

GROUP TASK MANAGEMENT BERBASIS GAMIFICATION

Adrian Shaksana Putra, Iwan Handoyo Putro, Petrus Santoso
Program Studi Teknik Elektro Universitas Kristen Petra
E-mail: m23414003@john.petra.ac.id, iwanhp@petra.ac.id, petrus@petra.ac.id

Abstrak - Di dalam dunia pekerjaan ataupun kuliah, semua orang dituntut memiliki kemampuan bekerja dalam kelompok untuk menyelesaikan sebuah proyek. Sejumlah aplikasi memungkinkan terciptanya koordinasi dan delegasi tugas dalam kelompok. Namun apabila tidak ditunjang dengan motivasi maka lambat laun kinerja akan berkurang. Untuk menciptakan motivasi tersebut maka salah satu cara yang dapat dilakukan adalah dengan implementasi konsep *gamification*. *Group Task Management* adalah aplikasi yang mengintegrasikan koordinasi dan delegasi tugas dalam proyek dengan konsep *gamification*. Dalam aplikasi ini terdapat fitur membuat grup pekerjaan atau proyek dimana pengguna dapat mengumpulkan sebuah kelompok untuk suatu proyek. Pengguna dapat mendelegasikan tugas dan memantau kinerja kelompok yang tergabung dalam suatu grup. Aplikasi diuji oleh 15 pengguna terbagi dalam 3 kelompok usia yaitu dewasa, mahasiswa dan pelajar SMA masing – masing 5 responden. Berdasarkan hasil kuisioner disimpulkan aplikasi ini dapat membantu dan memotivasi proses kerja kelompok.

Kata Kunci - Aplikasi mobile, task management, gamification.

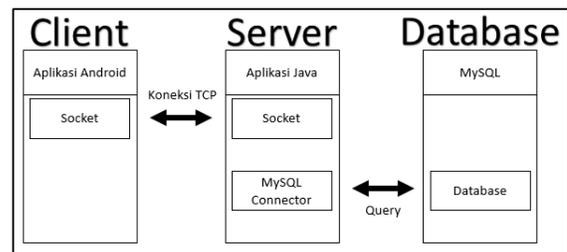
I. PENDAHULUAN

Di dalam dunia pekerjaan, karyawan dituntut untuk memiliki kemampuan bekerja dalam kelompok untuk menyelesaikan sebuah proyek. Sebuah proyek harus memiliki goal atau tujuan yang jelas, sehingga pendelegasian tugas dapat terlaksana dengan lancar. Untuk itu ada beberapa faktor yang harus diperhatikan agar sebuah kerja tim dapat menyelesaikan proyek seperti koordinasi, komunikasi, kontribusi, dukungan, motivasi, keseimbangan dan koherensi tugas dan anggota [1]. Masalah terbesar ialah bagaimana mengatur komposisi anggota kelompok dan bagaimana mendelegasikan sebuah tugas serta memonitor setiap kinerja anggota kelompok. Karena apabila komposisi anggota kelompok tidak memenuhi standar yang diperlukan untuk mengerjakan sebuah proyek maka proyek tidak akan bekerja dengan lancar, begitu juga dengan delegasi tugas dan pemantuan kinerja apabila tidak dikontrol dan direncanakan dengan baik maka proyek tidak akan selesai tepat waktu dan kemungkinan gagal. Tugas yang diberikan kepada kita dalam kelompok erat kaitannya dengan kewajiban. Berdasar artikel [2] menjelaskan tentang perbedaan keinginan dan kewajiban, dimana kewajiban terkadang tidak sama dengan keinginan. Sehingga terkadang kewajiban menjadi sesuatu hal yang mengurung kebebasan seseorang, hal ini mengakibatkan kita sering melalaikan kewajiban untuk keinginan kita. Dalam hubungannya dalam kerja kelompok adalah sebagai anggota adanya tekanan atau paksaan sehingga kebebasan seorang anggota terkukung. Maka dari itu diperlukan suatu cara untuk membungkus “kewajiban” ini menjadi suatu “keinginan”. Untuk membungkus hal tersebut maka konsep gamification dipakai. Gamification adalah suatu konsep yang dapat

menghadirkan suasana “game” dalam kegiatan sederhana [3]. Dengan gamification dapat mengubah pola pikir pengguna terhadap sesuatu yang merupakan rutinitas menjadi suatu yang menyenangkan. Karena itu perlu dikembangkan suatu aplikasi Task management yang mengintegrasikan unsur - unsur gamification agar dapat memotivasi pengguna menyelesaikan dan menjadwalkan tugas dan kegiatan dengan baik. Dalam peranan untuk membantu manajemen beberapa developer aplikasi smartphone membuat aplikasi yang menunjang biasa disebut Task management apps. Aplikasi ini memiliki fungsi sebagai pengatur, pencatat, dan pengingat tugas dan kegiatan sehari – hari yang dapat digunakan secara pribadi atau kelompok. Karena tingkat pengguna smartphone terus meningkat maka sarana terbaik untuk membantu ialah dengan menggunakan smartphone. Maka dari itu perlu dikembangkan suatu aplikasi group Task management yang mengintegrasikan unsur - unsur gamification agar dapat memotivasi pengguna menyelesaikan dan menjadwalkan kegiatan dengan baik.

