

# FREQUENCY CONTROL PLTMH DENGAN CAPACITIVE ENERGY STORAGE MENGGUNAKAN CUCKOO SEARCH ALGORITHM

Muhammad Ruswandi Djalal<sup>1</sup>, Dwi Ajiatmo<sup>2</sup>, Soedibyo<sup>3</sup>, Imam Robandi<sup>4</sup>

<sup>1,2,3,4</sup>Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember

<sup>1</sup>[muhammad.ruswandi13@mhs.ee.its.ac.id](mailto:muhammad.ruswandi13@mhs.ee.its.ac.id), <sup>2</sup>[ajiatio@gmail.com](mailto:ajiatio@gmail.com), <sup>3</sup>[dibyio\\_55@yahoo.com](mailto:dibyio_55@yahoo.com), <sup>4</sup>[robandi@ee.its.ac.id](mailto:robandi@ee.its.ac.id)

## Abstrak

Pemanfaatan PLTMH semakin banyak digunakan untuk kebutuhan listrik di daerah terpencil, karena memiliki kelebihan ekonomis, teknis, dan segi ramah lingkungan. PLTMH sangat cocok untuk digunakan pada daerah yang sulit dijangkau listrik PLN. Pada PLTMH, permasalahan yang sering terjadi adalah putaran yang tidak konstan terhadap generator yang disebabkan oleh perubahan beban dari konsumen. Sehingga menyebabkan terjadinya perubahan fluktuasi frekuensi pada sistem yang dapat mengakibatkan kerusakan baik dari sisi pembangkit maupun dari sisi peralatan listrik konsumen. Untuk menunjang kinerja optimal pembangkit listrik tenaga mikrohidro (PLTMH), dibutuhkan sebuah desain teknologi pengendali yang tepat. Oleh karena itu dalam penelitian ini akan dibahas sebuah strategi pengendali frekuensi berbasis *Capacitive Energy Storage (CES)* dan PID. *CES* merupakan peralatan yang dapat menyimpan energy dengan cepat dalam bentuk medan listrik pada kapasitor. Untuk mendapatkan kinerja yang optimal, diperlukan penalaan yang tepat untuk parameter *CES-PID*. Untuk mendapatkan parameter *CES-PID* yang optimal pada PLTMH ini digunakan metode kecerdasan buatan *Cuckoo Search Algorithm (CSA)*. Hasil simulasi menunjukkan bahwa penerapan *CSA* terhadap menala parameter *CES-PID* pada PLTMH, dapat mempercepat *settling time* respon perubahan frekuensi dan juga memperbaiki overshoot respon frekuensi sistem.

**Kata Kunci :** PLTMH, *CES-PID*, *Cuckoo*, Frekuensi, *Overshoot*, *Settling time*

## 1. Pendahuluan

Perubahan beban yang sering terjadi pada sebuah PLTMH dapat mengakibatkan kestabilan frekuensi menjadi terganggu. Selain itu juga dapat mengakibatkan roda gerak berputar lebih cepat. Akibatnya frekuensi listrik akan naik dan bila terlalu tinggi akan membahayakan peralatan listrik konsumen. Oleh karena itu, untuk menunjang kinerja PLTMH ini, pengaturan atau pengendalian frekuensi sangat diperlukan agar selalu berada pada daerah kerja antara 49 Hz – 51 Hz. Pada penelitian ini digunakan sebuah teknologi untuk mengoptimalkan kinerja dari unit pembangkit mikro hidro, yaitu dengan mengaplikasikan *Load Frequency Control (LFC)*. Imam Robandi, (2006). Mekanisme LFC ini dirancang dengan menggunakan *Capacitive Energy Storage (CES)*, yang memiliki kemampuan untuk memberikan kompensasi daya agar dapat mengurangi atau bahkan menghilangkan osilasi frekuensi yang diakibatkan oleh terjadinya perubahan beban daya listrik dari pelanggan. Untuk mendapatkan peredaman yang baik diperlukan penalaan parameter *CES* yang optimal sehingga dapat digunakan dengan baik dalam sistem. Untuk itu pada penelitian ini diusulkan menggunakan metode cerdas *Artificial Intelligent (AI)*, *cuckoo search algorithm (CSA)* untuk menala parameter *CES* dan PID dengan objective function meminimalkan nilai *Integral Time Absolute Error (ITAE)*.

Beberapa metode optimisasi telah diusulkan untuk menentukan nilai parameter control optimal, salah satunya adalah metode konvensional (*Conventional Method*). Tripathy, (1997). maupun metode kecerdasan buatan (*Artificial intelligence*), seperti *Particle Swarm Optimization (PSO)*, dan *Ant Colony Optimization (ACO)*. Miftakhur, (2010). *CSA* merupakan sebuah algoritma cerdas *metaheuristic* yang terinspirasi dari perilaku hidup (*behavior*) burung cuckoo yang parasit dalam menempatkan telurnya. Yang, (2009). Pada penelitian ini, akan ditinjau dan dianalisa respon frekuensi dari PLTMH serta dibandingkan dengan sistem tanpa *controller*, menggunakan PID, menggunakan *CES*, dan dengan menggunakan *CES-PID*.

