

Optimasi Interval *Fuzzy Time Series* Menggunakan *Particle Swarm Optimization* pada Peramalan Permintaan Darah : Studi Kasus Unit Transfusi Darah Cabang - PMI Kota Malang

Angga Dwi Apria Rifandi¹, Budi Darma Setiawan², Tibyani³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹angga.informatika.2013@gmail.com, ²s.budidarma@ub.ac.id, ³tibyani@ub.ac.id

Abstrak

Darah merupakan cairan penting yang secara alami dihasilkan dalam tubuh manusia. Ketika seorang manusia kehilangan banyak darah, maka diperlukan transfusi darah. Darah tersebut disediakan oleh pusat penyimpanan darah yang bertugas memperkirakan permintaan darah agar meminimalisir penuhnya penyimpanan maupun kerugian karena darah yang terbuang. Kurangnya pasokan darah dapat berdampak pada meningkatnya kematian pasien, sedangkan kelebihan pasokan darah hingga melewati masa simpannya (35 hari) juga harus dihindari. Demi meminimalisir kerugian tersebut diperlukan sebuah metode untuk melakukan peramalan permintaan darah yaitu menggunakan *fuzzy time series*. Demi meningkatkan akurasi, maka dilakukan optimasi interval dengan *particle swarm optimization* untuk menentukan interval terbaik dalam *fuzzy time series*. Berdasarkan hasil dari serangkaian pengujian, didapatkan solusi optimum bernilai *cost* (MSE) sebesar 60435.685 dengan jumlah partikel sebanyak 40, jumlah dimensi sebanyak 30, kombinasi nilai c_1 dan c_2 masing-masing 1.5 dan 1.5, bobot inersia sebesar 0.3, dan jumlah iterasi maksimum sebesar 950. Tingkat kesalahan dari sistem ini (MAPE) sebesar 7.50330% dari 12 data uji yang digunakan.

Kata kunci: permintaan darah, peramalan, *fuzzy time series*, *particle swarm optimization*

Abstract

Blood is an important fluid that naturally produced in the human body. When a human lost a lot of blood, a blood transfusion is needed. Blood for the transfusion is provided by a blood storage center in charge of estimating blood demand to minimize the excessive amount of blood in storage or wasted blood. Lack of blood supply can affect to the increased death of the patient, while an oversupply of blood until passes its shelf life (35 days) should also be avoided. In order to minimize the loss, a method to forecast the blood demand is needed, that is *fuzzy time series*. To increase the accuracy, the method is optimized with *particle swarm optimization* to determine the best interval in *fuzzy time series*. Based on the results of a series of tests, the optimum solution with average of cost value (MSE) of 60435.685 is obtained on 40 particles, 30 dimensions, 1.5 and 1.5 for the combination of c_1 and c_2 value respectively, the weight of inertia of 0.3, and the maximum number of iterations of 950. By using 12 testing data, the error rate generated by this system (MAPE) is 7.50330%.

Keywords: blood demand, forecasting, *fuzzy time series*, *particle swarm optimization*

1. PENDAHULUAN

Darah merupakan cairan penting yang secara alami diproduksi dalam tubuh manusia dan berfungsi untuk mengedarkan sari-sari makanan, oksigen serta zat-zat lain yang dibutuhkan oleh tubuh, transportasi bahan-bahan kimia yang dihasilkan oleh metabolisme dalam tubuh, dan juga sebagai mekanisme pertahanan tubuh terhadap virus ataupun bakteri (F & W,

2012). Ketika seorang manusia mengalami kehilangan darah dalam jumlah yang besar tubuh akan kesulitan dalam memproduksi darah baru untuk menggantikan darah yang hilang tersebut jika jumlah volume darah yang hilang lebih banyak daripada volume darah yang dihasilkan oleh tubuh. Sehingga diperlukan transfusi darah untuk menggantikan darah yang hilang agar nyawa manusia tersebut selamat.

Pusat penyimpanan darah bertanggung jawab untuk memenuhi permintaan transfusi tersebut. Dengan demikian, semua informasi yang ada tentang penggunaan terakhir komponen darah oleh rumah sakit ini sangat penting untuk membantu pengelola dari pusat penyimpanan darah untuk mengembangkan strategi keputusan dalam meningkatkan pemanfaatan sumberdaya tersebut yang tidak mudah karena darah dan komponen turunannya merupakan sumberdaya penting yang terbatas dan mudah rusak (Lang dalam Filho, Cezarino, & Salviano, 2012). Sel darah merah hanya memiliki masa hidup 35 hari sejak didonorkan, jika sudah melewati rentang waktu tersebut maka darah sudah tidak boleh lagi digunakan untuk transfusi (Rutherford, Cheng, & Bailie, 2016). Singkatnya, berdasarkan semua data yang tersedia, pengelola pusat penyimpanan darah memiliki tugas penting, yakni memperkirakan kebutuhan komponen darah untuk meminimalisir penuhnya penyimpanan maupun kerugian karena darah yang terbuang (Lang dalam Filho, Cezarino, & Salviano, 2012).

Kurangnya pasokan dapat berdampak pada meningkatnya jumlah kematian pasien. Di sisi lain, berlebihan pasokan darah hingga melewati masa simpannya juga tidak seharusnya terjadi, karena pendonor darah merupakan aset langka yang dibatasi waktu untuk mendonorkan darahnya (Belien & Force, 2012). Untuk meminimalisir kerugian tersebut perlu dilakukan sebuah metode untuk membantu menyesuaikan pasokan darah, yaitu dengan meramalkan permintaan darah.

Metode *fuzzy time series* telah terbukti menghasilkan peramalan dengan tingkat kesalahan yang cukup rendah. Pada penelitian yang menggunakan metode *fuzzy time series* dalam peramalan kebutuhan beban listrik harian di Malaysia (Efendi, Ismail, & Deris, 2015) didapatkan hasil bahwa peramalan yang dilakukan memiliki nilai MAPE yang cukup rendah, yaitu antara 1.23% sampai 1.63% dengan data uji 5 sampai 10 data. Kemudian, penelitian lain yang mencoba meramalkan pendaftar di Universitas Alabama dengan memodifikasi *fuzzy time series* Stevenson dan Porter (Saxeena, Sharma, & Easo, 2012) menghasilkan nilai AFER sebesar 0.34% dari hasil peramalan yang dilakukan.

