

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Organisasi

Selama melaksanakan kegiatan magang di PT Kompas Media Nusantara berlangsung, kedudukan yang ditempati adalah sebagai *Associate Quality Engineer Analyst Intern* pada divisi *Technology, Product, and Data*. Pada masa kerja magang, para anak magang di divisi TPD akan diawasi dan diberikan arahan, *feedback*, pelatihan, serta penilaian akhir oleh Yuni Putri Wulan Sari selaku supervisi. Segala Proses pengerjaan proyek dibimbing secara langsung oleh Christian Teguh selaku mentor *Quality Engineer Analyst* di *Squad E*. Pengerjaan setiap proyek dilakukan dalam satu atau beberapa *sprint* bersama seluruh anggota tim dengan tugas yang berbeda-beda, ada *developer* yang mengerjakan bagian *back end* dan *front end*, *Quality Engineer Analyst* yang bertugas menguji setiap fitur, serta ada *Leader* yang mengoordinasi setiap tugas-tugas yang harus dikerjakan dalam suatu proyek. *Squad E* bertanggung jawab dalam pengembangan fitur Pengelolaan *user* (*Guest/Anonymous, Register Only, dan Suber/Subscriber*) dalam halaman *sign in, sign up*, dan Kelola Akun, serta fitur penawaran paket-paket *membership kompas.id*.

#### 3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama pelaksanaan kegiatan magang di PT Kompas Media Nusantara, antara lain:

1. Melakukan pengujian beberapa *story* di Azure DevOps

Setiap *story* dalam satu siklus *sprint* memiliki waktu penyelesaian selama dua minggu, menyesuaikan dengan periode *sprint* tersebut. Saat *Story* pertama kali dibuat, *story* tersebut akan berada dalam kolom *New*, kemudian jika sudah mulai dikerjakan oleh *developer*, *story* akan digeser ke kolom *Active* oleh *developer* tersebut. Jika *story* sudah selesai dikerjakan dan di *review* oleh *Lead developer*, maka *story* tersebut akan digeser ke kolom *Ready to Test* oleh *developer*, disaat inilah *story* akan diuji oleh *Quality Engineer*. QE akan memindahkan *story* yang sudah siap untuk diuji ke kolom *Testing*. Pengujian dilakukan dengan memeriksa fungsionalitas yang telah ditentukan dalam *story* untuk memastikan bahwa setiap bagian telah memenuhi standar dan

berjalan sesuai dengan yang diharapkan. Saat pengujian selesai, *story* tersebut akan dipindahkan lagi ke kolom *On Review* oleh QE. Jika terdapat *story* yang belum terselesaikan dalam periode *sprint* tertentu, maka pengerjaan *story* tersebut akan dilanjutkan pada periode *sprint* berikutnya. Sistem ini dilakukan untuk memastikan bahwa setiap *task* terselesaikan secara bertahap dan berkelanjutan.

## 2. Membuat tiket *bug*

Ketika ditemukan masalah atau *bug* selama proses pengujian, langkah selanjutnya adalah membuat tiket *bug* di Azure DevOps. Tiket ini berfungsi sebagai dokumentasi resmi terkait kendala yang dihadapi, mulai dari deskripsi masalah, kondisi awal, *repro step*, *expected result*, bukti *bug*, hingga tingkat keparahannya (*Severity Level*). Proses ini memungkinkan tim pengembang untuk menerima masukan terkait isu yang ditemukan dan memprioritaskan perbaikan pada aspek-aspek yang krusial demi meningkatkan kualitas produk secara keseluruhan.

### 3.3 Uraian Pelaksanaan Magang

Berikut uraian pelaksanaan saat kerja magang di PT Kompas Media Nusantara.

#### 3.3.1 Aktivitas Setiap Minggu

Pekerjaan yang dilakukan tiap minggu selama pelaksanaan magang di Harian Kompas dapat dilihat pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu

Minggu ke -	Pekerjaan yang dilakukan
1	Berkenalan dengan lingkungan <i>Harian Kompas</i> , <i>set up</i> beberapa akun ( <i>Microsoft Teams</i> , <i>Postman</i> , dan <i>Azure DevOps</i> ) menggunakan akun <i>kompas</i> , memahami serta mempelajari <i>flow</i> kerja sebagai <i>Quality Engineer Analyst</i> di <i>Harian Kompas</i>

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu ke -	Pekerjaan Yang Dilakukan
2	<ul style="list-style-type: none"> <li>• Mulai memasuki Sprint 19</li> <li>• Mempelajari penggunaan Azure</li> <li>• Mempelajari metode Sprint</li> <li>• Mempelajari cara membuat test case</li> </ul>
3	<ul style="list-style-type: none"> <li>• Menguji story [BF] - Fix Buy Membership Without Login</li> <li>• Menguji notifikasi email untuk reminder H+8 paket berlangganan user habis masa berlaku</li> <li>• Mengecek API purchase token</li> <li>• Menguji story penambahan metadata dan fix logo Kompas.id</li> <li>• Menguji story add product type Kompas.com+</li> <li>• Mengikuti acara Code Prettier Refactor (CPR)</li> </ul>
4	<ul style="list-style-type: none"> <li>• Mulai memasuki Sprint 20</li> <li>• Melakukan Sprint Review, Planning, dan Retrospective bersama squad</li> <li>• Melakukan Regression Mobile untuk menguji fitur-fitur Kompas.id android version</li> <li>• Mengikuti acara Intensive Showcase Pairing (ISP)</li> <li>• Mengikuti acara Ngobrol Pakai Ilmu (NGOPI)</li> <li>• Melakukan pengujian bug dan membuat tiket bug di Azure</li> </ul>

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu ke -	Pekerjaan Yang Dilakukan
5	<ul style="list-style-type: none"> <li>• Mempelajari dan menguji alur fitur <i>seamless sign in and sign up through checkout page</i></li> <li>• Membuat dan menguji tiket bug Fix email yang didapatkan user Anon (tanpa login)</li> <li>• Membuat tiket bug Template email salah pada user subscriber</li> <li>• Membuat tiket bug email berbeda pada halaman pembayaran berhasil</li> <li>• Membuat tiket bug User tidak bisa aktivasi dengan email berbeda</li> <li>• Membuat dan menguji tiket bug Tanggal di Faktur Pembelian tidak real time</li> <li>• Membuat tiket bug Tautan Verifikasi Akun tidak berlaku</li> <li>• Membuat tiket bug Membership tidak masuk saat user sudah membeli paket berlangganan</li> <li>• Membuat tiket bug Fix flow user saat checkout dengan akun regon (tanpa login)</li> <li>• Membuat dan menguji tiket bug Flow user berlangganan dari halaman checkout menggunakan akun Regon (sudah terdaftar)</li> <li>• Meeting bersama anggota squad e untuk melakukan repro alur seamless login bersama-sama</li> <li>• Menguji story email berbeda pada halaman pembayaran berhasil</li> </ul>

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu ke -	Pekerjaan Yang Dilakukan
6	<ul style="list-style-type: none"> <li>• Mulai memasuki Sprint 21</li> <li>• Menguji story Check url Dynamic Menu</li> <li>• Menguji story Check New Wording TnC in Register Page &amp; Email Subscription</li> <li>• Menguji story Check back button ePaper</li> <li>• Menguji story Fix flow user saat checkout dengan akun regon (tanpa login)</li> <li>• Menguji story Template email salah pada user subscriber</li> <li>• Melakukan testing flow suber dan masih menemukan bug pada halaman onboarding</li> <li>• Melakukan testing flow magic linnk dan menemukan 2 bug</li> <li>• Menguji story check wordingan notification inbox</li> <li>• Mengerjakan regression mobile</li> </ul>
7	<ul style="list-style-type: none"> <li>• Deployment dan menguji proses seamless sign in dan sign up dari checkout page ke beta dan product</li> <li>• Menguji story Check Implement Footer fot OAuth User</li> <li>• Menguji story Check User Login Activity</li> <li>• Menguji story Check Notification Web Header Function</li> <li>• Membuat tiket bug Hapus Perangkat error di Aktivitas page</li> <li>• Backlog Grooming Squad E</li> </ul>

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu ke -	Pekerjaan Yang Dilakukan
8	<ul style="list-style-type: none"> <li>• Mulai memasuki Sprint 22</li> <li>• Menguji story Check Dynamic Wording SWG (Subscribe With Google)</li> <li>• Menguji story Check Template Email Subscription Reminder</li> <li>• Menguji story Hapus Perangkat error di Aktivitas page dan menemukan bug</li> <li>• Sprint Review, Retrospective, dan Planning</li> <li>• Membuat tiket bug dan menguji story Wording SWG di Halaman Berlangganan Belum Keupdate</li> <li>• Mengikuti acara 1on1 bersama Kak Wulan</li> <li>• Membuat tiket bug dan menguji story Email Subscription Reminder Tidak Berhasil Masuk</li> <li>• Mengerjakan regression mobile</li> <li>• Menemukan bug saat melakukan regression mobile pada fitur Tema Aplikasi</li> <li>• Menguji proses checkout menggunakan kupon untuk memastikan membership tetap berhasil didapatkan oleh user anon dan regon</li> <li>• Menemukan bug saat ingin checkout menggunakan kupon disc 3000 menggunakan metode pembayaran gopay</li> <li>• Menguji story Check API Available Coupon</li> </ul>

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu ke -	Pekerjaan Yang Dilakukan
9	<ul style="list-style-type: none"> <li>• Menguji dan menemukan bug pada story Check Dynamic Wording SWG</li> <li>• Deployment untuk Wording SWG di support site (untuk akun regon dan anon) ke beta dan product</li> <li>• Membuat tiket bug Membership tidak masuk pada User Regon</li> <li>• Membuat tiket bug Input Email masih ada di Halaman Checkout</li> <li>• Menguji story Check API Store IsSubOnboarding</li> <li>• Membuat tiket bug Membership dan Faktur Pembelian tidak masuk pada user Anon</li> <li>• Membuat tiket bug Membership tidak masuk pada user Regon</li> <li>• Menguji story Check New Attributes To Insider</li> <li>• Menguji proses checkout paket KDP dan bundling kompas.com</li> <li>• Menguji story Check API Notification Header</li> </ul>



Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu ke -	Pekerjaan Yang Dilakukan
10	<ul style="list-style-type: none"> <li>• Mulai memasuki Sprint 23</li> <li>• Menguji story membership dan Faktur Pembelian tidak masuk pada user Anon</li> <li>• Menguji story Membership tidak masuk pada user Regon</li> <li>• Melakukan sprint Review, Retrospective, dan Planning</li> <li>• Mengerjakan dan menemukan bug saat Regression Mobile</li> <li>• Menguji story Check Status User Kirim Opini for Guest User</li> <li>• Menguji story Kompas.id Logo error in Kirim Opini Page</li> <li>• Mengikuti acara HR wajib untuk anak magang (Self-Discovery: Charting Your Next Career Journey)</li> <li>• Membuat ppt untuk presentasi ISP Squad E</li> </ul>



Tabel 3.1 Pekerjaan yang dilakukan tiap minggu (lanjutan)

Minggu ke -	Pekerjaan Yang Dilakukan
11	<ul style="list-style-type: none"> <li>• Deployment bersama Squad e</li> <li>• Melakukan testing ulang untuk story Update Copy Notification Inbox for Subscription Reminder</li> <li>• Melakukan testing ulang untuk story Template email Subscription dikelola dari sre</li> <li>• Melakukan testing ulang untuk story Adjustment API Available Coupon today</li> <li>• Melakukan testing ulang untuk story Create API Store isSubOnboarding</li> <li>• Melakukan testing ulang untuk story Send new attributes to Insider</li> <li>• Memahami alur checkout membership paket bundling kompas.id dan kompas.com+</li> <li>• Menguji story Check User Berhasil Mendapatkan Email Active Subscription</li> <li>• Menguji story Check Implement Event Tracker Notification Header In Lit</li> <li>• Melakukan testing ulang untuk story Perbaikan login with facebook di halaman login (Oauth)</li> <li>• Melakukan testing ulang untuk story Implement Footer (Oauth)</li> <li>• Mengikuti acara ISP             <ul style="list-style-type: none"> <li>• Menguji story Check API Resend Email Activation Kompas.com</li> </ul> </li> <li>• Melakukan backlog Grooming Squad E</li> </ul>

### 3.3.2 Software yang Digunakan

Berikut beberapa perangkat lunak yang digunakan dalam proses pengujian fitur pengelolaan *user* dan *subscription* selama pelaksanaan magang:

1. Azure DevOps

*Azure DevOps* adalah sebuah platform yang menyediakan serangkaian fitur yang dapat membantu suatu tim dalam merencanakan, mengembangkan, mengelola, mengawasi, dan menguji proyek yang mereka miliki. *Tools* ini juga dapat digunakan untuk membuat *test case* dan *test plan*. Fitur *Sprints* pada *Azure* memungkinkan untuk segala *task* mulai dari proses pengerjaan sampai pengujian dapat terdokumentasi dengan rapi, sehingga semua skenario *task* yang perlu dikerjakan dan diuji tidak akan ada yang terlewat.