## 2. Pemodelan Sistem

### 2.1. Sistem Pembangkit Listrik Mikro Hidro

Pembangkit listrik mikro hidro pada prinsipnya memanfaatkan beda ketinggian dan jumlah debit air per detik yang ada pada aliran air saluran irigasi, sungai atau air terjun. Aliran air ini akan memutar poros turbin sehingga menghasilkan energi mekanik. Energi ini selanjutnya menggerakkan generator dan menghasilkan listrik. Lokasi pembangkit listrik mikro hidro yang paling baik adalah lokasi yang memiliki sumber air yang debit airnya selalu mengalir sepanjang tahun. Secara teori, nilai daya listrik yang dapat dihasilkan oleh pembangkit  $P_{th}$  sangat bergantung pada nilai debit

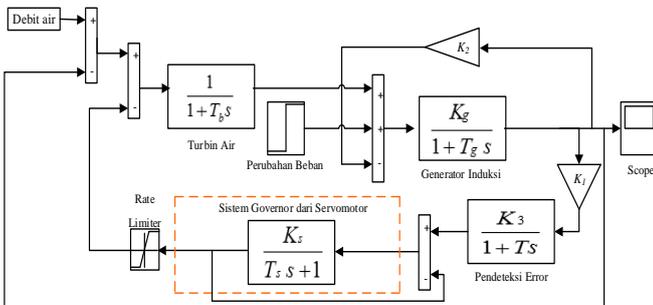
air yang melewati pipa pesat  $Q$  dan juga tinggi terjun air  $H$ . Persamaannya sebagai berikut :

$$P_{th}[W] = Q[m^3 / s].H[m].k[N / kg] \tag{1}$$

Adanya efisiensi dari turbin .turbin dan efisiensi generator  $\eta_{gen}$  yang ditentukan oleh masing-masing pabrikan dengan nilai di sekitar 0.85 menyebabkan nilai daya listrik nyata  $P_{real}$  yang terhasil akan lebih rendah dibandingkan dengan  $P_{th}$ .

$$P_{real}[W] = Q[m^3 / s].H[m].k[N / kg].\eta_{turbin}.\eta_{gen} \tag{2}$$

Untuk pompa yang digunakan sebagai turbin, nilai efisiensinya bervariasi antara 0,6 sampai 0,8. Adapun untuk *cross-flow turbines*, nilainya bervariasi antara 0,5 sampai 0,7. Sedangkan generator yang digunakan pada sistem pembangkit listrik tenaga mikro hidro di sini adalah menggunakan sebuah generator induksi, sebuah motor servo yang dioperasikan sebagai governor, dan beberapa komponen-komponen yang dimodelkan pada simulasi menggunakan program Matlab 2013 (Mfile & Simulink).



Gambar 1. Diagram blok sistem PLTMH

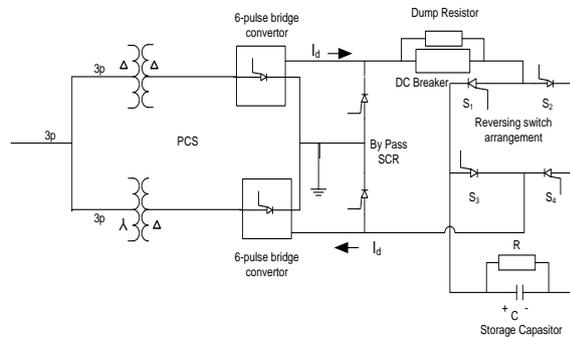
Dari blok pendeteksi error, sinyal  $\Delta\omega$  akan diteruskan ke blok servomotor yang digunakan sebagai governor. Pada blok ini terdapat parameter  $K_s$  dan  $T_s$ . Adapun pada sisi output governor ada sinyal yang diumpambalikan sebagai nilai masukan pada governor tersebut. Juga output dari governor ini diteruskan ke *rate limiter* yang berfungsi untuk membatasi sinyal pada nilai saturasi paling tinggi dan paling rendah yang telah ditentukan. Dari keluaran rate limiter ini, diteruskan sebagai masukan pada blok turbin air. Tabel 1 berikut ini menunjukkan parameter PLTMH yang digunakan pada penelitian ini. *Tsabit, (2015)*.

Tabel 1. Parameter PLTMH

Parameter	Nilai	Ket
Tb	1	Respon waktu turbin air (s)
Kg	1	Penguatan pengatur generator induksi (s)
Tg	13,33 3	Respon waktu generator induksi (s)
K1	5	Konstanta penguatan Error Detection
K2	8,52	Konstanta penguatan deviasi frekuensi
K3	0.004	Penguatan pengatur Error Detection
T	0,02	Respon waktu <i>Error Detection</i>
Ts	0,1	Konstanta waktu governor (s)
Ks	2,5	Penguatan pengatur governor
Sg	40	Rating daya generator mikrohidro (kVA)
pf	0,8	Faktordaya
Vg	400/2 31	Tegangan nominal generator (V)
$\omega$	1500	Kecepatan rotasi nominal (rpm)
fg	50	Frekuensi nominal mikrohidro (Hz)

2.2. Capacitive Energy Storage (CES)

CES merupakan sebuah peralatan untuk menyimpan dan melepaskan daya dalam jumlah yang besar secara simultan. CES menyimpan energi dalam bentuk medan listrik pada kapasitor. Sebuah CES terdiri dari *storage capacitor* dan *Power Conversion System (PCS)*. *Miftakhur, (2010)*.



