

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pelaksanaan kerja magang dimulai dengan melamar posisi untuk magang di PT Kompas Gramedia. Setelah tahap wawancara, surat penawaran kerja kemudian diberikan untuk ditandatangani. Kemudian penandatanganan kontrak kerja kemudian dilakukan sebagai Software Engineer di Customer Facing Platform (CFP) yang berfokus pada posisi *Quality Assurance/Engineer* dalam Corporate IT & IS (CITIS) PT Kompas Gramedia. Selama pelaksanaan kerja magang, penulis dibimbing oleh Bapak Kevin Alexander Surjadi selaku *Mobile Engineer* dari CFP selama 3 bulan pertama. Kemudian pembimbing penulis berubah menjadi Bapak Muhammad Iqbal selaku *Mobile Engineer Team Lead* selama 3 bulan terakhir pelaksanaan kerja magang. Pembimbing yang bersangkutan berperan dalam membimbing, mengawasi, memberikan arahan, materi pelatihan, informasi, dan masukan selama proses kerja magang berlangsung. Setiap hari Jum'at, dilakukan pertemuan mingguan bersama pembimbing lapangan untuk membahas tugas, kendala, atau saran yang dimiliki selama satu minggu tersebut.

Dalam CFP CITIS, Bapak Sigit Nurseto selaku *Software Engineering Manager* menerapkan *agile software development* dan *scrum process* dalam mengkoordinasi tim selama pengembangan dalam tim. Selama proses pengembangan, dilakukan *Daily Stand-Up Meeting* setiap hari untuk saling melaporkan proses, kendala, atau rencana yang akan dilakukan anggota tim. Setiap proses *sprint* berlangsung selama 2 minggu, dimana setiap akhir *sprint*, pada hari Jum'at, akan dilakukan review singkat melalui *OKR Review*. Setelah itu *sprint* berikutnya akan dimulai melalui proses *Sprint Review and Planning* pada hari Senin setelahnya.

Koordinasi antar anggota tim CFP CITIS dilakukan dengan berbagai *platform* seperti Slack, Figma, Microsoft Azure, Microsoft Teams, Git/GitLab, dan Dropbox Paper. Slack merupakan *platform* utama yang digunakan untuk melakukan komunikasi antar anggota. Figma digunakan sebagai *platform* untuk mendesain dan menampilkan hasil desain halaman *website* oleh tim desainer. Microsoft Azure dan Git/GitLab digunakan sebagai *platform* untuk melakukan *tracking progress task* dan *sub-task* yang akan, sedang, dan telah dilakukan selama *sprint* serta sebagai tempat untuk mengorganisir *project files* dan *repository* yang ada. Sedangkan Drop-

box Paper digunakan untuk menulis, menyimpan, dan mengorganisir dokumentasi-dokumentasi yang dilakukan oleh anggota tim.

3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama pelaksanaan kerja magang secara garis besar adalah untuk membantu dalam bagian *testing user requirement* saat pengembangan fitur baru di *website* CMS MyValue. Cakupan kerja secara umum yang dilakukan selama ini adalah sebagai berikut:

- Melakukan *Manual Testing* terhadap fitur-fitur yang ada di *website* CMS MyValue dan *mobile apps* MyValue.
- Merancang *test case scenario* berdasarkan *user requirements* yang diberikan.
- Membuat dan mengembangkan *automated test script* berdasarkan *test case scenario* yang sudah dibuat ataupun dengan *test case scenario* baru.
- Melakukan dokumentasi atas *test script* yang telah dibuat dan melaporkan *bug*, *error*, dan masalah lainnya yang ditemukan selama melakukan *testing* untuk membantu dalam menemukan dan mencari solusi dari masalah yang ditemukan tersebut.

3.3 Uraian Pelaksanaan Magang

Terhitung tanggal 1 Juli, pelaksanaan kerja magang dimulai, namun pada minggu tersebut hanya ada dua hari kerja yang dilakukan. Maka dari itu, uraian mengenai pelaksanaan kerja magang akan dihitung mulai dari hari Senin, 5 Juli, sebagai awal dari minggu pertama dan apa yang dilakukan pada dua hari pertama akan dimasukkan ke uraian minggu tersebut. Tabel 3.1 merupakan tabel deskripsi singkat pada tugas yang dilakukan selama pelaksanaan kerja magang setiap minggunya beserta uraian dari deskripsi singkat tersebut. Dikarenakan batas waktu selesai magang melebihi jadwal *deadline* pengumpulan laporan magang, maka uraian pelaksanaan tugas yang dicantumkan adalah uraian tugas dengan batas waktu tidak dilanjutkan setelah minggu kerja ke-24, dan tetap sesuai dengan prasyarat kelulusan pelaksanaan kerja magang.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	- <i>On-boarding</i> PT Kompas Gramedia - Pendalaman materi mengenai Vue.js
2	- Pendalaman materi mengenai Vue Js - <i>Manual Testing</i> CMS MyValue
3	- <i>Manual Testing</i> CMS MyValue - Pendalaman materi WebdriverIO
4	- <i>Manual Testing</i> CMS MyValue - Pendalaman materi Nightwatch.js
5	Pendalaman materi Cucumber BDD
6	- Pendalaman materi Cucumber BDD - Mulai merangkai <i>test scenario</i> untuk <i>automated testing</i>
7	- Mempelajari contoh <i>scenario</i> yang sudah ada - Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur <i>Authentication</i> untuk <i>website</i> MyValue
8	Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur <i>Authentication website</i> CMS MyValue
9	Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur <i>Inbox Notification</i> dan <i>Home Banner website</i> CMS MyValue
10	Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur-fitur pada <i>Member Management website</i> CMS MyValue
11	- Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur-fitur pada <i>Member Management website</i> CMS MyValue - Menulis Dokumentasi <i>scenario</i>
12	- <i>Manual Testing</i> untuk fitur <i>Voucher and Campaign Flash Sale</i> di MyValue Apps - Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur-fitur pada <i>Voucher and Transaction website</i> CMS MyValue
13	- Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur-fitur pada <i>Voucher and Transaction Management website</i> CMS MyValue - Menulis Dokumentasi <i>scenario</i>

Minggu Ke -	Pekerjaan yang dilakukan
14	- Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur-fitur pada <i>Content Management website CMS MyValue</i> - Melakukan <i>Manual Testing</i> untuk fitur <i>Flash Sale</i> yang telah diperbaiki
15	- Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur-fitur pada <i>Content Management website CMS MyValue</i> - Menulis <i>scenario</i> - Dokumentasi <i>scenario</i>
16	- Memulai proses <i>refactoring automated testing system</i> yang telah dibuat - Melakukan pendalaman mengenai proses <i>refactoring scenario</i> - <i>Manual Testing</i> fitur FOD CMS MyValue
17	Melanjutkan proses <i>refactoring automation testing system</i> yang telah dibuat
18	Melanjutkan proses <i>refactoring automation testing system</i> yang telah dibuat
19	Melanjutkan proses <i>refactoring automation testing system</i> yang telah dibuat
20	Melanjutkan proses <i>refactoring automation testing system</i> yang telah dibuat
21	Melanjutkan proses <i>refactoring automation testing system</i> yang telah dibuat
22	Melanjutkan proses <i>refactoring automation testing system</i> yang telah dibuat
23	Merangkai <i>test scenario</i> untuk <i>automated testing</i> fitur-fitur pada <i>Messaging Management dan Point Management website CMS My-Value</i>
24	Melakukan <i>Quality Check</i> dan melengkapi <i>scenario</i> yang kurang pada sistem yang telah dirancang

