang dimiliki pengendali ini masih kurang baik terhadap proses yang dikendalikan [3]. Hal tersebut menjadi permasalahan mendasar dan pengembangan kendali PI dibutuhkan agar mempunyai kemampuan adaptif yang lebih baik terhadap proses, sehingga mampu menghasilkan tanggapan respons yang juga lebih baik dan kokoh. Penelitian ini mengembangkan kendali PI dengan menambahkan algoritma kendali Fuzi, sehingga disebut dengan kendali PI-Fuzi [4], [5]. Kendali Fuzi ini bertujuan untuk mengatur nilai konstanta proporsional dan integral yang mampu lebih beradaptasi terhadap proses yang dikendalikan.

Penerapan dan pengujian kendali PI-Fuzi pada beberapa sistem *plant* dengan basis yang berbeda-beda telah banyak dilakukan. Uji kinerja kendali PI-Fuzi banyak diterapkan pada perangkat lunak MATLAB/Simulink dengan menggunakan beberapa jenis *plant* yang berbeda-beda, antara lain *Automatic Generation Control* (AGC) [6], *grid interactive inverter* [7], sistem konversi energi angin [8], Motor *Brushless DC* (BLDC) [9], Generator Induksi [10], dan magnet permanen motor multi-sinkron [11]. Penerapan dan pengujian pengendali PI-Fuzi pada [6], [8]-[11] merupakan penelitian yang berskala simulasi, di mana hasil dari penelitian tersebut adalah pengendali PI-Fuzi mampu menunjukkan kinerja yang lebih baik dari pengendali PI yang juga ikut diujikan pada beberapa sistem *plant* tersebut [6], [8]-[11]. Lain halnya pada penerapan dan pengujian pengendali PI-Fuzi pada *grid interactive inverter* yang memvalidasi hasil simulasi dan hasil eksperimen [7]. Hasil simulasi dan eksperimen tersebut menunjukkan bahwa kendali PI-Fuzi mampu menghasilkan respons dinamis yang cepat dan mampu melacak arus referensi dengan *overshoot* yang rendah dan *settling time* yang singkat [7].

Selain menggunakan perangkat lunak MATLAB/Simulink, uji kinerja kendali PI-Fuzi juga pernah dilakukan pada perangkat lunak LabVIEW, dengan menggunakan *plant* kendaraan listrik *hybrid* [12]. Penelitian tersebut juga berskala simulasi dan hasilnya adalah pengendali PI-Fuzi mampu membuat respons kecepatan menjadi kokoh.

Hasil dari beberapa penelitian kendali PI-Fuzi yang diterapkan dan diujikan pada beberapa sistem *plant* [6]-[12], menjadi suatu hal menarik dan penting untuk melakukan suatu penelitian yang lebih lanjut dalam menghasilkan suatu inovasi terbaru di era industri 4.0 ini. Oleh karena itu, penelitian ini melakukan penerapan kendali PI-Fuzi pada standar *Programmable Logic Controller* (PLC) IEC 61499 karena pengujiannya dapat dilakukan secara waktu nyata, serta sangat cocok untuk menerapkan suatu sistem besar yang berskala industri [13], [14]. Dalam melihat kinerja kendali PI-Fuzi berbasis IEC 61499 ini, pengendali PI-Fuzi tersebut divalidasi

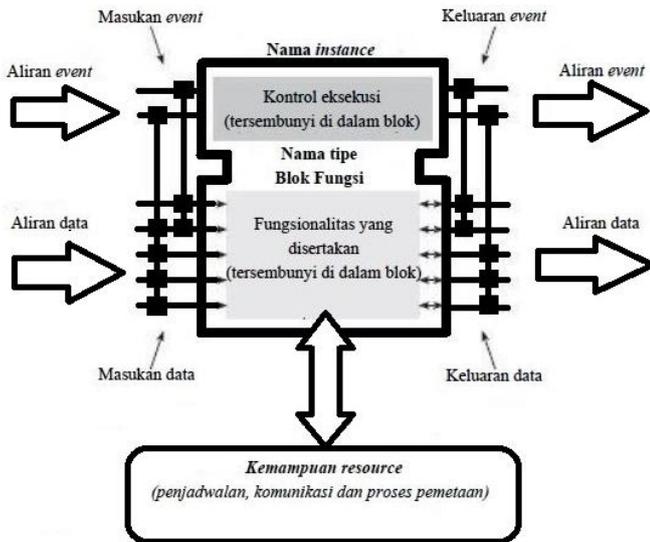
dan diujikan pada proses *coal mill* yang mengacu pada [15]. Penelitian ini bertujuan untuk membuat suatu pustaka baru pada salah satu perangkat lunak IEC 61499, yaitu 4DIAC-IDE yang nantinya dapat digunakan oleh para praktisi dalam merealisasikan sistem kendali yang berskala industri.

II. METODE

A. Pengendali PI-Fuzi Berbasis IEC 61499

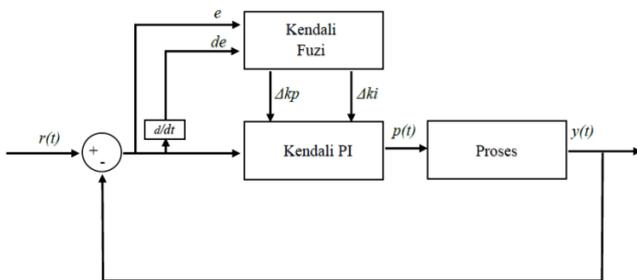
Awal mula lahirnya IEC 61499 terjadi pada awal tahun 1990, dengan pada saat itu komite teknis 65 dari *International Electrotechnical Commission* (IEC) memulai suatu proyek dalam mengembangkan model arsitektur umum untuk aplikasi modul perangkat lunak yang disebut juga dengan blok fungsi pada sistem distribusi *Industrial-Process Measurement and Control Systems* (IPMCS) [13]. Model yang mencakup IPMCS ini, memanfaatkan sistem "*Fieldbus*" seperti yang ditentukan dalam standar IEC 61158, serta memanfaatkan bahasa pemrograman yang didefinisikan dalam standar IEC 61131-3 untuk pengendali yang dapat diprogram. Proyek ini menghasilkan standar IEC 61499 untuk mode blok fungsi [13]. Blok fungsi dideskripsikan sebagai unit fungsional perangkat lunak yang mempunyai struktur data dan satu atau lebih algoritma yang dapat dimanipulasi. Definisi dari tipe blok fungsi ini memberikan deskripsi formal tentang struktur data dan algoritma yang akan diterapkan pada data dalam berbagai *instance*. Konsep tersebut bukanlah merupakan hal baru, tetapi berdasarkan praktik di industri pada umumnya blok fungsi kendali diterapkan dan dapat digunakan kembali dalam berbagai bentuk.

