ISSN Cetak: 2087-4286; ISSN On Line: 2580-6017

Unjuk Kerja Metode Imperalistt Competitive Alghorithm (ICA)Dalam Mengoptimalkan Kecepatan Motor DC

Yanuangga Gala Hartlambang[(1)], Machrus Ali[(2)], Agus Raikhani[(3)]
yanuangga@gmail.com, machrus7@gmail.com, agus.raikhani@gmail.com
[(1)]*Teknik Informatika, Universitas Darul 'Ulum, Jombang, Indonesia*
[(2,3)]*Teknik Elektro, Universitas Darul 'Ulum, Jombang, Indonesia*

ABSTRAK

Imperialist Competitive Algorithm (ICA) merupakan algoritma evolusioner yang terinspirasi dengan kompetisi kekuasaan (imperialist competitive). Algoritma optimasi ICA dikenalkan oleh Esmaeil Atashpaz dan pada tahun 2007. ICA mensimulasikan proses sosial politik dari imperialisme dan kompetisi kekuasaan. Pada Metode ICA ini, seperti algoritma evolusioner lainnya yaitu dengan dimulai dengan inisialisasi populasi awal. Setiap individu dari populasi disebut dengan negara (country). Beberapa negara terbaik dipilh sebagai negara penjajah dan sisanya membentuk koloni yang digunakan oleh penjajah. Negara imperialis bersama-sama dengan koloni yang dimilikinya membentuk beberapa empire (kerajaan). Setelah membentuk empire awal, koloni pada setiap empire bergerak menuju negara imperialis yang relevan. Pergerakan ini adalah model sederhana dari kebijakan asimilasi yang diberikan oleh negara imperialis. Total kekuatan dari sebuah empire tergantung pada kekuatan dari negara imperialis dan kekuatan dari koloninya. Fakta ini dimodelkan dengan mendefinisikan total kekuatan dari sebuah empire merupakan kekuatan dari negara imperialis ditambah dengan presentase dari rata-rata kekuatan koloninya. Penggunaan metode ini didasarkan pada pertimbangan bahwa, ICA merupakan jenis metode optimisasi yang sederhana, kemampuan mencapai konvergensi yang cepat, dan menghasilkan solusi yang baik.

Ketidaklinearan dari motor DC akan mempersulit dalam aplikasi yang memerlukan kecepatan kontrol secara otomatis. Sayangnya, non linear model dinamik dari motor DC memiliki keterbatasan pada desain dari ragkaian close-loop feedback kontroler. Karakteristik non linear dari motor DC seperti saturasi dan gesekan dapat menurunkan kinerja dari konvensional Kontrol. Pemodelan system pengaturan motor DC harus disesuaikan dengan karakteristik motor DC dan model pengaturannya. Metode kontrol Proporsional-Integral-Derivative (PID) banyak diterapkan di bidang industri. Kontroler ini memiliki parameter-parameter pengontrol, yaitu Kp, Ki, dan Kd. Ketiga parameter tersebut diturunkan dari perhitungan matematis pada metode PID konvensional. Metode osilasi Ziegler-Nichols merupakan sebuah metode penalaan PID yang dapat dilakukan secara otomatis tanpa memodelkan sistem. Sistem kontrol kecapatan motor DC yang dianggap paling baik adalah kontrol PID-ICA, kemudian PID-ZN, kemudian PID dan terakhir Nonkontrol. Hasil running program didapatkan nilai; overshoot tanpa kontrol 0 dengan settling time 7,634 detik, overshoot PID standart 1,513 dengan settling time 10 detik, overshoot PID-ZN 1,495 dengan settling time 2,023 detik, overshoot PID-ICA 1,103 dengan settling time 1,32 detik.

.

Kata kunci : ICA, PID, DC Motor

I. PENDAHULUAN

Kontrol cerdas berbasis Artificial Intelligent (AI) sudah banyak berkembang untuk memperbaiki kontrol konvensional. Oleh sebab itu pada tugas penelitian ini akan mendesain model kontrol motor DC menggunakan kontrol PID untuk mengontrol kecepatan motor DC. Kemudian akan menguji kedua kontrol tersebut pada sebuah model motor DC dengan perubahan kecepatan dan perubahan torsi beban. Hasil performance dari Model kontrol DC Series menggunkan PID dengan perubahan kecepatan dan perubahan beban torsi beban didapatkan memiliki steady state error, settling time dan overshoot yang lebih baik. Motor DC telah banyak digunakan dalam industri meskipun biaya pemeliharaannya lebih tinggi dari motor induksi. Proporsional-Integral-Derivative (PID) kontroler telah banyak digunakan untuk kecepatan dan posisi kontrol motor DC. Pencapaian jurnal digunakan untuk merancang sistem kontrol menggunakan kontroler *Ziegler-Nichols* dan *Iperalis Competitive Algorithm* dengan mempertimbangkan non linearitas yang efektif dari sistem. Dengan membandingkan metode ICA dan tanpa tuning system yang akan didapat dilihat hasil pencapaian pengoptimalan masing-masing metode.

II. IMPERIALIST COMPETITIVE ALGORITHM

Imperialisme merupakan kebijakan dalam memperluas kekuatan dan aturan pada pemerintahan diluar teritorialnya. Suatu negara berusaha mendominasi negara yang lainnya dengan aturan secara langsung atau dengan cara yang kurang begitu jelas seperti pengontrolan pasar-pasar barang atau bahan-bahan mentah. Pengontrolan pasar bahan mentah disebut neokolonisme. Pada awalnya, imperialisme hanya sebagai kontrol politik atas negara-negara lain dengan tujuan untuk menggunakan sumber daya yang dimiliki negara lain. Pada kasus-kasus tertentu, alasan mengontrol negara lain hanyalah mencegah penjajah musuh dari penguasaan negara itu. Apapun alasannya itu, negara penjajah akan berkompetisi untuk meningkatkan jumlah jajahannya dan menyebarkan kekuasaannya di seluruh dunia. Kompetisi ini mengakibatkan perkembangan bagi kerajaan yang kuat dan keruntuhan bagi kerajaan yang lemah.

Imperialisme telah merubah sikap publik menuju peradaban barat selama abad ke-19 dan ke-20. Pemahaman sosial Darwin menafsirkan bahwa budaya barat lebih tinggi dibandingkan budaya timur. Imperialisme mempertimbangkan perang salib sebagai hasil dari sikap. Kemudian sepanjang semua kesulitan ini, imperialisme membuat negara penjajah memulai dengan mengembangkan jajahannya (menyebarkan budayanya). Sebagai contoh, di pertengahan abad ke-18, dua penjajah bermusuhan, perancis dan inggris saling berkompetisi untuk menguasai India yang merupakan sebagai bagian dari ambisi imperialisisnya untuk menguasai seluruh dunia. Akhirnya yang dapat menguasai India adalah Inggris. Setelah mententramkan negara ini, inggris mulai membangun sekolah bahasa inggris, jalan, rel kereta, dan jalur telegraf. Inggris juga mencoba untuk merubah kepercayaan sosial dan adat yang dianggap salah jika dibandingkan dengan budaya barat. Budaya-budaya yang diperbaiki tersebut termasuk adat pembakaran diri yang diikuti oleh janda India sebagai tanda kesetiaan untuk suaminya. Mereka juga menggalakkan pernikahan usia dini pada anak perempuannya. Inggris membuat perubahan yang sama pada Malaya dengan menghapuskan perbudakan dan pajak sewenang-wenang dengan membuat system yang baru dalam pemeliharaan kesehatan. Indochina adalah contoh yang lain. Indochina merupakan jajahan Perancis. Perancis tertarik Indochina karena sumber

daya alamnya dan untuk mencegah Inggris dalam meningkatkan kekuatannya. Ini juga merupakan tempat yang baik untuk pengabar injil untuk menarik masuk orang-orang agar beragama kristen. Berdasarkan kebijakan asimilasi, Perancis bermaksud untuk membangun Perancis baru di Indochina melalui bangunan sekolah bahasa Perancis untuk memperluas bahasa dan budayanya. Walaupun kebijakan tersebut tidak berhasil dalam meningkatkan kekuasaan atas jajahannya, dan jajahan meminta untuk otonomi daerahnya, mereka membawa perkembangan sosial dan politik yang cepat untuk jajahannya.