Salah satu faktor yang dapat memengaruhi tingkat akurasi dari metode *fuzzy time series* adalah panjang interval dari partisi pada semesta pembicaraan (*universe of discourse*). Demi

meningkatkan akurasi peramalan, maka dalam penelitian ini akan digunakan algoritma *Modified Turbulence Particle Swarm Optimization* (MTPSO) yang merupakan modifikasi dari algoritma *Particle Swarm Optimization* (PSO).

PSO merupakan salah satu metode optimasi dalam *Swarm Intelligence*. Algoritma ini pertama kali dikembangkan oleh Eberhart dan Kennedy. PSO merupakan teknik komputasi evolusioner berbasis populasi yang terinspirasi dari perilaku sosial hewan seperti kawanan burung, kumpulan ikan, dan teori *swarming*. PSO dapat digunakan untuk menyelesaikan banyak permasalahan yang sama dengan Algoritma Genetika (Singh & Borah, 2014). PSO secara umum terbukti memiliki kinerja yang lebih baik daripada algoritma lainnya seperti Algoritma Genetika, *Memetic Algorithm*, *Antcolony System*, dan *Shuffled Frog Leaping* dalam hal tingkat keberhasilan dan kualitas solusi (Elbeltagi, Hegazy, & Grierson dalam Hsu, et al., 2010).

PSO telah berhasil dilakukan untuk melakukan optimasi interval *fuzzy time series* pada beberapa penelitian sebelumnya. Pada penelitian yang dilakukan oleh Huang (2011) yang menggunakan metode *aggregated fuzzy time series* dan *particle swarm optimization* untuk meramalkan jumlah pendaftar di Universitas Alabama didapatkan hasil MSE yang lebih rendah daripada metode lain yang sudah ada. Pada penelitian yang mencoba menerapkan algoritma MTPSO yang berasal dari PSO dalam meramalkan temperatur udara dan *Taiwan Future Exchange* (TAIFEX) (Hsu, et al., 2010) ditemukan bahwa PSO berhasil menurunkan tingkat kesalahan peramalan dibandingkan dengan penelitian sebelumnya yang dilakukan tanpa optimasi. Penelitian berikutnya yang meramalkan harga indeks saham (Singh & Borah, 2014) menunjukkan bahwa penerapan algoritma PSO untuk optimasi interval pada *m-factor fuzzy time series* menghasilkan nilai MAPE yang cukup rendah yaitu 0.6733% dan 0.4549% masing-masing untuk operasi *union* dan *intersection*.

2. DATA YANG DIGUNAKAN

Data yang digunakan dalam penelitian ini adalah data permintaan darah bulanan yang diperoleh dari Badan Pusat Statistik Kota Malang yang menghimpunnya dari Unit Transfusi Darah Cabang – PMI Kota Malang

dalam buku “Kota Malang dalam Angka” mulai tahun 2010 hingga tahun 2015. Sebanyak 60 data, yang merupakan jumlah permintaan darah dan komponen darah bulanan mulai tahun 2010 hingga 2014, akan digunakan sebagai data latih yang ditunjukkan pada Tabel 1.

Tabel 1. Data Latih

Bulan/Tahun	Jumlah	Bulan/Tahun	Jumlah
Januari/2010	4678	Juli/2012	4691
Februari/2010	4336	Agustus/2012	4784
Maret/2010	4703	September/2012	4657
April/2010	4858	Oktober/2012	4910
Mei/2010	4848	November/2012	4904
Juni/2010	4101	Desember/2012	4882
Juli/2010	4375	Januari/2013	5296
Agustus/2010	4403	Februari/2013	4555
September/2010	4121	Maret/2013	4978
Oktober/2010	5013	April/2013	5218
November/2010	4669	Mei/2013	5454
Desember/2010	4673	Juni/2013	6015
Januari/2011	4950	Juli/2013	5305
Februari/2011	4451	Agustus/2013	4940
Maret/2011	5125	September/2013	4676
April/2011	4808	Oktober/2013	4843
Mei/2011	4815	November/2013	5113
Juni/2011	5270	Desember/2013	5345
Juli/2011	5067	Januari/2014	5173
Agustus/2011	4575	Februari/2014	4864
September/2011	5039	Maret/2014	4924
Oktober/2011	5341	April/2014	5432
November/2011	5169	Mei/2014	5222
Desember/2011	5080	Juni/2014	5260
Januari/2012	5093	Juli/2014	5368
Februari/2012	5092	Agustus/2014	5465
Maret/2012	5056	September/2014	4523
April/2012	4843	Oktober/2014	5386
Mei/2012	4931	November/2014	4708
Juni/2012	4601	Desember/2014	4704

Kemudian sebanyak 12 data, yang merupakan jumlah permintaan darah dan komponen darah bulanan pada tahun 2015, akan digunakan sebagai data uji yang ditunjukkan pada Tabel 2.

Tabel 2. Data Uji

Bulan/Tahun	Jumlah
Januari/2015	5196
Februari/2015	4173
Maret/2015	4203
April/2015	4334
Mei/2015	4358
Juni/2015	4485
Juli/2015	4791

Agustus/2015	4570
September/2015	3573
Oktober/2015	4163
November/2015	3914
Desember/2015	4055

3. FUZZY TIME SERIES

Konsep *fuzzy time series* dikenalkan oleh Song dan Chissom pertama kali pada tahun 1993, di mana nilai-nilainya direpresentasikan dengan himpunan *fuzzy*.

Dalam melakukan peramalan dengan model *fuzzy time series* langkah-langkah yang dapat dilakukan ialah sebagai berikut (Huang, et al., 2011):

1. Menentukan Semesta Pembicaraan (*Universe of Discourse*).

Jika $Y(t)$ merupakan data historis pada tahun t , maka semesta pembicaraan didefinisikan sebagai:

$$U = (awal, akhir) \quad (1)$$

di mana $awal \leq Y(t) \leq akhir$.

2. Membagi Semesta Pembicaraan menjadi beberapa interval.

Semesta pembicaraan yang telah ditentukan akan dibagi menjadi n -interval dengan ukuran yang sama. Misalkan semesta pembicaraan U dibagi menjadi 6 interval, maka intervalnya adalah $u_1, u_2, u_3, u_4, u_5, u_6$.

