Sistem Pengaduk Cendol Otomatis Berbasis PID Dengan Metode *Model-Based Tuning* Untuk Menjaga Stabilitas Kecepatan Putaran

[1st]Haikal Pivaldo Prananda
Program Studi Teknik Elektro
Universitas Telkom
Purwokerto, Indonesia
haikalpvaldo@student.telkomuniversity.ac.id

[2nd]Yulian Zetta Maulana
Program Studi Teknik Elektro
Universitas Telkom
Purwokerto, Indonesia
yulianm@telkomuniversity.ac.id

[3rd]Risa Farrid Christianti
Program Studi Teknik Elektro
Universitas Telkom
Purwokerto, Indonesia
risabc@telkomuniversity.ac.id

***Abstrak* — Proses pengadukan pada pembuatan cendol membutuhkan kecepatan putaran yang stabil untuk menghasilkan tekstur dan homogenitas adonan yang baik. Namun, pada sebagian besar UMKM proses pengadukan masih dilakukan secara manual sehingga kecepatan putaran sulit dipertahankan ketika viskositas adonan meningkat. Penelitian ini bertujuan merancang dan mengimplementasikan sistem pengaduk cendol otomatis berbasis kendali *Proportional Integral Derivative* (PID) menggunakan metode *Model-Based Tuning*. Sistem menggunakan motor induksi tiga fasa yang dikendalikan oleh *Variable Frequency Drive* (VFD) dengan sensor *Hall Effect* sebagai umpan balik kecepatan. Model plant didapatkan melalui identifikasi sistem pada kondisi *open loop* dan menghasilkan nilai *best fit* sebesat 81,13%. Berdasarkan model tersebut diperoleh parameter PID sebesar Kp = 5,9349, Ki = 0,1819, dan Kd = 46,9086. Hasil implementasi menunjukkan bahwa sistem dapat mempertahankan kecepatan putaran pada *setpoint* 30 RPM, dengan *overshoot* sebesar 1,00% dan *steady state error* sebesar 0%. Perbandingan hasil simulasi dan *hardware* bnerdasarkan parameter *steady state* RPM menghasilkan rata-rata *error* sebesar 0,93% dengan tingkat akurasi 99,07%. Hasil penelitian menunjukkan bahwa metode *Model-Based Tuning* dapat menghasilkan parameter PID yang sesuai dengan karakteristik *plant* sehinggal sistem pengaduk cendol otomatis dapat bekerja secara stabil.**

***Kata kunci*— Cendol, Kendali PID, *Model-Based Tuning*, Motor Induksi, VFD**

I. PENDAHULUAN

Cendol merupakan minuman tradisional Indonesia yang memiliki peluang besar untuk dikembangkan menjadi usaha mikro, kecil, menengah (UMKM) [1]. Dalam proses pembuatannya, tahap pengadukan berperan penting untuk mendapatkan tekstur dan homogenitas adonan yang baik [2]. Namun, pada sebagin besar usaha UMKM proses pengadukan masih dilakukan secara manual, sehingga kecepatan putaran pengaduk sulit dipertahankan tetap stabil ketika viskositas adonan meningkat [3]. Kondisi tersebut membuat hasil pengadukan menjadi tidak merata dan kualitas produk yang dihasilkan tidak konsisten [4].

Beberapa penelitian telah mengembangkan sistem pengaduk otomatis menggunakan mikrokontroler, sensor, maupun pengendali *Proportional Integral Derivative* (PID) agar kecepatan motor lebih stabil [5][6][7]. Hasil penelitian tersebut menunjukkan bahwa pengendali PID mampu meningkatkan kinerja sistem dibandingkan sistem yang tidak menggunakan pengendali [7]. Namun, sebagian besar penelitian masih menggunakan metode penalaan PID secara konvensional atau *trial and error*, sehingga parameter pengendali yang didapatkan belum sepenuhnya sesuai dengan karakteristik plant dan belum mampu memberikan respons yang optimal [8][9].

Berdasarkan permasalahan tersebut, penelitian ini mengembangkan sistem pengaduk cendol otomatis yang menggunakan motor induksi tiga fasa, dan dikendalikan dengan *Variable Frequency Drive* (VFD) serta didukung oleh sensor *Hall Effect* sebagai umpan balik. Parameter PID ditentukan dengan metode *Model-Based Tuning* berdasarkan model sistem yang telah diidentifikasi, sehingga diharapkan mampu menghasilkan pengendalian kecepatan yang lebi stabil. Penelitian ini bertujuan membuat dan menerapkan sistem pengaduk cendol secara otomatis, serta menguji kinerja metode *Model-Based Tuning* dalam menjaga kecepatan putaran motor tetap stabil saat proses pengadukan berlangsung [10].

II. KAJIAN TEORI.

A. *Proportional Integral Derivative* (PID)

Proportional Integral Derivative (PID) adalah metode pengendalian yang digunakan untuk memastikan keluaran sistem tetap sesuai dengan nilai yang ditentukan. PID bekerja berdasarkan selisih antara nilai aktual dan nilai referensi atau *error*. Metode ini terdiri dari tiga parameter, yaitu Kp, Ki,dan Kd yang masing-masing memiliki peran yang berbeda. Pada kp berfungsi untuk mempercepat respons, Ki digunakan untuk mengurangi error, dan Kd digunakan untuk meningkatkan kestabilan sistem [11].

