PENGUJIAN APLIKASI BELAJAR BAHASA MANDARIN (BERBASIS ANDROID DAN WEB) MENGGUNAKAN METODE BLACK BOX



Disusun Oleh:

Nama : Putri Saman

NIM : 18523152

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGUJIAN APLIKASI BELAJAR BAHASA MANDARIN (BERBASIS ANDROID DAN WEB) MENGGUNAKAN METODE BLACK BOX

TUGAS AKHIR JALUR MAGANG

Disusun Oleh:

Nama	: Putri Saman
NIM	: 18523152

Yogyakarta, 8 Agustus 2022

Pembimbing,



Chanifah Indah Ratnasari, S.Kom., M.Kom.

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGUJIAN APLIKASI BELAJAR BAHASA MANDARIN (BERBASIS ANDROID DAN WEB) MENGGUNAKAN METODE BLACK BOX

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 8 Agustus 2022

Tim Penguji	
Ketua Penguji Chanifah Indah Ratnasari, S.Kom., M.Kom.	
Anggota 1 Irving Vitra Paputungan, S.T., M.Sc., Ph.D.	
Anggota 2 Sheila Nurul Huda, S.Kom., M.Cs.	

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Putri Saman

NIM : 18523152

Tugas akhir dengan judul:

PENGUJIAN APLIKASI BELAJAR BAHASA MANDARIN (BERBASIS ANDROID DAN WEB) MENGGUNAKAN METODE BLACK BOX

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 8 Agustus 2022



(Putri Saman)

HALAMAN PERSEMBAHAN

Laporan Tugas Akhir ini saya persembahkan untuk seluruh keluarga, kerabat dekat saya yang telah mendukung serta membantu dengan baik secara finansial, mental, dan doa selama saya menempuh pendidikan di kampus, dan diri saya sendiri yang telah bertahan menjalani dan berusaha dalam setiap jalan yang ditempuh sampai saat ini.

HALAMAN MOTO

"Janganlah kamu berduka cita, sesungguhnya Allah selalu bersama kita"

- QS. At-Taubah: 40

"Some lights, ambitious. Some lights, wandering. People's lights. All of them are precious."

- *Mikrokosmos's Lyrics*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan taufiq serta hidayahNya kepada penguji dalam menyelesaikan laporan tengah magang di PT Javan Cipta Solusi. Selawat serta salam tak lupa juga dihaturkan kepada Nabi Muhammad SAW yang selalu menjadi teladan bagi umatnya.

Laporan ini disusun sebagai salah satu persyaratan dalam kelulusan di penjaluran Magang pada Program Studi Informatika - Program Sarjana dan menjadi bukti serta dokumentasi dari pelaksanaan magang. Pada kesempatan ini, penguji ingin mengucapkan terima kasih kepada:

1. Kedua orang tua yang selalu memberikan doa, semangat serta dukungannya selama kegiatan magang berlangsung.
2. Bapak Hendrik, S.T., M.Eng. selaku Ketua Jurusan Informatika dan Bapak Dr. R. Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Program Studi Informatika – Program Sarjana.
3. Ibu Chanifah Indah Ratnasari, S.Kom., M.Kom. selaku dosen pembimbing yang selalu perhatian, sabar, dan senantiasa memberikan arahan kepada penguji.
4. Seluruh pimpinan di PT Javan Cipta Solusi yang telah memberikan kesempatan dan pengalaman magang.
5. Bapak Dicky Puja Pratama, Ibu Anita, Risca Putriyana, serta seluruh rekan kerja pada tim proyek yang telah membimbing selama proses magang berlangsung.
6. Para dosen Program Studi Informatika Universitas Islam Indonesia yang telah memberikan ilmu kepada penguji selama di perkuliahan.
7. Rekan seperjuangan magang yaitu Alya Jelita, Yuvirdha yang senantiasa menemani perjalanan dan telah memberikan banyak motivasi selama kegiatan magang berlangsung serta seluruh sahabat terdekat yang selalu memberikan dukungan, dan senantiasa menemani penguji ketika menjalani kegiatan magang.
8. Kim Namjoon, Kim Seokjin, Min Yoongi, Jung Hoseok, Park Jimin, Kim Taehyung, Jeon Jungkook, beserta jajaran *K-Pop Group* lainnya atas semua karya yang memberikan semangat sekaligus *mood booster* bagi penguji selama penyusunan Tugas Akhir

Laporan ini masih jauh dari kata sempurna, sehingga penguji menerima segala bentuk saran guna menyempurnakan laporan ini. Semoga laporan ini dapat bermanfaat bagi pembaca.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 8 Agustus 2022



(Putri Saman)

SARI

Saat ini telah banyak pengembangan berbagai macam jenis aplikasi dilakukan untuk menciptakan banyak peluang dan kebiasaan baru dalam kehidupan sehari-hari. Dalam proses pengembangan perangkat lunak tentu banyak *bug* atau *error* yang tidak bisa dihindari ketika menggunakan aplikasi. Oleh karena itu, setiap pengembangan aplikasi sudah seharusnya melewati tahapan pengujian untuk menghindari kesalahan atau *bug*, memastikan seluruh proses dapat berjalan dengan baik, serta menghasilkan aplikasi yang berkualitas untuk dipakai oleh masyarakat. PT Javan Cipta Solusi menjadi salah satu perusahaan yang telah mengimplementasikan pengujian perangkat lunak dalam setiap pengembangan perangkat lunak pada setiap proyek. Dalam melakukan pengujian dibutuhkan metode dan strategi sesuai dengan kebutuhan pada setiap tim proyek.

Implementasi pengujian perangkat lunak dilakukan pada Sistem Belajar Bahasa Mandarin yang merupakan aplikasi yang digunakan untuk meningkatkan kemampuan berbahasa Mandarin bagi masyarakat Indonesia menggunakan metode *Black Box*. Pengujian sistem diawali dengan pengujian terpisah pada setiap fitur dari sisi fungsional hingga non fungsional. Setelah itu, dilanjutkan pengujian regresi atau *regression testing* untuk memastikan bahwa perubahan yang dilakukan pada pengujian sebelumnya tidak mempengaruhi alur keseluruhan sebagai sebuah satu kesatuan aplikasi yang berjalan. Dapat disimpulkan bahwa pengujian menggunakan metode *black box* mampu memastikan setiap fitur dan keseluruhan alur pada Sistem Belajar Bahasa Mandarin dapat berjalan sesuai dengan kriteria yang ada. Dalam penerapan pengujian secara terpisah pada tiap fitur dan *regression testing* pada tim proyek terhadap sistem yang di uji ditemukan bahwa pengujian terpisah pada setiap fitur fungsional hingga non fungsional lebih cocok digunakan dalam jangka waktu pendek karena langsung berfokus pada inti kebutuhan dan permasalahan yang ada. Sedangkan, *regression testing* cocok digunakan untuk jangka waktu panjang karena sifatnya seperti pengujian untuk *maintenance* yang dilakukan secara berkala dan berulang setiap terjadi perubahan pada sistem untuk memastikan setiap perubahan yang dilakukan tidak berpengaruh dengan keseluruhan alur.

Kata kunci: Pengujian Perangkat Lunak, *Black Box*, *Fungsional*, *Non Fungsional, Regression Testing*, Sistem Belajar Bahasa Mandarin.

GLOSARIUM

Black Box	metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas.
Regression Test	pengujian alur keseluruhan sebagai sebuah kesatuan aplikasi yang seharusnya berjalan.
Test Case	deskripsi skenario alur pengujian.

DAFTAR ISI

HALAMAN PENGESAHAN DOSEN PEMBIMBING……...ii
HALAMAN PENGESAHAN DOSEN PENGUJI……...iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR…….. iv
HALAMAN PERSEMBAHAN…….v
HALAMAN MOTO……..vi
KATA PENGANTAR…….. vii
SARI……..ix
GLOSARIUM…… x
DAFTAR ISI……. xi
DAFTAR TABEL…… xii
DAFTAR GAMBAR…….. xiv
BAB I PENDAHULUAN……..……..……..……..…….. 1
1.1 Latar Belakang……..……..……..……..……. 1
1.2 Ruang Lingkup Magang……..……..……..…….2
1.3 Tujuan……..……..……..……..……..…….3
1.4 Manfaat……..……..……..……..…….4
1.5 Sistematika Penulisan……..……..……..……...4
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA……..…….5
2.1 Pengujian Perangkat Lunak……..……..……..……..……..5
2.2 Black Box……..……..……..……..……...5
2.3 White Box……..……..……..……..……... 6
2.4 Grey Box……..……..……..……..…… 6
2.5 Perbedaan Black Box, White Box, dan Grey Box……..……..…… 6
2.6 Regression Testing……..……..……..……... 7
2.7 Tinjauan Pustaka……..……..……..……... 7
BAB III PELAKSANAAN MAGANG……..……..……... 9
3.1 Manajemen Proyek……..……..……..……... 9
3.2 Gambaran Sistem Belajar Bahasa Mandarin……..……..……..12
3.3 Pengujian Fungsional dan Non Fungsional pada Sistem Belajar Bahasa Mandarin……..14
3.4 Pengujian Regression Testing pada Sistem Belajar Bahasa Mandarin…….. 22
3.4.1 Alur Website Sistem Belajar Bahasa Mandarin……..…… 22

3.4.2 Alur Aplikasi Android Sistem Belajar Bahasa Mandarin ……..……. 26
3.4.3 Alur dan Hasil Pengujian……..……..……..…….. 32

BAB IV REFLEKSI PELAKSANAAN MAGANG……..……..……... 38
4.1 Relevansi Akademik……..……..……..…….. 38
4.1.1 Implementasi Pengujian Perangkat Lunak menggunakan Black Box…… 38
4.1.2 Penerapan Taiga dan Regression Testing dalam Proyek……..……... 38
4.2 Pembelajaran Magang……..……..……..……..…… 39

BAB V PENUTUP……..……..……..……..……...41
5.1 Kesimpulan……..……..……..……..…….41
5.2 Saran……..……..……..……..……..…….41

DAFTAR TABEL

Tabel 3.1 *Role* Pekerjaan Dalam Proyek…….. 11
Tabel 3.2 Daftar Status Taiga Berdasarkan *Role* Pekerjaan……. 11
Tabel 3.3 Hasil Rekap Pengujian pada Taiga……16
Tabel 3.4 *Task* Penambahan Fitur…….17
Tabel 3.5 *Task* Menyimpan Data……..17
Tabel 3.6 *Task* Optimasi Tampilan…….……..21
Tabel 3.7 Skenario Pengujian *Website* menu *Article* dan *Book*……24
Tabel 3.8 Skenario Pengujian Android menu *Article* dan *Book*……...29
Tabel 3.9 Hasil Pengujian dan *Feedback*……. 33

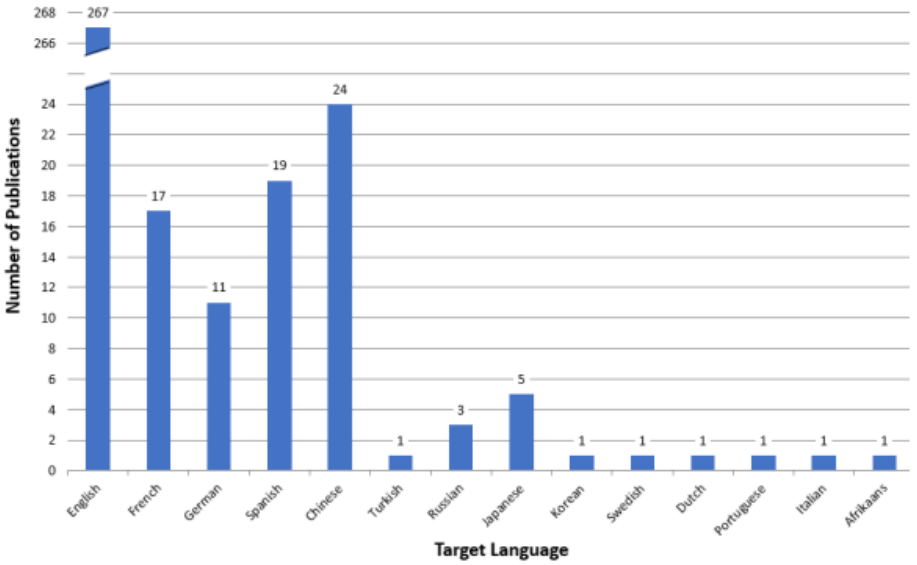
DAFTAR GAMBAR

Gambar 1.1 Pembelajaran Bahasa dan Jumlah Penelitiannya……2
Gambar 3.1 *Google Sheets Onboarding* Proyek…… 9
Gambar 3.2 Agenda *Sprint Review*…….. 10
Gambar 3.3 Aktivitas *Daily Scrum*…….. 11
Gambar 3.4 Gambaran Umum Menu *Website* Belajar Bahasa Mandarin……...13
Gambar 3.5 Gambaran Umum Menu Android Belajar Bahasa Mandarin……...13
Gambar 3.6 Alur *Task* pada Taiga…… 14
Gambar 3.7 *Task* dengan status "*Ready For Test*"……... 15
Gambar 3.8 *Detail Task* Proyek di Taiga……..16
Gambar 3.9 Rekap Dokumen Pengujian Taiga…… 17
Gambar 3.10 Alur admin dalam Pengelolaan Data *Website*…… 23
Gambar 3.11 Tampilan *Menu Books*…… 24
Gambar 3.12 Tampilan *Menu Articles*……. 24
Gambar 3.13 Tampilan *Menu Reading*…….26
Gambar 3.14 Halaman Kosakata……..27
Gambar 3.15 Tampilan reviu kosakata…….28
Gambar 3.16 Alur pengguna dalam menggunakan *Menu Reading*……..28
Gambar 3.17 Alur Pengujian……33

BAB I
PENDAHULUAN

1.1 Latar Belakang

Saat ini, teknologi telah banyak dimanfaatkan ke dalam berbagai bidang, salah satunya di bidang pendidikan. Penggunaan teknologi sebagai media pembelajaran bahasa telah banyak diterapkan. Dengan adanya *mobile technology* dan dunia virtual mampu menjadi media yang menciptakan peluang baru dalam meningkatkan minat pembelajaran bahasa. Setiap pengembangan aplikasi pembelajaran bahasa harus mampu melewati rangkaian proses hingga menjadi sebuah aplikasi yang layak untuk dipakai oleh masyarakat. Pengujian merupakan salah satu rangkaian penting dalam pengembangan perangkat lunak untuk pengecekan kelayakan dari sebuah aplikasi dan memastikan setiap bagian pada aplikasi dapat berfungsi dan berjalan dengan baik. Penemuan *bug* pada proses pengujian akan sangat berguna dalam meminimalisir kesalahan yang akan terjadi ketika aplikasi telah resmi dirilis dan digunakan oleh masyarakat. Dalam melakukan pengujian dibutuhkan ketelitian dan kesabaran untuk mengecek setiap bagian terkecil dari setiap fitur yang ada maupun pengecekan secara keseluruhan dari sistem yang berjalan untuk meminimalisir adanya *bug* pada aplikasi. Salah satu metode pengujian yang sering kali digunakan adalah metode *Black Box*. *Black box* adalah metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada input dan output aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum) (Iskandaria, 2012). Kegiatan belajar banyak dilakukan salah satunya terkait pembelajaran bahasa. Adapun penelitian yang menunjukkan bahwa Bahasa Mandarin menjadi pembelajaran bahasa yang cukup umum ditemui yang ditunjukkan pada Gambar 1.1.