Gambar 2. Capacitive Energy Storage

Kapasitor penyimpan terdiri dari beberapa kapasitor diskrit yang dihubungkan secara paralel, dengan kapasitansi  $C$ . *S.C.Tripathy, (1997) ; Rajesh, (2010)*. Rugi-rugi bocor dan dielektrik kapasitor bank pada CES dimodelkan dengan resistansi  $R$  yang dihubungkan secara paralel pada kapasitor. Kapasitor penyimpan dihubungkan ke jala-jala melalui *Power Conversion System (PCS)* 12-pulsa. PCS terdiri dari penyearah *ac to dc* dan inverter *dc to ac*. Thyristor *bypass* berfungsi untuk memberikan jalan bagi arus  $I_d$  saat terjadi kegagalan konverter. Pemutus dc memungkinkan arus  $I_d$  dialihkan ke tempat pembuangan energi resistor  $R_D$  jika konverter gagal. Dengan mengabaikan rugi-rugi, tegangan *bridge*  $E_d$  adalah seperti persamaan,

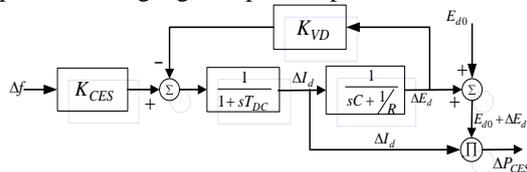
$$E_d = 2E_{d0} \cos \alpha - 2I_d R_D \tag{3}$$

$$E_{d0} = \frac{[E_{dmax}^2 + E_{dmin}^2]^{1/2}}{2} \quad (4)$$

Selama gangguan sistem terjadi, jika tegangan kapasitor terlalu rendah dan jika gangguan lain terjadi sebelum tegangan kembali ke nilai normal, energi akan lebih banyak ditarik dari kapasitor yang dapat menyebabkan kontrol yang terputus-putus. Untuk mengatasi masalah ini, batas bawah untuk tegangan kapasitor, diambil 30% dari nilai rating  $E_{d0}$ . Sehingga,

$$E_{dmin} = 30E_{d0} \quad (5)$$

Tegangan CES harus kembali ke nilai awal dengan cepat, sehingga setelah gangguan beban terjadi unit CES siap bekerja untuk gangguan beban berikutnya. Oleh karena itu, deviasi tegangan kapasitor digunakan sebagai sinyal umpan balik negatif dalam loop kontrol CES sehingga pemulihan tegangan cepat dicapai.



Gambar 2.15 Diagram blok CES

Berikut Deviasi tegangan kapasitor  $\Delta E_d$ ,

$$\Delta E_d = \left[ \frac{1}{sC + 1/R} \right] \Delta I_d \quad (6)$$

Daya output CES yang dilepas ke sistem pada saat terjadi perubahan beban adalah sebagai berikut,

$$\Delta P_{CES} = (E_{d0} + \Delta E_d) \cdot \Delta I_d \quad (7)$$

### 3. Cuckoo Search Algorithm Yang, (2009)

Burung cuckoo dalam berkembangbiak memiliki cara yang beragam dan unik. Dari sekian spesies cuckoo diketahui bahwa 59 diantaranya merupakan cuckoo yang bersifat parasit dalam berkembangbiak yaitu mereka memanfaatkan sarang burung lain yang berbeda spesies untuk menetas telur mereka. Bahkan tak jarang burung cuckoo yang menaruh telur pada sarang burung cuckoo yang lain. Ada beberapa jenis cuckoo yang membuang telur dari induk asli pada sarang untuk meningkatkan kemungkinan menetas telur mereka. Mungkin saja terjadi konflik antara burung host dan burung cuckoo pada saat burung cuckoo menaruh telurnya, sehingga burung host membuang telur cuckoo tersebut atau meninggalkan sarang mereka kemudian membuang sarang yang baru. Perilaku parasit yang lain adalah saat burung cuckoo menetas, biasanya telur burung cuckoo menetas lebih awal dari telur burung host, telur burung yang belum menetas tadi dibuang dari sarang agar anak cuckoo tersebut mendapatkan lebih banyak makanan. Penemu CS, Xin She Yang dan Suash Deb menggunakan beberapa aturan antara lain :

1. Setiap cuckoo meletakkan satu telur pada satu waktu dan membuang telur didalamnya pada sarang yang dipilih acak.
2. Sarang terbaik dengan kualitas telur tertinggi akan dilanjutkan ke generasi berikutnya.
3. Jumlah sarang burung yang tersedia berjumlah tetap, dan peluang telur yang diletakkan oleh cuckoo yang ditemukan burung inang sebesar  $p \in [0,1]$ . Dalam kasus ini, burung dapat membuang telur cuckoo atau meninggalkan sarang, lalu membuat sarang baru.