Pada minggu pertama proses pelaksanaan magang, penulis beserta dengan rekan *internship* lainnya melakukan *on-boarding* dengan tujuan untuk mengetahui struktur organisasi dari PT Kompas Gramedia , tim CITIS dan CFP dalam PT Kompas Gramedia . Selain itu, diadakan juga pertemuan pertama dengan anggota CFP dan juga bersama dengan rekan kerja magang dan supervisi masing-masing. Materi pembelajaran kemudian diberikan oleh supervisi lapangan saat diadakan pertemuan

serta untuk memberikan akses ke akun UdeMy untuk mempelajari Vue.js [4][5]. Selain itu diadakan juga *Objective & Key Results (OKR) Review* yang dipimpin oleh *Team Lead* untuk membahas objektif yang ingin dicapai oleh tim dan juga untuk membahas mengenai *progress* dan masalah yang ditemui selama *sprint* berlangsung.

Minggu kedua dimulai berpartisipasi dalam *Sprint Review and Planning* untuk *sprint* yang akan dijalankan selama dua minggu kedepan. Selain itu dilakukan juga pendalaman materi Vue.js melalui UdeMy, dokumentasi, serta sumber-sumber lainnya. Menjelang minggu berlangsung, penulis diberikan tugas untuk melakukan *Manual Testing* untuk beberapa fitur pembuatan konten pada *website* CMS MyValue yang dilakukan bersama dengan rekan kerja magang, seperti *Event Gathering*, *One Time Purchase Voucher (OTP)*, *Inbox Notification*, dan *Home Banner*. Hasil dari *Manual Testing* didokumentasikan dan dibahas bersama dengan tim *developer* untuk mengetahui hasil dan apa saja yang harus diperbaiki.

Pada minggu ketiga, dilakukan kembali *Manual Testing* untuk fitur-fitur sebelumnya untuk mengecek hasil perbaikan yang dilakukan. *Manual Testing* juga dilakukan untuk fitur lain seperti pembuatan jenis subyek notifikasi baru dan *FCM Devices*. Dokumentasi hasil testing dilakukan dan dilaporkan sesuai dengan prosedur yang dilakukan minggu sebelumnya. Menjelang akhir minggu dan akhir dari *sprint*, *OKR Review* kembali diadakan. Pendalaman materi juga masih befokus pada WebdriverIO. Divisi CFP juga mengadakan Townhall Meeting pada minggu ini, dimana keseluruhan anggota dari CFP berkumpul untuk saling berkenalan dan bertukar kabar satu sama lain.

Minggu keempat dimulai dengan *Sprint Review and Planning* oleh tim. Selain itu, diberikan juga materi pendalaman baru, Nightwatch.js oleh supervisi lapangan untuk dipelajari. Untuk bagian pembelajaran Nightwatch.js lebih lama dari materi yang lain dikarenakan dalam perancangan *automated testing system*, Nightwatch.js akan banyak digunakan [6][7].

Pada minggu kelima, pendalaman materi dilanjutkan ke Cucumber BDD, *open-source tool* yang akan digunakan bersamaan dengan Nightwatch.js untuk merancang *automated testing system* yang diinginkan [8]. Selain itu, mulai dilakukan instalasi dan persiapan *local repository* untuk memulai perancangan sistem. Dilakukan juga *sync repository* untuk contoh hasil system yang dibuat sebelumnya sebagai referensi penyusunan file dan pengerjaannya. *OKR Review* dilakukan menjelang akhir *sprint*.

Minggu keenam dimulai dengan *Sprint Review and Planning* dan tugas untuk mempelajari contoh *directory testing system* yang telah ada dan mencoba untuk membuat *automated test scenario* untuk *website MyValue* menggunakan Cucumber BDD dan Nightwatch.js. Contoh *scenario* yang dibuat adalah *scenario* untuk membuka halaman *website* dan berinteraksi dengan elemen untuk membuka *link* Google Playstore untuk MyValue App.

Minggu ketujuh diisi dengan membuat *testing scenario* untuk melakukan *User Login* dengan ketentuan hasil *login* sukses dan *login* gagal (dikarenakan salah input *password*). Hasil *test scenario* dijalankan dan dicommit ke *repository* yang ada (GitLab) untuk dilihat dan dilakukan *merge directory* yang ada. *Sprint* kali ini pun juga ditutup dengan *OKR Review*.

Minggu kedelapan dimulai dengan *Sprint Review and Planning*, dan memulai proses pembuaran *automated testing system* untuk *website CMS MyValue*. Proses ini dimulai dengan mempersiapkan *local directory* untuk tempat *project repository* yang ada dan dimulai dengan menyiapkan serangkaian *script* interaksi dan *step definition* sebelum *scenario* dibuat. *Feature* yang dibuat pertama kali adalah *feature* untuk *Authentication (Login-Logout)*.

Pada minggu kesembilan, pembuatan *feature* dan *scenario* dilanjutkan ke bagian fitur pembuatan konten *Inbox Notification* dan *Home Banner*. *Scenario* yang dirancang merupakan *scenario* yang dilakukan pada *manual test* beberapa minggu sebelumnya. Selain itu, dilakukan juga penambahan *script* interaksi dan *step definition* sesuai dengan apa yang dibutuhkan untuk membuat *scenario* yang dibutuhkan kedepannya. *OKR Review* kembali dilakukan untuk membahas *sprint* kali ini.

Pada minggu kesepuluh, dilakukan *Sprint Review and Planning* serta melakukan *repository commit* pada GitLab untuk diperiksa dan diakses supervisi dan anggota tim lainnya. Selain itu, dimulai juga proses merancang *scenario* untuk fitur-fitur yang ada pada tab bagian *Member Management* pada *website CMS MyValue*.

Minggu kesebelas berfokus kepada tugas untuk menyelesaikan perancangan *feature* dan *scenario* dalam tab bagian *Member Management*. Setelah *scenario* untuk fitur pada tab ini dibuat, *commit* dan *merge request* kembali dilakukan untuk diperiksa dan disapprove oleh supervisi lapangan. Dokumentasi juga ditambahkan pada *scenario* baru yang dibuat. Menjelang akhir dari *sprint*, *OKR Review* dilakukan kembali.

Sprint Review and Planning dilakukan untuk memulai minggu kedua belas. Proses membuat *automated testing system* berlanjut pada tab bagian *Voucher and Transaction Management*. Selain itu, dilakukan juga *Manual Testing* untuk fitur *flash Sale Voucher*, *Flash Sale Campaign* dan *Manual Testing Flash Sale Websocket RealTime* beberapa kali dalam sela satu minggu tersebut.