Metode berbasis *AI* telah ditemukan dan saat ini dikembangkan adalah *Imperialist Competitive Algorithm* (ICA), yang merupakan jenis metode optimisasi yang terinspirasi dari pola kompetisi kekuasaan (*Imperialist Competitive*) suatu Negara atau kerajaan yang saling berkompetisi atau menjajah.

2.1. Operasi *Imperialist Competitive Algorithm*

Imperialist Competitive Algorithm (ICA) merupakan algoritma evolusioner yang terinspirasi dengan kompetisi kekuasaan (*imperialist competitive*) [Atashpaz, 2007]. Algoritma optimasi ICA dikenalkan oleh Esmaeil Atashpaz dan pada tahun 2007. ICA mensimulasikan proses sosial politik dari imperialisme dan kompetisi kekuasaan. Pada Metode *ICA* ini, seperti algoritma evolusioner lainnya yaitu dengan dimulai dengan inisialisasi populasi awal. Setiap individu dari populasi disebut dengan negara (*country*). Beberapa negara terbaik dipilh sebagai negara penjajah dan sisanya membentuk koloni yang digunakan oleh penjajah. Negara imperialis bersama-sama dengan koloni yang dimilikinya membentuk beberapa *empire* (kerajaan). Setelah membentuk empire awal, koloni pada setiap empire bergerak menuju negara imperialis yang relevan. Pergerakan ini adalah model sederhana dari kebijakan asimilasi yang diberikan oleh negara imperialis. Total kekuatan dari sebuah empire tergantung pada kekuatan dari negara imperialis dan kekuatan dari koloninya. Fakta ini dimodelkan dengan mendefinisikan total kekuatan dari sebuah empire merupakan kekuatan dari negara imperialis ditambah dengan presentase dari rata-rata kekuatan koloninya. Penggunaan metode ini didasarkan pada pertimbangan bahwa, ICA merupakan jenis metode optimisasi yang sederhana, kemampuan mencapai konvergensi yang cepat, dan menghasilkan solusi yang baik.

2.2. Inisialisasi Kerajaan (*Empire*)

Dalam ICA populasi awal disebut "*country*", *cost country* diperoleh dari fungsi *objective*. Dari sejumlah country akan dipilih beberapa penjajah (*imperialist*) yang dianggap kuat untuk memimpin *empire*. Tujuan akhir dari optimisasi adalah mendaptakan solusi optimal untuk suatu permaslahan tertentu. ICA membentuk sebuah array dari nilai variabel yang akan dioptimisasi. Pada algoritma lain, seperti GA, array ini disebut dengan kromosom, maka di ICA ada istilah negara atau "*country"*. Sebuah negara (*country*) adalah 1 x *Nvar* array. Array ini didefinisikan sebagai berikut.

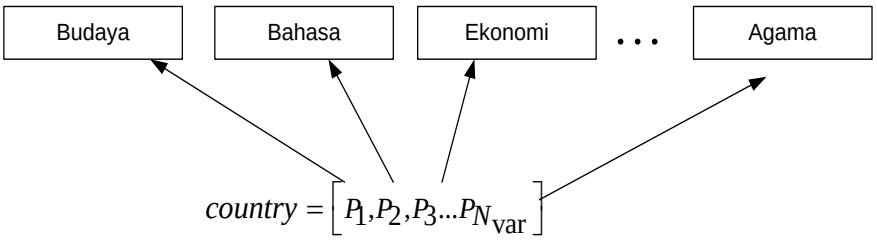
$$country = (\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \ldots \mathrm{P}_{\mathrm{N}_{\mathrm{var}}}) \qquad (2.1)$$

Dengan P_i adalah variabel yang akan dioptimisasi. Setiap variable dalam suatu negara dapat diinterpretasikan sebagai karakterstik sosio-politik dari sebuah negara. Dari sudut pandang ini, semua algoritma akan melakukan pencarian untuk negara yang terbaik

dimana negara ini memiliki kombinasi dari karakteristik sosial politik seperti budaya, bahasa, kebijakan politik, maupun agama. Dari sisi optimisasi, hal ini akan memicu penemuan solusi optimal dari permasalahan, solusi dengan nilai cost terbaik. Gambar 3.1 menunjukkan penggambaran dari negara dengan menggunakan beberapa karakteristik sosial politik. Cost dari sebuah negara didapatkan dengan mengevaluasi cost function *f* pada variabel (P_1,P_2,P_3,...,P_{Nvar}). Sehingga kita mendapatkan

$$cost = f(country) = f\left(\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \ldots \mathrm{P}_{\mathrm{N_{var}}}\right) \quad (2.2)$$

Untuk memulai optimisasi, negara awal yang berukuran N_{negara} dibentuk terlebih dahulu. Beberapa negara yang terbaik akan dipilih sebagai penjajah atau imperialis (*imperialist*) untuk memimpin sebuah *empire*. Sisa dari populasi akan membentuk jajahan atau koloni *(colony)* yang dimiliki oleh empire. Sebuah *empire* akan terdiri dari satu imperialis dan beberapa koloni.



Gambar 2.1 Kandidat Solusi Dari Permasalahan (Negara)

Pembagian koloni harus didasarkan kekuatan dari imperialis. Untuk membagi koloni berdasarkan imperialis dengan tepat, maka cost imperialis harus dinormalisasi terlebih dahulu dengan persamaan berikut.

$$C_n = c_n - \max\{c_\mathrm{i}\} \quad (2.3)$$

dengan C_n adalah cost yang sudah dinormalisasi dan c_n merupakan cost dari imperialis ke-*n*.

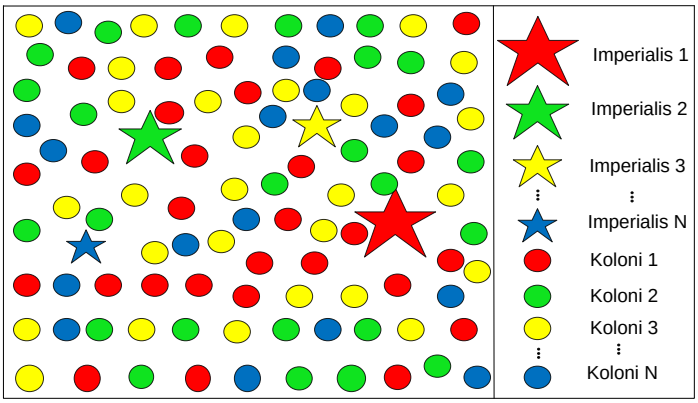
Dan kekuatan masing-masing imperialis didefinisikan sebagai berikut,

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \right| \quad (2.4)$$

kemudian jumlah koloni awal untuk sebuah empire ke-*n* adalah

$$N.C._n = round\{P_n.N_{col}\} \quad (2.5)$$

Dengan $N.C._n$ adalah jumlah awal koloni dari empire ke-*n* dan N_{col} merupakan jumlah koloni awal. Untuk membagi koloni-koloni, $N.C._n$ dari koloni secara random dipilih dan diberikan pada imerialist ke-*n*. Koloni tersebut dengan imperialis ke-*n* akan membentuk empire ke-*n*. Gambar 2.2 menunjukkan *empire* awal. Seperti yang ditunjukkan pada gambar tersebut, bahwa semakin besar *empire* semakin banyak koloni yang dimiliki. Pada gambar, imperialis 1 membentuk *empire* terkuat dan secara otomatis juga memiliki koloni terbanyak.



