

APLIKASI TEST REPORT DENGAN FITUR PUSH NOTIFICATION UNTUK CONTINUOUS REPORT TESTING BERBASIS ANDROID

Sri Dianing Asri*, Abdul Rahman Saleh **, Rimandani Litaningrum***

* Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana
Jln Meruya Selatan No. 1, Kembangan, Jakarta Barat
dianing.asri@mercubuana.ac.id

**Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana
Jln Meruya Selatan No. 1, Kembangan, Jakarta Barat
Warakasa22@mercubuana.ac.id

***Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana
Jln Meruya Selatan No. 1, Kembangan, Jakarta Barat
hi.rimanda@gmail.com

ABSTRACT

Sebuah report diperlukan dalam Automation testing yang menjalankan semua fungsi-fungsi pada fitur aplikasi dengan semua script yang telah dibuat. Pada saat script yang berisi perintah untuk menjalankan semua fitur maka dengan hanya sekali run akan menjalankan semua fungsi pada semua fitur yang ada pada sebuah aplikasi secara otomatis. Report diperlukan untuk mengetahui hasil dari sebuah pengujian dan persentase kesalahan ataupun hasil yang sudah valid, dan dengan menghubungkan calabash dengan testrail maka script automate testing yang dijalankan akan terhubung dengan testrail sehingga bisa generate report dari calabash menggunakan testrail report. Report tersebut harus diketahui oleh tim project yang bersangkutan maka dari itu dibuatlah sebuah fitur push notification dengan menggunakan metode FIFO dan menggunakan Firebase Cloud Messaging, berdasarkan report tersebut semua tim project bisa melihat karena notifikasi akan muncul di smartphone mereka. Dibuatnya fitur push notification pada aplikasi untuk menginfokan sebuah report testing adalah untuk mengantisipasi spam jika report dikirim melalui email.

Kata kunci: Report Testing, Testrail, Firebase Cloud Messaging, Algoritma FIFO

PENDAHULUAN

1.1. Latar Belakang

Saat ini pemanfaatan teknologi sudah merambah ke berbagai bidang kehidupan manusia. Kebutuhan akan teknologi yang cukup tinggi tentu menjadi tantangan tersendiri bagi para penggiat teknologi. Sadar akan hal itu maka bermunculan dibuatnya berbagai aplikasi, dan dengan banyaknya bermunculan dibuatnya aplikasi tersebut maka sangat diperlukan sekali untuk melakukan pengujian atau biasa disebut testing dalam sebuah aplikasi guna meminimalisir terjadinya kesalahan-kesalahan

ataupun bugs dan error yang akan terjadi ketika menggunakan aplikasi tersebut, namun dengan banyaknya fitur yang tersedia pada aplikasi maka tidak memungkinkan untuk melakukan pengujian secara berulang-ulang dari fitur awal hingga akhir jika terus menerus ada pengembangan fitur dan menjadikan fitur semakin banyak. Oleh karena itu adanya otomatisasi pengujian atau automation testing bertujuan untuk menguji sebuah aplikasi menggunakan automation tools tanpa harus melakukan testing dengan menjalankan semua fungsi-fungsi pada fitur-fitur aplikasi secara berulang dengan metode manual testing karena

otomatisasi ini ketika dijakankan satu kali maka akan menjalankan semua script yang telah di buat, ketika scriptnya berisi menjalankan semua fitur maka dengan cukup melakukan runningsatu kali akan menjalankan otomatis semua fungsi pada semua fitur yang ada pada sebuah aplikasi. Dan dengan adanya sebuah pengujian maka sangat diperlukan sebuah laporan atau biasa disebut report untuk mengetahui hasil valid dari sebuah pengujian dan persentase kesalahan maupun hasil yang sudah benar, dan dengan menghubungkan calabash dengan testrail maka script automate testing yang dijalankan akan terhubung dengan testrail sehingga bisa generate report dari calabash menggunakan testrail report. Dan tentu saja report tersebut harus diketahui oleh tim project yang bersangkutan maka dari itu dibuatlah sebuah fitur push notification guna untuk reminder dan menginfokan pada anggota project yang bersangkutan bahwa report testing sudah tersedia dan bisa dilihat hasilnya, bisa segera dilakukan fixing untuk bugs yang sudah ditemukan berdasarkan report tersebut sehingga semua tim project bisa melihat karena notifikasi akan muncul di smartphone mereka. Dibuatnya fitur push notif pada aplikasi untuk menginfokan sebuah report testing adalah untuk mengantisipasi spam jika report dikirim melalui email.