Minggu ketiga belas masih diisi dengan merancang *scenario* untuk tab bagian *Voucher and Transaction Management*. Setelah itu, dilakukan *commit* dan *merge request* untuk *scenario* yang telah dibuat. *OKR Review* juga dilakukan menjelang akhir dari *sprint*.

Pada minggu keempat belas, dilakukan *Sprint Review and Planning*. Dilakukan kembali *Manual Testing* untuk fitur pembuatan *Flash Sale Voucher* dan *Flash Sale Campaign* melalui *website* CMS MyValue selama beberapa hari. Hasil dari *Manual Testing* didokumentasikan untuk dibahas dan dilaporkan. Proses perancangan *automated testing system* juga dilanjutkan ke tab bagian *Merchant Management* dan *Content Management*.

Minggu kelima belas diisi dengan tugas melanjutkan perancangan *scenario* untuk bagian tab *Content Management*. Selain itu, dilakukan juga *retesting* untuk *Flash Sale* yang dilakukan minggu sebelumnya dan *commit* untuk *directory* atas *scenario* yang telah dibuat. Di minggu ini, dilakukan juga pembuatan dokumentasi untuk *scenario* pada *Merchant Management* dan *Content Management* serta *OKR Review* menjelang akhir *sprint*.

Minggu keenam belas dibuka dengan *Sprint Review and Planning*. Dari minggu ini, penulis ditugaskan untuk melakukan *refactoring* atas sistem yang telah dirancang hingga saat ini. *Refactoring* berfokus pada mengubah elemen *selector* pada *website* yang dipilih untuk dilakukan interaksi selama proses *testing* menjadi menggunakan *XPath* untuk membuat elemen lebih spesifik saat diinteraksi. Selain itu, *step definition* yang dibuat juga dimodifikasi sesuai dengan perubahan yang terjadi selama proses *refactoring*. Minggu ini dimulai dengan melakukan *research* dan pembelajaran mengenai penggunaan elemen *XPath* dan percobaan untuk menyesuaikan *script* yang ada dengan perubahan yang diinginkan.

Minggu ketujuh belas hingga minggu kedua puluh dua berfokus pada melakukan *refactoring* dari *automated testing system*. *Manual Testing* yang dilakukan selama minggu-minggu ini adalah testing untuk fitur FOD pada CMS MyValue dan MyValue App. Selain itu, proses *OKR Review* dan *Sprint Planning and Review* tetap berlangsung sesuai periode yang ada (setiap dua minggu), dan memasuki minggu keduapuluh dua menandakan proses *refactoring* berjalan sepanjang

tiga *sprint cycle*.

Minggu kedua puluh tiga dimulai dengan *Sprint Review and Planning*. Setelah proses *refactoring* dilakukan, perancangna sistem dilanjutkan dengan merangkai *scenario* untuk bagian fitur tab *Messaging Management* dan *Point Management*. Hasil *scenario* yang dibuat kemudian dilakukan *commit* dan *directory update*.

Minggu kedua puluh empat diisi dengan melakukan pengecekan secara keseluruhan sistem yang dirancang. Dilakukan penambahan *scenario* pada *Feature* yang memiliki *scenario* yang kurang atau tidak dibuat sebelumnya. Selain itu dilakukan juga *OKR Review* menjelang akhir dari *sprint*.

3.3.1 Analisis Kebutuhan

Automated Testing System yang dijadikan tujuan pelaksanaan magang dibuat dengan tujuan untuk membantu dalam proses *Manual Testing* yang dilakukan secara berulang dan memakan waktu. Sistem ini dibuat untuk sebagai sistem yang akan melakukan tugas *testing* untuk *scenario* umum pada tab-tab dalam *website* CMS dari MyValue. Masing-masing tab pada *website* CMS MyValue tersebut akan dipisah menjadi satu *Feature file* (*.feature*) dalam folder *testing*, dimana tiap *Feature* akan memiliki beberapa *Scenario* yang berisikan rangkaian langkah-langkah yang disebut sebagai *Step Definitions*. *Step definitions* dibuat dari kumpulan baris kode menggunakan bahasa *JavaScript*. Gambar 3.1 berikut merupakan skema singkat mengenai proses pembuatan suatu *Feature*, *Scenario*, dan *Step Definition*.



Gambar 3.1. Skema alur perancangan sebuah sebuah *Feature*

3.3.2 Alur Kerja dan *Framework* yang Digunakan

Pengenalan alur kerja dilakukan pada minggu pertama pelaksanaan kerja magang, dimana dijelaskan proses dan alur dari *scrum process* yang dilakukan beserta apa saja aplikasi dan platform yang akan digunakan. Supervisi lapangan memberikan akses ke akun Udemmy sebagai sumber materi untuk pembelajaran selama proses kerja magang, dan memberikan tutorial singkat mengenai aplikasi dan *platform* yang dipakai. Selain itu, dilakukan

juga penginstallan program yang dibutuhkan.

Tim CFP menerapkan *agile software development* dan *scrum process* untuk melakukan koordinasi dalam alur kerja mereka. Dimana *sprint* dilakukan setiap dua minggu, dimulai dengan *Sprint Review and Planning*, dan ditutup dengan *OKR Review* dalam satu *sprint cycle*. Selain itu, setiap akhir minggu dilakukan *Weekly Review* bersama supervisi lapangan untuk membahas mengenai *progress*, kendala atau apapun yang ditemukan selama satu minggu tersebut.

Dalam menjalankan tugas *Manual Testing*, dilakukan *testing* terhadap fitur yang diinginkan pada beberapa *platform* MyValue yang ada. Dimulai dari *website* MyValue, *website* CMS MyValue, hingga MyValue Apps. Hasil dari *Manual Testing* kemudian akan didokumentasikan sebagai bentuk laporan mengenai hasil *output* dari *testing*, *bug* yang kemungkinan dijumpai, hingga langkah-langkah yang dijalankan untuk mencapai *bug* yang dijumpai tersebut. Hasil dokumentasi tersebut akan dilaporkan dan dibahas untuk mencari solusi, dan setelah diperbaiki akan dilakukan *retesting* untuk memastikan kelancaran alur fitur tersebut.

Automated Testing System yang dirancang dalam pelaksanaan kerja magang ini difokuskan kepada bagian *website* CMS MyValue. Untuk pelaksanaan tugas magang ini, *website* yang menjadi tempat *testing* bukan merupakan *production website*, melainkan merupakan *staging website*. *Staging website* merupakan *website* yang dibuat sebagai tempat untuk *development* sebelum menambahkan *update* ataupun fitur baru kepada *production website*. Langkah awal yang dilakukan dalam merangkai *Feature* dan kumpulan *Scenario* di dalamnya adalah dengan menyusun alur dari untuk *scenario* yang akan dibuat. Alur tersebut kemudian akan diterjemahkan menjadi beberapa langkah yang sesuai dengan *Step Definition* yang telah dirancang.

