

Ekstraksi Kebutuhan Aplikasi Berdasarkan Feedback Pengguna Menggunakan Naïve Bayes dan Gamifikasi

Andre Rusli

Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
andre.rusli@umn.ac.id

Diterima 30 Maret 2018

Disetujui 8 Juni 2018

Abstrak—Requirements engineering is a series of activities which aims to elicit, analyze, evaluate, and document the requirements of a system that is being developed. The activities do not stop after the product is deployed but continues as the users use the product and provide feedbacks to the system and matter how decent the functionalities of a product are, if it cannot address the correct problem and/or opportunities of the stakeholders or users, the product cannot be considered useful. That being said, not all stakeholders are willing to participate in providing useful feedbacks to improve the product after deployment, for many reasons. Gamification is considered as an opportunity that can be utilized to improve the motivation of user to use a product by implementing game design elements into an existing software product, thus increasing user participation to contribute in providing useful feedbacks and evolving requirements of a software product. This research proposes a model to support engineers in motivating users to provide feedbacks using gamification and also Naïve Bayes Classifier to classify user feedbacks into categories needed by the developer to extract the requirements stated in the feedback, such as bug reports, feature request, user experiences, etc.

Kata Kunci—requirements engineering, gamification, Naïve Bayes, user feedback

I. PENDAHULUAN

Pemahaman dan definisi yang tepat akan suatu masalah merupakan hal yang penting dalam menciptakan suatu solusi *software* yang tepat. Untuk itu, diperlukan proses-proses untuk menemukan, memahami, memformulasi, menganalisis, dan menemui persetujuan terkait masalah apa yang harus diselesaikan, kenapa masalah tersebut harus diselesaikan, dan siapa yang harus terlibat dalam penyelesaian masalah tersebut. Secara umum, proses analisis kebutuhan meliputi hal-hal yang telah disebutkan di atas [1]. Tahapan analisis kebutuhan atau yang disebut juga dengan *requirements engineering* (RE) menjadi salah satu tahapan yang paling penting dalam *software development life cycle* (SDLC), karena hasil dari proses analisis kebutuhan tersebutlah yang akan menjadi pondasi dan landasan dari proyek yang akan dijalankan.

Setelah kebutuhan didefinisikan di awal, kebutuhan terhadap sistem dapat terus berubah dan berkembang seiring berjalannya waktu dan perkembangan teknologi. Setelah *system-to-be* selesai dibangun, muncul lagi kebutuhan untuk merancang dan membangun *system-to-be-next*. Objektif suatu sistem, struktur konseptual, kebutuhan dan asumsi yang telah didapatkan, dievaluasi, dispesifikasi, dan dianalisis sebelumnya mungkin perlu diubah dan/atau dikembangkan lebih lanjut karena berbagai alasan. Beberapa alasan tersebut mencakup *error* atau *bug* yang ditemukan pada sistem, perubahan prioritas dan *constraints* pada sistem, keunggulan dan keterbatasan sistem, dan banyak hal lainnya. Hal tersebut yang dinamakan dengan *requirements evolution* [1] atau evolusi kebutuhan.

Perubahan-perubahan tersebut dapat terjadi baik saat analisis kebutuhan dilakukan di awal, saat kebutuhan dievaluasi, saat perancangan maupun pembangunan *software*, atau bahkan setelah sistem dirilis ke publik dalam bentuk *user feedback*. *Feedback* pengguna, seperti yang dapat ditemukan pada pasar aplikasi seperti Google Play Store, dapat memiliki banyak informasi terkait *requirements* seperti *bugs* pada aplikasi [2], permintaan pengembangan fitur [3], dan juga ide-ide baru terkait fitur tertentu pada aplikasi atau bahkan fitur baru [4].

Partisipasi pengguna atau yang sering disebut dengan *stakeholder* dalam suatu proyek pengembangan piranti lunak sangatlah penting. Keterlibatan pengguna dalam menganalisis kebutuhan terbukti dapat meningkatkan penerimaan pengguna [5], mengurangi resiko kegagalan proyek [6], dan juga meningkatkan pemahaman pengguna terhadap sistem [7]. Adapula survei yang menunjukkan bahwa partisipasi pengguna dalam suatu proyek pengembangan *software* merupakan faktor paling penting yang dapat menentukan keberhasilan maupun kegagalan suatu proyek teknologi informasi [8]. Namun kenyataannya, tidak semua penggunanya melibatkan diri secara aktif dalam proses analisis kebutuhan suatu proyek karena berbagai alasan, termasuk dalam memberikan *feedback* terkait pengalaman yang dirasakan pengguna setelah

menggunakan suatu aplikasi atau *software*. Hal ini terutama terjadi pada proyek yang melibatkan banyak pengguna/*stakeholder* yang bervariasi, seperti proyek-proyek yang biasa dilakukan oleh *software producing organization* (SPO) dengan *multiple clients*, mencocokkan kebutuhan para pengguna dengan visi dan *road map* suatu produk semakin menambah kesulitan yang ada [9].