1.2. Perumusan Masalah

Dari latar belakang yang sudah dijelaskan sebelumnya, maka dapat diambil rumusan masalah yaitu :

1. Bagaimana agar mendapatkan push notification berupa report testing secara continuous?
2. Bagaimana cara membuka push notification dan membuka report testing?
3. Bagaimana cara membaca hasil report testing yang didapat melalui push notification?

1.3. Tujuan Penelitian dan Manfaat Penelitian

Tujuan dan manfaat yang ingin dicapai pada penelitian ini adalah:

1. Reminder melalui push notification report functional testing dalam sebuah project.
2. Semua anggota tim project dapat membaca hasil report testing dan bisa melakukan perbaikan berdasarkan hasil report yang didapatkan.
3. Tim project dapat langsung melihat report melalui smartphone masing-masing.
4. tim project dapat cepat melakukan perbaikan karena report diterima melalui smartphone dengan adanya fitur push notification

STUDI LITERATUR

2.1. Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak adalah teknologi berlapis. Mengacu pada IEEE, setiap pendekatan teknik (termasuk rekayasa perangkat lunak) harus bergantung pada komitmen organisasi terhadap kualitas. Manajemen kualitas total, Six Sigma, dan filosofi serupa yang mendorong budaya peningkatan proses yang berkesinambungan, dan budaya inilah yang pada akhirnya mengarah pada pengembangan pendekatan rekayasa perangkat lunak yang semakin efektif. Landasan yang mendukung rekayasa perangkat lunak adalah fokus terhadap kualitas. Software menyediakan fungsi dan kinerja yang diinginkan, struktur data yang memungkinkan program memanipulasi informasi yang cukup untuk dimengerti, serta dokumen yang menggambarkan operasi dan penggunaan dari program.

Tujuan dari rekayasa perangkat lunak adalah :

1. Memperoleh biaya produksi perangkat lunak yang rendah.
2. Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu.
3. Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis platform.
4. Menghasilkan perangkat lunak yang biaya perawatannya rendah.

Dasar untuk rekayasa perangkat lunak adalah lapisan proses. Proses rekayasa perangkat lunak adalah proses yang terus berulang, karena karakteristik perangkat lunak yang membutuhkan pemeliharaan dan pengembangan berkelanjutan agar perangkat lunak tidak kadaluarsa. Dalam proses pemeliharaan dilakukan koreksi kesalahan, adaptasi kebutuhan, peningkatan kemampuan atau fungsi dan bentuk pencegahan lainnya agar perangkat lunak tersebut tidak kadaluarsa. Alat rekayasa perangkat lunak merupakan unsur yang mendukung proses dan metode. Ketika alat-alat yang terhubung satu sama lain dan memberi informasi, serta informasi yang dibuat oleh salah satu alat dapat digunakan oleh yang lain, sistem untuk mendukung pengembangan perangkat lunak dapat dibangun dengan menggunakan bantuan komputer. (Pressman, R.S; Maxim, 2015)

2.2. Metode Waterfall

Metode penelitian yang digunakan dalam penelitian ini adalah metode Waterfall. Menurut Pressman [Pressman, R.S; Maxim, 2015] model waterfall

adalah model klasik yang bersifat sistematis, berurutan dalam membangun software.



Gambar 1. Waterfall Model (Roger S. Pressman, 2015)

2.3. Unified Modelling Language

Unified Modeling Language atau biasa disebut sebagai UML, adalah bahasa standard yang digunakan dalam menulis rancangan dari aplikasi. UML digunakan untuk menggambarkan, menentukan, membangun dan mendokumentasikan kebutuhan – kebutuhan dari software yang intensif. Dalam kata lain, layaknya seorang arsitek bangunan membuat rancangan untuk digunakan sebagai acuan oleh perusahaan konstruksi, maka seorang arsitek software membuat UML diagrams untuk membantu software developer dalam mengembangkan suatu aplikasi. Jika anda mengerti simbol – simbol yang digunakan dalam UML Diagram, anda akan lebih mudah untuk mengerti, anda dapat mulai membangun sebuah sistem dan menjelaskan desain tersebut dari mana (Pressman dan Maxim, 2015).