Penyusunan *test case* biasanya memiliki dua jenis berdasarkan *output*, *positive case* dan *negative case*. Namun perancangan *Automated Testing System* ini difokuskan pada *positive case*, dimana dilakukan pengecekan kesesuaian hasil *output* dari *input* yang diberikan saat *testing* dijalankan. *Framework* yang digunakan dalam perancangan *Automated Testing System* ini adalah *Cucumber BDD* dengan bahasa pemrograman berbasis *Javascript*, *Nightwatch.js*.

Cucumber BDD merupakan *framework* yang mendukung proses *Behaviour-Driven Development* (BDD) dalam *development*. BDD sendiri merupakan langkah yang digunakan untuk mencoba menutupi jarak antara pihak ‘bisnis’ dan pihak ‘tech’ dengan merancang sebuah *system* yang dapat dipakai oleh kedua pihak untuk mengilustrasikan alur dari *website* atau *web application* secara keseluruhan ataupun fitur tertentu saja. Sedangkan Nightwatch.js adalah *end-to-end testing framework* yang terintegrasi untuk membantu melakukan *testing* pada *web application* dan *website*. Nightwatch.js menggunakan *World Wide Web Consortium* (W3C) WebDriver API untuk menjalankan *browser*, perintah, dan pengecekan pada *Document Object Model elements* (DOM).

Penggunaan Cucumber BDD diharapkan dapat membantu dalam proses automasi, dan juga mempermudah untuk anggota yang masih awam dengan bahasa pemrograman untuk dapat membentuk *scenario* sendiri dikarenakan penggunaan bahasa Gherkin dalam merancang feature yang ada. Gherkin sendiri dikenal sebagai *business readable language*, karena *Feature* yang dibuat akan menjelaskan *business behaviour* tanpa perlu merincikan tahap implementasi yang lebih mengarah pada pihak ‘tech’. Bahasa ini menggunakan aturan penulisan biasa (*natural language*) untuk menjelaskan langkah-langkah yang dilakukan secara spesifik, yang kemudian akan diterjemahkan untuk memilih bagian fungsi apa yang dipilih untuk dijalankan dalam *scenario* tersebut pada bagian *step definitions*. Setelah menentukan *step definition* yang dimaksud, perintah yang ada akan direfer kembali ke *javascript* yang sudah dibuat dan dijalankan pada saat *testing*.

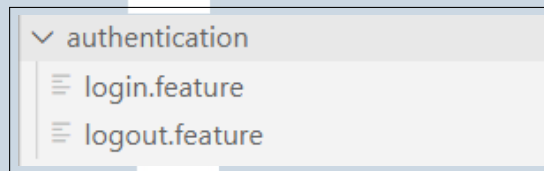
3.4 Perancangan dan Implementasi Sistem

3.4.1 Struktur File

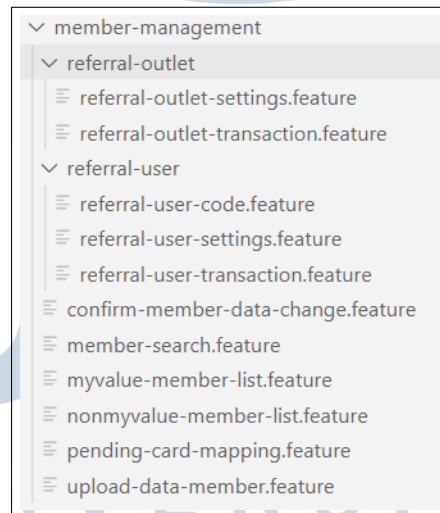
Selama perancangan *Automated Testing System* berlangsung, telah dibuat sejumlah 66 *Features*, dengan minimal terdapat satu *test scenario* di dalam *Feature* tersebut. *Feature* ini sendiri dibuat sesuai dengan jumlah fitur dan tab yang ada pada *website* CMY MyValue, dimana tiap tab akan dipisah menjadi *folder* dan *sub-folder*, dan setiap daftar fitur yang ada akan dibuat *Feature file* mereka masing-masing. Berikut merupakan *screenshot* daftar tab pada *website* CMS MyValue, beserta fitur di dalamnya. Berdasarkan

dari keterangan tersebut, struktur file dari *Feature* yang dibuat pada *Automated Testing System* yang dirancang beserta dengan *Feature* untuk proses *Authentication* adalah sebagai berikut.

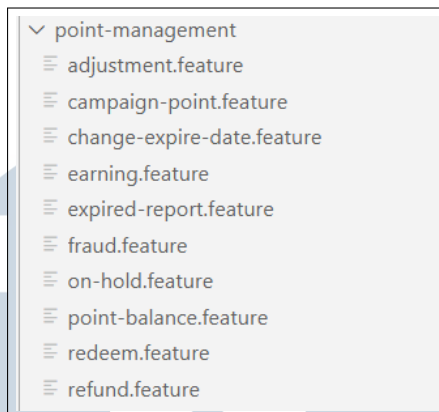
Pada Gambar 3.2, folder *Authentication* memuat *file feature* untuk login dan logout user. Kedua scenario ini dipisah untuk kepentingan organisasi data dan referensi untuk dapat digunakan ke *scenario* pada *feature* lainnya. Folder tanpa subfolder seperti pada Gambar 3.4 dan 3.8 memuat sesuai dengan list fitur yang ada pada tab masing-masing. Sedangkan untuk folder dengan subfolder seperti Gambar 3.3, Gambar 3.5, Gambar 3.6, dan Gambar 3.7 menampung fitur untuk list tambahan sesuai dengan fitur dalam tab tersebut.



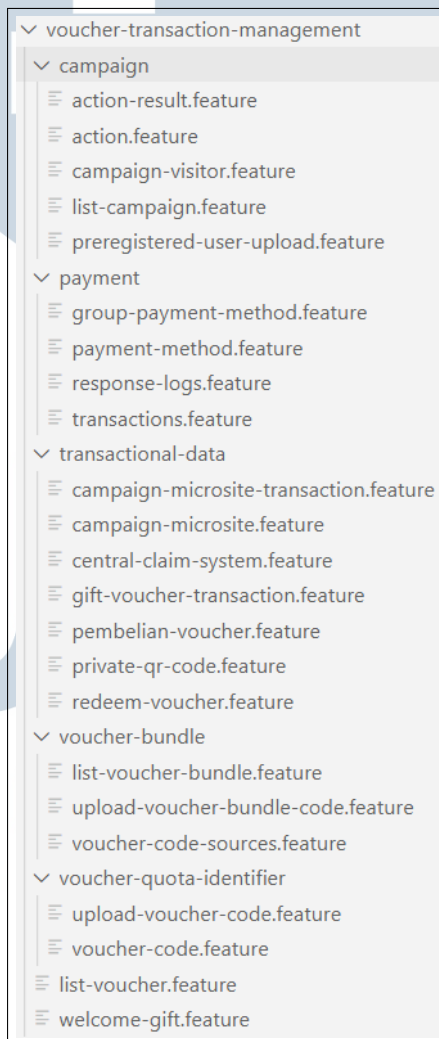
Gambar 3.2. Struktur *file* pada fitur *Authentication*