Gambar 1.1 Pembelajaran Bahasa dan Jumlah Penelitiannya

Pada laporan ini akan membahas terkait pengujian aplikasi pembelajaran bahasa Mandarin menggunakan metode *Black Box*. Pengujian sistem diawali dengan pengujian terpisah pada setiap fitur dari sisi fungsional hingga non fungsional. Setelah itu, dilanjutkan pengujian regresi atau *regression testing* untuk memastikan bahwa perubahan yang dilakukan pada pengujian sebelumnya tidak mempengaruhi alur keseluruhan sebagai sebuah satu kesatuan aplikasi yang berjalan. Pengujian dilakukan dengan menggunakan dokumen *acceptance test* yang berisi detail dari setiap pengujian yang dilakukan pada aplikasi seperti kesesuaian alur, fungsionalitas setiap fitur, dan estetika dari desain yang ada. Pengujian ini memastikan bahwa aplikasi pembelajaran bahasa Mandarin memiliki kualitas yang sesuai dengan kriteria yang ditetapkan dan seluruh fungsionalitasnya dapat berjalan dengan baik.

1.2 Ruang Lingkup Magang

Kegiatan magang di PT Javan Cipta Solusi dimulai dari awal bulan September 2021 hingga Februari 2022. PT Javan Cipta Solusi merupakan sebuah perusahaan yang bergerak dalam bidang jasa teknologi informasi yang memiliki komitmen untuk membangun proses bisnis yang efektif dan efisien bagi klien dengan mengimplementasikan teknologi. Selama kegiatan magang berlangsung, penguji terjun ke dalam 2 proyek yaitu *Maintenance IT Business Requirement Online* (ITBRO) dan Sistem Belajar Bahasa Mandarin.

Maintenance ITBRO yang merupakan aplikasi berbasis web yang dapat diakses melalui jaringan internet internal perusahaan dan digunakan untuk mengelola proses permintaan layanan TI atau perubahan layanan TI (*IT Service Request / IT Change Request*) sesuai dengan bisnis proses yang ada pada perusahaan. Penulis berkontribusi selama tiga

bulan menjadi *Quality Assurance* yang bertugas dalam melakukan pengecekan kelengkapan terhadap setiap dokumen serta pengujian pada fitur ITBRO.

Selain itu, penguji berpartisipasi dalam proyek Sistem Belajar Bahasa Mandarin yang merupakan aplikasi berbasis web dan *android* yang digunakan untuk membantu dalam proses belajar bahasa Mandarin dengan berbagai fitur yang tersedia. Dalam proyek ini, penguji berkontribusi penuh dari awal hingga akhir periode magang selama enam bulan penuh sebagai *Quality Assurance* yang bertugas untuk melakukan pengujian terhadap kualitas sistem yang dihasilkan. Proyek ini merupakan proyek utama penguji selama berkontribusi di masa periode magang, sehingga dalam laporan ini akan fokus membahas terkait pekerjaan yang telah dilakukan sebagai seorang *Quality Assurance* dalam melakukan pengujian pada Sistem Belajar Bahasa Mandarin.

Sistem Belajar Bahasa Mandarin memiliki hubungan yang saling terintegrasi dalam menampilkan data pada *Website* dan *Android*. Pada Sistem Belajar Bahasa Mandarin berbasis Android digunakan oleh *user* untuk belajar bahasa Mandarin dengan memanfaatkan seluruh fitur seperti membaca buku atau artikel, hingga menyimpan setiap kosakata. Sedangkan pada bagian *Website* digunakan untuk admin dalam mengelola setiap data yang akan ditampilkan pada *Android*. Beberapa aktivitas yang dilakukan selama bergabung dalam proyek Sistem Belajar Bahasa Mandarin, yaitu:

a. melakukan pengujian pada setiap bagian kecil pada sistem,
b. melakukan pengujian secara keseluruhan sistem berjalan,
c. membuat dokumentasi hasil pengujian menggunakan *template User Acceptance Test (*UAT*),*
d. melakukan demo pada setiap pengujian yang telah dilakukan kepada klien,
e. melakukan analisis bersama jika terjadi *bug,*
f. melaporkan serta membuat dokumentasi *issue* yang ditemukan pada sistem.

Dalam laporan Tugas Akhir ini akan berfokus pada seluruh pengujian yang dilakukan menggunakan metode *Black Box* terhadap Sistem Belajar Bahasa Mandarin berbasis *Website* dan *Android* untuk memastikan kelayakan dari sistem yang telah dibangun.

1.3 Tujuan

Laporan Tugas Akhir memiliki tujuan untuk memberikan gambaran secara detail terkait pengujian perangkat lunak yang dapat dilakukan terhadap sebuah sistem serta memastikan kelayakan dan kesesuaian fungsionalitas pada Sistem Belajar Bahasa Mandarin

berbasis *Website* dan *Android* dengan seluruh rangkaian pengujian yang akan dilakukan menggunakan metode *Black Box*.

1.4 Manfaat

Manfaat yang diperoleh dari pengujian yang dilakukan terhadap Sistem Belajar Bahasa Mandarin yaitu:

1. Mengetahui apa saja rangkaian pengujian yang dapat dilakukan terhadap sistem yang telah dibangun.
2. Mengetahui kualitas dari aplikasi yang telah dibangun berdasarkan dengan kriteria yang ditetapkan.

1.5 Sistematika Penulisan

Sistematika pengujian disusun untuk memberikan gambaran umum isi laporan. Adapun susunan sistematika pengujian pada laporan adalah sebagai berikut.

a. BAB I: Pendahuluan

   Bab ini berisikan latar belakang, ruang lingkup pekerjaan, tujuan, manfaat, dan sistematika pengujian.

b. BAB II: Kajian Pustaka

   Bab ini berisikan ringkasan hasil analisis dari penelitian terdahulu dan dasar teori yang berhubungan dengan topik laporan.

c. BAB III: Pelaksanaan Magang

   Bab ini berisi tahapan penguji dalam melakukan pengujian menggunakan metode *Black Box* pada aplikasi Belajar Bahasa Mandarin.

d. BAB IV: Refleksi Pelaksanaan Magang

   Bab ini berisi hal-hal yang diperoleh selama proses magang di PT Javan Cipta Solusi.

e. BAB V: Kesimpulan dan Saran

   Bab ini berisikan kesimpulan dari hasil pengujian menggunakan metode *Black Box* pada Aplikasi Belajar Bahasa Mandarin beserta saran bagi pihak terkait yang melakukan hal serupa.

BAB II
LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Pengujian Perangkat Lunak

Pengujian merupakan salah satu rangkaian penting dalam pengembangan perangkat lunak untuk pengecekan kelayakan dari sebuah aplikasi. Pengujian perangkat lunak ini menjamin kualitas perangkat lunak yang merupakan bagian dari daur hidup pengembangan perangkat lunak (Mustaqbal et al., 2015). Pengujian perangkat lunak mencakup beberapa hal yaitu pengujian menggunakan metode tertentu dan pengujian berdasarkan strategi pengujian. Pada penelitian Wibisono & Baskoro (2002) disebutkan bahwa metode atau teknik yang dipilih dalam melakukan pengujian dapat diintegrasikan dengan strategi yang berbentuk urutan langkah sehingga mempermudah penguji dalam mengeksekusi setiap data atau kasus uji secara teratur.

Dalam melakukan pengujian dibutuhkan ketelitian dan kesabaran untuk mengecek setiap bagian terkecil maupun pengecekan secara keseluruhan dari sistem yang berjalan untuk meminimalisir adanya *bug* pada aplikasi. Sehingga, penting dalam menentukan jenis pengujian yang akan dilakukan serta mendefinisikan keseluruhan strategi dan prosedur terkait langkah-langkah pengujian secara spesifik. Adanya perencanaan dan prosedur pengujian akan membawa konstruksi perangkat lunak yang teratur dan penemuan kesalahan pada setiap tahap dalam proses konstruksi (Saputra, 2018).

2.2 *Black Box*

Black Box merupakan metode pengujian yang lebih difokuskan pada spesifikasi modul atau fungsi-fungsi yang dikembangkan, baik dari sisi struktur data, kesalahan GUI, pengaksesan data dalam *database*, serta kesalahan kinerja (Suwirmayanti et al., 2020). *Black Box Testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain (Jaya, 2018). *Black box testing* dapat diterapkan pada tiga jenis pengujian utama, yaitu *functional testing*, *non-functional testing*, dan *regression testing* (Wibowo, 2021). Dalam metode ini, keluaran yang dihasilkan dalam sistem akan dievaluasi apakah sudah sesuai dengan keluaran yang diharapkan. Metode *Black Box* cenderung menemukan hal-hal secara fungsional maupun non fungsional pada sistem seperti, fungsi yang tidak benar, antarmuka yang tidak sesuai, kesalahan struktur data dan akses basis data, kesalahan performansi, dan kesalahan inisialisasi dan terminasi (Mustaqbal et al., 2015). Black box testing pada umumnya dibandingkan dengan 2 metode lain yaitu White Box

Testing dan Grey Box Testing (Jaya, 2018).

2.3 *White Box*

White Box merupakan metode pengujian yang berfokus dalam menganalisis adanya kesalahan atau ketidaksesuaian yang terdapat pada kode program yang telah dibuat tanpa melihat tampilan antarmuka dari sistem (Eriana, 2021). Pada metode *white box*, penguji perlu melakukan pengecekan kode program untuk mengetahui bahwa unit dari kode tersebut berperilaku tidak sesuai (Mubarak, 2021). *White box* menjadi salah satu pengujian yang bertujuan untuk menghindari adanya *logical error* dan pemahaman *script* atau kode program keliru yang dapat disebabkan oleh kesalahan pengetikan dalam kode yang tidak dapat dihindari (Irawan, 2017).

2.4 Grey Box

Grey Box merupakan metode gabungan dari *black box* dan *white box testing* dimana pengujian *software* dilakukan berdasarkan spesifikasi namun menggunakan cara kerja dari dalam (Dhaifullah et al., 2022). Meski tidak memiliki akses secara langsung ke *source code*, penguji yang menggunakan metode *grey box* setidaknya memiliki sebagian pengetahuan atau akses ke dalam informasi dokumen atau konfigurasi dasar pada sistem seperti *database internal* yang digunakan (Sholeh & Wardaya, 2019).

2.5 Perbedaan *Black Box*, *White Box,* dan *Grey Box*

Pengujian menggunakan metode *black box*, *white box,* dan *grey box* memiliki perannya masing-masing dalam memaksimalkan kualitas dari sistem yang telah dibangun. Adapun perbedaan antara 3 metode yang yang ditemukan yaitu,

a. *Black box* bergantung pada informasi yang dimiliki pada domain sistem baik dari sisi tampilan maupun fungsionalitas dari sistem, *White box* bergantung pada informasi struktur dan pengujian pada *source code* (Verma, 2017). Sedangkan *Grey box* menggunakan gabungan antara kedua informasi yang dimiliki metode tersebut dalam proses pengujiannya (Dhaifullah et al., 2022).
b. *Black box* memiliki fokus utama pada dalam menguji fungsi yang dikembangkan atau proses dan respon berdasarkan sistem telah dikembangkan, *White Box* berfokus pada struktur, percabang, dan perulangan atau *loop* pada kode program yang dibuat, sedangkan *Grey Box* berfokus pada informasi domain aplikasi dan sebagian akses

internal yang dimiliki dalam mendesain skenario tes yang dapat menguji spesifikasi modul pada sistem dengan cara kerja dari dalam (Dhaifullah et al., 2022).

2.6 *Regression Testing*

Regression testing adalah pengujian ulang pada perangkat lunak yang sebelumnya telah diuji sebagai akibat dari perubahan atau modifikasi yang telah dilakukan pada perangkat lunak untuk memastikan bahwa perubahan tersebut tidak memengaruhi fungsionalitasnya. Proses pelaksanaan *regression testing*, yaitu menjalankan kembali *test case* yang telah dijalankan untuk memeriksa adakah *bug* baru atau *bug* lama yang muncul kembali (Rosero et al., 2017). *Regression testing* sering berkaitan dengan pengujian pada sistem yang telah terjadi perubahan kode, namun pengujian ini juga dapat dilakukan pada tingkat pengujian unit, atau sistem (Biswas & Mall, 2011). *Regression testing* merupakan jenis pengujian yang penting karena hasil *regression testing* yang baik menjadi sebuah kunci rahasia dalam menghasilkan perangkat lunak yang berkualitas (Wahl, 1999). Perbedaan yang ditemukan antara *Regression testing* dan *Integration Testing* yaitu *Regression testing* merupakan pengujian ulang menggunakan data pengujian yang telah dieksekusi sebelumnya untuk memeriksa adanya *bug* sebagai akibat dari perubahan yang sebelumnya dilakukan, sedangkan *Integration Testing* merupakan pengujian secara keseluruhan dalam memastikan bahwa fitur yang baru ditambahkan terintegrasi dengan baik ke dalam kesatuan perangkat lunak (Board, 2021).

2.7 Tinjauan Pustaka

Terdapat beberapa penelitian yang telah dilakukan sebelumnya terkait pengujian perangkat lunak menggunakan metode *Black Box*. Pada penelitian Setiyani (2019), dilakukan implementasi pengujian perangkat lunak menggunakan metode *black box testing* pada sistem informasi inventory perusahaan farmasi di mana setiap pengujian yang dilakukan mampu menghasilkan keterangan keputusan yang jelas pada tiap data yang diuji yang mempermudah pengembang dalam melakukan perbaikan.

Penerapan pengujian perangkat lunak menggunakan metode *black box* lainnya dilakukan pada perancangan Aplikasi Pembelajaran Bahasa Inggris berbasis Android yang diimplementasikan secara langsung terhadap sampel objek penelitian dan menghasilkan aplikasi yang dapat digunakan dengan mudah dimanapun dan kapanpun (Azis et al., 2020). Pada penelitian lainnya terkait pengujian *black box* terhadap sistem aplikasi TOEFL sebagai media pelatihan Bahasa Inggris berbasis website yang dilakukan oleh Marbun dkk. (2016) disebutkan bahwa fungsionalitas sistem yang dapat berjalan secara maksimal pada saat

pengujian pertama tidak menutup kemungkinan terjadinya kesalahan, sehingga dibutuhkan *maintenance* secara rutin untuk lebih mengetahui kekurangan dan kemungkinan terjadinya kesalahan pada aplikasi.

Dalam penelitian lainnya juga membahas gambaran umum dari penggunaan *Regression Testing* (Wahl, 1999). Penelitian ini menghasilkan kesimpulan bahwa *Regression Testing* sering dianggap sebagai hal yang dilakukan hanya pada saat aktivitas pemeliharaan perangkat lunak, namun idealnya hal tersebut harus terjadi setiap kali terdapat perubahan pada kode atau spesifikasi pada perangkat lunak. Pengujian ini menjadi salah satu jenis pengujian yang penting bagi pengembang maupun pengguna untuk memastikan perangkat lunak dapat berfungsi dengan benar dan dapat membangun kepercayaan terhadap kualitas yang ada pada perangkat lunak.

