

Platform Gamifikasi Untuk Perkuliahan

David Kristiadi*¹, Khabib Mustofa²

¹STMM "MMTC" Yogyakarta; Jl Magelang km. 6 Yogyakarta, Indonesia

²Departemen Ilmu Komputer dan Elektronika, FMIPA, UGM, Yogyakarta, Indonesia
e-mail: *¹dv.davk@gmail.com, ²khabib.mustofa@ugm.ac.id

Abstrak

Desain gamifikasi di dalam perkuliahan memiliki banyak variasi. Platform yang fleksibel dibutuhkan untuk mengakomodasi hal tersebut. Penelitian ini bertujuan untuk mengembangkan platform gamifikasi untuk perkuliahan yang fleksibel, memiliki kinerja yang baik dan dapat diterima oleh pengguna.

Platform dikembangkan dengan menggunakan konsep Generic Gamification Platform (GGP). GGP merupakan jenis solusi gamifikasi yang mengaplikasikan prinsip-prinsip Service Oriented Architecture (SOA) dan meletakkan komponen gamifikasi (data, logic dan rewards) terpisah dari Sistem Informasi (SI). Platform dikembangkan dengan kemampuan untuk mengelola game mechanics (mekanika permainan), aksi, task, dan aturan permainan. Selain itu platform juga mampu membangkitkan aturan permainan secara otomatis, dan diintegrasikan dengan SI.

Hasil ujicoba menunjukkan bahwa platform gamifikasi untuk perkuliahan dapat dikembangkan. Platform memiliki tingkat fleksibilitas yang baik, memiliki kinerja yang baik, dan dapat diterima oleh pengguna (5 dosen dan 2 bukan dosen tetapi memahami aktivitas perkuliahan dengan baik). Tingkat fleksibilitas platform sebesar 85%. Waktu respon platform dalam mengeksekusi event kurang dari 336ms. Platform memiliki skor rata-rata System Usability Scale (SUS) sebesar 60.0 dan berada pada range batas bawah yang dapat diterima oleh pengguna.

Kata kunci— platform gamifikasi, gamifikasi untuk perkuliahan, GGP, Generic Gamification Platform

Abstract

Gamification in lecturing has a lot of variety designs. A flexible platform is needed for that matter. This research aims to develop a gamification platform for lecturing that flexible, has a good performance and acceptable by users. Generic Gamification Platform (GGP) concept is used to develop the platform. GGP is a kind of gamification solution that applies service oriented architecture Architecture (SOA) principles and puts gamification components (data, logic and rewards) and Information System (IS) separately. The platform has some capabilities such as able to manage game mechanics, actions, tasks and rules. The other platform's capabilities are able to auto generate rules and to be integrated to IS. The results of tests show that a gamification platform for lecturing can be developed. The platform has a good level of flexibility, has a good performance, and acceptable by users (5 lecturers and 2 non-lecturers but well knowing on lecturing activities). Its flexibility level is 85%. Its average of response time on event execution is lower than 336ms. Its System Usability Scale (SUS) average score is 60 and its acceptability range in low marginal.

Keywords— gamification platform, gamification for lecturing, GGP, generic gamification platform

1. PENDAHULUAN

Gamifikasi adalah penggunaan unsur-unsur desain *game* (permainan) pada konteks non-permainan [1]. Gamifikasi memberikan manfaat yang positif dalam bidang pendidikan yaitu peningkatan motivasi dan keterlibatan (peran aktif) peserta dalam aktivitas pembelajaran yang digamifikasi [2]. Manfaat tersebut diiringi dengan meningkatnya perolehan nilai siswa [3]. Manfaat positif tersebut menjadi daya tarik untuk menjadikan gamifikasi menjadi salah satu pilihan dalam mengelola aktivitas kelas/perkuliahannya.

Penerapan gamifikasi membutuhkan perencanaan yang cermat, menyeluruh dan fleksibel [4]. Hal tersebut berkaitan erat dengan aplikasi yang digunakan untuk implementasi gamifikasi (solusi gamifikasi). Di bidang pendidikan khususnya perkuliahan, terdapat beberapa solusi gamifikasi diantaranya sebagai *plugin* di dalam *Learning Management System* (LMS), aplikasi kerjasama dengan pihak ketiga dan aplikasi gamifikasi yang dikembangkan sendiri [5]. Walaupun demikian kadang pengguna merasa solusi gamifikasi yang ada kurang fleksibel dan dibatasi oleh lingkungan solusi gamifikasi [5].

Salah satu bentuk solusi gamifikasi berupa platform gamifikasi. Platform gamifikasi merupakan aplikasi yang menyediakan *Application Program Interface* (API) dan *Graphic User Interface* (GUI) untuk mengimplementasi desain gamifikasi. Salah satu konsep yang digunakan untuk membangun platform adalah *Generic Gamification Platform* (GGP). Konsep ini memisahkan komponen gamifikasi dari Sistem Informasi dan menerapkan prinsip-prinsip *Service Oriented Architecture* (SOA).

Fleksibilitas menjadi salah satu kebutuhan dan prinsip dalam pengembangan platform gamifikasi. Fleksibilitas platform tidak hanya berkaitan dengan integrasi terhadap aplikasi lain, namun juga pada kemampuannya untuk mengakomodasi desain gamifikasi yang berbeda-beda tanpa melakukan perubahan di dalam komponen platform. Fleksibilitas tersebut juga harus didukung oleh kinerja platform yang baik dan dapat diterima oleh pengguna. Sebab fleksibilitas platform yang baik tidak akan berguna tanpa disertai kinerja yang baik atau tidak dapat diterima oleh pengguna. Kebutuhan akan platform gamifikasi untuk perkuliahan yang fleksibel dengan kinerja yang baik serta dapat diterima oleh pengguna menuntun kepada penelitian untuk mengembangkan platform tersebut.