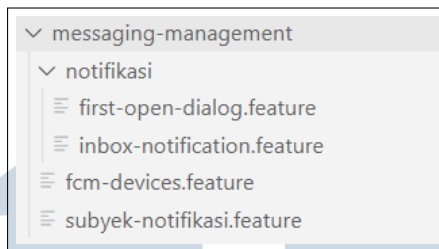
Gambar 3.3. Struktur *file* pada fitur *Member Management*



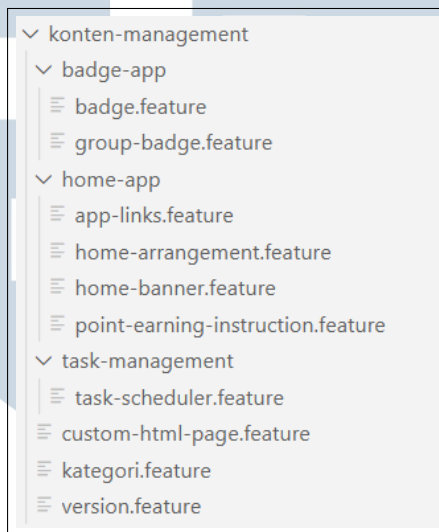
Gambar 3.4. Struktur file pada fitur *Point Management*



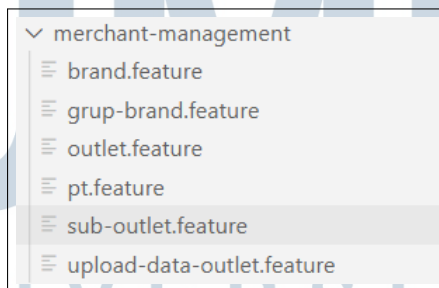
Gambar 3.5. Struktur file pada fitur *Voucher and Transaction Management*



Gambar 3.6. Struktur *file* pada fitur *Messaging Management*



Gambar 3.7. Struktur *file* pada fitur *Content Management*



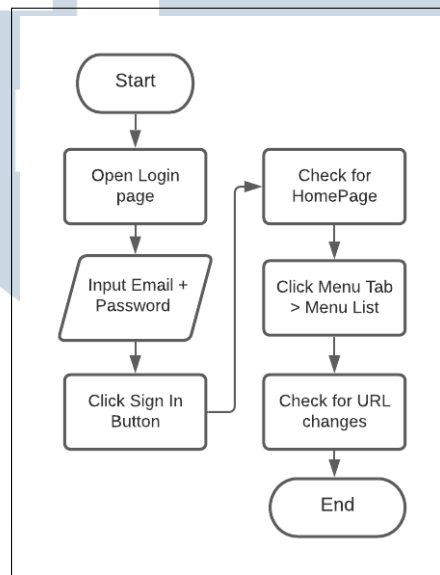
Gambar 3.8. Struktur *file* pada fitur *Merchant Management*

3.4.2 Flowchart

Berikut merupakan daftar gambar *flowchart* dari rangkaian *scenario* pada *Feature* pada *Automated Testing System*.

A. Submodule - Login & Open Page

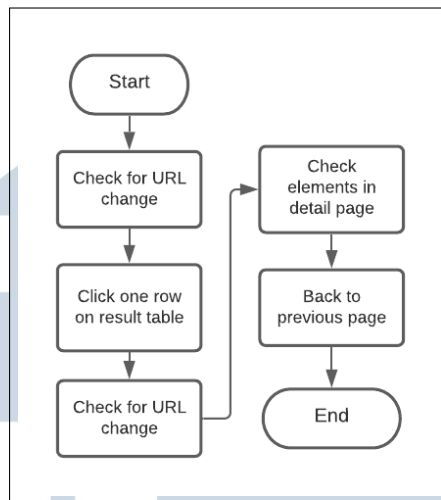
Scenario pada Gambar 3.9 ini merupakan bagian dari sub-modul untuk langkah '*Login & Open Page*' pada *flowchart* yang ada. Dalam submodul ini, dilakukan proses login user dan navigasi untuk membuka page dari fitur yang diinginkan.



Gambar 3.9. *Flowchart scenario* cek detail

B. Submodule - Check Detail

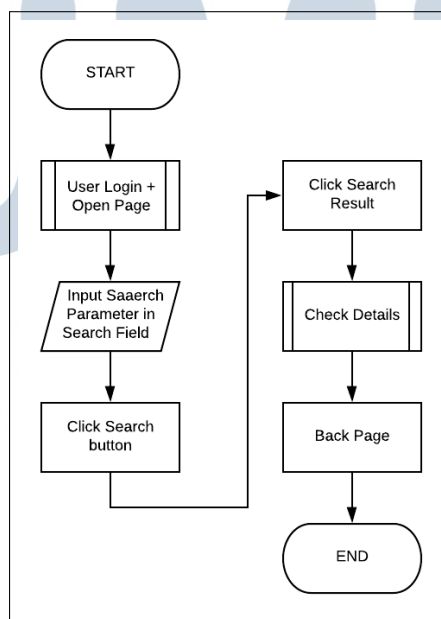
Scenario pada Gambar 3.10 ini merupakan bagian dari sub-modul untuk langkah '*Check Detail*' dan '*Check Data Detail*' pada *flowchart* yang ada. Dalam submodul ini, dilakukan pengecekan detail atas hasil dari pencarian yang dilakukan.



Gambar 3.10. *Flowchart scenario cek detail*

C. Scenario Search

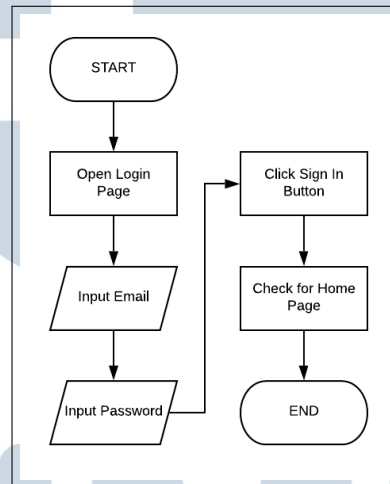
Untuk *scenario* pencarian, hampir seluruh *scenario* yang ada memiliki alur yang sama. Terlihat dari Gambar 3.11, *scenario* dimulai dengan memasukkan *parameter* pencarian sesuai dengan apa yang bisa dicari di *search field* yang ada di halaman *web*. Setelah tombol pencarian ditekan, hasil dari pencarian akan muncul dan dapat dilihat detailnya tergantung dengan jenis data.



