

# Pengembangan Fitur Notifikasi pada *Website Application Comic Strip rupi.co* Menggunakan Metode *Agile*

Moh. Roziq Bahtiar<sup>1</sup> dan Anggraini Mulwinda<sup>2</sup>

*Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang*  
*hi@roziqbahtiar.com<sup>1</sup>, anggrainimulwinda@mail.unnes.ac.id<sup>2</sup>*

**Abstrak**— *Rupi.co* merupakan website yang menjadi wadah bagi para kartunis untuk menyajikan karya berupa komik strip. Sejak diluncurkan 1 Januari - November 2015 jumlah *user* hanya 1620. Hal ini dirasa kurang memenuhi target pertumbuhan *user*. Pada *Rupi* hanya terdapat beberapa fitur diantaranya, *New, Popular, Top User, Upload*. Selain itu, terdapat beberapa fitur yang masih belum dikembangkan diantaranya notifikasi, *message*, dan komentar. *Rupi* juga belum menggunakan metode pengembangan *software*. Oleh karena itu, penulis melakukan studi pendahuluan awal menggunakan kuesioner. Hasilnya menunjukkan 21 dari 22 responden menginginkan adanya pengembangan fitur notifikasi. Penelitian ini bertujuan untuk mengetahui bagaimana implementasi *Agile Software Development* pada pengembangan fitur notifikasi di *Rupi.co*. Selain itu, untuk mengetahui pengalaman pengguna setelah berinteraksi dengan fitur notifikasi berdasarkan analisis *User Experience*. Metode penelitian yang digunakan adalah *Research and Development*. Pertama, studi pendahuluan bertujuan untuk mengungkapkan fakta dengan studi pustaka. Tahap pengembangan yaitu implementasi *Agile Scrum* pada fitur notifikasi. Pada tahap evaluasi, pengujian dengan *Black Box Testing* dan analisis *User Experience*. Hasil penelitian menunjukkan bahwa *Metode Agile Software Development* dapat diimplementasikan pada pengembangan fitur notifikasi di *Web Application Rupi.co*. Pengujian *Black box* menunjukkan hasil implementasi berjalan baik secara fungsional seperti yang direncanakan. Hasil analisis *User Experience*, menunjukkan semua kategori tidak ada yang di bawah rata-rata.

**Kata kunci**— *Metode Agile, Software Engineering, Software Development Methodology*

## I. PENDAHULUAN

*Rupi* merupakan website yang menjadi wadah bagi para kartunis dan komikus lokal untuk menyajikan karya mereka berupa komik strip yang menghibur. Terdapat satu syarat bagi para pengguna yang meng-upload komik strip, yakni mereka harus mengisi kolom *Source* yang berisikan informasi tentang komik tersebut. Baik nama si pembuat atau *link* web pembuat komik pada *form upload*. Jika mereka tidak mencantumkan sumber maka postingan akan dihapus oleh admin [1]. Sejak diluncurkan pertama kali pada 1 Januari - November 2015 jumlah *user* pada *rupi.co* hanya mencapai 1620. Hal ini dirasa kurang memenuhi target pertumbuhan *user* setiap bulannya.

Di sisi lain, dalam perkembangannya fitur masih terdapat beberapa fitur yang belum ada. Diantaranya seperti fitur notifikasi, *message*, dan komentar yang mungkin diinginkan *user Rupi* untuk terus berkembang mengikuti permintaan *user*. Oleh karena itu, masih membutuhkan validasi produk dan validasi *user* yang dapat memberikan solusi melalui produk tersebut. Penulis melakukan studi pendahuluan awal untuk mengetahui fitur mana yang akan dikembangkan penulis menggunakan kuisoner. Hasilnya menunjukkan 21 dari 22 responden menginginkan adanya pengembangan fitur notifikasi. *Rupi* juga belum pernah menarapkan suatu metode *software engineering* yang sesuai untuk mengembangkan suatu fitur.

*Software* komputer, dikatakan sukses jika memenuhi beberapa kriteria. Pertama, sesuai dengan kebutuhan *user* yang menggunakan *software* tersebut. Selain itu, dapat dijalankan dengan sempurna dalam waktu yang cukup lama. Terakhir, mudah untuk diubah, digunakan, serta diperbaharui untuk menjadi *software* yang lebih baik. Di sisi lain, *software* komputer dikatakan gagal ketika *user* tidak puas dengan *software* tersebut, mudah *error*, dan sulit diubah bahkan sulit untuk digunakan. Oleh karena itu, untuk menghasilkan *software* komputer yang sukses diperlukan *software engineering* [2].