2.4. Testrail Management Tools

Testrail adalah sebuah management tools yang bisa digunakan untuk pembuatan test case, test suite, dan pembuatan report metrics. Testrail bisa terintegrasi dan terhubung dengan berbagai tools tracking bugs yang lain. (Neha, Prof. A.J.Nathe; 2016; Study of Automated Software Quality Checking System and Its Tools)

2.5. Firebase Cloud Messaging

Firebase Cloud Messaging biasa dikenal dengan Google Cloud Messaging adalah solusi pesan lintas platform untuk pengiriman pemberitahuan atau push notification tanpa biaya. Firebase mewarisi infrastruktur inti GCM namun menyederhanakan pengembangan di sisi client. Cukup dengan menulis beberapa baris kode, FCM dapat diintegrasikan ke dalam aplikasi. Ini berisi server yang bertanggung jawab untuk menetapkan token pengenalan ke setiap perangkat yang berlangganan untuk mengirim dan menerima notifikasi. Ini juga memastikan bahwa pesan dikirim dan diterima oleh perangkat tertentu. (Expert Android Programming, Prajyot Mainkar, 2017)

2.6. JAVA

Java dikembangkan oleh Sun Microsystems pada 1995. Java adalah high level, general purpose dari bahasa Object Oriented Programming.

Fitur-fitur utama dari Java:

1. High Level dan general purpose: diciptakan untuk menyelesaikan satu task yang spesifik, Java memperbolehkan untuk menulis perintah atau instruksi pada komputer yang mudah untuk dibaca di sebuah environment yang open-ended.
2. Object Oriented: Java juga bisa disebut sebagai object oriented language, objek memungkinkan mendefinisikan entitas modular dalam program yang jauh lebih mudah dibaca manusia dan lebih mudah dikelola untuk dibuat besar. Skala proyek perangkat lunak. Pikiran yang kuat tentang konsep berorientasi objek sangat penting bagi pengembang perangkat lunak modern.
3. Platform Independent: Terakhir, Java didesain dengan maksud agar bisa menulis sekali, jalankan bahasa mana saja. Java digunakan untuk mengembangkan aplikasi desktop, server web, dan aplikasi web sisi klien. Ini adalah bahasa asli dari sistem operasi Android, yang beroperasi pada ponsel dan tablet Android.

Java telah terbiasa menulis video game dan terkadang bahkan di porting ke perangkat yang lebih kecil tanpa sistem operasi tradisional. (Java Programming for Beginners; Mark Lassof, 2017)

2.7. SPRING FRAMEWORK

Spring adalah framework open source. Mula-mula dibuat oleh Rod Johnson pada bukunya Expert One-on-One: J2EE Design and Development.

Spring diciptakan untuk mengatasi kompleksitas pengembangan aplikasi enterprise dan memungkinkan penggunaan JavaBeans plain-vanilla untuk mencapai hal-hal yang sebelumnya hanya mungkin dilakukan dengan EJB. Tapi kegunaan Spring tidak terbatas pada pengembangan sisi server. Setiap aplikasi Java bisa mendapatkan keuntungan dari Spring dalam hal kesederhanaan, testability, dan kopling longgar. (Spring in Action, 4th Edition: Covers Spring 4; Craig Walls, 2014)

2.8. ANDROID

Android merupakan sistem operasi dari Google. Android gratis dan open source. Android didasarkan pada versi mobile-centric dari sistem operasi Linux, pada intinya. Sebagai kerangka aplikasi, Android mengemas serangkaian fitur canggih untuk pengembang untuk membangun aplikasi dengan pengalaman pengguna dan logika kompleks yang kaya. Sebagai middleware, Android menawarkan sejumlah library untuk membantu pengembang membangun ide besar berikutnya dengan mudah. Kit Pengembangan Perangkat Lunak Android dari

Google berisi semua alat yang diperlukan untuk memungkinkan pengembang membuat code, mengembangkan, dan menguji aplikasinya pada perangkat Android. (Building Hybrid Android Apps with Java and Javascript; Gok and Kanna, 2013)

2.9. HTML

HTML adalah predominant markup language yang digunakan untuk mendeskripsikan content atau data atau World Wide Web. HTML5 adalah iterasi paling akhir dari bahasa HTML dan termasuk dari fitur-fitur baru, improvement dari fitur yang sudah ada, dan API Javascript.