Karakteristik utama dari blok fungsi IEC 61499 terdapat pada Gambar 1. Pada gambar tersebut menunjukkan bahwa blok fungsi IEC 61499 terdiri atas dua bagian utama, yaitu bagian atas dan bagian bawah [13]. Bagian atas blok fungsi disebut dengan bagian "Kontrol Eksekusi", berisi definisi untuk memetakan *event* ke fungsionalitas yang disertakan. Artinya, bagian atas berperan dalam menentukan fungsionalitas mana yang didefinisikan dalam bagian bawah yang dipicu pada saat kedatangan *event* di kontrol eksekusi dan kapan keluaran *event* dipicu. Standar menyebut hubungan sebab-akibat antara masukan *event*, keluaran *event* dan eksekusi fungsionalitas yang disertakan. Standar mendefinisikan sarana untuk memetakan hubungan antara *event* yang tiba pada masukan, eksekusi fungsionalitas yang disertakan dan pemicu keluaran *event* yang tergantung pada jenis blok fungsi. Bagian bawah blok fungsi berisi fungsionalitas yang disertakan dengan kemungkinan data internal ikut juga disembunyikan di dalam blok fungsi. Blok fungsi adalah jenis komponen perangkat lunak yang jika dirancang dengan baik, seharusnya tidak ada kebutuhan bagi pengguna untuk memiliki pemahaman rinci tentang desain internalnya.



Gambar 1. Karakteristik blok fungsi [13]

Komponen utama dalam sistem kalang tertutup atau *closed-loop* terdiri atas pengendali dan proses/plant. Kendali PI yang ditunjukkan pada (1) hingga (3) merupakan suatu pengendali yang masih banyak digunakan di industri proses karena strukturnya yang sederhana [5], [16]. Kendali PI mempunyai dua konstanta, yaitu konstanta proporsional (kp) dan konstanta integral (ki) yang disetel sedemikian rupa hingga menghasilkan tanggapan respons yang diinginkan dari sistem kalang tertutup [3] (Gambar 2). Meskipun masih banyak digunakan di industri proses, kendali PI mempunyai kemampuan adaptif terhadap proses yang masih kurang [3]. Perlunya pengembangan pengendali PI dibutuhkan dan pada penelitian ini kendali PI dikembangkan dengan menambahkan algoritma kendali Fuzi, sehingga disebut dengan kendali PI-Fuzi. Kendali Fuzi ini bertujuan untuk mengatur nilai konstanta proporsional dan integral yang mampu lebih beradaptasi terhadap proses yang dikendalikan, sehingga menghasilkan tanggapan respons yang jauh lebih baik dan kokoh.



Gambar 2. Diagram blok sistem *closed loop* dengan kendali PI-Fuzi [5], [16], [17]

$$p(t) = kp (e(t) + \frac{1}{T_i} \int_0^t e(t) dt) \quad (1)$$

$$p(t) = kp * e(t) + \frac{kp}{T_i} \int_0^t e(t) dt \quad (2)$$

$$p(t) = kp * e(t) + ki \int_0^t e(t) dt \quad (3)$$

dengan:

$$e(t) = y(t) - r(t) \quad (4)$$

$$\frac{de(t)}{dt} = \frac{e(t) - e(t-1)}{\Delta t} \quad (5)$$

keterangan:

p = sinyal keluaran pengendali

kp = konstanta proporsional

ki = konstanta integral

T_i = *time integral* (s)

y = hasil pengukuran/nilai keluaran sistem *closed-loop*

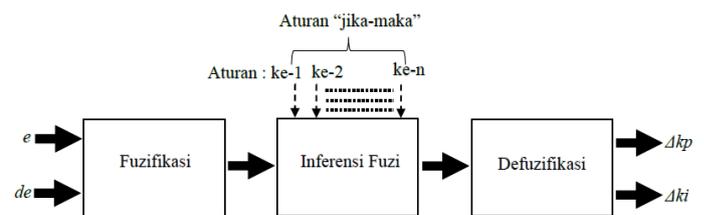
r = nilai *set-point*

Δt = perubahan waktu respons (s)

Tahapan kerja dari kendali PI-Fuzi ini adalah nilai *error* (e) yang merupakan selisih dari *set-point* dan hasil pengukuran, ditunjukkan pada (4) [3] dan turunan pertama *error* yang merupakan selisih dari *error* sekarang dan sebelumnya, ditunjukkan pada (5) [3] awal mulanya akan di-fuzifikasi berdasarkan variabel linguistik dari e dan de pada fungsi keanggotaannya, yang hasilnya berupa besar derajat kebenaran dari e dan de . Setelah itu, nilai derajat kebenaran dari e dan de akan diteruskan ke tahap inferensi fuzi dan di-defuzifikasi (Gambar 3), dengan pada tahap ini nilai derajat kebenarannya diagregasi dengan menggunakan *min* (\wedge) atau *max* (\vee), berdasarkan pada aturan “jika-maka” yang telah disusun. Selanjutnya, nilai hasil agregasi tersebut akan dikalikan dengan nilai variabel linguistik dari pengendali, lalu dijumlahkan dan dibagi dengan nilai hasil penjumlahan agregasi, sehingga menghasilkan nilai keluaran akhir dari pengendali, yaitu nilai perubahan dari konstanta proporsional dan integral (Δkp dan Δki) yang selanjutnya dijumlahkan dengan nilai awal dari kp dan ki . Tahapan ini ditunjukkan pada (6) dan (7) [5].

$$kp_{akhir} = \Delta kp + kp_{awal} \quad (6)$$

$$ki_{akhir} = \Delta ki + ki_{awal} \quad (7)$$



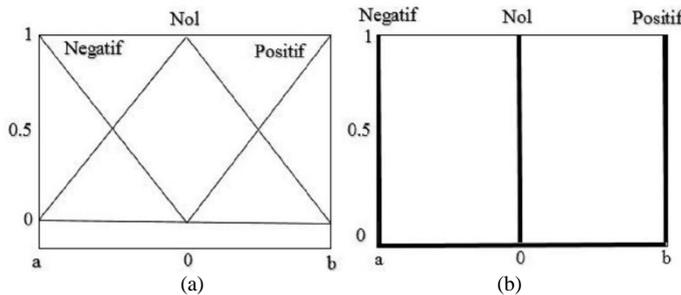
Gambar 3. Diagram blok proses Fuzi [18]

Terdapat dua tipe algoritma kendali Fuzi hingga saat ini, yaitu tipe Mamdani dan Sugeno [18]. Pada penelitian ini, tipe Sugeno diimplementasikan pada standar IEC 61499 dengan menggunakan fitur *Structure Text* (ST). Fungsi keanggotaan dari e dan de yang digunakan pada penelitian ini adalah tipe *triangular* (Gambar 4) yang dimodelkan dengan persamaan garis lurus melalui dua titik sesuai dengan (8). Selanjutnya, variabel linguistik dari e dan de terdiri dari “Negatif”, “NoI”, dan “Positif”, dengan nilai cakupan *universe* $[a,b]$. Nilai a merupakan batas bawah *universe*, sedangkan nilai b merupakan batas atas *universe*. Kedua nilai ini diatur

sedemikian rupa oleh operator, yang juga disesuaikan dengan pengetahuan dari proses.

Fungsi keanggotaan dari Δkp dan Δki yang digunakan pada penelitian ini adalah tipe *singleton* (Gambar 4) yang variabel linguistiknya juga terdiri dari “Negatif”, “Nol”, dan “Positif”, dengan nilai cakupan *universe* $[a,b]$. Nilai a merupakan persentase perubahan batas bawah, sedangkan nilai b merupakan persentase perubahan batas atas. Kedua nilai persentase perubahan ini, diatur sedemikian rupa oleh operator hingga mencapai tanggapan respons *closed-loop* yang diinginkan.

$$\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1} \quad (8)$$



Gambar 4. Fungsi keanggotaan e dan de (a) dan Δkp , Δki (b)

Aturan untuk Δkp dan Δki , serta persamaan defuzifikasi [18] yang digunakan pada penelitian ini dapat dilihat pada Tabel I dan (9).