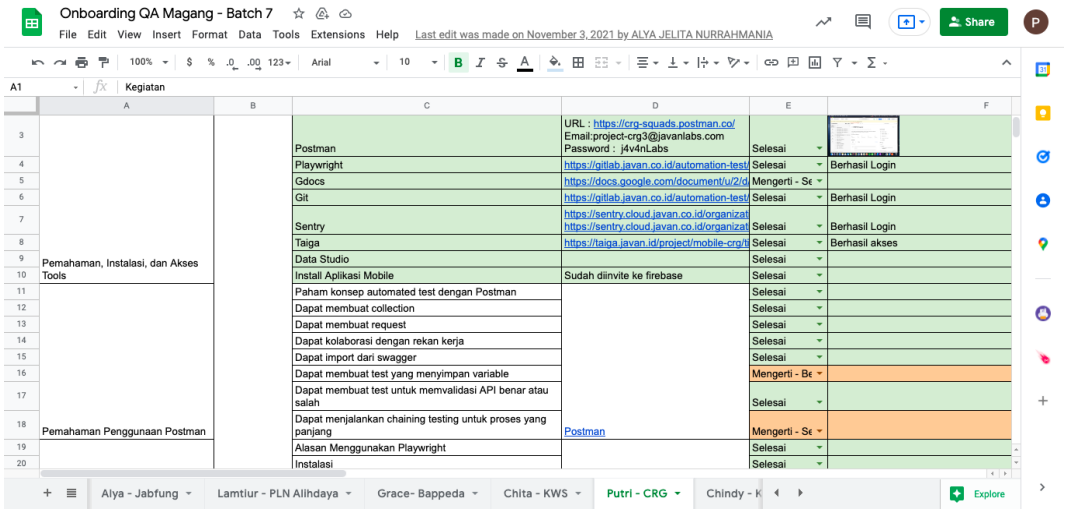
Berdasarkan seluruh penelitian yang telah disebutkan menunjukkan bahwa pengujian perangkat lunak sangat penting untuk dilakukan dan dapat dilakukan dengan dengan metode hingga strategi yang tepat dalam memastikan kualitas dari perangkat lunak. Pada penelitian Setiyani (2019) disebutkan bahwa *black box* menjadi jenis metode yang membuktikan dalam memberikan kemudahan penguji dalam hal pengujian fungsionalitas dan non fungsional dari perangkat lunak.

BAB III
PELAKSANAAN MAGANG

3.1 Manajemen Proyek

Sebelum terjun ke dalam proyek Sistem Belajar Bahasa Mandarin, penguji mengikuti rangkaian masa pengenalan pada proyek. Pengenalan proyek atau biasa dikenal dengan *Onboarding Project* merupakan kegiatan pengenalan proyek yang dilakukan ketika akan masuk ke dalam proyek yang sudah berjalan. Aktivitas pengenalan proyek atau *onboarding* dilakukan berkaitan dengan tugas utama penguji dalam melakukan pengujian, seperti penggunaan Taiga sebagai salah satu *tools management* proyek yang digunakan dalam mengatur segala tugas atau *task* pengujian, mendapatkan keseluruhan akses dan instalasi dari Sistem Belajar Bahasa Mandarin, serta melakukan pemahaman terkait alur Sistem Belajar Bahasa Mandarin.

Proses pemahaman dan adaptasi dalam melakukan pengujian Sistem Belajar Bahasa Mandarin dilakukan dalam kurun waktu yang cukup singkat kurang lebih 1 minggu. Kegiatan *onboarding* ini dipantau melalui *google sheets* yang telah diberikan *Supervisor* seperti pada Gambar 3.1. Sehingga, penguji setiap hari harus memberikan laporan harian melalui *google sheets* tersebut.



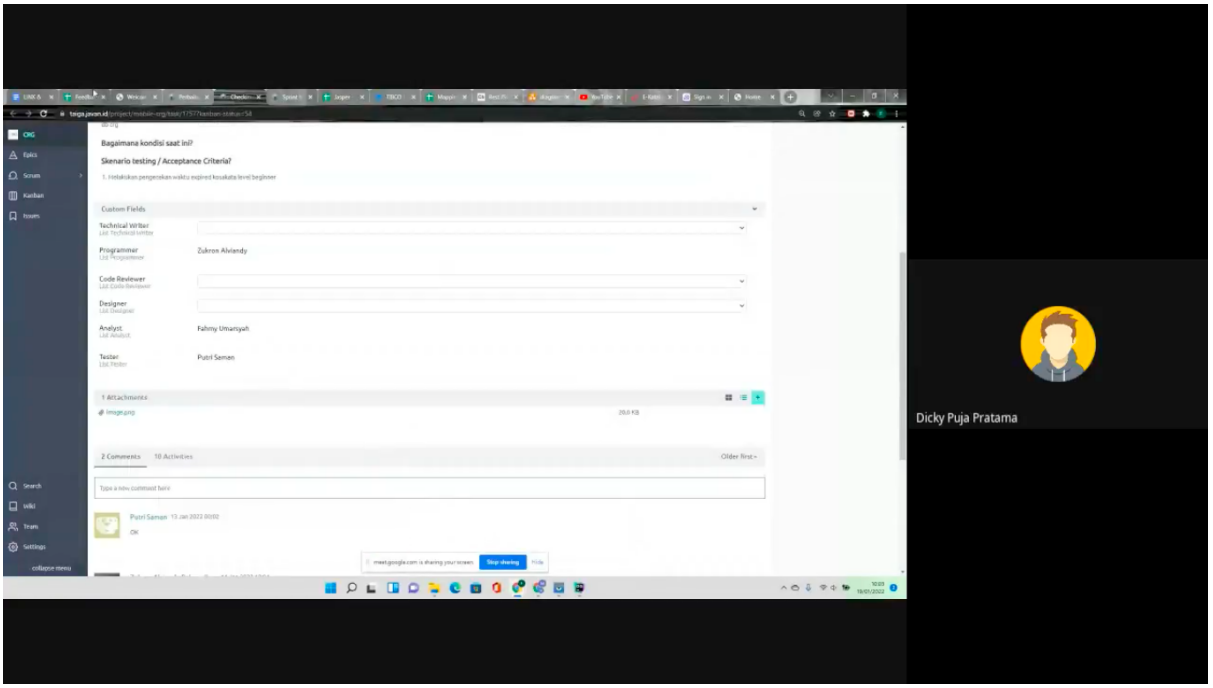
Gambar 3.1 *Google Sheets Onboarding* Proyek

Pengembangan proyek pada PT Javan Cipta Solusi menerapkan kerangka kerja *Scrum* di mana jangka waktu setiap *sprint* adalah 2 minggu. Aktivitas *sprint* diawali dengan *Sprint Planning* untuk mendefinisikan setiap pekerjaan setiap anggota tim dan diakhiri dengan *Sprint Review* untuk melaporkan kemajuan pekerjaan yang telah diselesaikan sekaligus menjadi sarana dalam pengumpulan kritik dan saran untuk *sprint* selanjutnya. Penulis ikut

memaparkan hasil kontribusi yang telah dikerjakan dalam aktivitas *Sprint Review* seperti pada Gambar 3.2. Adapun aktivitas Daily Scrum Meeting di mana penguji berperan untuk memaparkan hasil *Sprint Review* dan hasil Sprint Planning kepada klien seperti pada Gambar 3.2.



Gambar 3.2 Agenda *Sprint Review*



Gambar 3.3 Aktivitas *Daily Scrum*

Setiap anggota tim dalam pengembangan memiliki *role* pekerjaan masing-masing yang akan bekerja sesuai dengan tugas yang relevan. Adapun pembagian secara jelas terkait *role* dari setiap anggota tim pada proyek Sistem Belajar Bahasa Mandarin pada Tabel 3.1. Dalam mengelola setiap pekerjaan yang ada di dalam tim, perusahaan menggunakan alat manajemen proyek yaitu Taiga untuk mempermudah perkembangan dari setiap pekerjaan. Taiga membantu dalam *tracking* setiap pekerjaan melalui setiap *task* yang telah dibuat di mana masing-masing *task* memiliki status dan tenggat waktu yang harus segera diselesaikan. Setiap *role* pekerjaan memiliki status *task* tersendiri seperti yang ditunjukkan pada Tabel 3.2.

Tabel 3.1 *Role* Pekerjaan Dalam Proyek

Nama	***Role* Pekerjaan**	**Deskripsi**
Dicky	*Project Manager*	Bertanggung jawab penuh dalam tim proyek, melakukan pemantauan progres proyek secara berkala, dan menjadi jembatan antara tim internal dengan tim eksternal
Fahmy	*System Analyst*	Bertanggung jawab dalam melakukan analisis kebutuhan setiap *sprint*
Dwiky	*Programmer* Android	Bertanggung jawab dalam melakukan pengembangan sistem dan antarmuka di bagian Android
Zukron	*Programmer Website*	Bertanggung jawab dalam melakukan pengembangan sistem dan antarmuka di bagian *Website*
Putri Saman (Penulis)	*Quality Assurance (QA), Penulis*	Bertanggung jawab untuk memastikan seluruh fungsi pada sistem berjalan dengan baik dengan rangkaian pengujian yang dilakukan

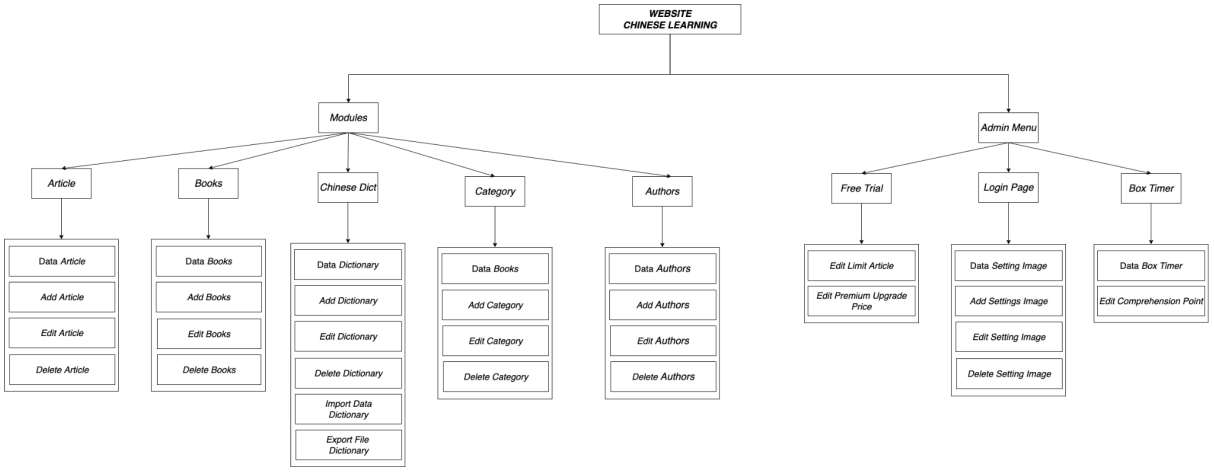
Tabel 3.2 Daftar Status Taiga Berdasarkan *Role* Pekerjaan

Assigned to	***Status Task***	**Deskripsi**
Analyst	*Backlog*	Daftar kebutuhan proyek yang akan dianalisis
	New	*Task* baru yang sedang dilengkapi setiap informasi oleh *analyst*
	Needs Info	*Task* yang dikembalikan kepada *Analyst* berisikan pertanyaan atau terdapat ketidakjelasan dalam proses analisis sehingga membutuhkan tanggapan atau

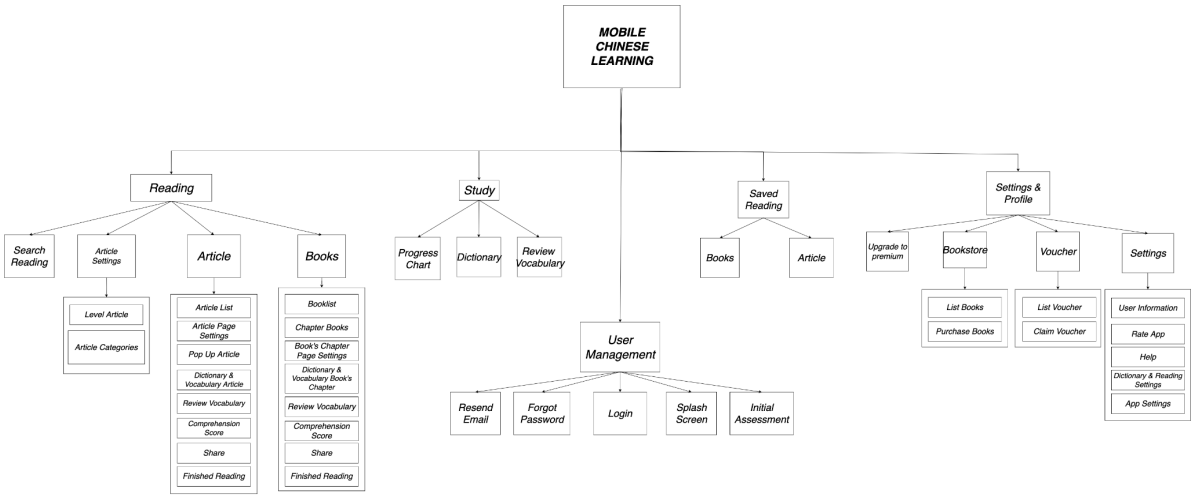
		penjelasan dari *Analyst*
Programmer	*To do*	*Task* yang telah dianalisis dan siap dikerjakan oleh *Programmer*
	In Progress	*Task* yang sedang dalam proses pengerjaan oleh *Programmer*
	Feedback	*Task* yang berisi pengembalian dari *Tester* karena terdapat *bug*/*issue*
QA/*Tester*	*Ready to Test*	*Task* yang diberikan *Programmer* sudah siap untuk dilakukan pengujian
	In Progress	*Task* yang sedang dalam proses pengujian oleh QA
	Closed	*Task* yang telah diuji dan sesuai dengan kriteria yang ada

3.2 Gambaran Sistem Belajar Bahasa Mandarin

Sistem Belajar Bahasa Mandarin atau *Chinese Learning System* memiliki tujuan untuk meningkatkan kemampuan berbahasa Mandarin bagi masyarakat Indonesia. Sistem terbagi menjadi dua bagian yaitu *Website* yang hanya digunakan oleh admin dan Android yang dapat digunakan oleh *user* dalam belajar. Aplikasi android memiliki berbagai fitur yang dapat menunjang *user* dalam belajar bahasa Mandarin. Setiap data yang ditampilkan pada aplikasi android harus diolah dengan sempurna agar tidak ada satupun data yang hilang. Oleh karena itu, peran admin pada *website Chinese Learning* sangat besar terhadap keberhasilan dari setiap fitur yang berjalan pada aplikasi *mobile*. *Website Chinese Learning* merupakan media bagi admin dalam melakukan pemasukan hingga pengelolaan data yang akan ditampilkan di aplikasi *Mobile*. Aplikasi *Chinese Learning* berbasis Android memiliki empat menu utama yaitu *Reading, Study, Saved, dan Profile,* sedangkan pada bagian *Website* terbagi menjadi dua menu utama yaitu *Modules* dan *Admin View*. Berikut merupakan gambaran umum menu dari aplikasi *Website* maupun Android *Chinese Learning* yang dapat dilihat pada Gambar 3.4 dan Gambar 3.5.



