
IMPLEMENTASI METODE *FORWARD CHAINING* UNTUK MENGELOMPOKKAN OBJEK BERDEKATAN PADA *SAME GAME*

¹Nicco Goetama, ²Emma Rosinta Simarmata, ³Christina N.M. Tobing

^{1,3}STMIK Time, ²Universitas Methodist Indonesia

¹nicco.goe92@gmail.com, ²emros2016@gmail.com

DOI: <https://doi.org/10.46880/jmika.Vol2No2.pp85-92>

ABSTRAK

Same game adalah salah satu jenis permainan komputer, dimana pemain dituntut untuk mencari dan menghilangkan sejumlah kotak sama yang berdekatan. Setiap kali pemain mengklik sebuah kotak, maka perangkat lunak harus mencari hubungan dari kotak tersebut ke kotak-kotak di sekelilingnya. Penulis merancang suatu aplikasi *Same game* yang menerapkan metode *Forward chaining* untuk mengelompokkan objek berdekatan yang sama. Perangkat lunak yang dirancang ini melakukan beberapa tahapan proses yaitu dimulai dari proses pengacakan posisi objek, proses penentuan posisi yang dapat dibuang, dan proses perhitungan nilai. Hasil dari penelitian ini adalah perangkat lunak *Same game* yang menerapkan konsep *Forward chaining* untuk mencari solusi.

Kata Kunci: *Forward Chaining, Same Game, Game.*

PENDAHULUAN

Latar Belakang

Setiap permainan memiliki aturan main. Hal ini mempermudah upaya menghasilkan ruang pencarian dan memberikan kebebasan pada para peneliti dari bermacam ambisi dan kompleksitas sifat serta kurangnya struktur permasalahan. Papan konfigurasi yang digunakan untuk memainkan permainan ini mudah untuk direpresentasikan pada komputer, tidak memerlukan bentuk yang kompleks untuk capture semantik dari domain permasalahan yang rumit.

Same game adalah salah satu jenis permainan komputer, dimana pemain dituntut untuk mencari dan menghilangkan sejumlah kotak sama yang berdekatan. Setiap kali pemain mengklik sebuah kotak, maka perangkat lunak harus mencari hubungan dari kotak tersebut ke kotak-kotak di sekelilingnya. Proses pencarian hubungan ini dapat menerapkan metode *Forward chaining*. Pada konsep ini, inferensi majemuk yang

menghubungkan permasalahan dengan solusinya disebut sebagai “chain” (rantai). Rantai yang dilalui dari masalah ke solusi disebut “forward chain”, yaitu penalaran dari fakta menuju konklusi berdasarkan fakta-fakta tersebut.

Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah mengimplementasikan metode *Forward chaining* untuk mengelompokkan objek pada *same game*.

Manfaat dari penelitian ini, yaitu aplikasi yang dibuat dapat digunakan sebagai sarana hiburan yang cukup menarik dan juga sebagai referensi bagi mahasiswa lainnya yang ingin mengangkat topik mengenai permainan.

ANALISIS SISTEM

Analisis sistem adalah aktivitas penguraian sistem ke dalam beberapa komponen yang membentuk sistem untuk mengkaji bagaimana komponen tersebut berinteraksi dan

berfungsi. Langkah analisa sistem dapat dibagi menjadi beberapa tahapan, yaitu:

1. Mengidentifikasi masalah yang melatarbelakangi diperlukannya perancangan sistem baru. Hal ini dapat dilakukan dengan mengkaji masalah-masalah yang dihadapi sekarang.
2. Mempelajari prosedur kerja dari sistem yang akan dibuat.
3. Mendefinisikan kebutuhan dari sistem yang akan dibuat.

Asumsi Permainan

Persyaratan yang harus dipenuhi oleh *Same game* yang akan dibangun dapat dikelompokkan menjadi persyaratan fungsional dan non-fungsional sistem. Beberapa persyaratan fungsional dari sistem dapat dirincikan sebagai berikut:

1. Permainan dimulai dari *level* 1 (satu) dengan ukuran kotak sebesar 8 x 8.
2. Batasan nilai yang diperoleh untuk setiap *level* agar melanjutkan ke *level* berikutnya dapat dijabarkan sebagai berikut:
 - a. *Level* 1 akan memiliki batasan skor 8.
 - b. *Level* 2 akan memiliki batasan skor 12.
 - c. *Level* 3 sampai *level* 4 akan memiliki batasan skor 16.
 - d. *Level* 5 sampai *level* 6 akan memiliki batasan skor 20.
 - e. *Level* 7 sampai *level* 8 akan memiliki batasan skor 24.
 - f. *Level* 9 sampai *level* 10 akan memiliki batasan skor 28.
 - g. *Level* 11 sampai *level* 12 akan memiliki batasan skor 32.
 - h. *Level* 13 sampai *level* 14 akan memiliki batasan skor 36.
 - i. *Level* 15 sampai *level* 16 akan memiliki batasan skor 40.

- j. *Level* 17 sampai *level* 18 akan memiliki batasan skor 44.
 - k. *Level* 19 sampai *level* 20 akan memiliki batasan skor 48.
 - l. *Level* 21 sampai *level* 22 akan memiliki batasan skor 52.
 - m. *Level* 23 ke atas akan memiliki batasan skor 60.
3. Ukuran kotak pada permainan dapat ditentukan sebagai berikut:
 - a. *Level* 2 akan memiliki ukuran kotak, yaitu 9 x 9.
 - b. *Level* 3 sampai *level* 4 akan memiliki ukuran kotak, yaitu 10 x 10.
 - c. *Level* 5 sampai *level* 6 akan memiliki ukuran kotak, yaitu 11 x 11.
 - d. *Level* 7 sampai *level* 8 akan memiliki ukuran kotak, yaitu 12 x 12.
 - e. *Level* 9 sampai *level* 10 akan memiliki ukuran kotak, yaitu 13 x 13.
 - f. *Level* 11 sampai *level* 12 akan memiliki ukuran kotak, yaitu 14 x 14.
 - g. *Level* 13 sampai *level* 14 akan memiliki ukuran kotak, yaitu 15 x 15.
 - h. *Level* 15 sampai *level* 16 akan memiliki ukuran kotak, yaitu 16 x 16.
 - i. *Level* 17 sampai *level* 18 akan memiliki ukuran kotak, yaitu 17 x 17.
 - j. *Level* 19 sampai *level* 20 akan memiliki ukuran kotak, yaitu 18 x 18.
 - k. *Level* 21 sampai *level* 22 akan memiliki ukuran kotak, yaitu 19 x 19.
 - l. *Level* 23 ke atas akan memiliki ukuran kotak, yaitu 20 x 20.
 4. Pada setiap *level*, gambar untuk setiap kotak akan ditentukan secara otomatis oleh komputer dengan memilih sembarang gambar secara acak.

5. Permainan dapat disimpan secara otomatis ketika pemain menyimpan permainan dan dapat dilanjutkan kembali ketika pemain kembali menjalankan permainan.

6. Permainan ini terdiri dari tiga jenis kesulitan, yaitu:

- a. Tingkat 'Mudah', dimana permainan dimainkan tanpa batasan waktu.
- b. Tingkat 'Sedang', dimana permainan dimainkan dengan adanya batasan waktu, namun permainan tidak akan berakhir. Tetapi skor yang diperoleh pemain akan dianggap sebagai tingkat 'Mudah'.
- c. Tingkat 'Susah', dimana permainan dimainkan dengan adanya batasan waktu. Permainan akan berakhir apabila waktu habis, namun pemain belum mampu mencapai nilai yang ditentukan.

7. Pada *level* 1, pemain diberikan 60 detik untuk menyelesaikan permainan untuk mencapai batasan nilai tertentu.

8. Sisa waktu pada setiap *level* akan diakumulasikan ke *level* berikutnya. Sebagai contoh jika pemain memerlukan 30 sekon untuk menyelesaikan permainan pada *level* pertama, maka pemain akan memiliki 90 sekon untuk menyelesaikan permainan pada *level* kedua.

9. Nilai ditentukan dengan menghitung total gambar yang telah dicocokkan. Semakin banyak gambar yang dikelompokkan secara bersama, maka nilai yang diperoleh merupakan kelipatan dari *level*. Cara perhitungannya adalah:

$$\{\text{Jumlah gambar}\} * \{\text{Level}\} * \{\text{Nilai Kesulitan}\}$$

Untuk tingkat 'Mudah', maka {Nilai Kesulitan} = 1

Untuk tingkat 'Sedang', maka {Nilai kesulitan} = 2

Untuk tingkat 'Susah', maka {Nilai Kesulitan} = 3

Kerangka kerja **PIECES** berikut digunakan untuk mengidentifikasi persyaratan non-fungsional yang berhubungan dengan kualitas sistem.

1. Performansi
 - a) Sistem harus mampu melakukan pengaturan gambar yang berdekatan.
 - b) Jenis gambar pada setiap kotak akan ditentukan secara acak.
2. Informasi
Sistem harus mampu menghitung dan menampilkan total nilai yang diperoleh pemain secara cepat dan akurat. Demikian juga halnya dengan sisa waktu untuk setiap *level*, sistem harus mampu menampilkan nilai pemain secara cepat dan akurat.
3. Ekonomis
Permainan harus dapat dimainkan pada berbagai versi dari sistem operasi Windows. *User* hanya perlu menginstall aplikasi permainan ini pada harddisk komputer.
4. Kontrol
Permainan harus dapat menampilkan pesan untuk *user* ketika permainan selesai atau ketika *user* berhasil menyelesaikan permainan.
5. Efisiensi
Permainan tidak memerlukan komponen atau aplikasi lainnya selain *Microsoft visual basic 2010*. Permainan harus dapat dimainkan setelah pemakai selesai meng-*install installer file*.
6. Pelayanan
Permainan harus memiliki desain gambar latar yang interaktif dan harus menyediakan fitur penyimpanan bagi *user* agar dapat

menyimpan permainan yang sedang dimainkan.

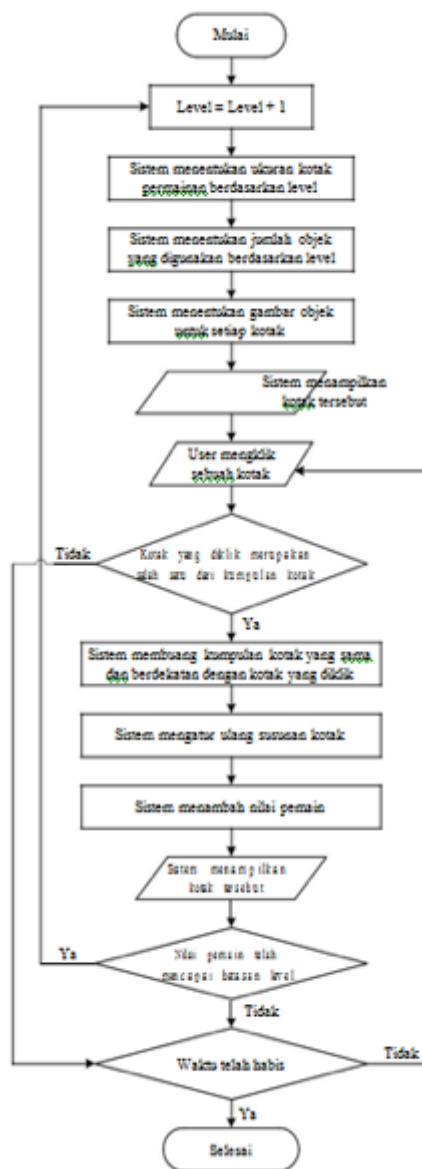
Prosedur Kerja Sistem

Same game merupakan sebuah permainan pencocokkan dan pengelompokkan objek sama yang terletak berdekatan. Objek yang digunakan dapat ditentukan secara manual oleh pemakai. Objek yang digunakan dapat berupa gambar huruf, warna, angka ataupun objek lainnya seperti kelereng dan sebagainya.

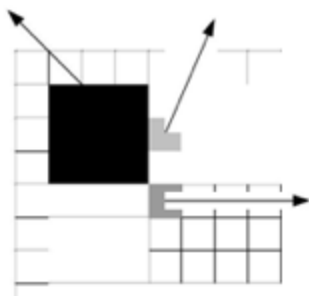
Prosedur kerja dari metode *forward chaining* dalam melakukan pengelompokkan objek sama pada *Same game* adalah sebagai berikut:

1. Pengecekan akan dimulai dari kotak yang diklik oleh pemakai.
2. Sistem akan melakukan pengecekan terhadap kotak yang terletak di sebelah kanan terlebih dahulu. Kemudian, proses akan dilanjutkan dengan pengecekan kotak yang terletak di bagian bawah dan dilanjutkan dengan kotak pada bagian kiri dan bagian atas. Namun, proses tidak dapat melakukan pengecekan terhadap kotak diagonal walaupun memiliki objek yang sama.
3. Kotak yang memiliki objek yang sama akan dikelompokkan menjadi satu kelompok.
4. Sebuah kotak akan dimasukkan ke dalam sebuah area apabila kotak tersebut merupakan kotak tetangga dari salah satu kotak dengan objek yang sama pada suatu area.

Proses kerja dari aplikasi *Same game* ini dapat dijabarkan seperti terlihat pada flowchart berikut:



Gambar 1. Flowchart Diagram dari Sistem



Gambar 2. Proses Pengelompokan Area

PERANCANGAN SISTEM

Perangkat lunak permainan ini dirancang dengan menggunakan bahasa pemrograman *Microsoft visual basic 2010* dan didukung oleh aplikasi *Adobe Photoshop CS* untuk merancang gambar latar (*background*) *interface* perangkat lunak.

Penggunaan Komponen

Aplikasi ini dirancang dengan menggunakan bahasa pemrograman *Microsoft visual basic 2010* dengan menggunakan beberapa objek dasar seperti :

1. *Button*, yang digunakan sebagai tombol eksekusi.
2. *Label*, yang digunakan untuk menampilkan keterangan.
3. *TextBox*, yang digunakan sebagai tempat penginputan data.
4. *RichTextBox*, yang digunakan untuk menampilkan hasil proses perhitungan.
5. *SaveFileDialog*, yang digunakan untuk menampilkan dialog save.
6. *PictureBox*, yang digunakan untuk menampilkan gambar.

Perancangan *Form*

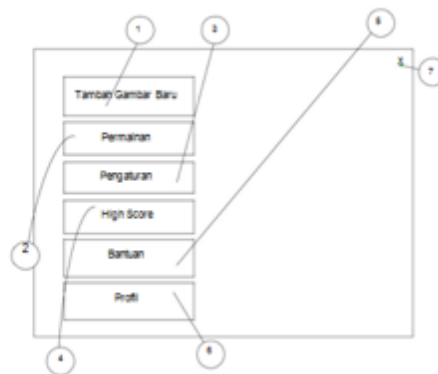
Rancangan tampilan dari perangkat lunak adalah sebagai berikut.

1. *Form* 'Utama'.

2. *Form* 'Pengaturan'.
3. *Form* 'Input Gambar'.
4. *Form* 'Permainan'.
5. *Form* 'High score'.
6. *Form* 'Mengenal'.

Form Main

Form 'Main' merupakan *form* awal (pembuka) dari perangkat lunak yang berfungsi untuk menyediakan fasilitas-fasilitas yang terdapat dalam perangkat lunak. Rancangan tampilan dari *form* 'Main' ini dapat dilihat pada Gambar 3.



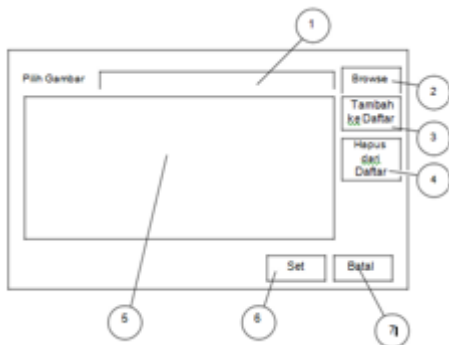
Gambar 3. Rancangan *Form* Main

Keterangan:

1. Link 'Pilih Gambar', berfungsi untuk menampilkan *form* 'Pilih Gambar'.
2. Link 'Permainan', berfungsi untuk menampilkan *form* 'Permainan'.
3. Link 'Pengaturan', berfungsi untuk menampilkan *form* 'Pengaturan'.
4. Link 'High score', berfungsi untuk menampilkan *form* 'High score'.
5. Link 'Profil', berfungsi untuk menampilkan *form* 'About'.
6. Link 'Help', berfungsi untuk menampilkan *form* 'Help'.
7. Link 'Keluar', berfungsi untuk menutup perangkat lunak

Form Pilih Gambar

Form ‘Pilih Gambar’ berfungsi sebagai tempat pemilihan gambar yang akan digunakan dalam permainan. Rancangan tampilan dari form ‘Pilih Gambar’ ini dapat dilihat pada Gambar 4.



Gambar 4. Rancangan Form Pilih Gambar

Keterangan:

1. Textbox ‘Pilih Gambar’, berfungsi untuk menampilkan lokasi gambar yang diinginkan.
2. Tombol ‘Browse’, berfungsi untuk membuka kotak dialog open untuk pemilihan gambar.
3. Tombol ‘Tambah ke Daftar’, berfungsi untuk menambahkan gambar yang dipilih ke daftar gambar.
4. Tombol ‘Hapus dari Daftar’, berfungsi untuk membuang gambar yang dipilih dari daftar gambar.
5. Listbox ‘Daftar Gambar’, berfungsi untuk menampilkan daftar gambar yang akan digunakan dalam permainan.
6. Tombol ‘Set’, berfungsi untuk menyimpan data gambar ke memori. Proses juga akan menutup form dan kembali ke form ‘Main’.
7. Tombol ‘Batal’, berfungsi untuk menutup form dan kembali ke form ‘Main’.
8. Data input tidak disimpan ke memori.

Perancangan Database

Database yang digunakan akan dirancang dengan menggunakan aplikasi *Microsoft Access* 2007. Tabel yang dirancang dapat dirincikan sebagai berikut:

1. Tabel *HighScore* yang digunakan untuk menyimpan data 10 nilai tertinggi dari *user*. Tabel 1 menunjukkan struktur tabel *HighScore* pada *database*.

Tabel 1 Tabel HighScore

NamaUser	Text
Nilai	Number
Tingkat	Number

2. Tabel *DaftarGambar* yang digunakan untuk menyimpan data gambar yang akan digunakan. Tabel 2 menunjukkan struktur tabel *DaftarGambar* pada *database*.

Tabel 2 Tabel DaftarGambar

Field Name	Data
Lokasi	Text

3. Tabel *DataTerakhir* yang digunakan untuk menyimpan data permainan terakhir dari setiap *user*. Tabel 3 menunjukkan struktur tabel *DataTerakhir* pada *database*.

Tabel 3 Tabel DataTerakhir

Field Name	Data Type
NamaUser	Text
Tingkatan	Number
Nilai	Number
Kesulitan	Text

4. Tabel *UserList* yang digunakan untuk menyimpan data *user* yang terdaftar dalam sistem. Tabel 4 menunjukkan struktur tabel *UserList* pada *database*.

Tabel 4 Tabel *UserList*

Field Name	Data Type
NamaUser	Text
Keterangan	Text

IMPLEMENTASI HASIL

Untuk menjalankan aplikasi metode *Forward chaining* untuk mengelompokkan objek berdekatan pada *Same game*, maka dapat mengklik file 'SameGame.exe' sehingga sistem akan menampilkan tampilan awal berikut:



Gambar 5. Tampilan Awal

Untuk bermain *Same game*, maka pertama kali harus memilih nama *user* terlebih dahulu dengan mengklik link 'Nama User', sehingga sistem akan menampilkan *form* Nama *User* berikut.



Gambar 6. Tampilan Pilih *User*

Setelah itu, pemain dapat memilih tingkat kesulitan dari permainan, yaitu dengan mengklik link 'Pengaturan', sehingga sistem akan menampilkan *form* Pengaturan berikut:



Gambar 7. Tampilan Pengaturan

Apabila pemain memilih tingkat 'Mudah', maka akan disediakan 80 detik untuk menyelesaikan *Same game*. Sedangkan, apabila pemain memilih tingkat 'Sedang' maka akan disediakan 60 detik untuk menyelesaikan *Same game*. Terakhir, apabila pemain memilih tingkat 'Susah' maka akan disediakan 45 detik untuk menyelesaikan *Same game*.

Setelah melakukan pemilihan nama *user*, dan pengaturan tingkat kesulitan maka pemain dapat memainkan *Same game*, dengan mengklik link 'Permainan' sehingga sistem akan menampilkan *form* berikut.



Gambar 8. Tampilan Permainan pada Saat Memulai Permainan

Pemain dapat memainkan permainan dengan melakukan *double click* pada gambar yang diinginkan. Apabila terdapat dua atau lebih gambar yang sama maka aplikasi akan membuang gambar yang sama tersebut dan menyusun ulang posisi gambar. Sementara itu, apabila pemain ingin mengetahui jumlah gambar sama yang dapat dibuang, maka pemain dapat melakukan click kiri pada gambar yang diinginkan. Apabila sampai batasan waktu yang diberikan, pemain masih belum mampu mencapai nilai yang ditentukan, maka permainan akan berakhir. Apabila pemain berhasil mencapai batasan nilai yang ditentukan sebelum waktu berakhir, maka pemain akan melanjutkan ke *level* berikutnya. Apabila permainan berakhir ataupun pemakai mengakhiri permainan, maka sistem akan menampilkan *form High score*

KESIMPULAN

Setelah selesai mengkonstruksi perangkat lunak, penulis dapat mengambil beberapa kesimpulan berikut:

1. Aplikasi menyediakan sebuah interface untuk bermain *Same game* pada sebuah komputer.
2. Metode *Forward chaining* dapat digunakan untuk menentukan objek yang sama pada *Same game*.
3. Perangkat lunak permainan *Same game* dapat dibuat dengan menggunakan bahasa pemrograman *Microsoft visual basic.NET 2010*, namun animasi yang dihasilkan masih kurang bagus.

DAFTAR PUSTAKA

Björner, T., & Hansen, C. B. S. (2010). Designing an Educational Game: Design Principles from a Holistic Perspective. *International Journal of Learning*, 17(10), 279-290.

Desiani, A., & Arhami. M. (2008). *Konsep Kecerdasan Buatan*, Yogyakarta: Penerbit Graha Ilmu.

Donald, C. (2006). *Game and e-learning*. Sunderland: Caspian

Hariyanto, B. (2004). *Rekayasa Sistem Berorientasi Objek*, Cetakan Pertama, Bandung: Penerbit Informatika.

Hartono, J. (2004). *Pengenalan Komputer*. Yogyakarta: Andi.

Junindar. (2008). *Panduan Lengkap Menjadi Programmer Membuat Aplikasi Penjualan Menggunakan VB.Net*. Cetakan ke-3. Jakarta: Media Kita.

Kusumadewi. S. (2002,) *Artificial Intelligence (Teknik dan Aplikasinya)*, Edisi 2, Yogyakarta: Penerbit Graha Ilmu

Salisah, F. N., Lidya, L., & Defit, S. (2015). Sistem Pakar Penentuan Bakat Anak dengan Menggunakan Metode Forward Chaining. *Jurnal Rekayasa dan Manajemen Sistem Informasi*, 1(1), 62-66.

Sandi, S. (2005). *Artificial Intelegencia*, Yogyakarta: Andi.

Supardi, Y. (2008). *Microsoft visual basic.NET untuk Segala Tingkat*, Jakarta: PT. Elex Media Komputindo.

Vitianingsih, A. V. (2016). Game Edukasi sebagai Media Pembelajaran Pendidikan Anak Usia Dini. *Jurnal INFORM*, 1(1), 1-8.