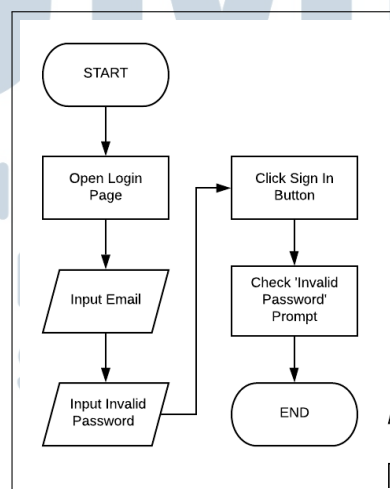
Gambar 3.11. *Flowchart scenario pencarian*

D. Authentication

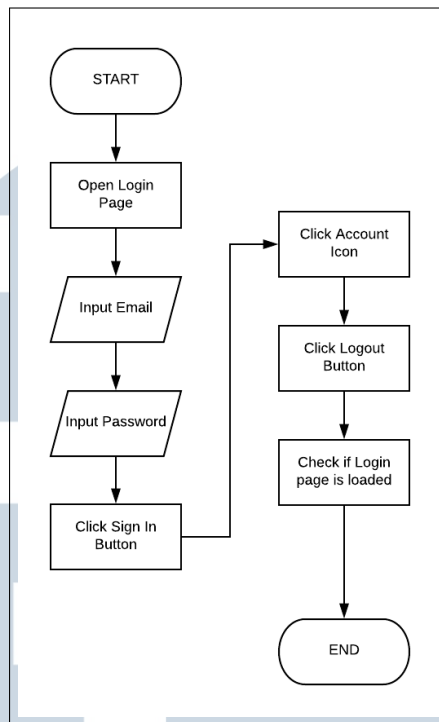
Authentication terdiri dari *scenario* untuk *User Login-Logout*. Pada *Login Feature*, dibuat dua *scenario*, *scenario* untuk *login* sukses (Gambar 3.12) dan *scenario* untuk *login* yang gagal (Gambar 3.13) karena *invalid password*. *Flowchart* ini dipisah dikarenakan untuk menyesuaikan dengan alur dari *scenario*, dimana kedua *scenario* merupakan *scenario* yang berbeda dan masing-masing memiliki alur satu arah saat *testing* dijalankan. Sedangkan untuk *Logout Feature* pada Gambar 3.14, *scenario* yang dibuat adalah *positive case* untuk *logout* sukses.



Gambar 3.12. *Flowchart scenario* login sukses



Gambar 3.13. *Flowchart scenario* login gagal

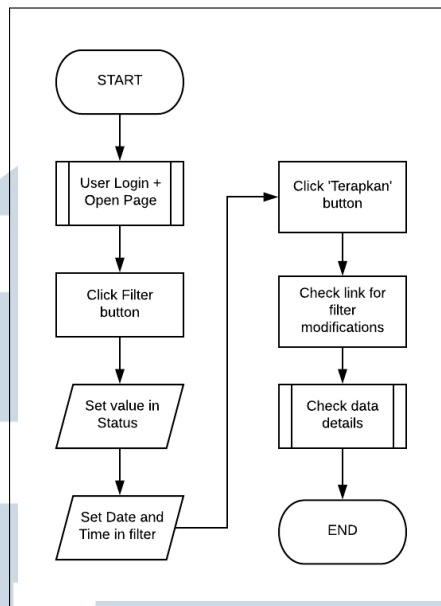


Gambar 3.14. Flowchart scenario logout sukses

E. Member Management

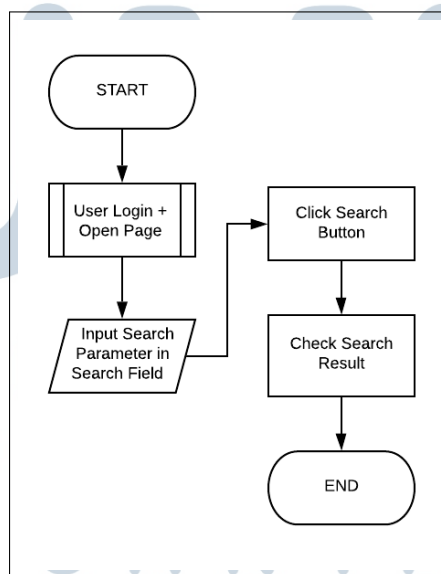
Member Management merupakan tab yang memiliki fitur yang berhubungan dengan *management* dari *member MyValue*, baik secara langsung dari aplikasi atau *website* ataupun melalui *MySantika*. Pada tab ini, hampir semua *Feature* memiliki *scenario* untuk melakukan pencarian, dengan *flowchart* yang sesuai dengan Gambar 3.11. Selain itu, ada juga beberapa *scenario* serupa pada beberapa *Feature* yang ada, yaitu *scenario* untuk melakukan *Filter* tabel yang menunjukkan data. Pada *flowchart* Gambar 3.15 dibawah ini, jenis *filter* yang dilakukan adalah *filter* untuk Status dan Tanggal Dibuat.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



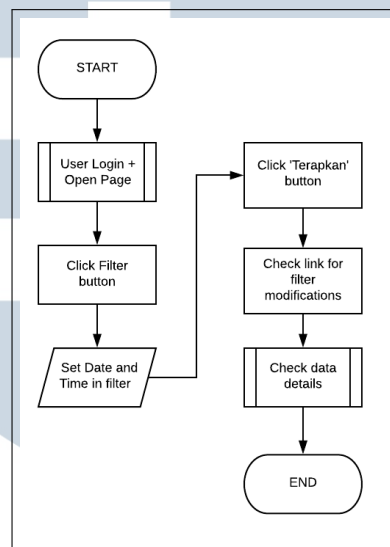
Gambar 3.15. *Flowchart scenario* Filter pada Member Management

Pada fitur Pencarian Member di Gambar 3.16 berikut, *flowchart* sedikit berbeda dari *flowchart* pencarian yang ada sebelumnya. *Scenario* pencarian ini memiliki langkah awal pencarian yang serupa, namun setelah pencarian dilakukan, dilakukan pengecekan data sesuai dengan *parameter* pencarian.

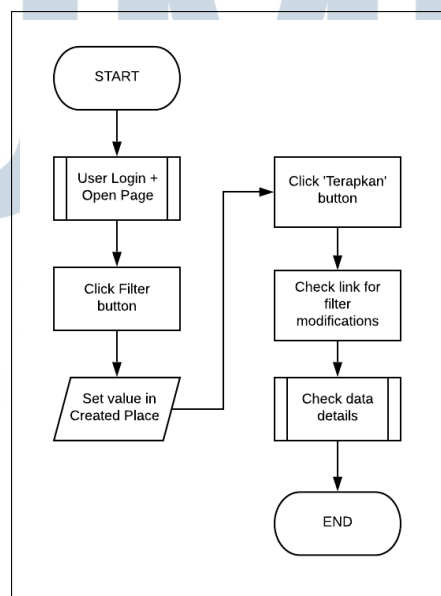


Gambar 3.16. *Flowchart scenario* Pencarian Member pada Member Management

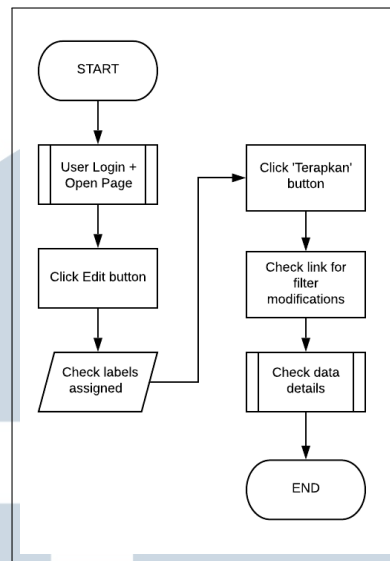
Fitur MyValue *Member List* dan NonMyValue *Member List* memiliki alur *flowchart* yang serupa. *Scenario* yang dilakukan juga serupa, dimana dilakukan pencarian beserta pengecekan detail sesuai dengan Gambar 3.11, *filter scenario* untuk tabel berdasarkan Tanggal Dibuat serta Tempat Dibuat yang dirancang secara terpisah (Gambar 3.17 dan Gambar 3.18), dan *scenario* untuk mengedit kolom tabel yang ditampilkan (Gambar 3.19).



Gambar 3.17. *Flowchart scenario filter* Member List pada Member Management (1)

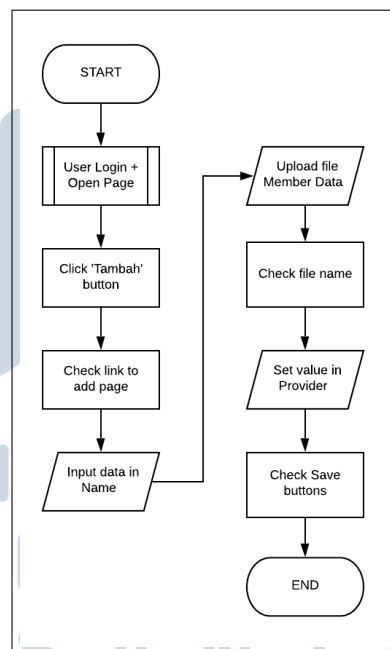


Gambar 3.18. *Flowchart scenario filter* Member List pada Member Management (2)



Gambar 3.19. *Flowchart scenario edit kolom Member List pada Member Management*

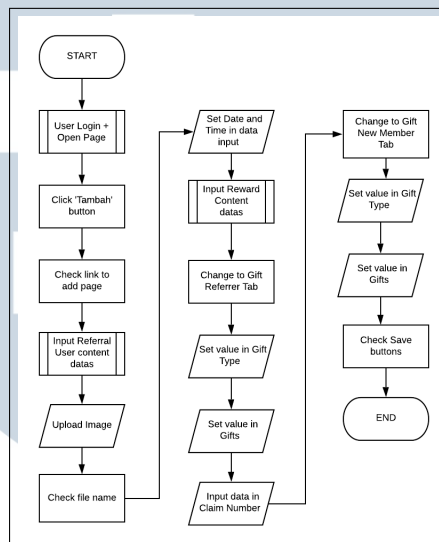
Fitur *Upload Data Member* memiliki *scenario filter status* dan tanggal yang serupa dengan Gambar 3.15 dan *scenario* menambahkan data anggota baru. *Scenario* tersebut dapat dilihat alurnya melalui *flowchart* pada Gambar 3.20 dibawah.



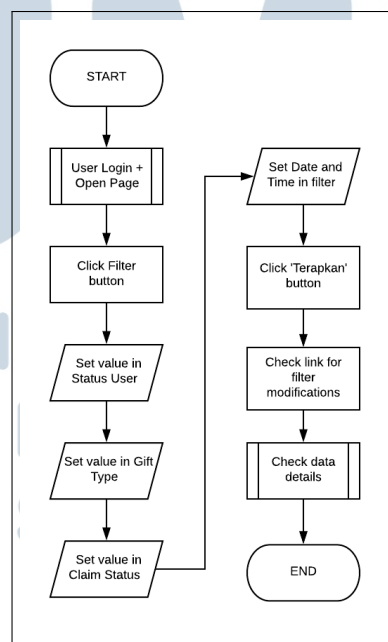
Gambar 3.20. *Flowchart scenario Add Member Data pada Member Management*

Fitur *Referral User* dibagi menjadi tiga Feature, *Referral User Code*, *Referral User Settings*, dan *Referral User Transactions*.

Pada ketiga fitur ini, masing-masing *Feature* memiliki *scenario* pencarian sesuai dengan Gambar 3.11. Selain itu, pada *Referral Settings*, terdapat *scenario* untuk menambahkan *Referral User Settings* dengan *flowchart* pada Gambar 3.21 di bawah. Terdapat juga *scenario filter status* dan tipe hadiah yang dimiliki *Referral User Transaction Feature* yang alurnya dijelaskan melalui *flowchart* pada Gambar 3.22.

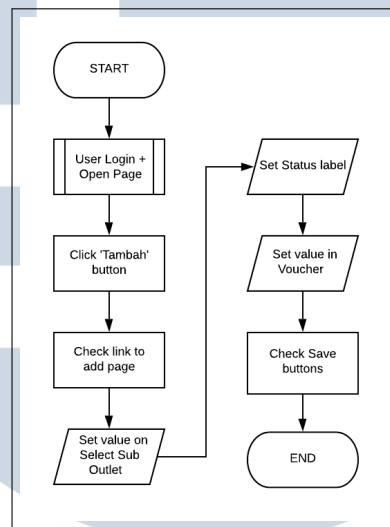


Gambar 3.21. *Flowchart scenario Add Referral User* pada Member Management

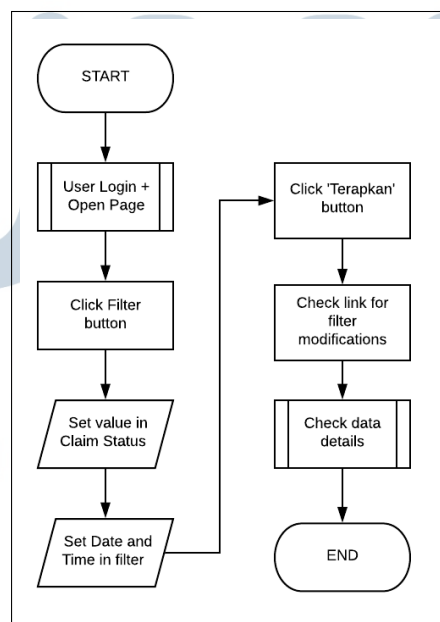


Gambar 3.22. *Flowchart scenario Filter Referral User* pada Member Management

Sedangkan pada fitur *Referral Outlet* terdapat dua *Feature*, *Referral Outlet Settings*, dan *Referral Outlet Transactions*. Kedua fitur ini juga memiliki *scenario* pencarian sesuai dengan Gambar 3.11. Selain itu, fitur ini juga memiliki *scenario* untuk menambah *Referral Outlet* (Gambar 3.23) dan untuk filter *Referral Outlet Transaction* sesuai dengan status klaim dan tanggal dibuat (Gambar 3.24).