Dikatakan bahwa, HTML5 bukan sebuah reformulasi dari versi bahasa sebelumnya termasuk semua element valid dari kedua HTML4 dan XHTML 1.0. (HTML5 & CSS3 For The Real World, 2nd Edition; Goldstein, Lazaris, Weyl; 2015)

1.12. CSS
 Teknologi di belakang style sheets disebut CSS, atau Cascading Style Sheets. CSS adalah bahasa yang digunakan untuk mendefinisikan pembuatan style seperti fonts, colors, positioning, yang mana mendeskripsikan bagaimana informasi pada webpage terformat dan terdisplay. Style CSS dapat disimpan langsung di webpage HTML atau bisa disimpan secara terpisah pada file style sheet. Ketika digunakan secara eksternal, aturan style sheet ditempatkan ditempatkan pada eksternal style sheet document dengan ekstensi .CSS. (HTML, CSS and JavaScript All in One, Sams Teach Yourself: Covering HTML5, CSS3, and jQuery, Second Edition; Julie C Meloni; 2014)

2.10. JAVASCRIPT

JavaScript dikembangkan 20 tahun yang lalu oleh Netspace Corporation Community, pembuat web browserlong-defunct Netscape web browser. JavaScript adalah bahasa scripting web pertama yang support browser.

JavaScript original dikenal dengan LiveScript dan pertama kali dikenalkan oleh Netscape Navigator 2.0 1995. Namanya diubah menjadi JavaScript karena untuk mengindikasikan adanya relasi marketing dengan Sun's Java Language. (HTML, CSS and JavaScript All in One, Sams Teach Yourself: Covering HTML5, CSS3, and jQuery, Second Edition; Julie C Meloni; 2014)

Aplikasi tim project yang akan dibangun digunakan oleh tiga jenis pengguna. Berikut adalah informasi mengenai daftar pengguna di PT. XYZ

Tabel 1 Analisa Pengguna

User	Tanggung Jawab	Hak Akses
Quality Assurance	Mengeksekusi testing di calabarh dan testrail, view report testing yang ada di aplikasi	View report testing
Software Engineer	View report testing yang ada di aplikasi	View report testing
Product Manager	View report testing yang ada di aplikasi	View report testing

3.2. Analisa Kebutuhan

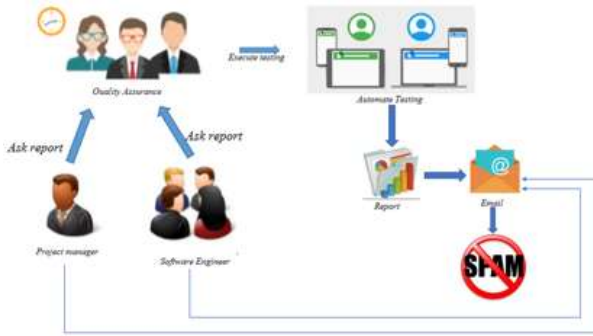
Analisa kebutuhan merupakan sebuah tahapan untuk menentukan apa saja yang dibutuhkan dalam pembuatan aplikasi ini. Analisa didapat dari wawancara dengan pihak terkait dan dari hasil itu kemudian diolah menjadi informasi yang digunakan untuk membangun aplikasi. Berikut adalah informasi mengenai hasil analisa kebutuhan :

1. Kebutuhan Input:
 - a. Data *Login*
 Data *Login* terdiri dari *username* dan *password* untuk melakukan akses aplikasi.
 - b. Data *Testrail Report*
 Data *testrail report* diperlukan untuk mengambil *report* yang akan ditampilkan.
2. Kebutuhan Proses
 - a. Proses mengolah data *testrail report* untuk diambil *endpoint reportnya*.
 Dan dimasukkan dalam *database*.
 - b. Proses mengambil data *endpoint* di *database* dan *diinsertkan* ke dalam *mobile* aplikasi dalam bentuk *push notification* dengan menggunakan *service google* yang bernama *firebase cloud messaging*.
3. Kebutuhan Output
 - a. Kebutuhan aplikasi menampilkan *report* dalam bentuk *endpoint/url* yang bisa *diview*.

