

Pembangkitan Nilai Belief Pada *Dempster-Shafer* Dengan *Particle Swarm Optimization* (PSO) Untuk Penentuan Pasal Kasus Penganiayaan

Merry Gricelya Nababan¹, Rekyan Regasari Mardi Putri², Indriati³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹merrygricelya@gmail.com, ²rekyan.rmp@ub.ac.id, ³indriati.tif@ub.ac.id

Abstrak

Tindak kejahatan terhadap tubuh dan nyawa terus mengalami peningkatan setiap tahunnya, hakim sebagai penentu keputusan terhadap terdakwa kejahatan memiliki peran yang sangat penting dalam memberikan keputusan. Namun, terdapat beberapa hal yang perlu dipertimbangkan oleh hakim dalam memberikan keputusan, sehingga permasalahan ketidakpastian dapat menjadi kendala hakim. Penulis menerapkan sebuah metode yang dapat menyelesaikan permasalahan ketidakpastian ini yaitu *Dempster-shafer* (D-S). Algoritme D-S mempunyai nilai *belief* yang berfungsi untuk menentukan pengaruh antar gejala yang didapatkan dari seorang pakar. Pada kasus ini pakar tidak dapat memberikan nilai *belief* karena harus sesuai dengan bukti dan saksi nyata. Maka dengan algoritme *Particle Swarm Optimization* (PSO) nilai *belief* akan dibangkitkan sekaligus melakukan optimasi untuk mendapatkan hasil yang maksimal. Sesuai dengan pengujian yang dilakukan dari data kasus penganiayaan didapatkan nilai *belief* maksimal berdasarkan pengujian parameter PSO. Hasil perhitungan akurasi sistem dengan menggunakan nilai *belief* yang telah dioptimasi dengan D-S pada 29 kasus penganiayaan menunjukkan akurasi sebesar 13,79%. Hasil akurasi ini tidak maksimal dikarenakan permasalahan yang kompleks dengan hasil keluaran(*output*) sistem lebih dari satu. Untuk penelitian selanjutnya, dapat digunakan metode *artificial neural network*(ANN) atau dengan algoritme *analytic hierarchy process* (AHP).

Kata Kunci: nilai belief, *Dempster-Shafer* (D-S), *Particle Swarm Optimization* (PSO), penganiayaan

Abstract

The crime against the body and life continues to increase every year, judges as decision makers against criminal defendants have a very important role in providing decisions. However, there are some things that the judge needs to consider in making decisions, so that the problem of uncertainty can be a judge's obstacle. The author applies a method that can solve the problem of this uncertainty is Dempster-shafer (D-S). D-S algorithm has belief value that serves to determine the influence between symptoms obtained from an expert. In this case the expert can not give the value of belief because must be in accordance with the evidence and real sanctions. So with Particle Swarm Optimization algorithm (PSO) belief value will be raised as well as doing optimization to get maximum results. In accordance with the test conducted from the case data of the penganiayaan obtained maximum belief value based on PSO parameter test. The result of system accuracy calculation by using belief value that has been optimized with D-S on 29 cases of abuse shows accuracy of 13.79%. The result of this accuracy is not maximal due to complex problems with the output (Output) of the system more than one. For further research, we can use artificial neural network (ANN) method or with algorithm analytic hierarchy process (AHP).

Keywords: belief value, *Dempster-Shafer* (D-S), *Particle Swarm Optimization* (PSO), persecution

1. PENDAHULUAN

Indonesia adalah salah satu negara yang berkembang yang sudah memiliki peraturan sendiri tentang hukum pidana yaitu Kitab Undang-Undang Hukum Pidana (KUHP) yang mengatur tentang hukum pidana Indonesia yang

di dalamnya berisi pasal-pasal. Salah satunya adalah pasal 351 KUHP yaitu tentang penganiayaan. Meskipun ketentuan mengenai perbuatan yang melawan hukum sudah diatur dalam hukum pidana tetapi pelanggaran tetap saja terjadi, hal tersebut didukung oleh data Badan Pusat Statistik Indonesia tingkat kekerasan terhadap tubuh dan nyawa mengalami

peningkatan setiap tahunnya (Statistik Kriminal, 2016). Seiring dengan perkembangan peradaban, tujuan utama dari hukum pidana adalah untuk menjamin masyarakat merasa aman dan nyaman (Susanto *et al*, 2010). Dalam memutuskan hukuman bagi seorang terdakwa orang yang berotoritas adalah hakim. Untuk itu dalam mewujudkan keamanan dan kesejahteraan masyarakat, setidaknya hakim harus pintar dalam memutuskan hukuman yang dapat membuat pelaku penganiayaan jera (Fikri, 2013).

Penganiayaan adalah tindakan yang dilakukan secara sengaja dengan tujuan tertentu, tindakan tersebut dapat ditunjukkan dengan terdapatnya luka-luka kepada orang lain sehingga mengakibatkan rasa sakit dan penderitaan diluar batas-batas kemanusiaan (Chazawi, 2000). Dalam penelitian Lingtogareng (2013) menyatakan bahwa bukan hal yang mudah dalam mengambil keputusan bahwa seorang terdakwa benar-benar bersalah secara hukum, selain itu seorang hakim harus mampu mengambil keputusan secara objektif artinya harus melihat dari banyak sisi yang perlu dipertimbangkan. Akan tetapi, hakim juga mempunyai hak untuk memutuskan fakta-fakta yang relevan terhadap kasus yang terjadi, sehingga seorang hakim mempunyai tanggung jawab yang besar terhadap terdakwa yang ditangani (Sidharta, 2015). Oleh sebab itu perlu dibuat suatu solusi untuk membantu hakim dalam memutuskan hukuman bagi terdakwa.

