

MancalaAHP: *Game* Tradisional Mancala Berbasis *Analytic Hierarchy Process*

Chandra Kusuma Dewa
Jurusan Teknik Informatika, Fakultas Teknologi Industri,
Universitas Islam Indonesia
Jl. Kaliurang Km 14 Yogyakarta 55501
ckusumadewa@gmail.com

Abstrak—Dalam makalah ini, dijabarkan penerapan metode *Analytic Hierarchy Process* (AHP) untuk *game* tradisional Mancala. Pada *game* yang telah berhasil dikembangkan, pemain dapat bermain melawan komputer yang menggunakan metode AHP untuk menghasilkan langkah permainan. Pembuatan langkah permainan selanjutnya dilihat sebagai sebuah proses pengambilan keputusan yang melibatkan sejumlah kriteria tertentu dari sekumpulan alternatif lubang dalam papan Mancala. Masing-masing alternatif lubang yang dimiliki oleh komputer akan diperbandingkan satu sama lain berdasarkan masing-masing kriteria, lalu lubang yang memiliki nilai bobot perbandingan paling tinggi akan dipilih oleh komputer. Hasil pengujian menunjukkan bahwa perolehan skor metode ini dapat mengungguli perolehan skor dari metode pemilihan lubang secara acak.

Kata kunci—Mancala; Game; Analytic Hierarchy Process

I. PENDAHULUAN

Mancala adalah salah satu jenis permainan tradisional yang menggunakan sebuah papan dengan lubang-lubang tertentu sebagai tempat untuk meletakkan sejumlah biji atau kerikil, yang selanjutnya disebut sebagai biji Mancala. Mancala dapat dimainkan oleh dua orang pemain yang akan secara bergantian memilih lubang dan memindahkan seluruh biji Mancala dari lubang tersebut satu per satu ke lubang-lubang lainnya dengan arah berlawanan jarum jam berdasarkan beberapa aturan tertentu. Pada akhir permainan, pemain yang bisa mendapatkan biji Mancala paling banyak dalam lubang nilai adalah pemenang [1].

Jika akan dikembangkan *game* versi elektronik dari permainan Mancala serta nantinya pemain dapat bermain melawan komputer, maka dibutuhkan sebuah metode tertentu agar komputer mampu menghasilkan langkah dengan menggunakan sejumlah pertimbangan tertentu agar permainan yang dihasilkan dapat menjadi lebih bervariasi dan lebih menarik. Metode yang digunakan juga harus dapat memastikan setiap langkah yang dihasilkan oleh komputer akan meningkatkan peluang agar komputer dapat memenangkan permainan dan mampu mengalahkan pemain.

Dari sejumlah pertimbangan yang dapat digunakan, salah satunya adalah tempat jatuhnya biji Mancala terakhir. Dalam aturan permainan Mancala, jika suatu lubang dipilih oleh salah

seorang pemain, serta biji Mancala terakhir akan jatuh ke lubang nilai milik pemain, maka giliran bermain selanjutnya akan tetap pada pemain tersebut. Jika kondisi tersebut terjadi, maka peluang pemain untuk bisa mendapatkan tambahan skor pada giliran bermain berikutnya akan menjadi semakin tinggi.

Kriteria kedua yang dapat digunakan adalah peluang untuk dapat mencuri biji Mancala lawan. Jika suatu alternatif lubang Mancala dipilih dan biji Mancala terakhir akan jatuh pada lubang kosong milik pemain, maka pemain tersebut memiliki kesempatan untuk mencuri biji Mancala lawan yang posisinya berhadapan langsung dengan tempat jatuhnya biji Mancala terakhir. Jika opsi ini digunakan, pemain pasti akan kehilangan giliran bermain selanjutnya. Meskipun demikian, jika tambahan biji Mancala yang bisa didapatkan cukup banyak, maka opsi ini layak untuk dipertimbangkan.

Selain itu, kriteria yang tidak kalah penting selanjutnya adalah kemungkinan biji Mancala dapat dicuri oleh pihak lawan. Jika kriteria sebelumnya lebih bersifat ofensif, maka kriteria ini lebih bersifat defensif. Penekanan utama pada kriteria ini adalah bagaimana agar biji Mancala dapat diamankan dan tidak dicuri oleh lawan. Jika pihak lawan dapat dicegah untuk dapat mencuri biji Mancala, maka peluang lawan untuk bisa menambah skor permainan dapat diminimalkan.

Terakhir, kriteria lain yang juga bisa digunakan yakni apakah biji Mancala bisa didapatkan jika suatu lubang dalam papan Mancala dipilih. Dari setiap alternatif lubang yang mungkin dapat dipilih, dimungkinkan terdapat alternatif-alternatif yang jika dipilih tidak akan menambah perolehan skor pemain. Fokus dari kriteria ini adalah memastikan bahwa setiap alternatif lubang yang dipilih dapat menambah perolehan skor.

II. ANALYTIC HIERARCHY PROCESS

Metode *Analytic Hierarchy Process* (AHP) pertama kali diusulkan oleh Saaty [2] serta merupakan salah satu metode yang penting dan banyak digunakan untuk permasalahan *multiple criteria decision making* (MCDM), yakni proses pembuatan keputusan yang melibatkan banyak kriteria yang saling bertentangan satu sama lainnya [3], [4]. Secara garis besar, metode ini terdiri dari tiga buah langkah utama,

diantaranya: langkah dekomposisi permasalahan, langkah penilaian perbandingan, serta langkah sintesa prioritas [2], [5].

Pada langkah pertama, dibangun representasi skematik secara hierarkis untuk tiap kriteria serta untuk tiap alternatif berdasarkan masing-masing kriteria. Hasil yang didapatkan dari langkah pertama selanjutnya digunakan pada langkah penilaian perbandingan untuk menyusun sejumlah *pairwise comparison matrix* yang menggambarkan perbandingan tingkat kepentingan untuk masing-masing kriteria ataupun untuk masing-masing alternatif. Pada langkah sintesa prioritas, dihitung *overall composite weight* berdasarkan keseluruhan *pairwise comparison matrix* untuk masing-masing alternatif, serta akan dipilih alternatif dengan nilai bobot paling tinggi [6].

Penentuan bobot skala perbandingan untuk dua buah kriteria ataupun alternatif dalam *pairwise comparison matrix* menggunakan nilai dalam rentang interval 1-9 yang dijabarkan pada Tabel I, sementara bentuk umum dari *pairwise comparison matrix* dijabarkan pada Persamaan 1.

TABLE I. BOBOT SKALA PERBANDINGAN PADA METODE AHP

Nilai Skala Bobot	Keterangan
1	Dua objek sama-sama penting
3	Salah satu objek sedikit lebih penting dari objek lain
5	Salah satu objek benar-benar lebih penting dari objek lain
7	Salah satu objek sangat lebih penting dari objek lain
9	Salah satu objek absolut lebih penting
2, 4, 6, 8	Skala bobot pertengahan dari bobot-bobot di atas

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \quad (1)$$

Hal yang harus diperhatikan pada pembentukan *pairwise comparison matrix* adalah penentuan bobot perbandingan tingkat kepentingan untuk masing-masing alternatif ataupun kriteria, serta tingkat konsistensi dari matriks tersebut [8]. Agar tingkat konsistensi dapat tetap terjaga, setiap elemen dari *pairwise comparison matrix* sedapat mungkin bisa memenuhi Persamaan 2 dan Persamaan 3.

$$a_{i,j} = a_{i,k} \cdot a_{k,j} \quad (2)$$

$$a_{i,j} = \frac{1}{a_{j,i}} \quad (3)$$

Misalkan akan dibandingkan tiga buah alternatif, yakni alternatif A, alternatif B, dan alternatif C untuk suatu kriteria tertentu, serta ditentukan bahwa alternatif A adalah 2 kali lebih penting dibandingkan dengan alternatif B, dan alternatif B adalah 3 kali lebih penting dibandingkan alternatif C. Untuk membentuk *pairwise comparison matrix* dari ketiga alternatif tersebut, maka langkah selanjutnya adalah menentukan nilai bobot perbandingan alternatif B terhadap alternatif A, alternatif C terhadap alternatif B, alternatif A terhadap alternatif C, dan alternatif C terhadap alternatif A.

Agar *pairwise comparison matrix* untuk ketiga alternatif tersebut dapat menjadi konsisten, maka alternatif B harus diberi

bobot 1/2 kali lebih penting dibandingkan dengan alternatif A serta alternatif C juga harus diberi bobot 1/3 kali lebih penting dari alternatif B. Terakhir, alternatif A harus diberi bobot 6 kali lebih penting dibandingkan dengan alternatif C, serta alternatif C harus diberi bobot 1/6 kali lebih penting dibandingkan dengan alternatif A.

Untuk mengukur tingkat konsistensi perbandingan dalam *pairwise comparison matrix*, Saaty [2] telah mengusulkan sebuah variabel yang disebut sebagai *consistency index* (CI). Menurut Taylor-III [9], nilai CI dihitung dari *eigenvalue* terbesar dalam *pairwise comparison matrix*, yang disimbolkan sebagai λ_{max} , dan jumlah alternatif ataupun kriteria yang akan diperbandingkan, yang disimbolkan sebagai n. Formula untuk menghitung nilai CI dijabarkan pada Persamaan 4.

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (4)$$

Setelah nilai CI berhasil dihitung, maka langkah selanjutnya adalah menghitung nilai *consistency ratio* (CR), nilai CR dihitung dengan membagi nilai CI dengan nilai *random index* (RI), yang nilainya ditentukan oleh jumlah alternatif atau jumlah kriteria yang akan diperbandingkan (n). Penentuan nilai RI dapat dilihat pada Tabel II, sementara formula untuk menghitung nilai CR dijabarkan pada Persamaan 5.

$$CR = \frac{CI}{RI} \quad (5)$$

TABLE II. NILAI RANDOM INDEX PADA METODE AHP

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Jika nilai CR dari suatu *pairwise comparison matrix* kurang dari atau sama dengan 10%, maka inkonsistensi dari matriks tersebut masih dapat diterima serta proses pembuatan keputusan menggunakan metode AHP dapat dilanjutkan, namun jika nilai CR lebih dari 10%, maka bobot-bobot dalam tabel *pairwise comparison matrix* tersebut harus disesuaikan kembali.

III. METODE AHP UNTUK GAME MANCALA

Pada bagian ini, dijabarkan bagaimana implementasi metode AHP untuk game Mancala. Langkah implementasi dimulai dengan penentuan kriteria yang akan digunakan, perbandingan antar kriteria, serta perbandingan antar alternatif untuk setiap kriteria.

A. Kriteria Yang Digunakan

Penerapan metode AHP untuk game Mancala dalam penelitian ini menggunakan empat buah kriteria, yakni tempat jatuhnya biji Mancala terakhir (JBT), kesempatan untuk mencuri biji Mancala lawan (MBL), peluang biji Mancala dicuri oleh lawan (DBL), dan apakah biji Mancala bisa didapatkan (BMD). Gambar 1 menunjukkan struktur hierarki dari kasus permasalahan yang diteliti, yakni pemilihan lubang Mancala oleh komputer berdasarkan keempat kriteria tersebut.

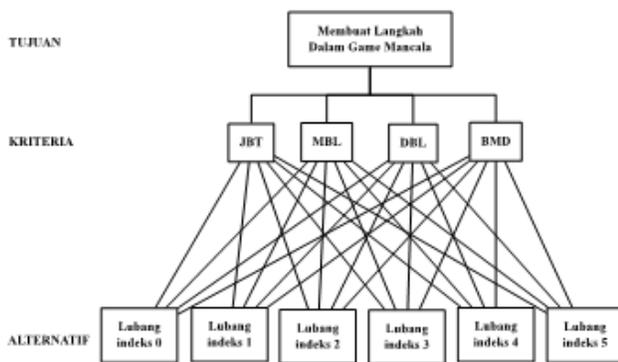


Fig. 1. Struktur Hierarki Permasalahan

B. Perbandingan Antar Kriteria

Ketika bermain melawan komputer, dalam rancangan game yang diusulkan, pemain nantinya dapat memilih strategi yang akan digunakan oleh komputer dalam bermain Mancala. Strategi bermain oleh komputer tersebut dibagi menjadi dua, yakni strategi menyerang serta strategi bertahan. Strategi menyerang akan menempatkan kriteria MBL lebih penting dibandingkan dengan kriteria DBL, sementara strategi bertahan akan menempatkan kriteria DBL lebih penting dibandingkan kriteria MBL. Detail dari strategi menyerang dan strategi bertahan ditunjukkan pada Tabel III.

TABLE III. STRATEGI MENYERANG DAN BERTAHAN KOMPUTER

Strategi Bermain Komputer	
Menyerang	Bertahan
JBT 2 kali lebih penting dari MBL	JBT 2 kali lebih penting dari DBL
MBL 2 kali lebih penting dari DBL	DBL 2 kali lebih penting dari MBL
DBL 2 kali lebih penting dari BMD	MBL 2 kali lebih penting dari BMD

Dari tabel di atas, dapat dilihat bahwa kriteria JBT tetap merupakan kriteria yang paling penting. Hal tersebut cukup beralasan mengingat jika kriteria ini terpenuhi, maka komputer dapat mempertahankan giliran bermain selanjutnya. Selain itu, terlihat pula bahwa kriteria BMD merupakan kriteria yang paling tidak penting dari ketiga kriteria lainnya karena kriteria ini hanya menekankan apakah tambahan skor akan didapatkan atau tidak, jika suatu lubang dipilih.

Strategi menyerang dan strategi bertahan komputer selanjutnya diimplementasikan ke dalam *pairwise comparison matrix* yang ditunjukkan pada Tabel IV dan Tabel V. Masing-masing elemen matriks dari kedua tabel tersebut dihitung berdasarkan formula pada Persamaan 2 dan Persamaan 3.

TABLE IV. PAIRWISE COMPARISON MATRIX UNTUK STRATEGI MENYERANG

Menyerang	JBT	MBL	DBL	BMD	Priority Vector
JBT	1	2	4	8	0.534
MBL	1/2	1	2	4	0.267
DBL	1/4	1/2	1	2	0.134
BMD	1/8	1/4	1/2	1	0.067
$\lambda_{max} = 4.0, CI = 0.0, CR = 0.0$					

TABLE V. PAIRWISE COMPARISON MATRIX UNTUK STRATEGI BERTAHAN

Bertahan	JBT	MBL	DBL	BMD	Priority Vector
JBT	1	4	2	8	0.534
MBL	1/4	1	1/2	2	0.134
DBL	1/2	2	1	4	0.267
BMD	1/8	1/2	1/4	1	0.067
$\lambda_{max} = 4.0, CI = 0.0, CR = 0.0$					

Pada Tabel IV dan Tabel V tersebut, priority vector merupakan *normalized eigenvector* dari *pairwise comparison matrix* yang nilainya dihitung dengan cara membagi setiap elemen dalam *eigenvector* matriks dengan jumlah kriteria ataupun jumlah alternatif. Nilai *priority vector* untuk masing-masing kriteria akan menunjukkan tingkat prioritas dari kriteria tersebut jika dibandingkan dengan kriteria-kriteria lainnya. Semakin tinggi nilai *priority vector* suatu kriteria, semakin tinggi pula prioritas kriteria tersebut dibandingkan kriteria-kriteria lainnya.

C. Perbandingan Antar Alternatif Untuk Kriteria JBT

Untuk kriteria tempat jatuhnya biji Mancala terakhir (JBT), tiap-tiap alternatif lubang dalam papan Mancala yang dapat dipilih oleh komputer akan diklasifikasikan ke dalam kelas-kelas tertentu berdasarkan tempat jatuhnya biji Mancala terakhir. Kelas untuk suatu lubang tertentu akan memiliki tingkat kepentingan yang lebih tinggi jika biji Mancala terakhir pada lubang tersebut jatuh ke lubang nilai milik komputer serta posisi lubang dekat dengan lubang nilai milik komputer.

D. Perbandingan Antar Alternatif Untuk Kriteria MBL

Pada kriteria kesempatan mencuri biji Mancala lawan (MBL), tiap-tiap alternatif lubang dalam papan Mancala yang dapat dipilih oleh komputer akan diklasifikasikan ke dalam kelas-kelas tertentu berdasarkan ada atau tidaknya kesempatan untuk mencuri biji Mancala lawan. Kelas untuk suatu lubang tertentu akan memiliki tingkat kepentingan yang lebih tinggi jika biji Mancala terakhir pada lubang tersebut dapat mencuri biji-biji Mancala dalam lubang lawan. Semakin banyak jumlah biji Mancala yang dapat dicuri jika suatu lubang dipilih, maka tingkat kepentingan untuk lubang tersebut akan semakin tinggi.

E. Perbandingan Antar Alternatif Untuk Kriteria DBL

Pada kriteria ketiga yang digunakan, yakni peluang biji Mancala dicuri oleh lawan (DBL), setiap alternatif lubang dalam papan Mancala milik komputer juga akan diklasifikasikan ke dalam kelas-kelas tertentu berdasarkan ada atau tidaknya peluang untuk dapat dicuri oleh lawan. Kelas untuk suatu lubang tertentu akan memiliki tingkat kepentingan yang lebih tinggi jika biji-biji Mancala dalam lubang tersebut dapat dicuri oleh lawan. Semakin banyak jumlah biji Mancala dalam suatu lubang yang dapat dicuri, maka tingkat kepentingan untuk lubang tersebut akan semakin tinggi.

F. Perbandingan Antar Alternatif Untuk Kriteria BMD

Untuk kriteria terakhir, yakni apakah biji Mancala bisa didapatkan (BMD), setiap alternatif lubang yang dapat dipilih oleh komputer juga akan diklasifikasikan ke dalam kelas-kelas

tertentu berdasarkan jumlah biji Mancala yang melewati lubang nilai milik komputer. Semakin banyak jumlah biji Mancala yang melewati lubang nilai milik komputer, maka kelas untuk lubang tersebut akan memiliki tingkat kepentingan yang semakin tinggi. Tambahan biji Mancala ke lubang nilai milik komputer yang didapatkan dari mencuri biji Mancala lawan tidak dihitung untuk kriteria ini.

IV. GAME MANCALA YANG DIKEMBANGKAN

Game Mancala pada penelitian ini dikembangkan dengan menggunakan bahasa pemrograman Java berbasis Java Standard Edition (Java SE) serta menggunakan Swing API. Masing-masing lubang dalam papan Mancala diimplementasikan menjadi objek JPanel yang diwujudkan dalam bangun persegi ataupun persegi panjang yang ditunjukkan pada Gambar 2.



Fig. 2. Tampilan Game yang Dikembangkan

Pada gambar di atas, enam buah persegi pada sisi bawah adalah lubang milik pemain, sementara enam persegi pada sisi atas adalah lubang milik komputer. Lubang nilai milik pemain adalah kotak persegi panjang pada sebelah kanan, sementara lubang nilai milik komputer adalah kotak persegi panjang sebelah kiri. Angka di dalam bangun persegi ataupun persegi panjang menyatakan jumlah biji Mancala yang ada pada lubang tersebut. Ketika tombol setting ditekan, pemain dapat memilih strategi yang akan digunakan oleh komputer yang ditunjukkan pada Gambar 3.



Fig. 3. Strategi Bermain Komputer

V. CONTOH KASUS DAN PEMBAHASAN

Misalkan diberikan kondisi papan Mancala yang ditunjukkan pada Gambar 4. Jika giliran bermain selanjutnya ada pada komputer, maka terdapat lima buah alternatif lubang yang dapat dipilih, yang ditunjukkan pada Tabel VI. Kelima alternatif lubang tersebut selanjutnya akan dibandingkan satu dengan lainnya berdasarkan keempat kriteria yang sudah dijabarkan sebelumnya, sehingga akan terbentuk empat buah *pairwise comparison matrix*. Dari keempat matriks tersebut,

selanjutnya akan dihitung *overall composite weight* untuk menentukan lubang mana yang sebaiknya dipilih oleh komputer.

indeks	5	4	3	2	1	0	
2	0	6	2	6	7	7	4
	6	5	1	1	0	1	

Fig. 4. Contoh Kasus Papan Mancala

TABLE VI. ALTERNATIF LUBANG DAPAT DIPILIH KOMPUTER

Indeks Lubang	Jumlah Biji Mancala
0	7
1	7
2	6
3	2
4	6

Untuk kriteria JBT, dari kelima alternatif lubang yang mungkin dapat dipilih oleh komputer pada Tabel VI, tidak ada satu lubang pun yang jika dipilih maka biji Mancala terakhir dari lubang tersebut akan jatuh ke lubang nilai milik komputer. Oleh karena itu, kelima lubang tersebut adalah sama-sama penting, sehingga bobot perbandingan untuk masing-masing lubang diberikan nilai 1. *Pairwise comparison matrix* untuk kriteria JBT dari kelima lubang tersebut ditunjukkan pada Tabel VII.

TABLE VII. PAIRWISE COMPARISON MATRIX UNTUK KRITERIA JBT

JBT	Lubang Indeks 0	Lubang Indeks 1	Lubang Indeks 2	Lubang Indeks 3	Lubang Indeks 4	Priority Vector
Lubang Indeks 0	1	1	1	1	1	0.2
Lubang Indeks 1	1	1	1	1	1	0.2
Lubang Indeks 2	1	1	1	1	1	0.2
Lubang Indeks 3	1	1	1	1	1	0.2
Lubang Indeks 4	1	1	1	1	1	0.2
$\lambda_{max} = 5.0, CI = 0.0, CR = 0.0$						

Untuk kriteria MBL, dari kelima alternatif lubang yang mungkin dipilih oleh komputer, hanya lubang dengan indeks 3 yang dapat mencuri biji Mancala lawan. Jika lubang dengan indeks 3 dipilih, maka biji Mancala terakhir dari lubang tersebut akan jatuh pada lubang kosong dengan indeks 5 milik komputer. Komputer selanjutnya dapat mencuri enam buah biji Mancala milik pemain yang letaknya berhadapan langsung dengan tempat jatuhnya biji Mancala terakhir.

Karena lubang dengan indeks 3 sedikit lebih penting jika dibandingkan dengan alternatif-alternatif lubang lainnya, maka lubang tersebut diberikan bobot 2 kali lebih penting dari

alternatif-alternatif lubang lain. *Pairwise comparison matrix* untuk kriteria MBL dari kelima alternatif lubang untuk kasus yang diberikan ditunjukkan pada Tabel VIII.

TABLE VIII. PAIRWISE COMPARISON MATRIX UNTUK KRITERIA MBL

MBL	Lubang Indeks 0	Lubang Indeks 1	Lubang Indeks 2	Lubang Indeks 3	Lubang Indeks 4	Priority Vector
Lubang Indeks 0	1	1	1	1/2	1	0.167
Lubang Indeks 1	1	1	1	1/2	1	0.167
Lubang Indeks 2	1	1	1	1/2	1	0.167
Lubang Indeks 3	2	2	2	1	2	0.333
Lubang Indeks 4	1	1	1	1/2	1	0.167
$\lambda_{max} = 5.0, CI = 0.0, CR = 0.0$						

Untuk kriteria DBL, tujuh buah biji Mancala dalam lubang milik komputer dengan indeks 1 berpeluang untuk dicuri oleh pemain. Jika lubang indeks 1 milik komputer akan dibandingkan dengan alternatif-alternatif lubang lainnya, maka lubang tersebut harus diberi bobot yang lebih tinggi dari alternatif-alternatif lain. Serupa dengan kriteria sebelumnya, karena lubang dengan indeks 1 sedikit lebih penting jika dibandingkan dengan lubang-lubang lainnya, maka lubang tersebut diberikan bobot nilai 2 kali lebih penting dari alternatif-alternatif lubang lain. *Pairwise comparison matrix* untuk kriteria DBL ditunjukkan pada Tabel IX.

TABLE IX. PAIRWISE COMPARISON MATRIX UNTUK KRITERIA DBL

DBL	Lubang Indeks 0	Lubang Indeks 1	Lubang Indeks 2	Lubang Indeks 3	Lubang Indeks 4	Priority Vector
Lubang Indeks 0	1	1/2	1	1	1	0.167
Lubang Indeks 1	2	1	2	2	2	0.333
Lubang Indeks 2	1	1/2	1	1	1	0.167
Lubang Indeks 3	1	1/2	1	1	1	0.167
Lubang Indeks 4	1	1/2	1	1	1	0.167
$\lambda_{max} = 5.0, CI = 0.0, CR = 0.0$						

Jika lubang milik komputer dengan indeks 0, lubang dengan indeks 1, lubang dengan indeks 2, dan lubang dengan indeks 4 dipilih, maka biji Mancala akan melewati lubang nilai sebanyak satu kali. Hal ini berarti tambahan satu skor untuk komputer bisa didapatkan. Untuk lubang dengan indeks 3, meskipun lubang tersebut dapat mencuri biji Mancala lawan jika dipilih, namun lubang tersebut akan memiliki nilai bobot prioritas yang lebih kecil jika dibandingkan dengan empat lubang lainnya untuk kriteria BMD.

Hal tersebut cukup beralasan mengingat jika lubang dengan indeks 3 dipilih, maka biji Mancala tidak akan melewati lubang nilai. Oleh karena itu, untuk kriteria BMD, keempat lubang tersebut akan memiliki bobot 2 kali lebih penting jika dibandingkan dengan lubang dengan indeks 3. *Pairwise comparison matrix* untuk kriteria BMD dari kasus tersebut ditunjukkan pada Tabel X.

TABLE X. PAIRWISE COMPARISON MATRIX UNTUK KRITERIA BMD

BMD	Lubang Indeks 0	Lubang Indeks 1	Lubang Indeks 2	Lubang Indeks 3	Lubang Indeks 4	Priority Vector
Lubang Indeks 0	1	1	1	2	1	0.222
Lubang Indeks 1	1	1	1	2	1	0.222
Lubang Indeks 2	1	1	1	2	1	0.222
Lubang Indeks 3	1/2	1/2	1/2	1	1/2	0.111
Lubang Indeks 4	1	1	1	2	1	0.222
$\lambda_{max} = 5.0, CI = 0.0, CR = 0.0$						

Setelah keempat *pairwise comparison matrix* berhasil didapatkan, maka langkah terakhir adalah menghitung *overall composite weight* untuk menentukan lubang mana yang sebaiknya dipilih. Tabel XI dan Tabel XII menunjukkan nilai *overall composite weight* dari masing-masing alternatif lubang untuk strategi menyerang dan untuk strategi bertahan.

Pada Tabel XI, nilai bobot menyerang didapatkan dari *priority vector* untuk masing-masing kriteria dari Tabel IV. Sementara itu, pada Tabel XII, nilai bobot bertahan didapatkan dari *priority vector* untuk masing-masing kriteria dari Tabel V. Bobot masing-masing alternatif lubang untuk masing-masing kriteria didapatkan dari *priority vector* milik keempat *pairwise comparison matrix* yang sudah disebutkan sebelumnya.

TABLE XI. OVERALL COMPOSITE WEIGHT UNTUK STRATEGI MENYERANG

	JBT	MBL	DBL	BMD	Composite Weight
Bobot Menyerang	0.534	0.267	0.134	0.067	-
Lubang Indeks 0	0.20	0.167	0.167	0.222	0.188
Lubang Indeks 1	0.20	0.167	0.333	0.222	0.210
Lubang Indeks 2	0.20	0.167	0.167	0.222	0.188
Lubang Indeks 3	0.20	0.333	0.167	0.111	0.225
Lubang Indeks 4	0.20	0.167	0.167	0.222	0.188

TABLE XII. OVERALL COMPOSITE WEIGHT UNTUK STRATEGI BERTAHAN

	JBT	MBL	DBL	BMD	Composite Weight
Bobot Bertahan	0.534	0.134	0.267	0.067	-
Lubang Indeks 0	0.20	0.167	0.167	0.222	0.188
Lubang Indeks 1	0.20	0.167	0.333	0.222	0.232
Lubang Indeks 2	0.20	0.167	0.167	0.222	0.188
Lubang Indeks 3	0.20	0.333	0.167	0.111	0.203
Lubang Indeks 4	0.20	0.167	0.167	0.222	0.188

Untuk menghitung nilai *composite weight* masing-masing alternatif lubang dari Tabel XI dan Tabel XII, dilakukan

penjumlahan dari perkalian antara bobot menyerang atau bobot bertahan dengan bobot masing-masing lubang berdasarkan masing-masing kriteria. Sebagai contoh, nilai *composite weight* dari lubang indeks 1 untuk strategi menyerang dihitung dari $(0.534 \times 0.2) + (0.267 \times 0.167) + (0.134 \times 0.333) + (0.067 \times 0.222) = 0.210$, sementara nilai *composite weight* dari lubang indeks 3 untuk strategi bertahan dihitung dari $(0.534 \times 0.2 + (0.34 \times 0.333) + (0.267 \times 0.167) + (0.067 \times 0.111) = 0.203$.

Berdasarkan Tabel XI, jika dipilih strategi menyerang, maka komputer akan memilih lubang indeks 3 dengan nilai *overall composite weight* tertinggi, yakni 0.225. Jika lubang tersebut dipilih, maka biji Mancala terakhir akan jatuh pada lubang dengan indeks 5 dan komputer dapat mencuri enam buah biji Mancala milik pemain. Kondisi akhir papan Mancala setelah lubang dengan indeks 3 dipilih ditunjukkan pada Gambar 5.

indeks	5	4	3	2	1	0	
	0	7	0	6	7	7	
9	0	5	1	1	0	1	4

Fig. 5. Kondisi Papan Jika Dipilih Strategi Menyerang

Untuk strategi bertahan, sesuai Tabel XII, maka komputer akan memilih lubang Mancala indeks 1 dengan nilai *overall composite weight* tertinggi, yakni 0.232. Jika lubang dengan indeks 1 dipilih, maka tujuh buah biji Mancala milik komputer dapat diselamatkan serta dapat terhindar dari peluang untuk dicuri oleh pemain. Kondisi akhir papan Mancala setelah lubang dengan indeks 1 dipilih ditunjukkan pada Gambar 6.

indeks	5	4	3	2	1	0	
	1	7	3	7	0	7	
3	7	6	1	1	0	1	4

Fig. 6. Kondisi Papan Mancala Jika Dipilih Strategi Bertahan

VI. PENGUJIAN

Pengujian dilakukan dengan cara membandingkan metode AHP strategi menyerang, metode AHP strategi bertahan, dan metode pemilihan lubang secara acak melalui 100 kali pertandingan antara komputer melawan komputer untuk setiap pasang metode. Hasil pengujian selanjutnya dapat dilihat pada Tabel XIII.

Dari tabel tersebut, dapat dilihat bahwa baik metode AHP strategi menyerang maupun metode AHP strategi bertahan hampir selalu memenangkan pertandingan jika dibandingkan dengan metode pemilihan lubang secara acak. Hal tersebut menunjukkan bahwa metode AHP dapat memperbesar peluang kemenangan karena pemilihan lubang dilakukan dengan

mempertimbangkan keempat kriteria yang sudah disebutkan sebelumnya.

TABLE XIII. HASIL PENGUJIAN METODE AHP UNTUK GAME MANCALA

Metode Dibandingkan	Jumlah Menang		Draw	Jumlah Game
	Komputer 1	Komputer 2		
Acak vs. AHP Menyerang	0	100	0	100
Acak vs. AHP Bertahan	0	99	1	100
AHP Menyerang vs. AHP Bertahan	0	0	100	100

VII. KESIMPULAN

Dari pengujian yang telah dilakukan, dapat ditarik kesimpulan bahwa Metode AHP dapat diimplementasikan ke dalam game Mancala dengan menggunakan keempat kriteria yang telah disebutkan sebelumnya. Langkah permainan Mancala yang dihasilkan dari metode AHP juga cukup bagus, serta metode ini dapat meningkatkan peluang untuk memenangkan permainan jika dibandingkan dengan metode pemilihan lubang secara acak.

DAFTAR PUSTAKA

- [1] A. de Voogt, "Mancala: Games that count," 2001. [Online]. Available: <http://www.penn.museum/documents/publications/expedition/PDFs/43-1/Mancala.pdf>
- [2] T. L. Saaty, *The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation*. New York: McGraw-Hill, 1980.
- [3] V. S. Lai, B. K. Wong, and W. Cheung, "Group decision making in a multiple criteria environment: A case using the ahp in software selection," *European Journal of Operational Research*, vol. 137, no. 1, pp. 134-144, 2002.
- [4] S. H. Zanakis, A. Solomon, N. Wishart, and S. Dublisch, "Multiple attribute decision making: A simulation comparison of select methods," *European Journal of Operational Research*, vol. 107, no. 3, pp. 507-529, 1998.
- [5] W. L. Winston, *Operational Research: Applications and Algorithms*. California: Duxbury Press, 1994.
- [6] M. Janic and A. Reggiani, "An application of the multiple criteria decision making (mcdm) analysis to the selection of a new hub airport," *European Journal of Transport and Infrastructure Research*, vol. 2, no. 2, pp. 113-142, 2002.
- [7] A. Ishizaka and M. Lusti, "An intelligent tutoring system for ahp," in *Proceedings of the 9th International Conference on Operational Research*, 2003, pp. 215-223.
- [8] D. Dalalah, F. AL-Oqila, and M. Hayajneh, "Application of the analytic hierarchy process (ahp) in multi-criteria analysis of the selection of cranes," *Jordan Journal of Mechanical and Industrial Engineering*, vol. 4, no. 5, pp. 567-578, 2010.
- [9] B. W. Taylor-III, *Management Science*, (seven edition). New Jersey: Pearson Prentice-Hall, 2002.