2. Postman

*Postman* adalah sebuah platform kolaborasi yang populer digunakan untuk menguji *Application Programming Interface (API)*. Terdapat fitur-fitur kolaborasi yang memungkinkan tim pengembang untuk dapat bekerja sama dalam melakukan pengembangan dan pengujian API. *Postman* juga memungkinkan para penguji untuk mengirimkan permintaan HTTP *GET*, *POST*, *PUT*, dan *DELETE*, dan lainnya ke server untuk menganalisis respons yang diterima.

3. Figma

*Figma* adalah suatu *software* yang dapat digunakan untuk membuat desain antarmuka pengguna (UI/UX) secara pribadi maupun kolaborasi. *Software* ini tidak hanya digunakan oleh para desainer saja, melainkan dapat digunakan oleh para QE sebagai acuan dalam melakukan pengujian suatu fitur, seperti memastikan *layout*, tampilan, dan elemen fitur tersebut telah sesuai dengan desain yang telah dibuat di *Figma*. Pada *software* ini dapat dibuat *flowchart* yang mampu membantu QE dan tim pengembang untuk lebih mudah memahami alur suatu fitur yang ingin dibuat. *Figma* memungkinkan untuk semua anggota tim memiliki pemahaman yang sama terkait fitur yang akan dibuat sehingga diharapkan mulai dari proses pembuatan, pengembangan, sampai pengujian memiliki titik keberhasilan yang sama.

4. Yopmail

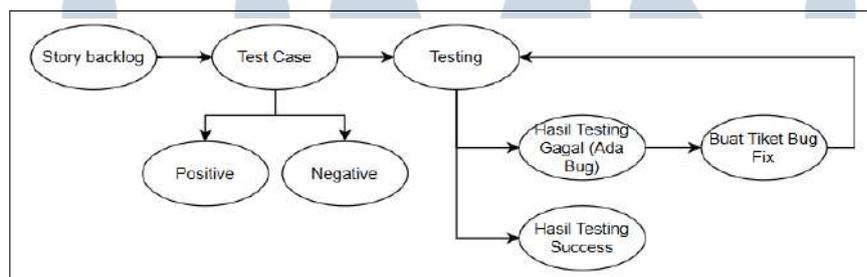
*Yopmail* adalah suatu layanan yang dapat digunakan untuk membuat email sementara tanpa perlu registrasi atau bayar terlebih dahulu. *Yopmail* dapat membantu QE untuk menguji fitur-fitur yang memerlukan penggunaan email, seperti saat melakukan simulasi *register* dan *login* akun, serta menguji apakah *user* berhasil mendapatkan email verifikasi dan *subscription* sesuai dengan kondisi *user* saat membeli paket langganan. Risiko penggunaan data pribadi saat melakukan pengujian dapat diminimalisir dengan menggunakan *yopmail* ini.

### 3.4 Proyek yang dikerjakan

Bagian ini menjelaskan proses pengerjaan proyek yang menjadi tanggung jawab selama magang. Penjabaran meliputi tahapan perancangan hingga implementasi yang dilakukan, dengan disertai dokumentasi berupa tangkapan layar (*screenshot*) dan penjelasan detail untuk mendukung hasil kerja magang di PT Kompas Media Nusantara.

#### 3.4.1 Alur Proses Testing

Proses *testing* suatu fitur atau aplikasi yang dilakukan oleh *Quality Engineer* memiliki beberapa tahap yang dapat dilihat pada Gambar 3.1.



Gambar 3.1. Alur Proses Testing

Pengujian ini diawali dengan adanya suatu *story backlog* yang telah ditetapkan oleh PM untuk dikerjakan pada *sprint* tertentu. *Story* ini akan dikerjakan oleh para *developer*, kemudian seorang QE akan membuat *test case* untuk digunakan sebagai poin pengujian. Skenario *test case* yang dibuat ada dua macam, yakni skenario positif dan negatif, menyesuaikan dengan kondisi desain dan kapasitas fitur yang akan diuji.

Ketika *story* telah selesai didevelop, akan dilakukan pengujian terhadap *test case* yang telah dibuat sebelumnya. Jika hasil *testing* tahap pertama ini ditemukan *bug* atau *error*, maka QE akan membuat tiket *bugfix* yang berisi informasi-informasi mengenai *bug* yang ditemukannya itu. *Story bugfix* ini akan dikembalikan ke *developer* sebagai panduan untuk memperbaiki *bug* tersebut. Setelah selesai diperbaiki, QE akan melakukan pengujian ulang untuk memastikan fitur tersebut telah sesuai dengan ketentuan. Proses pengujian ini akan terus berulang sampai QE berhasil mendapatkan hasil *testing* yang sukses.

### 3.4.2 Perancangan

Terdapat tahapan awal dalam pengerjaan proyek, yaitu proses perancangan yang dilakukan pada saat *Sprint Planning*. Perancangan ini mencakup pengumpulan dan pengorganisasian kebutuhan proyek yang diwujudkan dalam bentuk *story*, serta pembuatan *test case* sebagai acuan utama dalam proses pengujian.

#### A. Story pada Setiap Sprint

Pada metodologi *Agile* atau *Scrum*, terdapat *backlog* yang merupakan serangkaian tugas, fitur, atau pekerjaan yang telah dirancang pada saat *sprint planning* untuk mencapai tujuan proyek secara bertahap. Pada setiap *backlog* terdapat beberapa *story* yang memiliki atribut yang digunakan untuk membantu tim dalam mengelola pekerjaan menjadi lebih terorganisir dan menentukan prioritas setiap pekerjaan berdasarkan kebutuhan proyek. Beberapa atribut *story*, antara lain:

1. ID atau Nomor Unik

Setiap *story* memiliki nomor unik yang dapat digunakan untuk mengidentifikasi *story* tersebut dan mempermudah untuk dilacak. Contoh nomor uniknya seperti User Story 110719, Task 110720, dan Bug 110721.

2. State

Atribut ini menunjukkan tahap pengerjaan sebuah *story* sudah sampai mana, berikut beberapa tahapannya:

- (a) New, maknanya *story* belum mulai dikerjakan.

- (b) Active, maknanya *story* sedang dikerjakan namun belum selesai.

- (c) Resolved, maknanya *story* sudah selesai dikerjakan oleh *developer* dan sedang menunggu dicek ulang oleh *leader* dari sisi kodingan dan data Github.
- (d) Ready to Test atau Merged, maknanya *story* sudah selesai dicek oleh *leader* dan di *merged* oleh *developer*, sehingga sudah dapat dilakukan pengujian oleh *Quality Engineer*.
- (e) Testing, maknanya *story* sedang diuji oleh *Quality Engineer* atau QE dan dipastikan pekerjaan sudah sesuai dengan ketentuan yang telah diharapkan (*Acceptance Criteria*).
- (f) Closed, maknanya *story* sudah benar dan selesai diuji oleh QE serta *leader*. Proses *closed story* dilakukan saat *Sprint Review* oleh *leader*.

### 3. Tipe Story

Terdapat tiga tipe *story*, yaitu *User Story*, *Task*, dan *Bug*. *Story* dengan tipe *User Story* berisi pekerjaan atau fitur yang diinginkan oleh *user* dan ditulis dengan bahasa sederhana agar dapat dipahami oleh stakeholder, tim, dan *user*. *Story* dengan tipe *Task* berisikan langkah-langkah spesifik dari fitur atau pekerjaan yang telah ditetapkan di *User Story*. *Story* dengan tipe *Bug* berisikan kesalahan atau masalah yang ditemukan selama melakukan pengujian *Task* yang telah dikerjakan oleh para *developer*, seperti *error*, malfungsi, atau ketidaksesuaian dengan spesifikasi yang telah ditetapkan di *Acceptance Criteria* pada *User Story*.

### 4. Assigne

Menunjukkan atau menentukan siapa yang bertanggung jawab atas pengerjaan setiap item *story* yang ada. Penentuan ini akan ditentukan oleh *leader*.

### 5. Priority

Atribut ini menunjukkan tingkat seberapa penting suatu *story* untuk segera diselesaikan. Terdapat empat kategori, yaitu:

#### (a) Kategori 1

Menunjukkan prioritas tertinggi (*High*), sehingga setiap pekerjaan berupa fitur baru, perbaikan *bug*, maupun peningkatan sistem yang memiliki prioritas 1 harus segera dikerjakan dan diselesaikan sesegera mungkin.

(b) Kategori 2

Menunjukkan prioritas sedang (*Medium*). Pekerjaan tetap harus diselesaikan dan dikerjakan, namun pekerjaan dengan kategori ini dapat ditunda jika ada pekerjaan yang memiliki kategori 1.

(c) Kategori 3

Menunjukkan prioritas rendah (*Low*). Setiap pekerjaan bersifat opsional waktu penyelesaiannya berdasarkan sumber daya, waktu, dan risiko.

(d) Kategori 4

Menunjukkan prioritas terendah (*unimportant*). Setiap *story* dengan prioritas 4 berisi pekerjaan atau masalah (*bug*) yang pada dasarnya tidak terlalu memengaruhi produk, sehingga dapat dikerjakan dan diselesaikan setelah semua pekerjaan dengan tingkat prioritas 1, 2, dan 3 selesai.

6. Story Points

Poin ini memperkirakan waktu yang dibutuhkan oleh para *developer* yang bertanggung jawab dalam penyelesaian suatu *story*. Umumnya, terdapat empat macam poin yang digunakan, yaitu 1, 3, 5, dan 8 yang menunjukkan berapa jam yang dibutuhkan *developer* untuk mengerjakan *story* tersebut, namun di PT Kompas Media Nusantara, khususnya di *squad e*, poin-poin ini tidak digunakan untuk menunjukkan berapa lama waktu yang dibutuhkan untuk menyelesaikan suatu *story*, melainkan digunakan sebagai acuan seberapa berat *story* tersebut, semakin tinggi angkanya maka *story* tersebut dianggap cukup sulit atau rumit untuk dikerjakan. Hal ini sangat krusial karena dapat memengaruhi kinerja para *developer*, jika terdapat *developer* yang hanya mengambil *story* yang poinnya kecil saja terus menerus maka *developer* tersebut bisa saja dianggap memiliki kinerja yang tidak cukup baik.

7. Description

Berisikan penjelasan detail mengenai *item* yang harus dikerjakan dalam suatu *story*, baik itu tugas, fitur, atau *bug*. Deskripsi ini bertujuan untuk membantu tim memahami tujuan, konteks, dan detail penting yang dibutuhkan untuk menyelesaikan suatu *task story*.

8. Acceptance Criteria

Berisikan kriteria atau kondisi yang harus dipenuhi agar suatu *story* dianggap

selesai dan telah sesuai dengan ketentuan. *Acceptance criteria* terdapat pada semua tipe *story* yaitu *User Story*, *Task*, dan *Bug*. Kriteria ini ditentukan oleh *Product Manager* atau PM.

Selama pelaksanaan magang di PT Kompas Media Nusantara, pengujian *story backlog* dimulai dari sprint 19. Berikut beberapa *story* yang telah dilakukan pengujian selama kerja magang:

#### 1. Sprint 19

- (a) [Reminder Habis Langganan] Adjustment wording email reminder habis langganan

*Story* ini berisikan *task* untuk mengirimkan email pengingat (*reminder*) kepada para pelanggan yang masa tenggang paket berlangganannya telah habis atau berakhir (H+8). Pada email terdapat beberapa informasi yang disampaikan seperti masa periode langganan, periode masa tenggang langganan, dan semua akses yang didapatkan dari langganan yang pernah dibeli sudah tidak dapat digunakan lagi.

- (b) [BF][Kirim Opini] - Status user yang ditampilkan pada Header Kirim Opini sebagai Guest

*Story* ini berisi *task* untuk menyesuaikan kemunculan *button* Masuk pada *header* halaman kirim opini dengan status pengguna saat itu. *Button* Masuk hanya akan muncul jika status pengguna *Guest* atau *Anonymous (Anon)*. Jika status pengguna sudah *Regon (Register Only)* atau *Suber (Subscriber)*, maka *button* Masuk seharusnya tidak muncul lagi.

#### 2. Sprint 20

- (a) Update Copy TnC on Register Page

*Story* ini berisi *task* untuk memperbarui *wording* TnC yang berada di halaman Daftar Akun.