II. PERANCANGAN SISTEM

Desain sistem secara garis besar dari aplikasi *Group Task Management* adalah sebagai dapat digambarkan oleh Gambar 1.



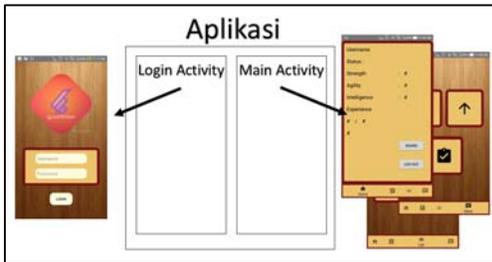
Gambar 1. Desain Sistem

Secara garis besar sistem terbagi menjadi 3 bagian besar yaitu *client*, *server* dan *database*. Bagian *client* adalah aplikasi *android* yang memiliki komponen *socket* untuk menciptakan koneksi TCP ke *server*. Bagian *server* adalah aplikasi *java* yang memiliki *socket* untuk menerima koneksi TCP dari *client*, bagian *server* juga memiliki komponen *MySQL Connector* untuk dapat melakukan proses *query* ke *database*.

A. Desain Client

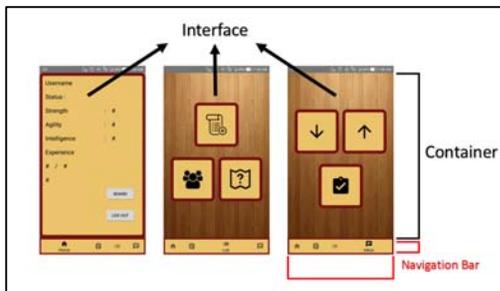
Aplikasi *client* berupa aplikasi *android* yang bisa digunakan di *smartphone android*. Aplikasi yang dibuat menggunakan *android API 19*. Aplikasi ini memiliki banyak *interface* yang didesain untuk mengakomodasi fungsi – fungsi yang akan dijalankan. Aplikasi *android* ini didesain memiliki 2 *activity*

yaitu 1 *activity login* dan 1 *main activity* seperti pada Gambar 2.



Gambar 2. Skema Aplikasi (1)

Main activity ini diisi oleh 1 *navigation bar* utk navigasi *interface* dan 1 *container* untuk menampilkan berbagai macam *interface* seperti pada Gambar 3.

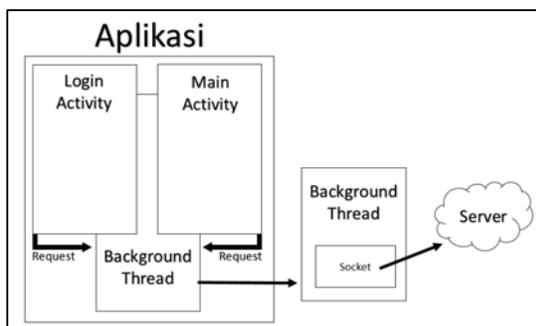


Gambar 3. Skema Aplikasi (2)

Untuk melakukan *request* ke *server* maka aplikasi ini mempunyai 1 *background thread* yang dipanggil setiap *user* ingin melakukan *request*. Didalam *background thread* terdapat komponen *socket* yang merupakan sarana untuk melakukan koneksi dan mengirim data/*request* ke *server* seperti pada Gambar 4.

Server memiliki beberapa entitas yang berelasi satu sama lainnya dan memiliki fungsi:

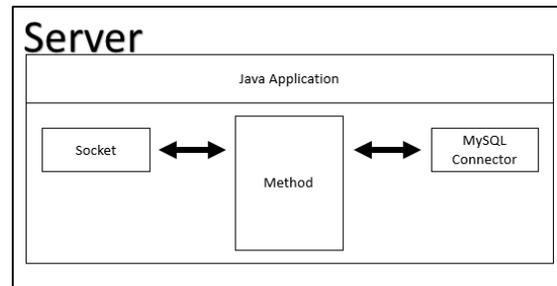
- *Socket*, bertipe *serversocket* dengan *ip address* dan *port* tertentu digunakan untuk menerima komunikasi TCP berupa pesan *string*.
- *Method*, berupa fungsi – fungsi yang dijalankan sesuai pesan *string* dari *client*. Fungsi – fungsi ini dieksekusi di dalam *thread*.
- *MySQL Connector*, berupa objek digunakan untuk proses *query* yang berfungsi mengambil data dari *database* sesuai pesan yang didapat oleh *client*. Objek ini diakses melalui *method*.