Aturan terakhir dapat disederhanakan dengan pendekatan fraksi  $p$  dari  $n$  sarang digantikan dengan sarang baru (dengan solusi baru secara acak). Untuk masalah maksimisasi, kualitas atau fitness dari sebuah solusi dapat sebanding dengan nilai dari objective function. Bentuk lain dari fitness dapat didefinisikan dalam cara yang sama dengan fungsi fitness pada algoritma genetika. Untuk mempermudah, dapat menggunakan representasi yang sederhana bahwa setiap telur dalam sarang merepresentasikan sebuah solusi, dan telur cuckoo merepresentasikan solusi baru, tujuannya adalah untuk menggunakan yang baru dan potensial solusi yang lebih baik (cuckoos) untuk mengganti solusi yang tidak bagus pada sarang. Kemudian telur-telur tadi akan dievolusikan, telur yang lebih akan menggantikan telur lainnya yang diukur oleh fitness, layaknya dalam GA. Dalam suatu sarang host bisa saja terdapat dua telur, dengan kata lain sarang bisa menyimpan lebih dari satu solusi hanya saja untuk menyederhanakan permasalahan, suatu sarang hanya akan menyimpan satu solusi (telur). Berdasarkan tiga aturan, langkah-langkah dasar Cuckoo Search (CS) dapat diringkas sebagai pseudo code yang ditunjukkan pada tabel berikut.

Tabel 2. Pseudo code dari Cuckoo Search (CS)

<p><b>Begin</b>                  Fungsi Objektif <math>f(x)</math>, <math>x = (x_1, \dots, x_d)^T</math>                  Inisialisasi populasi dari <math>n</math> sarang burung target <math>x_i</math> (<math>i = 1, 2, \dots, n</math>)  <b>While</b> (<math>t &lt; \text{generasiTotal}</math>) atau (kriteria lain untuk berhenti)                  Evaluasi kualitas dari masing-masing sarang burung cuckoo                  Pilih dari burung cuckoo secara acak dan lakukan random walk  <b>Jika</b> (<math>F_i &gt; F_j</math>)                  Gantikan burung cuckoo <math>j</math> dengan burung cuckoo <math>i</math>  <b>End If</b>                  Reset ulang sarang-sarang dengan kondisi terburuk (<math>P_a</math>)                  Simpan sarang-sarang yang berhasil lolos                  Urutkan solusi dan cari yang terbaik  <b>End While</b>                  Proses hasil dan visualisasi  <b>End</b></p>
--

Ketika generasi solusi baru  $x^{(t+1)}$  untuk seekor cuckoo  $i$ , Levy flight ditunjukkan sebagai berikut :

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Levy}(\lambda) \quad (3)$$

$\alpha > 0$  adalah ukuran tahapan yang seharusnya berhubungan dengan skala dari masalah yang menarik. Dalam kebanyakan kasus, dapat menggunakan  $\alpha = 1$ . Levy flight pada dasarnya memberikan jalan yang acak ketika tahapan acak panjang diambil dari distribusi Levy :

$$Levy \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (4)$$

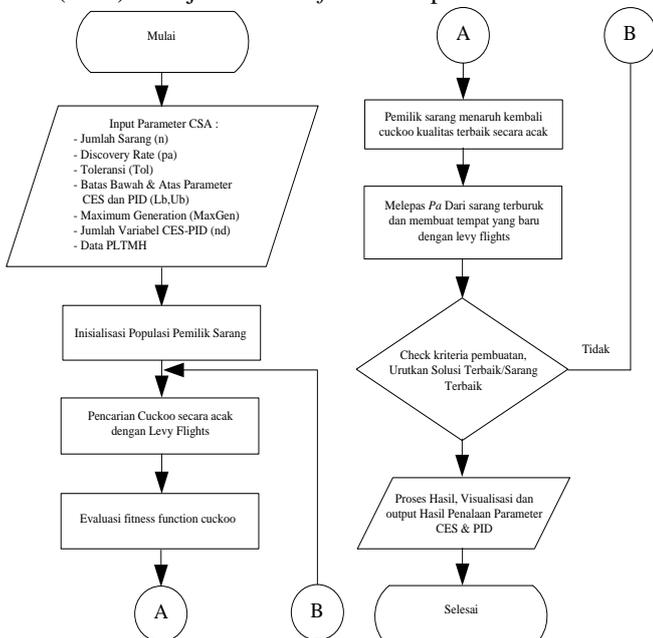
Mempunyai variasi yang tak terbatas dengan rata-rata tak terbatas. Tahapan penting bagian dari sebuah proses random walk dengan kekuatan rendah, distribusi tahapan panjang dengan bagian terakhir yang berat. Beberapa dari solusi baru seharusnya digenerasi oleh jalan berputar Levy mencapai solusi terbaik sejauh ini, kecepatan akan naik pada pencarian local. Tetapi, sebuah fraksi substansial dari solusi baru seharusnya digenerasi oleh medan acak yang jauh dan lokasi harus cukup jauh dari solusi terbaik, ini akan membuat sistem tidak akan terjebak pada optimasi lokal.