3. Melakukan fuzzifikasi data historis.

Setiap interval pada Langkah 2 mewakili variabel linguistik A_i ($1 \leq i \leq n$) yang menyatakan derajat relatif dari angka pada data historis. Masing-masing variabel linguistik merepresentasikan sebuah himpunan *fuzzy* yang dinyatakan dengan $A_i = \{\delta_j/u_j\}$, di mana $u_j = \{u_1, u_2, \dots, u_n\}$ dengan derajat keanggotaan berbeda $\delta_j = \{\delta_1, \delta_2, \dots, \delta_n\}$. Sehingga himpunan *fuzzy* A_i ($1 \leq i \leq n$) dapat dibentuk seperti berikut:

$$A_1 = 1/u_1 + 0.5/u_2 + 0/u_3 + \dots + 0/u_{n-1} + 0/u_n$$

$$A_2 = 0.5/u_1 + 1/u_2 + 0.5/u_3 + \dots + 0/u_{n-1} + 0/u_n$$

$$\dots$$

$$A_n = 0/u_1 + 0/u_2 + 0/u_3 + \dots + 0.5/u_{n-1} + 1/u_n \quad (2)$$

Untuk melakukan fuzzifikasi data historis, data tiap tahun dikonversi sesuai dengan interval yang mencakup data tersebut. Asumsikan terdapat dua data *time serial* $Y(t)$ dan $F(t)$ pada tahun t , di mana $Y(t)$ merupakan data historis pada tahun t dan $F(t)$ merupakan himpunan *fuzzy* dari $Y(t)$. Jika $Y(t)$ merupakan bagian dari

u_i ($1 \leq i \leq n$), maka $Y(t)$ difuzzifikasi menjadi $F(t) = A_i$.

4. Membuat fuzzy logical relationship (FLR).

Fuzzy logical relationship dibuat dari dua atau lebih himpunan fuzzy berurutan dalam time series. Dalam fase pelatihan, semua variabel linguistik dari satu keadaan menuju keadaan selanjutnya telah diketahui dan digunakan sebagai data latih. Misalkan terdapat data tahun $t-2$, $t-1$ dan t yang hasil fuzzifikasinya masing-masing adalah A_i, A_j , dan A_k , maka bentuk first-order fuzzy logical relationshipnya adalah $A_i \rightarrow A_j$, dan $A_j \rightarrow A_k$, sedangkan bentuk second-order fuzzy relationshipnya adalah $A_i, A_j \rightarrow A_k$.

5. Menyusun fuzzy logical relationship group (FLRG).

Fuzzy logical relationship group dibuat dengan mengumpulkan fuzzy logical relationship yang telah ada berdasarkan current state yang sama. Misalkan terdapat fuzzy logical relationship $A_i \rightarrow A_j$, $A_j \rightarrow A_k$, $A_k \rightarrow A_l$, $A_l \rightarrow A_i$, $A_i \rightarrow A_k$, $A_k \rightarrow A_j$, $A_j \rightarrow A_i$ dan $A_i \rightarrow A_i$. Maka bentuk fuzzy relation groupnya adalah:

$$\begin{aligned} G_1: A_i &\rightarrow A_i, A_j, A_k \\ G_2: A_j &\rightarrow A_i, A_k \\ G_3: A_k &\rightarrow A_j, A_l \\ G_4: A_l &\rightarrow A_i \end{aligned} \quad (3)$$

6. Menghitung nilai peramalan (defuzzifikasi).

Untuk meningkatkan akurasi peramalan, maka dalam penelitian ini digunakan metode peramalan yang menyatukan informasi global dan lokal pada latest fuzzy fluctuation (LFF) untuk menghitung hasil peramalan. Nilai hasil peramalan dapat dihitung menggunakan rumus berikut:

$$\text{Hasil_peramalan} = w_1 \times \text{Glob_info} + w_2 \times \text{Local_info} \quad (4)$$

Nilai w_1 dan w_2 masing-masing merupakan bobot adaptif untuk informasi global pada fuzzy logical relationship yang dinotasikan dengan Glob_info dan informasi lokal pada LFF yang dinotasikan dengan Local_info, di mana $w_1 + w_2 = 1$ dan $0 \leq w_1, w_2 \leq 1$.

Jika fuzzy logical relationship group terdiri atas lebih dari satu fuzzy logical relationship, maka nilai dari Glob_info adalah rata-rata dari titik tengah (midpoint) dari semua interval terkait dengan variabel linguistik yang ada pada next state dari semua fuzzy logical relationship. Bila terdapat first-order fuzzy logical relationship $A_{t-1} \rightarrow A_{t1}, A_{t2}, \dots, A_{tk}$ dan $m_{t1}, m_{t2}, \dots, m_{tk}$ masing-masing merupakan

midpoint dari variabel linguistik $A_{t1}, A_{t2}, \dots, A_{tk}$. Maka nilai Glob_info dapat dihitung dengan rumus berikut:

$$\text{Glob_info} = \frac{m_{t1} + m_{t2} + \dots + m_{tk}}{k} \quad (5)$$

Variabel Local_info merupakan informasi local yang berasal dari skema LFF. Skema LFF merupakan skema estimasi berdasarkan next state dan latest past dari current state. Jika terdapat sebuah n-order fuzzy logical relationship $A_{t-n}, A_{t-n+1}, \dots, A_{t-2}, A_{t-1} \rightarrow A_t$, dengan $n \geq 1$ dan $t \geq 2$, di mana A_{t-1} dan A_t masing-masing menyatakan latest past dalam current state dan next state, lalu m_{t-1} dan m_t merupakan midpoint dari fuzzy interval u_{t-1} dan u_t yang berkaitan dengan A_{t-1} dan A_t , di mana $u_{t-1} = (bs_{t-1}, be_{t-1})$ dan $u_t = (bs_t, be_t)$, maka skema LFF dapat dirumuskan sebagai berikut:

$$\text{Local_info} = bs_t + \frac{be_t - bs_t}{2} \times \frac{m_t - m_{t-1}}{m_t + m_{t-1}} \quad (6)$$

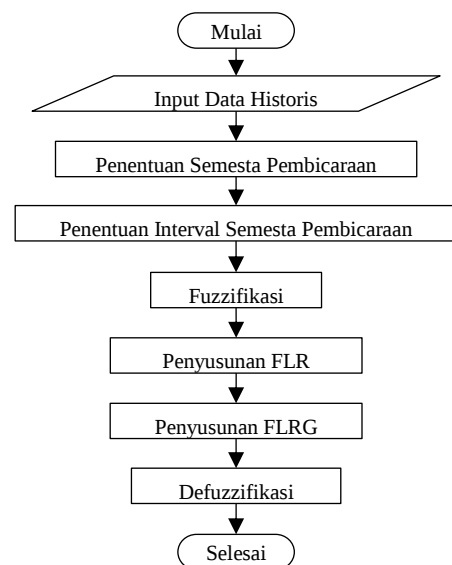
Untuk data tanpa data aktual, maka informasi global dan informasi lokal masing-masing dirumuskan sebagai berikut:

$$\text{Glob_info} = m_{t-1} \quad (7)$$

$$\text{Local_info} = bs_{t-1} + \frac{be_{t-1} - bs_{t-1}}{2} \times \frac{m_{t-1} - m_{t-2}}{m_{t-1} + m_{t-2}} \quad (8)$$

Di mana Glob_info dihitung berdasarkan nilai midpoint dari himpunan fuzzy dari data antesedennya dan Local_info dihitung berdasarkan data anteseden dan satu data sebelum antesedennya.