*Software engineering* adalah bidang multidisipliner, hal ini mengakibatkan studi penelitian di bidang ini jauh lebih bervariasi, daripada contohnya dalam ilmu-ilmu pengetahuan alam seperti fisika [3]. Terdapat beberapa metode *software Development* dalam *software engineering* diantaranya *Waterfall, Prototyping Model, Spiral Model, Iterative Incremental Model*, dan *Agile* [4]. Metode tersebut termasuk ke dalam metode pengembangan tradisional [5]. Sedangkan, metode *Agile* merupakan pendekatan yang baru dalam pengembangan *software*. Metode ini mulai diperkenalkan pada akhir tahun 90'an dan sekarang sudah digunakan secara luas sebagai metode utama dalam pengembangan perangkat lunak [6].

Metode ini bertujuan untuk membuat unit *software development* lebih responsif terhadap perubahan. Perubahan tersebut adalah perubahan terhadap pesatnya perkembangan teknologi, perubahan bisnis, dan kebutuhan produk. *Agile Development* pada dasarnya menggunakan dasar pengembangan yang iteratif. Bila dibandingkan dengan metode pengembangan tradisional metode ini lebih mengutamakan sudut pandang pengguna (*people-centric*). Proses dari metode *Agile* ini menggunakan *feedback* dari pengguna sebagai kontrol utama daripada perencanaan dalam hal pengembangan produk. *Feedback* ini diperoleh dari pengujian secara berkala dan rilis dari perkembangan *software* [5].

Rupi merupakan startup yang terdiri dari empat *co-founder*. Terdiri dari satu *co-founder* teknis, satu *co-founder* desain, dan dua *co-founder* bisnis. Rupi masih tergolong baru. Dari segi tampilan situs Rupi masih sederhana. Pada menu bar pengguna dapat melihat menu utama Rupi yang terdiri dari *New, Popular, Top User, Top Post, Upload*, dan kategori komik yang terdapat dalam menu *Subrupi*.

Berdasarkan fakta tersebut, penulis ingin mengembangkan fitur notifikasi dengan suatu metode *software development* yang sesuai. Metode dalam *software engineering* berkembang seiring dengan bergesernya aspek yang mempengaruhi dalam pembuatan *software* itu sendiri. Jika pada tahun 80'an aspek teknis sangat penting, saat ini aspek non teknis justru yang berpengaruh pada pembuatan *software*. Metode *Agile* hadir sebagai sebuah pendekatan *software engineering* informal yang memiliki langkah kerja namun tidak terikat oleh aturan tertentu. Selain itu, metode ini juga dapat diterapkan pada tim berskala kecil. Terdapat prinsip-prinsip yang harus dipatuhi dalam menerapkan metode ini. Diantaranya menyampaikan sedini mungkin dan sesering mungkin kepada pengguna, tidak hanya fokus ke desain dan arsitektur antar muka. Akan tetapi, fokus pada setiap bit pekerjaan. Proses ini secara sederhana sudah meringkas bagaimana praktik *Agile Software Development*.

Oleh karena itu, maka penelitian tentang pengembangan fitur notifikasi pada *website application comic strip* Rupi.co menggunakan Metode *Agile* menarik dan penting dilakukan penelitian. Untuk itu maka peneliti tertarik untuk meneliti masalah ini.

Tujuan dari penelitian ini adalah bagaimana mengimplementasikan *Agile Software Development* pada pengembangan fitur notifikasi website Rupi.co. Selain itu, untuk mengetahui pengalaman pengguna setelah berinteraksi dengan fitur notifikasi berdasarkan analisis *User Experience*.