Kendala yang sering dirasakan oleh pihak hakim adalah banyaknya suatu pasal dalam KUHP yang dilanggar oleh tersangka tindak kejahatan. Hal ini membuat hakim harus memilih pasal-pasal yang tepat dan benar sesuai dengan KUHP (Pratiwi, 2005). Terdapat beberapa hal yang harus dipertimbangkan oleh seorang hakim sebelum memberikan putusan pelanggaran terhadap pasal dalam KUHP. Pertimbangan hakim pada saat memberikan keputusan adalah menetapkan terlebih dahulu fakta-fakta atau perbuatan yang dilakukan oleh terdakwa. Berdasarkan fakta tersebut hakim memberikan keputusan hukum kepada terdakwa (Nurjanah, 2016).

Berdasarkan permasalahan diatas maka penulis ingin mencoba membantu para hakim dalam mengambil keputusan penentuan pasal untuk kasus tindak pidana penganiayaan pada pasal-pasal yang terdapat dalam KUHP. Terdapat beberapa metode untuk menyelesaikan permasalahan ketidakpastian, salah satunya

adalah metode *Dempster-Shafer*. Metode *Dempster-Shafer* adalah metode yang dapat digunakan mendiagnosis keadaan yang tidak konsisten akibat adanya pengurangan ataupun penambahan fakta baru yang akan memberikan perubahan terhadap aturan yang sudah ada (Ahmazedah *et al*, 2001).

Metode *Dempster-Shafer* mempunyai *frame of discernment* yang berfungsi sebagai skema pembicaraan dari sekumpulan hipotesis yang disebut dengan Densitas atau nilai *belief* (Hassiana, 2016). Nilai ini didapatkan dari pakar, pasal beserta gejala yang dilakukan terdakwa tidak dapat diprediksi pengaruhnya satu dengan yang lain karena harus sesuai dengan bukti nyata dan sanksi mata. Oleh sebab itu penulis ingin menerapkan sebuah metode untuk pembangkitan nilai *belief* pada penelitian ini dengan algoritme *Particle Swarm Optimization*.

Particle swarm optimization adalah salah satu metode *heuristics* yang berfokus pada pencarian lokal sehingga pada ruang pencarian yang kecil dapat menemukan solusi yang optimal (Anggodo *et al*, 2017a). Selain itu *Particle swarm optimization* juga dapat menyelesaikan berbagai permasalahan yang kompleks (Anggodo dan Mahmudy, 2017; Anggodo *et al*, 2017b). Berdasarkan penjabaran sebelumnya maka fokus penelitian ini adalah pembangkitan nilai *belief* pada *Dempster-Shafer* dengan *Particle Swarm Optimization* untuk penentuan pasal kasus penganiayaan.

2. DASAR TEORI

2.1 *Dempster-Shafer* (D-S)

Metode *Dempster-Shafer* dikenal juga sebagai teori fungsi keyakinan. Teori *Dempster-Shafer* ini diperkenalkan oleh A.P Dempster pada tahun 1968 beserta Glen Shafer pada tahun 1976 pada saat mereka sedang melakukan percobaan untuk mengadaptasikan teori probabilitas ke dalam sistem pakar (Dempster, A. P, 1990).

Belief adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 (nol) maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian. Berikut ini dapat diformulasikan fungsi *belief*:

$$Bel(X) = \sum_{Y \subseteq X} m(Y), \quad (1)$$

sedangkan *Plausibility* (Pls) dapat

diformulasikan sebagai (Wu *et al*, 2002):

$$Pls(X) = 1 - Bel(X') = 1 - \sum_{Y \subseteq X'} m(X'), \quad (2)$$

dimana:

- Bel(X) = *Belief* (X)
- Pls(X) = *Plausibility* (X)
- m(X) = *mass function* dari (X)
- m(Y) = *mass function* dari (Y)

Plausibility juga bernilai 0 sampai 1, jika yakin akan X' maka dapat dikatakan *Belief* (X') = 1 sehingga dari rumus di atas nilai Pls (X) = 0. Pada teori *Dempster-Shafer* juga dikenal adanya *frame of discernment* yang dinotasikan dengan Θ . *FOD* ini merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment* (O'neill, 2000), dimana:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}, \quad (3)$$

dimana:

- Θ = FOD atau *environment*
- $\theta_1 \dots \theta_n$ = elemen/unsur bagian dalam *environment*

Environment mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori *Dempster-Shafer* disebut dengan *power set* dan dinotasikan dengan $P(\Theta)$, setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1.

$$m = P(\Theta) \rightarrow [0,1], \quad (4)$$

sehingga dapat dirumuskan:

$$\sum_{X \in P(\Theta)} m(X) = 1 \approx \sum_{X \in P(\theta)} m(X) = 1, \quad (5)$$

dengan $P(\Theta)$ = *power set* dan $m(X)$ = *mass function* dari (X),

sebagai contoh:

$$P(\text{hostile}) = 0,7$$

$$P(\text{non-hostile}) = 1 - 0,7 = 0,3$$

Pada contoh di atas *belief* dari *hostile* adalah 0,7 sedangkan *disbelieffhostile* adalah 0,3. dalam teori *Dempster-Shafer*, *disbelief* dalam *environment* biasanya dinotasikan $m(\theta)$.