Gambar 3.4 Gambaran Umum Menu *Website* Belajar Bahasa Mandarin

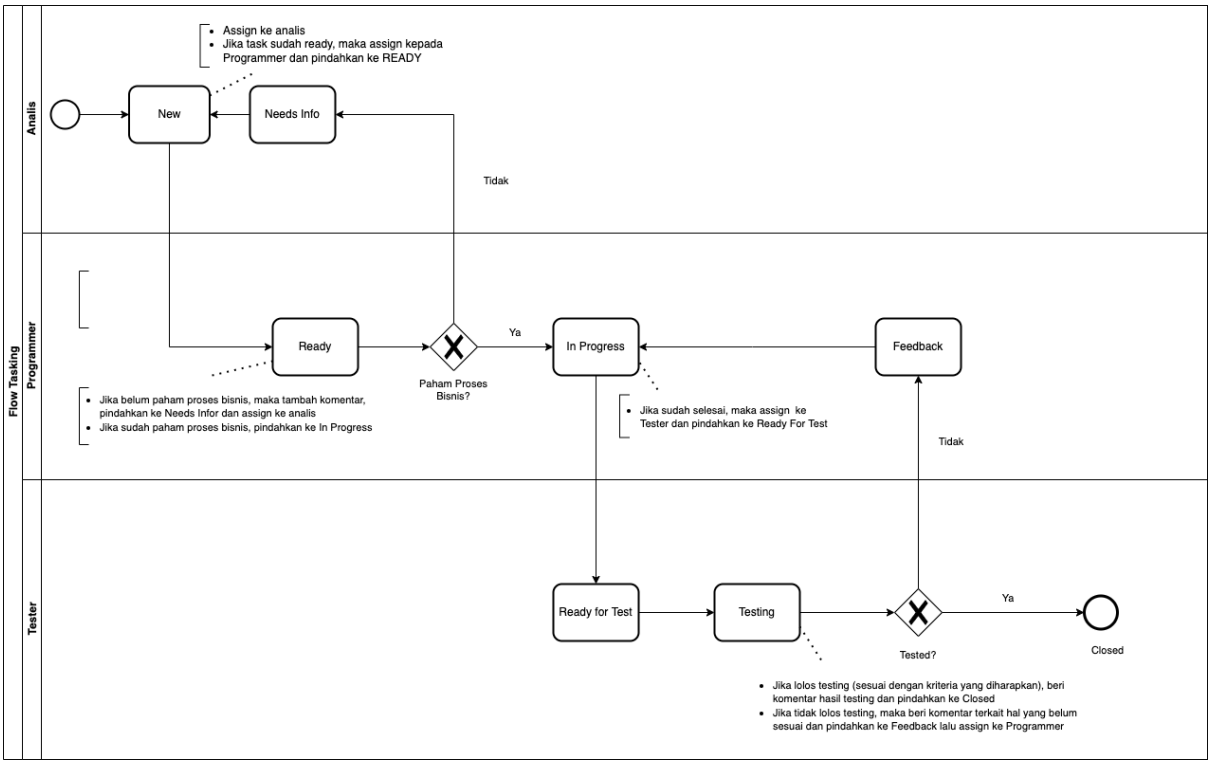


Gambar 3.5 Gambaran Umum Menu Android Belajar Bahasa Mandarin

Pengujian Sistem Belajar Bahasa Mandarin dilakukan secara manual menggunakan metode *Black Box Testing*. Pemilihan metode *Black Box* dipilih karena proses pengujian didasarkan pada detail dari Sistem Belajar Bahasa Mandarin yang mencakup fungsi-fungsi pada aplikasi, kesesuaian berjalannya alur, hingga pengecekan tampilan aplikasi. Hal tersebut selaras dengan fokus utama yang dimiliki oleh metode *Black Box*. Setiap fitur yang terdapat pada *website chinese learning* terintegrasi dengan aplikasi *mobile chinese learning*, sehingga setiap fitur yang berada pada *website* maupun *mobile* perlu diuji keberhasilan dan kesesuaiannya. Pengujian sistem diawali dengan pengujian terpisah pada setiap fitur dari sisi fungsional hingga non fungsional. Setelah itu, dilanjutkan pengujian regresi atau *regression testing* untuk memastikan bahwa perubahan yang dilakukan pada pengujian sebelumnya tidak mempengaruhi alur keseluruhan sebagai sebuah satu kesatuan aplikasi yang berjalan.

3.3 Pengujian Fungsional dan Non Fungsional pada Sistem Belajar Bahasa Mandarin

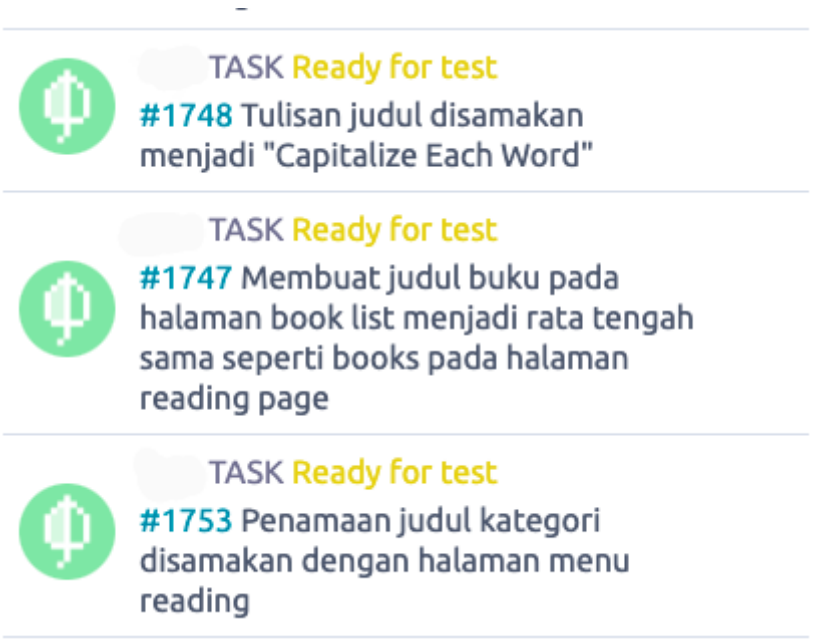
Pengujian menggunakan metode *black box* difokuskan dalam menemukan hal-hal seperti fungsi yang tidak benar, kesalahan performansi, kesalahan kinerja pada aplikasi hingga perbaikan pada antarmuka yang tidak sesuai. Pada sub bab ini akan dibahas terkait pengujian yang meliputi fungsional dan non fungsional pada Sistem Belajar Bahasa Mandarin. Aktivitas pengujian dilakukan melalui Taiga pada setiap Task yang sudah siap diuji dengan status "*Ready For Test*". Taiga merupakan perangkat lunak manajemen proyek sumber terbuka yang mendukung tim dalam bekerja secara *Agile* di seluruh kerangka kerja *Scrum* dan *Kanban* (Taiga, 2021). Adapun mekanisme penggunaan Taiga yakni pada setiap *task* yang telah dibuat oleh *System Analyst* pada Taiga akan melewati *Programmer* terlebih dahulu sebelum *task* tersebut masuk ke dalam tahap pengujian. Alur pengerjaan *task* di setiap *role* pada Taiga dapat dilihat pada Gambar 3.6.



Gambar 3.6 Alur *Task* pada Taiga

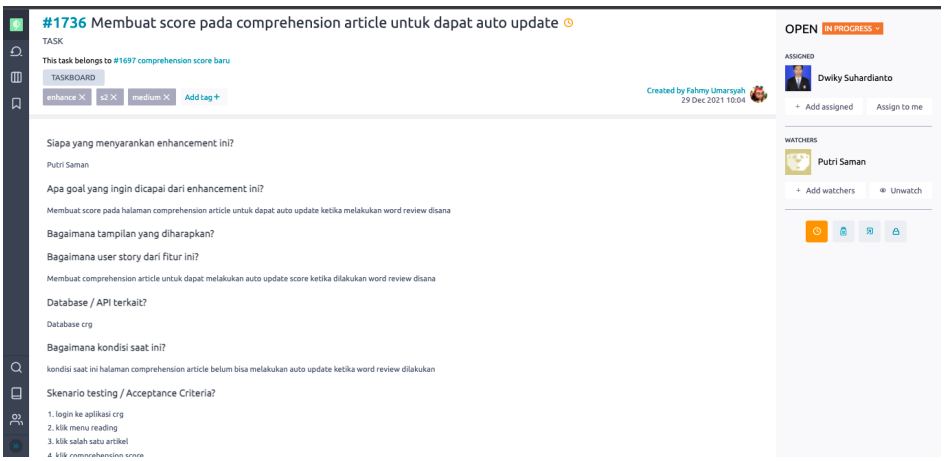
Kebutuhan tersebut dianalisis di setiap *sprint* dan dibuat *task* oleh *System Analyst*. Selain itu, beberapa *task* juga dibuat oleh *System Analyst* berdasarkan hasil *feedback* yang ditemukan pada *bug* atau saran pada *task* sebelumnya dari penguji sebagai *Tester*. Berikut adalah tahapan yang dilakukan penguji sebagai *Tester* dalam melakukan pengujian pada Taiga.

a. Penguji menerima *task* dengan status "*Ready For Test*". Setiap *task* yang sudah memiliki status "*Ready For Test"* dianggap telah melalui proses eksekusi oleh *Programmer* dan layak untuk diuji oleh *Tester* seperti pada Gambar 3.7.



Gambar 3.7 *Task* dengan status "*Ready For Test*"

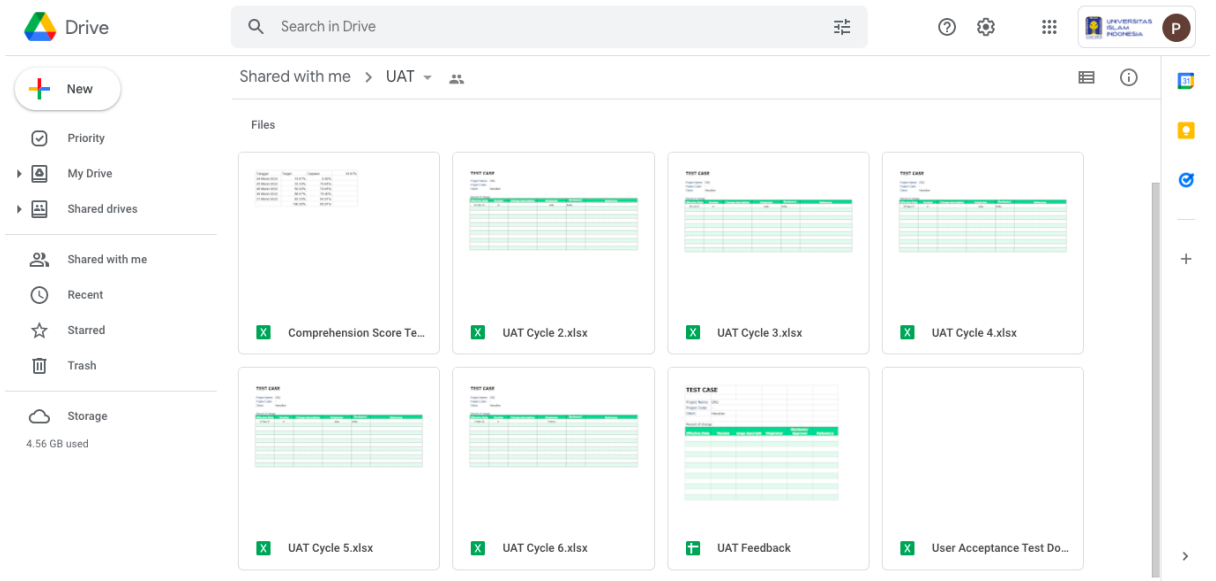
b. Setiap *task* yang dibuat oleh *System Analyst* wajib berisi informasi terkait *detail* penemuan *bug*, hasil yang diharapkan, alur pengecekan, serta kriteria yang harus dicapai seperti pada Gambar 3.8. Seluruh informasi yang tertera pada *Task* wajib dipahami oleh penguji sebelum melakukan pengujian. Apabila terdapat informasi yang kurang jelas terhadap keterangan yang ada di dalam *Task*, penguji dapat menuliskan pertanyaan untuk informasi lebih *detail* melalui *comments field* pada *task* tersebut agar dapat dilihat dan dijawab oleh sesama rekan tim.



Gambar 3.8 *Detail Task* Proyek di Taiga

c. Ketika hasil pengujian telah sesuai dengan seluruh kriteria yang harus dicapai maka penguji akan mengubah status *task* menjadi "*Closed*". Apabila hasil pengujian tidak sesuai karena ditemukan *bug* atau terdapat saran tambahan terkait *task* tersebut, maka penguji akan melampirkan keterangan tersebut melalui *comments field* di *task* tersebut dan *task* di*assign* kepada programmer untuk perbaikan.

d. Setiap hasil pengujian melalui Taiga yang dilakukan pada tiap *sprint* akan direkap menggunakan instrumen berupa dokumen *User Acceptance Test* (UAT) seperti pada Gambar 3.9. Dalam dokumen UAT terdapat poin-poin yang ada pada *task* Taiga. Dokumen tersebut akan digunakan sebagai laporan progres dari *sprint* yang berjalan sekaligus menjadi acuan bagi penguji melakukan demo secara *realtime* pada setiap *task* pengujian untuk klien.



Gambar 3.9 Rekap Dokumen Pengujian Taiga

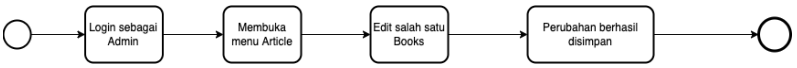
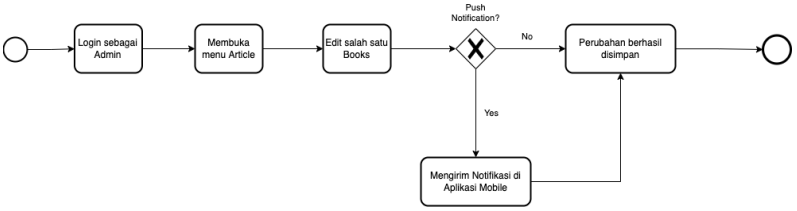
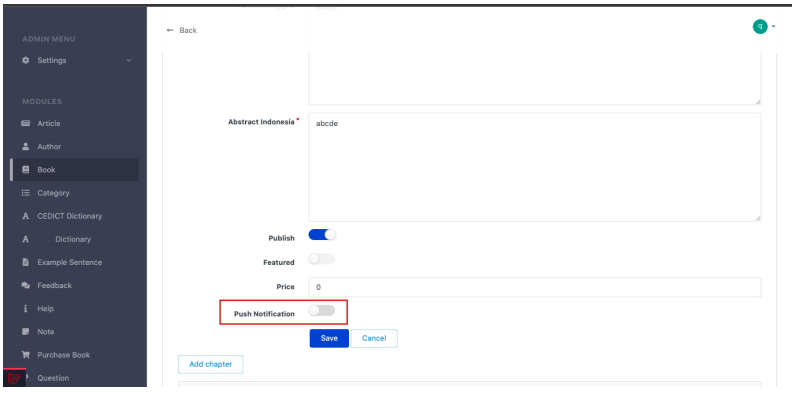
Penerapan Taiga menggunakan pengujian dengan metode *black box* dapat dilihat ke dalam 2 sisi pengujian yaitu pada bagian Fungsional dan Non Fungsional pada sistem. Contoh dari *task* pengujian yang dilakukan ditunjukkan pada Tabel 3.3.

Tabel 3.3 Hasil Rekap Pengujian pada *Task* Taiga

Total *Task*	*Mobile Task*	*Website Task*	Fokus *Task*	
			Fungsional	Non Fungsional
326	149	174	309	17

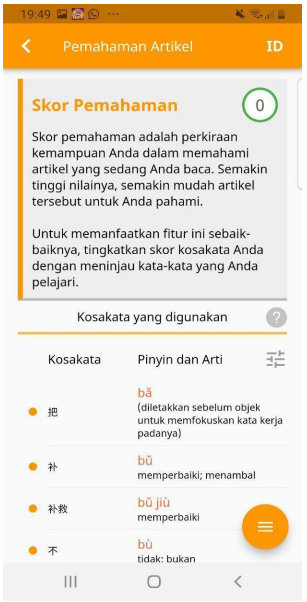
a. *Task* Fungsional merupakan *task* yang dilakukan pengujian pada keberhasilan dari proses yang dilakukan oleh sistem. Adapun contoh *task* fungsional yang dapat dilihat pada Tabel 3.4 dan Tabel 3.5.