Gabungan ketiga parameter menghasilkan sinyal kendali u(t) yang digunakan untuk mengatur respons sistem. Persamaan umum pengendeli PID dapat dinyatakan sebagai berikut:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

Dengan u(t) sinyal keluaran pengendali PID, e(t) sinyal error yang didapatkan dari selisih antara setpoint dan nilai

aktual, serta Kp, Ki, dan Kd menyatakan pada masing-masing konstanta proporsional, integral, dan derivatif [12].

B. *Model-Based Tuning*

Model-Based Tuning adalah metode pendekatan dalam penyetelan parameter kontrol PID dengan mengacu pada model matematis dari sistem (*plant*) yang ada [13]. Model tersebut biasanya dihasilkan dalam bentuk fungsi alih (*transfer function*) yang menggambarkan hubungan antara sinyal input dan sinyal output sistem, sehingga menghasilkan persamaan sebagai berikut [14]:

$$G(s) = \frac{Y(s)}{U(s)} \quad (2)$$

Dengan G(s) adalah fungsi alih, Y(s) merupakan keluaran sistem, dan U(s) merupakan masukan sistem []. Salah satu bentuk fungsi alih yang umum digunakan adalah orde dua, yang dinyatakan persamaan sebagai berikut [14]:

$$G(s) = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} \quad (3)$$

Koefisien $a_2, a_1, a_0, b_1, danb_0$ didapatkan melalui proses identifikasi sistem dan kemudian digunakan sebagai dasar untuk menentukan parameter PID yang sesuai dengan karakteristik plant menggunakan metode *Model-Based Tuning* [14].

C. *Variable Frequency Drive* (VFD)

Variable Frequency Drive (VFD) adalah alat elektronika yang digunakan untuk mengatur kecepatan motor induksi dengan mengubah frekuensi dan tegangan listrik yang diberikan. Prinsip kerja VFD memungkinkan motor Listrik AC berjalan dengan kecepatan yang bisa diatur sesuai dengan kebutuhan proses, sehingga lebih fleksibel dibandingkan cara pengaturan kecepatan yang biasa digunakan. VFD banyak digunakan di berbagai industri karena mampu meningkatkan penggunaan energi secara lebih efisien, membuat proses *start* dan *stop* motor, serta memberikan kontrol kecepatan yang lebih stabil [15].

D. *Pulse Width Modulation* (PWM)

Pulse Width Modulation (PWM) adalah metode untuk mengatur nilai tegangan rata-rata dengan memvariasikan lebar pulsa (*duty cycle*) pada setiap periode sinyal. Semakin besar *duty cycle* yang diberikan, semakin tinggi tegangan rata-rata yang dihasilkan, sedangkan *duty cycle* yang lebih kecil menghasilkan tegangan rata-rata yang lebih rendah. PWM bekerja dengan mengatur keadaan aktif (ON) dan tidak aktif (OFF) secara cepat, sehingga membentuk rangkaian pulsa yang bisa digunakan untuk mengatur besarnya daya yang diterima oleh beban [16].

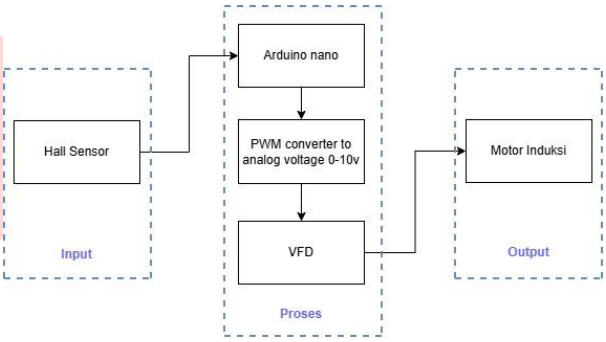
E. Identifikasi sistem

Identifikasi sistem Adalah metode untuk memperoleh representasi matematis dari sistem tersebut berdasarkan data input dan output yang diperoleh melalui pengujian. Metode ini digunakan ketika karakteristik sistem sulit ditentukan secara analitis atau terdapat banyak parameter yang tidak diketahui. Hasil identifikasi sistem bisa dilakukan dengan menggunakan fungsi transfer, model state-space, atau model matematis lainnya yang digunakan untuk analisis, simulasi, dan perancangan sistem kendali [17].

III. METODE

A. Rancangan Sistem

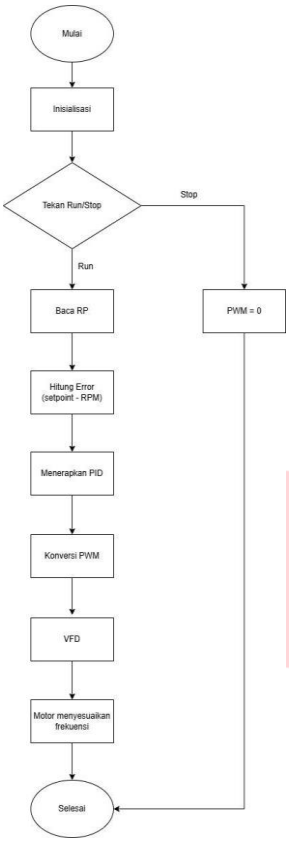
Rancangan sistem adalah tahap awal dari penelitian yang bertujuan untuk menggambarkan hubungan antar komponen pada sistem pengaduk cendol otomatis. Blok diagram memberikan gambaran aliran data dan sinyal kendali mulai dari *input*, proses pengendalian, sampai *output* sistem. Rancangan ini menjadi dasar dalam implementasi pengendali PID berbasis *Model-Based Tuning* untuk menjaga kestabilan kecepatan putaran motor.