Sedangkan *mass function* (m) dalam teori *Dempster-Shafer* adalah tingkat kepercayaan dari suatu *evidence* (gejala), sering disebut dengan *evidencemeasure* sehingga dinotasikan dengan (m).

Pada aplikasi sistem pakar dalam satu

penyakit terdapat sejumlah *evidence* yang akan digunakan pada faktor ketidakpastian dalam pengambilan keputusan untuk diagnosis suatu penyakit. Untuk mengatasi sejumlah *evidence* tersebut pada teori *Dempster-Shafer* menggunakan aturan yang lebih dikenal dengan *Dempster's Rule of Combination*.

$$m1 \oplus m2(Z) = \sum_{X \cap Y = Z} m1(X)m2(Y), \quad (6)$$

dimana:

- $m1 \oplus m2(Z)$ = *mass function* dari *evidence* (Z)
- $m1(X)$ = *mass function* dari *evidence* (X)
- $m2(Y)$ = *mass function* dari *evidence* (Y)
- \oplus = operator *direct sum*

Secara umum formulasi untuk *Dempster's Rule of Combination* adalah:

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1 - k}, \quad (7)$$

dimana: k = Jumlah *evidential conflict*.

Besarnya jumlah *evidential conflict* (k) dirumuskan dengan:

$$k = \sum_{X \cap Y = \theta} m1(X)m2(Y), \quad (8)$$

sehingga bila persamaan (7) disubstitusikan ke persamaan (8) akan menjadi:

$$m1 \oplus m2(Z) = \frac{\sum_{X \cap Y = Z} m1(X)m2(Y)}{1 - \sum_{X \cap Y = \theta} m1(X)m2(Y)}, \quad (9)$$

dimana:

- $m1 \oplus m2(Z)$ = *mass function* dari *evidence* (Z)
- $m1(X)$ = *mass function* dari *evidence*(X)
- $m2(Y)$ = *mass function* dari *evidence* (Y)
- k= jumlah *evidential conflict*

2.2 Particle Swarm Optimization (PSO)

Algoritme *Particle Swarm Optimization* adalah sebuah algoritme dengan teknik komputasi evolusi yang termotivasi melalui sekumpulan perilaku sosial. Algoritme ini pertama kali diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995 (Kennedy dan Eberhart, 1995). Algoritme *Particle Swarm Optimization* akan memodelkan aktifitas solusi terbaik pada ruang pencarian, posisi partikel

yang terdapat pada ruang solusi adalah variabel-variabel optimasi yang digunakan sebagai kandidat optimasi. Setiap posisi tersebut akan dihubungkan dengan nilai objektif atau disebut sebagai nilai *fitness* (Marini dan Walzcak, 2015). *Particle Swarm Optimization* berbeda dengan algoritme optimasi lainnya karena tidak menggunakan gradien informasi dalam pencarian solusi sehingga tidak berakibat kesalahan fungsi persyaratan terus menerus (Nouaouria, 2013).

Penelitian Sedighizadeh *et al* (2009) terdapat beberapa istilah yang sering digunakan pada algoritme *Particle Swarm Optimization* antara lain:

1. *Swarm* : populasi yang terdapat pada suatu algoritme
2. Partikel : bagian(anggota) dari swarm, setiap partikel akan merepresentasikan solusi untuk permasalahan yang akan diselesaikan.
3. *Pbest (Personal Best)* : posisi terbaik yang pernah dicapai untuk mendapatkan solusi yang terbaik.
4. *Gbest(Global Best)* : posisi terbaik partikel secara keseluruhan
5. *Velocity*(vektor) : vektor yang menggerakkan proses optimasi yang menentukan arah suatu partikel diperlukan untuk berpindah yang berfungsi untuk memperbaiki posisinya semula.
6. *Inertia weight* : bobot *inertia*, digunakan untuk pengontrol dampak dari adanya *velocity* yang diberikan oleh suatu partikel.
7. *Koefisien akselerasi* : koefisien akselerasi akan mempengaruhi jarak maksimum yang dapat diambil oleh sebuah partikel dalam sebuah iterasi.

Terdapat beberapa tahapan-tahapan dalam pengimplementasian algoritme PSO, berikut ini adalah tahapan-tahapan dalam PSO:

1. Melakukan inialisasi populasi dari partikel dengan posisi dan *velocity* secara *random* dalam suatu ruang dimensi pencarian.
2. Mengevaluasi fungsi *fitness* optimasi yang diinginkan di dalam variabel pada setiap partikel.
3. Membandingkan evaluasi *fitness* partikel dengan *Pbest*nya. Jika nilai yang ada lebih baik dibandingkan dengan nilai *Pbest*, maka nilai tersebut akan diset sebagai *Pbest*.