TABEL I. ATURAN ΔKP DAN ΔKI

e	de	$\Delta kp/\Delta ki$
Negatif	Negatif	Positif
Negatif	Nol	Positif
Negatif	Positif	Negatif
Nol	Negatif	Negatif
Nol	Nol	Nol
Nol	Positif	Nol
Positif	Negatif	Nol
Positif	Nol	Positif
Positif	Positif	Positif

$$u = \frac{a_1*s_1 + a_2*s_2 + \dots + a_n*s_n}{a_1 + a_2 + \dots + a_n} \quad (9)$$

keterangan:

u = keluaran sinyal pengendali (Δkp dan Δki)

a = nilai agregasi

s = nilai *universe* Δkp dan Δki

Pada penelitian ini, aturan dari Δkp and Δki memiliki tipe aturan yang sama karena kedua konstanta ini memiliki karakteristik yang hampir sama [3]. Pertama, ketika kondisi nilai e dan de negatif dan negatif, negatif dan nol, respons membutuhkan *rise time* yang lebih cepat, sehingga nilai Δkp dan Δki harus positif. Selanjutnya, ketika nilai e dan de bernilai negatif dan positif, nol dan negatif, respons perlu diperkecil dan mengurangi *overshoot*, sehingga nilai Δkp dan Δki harus negatif. Terakhir, ketika nilai e dan de adalah nol dan nol, nol dan positif, positif dan negatif, positif dan nol, positif dan positif, respons perlu memperoleh *settling time*

yang lebih cepat, sehingga nilai Δkp dan Δki perlu ditingkatkan dengan nol dan kemudian menjadi nilai positif.

B. Validasi, Pengujian, dan Analisis Data

Dalam melihat kinerja dari blok fungsi kendali PI-Fuzi berbasis IEC 61499 yang dibuat menggunakan perangkat lunak 4DIAC-IDE 1.8.4., blok fungsi kendali PI-Fuzi berbasis IEC 61499 ini akan divalidasi dan diujikan pada proses berskala industri, yaitu *coal mill* yang modelnya mengacu pada [15]. Kemudian dalam analisis data, beberapa tanggapan respons keluaran dari parameter atau variabel proses dari *coal mill* yang telah diukur pada penelitian ini, akan dibandingkan dengan hasil dari pengendali PI, dengan menghitung parameter tertentu.

III. HASIL DAN PEMBAHASAN

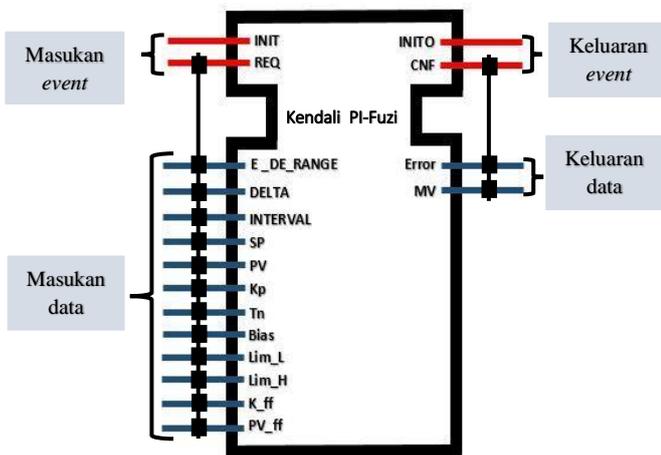
A. Blok Fungsi Kendali PI-Fuzi Berbasis IEC 61499

Penelitian ini bertujuan untuk menghasilkan suatu pustaka atau blok fungsi baru pada perangkat lunak IEC 61499, yaitu 4DIAC-IDE yang nantinya dapat digunakan oleh para praktisi dalam merealisasikan sistem kendali yang berskala industri, dengan tentunya sangat berbeda pada penelitian pengendali PI-Fuzi yang telah dilakukan [6]–[12]. Blok fungsi terdiri atas dua tipe, yaitu tipe *basic* dan *composite* [13]. Blok fungsi tipe *basic* merupakan blok fungsi yang dibentuk melalui suatu algoritma bermode *Structure Text* (ST) (Gambar 8) dan langkah-langkah eksekusi dari masing-masing algoritma pada blok fungsi dimodelkan berdasarkan *Execution Control Chart* (ECC) (Gambar 7). Sedangkan Blok fungsi tipe *composite* merupakan blok fungsi yang terdiri atas beberapa jaringan blok fungsi tipe *basic* dan tersusun secara sistematis.

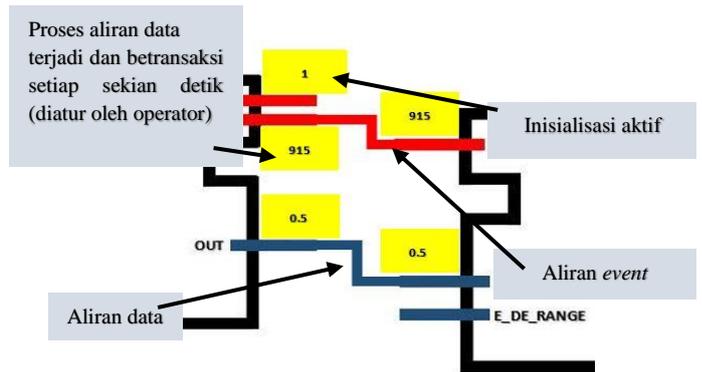
Blok fungsi kendali PI-Fuzi berbasis IEC 61499 yang dibuat pada penelitian ini merupakan blok fungsi yang bertipe *composite*, dengan dalam pembuatannya menggunakan perangkat lunak 4DIAC-IDE versi 1.8.4. Untuk lebih jelasnya, Gambar 5 menunjukkan wujud dari blok fungsi kendali PI-Fuzi berbasis IEC 61499 beserta dengan penjelasannya.

Pada Gambar 5 dan 6, blok fungsi kendali PI-Fuzi berbasis IEC 61499 ini mempunyai terminal *event* yang ditunjukkan dengan warna merah dan terminal data yang ditunjukkan dengan warna biru. Terminal *event* mempunyai 2 masukan dan 2 keluaran. Pada bagian masukan, terdapat terminal “INIT” yang berfungsi untuk menginisialisasi atau mengadakan suatu layanan tertentu dari blok fungsi agar proses transmisi data dapat dilakukan. Sedangkan untuk terminal “REQ” berfungsi untuk menginisialisasi atau mengadakan sebuah permintaan untuk memperoleh data dari agen eksternal. Selanjutnya pada bagian keluaran, terdapat terminal “INITO” yang berfungsi menandakan bahwa layanan antarmuka blok fungsi telah menyelesaikan inisialisasi, dengan hasil terminal ini dapat diteruskan ke terminal masukan “INIT” yang terdapat pada blok fungsi lainnya. Sedangkan untuk terminal “CNF” berfungsi untuk mengkonfirmasi *event* yang keluar ketika blok fungsi telah menyelesaikan pengiriman permintaan ke agen eksternal.