GAMBAR 1
(Blok Diagram Sistem)

Gambar 1 menunjukkan proses sistem dimulai dengan pengukuran kecepatan putaran motor menggunakan sensor *Hall Effect* 3144E sebagai sinyal umpan balik. Datanya diukur dalam RPM dan diproses oleh Arduino Nano, yang kemudian dibandingkan dengan nilai *setpoint* untuk mencari nilai *error*. Setelah itu, Arduino melakukan perhitungan sinyal kendali menggunakan algoritma PID dan menghasilkan sinyal PWM yang diubah menjadi tegangan analog 0-10 Volt melalui PWM *to* Analog. Tegangan analog ini berfungsi sebagai *input* untuk VFD dalam mengatur frekuensi keluaran, sehingga kecepatan motor induksi dapat dikendalikan sesuai dengan *setpoint* yang ditentukan. Semua proses ini dilakukan secara terus-menerus dengan mekanisme umpan balik untuk menjaga stabilitas kecepatan putaran motor meskipun terdapat perubahan beban selama proses pengadukan.

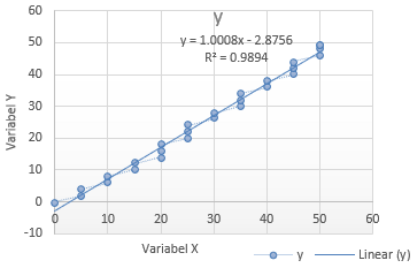
Gambar 2 menunjukkan *flowchart* sistem dimulai dengan inisialisasi seluruh komponen, yaitu Arduino Nano, Sensor *Hall Effect*, modul PWM *to* Analog, dan VFD. Setelah itu, sistem menunggu instruksi dari tombol Run/Stop. Jika dalam posisi Stop, nilai PWM bernilai 0 sehingga motor berhenti beroperasi. Sebaliknya, jika dalam posisi Run, sensor *hall effect* akan membaca RPM motor, kemudian nilai tersebut dibandingkan dengan *setpoint* untuk mendapatkan nilai *error*. Nilai *error* kemudian diproses dengan pengendali PID untuk menghasilkan sinyal PWM, yang kemudian dikonversikan menjadi tegangan analog 0-10 Volt, dan dikirimkan ke VFD. VFD kemudian mengatur frekuensi keluaran motor agar kecepatan putaran sesuai dengan *setpoint*. Proses ini akan terus berlangsung selama sistem berada dalam keadaan Run.



GAMBAR 2
(Flowchart sistem)

B. Kalibrasi Sistem

Kalibrasi sistem dilakukan agar pembacaan kecepatan putaran motor yang dilakukan oleh sensor *hall effect* 3144E sesuai dengan kondisi sebenarnya. Proses kalibrasi ini dilakukan dengan membandingkan nilai RPM yang dibaca sensor terhadap hasil pengukuran menggunakan tachometer sebagai acuan. Data dari pengukuran selanjutnya dianalisis dengan menggunakan metode regresi linear untuk mendapatkan faktor koreksi yang berupa nilai *slope* dan *offset*.



GAMBAR 3
(Grafik regresi linear)

Gambar 3 menunjukkan adanya hubungan linear antara hasil pembacaan sensor dan alat acuan dengan tingkat kesesuaian yang baik dan didapatkan hasil nilai *slope* sebesar 1,0008 dan *offset* -2,8756. Nilai *slope* dan *offset* yang didapatkan kemudian diterapkan pada program Arduino sebagai faktor koreksi sehingga pembacaan RPM menjadi lebih akurat sebelum digunakan sebagai sinyal umpan balik dalam sistem pengendali PID.

C. Pengujian Sistem

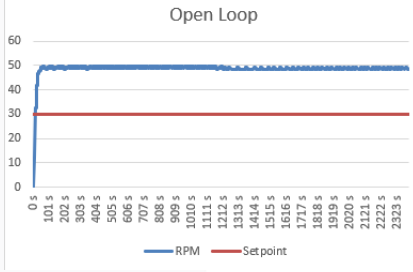
Pengujian sistem dilakukan untuk menilai performa sistem pengaduk cendol otomatis dalam menjaga kecepatan putaran motor sesuai dengan nilai *setpoint*. Pengujian dimulai pada kondisi *open loop* tanpa pengendali PID dengan mencatat data RPM menggunakan sensor *Hall Effect* selama proses pengadukan adonan cendol sebanyak 5 liter pada setpoint 30 RPM. Data yang didapat dari pengujian ini digunakan untuk mendapatkan model *plant* melalui identifikasi sistem di MATLAB sebagai dasar dalam menentukan parameter pengendali.