3.3. Analisa Proses Berjalan

METHODOLOGI

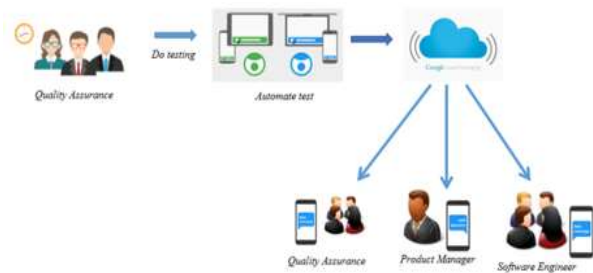
3.1 Analisa Pengguna



Gambar 1 Rich Diagram Proses Berjalan.

Penjelasan dari *Rich Diagram* di atas yaitu di PT. XYZ, *Quality Assurance* mengeksekusi proses *testing* dengan menggunakan *automation testing* menggunakan *calabash* kemudian *automate testing* tersebut akan *generate report* dengan menggunakan *tools* yang bernama *testrail report*, ketika *report* sudah tergenerate maka akan diberitahukan via email. Kendalanya adalah ketika *report* diberitahukan lewat email adalah masuknya pesan tersebut ke dalam SPAM, tertimpanya pesan tersebut dengan pesan-pesan yang lain sehingga membuat tim *project* tidak *aware* terhadap *report* tersebut. Dan tidak *aware*nya tim *project*, pesan *report* yang masuk ke SPAM, maupun tertimpa dengan pesan yg lain akan menyebabkan tim *project* yang bersangkutan bertanya secara langsung ke tim *Quality Assurance* mengenai *report testing*.

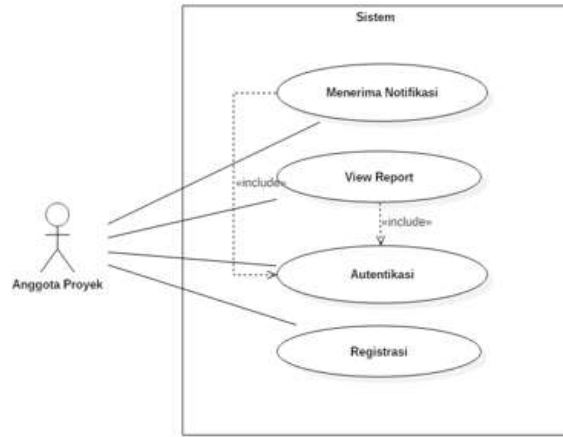
3.4. Perancangan Sistem



Gambar 2 Perancangan Sistem

1. *Quality Assurance* execute *automate testing* dan akan *generate report testing* melalui *testrail*.
2. Data *test report* yang berupa *testruns* akan masuk ke antrean di *firebase cloud messaging* untuk dikirimkan sebagai *push notification* di *smartphone* anggota proyek.
3. Anggota proyek akan menerima *push notification* di masing-masing *smartphone* yang isinya adalah *test report*.
4. *Use Case Diagram*

Diagram ini menggambarkan interaksi antara aktor dengan sistem. *Use case diagram* dari aplikasi *test report* berbasis android ini dapat dilihat pada gambar berikut:



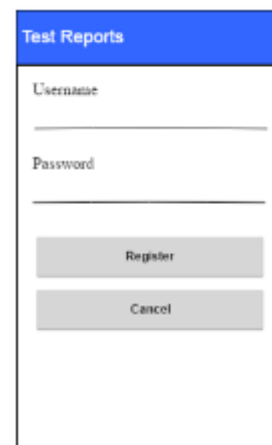
Gambar 3 Use Case Aplikasi Test Report

3.5. Desain Antar Muka

Desain antarmuka dari aplikasi ini adalah sebagai berikut:

1. Halaman Registrasi

Halaman registrasi digunakan *user* untuk dapat *register* atau daftar ke dalam aplikasi.



