PENGEMBANGAN ALGORITMA *PARTICLE SWARM OPTIMIZATION* UNTUK OPTIMALISASI DISPERSI *BATCH* PADA PROSES PRODUKSI

Misra Hartati[1], Iwan Vanany[2], Budi Santosa[3]
Jurusan Teknik Industri, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember
Keputih Sukolilo Surabaya Jawa Timur 60111 Telp: +62 (31) 592 3411
Email: misra10@mhs.ie.its.ac.id / misrahartati@yahoo.com

ABSTRAK

Salah satu konsep dan instrumentasi mutu dan keamanan pangan yang disarankan untuk mendukung dan menjamin mutu makanan adalah pemberian informasi lengkap mengenai posisi suatu produk dan jalur distribusi yang ditempuh, sehingga memudahkan upaya pelacakan produk, konsep ini dinamakan dengan sistem traceability. Tujuan melakukan sistem traceability adalah untuk meminimasi biaya karena krisis keamanan pangan. Untuk itu perusahaan harus mempunyai sistem traceability yang baik agar biaya krisis keamanan pangan bisa diminimalisir.

Sistem traceability dapat diklasifikasikan menjadi dua macam, yaitu iternal traceability dan chain traceability. Pada penelitian ini akan membahas mengenai internal traceability yaitu pengaturan batch pada proses produksi. Tujuan penelitian yaitu untuk meminimalisasi batch dispersion dan menentukan kuantitas masing-masing batch yang optimal. Dengan menggunakan pendekatan metaheuristik yaitu algoritma PSO (particle Swarm Optimization) diharapkan dapat menemukan solusi dari fungsi tujuan yang telah ditetapkan. Penelitian dilakukan pada proses produksi sosis di salah satu perusahaan makanan Perancis dengan karakteristik produk 3 level (bahan baku, komponen dan produk jadi).

Dari hasil penelitian yang dilakukan, dengan mengusulkan algoritma PSO didapatkan jumlah dispersi minimal yang terjadi pada proses produksi sosis dengan karakteristik produk 3-level adalah sebesar 20 dispersi (dawnward dispersion = 8 dan upward dispersion = 12). Jadi dapat diartikan bahwa jumlah batch bahan baku yang digunakan pada batch produk jadi dan jumlah batch komponen yang dibeli yang digunakan untuk batch produk jadi adalah 20 dispersi. Dengan jumlah minimal dispersi ini diharapkan dapat mengurangi terjadinya kontaminasi produk dan sekaligus dapat meminimalkan biaya produksi.

Kata kunci: *chain traceability; internal traceability; krisis keamanan pangan; PSO (Particle Swarm Optimization); traceability*

Pendahuluan

Untuk menangani krisis keamanan pangan diperlukan suatu sistem yang bisa memberikan informasi tentang pergerakan pangan dari hulu ke hilir. Salah satu sistem yang bisa digunakan adalah sistem *traceability*, dimana sistem ini akan memberikan informasi-informasi mengenai pergerakan produk. Secara umum tujuan perusahaan melakukan sistem *traceability* yaitu untuk meminimalkan biaya karena krisis keamanan pangan. Jika masalah keamanan pangan berasal dari *batch* bahan baku, perusahaan akan mengidentifikasi dan me-*recall* dari *batch* bahan baku. Jika krisis keamanan pangan berasal dari produk jadi, maka akan diidentifikasi dan di-*recall batch* bahan baku dan *batch* produk jadi. Jadi bisa disimpulkan bahwa untuk miminimalkan biaya krisis keamanan pangan, perusahaan harus meminimalkan jumlah produk *recall*. Masalah penarikan produk akan keamanan pangan ini telah memaksa timbulnya regulasi mengenai *traceability*, khususnya di negara Amerika dan Uni Eropa, bahkan pada *General Food Law Regulation* Uni Eropa (EC No. 178, artikel 18) telah diberlakukan mulai tanggal 1 Januari 2005. Regulasi-regulasi tersebut memperlihatkan elemen-elemen penting, termasuk aturan *traceability* dan penarikan produk berbahaya (*recall procedures*) yang terdapat di pasaran.