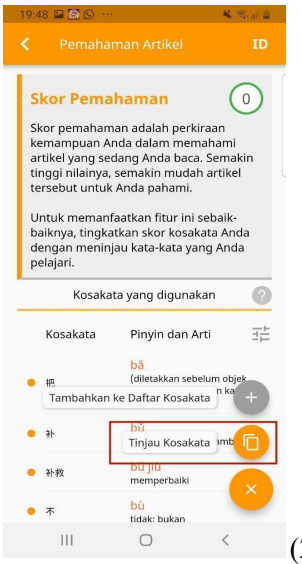
Tabel 3.4 *Task* Penambahan Fitur

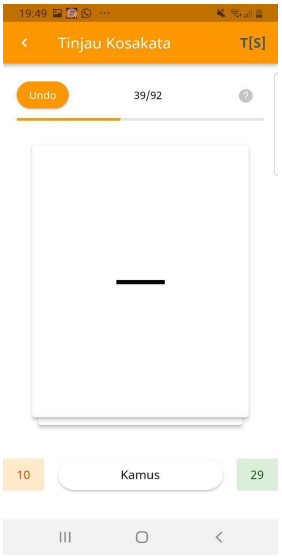
Judul *Task*	Menambahkan Fitur *Push Notification* saat *Editing Books* pada *Website Chinese Learning*
Deskripsi *Task*	*Website Chinese Learning* memiliki data buku yang ditampilkan pada Aplikasi *Mobile Chinese Learning* yang dapat ditambah, diubah, dan dihapus sesuai kebutuhan dan ketentuan dari admin. Setiap perubahan yang terjadi dapat diberitahukan kepada pengguna dengan adanya Fitur *Push Notification*. Pengujian dilakukan untuk memastikan Fitur *Push Notification* telah berhasil ditambahkan oleh *Programmer* dan berhasil berfungsi dengan kriteria yang ingin dicapai.
Kondisi Saat Ini	
Kriteria yang ingin dicapai	- Alur berubah menjadi :  - Menambahkan fitur dalam bentuk *button toggle Push Notification* untuk mengirim notifikasi perubahan yang terjadi ke Aplikasi *Mobile Chinese Learning*.
Result	

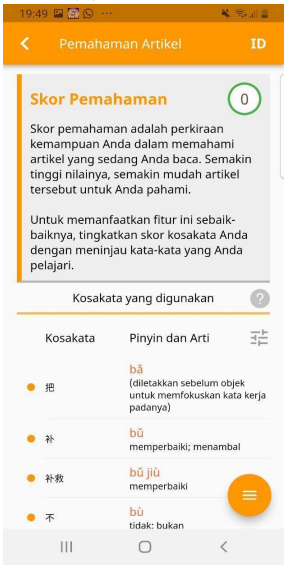
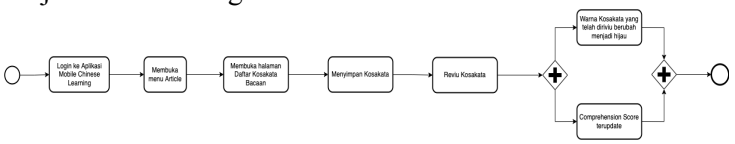
Tabel 3.5 *Task* Menyimpan Data

Judul *Task*	Sistem mampu menyimpan kosakata yang telah direviu
Deskripsi *Task*	Aplikasi *Mobile Chinese Learning* memiliki halaman kosakata Mandarin pada setiap bacaan yang ada. Setiap kosakata dapat

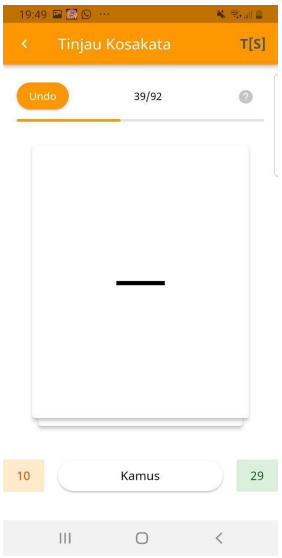
	disimpan dan direviu untuk memperdalam pemahaman pada setiap kata yang ada di bacaan. Perubahan yang terjadi diharapkan dapat disimpan sesuai dengan kriteria yang ada. Pengujian dilakukan untuk bahwa sistem mampu menyimpan kosakata sesuai dengan kriteria yang ingin dicapai.
Kondisi Saat Ini	Alur yang seharusnya terjadi:  Kondisi sebelum reviu kosakata pada halaman kosakata bacaan:  Ketika melakukan reviu kosakata (1)(2), tidak terjadi perubahan pada *Comprehension Score* (3). *Comprehension Score* adalah skor yang didapatkan setiap melakukan reviu kosakata. Pengguna akan mendapatkan skor terbaru atau skor akhir ketika kosakata telah sepenuhnya selesai di dalam bacaan.

<table>
<tr>
<td></td>
<td>(1)


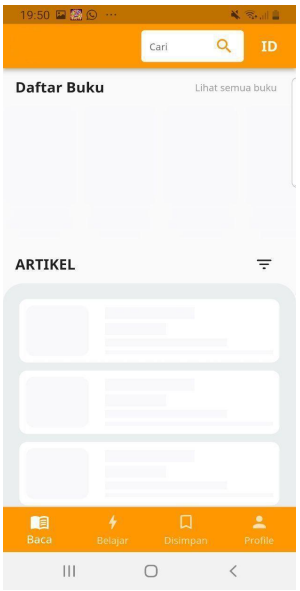
(2)


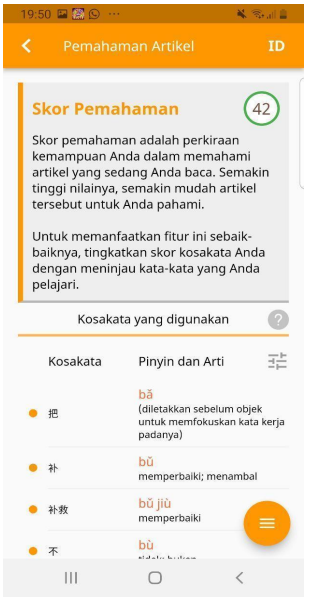
(3)

</td>
</tr>
<tr>
<td>Kriteria yang ingin dicapai</td>
<td>Berjalan sesuai dengan alur berikut


Berhasil menyimpan hasil reviu kosakata, dengan kondisi:
a. Warna pada kata yang telah direviu langsung berubah menjadi hijau
b. Jumlah comprehension score langsung berubah sesuai dengan reviu yang telah dilakukan.</td>
</tr>
<tr>
<td>Result</td>
<td>Setelah melakukan pengujian dengan mereviu kosakata pada halaman bacaan, sudah sesuai dengan alur yang diharapkan.
- Di bawah ini adalah tampilan ketika user melakukan reviu kosakata hingga selesai</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>


- Kondisi pada halaman kosakata akan berubah ketika User telah selesai melakukan reviu kosakata dan melakukan refresh terlebih dahulu halaman bacaan (1) lalu kembali membuka Halaman Kosakata pada Bacaan yang dipilih (2). Hal ini membuat hasil pengujian sudah sesuai dengan kriteria dan alur yaitu karena hasil reviu berhasil tersimpan dengan kondisi tersebut, namun terdapat catatan yang dapat dioptimalkan.



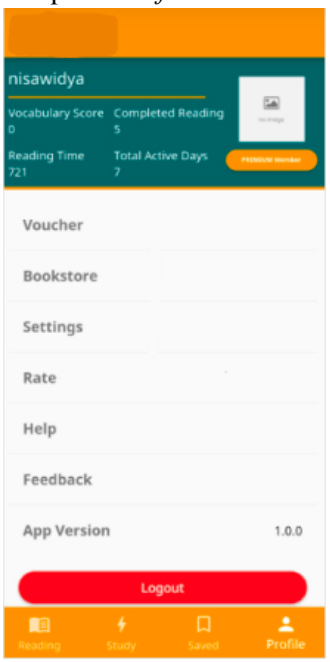
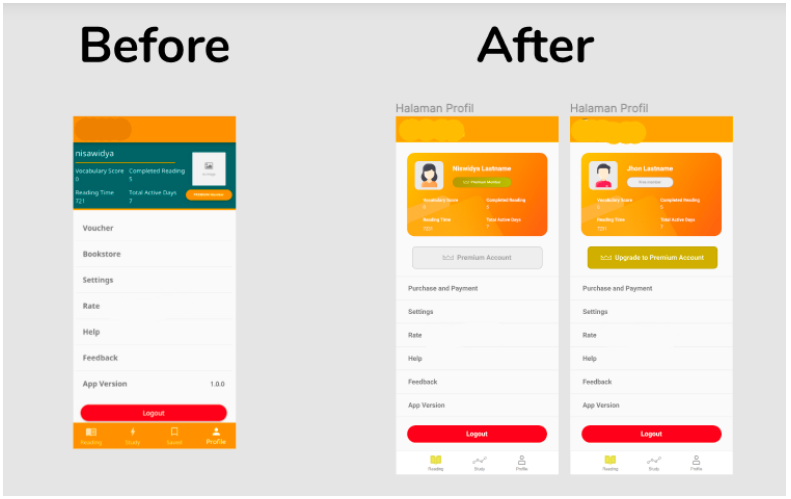
(1)


(2)
</td>
</tr>
<tr>
<td>Feedback</td>
<td>Tester memberikan catatan Feedback sebelum task Closed yang dapat dioptimalkan pada fitur tersebut.

- Membuat Halaman Kosakata dapat Auto-Update ketika selesai melakukan reviu. Hal ini memiliki tujuan yang sama dengan kriteria task yang ingin dicapai yaitu “Sistem mampu menyimpan kosakata yang telah direviu” namun dioptimalkan agar user tidak perlu melakukan refresh lagi pada halaman utama.</td>
</tr>
</table>

b. *Task* non fungsional merupakan task yang dilakukan pengujian pada sisi "quality attributes" dari suatu sistem. *Task* ini biasanya ditujukan untuk melakukan perubahan tampilan, desain, hingga penempatan setiap *button* pada sistem. Adapun contoh task non fungsional yang dapat dilihat pada Tabel 3.6.

Tabel 3.6 Contoh *Task* Optimasi Tampilan

Judul *Task*	Optimasi tampilan *profile*
Deskripsi *Task*	Melakukan perubahan desain pada menu *Profile* menjadi lebih baik dari desain sebelumnya. Pengujian dilakukan untuk memastikan perubahan berhasil dilakukan di bagian *Profile* sesuai dengan desain yang telah dibuat pada versi terbaru aplikasi android.
Kondisi Saat Ini	Tampilan *Profile* 
Kriteria yang ingin dicapai	

Result	Tampilan pada Aplikasi *Mobile Chinese Learning* berhasil diubah pada versi terbaru.

3.4 Pengujian *Regression Testing* pada Sistem Belajar Bahasa Mandarin

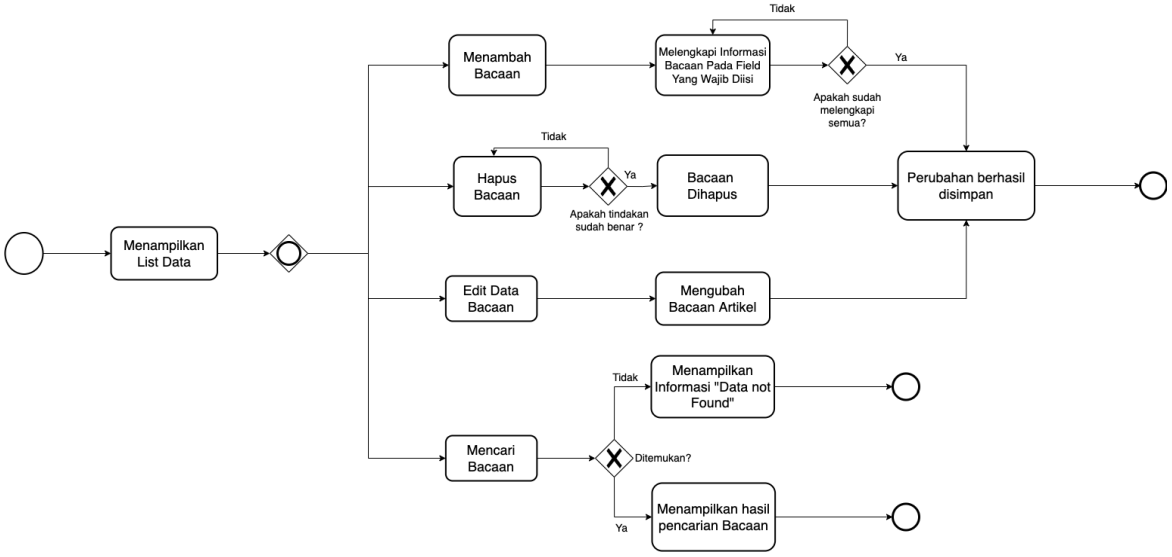
Pengujian ulang terhadap alur sistem diperlukan sebagai validasi bahwa perubahan dan perbaikan yang didapatkan pada pengujian secara terpisah menggunakan Taiga tidak mempengaruhi alur secara keseluruhan yang ada. Oleh karena itu, dibutuhkan pengujian secara berulang pada alur yang telah ada untuk memastikan seluruh skenario berhasil dijalankan sesuai dengan alur pada sistem. Regression testing dipilih dalam proses pengujian karena *regression testing* termasuk pengujian berulang menggunakan data pengujian yang telah dieksekusi sebelumnya untuk memeriksa adanya *bug* sebagai akibat dari perubahan yang sebelumnya dilakukan. Data pengujian yang digunakan adalah tabel skenario pengujian yang telah dibuat berdasarkan analisis pada setiap alur modul yang ada. Pengujian *Regression Testing* menggunakan metode *Black Box* yang dilakukan penguji secara manual.

Sistem Belajar Bahasa Mandarin berbasis *Website* maupun Android memiliki fungsi yang berbeda namun saling berhubungan. *Website* hanya digunakan oleh admin dalam mengelola setiap data yang ada untuk ditampilkan pada aplikasi Android yang dapat diakses oleh pengguna. Dalam pelaksanaan magang, pengujian Sistem Belajar Bahasa Mandarin berbasis *Website* dan Android, penguji telah membuat dan menyelesaikan sebanyak 326 *test case* yang telah didokumentasikan di *Spreadsheet* proyek. Akan tetapi, terdapat keterbatasan alur regresi yang tidak dapat diperlihatkan karena tidak mendapatkan izin perusahaan serta keterbatasan waktu dalam pengerjaan laporan Tugas Akhir ini sehingga akan disampaikan hasil pengujian pada proses regresi hanya mencakup Modul *Reading*. Bagian *Reading* dipilih sebagai studi kasus karena objek utama yang digunakan pengguna dalam belajar Bahasa Mandarin adalah kumpulan artikel maupun buku yang mengandung kosakata Mandarin yang dimasukkan admin melalui *Website*. Modul *Reading* terbagi menjadi dua yaitu, *Articles* dan *Books*. Berikut disampaikan keseluruhan alur dan hasil yang dilakukan dalam pengujian regresi pada website dan android Sistem Belajar Bahasa Mandarin.

3.4.1 Alur *Website* Sistem Belajar Bahasa Mandarin

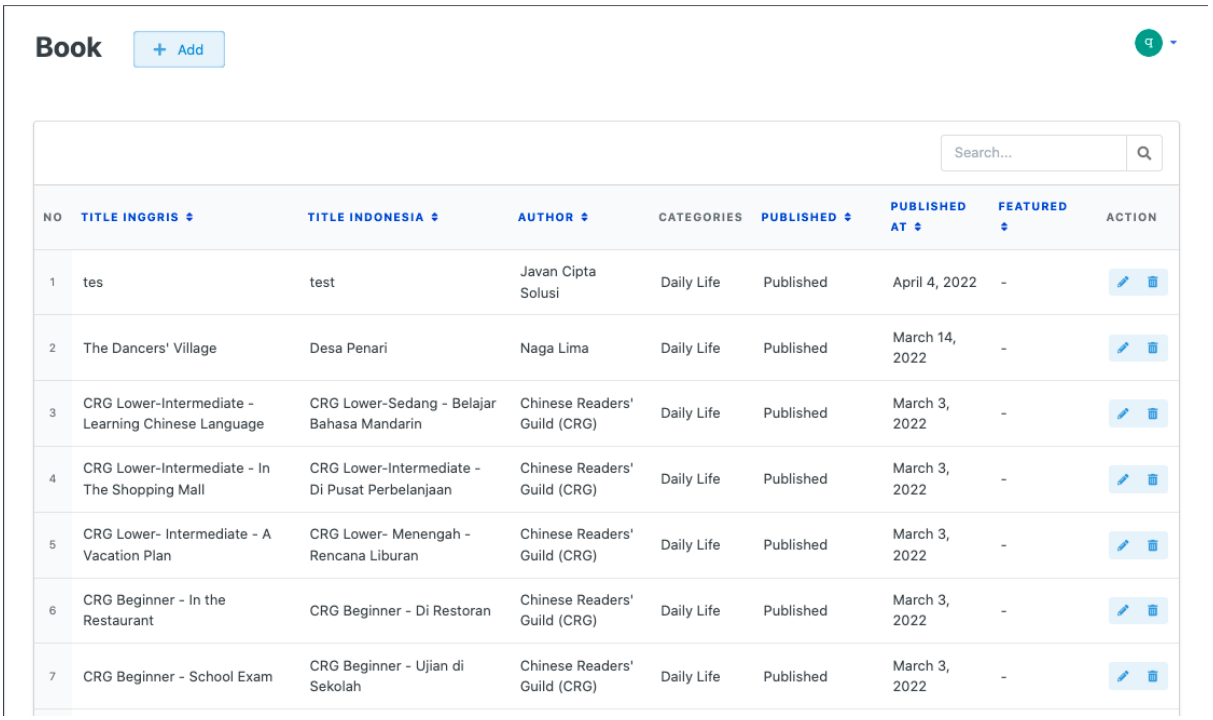
Pengguna dapat memanfaatkan seluruh bacaan seperti artikel atau buku yang terdapat pada Aplikasi Android *Chinese Learning* untuk mendalami pengetahuan seputar Bahasa Mandarin. Perbedaan antara artikel atau buku terdapat pada isi di dalamnya, di mana buku merupakan bacaan yang memiliki lebih dari satu *Chapter* sedangkan artikel tidak memiliki

Chapter. Setiap data bacaan yang muncul pada *menu Reading* di Aplikasi *Android Chinese Learning* dimasukkan dan dikelola oleh admin *Website*. Alur yang dilakukan admin ketika mengelola data pada *Website* dapat dilihat pada Gambar 3.10.

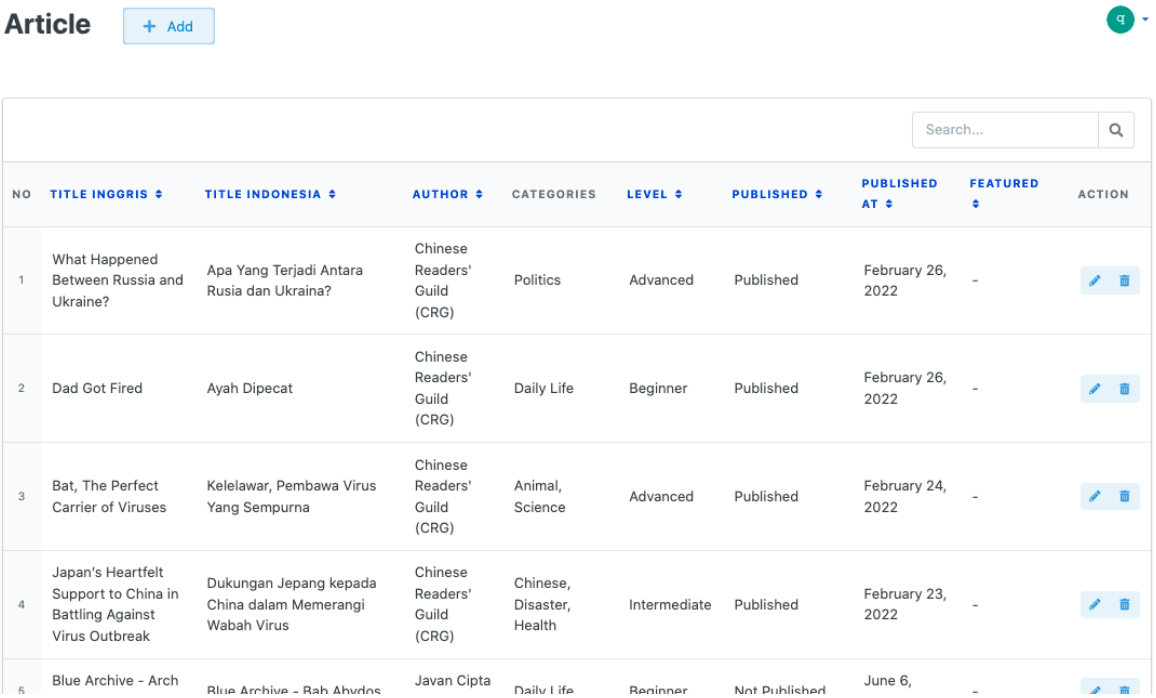


Gambar 3.10 Alur admin dalam Pengelolaan Data *Website*

Aktivitas perubahan ataupun penambahan data akan tersimpan otomatis ke Aplikasi Android ketika admin menyimpan seluruh perubahan. Alur kerja menu Artikel dan Buku pada *Website Chinese Learning* memiliki konsep yang sama di mana keduanya dapat menambah, menghapus, mengubah, serta mencari setiap data yang telah tersimpan. Pada Gambar 3.11 dan Gambar 3.12 menunjukkan tampilan menu Buku dan Artikel.



Gambar 3.11 Tampilan *Menu Books*



Gambar 3.12 Tampilan *Menu Articles*

Penguji membuat skenario pengujian berdasarkan alur pada Gambar 3.10. Skenario pengujian ini dijadikan acuan dalam melakukan pengujian setiap elemen sesuai dengan kebutuhan dan hasil yang diharapkan pada sistem. Skenario pengujian dibuat ke dalam bentuk tabel yang mencakup informasi sebagai berikut.

ID	:	Kode spesifik yang berbeda-beda pada tiap kasus pengujian
Sub-modul	:	Sub dari modul utama yang akan diuji
Test Case Description	:	Deskripsi atau penjelasan kasus pengujian dari bagian yang akan diuji
Test Steps	:	Langkah-langkah dalam melakukan pengujian
Expected Result	:	Hasil yang diharapkan dari proses pengujian

Skenario pengujian pada menu *Article* dan *Books* dapat dilihat pada Tabel 3.7.

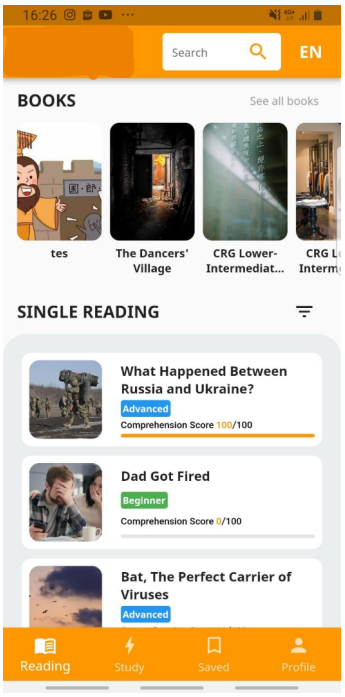
Tabel 3.7 Skenario Pengujian *Website* menu *Article* dan *Book*

ID	***Sub-Modul***	***Test Case Description***	***Test Steps***	***Acceptance Criteria***
W01	*Articles*	Menampilkan *list data article* yang tersimpan	1. Buka *website* 2. *Login as* admin 3. Masuk ke Menu *Article*	Berhasil menampilkan seluruh data artikel yang tersimpan

W02	*Articles*	Menambah artikel baru dengan mengisi seluruh *field* yang wajib diisi	1. Buka *website* 2. *Login as* admin 3. Masuk ke Menu *Article* 4. *Klik Add*	Berhasil menyimpan artikel baru
W03	*Articles*	Mengubah data artikel yang sudah ada	1. Buka *website* 2. *Login as* ″admin″ 3. Masuk ke Menu *Article* 4. Klik *Edit* pada salah satu artikel	Berhasil menyimpan perubahan pada Artikel
W04	*Articles*	Menghapus data artikel	1. Buka *website* 2. *Login as* ″admin″ 3. Masuk ke *Menu Article* 4. Klik *Delete* pada salah satu artikel	Berhasil menghapus artikel yang dipilih
W05	*Articles*	Mencari Artikel	1. Buka *website* 2. *Login as* ″admin″ 3. Masuk ke Menu *Article* 4.Masukkan *Keywords* yang ingin dicari pada *Search Bar*	Berhasil menampilkan hasil pencarian Artikel
W06	*Book*	Menampilkan *list data article* yang tersimpan	1. Buka *website* 2. *Login as* ″admin″ 3. Masuk ke Menu *Book*	Berhasil menampilkan seluruh data Buku yang tersimpan
W07	*Book*	Menambah buku baru dengan mengisi seluruh *field* yang wajib diisi	1. Buka *website* 2. *Login as* ″admin″ 3. Masuk ke Menu *Book* 4. *Klik Add*	Berhasil menyimpan Buku baru
W08	*Book*	Mengubah data buku yang sudah ada	1. Buka *website* 2. *Login as* ″admin″ 3. Masuk ke Menu *Book* 4. Klik *Edit* pada salah satu artikel	Berhasil menyimpan perubahan pada Buku
W09	*Book*	Menghapus data buku	1. Buka *website* 2. *Login as* ″admin″ 3. Masuk ke *Menu Book* 4. Klik *Delete* pada salah satu artikel	Berhasil menghapus Buku yang dipilih
W10	*Book*	Mencari Buku	1. Buka *website* 2. *Login as* ″admin″ 3. Masuk ke Menu *Book* 4.Masukkan *Keywords* yang ingin dicari pada *Search Bar*	Berhasil menampilkan hasil pencarian Buku

3.4.2 Alur Aplikasi Android Sistem Belajar Bahasa Mandarin

Setelah melalui tahapan penginputan dan pengolahan data oleh admin pada *Website Chinese Learning*, maka seluruh data bacaan akan ditampilkan di Aplikasi Android sesuai perubahan yang terjadi pada *Website* seperti pada Gambar 3.13.



Gambar 3.13 Tampilan *Menu Reading*

Pengguna dapat memanfaatkan bacaan yang terdapat pada *Menu Reading* untuk mempelajari Bahasa Mandarin secara jauh lebih mendalam karena pada setiap bacaan baik artikel maupun buku berisi paragraf Bahasa Mandarin yang dapat diartikan kedalam Bahasa Indonesia dan Inggris. Dalam setiap paragraf memiliki kumpulan kosakata yang berbeda-beda, oleh karena itu terdapat halaman kosakata yang dapat membantu pengguna mempelajari setiap huruf alphabet yang digunakan dalam membaca huruf Mandarin atau disebut dengan *pinyin* seperti pada Gambar 3.14.



Gambar 3.14 Halaman Kosakata

Pada Gambar 3.14 menunjukkan bahwa setiap kosakata dapat disimpan, ditinjau, dan memiliki kondisi keterangan warna yang berbeda-beda. Keterangan warna pada kosakata memiliki makna yang berbeda-beda, yaitu:

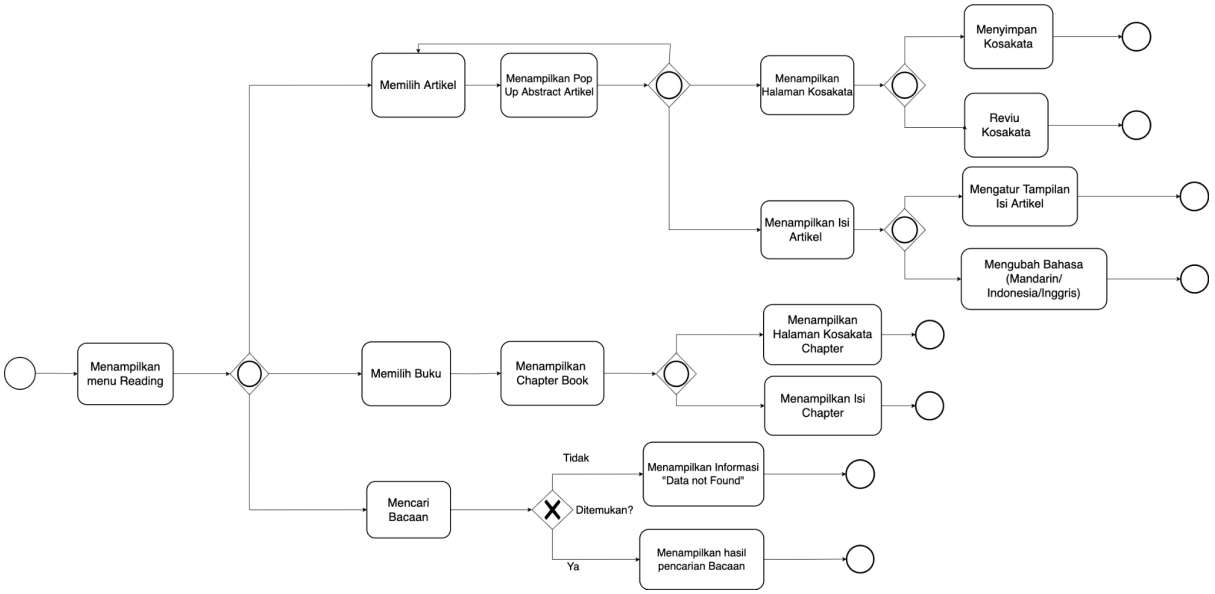
- Warna Abu-Abu, menunjukkan kondisi bahwa kosakata tersebut belum disimpan
- Warna Kuning, menunjukkan kondisi kosakata telah disimpan namun belum di *review*
- Warna Hijau, menunjukkan kosakata yang telah disimpan dan direviu

Tampilan Aplikasi *Android Chinese Learning* ketika melakukan reviu dapat dilihat pada Gambar 3.15 yang berupa *card view* yang menampilkan kosakata yang berbeda-beda.



Gambar 3.15 Tampilan reviu kosakata

Card tersebut dapat di*swipe* ke kanan ketika *user* sudah paham, dan *swipe* ke kiri ketika *user* belum paham. Berdasarkan penggunaannya, penguji membuat gambaran terkait alur yang dapat dilakukan pengguna dalam menggunakan setiap fitur pada *Menu Reading* seperti pada Gambar 3.16.



Gambar 3.16 Alur pengguna dalam menggunakan *Menu Reading*

Dengan adanya alur berdasarkan penggunaan *user* pada *menu Reading*, membantu penguji dalam membuat skenario pengujian pada setiap fitur secara keseluruhan. Skenario pengujian ini dijadikan acuan dalam melakukan pengujian setiap elemen sesuai dengan

kebutuhan dan hasil yang diharapkan pada sistem. Berikut skenario pengujian pada *Menu Reading* pada Tabel 3.8.

Tabel 3.8 Skenario Pengujian Android menu *Article* dan *Book*

ID	***Sub-Modul***	***Test Case Description***	***Test Steps***	***Expected Result***
A01	*Book*	Tampilkan *book* yang ada	1. *Login as user* 2. Klik *menu reading* 3. Menggeser kanan kiri *list book* 4. KIik Lihat semua buku	Dapat menampilkan daftar buku yang ada
A02	*Book*	Tampilkan *chapter* dari *book*	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book*	Dapat menampilkan *Chapter* dari Buku yang dipilih
A03	*Book*	Menampilkan halaman kosakata sesuai *chapter*	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik lingkaran yang didalamnya ada angka pada *list chapter*	Dapat menampilkan halaman kosakata sesuai dengan *chapter* yang dipilih
A04	*Book*	Dapat melakukan *filter* kosakata	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik lingkaran yang didalamnya ada angka pada list *chapter* 5. Klik *Filter* Daftar Kosakata	Dapat menampilkan hasil *filter* berdasarkan kriteria berikut: - *Default filter* adalah *unsaved vocab* - Kosakata yang belum disimpan berwarna abu-abu - Ketika sudah disimpan berwarna kuning - Ketika telah direview berwarna hijau - Jika seluruh kosakata telah disimpan / *direview*, maka dibagian *unsaved vocab* harusnya kosong
A05	*Book*	Dapat menambahkan kosakata ke Daftar Kosakata	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik *button* yang ada gambar 3 garis *horizontal* 5. Klik *add to word list*/tambahkan ke daftar kosakata	Dapat menyimpan kosakata pada *chapter* tersebut dengan kondisi seperti berikut. - Sebelum kosakata disimpan, kosakata masih berwarna abu-abu

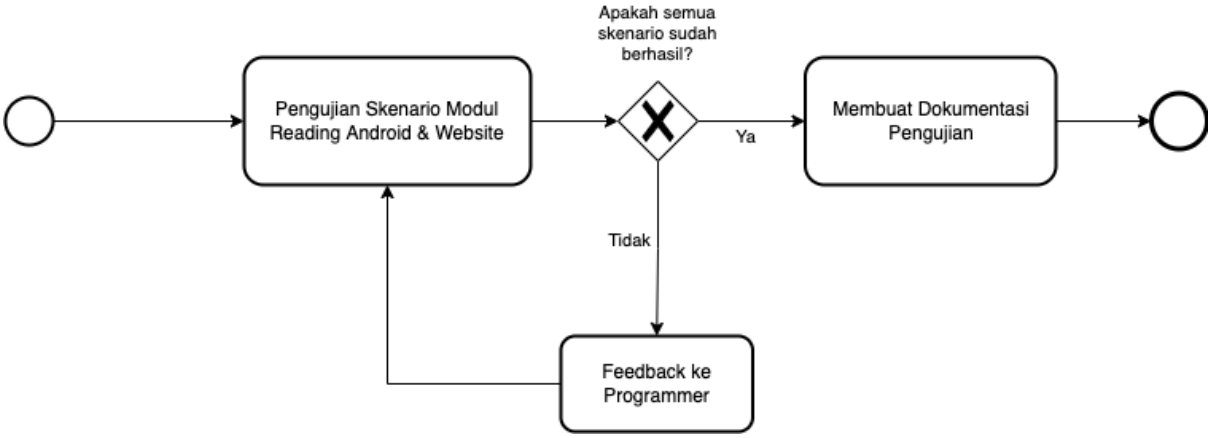
				- Ketika sudah disimpan namun belum reviu berwarna kuning - Ketika sudah disimpan, dan melakukan reviu pada kosakata tersebut berwarna hijau
A06	*Book*	Dapat melakukan *review* kosakata yang ada pada *chapter* tersebut	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik *button* yang ada gambar 3 garis *horizontal* 5. Klik Tinjau Kosakata 6. Klik tombol *switch* di sebelah kanan atas	Dapat melakukan *review* kosakata yang ada pada *chapter* tersebut dan ketika telah *direview* kata berubah menjadi hijau.
A07	*Book*	Tampilkan paragraf dari *chapter*	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik salah satu *chapter*	Dapat menampilkan paragraf dari salah satu *chapter*
A08	*Book*	Mengganti Bahasa pada paragraf salah satu *chapter*	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik salah satu *chapter* 5. Klik "A Z" 6. Pilih salah satu *page color*	Dapat mengubah Bahasa pada paragraf sesuai dengan Bahasa yang dipilih antara Inggris, Mandarin, dan Indonesia
A09	*Book*	Mengganti *background* pada paragraf salah satu *chapter*	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik salah satu *chapter* 5. Klik "A Z" 6. Pilih salah satu *page color*	Dapat mengganti *background* pada paragraf salah satu *chapter*
A10	*Book*	Merubah ukuran *font size* paragraf salah satu *chapter*	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik salah satu *chapter* 5. Klik "A-Z" 6. Pilih salah satu *Font Size*	Dapat merubah ukuran *font* pada *paragraf* salah satu *chapter*
A11	*Book*	Memainkan *audio* dari salah satu *chapter*	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *book* 4. Klik salah satu *chapter* 5. Klik *play audio*	Dapat memainkan *audio* dari salah satu *chapter*
A12	*Book*	Menandai *chapter* telah selesai dibaca	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu book 4. Klik salah satu *chapter* 5. Klik *Finished Reading*	Dapat menandai *chapter* yang telah dibaca

			6. Keluar *pop up* "*Already mark as finished*"	
A13	*Article*	Tampilkan *Article* yang ada	1. *Login as user* 2. Klik menu *reading* 3. Menggeser kanan kiri *list Article* 4. KIik Lihat semua *Article*	Dapat menampilkan daftar Artikel yang ada
A14	*Article*	Menampilkan halaman kosakata pada Artikel yang dipilih	1. *Login as user* 2. Klik menu *reading* 3. Klik salah satu *Article* 4. Klik lingkaran yang didalamnya ada angka pada *list chapter*	Dapat menampilkan halaman kosakata sesuai dengan Artikel yang dipilih
A15	*Article*	Dapat melakukan *filter* kosakata pada artikel tersebut	1. *Login as user* 2. Klik menu *reading* 3. Klik salah satu *Article* 4. Klik *Filter* Daftar Kosakata	Dapat menampilkan hasil *filter* berdasarkan kriteria berikut: - *Default filter* adalah *unsaved vocab* - Kosakata yang belum disimpan berwarna abu-abu - Ketika sudah disimpan berwarna kuning - Ketika telah *direview* berwarna hijau - Jika seluruh kosakata telah disimpan / *direview*, maka dibagian *unsaved vocab* harusnya kosong
A16	*Article*	Dapat menambahkan kosakata ke Daftar Kosakata	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *Article* 4. Klik *button* yang ada gambar 3 garis *horizontal* 5. Klik *add to word list*/tambahkan ke daftar kosakata	Dapat menyimpan kosakata pada artikel tersebut dengan kondisi seperti berikut. - Sebelum kosakata disimpan, kosakata masih berwarna abu-abu - Ketika sudah disimpan namun belum reviu berwarna kuning - Ketika sudah disimpan, dan melakukan reviu pada kosakata tersebut berwarna hijau
A17	*Article*	Dapat melakukan reviu kosakata	1. *Login as user* 2. Klik *menu reading* 3. Klik *salah* satu *Article* 4. Klik *button* yang ada gambar 3 garis *horizontal* 5. Klik Tinjau Kosakata	Dapat melakukan *review* kosakata yang ada pada *chapter* tersebut dan ketika telah *direview* kata berubah menjadi hijau.

A18	*Article*	Menampilkan Isi Paragraf Artikel	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *Article* 4. Klik Baca	Dapat menampilkan paragraf dari Artikel yang dipilih
A19	*Article*	Mengganti Bahasa pada paragraf artikel	1. *Login as user* 2. Klik *menu reading* 3. Klik salah satu *single reading* 4. Muncul *pop up single reading* 5. Klik tombol *read* 6. Klik + salah satu paragraf 7. Muncul *translate* dari paragraf	Dapat mengubah Bahasa paragraf sesuai dengan Bahasa yang dipilih antara Inggris, Mandarin, dan Indonesia

3.4.3 Alur dan Hasil Pengujian

Pengujian modul *Reading* diawali dengan pembuatan skenario pengujian. Pada pengujian *Regression Test Website Chinese Learning* akan berfokus pada alur pemasukan, serta pengelolaan setiap data *Articles* dan *Books* agar dapat berjalan sesuai dengan alur pada Gambar 3.10. Setelah seluruh pengujian pada *Website* berhasil, langkah selanjutnya adalah pengujian pada aplikasi Android *Chinese Learning* untuk memeriksa kesesuaian data antara *Website* dan Android. Selain itu, pengujian *Regression Test* pada aplikasi Android juga berfokus pada keberhasilan setiap proses pada Gambar 3.16. Ketika pengujian fitur secara terpisah pada Taiga dieksekusi secara satu per satu pada setiap *task* tanpa mempengaruhi proses yang terjadi pada *task* lainnya, *Regression Testing* dieksekusi dengan cara yang berbeda yaitu pengujian berpacu kepada skenario pengujian yang telah dibuat berdasarkan alur penggunaan *website* dan aplikasi *Android*. Seluruh *test case* yang telah dibuat dari rancangan alur harus berhasil sesuai dengan kriteria yang diharapkan agar dapat berpindah ke tahapan selanjutnya. Apabila terdapat ketidaksesuaian pada salah satu *test case*, maka proses pengujian akan dialihkan menjadi *Feedback* ke *Programmer* untuk diperbaiki dan diuji kembali ketika sudah diperbaiki. Alur pengujian dapat dilihat pada Gambar 3.17.



Gambar 3.17 Alur Pengujian

Setiap skenario dieksekusi hingga berhasil berjalan sesuai dengan kriteria yang telah ditentukan. Seluruh *feedback* yang didapatkan selama pengujian dibuat ke dalam bentuk tabel yang mencakup informasi sebagai berikut.

ID : Kode spesifik yang berbeda-beda pada tiap kasus pengujian

Status : Hasil yang diperoleh, diisi dengan "*PASSED*" apabila sesuai dan "*FAILED*" apabila tidak sesuai dengan *expected result*.

Result : Keterangan tambahan dalam pengujian. Dapat diisi ketika kondisi status "*Failed*" yang berarti berisi keterangan kondisi ketidaksesuaian maupun kondisi status "*Passed*" yang berarti kondisi pengujian sukses dengan beberapa catatan tambahan.

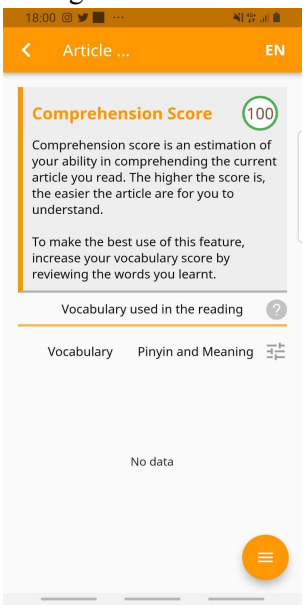
Follow Up : Tindak lanjut dari pengujian ketika status *test case* "*Failed*".

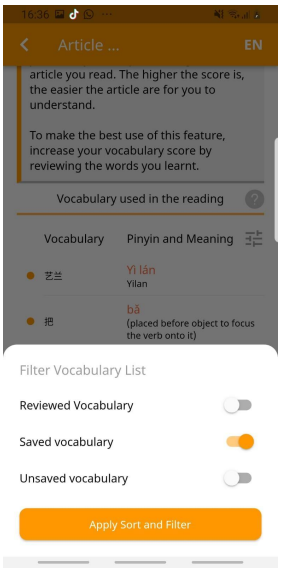
Hasil pengujian dan *feedback* yang didapatkan dapat dilihat pada Tabel 3.9.

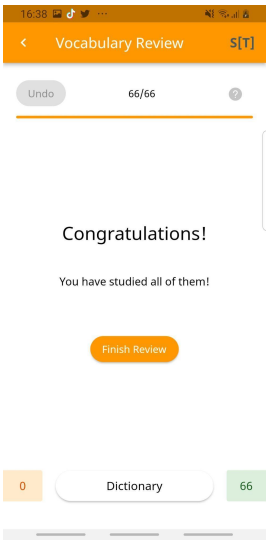
Tabel 3.9 Hasil Pengujian dan *Feedback*

ID	***Status***	***Result***	***Follow Up***
W01	*PASSED*	Berhasil menampilkan seluruh data yang tersimpan	
W02	*PASSED*	Berhasil dengan beberapa kondisi, - Kondisi 1, dapat menyimpan artikel baru ketika mengisi seluruh *field* wajib - Kondisi 2, gagal menyimpan artikel baru ketika tidak mengisi seluruh *field* wajib pada form pengujian	
W03	*PASSED*	Berhasil mengubah isian data pada artikel dan tersimpan sesuai dengan perubahan yang dilakukan	

W04	*PASSED*	Data artikel yang dipilih berhasil terhapus
W05	*PASSED*	Berhasil dengan beberapa kondisi: - Ketika penguji mencari sesuai dengan *keyword* artikel yang ada, maka artikel tersebut berhasil ditampilkan - Ketika penguji mencari keyword yang tidak sesuai dengan artikel yang ada, informasi yang ditampilkan berupa "*Data Not Found*"
W06	*PASSED*	Berhasil menampilkan seluruh data buku yang tersimpan
W07	*PASSED*	Berhasil dengan beberapa kondisi: - Kondisi 1, dapat menyimpan buku baru ketika mengisi seluruh *field* wajib - Kondisi 2, gagal menyimpan buku baru ketika tidak mengisi seluruh *field* wajib pada *form* pengujian
W08	*PASSED*	Berhasil mengubah isian data pada buku dan tersimpan sesuai dengan perubahan yang dilakukan
W09	*PASSED*	Data buku yang dipilih berhasil terhapus
W10	*PASSED*	Berhasil dengan beberapa kondisi: - Ketika penguji mencari sesuai dengan *keyword* sesuai dengan buku yang ada, maka buku tersebut berhasil ditampilkan - Ketika penguji mencari *keyword* yang tidak sesuai dengan buku yang ada, informasi yang ditampilkan berupa "*Data Not Found*"
A01	*PASSED*	Berhasil menampilkan seluruh data buku sesuai pada menu *Reading* sesuai dengan data buku pada bagian *website*
A02	*PASSED*	Berhasil menampilkan *chapter* sesuai dengan buku yang dipilih
A03	*PASSED*	Berhasil menampilkan halaman kosakata pada setiap *chapter* buku yang dipilih
A04	*PASSED*	Berhasil melakukan *filter* sesuai dengan kriteria yang ditentukan
A05	*PASSED*	Berhasil menyimpan kosakata pada masing-masing *chapter* yang dipilih sesuai dengan kriteria yang ditentukan

A06	*PASSED*	Berhasil melakukan *review* kosakata pada *chapter* yang dipilih sesuai dan setiap kata yang telah di *review* berubah tanda dari warna kuning ke hijau	
A07	*PASSED*	Berhasil menampilkan isi paragraf dari chapter yang dipilih	
A08	*PASSED*	Berhasil mengubah *setting* bahasa paragraf sesuai dengan bahasa yang dipilih antara Inggris, Mandarin, dan Indonesia	
A09	*PASSED*	Berhasil mengganti tampilan warna *background* saat membaca isi *chapter* yang dipilih	
A10	*PASSED*	Berhasil mengubah ukuran *font* pada paragraf *chapter* yang dipilih	
A11	*PASSED*	Berhasil memainkan *audio* dari salah satu *chapter* yang dipilih	
A12	*PASSED*	Berhasil menandai *chapter* yang telah dibaca	
A13	*PASSED*	Berhasil menampilkan daftar bacaan artikel	
A14	*FAILED*	Gagal menampilkan isi kosakata yang terdapat pada artikel, isi halaman kosakata kosong 	Setelah dilakukan perbaikan oleh Programmer, halaman kosakata berhasil menampilkan isi kosakata pada artikel yang dipilih. 
A15	*FAILED*	Tidak dapat melakukan *filter* karena kosakata pada *test case* A14 tidak tampil	Kosakata berhasil difilter.

			
A16	*FAILED*	Tidak dapat menambahkan kosakata ke Daftar Kosakata karena kosakata pada *test case* A14 tidak tampil	
A17	*FAILED*	Tidak dapat melakukan review kosakata karena kosakata pada *test case* A14 tidak tampil	Berhasil melakukan review kosakata 

			
A18	*PASSED*	Berhasil menampilkan isi paragraf pada artikel yang dipilih	
A19	*PASSED*	Berhasil mengganti bahasa paragraf pada artikel yang dipilih	

Seluruh pengujian *Regression Testing* dinyatakan berhasil dengan beberapa proses *feedback*. Adapun empat *test case* pada modul Reading yang dilakukan perbaikan karena gagal pada salah satu *test case* ketika dieksekusi yaitu gagal menampilkan isi kosakata yang menyebabkan halaman kosakata menjadi kosong pada *test case* A14. Kesalahan pada satu *test case* ini berpengaruh dengan tiga *test case* lainnya yaitu A15, A16,17 karena jika halaman kosakata kosong maka alur selanjutnya yaitu melakukan filter, menambah kosakata, dan melakukan *review* tidak akan berhasil dilakukan. Oleh karena itu dilakukan perbaikan hingga berhasil lalu dapat dilanjutkan ke proses selanjutnya. Pengujian *regression testing* dilakukan secara terus menerus selama masa *maintenance* sistem untuk memastikan setiap perubahan pada pengujian unit tidak berpengaruh pada keseluruhan alur sistem dan sistem dapat digunakan dengan baik dari awal hingga akhir.