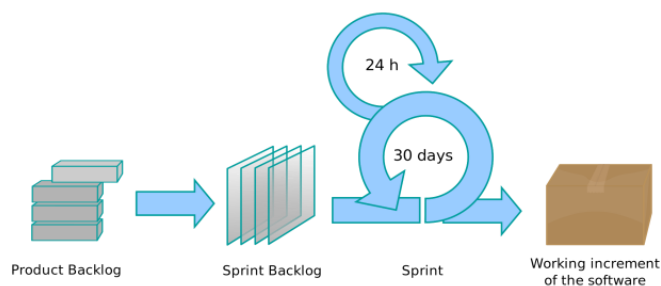
Penelitian ini diharapkan dapat bagi pengguna, dapat mengetahui aktivitas tentang postingan mereka yang sudah di upload pada Rupi.co. Pengguna akan mengetahui siapa saja yang menyukai postingannya yang di *upload* di Rupi.co melalui fitur notifikasi. Selain itu, dapat menambah wawasan mengenai metode *software development* khususnya bagi mahasiswa dengan bidang studi teknik informatika atau yang sejenisnya.

## II. METODE PENELITIAN

Penelitian ini termasuk dalam penelitian *Research and Development* (R & D). Penelitian dan pengembangan juga didefinisikan sebagai suatu metode penelitian yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut [7].

Produk yang dikembangkan dalam penelitian ini adalah website *rupi.co*. Terdapat tiga tahap yang akan dilakukan dalam penelitian *Research and Development*. Tahapan yang akan dilakukan dalam penelitian ini meliputi: studi pendahuluan, pengembangan, dan evaluasi. Pertama, tahapan ini adalah mengumpulkan data dan informasi yang dibutuhkan dalam penelitian, kegiatan yang dilakukan meliputi : (1) Melakukan studi pustaka yaitu mencari referensi dengan cara menelusuri sumber-sumber tulisan yang pernah dibuat sebelumnya terkait dengan pengembangan menggunakan metode *Agile*; (2) Mencari data lapangan tentang fitur apa yang akan di kembangkan dalam penelitian ini berdasarkan pengguna di *Rupi.co*; (3) Melakukan analisis temuan secara deskriptif mengenai tahapan yang harus dikerjakan pada tahap selanjutnya berdasarkan studi pustaka yang dilakukan dengan data temuan di lapangan. Untuk hasil dari studi pendahuluan mengenai fitur apa yang akan dikembangkan dulu pada penelitian ini, maka penulis melakukan survei kepada *user* di *Rupi.co* menggunakan kuesioner. Jumlah responden sebanyak 22 *user* yang terdiri dari 14 *user* aktif dan 8 *user* pasif. Hasil untuk fitur notifikasi menunjukkan semua *user* aktif menginginkan adanya fitur notifikasi. Sedangkan untuk *user* pasif, 7 dari 8 menginginkan adanya fitur notifikasi. Sebanyak 1 *user* pasif tidak menginginkan adanya fitur notifikasi.

Kedua tahap pengembangan, metode pengembangan perangkat lunak yang digunakan yaitu *Scrum*, merupakan salah satu jenis pemodelan yang termasuk ke dalam metode *Agile Software Development*. *Scrum* adalah sebuah kerangka kerja untuk menyelesaikan permasalahan kompleks yang senantiasa berubah. Pada saat yang bersamaan dapat menghasilkan produk bernilai tinggi dengan kreatif dan produktif [8]. Proses yang dilakukan dalam metode ini adalah *Product Backlog, Sprint Backlog, Sprint* dan *Increment*. Hal ini dapat dilihat pada Gambar 1.



Gambar 1. *Scrum process*

Ketiga, tahap evaluasi menggunakan *Black Box Testing* dan analisis *User Experience*. Hal ini dilakukan untuk mengetahui pengalaman menggunakan setelah pengembangan fitur notifikasi.

III. HASIL DAN PEMBAHASAN

*Product Backlog* adalah daftar terurut, dari setiap hal yang berkemungkinan dibutuhkan di dalam produk [8]. Pada penelitian ini prioritas fitur yang dikembangkan adalah notifikasi sesuai pada Tabel I.