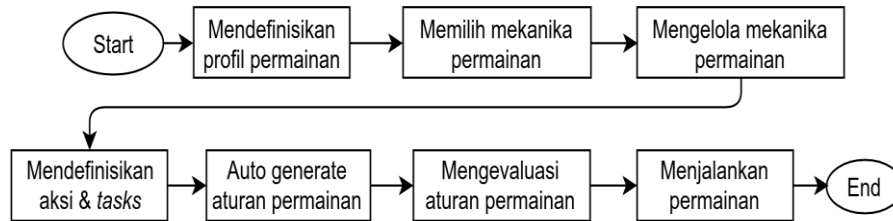
Beberapa penelitian tentang solusi gamifikasi telah dilakukan. Penelitian tersebut berkaitan dengan upaya membangun platform yang fleksibel. Platform yang fleksibel tersebut mudah dimodifikasi untuk digunakan di dalam aplikasi atau lingkungan lain selain dari yang telah dirancang [6]. Upaya yang dilakukan adalah dengan membangun platform gamifikasi yang berorientasi servis [8, 9], menggunakan prinsip *event-driven* [8], adaptif [9], dapat dikonfigurasi dan mendukung mekanika permainan yang bermacam-macam [5]. Namun dari publikasi tersebut belum ada yang secara khusus membahas tentang fleksibilitas platform dalam mengakomodasi desain gamifikasi yang berbeda-beda.

2. METODE PENELITIAN

Studi pustaka terhadap literatur-literatur yang berkaitan dengan desain gamifikasi di perkuliahan dan desain platform gamifikasi mengawali proses penelitian. Dari desain-desain tersebut kemudian dilakukan analisa terhadap mekanika permainan yang digunakan, aturan-aturan yang digunakan, serta ringkasan umum desain yang dilakukan. Mekanika permainan adalah metode dan alat untuk merepresentasikan data permainan. Beberapa diantaranya adalah poin, *badge*, *leaderboard*, *level* dsb. Banyaknya penggunaan mekanika permainan tidak menjamin keberhasilan penerapan gamifikasi tetapi pada penggunaannya secara efektif [5]. Oleh karenanya platform yang dibangun harus membebaskan *game master* (dosen) untuk memilih mekanika permainan yang akan digunakan, dan hanya poin yang wajib digunakan [10]. Mekanika permainan yang didukung oleh platform adalah poin, *badge*, papan skor, *level*, *progression*, *avatar*, status, notifikasi dan *activity feed*.

2.1 Tahapan desain gamifikasi

Tahapan desain gamifikasi menggunakan platform dirancang seperti pada Gambar 1. Tahap mendefinisikan permainan berkaitan dengan pengisian data profil matakuliah yang digamifikasi seperti nama mata kuliah, tanggal mulai dan berakhirnya kuliah.



Pada tahap memilih mekanika permainan, *game master* memilih mekanika permainan yang akan digunakan. Mekanika permainan yang telah dipilih akan diaktifkan dan dapat didefinisikan lebih detail di tahap mengkonfigurasi mekanika permainan. Tahap mendefinisikan aksi dan *tasks* dilakukan pendefinisian aksi dan kumpulan aksi yang akan dinilai di dalam permainan. Selanjutnya pada tahap auto-generate aturan permainan, aksi dan *task* yang telah didefinisikan oleh platform akan dipindai dan dimasukkan ke dalam *template* aturan permainan. Hasil dari proses ini adalah aturan permainan. Selanjutnya *game master* dapat mengevaluasi aturan permainan yaitu dengan menambah, mengurangi atau memodifikasi aturan. Setelah semua aturan permainan dapat berfungsi dengan baik permainan dapat dijalankan.

2.2 Perancangan aksi, task, event, eventTask dan aturan permainan

2.2.1 Aksi (a)

Aksi merupakan tindakan-tindakan yang dilakukan oleh pemain untuk memperoleh poin. Aksi memiliki beberapa data penyusun yaitu frekuensi, nilai poin, dan tipe poin. Frekuensi adalah jumlah kemunculan aksi yang diperbolehkan dalam sebuah permainan. Hal ini memungkinkan sebuah aksi yang sama dapat diulang pada sebuah permainan, misalnya pada aksi “kehadiran” pada Tabel 1 dapat diulang sebanyak 14 kali. Nilai poin adalah sejumlah angka yang diterima ketika mengerjakan aksi. Tipe poin menentukan tipe perhitungan poin yang akan diterima oleh pemain. Tipe poin tersebut adalah :

1. *Fixed*. Seluruh poin yang didefinisikan akan diperoleh hanya dengan mengerjakan aksi.
2. *Relative*. Poin yang akan diterima pemain ditentukan oleh perbandingan nilai aksi dengan nilai aksi maksimal dikalikan nilai poin, seperti pada persamaan (1)

$$\text{Poin yang diterima} = \frac{\text{nilai aksi}}{\text{nilai aksi maksimal}} \times \text{nilai poin} \quad (1)$$

Semisal pemain melakukan aksi “Kehadiran”, maka pemain mendapat poin 25. Ketika pemain melakukan aksi “*Final Project*” dan mendapatkan nilai 7, maka pemain mendapat poin $(7/10) \times 100 = 70$.

Tabel 1 Contoh pendefinisian aksi

No	Aksi	Tipe	Kemunculan	Poin	Nilai Max
1	Kehadiran	<i>Fixed</i>	14	20	-
2	Tugas	Relatif	2	100	10
3	Kuis	Relatif	1	200	10
4	Ujian	Relatif	1	400	10
5	Presentasi	Relatif	1	200	10

2.2.2 Task (t)

Task atau sering disebut *mission* atau *challenge* merupakan kumpulan dari beberapa aksi yang harus dikerjakan pemain untuk mendapatkan bonus poin atau *reward (badge)*.

