

Rancang Bangun “Chomical”: *Game Tower Defense* dengan Menerapkan Eksperimen Kimia Menggunakan Kerangka Kerja libGDX

Annisa U Istighotsah, Imam Kuswardayan, dan Dwi Sunaryono
Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Teknik Kimia, Surabaya 60111 Indonesia
E-mail: imam@its.ac.id

Abstrak—*Game* simulasi merupakan *game* yang menerapkan suatu aktivitas nyata ke dalam permainan komputer. Perangkat lunak pada tulisan ini mengaplikasikan konsep simulasi sebagai media pembelajaran untuk bidang kimia. Sasaran pengguna adalah siswa sekolah menengah atas. Aturan permainannya yaitu pemain harus melakukan eksperimen kimia untuk membuat peluru. Peluru tersebut digunakan untuk menyerang musuh pada pertempuran yang merupakan *tower defense game*. Dengan mengalahkan musuh, pemain dapat menemukan zat kimia baru untuk eksperimen selanjutnya. Pengembangan dilakukan dengan kerangka kerja libGDX. *Game* dikembangkan dengan bahasa pemrograman Java dan implementasi kode dilakukan pada perangkat lunak lingkungan pengembangan Eclipse. Uji coba dilakukan untuk mengetahui apakah perangkat lunak dapat berjalan sesuai yang diharapkan. Melalui pengujian fungsional, didapatkan hasil yang menunjukkan bahwa aplikasi dapat berjalan dengan baik. Pengujian kepada responden menunjukkan bahwa *game* diterima dengan baik dan membantu pemain dalam mempelajari berbagai percobaan kimia.

Kata Kunci—Eksperimen Kimia, libGDX, Simulasi, *Tower Defense*.

I. PENDAHULUAN

PERKEMBANGAN teknologi yang begitu pesat saat ini membuat keberadaan *game* bukanlah hal yang asing lagi.

Game memiliki banyak jenis dan dapat dimainkan melalui berbagai perangkat mulai dari telepon seluler hingga komputer. *Game* merupakan sarana bagi masyarakat untuk hiburan dan media untuk melepas penat setelah beraktifitas.

Kesulitan yang muncul ketika merancang sebuah *game* edukasi yaitu dalam memadukan aspek menyenangkan dengan konten edukatif yang hendak disampaikan. Dalam *game* edukasi, selain mementingkan unsur pendidikannya, unsur menyenangkan tetaplah harus diperhatikan karena tujuan dasar dari *game* adalah untuk permainan dan hiburan. *Game* edukatif yang mengangkat tema kimia jumlahnya yang sedikit yang beredar di internet atau toko aplikasi seperti Google Play atau App Store. *Game-game* yang ditemukan sebagian besar berjenis kuis tanya jawab ataupun teka-teki berisi soal-soal kimia. Aturan main pada *game-game* tersebut kebanyakan terbatas pada bagaimana pemain harus menyelesaikan soal-soal yang diberikan.

Oleh karena itu, perancangan perangkat lunak pada artikel ini mencoba menerapkan penggabungan tipe *game* simulasi dan *tower defense* yang belum pernah dilakukan sebelumnya.

Percobaan kimia merupakan aspek kehidupan nyata yang dapat disimulasikan dalam permainan komputer sebagai bagian edukatif permainan, sementara bagian *tower defense* berfungsi sebagai motivator dan aspek menyenangkan dalam *game*. Dengan menggabungkan kedua jenis *game* tersebut, diharapkan akan terbentuk suatu *game* edukatif mengenai simulasi percobaan kimia yang memiliki aspek *action* yang menyenangkan untuk dimainkan.

II. TAHAP KAJIAN PUSTAKA

A. Permainan Simulasi

Game simulasi merupakan *game* yang berusaha meniru kejadian atau proses yang terjadi pada kehidupan nyata. Genre simulasi ini memiliki beberapa sub-genre antara lain simulasi kehidupan, simulasi manajemen, simulasi kegiatan olahraga dan tipe simulasi lainnya.