Gambar 4. Skema Aplikasi (3)

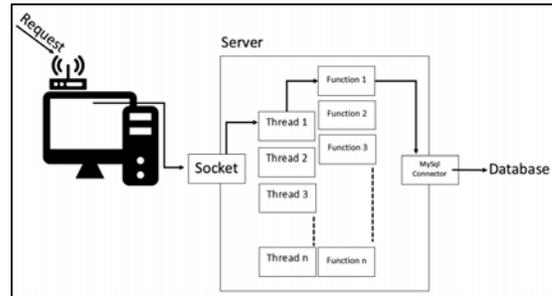
B. Desain Server

Secara garis besar desain *server* ada pada Gambar 5



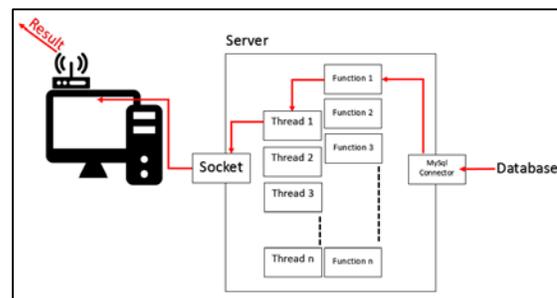
Gambar 5. Desain Server

Server yang digunakan adalah *server* dari pemrograman java. *Server* ini didesain untuk berjalan di komputer lokal yang terkoneksi dengan jaringan internet. *Server* yang didesain mengimplementasikan *multithreading* dimana setiap kali *server* menerima suatu koneksi dari aplikasi *android* maka *server* akan membuat satu *thread* untuk memproses *request* aplikasi *android* seperti pada Gambar 6.



Gambar 6. Alur Server (1)

Dalam *thread* ini diisi oleh berbagai fungsi – fungsi yang hanya bisa diakses oleh *keyword* tertentu dari *request* aplikasi *android*. Fungsi – fungsi tersebut akan menjalankan sebuah proses *query* yang akan diteruskan ke *database* setelah menerima hasil/nilai dari *database* maka nilai tersebut akan diproses untuk dikembalikan ke aplikasi *android*. Setelah nilai dikembalikan ke aplikasi *android* maka *thread* tersebut akan ditutup seperti pada Gambar 7.



Gambar 7. Alur Server (2)

C. Desain Database

Database pada aplikasi terdiri 5 macam tabel untuk menyimpan data – data yang dibutuhkan. Berikut merupakan nama tabel – tabel yang dibuat:

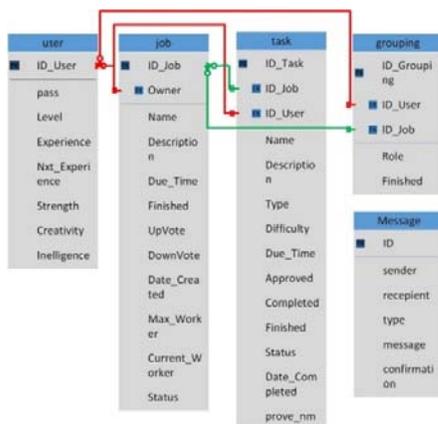
- Tabel *user*
Tabel ini berfungsi menyimpan data setiap *user* yang terdaftar dalam sistem. Data – data tersebut meliputi status *user* tersebut, kata sandi untuk mengakses aplikasi dan *username* sebagai pengidentifikasi.
- Tabel *job*
Tabel ini berfungsi untuk menyimpan data – data dari semua *job* yang *user – user* dalam sistem telah buat. Data-data yang bersangkutan meliputi parameter dari *job* seperti nama, deskripsi, tenggat waktu, ketersediaan *worker* dan masih banyak lagi.
- Tabel *task*
Tabel ini berfungsi untuk menyimpan data – data yang berhubungan dengan semua *task* yang *user* buat. Secara garis besar data – data tersebut meliputi status dari *task* tersebut dan parameter - parameter pengidentifikasi *task* tersebut.
- Tabel *message*
Tabel ini berfungsi untuk menyimpan pesan – pesan yang dikirim oleh *user – user*.
- Tabel *grouping*
Tabel ini berfungsi untuk mendaftarkan nama – nama *user* yang tergabung dalam suatu grup.