4. Melakukan *Update Velocity* dan posisi untuk masing-masing partikel.
5. Kembali ke langkah 2 sampai kriteria terpenuhi, biasanya berhenti pada nilai *fitness* yang cukup baik atau sampai pada jumlah maksimum iterasi (Kennedy dan Eberhart, 1995).

Setiap partikel akan mempertahankan posisinya, yang terdiri dari *fitness* yang telah dievaluasi. Selain itu, setiap partikel dapat mengingat nilai *fitness* terbaik yang pernah dicapai selama pengoperasian algoritme berlangsung, disebut sebagai *fitness* partikel terbaik dan kandidat solusi yang dicapai oleh *fitness* ini disebut sebagai posisi terbaik partikel (*Pbest*). Algoritme PSO juga akan mempertahankan nilai *fitness* terbaik secara keseluruhan yang disebut sebagai (*Gbest*).

Untuk melakukan perubahan kecepatan pada algoritme PSO direpresentasikan pada persamaan 10.

$$V_i^{k+1} = W V_i^k + C_1 \text{rand1} \times (Pbest - X_i^k) + C_2 \text{rand2} (Gbest - X_i^k), \quad (10)$$

keterangan:

- V_i^k : kecepatan agen I pada iterasi k
- W : fungsi pemberat (Inertia)
- C_i : faktor pemberat
- Rand : nilai acak antara 0 dan 1
- X_i^k : posisi terakhir agen I pada iterasi K
- Pbest* : best dari agen i
- Gbest* : nilai *Pbest* terbaik dari kawan

Bobot inertia diperkenalkan oleh Shi dan Eberhart, yang digunakan untuk menyeimbangkan eksplorasi lokal yang berfungsi untuk mengurangi bobot selama iterasi dan memungkinkan algoritme untuk mengeksplorasi beberapa daerah spesifik (Chen, 2011).

Pada setiap iterasi, nilai fungsi inersia di *update* melalui persamaan 11.

$$W = W_{max} - \frac{w_{max} - w_{min}}{\text{iter max}} \times \text{iter}, \quad (11)$$

dimana:

- W_{max} : nilai inertia awal
- W_{min} : nilai inertia akhir
- itermax : jumlah iterasi maksimum
- iter : jumlah iterasi terakhir

Seiring dengan berubahnya kecepatan, maka terjadi perubahan pula pada posisi agen setiap iterasi yang dapat di hitung dengan persamaan 12 berikut ini (Marini dan Walczack, 2015).

$$X_i^{k+1} = X_i^k + V_i^{k+1}, \quad (12)$$

Keterangan :

X_i^{k+1} = Posisi agen terakhir

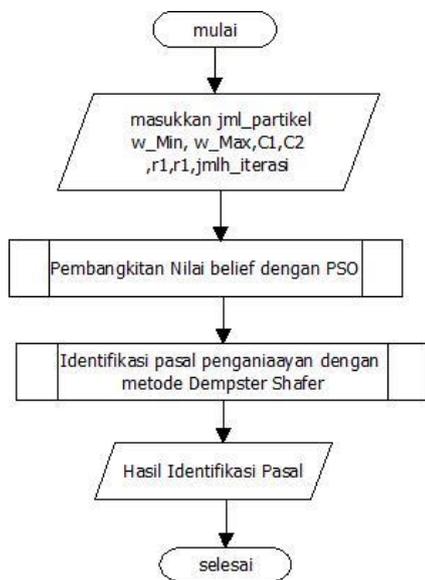
X_i^k = Posisi agen sebelumnya

V_i^{k+1} = Kecepatan agen terkini

3. PERANCANGAN SISTEM

3.1 Alur Perancangan Sistem

Pada tahap alir perancangan sistem, terdapat penjelasan mengenai bagaimana proses yang terjadi didalam pembangkitan nilai *belief* D-S dengan PSO, yang ditunjukkan oleh Gambar 2 berikut ini.