Gambar 2.2 *Empire* awal

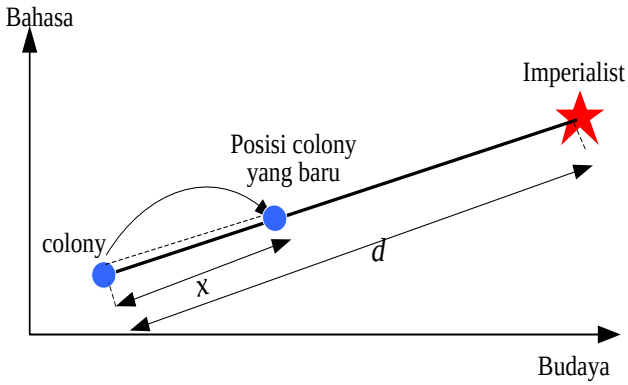
2.3. Pergerakan Koloni dari Sebuah *Empire* Menuju *Imperialist*

Asimilasi adalah proses dimana kelompok minoritas dengan cepat beradaptasi untuk menjadi sebuah kelompok yang memiliki budaya yang kuat. Kebijakan asimilasi membuat negara-negara imperialis mencoba untuk mendekatkan koloninya dan membuat menjadi bagian dari negaranya. Lebih tepatnya, negara imperialis membuat koloninya bergerak menuju dirinya sendiri. Gambar 2.4 menunjukkan pergerakan dari koloni menuju imperialisnya. Dengan memperhitungkan permasalahan optimisasi 2 dimensi, koloni di tarik oleh imperialis pada sumbu budaya dan bahasa. Koloni akan menjadi lebih dekat dengan imperialis pada sumbu tersebut. Pergerakan ini apabila dilanjutkan terus menerus maka akan membuat semua koloni akan berpindah menuju imperialis.

Dalam ICA, kebijakan asimilasi dimodelkan dengan menggerakkan semua koloni menuju imperialis. Pergerakan ini ditunjukkan oleh Gambar 2.3. Dimana sebuah koloni bergerak menuju imperialis sebesar x unit. Posisi baru koloni ditunjukkan dengan warna yang lebih gelap. Arah dari pergerakan adalah vektor dari koloni ke imperialis. Pada Gambar ini, x adalah variabel random yang terdistribusi seragam.

$$x \sim U(0, \beta x d) \tag{2.6}$$

Nilai β adalah sebuah angka yang lebih dari 1 sehingga membuat koloni bergerak lebih dekat dengan imperialisnya dari kedua sisi dan d adalah jarak antara koloni dan negara imperialis.

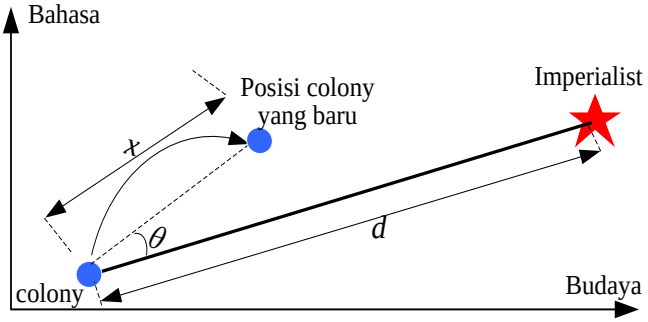


Gambar 2.3 Pergerakan Koloni Menuju Imperialis

Asimilasi koloni oleh negara imperialis tidak menghasilkan pergerakan yang secara langsung menuju imperialis. Artinya, arah pergerakan belum tentu vektor dari koloni ke imperialis. Untuk memodelkan kenyataan ini jumlah acak penyimpangan ditambahkan ke arah gerakan tujuannya adalah untuk meningkatan kemampuan daerah pencarian di sekitar negara imperialis. Gambar 2.4 menunjukkan arah yang baru. Pada gambar ini θ adalah parameter yang terdistribusi seragam.

$$\theta \sim U(-\gamma, \gamma) \tag{2.7}$$

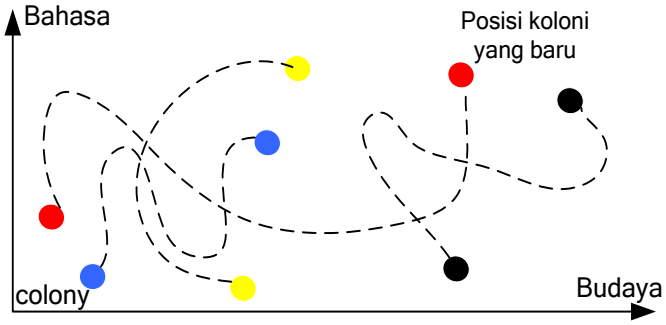
Dengan, γ adalah parameter yang mengatur penyimpangan dari arah awal. Namun, nilai β dan γ tidak dipilih sembarangan, dalam sebagian besar impelementasi, nilai β sekitar 2 dan nilai γ sekitar $\pi/4$ (rad) untuk menghasilkan konvergensi yang baik untuk menuju global minimum.



Gambar 2.4 Pergerakan Koloni Menuju Imperialis dalam Penyimpangan Acak

2.4. Revolusi

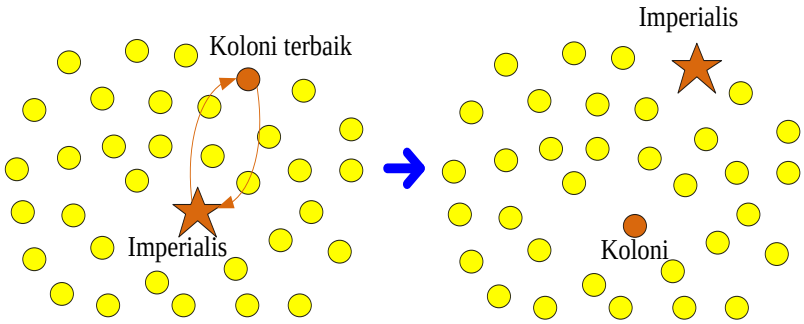
Revolusi adalah perubahan dasar pada struktur organisasi yang mengambil tempat secara relatif pada periode waktu. Pada terminologi ICA, revolusi menyebabkan sebuah negara tiba-tiba berubah karakteristik sosial politiknya. Artinya, walaupun sudah diasimilasi oleh imperialis, koloni secara acak merubah posisinya pada sumbu sosial politik. Gambar 2.5 menunjukkan revolusi pada sumbu bahasa-budaya. Revolusi meningkatkan eksplorasi dari algoritma dan mencegah konvergensi negara menuju lokal minimum. Kecepatan revolusi pada algoritma menunjukkan persentase koloni-koloni pada setiap koloni yang akan merubah posisinya secara random. Nilai revolusi yang sangat tinggi menurunkan kekuatan eksploitasi algoritma dan dapat mengurangi kecepatan konvergensi. Pada simulasi penelitian ini, kecepatan revolusi adalah 0,3. Artinya, 30% dari koloni dalam *empire* akan merubah posisinya secara random.