Dari tabel – tabel diatas akan diimplementasikan sebuah relasi yang bisa dilihat pada Gambar 8.

D. Desain Layanan

Secara garis besar skema interaksi dari layanan dapat dilihat pada Gambar 9.

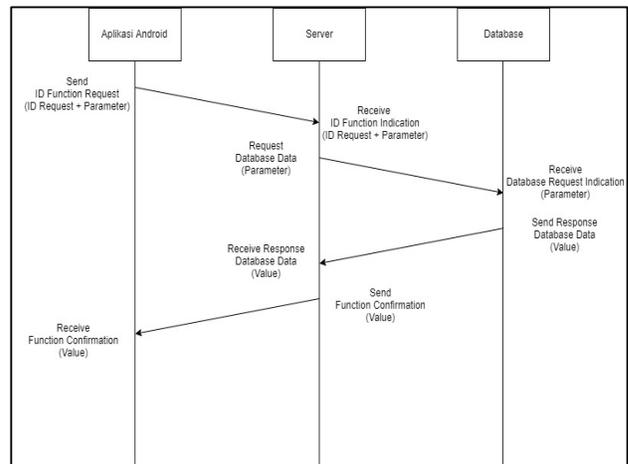
Aplikasi *android* mengirim pesan kepada *server*. Pesan yang disampaikan merupakan data *string* yang memuat ID layanan (ID Request) beserta parameter yang dibutuhkan untuk layanan tersebut. Setelah pesan diterima oleh *server* maka *server* akan mengolah parameter sesuai layanan yang telah dipilih lewat ID Request. Proses pengolahan data membutuhkan *database* maka *server* akan berhubungan dengan *database*. Setelah pengolahan/perubahan/pengambilan data ke *database* selesai dilakukan maka *server* akan memberikan notifikasi atau data yang diolah kembali ke aplikasi *android*.



Gambar 8. Entity Relation Diagram

Beberapa layanan tertentu tidak dapat diakses *user* apabila *user* tidak memerankan peran yang bersangkutan. Aplikasi ini memiliki *user* dengan 2 macam peran, peran tersebut adalah *leader* dan *worker*. Setiap peran memiliki layanan masing –

masing yang eksklusif untuk peran tersebut, daftar layanan dapat dilihat pada Tabel 1, Tabel 2, Tabel 3.



Gambar 9. Time Sequence Diagram

Tabel 1. Daftar Layanan Umum

Daftar Layanan	Fungsi	Parameter
Request Login	Mengkonfirmasi User yang akan masuk ke aplikasi	ID_Request, Username, Password
Request Home	Menampilkan data milik <i>user</i>	ID_Request, Username
Request View Job	Menampilkan job milik <i>user</i> dan forum job	ID_Request, Username
Request View Task	Menampilkan task dari job yang bersangkutan	ID_Request, Username, ID_Job
Request Apply Task	Layanan untuk <i>user</i> mendaftar task dari job yang bersangkutan	ID_Request, Username, ID_Task
Request Create Job	Layanan untuk <i>user</i> membuat Job	ID_Request, Username
Request View Message	Layanan untuk <i>user</i> melihat pesan seperti pesan undangan, pendaftaran kegiatan dan vote	ID_Request, ID_User
Request Reply Message	Layanan untuk <i>user</i> membalas pesan - pesan tertentu dengan opsi pilihan yang sudah disediakan	ID_Request, ID_User, ID_Job

Tabel 2. Daftar Layanan *Leader*

Daftar Layanan	Fungsi	Parameter
Request Finish Job	Layanan untuk <i>user</i> (leader) mengakhiri job yang <i>user</i> pimpin sebelum tenggat waktu	ID_Request, ID_User, ID_Job
Request Delete Task	Layanan untuk <i>user</i> (leader) menghapus task yang telah dibuat dari job yang bersangkutan	ID_Request, Username, ID_Task
Request Approve Task	Layanan untuk <i>user</i> (leader) menyetujui task yang telah diselesaikan <i>user</i> (worker) yang mendaftar task tersebut	ID_Request, Username, ID_Task
Request Create Task	Layanan untuk <i>user</i> membuat task dari job yang bersangkutan	ID_Request, ID_Job
Request View Worker	Layanan untuk <i>user</i> (leader) melihat <i>user</i> (worker) yang tergabung dalam job	ID_Request, ID_Job
Request Invite Worker	Layanan untuk <i>user</i> (leader) mengundang <i>user</i> (worker) untuk bergabung dalam job	ID_Request, ID_Job, ID_User
Request Assign Worker	Layanan untuk <i>user</i> (leader) mengganti task <i>user</i> (worker) ke task lain	ID_Request, ID_User, ID_Task
Request Remove Worker	Layanan untuk <i>user</i> (leader) mengeluarkan <i>user</i> dari job	ID_Request, ID_User, ID_Job