Diagram alir fuzzy time series dalam penelitian ini ditunjukkan dalam Gambar 1.



Gambar 1. Diagram Alir Fuzzy Time Series

4. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) merupakan salah satu metode optimasi dalam *swarm intelligence* yang terinspirasi dari perilaku sosial yang terdapat di alam. Algoritma ini pertama kali dikembangkan oleh Eberhart dan Kennedy. PSO merupakan teknik komputasi evolusioner berbasis populasi yang terinspirasi dari perilaku sosial hewan seperti kawanan burung, kumpulan ikan, dan teori *swarming*.

Dalam PSO sejumlah partikel membentuk kerumunan (*swarm*) yang berkembang atau “terbang” melintasi ruang multidimensional untuk mencari daerah “kaya makanan” di mana terdapat solusi yang optimal. Dalam ruang pencarian N -dimensional masing-masing partikel memiliki dua vektor, yaitu vektor posisi (P_i) dan kecepatan (V_i). $P_i = [p_{i1}, p_{i2}, \dots, p_{iN}]$ dan $V_i = [v_{i1}, v_{i2}, \dots, v_{iN}]$ adalah dua vektor yang terkait dengan tiap partikel i . Selama masa pencarian, anggota *swarm* saling berinteraksi dengan cara tertentu untuk mengoptimalkan pencarian mereka. Posisi tiap partikel direpresentasikan berdasarkan dimensi permasalahannya yang dapat berupa bilangan biner, real, atau integer.

Terdapat beragam paradigma *particle swarm*, tetapi yang sering digunakan adalah model *Gbest* di mana keseluruhan populasi dianggap sebagai satu ketetanggaan dalam proses pencarian. Dalam setiap iterasi, partikel dengan solusi terbaik akan membagikan koordinat posisinya dengan anggota lainnya, kemudian tiap partikel memperbarui posisinya berdasarkan hasil pencarian terbaik individu (*Pbest*) dan keseluruhan (*Gbest*) (AlRashidi & El-Naggar, 2010). Berikut adalah tahapan dari algoritma PSO (Hsu, et al., 2010):

1. Inisialisasi posisi (P_i) dan kecepatan (V_i) untuk $1 \leq i \leq N$, di mana N merupakan jumlah partikel.
2. Menghitung nilai *fitness* atau *cost* dari partikel i .
3. Memperbarui posisi terbaik (*Pbest*) dari partikel i berdasarkan nilai *fitness* partikel tersebut pada iterasi saat ini dibandingkan dengan iterasi sebelumnya.
4. Memperbarui posisi terbaik dari keseluruhan partikel (*Gbest*) dari berdasarkan nilai *fitness* atau *cost* dari seluruh partikel.
5. Menghitung kecepatan partikel menggunakan rumus :

$$V_i = \omega \times V_i + c_1 \times rand_1() \times (Pbest - P_i) + c_2 \times rand_2() \times (Gbest - P_i) \quad (9)$$

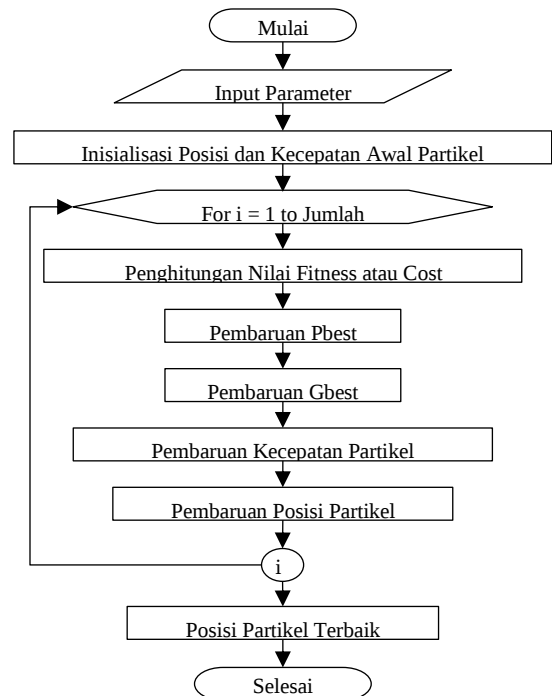
Di mana c_1 dan c_2 merupakan konstanta percepatan positif (biasanya $c_1 = c_2 = 2$), di mana masing-masing merupakan koefisien kognitif dan koefisien sosial, lalu $rand_1()$ dan $rand_2()$ merupakan bilangan random dalam range $[0,1]$, dan ω merupakan bobot inersia.

6. Memperbarui posisi partikel menggunakan rumus :

$$P_i = P_i + V_i \quad (10)$$

7. Untuk masing-masing partikel ulangi langkah 1-6.
8. Selama solusi optimal belum ditemukan atau belum mencapai iterasi maksimum, ulangi langkah 2-7.

Diagram alir PSO ditunjukkan pada Gambar 2.



