

RANCANG BANGUN GAME TRADISIONAL MANCALA MENGUNAKAN PENDEKATAN BERORIENTASI OBJEK

Chandra Kusuma Dewa

Jurusan Teknik Informatika, Fakultas Teknologi Industri

Universitas Islam Indonesia

ckusumadewa@gmail.com

Abstrak

Dalam era teknologi saat ini, banyak permainan tradisional yang sudah mulai ditinggalkan. Berbagai macam dan jenis permainan tradisional kemudian digantikan perannya oleh beragam permainan elektronik seperti game-game yang dapat dijalankan pada perangkat PC maupun pada perangkat mobile. Agar tidak punah, salah satu upaya yang dapat dilakukan adalah membuat game versi elektronik dari permainan-permainan tradisional tersebut. Pada makalah ini, dijabarkan proses rancang bangun untuk salah satu game tradisional, yakni Mancala, dengan menggunakan pendekatan berorientasi objek. Game yang dikembangkan berbasis desktop, serta menggunakan Java Standard Edition (Java SE) yang menggunakan library Swing API. Hasil pengujian terhadap game menunjukkan bahwa semua aturan permainan Mancala telah berhasil diimplementasikan.

Kata Kunci: Mancala, rancang bangun, pendekatan berorientasi objek.

1. Pendahuluan

Mancala merupakan salah satu jenis permainan papan tradisional yang sebenarnya cukup populer. Permainan ini memiliki cukup banyak variasi, yang menurut Murray (1952) dan Russ (2000) bahkan hampir dimainkan di seluruh dunia. Meskipun demikian, tidak menutup kemungkinan bahwa nantinya permainan ini akan lambat laun ditinggalkan seiring dengan banyaknya permainan-permainan elektronik yang jauh lebih menarik yang jumlah dan jenisnya juga cukup beragam.

Agar hal tersebut tidak terjadi, berbagai *game* versi elektronik dari beragam permainan tradisional pun mulai banyak dikembangkan, tidak terkecuali permainan tradisional Mancala. *Game-game* versi elektronik dari permainan-permainan tradisional tersebut selanjutnya dapat digunakan sebagai media pengenalan tentang permainan-permainan tradisional serta aturan-aturan dalam permainan tersebut, mengingat *game* komputer merupakan sarana pembelajaran yang sangat *powerful* (Prensky, 2005).

Pada penelitian ini, diangkat topik tentang penerapan teknik analisis dan desain berorientasi objek untuk pengembangan *game* tradisional Mancala. Teknik analisis dan desain berorientasi objek dipilih karena teknik ini sangat

cocok untuk diterapkan pada proyek ataupun aplikasi yang menggunakan objek sebagai fokus utama (Munassar dan Govardhan, 2011).

Dalam makalah ini, paradigma penggunaan objek sebagai fokus utama juga dianggap cukup sesuai dengan *game* Mancala yang dikembangkan. Segala sesuatu yang ada dalam *game* selanjutnya dilihat sebagai objek, sehingga proses pengembangan aplikasi akan menjadi lebih sederhana, lebih cepat, serta lebih *reliable* (Shoval, 2007).



Gambar 1 Papan Permainan Mancala (Noraziah, dkk., 2013)

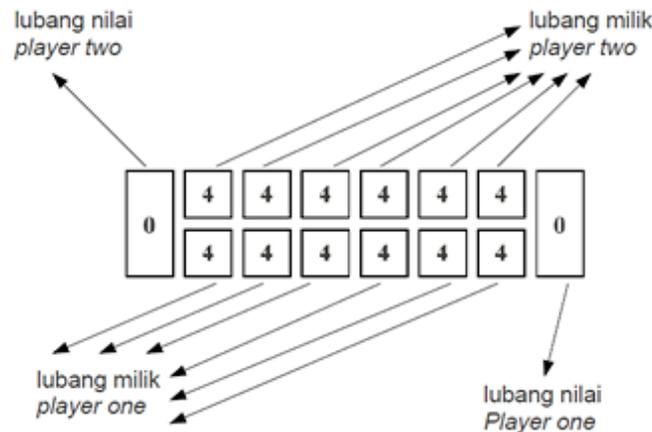
Sistematika penulisan yang digunakan dalam makalah ini adalah sebagai berikut: pada bagian 2 akan dibahas mengenai tinjauan pustaka yang digunakan dalam penelitian ini, pada bagian 3 akan dibahas mengenai analisis sistem, pada bagian 4 akan dibahas mengenai desain sistem, pada bagian 5 akan dibahas mengenai implementasi dan pengujian sistem, serta pada bagian 6 akan dibahas mengenai kesimpulan dan saran.