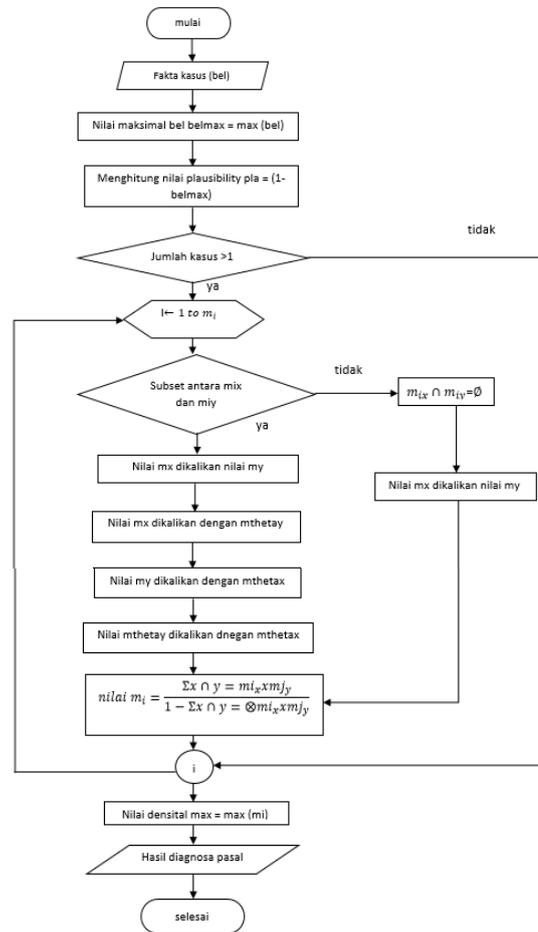


Gambar 1. Diagram Alir Perancangan Sistem

3.2 Alur Perancangan Sistem Perhitungan Algoritme Dempster-Shafer (D-S)

Proses *Dempster-Shafer* ini berfungsi untuk mengidentifikasi pasal terkait penganiayaan. Perhitungan dengan D-S akan dilakukan terlebih dahulu untuk mendapatkan nilai *fitness* yang digunakan untuk mengukur akurasi nilai *belief* yaitu dengan membandingkan hasil perhitungan dengan metode D-S dengan hasil diagnosis pakar. Perhitungan dalam D-S dimulai dengan memasukkan nilai *belief* tiap kasus penganiayaan ke dalam basis data sebagai dasar perhitungan. Kemudian dari data kasus yang telah dimasukkan oleh pengguna maka diperoleh kemungkinan pasal dan nilai densitasnya untuk menghitung nilai *belief* dan *plausibility*. Jika hanya satu gejala yang dimasukkan oleh pengguna maka pasal yang akan dikenakan

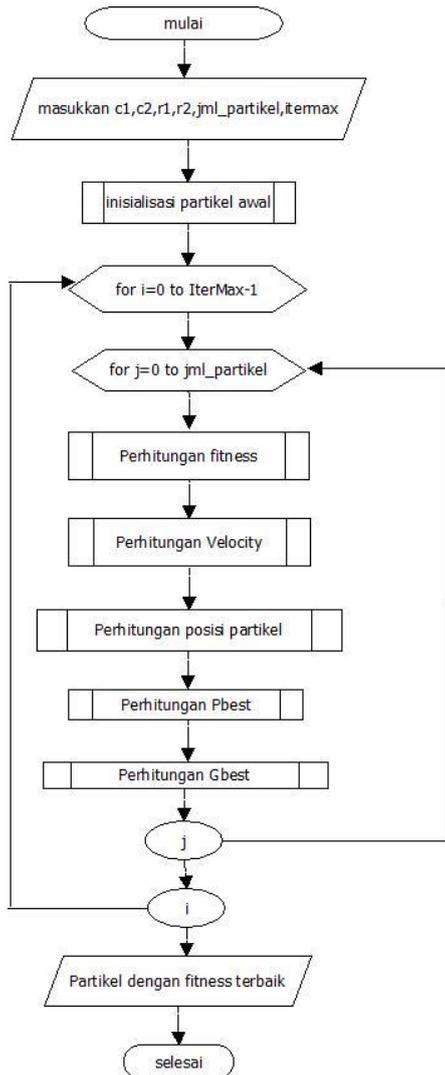
dilihat dari nilai *belief* yang paling tertinggi. Diagram alir proses perhitungan metode D_S dapat dilihat pada Gambar 3.



Gambar 2. Diagram Alir D-S

3.3 Alur Perancangan Sistem Perhitungan Algoritme Particle Swarm Optimization

Proses perhitungan menggunakan algoritme PSO untuk membangkitkan nilai *belief* pada D-S. Pada Gambar 4 dibawah ini ditunjukkan bagaimana alir algoritme PSO. Algoritme PSO mempunyai beberapa proses, diantaranya adalah inialisasi partikel awal, nilai *Pbest*, nilai *Gbest*, kecepatan awal, kemudian melakukan perhitungan *fitness*, *update* kecepatan, *update* posisi dan mendapatkan *Pbset* dan *Gbest* yang baru dan yang terakhir akan mendapatkan partikel dengan *fitness* terbaik. Proses-proses tersebut akan ditunjukkan pada Gambar 4 berikut ini:

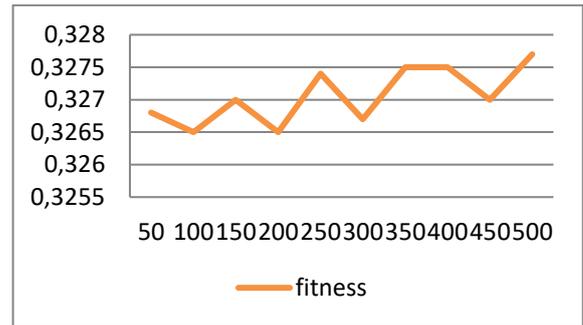


Gambar 3. Diagram Alir PSO

4. PENGUJIAN DAN ANALISIS

4.1 Pengujian Dan Analisis Jumlah Partikel

Pengujian ini berfungsi untuk membandingkan nilai *fitness* yang dihasilkan berdasarkan parameter yang diubah. Proses pengujian Nilai partikel dimulai dengan nilai partikel 50 sampai 500, dengan kelipatan 50. Skenario uji coba dilakukan dengan jumlah iterasi 50, dengan nilai bobot batas bawah (W_{min}) dan bobot batas atas (W_{max}) 0,4 dan 0,9. Nilai koefisien akselerasi ($C1,C2$) 1. Pengujian dilakukan sebanyak 5 kali untuk setiap partikel.