- (b) [QA] Magic Link

*Story* ini berisi *task* untuk membuat fitur *magic link* yang dapat digunakan pengguna untuk membeli paket berlangganan dengan lebih mudah dan efisien. Melalui adanya *magic link*, pengguna tidak perlu membuka banyak tab untuk melakukan *register* atau *login* terlebih dahulu, melainkan pengguna dapat langsung membeli paket langganan.

Jika sudah selesai *payment*, pengguna akan menerima email berisi *magic link* yang jika di klik secara otomatis akan mengarahkan pengguna untuk melakukan *login* ataupun *register* akun. Hal ini dapat memberikan keuntungan bagi pengguna untuk dapat mengaktifkan *membership* yang telah dibelinya ke akun pribadi atau orang lain.

### 3. Sprint 21

(a) [Subscription] Dynamic Wording on SWG Button

*Story* ini berisi *task* untuk memperbarui *wording* judul dan deskripsi pada *button* SWG (*Subscribe With Google*) di halaman berlangganan (*support site*) dan *paywall*.

(b) [Subscription][BE] Template email Subscription dikelola dari sre

*Story* ini berisi *task* untuk mengirimkan email *reminder* kepada para pelanggan dengan tiga kondisi. Pertama, kondisi ketika masa berlaku *membership* pelanggan akan berakhir 5 hari lagi (H-5). Kedua, ketika masa berlaku *membership* telah berakhir namun masih diberikan kesempatan untuk memasuki masa tenggang (H+1). Ketiga, ketika pelanggan berada di hari terakhir masa tenggang *membership* (H+7).

### 4. Sprint 22

(a) [Promotion] Adjustment API Available Coupon today

*Story* ini berisi *task* untuk menambahkan API *productId* sehingga dapat menunjukkan kupon apa saja yang tersedia di hari saat pengguna ingin membeli paket langganan di halaman *checkout*.

(b) [Subscription] Create API Store *isSubOnboarding*

*Story* ini berisi *task* untuk membuat API Store *isSubOnboarding* yang digunakan untuk melacak dan mengelola status *onboarding* pengguna. Jika akun pengguna sudah pernah melewati halaman *onboarding*, maka API *isSubOnboarding* akan berisi *true* dan akun pengguna tidak akan mendapatkan halaman *onboarding* lagi.

### 5. Sprint 23

(a) [Support Site] Notification Header

*Story* ini berisi *task* untuk memindahkan fitur notifikasi dari menu ke header *kompas.id* yang hanya akan muncul jika *user* sudah login.

(b) [Support Site] Event Tracker page\_viewed Magic Link

*Story* ini berisi *task* untuk menampilkan *event tracker page\_viewed* di halaman Konfirmasi Paket Langganan. *Tracker* ini disesuaikan dengan saat pengguna ingin membeli paket langganan, kemudian memasukkan email di halaman *checkout*.

## B. Test Case

Sebelum melakukan pengujian terhadap fitur-fitur yang telah dibuat atau dikembangkan, perlu adanya rancangan yang disebut *test case* terlebih dahulu. *Test Case* diawali dengan mendefinisikan fitur yang sedang diuji, kemudian membuat skenario dari fitur yang ingin diuji. Setelah itu, dilanjutkan dengan membuat alur pengujian dengan menggunakan *Gherkin Scenario* yang dinyatakan dalam format *Given, When, Then*. Format *Given* menjelaskan kondisi awal *user* sebelum tindakan pengujian dilakukan, format *When* menjelaskan tindakan atau peristiwa yang memicu respons sistem, dan format *Then* menjelaskan hasil yang diharapkan setelah tindakan dilakukan. Semua proses pengujian dilakukan secara manual seperti yang tertera pada kolom *Automation Coverage*. Berikut beberapa *test case* yang dibuat saat menguji *story* pada *sprint* 19 sampai *sprint* 23:

### 1. Story [Reminder Habis Langganan] Adjustment wording email reminder habis langganan

*Test case* untuk menguji apakah *user* berhasil mendapatkan email *remainder* H+8 atau tidak dapat dilihat pada Gambar 3.2.

Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
Email Reminder Habis Langganan	Sending email to expired users	Given user subscriber And the subscription was expired And user enter the grace period When the grace period end Then user receives a reminder email	Manual	Email is only sent to users whose grace period has expired	Subscriber	Positive
Email Reminder Habis Langganan	Displaying subscription details in email	Given user subscriber And the grace period was over When the email reminder is sent Then the email contains the correct subscription period and grace period details	Manual	Correct subscription and grace period details are displayed	Subscriber	Positive

Gambar 3.2. Test Case Email Reminder H+8

Terdapat dua *Test Scenario* yang dirancang dengan dua fokus spesifik yang berbeda untuk memastikan apakah *story* benar-benar telah sesuai dengan yang telah diharapkan. *Scenario* pertama digunakan untuk menguji apakah *user* berhasil mendapatkan email *reminder* H+8 atau tidak. Proses *testing* dimulai ketika sistem mendeteksi bahwa status *membership user* telah kedaluwarsa dan memasuki masa tenggang (*Grace Period*). Ketika masa

tenggang *user* berakhir, maka sesuai dengan hasil yang diharapkan (*expected result*), *user* akan menerima email *reminder* (H+8).

*Senario* kedua digunakan untuk menguji apakah isi dari email *reminder* yang dikirim ke *user* sudah sesuai dengan yang telah ditentukan oleh PM. Proses *testing* ini diawali dengan *user* *suber* memiliki *membership* yang masa tenggangnya telah berakhir. Ketika *user* mendapatkan email *reminder*, maka berdasarkan *expected result*, email akan berisi informasi akurat mengenai *membership* dan periode masa tenggang *user*. Kedua *scenario testing* ini diuji secara manual atau langsung oleh QE, tanpa menggunakan alat otomatisasi. Tipe pengguna yang diuji adalah pengguna yang telah memiliki *membership* atau berlangganan (*subscribe*). Kedua skenario yang diuji adalah *positive case type*, yakni ketika pengujian dilakukan menggunakan input valid atau sesuai dengan ketentuan untuk memastikan suatu fitur berfungsi sebagaimana mestinya atau normal.

2. Story [BF][Kirim Opini] - Status user yang ditampilkan pada Header Kirim Opini sebagai Guest

*Test case* untuk menguji apakah *button* Masuk yang ada di *header* kompas.id tetap muncul walaupun status *user* saat itu adalah *user register only* atau *subscription* dapat dilihat pada Gambar 3.3.

Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
Kirim Opini	Verify "Masuk" button appears for Guest users	Given user anonymous When user open Kirim Opini page Then the "Masuk" button is displayed in the header	Manual	"Masuk" button appears in header for anonymous users	Anonymous	Positive
Kirim Opini	Verify "Masuk" button is hidden for logged-in users	Given user is already login (as Regon or Suber) When user open Kirim Opini page Then the "Masuk" button is not displayed in the header	Manual	"Masuk" button does not appear for logged-in users	Regon/Subscribe	Positive

Gambar 3.3. Test Case Button Masuk pada Halaman Kirim Opini

*Test case* ini juga memiliki dua skenario yang sama-sama menguji fitur bernama Kirim Opini. Skenario pertama untuk memastikan *button* Masuk hanya akan terlihat pada pengguna dengan status *anonymous* atau *anon*. Pengujian ini diawali dengan tipe *user anonymous* yang memasuki halaman kirim opini, kemudian sesuai dengan *expected result*, *user* akan menemukan *button* Masuk di bagian *header*. Skenario kedua memastikan *button* Masuk tidak akan terlihat pada pengguna yang telah melakukan *login*. Pengujian ini diawali dengan *user regon* maupun *suber* memasuki halaman kirim opini, sesuai dengan *expected result*, maka *user* tidak akan menemukan *button* Masuk lagi karena mereka telah melakukan *login*. Kedua skenario positif ini diuji secara manual.

### 3. Story Update Copy TnC on Register Page

*Test case* untuk menguji apakah *wording* TnC di halaman daftar akun telah diperbaharui dan sesuai dengan ketentuan dapat dilihat pada Gambar 3.4.

Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
Wording TnC on Register Page	TnC words updated	Given user anon When user open Register page Then the updated TnC wording is displayed correctly	Manual	Updated TnC wording is displayed as per the requirement	Anonymous	Positive

Gambar 3.4. Test Case Wording TnC pada Halaman Daftar Akun

Skenario pengujian yang dibuat hanya satu saja karena fitur yang diuji memiliki fungsi yang sederhana, sehingga hanya memerlukan satu langkah pengujian. Proses pengujian diawali dengan *user anon* memasuki halaman *register*, kemudian *wording* TnC yang ada pada halaman *register* tersebut sesuai dengan desain figma. Skenario positif ini diuji secara manual.

### 4. Story Magic Link

Fitur *Magic Link* dibuat untuk mengatasi kendala yang mengharuskan *user guest* atau *anonymous* melakukan *sign in* atau *sign up* terlebih dahulu di halaman terpisah saat pengguna ingin membeli *membership*. Hal ini berpotensi menurunkan persentase pengguna untuk membeli paket berlangganan *kompas.id*. Fitur ini dibuat untuk pengguna yang membeli langganan dalam kondisi *anonymous* atau *anon*. *Test case* yang telah dibuat untuk melakukan pengujian *story* ini dapat dilihat pada Gambar 3.5.

No	Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
1	Magic Link	Anonymous user buy package membership via magic link with Register Only email	Given user anonymous And user on checkout page And user enter register only email on field Email And user click button Lanjut Pembayaran When user finish transaction Then user get an Active Subscription email	Manual	User get magic link email for subscription already active after payment membership with Register Only email success	Anonymous	Negative
2	Magic Link	Validation of Transaction Date on Purchase Invoice	Given user anonymous And user buy membership And user finish transaction And user enter Konfirmasi Paket Langganan page When user click Linkah Faktur Pembelian And user open the file Then user see the transaction date according to the membership purchase time	Manual	The transaction date on the purchase invoice must match the time the membership purchase was made	Anonymous	Positive
3	Magic Link	Anonymous user get membership after purchasing a subscription package with Anonymous or Register Only email	Given user anonymous And user on checkout page And user enter anonymous email on field Email And user click button Lanjut Pembayaran And user finish transaction And user enter Konfirmasi Paket Langganan page When user enter register using the anonymous package email used during the membership checkout And user enter Success of homepage Then user get membership	Manual	After login or registration is complete, the membership will be immediately added to the user's account	Anonymous	Positive
4	Magic Link	Anonymous user activate membership with different account	Given user anonymous And user buy membership And user get email magic link And user open Success in new tab And user login with regon account When user click button Aktivas Langganan in email And user enter Aktivas Paket Langganan page And user click button Aktivas Dengan Email Lain Then user enter login page	Manual	When user click the Aktivas Dengan Email Lain button, user will be directed to the login page and successfully activate the membership they purchased to the account used during login/register	Anonymous	Negative

Gambar 3.5. Test Case Magic Link

*Test case* ini dibuat untuk menguji fitur *Magic Link* yang tersedia dalam proses pembelian *membership*, seperti pada skenario pertama untuk menguji kondisi saat pengguna membeli *membership* menggunakan email *register only* atau *regon*, apakah pengguna akan mendapatkan email yang sesuai

dengan yang telah ditetapkan atau tidak. Lalu skenario kedua untuk menguji kondisi ketika pengguna sudah membeli *membership* dan ingin mengunduh faktur pembeliannya, apakah semua data yang ada dalam faktur tersebut sesuai dengan keadaan saat pembelian *membership* dilakukan atau tidak. Kondisi ketiga untuk menguji apakah pengguna berhasil mendapatkan *membership* yang telah dibelinya setelah berhasil *login* atau *register* menggunakan akun yang sama saat di halaman *checkout*. Terakhir, kondisi keempat dilakukan pengujian untuk memastikan apakah pengguna dapat mengaktivasi *membership* yang telah dibelinya ke akun yang berbeda dengan yang dimasukkan saat di halaman *checkout* sebelumnya.

Semua tipe *user* yang diuji adalah *user anonymous*. Skenario pertama dan keempat merupakan tipe skenario negatif karena skenario ini menguji fitur *magic link* menggunakan input yang tidak valid atau kondisi yang tidak sesuai dengan spesifikasi untuk menguji bagaimana fitur atau aplikasi meresponsnya. Seperti pada skenario pertama ketika *user* memasukkan email akun *regon* di halaman *checkout*, hal ini tentu tidak sesuai karena kondisi normalnya adalah *user anon* akan memasukkan email akun *anon* juga saat membeli *membership* di halaman *checkout*. Lalu ketika *user* ingin mengaktivasi *membership* miliknya menggunakan akun yang berbeda saat *checkout*, hal ini juga tidak sesuai karena normalnya *user* akan mengaktivasi *membership* miliknya ke akun yang sama dengan saat *user checkout*. Skenario kedua dan ketiga merupakan tipe skenario positif karena memang kondisi normalnya seperti itu.