Konsep gamifikasi tumbuh dan berkembang sebagai suatu tren yang menjanjikan dalam berbagai area. Penelitian yang dilakukan M2 Research pada tahun 2011 memprediksi bahwa pada pasar gamifikasi akan mencapai 2.8 miliar dolar Amerika pada tahun 2016 [10]. Penggunaan elemen-elemen *game design* pada aplikasi-aplikasi maupun proses dalam cakupan *non-game* dianggap dapat meningkatkan partisipasi pengguna dan juga pada akhirnya meningkatkan kualitas suatu produk [11]. Kontribusi pertama dari penelitian ini yaitu adalah untuk mengajukan ide bahwa implementasi konsep gamifikasi dapat meningkatkan partisipasi pengguna dalam memberikan *feedback* terhadap suatu aplikasi yang dapat menjadi sumber informasi terkait munculnya *requirements* baru atau kesalahan-kesalahan yang terjadi pada analisis kebutuhan yang telah dilakukan sebelumnya.

Namun sekedar mendapatkan *feedback* pengguna saja tidak langsung dapat menyelesaikan semua masalah, karena walaupun motivasinya masih harus ditingkatkan, tapi jumlah *feedback* dari pengguna ini bisa datang dalam jumlah yang cukup besar dan untuk mengklasifikasi lalu mengekstrak *requirements* dari sana secara manual dapat memakan waktu. Selain itu, tidak semua *feedback* dari pengguna dapat dinilai berguna dan berkualitas bagi pengembangan proyek seperti *spam* atau informasi yang sama sekali tidak ada kaitannya dengan aplikasi. Salah satu algoritma *machine learning* yang dapat digunakan untuk membantu proses klasifikasi *user feedback* adalah Naïve Bayes.

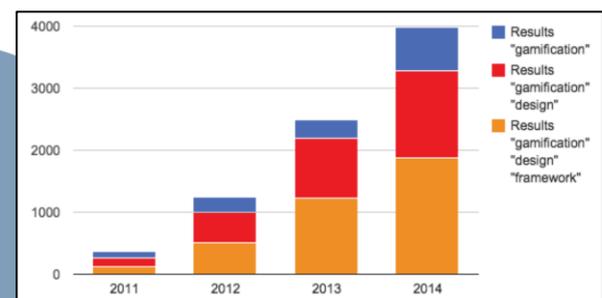
Dalam penelitian sebelumnya [12], telah dilakukan klasifikasi *review* aplikasi yang terdapat di Google Play dan Apple Stores, namun belum ada kajian terkait penggunaan gamifikasi untuk meningkatkan motivasi pengguna untuk memberikan *feedback* yang berkualitas. Berdasarkan hasil dari penelitian tersebut pula, Naïve Bayes merupakan algoritma klasifikasi yang tepat karena dapat mencapai tingkat akurasi yang tinggi dengan *training set* yang relatif kecil serta memiliki kecepatan yang tinggi dibandingkan algoritma lain yang dikaji pada penelitiannya yaitu Maximum Entropy dan Decision Tree Learning [12]. Adapun kontribusi berikutnya dari penelitian ini adalah untuk mengintegrasikan *automatic app reviews classifier* menggunakan Naïve Bayes ke dalam model yang diajukan.

Selanjutnya, karya ilmiah ini terstruktur sebagai berikut. Bab II akan menjelaskan tentang gamifikasi

beserta beberapa literatur terkait. Bab III menjelaskan mengenai klasifikasi *feedback* pengguna menggunakan Naïve Bayes, bab IV menjabarkan model yang diajukan pada penelitian ini, dan bab V menyajikan kesimpulan dari penelitian yang sedang dijalankan serta tahapan-tahapan berikutnya yang dapat dilakukan untuk mengembangkan penelitian ini.

II. GAMIFIKASI

Gamifikasi merupakan metode yang belakangan ini mendapatkan banyak sorotan di dunia teknologi. Gambar 1 di bawah menunjukkan hasil pencarian terhadap penelitian akademis seputar gamifikasi beserta perkembangannya dari tahun 2011 sampai 2014. Dapat dilihat bahwa jumlah penelitian akademis mengenai “gamifikasi” beserta “design” dan “framework” terkait gamifikasi meningkat dengan cukup pesat dari tahun ke tahun.



Gambar 1. Hasil Pencarian Tentang Gamifikasi pada Penelitian Akademis [13]

Gamifikasi didefinisikan sebagai penggunaan elemen-elemen pada *game design* pada konteks non-game [14]. Walau begitu, tujuan penggunaan elemen-elemen *design* dalam gamifikasi agak berbeda dengan *game* pada umumnya. Pada gamifikasi, penggunaan elemen-elemen *game design* lebih ditujukan untuk meningkatkan *user engagement* pada konteks sistem yang sudah ada. Sedangkan, pada *game*, penggunaan elemen-elemen umumnya ditujukan murni untuk *entertainment*. Oleh karena itu, perancangan sistem yang digamifikasi tidak bisa disamakan dengan proyek pembuatan *game*.