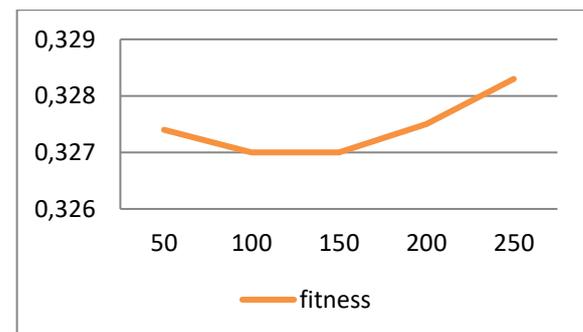


Gambar 4. Pengujian jumlah partikel terhadap nilai *fitness*

Berdasarkan Gambar 5 menunjukkan bahwa jumlah partikel yang semakin besar dapat menentukan nilai *fitness* akan semakin baik juga, tapi tidak menutup kemungkinan untuk jumlah partikel yang kecil, karena PSO adalah algoritme yang *stochastic* atau bersifat acak. Dari hasil pengujian nilai *fitness* terbaik berada pada partikel 500.

4.2 Pengujian Dan Analisis Jumlah Iterasi

Proses pengujian jumlah iterasi dimulai dengan nilai iterasi 50 sampai 500, dengan kelipatan 50. Skenario uji cobadilakukan dengan jumlah partikel yang paling baik, yaitu 500, dengan nilai bobot batas bawah (W_{min}) dan bobot batas atas (W_{max}) 0,4 dan 0,9. Nilai koefisien akselerasi ($C1,C2$) 1. Berikut ini adalah hasil pengujian jumlah iterasi:



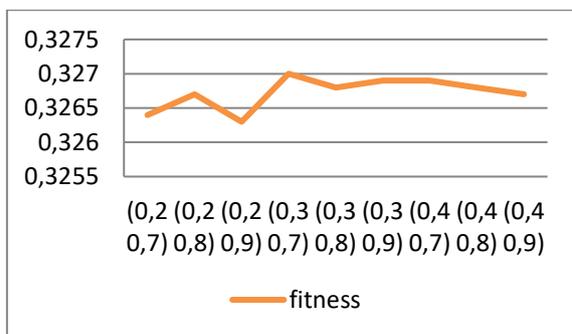
Gambar 5. Pengujian jumlah iterasi terhadap nilai *fitness*

Berdasarkan Gambar 6 jumlah iterasi yang semakin tinggi menghasilkan nilai *fitness* yang tinggi. Namun membutuhkan komputasi yang cukup lama. Dari Gambar 6 jumlah iterasi yang paling optimal terdapat pada iterasi 250 dengan nilai *fitness* 0,3282.

4.3 Pengujian W_{min} dan W_{max}

Pengujian ini berfungsi untuk mendapatkan nilai W_{min} dan W_{max} yang

paling optimal yang akan mempengaruhi perhitungan nilai *fitness*. Proses pengujian ini diujikan terhadap W_{min} 0,2, 0,3, 0,4 dan W_{max} adalah 0,7, 0,8, 0,9. Skenario uji coba dilakukan dengan jumlah partikel yang paling baik, yaitu 500, dengan iterasi 250, dan nilai koefisien akselerasi (C1,C2) adalah 1. Berikut ini adalah hasil pengujian W_{min} dan W_{max} .

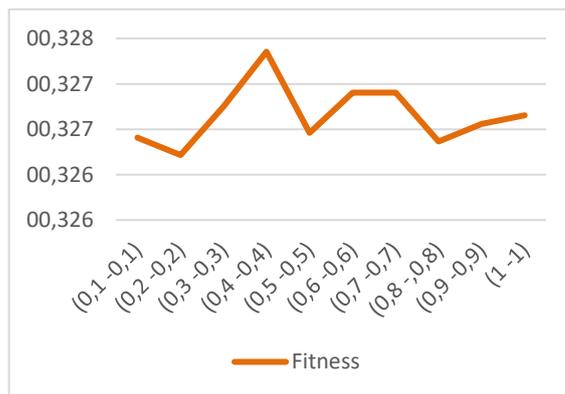


Gambar 6. Pengujian W_{min} dan W_{max}

Dari hasil grafik, Nilai *fitness* terendah terdapat pada W_{min} dan W_{max} 0,2 dan 0,9. Tetapi nilai *fitness* yang paling baik terdapat pada W_{min} dan W_{max} 0,3 dan 0,7, hal ini dapat membuktikan bahwa semakin besar nilai W_{min} dan W_{max} belum menentukan semakin baik nilai *fitness*-nya karena PSO merupakan metode *stochastic* atau bersifat acak.

4.4 Pengujian V_{min} dan V_{max}

Pengujian ini berfungsi untuk mendapatkan nilai V_{min} dan V_{max} yang paling optimal yang akan mempengaruhi perhitungan nilai *fitness*. Proses pengujian ini diujikan terhadap V_{max} adalah 0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8, 0,9, 1 dan V_{min} adalah -0,1, -0,2, -0,3, -0,4, -0,5, -0,6, -0,7, -0,8, -0,9, -1. Skenario uji coba dilakukan dengan jumlah partikel yang paling baik, yaitu 500, dengan iterasi 250 dan nilai W_{min} 0,3 dan W_{max} 0,7. Berikut ini adalah hasil pengujian terhadap V_{min} dan V_{max} .