##### 5. Story [Subscription] Dynamic Wording on SWG Button

*Test case* untuk menguji *wording* SWG *button* dapat dilihat pada Gambar 3.6.

No	Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
1	Dynamic Wording on SWG Button	Updating button wording SWG on support site	Given user is on the subscription page. When SWG button is displayed Then the button title and description show the updated wording	Manual	SWG payment method on the support site displays the updated wording	Anon/Regon	Positive
2	Dynamic Wording on SWG Button	Updating button wording SWG on paypal	Given user is on the subscription paypal When the SWG button is displayed Then the button title and description show the updated wording	Manual	SWG payment method on the paypal displays the updated wording	Anon/Regon	Positive

Gambar 3.6. Test Case Dynamic Wording SWG

Terdapat dua kondisi yang perlu dilakukan pengujian terkait *story* ini, sehingga dibuatlah *test case* untuk menguji saat *user* melihat metode *payment* SWG di halaman berlangganan (*support site*) dan saat *user* menemukan SWG di *paypal*. Pengujian skenario pertama diawali dengan *user anon* atau *regon* memasuki halaman berlangganan, kemudian saat melihat *button*

SWG, judul dan deskripsi pada *button* tersebut sesuai dengan desain di figma. Lalu pengujian pada skenario kedua diawali dengan *user anon* atau *regon* berada di halaman berlangganan *paywall*, kemudian melihat *button* SWG dan memastikan apakah informasi atau deskripsi pada *button* tersebut telah sesuai dengan desain figma. Kedua *test case* ini termasuk positif karena fitur ini cukup sederhana dan hanya memerlukan dua *test case* saja.

6. Story [Subscription][BE] Template email Subscription dikelola dari sre  
*Test case* untuk menguji apakah *user* berhasil mendapatkan *email reminder* jika *membership* miliknya akan selesai masa berlakunya 5 hari lagi (H-5), *membership* sudah habis masa berlaku dan memasuki hari tenggang (H+1), serta *membership* selesai memasuki masa tenggang (H+7) dapat dilihat pada Gambar 3.7.

No	Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
1	Email Reminder Subscription	Sending reminder email on H-5 membership end	Given user suber When the subscription will expire in 5 more days Then user receives a reminder email	Manual	Email reminder is sent for users whose membership expires in 5 days	Expiring Users	Positive
2	Email Reminder Subscription	Sending reminder email on H+1 membership end	Given user suber When the subscription already expired one day Then user receives a reminder email	Manual	Email reminder is sent for users in H+1 grace period	Expired Users	Positive
3	Email Reminder Subscription	Sending reminder email on H+7 membership end	Given user suber When the last day of the grace period Then user receives a reminder email	Manual	Email reminder is sent for users on the last grace period day (H+7)	Expired Users	Positive
4	Email Reminder Subscription	Displaying subscription details in email H-5	Given user suber And the membership was over When the email reminder is sent Then the email contains the correct details for H-5 case	Manual	Correct subscription details are displayed for H-5 remind	Expiring Users	Positive
5	Email Reminder Subscription	Displaying subscription details in email H+1	Given user suber And the membership was over When the email reminder is sent Then the email contains the correct details for H+1 case	Manual	Correct subscription details are displayed for H+1 remind	Expired Users	Positive
6	Email Reminder Subscription	Displaying subscription details in email H+7	Given user suber And the grace period was over When the email reminder is sent Then the email contains the correct details for H+7 case	Manual	Correct subscription details are displayed for H+7 remind	Expired Users	Positive

Gambar 3.7. Test Case Email Reminder Subscription

Terdapat 6 kondisi positif yang dilakukan *testing*, yakni kondisi saat *user* berhasil menerima *email reminder* untuk H-5, H+1, dan H+7, juga pengujian untuk memastikan apakah isi dari *email reminder* yang diterima oleh masing-masing *user* sudah sesuai dengan desain figma yang telah ditentukan oleh PM atau belum. Semua tipe *user* yang diuji adalah *user* dengan masa berlaku langganan telah selesai. Semua kondisi yang diuji merupakan kondisi positif karena cakupan pengujian yang dapat dilakukan terbatas hanya pada kondisi normal, dan juga karena memang tidak ada desain untuk alur negatif dari PM seperti pada *story Magic Link*.

7. Story [Promotion] Adjustment API Available Coupon today  
*Test case* untuk menguji apakah *API ProductId* berhasil menunjukkan kupon-

kupon apa saja yang dapat digunakan setiap harinya dapat dilihat pada Gambar 3.8.

No	Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
1	API Available Coupon Today	API returns the list of available coupons	Given the user is on the checkout page When the API is called to fetch available coupons for today Then the API returns the correct list of available coupons along with the corresponding productid.	Manual	API response includes valid coupons and productid for the current day	Anon/Regon	Positive
2	API Available Coupon Today	API returns the list of unavailable coupons	Given there are no coupons available for today When the API is called to fetch available coupons Then the API returns an zero in list	Manual	API response returns zero in the list of coupons not available today	Anon/Regon	Positive

Gambar 3.8. Test Case Adjustment API Available Coupon Today

Proses pengujian fitur *API Available Coupon Today* diawali saat user *anon* maupun *regon* berada di halaman *checkout*, *API productId* akan langsung memberitahu kupon-kupon apa saja yang dapat digunakan pada saat itu juga dan untuk kupon yang belum bisa digunakan pada saat itu akan ditunjukkan dengan memberikan nilai nol (0) pada *API productId*. Kedua skenario ini merupakan tipe skenario positif yang menguji fitur sesuai dengan kondisi normal yang diharapkan dan telah ditentukan oleh PM.

#### 8. Story [Subscription] Create API Store isSubOnboarding

*Test case* untuk menguji apakah *API isSubOnboarding* benar akan terisi *true* jika *user* sudah pernah melewati *onboarding page* dan terisi *false* jika belum pernah melewatinya dapat dilihat pada Gambar 3.9.

No	Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
1	API Store isSubOnboarding	User has completed onboarding	Given the user is in the onboarding page When the API is called to store the isSubOnboarding flag Then the system sets API isSubOnboarding = true	Manual	The API response returns "true" for users who have completed onboarding	Suber	Positive
2	API Store isSubOnboarding	User has not completed onboarding	Given the user has not accessed the onboarding page When the API isSubOnboarding is called Then the response returns API isSubOnboarding = false	Manual	The API response returns "false" for users who have not completed onboarding	Suber	Positive

Gambar 3.9. Test Case API Store isSubOnboarding

Pembuatan API ini ditujukan agar sistem dapat melacak dan mengelola status *onboarding user*, sehingga *user* tidak akan melewati halaman *onboarding* lagi untuk kedua kali dan seterusnya. Pengujian fitur *API Store isSubOnboarding* dilakukan pada dua skenario, ketika pengguna telah melewati halaman *onboarding* dan ketika pengguna belum pernah melewati halaman *onboarding*. Pada skenario pertama diawali dengan kondisi pengguna berada di halaman *onboarding*, kemudian API akan dipanggil untuk menyimpan tanda pada *API isSubOnboarding* sehingga menghasilkan nilai *true*.

Skenario kedua diawali dengan kondisi pengguna belum pernah mengakses halaman *onboarding*, maka API akan dipanggil untuk diisi dengan nilai *false*. Tipe pengguna yang digunakan selama proses pengujian adalah *user*

*suber*, karena hanya pengguna yang telah memiliki *membership* saja yang bisa mengakses halaman *onboarding*. Kedua skenario ini termasuk dalam skenario positif, menyesuaikan spesifikasi fitur yang telah ditetapkan oleh PM, yakni API *isSubOnboarding* berisi nilai *true* atau *false* sesuai dengan kondisi yang telah ditetapkan.

#### 9. Story [Support Site] Notification Header

*Test case* untuk menguji apakah fitur *notification* berhasil dipindah dari menu ke *header* untuk *user* yang sudah *login* kompas.id dapat dilihat pada Gambar 3.10.

No	Feature	Test Scenario	Step to Reproduce / Test Case	Automation Coverage	Expected Result	User Type	Case Type
1	Notification Header	Displaying notification header for logged-in users	Given the user is logged in When the user accesses the Support Site page Then the notification is displayed in the header	Manual	Notification header appears correctly for logged-in users	Registered	Positive
2	Notification Header	Hiding notification header for Guest users	Given the user is a Guest or Anonymous When the user accesses the Support Site page Then the notification is not displayed in the header	Manual	Notification header does not appear for Guest users	Guest	Positive
3	Notification Header	Removing notification from menu	Given the user is logged in When the user accesses the Support Site page And user opens menu Then the notification feature is not displayed in the menu	Manual	Notification is not visible in the menu for logged-in users	Registered	Positive

Gambar 3.10. Test Case Notification Header

Fitur notifikasi yang semulanya berada di bagian menu, dipindah ke bagian *header* dan hal ini hanya akan ditemukan jika *user* dalam kondisi sudah *login* (*user Regon* dan *Suber*) saja. Halaman *Support Site* merupakan halaman berlangganan yang berisikan paket-paket *membership* yang ditawarkan oleh kompas.id. Perpindahan ini mengharuskan penghapusan fitur notifikasi yang awalnya berada di menu, sehingga fitur notifikasi tidak *double* (Ada di menu dan ada juga di *header*).

Skenario pertama pada pengujian fitur *Notification Header* ini dilakukan ketika *user* yang sudah *login* dan mengakses halaman berlangganan akan melihat fitur *notification* di *header* halaman tersebut. Skenario kedua dilakukan ketika *user* masih dalam keadaan *guest* dan mengakses halaman berlangganan, maka *user* tidak akan melihat fitur notifikasi di *header* halaman tersebut. Skenario ketiga memiliki alur yang serupa dengan skenario pertama. Namun, pada skenario ketiga ini, *user* melakukan pengujian tambahan dengan membuka menu untuk memastikan bahwa notifikasi hanya muncul di *header* dan tidak lagi muncul di dalam menu. Semua skenario yang diuji merupakan skenario positif karena *story* ini hanya mencakup alur kerja normal, sehingga pengujian disesuaikan.

#### 10. Story [Support Site] Event Tracker page\_viewed Magic Link

*Test case* untuk menguji apakah *event tracker page\_view* sudah memberikan

kode *user\_type* yang sesuai dengan kriteria yang telah ditetapkan oleh PM dapat dilihat pada Gambar 3.11.

No	Feature	Test Scenario	Step to Reproduce   Test Case	Automation Coverage	Expected Result	User Type	Case Type
1	Event Tracker Page Viewed Magic Link	Triggering event tracker for anonymous email when user views Confirmation page	Open user is on the Checkout page And user entry anonymous email on field Email When user see Konfirmasi Paket Langganan Page Then system will track event <i>page_viewed</i> with <i>page_title</i> = Konfirmasi Paket Langganan Page	Manual	The <i>page_viewed</i> event tracker assigns <i>user_type</i> = G for users with an anonymous email when checking out a membership	Anon	Positive
2	Event Tracker Page Viewed Magic Link	Triggering event tracker for register only email when user views Confirmation page	Open user is on the Checkout page And user entry register only email on field Email When user see Konfirmasi Paket Langganan Page Then system will track event <i>page_viewed</i> with <i>page_title</i> = Konfirmasi Paket Langganan Page	Manual	The <i>page_viewed</i> event tracker assigns <i>user_type</i> = S for users with an register only email when checking out a membership	Anon	Negative
3	Event Tracker Page Viewed Magic Link	Triggering event tracker for subscription email when user views Confirmation page	Open user is on the Checkout page And user entry an already subscription email on field Email When user see Konfirmasi Paket Langganan Page Then system will track event <i>page_viewed</i> with <i>page_title</i> = Konfirmasi Paket Langganan Page	Manual	The <i>page_viewed</i> event tracker assigns <i>user_type</i> = S for users with a subscription email when checking out a membership	Anon	Negative

Gambar 3.11. Test Case Event Tracker *page\_views*

Proses pengujian dilakukan ketika *user* dalam keadaan *anon* dan membeli paket langsung di halaman *checkout*. Kemudian, *user* memasukkan email dengan kondisi yang berbeda-beda, yakni email yang masih baru dan belum terdaftar sama sekali (*anon*) maupun email yang sudah terdaftar namun belum memiliki *membership* (*regon*), dan email yang sudah terdaftar serta memiliki *membership* aktif (*suber*). Jika *user* memasukkan email *anon* sebelumnya, maka saat memasuki halaman Konfirmasi Paket Langganan, *event tracker page\_views* akan memberikan kode *user\_type* = G, jika *user* memasukkan email *regon* ataupun *suber*, maka *user* akan mendapatkan kode *user\_type* = S. Pengujian email *regon* dan *suber* dianggap sebagai *test case negative* karena kondisi ini tidak sesuai dengan ekspektasi namun memungkinkan untuk dilakukan oleh *user*.