Setelah itu, model *plant* yang dihasilkan diterapkan dalam metode *Model-Based Tuning* untuk menetapkan nilai Kp, Ki, dan Kd, yang kemudian diimplementasikan pada Arduino Nano untuk pengendalian *closed loop* tanpa beban. Selanjutnya, sistem diuji kembali pada kondisi operasi yang sama tetapi pada hal ini menggunakan beban untuk mengevaluasi kemampuan pengendali dalam mempertahankan stabilitas kecepatan motor. Kinerja sistem dianalisis berdasarkan *rise time*, *settling time*, *overshoot*, dan *steady state error*, sedangkan hasil pengujian dan pembahasannya dijelaskan pada hasil dan pembahasan.

IV. HASIL DAN PEMBAHASAN

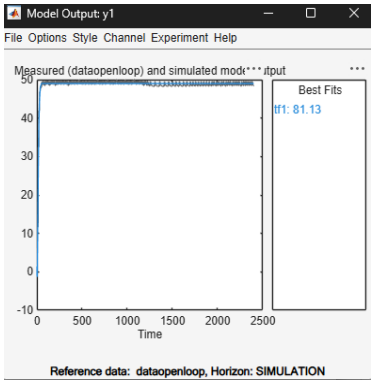
A. Pengujian Sistem *Open Loop*

Pengujian *open loop* dilakukan untuk mengetahui respons awal tanpa adanya pengendali PID. Pengujian ini menggunakan adonan 5 liter dengan durasi 40 menit (2400 detik) dengan motor dioperasikan melalui sinyal PWM tanpa umpan balik.



GAMBAR 4
(Grafik sistem *open loop*)

Gambar 4 menunjukkan bahwa RPM motor mengalami peningkatan yang signifikan pada awal pengujian hingga mencapai kisaran 48-49 RPM, kemudian cenderung stabil meskipun terdapat sedikit variasi. Hasil pengujian ini digunakan sebagai data identifikasi sistem untuk mendapatkan model dan fungsi transfer *plant*.



GAMBAR 5
(Hasil identifikasi pada *software)*

Gambar 5 menunjukkan bahwa didapatkan suatu model sistem dengan nilai *best fit* sebesar 81,13%. Nilai tersebut menunjukkan bahwa model hasil identifikasi berhasil mempresentasikan karakteristik sistem yang sebenarnya berdasarkan data pengujian *open loop* yang telah didapatkan. Hasil identifikasi menghasilkan fungsi transfer, sebagai berikut:

$$G(s) = \frac{-0{,}008956\, s + 0{,}003586}{s^2 + 0{,}2503\, s + 0{,}01867} \quad (4)$$

Berdasarkan persamaan (4) menunjukkan bahwa persamaan tersebut merupakan model matematis plant yang didapatkan dari data respons *open loop* menggunakan MATLAB *System Identification Toolbox*. Model matematis ini disusun berdasarkan bentuk umum fungsi transfer pada persamaan (3) dan direpresentasikan sebagai fungsi transfer orde dua. Pada fungsi transfer ini, pembilang (*numerator*) adalah (-0,008956s + 0,003586), sedangkan penyebut (d*enominator*) adalah (s^2 + 0,2503 s + 0,01867). Fungsi transfer ini menggambarkan hubungan antara sinyal *input* (PWM) dan sinyal *output* (RPM), serta digunakan sebagai model *plant* dalam proses *Model-Based Tuning* untuk memperoleh parameter pengendali PID.

TABEL 1
(HASIL RESPON SISTEM)

Parameter Respon Sistem	**Nilai**
Rise Time (s)	12,9
Settling Time (s)	55,4
Overshoot (%)	6,45
Steady State RPM	48,92
Steady State Error (%)	63,06%

Berdasarkan tabel 1 menunjukkan hasil respon sistem didapatkan *rise time* sebesar 12,9 s, *settling time* sebesar 55,4 s, *overshoot* sebesar 6,45%, serta *steady state error* sebesar 63,06%. Nilai *steady state* RPM menunjukkan bahwa tanpa adanya pengendali, kecepatan putaran motor mengikuti karakteristik alami plant berdasarkan sinyal *input* yang diterima. Sementara itu, *steady state error* yang masih cukup tinggi menunjukkan bahwa sistem belum dapat menjaga kecepatan yang diinginkan. Hasil tersebut menunjukkan bahwa sistem masih memerlukan suatu metode pengendalian agar hasil respons sistem menjadi lebih stabil serta mampu mempertahankan kecepatan putaran motor mendekati nilai *setpoint* yang diinginkan.

B. Penentuan Parameter PID

Perancangan pengendali PID dilakukan melalui metode *Model-Based Tuning* dengan memanfaatkan model *plant* yang diperoleh dari hasil identifikasi sistem dalam keadaan tanpa beban. Tujuannya untuk mendapatkan parameter dasar yang stabil sebelum diuji dengan adonan sebenarnya, didapatkan persamaan sebagai berikut:

$$G(s) = \frac{0{,}0002545}{s^2 + 0{,}04407\, s + 0{,}00136} \quad (5)$$

Berdasarkan persamaan (5) menunjukkan bahwa model matematis *plant* yang digunakan pada simulasi kendali *closed loop* di MATLAB *Simulink*. Model ini didapatkan berdasarkan bentuk umum fungsi transfer pada persamaan (2) dan (3) dan direpresentasikan sebagai *transfer function* orde dua. Model ini dikategorikan sebagai sistem orde dua karena pangkat tertinggi pada *polinomial* penyebut adalah s^2. Pada fungsi transfer ini pembilang adalah (0,0002545), sedangkan penyebut *denominator* adalah ($s^2 + 0{,}04407s + 0{,}00136$).