Gambar 7. Pengujian V_{min} dan V_{max}

Berdasarkan Grafik pada Gambar 8, dapat dilihat bahwa nilai V_{min} dan V_{max} memiliki pengaruh terhadap nilai *fitness* pada algoritme. Dari hasil grafik, Nilai *fitness* terendah terdapat pada V_{min} dan V_{max} 0,2 dan -0,2. Tetapi nilai *fitness* yang paling baik terdapat pada V_{min} dan V_{max} 0,4 dan -0,4. Sedangkan nilai V_{min} dan V_{max} yang semakin besar belum menentukan nilai *Fitness* akan semakin baik juga, hal ini bias terjadi karena algoritma PSO bersifat *stochastic*.

4.5 Pengujian Akurasi

Pengujian hasil akurasi pembangkitan nilai *belief* dilakukan untuk mengetahui hasil yang diberikan oleh pakar dan hasil sistem untuk penentuan pasal kasus penganiayaan. Data yang digunakan dalam pengujian ini sebanyak 29 data kasus. Berikut ini adalah hasil pengujian akurasi.

Tabel 1. Pengujian Akurasi

Kasus	Diagnosis Pakar	Diagnosis Sistem	Nilai Belief dari Hasil Optimasi
1	352 A2, 351 A5	352 A2, 351 A5	0.9993
2	355 A1, 356	355 A1, 356	0.5832
3	355 A2, 356	355 A2, 356	0.5193
4	355 A1	351 A1,356	0.5832
5	355 A2	355 A2, 356	0.5193
6	353 A1, 356	353 A1, 356	0.6574
7	353 A2, 356	353 A1, 355 A1, 356	0.7590
8	353 A3, 356	353 A3, 355 A2,356	0.6759
9	353 A1	351 A3, 356	0.6574
10	353 A2	353 A2, 355 A1, 356	0.7590
11	353 A3	353 A3, 355 A2, 356	0.6759
12	354 A1, 356	354 A1,355 A1, 356	0.6188
13	354 A2, 356	354 A2, 355 A2, 356	0.5510
14	354 A1	354 A1, 355 A1, 356	0.6188
15	354 A2	354 A2, 355 A1, 356	0.5510
16	351 A1, 356	351 A1, 353 A1, 356	0.6975

17	351 A2, 356	351 A2,353 A2, 354 A1, 355 A1, 356	0.7825
18	351 A3, 356	351 A3, 343 A3, 354 A2, 355 A2, 356	0.6968
19	351 A4, 356	351 A1,351 A2,351 A3,351 A4, 352 A1, 353 A1,353 A2,353 A3,354 A1,354 A2,355 A1,355 A2, 356	0.6401
20	351 A5, 356	352 A1,356	0.7221
21	351 A1	351 A1, 353 A1, 356	0.6975
22	351 A2	351 A2, 353 A2, 354 A1, 355 A1, 356	0.7825
23	351 A3	351 A3, 353 A3, 354 A2, 355 A1, 356	0.6968
24	351 A4	351 A1,351 A2,351 A3,351 A4, 352 A1, 353 A1,353 A2,353 A3,354 A1,354 A2,355 A1,355 A2, 356	0.6401
25	351 A5	352 A1, 356	0.7221
26	358-1, 356	351 A2,353 A2, 354 A1, 355 A1, 356, 358-1	0.8054
27	358-2, 356	351 A3,353 A3,354 A2,355 A2, 356, 358 -2	0.7172
28	358-1	351 A2, 353 A2, 354 A1, 355 A1, 356, 358-1	0.8054
29	358-2	351 A3,353 A3,354 A2,355 A2, 356, 358 -2	0.7172

5. KESIMPULAN

Penelitian ini adalah pembangkitan nilai *belief* pada *Dempster-Shafer* dengan algoritme *Particle Swarm Optimization* untuk penentuan pasal pada kasus penganiayaan, pada penelitian ini PSO dapat membangkitkan nilai *belief* dengan hasil yang maksimal. Nilai tersebut akan digunakan pada D-S untuk mendeteksi pasal kasus penganiayaan. Akurasi yang didapatkan dalam pembangkitan nilai *belief* pada *Dempster-Shafer* dengan *Particle Swarm Optimization* untuk penentuan pasal kasus penganiayaan ini sebesar 13,79%. Penerapan Algoritme *Dempster-Shafer* tidak memperoleh hasil yang maksimal, hal ini terjadi karena *output* dari sistem ini lebih dari satu pasal sehingga D-S akan menghitung setiap nilai *belief* yang saling berkaitan. Selain itu, berdasarkan beberapa penelitian D-S cenderung digunakan untuk

penelitian yang menghasilkan satu *output*.