BAB IV
REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Sub-bab ini menjelaskan terkait gap antara teori pada Bab II dengan pelaksanaan di lapangan selama proses magang berlangsung. Terdapat beberapa poin yang akan dibahas, yaitu terkait implementasi *black box* dalam pengujian perangkat lunak, penerapan Taiga pada pengujian dan *regression testing* dalam proyek.

4.1.1 Implementasi Pengujian Perangkat Lunak menggunakan Black Box

Terjun secara langsung menjadi seorang *Quality Assurance* (QA) yang melakukan pengujian perangkat lunak menjadi pengalaman baru yang sebelumnya belum pernah dilakukan dan dipelajari di masa perkuliahan. Ruang lingkup yang harus dikuasai sebagai QA dalam melakukan pengujian perangkat lunak dengan metode *black box* sangat luas. Sebelum melakukan pengujian, penguji harus mampu beradaptasi dengan penggunaan seluruh *tools* yang digunakan sistem. Tak hanya mempelajari *tools* dan sistem, pembelajaran dokumen *handbook* sistem yang berisi seluruh alur, kegunaan setiap fitur, standar operasional pada sistem yang dibangun juga dilakukan sebagai salah satu rangkaian pemahaman sebelum melakukan pengujian perangkat lunak menggunakan *black box*. Pada pelaksanaannya, penguji merasakan bahwa pemahaman yang baik bagi penguji terhadap sistem cenderung akan lebih mudah dalam melakukan pengujian secara manual karena mampu meningkatkan *human-sense* atau kepekaan serta paham dengan setiap aktivitas fungsional yang dimiliki sistem. Selain itu, penguji memiliki tanggung jawab sebagai QA untuk bertemu langsung dengan klien pada setiap akhir *sprint*. Penguji berperan untuk menceritakan serta mendemokan seluruh pengujian fungsional yang telah dilakukan pada setiap *sprint*.

4.1.2 Penerapan Taiga dan *Regression Testing* dalam Proyek

Dalam penerapannya, pengujian fungsional dan non fungsional pada sistem dilakukan secara terpisah berdasarkan *task* menggunakan *tools management* yaitu Taiga berdasarkan hasil analisis yang telah dilakukan pada setiap fitur. Adanya bantuan *tools management* seperti Taiga sangat membantu pengerjaan setiap progres sesuai dengan porsi *role* masing-masing. Pengujian pada Taiga membuat penguji dapat mengerjakan pengujian pada *task* yang telah tersedia tanpa menunggu *task* lainnya selesai terlebih dahulu. Penemuan dan perbaikan setiap *bug* yang ditemukan pada *task* jauh lebih cepat dikerjakan karena dapat

langsung disampaikan *feedback* kepada *Programmer* untuk langsung diperbaiki. Selain melakukan pengujian pada *task* yang diberikan, penguji bertanggung jawab untuk merangkum seluruh *task* yang telah dikerjakan pada setiap *sprint* ke dalam format dokumen *User Acceptance Test* (UAT) yang dimiliki perusahaan.

Setelah pengujian terpisah pada sisi fungsional dan non fungsional yang dilakukan di Taiga, dilanjutkan pengujian *Regression Test*. Sebelum melakukan pengujian *Regression Testing*, penguji perlu membuat dokumen UAT regresi yang berisi skenario keseluruhan aplikasi secara keseluruhan alur yang terpisah. Seluruh skenario disatukan ke dalam dokumen lalu dilakukan pengecekan secara berulang setiap terjadi perubahan berdasarkan skenario yang dibuat. Nyatanya walaupun *regression testing* termasuk pengujian ulang secara keseluruhan akibat perubahan yang terjadi pada pengujian sebelumnya, masih terdapat beberapa kesalahan yang terjadi ketika pengujian secara keseluruhan. Pada pelaksanaan di lapangan, *regression testing* memakan waktu yang lama dalam penerapannya karena masih dilakukan secara manual tanpa bantuan mesin otomasi pengujian dan harus dilakukan berkali-kali ketika terjadi ada perubahan atau kesalahan.

4.2 Pembelajaran Magang

Banyak pembelajaran yang telah didapatkan selama menjalani kegiatan magang dalam kurun waktu 6 bulan sebagai *Quality Assurance* atau *Tester*. Dari sisi *hard skill*, penguji belajar cara melakukan analisis kebutuhan atau permasalahan yang ada pada sistem, mempelajari setiap SOP dari penggunaan *tools management* hingga prosedur pengujian yang akan dilakukan, hingga mempertajam ketelitian penguji ketika melakukan pengujian. Banyak ilmu yang baru dipelajari mulai dari mempelajari setiap *tools* yang digunakan pada proyek, memahami prosedur yang dilakukan saat pengujian di mana diawali dengan analisis hingga mendokumentasikan hasil pengujian. Analisis yang dilakukan di awal sangat membantu penguji dalam memahami karakteristik dari sistem yang akan diuji. Adapun *soft skill* yang diasah ketika terjun langsung di lingkungan kerja perusahaan di antaranya adalah komunikasi yang dilakukan selama masuk ke dalam proyek terutama komunikasi antar tim internal maupun klien perusahaan. Walaupun bentuk komunikasi yang dilakukan selama magang menggunakan *Zoom Meeting* dengan kondisi *full Work From Home*, penguji sadar bahwa dengan menyalakan kamera dan berpakaian rapi merupakan salah satu bentuk etika dalam menghargai lawan bicara yang berada di *Zoom Meeting*. Penguji menemukan beberapa kondisi yang berperan besar dalam mengasah kemampuan komunikasi diantaranya ketika penguji diberikan *task* yang kurang jelas dan tidak sesuai, penguji berusaha untuk mencari

tahu dan bertanya untuk menemukan jawaban agar *issue* dapat terselesaikan. Pertanyaan yang disampaikan harus dapat disampaikan dengan jelas agar pertanyaan tersebut dapat diterima dan dimengerti dengan baik. Hal tersebut perlu dipersiapkan agar tidak membuang waktu pengerjaan *task* lebih lama. Selain itu, proyek yang penguji pegang harus berhubungan langsung dengan klien, membuat penguji dituntut untuk berani dalam membangun komunikasi seperti bertanya terkait informasi yang belum jelas, melakukan demo secara langsung kepada klien, serta mampu menerima kritik dan saran yang diberikan oleh klien.

Selama menjalani kegiatan magang, ada beberapa hambatan dan tantangan yang dialami. Di awal pengerjaan *task*, penguji sempat merasa kesulitan dalam melakukan pengujian secara terperinci dan hanya berpatokan terhadap alur skenario tanpa memahami masalah dan tujuan dari *task* tersebut. Hal ini disebabkan karena sebelumnya penguji belum pernah memiliki pengalaman melakukan pengujian baik secara manual maupun otomatis. Kesulitan juga dirasakan dalam memahami *tools* yang dipakai di dalam proyek yang belum pernah penguji gunakan sebelumnya seperti Taiga, DBeaver, dan Collab. Kesulitan tersebut seiring berjalan waktu dapat diatasi dengan belajar beradaptasi dan aktif dalam berkomunikasi ketika menemukan kesulitan. Selain itu, tantangan yang dihadapi sebagai seorang *Tester* pada proyek adalah ketika penguji harus mampu mengerjakan pengujian secara manual pada *task* maupun diluar *task* sesuai dengan tenggat waktu yang ditentukan. Dalam sehari dapat terdapat sekitar 10-20 *task* masuk untuk diuji. Pengujian yang dilakukan secara manual menantang penguji dalam menguji tingkat kepekaan terhadap sistem dalam menyadari adanya kesalahan pada sistem. Hal tersebut dapat ditingkatkan dengan mengeksplor sistem yang akan diuji dan kriteria capaian dari *task* yang diberikan. Seluruh proses tersebut memerlukan waktu untuk dikerjakan agar tidak terjadi kesalahan. Hal tersebut juga membuat penguji harus lebih cekatan dalam memahami setiap *task* agar tidak ada yang melewati tenggat waktu yang ditentukan.

BAB V
PENUTUP

5.1 Kesimpulan

Pengujian pada Sistem Belajar Bahasa Mandarin berbasis *website* dan android menggunakan metode *Black Box* telah dilakukan hingga seluruh *test case* berhasil sesuai dengan kriteria yang ditentukan. Rangkaian proses pengujian telah dilakukan dimulai dengan pengujian setiap bagian fungsional hingga non fungsional pada aplikasi menggunakan Taiga, lalu pengujian secara keseluruhan alur aplikasi atau *Regression Testing*. Dalam melakukan kedua jenis pengujian tersebut sangat penting untuk melakukan beberapa analisis di awal untuk menentukan capaian tenggat waktu yang dibutuhkan, aspek apa saja yang perlu diuji, batasan pengujian, hingga kriteria yang ingin dicapai serta keinginan dari *user*. Sebagai *tester*, memperbanyak eksplor sistem sebelum memulai pengujian akan sangat membantu memahami alur kerja yang dimiliki sistem.

Pengujian berdasarkan task Taiga cocok diterapkan pada proyek dengan jangka waktu cepat karena langsung berfokus pada kebutuhan yang di uji atau permasalahan yang ada. Adapun *regression testing* cocok digunakan untuk jangka waktu lama karena sifatnya seperti pengujian untuk *maintenance* yang dilakukan secara berkala dan berulang setiap terjadi perubahan pada sistem untuk memastikan setiap perubahan yang dilakukan tidak berpengaruh dengan keseluruhan alur.

5.2 Saran

Adapun saran untuk penelitian selanjutnya yaitu mengimplementasikan *regression testing* menggunakan salah satu *automation testing tools*. *Regression testing* pada dasarnya menguji alur kerja sistem atau penggunaan sistem dari awal hingga akhir. Hasil dari pengeksekusian uji regresi menggunakan *automation testing tools* dapat dibandingkan dengan uji regresi secara manual. Perbandingan dapat dilihat dari segi waktu dan seberapa banyak penemuan *bug* yang didapatkan selama pengujian.

DAFTAR PUSTAKA

Iskandaria (2012). Contoh Pengujian Black Box. https://www.kafegue.com/ContohPengujian-Black-Box-Testing/

Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (STUDI KASUS : APLIKASI PREDIKSI KELULUSAN SMNPTN). *Jurnal Ilmiah Teknologi Infomasi Terapan*, *1*(3). https://doi.org/10.33197/jitter.vol1.iss3.2015.62

Saputra, A. H. (2018, January 5). *PENGUJIAN PERANGKAT LUNAK DENGAN TECHNICAL REVIEW*. https://dspace.uii.ac.id/handle/123456789/5926

Suwirmayanti, N. L. G. P., I Komang Agus Ady Aryanto, Putra, I. G. A. N. W., Sukerti, N. K., & Hadi, R. (2020). Penerapan Helpdesk System dengan Pengujian Blackbox Testing. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, *2*(02). https://doi.org/10.46772/intech.v2i02.290

Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT*, *3*(1), 45–48. https://doi.org/10.30591/jpit.v3i1.647

Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (STUDI KASUS : APLIKASI PREDIKSI KELULUSAN SMNPTN). Jurnal Ilmiah Teknologi Infomasi Terapan, 1(3). https://doi.org/10.33197/jitter.vol1.iss3.2015.62

Eriana, E. sita. (2021). PENGUJIAN SISTEM INFORMASI APLIKASI PERPUSTAKAAN BERBASIS WEB DENGAN WHITE BOX TESTING. *Jurnal ESIT (E-Bisnis, Sistem Informasi, Teknologi Informasi)*, *15*(2). https://doi.org/13057

Mubarak, R. (2021). IMPLEMENTASI METODE WHITE BOX TESTING PADA PROSES QUALITY ASSURANCE PERANGKAT LUNAK BERBASIS WEB DAN MOBILE COLLECTION SYSTEM. *Jurnal ESIT (E-Bisnis, Sistem Informasi, Teknologi Informasi)*, *15*(1). https://doi.org/14044

Irawan, Y. (2017). Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja Upt. BLK Kabupaten Kudus dengan Metode Whitebox Testing. *SPEED - Sentra Penelitian Engineering Dan Edukasi*, *9*(3). https://doi.org/10.55181/speed.v9i3.353

Verma, A., Khatana, A., & Chaudhary, S. (2017). A comparative study of black box testing and white box testing. *International Journal of Computer Sciences and Engineering*, *5*(12), 301–304. https://doi.org/10.26438/ijcse/v5i12.301304

Wibisono, W., & Baskoro, F. (2002). PENGUJIAN PERANGKAT LUNAK DENGAN MENGGUNAKAN MODEL BEHAVIOUR UML. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *1*(1), 43. https://doi.org/10.12962/j24068535.v1i1.a95

Wahl, N. J. (1999). An overview of regression testing. *ACM SIGSOFT Software Engineering Notes*, *24*(1), 69–73. https://doi.org/10.1145/308769.308790

Rosero, R. H., Gomez, O. S., & Rodriguez, G. (2017, October). An approach for regression testing of database applications in incremental development settings. *2017 6th International Conference on Software Process Improvement (CIMPS)*. http://dx.doi.org/10.1109/cimps.2017.8169952

Biswas, S., & Mall, R. (2011). Regression Test Selection Techniques: A Survey. Informatica, 35, 289–321.

Azis, N., Pribadi, G., & Nurcahya, M. S. (2020). Analisa dan Perancangan Aplikasi Pembelajaran Bahasa Inggris Dasar Berbasis Android. *Ikraith-Informatika*, *4*(3), 1–5. https://doi.org/815

Marbun, Y. Y., Isnanto, R. R., & Martono, K. T. (2016). Pembuatan Aplikasi TOEFL sebagai Media Pelatihan Bahasa Inggris Berbasis Web. *Jurnal Teknologi Dan Sistem Komputer*, *4*(1), 83. https://doi.org/10.14710/jtsiskom.4.1.2016.83-92

Setiyani, L. (2019). PENGUJIAN SISTEM INFORMASI INVENTORY PADA PERUSAHAAN DISTRIBUTOR FARMASI MENGGUNAKAN METODE BLACK BOX TESTING. *Techno Xplore : Jurnal Ilmu Komputer Dan Teknologi Informasi*, *4*(1), 1–9. https://doi.org/10.36805/technoxplore.v4i1.539

Board, S. E. (2021, December 15). *Difference between regression and integration testing 2022*. Softwaretestingo Editorial Board. https://www.softwaretestingo.com/regression-and-integration-testing/

Wibowo, P. T. J. (2021, December 3). Apa Itu black-box testing? *Warta Ekonomi*. https://wartaekonomi.co.id/read378402/apa-itu-black-box-testing

Dhaifullah, I. R., H, M. M., Salsabila, A. A., & Yakin, M. A. (2022). Survei teknik pengujian software. *Journal Automation Computer Information System (JACIS)*, *2*(1), 31–38. https://doi.org/10.47134/jacis.v2i1.42

Sholeh, A. N., & Wardaya, M. S. S. (2019). ANALISIS DAN PENGUJIAN KERENTANAN SISTEM INFORMASI PERPUSTAKAAN. *Jurnal Mandiri : Ilmu Pengetahuan, Seni, Dan Teknologi*, *3*(1), 116–131. https://doi.org/10.33753/mandiri.v3i1.65