Tabel 3. Daftar Layanan *Worker*

Daftar Layanan	Fungsi	Parameter
Request Abandon Job	Layanan untuk <i>user</i> (worker) guna meninggalkan job yang artinya <i>user</i> tersebut ingin keluar dari grup tersebut	ID_Request, ID_User, ID_Job
Request Finish Task	Layanan untuk <i>user</i> (worker) memberi indikasi penyelesaian task dari job yang bersangkutan	ID_Request, Username, ID_Task
Request Abandon Task	Layanan untuk <i>user</i> (worker) meninggalkan task dari job yang bersangkutan	ID_Request, Username, ID_Task
Request Vote	Layanan untuk <i>user</i> (worker) melakukan proses voting di akhir job sebelum mendapatkan reward	ID_Request, ID_User, ID_Job

E. Desain Gamification

Berikut merupakan daftar penjelasan terminologi yang akan digunakan dalam penjelasan:

- *User*, istilah lain dari pengguna yang sudah menginstall aplikasi dan terdaftar di sistem
- *Worker*, istilah lain dari pengguna yang menjadi anggota dalam sebuah grup
- *Leader*, istilah lain dari pengguna yang menjadi ketua dalam sebuah grup
- *Job*, istilah lain dari 1 grup.
- *Job offering*, istilah lain dari forum untuk melihat grup – grup yang tersedia.
- *Task*, istilah lain dari tugas yang diberikan untuk anggota grup oleh ketua grup.

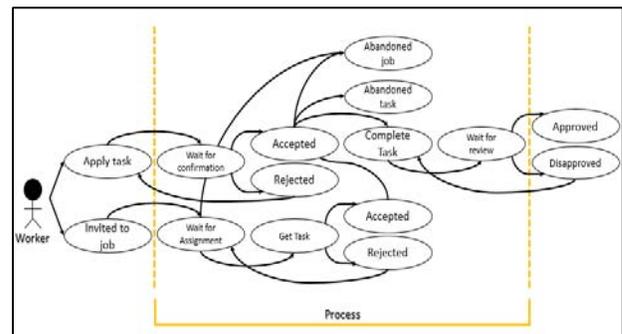
1. User

Setiap *User* yang terdaftar dalam sistem memiliki status/parameter. Pada awal *user* terdaftar dalam sistem nilai parameter yang diberikan sistem ke *user* adalah nilai *default*. Beberapa parameter antara lain adalah:

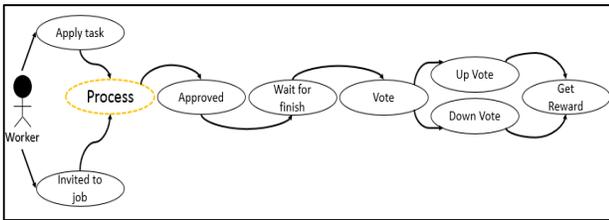
- *Level* dan *Experience*, *Experience* adalah poin yang didapatkan setelah menyelesaikan *task*, setelah *experience* mencapai poin tertentu maka *level* akan bertambah, pada mulanya *experience* bernilai 0 dan *level* bernilai 1.
- *Intelligence*, pekerjaan yang berhubungan dengan perhitungan, perencanaan, dan perancangan seperti kalkulasi anggaran, perhitungan matematis, perencanaan jadwal, pembagian tugas, inventaris, perancangan gambar teknis, *rundown* kegiatan, dan laporan tertulis
- *Creativity*, pekerjaan yang berhubungan dengan *skill* berbicara, mendesain presentasi, melakukan dekorasi dan melakukan dokumentasi.
- *Strength*, pekerjaan yang berhubungan dengan mengakomodasi peralatan dan kebutuhan member yang lain, melakukan pekerjaan dengan tenaga fisik seperti survei, berbelanja dan pekerjaan tangan.

Parameter *intelligence*, *Creativity* dan *strength* yang dimiliki saat pertama kali mendaftar di sediakan secara *default* yaitu 1,1,1 .

Satu *User* memiliki 2 peran sekaligus yaitu *worker* dan *leader*. *User* memerankan *worker* ketika *user* bergabung dalam *job* yang dibuat oleh *leader*, maka secara sistem *user* tersebut tergabung menjadi *worker* dalam *job* tersebut. *Worker* merupakan *user* yang ditugaskan untuk menyelesaikan *task* yang diterima oleh mereka. Alur peran dapat dilihat pada Gambar 10 dan Gambar 11.