Gambar 4 Desain Antarmuka Halaman Registrasi

2. Halaman Login

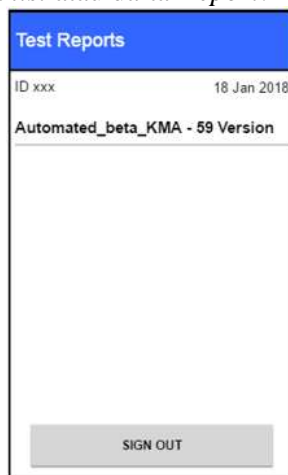
Halaman *login* digunakan *user* untuk dapat *login* atau masuk ke dalam aplikasi



Gambar 5 Desain Antarmuka Halaman Login

3. Halaman List Report

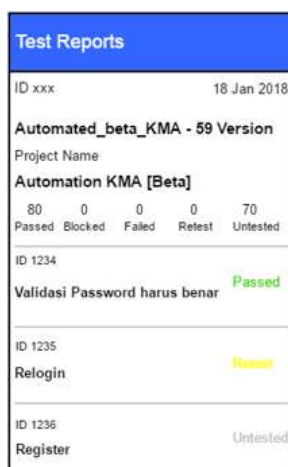
Halaman *list report* digunakan *user* untuk melihat *list* atau daftar *report*.



Gambar 6 Desain Antarmuka Halaman List Report

4. Halaman View Report

Halaman *view report* digunakan *user* untuk mengetahui detail dari masing-masing *report*.



Gambar 7 Desain Antarmuka Halaman View Report

HASIL DAN DISKUSI

5.1. Implementasi Sistem

Lingkungan Implementasi adalah tahap penerapan sekaligus pengujian bagi “Pembuatan Aplikasi Tim Project Berbasis Android dengan Penambahan Fitur *Push Notification* untuk *Continuous Report Testing* dengan menggunakan Algoritma FIFO” berdasarkan hasil analisa dan perancangan yang sudah dilakukan sebelumnya.

1. Implementasi Perangkat Keras

Spesifikasi minimum perangkat keras untuk dapat menjalankan aplikasi ini adalah sebagai berikut :

1. Processor Intel Pentium Dual Core E6300 2.8Ghz
2. Memory 2GB
3. Harddisk 40GB
4. Koneksi Internet 512 Kbps
5. Monitor resolusi 1366 x 768 pixels
6. Keyboard, Mouse

2. Implementasi Perangkat Lunak

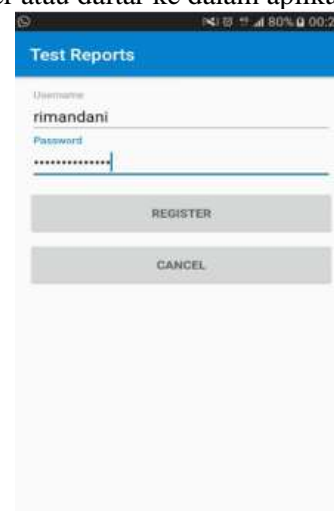
Spesifikasi perangkat lunak yang digunakan untuk membangun aplikasi ini adalah sebagai berikut :

1. Sistem operasi : Windows 7 32bit
2. Program aplikasi : Java, MySQL 5.6
3. Perangkat tambahan : Mozilla Firefox 5x

5.2. Hasil Implementasi

1. Implementasi Halaman Registrasi

Halaman registrasi digunakan *user* untuk dapat register atau daftar ke dalam aplikasi.



Gambar 8 Implementasi Halaman Registrasi

2. Implementasi Halaman Login

Halaman *login* digunakan *user* untuk dapat *login* atau masuk ke dalam aplikasi.



Gambar 9 Implementasi Halaman Login

4. Implementasi Halaman List Report

Halaman *list report* digunakan *user* untuk melihat *list* atau daftar *report*.



Gambar 10 Implementasi Halaman List Report

5. Implementasi Halaman View Report

Halaman *view report* digunakan *user* untuk mengetahui detail dari masing-masing *report*.



Gambar 11 Implementasi Halaman View Report

5.3. Hasil Pengujian

1. Skenario Uji Coba

Pengujian yang dijelaskan dalam laporan ini mewakili keseluruhan pengujian sistem yang dilakukan.