Gambar 2. Diagram Alir PSO

4.1. Turbulent Particle Swarm Optimization

Untuk mengarahkan partikel secara efektif dalam ruang pencarian, jarak perpindahan maksimum dalam sebuah iterasi dan jangkauan perpindahannya harus dibatasi di antara kecepatan maksimumnya. Metode untuk menggerakkan partikel yang “malas” (partikel dengan kecepatan di bawah *threshold*) agar mencari solusi yang lebih baik disebut *turbulent particle swarm optimization* (TPSO) yang dihitung dengan rumus berikut (Hsu, et al., 2010):

$$V_i = \begin{cases} V_{max}, & \text{jika } V_i > V_{max} \\ -V_{max}, & \text{jika } V_i < -V_{max} \\ -V_{max} + 2 \times V_{max} \times rand(), & \text{jika } |V_i| < V_s \\ V_i, & \text{untuk kondisi lainnya} \end{cases} \quad (11)$$

Di mana V_i merupakan kecepatan partikel yang dibatasi dengan rentang $\pm V_{max}$, $rand()$ merupakan bilangan random dalam range $[0,1]$, V_s merupakan *threshold* minimum kecepatan. Untuk menghitung V_{max} , dapat digunakan rumus berikut (Marini & Walczak, 2015):

$$V_i^{max} = k \frac{(P_{i,max} - P_{i,min})}{2} \quad (12)$$

Di mana k merupakan konstanta dengan rentang 0 sampai 1, $P_{i,max}$ merupakan batas maksimum posisi partikel i , dan $P_{i,min}$ merupakan batas minimum posisi partikel.

5. PSO UNTUK OPTIMASI INTERVAL FUZZY TIME SERIES

Dalam penelitian ini, algoritma *particle swarm optimization* digunakan untuk melakukan optimasi interval *fuzzy time series* demi meningkatkan hasil akurasi peramalan. Berikut ini merupakan langkah-langkah peramalan dengan *Fuzzy Time Series* dan PSO (Huang, et al., 2011):

1. Menentukan Semesta Pembicaraan (*Universe of Discourse*).
2. Inisialisasi posisi (P_i) dan kecepatan (V_i) untuk $1 \leq i \leq N$, di mana N merupakan jumlah partikel.
3. Membagi Semesta Pembicaraan yang telah dibuat menjadi beberapa interval berdasarkan posisi partikel.
4. Melakukan fuzzifikasi data historis.
5. Membuat *fuzzy logical relationship*.
6. Menyusun *fuzzy logical relationship group*.
7. Menghitung nilai peramalan.
8. Menghitung akurasi peramalan dengan nilai MSE.
9. Menghitung nilai *cost* dari partikel i menggunakan nilai MSE tersebut.
10. Memperbarui posisi terbaik (P_{best}) dari partikel i berdasarkan nilai *cost* partikel tersebut pada iterasi saat ini dibandingkan dengan iterasi sebelumnya.
11. Memperbarui posisi terbaik dari keseluruhan partikel (G_{best}) dari berdasarkan nilai *cost* dari seluruh partikel.
12. Menghitung kecepatan partikel
13. Memperbarui posisi partikel
14. Untuk masing-masing partikel ulangi langkah 1-13.

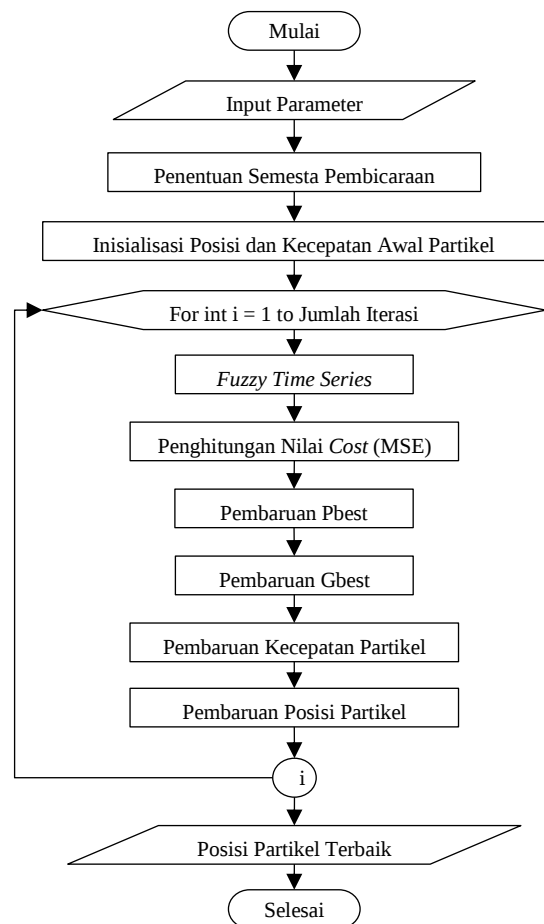
15. Selama solusi optimal belum ditemukan atau belum mencapai iterasi maksimum, ulangi langkah 2-14.

Diagram alir optimasi interval fuzzy time series menggunakan PSO ditunjukkan pada Gambar 3.

Dalam proses optimasi oleh PSO ini representasi partikel PSO yang digunakan adalah *real-coded* (pengkodean real) karena menyesuaikan dengan batasan interval *fuzzy time series* berupa bilangan real sebagai posisi partikel dalam dimensi pencarian. Posisi partikel berkaitan dengan batas interval pada *fuzzy time series* sebagai posisi partikel, hanya saja batas interval yang menjadi batas bawah maupun batas atas semesta pembicaraan tidak diikutsertakan dalam representasi partikel karena bersifat tetap. Jumlah dimensi partikel ditentukan dari jumlah interval *fuzzy time series* yang diinginkan seperti pada rumus berikut.

$$\text{jumlah_dimensi} = \text{jumlah_interval} - 1 \quad (13)$$

Sehingga bila terdapat 6 interval, sesuai rumus 13, maka jumlah dimensi partikel adalah 5. Representasi partikel dengan 5 dimensi dicontohkan pada Tabel 3.



Gambar 3. Diagram Alir Optimasi Interval Fuzzy Time Series Menggunakan PSO

Tabel 3. Representasi Partikel

Partikel	b ₁	b ₂	b ₃	b ₄	b ₅
P _i	4210	4357	5371	5865	6063

Keterangan:

P_i : Partikel ke-*i*

b₁ : Batas atas interval kesatu dan batas bawah interval kedua

b₂ : Batas atas interval kedua dan batas bawah interval ketiga

b₃ : Batas atas interval ketiga dan batas bawah interval keempat

b₄ : Batas atas interval keempat dan batas bawah interval kelima

b₅ : Batas atas interval kelima dan batas bawah interval keenam

Karena posisi partikel pada dimensi tertentu merupakan batasan interval, maka posisi partikel tersebut harus diurutkan secara *ascending*. Posisi partikel dengan nilai terkecil akan menempati dimensi b₁, kemudian selanjutnya menempati b₂, dan seterusnya hingga partikel dengan nilai terbesar menempati b₅.

Untuk menunjukkan nilai *cost* tiap partikel, *mean square error* (MSE) digunakan sebagai fungsi obyektif. Kemudian *mean absolute percentage error* (MAPE) digunakan sebagai kriteria evaluasi untuk menunjukkan akurasi peramalan yang dihasilkan oleh sistem. Nilai MSE dan MAPE dihitung menggunakan rumus berikut:

$$MSE = \frac{\sum_{i=1}^N (f v_i - av_i)^2}{N} \quad (14)$$

$$MAPE = \sum_{i=1}^N \frac{|av_i - f v_i|}{av_i} \times \frac{100\%}{N} \quad (15)$$

di mana *N* menyatakan jumlah data historis dalam *time series*, *f v_i* menyatakan nilai hasil peramalan data ke-*i* dan *av_i* menyatakan data aktual ke-*i*.

6. HASIL DAN PEMBAHASAN

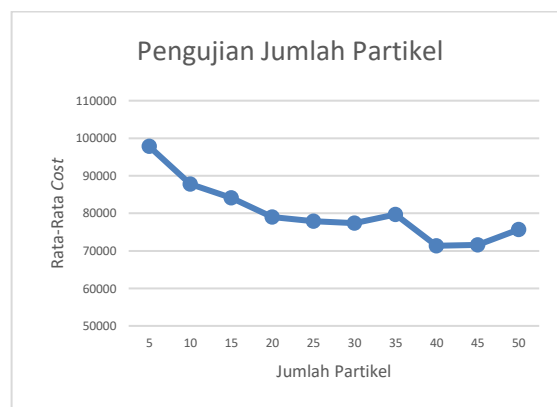
Hasil dari pengujian yang dilakukan berdasarkan rancangan pengujian yang telah dirancang sebelumnya akan dianalisis, sehingga hasil yang telah diperoleh dari serangkaian pengujian yang telah dilakukan dapat menghasilkan suatu kesimpulan mengenai penelitian yang dijalankan ini. Setiap pengujian diulang sebanyak 10 kali untuk mendapatkan rata-rata nilai *cost* dari setiap pengujian.

6.1. Pengujian Jumlah Partikel

Pengujian ini digunakan untuk menentukan jumlah partikel demi menghasilkan interval

paling optimum dalam *Fuzzy Time Series*. Untuk pengujian ini, jumlah partikel yang digunakan adalah 5, 10, 15, 20, 25, 30, 35, 40, 45, dan 50. Grafik hasil pengujian jumlah partikel dapat dilihat pada Gambar 4.

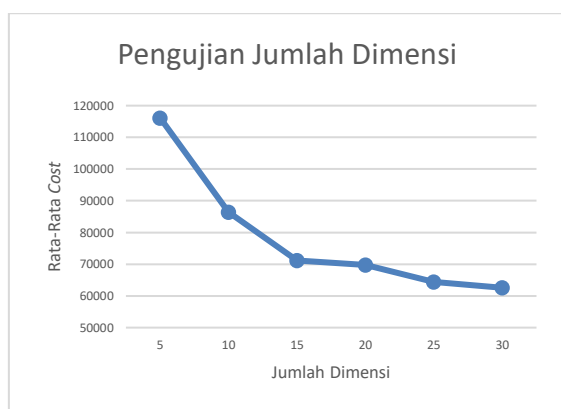
Berdasarkan grafik hasil pengujian pada Gambar 4, rata-rata nilai *cost* terendah didapatkan dari jumlah partikel 40. Dari grafik tersebut dapat dilihat bahwa rata-rata nilai *cost* cenderung menurun dengan bertambahnya jumlah partikel, meskipun tidak seluruhnya menurun karena pengaruh inisialisasi partikel secara acak yang menyebabkan hasil pengujian sedikit berbeda antara satu pengujian dengan lainnya. Ini mengindikasikan bahwa bertambahnya jumlah partikel memiliki kemungkinan lebih besar untuk menurunkan nilai *cost*. Menurunnya nilai *cost* seiring bertambahnya jumlah partikel disebabkan karena semakin banyak jumlah partikelnya maka semakin banyak posisi dalam ruang pencarian yang dapat dijelajahi, meskipun hal ini tidak sepenuhnya menjamin dapat menurunkan nilai *cost* sebab walaupun terdapat banyak partikel tetapi saat inisialisasi berada jauh dari titik optimum maka untuk menuju titik optimum akan memerlukan waktu lebih lama dibanding partikel yang sudah berada di dekat titik optimum. Selain itu jumlah partikel yang terlalu besar dapat berakibat pada bertambahnya waktu komputasi yang dibutuhkan sehingga berakibat pada menurunnya kinerja sistem.

**Gambar 4.** Grafik Hasil Pengujian Jumlah Partikel

6.2. Pengujian Jumlah Dimensi PSO

Pengujian ini digunakan untuk menentukan jumlah dimensi PSO demi menghasilkan interval paling optimum dalam *Fuzzy Time Series*. Jumlah dimensi yang digunakan adalah 5, 10, 15, 20, 25, dan 30. Grafik hasil pengujian jumlah dimensi dapat dilihat pada Gambar 5.

Berdasarkan grafik hasil pengujian pada Gambar 5, rata-rata nilai *cost* terendah didapatkan pada jumlah dimensi 30. Terlihat pada grafik bahwa nilai *cost* cenderung menurun seiring dengan bertambahnya jumlah dimensi. Hal ini menunjukkan bahwa bertambahnya jumlah dimensi dapat menurunkan nilai *cost*, karena banyaknya jumlah dimensi akan berpengaruh pada jumlah interval *fuzzy time series*. Bertambahnya jumlah interval *fuzzy time series* mengakibatkan rentang tiap interval secara umum semakin mengecil. Hal ini berpengaruh pada penyusunan FLR yang dapat menurunkan tingkat kesalahan peramalan.