2. Tinjauan Pustaka

2.1 Permainan Mancala

Untuk dapat memainkan permainan Mancala, dibutuhkan sebuah papan permainan yang didalamnya terdapat sejumlah lubang-lubang tertentu sebagai tempat untuk menampung sejumlah biji atau kerikil yang selanjutnya disebut sebagai biji Mancala. Pada awal permainan, masing-masing pemain akan mendapatkan jumlah biji Mancala yang sama. Karena permainan ini dapat dimainkan oleh dua orang pemain saja, maka lubang-lubang yang ada pada papan permainan selanjutnya dibagi kepemilikannya untuk masing-masing pemain.

Masing-masing pemain nantinya akan mendapatkan dua buah tipe lubang, yakni sejumlah lubang biasa serta sebuah lubang nilai. Nantinya, setiap biji Mancala yang masuk ke lubang nilai merupakan tambahan skor untuk pemain tersebut. Dengan kata lain, lubang nilai akan menampung jumlah skor untuk masing-masing pemain.



Gambar 2 Ilustrasi Papan Permainan Mancala

Pada Gambar 2, dapat dilihat pembagian kepemilikan untuk masing-masing lubang pada papan permainan. Pada gambar tersebut, lubang biasa milik pemain digambarkan sebagai bangun persegi, sementara lubang nilai digambarkan sebagai bangun persegi panjang. Angka dalam lubang selanjutnya menyatakan jumlah biji Mancala yang ada dalam lubang tersebut.

Menurut Irving, dkk. (2000), permainan Mancala memiliki versi aturan-aturan yang beragam, tergantung tempat dimana permainan ini dimainkan. Dalam penelitian ini, digunakan aturan-aturan permainan sebagai berikut (Voogt, 2001; Sian dan Samuel, 2010):

- 1) Langkah permainan dibuat oleh pemain dengan cara memilih salah satu lubang tidak kosong yang bukan merupakan lubang nilai.
- 2) Pemain selanjutnya akan memindahkan satu per satu biji Mancala dari lubang yang dipilih ke lubang-lubang lainnya dengan arah berlawanan dengan jarum jam (*counter clockwise*). Langkah ini selanjutnya disebut sebagai "*sowing*".
- 3) Tempat jatuhnya biji Mancala terakhir akan menentukan apa yang akan terjadi selanjutnya.
- 4) Jika biji Mancala terakhir jatuh ke lubang nilai milik pemain, maka giliran bermain selanjutnya akan tetap pada pemain tersebut. Jika tidak, maka pemain yang lain yang akan mendapat giliran bermain selanjutnya.

- 5) Jika biji Mancala terakhir jatuh ke lubang kosong milik pemain itu sendiri, maka pemain dapat mencuri biji lawan yang letaknya berhadapan langsung dengan tempat jatuhnya biji Mancala terakhir.
- 6) Jika seluruh lubang nilai milik salah satu pemain sudah kosong, maka seluruh sisa biji mancala akan dimiliki oleh pemilik lubang tersebut dan akan dipindahkan seluruhnya ke lubang nilai pemilik lubang.
- 7) Pada akhir permainan, pemain yang bisa mendapatkan biji Mancala yang paling banyak adalah pemenang.

2.2 Analisis dan Desain Berorientasi Objek

Berbeda dengan metode pendekatan tradisional, sebuah sistem selanjutnya dilihat sebagai sekumpulan objek yang saling berinteraksi antara satu dengan lainnya dalam pendekatan berorientasi objek. Masing-masing objek nantinya akan memiliki sekumpulan local state serta sekumpulan operasi untuk *local state* tersebut. Untuk membangun sistem dengan pendekatan berorientasi objek, tahapan pembangunan sistem akan dibagi menjadi dua tahapan, yakni tahapan analisis dan tahapan desain.

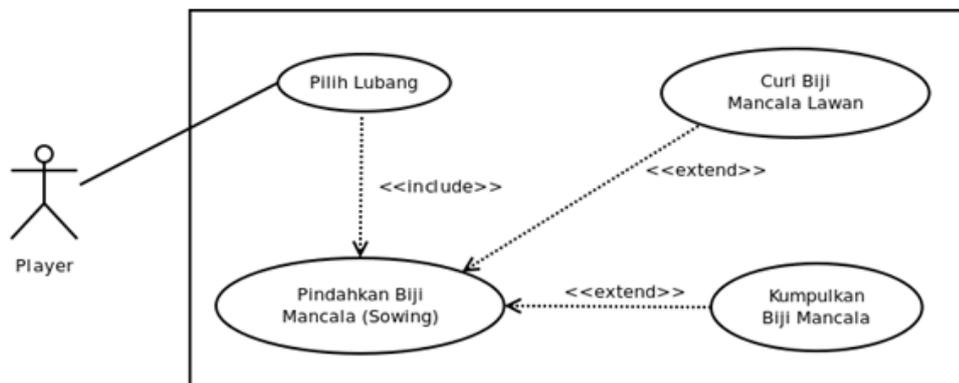
Pada tahapan analisis, dilakukan proses analisis, proses spesifikasi, serta proses definisi dari sistem yang akan dibangun dengan membuat serangkaian model agar pemahaman terhadap sistem dapat menjadi lebih mudah untuk dilakukan (Munassar dan Govardhan, 2011). Model yang dibangun pada tahapan ini akan menggambarkan tingkah laku (*behaviour*) dari sistem yang akan dibangun (Sommerville, 2007).

Pada tahapan desain, menurut Sommerville (2007), dibangun *object-oriented model* yang nantinya akan mengimplementasikan *software requirements* pada tahapan sebelumnya. *Object-Oriented model* sendiri merupakan model konseptual aplikasi yang menggambarkan *data/information domain* untuk aplikasi secara riil (Shoval, 2007).

3. Analisis Sistem

Pada tahapan analisis, untuk menggambarkan fungsionalitas dari sistem yang akan dibangun, selanjutnya digunakan *use case diagram*, sementara itu, untuk menggambarkan objek-objek apa saja yang nantinya akan ada pada sistem, akan digunakan *class diagram* (Munassar dan Govardhan, 2011). Pada bagian ini, akan dijabarkan tahapan analisis sistem untuk *game* Mancala.

Gambar 3 selanjutnya menunjukkan *use case diagram* dari *game Mancala* yang dikembangkan. Pada gambar tersebut, dapat dilihat bahwa terdapat sebuah aktor dari *game*, yakni *player* atau pemain. Pemain sendiri hanya dapat mengakses sebuah fungsi pada *game*, yakni memilih lubang untuk membuat langkah permainan. Ketika fungsi memilih lubang diakses oleh pemain, maka fungsi tersebut akan memanggil fungsi memindahkan biji Mancala (*sowing*) yang dapat pula memanggil fungsi mencuri biji Mancala ataupun mengumpulkan biji Mancala. Detail fungsi-fungsi tersebut dapat dilihat pada Tabel 1.

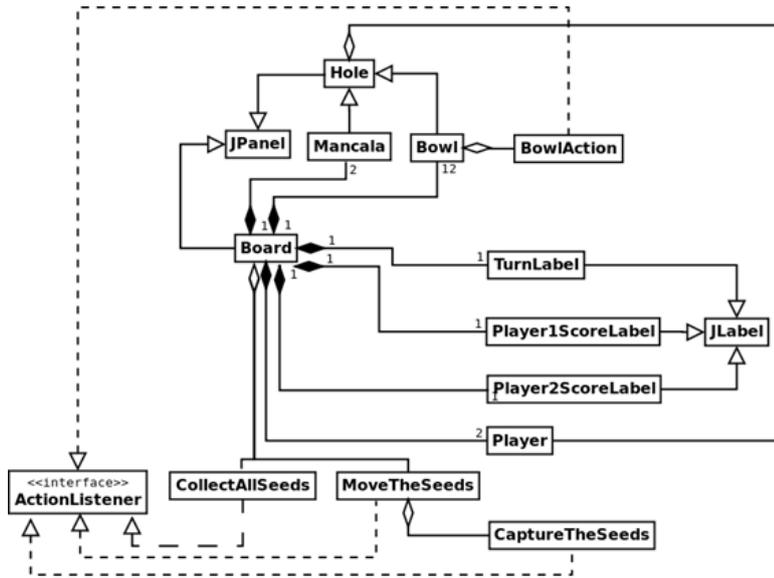


Gambar 3 Diagram *Use Case* Game Mancala

Tabel 1 Detail Diagram *Use Case* Game Mancala

Nama Use Case	Keterangan
Pilih Lubang	Pada fungsi ini, <i>player</i> dapat membuat langkah permainan dengan memilih salah satu lubang yang tidak kosong milik <i>player</i> tersebut.
Pindahkan Biji Mancala (<i>Sowing</i>)	Pada fungsi ini, aplikasi <i>game</i> Mancala akan memindahkan biji-biji Mancala dalam lubang yang dipilih oleh <i>player</i> satu per satu ke lubang-lubang lainnya.
Curi Biji Mancala Lawan	Fungsi ini akan dipanggil ketika biji Mancala terakhir pemain jatuh ke lubang kosong milik pemain tersebut. Fungsi ini selanjutnya akan mengambil seluruh biji Mancala lawan yang posisinya ada tepat di hadapan tempat jatuhnya biji Mancala terakhir milik pemain, untuk dimasukkan ke lubang nilai pemain tersebut.
Kumpulkan biji Mancala	Fungsi ini akan dipanggil ketika seluruh lubang milik salah satu pemain dalam kondisi kosong. Ketika kondisi ini terjadi, maka sisa seluruh biji Mancala akan dimasukkan ke lubang nilai pemilik lubang.

Gambar 4 menunjukkan diagram kelas untuk tahapan analisis dari *game Mancala*. Pada gambar tersebut terlihat beberapa objek yang ada pada *game*. Terlihat pula bahwa nantinya objek-objek tersebut akan memanfaatkan beberapa kelas pada Java Swing, yakni kelas *JPanel* untuk objek *Hole* dan *Board* serta *JLabel* untuk beberapa label dalam *game*. Selain itu, pada diagram kelas tersebut juga terdapat beberapa kelas yang mengimplementasikan *interface ActionListener* untuk menangani beberapa aksi yang ada dalam *game*. Detail dari masing-masing objek dapat dilihat pada Tabel 2.



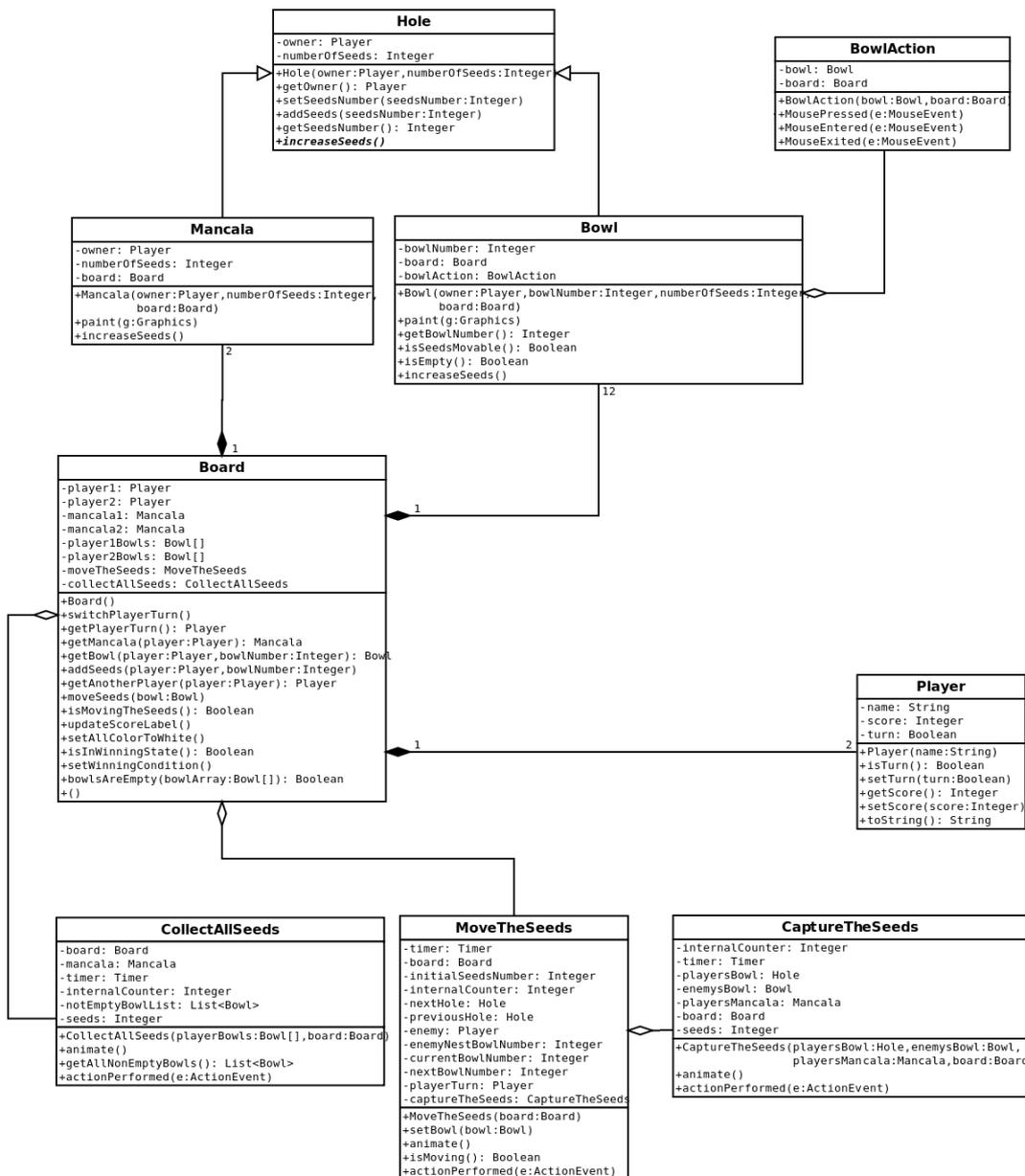
Gambar 4 Diagram Kelas Game Mancala (tahap analisis)

Tabel 2 Detail Diagram Kelas Game Mancala

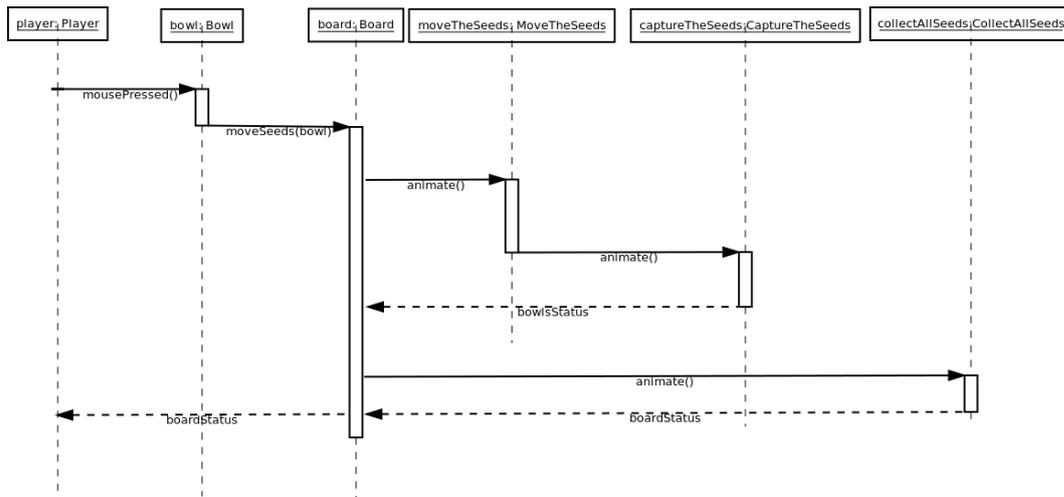
Nama Kelas	Keterangan
Board	Kelas ini nantinya akan bertanggung jawab membuat objek <i>board</i> , yang merupakan representasi papan dari <i>game</i> Mancala. Objek <i>board</i> nantinya akan terdiri dari 12 objek <i>bowl</i> yang merupakan lubang milik pemain, serta dua buah Mancala yang merupakan lubang nilai tiap pemain
Hole	Kelas Hole merupakan <i>super class</i> dari kelas <i>bowl</i> dan Mancala yang merepresentasikan lubang dalam papan Mancala.
Bowl	Kelas Bowl akan membuat objek <i>bowl</i> yang merupakan representasi lubang untuk masing-masing pemain. Kelas ini memiliki relasi dengan kelas BowlAction yang merupakan aksi ketika objek ini di-klik pemain
Mancala	Kelas ini nantinya akan merepresentasikan objek lubang-lubang nilai milik masing-masing pemain
Player	Kelas ini berfungsi untuk membuat dua buah objek yang merepresentasikan dua orang pemain pada <i>game</i> Mancala.
BowlAction	BowlAction merupakan kelas implementasi dari <i>interface</i> ActionListener. Kelas ini merupakan kelas aksi ketika pemain memilih salah satu lubang dalam papan Mancala.
MoveTheSeeds	Kelas ini juga merupakan implementasi dari <i>interface</i> ActionListener. Objek dari kelas ini merupakan aksi ketika objek papan memindahkan biji-biji Mancala dari lubang yang dipilih oleh pemain (representasi dari <i>sowing</i>).
CaptureTheSeeds	Kelas ini juga merupakan implementasi dari <i>interface</i> ActionListener. Objek dari kelas ini merupakan aksi ketika pemain mencuri biji Mancala lawan.
CollectAllSeeds	Kelas ini juga merupakan implementasi dari <i>interface</i> ActionListener. Objek dari kelas ini merupakan aksi ketika ada salah seorang pemain yang seluruh lubangnya kosong, sehingga objek papan akan mengumpulkan seluruh biji Mancala ke masing-masing lubang nilai pemain.
TurnLabel, Player1ScoreLabel, Player2ScoreLabel	Ketiga kelas ini merupakan turunan dari kelas JLabel yang nantinya akan merepresentasikan label giliran bermain, serta label skor untuk masing-masing pemain.

4. Desain Sistem

Tahapan desain sistem merupakan kelanjutan dari tahapan analisis sistem yang selanjutnya akan diikuti dengan tahapan implementasi. Model yang dihasilkan pada tahapan ini merupakan pemutakhiran (*refinement*) dan bentuk formal (*formalization*) dari model pada tahapan analisis (Jacobson, dkk., 1992). Gambar 5 selanjutnya menunjukkan *refinement* dari diagram kelas pada Gambar 4, sementara Gambar 6 menunjukkan *sequence diagram* yang merupakan *refinement* dari *use case diagram* pada Gambar 3.



Gambar 5 Diagram Kelas Game Mancala (tahapan desain)



Gambar 6 Diagram sekuens *game* Mancala

5. Implementasi dan Pengujian Sistem

Implementasi *game* Mancala selanjutnya menggunakan bahasa Java berbasis Java Standard Edition (JSE) dengan menggunakan *library* Swing API. Tampilan *game* yang sudah jadi dapat dilihat pada Gambar 7. Pemain nantinya dapat membuat langkah permainan dengan melakukan aksi klik pada salah satu lubang yang akan dipilih.



Gambar 7 Implementasi *Game* Mancala

Pengujian aplikasi dilakukan menggunakan teknik *black-box testing*, yakni dengan cara menguji apakah seluruh aturan pada permainan Mancala sudah berhasil diimplementasikan. Hasil pengujian selanjutnya ditunjukkan pada Tabel 3. Pada tabel tersebut, terlihat bahwa seluruh aturan pada permainan Mancala telah berhasil diimplementasikan.

Tabel 3 Pengujian *Game Mancala*

Aturan yang diuji	Hasil Pengujian
Memindahkan biji Mancala (Sowing)	Berhasil diimplementasikan
Pergantian kesempatan bermain	Berhasil diimplementasikan
Mencuri biji Mancala Lawan	Berhasil diimplementasikan
Mengumpulkan semua biji Mancala	Berhasil diimplementasikan

6. Kesimpulan dan Saran

Dari proses rancang bangun serta pengujian *game Mancala* yang telah dilakukan pada penelitian ini, maka dapat ditarik kesimpulan bahwa *game Mancala* yang dikembangkan telah berhasil mengimplementasikan aturan-aturan permainan pada permainan Mancala sesungguhnya. Pengembangan selanjutnya dapat dilakukan dengan melakukan proses *porting game Mancala* ke *platform Andorid* agar *game* dapat dimainkan pada perangkat berbasis *mobile*.

Daftar Pustaka

- de Voogt, A. J., 2001. *Mancala: Games That Count*. [Online] Available at: <http://www.penn.museum/documents/publications/expedition/PDFs/43-1/Mancala.pdf> [Accessed 18/3/2014].
- Irving, G., Donkers, H. H. L. M. & Uiterwijk, J. W. H. M., 2000. Solving Kalah. *ICGA Journal*, 23(3), pp.139-148.
- Jacobson, I., Christerson, M., & Oevergaard, G., 1992. *Object-Oriented Software Engineering – A Use Case Driven Approach*. Boston: Addison-Wesley Professional.
- Munassar, N. M. A. & Govardhan, A., 2011. Comparison Between Traditional Approach and Object-Oriented Approach in Software Engineering Development. *International Journal of Advanced Computer Science and Applications*, 2(6), pp. 70-76.
- Murray, H. J. R., 1952. *A History of Board Games other than Chess*. London: Oxford at the Clarendon Press.
- Noraziah, C. P., Alwi, A., Din, A. M. & Safwan, A., 2013. The Application of Neural Networks and Min-Max Algorithm in Digital Congkak. *The 4th International Conference on Computing and Informatics*, Serawak, pp. 222-227.
- Prensky, M., 2005. Computer Games and Learning: Digital Game-Based Learning. *Handbook of Computer Game Studies*, 18, pp. 97-12.
- Russ, L., 2000. *The Complete Mancala Games Book*. New York: Marlow and Company.
- Shoval, P., 2007. *Functional and Object Oriented Analysis and Design: An Integrated Methodology*. Hershey: Idea Group Publishing.
- Sian & Samuel, 2010. *More About Mancala*. [Online] Available at: <http://woodsgood.ca/mancalarules.htm> [Accessed 24/6/2014].
- Sommerville, I., 2007. *Software Engineering Eight Edition*. Essex: Pearson Education Limited.