Kumpulan aksi terdiri dari nama aksi dengan jumlah minimal (*count*) yang harus sudah dikerjakan pada posisi aksi (*at*) tertentu. Sebagai contoh *task* “*find your sword*” pada Tabel 2, pemain dianggap mengerjakan *task* tersebut ketika pada Aksi Kehadiran pada posisi (*at*) yang ke 4, pemain telah mengerjakan minimal sebanyak (*count*) 2 dan pada Aksi Kuis pada posisi (*at*) yang ke 1, pemain telah mengerjakannya minimal sebanyak (*count*) 0. Ketika kondisi tersebut dipenuhi, maka pemain mendapatkan badge “*Sword Badge*” dan bonus poin 40.

Tabel 2 Contoh pendefinisian *tasks*

<i>Tasks</i>	<i>Aksi</i>	<i>Bonus Poin</i>	<i>Badges</i>
<i>Find your sword</i>	Kehadiran (<i>c</i> :2, <i>at</i> : 4), Kuis (<i>c</i> :0, <i>at</i> :1)	40	<i>Sword Badge</i>
<i>Facing The Boss</i>	Kehadiran (<i>c</i> :7, <i>at</i> :8), Tugas(<i>c</i> :0, <i>at</i> :1)	30	<i>Master Badge</i>

Keterangan: *c* = *count*/jumlah minimal aksi yang telah dikerjakan; *at* = posisi aksi terakhir yang dikerjakan

2.2.3 Event (*e*)

Platform yang dikembangkan juga menggunakan istilah *event* dan menggunakan asumsi bahwa segala yang berkaitan dengan gamifikasi dapat diwakili oleh *event* atau *user action* [8]. *Event* (*e*) didefinisikan sebagai sebuah peristiwa dimana sebuah aksi (*a*) dikerjakan oleh pemain (*p*) di dalam suatu permainan (*g*) jika dinotasikan seperti pada persamaan 2.

$$e = (g, p, m, a) \quad (2)$$

dengan $a = \langle \text{code}, \text{at}, \text{value} \rangle$

Pemain (*p*) dapat bertindak sebagai tim atau pemain tunggal, untuk membedakannya ditambahkan sebuah *flag* mode tim (*m*). Pemain sebagai tim ketika *m* bernilai *true* dan sebagai pemain tunggal ketika *m* bernilai *false*. Eksekusi *event* dengan nilai *m*=*true* akan menyebabkan perubahan poin bagi semua anggota tim. Sedangkan perubahan akibat eksekusi *event* dengan *m*=*false* hanya akan diterima oleh pemain sendiri. Di dalam program, *event* akan dinyatakan ke dalam sebuah objek seperti pada Gambar 2 (a).

```

5 define Event={
6   game : '', // game code
7   player : '', // username
8   teamMode:false, //teamMode true or false
9   action: {
10    at:'', // action position
11    code:'', // action code
12    value:'' // action value
13  }
14 /* e = { game:'', player:'',
15   teamMode:false,
16   action:{ at:0,code:'' ,
17   rev:'', value:'' }
18 */
19 construct:function(e){
20   this.action = e.action;
21   this.player = e.player;
22   this.game = e.game;
23   this.teammode = e.teamMode||false;
24 }
25 }

```

(a)

```

4 define EventTask{
5   game:'', //gameCode
6   player:'', // username
7   task:'', // task Code
8   // actions={
9   // 'actionName':{
10  // count:number, at:number},...}
11  actions:{},
12  /*et = { game:'gamecode',
13   player:'username',
14   task:'taskCode',
15   actions:{actions}} */
16  constructor:function(et){
17   this.game = et.game;
18   this.player = et.player;
19   this.task = et.task;
20   this.actions = et.actions;
21 }
22 }

```

(b)

Gambar 2 Potongan program
(a) Pendefinisian *event*. (b) Pendefinisian *eventTask*

Parameter *g* mewakili kode permainan, *p* mewakili *username* pemain atau nama tim, *m* mewakili *flag* mode tim dan *a* mewakili aksi yang dikerjakan pemain. Aksi (*a*) yang dikerjakan pemain merangkum beberapa data yaitu kode aksi (*code*), posisi aksi (*at*) atau urutan aksi yang sedang dikerjakan, dan nilai aksi (*value*) nilai yang diperoleh pemain (*true/false* untuk tipe aksi *fixed* dan angka untuk tipe aksi relatif).

2.2.4 EventTask (et)

EventTask merupakan peristiwa dimana pemain (p) dari suatu permainan (g) telah melakukan sejumlah aksi (a_n) yang dipersyaratkan oleh suatu *task* (t). Hasil dari eksekusi *eventTask*, pemain mendapatkan *rewards* sesuai yang telah didefinisikan di *task*. Jika dinotasikan akan seperti pada persamaan 3.

$$et = (g, p, t, a_n) \quad (3)$$

dengan $a_n = \{code_1: \langle count, at \rangle, code_2: \langle count, at \rangle, \dots, code_n: \langle count, at \rangle\}$