**3.1. Penalaan CES-PID dengan CSA**

Gambar 5 menunjukkan diagram alir algoritma metode *Cuckoo Search Algorithm* (CSA) yang digunakan pada penelitian ini untuk menala parameter CES dan PID. Fungsi objektif yang digunakan untuk menguji kestabilan sistem adalah dengan *Integral Time Absolut Error* (ITAE).

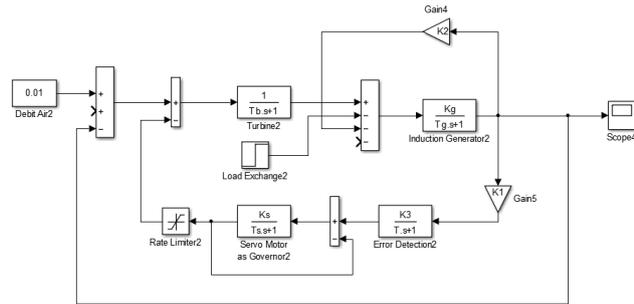
$$ITAE = \int_0^t |\Delta\omega(t)| dt \quad (5)$$

Parameter CES yang ditala oleh CSA adalah KCES, tw, Tdc, Kp, Ki dan Kd. Adapun untuk diagram alir proses penalaan parameter CES dengan menggunakan metode *Cuckoo Search Algorithm* (CSA) ditunjukkan oleh *flowchart* pada Gambar 5.

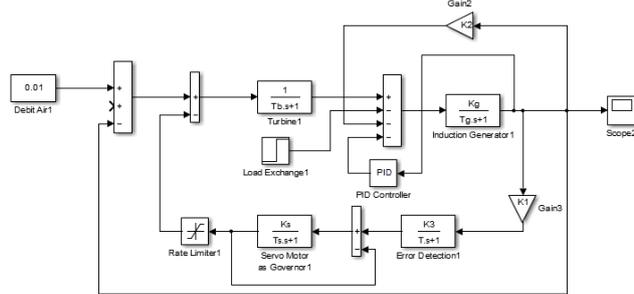


**Gambar 5. Diagram alir penelitian**

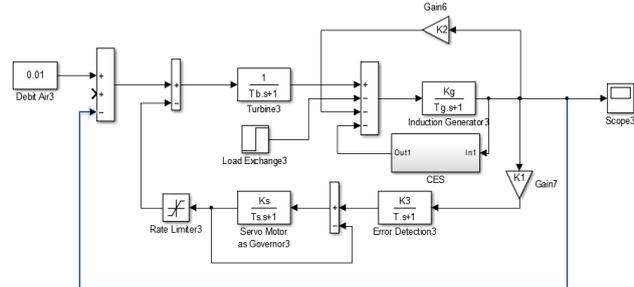
Berikut pemodelan untuk masing-masing model kontrol pada Simulink Matlab 2013, untuk sistem PLTMH tanpa Kontrol, dengan PID-CSA, dengan CES-CSA dan dengan CES-PID-CSA yang ditala CSA.



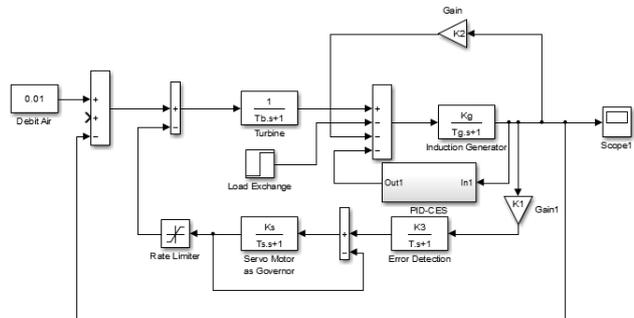
**Gambar 6. Pemodelan Simulink Tanpa Kontrol**



**Gambar 7. Pemodelan Simulink Kontrol PID**



**Gambar 8. Pemodelan Simulink Kontrol CES**



**Gambar 9. Pemodelan Simulink Kontrol CES-PID**

Adapun data parameter-parameter CSA berikut.

**Tabel 3. Parameter CSA**

Parameter	Nilai
Number of nests	25
Discovery rate of alien eggs/solutions	0.25
Toleransi	1.0e-5
Max Generation	50
Jumlah Variabel (nd)	5

Berikut hasil optimasi dengan metode CSA.

**Tabel 4. Hasil Optimasi CES-PID dengan CSA**

Total number of iterations=2551			
fmin =			
7.8005e-10			
bestnest =			
83.3173	0.0384	14.2518	0.1845
0.1352			

Hasil optimasi CSA didapatkan nilai *fitness function* sebesar  $7.8005e-10$ , dengan 50 iterasi, serta nilai *bestnest* adalah hasil optimasi sarang terbaik dari telur cuckoo, dan merupakan parameter hasil optimasi Kces, Tdc, Kp, Ki dan Kd.