TABEL I. *PRODUCT BACKLOG*

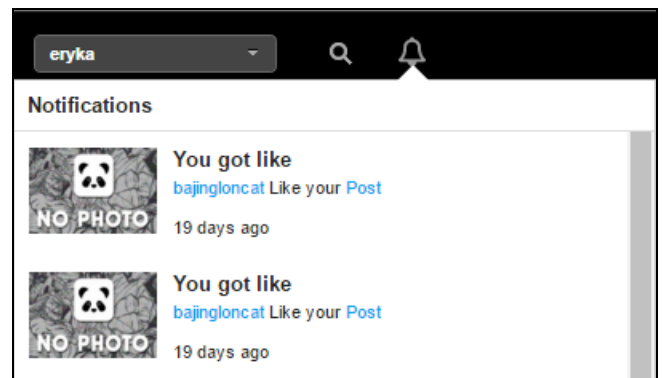
| No | Fitur      | Prioritas                                   |
|----|------------|---|
| 1  | Notifikasi | 14 <i>User</i> aktif menginginkan fitur ini |
| 2  | Komentar   | 12 <i>User</i> aktif menginginkan fitur ini |
| 3  | Pesan      | 11 <i>User</i> aktif menginginkan fitur ini |

*Sprint Backlog* adalah sekumpulan *Product Backlog Item* yang telah dipilih untuk dikerjakan di *Sprint*. Hasil dari penelitian ini ditunjukkan pada Tabel II.

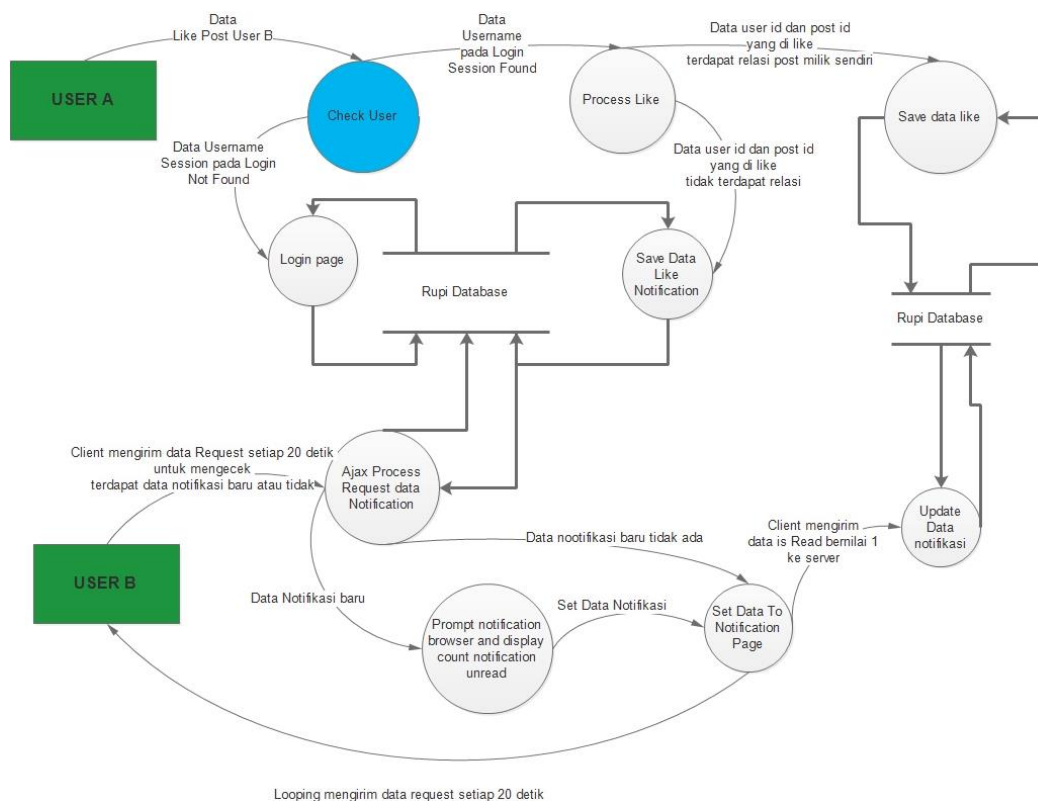
TABEL II. *TIMELINE SPRINT BACKLOG*

|                                | Lama Sprint                     | 2 Minggu        |                          |
|--------------------------------|---------------------------------|-----------------|--------------------------|
|                                | Jumlah Hari Kerja               | 8 Hari          |                          |
| Peran dalam Tim                | Jumlah hari kerja selama sprint | Jumlah jam/hari | Total jam Sprint pertama |
| Manager Project (Scrum Master) | 5                               | 4               | 20                       |
| Programmer 1                   | 8                               | 5               | 40                       |
| Programmer 2                   | 7                               | 5               | 35                       |