Gambar 5. Grafik Hasil Pengujian Jumlah Dimensi

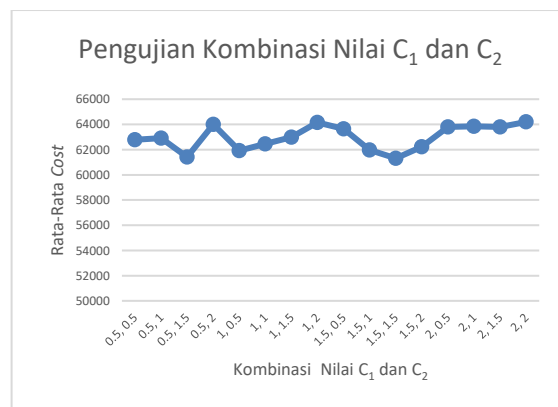
6.3. Pengujian Kombinasi Nilai Konstanta Akselerasi PSO (c_1 dan c_2)

Pengujian ini digunakan untuk menentukan kombinasi nilai konstanta akselerasi (c_1 dan c_2) demi menghasilkan interval paling optimum dalam *Fuzzy Time Series*. Kombinasi nilai c_1 dan c_2 yang digunakan adalah 0.5 dan 0.5, 0.5 dan 1, 0.5 dan 1.5, 0.5 dan 2, 1 dan 0.5, 1 dan 1, 1 dan 1.5, 1 dan 2, 1.5 dan 0.5, 1.5 dan 1, 1.5 dan 1.5, 1.5 dan 2, 2 dan 0.5, 2 dan 1, 2 dan 1.5, 2 dan 2. Grafik hasil pengujian kombinasi dapat dilihat pada Gambar 6.

Berdasarkan grafik hasil pengujian pada Gambar 6, rata-rata nilai *cost* terendah didapatkan dari kombinasi 1.5 dan 1.5. Rata-rata nilai *cost* naik turun pada kombinasi 0.5, 0.5 hingga 1, 0.5. Rata-rata nilai *cost* mulai meningkat pada kombinasi 1, 0.5 hingga 1, 2 kemudian menurun kembali pada kombinasi 1.5, 0.5 hingga 1.5, 1.5 lalu mengalami kenaikan pada kombinasi 1.5, 2 hingga 2, 0.5. Pada kombinasi 2, 0.5 hingga 2, 2 rata-rata nilai *cost* tidak mengalami perubahan signifikan.

Ketidakstabilan ini terjadi karena masing-masing merupakan koefisien kognitif dan sosial yang mempengaruhi pergerakan partikel. Jika c_1

terlalu besar dibanding c_2 maka partikel cenderung hanya bergerak di sekitar posisi terbaik personalnya ($Pbest$) dan tidak menuju posisi terbaik global ($Gbest$), hal ini menyebabkan partikel dengan nilai *cost* yang tinggi akan membutuhkan iterasi yang lebih banyak untuk mencapai titik optimum. Tetapi jika c_2 terlalu besar dibanding c_1 maka partikel akan berpindah terlalu cepat mendekati posisi terbaik global ($Gbest$), hal ini dikhawatirkan partikel akan melewati titik yang seharusnya menjadi titik optimum.



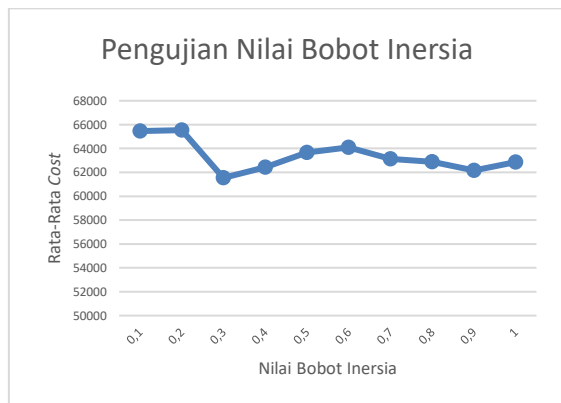
Gambar 6. Grafik Hasil Pengujian Kombinasi Nilai C_1 dan C_2

6.4. Pengujian Nilai Bobot Inersia (ω)

Pengujian ini digunakan untuk menentukan nilai bobot inersia (ω) demi menghasilkan interval paling optimum dalam *Fuzzy Time Series*. Untuk pengujian ini nilai bobot inersia yang digunakan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan 1. Grafik hasil pengujian nilai bobot inersia dapat dilihat pada Gambar 7.

Berdasarkan grafik hasil pengujian pada Gambar 7, rata-rata nilai *cost* terendah didapatkan pada nilai bobot inersia 0.3. Rata-rata nilai *cost* cenderung meningkat pada bobot inersia 0.4 hingga 0.6, baru pada bobot inersia 0.7 hingga 0.9 rata-rata nilai *cost* mengalami penurunan. Ketidakstabilan ini disebabkan karena nilai bobot inersia akan memengaruhi kegunaan dari kecepatan partikel pada iterasi sebelumnya. Semakin besar nilai bobot inersia, maka semakin besar pengaruh kecepatan partikel pada iterasi sebelumnya terhadap hasil pembaruan kecepatan partikel. Sedangkan jika nilai bobot inersia terlalu kecil maka kecepatan partikel pada iterasi sebelumnya tidak terlalu berpengaruh, hal ini berdampak pada kecepatan gerak partikel yang menjadi lambat sehingga menyebabkan partikel yang berada jauh dari posisi partikel terbaik akan membutuhkan lebih

banyak iterasi untuk mencapai posisi partikel terbaik tersebut.

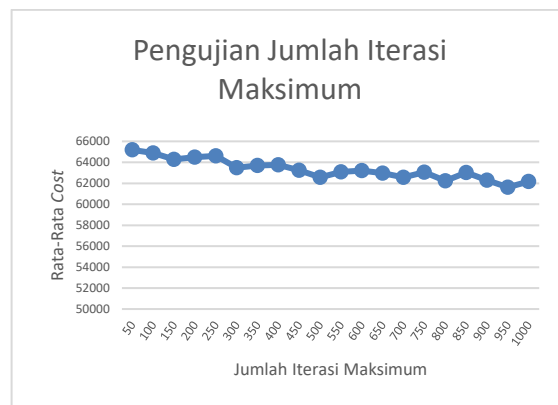


Gambar 7. Grafik Hasil Pengujian Nilai Bobot Inersia

6.5. Pengujian Jumlah Iterasi Maksimum

Pengujian ini digunakan untuk menentukan jumlah iterasi maksimum demi menghasilkan interval paling optimum dalam *Fuzzy Time Series*. Untuk pengujian ini jumlah iterasi yang digunakan adalah 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, dan 1000. Grafik hasil pengujian jumlah iterasi maksimum dapat dilihat pada Gambar 8.