Gambar 2.5 Revolusi ; Sebuah Perubahan Tiba-tiba dalam Karakteristik Sosial Politik Sebuah Negara

2.5. Pertukaran Posisi Antara *Imperialist* dan Sebuah Koloni

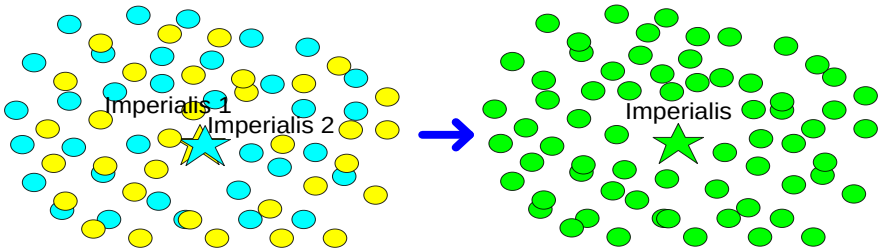
Ketika koloni bergerak menuju *imperialist*, sebuah koloni mungkin bisa memiliki cost yang lebih baik daripada yang dimiliki *imperialist*nya. Ketika hal ini terjadi maka akan terjadi pertukaran antara *imperialist* dengan koloni. Sehingga ICA akan melanjutkan dengan *imperialist* yang baru dan koloni-koloninya bergerak menuju posisi *imperialist* yang baru tersebut. Gambar 2.6 menunjukkan pertukaran posisi imperialis dan koloni.



Gambar 2.6 Pertukaran Posisi Antara *Imperialist* dan Sebuah *Colony*

2.6. Penggabungan empire yang sama

Pada pergerakan koloni dan imperialis menuju glonbl minimum, beberapa imperialis mungkin akan bergerak ke posisi yang sama. Jika jarak antara dua imperialis kurang dari jarak *treshold*, maka keduanya akan membentuk *empire* yang baru dan imperialis baru pada posisi dimana kedua imperialis itu bertemu. Gambar 2.7 menunjukkan proses penggabungan dari dua empire.



Gambar 2.7 Penggabungan Dua *Empire* yang Memiliki Posisi Sama

2.7. Perhitungan Kekuatan Total dari Sebuah *Empire*

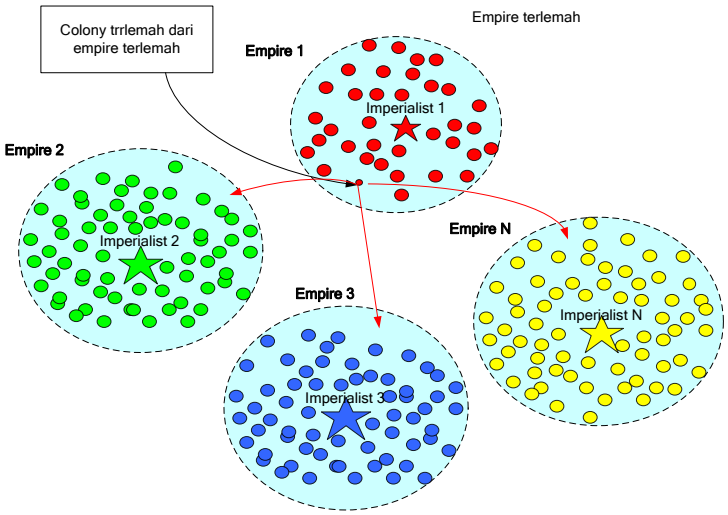
Kekuatan total dari sebuah *empire* sangat ditentukan oeh kekuatan dari negara *imperialist*. Akan tetapi kekuatan dari koloni juga memiliki pengaruh walaupun kecil. Total *cost* dari sebuah *empire* didefinisikan sebagai jumlah antara *cost imperialist* dengan rata-rata *cost* koloni-koloni yang dimiliki *imperialist* dari satu *empire*. Namun nilai rata-rata biaya koloni yang dimiliki suatu *empire* dipengaruhi oleh nilai ξ yang menunjukkan pengaruh kontribusi dari *koloni*.

$$T.C._n = \cos t(imperialist_n) + \xi\, mean\{Cost(colonies\ of\ empire_n)\} \quad (2.8)$$

Dengan $T.C._n$ adalah total *cost* dari *empire* ke-n dan ξ adalah nilai positif dengan nilai yang dianggap kurang dari 1, nilai yang kecil, sehingga menyebabkan kekuatan total *empire* lebih dipengaruhi oleh imperialis daripada koloni. Nilai 0,01 untuk ξ telah menunjukkan hasil yang bagus untuk implementasi.

2.8. Kompetisi kekuasaan (Imperialist Competition)

Semua *empire* berusaha untuk memiliki koloni dari empire yang lain dan menguasai mereka. Kompetisi kekuasaan secara berangsur menurunkan kekuatan dari *empire* yang lemah dan meningkatkan kekuatan *empire* yang kuat. Kompetisi ini dimodelkan dengan hanya mengambil beberapa atau satu koloni terlemah yang dimiliki oleh *empire* yang terlemah diantara semua *empire* dan membuat kompetisi antara semua *empire-empire* untuk memiliki koloni tersebut. Gambar 2.8 menunjukkan pemodelan kompetisi kekuasaan.



Gambar 2.8 Kompetisi Kekuasaan

Berdasarkan total kekuatan, pada kompetisi ini, setiap *empire* akan memiliki kemungkinan mengambil koloni tersebut. dengan kata lain, koloni belum tentu akan dimiliki oleh empire yang paling kuat, namun empire tersebut memiliki peluang yang lebih besar untuk memiliki koloni.

Untuk memulai kompetisi, maka terlebih dahulu mencari peluang kepemilikan dari setiap empire berdasarkan pada total kekuatan masing-masing empire. Peluang

kepemilikan P_p sebanding dengan toal kekuatan yang dimiliki *empire*. Normalisasi total *cost* dari sebuah *empire* secara sederhana didapatkan dengan,

$$N.T.C._n = T.C._n - \max\{T.C._i\} \quad (2.9)$$

Dimana *T.C.n* dan *N.T.C.n* adalah total *cost* dan total *cost* yang telah dinormalisisai dari *empire* ke-*n*, secara berurutan. Setelah total *cost* dinormalisasi, peluang kepemilikan dari tiap *empire* diberikan sebagai berikut.

$$p_{p_n} = \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{imp}} N.T.C._i} \right| \quad (2.10)$$

Untuk membagi koloni yang terlemah kepada *empire-empire*, vektor *P* dibentuk sebgai berikut,

$$P = [p_{p_1}, p_{p_2}, p_{p_3}, ..., p_{N_{imp}}] \quad (2.11)$$

kemudian dibuat sebuah vektor *R* dengan ukuran yang sama seperti *P* yang elemennya terdistribusi seragam secara random.

$$R = [r_1, r_2, r_3, ..., r_{N_{imp}}] \quad (2.12)$$

$$r_1, r_2, r_3, ..., r_{Nimp} \sim U(0,1) \quad (2.13)$$

kemudian vektor *D* dibentuk dengan mengurangkan *R* dari *P*

$D=P-R= [D_1, D_2, D_3, ..., D_{Nimp}] =$

$$[p_{p_1} - r_1, p_{p_2} - r_2, p_{p_3} - r_3, ..., p_{N_{imp}} - r_{N_{imp}}] \quad (2.14)$$

Berdasarkan vektor *D*, koloni yang disebutkan dikendalikan oleh *empire* yang memiliki *D* yang paling besar.

Proses dari pemilihan sebuah *empire* yang sama untuk proses *roullete wheel* yang digunakan pada pemillihan orang tua GA. Namun pada metode ini pemilihan dilakukan lebih cepat daripada *roullete wheel* konvensional. Karena ICA tidak memerlukan perhitungan fungsi distribusi komulatif dan seleksi didasarkan hanya pada nilai peluang. Oleh karena itu, proses seleksi *empire* hanya dapat menggantikan *roullete wheel* pada GA dan meningkatkan kecepatan eksekusi.

2.9. Eliminasi *Empire* Terlemah

Empire terlemah akan runtuh dalam kompetisi *imperialist* dan *koloni* dari *empire* tersebut akan dibagikan kepada *empire* yang lain. Diasumsikan sebuah *empire* akan runtuh dan tereliminasi ketika *empire* tersebut kehilangan semua *koloni*nya.

2.10. *Konvergensi*

Setelah semua *empire* runtuh kecuali satu yang terkuat maka semua *koloni* akan dikontrol atau menjadi milik dari *empire* yang terkuat. Pada dunia ideal yang baru, semua *koloni* akan akan mempunyai posisi dan *cost* yang sama dengan imperialist. Pada kondisi ini, maka kompetisi kekuasaan berakhir dan algoritma berhenti.

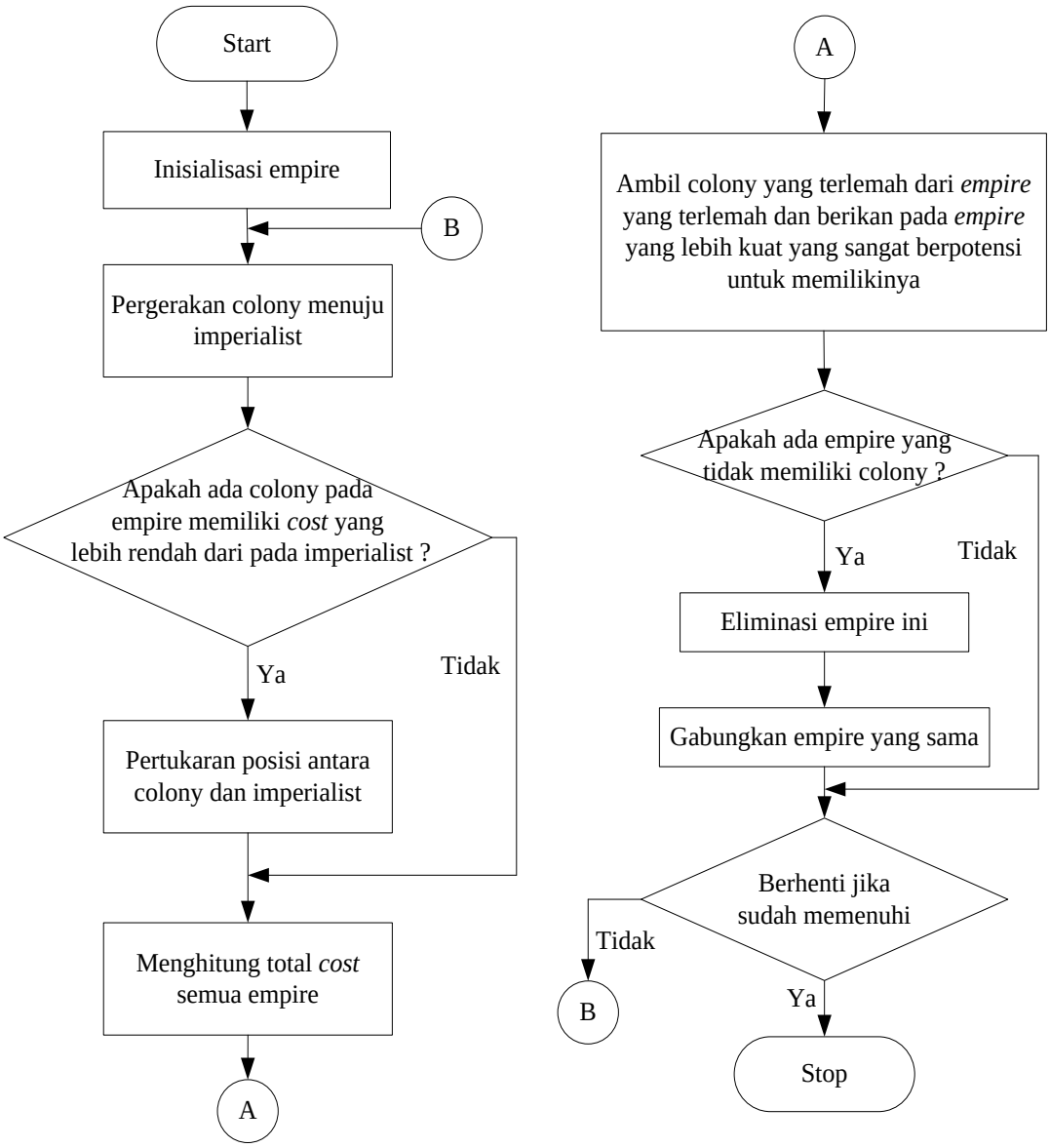
Adapun langkah-langkah utama pada ICA dapat dirangkum dalam pseudo-code berikut;

1. Pilih titik random pada fungsi dan inisialisasi empire (2.2 - 2.5).
2. Gerakkan koloni menuju imperialis yang relevan (asimilasi) (2.6 dan 2.7).
3. Secara random, ubahlah posisi dari beberapa koloni (revolusi)
4. Jika ada sebuah koloni yang memiliki cost lebih baik dari pada imperialis, ubahlah posisi dari koloni tersebut dengan imperialis.

5. Gabungkan empire yang sama
6. Hitung total cost dari semua empire (2.8)
7. Ambil koloni terlemah dari empire terlemah dan berikan kepada salah satu empire (Imperialistic competition) (2.9 - 2.14)
8. Hilangkan empire yang paling lemah
9. Jika kondisi berhenti dipenuhi, berhenti, jika tidak, ke langkah 2

Dari langkah-langkah tersebut diharapkan negara-negara untuk bertemu pada global minimum dari *cost function*. Kriteria yang bereda dapat digunakan untuk menghentikan algoritma. Salah satunya adalah menggunakan maksimum iterasi dari algoritma, yang disebut maksimum dekade. Dan juga bisa dengan cara yang lain, yaitu ketika hanya satu *empire* yang tersisa maka ICA akan berhenti.

Flowchart pada Gambar 2.9 mempresentasikan prosedur komputasi dari metode yang diusulkan.