Menurut Wang et al. (2010), pengaturan dan ukuran *batch* merupakan salah satu faktor yang menyebabkan terjadinya produk *recall*. Dengan membatasi jumlah *batch* dapat mengurangi masalah keamanan pangan dan dapat mengoptimalkan sistem *traceability*. Dupuy et al. (2005) mengusulkan model optimasi *traceability* dengan mengurangi *batch dispersion*. Penelitian yang

dilakukan oleh Dupuy et al. (2005) hanya bisa menyelesaikan kasus-kasus kecil yang jumlah batch dari bahan baku / komponen sedikit, untuk kasus-kasus yang jumlah *bach*nya besar (jumlah produksi besar), pendekatan yang diusulkan oleh Dupuy et al. (2005) ini tidak memiliki solusi penyelesaian. Oleh karena itu penelitian ini mengusulkan pendekatan metaheuristik untuk menyelesaikan permasalahan dalam penelitian yang dilakukan oleh Dupuy et al. (2005).

Pada penelitian ini akan diusulkan algoritma metaheuristik yaitu *Particle Swarm Optimization* (PSO) untuk menyelesaikan permasalahan optimasi dispersi *batch* pada proses produksi dengan karakteristik 3-*level* (BOM 3-*level*). Pemilihan algoritma ini dikarenakan PSO memiliki kelebihan antara lain mudah diimplementasikan, tidak membutuhkan perhitungan turunan (Marinakis et al., 2010), simpel dan efisien serta apabila dibandingkan dengan *Genetic Algorithm* (GA), PSO lebih cepat menemukan solusi yang mendekati optimal (Shi et al., 2007, Sedighizadeh dan Masehian, 2009).

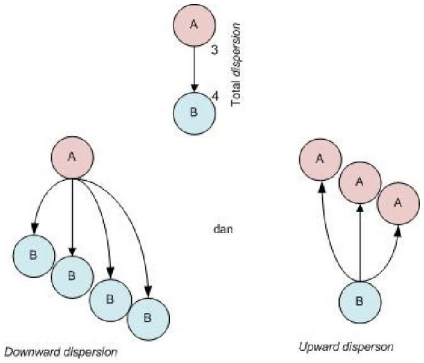
Traceability

Traceability mempunyai beberapa pengertian. Menurut ISO 22005:2007, *traceability* diartikan kemampuan untuk menelusuri pergerakan pakan atau makanan pada tahap produksi, proses dan distribusi. Smith and Furness (2006) mengatakan bahwa *traceability* lebih menekankan pada penelusuran, menarik perhatian terhadap pentingnya mencatat informasi yang penting untuk memuaskan kebutuhan yang ditelusuri.

Sistem *traceability* membatasi pengaruh masalah potensial keamanan pangan, dengan bantuan *traceability* dapat diketahui dengan tepat produk yang mana yang tidak baik dan jaringan *supply* mana yang rumit. Akan tetapi *traceability* sendiri tidak merubah keamanan dan mutu dari produk (Grunow et al. 2008).

Klasifikasi *traceability* terdiri dari *internal traceability* dan *chain traceability*. Menurut Moe (1998) *internal traceability* merupakan penelusuran dengan melacak internal *batch* produk pada satu langkah dalam rantainya, misalnya pada proses produksi. Sedangkan *chain traceability* merupakan penelusuran dengan melacak produk melalui rantai produksi mulai dari panen sampai tansport, penyimpanan, proses, distribusi dan sales.