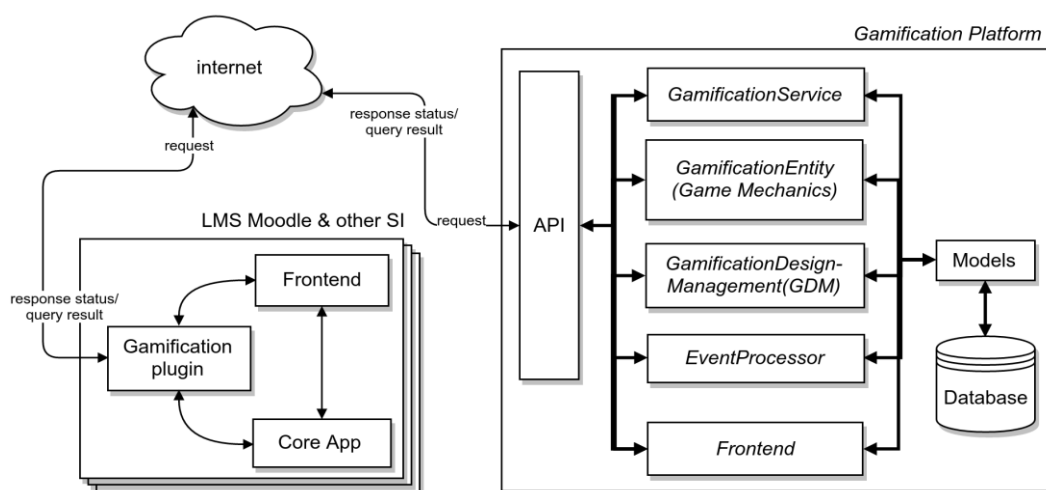
EventTask di dalam program dinyatakan dalam bentuk sebuah objek seperti pada gambar Gambar 2 (b). Parameter g mewakili kode permainan, p mewakili *username* pemain, t mewakili kode *task* dan parameter a_n merupakan rangkuman aksi yang telah dikerjakan pemain. Rangkuman aksi ini berisi daftar kode aksi (*code*) yang disertai dengan data cacah aksi yang telah dikerjakan (*count*) dan posisi aksi (*at*) yang terakhir dikerjakan.

2.2.5 Rule (aturan permainan)

Rule atau aturan permainan menggunakan prinsip aturan *if-then*. Representasi aturan permainan mengikuti format yang telah ditentukan di *Domain Specific Language* (DSL) dari *library* Nools (<https://github.com/C2FO/nools>). Aksi dan *task* menjadi dasar dalam pembuatan aturan permainan. Pada bagian premis, diambil beberapa properti dari entitas aksi dan *task* menjadi data pembanding seperti kode permainan, dan kode aksi/*task*. Pada bagian konklusi properti poin atau *reward* diambil untuk dijadikan sebagai masukan dalam proses memperbaharui poin dan *reward* pemain dalam permainan. Dengan prinsip ini aksi dan *task* yang telah didefinisikan, oleh sistem dapat di *auto-generate* menjadi aturan permainan dengan menyediakan *default template*.

2.3 Rancangan platform

Platform gamifikasi yang dikembangkan seperti pada Gambar 3. Rancangan ini dikembangkan dari solusi gamifikasi GGP [11]. GGP menerapkan prinsip-prinsip SOA dan menempatkan secara penuh komponen solusi gamifikasi (data dan logic) terpisah dari SI. Pada Gambar 3, *frontend* merupakan antarmuka bagi pengguna untuk mengakses komponen-komponen di dalam platform atau SI.



Gambar 3 Rancangan platform gamifikasi pengembangan dari konsep GGP [11]

Komponen *EventProcessor* mengerjakan fungsionalitas untuk menambah *event*. Fungsionalitas platform untuk pengelolaan gamifikasi, pengelolaan aksi dan *tasks*, pengelolaan tim dan pemain, pengelolaan aturan permainan dirangkum dalam komponen *GamificationDesign-Management(GDM)*. Komponen *GamificationEntity* merangkum fungsionalitas platform untuk pengelolaan entitas gamifikasi. Komponen *GamificationService* mewakili proses pengelolaan *runtime* permainan. Proses CRUD *database* dikerjakan oleh komponen *Model*. *Plugin* gamifikasi berfungsi menjembatani integrasi antara SI dengan platform. Melalui *plugin* ini, SI dapat melakukan *request* penambahan *event* atau *request* status dan profil pemain. API merupakan webservice yang mengkoordinasikan *request* dari SI kepada komponen-komponen platform yang lain, kemudian memberikan respon yang berupa hasil *query* atau status terhadap *request* yang diinputkan misalnya status HTTP 200 untuk operasi berhasil atau status HTTP 4xx untuk operasi gagal.

2.3.1 Integrasi Platform dengan SI

Integrasi platform dengan SI menggunakan contoh kasus LMS Moodle versi 2.6 dan 3.1. *Plugin* gamifikasi dibangun mengikuti arsitektur LMS, dan dibekali kemampuan berkomunikasi dengan platform melalui *RESTful web service*. Data yang dipertukarkan antara LMS dengan platform ketika permainan sedang berlangsung adalah *event*. Oleh karenanya, data penyusun *event* harus disinkronisasi terlebih dahulu. Data tersebut adalah data pemain, aksi dan permainan. Data pemain disinkronisasi dengan data siswa di LMS. Setiap aktivitas (kuis, ujian/*assignment*) di LMS disinkronisasi dengan jenis aksi dan posisi aksi (*at*). Data permainan disinkronisasi dengan data matakuliah/*course* di LMS. Di dalam LMS melalui *plugin* dibuat proses untuk *me-request* sinkronisasi data, *me-request* penambahan *event* dan proses mencatat hasil sinkronisasi.

2.4 Rancangan Pengujian

2.4.1 Uji Fleksibilitas

Fleksibilitas adalah salah satu kebutuhan non-fungsional platform. Fleksibilitas merupakan kemampuan platform untuk mengakomodasi desain gamifikasi yang berbeda-beda dengan melakukan modifikasi aturan permainan dan pemilihan mekanika permainan tanpa melakukan perubahan pada komponen internal platform. Sehingga platform dapat digunakan untuk matakuliah yang berbeda-beda [7], tanpa melakukan modifikasi komponen internal platform [11, 7]. Untuk mengukur kemampuan platform dalam mengakomodasi desain gamifikasi perkuliahan dirancang sebuah uji fleksibilitas. Di dalam uji fleksibilitas dilakukan implementasi desain gamifikasi yang digunakan oleh beberapa publikasi. Selama pengujian dilakukan pencatatan terhadap beberapa parameter berikut:

1. Mekanika permainan. Parameter ini berkaitan dengan mekanika permainan yang digunakan oleh desain tetapi tidak tersedia di dalam platform.
2. Integrasi LMS/SI. Parameter ini berkaitan dengan perlu atau tidak integrasi dengan LMS/SI melalui *plugin* gamifikasi yang telah dibuat.
3. Modifikasi aturan. Parameter ini berkaitan dengan perlu atau tidak dilakukan modifikasi aturan permainan.
4. Modifikasi *Plugin*. perlu tidaknya modifikasi dilakukan modifikasi *plugin* gamifikasi.
5. Modifikasi Platform, perlu atau tidaknya dilakukan modifikasi internal platform untuk mengakomodasi desain gamifikasi.
6. Kesimpulan. Berupa keterangan desain gamifikasi dapat diakomodasi sepenuhnya, sebagian, atau tidak sama sekali beserta saran agar dapat diakomodasi sepenuhnya.