*Sprint* merupakan sebuah batasan waktu selama satu bulan atau kurang, di mana sebuah inkremen yang ‘selesai’, berfungsi, dan berpotensi untuk dirilis. (1) *User Interface Design*, pada tahap ini peneliti membuat berbagai macam desain untuk *user interface* fitur notifikasi termuat pada Gambar 2. (2) *Flow system*, pada tahap ini merupakan proses perancangan alur bisnis dari fitur notifikasi yang akan dikembangkan. Secara ringkas termuat dalam Gambar 3. (3) *Database Design*, pada Gambar 4 menunjukkan relasi diagram dari masing-masing entitas objek yang terlibat dalam sistem. *Members* memiliki relasi mendapatkan notifikasi dan *members* dapat melakukan *like* pada setiap postingan yang ada di Rupi.

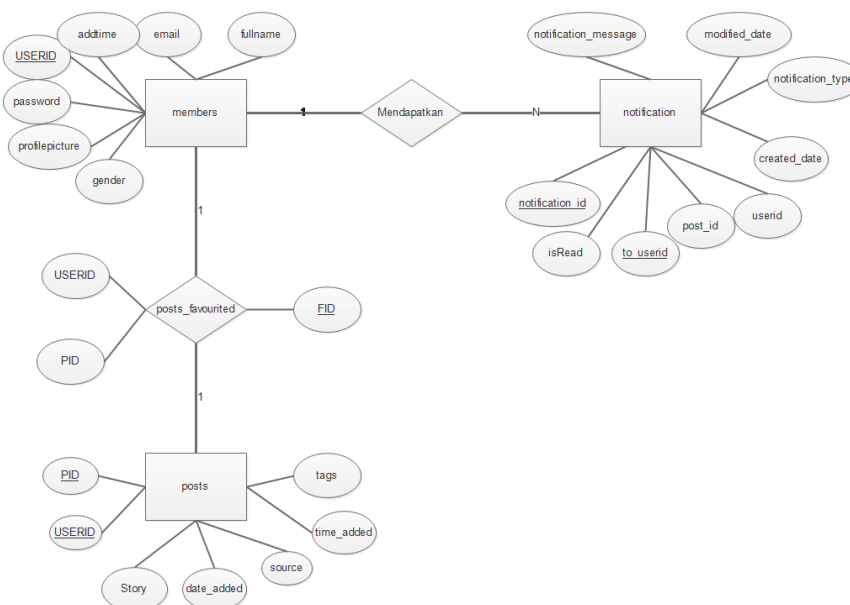


Gambar 2. Desain *user interface* Halaman Notifikasi



Looping mengirim data request setiap 20 detik

Gambar 3. DFD (*Data Flow Diagram*)



Gambar 4. ERD (Entity Relation Diagram)

Berikut ini merupakan hasil transformasi dari desain database yang sudah dirancang pada proses sebelumnya yang menghasilkan database dalam bentuk fisik. Dari perancangan menghasilkan empat buah tabel yang berhubungan dengan sistem notifikasi. (a) tabel notifikasi disajikan pada Gambar 5, (b) tabel posts disajikan pada Gambar 6, (c) tabel posts\_favourited disajikan pada Gambar 7, dan (d) tabel members disajikan pada Gambar 8.

Gambar 5 merupakan bentuk database fisik dari Mysql dengan phpmyadmin. Terdapat primary key notification\_id dan user id sebagai foreign key.

Gambar 6 merupakan bentuk fisik dari tabel posts pada Mysql dengan menggunakan phpmyadmin dengan primary key PID dan field-field lain untuk menyimpan data dari sebuah postingan.

Gambar 7 merupakan bentuk fisik dari tabel posts\_favourited pada Mysql dengan menggunakan

phpmyadmin terdapat primary key FID dengan USERID dan PID sebagai foreign key. Tabel ini adalah untuk menghubungkan siapa yang melakukan like pada postingan tertentu dengan menyimpan informasi USERID dan PID.

Gambar 8 merupakan bentuk fisik dari tabel members dengan Mysql menggunakan phpmyadmin. Dengan primary key USERID elemen terpenting yang digunakan dalam sistem notifikasi ini adalah USERID dari member dan foto profile dari user tersebut.