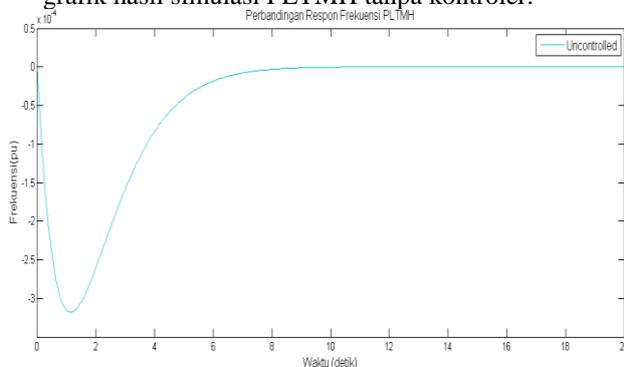
**Tabel 5. Hasil Penalaan Parameter CES-PID**

Parameter	Konstrain		Hasil CSA
	Bawah	Atas	
<i>Kces</i>	80	90	83.3172
<i>Tdc</i>	0.03	0.06	0.0384
<i>Kp</i>	10	15	14.2518
<i>Ki</i>	0.1	0.5	0.1845
<i>Kd</i>	0	1	0.1352

**4. Hasil Simulasi Dan Analisis**

**4.1 Respon Frekuensi PLTMH tanpa Controller**

Simulasi pertama adalah simulasi open loop PLTMH tanpa controller. Berikut ditunjukkan pada grafik hasil simulasi PLTMH tanpa kontroler.



**Gambar 7. Grafik frekuensi PLTMH uncontroller**

Gambar grafik di atas menunjukkan grafik respon frekuensi PLTMH open loop atau tanpa dilengkapi controller. Dari grafik di atas didapatkan nilai overshoot dan settling time PLTMH yang sangat besar, seperti pada tabel berikut.

**Tabel 6 Nilai overshoot dan settling time**

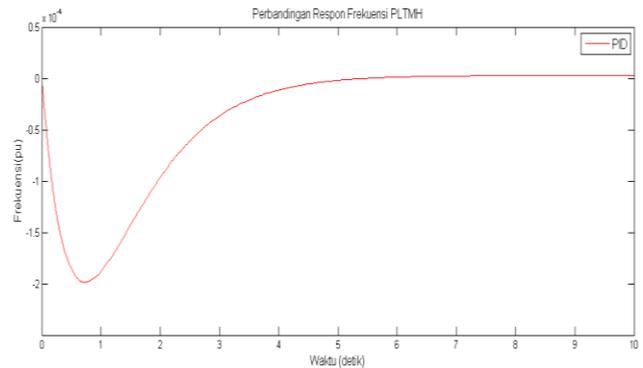
Karakteristik	Nilai
<i>Overshoot (pu)</i>	-0.0003179 & 3.523e-09
<i>Settling time (detik)</i>	7,8

Nilai overshoot dan settling time yang dihasilkan dari simulasi pembangkit listrik mikro hidro tanpa controller diatas merupakan nilai overshoot dan settling time yang paling besar diantara semua simulasi yang dilakukan, yaitu sebesar -0.0003179

& 3.523e-09. Hal ini disebabkan karena *control* frekuensi dari sistem hanya dilakukan atau dibebankan kepada governor saja, sehingga sistem yang tidak dilengkapi *controller* frekuensi pada generatormya ini menjadi sistem yang paling tidak stabil diantara sistem lain yang diberi *controller*.

**4.2 Respon Frekuensi PLTMH Menggunakan Controller PID**

Berikutnya simulasi dengan menggunakan kontrol PI, dari hasil simulasi didapatkan grafik berikut :



**Gambar 9. Grafik respon frekuensi PLTMH dengan controller PID**

Dapat dilihat dari gambar di atas bahwa ketika terjadi perubahan beban atau gangguan (dalam hal ini perubahan beban), frekuensi sistem berosilasi terlebih dahulu sebelum mencapai kondisi tunaknya (*steady state*). Adapun nilai overshoot yang terjadi dan juga settling time nya detailnya dapat dilihat pada tabel berikut.