2.4.2 Uji kinerja

Ujicoba kinerja dilakukan untuk mengetahui waktu respon, tingkat konsumsi CPU dan konsumsi memori platform ketika menangani request penambahan *event*. Layanan penambahan *event* melibatkan beberapa proses yang kompleks di dalam platform seperti proses eksekusi *event* dan proses eksekusi *eventTask*. Pengukuran waktu respon dengan cara menghitung selisih waktu pada awal dan akhir setiap proses tersebut. Waktu respon rata-rata didapatkan dengan menghitung waktu total dibagi dengan jumlah *event* atau *eventTask* yang berhasil dieksekusi oleh platform. Pengukuran tingkat konsumsi memori dan CPU dilakukan setiap 300 ms. Pengukuran tersebut dimulai pada awal inisialisasi hingga akhir uji kinerja.

Uji kinerja dilakukan dengan cara mengirimkan sejumlah request penambahan *event* kepada platform. Untuk itu dibuat lingkungan ujicoba yang terdiri dari 2ⁿ permainan dengan $n = (1, 2, \dots, 7)$. Setiap permainan diikuti oleh 20 pemain. Setiap permainan memiliki 5 macam aksi tunggal (*teamMode=false*) dengan komposisi seperti pada Tabel 1. Ditentukan 2 macam *task*, dengan jumlah minimal (*count*) setiap aksi ditentukan secara random dan posisi aksi (*at*) adalah nilai frekuensi aksi. Dengan ini setiap permainan akan ada 400 *event* dan 40 *eventTask* jika diasumsikan semua pemain mengerjakan semua *event* dan *eventTask* yang dibuat.

2.4.3 Uji usabilitas

Untuk mengetahui tingkat penerimaan calon pengguna terhadap platform yang telah dibuat dilakukan uji Usabilitas. Uji usabilitas menggunakan metode *System Usability Scale (SUS)* [12]. Ujicoba dilakukan dengan melibatkan 9 responden. Jumlah ini memenuhi syarat jumlah minimal untuk melakukan uji usabilitas yaitu 5 orang [13]. Responden yang ikut serta dalam uji usabilitas adalah pengajar diperguruan atau memahami aktivitas perkuliahan dengan baik. Responden memiliki latar belakang yang bervariasi. Hal ini bertujuan untuk membuktikan bahwa platform dapat digunakan oleh semua dosen. Skenario yang digunakan untuk ujicoba ini sebagai berikut :

1. Responden mendengarkan penjelasan tentang gamifikasi dan cara penggunaan platform.
2. Responden mempraktekkan penggunaan platform (dengan bantuan dan tanpa bantuan)
3. Responden membuat desain gamifikasi dari soal yang disediakan, kemudian menginputkannya ke dalam platform tanpa disertai bantuan.
4. Setelah responden melewati tahapan di atas, responden diperbolehkan mengisi kuis. Pertanyaan kuis terbagi menjadi 3 bagian. Bagian pertama kuis berisi pertanyaan untuk mengetahui tingkat pemahaman responden tentang konsep gamifikasi perkuliahan dan penggunaan platform. Bagian kedua kuis berisi pertanyaan dari uji usabilitas menggunakan metode SUS. Bagian ketiga adalah pendapat pengguna tentang fitur platform dan saran pengguna untuk perbaikan.

2.5 Implementasi

Gambar 4 adalah hasil implementasi pada platform dan *plugin* gamifikasi. Implementasi platform dilakukan dengan menggunakan lingkungan pemrograman Javascript, dan *database* Mysql. Sedangkan untuk *plugin* gamifikasi menyesuaikan dengan lingkungan LMS moodle yaitu pemrograman PHP dan *database* MySQL.



Gambar 4 Hasil implementasi platform dan *plugin* gamifikasi.

- (a) Tampilan permainan pada platform gamifikasi; (b) Tampilan leaderboard di *plugin* gamifikasi;
(c) Tampilan status dan profil pemain di *plugin* gamifikasi.

3. HASIL DAN PEMBAHASAN

Uji fleksibilitas yang dilakukan adalah dengan cara mengimplemetasikan desain gamifikasi yang digunakan pada beberapa publikasi. Ada 10 desain gamifikasi dari 10 publikasi yang diujikan. Tabel 3 adalah hasil uji coba implementasi desain. Hasil uji fleksibilitas menunjukkan bahwa dari 10 desain gamifikasi yang digunakan untuk uji fleksibilitas, 85% dapat diimplementasikan menggunakan platform. Dari hasil tersebut terbukti bahwa platform fleksibel atau dapat mengakomodasi desain gamifikasi yang bermacam-macam tanpa harus merubah internal platform.