Dalam merancang dan membangun suatu sistem yang tergamifikasi, terdapat beberapa *framework* yang dapat digunakan sebagai acuan dalam membuat suatu sistem yang digamifikasi. Pada penelitian yang dilakukan Mora dkk. [13], setelah mengkaji 18 *framework* yang ada, ditemukan bahwa ada 10 *game design items* pada gamifikasi yang dianggap “*the ten most meaningful (in terms of results and heterogeneity)*”. Adapun sepuluh hal tersebut adalah sebagai berikut.

1. *Viability*: studi pendahuluan, evaluasi dan analisis potensi untuk mengaplikasikan gamifikasi pada suatu sistem.
2. *Stakeholders*: teknik yang digunakan untuk mengidentifikasi dan berfokus pada pihak-

- pihak yang berinteraksi pada proses perancangan.
3. *Loop*: mekanika game yang digabung dengan *reinforcement* dan *feedback* yang digunakan untuk menarik pengguna ke dalam *key system actions*.
 4. *Endgame*: akhir permainan yang sudah ditentukan di awal, biasanya menarik pengguna untuk menggunakan seluruh kemampuannya.
 5. *On-boarding*: cara memulai pemakaian sistem bagi partisipan baru.
 6. *Rules*: aturan main yang ditentukan oleh *system designer*.
 7. *Metrics*: standar pengukuran efisiensi, performa, progres, proses, ataupun kualitas.
 8. *Ethics*: cabang filosofi yang melibatkan mensistematisasi, mempertahankan, dan merekomendasikan konsep benar dan salah.
 9. *UI/UX*: segala hal yang dirancang ke dalam sistem yang digamifikasi, termasuk interaksi, perilaku, sikap, dan emosi pemain.
 10. *Technology*: komponen *software* yang diperlukan dan digunakan dalam pembangunan sistem.

Dari 18 *framework* gamifikasi yang dikaji, terdapat dua *framework* yang di dalamnya membahas kesepuluh *item* yang disebutkan di atas, baik secara implisit maupun eksplisit, sedangkan pada 16 *framework* lainnya tidak semua *item* tersebut dibahas. Salah satunya adalah *framework* yang diajukan oleh Werbach dan Hunter, yang dinamakan Six Steps to Gamification [15]. Adapun keenam langkah tersebut adalah sebagai berikut.

1. *DEFINE business objectives*. Salah satu tujuan dari gamifikasi adalah untuk menciptakan sesuatu yang disukai orang namun itu bukanlah tujuan satu-satunya dari gamifikasi. *Business objective* yang ingin dikaji harus didefinisikan dengan jelas. Bila pengguna menyukai sistem yang dikembangkan namun tidak meningkatkan produktivitas, efisiensi, maupun kesejahteraan di lingkungan kerja, maka hasil implementasi gamifikasi dapat melenceng dari tujuan awalnya.
2. *DELINEATE target behavior*. Langkah berikutnya adalah jabarkan secara konkrit perilaku atau *behavior* apa yang diinginkan dari target pengguna sistem untuk mencapai tujuan awal atau *business objective* yang dikaji.
3. *DESCRIBE your players*. Werbach pada *interview* dengan pihak dari Engaging Leader [16], menyebutkan bahwa penting untuk menganggap pengguna sebagai *players/pemain*, bukan sekedar pengguna atau

karyawan. Pemain ini yang menggunakan sistem yang dibuat secara sukarela, dan perlu merasakan kesenangan dan hasrat untuk berkembang dalam sistem yang dibuat. Ada banyak tipe pemain yang dapat dimotivasi oleh berbagai hal, seperti kompetisi atau tantangan, dan ada juga yang termotivasi karena sosialisasi yang dapat dilakukan pada sistem. Penting untuk mengetahui tipe pemain untuk merancang sistem dan menggunakan elemen-elemen yang tepat untuk meningkatkan partisipasi dan motivasi pemain pada sistem yang digamifikasi.

4. *DEVISE activity cycles*. Merancang bagaimana sistem yang digamifikasi bekerja. Dibagi menjadi dua level yaitu mikro dan makro. Pada level mikro, harus dipikirkan bagaimana hal-hal sederhana seperti *feedback* dapat memotivasi pengguna untuk terus bermain. Sedangkan pada level makro, lebih difokuskan pada yang disebut dengan "Player Journey", yang adalah perjalanan pemain dari awal sebagai pemula yang lalu terus berkembang dan tetap termotivasi untuk bermain sampai ke tahap dimana pemain dapat dianggap sukses.
5. *DON'T forget the fun!* Langkah selanjutnya adalah dimana *developer* harus berhenti memikirkan hal teknis sejenak, agar tidak melupakan *fun* dalam permainan. *System designer* harus memikirkan pertanyaan-pertanyaan seperti "All right, would people actually find this fun and engaging? If not, what can I do to make it feel more fun". Namun tetap harus dilakukan tanpa melupakan struktur, hal teknis, dan tujuan awal sistem digamifikasikan.
6. *DEPLOY the appropriate tools*. Pada tahap ini, cari dan tentukan tools yang dibutuhkan untuk merancang dan membangun sistem yang digamifikasi. Baik itu software tools maupun mekanisme-mekanisme sederhana dalam game. Tentukan elemen-elemen game (*dynamics, mechanics, dan components*) yang ingin digunakan, lalu bangun sistemnya.