### 3.4.3 Implementasi

#### A. Tabel Summary

Setelah membuat *test case*, QE siap untuk melakukan pengujian secara langsung. Berdasarkan hasil uji, ada dua kemungkinan yang didapatkan, yaitu hasil uji *success* yang menandakan *story* telah sesuai dengan ketentuan dan hasil uji gagal yang menandakan adanya *error* atau spesifikasi yang kurang sesuai dengan yang telah ditentukan. Maka dari itu, dibuatlah tabel yang berisi ringkasan (*summary*) dari hasil setiap *story* yang telah dilakukan pengujian. Tabel *summary* ini dapat dilihat pada Tabel 3.2.

Tabel 3.2. Table Summary

No.	Story	Bug	Severity	Testing Tahap 1	Testing Tahap 2
1	[Reminder Habis Langganan] Adjusment wording email reminder habis langganan	Tidak Ada	-	Pass	-
2	[BF][Kirim Opini] - Status user yang ditampilkan pada Header Kirim Opini sebagai Guest	Tidak Ada	-	Pass	-
3	Update Copy TnC on Register Page	Tidak Ada	-	Pass	-
4	[QA] Magic Link	Ada	Medium, High	Failed	Pass
5	[Subscription] Dynamic Wording on SWG Button	Ada	Low	Failed	Pass
6	[Subscription] Template email Subscription dikelola dari sre	Ada	High	Failed	Pass
7	[Promotion] Adjustment API Available Coupon today	Tidak Ada	-	Pass	-
8	[Subscription] Create API Store isSubOnboarding	Tidak Ada	-	Pass	-
9	[Support Site] Notification Header	Tidak Ada	-	Pass	-

Tabel 3.2 Table Summary (lanjutan)

No.	Story	Bug	Severity	Testing Tahap 1	Testing Tahap 2
10	[Support Site] Event Tracker page_viewed Magic Link	Ada	Low	Failed	Pass

## B. Tiket Bugfix

Pada tahap pengujian, beberapa *bug* berhasil diidentifikasi dan dianalisis untuk memastikan kualitas fitur yang dikembangkan sesuai dengan kebutuhan pengguna. Berikut adalah beberapa contoh *bug* yang ditemukan selama magang di PT Kompas Media Nusantara:

### 1. Story [QA] Magic Link

Pada *Story Magic Link* ditemukan cukup banyak *bug* sehingga perlu dibuatkan tiket *bugfix* nya. Berikut beberapa *bug* yang ditemukan:

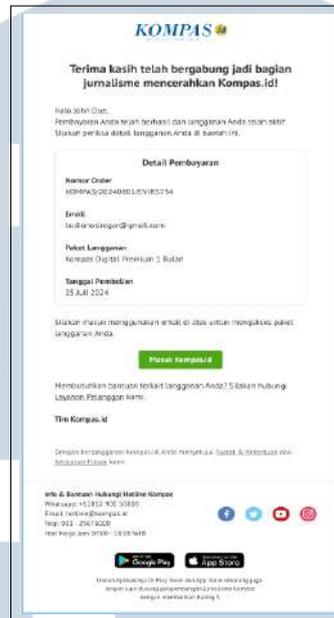
#### (a) [BF] - Fix flow user saat checkout dengan akun (tanpa login)

Kasus ini berisi kondisi *bug* saat pengguna anon membeli paket berlangganan langsung dari halaman *checkout* menggunakan email *regon* tanpa melakukan *login* terlebih dahulu, namun saat sudah selesai *payment*, *user* justru mendapatkan email aktivasi *membership*. *Story* ini memiliki tingkat *severity Medium* dan dapat dilihat pada Gambar 3.12.

<b>Description :</b>	<ul style="list-style-type: none"> <li>Email active subscription tidak masuk setelah user berhasil melakukan pembayaran.</li> </ul>
<b>Pre-Condition :</b>	<ul style="list-style-type: none"> <li>User anon masuk ke halaman checkout tanpa login</li> <li>User anon membeli paket berlangganan dengan email yang sudah terdaftar (Regon)</li> </ul>
<b>Repro Steps :</b>	<ul style="list-style-type: none"> <li>User anon langsung memasuki halaman checkout tanpa login terlebih dahulu</li> <li>User membeli paket yang diinginkan menggunakan email yang sudah terdaftar (Regon)</li> <li>User melanjutkan proses pembayaran</li> <li>User berhasil melakukan pembayaran</li> <li>User melihat halaman pembayaran sukses</li> <li>User call email</li> <li>User mendapatkan email active subscription</li> </ul>
<b>Actual Result :</b>	<ul style="list-style-type: none"> <li>user salah mendapatkan email, seharusnya yang didapat email active subscription tapi yang di dapat malah email aktivasi langganan</li> </ul>
<b>Expected Result :</b>	<ul style="list-style-type: none"> <li>seharusnya user mendapatkan email active subscription dan tidak mendapatkan email aktivasi langganan</li> </ul>

Gambar 3.12. Tiket bugfix user gagal mendapatkan email subscription active

Seharusnya email yang didapat oleh *user* adalah email seperti Gambar 3.13.



Gambar 3.13. Expected Result email yang seharusnya didapat user dengan email regon

(b) [BF] - Tanggal di Faktur Pembelian tidak real time

*Story* ini memiliki tingkat *severity Medium* dan dapat dilihat pada Gambar 3.14.

<p><b>Description :</b></p> <ul style="list-style-type: none"> <li>Tanggal pembelian paket berlangganan tidak sesuai dengan tanggal yang ada di faktur pembelian</li> </ul> <p><b>Pre-Condition :</b></p> <ul style="list-style-type: none"> <li>user anon masuk ke halaman beli paket berlangganan tanpa login</li> </ul> <p><b>Repro Steps :</b></p> <ol style="list-style-type: none"> <li>1. user anon masuk ke halaman berlangganan</li> <li>2. user anon pilih paket yang ingin dibeli</li> <li>3. user anon masuk ke halaman checkout</li> <li>4. user anon masukkan email yang belum terdaftar</li> <li>5. user anon selesaikan transaksi</li> <li>6. user anon masuk ke halaman pembayaran berhasil</li> <li>7. user anon masuk ke halaman konfirmasi paket langganan</li> <li>10. user cek faktur pembelian</li> </ol> <p><b>Actual Result :</b></p> <ul style="list-style-type: none"> <li>Tanggal pembelian paket berlangganan tidak sesuai dengan real time saat waktu pembelian</li> </ul> <p><b>Expected Result :</b></p> <ul style="list-style-type: none"> <li>Tanggal pembelian paket berlangganan sesuai dengan real time saat waktu pembelian</li> </ul>
--

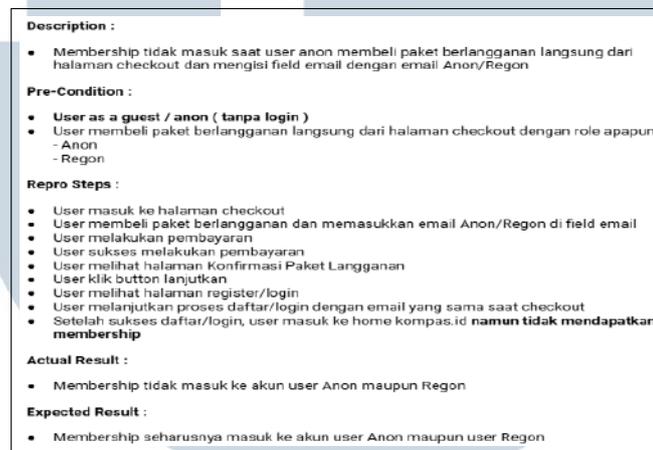
Gambar 3.14. Tiket bugfix tanggal di faktur pembelian tidak real time

*Bug* ini berisikan kondisi saat *user anonymous* membeli paket langganan, namun setelah selesai membayar dan *user* mengunduh faktur pembelian yang terdapat di halaman konfirmasi paket langganan, tanggal pada faktur tersebut tidak sesuai *real time* dengan saat

pembelian *membership* itu dilakukan. *Expected Result story* ini seharusnya tanggal di faktur pembelian sama dengan saat pembelian itu dilakukan.

(c) [BF] - Membership tidak masuk saat user sudah membeli paket berlangganan

*Story* ini memiliki tingkat *severity High* dan dapat dilihat pada Gambar 3.15.



Gambar 3.15. Tiket bugfix user gagal mendapatkan membership

*Bug* ini berisikan kondisi saat *user anon* membeli paket langganan langsung di halaman *checkout* tanpa *login* terlebih dahulu dan memasukkan email *user anon* atau *regon* di *field* email. Saat *user* sudah menyelesaikan tahap *payment* lalu memasuki halaman konfirmasi paket langganan dan menyelesaikan proses *register* atau *login*, *user* tidak mendapatkan *membership* yang telah dibeli tersebut. Seharusnya, saat *user* memasuki halaman *home* kompas.id setelah menyelesaikan proses *register* atau *login*, *user* akan mendapatkan *membership* mereka.

(d) [BF] - User tidak bisa aktivasi dengan email berbeda

*Story* ini memiliki tingkat *severity Medium* dan dapat dilihat pada Gambar 3.16.

<p><b>Description :</b></p> <ul style="list-style-type: none"> <li>• user tidak bisa aktivasi dengan email berbeda</li> </ul> <p><b>Pre-Condition :</b></p> <ul style="list-style-type: none"> <li>• user anon masuk ke halaman checkout tanpa login</li> </ul> <p><b>Repro Steps :</b></p> <ul style="list-style-type: none"> <li>• 1. user masuk ke halaman checkout</li> <li>2. user masukkan email anon di field email</li> <li>3. user selesaikan transaksi</li> <li>4. user masuk ke halaman pembayaran berhasil</li> <li>5. user menerima email magic link</li> <li>6. user open new tab kompas dan login akun regon</li> <li>7. user klik button aktivasi langganan</li> <li>8. user masuk ke halaman aktivasi paket langganan</li> <li>9. user klik button aktivasi dengan email lain</li> <li>10. user masuk ke halaman bad gateway</li> </ul> <p><b>Actual Result :</b></p> <ul style="list-style-type: none"> <li>• user masuk ke halaman bad gateway</li> </ul> <p><b>Expected Result :</b></p> <ul style="list-style-type: none"> <li>• user masuk ke halaman daftar akun</li> </ul>
--

Gambar 3.16. Tiket bugfix user tidak bisa aktivasi dengan email berbeda

*Bug* ini berisikan kondisi setelah *user* anon membeli paket langganan kemudian membuka *new tab* kompas.id dan *login* dengan akun *regon*, lalu *user* membuka email *magic link* yang diterima dan klik *button magic link* tersebut untuk aktivasi langganan. Ketika *user* memasuki halaman aktivasi paket langganan dan *user* klik *button* aktivasi dengan email lain, *user* memasuki halaman *bad gateway*. Seharusnya setelah *user* klik *button* aktivasi dengan email lain, *user* memasuki halaman *login* akun.

Selain empat tiket *bugfix* di atas, terdapat beberapa tiket *bugfix* lainnya yang telah dibuat saat pengujian *story Magic Link* ini, antara lain:

- (a) [BF] - Template email salah pada user subscriber
- (b) [BF] - Tautan Verifikasi Akun tidak berlaku
- (c) [BF] - Email berbeda pada halaman pembayaran berhasil
- (d) [BF] - Flow user berlangganan dari halaman checkout menggunakan akun Regon (sudah terdaftar)

## 2. Story [Subscription] Dynamic Wording on SWG Button

Pada saat pengujian *story* ini ditemukan *bug* yaitu *wording* SWG belum terupdate dari JSON yang telah dibuat sebelumnya. *Story bugfix* ini memiliki *severity Low* dan dapat dilihat pada Gambar 3.17.