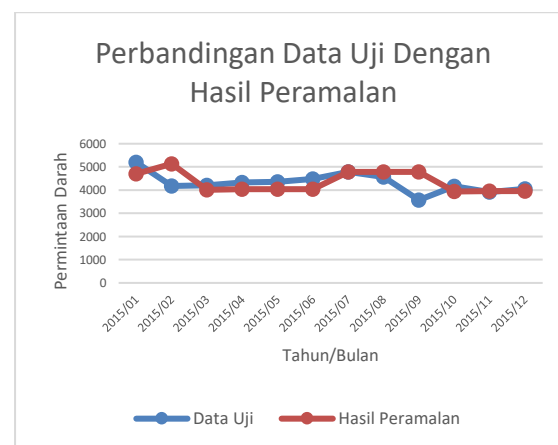
Berdasarkan grafik hasil pengujian pada Gambar 8, rata-rata nilai *cost* terendah didapatkan dari jumlah iterasi 950. Rata-rata nilai *cost* tidak menurun seiring dengan bertambahnya jumlah iterasi. Jumlah iterasi yang besar dapat menurunkan nilai *cost* partikel karena memberikan waktu bagi partikel untuk bergerak menuju titik optimal. Tetapi jumlah iterasi yang terlalu besar tidak menjamin nilai *cost* akan menurun, karena terdapat kemungkinan partikel yang sudah berada di dekat titik optimum akan bergerak menjauh dari titik optimum tersebut.



Gambar 8. Grafik Hasil Pengujian Jumlah Iterasi Maksimum

6.6. Hasil Penerapan Parameter Optimum

Berdasarkan pengujian yang dilakukan terhadap 12 data uji yang telah ditentukan sebelumnya, tingkat kesalahan peramalan (MAPE) yang dihasilkan oleh sistem adalah 7.50330%. Pada Gambar 9 ditunjukkan grafik perbandingan antara data uji dengan hasil peramalannya.



Gambar 9. Perbandingan Data Uji Dengan Hasil Peramalan

7. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan dalam penelitian optimasi interval *fuzzy time series* menggunakan *particle swarm optimization* pada peramalan permintaan darah, berdasarkan hasil pengujian yang diperoleh, nilai akurasi yang dihasilkan oleh sistem dalam penelitian optimasi interval *fuzzy time series* menggunakan *particle swarm optimization* pada peramalan permintaan darah adalah sebesar 92.49670% dengan tingkat kesalahan (MAPE) sebesar 7.50330% yang didapatkan dari hasil perhitungan kesalahan dari data aktual dengan hasil peramalan pada 12 data uji.

Selain itu dapat disimpulkan bahwa besaran nilai parameter *particle swarm optimization* yang berbeda pada tiap kali pengujian memengaruhi nilai *cost* yang dihasilkan oleh sistem. Dari beberapa kali pengujian yang telah dilakukan dalam penelitian ini, didapatkan besaran parameter yang dapat memberikan interval optimum dalam peramalan *fuzzy time series*. Nilai parameter tersebut adalah 40 untuk jumlah partikel, 30 untuk jumlah dimensi, 1.5 dan 1.5 masing-masing untuk kombinasi nilai c_1 dan c_2 , 0.3 untuk bobot inersia, dan 400 untuk jumlah iterasi maksimum.

DAFTAR PUSTAKA

- AlRashidi, M. R. & El-Naggar, K. M., 2010. Long Term Electric Load Forecasting based on Particle Swarm Optimization. *Applied Energy*, pp. 320-326.
- Badan Pusat Statistik Kota Malang, 2011. *MALANG DALAM ANGKA: Malang City in Figures 2011*. Malang: Badan Pusat Statistik Kota Malang.
- Badan Pusat Statistik Kota Malang, 2012. *MALANG DALAM ANGKA: Malang City in Figures 2012*. Malang: Badan Pusat Statistik Kota Malang.
- Badan Pusat Statistik Kota Malang, 2013. *MALANG DALAM ANGKA: Malang City in Figures 2013*. Malang: Badan Pusat Statistik Kota Malang.
- Badan Pusat Statistik Kota Malang, 2014. *MALANG DALAM ANGKA: Malang City in Figures 2014*. Malang: Badan Pusat Statistik Kota Malang.
- Badan Pusat Statistik Kota Malang, 2015. *MALANG DALAM ANGKA: Malang City in Figures 2015*. Malang: Badan Pusat Statistik Kota Malang.
- Badan Pusat Statistik Kota Malang, 2016. *MALANG DALAM ANGKA: Malang City in Figures 2016*. Malang: Badan Pusat Statistik Kota Malang.
- Belien, J. & Force, h., 2012. Supply Chain Management of Blood Products: A Literature Review. *European Journal of Operational Research*, 217(1), pp. 1-16.
- Efendi, R., Ismail, Z. & Deris, M. M., 2015. A New Linguistic Out-Sample Approach of Fuzzy Time Series for Daily Forecasting of Malaysian Electricity Load Demand. *Applied Soft Computing*, pp. 422-430.
- Febriana, W. E. & Widodo, D. A., 2012. Analisis Peramalan Kombinasi terhadap Jumlah Permintaan Darah di Surabaya (Studi Kasus: UDD PMI Kota Surabaya). *JURNAL SAINS DAN SENI ITS*, Volume 1, pp. 20-24.
- Filho, O. S. S., Cezarino, W. & Salviano, G. R., 2012. *A Decision-making Tool for Demand Forecasting of Blood Components*. Bucharest, ScienceDirect, pp. 1499-1504.
- Hsu, L.-Y. et al., 2010. Temperature Prediction and TAIFEX Forecasting Based on Fuzzy Relationship and MTPSO techniques. *Expert Systems with Applications*, pp. 2756-2770.
- Huang, Y.-L. et al., 2011. A Hybrid Forecasting Model for Enrollments based on Aggregated Fuzzy Time Series and Particle Swarm Optimization. *Expert Systems with Applications*, pp. 8014-8023.
- Marini, F. & Walczak, B., 2015. Particle Swarm Optimization (PSO). A Tutorial. *Chemometrics and Intelligent Laboratory Systems*, Volume 149, pp. 153-165.
- Rutherford, C., Cheng, S. Y. & Bailie, K., 2016. *Evidence of Bullwhip in the Blood Supply Chain*. Trondheim, Heriot-Watt University Research Gateway.
- Saxeena, P., Sharma, K. & Easo, S., 2012. Forecasting Enrollments based on Fuzzy Time Series with Higher Forecast Accuracy Rate. *International Journal of Computer Technology & Applications*, p. 957961.
- Singh, P. & Borah, B., 2014. Forecasting Stock Index Price Based on M-Factors Fuzzy Time Series and Particle Swarm Optimization. *International Journal of Approximate Reasoning*, pp. 812-833.