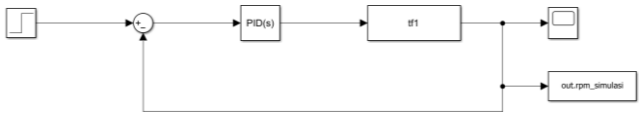
Model matematis yang diperoleh kemudian digunakan sebagai model *plant* pada simulasi sistem kendali *closed loop* di MATLAB *Simulink*. Model ini menjadi dasar proses PID *Tuner* untuk menentukan parameter pengendali Kp, Ki, dan Kd sebelum diterapkan pada sistem *hardware*, sehingga parameter yang diperoleh telah disesuaikan dengan karakteristik *plant* yang didapatkan dari proses identifikasi.

TABEL 2
(HASIL TUNING PARAMETER KENDALI)

Parameter Pengendali	K_p	K_i	K_d
P (merah)	5,675	0	0
PI (biru)	4,085	0,1253	0
PID (hijau)	5,9349	0,1819	46,9086

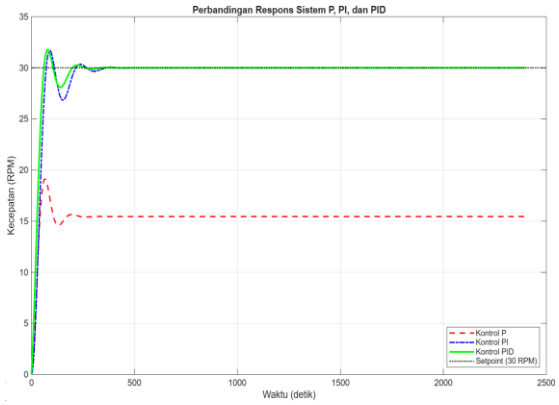
Berdasarkan tabel 2, menunjukkan proses *tuning* dilakukan dengan menggunakan PID *Tuner* di MATLAB untuk memperoleh parameter pengendali yang sesuai dengan karakteristik plant. Pada fase ini dilakukan pengujian terhadap tiga jenis pengendali diantaranya pengendali P, PI, dan PID, sehingga dapat dibandingkan karakteristik respons yang dihasilkan oleh masing-masing pengendali terhadap *setpoin*t yang ditentukan.

Ketiga variasi parameter diatas disimulasikan untuk membandingkan karakteristik respons kecepatannya.



GAMBAR 6
(Hasil simulink)

Gambar 6 menunjukkan model *plant* diimplementasikan ke dalam MATLAB *Simulink* untuk membuat sistem kendali *closed loop*. Model simulasi terdiri dari blok step sebagai masukan *setpoint*, blok PID *Controller* yang berisi parameter yang dihasilkan dari *tuning*, blok LTI *System* sebagai model plant hasil identifikasi sistem yang sudah didapatkan, blok *scope* untuk menampilkan hasil respon sistem, serta blok Out.RPM yang digunakan untuk menyimpan data keluaran berupa kecepatan putaran motor. Melalui model simulasi tersebut, karakteristik respons dari setiap jenis pengendali dapat diamati sebelum diterapkan pada sistem nyata.



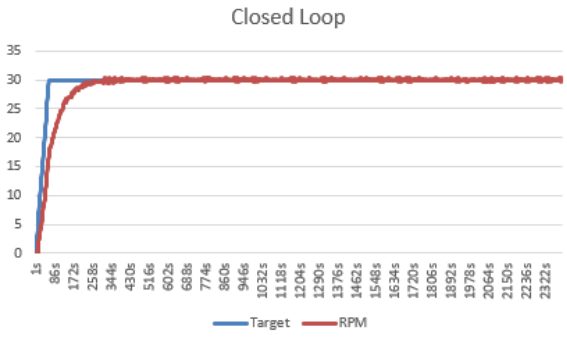
GAMBAR 7
(Perbandingan respons simulasi sistem)

Gambar 7 menunjukkan bahwa pengendali P menghasilkan respons yang sederhana, tetapi masih meninggalkan *error* karena hanya menggunakan aksi proporsional. Pengendali PI dapat mengurangi *steady state error* melalui aksi integral, namun menghasilkan waktu respons yang lebih lambat dibandingkan dengan pengendali PID. Sementara itu, pengendali PID memberikan performa yang paling baik karena menggabungkan aksi proporsional, integral, dan derivatif, sehingga dapat menghasilkan respons yang lebih cepat, mengurangi *steady state error*, serta meningkatkan kestabilan sistem.

Hasil simulasi menunjukkan bahwa parameter pengendali PID dengan Kp = 5,9349, Ki = 0,1819, dan Kd = 46,9086 memberikan karakteristik respons yang lebih baik dibandingkan dengan pengendali P dan PI. Respons sistem dapat mencapai kondisi tunak dengan lebih cepat, memiliki *steady state error* yang mendekati nol, serta menghasilkan respons sistem yang lebih konsisten. Oleh karena itu, parameter PID dari *Model-Based Tuning* dipilih sebagai parameter pengendali yang akan diterapkan dalam tahap sistem kendali pada implementasi hardware.