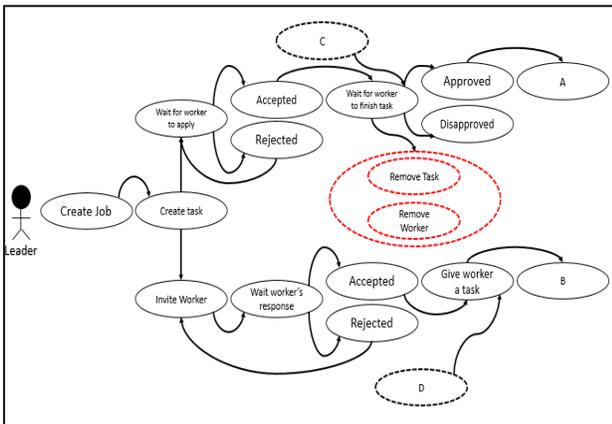


Gambar 10. Alur *Worker*

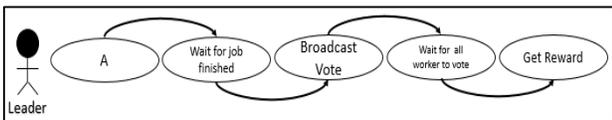


Gambar 11. Alur Worker Setelah Task di Approve

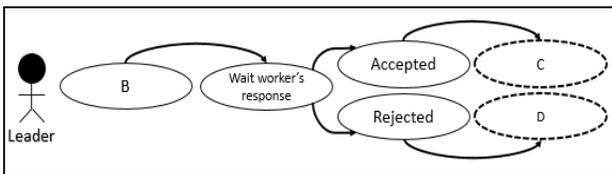
User memerankan perannya sebagai leader ketika user membuat suatu job maka secara sistem user tersebut menjadi leader dari job tersebut. Leader merupakan pemimpin dari sebuah job yang bertugas untuk mengurus user yang bertugas menjadi worker mulai dari manajemen worker – worker, pendelegasian dan pemeriksaan task. Alur peran dapat dilihat pada Gambar 12, Gambar 13 dan Gambar 14.



Gambar 12. Alur Leader



Gambar 13. Alur Leader Setelah Semua Task di Approve



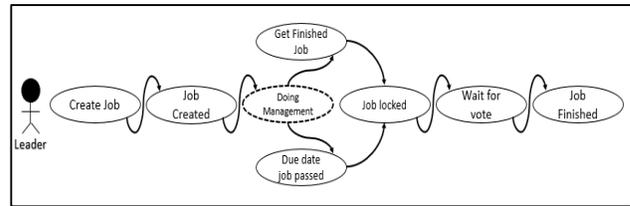
Gambar 14. Alur Leader Saat Memberikan Task

2. Job dan Task

Job merupakan istilah lain dari grup didalam aplikasi ini yang dibuat oleh user. User yang menciptakan sebuah job akan otomatis menjadi leader dari job tersebut. Job ini digunakan sebagai tempat/sarana berkumpulnya worker – worker dan leader, dimana leader dapat melakukan fungsi – fungsinya seperti memantau progress, mengulas pekerjaan dan memanajemen beban dan kuantitas/jumlah worker. Alur job dapat dilihat pada Gambar 15.

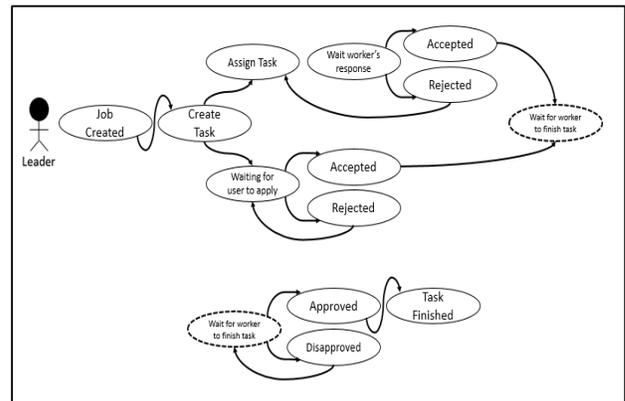
Siklus hidup sebuah job bergantung pada keputusan leader atau tenggat waktu job. Ketika leader memutuskan untuk mengakhiri job sebelum tenggat waktu atau job sudah berada di tenggat waktu maka kondisi job tersebut akan masuk ke fase

voting dimana job dan task – task yang berkaitan dalam job tersebut dikunci (tidak bisa di ulas atau ditambahkan). Fase voting adalah fase dimana para worker dalam job tersebut mendapat hak untuk voting untuk tingkat keberhasilan job tersebut. Setelah semua worker memberikan vote maka job tersebut dinyatakan selesai.