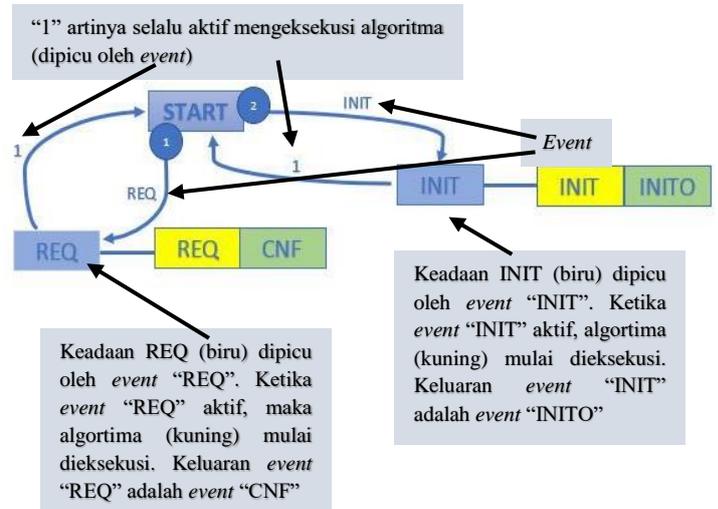
Terminal data mempunyai 12 masukan dan 2 keluaran. Bagian masukan terminal variabel pengendali, nilainya diset oleh operator dan dipicu oleh masukan *event* "REQ". Variabel tersebut terdiri dari "E_DE_RANGE", "DELTA", "INTERVAL", "SP", "PV", "Kp", "Tn", "Bias", "Lim_L", "Lim_H", "K_ff", dan "PV_ff". "E_DE_RANGE" merupakan nilai batas bawah dan batas atas *universe* dari *error* (*e*) dan turunan pertama dari *error* (*de*). Nilai variabel *universe* *e* dan *de* pada blok fungsi yang dibuat ini diatur dengan nilai tunggal. Misalnya jika nilai masukan diatur sebesar 10, maka batas bawah *e* dan *de* adalah -10 dan batas atas *e* dan *de* adalah 10. "DELTA" merupakan nilai persentase perubahan batas bawah dan batas atas dari *kp* dan *ki* (Δkp dan Δki) yang nilainya diset juga dengan nilai tunggal seperti pada terminal data "E_DE_RANGE". "INTERVAL" merupakan nilai perubahan waktu respons dari kalang kendali dalam satuan detik (s). "SP" merupakan nilai referensi parameter yang dikendalikan (*set-point*). "PV" merupakan nilai variabel proses (*y*) atau hasil pengukuran parameter dari kalang kendali. "Kp" dan "Tn" merupakan konstanta proporsional dan waktu integrasi dalam satuan detik (s). "Bias" merupakan nilai tambahan pengendali agar mampu mencapai keadaan stabilnya (*steady-state*). "Lim_L" dan "Lim_H" merupakan nilai batas bawah dan batas atas dari keluaran pengendali (*p*) atau *Manipulated Variable* (MV). "K_ff" dan "PV_ff" merupakan nilai *gain* dan variabel proses yang digunakan untuk kasus sistem umpan maju (*feedforward*). Selanjutnya pada bagian keluaran terminal variabel pengendali, terdiri dari "Error" dan "MV", serta keduanya dipicu oleh keluaran *event* "CNF". "Error" merupakan nilai selisih dari "SP" dan "PV" dari terminal masukan, sedangkan "MV" merupakan singkatan dari *Manipulated Variable* atau nilai keluaran dari pengendali.



Gambar 5. Blok fungsi kendali PI-Fuzi berbasis IEC 61499



Gambar 6. Aliran event dan data pada blok fungsi kendali PI-Fuzi berbasis IEC 61499



Gambar 7. Execution Control Chart (ECC) pada blok fungsi kendali PI-Fuzi berbasis IEC 61499

```

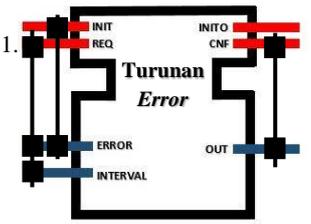
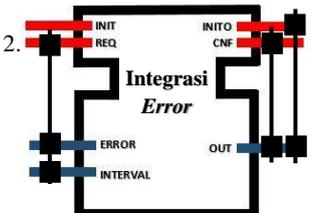
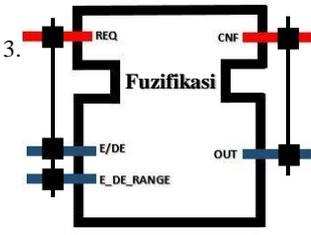
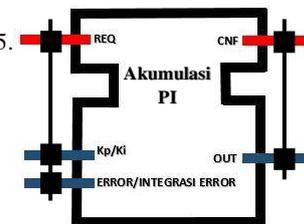
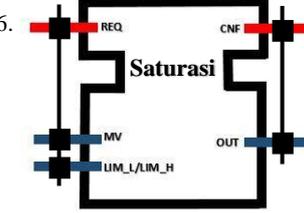
VAR
  occurred_error : REAL;
END_VAR;
occurred_error:=error;
a:=-e_range;
IF (occurred_error>=a) AND (occurred_error<=0) THEN
  (*Equation of N*)
  degree_of_membership := occurred_error/a;
  ELSE IF (occurred_error<a) THEN
  degree_of_membership := 1;
  ELSE
  degree_of_membership := 0.0;
  END_IF;

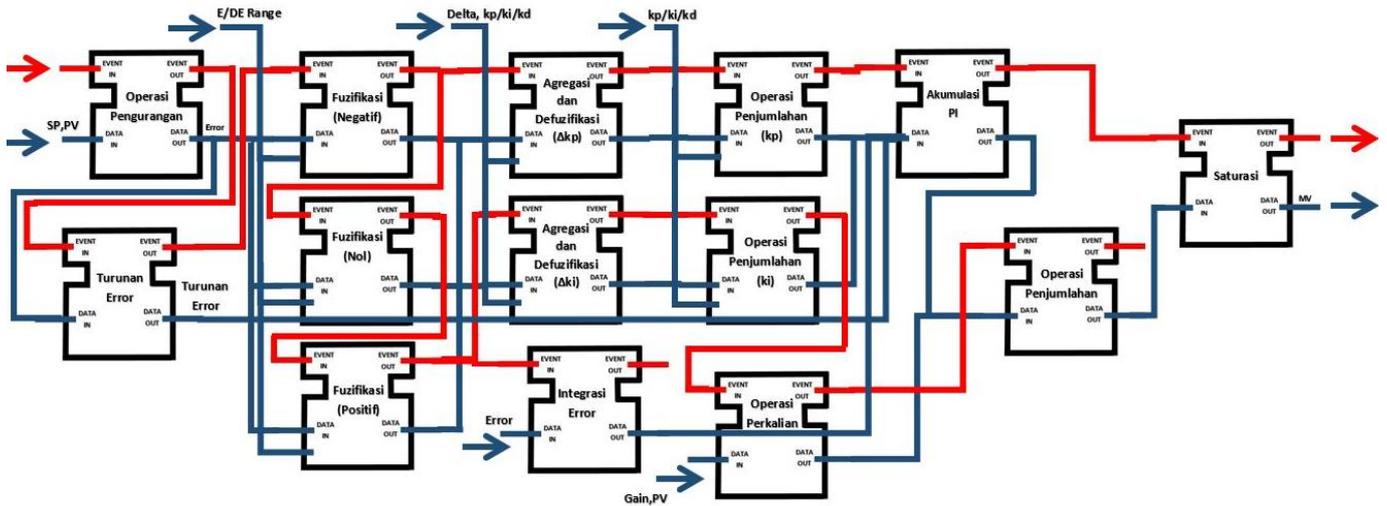
```

Gambar 8. Structure Text (ST) pada blok fungsi tipe basic yang digunakan pada composite blok fungsi kendali PI-Fuzi berbasis IEC 61499 di 4DIAC-IDE

Dalam jaringan *composite* dari blok fungsi kendali PI-Fuzi berbasis IEC 61499 terdapat beberapa blok fungsi tipe *basic* yang dibuat menggunakan fitur algoritma *structure text*. Tabel II menunjukkan daftar dari blok fungsi tipe *basic* yang digunakan pada *composite* blok fungsi kendali PI-Fuzi berbasis IEC 61499. Gambar 9 menunjukkan arsitektur jaringan dari blok fungsi tipe *basic* yang digunakan pada *composite* blok fungsi kendali PI-Fuzi berbasis IEC 61499.