2. Pengujian

Tabel 2 Skenario Pengujian

No	Interface yang diuji	Langkah-langkah	Expected result	Actual result	Status
1	Halaman Login	1. Masukkan valid username 2. Masukkan valid password 3. Klik button SIGN IN	Masuk ke halaman utama yaitu halaman list report	Masuk ke halaman utama yaitu halaman list report	Pass
2	Halaman utama(halaman list report)	Klik salah satu list	Masuk ke halaman view report	Masuk ke halaman view report	Pass
3	Halaman utama(halaman list report)	Klik wordings list	Keluar menuju halaman login	Keluar menuju halaman login	Pass
4	Halaman View Report	Klik wordings list	Keluar menuju halaman login	Keluar menuju halaman login	Pass
5	Halaman Register	1. Masukan valid username 2. Masukan valid password 3. Klik button REGISTER	Username dan password berhasil terdaftar dan bisa digunakan untuk login	password berhasil terdaftar dan bisa digunakan untuk login	Pass
6	Halaman Registrasi	Klik button CAN	Masuk kembali ke halaman login	Masuk kembali ke halaman login	Pass
7	Menonima Push Notification	1. Login ke aplikasi 2. Run automate	Menonima Push Notification	Menonima Push Notification	Pass
8	Push Notification lebih dari 3	1. Login ke aplikasi 2. Run automate sering 3. Jalankan puller backend	Menonima Push Notification sesuai dengan timerlang, timerlang tercapai akan masuk sebagai push yang paling awal dan seterusnya	Notification sesuai dengan timerlang, timerlang tercapai akan masuk sebagai push yang paling awal dan seterusnya	Pass
9	Push Notification lebih dari 1 device	1. Login ke aplikasi 2. Run automate sering	Menonima Push Notification sesuai dengan timerlang di lebih dari 1 device, timerlang tercapai akan masuk sebagai push yang paling awal dan seterusnya	Notification sesuai dengan timerlang di lebih dari 1 device, timerlang tercapai akan masuk sebagai	Pass
10	Push Notification di hari dan waktu yang berbeda	1. Login ke aplikasi 2. Run automate sering 3. Jalankan	Menonima push notification sesuai dengan hari dihasilkannya report oleh automate sering dan hari	Menonima push notification sesuai dengan hari dihasilkannya report oleh	Pass

KESIMPULAN

Berdasarkan serangkaian teori, analisa, perancangan, implementasi dan pengujian perangkat lunak yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Mendapatkan reminder sebuah push notification report functional testing dengan menginstall aplikasi test report dan melakukan login setelah mendaftar dalam aplikasi test report.
2. Semua tim project menerima push notification setelah menginstall aplikasi di android device masing-masing dan mendaftar ke dalam aplikasi serta melakukan login ke dalam aplikasi.
3. Semua tim project bisa membaca report testing dengan cara membuka push notification yang mereka dapat di android device.
4. Tim project bisa mengefisiensi waktu dan mengerjakan pekerjaan yang lain tanpa bertanya mengenai report testing kepada tim Quality Assurance dengan hanya membuka report testing sesuai push notification yang mereka dapat di android device masing-masing.
5. Perbaikan atau bugs fixing bisa seketika dilakukan oleh software engineer setelah membaca report testing melalui fitur push notification yang didapat di android device masing-masing

- Pressman Roger S, Maxim Bruce R. 2015. Software Engineering A Practitioner's Approach. New York: Mc Graw Hill Education.
- Walls Craig. 2014. Spring in Action, 4th Edition: Covers Spring 4. Manning Publication.

DAFTAR PUSTAKA

- Gok, Khanna. (2013). Building Hybrid Android Apps with Java and JavaScript: O'Reilly Media, Inc.
- Goldstein Alexis, Lazaris Louis, Weyl Estelle . 2015. HTML5 & CSS3 For The Real World 2nd Edition. SitePoint.
- Lassof Mark. 2017. Java Programming for Beginners. Packt Publishing.
- Mainkar Prajyot. 2017. Expert Android Programming. Packt Publishing.
- Meloni Julie C. 2014. HTML, CSS and JavaScript All in One, Sams Teach Yourself: Covering HTML5, CSS3, and jQuery, Second Edition. Sams.