<p><b>Description :</b></p> <ul style="list-style-type: none"> <li>• wording SWG di halaman berlangganan belum terupdate dengan desain yang telah dibuat di Json</li> </ul> <p><b>Pre-Condition :</b></p> <ul style="list-style-type: none"> <li>• user anonymous</li> </ul> <p><b>Repro Steps :</b></p> <ol style="list-style-type: none"> <li>1. user masuk ke halaman checkout</li> <li>2. user melihat paket langganan Subscribe With Google (SWG)</li> <li>3. user melihat wording paket SWG belum terupdate</li> </ol> <p><b>Actual Result :</b></p> <ul style="list-style-type: none"> <li>• Wording SWG pada halaman berlangganan masih error. Wordingan tidak keupdate di <a href="#">Berlangganan Berita Premium Kompas.id</a> walaupun sudah update untuk json nya</li> </ul> <p><b>Expected Result :</b></p> <ul style="list-style-type: none"> <li>• Wording SWG sudah keupdate dan sesuai dengan desain figma</li> </ul>
--

Gambar 3.17. Tiket bugfix wording SWG belum terupdate

Wording SWG sebenarnya sudah berhasil diperbarui atau diubah di JSON, namun saat dicek secara tampilan *frontend*, ternyata *wording* untuk SWG tersebut belum diperbarui, sehingga perlu dilakukan perbaikan lebih lanjut dan dibuatkan tiket *bugfix* nya.

3. Story [Subscription] Template email Subscription dikelola dari sre  
 Pada saat proses pengujian *story* ini, ditemukan *bug* dengan kondisi isi email *subscription reminder* untuk *5 days before exp (H-5)* dan *1 day after exp (H+1)* tidak berhasil masuk ke email *user*, sedangkan untuk email H+7 berhasil diterima oleh *user*. *Story bugfix* ini memiliki *severity High*. Oleh karena itu, perlu dibuatkan tiket *bugfix* terkait hal ini yang dapat dilihat pada Gambar 3.18.

<p><b>Description :</b></p> <ul style="list-style-type: none"> <li>• Template email Subscription Reminder untuk 5 days before exp (H-5) dan 1 day after exp (H+1) tidak ada isinya / kosong saat diterima oleh user</li> </ul> <p><b>Pre-Condition :</b></p> <ul style="list-style-type: none"> <li>• user suber</li> </ul> <p><b>Repro Steps :</b></p> <ul style="list-style-type: none"> <li>• Given user suber</li> <li>• When masa berlaku membership user habis sesuai dengan ketentuan (5 hari lagi/sudah habis 1 hari sebelumnya) And user mendapatkan email reminder sesuai dengan kondisi user</li> <li>• Then template email reminder yang diterima berisi informasi yang sesuai kondisi user</li> </ul> <p><b>Actual Result :</b></p> <ul style="list-style-type: none"> <li>• Template email reminder yang diterima user tidak ada informasinya atau kosong</li> </ul> <p><b>Expected Result :</b></p> <ul style="list-style-type: none"> <li>• Email reminder yang diterima oleh user, baik email reminder untuk H-5 ataupun H+1 berisi informasi yang sesuai dengan kondisi tersebut / tidak kosong</li> </ul>
--

Gambar 3.18. Tiket bugfix template email reminder kosong

Email *reminder* H-5 diberikan sebagai peringatan bahwa masa berlaku *membership user* akan selesai 5 hari lagi dan H+1 sebagai peringatan bahwa masa berlaku *membership user* sudah selesai dan *membership user* sedang memasuki masa tenggang. *Bug* ini ditemukan pada akun *user suber*.

#### 4. Story [Support Site] Event Tracker page\_viewed Magic Link

Pada tahap pertama pengujian ini, ditemukan *bug* dengan kondisi *user* tidak menerima *user\_type* yang sesuai dengan tipe email yang digunakan saat membeli *membership*. *user\_type* yang diterima oleh *user* yang menggunakan tipe email *anon/guest* saat membeli *membership* sudah benar namun, hal ini diakibatkan karena sistem data berisi *null*, bukan karena email *user* terdeteksi sebagai email *guest*. Lalu, untuk tipe email *regon* dan *suber* tidak memberikan hasil *user\_type* apapun. Tampilan tiket *bugfix* yang telah dibuat dapat dilihat pada Gambar 3.19.

<p><b>Description :</b></p> <ul style="list-style-type: none"><li>• <i>user_type</i> yang diterima tidak sesuai dengan tipe email user yang digunakan saat membeli <i>membership</i></li></ul> <p><b>Pre-Condition :</b></p> <ul style="list-style-type: none"><li>• user beli <i>membership</i> di halaman checkout menggunakan email <i>anon</i>, <i>regon</i>, dan <i>suber</i></li></ul> <p><b>Repro Steps :</b></p> <ul style="list-style-type: none"><li>• user <i>anon</i> membeli paket berlangganan di halaman checkout</li><li>• user menyelesaikan <i>payment</i></li><li>• user masuk ke halaman Konfirmasi Paket Langganan</li><li>• user mengecek <i>user_type</i> yang diterima sesuai dengan tipe email user yang digunakan saat checkout <i>membership</i></li></ul> <p><b>Actual Result :</b></p> <ul style="list-style-type: none"><li>• <i>user_type</i> untuk email yang belum terdaftar (email <i>Anon</i>) berhasil mendapat <i>user_type</i> G namun dikarenakan error (data: null), bukan karena email terdeteksi sebagai email yang belum terdaftar</li><li>• <i>user_type</i> untuk email sudah terdaftar (email <i>Regon</i> dan <i>Suber</i>) tidak muncul</li></ul> <p><b>Expected Result :</b></p> <ul style="list-style-type: none"><li>• <i>user_type</i> untuk email belum terdaftar = G</li><li>• <i>user_type</i> untuk email sudah terdaftar = S</li></ul>
---

Gambar 3.19. Tiket bugfix event tracker page\_viewed

Seharusnya *user* yang membeli *membership* dengan tipe email yang belum terdaftar (*guest*) akan mendapatkan *user\_type* = G, lalu *user* yang membeli *membership* dengan tipe email yang sudah terdaftar (*regon* dan *suber*) akan mendapatkan *user\_type* = S.

### C. Testing Success

Hasil pengujian yang sukses menunjukkan bahwa fungsi atau fitur yang diuji telah berjalan sesuai dengan spesifikasi yang ditetapkan. Bentuk tangkapan

layar (*screenshot*) dapat dijadikan sebagai bukti visual ketika hasil pengujian suatu *story* sudah dapat memberikan hasil yang sesuai dengan ketentuan dari PM (*Product Management*). Keberhasilan ini memastikan sistem dapat berfungsi dengan benar dalam kondisi yang telah dirancang. Berikut beberapa hasil pengujian *story* yang telah sesuai dengan *Acceptance Criteria*:

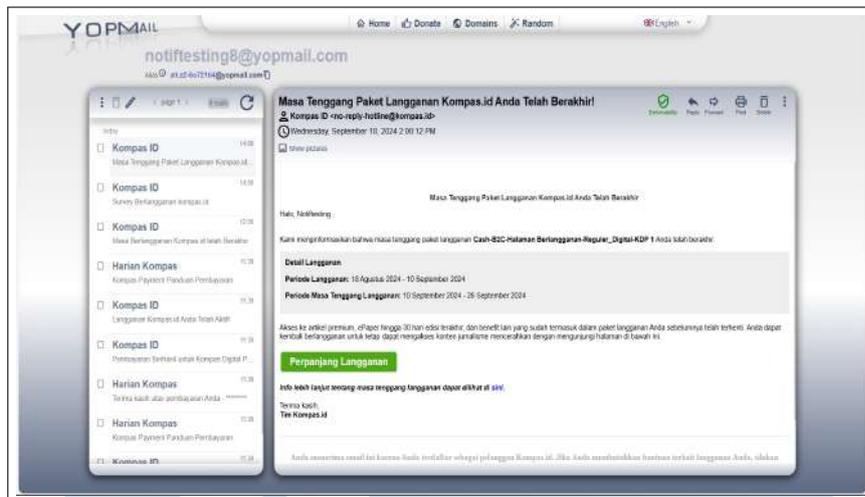
1. [Reminder Habis Langganan] Adjustment wording email reminder habis langganan

Saat menguji *story* ini, QE (*Quality Engineer*) langsung mendapatkan hasil *testing success* tanpa menemukan adanya *bug* karena hasilnya telah sesuai dengan yang telah ditentukan. *User* telah berhasil mendapatkan email *reminder* saat masa tenggang *membershipnya* telah berakhir (H+8). Proses pengujian untuk mendapatkan email *reminder* ini dibantu oleh *developer* bagian *backend* karena harus mengubah waktu *expired membership* nya di database, sedangkan anak magang tidak mendapatkan akses ke database. *Wording* atau *template* isi email *reminder* berdasarkan *expected result* dapat dilihat pada Gambar 3.20.



Gambar 3.20. Expected result template isi email reminder H+8

Email *reminder* yang didapat *user* telah sesuai dengan ketentuan, berikut bukti dari hasil *testing success* ini dapat dilihat pada Gambar 3.21.



Gambar 3.21. Hasil testing success email reminder H+8

- [BF][Kirim Opini] - Status user yang ditampilkan pada Header Kirim Opini sebagai Guest

Saat pengujian *story* ini dilakukan, QE langsung mendapatkan hasil *testing success*. Proses pengujian ini dilakukan di *cloud* milik Kompas.id. Hasil *testing success* yang didapatkan QE dapat dilihat pada Gambar 3.22.



Gambar 3.22. Hasil testing success button login pada halaman Kirim Opini

- Update Copy TnC on Register Page  
Selama proses pengujian *story* ini, tidak ada *bug* yang ditemukan. QE langsung mendapatkan hasil *testing success* saat menguji *wording* TnC yang ada di halaman registrasi akun. *Wording* TnC sebelum diubah dapat dilihat pada Gambar 3.23.



Gambar 3.23. wording TnC pada halaman register sebelum diperbarui

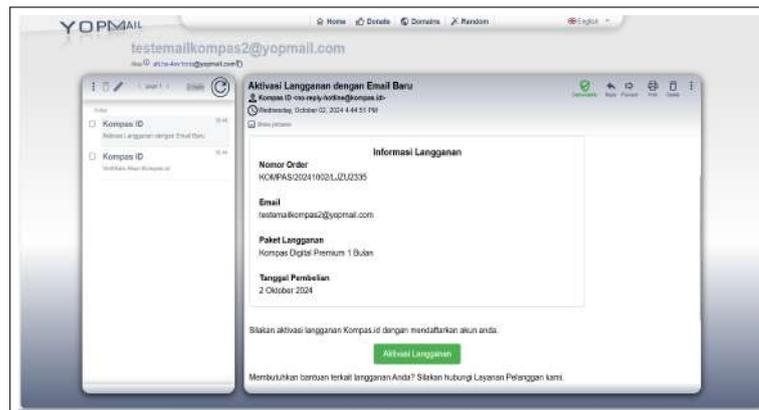
Namun setelah *wording* TnC diubah menyesuaikan dengan desain figma menjadi seperti pada Gambar 3.24.



Gambar 3.24. Hasil testing success wording TnC pada halaman register

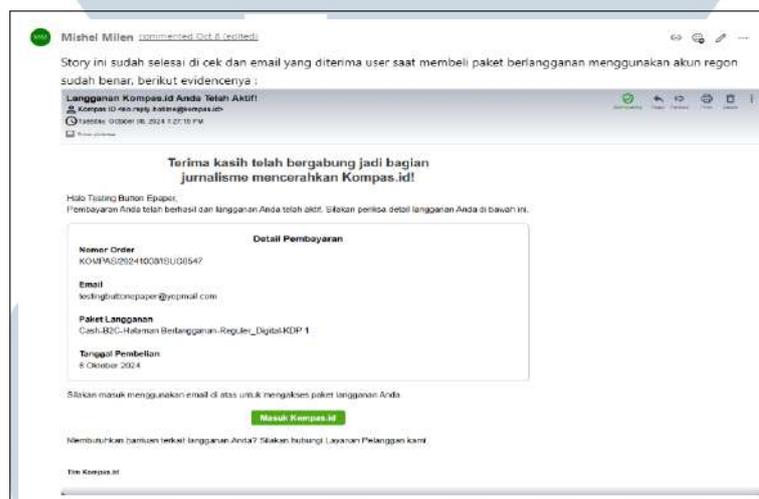
4. [QA] Magic Link Proses pengujian *story Magic Link* menemukan beberapa *bug* yang telah dibuatkan tiket *bugfix* nya. Berikut beberapa tiket *bugfix* yang telah dibuat beserta hasil *testing bug* nya:

- (a) [BF] - Fix flow user saat checkout dengan akun regon (tanpa login)  
Setelah pengujian tahap pertama ditemukan *bug*, pada pengujian tahap kedua (setelah tiket *bugfix* dibuat), proses pengujian mendapatkan hasil *testing success*. Hasil *testing* tahap satu dapat dilihat pada Gambar 3.25.



Gambar 3.25. Hasil testing tahap satu user mendapat email subscription active

Pada *testing* tahap dua, *user* telah berhasil mendapatkan email yang seharusnya diterima berdasarkan *expected result* yang telah ditentukan dan dapat dilihat pada Gambar 3.26.