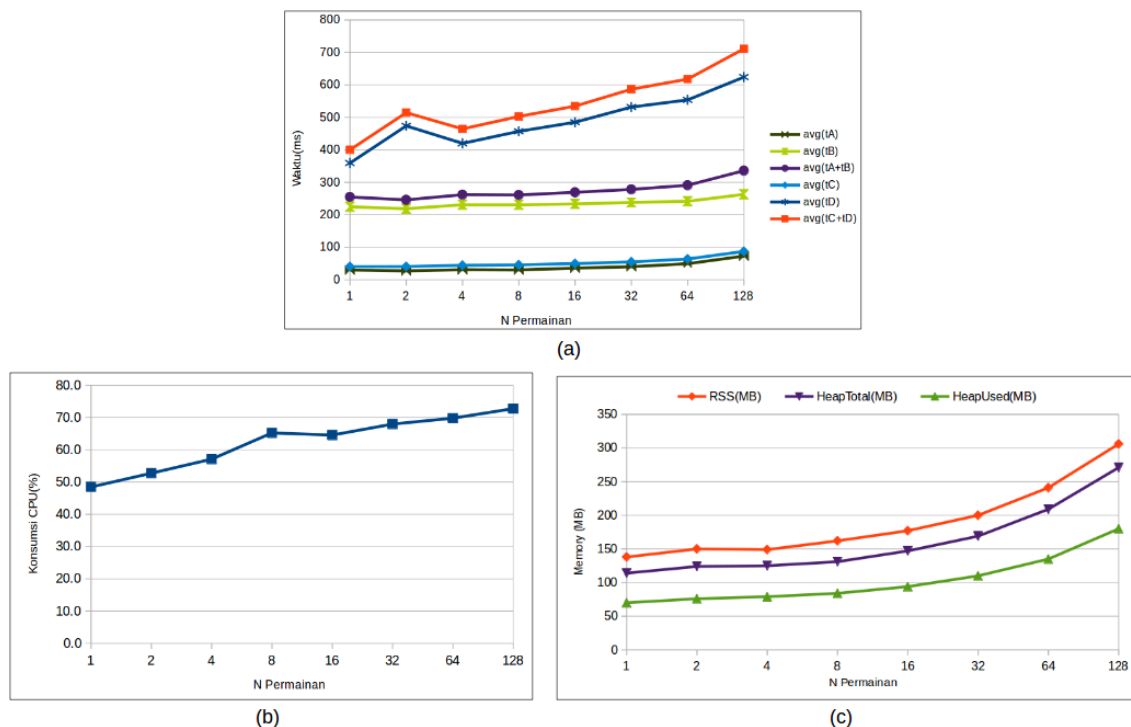
Tabel 3 Hasil uji fleksibilitas platform

No	Publi-kasi	Mekanika		Integrasi SI	Modifikasi			Kesimpulan
		Yg di-gunakan	Yg tdk tersedia		Atu-ran	Plugin	Plat-form	
1	[18]	(a), (b), (c), (d), (f)	-	√ (Moodle)	-	√	-	Desain terakomodasi sepenuhnya. Pengembangan <i>plugin</i> untuk aktivitas forum
2	[19]	(a), (b), (e)	-	-	-	-	-	Desain terakomodasi sepenuhnya
3	[5]	(a), (b), (c), (d)	<i>Virtual goods</i>	√ (Moodle)	-	-	√ [seba-gian]	Terakomodasi sebagian, dukungan terhadap virtual goods perlu ditambah
4	[4]	(a), (b), (c)	-	√ (Dapat menggu-nakan Moodle)	-	√	-	Terakomodasi sepenuhnya.
5	[3]	(a), (b), (d)	-	√ (Dapat menggu-nakan Moodle)	-	√	-	Terakomodasi sepenuhnya. Pengembangan <i>plugin</i> untuk aksi survei
6	[20]	(a), (b)	-	√	-	√	-	Terakomodasi sepenuhnya. Pengembangan <i>plugin</i> untuk aktivitas diforum seperti <i>post</i> , <i>assist</i> dan <i>like</i>
7	[14]	(a), (e)	-	-	√	-	-	Terakomodasi sepenuhnya
8	[15]	(a), (b), (d), (g), (h)	-	-	√	√	-	Terakomodasi sepenuhnya. Pengem-bangan <i>front end plugin</i> untuk menampilkan <i>leaderboard</i> sesuai desain.
9	[16]	(a), (c),(e)	5 kategori poin, <i>virtual economy</i>	√	√	√	√	Tidak terakomodasi. Perlu pengembangan platform untuk mendukung 5 jenis kategori poin dan mekanika <i>virtual economy</i>
10	[17]	(a),(e)	-	-	-	-	-	Terakomodasi sepenuhnya.

Keterangan: (a) poin/skor. (b) papan skor. (c) *badges*. (d) *progress/completion track*. (e) *level*. (f) *status*. (g) *activity feed/instant feedback*. (h) *avatar*. √ : ya. - : tidak

Agar dapat mengakomodasi seluruh desain, perlu dilakukan modifikasi platform dengan menambah dukungan terhadap penggunaan mekanika *virtual goods*, *virtual economy* dan poin dengan bermacam-macam kategori. Internal komponen di dalam rancangan (Gambar 3) harus dimodifikasi seperti (a) Komponen *GamificationEntity* perlu ditambahkan kemampuan untuk mengelola mekanika tersebut. (b) *Komponen GamificationDesignManagement* perlu ditambahkan pilihan kategori poin ketika mendefinisikan aksi. (c) *Komponen EventProcessor* perlu ditambahkan proses yang mengeksekusi *event* yang menyebabkan perubahan data pemain terkait mekanika tersebut. (d) Di dalam Model perlu ditambahkan modul untuk mengakses *database* yang berkaitan dengan data mekanika tersebut. (e) *Komponen Frontend* harus ditambahkan antarmuka yang berkaitan dengan pengelolaan mekanika poin dengan kategori yang bermacam, *virtual goods* dan *virtual economy*

Ujicoba kinerja platform dilakukan secara lokal, yaitu menggunakan satu komputer sebagai server dan klien. Pengujian menggunakan spesifikasi komputer dengan sistem operasi Linux Mint 17.3 64Bit, CPU AMD A6-4455 2.1 GHz dan memory 7623MB. Gambar 5 (a) adalah grafik waktu respon rata-rata dalam mengeksekusi *event* dan *eventTask*. Hingga cacah permainan yang dijalankan sekaligus sejumlah 128 permainan. Waktu respon rata-rata untuk mengeksekusi *event* (t_A+t_B) adalah 336ms. Waktu respon yang dibutuhkan platform untuk melayani *eventTask* (t_C+t_D) adalah 711 ms. Waktu respon tersebut dapat dikatakan cepat karena tidak lebih dari 1 detik [21]. Selain itu mengingat juga akan banyaknya permainan dan *event* yang harus diproses oleh platform. Penambahan cacah permainan yang dijalankan, selain menambah variasi *event* yang dieksekusi, juga menambah layanan yang memastikan *runtime* permainan tetap ada di memori.



Gambar 5: Grafik hasil uji kinerja platform dalam melayani *request* penambahan *event*
 (a) waktu respon platform; (b) tingkat penggunaan CPU; (c) Tingkat penggunaan memori

Hingga cacah permainan yang dijalankan sekaligus mencapai 128 permainan, rata-rata tingkat konsumsi CPU yang tertinggi adalah 73% Gambar 5 (b) dan rata-rata konsumsi memori (RSS- Resident set/ alokasi memori yang digunakan oleh aplikasi) yang tertinggi adalah 306MB

Gambar 5 (c). Ini berarti bahwa platform dapat bekerja dengan baik tanpa harus menghabiskan sumber daya yang ada. Sehingga dapat disimpulkan bahwa platform dapat bekerja dengan baik walaupun dalam lingkungan yang terbatas. Untuk meningkatkan hasil uji kinerja dapat dilakukan (1) memperbesar kapasitas hardware yang digunakan untuk menjalankan platform seperti meningkatkan kecepatan CPU dan kapasitas memori, (2) meminimalisir bug, (3) mengganti algoritma yang kurang efisien dengan yang lebih efisien.

Untuk hasil uji usability, dari 9 responden yang ikut serta hanya 7 responden yang dapat menyelesaikan tahapan uji usability dan memiliki jawaban benar minimal 80% pada pertanyaan bagian pertama. Dari ketujuh responden tersebut 5 adalah dosen dan 2 bukan dosen namun memahami aktivitas perkuliahan dengan baik. Sebanyak 42,8 % responden memberikan penilaian dengan total skor SUS bernilai lebih dari sama dengan 70 dan yang lainnya kurang dari 70. Skor SUS sama dengan 70 merupakan batas minimal yang termasuk dalam *range* dapat diterima sepenuhnya oleh pengguna [22]. Rata-rata total skor SUS adalah 60.0. Skor tersebut berdasarkan perbandingan rata-rata skor SUS [22] memiliki peringkat D, sifat OK dan berada pada batas bawah untuk dapat diterima pengguna. Tabel 4 adalah rata-rata skor SUS dari setiap pernyataan SUS.

Tabel 4 Hasil uji usability platform gamifikasi

No	Pernyataan	Rata ² skor SUS
Q1	Saya akan sering menggunakan sistem ini	2.3
Q2	Sistem ini seharusnya sederhana	1.1
Q3	Sistem ini mudah digunakan	2.6
Q4	Saya perlu bantuan tenaga teknis agar dapat menggunakan sistem ini	2.6
Q5	Beberapa fungsi pada sistem ini terintegrasi dengan baik	3.1
Q6	Terlalu banyak ketidakkonsistenan pada sistem ini	2.6
Q7	Pada umumnya orang-orang akan belajar menggunakan sistem dengan cepat	2.3
Q8	Sistem sangat rumit digunakan	2.6
Q9	Saya merasa percaya dan nyaman menggunakan sistem ini	2.7
Q10	Saya harus belajar banyak hal sebelum menggunakan sistem ini	2.1
	Total Skor SUS rata-rata = total * 2.5	60.0

Pernyataan Q4, Q7, dan Q10 memiliki skor rata-rata di atas 2 ini menunjukkan bahwa platform cenderung mudah dipelajari. Informasi yang cukup tentang konsep dan cara penggunaan platform, membuat pengguna dapat menyelesaikan tugas membuat desain gamifikasi secara mandiri/tanpa bantuan teknis. Skor pada Pernyataan Q2 bernilai 1.1. Skor tersebut menunjukkan bahwa perlu dilakukan perbaikan terhadap tahapan membuat desain gamifikasi pada Gambar 1 dan antarmuka. Rata-rata skor SUS pada Pernyataan Q5 dan Q6 masing-masing bernilai 3.1 dan 2.6. Skor tersebut berarti pengguna menemukan bahwa platform dapat berfungsi dengan baik. Skor tersebut juga mendukung hasil uji kinerja, yaitu bahwa platform memiliki kinerja yang baik.

Untuk meningkatkan hasil uji usability platform perlu dilakukan beberapa hal yaitu (1) pengembangan lebih lanjut pada bagian antar muka, (2) penyederhanaan tahapan desain gamifikasi, (3) menyediakan template desain gamifikasi atau mendukung kemampuan platform untuk menduplikasi desain gamifikasi dan (4) memperbanyak pelatihan, manual dan tutorial penggunaan platform.

4. KESIMPULAN

Penelitian tentang pengembangan platform gamifikasi untuk perkuliahan dengan Konsep *Generic Gamification Platform* (GGP) berhasil dilakukan. Penelitian ini menghasilkan beberapa kesimpulan, yaitu:

1. Platform yang telah dikembangkan memiliki fleksibilitas yang baik/dapat mengakomodasi desain gamifikasi perkuliahan yang berbeda-beda tanpa membutuhkan modifikasi internal platform. Dari 10 desain gamifikasi yang berbeda dapat diakomodasi menggunakan platform sebesar 85%.
2. Platform yang telah dikembangkan memiliki kinerja yang baik. Waktu respon rata-rata kurang dari 1 detik yaitu untuk eksekusi event kurang dari 336 ms dan eksekusi eventTask kurang dari 711 ms. Tingkat konsumsi CPU 73% dan memori 306 MB.
3. Platform yang telah dikembangkan dapat diterima oleh pengguna (5 dosen dan 2 bukan dosen tetapi memahami dengan baik aktivitas perkuliahan). Total skor SUS rata-rata platform adalah 60. Dengan skor tersebut platform memiliki peringkat kata sifat "OK", nilai D dan berada pada range batas bawah yang dapat diterima oleh pengguna.
4. Bagian platform yang terbukti mendukung fleksibilitas platform adalah bagian pemilihan mekanika permainan, pengelolaan mekanika permainan, pendefinisian aksi, pendefinisian task, pengelolaan aturan permainan, dan integrasi dengan aplikasi lain.

5. SARAN

Tahapan desain gamifikasi dan antar muka platform perlu dibuat menjadi lebih sederhana untuk lebih memudahkan pengguna dalam menggunakan aplikasi. Pengembangan platform untuk mendukung mekanika permainan yang belum didukung seperti *virtual goods*, dan *virtual economy*. Pengembangan *plugin* LMS Moodle lebih lanjut untuk mendukung aktivitas selain *assignment dan quize*. Pengembangan *plugin* gamifikasi untuk LMS selain Moodle.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Balitbang SDM Kominfo yang telah memberi dukungan finansial terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Deterding, S., Dixon, D., Khaled, R. & Nacke, L.E., 2011, Gamification: Toward a Definition, *CHI 2011 Gamification Workshop Proceedings*, hal. 12–15.
- [2] Hamari, J., Koivisto, J. & Sarsa, H., 2014, Does Gamification Work? — A Literature Review of Empirical Studies on Gamification, *2014 47th Hawaii International Conference on System Science*, vol. 47, hal. 3025–3034.
- [3] Domínguez, A., Saenz-de-navarrete, J., Fernández-sanz, L., Pagés, C. & De-Marcos, L., 2013, Gamifying learning experiences: Practical implications and outcomes, *Computers & Education*, vol. 63, hal. 380–392.
- [4] Ibanez, M.-B., Di-Serio, A. & Delgado-kloos, C., 2014, Gamification for Engaging Computer Science Students in Learning Activities: A Case Study, *IEEE Transactions on Learning Technologies*, Vol. 7, No. 3, Juli-September 2014, hal. 291–301.
- [5] Dicheva, D., Irwin, K., Dichev, C. & Talasila, S., 2014, A Course Gamification Platform Supporting Student Motivation and Engagement, *2014 International Conference on Web and Open Access to Learning (ICWOAL)*, hal. 1–4.

-
- [6] IEEE, 2010, Systems and software engineering -- Vocabulary, *ISO/IEC/IEEE 24765:2010(E)*, hal. 1–418.
- [7] Haaranen, L., Hakulinen, L., Ihantola, P. & Korhonen, A., 2014, Software Architectures for Implementing Achievement Badges – Practical Experiences, *2014 International Conference on Teaching and Learning in Computing and Engineering Software*, hal. 41–46.
- [8] Herzig, P., Ameling, M. & Schill, A., 2012, A Generic Platform for Enterprise Gamification, *2012 Joint Working Conference on Software Architecture & 6th European Conference on Software Architecture*, hal. 219–223.
- [9] Monterrat, B., Lavoué, É. & George, S., 2014, Motivation for Learning: Adaptive Gamification for Web-based Learning Environments, *6th International Conference on Computer Supported Education scenario*, hal. 117–125.
- [10] Zichermann, G. & Cunningham, C., 2011, Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps, 1 ed. Sebastopol, CA: O'Reilly Media.
- [11] Herzig, P., Ameling, M., Wolf, B. & Schill, A., 2015, Implementing Gamification: Requirements and Gamification Platforms, *Gamification in Education and Business*, hal. 1–21.
- [12] Brooke, J., 1996, SUS - A quick and dirty usability scale, *Usability Evaluation in Industry*, no. 189, hal. 189–194.
- [13] Sauro, J., 2013, 10 Things to Know About the System Usability Scale (SUS). <http://www.measuringu.com/blog/10-things-SUS.php>, diakses tgl 12 Maret 2016.
- [14] Laskowski, M. & Badurowicz, M., 2014, Gamification in higher education: a case study, *Management, Knowledge and Learning (MakeLearn) International Conference 2014*, hal. 971–975.
- [15] Todor, V. & Pitică, D., 2013, The gamification of the study of electronics in dedicated e-learning platforms, *36th Int. Spring Seminar on Electronics Technology*, hal. 428–431.
- [16] Swacha, J. & Baszuro, P., 2013, Gamification - based e - learning Platform for Computer Programming Education, *X World Conference on Computers in Education*, hal. 122–130.
- [17] Lambruschini, B.B. & Pizarro, W.G., 2015, Tech - Gamification in university engineering education: Captivating students, generating knowledge, in *International Conference on Computer Science & Education (ICCSE 2015)*, hal. 295–299.
- [18] Amriani, A., Aji, A.F., Utomo, A.Y. & Junus, K.M., 2013, An Empirical Study of Gamification Impact on E-Learning Environment, *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, hal. 265–269.
- [19] Borys, M. & Laskowski, M., 2013, Implementing game elements into didactic process: a case study, *Management, Knowledge and Learning International Conference 2013*, hal. 819–824.
- [20] Grant, E.S., Shankararaman, V. & Loong, J.L.K., 2014, Experimenting with Gamification in the Classroom, *2014 IEEE 6th International Conference on Engineering Education gamification*, hal. 79–83.
- [21] Neilsen, J., 2010, Website Response Times. <https://www.nngroup.com/articles/website-response-times/>, diakses tgl 07 November 2016.
- [22] Bangor, A., Staff, T., Kortum, P., Miller, J. & Staff, T., 2009, Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale, *Jurnal of Usability Studies*, vol. 4, no. 3, hal. 114–123.
-