C. Pengujian Sistem *Closed Loop Hardware*

Pengujian *hardware* dilakukan untuk melihat kinerja sistem kontrol secara langsung ketika mesin menerima beban berupa adonan. Parameter PID yang dimasukkan ke dalam mikrokontroler merupakan hasil tuning sebelumnya, yaitu Kp = 5,9349, Ki = 0,1819, dan Kd = 46,9086, dengan target kecepatan *setpoint* sebesar 30 RPM. Hasil pembacaan kecepatan motor dari pengujian ini ditampilkan pada gambar dibawah ini.

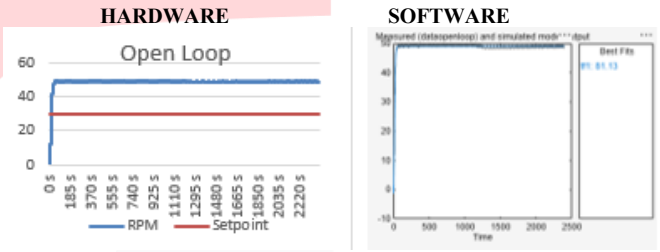


GAMBAR 8
(Hasil respon sistem kecepatan motor pada pengujian *hardware*)

Gambar 8 penerapan sistem kendali PID pada hardware menunjukkan keberhasilan dalam menstabilkan pengaduk dalam kondisi operasional yang nyata. Sistem ini dapat mencapai dan mempertahankan target 30 RPM dengan konsisten tanpa mengalami *overshoot*, yang memungkinkan mesin beroperasi dengan aman dan stabil saat mengelola beban adonan yang kental. Semua jenis fluktuasi kecil yang terlihat masih berada dalam batas toleransi fisik dari mesin dan pembacaan sensor.

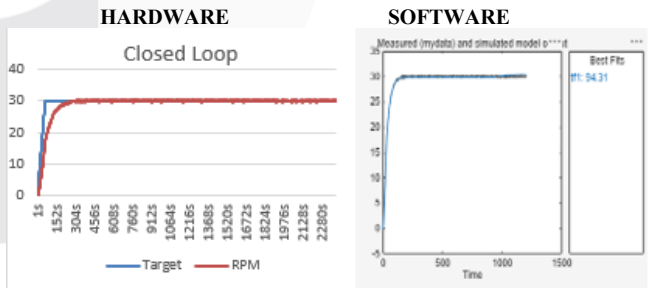
D. Perbandingan Sistem Menggunakan *Software* dan *Hardware*

Pengujian perbandingan antara *software* dan *hardware* dilakukan untuk menilai kecocokan respons sistem hasil simulasi terhadap implementasi pada kondisi nyata (*hardware*) pada kondisi *open loop* atau *closed loop*. Tujuan dari perbandingan ini untuk mengetahui kemampuan model matematis hasil identifikasi sistem dalam merepresentasikan karakteristik *plant* serta menilai performa pengendali PID yang diperoleh menggunakan metode *Model-Based Tuning*.



GAMBAR 9
(Peerbandingan hasil tuning software dan hardware open loop)

Gambar 9 grafik *open loop* didapatkan hasil hardware menunjukka bahwa RPM aktual tetap stabil di sekitar 50 RPM, tetapi terdapat osilasi kecil yang diakibatkan oleh gangguan sensor dan sifat alami motor AC. Hasil *software* menunjukkan tingkat kesesuian *best fit* sebesar 81,13% yang menghasilkan model matematis yang diidentifikasi cukup mewakili sistem nyata, sehingga model ini sesuai digunakan sebagai dasar dalam merancang kontroler PID.



GAMBAR 10
(Perbandingan hasil tuning software dan hardware closed loop)

Gambar 10 grafik *closed loop*, diperoleh hasil dari hardware menunjukkan bahwa RPM actual dapat mengikuti nilai target 30 RPM dengan sempurna, Dimana respons sistem meningkat secara perlahan, dan kemudian stabil, tumpeng tindih dengan garis target, tanpa ada peningkatan melebihi batas yang signifikan. Parameter PID pada *hardware* berhasil bekerja dengan baik. Dalam simulasi *software*, sistem menampilkan respons yang hampir sesuai, dengan *best fit* sebesar 94,31% yang merupakan peningkatan yang cukup besar dibandingkan dengan kondisi *open loop*.

Nilai tersebut mengindikasikan bahwa model *software* sangat mendekati sistem *closed loop*, sehingga dapat disimpulkan bahwa kendali PID berbasis *Model-Based Tuning* yang dilakukan berhasil menghasilkan sistem kendali yang stabil.

TABEL 3
(Perbandingan analisis respons sistem)

Perbandingan Analisis Respon Sistem					
Metode Parameter PID	***Rise Time* (s)**	***Settling Time* (s)**	***Overshoot* (%)**	***Error Steady State* (%)**	***Steady State RPM***
Open Loop (Software)	12,9	55,4	6,45	63,06	48,92
Open Loop (Hardware)	73	146	1,39	63,33	49
Nilai *Error Open Loop* RPM(%)					0,16
Closed Loop (Software)	7,19	18,8	2,57	1,67	29,5
Closed Loop (Hardware)	112	280	1,00	0	30
Nilai *Error Closed Loop* RPM(%)					1,69
Rata-rata *error* (%)					0,93
Akurasi (%)					99,07

Berdasarkan tabel 3, menunjukkan perbandingan respons sistem antara hasil *software* dan *hardware* pada *kondisi open loop* dan *closed loop*. Parameter yang dibandingkan meliputi *rise time*, *settling time*, *overshoot*, *steady state error*, dan *steady state* RPM. Perhitungan nilai pada *error* menggunakan persamaan (6), dilakukan berdasarkan parameter *steady state* RPM, dengan hasil nilai *error* sebesar 0,16% pada kondisi *open loop* dan hasil *error* 1,69% pada kondisi *closed loop*. Berdasarkan kedua nilai tersebut didapatkan rata-rata *error* dengan persamaan (7) sebesar 0,93%, dengan hasil tingkat akurasi pada persamaan (8) sebesar 99,07% yang menunjukkan bahwa hasil *hardware* memiliki tingkat yang baik terhadap hasil *software* pada kondisi *steady state*.

$$Nilai\ Error = \left|\frac{hardware - software}{software}\right| \times 100\% \quad (6)$$

$$Rata - rata\ Error = \frac{Error\ open\ loop - Error\ close}{2} \quad (7)$$

$$Akurasi = 100 - rata\ rata\ error \quad (8)$$

Pada kondisi *open loop* didapatkan nilai *overshoot* sebesar 6,45% pada *software* dan 1,39% pada *hardware*, dengan *steady state error* masing-masing sebesar 63,06% dan 63,33%. Sedangkan dalam kondisi *closed loop*, nilai *overshoot* sebesar 2,57% pada *software* dan 1,00% pada *hardware*, sedangkan *steady state error* sebesar 1,67% pada simulasi dan 0% pada implementasi *hardware*. Penurunan nilai *overshoot* dan *steady state error* dari *open loop* ke *closed loop* menunjukkan bahwa pengendali PID berhasil meningkatkan karakteristik respons sistem dengan mengurangi lonjakan serta memperbaiki kemampuan sistem dalam mencapai dan menajaga kecepatan putaran sesuai setpoint 30 RPM.

Meskipun terdapat perbedaan nilai *rise time*, *settling time*, dan *overshoot* antara hasil simulasi dan *hardware*, perbedaan ini merupakan hal yang wajar karena model matematis merupakan pendekatan terhadap sistem nyata sehingga belum mampu merepresentasikan seluruh karakteristik fisik sistem secara sempurna, seperti pengaruh gesekan mekanik, inersia motor dan *gearbox*, respon *inverter*, toleransi komponen, serta ketidakpastian pembacaan sensor. Namun, hasil pengujian menunjukkan bahwa parameter PID yang didapatkan melalui metode *Model-Based Tuning* tetap mampu menghasilkan respon sistem yang stabil dan mempertahankan kecepatan putaran mendekati *setpoint*. Hal ini diperkuat oleh nilai *best fit* yang tinggi serta rata-rata *error* sebesar 0,93% pada parameter *steady state* RPM, sehingga model matematis yang digunakan dapat dinyatakan cukup valid sebagai dasar dalam merancang pengendali PID pada sistem pengaduk otomatis.

V. KESIMPULAN

Berdasarkan hasil perancangan, pengujian, dan analisis yang telah dilakukan, dapat disimpulkan bahwa sistem pengaduk cendol otomatis yang menggunakan motor induksi tiga fasa dengan kendali PID melalui metode *Model-Based Tuning* telah berhasil dirancang dan diimplementasikan dengan baik. Proses identifikasi sistem dalam keadaan *open loop* menghasilkan model matematis berupa fungsi transfer dengan nilai *best fit* sebesr 81,13% yang menunjukkan bahwa model dapat merepresentasikan karakteristik sistem aktual dengan tingkat kesesuaian yang baik. Model tersebut kemudian digunakan dalam proses *Model-Based Tuning* untuk mendapatkan parameter pengendali PID, yaitu Kp = 5,9349, Ki = 0,1819, dan Kd = 46,9086. Hasil dari implementasi menunjukkan bahwa pengendali PID dapat menjaga kecepatan motor mendekati setpoint 30 RPM pada kondisi *closed loop* dengan beban 5 liter, serta menghasilkan respons sistem yang lebih stabil dengan pengurangan *overshoot* dan *steady state error* dibandingkan pada kondisi *open loop*. Selain itu, hasil perbandingan antara *software* dan implementasi *hardware* menunjukkan tingkat kesesuaian yang baik, dengan rata-rata *error* sebesar 0,93% dan akurasi mencapai 99,07%, sehingga metode *Model-Based Tuning* terbukti efektif dalam meningkatkan kestabilan sistem pengendalian kecepatan pada pengaduk cendol otomatis.

REFERENSI

[1] D. Dewita, S. W. Sidauruk, D. Desmelati, M. Syahda, and T. Hidayat, “Karakteristik fisikokimia dan umur simpan cendol sagu instan dengan variasi kemasan dan metode pengeringan,” *J. Pengolah. Has. Perikan. Indones.*, vol. 28, no. 4, pp. 336–348, 2025, doi: 10.17844/jphpi.v28i4.62300.

[2] D. Fitriani and Z. Amalia Dina, "Penggunaan Tepung Umbi Garut Sebagai Pengganti Tepung Tapioka Dalam Pembuatan Es Cendol," *J. Pariwisata Vokasi*, vol. Vol. 5, no. 2, pp. 30–44, 2024.

[3] I. G. Utama Ningrat Wiratna Ayu Agung, N. P. Atmaja Dharmadewi Cempaka, and N. M. Mentari Indah, "Pengembangan Produk dan Promosi Digital Pada UMKM Produk Cendol Di Desa Peguyangan," Denpasar, 2021.

[4] R. A. Yasmin, W. D. Ayu, and L. Rijai, "Rancang Bangun Mesin Pengaduk Dodol Kapasitas 5Kg," *Pros. Semin. Nas. Kefarmasian Ke-3*, pp. 125–128, 2024.

[5] B. Artono, D. Prakoso Nur, and T. Lestari Widya Annisa, "Sistem Monitoring Mesin Pengaduk Adonan Roti Otomatis Berbasis Internet of Things (IoT)," *J. Comput. Sci. Eng.*, vol. 4, no. 1, pp. 25–41, Feb. 2023, doi: 10.36596/jcse.v1i1.4.

[6] J. Wilewska, W. Orciuch, A. Dudała, P. Gierycz, and Ł. Makowski, "Power Consumption and Mixing Intensity of Jet Flow Mixer in Industrial Tank," *Energies*, vol. 18, no. 15, pp. 1–19, Jul. 2025, doi: 10.3390/en18153975.

[7] M. Diah Ika Putri, A. Ma'arif, and R. Dwi Puriyanto, "Pengendali Kecepatan Sudut Motor DC Menggunakan Kontrol PID dan Tuning Ziegler Nichols," *Techno (Jurnal Fak. Tek. Univ. Muhammadiyah Purwokerto)*, vol. 23, no. 1, pp. 09–18, 2022, doi: 10.30595/techno.v23i1.10773.

[8] J. A. Niembro-Ceceña, R. A. Gómez-Loenzo, J. Rodríguez-Reséndiz, O. Rodríguez-Abreo, and Á. Odry, "Auto-Regression Model-Based Off-Line PID Controller Tuning: An Adaptive Strategy for DC Motor Control," *Micromachines*, vol. 13, no. 8, pp. 1–21, Aug. 2022, doi: 10.3390/mi13081264.

[9] B. Triyono, R. Fadilah, T. Tohir, and Supriyanto, "Implementasi Sistem Kendali Kecepatan Motor DC Berbasis PID Ziegler-Nichols Pada Alat Pengaduk Cairan Viskos," Bandung, Jul. 2023.

[10] A. Kuniawan, H. Istiqlaliyah, and H. Mahmudi, "Rangkaian Kelistrikan Pada Mesin Pengaduk Petis Kapasitas 15 Kg / Jam," *Prosidig Semnas INOTEK*, vol. 9, pp. 1359–1365, 2025.

[11] E. Gunawan Indra, M. Ali, and H. Nurohmah, "Optimasi Thermal Oil Heater Menggunakan PSO Sebagai Tunning PID Controller," *Nucleus*, vol. 1, no. 2, pp. 53–64, 2022.

[12] M. Suhaimi Naufal, A. Murtono, and Fathoni, "Sistem Mesin Roasting Kopi Guna Peningkatan Kualitas Produk Petani Berbasis Mikrokontroller Dengan Metode PID," *Elkolind*, vol. 9, no. 3, pp. 191–198, Sep. 2022, doi: 10.33795/elkolind.v9i3/403.

[13] D. Vrančić, P. Moura Oliveira, P. Bisták, and M. Huba, "Model-Free VRFT-Based Tuning Method for PID Controllers," *Mathematics*, vol. 11, no. 3, pp. 1–29, Feb. 2023, doi: 10.3390/math11030715.

[14] A. Rahman Hidayat, A. Dwi Risdhayanti, Fitri, D. Ayu Permatasari, and W. Tri Wahono, "Pemodelan Identifikasi Sistem untuk Pengaturan Kecepatan Motor DC dengan Kontrol PID," *J. Elektron. dan Otomasi Ind.*, vol. 11, no. 1, pp. 253–262, 2024, doi: 10.33795/elkolind.v11i1.5409.

[15] M. K. Munfiqoh and D. Aribowo, "Pengendalian Kecepatan Motor Induksi Tiga Fasa Menggunakan Variable Frequency-Drive (VFD) Untuk Mendeteksi Aliran dan Tekanan Air Pada Modul Pumps Training System PT. Festo Indonesia," *J. Sains dan Teknol.*, vol. 1, no. 2, pp. 24–34, Dec. 2022.

[16] F. Hazrina, R. P. Dewi, B. Widianingsih, L. Sari, and M. Zulfahmi Muassar, "Kontrol Kecepatan Berbasis PWM (Pulse Width Modulation) Untuk Mesin Pemarut Kelapa Bertenaga Surya," vol. 16, no. 02, pp. 485–492, Jul. 2025, doi: 10.35970/infotekmesin.v16i2.2794.

[17] M. Rakha Firdaus, T. Arif Berbudi, S. Nurrahma, G. Izzaulhaq, and I. Hudati, "Identifikasi Sistem Motor DC dan Penerapan Kendali PID, LQR, dan Servo Tipe 1 Berbasis Arduino-MATLAB," *J. List. Instrum. dan Elektron. Terap.*, vol. 4, no. 1, pp. 1–9, Apr. 2023.