Dalam membangun sistem yang digamifikasi, keenam langkah tersebut tidak hanya cukup dilakukan dalam satu iterasi dan sistem selesai dibangun. *Developer* harus memperhatikan *feedback* yang didapat dari pengguna dan memperbaiki sistem yang ada sesuai kebutuhan pengguna atau menambahkan fitur-fitur yang dirasa perlu.

III. KLASIFIKASI FEEDBACK PENGGUNA DENGAN NAÏVE BAYES

Pada penelitiannya, Maalej dan Nabil [12] mengimplementasikan tiga algoritma klasifikasi *machine learning* yaitu Naïve Bayes, Maximum Entropy, dan Decision Tree untuk mengklasifikasi *user review* dari suatu aplikasi yang dapat ditemukan di

Google Play Store atau Apple App Store. *User reviews* yang didapat kemudian diklasifikasikan ke dalam empat kategori yaitu *bug reports*, *feature request*, *user experiences*, dan *ratings* [12].

Bug reports menjelaskan masalah dengan aplikasi terkait yang harus diperbaiki, seperti *crash*, perilaku sistem yang tidak diinginkan, atau isu-isu terkait performa aplikasi. Pada kategori *feature request*, pengguna menanyakan fungsionalitas atau konten yang dianggap kurang atau belum ada atau yang dianggap baik dan patut ditambahkan pada *update* berikutnya. *User experiences* menggambarkan pengalaman yang dirasakan pengguna ketika menggunakan aplikasi dan fitur-fiturnya pada kondisi-kondisi tertentu. Dan yang terakhir adalah *ratings*, yang merupakan representasi dari jumlah bintang yang diterima aplikasi.

Dari hasil penelitian yang dilakukan, disimpulkan untuk mengklasifikasi *user review* ke dalam empat kategori di atas, dapat dikatakan bahwa Naïve Bayes merupakan algoritma yang lebih tepat dibanding algoritma *classifier* lainnya karena dapat mencapai tingkat akurasi yang tinggi dengan ukuran *training set* yang kecil, dengan waktu dua kali lebih cepat dibanding algoritma lainnya [12]. Adapun klasifikasi dengan Naïve Bayes dilakukan setelah data *user review* diperoleh, dikategorikan berdasarkan beberapa *keyword* yang sudah dipilih sebelumnya, dan kalimat-kalimatnya diproses dengan menggunakan teknik *natural language processing (NLP)* seperti *stopword removal*, *stemming*, *lemmatization*, *tense detection*, dan *bigrams*, yang berada di luar cakupan penelitian ini.

Setelah kalimat diproses menjadi *bag of words*, dengan algoritma Naïve Bayes Classifier, setiap fitur dipertimbangkan dalam menentukan kelas/*label* apa saja yang dapat diberikan terhadap suatu input. Untuk menentukan *label* yang diberikan, Naïve Bayes menghitung probabilitas awal dari setiap *label*, yang bisa didapat dari pengecekan frekuensi kemunculan *label* tersebut pada *training set* [17].

Naïve Bayes merupakan algoritma yang sangat populer sebagai *binary classifier* [17], yang dapat menentukan apakah suatu input termasuk ke dalam satu kategori atau tidak. Dalam klasifikasi ke dalam lima kategori sebagai contoh, suatu *user feedback* dapat tergolong ke dalam lebih dari satu kategori, dan pada penelitian ini, suatu input yang berupa *feedback* dapat diklasifikasi secara biner (termasuk dalam satu kategori atau tidak) sebanyak lima kali. Model yang digunakan pada penelitian ini adalah *multinomial document model*, dimana tidak seperti *Bernoulli document model*, frekuensi kemunculan suatu kata dalam suatu dokumen dianggap penting [18]. Selain itu, pada dokumen dengan ukuran *vocabulary* yang besar tingkat kesalahan dapat dikurangi hingga 27% dengan menggunakan *multinomial model* jika dibandingkan dengan *multi-variate Bernoulli model* [19].

Hal pertama yang harus dilakukan untuk mengklasifikasikan suatu dokumen dengan Naïve Bayes Classifier adalah membuat suatu *training set* yang akan dijadikan acuan dalam menentukan probabilitas dokumen baru yang masuk nantinya. Setelah *training set* tersedia, maka dapat dilakukan perhitungan *probability* dengan *multinomial text classification model* sebagai berikut [18].