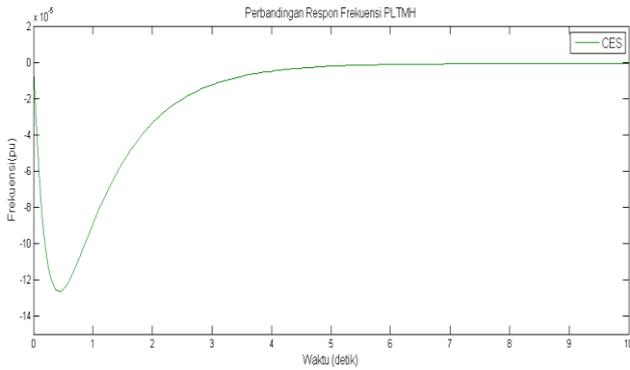
**Tabel 7. Nilai overshoot dan settling time sistem dengan controller PID**

Karakteristik	Nilai
<i>Overshoot (pu)</i>	-0.0001984 & 2.82e-06
<i>Settling time (detik)</i>	5.8

Dapat dilihat dari nilai yang tertera pada tabel di atas bahwa dengan pemasangan controller PID, frekuensi sistem berosilasi antara -0,0001984 pu hingga 2.82e-06 pu terhadap kondisi steady state, ketika terjadi gangguan seperti perubahan beban. Kemudian, peredaman osilasi tersebut membutuhkan waktu 5.8 detik untuk mencapai kondisi tunaknya. Jika dibandingkan dengan respon frekuensi sistem yang tidak dilengkapi dengan controller PID maupun CES, nilai overshoot nya menjadi lebih kecil, namun nilai waktu peredamannya (*settling time*) menjadi lebih lama dibandingkan dengan sistem yang tidak dipasang controller.

**4.3. Respon Frekuensi PLTMH Menggunakan Controller CES**

Berikutnya simulasi dengan menggunakan kontroler CES, dari hasil simulasi didapatkan :



**Gambar 9. Grafik respon frekuensi PLTMH yang dilengkapi controller CES**

Dapat dilihat dari gambar di atas, hasil respon frekuensi untuk sistem PLTMH dengan controller CES yang telah dipasang tanpa controller PID didalamnya. Dari grafik tersebut juga dapat diketahui bahwa nilai overshoot dan nilai settling time nya yaitu seperti pada tabel berikut.

**Tabel 8. Nilai overshoot dan settling time sistem dengan controller CES**

Karakteristik	Nilai
Overshoot (pu)	-0,0001264
Settling time (detik)	>10

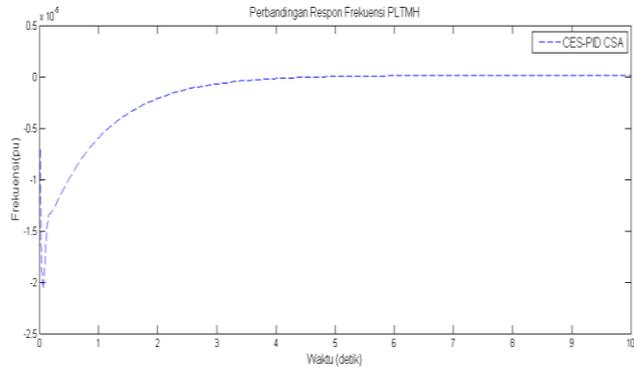
Nilai overshoot yang pada sistem ini sebesar -0,0001264 pumenunjukkan bahwa ketika sistem mengalami perubahan beban atau gangguan, maka frekuensi sistem berisolasi turun sebesar -0,0001264 pu, dan akan kembali pada nilai tunak nya setelah setelah >10 detik lebih. Tentunya settling time sebesar ini merupakan kondisi yang tidak baik untuk sebuah sistem pembangkit listrik ditinjau dari kestabilannya. Apabila frekuensi tidak kembali pada kondisi steady state setelah terjadi perubahan beban atau gangguan lainnya, maka sistem pembangkit itu sendiri akan terganggu bahkan hingga gangguan itu juga dapat dirasakan oleh pelanggan listrik pembangkit tersebut.

**4.4. Respon Frekuensi PLTMH Menggunakan Controller CES-PID CSA**

Selanjutnya dilakukan simulasi dengan penambahan komponen control blok PID pada controller CES, karena pada simulasi sistem sebelumnya yang hanya menggunakan controller CES saja tanpa PID. Parameter CES dan PID disini akan ditala dengan metode optimasi Cuckoo Search Algorithm (CSA).

Gambar 10 berikut menunjukkan nilai perubahan frekuensi terhadap waktu untuk sistem pembangkit mikro hidro dengan menggunakan controller CES sekaligus PID didalamnya. Dari grafik tersebut, diperoleh nilai osilasi frekuensi yaitu hanya -2.067e-05 dan 1.213e-07. Dan juga

waktu peredaman yang lebih singkat dibandingkan sistem yang lain. Sehingga sistem ini merupakan sistem yang memiliki respon frekuensi paling bagus dibandingkan dengan tiga sistem lain yang disimulasikan sebelumnya. Karena memiliki nilai overshoot yang semakin mendekati nol, dan nilai settling time yang semakin kecil.



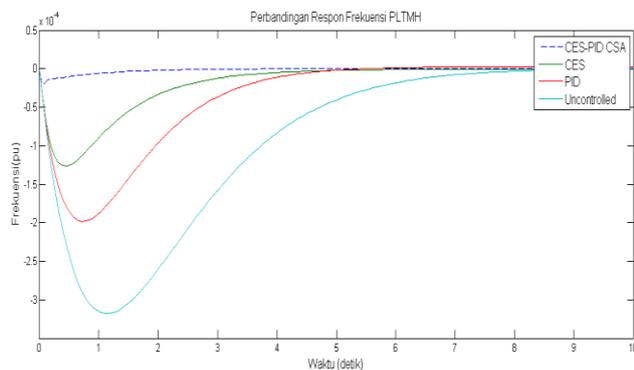
**Gambar 10. Grafik respon frekuensi PLTMH yang dilengkapi controller CES**

Adapun nilai overshoot dan settling time nya dapat dilihat pada tabel berikut.