Genre *game* simulasi berfungsi untuk memberi pengalaman bermain yang semirip mungkin dengan kenyataan. Salah satu contoh *game* simulasi yaitu Trauma Center [1], sebuah *game* pada platform Nintendo yang bertema kedokteran. Pada Trauma Center, pemain bertugas untuk melakukan operasi pada pasien dengan peralatan dan instruksi yang telah disediakan.

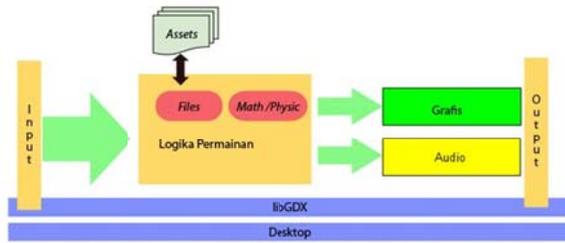
B. Permainan Tower Defense

Tower defense merupakan sub-genre dari kategori *game* strategi. Tujuan dari *game* adalah untuk mencegah musuh memasuki suatu tempat yang dilindungi oleh pemain dengan cara menyerang dan menahan musuh untuk memasuki tempat yang dilindungi tersebut. Cara menahan musuh bervariasi mulai dari menembak, memasang jebakan dan memperlambat pergerakan musuh.

Setiap kali pemain berhasil mengalahkan musuh, pemain akan mendapatkan semacam poin atau uang. Poin tersebut dapat dipakai untuk menambah objek tertentu untuk melawan musuh dengan level yang lebih tinggi. Objek-objek *game* tersebut dapat berupa senjata baru atau kekuatan yang lebih tinggi dan dapat digunakan seiring dengan berlangsungnya permainan.

C. Kombinasi Desain Game

Dengan kompleksitas *game* yang semakin berkembang akhir-akhir ini, menggunakan satu skenario dapat menghasil-



Gambar 1. Diagram Modul Kerangka Kerja libGDX

kan sekumpulan aturan *game* yang rumit dan susah dimengerti. Oleh karena itu, model desain pada [2] menawarkan solusi reduksi kompleksitas dengan menyusun sebuah konsep desain *game* yang terdiri dari gabungan desain *game-game* yang lebih sederhana. Kombinasi antar dua *game* atau lebih berakibat dengan munculnya tren desain *game* misalnya *mini-game* dan *bonus stage*.

Bagian dari *game* yang dapat dikombinasikan antar lain aturan main *game* (*gameplay*). Kombinasi aturan main menggabungkan aturan-aturan pada *game* sehingga tersusun *game* dengan pengalaman bermain yang lebih luas. Ada tiga macam pendekatan kombinasi yang dilakukan menurut [2] dalam penggabungan model aturan main, yaitu kombinasi antar hadiah (*reward*) dengan kegigihan (*persistence*), kombinasi cerita *game* dengan pembekalan pengetahuan serta kombinasi sosialisasi dengan umpan balik.

D. libGDX

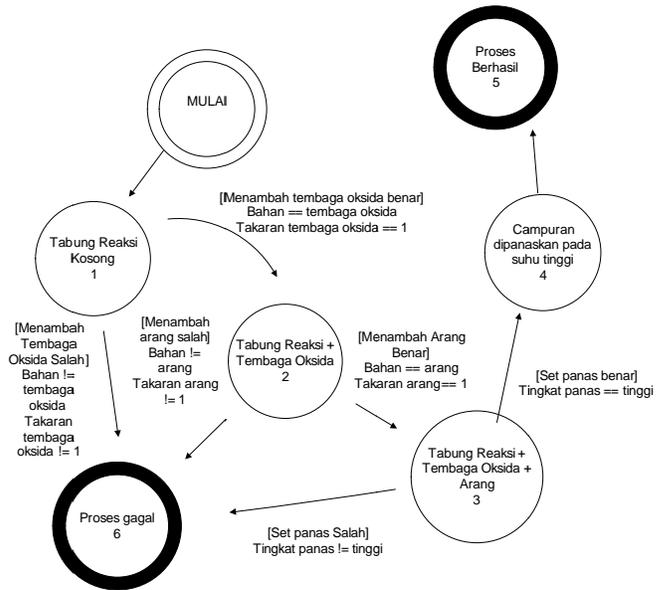
Kerangka kerja libGDX adalah sebuah kerangka kerja pengembangan *game* 2D dan 3D berbasis Java yang menyediakan fitur-fitur yang dibutuhkan dalam pengembangan *game*. *Game-game* yang dikembangkan dengan libGDX dapat ditujukan secara spesifik di berbagai platform yang telah didukung oleh libGDX. Platform-platform tersebut antara lain Windows, Android versi 1.5 ke atas, Mac OS X, iOS (membutuhkan lisensi iOS), Linux, dan peramban dengan JavaScript seperti Chrome, Safari, Opera dan Firefox. Secara umum, modul libGDX pada suatu arsitektur *game* dapat digambarkan seperti pada Gambar 1.

III. ANALISIS DAN PERANCANGAN

A. Rancangan Antarmuka

Antarmuka permainan pada *game* secara umum dibagi menjadi lima layar utama sebagai berikut.

1. Layar utama permainan yaitu layar yang pertama kali muncul ketika memulai permainan.
2. Layar peta yang berisi ikon level untuk memilih level pertempuran yang diinginkan dan ikon lab untuk navigasi menuju halaman menu lab.
3. Layar pertempuran.



Gambar 2. Diagram Finite State Machine Proses Ekstraksi Tembaga

4. Layar menu laboratorium dimana pemain dapat memilih tipe eksperimen yang hendak dilakukan.
5. Layar laboratorium dimana pemain melakukan eksperimen kimia.

B. Aturan Simulasi

Pada bagian simulasi, pemain harus memperhatikan hal-hal berikut ketika melakukan eksperimen.

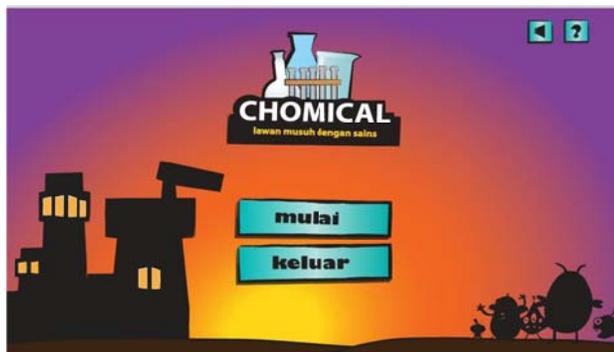
1. Takaran zat.
2. Urutan memasukkan.
3. Aturan memanaskan.
4. Pengadukan

Penalti yang didapat ketika tidak memperhatikan aturan tersebut yaitu gagalnya percobaan. Jika hal ini terjadi, *game* akan mengembalikan posisi pemain ke awal percobaan. Gambar 2 menunjukkan diagram alur proses ekstraksi tembaga pada *game*.

C. Perancangan Data

Penanganan data diaplikasikan pada pembuatan "Chomical" untuk menyimpan data-data yang dibutuhkan dalam permainan. Data disimpan pada *file* berekstensi .json dan dipanggil menggunakan penerjemah (*parser*) yang ada pada libGDX.

Data yang disusun antara lain data-data zat kimia. Tiap entitas data kimia memiliki atribut berupa nomor unik identitas, nama zat kimia, jumlah yang dimiliki pemain dan status *unlock/locked*-nya. Data lain yang disimpan pada .json yaitu data pemain. Data ini menyimpan kondisi pemain misalnya jumlah koin yang dimiliki pemain dan data nilai yang telah diperoleh.



Gambar 3. Tampilan Layar Utama

IV. IMPLEMENTASI

A. Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan menggunakan modul scene2D pada libGDX. Modul ini dipakai untuk implementasi objek-objek *game* seperti tombol, dialog, dan tabel. Hasil implementasi layar utama dapat dilihat pada Gambar 3.

B. Implementasi Simulasi

Implementasi simulasi mengacu pada diagram alur pada Gambar 2. Pada gambar tersebut terdapat tiga tahap utama dalam melakukan eksperimen ekstraksi tembaga yaitu menambahkan tembaga oksida, menambahkan arang dan memanaskan campuran dengan suhu tinggi.

Gambar 4 merupakan implementasi kode semu (*pseudocode*) untuk tahap memasukkan tembaga oksida. Pada tahap ini, *game* akan mengecek apakah bahan dan jumlah bahan yang dimasukkan pemain sudah benar. Pengecekan ini juga diimplementasikan dengan kode semu pada Gambar 5 mengenai tahap memasukkan arang.

Tahap pemanasan diimplementasikan dengan kode semu pada Gambar 6. Pada tahap ini, *game* mengecek apakah pemain mengeset tingkat pemanasan yang benar.

C. Implementasi Data

Data-data bahan kimia yang dipakai pada *game* disimpan pada file kimia.json. Implementasi data bahan kimia ke JSON dapat dilihat pada kode semu pada Gambar 7.

V. UJI COBA

Perangkat lunak pada artikel ini melalui proses pengujian yang dilakukan dengan dua skenario, yaitu pengujian fungsional dan pengujian terhadap pengguna.

Pengujian fungsional dilakukan dengan metode kotak hitam (*black box*). Hasil rekapitulasi pada pengujian ini terdapat pada Tabel 1. Aspek-aspek yang diuji pada pengujian ini meliputi uji coba kebenaran fungsi aturan main *game*.

```

1 Function MenambahTembagaOksida
2 (Input indeksBahan, Input JumlahBahan){
3 State □ State_tabung_reaksi_kosong
4 If indeksBahan == indexTembagaOksida and jumlahBahan == 1
5     State <- State_Tembaga_Oksida_Ditambah
6 Else
7     State <- State_Proses_Gagal
8 }

```

Gambar 4. Kode Semu Tahap Memasukkan Tembaga Oksida

```

1 Function MenambahArang
2 (Input indeksBahan, Input JumlahBahan){
3 If indeksBahan == indeksBahanArang and jumlahBahan == 1
4     State <- State_Arang_Ditambah
5 Else
6     State <- State_Proses_Gagal
7
8 }

```

Gambar 5. Kode Semu Tahap Memasukkan Arang

```

1 Function SetPanas(Input pilihanPanas){
2 If(pilihanPanas == inputPanasTinggi)
3     State <- State_Campuran_Dipanasikan
4 else
5     State <- State_Proses_Gagal
6 }

```

Gambar 6. Kode Semu Tahap Memanaskan Campuran

Hasil rekapitulasi pada pengujian untuk fungsional kontrol dan antarmuka dapat dilihat pada Tabel 2. Pada uji coba kontrol yang diuji adalah kebenaran antarmuka dan tombol-tombol pada *game* dalam menjalankan fungsi sesuai yang ditugaskan.

Pengujian pengguna dilakukan dengan memberikan tes awal dan tes akhir kepada responden yang merupakan siswa SMA. Tabel 3 berisi rekapitulasi tes awal dan tes akhir dari responden.

Selain itu responden juga diberi kuisioner untuk menilai beberapa aspek permainan misalnya tampilan antarmuka dan skenario aturan permainan. Hasil rekapitulasi pada pengujian pengguna dengan kuisioner dapat dilihat pada Tabel 4.

VI. KESIMPULAN

Melalui proses yang dijalankan dalam pengembangan aplikasi, dapat ditarik kesimpulan sebagai berikut.

1. *Game* dapat menerapkan aturan permainan dengan mengkombinasikan simulasi dan *tower defense*. Pada *game* ini, pemain mengalahkan musuh pada *tower defense game* dan membuat senjata pada simulasi.
2. *Game* mampu mengimplementasikan simulasi percobaan kimia sesuai dengan proses pada referensi.
3. libGDX dapat diaplikasikan dalam pembangunan *game* yaitu dalam hal pengembangan antarmuka permainan menggunakan scene2D, pengelolaan aset, manajemen file dan kontrol masukan.

```

1 {
2   nama: "aluminium sulfat",
3   tipe: bahan,
4   info: "aluminium sulfat info",
5   jumlah: 4,
6   status: locked,
7   harga: 10,
8   id: 1
9 }
10

```

Gambar 7. Implementasi Data Kimia Pada JSON

Tabel 1.
Rekapitulasi Pengujian Aturan Main

No	Kode Pengujian	Nama Pengujian	Hasil Pengujian
1	CHOM-UC-01	Pengujian penemuan bahan kimia level 1	Berhasil
2	CHOM-UC-02	Pengujian penemuan bahan kimia level 2	Berhasil
3	CHOM-UC-03	Pengujian penemuan bahan kimia level 3	Berhasil
4	CHOM-UC-04	Pengujian penemuan bahan kimia level 4	Berhasil
5	CHOM-UC-05	Pengujian naik level	Berhasil
6	CHOM-UC-06	Pengujian kondisi kalah	Berhasil
7	CHOM-UC-07	Pengujian kondisi menang	Berhasil

- Pada evaluasi pengujian kegunaan melalui kuisioner, responden rata-rata memberikan nilai 81,75% untuk antarmuka permainan, 82,25% untuk aturan main, 88,50% untuk simulasi dan 93,75% untuk pengetahuan pada *game*. Rata-rata penilaian keseluruhan adalah sebesar 86,56%. Dengan nilai tersebut dapat disimpulkan bahwa responden memberikan penilaian cukup baik untuk *game*.
- Evaluasi pengujian menggunakan tes awal dan tes akhir menunjukkan *game* "Chomical" dapat meningkatkan pemahaman pemain dalam menyelesaikan soal kimia sebesar 42% dari nilai sempurna (100%).

Saran untuk pengembangan lebih lanjut antara lain yaitu penambahan efek visual yang lebih baik sehingga *game* lebih menarik untuk pemain. Saran lainnya yaitu untuk menambah jumlah simulasi sehingga pemain memiliki pilihan lebih bervariasi.

Tabel 2.
Rekapitulasi Pengujian Kontrol dan Antarmuka

No	Kode Pengujian	Nama Pengujian	Hasil Pengujian
1	CHOM-UI-01	Pengujian navigasi antarmuka	Berhasil
2	CHOM-UI-02	Pengujian kontrol pertempuran	Berhasil
3	CHOM-UI-03	Pengujian kontrol simulasi	Berhasil

Tabel 4.
Rekapitulasi Pengujian Kuisioner

No.	Aspek Pengujian	Nilai Akhir	Persentase
1	Antarmuka permainan	3,27	81,75%
2	Aturan main	3,29	82,25%
3	Simulasi	3,54	88,50%
4	Penyampaian pengetahuan	3,75	93,75%
Rata-rata		3,46	86,56%

Tabel 3.
Rekapitulasi Pengujian Tes Awal dan Tes Akhir

Responden	Nilai Tes Awal	Nilai Tes Akhir
Responden 1	40	93
Responden 2	26	33
Responden 3	20	73
Responden 4	60	86
Responden 5	20	80
Responden 6	26	80
Rata-rata	32	74

UCAPAN TERIMA KASIH

Penulis A.I. mengucapkan terima kasih kepada kru dan admin Laboratorium Pengembangan Permainan Teknik Informatika ITS atas dukungannya selama pengembangan perangkat lunak pada artikel ini dilaksanakan.

DAFTAR PUSTAKA

- "Official Trauma Center," 2008. [Online]. Available: <http://www.traumacenter.org/>. [Diakses 4 Mei 2013].
- T. Zarranandia, "Designing Educational Games by Combining Other Game Designs," in *2012 IEEE 12th International Conference on Advanced Learning Technologies (ICALT)*, Leganés, 2012.
- K. Hutchings, *Classic Chemistry Experiments*, C. Osborne and J. Johnston, Eds., London: The Royal Society of Chemistry, 2000.
- M. Jovanovic, "Surviving the Design of Educational Games: Borrowing from Motivation and Multimodal," in *Conference on Human System Interactions 2008*, Belgrade, 2008.
- "libGDX Documentation," [Online]. Available: <http://code.google.com/p/libgdx/wiki/>. [Diakses 12 Mei 2013].