- Tentukan dan kumpulkan daftar seluruh kata yang ada dan dipertimbangkan (V)
- Hitung hal-hal dibawah ini:
 - N : jumlah total seluruh dokumen (*feedback* dari pengguna) yang ada
 - N_k : jumlah dokumen yang diklasifikasikan ke dalam suatu kategori k , untuk setiap kategori yang ada
 - x_{it} : frekuensi kemunculan suatu kata w_i dalam suatu dokumen D_i , untuk setiap kata yang ada dalam V – dengan kata lain, hitung $n_k(w_i)$: frekuensi kemunculan kata w_i dalam dokumen dengan kategori k
- Hitung probabilitas kemunculan suatu kata (w_i) dalam semua dokumen yang termasuk dalam suatu kategori (C_k) dengan juga menerapkan *Laplace's law of succession* untuk menghindari *zero probability problem* [18], menggunakan rumus (1) atau (2) di bawah ini, dimana z_{ik} merupakan variabel indikator yang bernilai 1 ketika suatu dokumen termasuk dalam satu kategori, dan bernilai 0 ketika tidak.

$$P(w_t|C_k) = \frac{1 + \sum_{i=1}^N x_{it} z_{ik}}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^N x_{is} z_{ik}} \quad (1)$$

$$P(w_t|C_k) = \frac{1 + n_k(w_t)}{|V| + \sum_{s=1}^{|V|} n_k(w_s)} \quad (2)$$

- Hitung probabilitas terklasifikasinya suatu dokumen ke dalam suatu kategori $P(C_k)$ dengan menggunakan rumus (3) di bawah ini.

$$P(C_k) = \frac{N_k}{N} \quad (3)$$

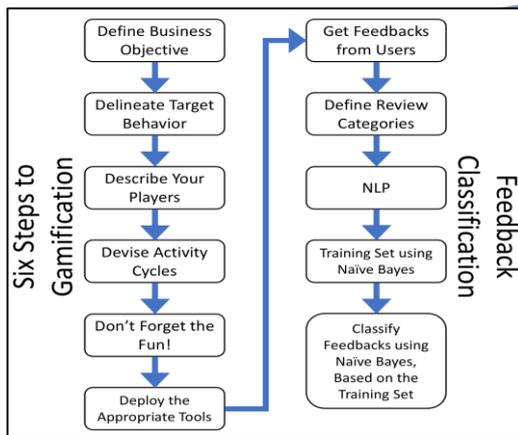
Langkah berikutnya adalah untuk menerima dokumen baru yang belum diklasifikasikan (D), lalu akan diklasifikasikan ke masing-masing kategori yang telah ditentukan berikutnya. Berikut rumus (4) yang menampilkan formula untuk menghitung probabilitas suatu dokumen/*feedback* untuk diklasifikasikan ke dalam suatu kategori. Dimana x adalah frekuensi kemunculan suatu kata dalam dokumen D .

$$P(C_k|D) = \frac{P(C_k|x)}{\alpha P(x|C_k)P(C_k)} \quad (4)$$

$$\alpha P(C_k) \prod_{j=1}^{|V|} P(w_t|C_k)^{x_t}$$

IV. MODEL UNTUK EKSTRAKSI KEBUTUHAN APLIKASI DENGAN GAMIFIKASI DAN NAÏVE BAYES

Berdasarkan penelitian-penelitian yang telah disebutkan sebelumnya, dapat dilihat bahwa *requirements evolution* merupakan suatu hal yang nyata dan terjadi pada suatu proyek pengembangan sistem, dan salah satu masukan yang berguna bagi *developer* adalah *feedback* yang didapat dari penggunaannya. Penelitian ini memperkenalkan suatu model yang dianggap mampu meningkatkan motivasi pengguna untuk memberikan *feedback* yang berguna bagi *developer* dan juga cara untuk mengklasifikasikan *feedback* tersebut sehingga dapat mempermudah *developer* untuk menyimpulkan fitur-fitur mana saja yang harus dikembangkan, *bugs and errors* yang harus diperbaiki, dan lain sebagainya.



Gambar 2. Model Ekstraksi Kebutuhan Aplikasi

Model yang diajukan dibagi menjadi dua bagian secara garis besar seperti yang dapat dilihat pada Gambar 2. Bagian pertama adalah implementasi gamifikasi untuk meningkatkan partisipasi pengguna untuk memberikan *feedback* yang berguna untuk pengembangan produk ke depannya, yang menerapkan Six Steps to Gamification, *framework* gamifikasi yang diajukan oleh Werbach dan Hunter [15]. Lalu pada bagian selanjutnya, setelah *feedback* didapatkan, maka *developer* harus menentukan kategori-kategori *feedback* sesuai dengan kebutuhannya, lalu *feedback* yang didapat diproses dengan NLP, yang lalu bila belum ada *training set* yang dapat dijadikan acuan bagi Naïve Bayes untuk menentukan kategori *feedback*, maka *training set* tersebut harus dibuat terlebih dahulu, dan akhirnya *developer* dapat menggunakan Naïve Bayes untuk mengklasifikasikan *feedback* yang datang berikutnya.