**Tabel 9. Nilai overshoot dan settling time untuk PLTMH dengan controller CES-PID CSA**

Karakteristik	Nilai
Overshoot (pu)	-2.067e-05 dan 1.213e-07
Settling time (detik)	4.6

Dari hasil yang telah disimulasikan, berikut ini dapat dilihat keempat model yang telah disimulasikan.



**Gambar 11. Grafik perbandingan respon frekuensi keempat kontroler**

Dari empat model sistem pembangkit listrik mikro hidro yang telah disimulasikan, dapat disimpulkan bahwa suatu pembangkit listrik mikro hidro mutlak memerlukan controller sebagai peredam osilasi frekuensi yang terjadi akibat perubahan beban. Controller CES-PID yang diajukan dalam penelitian ini memiliki pengaruh yang signifikan dalam peredaman osilasi tersebut dan cocok diterapkan untuk sistem pembangkit mikro hidro, di mana untuk metode penalaannya dengan

menggunakan metode cerdas *Artificial Intelligent Cuckoo Search Algorithm*.

### 5. Kesimpulan

Dengan menggunakan metode cerdas *Cuckoo Search Algorithm (CSA)* sebagai metode penalaan *controller Capacitive Energy Storage (CES)& PID*, didapatkan hasil penalaan parameter nilai *CES-PID* yang optimal di mana, *K<sub>ces</sub>* sebesar 83.3172, *T<sub>dc</sub>* sebesar 0.0384, *K<sub>p</sub>* sebesar 14.2518, *K<sub>i</sub>* sebesar 0.1845 dan *K<sub>d</sub>* sebesar 0.1352.

Dari hasil simulasi dapat disimpulkan bahwa dengan menggunakan kombinasi *energy storage CES-PID* pada sistem kontrol *Load Frequency Control (LFC)* yang dirancang, dapat memperbaiki respon frekuensi sebuah sistem PLTMH. Hal ini ditunjukkan dengan nilai *settling time* sebesar 4.6 detik yang merupakan nilai *settling time* tercepat dan juga nilai *overshoot* sebesar -2.067e-05 dan 1.213e-07 yang merupakan nilai *overshoot* terkecil dari model kontroler yang lain.

### Daftar Pustaka :

- Chatterjee, A. (2012) : *Transient Performance Improvement of Grid Connected Hydro System Using Distributed Generation and Capacitive Energy Storage Unit*. Sciencedirect. 2012.
- Hadi, S. (1999) : *Power System Analysis*. McGraw-Hill 1999.
- Kundur, P. (1993) : *Power System Stability and Control*, McGraw-Hill, Inc. 1993.
- Mukherjee, V. (2009) : *Application of Capacitive Energy Storage for Transient Performance Improvement of Power System*. Sciencedirect. 2009.
- Miftakhur R, M, D, dkk. (2010) : *Koordinasi Optimal Capacitive Energy Storage (CES) dan Automatic Voltage Regulator (AVR) Pada Sistem Tenaga Listrik Menggunakan Ant Colony Optimization (ACO)*. EECCIS 2010.
- Miftakhur, R, M, D, dkk. (2010) : *Optimal Tuning of Capacitive Energy Storage on Wind-Diesel Hybrid Power System on Isolated Area using Particle Swarm Optimization*. ICAST 2010.
- S.C. Tripathy. (2010) : *Improved Load Frequency Control With Capacitive Energy Storage*. Energy Cottut-rs. 1997
- S.C. Tripathy. (1996) : *Dynamic Performance Of Wind-Diesel Power System With Capacitive Energy Storage*. Energy Convers. 1996.
- Rajesh, J, A. (2010) : *Automatic Generation Control of an Interconnected Power System with Capacitive Energy Storage*, International Journal of Electrical Power and Energy System Engineering. 2010.
- Rajesh, J, A. (2010) : *Effect of Capacitive Energy Storage on Automatic Generation Control*, IPEC. 2005.

Robandi, I. (2006) : *Modern Power System Control*. ANDI, Yogyakarta. 2006.

Robandi, I. (2006) : *Desain Sistem Tenaga Modern: Optimasi, Logika Fuzzy, dan Algoritma Genetika*. ANDI. 2006.

Tsabit Mustarin. (2010) : *Desain Optimal Load Frekuensi Kontrol dengan Superconducting Magnetic Energy Storage pada PLTMH menggunakan*. TA ITS. 2015.

Yang, X, S. (2013) : *Cuckoo Search and Firefly Algorithms Theory and Applications*. 2013.

Yang, X, S. (2008) : *Nature-Inspired Metaheuristic Algorithms*. Frome: Luniver Press. 2008.