

Penggunaan Unsur *Soft Computing* Dalam Permainan “Terima Atau Tolak: Olimpiade SC” untuk Menentukan Strategi Terbaik

Eza Budi Perkasa
Institut Sains dan Bisnis Atma Luhur
Pangkalpinang, Indonesia
ezabudiperkasa@atmaluhur.ac.id

Devi Irawan
Institut Sains dan Bisnis Atma Luhur
Pangkalpinang, Indonesia
deviirawan@atmaluhur.ac.id

Rendy Rian Chrisna Putra
Institut Sains dan Bisnis Atma Luhur
Pangkalpinang, Indonesia
rendyriancp@atmaluhur.ac.id

Abstrak—Permainan “Terima Atau Tolak Olimpiade SC” adalah sebuah permainan yang diadaptasi dari permainan “Deal Or No Deal”. Permainan ini dimainkan oleh lima pemain yang mempunyai peran berbeda. Tiap-tiap pemain memiliki cara yang berbeda dalam bermain. Permainan ini juga merupakan permainan yang membutuhkan strategi bagi para pemainnya. Strategi ini diperlukan agar pemain tersebut memperoleh nilai yang sebesar-besarnya dibandingkan lawannya. Pada penelitian ini, akan diterapkan unsur *soft computing*. Unsur tersebut digunakan oleh komputer sebagai strategi dalam permainan ini. Tujuan yang hendak diperoleh adalah menentukan unsur *soft computing* yang dapat digunakan sebagai metode untuk menyusun strategi terbaik pada permainan tersebut. Pencarian strategi terbaik dilakukan dengan penghitungan peluang kemenangan dan tingkat kesuksesan. Peluang kemenangan digunakan untuk menggambarkan langkah yang diambil dan tingkat kesuksesan digunakan untuk menganalisis hasil akhir permainan. Tingkat kesuksesan dalam permainan dapat bersifat non-kumulatif apabila pemain mampu mempertahankan posisi pertama atau kumulatif apabila gagal. Nilai-nilai tersebut didapat dari beberapa percobaan. Hasil yang didapat adalah probabilistic reasoning memperoleh tingkat kesuksesan kumulatif tertinggi. Dari hasil tersebut, dapat disimpulkan bahwa probabilistic reasoning merupakan metode yang digunakan untuk menyusun strategi bermain yang terbaik. Ke depannya, peneliti selanjutnya diharapkan untuk dapat memperluas atau mempersempit ruang lingkup permainan agar penentuan strategi terbaik dapat dilakukan dengan lebih teliti.

Kata kunci — kecerdasan buatan, peluang kemenangan, permainan, *soft computing*, strategi bermain

I. PENDAHULUAN

Di zaman yang serba canggih ini, komputer telah menjadi sebuah kebutuhan bagi manusia. Tidak seperti dahulu kala ketika komputer masih merupakan barang mewah sehingga sulit didapat. Sekarang, kita bisa menemukan komputer dimanapun: Di *supermarket*, di instansi-instansi, bahkan kita sendiri dapat memilikinya dengan sangat mudah. Hal ini dikarenakan seperangkat komputer mampu mempermudah pekerjaan manusia [1], baik pekerjaan yang ringan sekalipun, maupun pekerjaan berat yang belum tentu manusia dapat melakukannya. Bahkan, suatu saat komputer dapat memiliki tingkat kecerdasan yang lebih tinggi dari manusia yang merupakan pemiliknya [2]. Salah satu contohnya adalah komputer catur milik IBM yang bernama Deep Blue yang berhasil mengalahkan juara dunia bertahan dalam permainan catur [3].

Saat ini, permainan tidak hanya berguna sebagai media hiburan saja, namun juga sebagai media simulasi kehidupan sehari-hari. Contoh permainan yang dapat digunakan sebagai media simulasi tersebut adalah Stackelberg [4] yang merupakan permainan dalam bidang sosial. Selain itu, permainan juga dapat menerapkan unsur *soft computing* di dalamnya. *Soft computing* dapat dikatakan sebagai pencari solusi yang lebih murah dan dapat memungkinkan toleransi terhadap *input* yang diberikan pengguna dibandingkan dengan komputasi konvensional (*hard computing*) [3]. *Soft computing* dapat membuat pemain komputer (*non-human player*) lebih memiliki karakter layaknya manusia dibandingkan bila tanpa menggunakan *soft computing*. Hal tersebut dikarenakan *soft computing* terilhami dari kemampuan manusia untuk mengambil keputusan [5]. Adapun permainan yang menerapkan *soft computing* antara lain permainan Bridge dengan menggunakan jaringan syaraf tiruan [6] serta permainan balap mobil yang menggunakan arsitektur *fuzzy* berbasis aturan [7] dan jaringan syaraf tiruan [8]. Dalam hal ini, *soft computing* berperan sebagai penyusun strategi bermain sesuai aturan yang telah diberikan pada permainan bersangkutan.

Dari uraian sebelumnya, peneliti bermaksud untuk merancang sebuah permainan yang menerapkan *soft computing*. Dalam penelitian ini, peneliti bermaksud untuk mencari tahu metode *soft computing* yang digunakan untuk menyusun strategi bermain terbaik. Metode-metode tersebut merupakan unsur pokok *soft computing*: *Probabilistic reasoning*, logika *fuzzy*, jaringan syaraf tiruan, dan algoritma genetika. Keempat unsur tersebut sebenarnya bukanlah pesaing antara satu dengan yang lainnya [1]. Akan tetapi, penelitian ini akan membandingkan keempat unsur tersebut secara independen. Dari hasil perbandingan kinerja setiap unsur tersebut, dapat diketahui unsur *soft computing* yang terbaik dalam penyusunan strategi bermain.

II. TINJAUAN PUSTAKA

A. Tinjauan Penelitian Terdahulu

Salah satu penelitian yang menerapkan *soft computing* di dalam permainan dapat dilihat di [9]. Tujuan dari penelitian tersebut adalah mensimulasikan efek *noise* di dalam permainan. Metode yang digunakan adalah metode optimasi evolusioner yang diberi nama “Wilcoxon Tournament”, metode pelestarian individu menggunakan metode *Incremental Temporal Average* (ITA) dan *Wilcoxon-test based Partial Order* (WPO) yang dibandingkan kecepatannya, serta distribusi *noise*-nya menggunakan model Gaussian. Hasil yang didapat dari penelitian ini adalah metode ITA lebih cepat karena metode tersebut menghindari komparasi setiap individu yang membutuhkan waktu lebih banyak. Walaupun demikian, dalam keadaan yang berbeda dengan saat penelitian, bisa saja metode WPO lebih baik dibandingkan ITA.

Penelitian lainnya yang mempunyai topik yang serupa adalah [10]. Tujuan penelitian tersebut adalah memperkenalkan konsep permainan bimatriks dengan ketidakpastian. Metode penelitian yang digunakan adalah tiga kriteria keputusan dalam *framework* teori ketidakpastian yang meliputi kriteria nilai terharap, kriteria nilai optimistik, dan kriteria nilai tak pasti. Hasil dari penelitian ini adalah diusulkannya tiga strategi ekuilibrium ketidakpastian untuk menghadapi situasi pemberian keputusan yang berbeda.

Soft computing juga dapat diterapkan pada permainan kartu seperti diuraikan lebih lanjut di [6]. Penelitian tersebut bertujuan menerapkan jaringan syaraf dalam permainan *contract bridge*, khususnya dalam Permasalahan *Double Dummy Bridge*. Metode yang digunakan adalah arsitektur jaringan syaraf Elman dengan pembelajaran terawasi yang difokuskan pada algoritma Runut Balik Elastik. Hasil dari penelitian ini adalah jaringan syaraf Elman memiliki performa yang superior sehingga dapat membantu pemain pemula dan semi-profesional untuk memperbaiki kemampuan mereka dalam bermain *bridge*.

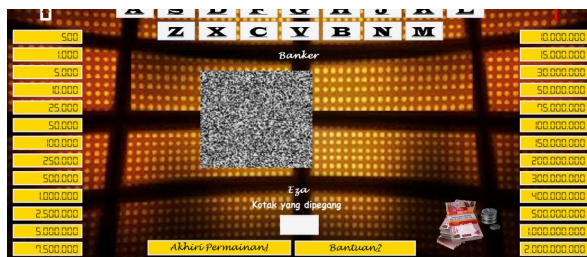
Referensi [7] mencoba untuk menerapkan unsur *soft computing* di dalam permainan balap mobil. Penelitian tersebut bertujuan mengusulkan sebuah arsitektur dalam permainan simulasi balap mobil menggunakan logika *fuzzy*. Metode yang digunakan adalah arsitektur *fuzzy* berbasis aturan yang dibagi menjadi enam modul: *Gear Control*, *Target Speed*, *Speed Control*, *Steering Control*, *Learning*, dan *Opponents Modifier*. Pengujian dilakukan pada ajang 2009 Simulated Car Racing Championship pada tiga kelompok trek: CEC2009, GECCO2009, dan CIG2009. Hasil yang diperoleh dari penelitian ini adalah arsitektur simulasi mobil balap yang diusulkan sebelumnya dapat memenangkan kejuaraan tersebut pada posisi akhir teratas.

Penelitian kali ini didasarkan dari penelitian oleh [11]. Tujuan dari penelitian tersebut adalah melihat kemampuan strategi AI dalam menyusun strategi bermain jika diterapkan unsur *soft computing* tersebut. Metode yang digunakan adalah *probabilistic reasoning* serta alat bantu yang disebut Tabel Peluang Kemenangan. Tabel Peluang Kemenangan digunakan untuk menghitung peluang kemenangan pada kondisi permainan yang diberikan. Hasil yang diperoleh pada penelitian ini adalah peluang kemenangan bernilai tidak eksak dan tergantung pada kondisi permainan yang terjadi saat itu. Selain itu, peluang kemenangan bukan merupakan tolak ukur sukses tidaknya AI dalam menyusun strategi.

Perbedaan penelitian terdahulu dengan penelitian kali ini terletak pada metode AI yang digunakan. Pada penelitian terdahulu, unsur *soft computing* yang diuji hanya berjumlah satu buah untuk setiap permainan. Pada penelitian saat ini, peneliti membandingkan empat unsur *soft computing* dalam satu permainan.

B. Tinjauan Objek Penelitian

“Terima Atau Tolak: Olimpiade SC” (TAT OSC) adalah sebuah permainan yang diadaptasi dari permainan “Deal Or No Deal” (DOND) yang pernah ditayangkan di stasiun televisi milik beberapa negara (termasuk Indonesia). Permainan ini merupakan *spin-off* dari permainan sebelumnya, yaitu “Terima Atau Tolak” (TAT). Seluruh permainan tersebut memiliki aturan yang serupa. TAT OSC dan DOND dimainkan oleh pemain-pemain yang masing-masing memiliki peran berbeda: Kontestan dan *banker*. Tujuan yang hendak dicapai dari permainan ini adalah mendapatkan uang sebesar-besarnya dengan cara apapun. Kontestan mendapatkan uang dengan cara memilih satu kotak uang, kemudian membuang kotak-kotak sisanya satu per satu. Setelah beberapa kotak terbuang, *banker* akan menawarkan kotak yang dipegang kontestan dengan harga tertentu. Penawaran *banker* ini tak akan pernah melebihi nilai tertinggi yang masih ada pada *money tree*. Pada sesi penawaran ini, kontestan dapat memilih untuk menerima penawaran jika dirasa nilai uang dalam kotak yang dipegangnya bernilai kecil atau menolak penawaran dan melanjutkan permainan.



Gambar 1 Antarmuka Permainan “Terima Atau Tolak”

Permainan TAT OSC mendapatkan namanya karena pemain manusia (*human player*) akan menghadapi empat lawan yang memiliki pola pikir yang berbeda-beda. Perbedaan ini disebabkan oleh unsur *soft computing* yang digunakan oleh setiap lawan. Keempat unsur *soft computing* (SC) yang diadu dalam permainan inilah yang menyebabkan kata “Olimpiade” tertulis sebagai subjudul dari permainan ini.

Secara keseluruhan, TAT OSC memiliki aturan yang serupa dengan DOND. TAT OSC dapat dimainkan oleh lima pemain: Satu kontestan dan empat *banker*. Keempat *banker* yang akan dihadapi kontestan adalah Banker (menggunakan *probabilistic reasoning*), Zhuang (menggunakan logika *fuzzy*), Bankaa (menggunakan jaringan syaraf tiruan), dan Banquero (menggunakan algoritma genetika). Kelima pemain tersebut memiliki tujuan yang sama, yaitu memperoleh uang sebesar-besarnya. Akan tetapi, cara pemain mencapai tujuan tersebut tergantung dengan perannya. Kontestan dapat memperoleh uang dengan memilih satu kotak uang. Kemudian, kontestan akan mencoba untuk mempertahankannya hingga akhir permainan. Sementara itu, setiap *banker* memperoleh tujuannya dengan cara menawarkan sejumlah uang kepada kontestan dengan nilai tertentu. Tidak seperti versi TAT

sebelumnya, pada TAT OSC, dimungkinkan bagi *banker* untuk tidak melakukan penawaran. Hal tersebut dikarenakan penghitungan nilai penawaran yang tidak sesuai dengan batasan yang ditetapkan TAT OSC.

III. METODE PENELITIAN

A. Rancangan Sistem Inferensi

Banker menggunakan *probabilistic reasoning*. Untuk merumuskan nilai penawarannya, Banker dapat memilih untuk menggunakan nilai penawaran terbaik saja selama permainan atau menggunakan nilai yang acak. Jika Banker memilih untuk menggunakan nilai penawaran terbaik saja, maka variabel n (penambah persentase penawaran) yang dipilih selalu bernilai 5. Sebaliknya, jika Banker memilih nilai acak, maka nilai variabel n akan diacak pada setiap ronde.

Unsur *soft computing* yang digunakan oleh Zhuang adalah logika *fuzzy*. Perumusan nilai penawaran dilakukan dengan pembentukan himpunan *fuzzy* (fuzzifikasi) pada Tabel 1.

Tabel 1 Himpunan *Fuzzy* Permainan “Terima Atau Tolak Olimpiade SC”

Variabel		Himpunan <i>Fuzzy</i>		Rentang
Nama	Simbol	Nama	Simbol	
Perbandingan <i>expected value</i>	e	Kecil	k	$[0, d_1]$
		Sedang	s	$[d_2, d_3]$
		Besar	b	$[d_4, \infty)$
Ronde	r	Rendah	rd	$[0, d_5]$
		Menengah	mg	$[d_6, d_7]$
		Tinggi	tg	$[d_8, 8]$
Penambah persentase penawaran	n	Kecil	kc	$[0, d_9]$
		Besar	bs	$[d_{10}, 5]$

Expected value (EV) merupakan rata-rata dari nilai uang yang belum terpilih atau dapat dikatakan:

$$EV = \frac{\sum m}{b} \quad .!$$

dengan $\sum m$ dan b berturut-turut merupakan jumlah nilai uang yang masih berada di *money tree* (nilai yang belum terpilih) dan jumlah kotak yang masih belum terbuka (termasuk kotak yang dipegang kontestan). Variabel e merupakan perbandingan EV ronde saat itu dan ronde sebelumnya dalam persen.

Variabel d_1 hingga d_{10} merupakan nilai yang ditentukan sebelum permainan berlangsung. Nilai-nilai variabel tersebut adalah sebagai berikut.

- d_1 = Bilangan acak dari 21 sampai 100
- $d_2 = d_1 - 20$
- $d_3 = d_2 + 70$
- $d_4 = d_3 - 20$
- d_5 = Bilangan acak dari 2 sampai 5
- $d_6 = d_5 - 1$
- $d_7 = d_6 + 3$
- $d_8 = d_7 - 1$
- d_9 = Bilangan acak dari 2 sampai 4
- $d_{10} = d_9 - 1$

Rentang nilai pada Tabel 1 digunakan untuk menentukan fungsi keanggotaan berikut.

1) Perbandingan EV

$$\mu_k(e) = \begin{cases} 1; e \leq d_1 - 10 \\ \frac{d_1 - e}{10}; d_1 - 10 \leq e \leq d_1 \\ 0; e \geq d_1 \end{cases} \quad .!$$

$$\mu_s(e) = \begin{cases} 0; e \leq d_2 \text{ atau } e \geq d_3 \\ \frac{e-d_2}{20}; d_2 \leq e \leq d_2+20 \\ 1; d_2+20 \leq e \leq d_3-20 \\ \frac{d_3-e}{20}; d_3-20 \leq e \leq d_3 \end{cases}$$

$$\mu_b(e) = \begin{cases} 0; e \leq d_4 \\ \frac{e-d_4}{10}; d_4 \leq e \leq d_4+10 \\ 1; e \geq d_4+10 \end{cases}$$

2) Ronde

$$\mu_{rd}(r) = \begin{cases} 1; r \leq d_5-2 \\ \frac{d_5-r}{2}; d_5-2 \leq r \leq d_5 \\ 0; r \geq d_5 \end{cases}$$

$$\mu_{mg}(r) = \begin{cases} 0; r \leq d_6 \text{ atau } r \geq d_7 \\ 1; r = d_6+1 \\ 0,5; r = d_7-1 \end{cases}$$

$$\mu_{tg}(r) = \begin{cases} 0; r \leq d_8 \\ \frac{r-d_8}{2}; d_8 \leq r \leq d_8+2 \\ 1; r \geq d_8+2 \end{cases}$$

3) Penambah persentase penawaran

$$\mu_{kc}(n) = \begin{cases} 1; n \leq d_9-2 \\ \frac{d_9-n}{2}; d_9-2 \leq n \leq d_9 \\ 0; n \geq d_9 \end{cases}$$

$$\mu_{bs}(n) = \begin{cases} 0; n \leq d_{10} \\ \frac{n-d_{10}}{2}; d_{10} \leq n \leq d_{10}+2 \\ 1; n \geq d_{10}+2 \end{cases}$$

Berdasarkan jumlah variabel *fuzzy*, dapat dibuat *rule* inferensi sebanyak sembilan buah. Kesembilan *rule* tersebut adalah sebagai berikut.

- 1) IF *e* Besar AND *r* Tinggi THEN *n* Kecil
- 2) IF *e* Besar AND *r* Menengah THEN *n* Kecil
- 3) IF *e* Besar AND *r* Rendah THEN *n* Kecil
- 4) IF *e* Sedang AND *r* Tinggi THEN *n* Kecil
- 5) IF *e* Sedang AND *r* Menengah THEN *n* Besar
- 6) IF *e* Sedang AND *r* Rendah THEN *n* Besar
- 7) IF *e* Kecil AND *r* Tinggi THEN *n* Besar
- 8) IF *e* Kecil AND *r* Menengah THEN *n* Besar
- 9) IF *e* Kecil AND *r* Rendah THEN *n* Besar

Adapun pencarian nilai n yang tegas (defuzzifikasi) dilakukan dengan metode penghitungan rata-rata terbobot. Penghitungan rata-rata terbobot dapat dirumuskan secara matematis sebagai berikut.

$$n = \frac{\sum_{i=1}^9 (\alpha_i \times z_i)}{\sum_{i=1}^9 \alpha_i} \quad .1$$

dengan α adalah nilai minimum dari fungsi keanggotaan e dan r , nilai z didapat dengan mensubstitusikan nilai fungsi keanggotaan n dengan α sesuai *rule* yang diberikan, dan i adalah nomor *rule* inferensi. Nilai n yang didapat dari persamaan tersebut dibulatkan ke bilangan bulat terdekat.

Dalam penentuan nilai penawaran, Bankaa menggunakan jaringan syaraf tiruan. Unsur *soft computing* yang digunakan tersebut merupakan satu-satunya unsur yang membutuhkan pembelajaran sebelum permainan berlangsung. Pembelajaran dilakukan dengan mengenalkan data sampel yang diberikan. Data sampel yang diberikan meliputi:

- 1) Ronde yang benar (1 hingga 8) dan penambah persentase penawaran yang benar (0 hingga 5) sebanyak tiga buah;
- 2) Ronde yang salah dan penambah persentase penawaran yang benar sebanyak satu buah;
- 3) Ronde yang benar dan penambah persentase penawaran yang salah sebanyak satu buah; dan
- 4) Ronde yang salah dan penambah persentase penawaran yang salah sebanyak satu buah.

Nilai parameter inferensi dengan jaringan syaraf ini ditentukan sebelum permainan berlangsung dan merupakan nilai acak berikut.

- 1) Bobot ronde awal: 0,000 hingga 1,000.
- 2) Bobot penambah persentase penawaran: 0,000 hingga 1,000.
- 3) Bias awal: 0,000 hingga 1,000.
- 4) *Learning rate*: 0,000 hingga 1,000.
- 5) Jumlah maksimum *epoch*: 1 hingga 100.

Jika w_r adalah bobot ronde, w_n adalah bobot penambah persentase penawaran, b adalah bias, dan α adalah *learning rate*, maka nilai *output* setiap data sampel, y_{in} , dapat dihitung dengan persamaan berikut

$$y_i = b + r \times w_r + n \times w_n \quad .11$$

Nilai respon *output*, y , dapat dihitung dengan fungsi aktivasi berikut.

$$y = \begin{cases} 1; & y_i \geq 1 \\ 0; & y_i < 1 \end{cases} \quad .12$$

Nilai respon tersebut digunakan untuk menentukan apakah nilai n yang digunakan termasuk nilai yang valid atau tidak valid. Target nilai respon tersebut adalah 1 (valid) untuk data sampel pertama hingga ketiga dan 0 (tidak valid) untuk data sampel lainnya. Nilai w_r , w_n , dan b selanjutnya diperbarui dengan persamaan berikut.

$$w_r(\text{baru}) = \begin{cases} w_r(\text{lama}); & y = \text{target} \\ w_r(\text{lama}) + \alpha \times \text{target} \times r; & y \neq \text{target} \end{cases} \quad .13$$

$$w_n(\text{baru}) = \begin{cases} w_n(\text{lama}); & y = \text{target} \\ w_n(\text{lama}) + \alpha \times \text{target} \times n; & y \neq \text{target} \end{cases} \quad .14$$

$$b(\text{baru}) = \begin{cases} b(\text{lama}); & y = \text{target} \\ b(\text{lama}) + \alpha \times \text{target}; & y \neq \text{target} \end{cases} \quad .15$$

Nilai-nilai parameter yang telah diperbarui ini akan digunakan untuk mengenalkan data selanjutnya. Apabila sudah mencapai data terakhir, maka *epoch* ditambah satu dan pembelajaran diulang ke data pertama dan seterusnya hingga *epoch* maksimum.

Penentuan nilai penawaran Bankaa dilakukan dengan memilih secara acak n yang memiliki nilai 0, 1, 2, 3, 4, atau 5 dan memiliki nilai y sama dengan 1 pada ronde berjalan. Jika tak ada satupun nilai n yang memenuhi syarat tersebut, maka Bankaa tak dapat melakukan penawaran pada ronde itu dan akan menyatakan “lewat” (pas).

Selama permainan, Banquero menerapkan algoritma genetika. Dalam hal ini, nilai n yang akan dipilih direpresentasikan dengan *string*. Panjang *string* yang digunakan adalah tiga bit sehingga terdapat delapan kemungkinan nilai n . Dari kedelapan kemungkinan tersebut, kombinasi yang dipakai adalah *string* 000 sampai 101 (0 sampai 5 dalam desimal). Representasi nilai n ini

disebut juga “kromosom”. Adapun nilai-nilai parameter yang digunakan adalah nilai acak yang ditentukan sebelum permainan berlangsung sebagai berikut.

- 1) Ukuran populasi: 2 hingga 20.
- 2) Peluang *crossover*: 0,01 hingga 1,00.
- 3) Peluang mutasi: 0,000 hingga 0,100.
- 4) Peluang pelestarian kromosom: 0,00 hingga 1,00.

Tahap-tahap pencarian nilai n menggunakan algoritma genetika ini adalah sebagai berikut.

- 1) Tentukan populasi awal secara acak dengan jumlah kromosom sama dengan ukuran populasi.
- 2) Seleksi kromosom yang akan digunakan menggunakan metode roda *roulette* dengan cara:
 - a) Hitung *fitness* setiap kromosom dengan menggunakan fungsi linear [$f(x) = x$].
 - b) Hitung *fitness* relatif setiap kromosom. Nilai *fitness* relatif sama dengan nilai *fitness* kromosom bersangkutan dibagi total nilai *fitness* seluruh kromosom.
 - c) Tentukan nilai *fitness* relatif kumulatif dari setiap kromosom.
 - d) Bangkitkan nilai acak dari 0,000 hingga 1,000 sebanyak ukuran populasi.
 - e) Kromosom baru hasil seleksi merupakan kromosom semula yang nilai acaknya lebih dari atau sama dengan *fitness* relatif kumulatif kromosom sebelumnya dan kurang dari *fitness* relatif kumulatif kromosom bersangkutan.
- 3) Lakukan *crossover* dengan cara:
 - a) Bangkitkan nilai acak dari 0,000 hingga 1,000 sebanyak ukuran populasi.
 - b) Pilih nilai acak yang kurang dari atau sama dengan peluang *crossover*. Jika jumlah kromosomnya ganjil, buang kromosom hasil pemilihan yang terakhir.
 - c) Tentukan titik tempat *crossover* dilakukan dengan membangkitkan bilangan acak dari 1 hingga 2. Tukar bit pada kromosom bersangkutan pada posisi setelah bilangan acak tersebut hingga bit terakhir dengan bit yang sesuai pada kromosom pasangan.
- 4) Lakukan mutasi dengan cara:
 - a) Bangkitkan bilangan acak dari 0,000 hingga 1,000 sebanyak ukuran populasi dikali tiga. Bilangan pengali ini didapatkan dari panjang kromosom yang digunakan (tiga bit).
 - b) Gantilah bit yang terkena mutasi (bilangan acaknya kurang dari peluang mutasi) dengan komplemennya.
- 5) Lestarkan kromosom dengan cara:
 - a) Bangkitkan bilangan acak dari 0,000 hingga 1,000 sebanyak ukuran populasi.
 - b) Tentukan kromosom yang akan diganti dengan memperhatikan nilai acak kromosom bersangkutan. Kromosom yang akan diganti adalah kromosom yang memiliki nilai acak kurang dari peluang pelestarian kromosom. Kromosom penggantinya adalah kromosom terbaik pada populasi awal.

Satu siklus manipulasi kromosom (satu generasi) ini menentukan nilai n untuk ronde 1. Untuk ronde-ronde selanjutnya, maka populasi awalnya adalah populasi hasil manipulasi. Nilai n yang dipilih adalah kromosom yang memiliki nilai *string* bit terbesar selain kromosom yang memiliki *string* bit 110 atau 111. Apabila populasi hasil manipulasi hanya memuat kromosom dengan *string* bit 110 dan/atau 111 saja, maka Banquero akan menyatakan “lewat” pada ronde berjalan.

B. Teknik Pengujian dan Analisis

Dalam permainan TAT OSC, nilai peluang kemenangan tiap-tiap pemain tidaklah eksak. Hal tersebut dipengaruhi oleh kondisi permainan saat itu. Untuk itu, peneliti menggunakan Tabel Peluang Kemenangan (TPK) [11] sebagai alat bantu penghitungan peluang kemenangan. Struktur TPK dapat dilihat pada Gambar 2.

	46.594.415	48.924.136	51.253.856	53.583.577	55.913.298	58.243.019
500	B	B	B	B	B	B
1.000	B	B	B	B	B	B
5.000	B	B	B	B	B	B
10.000	B	B	B	B	B	B
25.000	B	B	B	B	B	B
50.000	B	B	B	B	B	B
100.000	B	B	B	B	B	B
250.000	B	B	B	B	B	B
500.000	B	B	B	B	B	B
1.000.000	B	B	B	B	B	B
2.500.000	B	B	B	B	B	B
5.000.000	B	B	B	B	B	B
100.000.000	K	K	K	K	K	K
150.000.000	K	K	K	K	K	K
200.000.000	K	K	K	K	K	K
300.000.000	K	K	K	K	K	K
400.000.000	K	K	K	K	K	K
500.000.000	K	K	K	K	K	K
1.000.000.000	K	K	K	K	K	K
2.000.000.000	K	K	K	K	K	K

Gambar 2 Tabel Peluang Kemenangan (Sumber: [11])

Dari Gambar 2, dapat dijelaskan struktur TPK sebagai berikut.

- 1) Angka-angka yang tersusun per baris merupakan kemungkinan nilai uang dalam kotak yang dipegang kontestan. Jumlah kombinasinya sama dengan jumlah kotak uang yang tersisa dan nilainya sama dengan nilai uang yang masih berada di *money tree*.
- 2) Angka-angka yang tersusun per kolom merupakan kemungkinan nilai penawaran *banker* saat itu. Jumlah kombinasinya adalah enam dan nilainya dapat dihitung menggunakan persamaan

$$o_n = \frac{(10 \times r + n) \% \times \sum m}{b} \quad .1.$$

dengan o_n adalah nilai penawaran (dibulatkan ke bilangan bulat terdekat), r adalah ronde berjalan, n adalah angka yang dipilih (bernilai 0 sampai 5), $\sum m$ adalah jumlah nilai uang yang masih berada di *money tree* (nilai yang belum terpilih), dan b adalah jumlah kotak yang masih belum terbuka (termasuk kotak yang dipegang kontestan).

- 3) Huruf yang terletak di sel menandakan pemain yang dapat memenangkan permainan jika kombinasi nilai uang dan penawarannya sama dengan nilai yang ditunjukkan di baris dan kolom bersangkutan. Huruf B berarti pemenangnya adalah *banker*, huruf K berarti pemenangnya adalah kontestan, dan huruf S berarti seri. Pengisiannya mengikuti ketentuan berikut.
 - a) Jika nilai uang kurang dari nilai penawaran, maka berikan huruf K bila kontestan menerima penawaran atau B bila kontestan menolaknya.
 - b) Jika nilai uang lebih dari nilai penawaran, maka berikan huruf B bila kontestan menerima penawaran atau K bila kontestan menolaknya.
 - c) Huruf S hanya dapat diberikan apabila nilai uang sama dengan nilai penawaran.

Nilai peluang kemenangan masing-masing *banker*, w_B , dapat dihitung menggunakan rumus berikut.

$$w_B = \frac{\text{Jumlah huruf B pada TPK}}{6 \times b} \times 100 \% \quad .2.$$

Pengujian dilakukan dengan menghitung peluang kemenangan menggunakan TPK pada kondisi permainan yang berbeda-beda dan memperhatikan medali yang diperoleh. Pengujian dilakukan dengan menghitung peluang kemenangan menggunakan TPK dan memperhatikan medali yang diperoleh. Dalam penelitian kali ini, kondisi yang diuji tersebut adalah sebagai berikut.

- 1) Semua penawaran ditolak hingga akhir permainan.
- 2) Penawaran Banker pada ronde 8 diterima.
- 3) Penawaran Zhuang pada ronde 8 diterima.
- 4) Penawaran Bankaa pada ronde 8 diterima.
- 5) Penawaran Banquero pada ronde 8 diterima.
- 6) Penawaran Banker pada ronde 4 diterima.
- 7) Penawaran Zhuang pada ronde 4 diterima.
- 8) Penawaran Bankaa pada ronde 4 diterima.
- 9) Penawaran Banquero pada ronde 4 diterima.
- 10) Penawaran Banker pada ronde 1 diterima.
- 11) Penawaran Zhuang pada ronde 1 diterima.
- 12) Penawaran Bankaa pada ronde 1 diterima.
- 13) Penawaran Banquero pada ronde 1 diterima.

Pemilihan Banker, Zhuang, Bankaa, dan Banquero sebagai pemain percobaan dilakukan karena keempat *banker* tersebut adalah pemain AI dalam permainan TAT OSC. Ronde 1, 4, dan 8 dipilih karena ketiga ronde tersebut merupakan *milestone* dari permainan (ronde awal, pertengahan, dan akhir) sehingga dapat mewakili seluruh ronde. Pengujian dilakukan dalam dua siklus agar dapat dibandingkan perbedaan cara penyusunan strategi AI pada masing-masing siklus. Setiap siklus berisi tiga belas kondisi yang sama. Pengujian dalam siklus ini dilakukan karena terdapat pembangkit nilai acak dalam permainan.

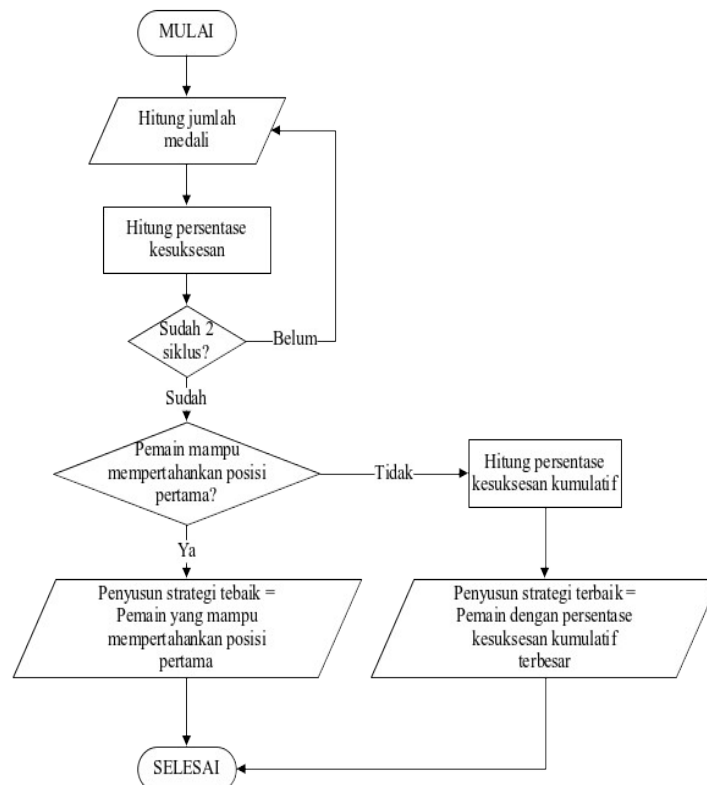
Pada penelitian kali ini, peluang kemenangan yang dihitung meliputi peluang kemenangan sebelum dan setelah penawaran. Peluang kemenangan sebelum penawaran dihitung menggunakan TPK yang utuh dengan asumsi nilai penawarannya valid dan akan ditolak kontestan seluruhnya. Peluang kemenangan setelah penawaran dihitung menggunakan TPK yang di-*pruning*, yaitu TPK yang hanya terdiri dari satu kolom yang *header*-nya menunjukkan nilai penawaran yang diajukan. Nilai peluang kemenangan setelah penawaran dapat berbeda antara satu *banker* dengan *banker* lainnya karena tergantung dari nilai dan status penawaran (diterima atau ditolak oleh kontestan) masing-masing.

Peneliti akan memperhatikan urutan total kesuksesan *banker* dalam dua siklus penelitian. Total kesuksesan *banker* dinyatakan dalam persentase. Apabila *banker* memperoleh P medali platinum, G medali emas, S medali perak, dan B medali perunggu dalam tiga belas percobaan yang diuraikan sebelumnya, maka persentase kesuksesan *banker* pada satu siklus tersebut adalah s . Nilai s dapat dirumuskan sebagai berikut.

$$s = \frac{P+G+S+B}{13} \times 100 \% \quad .3.$$

Pada kasus terjadi dua atau lebih *banker* memperoleh nilai s yang sama, penentuan urutan kesuksesan ditetapkan melalui perolehan medali secara individual, dimulai dari platinum. Jika jumlah medali platinumnya sama, maka perhatikan jumlah medali emasnya dan seterusnya. Jika *banker* bersangkutan dapat mempertahankan urutan pertama pada kedua siklus, maka secara otomatis ia dinyatakan sebagai penyusun strategi terbaik. Jika urutan tersebut gagal dipertahankan, maka *banker* dengan persentase kesuksesan kumulatif terbesar yang dinyatakan sebagai penyusun strategi terbaik. Persentase tersebut ditetapkan karena pemain yang mengumpulkan medali terbanyak dalam dua siklus percobaan dianggap lebih dominan (agresif) dalam merumuskan strategi. Persentase kesuksesan kumulatif pemain adalah s_k . Nilai s_k dirumuskan mirip dengan nilai s , hanya penyebut pecahannya diganti menjadi 26. *Tiebreaker* untuk nilai s_k sama dengan *tiebreaker* untuk nilai s (perolehan medali secara individual).

$$s_k = \frac{P+G+S+B}{26} \times 100\%$$



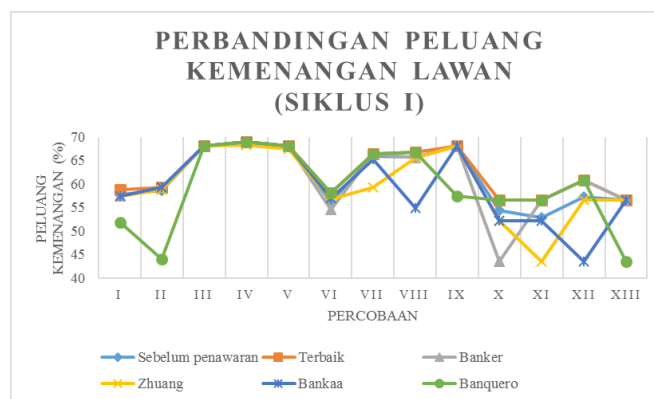
Gambar 3 Alur Penetapan Penyusun Strategi Terbaik

Peluang kemenangan bukanlah tolak ukur sukses atau tidaknya *banker* menyusun strategi. Peluang kemenangan hanya menyatakan seberapa besar “usaha” *banker* (pemain non-manusia/komputer) untuk mengalahkan kontestan (pemain manusia). Peneliti juga akan menganalisis keunggulan dan kelemahan masing-masing *banker* dalam menyusun strategi.

IV. HASIL DAN PEMBAHASAN

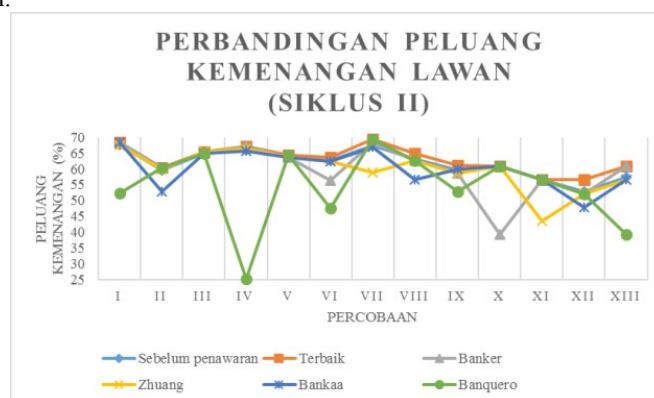
A. Peluang Kemenangan Banker

Dari Gambar 4, dapat dilihat bahwa Banquero memiliki usaha yang minimum untuk memenangkan permainan. Hal ini dikarenakan nilai peluang kemenangannya yang bernilai cukup jauh berbeda dibanding pesaingnya pada percobaan II dan XIII. Selain itu, nilai peluang kemenangan keempat *banker* mencapai nilai minimum pada percobaan X (Banker), XI (Zhuang), XII (Bankaa), dan XIII (Banquero). Hal tersebut terjadi karena kontestan langsung menerima penawaran *banker* bersangkutan pada ronde 1. Bagaimanapun, seluruh pernyataan tersebut bukanlah menjadi tolak ukur kesuksesan *banker* dalam penyusunan strategi.



Gambar 4 Grafik Perbandingan Peluang Kemenangan (Siklus I)

Dari Gambar 5, dapat dilihat bahwa peluang kemenangan dapat menjadi minimum tanpa harus menunggu penerimaan penawaran dari kontestan. Hal tersebut dapat terjadi dikarenakan pemilihan nilai parameter yang salah. Nilai minimum tersebut terjadi pada Banquero di percobaan IV. Pada percobaan tersebut, penawaran Banquero tidak diterima, namun peluang kemenangannya bernilai paling kecil, sangat jauh dibandingkan Bankaa yang penawarannya diterima. Ini terjadi karena terdapat banyak peluang kemenangan yang bernilai nol. Selain itu, tidak seperti siklus I, nilai peluang kemenangan *banker* yang penawarannya diterima pada percobaan X hingga XIII tidak sama. Hal tersebut terjadi dikarenakan kombinasi nilai uang pada *money tree* yang berbeda cukup signifikan dan juga nilai penambah persentase penawaran yang berbeda-beda. Harap diingat bahwa kombinasi nilai uang pada *money tree* merupakan kemungkinan nilai uang yang berada di kotak uang yang dipertahankan kontestan hingga akhir permainan.



Gambar 5 Grafik Perbandingan Peluang Kemenangan (Siklus II)

B. Perolehan Medali dan Persentase Kesuksesan

Dari Tabel 2, dapat dilihat bahwa Banker merupakan pemain AI dengan tingkat kesuksesan sempurna ($s = 100\%$). Banquero dan Zhuang memiliki tingkat kesuksesan yang sama, namun Banquero lebih unggul karena memperoleh medali emas lebih banyak. Bankaa menjadi juru kunci pada siklus I karena hanya memperoleh nilai s sekitar 61,5% saja.

Tabel 2 Perolehan Medali dan Persentase Kesuksesan (Siklus I)

Pemain	Pl	E	Pr	Pg	Total	% Sukses
Banker	7	2	2	2	13	100,0000000
Banquero	3	7	0	1	11	84,61538462
Zhuang	3	1	2	5	11	84,61538462
Bankaa	2	1	2	3	8	61,53846154

Dari Tabel 3, dapat dilihat bahwa unggulan pada siklus II (Banker) harus “puas” menempati posisi juru kunci dengan nilai s sekitar 69% saja. Banquero yang memiliki rekor buruk pun memiliki nilai s yang sama dengan Banker. Banquero menempati posisi yang lebih tinggi dikarenakan medali platinyanya yang lebih banyak. Sementara itu, Zhuang mengambil alih posisi Banker dengan persentase kesuksesan mencapai hampir 85%. Bankaa yang memiliki usaha yang biasa saja menempati posisi kedua walaupun memiliki nilai s yang sama dengan Zhuang dikarenakan kalah dalam perolehan medali platinum. Tidak seperti siklus I, pada siklus II ini tidak ada satu pun *banker* yang memperoleh kesuksesan sempurna ($s = 100\%$) karena setiap *banker* pernah gagal memperoleh medali sekurang-kurangnya satu kali.

Tabel 3 Perolehan Medali dan Persentase Kesuksesan (Siklus II)

Pemain	Pl	E	Pr	Pg	Total	% Sukses
Zhuang	3	1	4	3	11	84,61538462
Bankaa	1	3	4	3	11	84,61538462

Banquero	5	3	0	1	9	69,23076923
Banker	2	3	2	2	9	69,23076923

Berdasarkan hasil pengujian yang dilakukan dalam dua siklus, ditemukan bahwa pemain yang menempati posisi pertama pada siklus I (Banker) tidak berhasil mempertahankan posisinya pada siklus II (menurun ke posisi terakhir). Oleh karena ketidakmampuan mempertahankan posisi puncak tersebut, maka peneliti akan mencari penyusun strategi terbaik dengan metode lainnya. Metode tersebut adalah metode kesuksesan kumulatif.

Pada Tabel 4, terlihat bahwa Zhuang ditempatkan di bawah Banker walaupun memiliki nilai s_k yang sama karena mengumpulkan medali platinum yang lebih sedikit. Banquero tidak pernah mengumpulkan medali perak dan hampir selalu memperoleh medali emas (setengah dari total medali keseluruhan). Bankaa menempati posisi terakhir serta lebih banyak mengoleksi medali perak dan perunggu. Hanya medali platinum yang berhasil diperoleh para *banker* setidaknya-tidaknnya satu kali.

Tabel 4 Perolehan Medali dan Persentase Kesuksesan (Kumulatif)

Pemain	Pl	E	Pr	Pg	Total	% Sukses
Banker	9	5	4	4	22	84,61538462
Zhuang	6	2	6	8	22	84,61538462
Banquero	8	10	0	2	20	76,92307692
Bankaa	3	4	6	6	19	73,07692308

C. Analisis Keunggulan dan Kelemahan Penyusunan Strategi

Ringkasan hasil analisis keunggulan dan kelemahan unsur *soft computing* dalam permainan TAT OSC pada Tabel 5 menguatkan definisi *soft computing*. Definsi tersebut menyatakan bahwa unsur-unsur *soft computing* sejatinya bukan merupakan pesaing satu dengan lainnya, namun saling melengkapi [1]. Hal tersebut terlihat dari kelemahan pada satu unsur merupakan kekuatan bagi unsur lainnya, misalnya frekuensi pernyataan “lewat” dan tingkat resiko dalam bermain.

Tabel 5 Keunggulan dan Kelemahan Penyusunan Strategi

Unsur	Keunggulan	Kelemahan
<i>Probabilistic reasoning</i>	Penawaran hampir selalu tinggi	Cenderung “rakus”
	Berani dalam mengambil resiko	Terlalu bermain “ngotot”
	Usaha tinggi	Resiko bermain tinggi
	Tidak pernah “lewat”	Karakteristik sulit ditebak
	Agresivitas tinggi	
Logika <i>fuzzy</i>	“Rakus” namun dapat dikendalikan	Penawaran cenderung rendah
	Tidak terlalu “berambisi” dalam bermain	Peluang kemenangan cenderung bernilai sedang
	“Cerdas” dan “bijak” dalam mengambil keputusan	
	Tidak pernah “lewat”	
	Resiko bermain rendah	
	Tingkat kemenangan wajar	
Jaringan syaraf tiruan	Nilai evaluasi sebagai pembatas nilai acak	Dapat “lewat” sewaktu-waktu
	Usaha normal	Kemenangan tidak terlalu total
Algoritma genetika	Mampu mempertahankan nilai penawaran yang tinggi	Nilai yang dipertahankan dapat berupa nilai yang tidak diizinkan sistem
	Mampu “menebak” nilai dengan cukup ampuh	Sering “lewat” Tingkat kemenangan cenderung “ter-“ (<i>terbaik</i> atau bahkan <i>terburuk</i>)

V. KESIMPULAN

Berdasarkan hasil dari penelitian yang telah dilakukan, maka dapat disimpulkan bahwa *probabilistic reasoning* merupakan unsur *soft computing* yang digunakan sebagai metode untuk menyusun strategi bermain terbaik dalam permainan TAT OSC. Tingkat kebaikan strategi tersebut diukur dari kesuksesan pemain AI dalam memenangkan permainan.

Prototipe permainan ini masih jauh dari sempurna. Ke depannya, diharapkan peneliti selanjutnya dapat melakukan hal-hal berikut.

- Menggunakan unsur *soft computing* lainnya yang merupakan pengembangan dari unsur-unsur pokok sehingga dapat dilihat cara penyusunan strateginya.
- Mengubah rumusan penawaran *banker* dengan memperluas atau mempersempit keadaan permainan ataupun merubah rumus penawaran serta menggunakan metode inferensi *soft computing* lainnya.
- Menambah jumlah siklus dan/atau jumlah percobaan permainan agar penentuan strategi terbaik dapat dilakukan dengan lebih teliti.

PENGAKUAN

Makalah ini dipublikasikan sebagai luaran tambahan dari kegiatan penelitian internal Institut Sains dan Bisnis Atma Luhur.

DAFTAR PUSTAKA

- [1] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu, 2003.
- [2] R. Kurzweil, *The Singularity is Near: When Humans Transcend Biology*, London: Penguin Books, 2005.
- [3] T. Sutojo, E. Mulyanto, dan V. Suhartono, *Kecerdasan Buatan*, Yogyakarta: ANDI, 2011.
- [4] Sinha, A., Malo, P., Franstev, A., dan Deb, K. (2014). Finding Optimal Strategies in a Multi-Period Multi-Leader-Follower Stackelberg Game Using an Evolutionary Algorithm. *Computers and Operations Research*. 41. hal. 374-385.
- [5] Y. A. Abdullah (2011). 'Perkembangan Soft Computing' [Online]. Tersedia: http://yopy-fst07.web.unair.ac.id/artikel_detail-24412-Kuliah%20Komcer-Perkembangan%20Soft%20Computing.html.
- [6] M. Dharmalingam dan R. Amalraj (2014). Supervised Elman Neural Network Architecture for Solving Double Dummy Bridge Problem in Contract Bridge. *International Journal of Science and Research*. 3. hal. 2745-2750.
- [7] E. Onieva, D.A. Pelta, V. Milanés, dan J. Pérez (2011). A fuzzy-rule-based driving architecture for non-player characters in a car racing game. *Soft Computing*. 15. hal. 1617-1629.
- [8] U. Markowska-Kaczmar dan M. Koldowski (2015). Spiking neural network vs multilayer perceptron: who is the winner in the racing car computer game. *Soft Computing*. 19. hal. 3465-3478.
- [9] J. J. Mereño, A.P. Castillo, A. Mora, A. Fernández-Ares, A.I. Esparcia-Alcázar, C. Cotta, *et al.* "Studying and Tackling Noisy Fitness in Evolutionary Design of Game Characters", dalam *Proc. of the ICOECTA*, 2014, hal. 76-85.
- [10] J. Gao (2013). Uncertain bimatrix game with applications. *Fuzzy Optimization and Decision Making*. 12. hal. 65-78.
- [11] E. B. Perkasa, A. Dendi, dan F. Panca (2015). Prototipe Permainan "Terima Atau Tolak" Dengan Metode *Probabilistic Reasoning*. *Jurnal Teknologi Informatika dan Komputer Atma Luhur*. 2(1). hal. 8-16.