TABEL II. DAFTAR BLOK FUNGSI TIPE BASIC PADA COMPOSITE BLOK FUNGSI KENDALI PI-FUZI BERBASIS IEC 61499

Blok Fungsi Tipe Basic	Algoritma
<p>1.</p> 	<p>Algoritma 1 Blok Fungsi Turunan Error</p> <p>1: Masukan: ERROR, INTERVAL 2: Inisialisasi: $ERROR(t-1) = 0$ 3: $OUT = (ERROR(t) - ERROR(t-1)) / INTERVAL$ 4: $ERROR(t-1) = ERROR(t)$ 5: Keluaran: OUT</p>
<p>2.</p> 	<p>Algoritma 2 Blok Fungsi Integrasi Error</p> <p>1: Masukan: ERROR, INTERVAL 2: Inisialisasi: $OUT = 0$ 3: $OUT = (OUT + ERROR(t)) * INTERVAL$ 4: Keluaran: OUT</p>
<p>3.</p> 	<p>Algoritma 3 Blok Fungsi Fuzifikasi</p> <p>1: Masukan: E/DE, E_DE_RANGE Fungsi Keanggotaan Negatif 2: <i>if</i> $(E/DE \geq a)$ and $(E/DE \leq 0)$ <i>do</i> 3: $OUT = \text{Persamaan (8)}$ 4: <i>if else</i> $(E < a)$ <i>do</i> 5: $OUT = 1$ 6: <i>else</i> 7: $OUT = 0$ 8: <i>end if</i> 9: Keluaran: OUT</p> <p>Fungsi Keanggotaan Nol 1: Masukan: E/DE, E_DE_RANGE 2: <i>if</i> $(E/DE \geq a)$ and $(E/DE \leq 0)$ <i>do</i> 3: $OUT = \text{Persamaan (8)}$ 4: <i>else if</i> $(E/DE \geq 0)$ and $(E/DE \leq -b)$ <i>do</i> 5: $OUT = \text{Persamaan (8)}$ 6: <i>else</i></p>
<p>7.</p> 	<p>7: $OUT = 0$ 8: <i>end if</i> 9: Keluaran: OUT</p> <p>Fungsi Keanggotaan Positif 1: Masukan: E/DE, E_DE_RANGE 2: <i>if</i> $(E/DE \geq 0)$ and $(E/DE \leq b)$ <i>do</i> 3: $OUT = \text{Persamaan (8)}$ 4: <i>else if</i> $(E/DE > b)$ <i>do</i> 5: $OUT = 1$ 6: <i>else</i> 7: $OUT = 0$ 8: <i>end if</i> 9: Keluaran: OUT</p>
<p>4.</p> 	<p>Algoritma 4 Blok Fungsi Agregasi dan Defuzifikasi</p> <p>1: Masukan: Kp, Ki, DELTA, DERAJAT KEBENARAN 2: Proses agregasi 3: Perkalian hasil agregasi dan nilai <i>universe</i> yang mengacu pada aturan di Tabel I 4: Penjumlahan nilai agregasi 5: Penjumlahan hasil kali agregasi dan <i>universe</i> 6: $OUT = \text{Proses defuzikasi } \Delta k_p \text{ dan } \Delta k_i \text{ (Persamaan (9))}$ 7: Keluaran: OUT</p>
<p>5.</p> 	<p>Algoritma 5 Blok Fungsi Akumulasi PI</p> <p>1: Masukan: Kp, Ki, ERROR, INTEGRASI ERROR 2: $OUT = \text{Persamaan (3)}$ 3: Keluaran: OUT</p>
<p>6.</p> 	<p>Algoritma 6 Blok Fungsi Saturasi</p> <p>1: Masukan: MV, LIM_L, LIM_H 2: <i>if</i> $MV < LIM_L$ <i>do</i> 3: $OUT = LIM_L$ 4: <i>else if</i> $MV > LIM_H$ <i>do</i> 5: $OUT = LIM_H$ 6: <i>else</i> 7: $OUT = MV$ 8: <i>end if</i> 9: Keluaran: OUT</p>



Gambar 9. Arsitektur jaringan dari blok fungsi tipe *basic* yang digunakan pada *composite* blok fungsi kendali PI-Fuzi berbasis IEC 61499

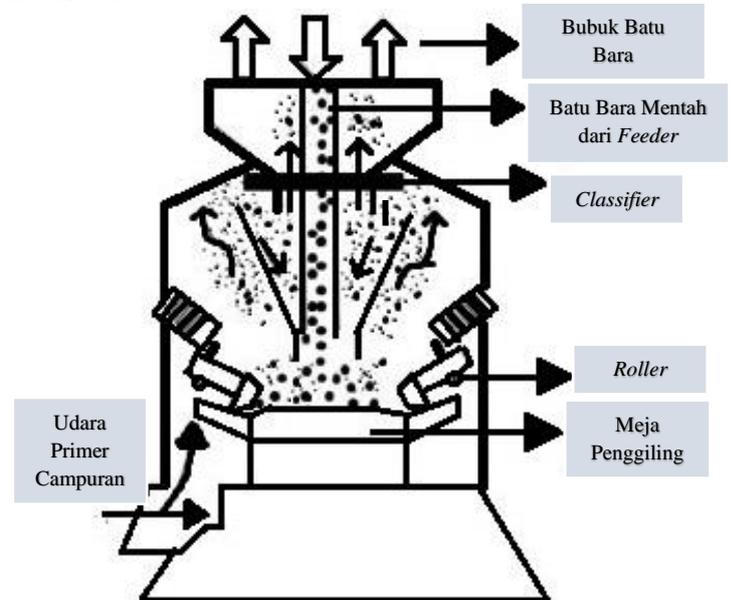
B. Validasi dan Pengujian Blok Fungsi Kendali PI-Fuzi Berbasis IEC 61499

Coal mill atau pabrik batu bara merupakan salah satu komponen penting pada sistem pembangkit yang berpengaruh pada efisiensi dan kehandalan dari sebuah pembangkit listrik tenaga batu bara [19]. Pabrik batu bara menggiling batu bara mentah ke dalam bentuk bubuk dan udara panas yang memasuki batu bara mengeringkan dan membawa batu bara bubuk ke tungku pembakaran. Pabrik batu bara dikategorikan menjadi tiga, dengan kategori ini dibagi berdasarkan kecepatan, antara lain pabrik dengan kecepatan rendah, kecepatan sedang, dan pabrik dengan kecepatan tinggi [20]. Di antara ketiga kategori tersebut, pabrik dengan kecepatan sedang adalah salah satu jenis pabrik batu bara yang paling populer pada sistem pembangkit listrik tenaga uap batu bara dan dalam penelitian ini digunakan untuk validasi dan pengujian blok fungsi kendali PI-Fuzi berbasis IEC 61499. Pabrik batu bara dengan kecepatan sedang memiliki bentuk meja penggiling yang seperti mangkuk (*bowl mill*) (Gambar 10). Batu bara mentah dari *feeder* awal mulanya memasuki gilingan melalui saluran ke meja yang berputar dengan kecepatan konstan.