Gambar 15. Alur Job

Task merupakan istilah lain dari kegiatan di dalam aplikasi ini. Task diciptakan oleh leader dalam suatu grup/job. Task yang diciptakan dapat diberikan ke worker – worker yang tergabung dalam grup/job oleh leader, atau dapat dilihat oleh worker – worker di dalam halaman forum (lihat Job Offer). Alur task dapat dilihat pada Gambar 16.



Gambar 16. Alur Task

Task diciptakan di dalam job, maka dari itu job harus diciptakan terlebih dahulu. Task yang diciptakan akan otomatis menjadi task milik job tsb. Task dapat dimiliki worker dengan 2 cara yaitu apabila worker mendaftar/melamar task tersebut ke Leader yang memiliki task tersebut atau task dapat didelegasikan secara langsung oleh leader kepada worker yang sudah tergabung dalam job yang bersangkutan. Siklus hidup sebuah task bergantung kepada penyelesaian task tersebut dan tenggat waktu task tersebut. Task yang diselesaikan oleh worker harus diulas oleh leader terlebih dahulu apabila task mendapat ulasan baik maka task tersebut sudah berakhir, tetapi apabila mendapat ulasan kurang baik maka worker dihimbau untuk memperbaiki pekerjaannya untuk diulas kembali oleh leader.

3. Job Offer

Job Offer merupakan halaman yang dapat diakses oleh user yang berisi job yang di buat oleh Leader. Di halaman ini user dapat melihat job yang tersedia beserta task di dalamnya. Apabila user tertarik dengan suatu job dan task maka user dapat mendaftar pada task tersebut. Setelah mendaftarkan maka sistem akan mengirim message kepada user yang menjadi leader dari job yang bersangkutan tentang perihal seorang user mendaftar pada suatu task. Leader dapat menolak

atau menyetujui permintaan pendaftaran tersebut. Apabila *leader* menyetujui pendaftaran maka *user* tadi yang mendaftar otomatis menjadi *worker* untuk *job* tersebut serta mengemban *task* yang tadi didaftarkan di awal, sekaligus *user* tersebut tergabung menjadi dalam *job* tersebut.

4. *Reward dan Losses*

Untuk meningkatkan *level* maka poin *experience user* harus mencapai poin tertentu. Perhitungan *experience* menggunakan progresi eksponensial [4].

$$Experience\ level\ n = 1500 \times (n^{1.5}) \tag{1}$$

Reward dan *losses* aplikasi ini ialah penambahan dan pengurangan *Experience* dan parameter. *Reward* terbagi menjadi 3 yaitu:

- *Reward Task*, *reward* didapat dari penyelesaian setiap individual *task*. *Reward* diberikan ketika *job* berakhir. Bentuknya berupa penambahan parameter sesuai jumlah bintang dan *experience* pada Tabel 4.

Tabel 4. *Experience Task*

Jumlah Bintang	Jumlah <i>experience task</i>
1	110
2	140
3	280

- *Reward Leader* adalah 75 % *experience* yang didapat oleh keseluruhan *task* yang *worker* sudah selesaikan. *Experience* ini diberikan khusus untuk *leader* berdasar hasil *vote* dari *worker* pada saat *job* berakhir. *Experience* akan ditambah apabila *leader* mendapat *up vote* dari lebih banyak daripada *down vote*. *Experience* tidak akan diberikan apabila *down vote* lebih banyak daripada *up vote* dari keseluruhan hasil *vote*.

Losses terbagi menjadi 2 yaitu:

- *Losses Task*, *losses ini* didapat apabila *Task* tidak diselesaikan atau melebihi tenggat waktu. Bentuknya berupa pengurangan parameter sesuai jumlah bintang dan *experience* pada Tabel 4.
- *Losses leave*, *Losses ini* didapat apabila *worker* meninggalkan grup sendiri. Bentuknya berupa pengurangan parameter sebesar 2 kali jumlah bintang dan *experience* sebesar 1000.

III. PENGUJIAN SISTEM DAN ANALISA

Pengujian ini dilakukan dengan memberikan uji coba aplikasi kepada 15 calon pengguna dalam kurun waktu tertentu (7 hari). Setelah itu calon pengguna akan diberi angket untuk mengukur efektifitas aplikasi yang dinilai berdasarkan apakah fitur – fitur yang sudah diimplementasikan membantu untuk mengatur jadwal kegiatan grup. Berikut merupakan hasil angket dari grup dewasa:

Tabel 5. Deskripsi Fitur

List Fitur	Deskripsi fitur
1	Membuat grup
2	Pendelegasian tugas
3	Perekrutan member grup
4	Monitor progress
5	Feedback
6	Messaging
7	Sistem poin

Tabel 6. Hasil Angket Grup Dewasa

List Fitur	Respon		
	YA	TIDAK	
1	4	1	
2	4	1	
3	3	2	
4	3	2	
5	4	1	
6	3	2	
7	4	1	
Aplikasi membantu mengatur jadwal kegiatan			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1	3	1	0

Berikut merupakan kesimpulan dari kekurangan aplikasi menurut grup dewasa:

- UI kurang *informative* sehingga untuk pengguna baru mengalami kesulitan untuk mengeksplorasi aplikasi.
- Tidak ada pilihan supaya *job* yang baru dibuat tidak ditampilkan di halaman *job offering*.
- Tidak ada fitur untuk saling mengingatkan antara *worker* dan *leader*.
- Tidak ada fitur *feedback* yang bersifat komentar.
- Tidak ada fitur *chat* yang bisa memudahkan komunikasi.

Berikut merupakan hasil angket dari grup mahasiswa:

Tabel 7. Hasil Angket Grup Mahasiswa

List Fitur	Respon		
	YA	TIDAK	
1	5	0	
2	4	1	
3	5	0	
4	5	0	
5	5	0	
6	5	0	
7	5	0	
Aplikasi membantu mengatur jadwal kegiatan			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1	3	1	0

Berikut merupakan kesimpulan dari kekurangan aplikasi menurut grup mahasiswa:

- Desain UI kurang menarik dan membingungkan pengguna.
- Kurang nya sanksi yang diberikan bagi *Worker* yang tidak mengerjakan *task*
- Aplikasi tidak memiliki ikatan yang kuat.

Berikut merupakan hasil angket dari grup SMA:

Tabel 8. Hasil angket grup SMA

List Fitur	Respon		
	YA	TIDAK	
1	5	0	
2	5	0	
3	5	0	
4	5	0	
5	5	0	
6	5	0	
7	5	0	
Aplikasi membantu mengatur jadwal kegiatan			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
3	2	0	0

Berikut merupakan kesimpulan dari kekurangan aplikasi menurut grup SMA:

- *Reward* yang didapat kurang menggugah.
- Aplikasi hanya dapat menerima gambar.
- Aplikasi hanya dapat upload 1 bukti/*prove*.
- Design UI yang kurang menarik.

Dalam kelompok SMA hanya 1 orang saja yang menjadi *leader* dan 4 orang lainnya menjadi *worker*. Khusus untuk pengguna pelajar SMA sudah menguji coba aplikasi secara nyata dengan pekerjaan nyata yaitu membuat presentasi. Dalam kasus ini para pelajar SMA memiliki tugas kelompok membuat presentasi. Pengguna pelajar SMA menggunakan aplikasi untuk delegasi tugas dan mengirim *feedback* hingga tugas membuat presentasi selesai.

IV. KESIMPULAN

Unsur *reward* dan *losses* yang sudah diimplementasikan dapat dipahami oleh grup dewasa dan mahasiswa. Untuk grup SMA unsur *reward* dan *losses* dapat dipahami dan sudah dicoba secara nyata sehingga membantu responden memotivasi penyelesaian pekerjaan, hal ini dibuktikan lewat pekerjaan grup SMA yang sudah diselesaikan.

Unsur *competitive* tidak diimplementasikan dikarenakan dalam fitur poin yang diimplementasikan kurang mempresentasikan kemampuan individu. Contoh pada kasus penguji dewasa yang memiliki poin *strength* 3 dan penguji SMA yang memiliki poin *strength* 3 belum tentu memiliki kapasitas yang sama dalam kehidupan nyata maka dari itu kemampuan tersebut tidak bisa dibandingkan dengan adil.

DAFTAR PUSTAKA

- [1] M. Hoegl and H. G. Gemuenden. (2001). Teamwork Quality and the Success of Innovative Project: A Theoretical Concept and Empirical Evidence. *JSTOR*, pp. 435-449.
- [2] N. Eyal. (2013). Why Behavior Change Apps Fail To Change Behavior. 13 July 2013. [Online]. Available: <https://techcrunch.com/2013/07/13/why-behavior-change-apps-fail-to-change-behavior/>.
- [3] S. Deterding, D. Dixon, R. Khaleed and L. Nacke. (2011). From Game Design Elements to Gamefulness: Defining “Gamification”. *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pp. 9-15.
- [4] C. Bateman. (2006). Only A Game: Mathematic of XP. 8 Agustus 2006. [Online]. Available: https://onlyagame.typepad.com/only_a_game/2006/08/mathematics_of_.html.