Batch Dispersion



Gambar 1. Ilustrasi Perhitungan *Downward* dan *Upward Dispersion*
(Lobna dan Mounir, 2011)

Pada proses produksi, untuk mengevaluasi akurasi dari *traceability* diperkenalkan cara pengukuran yang baru yaitu *downward dispersion, upward dispersion* dan *batch dipersion* (Dupuy *et al*. 2002). *Downward disperson* pada *batch* bahan baku merupakan jumlah *batch* produk akhir yang mengandung bagian dari *batch* bahan baku tersebut. *Downward dispersion* (*tracing*) merupakan kapasitas pada *supply chain*, untuk menemukan asal usul dan karakteristik dari suatu produk dari satu atau beberapa kriteria yang ada. *Upward dispersion* pada *batch* produk akhir yaitu jumlah *batch* bahan baku yang digunakan memproduksi *batch* ini. *Upward traceability* (*tracking*) merupakan kapasitas pada *supply chain*, untuk menemukan lokasi dari suatu produk dari satu atau beberapa kriteria yang ada (Dupuy *et al*. 2002). *Batch dispersion* adalah jumlah dari *downward*

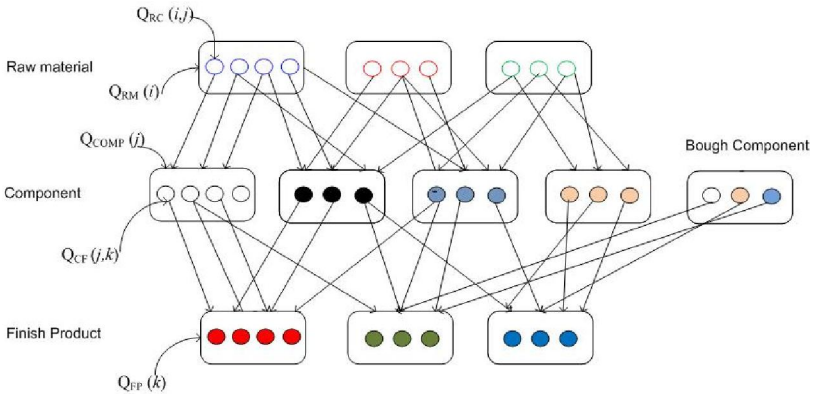
dispersion pada *batch* bahan baku (A) dan *upward dispersion* pada *batch* produk akhir (B) seperti yang dilihatkan pada gambar 1.

Masalah Kompleksitas

Dupuy et.al (2005) mengusulkan model matematika berdasarkan program MILP (*Mixed Integer Linier Programming*) dalam mengoptimasi dispersi pada industri makanan. Pengoptimasian dilakukan dengan bantuan *software* LINGO 6.0. Pendekatan ini hanya bisa digunakan pada permasalahan yang kecil. Sedangkan kenyataannya permasalahan yang terjadi di industri adalah permasalahan yang komplek. Pembuktian bahwa hal ini merupakan permasalahan NP-*complete* diperlihatkan pada persamaan 1.

$$C = E \sum_{j=1}^{j=B_{RM}} \left((Q_{i,j})^n \begin{cases} i = Q_m \\ i = 0 \end{cases} \right. \quad (1)$$

Sebuah perhitungan kompleksitas ditunjukkan jumlah dari kombinasi C. C adalah jumlah kombinasi, E : jumlah level produksi, Q_{max} : maksimum kuantitas, Q_{ij} : jumlah kuantitas bahan baku yang dialokasikan (*integer*), B_{RM}: jumlah *batch* bahan baku, n: kuantitas produk untuk proses produksi, i: indeks bahan baku dan j: indeks produk. Ilustrasi dispersi *batch* pada karakteristik produk 3-level ditunjukkan pada gambar 2.



Gambar 2. Ilustrasi dispersi *batch* pada karakteristik produk 3-level

***Particle Swarm Optimization* (PSO)**

PSO merupakan algoritma berbasis populasi yang mengeksploitasi individu dalam populasi menuju daerah penyelesaian dalam daerah pencarian. Dalam PSO, populasi disebut dengan *swarm* dan individu disebut dengan *particle*. Tiap partikel berpindah dengan kecepatan yang diadaptasi dari daerah pencarian dan disimpan sebagai posisi terbaik yang pernah dicapai. Tahapan pengerjaan PSO adalah sebagai berikut (Santosa dan Willy, 2011):

1. Penentuan nilai parameter
2. Inisialisasi kecepatan dan posisi partikel secara random
3. Evaluasi *fitness* dari masing-masing partikel berdasarkan posisinya
4. Tentukan partikel dengan *fitness* terbaik, dan tetapkan sebagai Gbest.
5. Ulangi langkah berikut sampai *stopping criteria* dipenuhi.