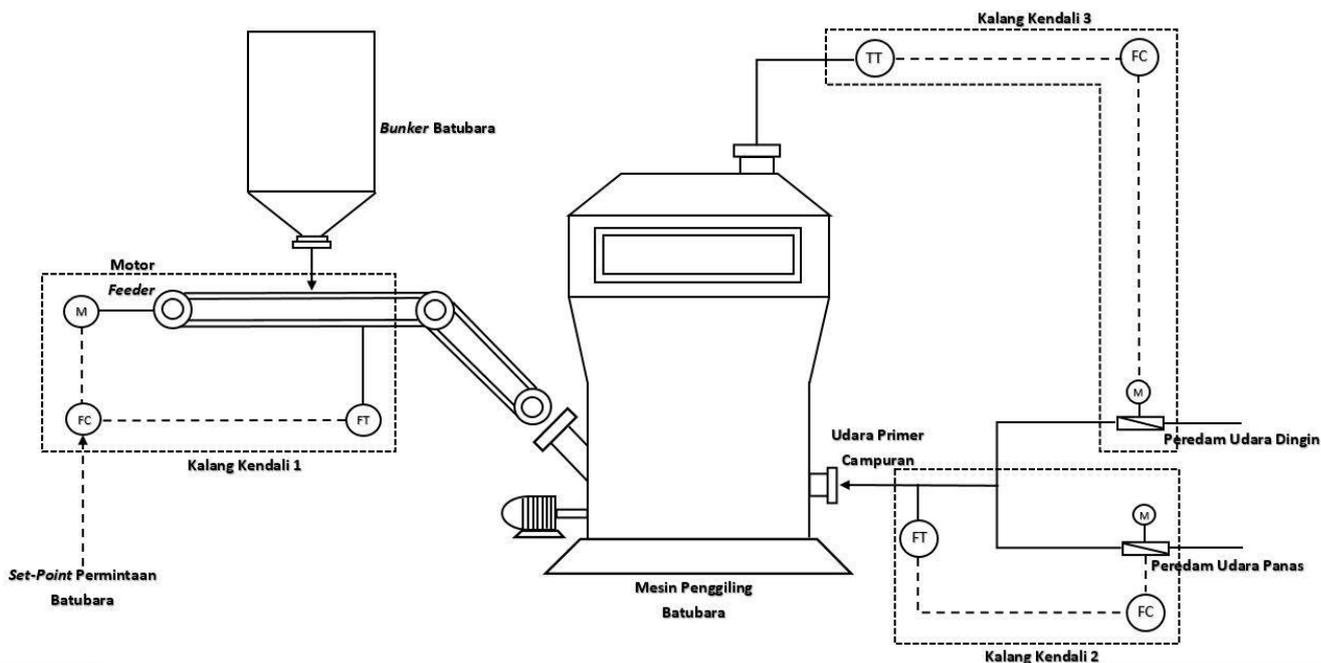
Batu bara ini bergerak ke luar karena adanya gaya sentrifugal dan hancur, lalu jatuh ke bawah *roller*. Batu bara *ground* diambil dan dikeringkan dengan udara primer campuran berkecepatan tinggi. Partikel yang lebih berat jatuh kembali ke meja untuk digiling lebih lanjut, sementara partikel yang lebih ringan diteruskan ke *classifier*. Partikel yang lebih ringan diteruskan keluar ke *burner* sebagai bahan bakar dalam bentuk bubuk dan partikel yang lebih berat jatuh kembali ke meja penggiling untuk diproses lebih lanjut.

Produk akhir dari *coal mill* adalah berupa batu bara halus dari proses pelumatan atau *pulverised*, setelah itu batu bara halus ini akan diproses lebih lanjut pada unit lainnya. Penelitian ini memvalidasi dan menguji kendali PI-Fuzi berbasis IEC 61499 pada *coal mill* yang mengacu pada [15] dan kalang kendali yang mengacu pada [20]. Pada [20]

terdapat tiga kalang kendali yang digunakan, yaitu kendali aliran batu bara, kendali aliran udara, dan kendali suhu keluaran. Kendali aliran batu bara digunakan untuk mengendalikan aliran pada batu bara yang akan melalui *feeder* dengan mengatur kecepatan pada *feeder* tersebut. Kendali aliran udara digunakan dengan mengatur peredam udara panas dan kendali suhu keluaran digunakan dengan mengatur peredam udara dingin. Gambar 11 menunjukkan diagram *piping and instrumentation (P&ID)* dari *coal mill* yang digunakan dalam penelitian ini. Skema kendali atau pengujian blok fungsi kendali PI-Fuzi berbasis IEC 61499 pada *coal mill* dilakukan dengan mengatur beberapa variasi *set-point* atau nilai permintaan batu bara yang akan diproses dalam mesin penggiling.



Gambar 10. Struktur fisik dari *Bowl Mill* [19]



Gambar 11. Diagram Piping and Instrumentation Coal Mill [15]

Nilai permintaan batu bara tersebut diatur pada kalang kendali pertama. Terdapat 6 variasi nilai permintaan batu bara yang digunakan pada penelitian ini, dengan rentangnya berada pada 10 s.d. 15 kg/s. Pengujian blok fungsi kendali PI-Fuzi berbasis IEC 61499 pada *coal mill* yang mengacu pada [15], dioperasikan selama tiga jam secara waktu nyata pada perangkat lunak 4DIAC-IDE versi 1.8.4. Beberapa parameter atau variabel proses dari *coal mill* dianalisis, dengan parameter tersebut adalah aliran batu bara, akumulasi massa, suhu keluaran, kelembaban, dan kehalusan

Dengan adanya beberapa variasi permintaan batu bara atau *set-point* yang diatur oleh operator pada penelitian ini, nilai perubahan permintaan batu bara tersebut dapat diikuti dengan perubahan aliran batu bara yang masuk ke dalam mesin penggiling. Terlihat pada hasil analisis aliran batu bara, aliran batu bara dengan menggunakan kendali PI-Fuzi terlihat lebih baik dari hasil aliran batu bara dengan menggunakan kendali PI (Tabel IV). Dengan menggunakan kendali PI-Fuzi, nilai permintaan batu bara yang masuk ke dalam mesin penggiling dapat beberapa kali terpenuhi pada setiap perubahan yang dilakukan oleh operator (Tabel III). Hanya ketika permintaan batu bara diset sebesar 10 dan 11,5 kg/s, kendali PI-Fuzi belum mampu memenuhi permintaan tersebut (Tabel III). Sedangkan dengan menggunakan kendali PI, terlihat bahwa nilai permintaan batu bara yang masuk ke dalam mesin penggiling hanya satu yang dapat terpenuhi, yaitu pada saat diset 11,5 kg/s (Tabel III).

Dari kedua hasil tersebut, rentang nilai aliran batu bara yang mampu dihasilkan kendali PI-Fuzi adalah sebesar 12,240 s.d. 15,101 kg/s dengan nilai rerata yang diperoleh sebesar 13,584 kg/s (Tabel IV). Kemudian untuk rentang nilai aliran batu bara yang mampu dihasilkan kendali PI adalah sebesar 11,268 s.d. 12,279 kg/s dengan nilai rerata yang diperoleh sebesar 11,582 kg/s (Tabel IV). Mengacu pada [19] kedua

hasil dari algoritma pengendali yang digunakan sama-sama mampu mencapai rentang kerja dari parameter aliran batu bara tersebut, namun kendali PI-Fuzi mampu menunjukkan hasil yang lebih efisien.