Dalam mengimplementasikan gamifikasi sesuai dengan model diatas, berikut penjelasan keenam panduan berdasarkan Six Steps to Gamification [15].

1. Define Business Objective

Pada tahap ini, pihak pengembang aplikasi harus mendefinisikan secara jelas apa objektif yang ingin dicapai dengan model ini. Secara umum, objektif dari gamifikasi pada model yang diajukan adalah untuk mendapatkan informasi terkait evolusi kebutuhan berdasarkan masukan dari pengguna yang didapat dari *feedback*. Maka elemen-elemen gamifikasi yang digunakan pun harus searah agar dapat memotivasi pengguna untuk turut serta berpartisipasi dalam memberikan *feedback* yang berguna demi mendukung agar tujuan tersebut dapat tercapai.

2. Delineate Target Behavior

Untuk mencapai objektif yang telah disebutkan di atas yaitu untuk mendapatkan *feedback* yang berguna sebagai masukan terkait kebutuhan aplikasi yang harus dikembangkan, pengguna diharapkan untuk dapat secara aktif berpartisipasi memberikan *feedback* yang berarti, sesuai kebutuhan *developer*. Panduan terkait syarat-syarat agar suatu *feedback* menjadi berarti dapat diinformasikan kepada pengguna, penggunaan elemen-elemen gamifikasi yang adapun harus dijelaskan sehingga perilaku pengguna dapat disesuaikan dengan tujuan diimplementasikannya gamifikasi untuk ekstraksi kebutuhan.

3. Describe Your Players

Tidak semua pengguna memiliki karakteristik yang sama. Sesuai dengan tipenya, elemen-elemen gamifikasi yang baik digunakan pada sistem bisa berbeda. Gambar 3 menampilkan tipe-tipe pengguna yang harus dipertimbangkan dalam membuat suatu sistem yang digamifikasi (*gamified system*). Menurut Marczewski [20], secara umum pengguna dapat dikategorikan ke dalam 6 tipe yaitu Philanthropist, Socialiser, Player, Achiever, Free Spirit, dan Disruptor.



Gambar 3. Tipe-Tipe Pengguna [20]

Philanthropist termotivasi oleh tujuan dan arti dari aktivitas yang dijalankannya. Pengguna dengan tipe ini ingin membantu orang lain, tidak egois, dan ingin dapat menjadi bermanfaat untuk sesuatu tanpa mengharapkan imbal balik. **Socialiser** termotivasi oleh keterhubungan pengguna dengan pengguna lainnya, suka berinteraksi dan membuka koneksi social yang baru. **Player** dapat dimotivasi dengan *reward* atau

hadiah. Mereka secara sukarela turut berpartisipasi dalam permainan karena tertarik dengan *reward* yang bisa didapatkan dari suatu aktivitas.

Achiever dapat termotivasi dengan keahlian dari suatu aktivitas. Tipe ini sangat tertarik dengan hal baru dan aktivitas yang dapat dilakukan untuk mengembangkan diri mereka dan bahkan menjadi ahli di suatu bidang. **Free Spirit** suka mencoba sesuatu yang baru dan bereksplorasi, tipe ini dapat dimotivasi dengan adanya otonomi dan kebebasan mengekspresikan diri sendiri. Yang terakhir adalah **Disruptor**, tipe ini dapat dimotivasi dengan perubahan dan kemampuan mereka untuk menjatuhkan sistem yang ada baik secara positif maupun negatif [20].

Setiap tipe memiliki karakteristiknya masing-masing yang bisa disesuaikan dengan kebutuhan sistem dan elemen-elemen gamifikasi yang digunakan untuk meningkatkan motivasi pengguna untuk menggunakan sistem. Seperti contohnya bagi pemain dengan tipe Player, elemen gamifikasi seperti *points* dan *leaderboard* dapat digunakan untuk menarik pemain untuk berpartisipasi dalam sistem karena sifat Player yang menginginkan *reward* atas usaha yang telah dilakukannya. Berbeda dengan Socialiser, contohnya, yang mungkin dapat tertarik dengan fitur yang memungkinkan mereka untuk dapat bersosialisasi dalam sistem dan berkolaborasi dengan pengguna lainnya.

Marczewski menyajikan hasil dari *user type test* yang telah dilakukan terhadap lebih dari 10.000 orang menunjukkan bahwa Philanthropist dan Free Spirit merupakan 2 tipe yang paling populer dengan 20% dan 19% dari total pengguna yang telah melakukan *test* tersebut [21]. Namun, hasil yang disajikan tersebut tidak dapat digunakan langsung pada proyek yang ingin dilakukan karena pada akhirnya, setiap proyek bisa memiliki tipe-tipe pengguna yang berbeda dan *developer* harus mengetahui tipe-tipe calon pengguna dari sistem yang dibangun.

4. Devise Activity Cycles

Pada suatu permainan, biasanya selalu memiliki awal dan kemungkinan juga memiliki akhir. Namun, di antara kedua hal tersebut, terdapat banyak kondisi yang berulang dan bercabang. Ada dua jenis *cycles* atau *loops* yang harus dipertimbangkan dalam gamifikasi, yaitu *engagement loops (micro level)* dan *progression stairs (macro level)* [15].

Pada level mikro, aksi yang dilakukan pengguna berangkat dari motivasi awal untuk berpartisipasi dalam sistem, lalu harus dilanjutkan dengan *feedback* yang berupa respon dari sistem. *Feedback* (dari sistem) merupakan cara efektif untuk memotivasi pengguna untuk terus menggunakan sistem, yang dalam hal ini adalah untuk memberikan masukan terkait kebutuhan-kebutuhan yang berubah atau bertambah. *Feedback* sistem dapat bermacam-macam bentuknya, bahkan *rewards* seperti poin dan *achievements* pun merupakan *feedback* dari sistem atas aksi yang dilakukan pengguna.

Sedangkan pada level makro, sistem harus dirancang dan dibangun dengan memikirkan progres yang dapat dirasakan pengguna ketika sistem digunakan. Salah satu contoh elemen yang dapat digunakan untuk menunjukkan progres pengguna adalah *level* dan *difficulty*. Semakin lama pengguna berpartisipasi, maka *levelnya* semakin tinggi dan semakin sulit dan menantang pula hal yang harus dilakukan.

5. Don't Forget the Fun

Hal yang tidak boleh dilupakan ketika merancang suatu sistem dengan gamifikasi adalah untuk berhenti sejenak dan berpikir, apakah sistem yang dirancang itu menyenangkan? Banyak pertanyaan-pertanyaan yang dapat ditanyakan oleh *developer* saat ada di tahap ini. Salah satu pertanyaan yang dapat dipikirkan adalah, bila tidak ada *rewards* secara ekstrinsik yang didapatkan pengguna, bila pengguna tidak diminta untuk menggunakan sistem, apakah pengguna akan ingin menggunakan sistem hanya karena sistem tersebut menarik dan dirasa menyenangkan? Bila tidak, maka ada hal-hal yang harus diperbaiki untuk meningkatkan rasa *fun* tersebut.

6. Deploy the Appropriate Tools

Tahap terakhir dari bagian pertama model yang diajukan pada penelitian ini adalah untuk menentukan mekanika dan komponen yang akan digunakan pada sistem yang digamifikasi, dan implementasi sesuai dengan perancangan yang telah dibuat sebelumnya.

Setelah keenam langkah di atas dilakukan, sistem yang telah digamifikasi selesai dirancang dan dibangun, dan elemen-elemen yang digunakan pun telah terimplementasi ke dalam sistem. Setelah sistem yang telah digamifikasi ini dirilis ke publik dan pengguna secara aktif berpartisipasi untuk memberikan *feedback* terkait produk yang dikaji untuk mengetahui kebutuhan-kebutuhan yang berevolusi atau yang masih kurang pada produk yang dibuat sebelumnya, saatnya beranjak ke bagian kedua dari model yang diajukan, yaitu mengklasifikasikan *feedback* tersebut.

7. Get Feedback from Users

Feedback yang didapat dari pengguna lalu disimpan dan diproses – bisa menggunakan NLP – untuk mengekstrak kata-kata yang memang relevan dan dapat digunakan. Output dari tahap ini adalah *bag of words*, atau kumpulan kata-kata dari setiap *feedback* dari pengguna yang nantinya akan digunakan untuk membuat training set dan masukan yang akan diklasifikasi.

8. Define Review Categories

Untuk mengklasifikasikan *feedback* ke dalam beberapa kategori, *developer* harus menentukan terlebih dahulu kategori tersebut. Pada penelitian yang dilakukan sebelumnya, *review* pengguna yang didapatkan dari pasar aplikasi diklasifikasikan ke dalam empat kategori yaitu *bug reports*, *feature request*, *user experiences*, dan *ratings* [12]. Namun

tentu saja kategori-kategori tersebut dapat berbeda sesuai kebutuhan *developer*.

9. Define the Training Sets

Pada tahap ini, *feedback* dari pengguna sudah didapatkan dan diproses dan kategori yang diinginkan pun sudah ditentukan. Untuk membentuk *training set*, *developer* secara manual memilah-milah suatu set dokumen dan lalu mengklasifikasikannya ke dalam kategori-kategori yang ada.

10. Classify the Feedbacks

Setelah *training set* dibentuk dan probabilitas-probabilitas yang dibutuhkan sudah dihitung, maka langkah selanjutnya adalah untuk melakukan perhitungan probabilitas suatu dokumen baru termasuk ke dalam satu atau lebih kategori yang telah ditentukan sebelumnya, dengan menggunakan formula yang tercatat di rumus (4).

V. SIMPULAN

Pada penelitian ini, diajukan sebuah model yang dapat digunakan untuk meningkatkan motivasi pengguna agar turut berpartisipasi secara aktif dalam memberikan masukan berupa *user feedback* yang berguna sebagai *requirements* yang dapat digunakan *developer* untuk mengembangkan suatu produk lebih lanjut ke depannya dengan menggunakan *framework* gamifikasi yang dinamakan Six Steps to Gamification yang dipublikasi oleh Werbach dan Hunter [15]. Selain itu, model pada penelitian ini juga mengimplementasikan Naïve Bayes dengan *multinomial document model* sebagai *text classifier* yang dapat membantu *developer* dalam mengklasifikasikan *feedback* dari pengguna yang banyak jumlahnya.

Dengan klasifikasi yang dibantu menggunakan Naïve Bayes Classifier, *developer* diharapkan dapat menghemat waktu dan bekerja lebih efisien dalam mengkaji kebutuhan-kebutuhan suatu produk yang berubah, berkembang, atau bertambah, dan juga dapat membantu proses *requirements prioritization* berdasarkan jumlah dan urgensi yang dapat diketahui berdasarkan *feedback* pengguna.

Langkah selanjutnya yang dapat dilakukan adalah untuk mengimplementasikan model yang telah diajukan dan merancang bangun suatu *feedback system* untuk memotivasi pengguna dan mengklasifikasikan *feedback* yang didapat sesuai dengan kategori yang ditentukan sebelumnya oleh *developer*. Setelah itu, model dapat diuji dan dievaluasi keefektifannya. Selain itu, penelitian dari segi natural language processing pun dapat digali lebih dalam untuk menentukan proses-proses yang lebih baik dalam mengolah *feedback* dari pengguna.

VI. DAFTAR PUSTAKA

- [1] A. v. Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specification*, Chichester: John Wiley & Sons, 2009.
- [2] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," Rio de Janeiro, 2013.
- [3] C. Jacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," San Francisco, 2013.
- [4] L. V. G. Carreno and K. Winbladh, "Analysis of user comments: An approach for software requirements evolution," San Francisco, 2013.
- [5] S. Kujala, "User Involvement: A Review of the Benefits and Challenges," 2003.
- [6] K. E. Emam, S. Quintin and N. H. Madhavji, "User participation in the requirements engineering process: An empirical study," 1996.
- [7] L. Damodaran, "User involvement in the systems design process-a practical guide for users," 1996.
- [8] The Standish Group, "CHAOS Summary 2009: The 10 Laws of CHAOS," The Standish Group International, Incorporated, 2009.
- [9] G. Lucassen, F. Dalpiaz, J. M. van der Werf and S. Brikkemper, "Bridging the Twin Peaks: the Case of the Software Industry," Florence, 2015.
- [10] S. Peterson, "gamesindustry.biz," 2012. [Media perantara]. Alamat situs: <https://www.gamesindustry.biz/articles/2012-05-21-gamification-market-to-reach-usd2-8-billion-in-2016>. [Diakses 23 March 2018].
- [11] C. Eickhoff, C. G. Harris, A. P. de Vries and P. Srinivasan, "Quality through flow and immersion: gamifying crowdsourced relevance assessments," Portland, 2012.
- [12] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? On automatically classifying app reviews," Ottawa, 2015.
- [13] A. Mora, D. Riera, C. Gonzales and J. Arnedo-Moreno, "A Literature Review of Gamification Design Frameworks," Skovde, 2015.
- [14] S. Deterding, D. Dixon, R. Khaled and L. E. Nacke, "Gamification: Toward a Definition," Vancouver, 2011.
- [15] K. Werbach and D. Hunter, *For the Win: How Game Thinking Can Revolutionize Your Business*, Wharton Digital Press, 2012.
- [16] K. Werbach, "Engaging Leader," 2013. [Media perantara]. Alamat situs: <http://www.engagingleader.com/6-steps-to-effective-gamification-transcript/>. [Diakses 26 March 2018].
- [17] S. Bird, E. Klein and E. Loper, *Natural Language Processing with Python*, First Edition ed., Sebastopol: O'Reilly Media Inc., 2009.
- [18] H. Shimodaira, "Text Classification using Naive Bayes," Edinburgh, 2018.
- [19] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI-98 workshop on learning for text categorization*, hal. 41-48, 1998.
- [20] A. Marczewski, "Gamified UK," 2015. [Media perantara]. Alamat situs: <https://www.gamified.uk/user-types/>. [Diakses 26 March 2018].
- [21] A. Marczewski, "Gamified UK," 2016. [Media perantara]. Alamat situs: <https://gamified.uk/UserTypeTest2016/user-type-test-results.php?lid=#.Wrs2kWaB3Bl>. [Diakses 26 March 2018]