Berdasarkan penelitian pembangkitan nilai *belief* pada metode *Dempster-Shafer* dengan algoritme *Particle Swarm Optimization* untuk penentuan pasal pada kasus penganiayaan masih memiliki kekurangan. Oleh sebab itu penulis memberikan saran dengan menambahkan metode *artificial neural network* (ANN) untuk mendapatkan hasil akurasi yang lebih baik, dapat menggunakan metode *Bacpropagation*. Kemudian menambahkan data kasus untuk pengujian dan analisis, menambahkan Rule pada algoritme *Dempster-Shafer* atau menggunakan metode optimasi lainnya seperti Algoritme Genetika dan *Ant colony Algorithm* dan mengimplementasikan algoritme *Analytical Hierarchy Process* (AHP) untuk menghasilkan hasil yang lebih baik.

DAFTAR PUSTAKA

- Ahmadzadeh, M.R. dan Petrou, Maria. 2001. *Knowledge Fusion Based on D-S Theory. Its Application on Expert System for Software Fault Diagnosis*. IEEE.
- Anggodo, Y. P., Cahyaningrum, Winda., Fauziyah A. N., Khoiriyah I. L., Kartikasari, Oktavianis. 2017a. *Hybrid K-means dan Particle Swarm Optimization Untuk Clustering Nasabah Kredit*. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), vol. 4, no.01, hlm.8-18.
- Anggodo, Y. P., Ariyani A. K., Ardi M. K., dan Mahmudy W.F., 2017b. *Optimization Of Multi-Trip Vehicle Routing Problem With Time Windows Using Genetic Algorithm*. Journal of Environmental Engineering & Sustainable Technology (JEEST), vol. 3, no. 02, pp 92-97
- Anggodo, Y. P. Dan Mahmudy, W. F. 2017. *Automatic Clustering and Optimized Fuzzy Logical Relationships For Minimum Living Needs Forecasting*. Journal of Environmental Engineering & Sustainable Technology (JEEST), vol. 4, no. 01, pp. 1-7.
- Ardian, A. D. 2016. Sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan dengan menggunakan metode *Forward Chaining* dan *Iterative Dichotomizer 3* (ID3). Malang: Universitas Brawijaya.
- Chazawi, Adami. 2000. *Kejahatan Terhadap*

- Tubuh Dan Nyawa*. Malang: Rajawali Pers.
- Chen H. L., Yang, Bo., Wang, Gang., Liu, Jie., Xu, Xin., Wang S. J., Liu D. Y., 2011. *A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest. Knowledge-Based System*, vol. 24, pp.1349-1359.
- Dempster, A. P. 1968 *.A generalization of Bayesian inference*. *Journal of the Royal Statistical Society*, vol. 30 pp.205-247.
- Fikri. 2013. Analisis Yurdis Terhadap Delik Penganiayaan Berencana. *Jurnal Ilmu Hukum Legal Opinion*.vol. 1
- Hasianna, Noviani. 2016. *Penerapan Metode Dempster-Shafer Untuk Deteksi Penyakit Infeksi Saluran Pernafasan Dengan Pembangkitan Nilai Bilief Menggunakan Algoritma Genetika*. Malang: Universitas Brawijaya.
- Kennedy dan Eberhart. 1995. A New Optimizer Using Particle Swarm Theory. *Sixth IEEE International Symposium on Micro Machine and Human Science*.
- Lingtogareng, Jerol. 2013. *Analisa Keyakinan Hakim Dalam Pengambilan Keputusan Perkara Pidana di Pengadilan*. *Lex Crimen*. vol. 2, no. 3.
- Marini dan Walzcak, 2015. Particle Swarm Optimization (PSO) Atutorial. *Chemometrics and Intelligent Laboratory Systems*. vol. 149, pp. 153–165.
- Nouaouria, N., Boukadoum, M. Dan Proulx, R. 2013. *Particle Swarm Clasification:Asurvey and Positioning*. *Pattern Recognition*. Vol. 46. pp.2028-2044.
- Nurjanah, Siti. 2016. Pidana dan Pemidanaan Dalam Perundang-Undangan Di Indonesia.
- O'Neill, Adrian.. 2000. *Dempster-Shafer Teory*. <<http://www.Aonaware.com/binaries/dempster.pdf>> [Diakses tanggal : 5 Maret 2017]
- Pratiwi, S. W. 2016. Analisis Penjatuhan Putusan Lepas Dari Segala Tuntutan Hukum Oleh Hakim Pengadilan Negeri Surabaya No. 3175/Pid.B/2010/Pn.Sby Terhadap Kasus Anak Yang Melakukan Pembunuhan Terhadap Anak
- Kandungnya. Medan: Universitas Sumatera Utara
- Sedighzadeh, Davoud dan Masehian, Ellips. 2009. *Particle Swarm Optimization Methods, Taxonomy and Applications*. *International Journal of Computer Theory and Engineering*. vol. 1, no. 05, pp.486-502.
- Sidharta, Arief. 2015. Etika Dan Kode Etik Profesi Hukum. Kalimantan: Universitas Palangka Raya. Tersedia di <<http://journal.unpar.ac.id>>
- Statistik Kriminal, 2016. Badan Pusat Statistik.<<https://www.bps.go.id/linkTabelStatis/view/id/1570>> [Diakses Tanggal 6 Maret 2017]
- Susanto, Riki. 2010. Hukum Pidana (Criminal Law). Depok :Universitas Indonesia.