Gambar 3.26. Hasil testing success tahap dua user mendapat email subscription active

(b) [BF] - Tanggal di Faktur Pembelian tidak real time

Pada pengujian tahap pertama, ditemukan *bug* dengan kondisi tanggal pada faktur pembelian tidak sama dengan tanggal saat paket langganan tersebut dibeli. Hasil *testing* tahap satu ini dapat dilihat pada Gambar 3.27.



Gambar 3.27. Hasil testing tahap satu tanggal faktur pembelian

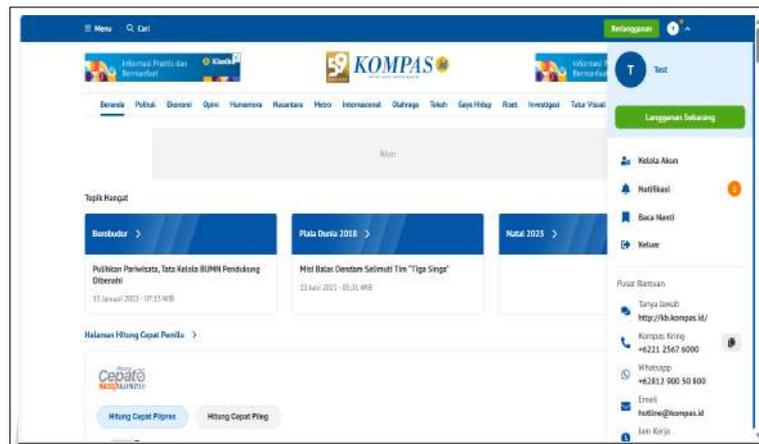
Setelah itu dilakukan perbaikan untuk menyesuaikan tanggal pembelian tersebut, sehingga pada pengujian tahap kedua didapatkan hasil *testing success*. Proses pengujian yang dilakukan saat tahap satu diulang kembali dan mendapatkan hasil seperti pada Gambar 3.28.



Gambar 3.28. Hasil testing success tahap dua tanggal faktur pembelian

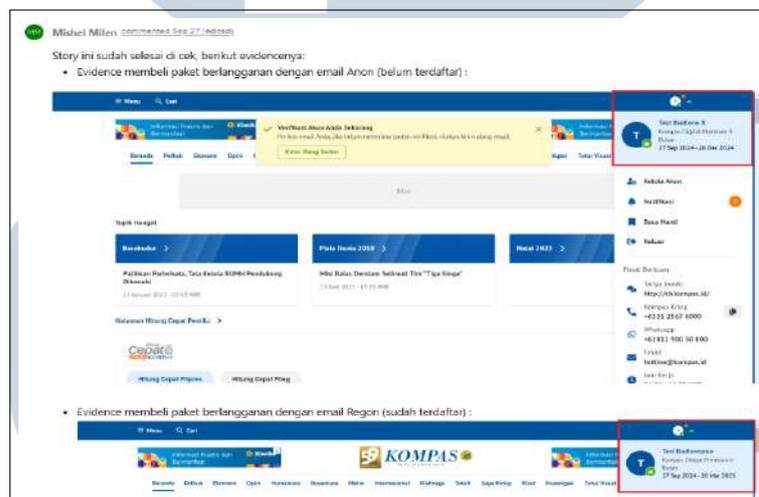
(c) [BF] - Membership tidak masuk saat user sudah membeli paket berlangganan

Saat pengujian tahap pertama, ditemukan *bug* dengan kondisi *user* tidak berhasil mendapatkan *membership* padahal sudah melakukan *payment* dengan berhasil. Hasil pengujian tahap satu dapat dilihat pada Gambar 3.29.



Gambar 3.29. Hasil testing tahap satu membership user

Pada pengujian tahap kedua telah dilakukan perbaikan oleh para *developer*, sehingga *user* sudah berhasil mendapatkan *membership* sesuai dengan yang telah dibeli sebelumnya. Hasil *testing success* ini dapat dilihat pada Gambar 3.30.



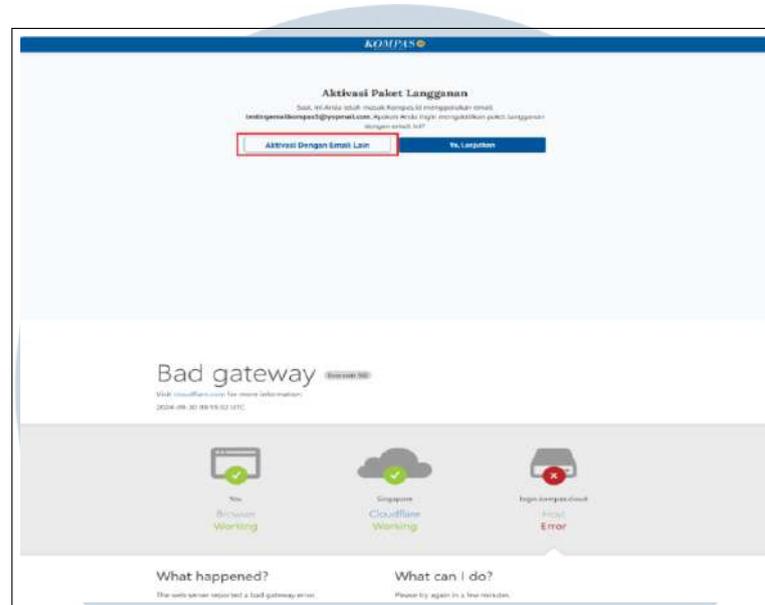
Gambar 3.30. Hasil testing success tahap dua membership user

Hasil pengujian juga dilakukan terhadap dua kondisi, yakni kondisi saat *user anon* membeli paket langganan menggunakan akun email *anon* dan akun email yang sudah terdaftar di *kompas.id*.

(d) [BF] - User tidak bisa aktivasi dengan email berbeda

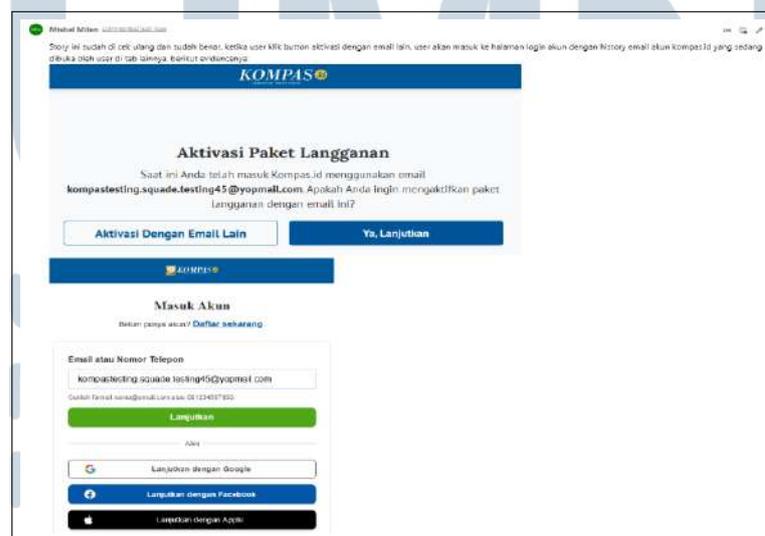
Pada pengujian tahap pertama, ditemukan *bug* pada *story* ini dengan kondisi ketika *user* ingin mengaktivasi *membership* yang telah dibelinya ke akun email yang berbeda, *user* akan menemukan halaman *bad*

gateway. Hasil pengujian tahap pertama ini dapat dilihat pada Gambar 3.31.



Gambar 3.31. Hasil testing tahap satu aktivasi dengan email berbeda

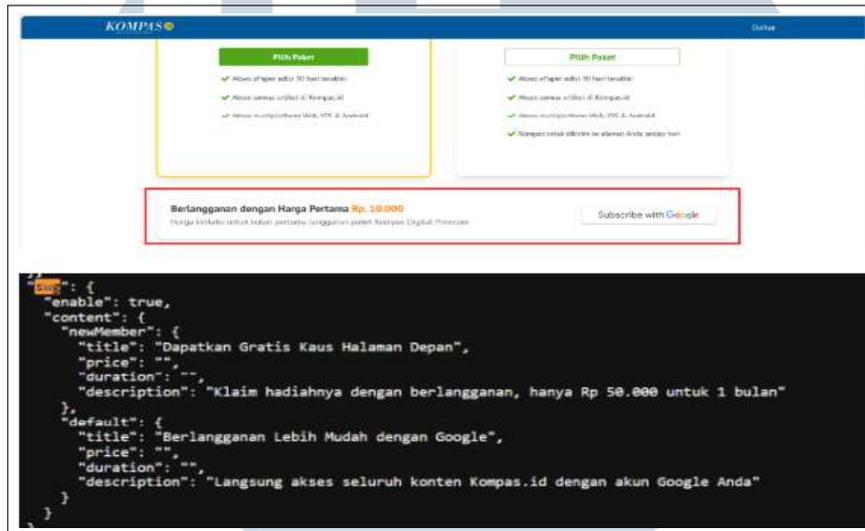
Setelah dibuatkan tiket *bug* nya, *problem* ini telah ditangani oleh *developer* dan setelah dilakukan pengujian ulang pada tahap kedua, hasilnya sudah sesuai dengan kriteria. Ketika *user* klik *button* aktivasi dengan email lain, *user* masuk ke halaman *login* akun seperti pada Gambar 3.32.



Gambar 3.32. Hasil testing success tahap dua aktivasi dengan email berbeda

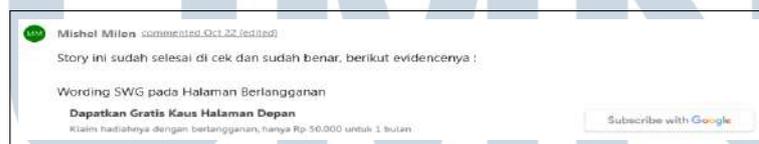
## 5. [Subscription] Dynamic Wording on SWG Button

Pada pengujian tahap satu, ditemukan *bug* pada *story* ini dengan kondisi *wording* SWG yang telah diperbarui di JSON, tidak tampil di bagian *frontend*, sehingga perlu dilakukan perbaikan dan pembuatan tiket *bugfix* untuk dijadikan sebagai dokumentasi. Hasil pengujian tahap satu dapat dilihat pada Gambar 3.33.



Gambar 3.33. Hasil testing tahap satu wording SWG

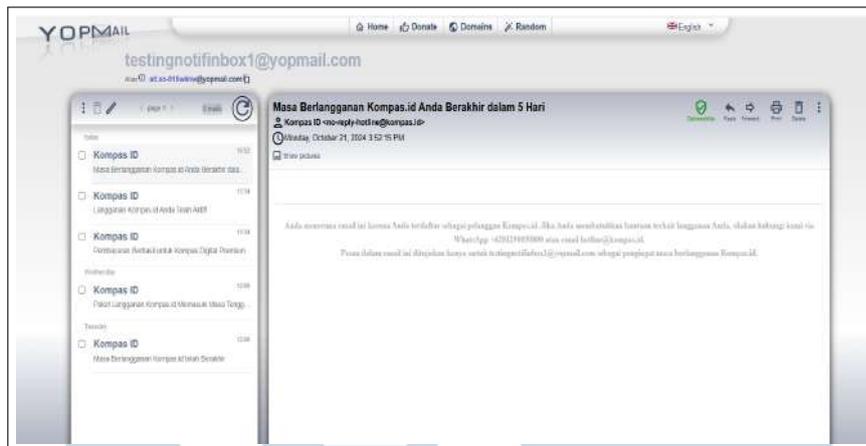
Setelah dilakukan pengujian kembali di tahap dua, hasil yang didapatkan adalah *wording* SWG pada *frontend* sudah sesuai dengan data yang telah dibuat di JSON sebelumnya. Hasilnya dapat dilihat pada Gambar 3.34.



Gambar 3.34. Hasil testing success tahap dua wording SWG

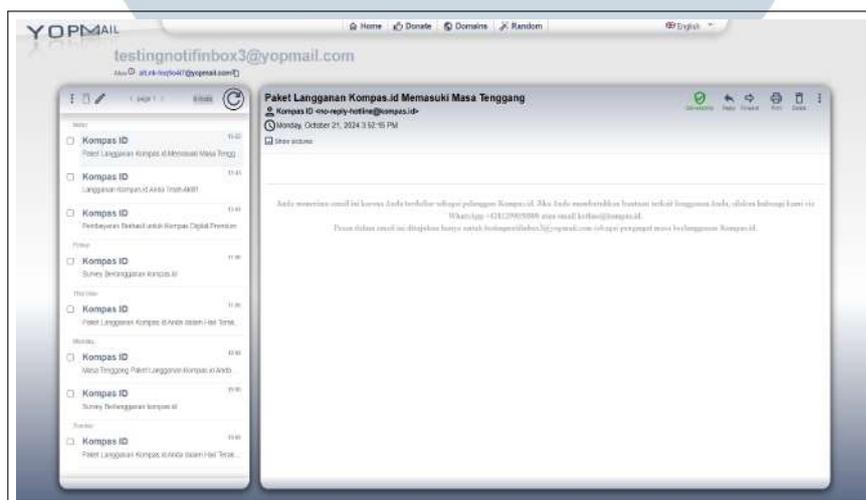
#### 6. [Subscription] Template email Subscription dikelola dari sre

Proses pengujian tahap satu *story* ini juga ditemukan *bug* dengan kondisi isi email *reminder* H-5 dan H+1 yang diterima *user* kosong atau tidak ada isinya sama sekali. Berikut *evidence bug* yang didapat saat pengujian tahap satu email *reminder* H-5 yang dapat dilihat pada Gambar 3.35.