Setelah fitur selesai dikerjakan, maka langkah selanjutnya adalah pengujian akhir yaitu pengujian test case dengan menggunakan metode Black Box dan pengujian User Experience. Sebelum pengujian User Experience perlu dilakukan uji test case, gunanya adalah untuk memastikan apakah sistem dapat digunakan dengan baik sesuai skenario perancangan di scrum planning. Metode untuk pengujian ini menggunakan Black Box Testing termuat pada Tabel III.

| # | Column               | Type         | Collation         | Attributes                  | Null | Default | Extra                       | Action           |
|---|----------------------|--------------|-------------------|-----------------------------|------|---------|-----------------------------|------------------|
| 1 | notification_id      | bigint(255)  |                   |                             | No   | None    | AUTO_INCREMENT              | Change Drop More |
| 2 | userid               | bigint(255)  |                   |                             | Yes  | NULL    |                             | Change Drop More |
| 3 | to_userid            | bigint(255)  |                   |                             | Yes  | NULL    |                             | Change Drop More |
| 4 | notification_type    | varchar(255) | latin1_swedish_ci |                             | Yes  | NULL    |                             | Change Drop More |
| 5 | notification_message | varchar(255) | latin1_swedish_ci |                             | Yes  |         |                             | Change Drop More |
| 6 | post_id              | bigint(255)  |                   |                             | Yes  | NULL    |                             | Change Drop More |
| 7 | isRead               | int(10)      |                   |                             | No   | None    |                             | Change Drop More |
| 8 | create_date          | timestamp    |                   | on update CURRENT_TIMESTAMP | Yes  | NULL    | ON UPDATE CURRENT_TIMESTAMP | Change Drop More |
| 9 | modifie_date         | datetime     |                   |                             | Yes  | NULL    |                             | Change Drop More |

Gambar 5. Bentuk fisik tabel notifikasi

| #  | Column      | Type         | Collation       | Attributes | Null | Default    | Extra          | Action           |
|----|-------------|--------------|-----------------|------------|------|------------|----------------|------------------|
| 1  | PID         | bigint(20)   |                 |            | No   | None       | AUTO_INCREMENT | Change Drop More |
| 2  | USERID      | bigint(20)   |                 |            | No   | 0          |                | Change Drop More |
| 3  | story       | text         | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 4  | tags        | varchar(200) | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 5  | source      | varchar(200) | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 6  | CID         | bigint(20)   |                 |            | No   | None       |                | Change Drop More |
| 7  | nsfw        | int(1)       |                 |            | No   | 0          |                | Change Drop More |
| 8  | pic         | varchar(20)  | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 9  | youtube_key | varchar(20)  | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 10 | fod_key     | varchar(20)  | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 11 | vfy_key     | varchar(50)  | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 12 | vmo_key     | varchar(50)  | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 13 | url         | text         | utf8_general_ci |            | No   | None       |                | Change Drop More |
| 14 | time_added  | varchar(20)  | utf8_general_ci |            | Yes  | NULL       |                | Change Drop More |
| 15 | date_added  | date         |                 |            | No   | 0000-00-00 |                | Change Drop More |
| 16 | active      | char(1)      | utf8_general_ci |            | No   |            |                | Change Drop More |
| 17 | phase       | bigint(1)    |                 |            | No   | 0          |                | Change Drop More |
| 18 | favclicks   | bigint(50)   |                 |            | No   | 0          |                | Change Drop More |
| 19 | last_viewed | varchar(20)  | utf8_general_ci |            | No   |            |                | Change Drop More |
| 20 | viewed      | bigint(50)   |                 |            | No   | None       |                | Change Drop More |

Gambar 6. Bentuk fisik tabel *posts*

| # | Column | Type       | Collation | Attributes | Null | Default | Extra          | Action           |
|---|--------|------------|-----------|------------|------|---------|----------------|------------------|
| 1 | FID    | bigint(20) |           |            | No   | None    | AUTO_INCREMENT | Change Drop More |
| 2 | USERID | bigint(25) |           |            | No   | 0       |                | Change Drop More |
| 3 | PID    | bigint(25) |           |            | No   | 0       |                | Change Drop More |

Gambar 7. Bentuk fisik tabel *posts\_favourited*

| #  | Column           | Type           | Collation       | Attributes | Null | Default | Extra          | Action           |
|----|------------------|----------------|-----------------|------------|------|---------|----------------|------------------|
| 1  | USERID           | bigint(20)     |                 |            | No   | None    | AUTO_INCREMENT | Change Drop More |
| 2  | email            | varchar(80)    | utf8_general_ci |            | No   |         |                | Change Drop More |
| 3  | username         | varchar(80)    | utf8_general_ci |            | No   |         |                | Change Drop More |
| 4  | password         | varchar(50)    | utf8_general_ci |            | No   |         |                | Change Drop More |
| 5  | pwd              | varchar(50)    | utf8_general_ci |            | No   | None    |                | Change Drop More |
| 6  | fullname         | varchar(200)   | utf8_general_ci |            | No   |         |                | Change Drop More |
| 7  | gender           | varchar(6)     | utf8_general_ci |            | No   |         |                | Change Drop More |
| 8  | description      | text           | utf8_general_ci |            | No   | None    |                | Change Drop More |
| 9  | instagramaccount | varchar(255)   | utf8_general_ci |            | No   | None    |                | Change Drop More |
| 10 | country          | varchar(100)   | utf8_general_ci |            | No   |         |                | Change Drop More |
| 11 | posts            | int(20)        |                 |            | No   | 0       |                | Change Drop More |
| 12 | yourviewed       | int(20)        |                 |            | No   | 0       |                | Change Drop More |
| 13 | profileviews     | int(20)        |                 |            | No   | 0       |                | Change Drop More |
| 14 | youviewed        | bigint(20)     |                 |            | No   | 0       |                | Change Drop More |
| 15 | addtime          | varchar(20)    | utf8_general_ci |            | No   |         |                | Change Drop More |
| 16 | lastlogin        | varchar(20)    | utf8_general_ci |            | No   |         |                | Change Drop More |
| 17 | verified         | char(1)        | utf8_general_ci |            | No   | 1       |                | Change Drop More |
| 18 | status           | enum('1', '0') | utf8_general_ci |            | No   | 1       |                | Change Drop More |
| 19 | profilepicture   | varchar(100)   | utf8_general_ci |            | No   |         |                | Change Drop More |
| 20 | remember_me_key  | varchar(32)    | utf8_general_ci |            | Yes  | NULL    |                | Change Drop More |
| 21 | remember_me_time | datetime       |                 |            | Yes  | NULL    |                | Change Drop More |
| 22 | oauth_provider   | varchar(20)    | utf8_general_ci |            | No   | None    |                | Change Drop More |

Gambar 8. *Members*



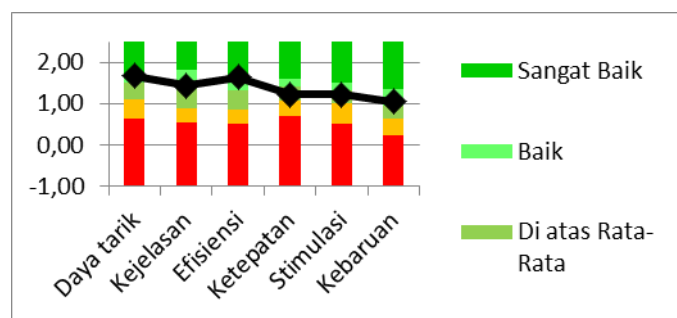
TABEL III. HASIL PENGUJIAN TEST CASE BLACK BOX TESTING

| No | Senkario   | Hasil |
|----|--|-------|
| 1. | User dapat melakukan like pada postingan User lain   | Ok    |
| 2. | User melakukan like di postingannya sendiri  | Ok    |
| 3. | User dapat melihat daftar notifikasi   | Ok    |
| 4. | User dapat melihat icon lonceng pada pojok kanan atas setelah login di Rupi.co   | Ok    |
| 5. | User B dapat melihat jumlah notifikasi baru dengan munculnya bubble berwarna merah dengan tulisan putih di atas icon lonceng | Ok    |
| 6. | User dapat mengunjungi postingan mana yang di-like oleh user lain  | Ok    |
| 7. | User dapat mengunjungi profile User yang melakukan like pada postingan   | Ok    |
| 8. | User dapat melakukan scrolling pada daftar notifikasi jika daftar notifikasi melebihi box notifikasi                         | Ok    |

Hasil dari pengujian *test case* menggunakan metode *black box* menunjukkan semua sistem yang dikembangkan dapat berjalan baik sesuai yang direncanakan. Namun, jika ingin dikembangkan lebih lanjut masih ada fitur kecil dari notifikasi yang belum menemui kesempurnaan. Tujuan dari penelitian ini adalah untuk menunjukkan bahwa metode tersebut dapat diterapkan atau tidak untuk pengembangan *software* dengan kondisi sumber daya manusia dan sumber dana yang terbatas.

Setelah melakukan pengujian *test case*, Sekarang dapat dilanjutkan untuk pengujian *User Experience*. Analisis *User Experience* ini melibatkan 22 responden. Responden *User Experience* adalah responden yang mengisi kuesioner pada saat studi pendahuluan. Perhitungan analisis menggunakan *User Experience Tools*.

Dari lembar *benchmark*, dapat diperoleh informasi bahwa kategori daya tarik, kejelasan, efisiensi, dan kebaruan bernilai baik. Sementara ketepatan dan stimulasi bernilai di atas rata-rata. Dapat dilihat pada Gambar 9.



Gambar 9. Benchmark (tolak ukur) kategori

Untuk analisis *User Experience*, menunjukkan hasil dalam semua kategori tidak ada yang di bawah rata-rata. Hal ini berarti bahwa pengembangan fitur ini dapat memberikan pengalaman pengguna (*User Experience*) yang baik kepada pengguna saat berinteraksi dengan Rupi. Baik dari sisi daya tarik, kejelasan, efisiensi, ketepatan, stimulasi dan kebaruan.

#### IV. SIMPULAN DAN SARAN

Berdasarkan hasil yang diperoleh dapat disimpulkan bahwa metode *Agile Software Development* dapat diimplementasikan pada pengembangan fitur notifikasi, yang dilakukan pada *Web Application* Rupi.co. Pengujian *black box* menunjukkan hasil implementasi berjalan secara fungsional seperti yang direncanakan dengan baik. Hasil dari analisis *User Experience* juga menunjukkan bahwa fitur notifikasi dapat memberikan pengalaman pengguna yang baik kepada pengguna saat berinteraksi dengan Rupi.co. Hasil dari pengembangan fitur notifikasi berupa produk Rupi.co yang memiliki fitur notifikasi. Hasil analisis *User Experience*, menunjukkan semua kategori tidak ada yang dibawah rata-rata. Hal ini berarti bahwa pengembangan fitur ini dapat memberikan pengalaman pengguna (*User Experience*) yang baik kepada pengguna saat berinteraksi dengan Rupi.co. Baik dari sisi daya tarik, kejelasan, efisiensi, ketepatan, stimulasi, dan kebaruan.

#### REFERENSI

- [1] Putri, S. 2015. *Tidak Mau Disamakan dengan 9GAG atau ICak, Rupi Jadi Platform Bagi Komikus dan Kartunis Lokal*. <https://id.techinasia.com/rupi-website-komik-strip-lucu-anti-meme-indonesia>, diakses pada tanggal 29 September 2015.
- [2] Pressman, R. 2002. *Software Engineering: A Practitioner's Approach*, 5th Edition, McGraw-Hill International Editions. New York: McGraw-Hill International.
- [3] Stol, K. J. dan Fitzgerald, B. 2015. *Theory-Oriented Software Engineering. Science of Computer Programming*, 101: 79-98.
- [4] Popli, R., Anita, & Chauhan, N. 2013. A Mapping Model for Transforming Traditional *Software Development* Methods to *Agile Methodology*. *International Journal of Software Engineering & Applications*, 4 (4): 53-64. DOI: 10.5121/ijsea.2013.4405
- [5] Tiwari, V. 2010. Some Observations on Open Source *Software Development* on *Software Engineering Perspectives*. *International Journal of Computer Science & Information Technology*, 2 (6): 113-125. DOI: 10.5121/ijcsit.2010.2611.
- [6] Mishra, D. dan Mishra, A. 2011. *Complex Software Project Development: Agile Methods Adoption*. *Journal of Software Maintenance and Evolution: Research and Practice*. DOI: 10.1002/smr.528
- [7] Sugiyono. 2009. *Metode Penelitian Kuantitatif, Kualitatif, dan R & D*. Bandung: Alfabeta.
- [8] Schwaber, K. dan Sutherland, J. 2013. *The Scrum Guide*.