TABEL III. HASIL ANALISIS PERMINTAAN BATU BARA

Pengendali	Permintaan Batu bara (<i>Set-Point</i>) (kg/s)					
	10	11,5	12,5	13,5	14,5	15
PI	BT	T	BT	BT	BT	BT
PI-Fuzi	BT	BT	T	T	T	T

keterangan:
 T: Terpenuhi
 BT: Belum Terpenuhi

TABEL IV. HASIL ANALISIS ALIRAN BATU BARA

Pengendali	Nilai Aliran (kg/s)		Rerata (kg/s)
	Nilai Maksimum	Nilai Minimum	
PI	12,279	11,268	11,852
PI-Fuzi	15,101	12,240	13,584

Perubahan nilai aliran batu bara yang masuk ke dalam mesin penggiling, dapat diikuti juga dengan berubahnya nilai parameter akumulasi massa batu bara. Berdasarkan nilai reratanya, ketika nilai permintaan batu bara dinaikkan, nilai akumulasi massa juga ikut meningkat. Begitu pula ketika nilai permintaan batu bara diturunkan, nilai akumulasi massa juga ikut menurun (Tabel V). Hasil analisis akumulasi massa batu bara (Tabel V) menunjukkan dengan menggunakan kendali PI-Fuzi, hasil akumulasi massa terlihat lebih baik dari hasil akumulasi massa batu bara dengan menggunakan kendali PI. Dengan menggunakan kendali PI-Fuzi, nilai akumulasi massa batu bara dapat meningkat (nilai rerata). Rentang nilai akumulasi massa yang mampu dihasilkan kendali PI-Fuzi

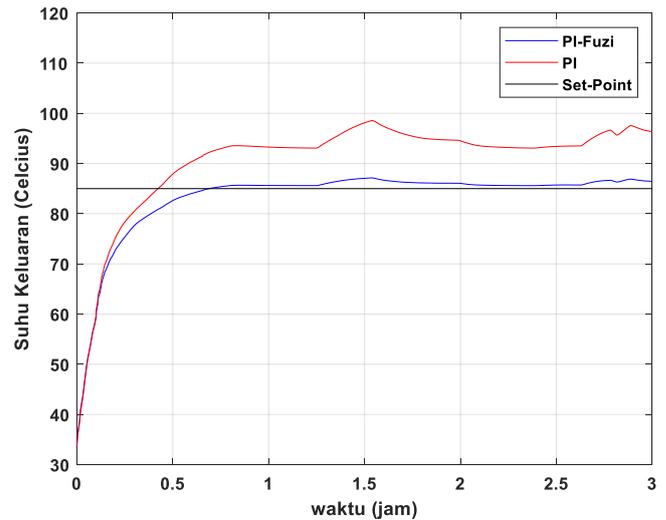
adalah sebesar 452,421 s.d. 2821 kg dengan nilai rerata yang diperoleh sebesar 2196 kg (Tabel V). Kemudian untuk rentang nilai aliran batu bara yang mampu dihasilkan kendali PI adalah sebesar 404,411 s.d. 2001,3 kg dengan nilai rerata yang diperoleh sebesar 1899 kg (Tabel V). Mengacu pada [19] kedua hasil dari algoritma pengendali yang digunakan sama-sama mampu mencapai rentang kerja dari parameter akumulasi massa batu bara tersebut, namun kendali PI-Fuzi mampu menunjukkan hasil yang lebih efisien dan mampu mengompensasi lambatnya *rise time* yang dihasilkan oleh kendali PI-Fuzi daripada kendali PI (Tabel V).

TABEL V. HASIL ANALISIS AKUMULASI MASSA BATU BARA

Pengendali	Nilai Akumulasi Massa (kg)		Rise Time (jam)	Rerata (kg)
	Nilai Maksimum	Nilai Minimum		
PI	2001,3	404,411	±0,524	1899
PI-Fuzi	2821	452,421	±0,597	2196

Suhu keluaran batu bara menunjukkan pengaruh pada parameter kelembaban dan kehalusan batu bara. Ketika nilai suhu keluaran batu bara meningkat, maka nilai kelembaban batu bara menurun. Hal ini ditunjukkan pada hasil rerata kelembaban batu bara bahwa dengan menggunakan kendali PI-Fuzi, nilai kelembaban batubara terlihat lebih meningkat dan lebih baik dari hasil nilai kelembaban batu bara dengan menggunakan kendali PI (Tabel VII). Sedangkan untuk pada parameter kehalusan batu bara, ketika nilai suhu keluaran batu bara meningkat, nilai kehalusan batu bara pun ikut meningkat. Hal ini ditunjukkan pada hasil rerata kehalusan batu bara bahwa dengan menggunakan kendali PI nilai kehalusan batu bara terlihat lebih meningkat dan lebih baik dari hasil rerata kehalusan batu bara dengan menggunakan kendali PI-Fuzi (Tabel VIII).

Hasil analisis suhu keluaran batu bara menunjukkan bahwa suhu keluaran batu bara dengan menggunakan kendali PI-Fuzi terlihat lebih baik dari hasil suhu keluaran batu bara dengan menggunakan kendali PI (Tabel VI). Dengan menggunakan kendali PI-Fuzi, nilai suhu keluaran batu bara dapat menurun dan mampu mendekati *set-point* (85°C) (Gambar 12). Rentang nilai suhu keluaran yang mampu dihasilkan kendali PI-Fuzi adalah sebesar 33,637 s.d. 87,136 °C dengan nilai rerata yang diperoleh sebesar 83,296 °C (Tabel VI). Kemudian untuk rentang nilai suhu keluaran batu bara yang mampu dihasilkan kendali PI adalah sebesar 33,722 s.d. 98,560 °C dengan nilai rerata yang diperoleh sebesar 90,702 °C (Tabel VI). Mengacu pada [19] kedua hasil dari algoritma pengendali yang digunakan sama-sama mampu mencapai rentang kerja dari parameter suhu keluaran batu bara tersebut, namun kendali PI-Fuzi mampu menunjukkan hasil yang lebih efisien dan mampu mengompensasi lambatnya *rise time* yang dihasilkan oleh kendali PI-Fuzi daripada kendali PI (Tabel VI). Hasil rerata suhu keluaran batu bara yang diperoleh kendali PI-Fuzi, terlihat bahwa kendali PI-Fuzi dapat menjaga proses evaporasi di dalam mesin penggiling dengan menghasilkan suhu keluaran yang tidak begitu tinggi dan tidak begitu rendah [19].



Gambar 12. Respon suhu keluaran batu bara

TABEL VI. HASIL ANALISIS SUHU KELUARAN BATU BARA

Pengendali	Nilai Suhu Keluaran (°C)		Rise Time (jam)	Rerata (°C)
	Nilai Maksimum	Nilai Minimum		
PI	98,560	33,722	±0,422	90,702
PI-Fuzi	87,136	33,637	±0,687	83,296

Pada hasil analisis kelembaban batubara (Tabel VII), rentang nilai kelembaban yang mampu dihasilkan kendali PI-Fuzi adalah sebesar 0,011 s.d. 0,130 dengan nilai rerata yang diperoleh sebesar 0,021 (Tabel VII). Kemudian untuk rentang nilai kelembaban batu bara yang mampu dihasilkan kendali PI adalah sebesar $-1,155 \times 10^{-5}$ s.d. 0,130 dengan nilai rerata yang diperoleh sebesar 0,010 (Tabel VII). Mengacu pada [19] kedua hasil dari algoritma pengendali yang digunakan sama-sama mampu mencapai rentang kerja dari parameter kelembaban batu bara tersebut, namun kendali PI-Fuzi mampu menunjukkan hasil yang lebih efisien dan mampu mengompensasi lambatnya *fall time* yang dihasilkan oleh kendali PI-Fuzi daripada kendali PI (Tabel VII).