1. Penentuan Nilai Parameter

Parameter yang digunakan dalam algoritma PSO antara lain: N (jumlah partikel yang dibangkitkan), C1 (*learning rates* eksplorasi lokal), C2 (*learnng rates* eksplorasi global), θmin (koefisien inersia minimum), θmax (koefisien inersia maksimum) dan iterasi maksimum.

2. Inisialisasi Kecepatan dan Posisi Partikel Secara Random

- Membangkitkan partikel

Partikel yang akan dibangkitkan merupakan solusi posisi awal [$X_p(0)$] partikel. Partikel ini terdiri dari matrik bilangan biner (matrik penugasan), matrik kuantitas masing-masing *batch* (matrik komposisi) dan matrik untuk *batch* yang dibeli. Permasalahan yang akan diteliti adalah

mengenai dispersi *batch* pada proses produksi, karena proses produksi terdiri dari BOM - 3 *level*, maka pembangkitan partikel ini dilakukan dengan beberapa tahap, yaitu:

a. Membangkitkan matrik penugasan
Matrik ini berupa matrik bilangan biner (0,1) untuk setiap *level*.

- *Level 1-2* à *Batch* bahan baku *i* yang digunakan pada *batch* komponen *j*
Matrik ini merupakan matrik *disassembly*, 1 jika *batch* bahan baku *i* digunakan pada *batch* komponen *j*, 0 jika *batch* bahan baku *i* tidak digunakan pada *batch* komponen *j*.

Tabel 1. Contoh Matrik Penugasan *level* 1-2

	Comp1	Comp2	Comp3	Comp4	Comp5	Comp6
RM1	1	1	1	1	0	1
RM2	1	0	1	0	1	0
RM3	1	0	1	1	1	1
RM4	1	1	1	0	1	1

- *Level 2-3* à *Batch* komponen *j* yang digunakan pada *batch* produk jadi *k*
Matrik ini merupakan matrik *assembly*, 1 jika *batch* komponen *j* digunakan pada *batch* produk jadi *k*, 0 jika *batch* komponen *j* tidak digunakan pada *batch* produk jadi *k*.

Tabel 2. Contoh Matrik Penugasan *level* 2-3

	Comp1	Comp2	Comp3	Comp4	Comp5	Comp6
FP1	1	0	1	0	1	1
FP2	1	1	1	1	1	1
FP3	1	1	0	1	1	0
FP4	0	0	1	0	0	1

b. Menentukan Matrik Kuantitas

Matrik kuantitas ditentukan berdasarkan matrik bilangan biner. Jika matrik bilangan biner terdiri dari angka 1 dan 0, matrik kuantitas terdiri dari kuantitas dari *batch* yang digunakan pada masing-masing level BOM pada proses produksi. Jika pada matrik bilangan biner bernilai 1, maka bilangan pada matrik kuantitas bernilai dan jika pada matrik bilangan biner bernilai 0, maka bilangan pada matrik kuantitas tidak bernilai (0).

- *Level 1-2* à Kuantitas *batch* bahan baku *i* yang digunakan pada *batch* komponen *j*

Tabel 3. Contoh matrik kuantitas untuk *level* 1-2

	Comp1	Comp2	Comp3	Comp4	Comp5	Comp6
RM1	223	240	137	193	0	207
RM2	169	0	551	0	480	0
RM3	288	0	162	103	253	194
RM4	107	254	179	0	247	413
Tot	**787**	**494**	**1029**	**296**	**980**	**814**

Tabel diatas menjelaskan bahwa jumlah bahan baku 1 yang diberikan kepada komponen 1 berjumlah 223 Kg. jumlah bahan baku 1 yang diberikan kepada komponen 2 berjumlah 240 Kg. jumlah bahan baku 2 yang diberikan kepada komponen 1 berjumlah 169 Kg dan seterusnya.

- *Level 2-3* à Kuantitas *batch* komponen *j* yang digunakan pada *batch* produk jadi *k*

Tabel 4. Contoh matrik kuantitas untuk *level* 2-3

	Comp1	Comp2	Comp3	Comp4	Comp5	Comp6
FP1	351	0	649	0	507	493
FP2	421	330	249	292	456	251
FP3	15	164	0	4	17	0
FP4	0	0	131	0	0	70
Tot	**787**	**494**	**1029**	**296**	**980**	**814**

Tabel diatas menjelaskan bahwa kuantitas komponen 1 (Comp1) yang diberikan kepada produk jadi 1 (SFP1) berjumlah 351 Kg. kuantitas komponen 1 (Comp1) yang diberikan kepada produk jadi 2 (SFP2) berjumlah 421 Kg. jumlah komponen 2 (Comp2) yang diberikan kepada produk jadi 2 (SFP2) berjumlah 330 Kg dan seterusnya.

c. Menentukan kuantitas *batch* komponen yang dibeli

Batch komponen yang dibeli ini digunakan jika terjadi kekurangan pada *batch* produk jadi. Jika kekurangan, maka untuk memenuhi kuantitas *batch* produk jadi diambil dari *batch* komponen yang dibeli. Variabel yang mewakili kuantitas dari *batch* komponen yang dibeli adalah variabel biner [X_{BF}( )], 1 jika *batch* komponen yang dibeli digunakan pada *batch* produk jadi *k*, 0 jika *batch* komponen yang dibeli tidak digunakan pada *batch* produk jadi *k*.

3. Evaluasi *fitness* dari masing-masing partikel berdasarkan posisinya

Setelah matrik penugasan dan kuantitas diketahui, kemudiaan dilakukan perhitungan fungsi tujuan yaitu minimasi dari dispersi *batch* (persamaan 2). dari hasil *running* pada software MATLAB didapatkan hasil dari fungsi tujuan (fungsi_gbest) untuk jumlah N= 175 dan iterasi = 75 adalah sebesar 12 dispersi.

$$Z = \sum_{i=1}^{M}\sum_{k=1}^{P} Y(i,k) + \sum_{l=1}^{Q}\sum_{k=1}^{P} X_{BF}(l, \qquad (2)$$

Keterangan:

Y(i,k) = Variabel biner, brnilai 1 jika batch bahan baku i digunakan pada batch produk jadi k dan bernilai 0 jika batch bahan baku i tidak digunakan pada batch produk jadi k

X_{BF} (l,k) = Persamaan variabel biner, bernilai 1 jika batch komponen yang dibeli l digunakan pada batch produk jadi k dan bernilai 0 jika batch komponen yang dibeli l tidak digunakan pada batch produk jadi k

M = Jumlah *batch* bahan baku

P = Jumlah *batch* produk jadi

Q = Jumlah *batch* komponen yang dibeli

i,k, = indeks *batch* bahan baku, produk jadi dan komponen yang dibeli.

4. Tentukan partikel dengan *fitness* terbaik, dan tetapkan sebagai Gbest.

Tabel 3. Nilai Pbest dan Gbest dari fungsi tujuan

	Iterasi 1		Iterasi 2	
N	P_best	Gbest	P_best	Gbest
1	32	32	32	30
2	33		31	
3	33		32	
4	32		34	
5	34		31	
6	34		31	
7	33		30	
8	35		31	
9	32		31	
10	35		30	

Setelah nilai *fitness* didapatkan, kemudian ditentukan P*best* dan G*best* untuk semua partikel pada iterasi pertama. P*best* adalah nilai *fitness* terbaik untuk masing-masing partikel, sedangkan G*best* adalah nilai *fitness* terbaik dari semua partikel pada iterasi pertama. Contoh nilai P*best* dan G*best* dari hasil *running* MATLAB dengan N=10 dan iterasi = 2 bisa dilihat pada tabel 3.

5. Perbarui iterasi, koefisien inersia, kecepatan dan posisi partikel

Jika solusi optimal belum ditemukan pada iterasi awal, maka dilakukan pembaharuan iterasi. Pembaharuan iterasi bisa dilakukan dengan menggunakan persamaan-persamaan berikut (Santosa dan Willy, 2011):

Koefisien inersia

$$\theta(t) = \theta_{max} - \left(\frac{\theta_{max}-\theta_{min}}{t_{max}}\right) \quad (3)$$

Keterangan:

: koefisien inersia

θ_i : koefisien inersia maksimum

θ : koefisien inersia minimum

t_{max} : iterasi maksimum

t : iterasi

***Update* kecepatan**

$$V_p(t) = \theta(t)V_p(t-1) + c_1 r_1 \left(X_p^L - X_p(t-1)\right) + c_2 r_2 (X^G - X_p(t- \quad (4)$$

***Update* posisi partikel**

$$X_i(t) = V_i(t) + X_i(t- \quad (5)$$

X = Posisi partikel

V = Kecepatan partikel

i = indeks partikel

t = iterasi ke-t

N = ukuran dimensi ruang

$X_i^L = x_{i1}^L, x_{i2}^L, \ldots,$ à *local best*

$X^G = x_1^G, x_2^G, \ldots.$ à *global best*

c_1 = *learning factor*

r = bilangan random (0-1)

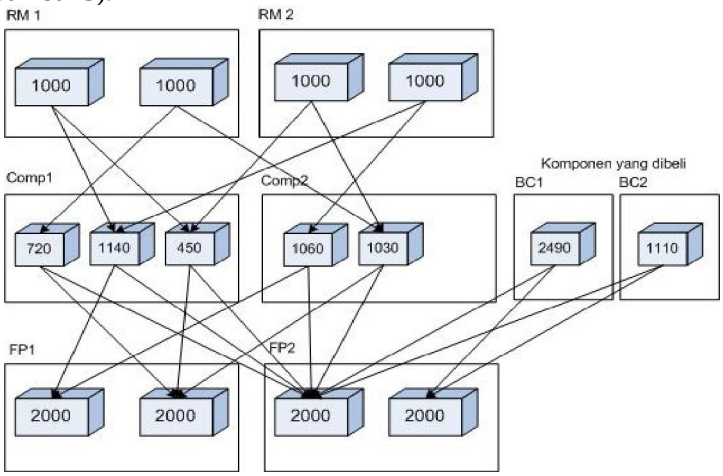
6. Ulangi langkah berikut sampai *stopping criteria* dipenuhi.

Stopping kriteria dilakukan dengan melihat selisih diantara solusi yang sekarang dengan solusi sebelumnya, yaitu apakah sudah mendekati nol atau tidak, jika sudah maka iterasi dihentikan.

Hasil

Disajikan 2 tipe bahan baku, 2 tipe komponen dan 2 tipe produk jadi pada proses produksi sosis. Jumlah *bach* masing-masing level yaitu 4 *batch* bahan baku (RM), 6 *batch* komponen (Comp) dan 4 *batch* produk jadi (FP). Tahap awal yang dilakukan adalah pembuatan algoritma (koding) berdasarkan formulasi matematis yang diusulkan oleh Dupuy, et al (2005). Fungsi tujuan dari formulasi matematis yang diusulkan oleh Dupuy, et al (2005) adalah minimalisasi dari dispersi *batch*. Dari hasil *running* dengan menggunakan *software* MATLAB diketahui jumlah minimal dispersi yang terjadi adalah sebesar 12 dispersi. Output dari running MATLAB ini ditunjukkan

pada sebuah gambar ilustrasi dispersi *batch* yang terjadi pada proses produksi dengan karakteristik produk 3-*level* (Gambar 3).



Gambar 3. Solusi optimal yang diperoleh dengan algoritma PSO

Dengan bantuan gambar di atas akan terlihat *dawnward* dan *upward dispersion* yang terjadi pada proses produksi sosis. Dengan menggunakan persamaan 3 dan 4 didapatkan jumlah dari *dawnward dispersion* sebesar 8 dan *upward dispersion* sebesar 12, jadi total dispersi yang terjadi pada proses produksi sosis dengan karakteristik 3-*level* adalah sebesar 20 dispersi. Perhitungan *dawnward* (D_D_i) dan *upward dispersion* (U_D_i) dapat dilihat dari persamaan 3 dan 4:

$$D\_DISP(i) = \sum_{k=1}^{P} Y(i, \quad (3)$$

$$U\_DISP(k) = \sum_{i=1}^{M} Y(i,k) + \sum_{i-1}^{Q} x_{RF}(l,k) \quad (4)$$

Kesimpulan dan Saran

Model algoritma metaheuristik yang diusulkan adalah algoritma *particle swarm optimization* (PSO). Dengan bantuan algoritma ini bisa dilakukan perhitungan dispersi *batch* untuk masalah yang lebih kompleks. Dapat diketahui bahwa hasil perhitungan optimasi dispersi *batch* dengan menggunakan algoritma PSO adalah sebesar 20 dispersi yang terdiri dari 12 *dawnward dispersion* dan 8 *upward dispersion*. Dengan meminimasi *batch* dispersion diharapkan mampu memudahkan dalam melakukan *tracking dan tracing* terhadap produk jika terjadi kontaminasi dan juga dapat meminimasi produk *recall*.

Dalam penelitian ini belum mempertimbangkan faktor biaya. Oleh karena itu, peneliti mengharapkan untuk penelitian selanjutnya mempertimbangkan faktor biaya sehingga bisa terlihat seberapa besar pengurangan biaya yang terjadi dalam proses produksi dengan melakukan pengendalian terhadap dispersi *bach*.

Daftar Pustaka

Dupuy C., Botta-Genoulaz V., Guinet A., (2005), "Batch Dispersion Model to Optimize Traceability in Food Industry, *Journal of Food Engineering*, Vol. 70, hal. 333-339.

Grunow M, Rong A, and Akkerman R., (2008), "Reducing Dispersion in Food Distribution", *Proceedings of the 9th Asia Pasific Industrial Engineering and Management System Conference*, hal. 618-628.

Lobna K. and Mounir B., (2011), "A Production Model to Reduce Batch Dispersion and Optimize Traceability", *Journal of IEEE*.

International Organization for Standardization [ISO] 22005, (2007), Traceability in the Feed and Food Chain-General Principles and Basic Requirements For System Design and Implementation, Geneva.

Marinarkis Y., Marinaki M., Dounias G., (2010), "A Hybrid Particle Swarm Optimization Algorithm for The Vehicle Routing Problem", *Engineering Applications of Artificial Intelligence*, Vol. 23, hal. 463–472.

Moe T., 1998, "Perspective on Traceability in Food Manufacture", *Trends in Food Science and Technology*, Vol. 9, hal. 211-214.

Santosa B., Willy P., (2011), "Metoda Metaheuristik: Konsep dan Implementasi", Surabaya, Guna Widya.

Sedighizadeh and Masehian, (2009), "Particle Swarm Optimization Methods, Taxonomy and Applications", *International Journal of Computer Theory and Engineering*, Vol. 1, No. 5.

Shi X.H., Liang Y.C., Lee H.P., Lu C., Wang Q.X., (2007), "Particle swarm Optimization-based Algorithms For TSP and Generalized TSP", *Information Processing Letters*, Vol. 103, hal. 169–176.

Smith I. and Furness A., (2006), "Improving Traceability in Food Processing and Distribution". Woodhead Publishing Limited Cambridge England.

Wang X., Li D., O'brien C., Li Y., (2010), "A Production Planning Model to Reduce Risk and Improve Operations Management" , *International Journal of Production Economics*, Vol. 124 (2), hal. 463–47.