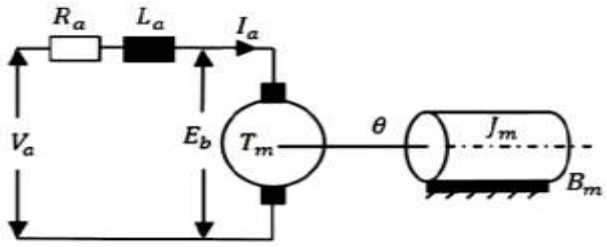


Gambar 2.9 *Flowchart* Algoritma Komputasi ICA

III. MODEL MOTOR DC

Sebuah motor listrik mengubah energi listrik menjadi energi mekanik dengan menggunakan berinteraksi medan magnet. Motor listrik yang digunakan untuk berbagai operasi perumahan, komersial, dan industri. Gulungan motor DC shunt terdiri dari medan shunt dihubungkan secara paralel dengan armature. Medan shunt memiliki ketahanan yang lebih tinggi dan arus yang lebih rendah dibandingkan dengan medan shunt seri. Akibatnya, motor ini memiliki kecepatan dan kontrol posisi yang sangat baik. Oleh karena itu DC shunt motor biasanya digunakan aplikasi yang membutuhkan tenaga kuda lima kali atau lebih. Persamaan menggambarkan perilaku dinamis dari motor DC berdasarkan diagram skematik pada Gambar (3.1) diberikan oleh persamaan berikut;

$$\boldsymbol{V_a = R_a . i_{a(t)} + La . \frac{di_{a(t)}}{dt} + e_{b(t)}} \quad \text{……..(3.1)}$$



Gambar 3.1. Skematik diagram motor DC

$$e_b(t) - K_b . w(t) \quad \text{…….. (3.2)}$$

$$T_m(t) - K_T . i_a(t) \quad \text{……. (3.3)}$$

$$T_m(t) = J_m . \frac{dw(t)}{dt} + B_m . w(t) \quad \text{…….. (3.4)}$$

Kemudian

$$T_m(t) = J_m . \frac{d\theta(t)}{dt} + B_m . \frac{d\theta(t)}{dt} \quad \text{…….. (3.5)}$$

Transfer fungsi akan diberikan:

$$\Theta\ (s) / v\ (s) = Kb / [J\ La\ S3 + (Ra\ J + B\ La)\ S2 + (KB2 + Ra\ B)\ S] \quad \text{….. (3.6)}$$

Dimana:

V_a = Angker tegangan (V)
R_a = resistansi angker (Ω)
L_a = induktansi angker (H)
I_a = arus dinamo (A)
E_b = back emf (V)
w = kecepatan sudut (rad / s)
T_m = torsi motor (Nm)
θ = posisi sudut dari poros rotor (rad)
J_m = rotor inersia (kg)
B_m = koefisien gesekan viskos (Nms / rad)
K_T = konstanta torsi (Nm / A)
K_b = konstanta back emf (Vs / rad)

Dengan memasukkan parameter Motor DC didapat R_a = 2,45 Ω, L_a = 0,035 H, K_b = 1,2 Vs/rad, J_m = 0,022 kgm^2, B_m = 0,5.10^{-2} Nms/rad

Sehingga Transfer function motor DC yang digunakan adalah:

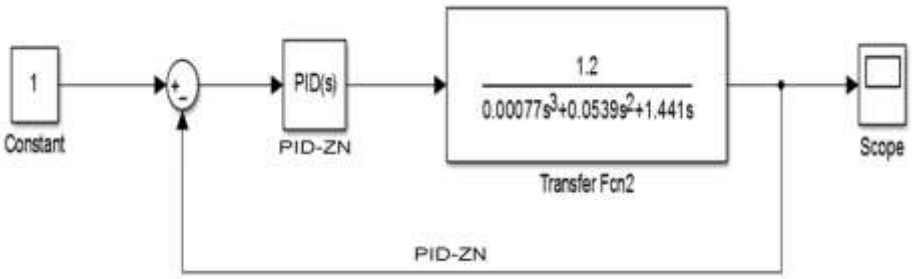
$$\frac{\theta(s)}{V_a(s)} = \frac{1{,}2}{0{,}00077s^3+0{,}0539s^2+1{,}441s} \quad \text{..... (3.7)}$$

IV. ANALISA DAN PEMBAHASAN

Tuning Ziegler-Nichols:

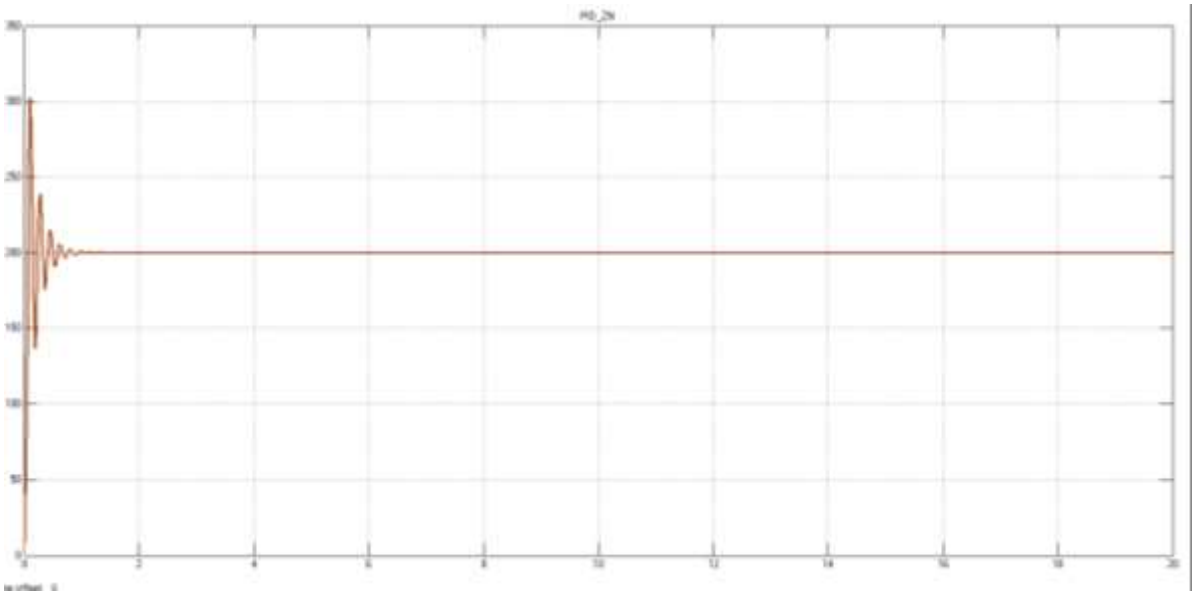
Ada dua metode untuk penentuan parameter kontroler PID Ziegler-Nichols disebut tala aturan. Tetapi metode yang diterima secara luas untuk tuning kontroler PID adalah metode sederhana. Pertama, mengatur controller ke mode P saja. Berikutnya, mengatur gain dari controller (Kp) ke nilai yang kecil. Akhirnya, menyesuaikan sampai respon diperoleh yang menghasilkan osilasi terus menerus. Hal ini dikenal sebagai gain tertinggi (Kp) atau ditunjukkan bahwa periode osilasi dikenal sebagai periode utama (Ku). Langkah-langkah yang diperlukan untuk metode yang diberikan di bawah: Koefisien integral (Ki) dan derivatif (Kd) harus mengatur ke nol. Secara bertahap meningkatkan koefisien proporsional dari nol sampai sampai sistem hanya mulai berosilasi terus menerus. Koefisien proporsional pada titik ini disebut gain tertinggi (Tu).

Tunning Ziegler_Nicols didapatkan Critikal gain Ku = 84 dan Kritikal periode Tu = 0,15 detik dan PID controller dapat dilihat pada gambar 4.1. dimana Kp = 49,41, ki = 0,075 dan Kd = 0,01875



Gambar 4.1. Block Diagram PID Controller Tuning Ziegler-Nichols

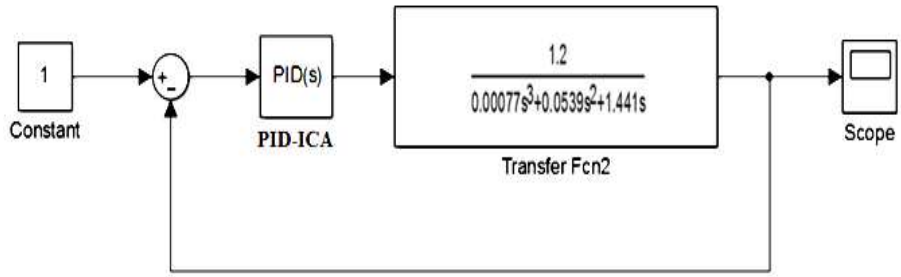
Hasil running program matlab 2013a dapat dilihat pada Gambar 4.2. dibawah ini:



Gambar 4.2. Hasil Running Program simulink motor DC Ziegler-Nichols.

Tuning Iperalist Competitive Algorithm (ICA)

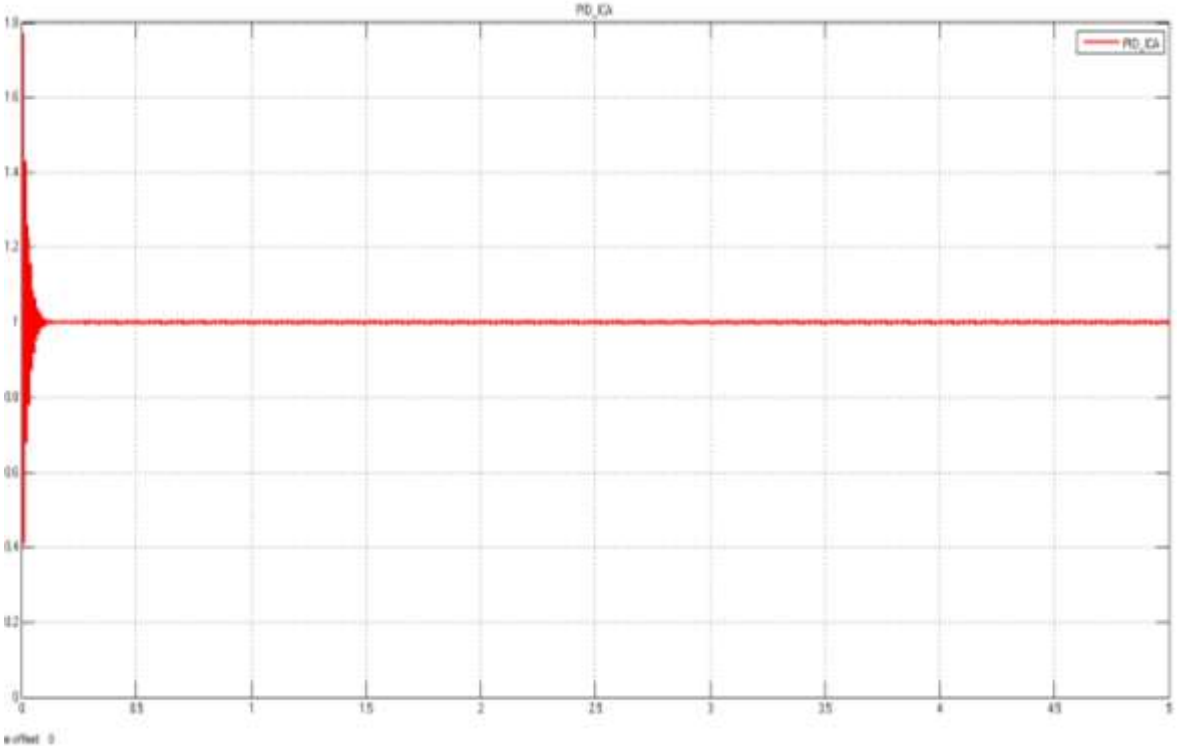
Simulasi model Motor Induksi dengan PID yang ditunning oleh Imperialist Competitive Algorithm Kontroller dapat dilihat pada gambar 4.3. dibawah ini:



Gambar 4.3. Model Motor Induksi dengan PID-ICA kontroller

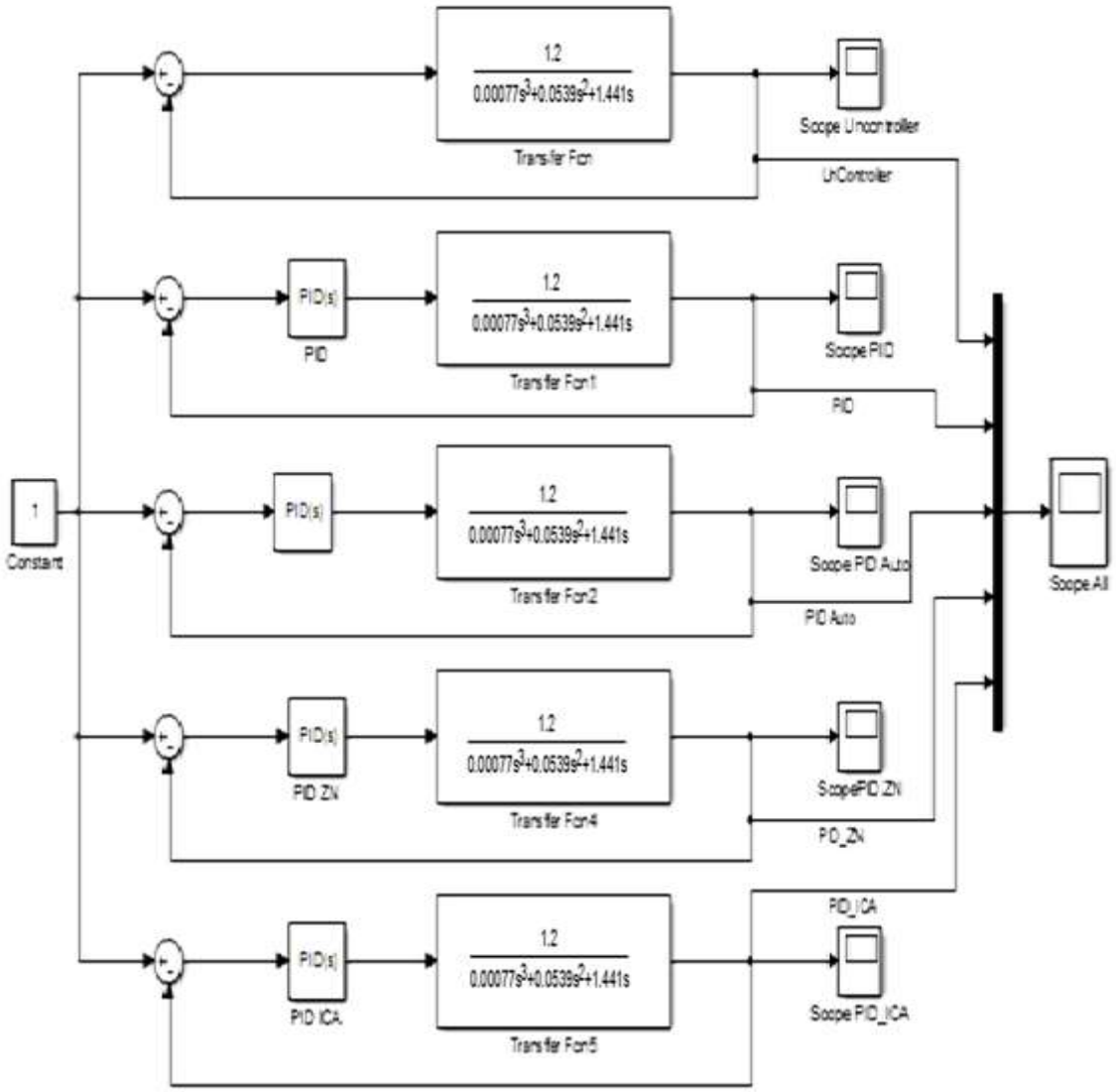
Hasil auto tunning pengaturan kecepatan Motor Induksi dengan PID yang tunning ICA didapatkan nilai konstanta Kp = 39.3369, Ki = 0 dan Kd = 0.9839.

Hasil running program matlab 2013a dapat dilihat pada Gambar 4.4. dibawah ini:



Gambar 4.4. Hasil Running Program simulink Motor Induksi ditunning dengan ICA.

Haril hasil grafik di atas dapat diartikan bahwa terjadi overshoots maks sebesar 1,75 pada saat t = 0,005 detik, dengan settlingtime 0.15 detik.

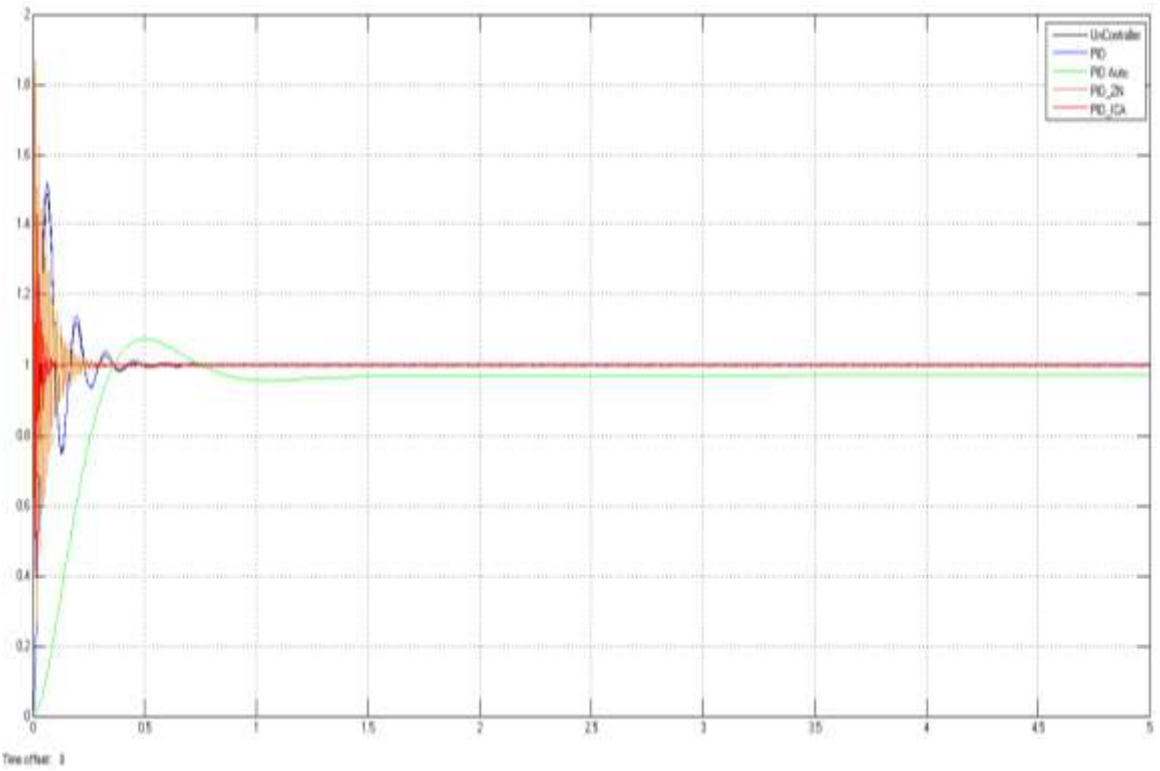


Gambar 4.5. Sistem kontrol kecepatan Motor DC dengan berbagai model control

Dari hasil running program didapatkan nilai Kp, Ki dan Kd dari masing-masing kontroler seperti pada table 1. dengan gambar respon seperti pada gambar 4.6. dibawah ini:

Tabel 1. Nilai Kp, Kd dan Ki

	Tanpa kontrol	PID standart	PID Auto Tunning	PID Ziegler-Nichols	PID-ICA
Kp	-	1	20.1979	49.41	39.3369
Ki	-	1	2.8623	0.01875	0
Kd	-	0	0.1264	0.075	0.9839
Overshoots	-	0,34	0.08	0.5	1,75
Setlingtime	16,01	25,5	5,26	1,75	0,15



Gambar 4.6. respon berbagai model kontrol

Hasil running program didapatkan nilai; overshoot tanpa kontrol 0 dengan settling time 7,634 detik, overshoot PID standart 1,513 dengan settling time 10 detik, overshoot PID-ZN 1,495 dengan settling time 2,023 detik, overshoot PID-ICA 1,103 dengan settling time 1,32 detik.

V. KESIMPULAN

Dari hasil analisis dan pembahasan pada penelitian maka didapat kesimpulan sebagai berikut: Sistem kontrol kecapatan motor DC yang dianggap paling baik adalah kontrol PID-ICA, kemudian PID-ZN, kemudian PID dan terakhir Nonkontrol. Hasil running program didapatkan nilai; overshoot tanpa kontrol 0 dengan settling time 7,634 detik, overshoot PID standart 1,513 dengan settling time 10 detik, overshoot PID-ZN 1,495 dengan settling time 2,023 detik, overshoot PID-ICA 1,103 dengan settling time 1,32 detik.

DAFTAR PUSTAKA

Dwi Hartanto, Thomas Wahyu. "***Analisis Dan Desain System Kontrol Dengan MATLAB***". Andy.Yokyakarta. 2001.

E. Atashpaz-Gargari and C. Lucas, "***Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition***," in Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, 2007, pp. 4661-4667.

E. Atashpaz-Gargari, F. Hashemzadeh, R. Rajabioun, C. Lucas, ***Colonial competitive algorithm: A novel approach for PID controller design in MIMO distillation column process***, International Journal of Intelligent Computing and Cybernetics. 1(3) (2008) 337-355.

H. Shayeg, A. Safari and H. A. Shayanfar, "***Multimachine Power System Stabilizer Design Using Iperalis Competitive Algorithm Algorithm***", International journal of Electrical Power and Energy System Engineering, 2008, 226-233.

J. Bates and M.E. Elbuluk and D.S. Zinger, ***"Neural Network Control of a Chopper Fed DC Motor***, 24th Annual IEEE 20-24 June 1993, pp. 893-899.

J. M. Zurada, ***Artificial Neural Networks***, copy right 1992 by west publishing company in the United States of America, pp. 185-208.

K Ogata, ***Modern Control Systems***, University of Minnesota, Prentice Hall, 1987.

M. Azizur Rahman, Fellow, IEEE, and M. Ashraful Hoque; ***On-Line Self-Tuning ANN-Based Speed Control of a PM DC Motor***, IEEE/ASME Transactions ON Mechatronics, VOL. 2, NO. 3, September 1997

M. I. Mahmoud, B. A. Zalam, M. A. Bardiny, E. A. Gomah, ***A Simplification Technique for an Adaptive Neural Network Based Speed Controller for Implementation on PLC for DC drive***, AIML 06 International Conference, 13 -15 June 2006, Sharm El Sheikh, Egypt.

O. Dwyer,.***PI And PID Controller Tuning Rules For Time Delay Process***: A Summary. Part 1: PI Controller Tuning Rules.. , Proceedings Of Irish Signals And Systems Conference, June 1999.

R. Rajabioun, F. Hashemzadeh, E. Atashpaz-Gargari, B. Mesgari, F. Rajaei Salmasi, Identification of a MIMO evaporator and its decentralized PID controller tuning using Colonial Competitive Algorithm, In the proceeding of IFAC World Congress, Seoul, Korea, 2008, pp. 9952-9957.