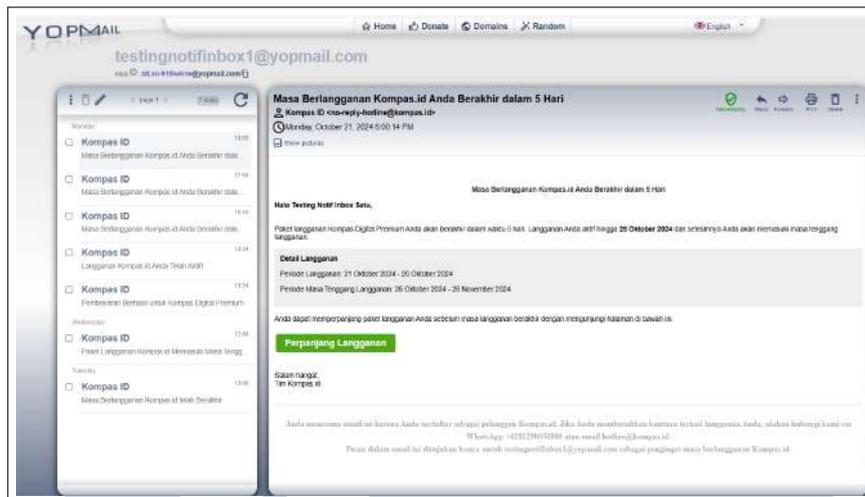
Gambar 3.35. Hasil testing tahap satu isi email reminder H-5

Berikut *evidence bug* yang juga diterima *user* saat melakukan pengujian tahap satu untuk email *reminder* H+1 dapat dilihat pada Gambar 3.36.



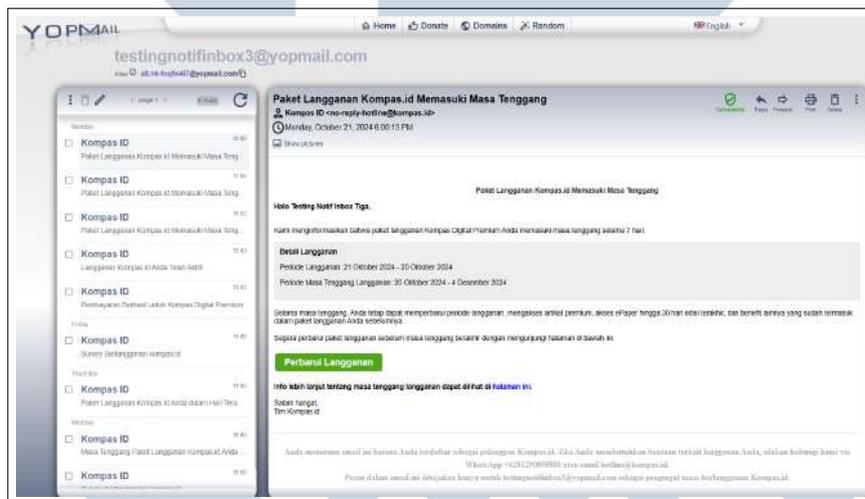
Gambar 3.36. Hasil testing tahap satu isi email reminder H+1

Setelah dilakukan perbaikan, dilakukan pengujian ulang pada tahap dua yang sudah berhasil menampilkan isi email *reminder* H-5 yang diterima oleh *user* seperti pada Gambar 3.37.



Gambar 3.37. Hasil testing success tahap dua isi email reminder H-5

Lalu untuk isi email *reminder* H+1 terbaru yang diterima oleh *user* dapat dilihat pada Gambar 3.38.



Gambar 3.38. Hasil testing success tahap dua isi email reminder H+1

## 7. [Promotion] Adjustment API Available Coupon today

Proses pengujian *story* ini langsung mendapatkan hasil *testing success*, sehingga pengujian hanya berhenti sampai *testing* tahap satu saja. API *productId* berhasil memberikan nilai ke kupon-kupon yang *available* untuk digunakan di hari tersebut dan memberikan nilai nol (0) untuk kupon yang tidak *available* untuk digunakan di hari tersebut. Hasil *testing* ini juga telah sesuai dengan ketentuan yang telah ditetapkan dan dapat dilihat pada Gambar 3.39.



Gambar 3.39. Hasil testing success adjustment API available coupon today

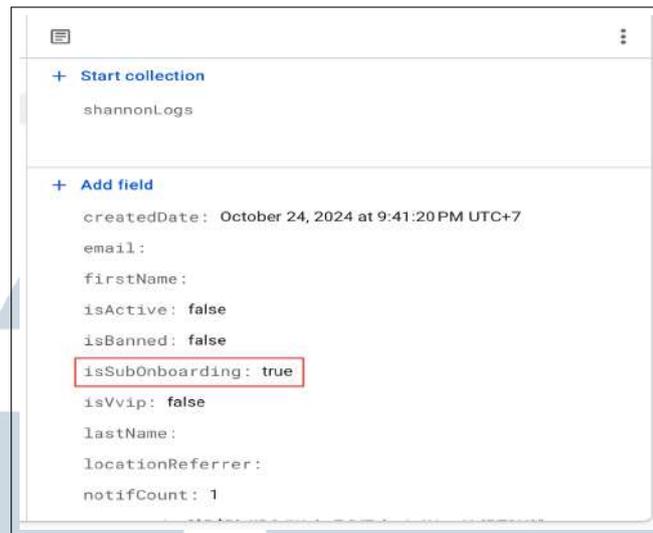
#### 8. [Subscription] Create API Store isSubOnboarding

Pengujian *story* ini hanya sampai *testing* tahap satu saja karena QE langsung mendapatkan hasil *testing success*. Proses pengujian API *isSubOnboarding* ini dibantu oleh *developer* bagian *backend* untuk melihat hasil pengujiannya lewat *firestore*, sedangkan QE mengirimkan API nya lewat *postman*. Hasil yang didapat sebelum *send* API dari *postman* dapat dilihat pada Gambar 3.40.



Gambar 3.40. Tampilan before API isSubOnboarding

terlihat bahwa sebelum API *update isOnBoarding* dikirim dari *postman*, API *isSubOnboarding* belum ada. Setelah API *update isSubOnboarding* dikirim dari *postman* dapat dilihat pada Gambar 3.41.



Gambar 3.41. Tampilan after API isSubOnboarding

Setelah API ini terkirim dari *postman*, *isSubOnboarding* sudah ada dan bernilai *true*.

#### 9. [Support Site] Notification Header

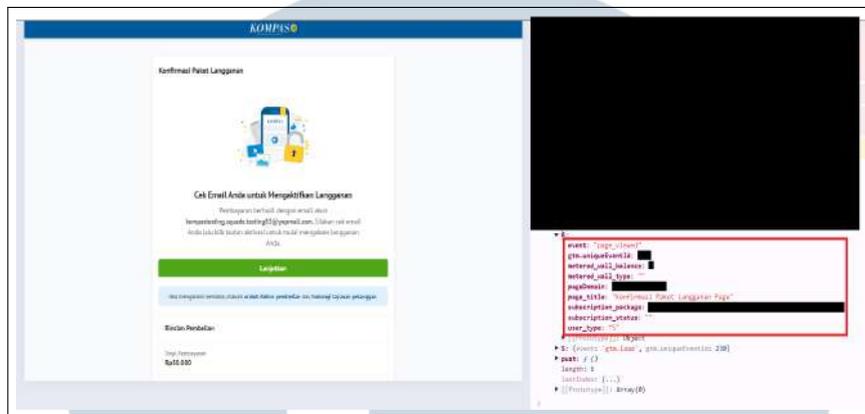
Selama proses pengujian *story* ini tidak ditemukan adanya *bug*. QE mendapatkan hasil *testing success* saat pertama kali melakukan *testing* sehingga proses *testing* ini hanya sampai tahap satu saja. Fitur *notification* telah berhasil dipindahkan dari *menu* ke bagian *header* *kompas.id* dan hanya tampil ketika *user* dalam keadaan *regon* atau *suber* saja. Berikut hasil *testing* yang dapat dilihat pada Gambar 3.42.



Gambar 3.42. Hasil testing success notification header

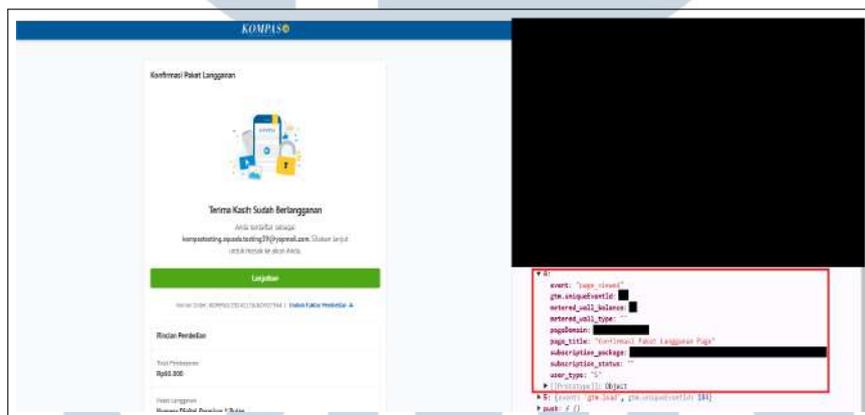


Pengujian tahap dua untuk tipe email *regon* juga sudah sesuai dan dapat dilihat pada Gambar 3.45.



Gambar 3.45. Hasil testing success tahap dua event tracker page\_viewed email Regon

Kemudian, pengujian tahap dua pada tipe email *suber* juga sudah sesuai dan dapat dilihat pada Gambar 3.46.



Gambar 3.46. Hasil testing success tahap dua event tracker page\_viewed email Suber

### 3.5 Kendala dan Solusi yang Ditemukan

Selama proses pelaksanaan magang di PT Kompas Media Nusantara terdapat beberapa kendala yang dialami, antara lain:

1. Pada setiap akhir masa *sprint*, beban kerja sering kali mengalami peningkatan yang signifikan, terutama karena adanya banyak *story* yang baru siap diuji menjelang *sprint review* dan *sprint planning*. Hal ini menyebabkan banyaknya *story* yang harus diuji dalam waktu singkat, sementara di awal

*sprint* justru sering kali tidak ada *story* yang siap untuk diuji, sehingga waktu kerja terasa lebih senggang di periode tersebut.

2. Beberapa *story* tidak terdapat informasi apapun, baik langkah pengujian, deskripsi tugas, sampai *acceptance criteria*. Hal ini dapat menyulitkan tim *developer* atau QE untuk menentukan cara pengembangannya atau metode pengujiannya. Selain itu juga dapat memperlambat proses pengujian *story*.

Dari semua kendala yang dirasakan saat proses kerja magang, terdapat beberapa solusi yang disarankan oleh mentor untuk mengatasi kendala-kendala tersebut, antara lain:

1. Solusi yang dapat dilakukan untuk mengatasi ketidakseimbangan beban kerja adalah dengan mencoba untuk mengimplementasikan *automation testing* pada fitur atau *story* yang memiliki alur pengujian berulang ataupun *story* yang sering diuji. Hal ini dapat membantu mempercepat proses pengujian dan membuat QE bisa lebih berfokus pada pengujian fitur yang kompleks.
2. Setiap anggota yang membuat *story user*, *task*, maupun tiket *bugfix* harus menyertakan informasi seperti langkah pengujian, deskripsi tentang *story* tersebut, sampai hasil yang diharapkan akan diterima oleh *user*. Hal ini dapat membantu tim pengembang untuk memahami maksud dan cara pengembangan *story* tersebut dengan lebih baik, juga membantu QE untuk dapat melakukan pengujian dengan lebih efisien, dan memastikan semua anggota tim memiliki pemahaman yang sama sehingga dapat mengurangi terjadinya kesalahan akibat pemahaman yang berbeda-beda.

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA