

# Optimisasi Waktu Tempuh Pada Multi-AGV Menggunakan *Particle Swarm Optimization*

Anugrah Kusumo Pamosoaji<sup>1</sup>

**Abstract.** An algorithm for velocity planning on a continuous-curvature path of a pair of automated guided vehicles (AGVs) that minimizing its travelling time is presented. A class of 3-degree Bezier curves is used as basic form of the path. In addition, constraints of maximum allowable linear and radial accelerations are considered. The velocity plan algorithm is generated based on the characteristics of the path's control points and maximum allowable radial velocity on some the path's points. A set of properties of the allowable radial velocity is discussed. The verification of the new algorithm is revealed in the simulation results.

**Keywords.** AGV, material handling, velocity planning, travelling time optimization

**Abstrak.** Suatu algoritma untuk perencanaan kecepatan pada suatu lintasan kurva kontinu dari sepasang automated guided vehicle (AGV) yang meminimalkan waktu perpindahan ditunjukkan pada artikel ini. Sebuah kelas dari kurva Bezier derajat 3 digunakan sebagai bentuk dasar lintasan. Selibuhnya, batasan maksimal yang diperkenankan pada percepatan linier dan radial diberikan. Algoritma perencanaan kecepatan dibangkitkan berdasarkan karakteristik pada titik kendali lintasan dan kecepatan radial maksimum yang diperkenankan pada titik lintasan tersebut. Diskusi dilakukan pada seperangkat sifat kecepatan radial yang diperkenankan. Verifikasi dari algoritma baru dilakukan dengan hasil simulasi.

**Kata kunci.** AGV, penanganan material, perencanaan kecepatan, optimisasi waktu pemindahan.

## I. PENDAHULUAN

Penanganan material merupakan bagian penting dalam manufaktur. Manufaktur berskala besar biasanya membutuhkan sistem penanganan material yang cepat, tepat, dan aman. Keamanan pada penanganan material menjadi isu yang penting ketika penggunaan otomatisasi diberlakukan. Pada penggunaan *automated guided vehicle* (AGV), aspek keamanan menjadi penting karena AGV sendiri merupakan kendaraan yang dapat bergerak bebas. Ini berbeda dengan *automated conveyor* yang pergerakannya terbatas pada jalur yang telah ditetapkan. Penggunaan AGV dalam penanganan material seperti pada *warehouse* lazimnya menggunakan mekanisme navigasi *line follower*. Mekanisme ini sangat efektif.

AGV dapat dimodelkan sebagai sebuah robot beroda yang memiliki mekanisme gerak tertentu, yang merupakan kombinasi dari kecepatan traksi (lurus ke depan) dan kecepatan radial (gerak berputar), tergantung konfigurasi rodanya. Beberapa konfigurasi roda yang digunakan diantaranya adalah konfigurasi *rear-steered wheel* dengan aktuator *steer* berada di pinggir dan (Tamba & Hong, 2009) dan *rear-steered wheel* dengan aktuator *steer* berada di tengah (Widyotriatmo & Hong, 2011). Pada prinsipnya, agar sebuah AGV dapat bergerak secara otomatis, AGV tersebut harus mempunyai sistem kendali yang mampu mengendalikan pergerakannya dari titik awal menuju tujuan tanpa mengalami tabrakan dengan obyek lain (LaValle, 2006).

Ada dua tujuan yang hendak dicapai dari penelitian ini. Pertama, penelitian ini bertujuan mengembangkan sebuah algoritma untuk menentukan jalur yang dilalui oleh sebuah AGV dari konfigurasi (didefinisikan sebagai posisi dan orientasi) awal menuju konfigurasi tujuan. Kedua, membangun sebuah algoritma untuk menentukan kecepatan pada titik-titik sampling pada jalur yang telah ditentukan tersebut sedemikian rupa sehingga waktu yang ditempuh oleh AGV

---

<sup>1</sup> Anugrah Kusumo Pamosoaji, Program Studi Teknik Industri, Universitas Atma Jaya Yogyakarta, Kampus III Gedung Bonaventura, Jl. Babarsari No. 44, Yogyakarta, Indonesia (email: kusumo\_pamosoaji@mail.uajy.ac.id)

menjadi seminimum mungkin.

Penelitian ini merupakan studi awal menuju pembuatan perangkat lunak untuk memberikan rekomendasi pada manajemen pergudangan apabila akan menerapkan AGV dalam operasional pergudangan. Dengan menggunakan perangkat lunak yang akan didesain, diharapkan dalam sebuah gudang/warehouse, tidak ada lagi AGV yang berhenti di tengah jalan ketika menjalankan sebuah misi *pick-and-place*.

*Automated Guided Vehicle* (AGV) merupakan kendaraan yang didesain khusus untuk keperluan memindahkan material dari satu lokasi ke lokasi lain secara otomatis. Produk-produk AGV yang ada di pasaran banyak yang dikendalikan menggunakan komputer *workstation* yang berkomunikasi dengan AGV menggunakan *Local Area Network* (Aoki dkk, 2013). Komputer akan memberikan instruksi kepada semua AGV yang sedang beroperasi. Instruksi-instruksi tersebut merupakan keputusan komputer setelah melakukan kalkulasi dari data-data tentang situasi ruang kerja terkini.

Penelitian tentang pengendalian AGV sudah banyak dilakukan. Beberapa diantaranya adalah penelitian yang dilakukan oleh Peng dan Akella (2005), Tamba, dkk. (2009), dan Widyotriatmo dan Hong (2011). Namun demikian, penelitian-penelitian tersebut belum melibatkan perencanaan jalur dan kecepatan. Beberapa hasil penelitian tentang perencanaan jalur diantaranya adalah penelitian dari Haddad dkk. (2010), Skrzjanc dan Klancar (2010), dan Dolinskaya dan Maggiar (2012). Haddad dkk. (2010) memusatkan penelitiannya pada kalkulasi dan *plotting* kecepatan robot beroda satu dengan halangan berupa batasan kecepatan maksimum. Penelitian ini tidak mempertimbangkan penghalang lain berupa percepatan radial yang secara natural berkaitan dengan radius tikungan maksimum.

Percepatan radial digunakan dalam penelitian yang dilakukan oleh Skrzjanc dan Klancar (2010). Penelitian ini menggunakan kurva Bezier derajat tiga dengan mengidentifikasi besarnya kecepatan radial maksimum kecepatan traksi maksimum yang diinginkan pada setiap titik sampel pada jalur.

*Particle Swarm Optimization* (PSO) yang digunakan untuk mencari waktu tempuh minimum dari AGV yang lebih lambat pertama kali diperkenalkan oleh Clerc dan Kennedy

(2002). Konsepnya mengadopsi interaksi antar partikel dalam mekanika. Sebuah partikel mewakili satu set variabel dan kombinasi nilai variabel-variabel tersebut merupakan input dari sebuah fungsi obyektif. Nilai variabel-variabel dalam setiap partikel akan diubah sedemikian sehingga partikel tersebut mendekati partikel lain yang memiliki nilai *fitness* terendah. Proses iterasi ini akan berulang sedemikian sehingga mayoritas partikel akan berkumpul pada sebuah posisi yang nilai *fitness*-nya terendah. Partikel dengan nilai *fitness* terendah akan menjadi solusi terbaik dari masalah optimisasi.

## II. METODOLOGI

### Definisi masalah

Secara formal, masalah yang hendak dipecahkan adalah sebagai berikut: didefinisikan  $x$ ,  $y$ , dan  $\theta$  sebagai proyeksi posisi AGV pada sumbu  $x$  global, sumbu  $y$  global, dan sudut orientasi AGV menurut sumbu  $x$ . AGV tersebut harus mencapai sebuah orientasi tujuan  $(x_g, y_g, \text{ dan } \theta_g)$  dengan  $x_g$ ,  $y_g$ , dan  $\theta_g$  adalah proyeksi posisi titik tujuan pada sumbu  $x$  global, sumbu  $y$  global, dan sudut orientasi tujuan AGV menurut sumbu  $x$  global. Selain itu, algoritma ini harus mampu menetapkan besarnya kecepatan pada beberapa titik sampling pada jalur sehingga kedua AGV tersebut tidak bertabrakan. Selain itu juga waktu yang ditempuh oleh AGV yang lebih lama dapat diminimalkan. Secara formal, obyektif dari penelitian ini adalah mencari jalur dan plot kecepatan semua AGV sehingga fungsi obyektif  $J$  dapat diminimalkan, dengan  $J$  dirumuskan sebagai berikut:

$$J = \min(\max\{T_1 - T_2\}) \quad \dots(1)$$

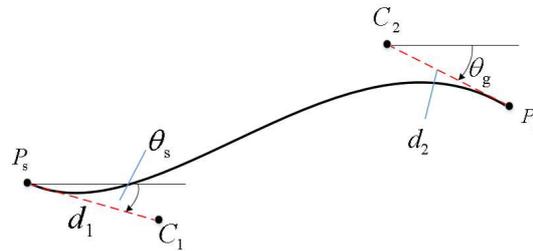
dengan  $T_1$  dan  $T_2$  adalah waktu tempuh AGV-1 dan AGV-2, dengan beberapa kondisi pembatas sebagai berikut:

$$P \leq Q \quad \dots(2)$$

dimana,  $P = [v_r \ v_t \ a_r \ a_t \ \bar{\rho}]^T$  dan  $Q = [\bar{v}_r \ \bar{v}_t \ \bar{a}_r \ \bar{a}_t \ \bar{\rho}]^T$ ,  $a_r$  dan  $\bar{a}_r$  adalah percepatan radial dan percepatan radial maksimum;  $a_t$  dan  $\bar{a}_t$  adalah percepatan traksi dan percepatan traksi maksimum;  $v_r$  dan  $\bar{v}_r$  adalah kecepatan radial dan kecepatan radial maksimum yang dirumuskan dengan

$$v_r = \sqrt{a_r/\kappa} \quad \dots(3)$$

dengan  $\kappa$  adalah kurvatur pada sebuah titik sampel;  $v_t$  dan  $\bar{v}_t$  adalah kecepatan traksi dan



Gambar 1. Kurva Bezier derajat tiga.

kecepatan traksi maksimum;  $\rho$  dan  $\bar{\rho}$  adalah jarak antar-AGV dan jarak antar-AGV maksimum. Penerapan batasan-batasan kecepatan traksi, kecepatan radial, percepatan radial, dan percepatan traksi tersebut sesuai dengan kenyataan di lapangan, yaitu bahwa forklift sendiri mempunyai karakteristik fisik yang membatasi pergerakannya dan juga batasan-batasan yang dibuat untuk menjamin keamanan saat operasi.

**Model jalur berbasis kurva Bezier derajat tiga**

Untuk mengkalkulasi jalur yang dilalui oleh AGV, studi ini menggunakan sebuah model kurva yang dinamakan kurva Bezier derajat tiga. Kurva ini dirumuskan sebagai berikut:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_0 + a_1\xi + a_1\xi^2 + a_2\xi^3 \\ b_0 + b\xi + b_1\xi^2 + b_2\xi^3 \end{bmatrix} \dots(4)$$

dengan  $(x, y)$  merupakan posisi titik pada kurva yang harus dilalui oleh AGV dan  $\xi$  adalah parameter kurva yang nilainya berada pada interval  $[0,1]$ , sehingga  $\xi = 0$  adalah nilai parameter untuk titik awal kurva dan  $\xi = 1$  adalah nilai parameter untuk titik akhir kurva. Dengan demikian dapat dikatakan bahwa sebuah AGV akan bergerak dari  $\xi = 0$  ke  $\xi = 1$ . Parameter-parameter  $a_0 \dots a_2$  dan  $b_0 \dots b_2$  adalah konstan untuk sebuah jalur.

Sebuah kurva Bezier derajat tiga mempunyai empat titik kendali, seperti yang diperlihatkan pada Gambar 1, yaitu  $P_s, P_g, C_1,$  dan  $C_2$ . Titik kendali  $P_s$  dan  $P_g$  adalah titik kendali yang berdempetan dengan titik awal (diparameterisasi dengan  $\xi = 0$ ) dan titik akhir (diparameterisasi dengan  $\xi = 1$ ). Dengan demikian kedua titik tersebut tidak berubah posisinya. Titik  $C_1$  dan  $C_2$  adalah titik kendali dinamis yang membentuk jarak  $d_1$  dan  $d_2$  dengan  $P_s$  dan  $P_g$  dan sudut  $\theta_s$  dan  $\theta_g$ . Pada dasarnya, titik  $C_1$  dan  $C_2$  sangat

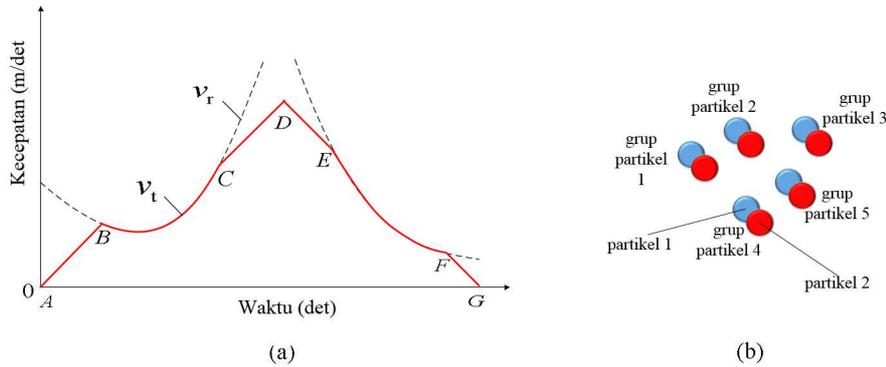
menentukan bentuk kurva, termasuk juga kurvatur di setiap titiknya. Akan tetapi posisi  $C_1$  dan  $C_2$  ini dapat digantikan dengan variabel jarak  $d_1$  dan  $d_2$ . Dengan demikian,  $d_1$  dan  $d_2$  dapat digunakan menjadi variabel penentu pada pembentukan kurva Bezier derajat tiga.

Asumsikan bahwa nilai  $d_1$  dan  $d_2$  dari sebuah jalur telah ditentukan (yang berarti bentuk jalur sudah terdefinisi). Sebelum menentukan besarnya kecepatan traksi, maka terlebih dahulu ditentukan titik-titik sampel pada lokasi-lokasi tertentu pada jalur. Oleh karena itu perlu ditetapkan berapa sampel yang perlu dibuat. Semakin banyak sampel maka akurasi penentuan waktu tempuh akan lebih akurat. Dalam studi ini, metode sampling dilakukan dengan menentukan bahwa titik-titik yang disampling adalah titik-titik dengan parameter  $\xi = k/N$ , dengan  $k = 0,1,2, \dots, N$  dan  $N$  adalah jumlah sampel.

**Penentuan kecepatan traksi dan waktu tempuh**

Setelah sebuah jalur berhasil ditentukan, maka langkah selanjutnya adalah menentukan besarnya kecepatan traksi untuk setiap titik sampling pada jalur yang memenuhi batasan pada persamaan (2). Langkah-langkah untuk menentukan kecepatan traksi adalah sebagai berikut:

1. Tentukan besarnya kurvatur pada setiap titik sampling untuk jalur yang telah ditentukan.
2. Tentukan besarnya kecepatan radial maksimum yang diperbolehkan, menurut persamaan (3).
3. Identifikasi titik-titik sampling yang memiliki percepatan traksi sama dengan  $\bar{a}_t$ . Titik-titik ini akan menjadi batas antara area dengan  $v_t > \bar{v}_r$  (area B-C dan E-F pada Gambar 2(a)) dengan area dengan  $v_t < \bar{v}_r$  (area A-B dan C-D-E pada Gambar 2(a)).



Gambar 2. (a) Konsep plot kecepatan sebuah AGV; (b) Konsep *particle-group swarm optimization*

4. Dengan memulai pemindaian pada setiap titik sampel dimulai dari kecepatan traksi awal  $v_t = 0$ , terapkan percepatan traksi dengan percepatan maksimum  $a_t = \bar{a}_t$  sampai pada titik di mana  $v_t > \bar{v}_r$ .
5. Setelah tiba pada titik batas  $v_t > \bar{v}_r$  dan  $v_t < \bar{v}_r$  (misalnya di titik B), terapkan  $v_t = \bar{v}_r$  sampai pada batas area berikutnya (mulai dari B sampai C).
6. Jika sampai pada titik di mana perlambatan yang dibutuhkan untuk mencapai titik tujuan adalah  $-\bar{a}_t$ , maka terapkan kecepatan traksi dengan percepatan traksi  $a_t = -\bar{a}_t$ .
7. Jika dalam kecepatan traksi hasil perhitungan melebihi kecepatan traksi maksimum  $\bar{v}_t$ , maka nilai  $v_t$  diganti menjadi  $v_t = \bar{v}_t$ .
8. Proses penentuan kecepatan traksi dilakukan per waktu sampling, sehingga waktu tempuh  $T_1$  dan  $T_2$  untuk AGV-1 dan AGV-2 dapat langsung dikalkulasi dengan menggunakan banyaknya langkah pemindaian yang dilakukan dari titik awal ke titik akhir.

**Pencarian waktu tempuh terpendek berbasis GBPSO**

Algoritma *particle swarm optimization-based search* yang digunakan adalah *group-based particle swarm optimization* (GBPSO) (Pamosoaji & Hong, 2015). Pada GBPSO, ‘partikel’ yang diperkenalkan oleh Clerc dan Kennedy (2002) dimodifikasi menjadi sebuah ‘grup partikel’. Setiap grup partikel terdiri atas partikel yang jumlahnya sama dengan jumlah AGV. Setiap partikel mewakili parameter-parameter jalur AGV, yaitu  $d_1$  dan  $d_2$ . Visualisasi dari *particle-group swarm optimization* diperlihatkan pada Gambar 2(b). Langkah-

langkah pencarian waktu tempuh terpendek untuk AGV yang lebih lambat adalah sebagai berikut:

1. Untuk setiap grup partikel, masing-masing partikel melakukan langkah-langkah penentuan jalur yang akan dilewati oleh AGV yang berkaitan, dengan menggunakan rumus berikut:

$$\Delta d_{1,next} = c_0 \Delta d_{1,previous} + c_1 r_1 (d_1^{best-local} - d_1) + c_2 r_2 (d_1^{best-global} - d_1) \dots(5a)$$

$$\Delta d_{2,next} = c_0 \Delta d_{2,previous} + c_1 r_1 (d_2^{best-local} - d_2) + c_2 r_2 (d_2^{best-global} - d_2) \dots(5b)$$

$$d_{1,next} = d_{1,previous} + \Delta d_{1,next} \dots(6a)$$

$$d_{2,next} = d_{2,previous} + \Delta d_{2,next} \dots(6b)$$

Kemudian waktu tempuh dikalkulasi dengan menggunakan langkah-langkah 1) sampai 8) yang telah dibahas.

2. Jika ada titik sampling AGV-1 dan AGV-2 yang tidak memenuhi batasan jarak  $\rho > \bar{\rho}$ , maka pencarian  $d_1$  dan  $d_2$  dilanjutkan untuk grup partikel berikutnya.
3. Untuk setiap iterasi, waktu tempuh terlama pada setiap grup partikel dikalkulasi, kemudian dicari grup partikel yang nilai *fitness*-nya paling kecil, dikalkulasi dengan persamaan (1).
4. Nilai parameter jalur kedua AGV dengan waktu tempuh AGV terlama yang paling pendek menjadi solusi optimal.

### III. HASIL DAN PEMBAHASAN

Simulasi dilakukan dengan skenario yang ditunjukkan pada Tabel 1. Dengan menggunakan metode penentuan jalur berbasis kurva Bezier berderajat tiga, penentuan kecepatan traksi, dan pencarian waktu tempuh terpendek untuk AGV yang terlama, maka didapat jalur AGV yang bebas tabrakan seperti pada Gambar 3. Sedangkan plot kecepatan traksi yang dihasilkan ditunjukkan pada Gambar 4.

Simulasi dilakukan 10 kali dengan menggunakan perangkat lunak simulasi MATLAB. Jumlah grup partikel yang digunakan dalam simulasi ini adalah 20 grup partikel dan jumlah iterasi 30 kali. Dari Tabel 2 dapat dilihat bahwa dengan metode GBPSO, didapat jalur-jalur untuk AGV-1 dan AGV-2 yang masing-masing ditempuh dalam  $25,79 \pm 0,137$  m dan

$25,96 \pm 0,23$  cm. Dengan simpangan baku pada waktu tempuh yang berada di bawah 1%, dapat disimpulkan bahwa metode GBPSO sukses menghasilkan jalur dan kecepatan traksi untuk AGV-1 dan AGV-2 yang konvergen ke waktu tempuh untuk AGV terlama yang minimum.

Tabel 2 memberikan informasi bahwa jarak minimum antar AGV yang dihasilkan seluruhnya berada di atas batas minimum jarak antar AGV yang dihasilkan, yaitu 10 m, dengan rata-rata  $10,863 \pm 0,23$  m. Dengan demikian, metode GBPSO cukup aman untuk diterapkan pada kasus 2 AGV.

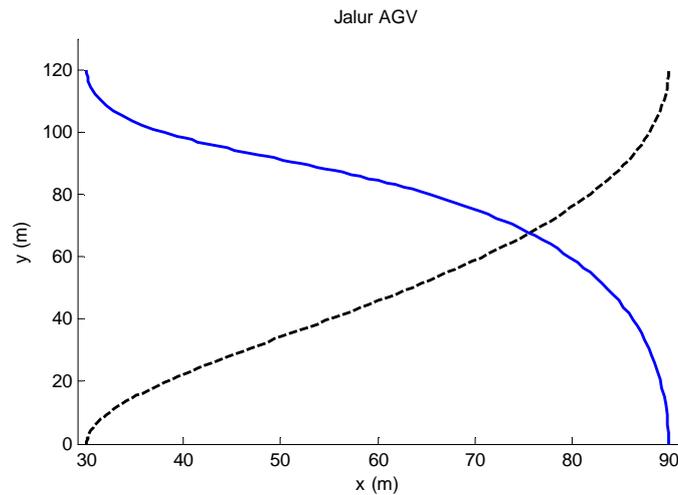
Nilai parameter jalur AGV-1 berada pada  $d_1 = 77,91 \pm 14,17$  cm dan  $d_2 = 60,29 \pm 9,60$  dan nilai parameter jalur untuk AGV-2 berada pada  $d_1 = 73,56 \pm 14,32$  dan  $d_2 = 87,65 \pm 11,64$ . Nilai simpangan baku yang lebih besar dari 10%

Tabel 1. Skenario simulasi untuk 2 AGV

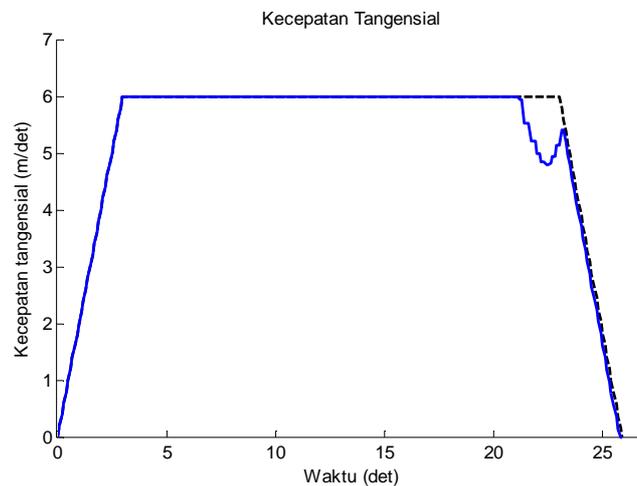
	AGV-1	AGV-2
Jarak minimum antar AGV yang diizinkan: 10 m		
Kecepatan traksi max	6 m/det	6 m/det
Percepatan traksi max	2 m/det <sup>2</sup>	2 m/det <sup>2</sup>
Percepatan radial max	1 m/det <sup>2</sup>	1 m/det <sup>2</sup>
Posisi awal	(30 m, 0 m)	(90 m, 0 m)
Orientasi awal	90°	90°
Posisi akhir	(90 m, 120 m)	(30 m, 120 m)
Orientasi akhir	90°	90°
Kecepatan awal	0 m/det	0 m/det
Kecepatan akhir	0 m/det	0 m/det

Tabel 2. Hasil simulasi

Simulasi ke-	AGV-1				AGV-2				Jarak minimum antar AGV (m)
	$d_1$ (cm)	$d_2$ (cm)	Panjang jalur (m)	Waktu tempuh (det)	$d_1$ (cm)	$d_2$ (cm)	Panjang jalur(m)	Waktu tempuh (det)	
1	28.86	66.79	137.41	25.90	107.40	42.24	141.98	25.30	10.16
2	107.69	62.82	143.73	25.90	83.46	123.59	148.54	25.40	11.28
3	123.35	86.72	148.93	25.40	40.18	57.61	137.40	25.80	10.08
4	106.99	23.04	141.83	26.20	55.55	69.12	139.03	26.10	10.55
5	29.79	29.65	135.58	25.50	57.80	137.57	148.7258	26.00	11.52
6	5.17	5.40	134.26	25.30	64.22	140.15	149.87	25.80	10.67
7	61.43	71.44	139.61	26.20	134.16	84.35	150.86	25.80	12.22
8	77.52	80.50	141.76	26.50	150.99	106.98	159.08	27.90	11.47
9	136.17	80.75	150.83	25.80	31.63	48.99	136.54	25.70	10.37
10	102.11	95.76	146.54	25.20	10.24	65.93	137.30	25.80	10.31
Rata-rata	77.91	60.29	142.05	25.79	73.56	87.65	144.93	25.96	10.86
Ragam	2006.83	921.78	30.84	0.19	2050.30	1355.41	57.34	0.52	0.51
Simpangan baku	14.17	9.60	1.76	0.137	14.32	11.64	2.40	0.23	0.23



Gambar 3. Contoh jalur AGV hasil kalkulasi



Gambar 4. Contoh kecepatan tangensial hasil kalkulasi: kecepatan AGV-1 (garis putus-putus) dan kecepatan AGV-2 (garis kontinu).

menunjukkan bahwa nilai parameter jalur untuk kedua AGV memiliki banyak variasi, sehingga jalur yang dihasilkan tidak unik. Fakta ini diikuti pula oleh panjang jalur yang simpangan bakunya  $> 20\%$ , sehingga dapat disimpulkan bahwa solusi waktu tempuh AGV terlama yang terpendek tidak ditentukan oleh nilai parameter jalur tertentu.

#### IV. SIMPULAN

Dari hasil 10 kali simulasi MATLAB dapat disimpulkan bahwa metode GBPSO memberikan hasil yang sangat konvergen untuk waktu minimum yang ditempuh oleh AGV yang lebih lama, yang mencapai simpangan baku di bawah

1%. Selain itu jalur yang dihasilkan mampu menjaga jarak antar AGV sehingga tidak berada pada jarak di bawah 10 m. Metode GBPSO ini akan dikembangkan juga untuk kasus-kasus lebih dari dua AGV. Masalah-masalah penentuan jalur dan kecepatan AGV yang juga memperhatikan jarak terpendek juga akan menjadi sasaran penelitian berikutnya.

#### DAFTAR PUSTAKA

- Aoki, K.; Furuno, H.; Nagaoka, J.; Furukawa, K. (2013). 'Autonomous forklift automatic guided vehicle.' *Hitachi Review*, Vol. 62 (4), pp. 285 – 287.
- Bobrow, J.E.; Dubowsky, S.; Gibson, J.S. (1985). 'Time-optimal control for robotic along specified paths.'

- International Journal of Robotics Research*, Vol. 4 (3), pp. 3 – 17.
- Clerc, M.; Kennedy, J. (2002). 'The particle swarm: explosion, stability, and convergence in a multidimensional complex space.' *IEEE Transactions on Evolutionary Computation*, Vol. 6 (1), pp. 58 – 73.
- Gan, Z.; Tao, L.; Ying, Q. (2013). 'Automated guided vehicles dynamic scheduling based on annealing genetic algorithm.' *TELKOMNIKA*, Vol. 11 (5), pp. 2508–2515.
- LaValle, S.M. (2006). *Planning Algorithm*. New York: Cambridge University Press.
- Pamosoaji, A.K.; Hong, K.-S. (2015). *Group-based Particle Swarm Optimization for Multiple-Vehicles Trajectory Planning*, Proceedings of The 15th International Conference on Control, Automation and Systems (ICCAS 2015), pp. 862 – 867.
- Peng, J.; Akella, S. (2005). 'Coordinating multiple robots with kinodynamic constraints along specified paths.' *International Journal of Robotics Research*. Vol. 24 (4), pp. 295 – 310.
- Skrjanc, I.; Klancar, G. (2010). 'Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bezier curves.' *Robotics and Autonomous Systems*, Vol. 58 (1), pp. 1 – 9.
- Tamba, T.A.; Hong, B.; Hong, K.-S. (2009). 'A path following control of an unmanned autonomous forklift,' *International Journal of Control, Automation, and Systems*, Vol. 7 (1), pp. 113 – 122.
- Widyotriatmo, A.; Hong, K.-S. (2011). 'A navigation function-based control of multiple wheeled robots.' *IEEE Transactions on Industrial Electronics*, Vo. 47 (4), pp. 722 – 732.