TABEL VII. HASIL ANALISIS KELEMBABAN BATU BARA

Pengendali	Nilai Kelembaban (Fraksi)		Fall Time (jam)	Rerata (Fraksi)
	Nilai Maksimum	Nilai Minimum		
PI	0,130	$-1,155 \times 10^{-5}$	±0,813	0,010
PI-Fuzi	0,130	0,011	±0,865	0,021

Kemudian untuk hasil analisis kehalusan batu bara (Tabel VIII), rentang nilai kehalusan yang mampu dihasilkan kendali PI-Fuzi adalah sebesar 74,980 s.d. 81,821 % dengan nilai rerata yang diperoleh sebesar 75,338 % (Tabel VIII) dan untuk rentang nilai kehalusan batu bara yang mampu dihasilkan kendali PI adalah sebesar 74,931 s.d. 81,823% dengan nilai rerata yang diperoleh sebesar 75,382 % (Tabel VIII). Mengacu

pada [19] kedua hasil dari algoritma pengendali yang digunakan sama-sama mampu mencapai rentang kerja dari parameter kehalusan batu bara tersebut, namun walaupun kendali PI-Fuzi belum mampu menunjukkan hasil yang lebih baik daripada kendali PI, nilai rerata kehalusan batu bara yang dihasilkan masih tetap memenuhi standar yang diinginkan, yaitu di atas 70 % [19].

TABEL VIII. HASIL ANALISIS KEHALUSAN BATU BARA

Pengendali	Nilai Kehalusan (%)		Fall Time (jam)	Rerata (%)
	Nilai Maksimum	Nilai Minimum		
PI	81,823	74,931	±0,547	75,382
PI-Fuzi	81,821	74,980	±0,503	75,338

IV. PENUTUP

Blok fungsi kendali PI-Fuzi berbasis IEC 61499 sangat layak digunakan oleh para praktisi dalam merealisasikan sistem kendali yang berskala industri karena memiliki kinerja yang baik. Hasil pengujian dan validasi kendali PI-Fuzi menggunakan *coal mill* mampu mencapai rentang kerja dari setiap parameter. Nilai rerata parameter aliran batu bara mencapai sebesar 13,584 kg/s, akumulasi massa sebesar 2196 kg, suhu keluaran sebesar 83,296 °C, kelembaban sebesar 0,021, dan kehalusan sebesar 75,338 %. Untuk penelitian ke depan, pengendali PI-Fuzi yang telah diterapkan ini perlu dikaji lebih lanjut lagi kinerjanya, misalnya terkait kekokohan.

REFERENSI

[1] K. Schwab, "The Fourth Industrial Revolution," World Economic Forum, pp. 172, 2016.
 [2] J. J. Downs and E. F. Vogel, "A Plant Wide Industrial Process Control Problem," *Computers Chem. Eng.*, vol. 17, 1993.
 [3] D. E. Seborg, *Process Dynamics and Control*, 3rd ed. Hoboken, N.J: John Wiley & Sons, Inc, 2011.
 [4] A. N. I. Wardana, "Improved PID Controller with Fuzzy Logic for Grate Coolers in Cement Plants," *ZKG International*, vol.56, no.11, pp.82-85, 2004.
 [5] D. E. P. Lebukun, A. N. I. Wardana, and N. Effendy, "Implementation of Plant-Wide PI-Fuzzy Controller in Tennessee Eastman Process," in

International Seminar on Application for Technology of Information and Communication (ISEmantic), pp. 5, 2019.
 [6] R. K. Sahu, S. Panda, and G. T. Chandra Sekhar, "A Novel Hybrid PSO-PS Optimized Fuzzy PI Controller for AGC in Multi Area Interconnected Power Systems," *International Journal of Electrical Power & Energy Systems*, vol. 64, pp. 880–893, Jan. 2015.
 [7] S. Ozdemir, O. Kaplan, I. Sefa, and N. Altin, "Fuzzy PI controlled Inverter for Grid Interactive Renewable Energy Systems," *IET Renewable Power Generation*, vol. 9, no. 7, pp. 729–738, Sep. 2015.
 [8] K. Bedoud, M. Ali-rachedi, T. Bahi, and R. Lakel, "Adaptive Fuzzy Gain Scheduling of PI Controller for Control of the Wind Energy Conversion Systems," *Energy Procedia*, vol. 74, pp. 211–225, Aug. 2015.
 [9] A. M. Ahmed, A. Ali-Eldin, M. S. Elksasy, and F. F. Areeed, "Brushless DC Motor Speed Control using both PI Controller and Fuzzy PI Controller," *International Journal of Computer and Application (IJCA)*, vol. 109, no. 10, pp. 29–35, Jan. 2015.
 [10] I. Filip and I. Szeidert, "Adaptive Fuzzy PI Controller with Shifted Control Singletons," *Expert Systems with Applications*, vol. 54, pp. 1–12, Jul. 2016.
 [11] Z. Tian, S. Li, Y. Wang, and Q. Zhang, "Multi Permanent Magnet Synchronous Motor Synchronization Control based on Variable Universe Fuzzy PI Method," *Engineering Letters*, pp. 9, 2015.
 [12] V. Kumar, K. P. S. Rana, and P. Mishra, "Robust Speed Control of Hybrid Electric Vehicle using Fractional Order Fuzzy PD and PI Controllers in Cascade Control Loop," *Journal of the Franklin Institute*, vol. 353, no. 8, pp. 1713–1741, May 2016.
 [13] Lewis and Zoitl, *Modelling Control Systems Using IEC 61499*. Institution of Engineering and Technology, 2014.
 [14] V. Vyatkin, "IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review," *IEEE Trans. Ind. Inf.*, vol. 7, no. 4, pp. 768–781, Nov. 2011.
 [15] S. Y. Irwan and A. N. I. Wardana, "Distributed Coal Mill Simulator based on IEC 61499," in *The 5th UGM International Conference of Science and Technology*, 2019.
 [16] K. Ogata, *Modern control engineering*, 5th ed. Boston: Prentice-Hall, 2010.
 [17] J. Guo, G. Wu, and S. Guo, "Fuzzy PID Algorithm-Based Motion Control for the Spherical Amphibious Robot," in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Beijing, China, 2015, pp. 1583–1588.
 [18] J. Jantzen, *Foundations of Fuzzy Control (A Practical Approach)*, 2nd ed. Wiley, 2013.
 [19] V. Agrawal, B. K. Panigrahi, and P. M. V. Subbarao, "A Unified Thermo-Mechanical Model for Coal Mill Operation," *Control Engineering Practice*, vol. 44, pp. 157–171, Nov. 2015.
 [20] V. Agrawal, B. K. Panigrahi, and P. M. V. Subbarao, "Review of Control and Fault Diagnosis Methods Applied to Coal Mills," *Journal of Process Control*, vol. 32, pp. 138–153, Aug. 2015.