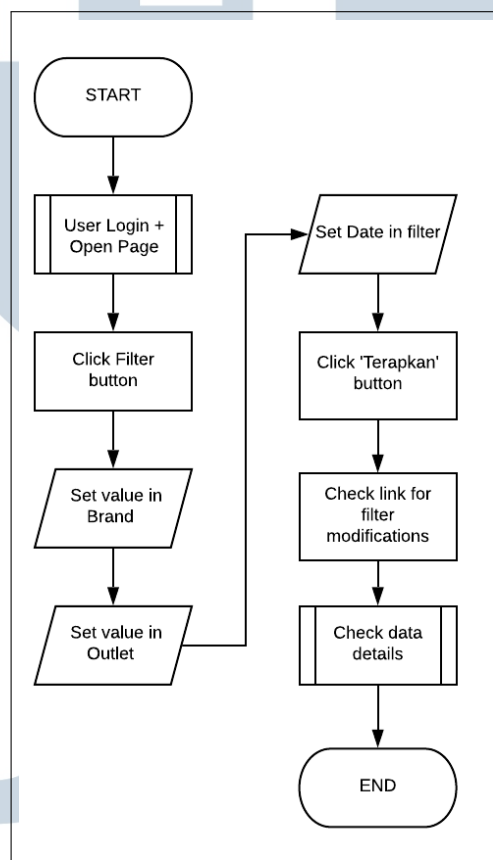
Gambar 3.23. *Flowchart scenario Add Referral Outlet* pada Member Management



Gambar 3.24. *Flowchart scenario Filter Referral Outlet* pada Member Management

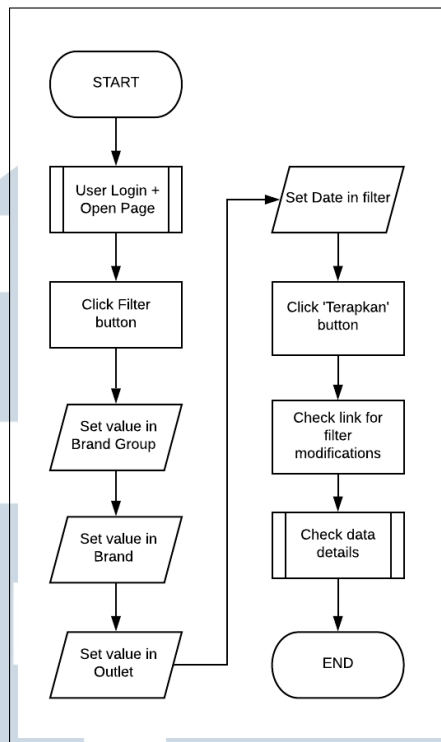
F. Point Management

Point Management mengelola data mengenai distribusi poin serta laporan terkait poin dari MyValue, baik poin yang digunakan dalam transaksi hingga yang hangus karena tidak terpakai. Tab ini memiliki sepuluh *Feature*, dimana ada beberapa *Feature* yang hanya memiliki scenario untuk pencarian data dan *filter* data. *Scenario filter* data pada tab ini adalah *filter* untuk jenis *Merchant* dan tanggal (Gambar 3.25 dan Gambar 3.26).



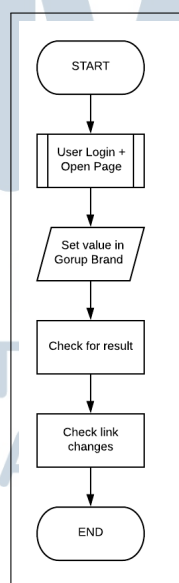
Gambar 3.25. Flowchart scenario Filter pada Point Management (1)

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.26. Flowchart scenario Filter pada Point Management (2)

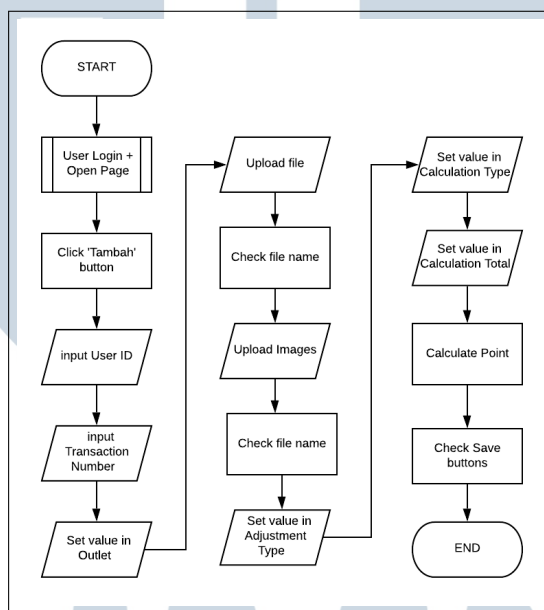
Pada Gambar 3.27, *Point Balance Feature* menyajikan laporan pada *Group Brand* terkait data transaksi poin yang terjadi pada *Group Brand* tersebut. Data ini ditampilkan secara *realtime* sesuai dengan *Group Brand* yang dipilih.



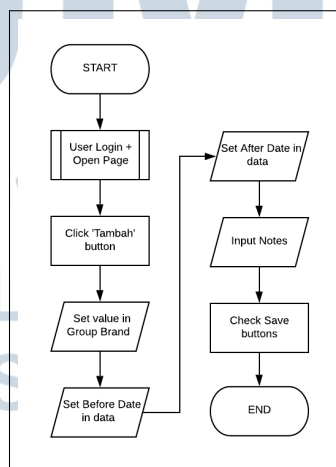
Gambar 3.27. Flowchart scenario Point Balance pada Point Management

Feature Earning, Refund, Redeem, dan On Hold hanya memiliki *scenario* untuk pencarian data dan *filter* data sesuai dengan Gambar 3.26. Sedangkan pada *Fraud*, *scenario* yang dimiliki adalah *scenario* pencarian data dan *filter* sesuai dengan Gambar 3.25.

Feature Adjustment dan Change Expired Date memiliki *scenario* pencarian data dan *filter* sesuai dengan Gambar 3.25. Selain itu terdapat juga *scenario* penambahan data dengan alur terlihat pada Gambar 3.28 dan Gambar 3.29.

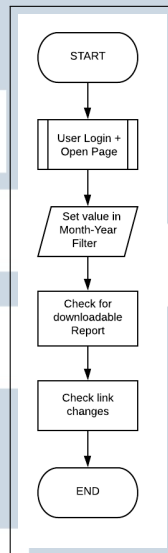


Gambar 3.28. Flowchart *scenario* Adjustment pada Point Management



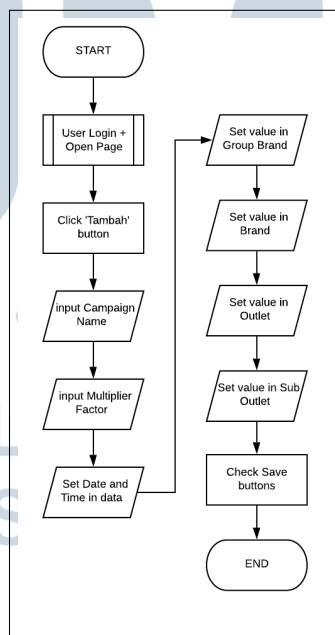
Gambar 3.29. Flowchart *scenario* Expired Date pada Point Management

Pada Gambar 3.30, *Feature Expired Report* menampilkan halaman yang menentukan laporan mengenai poin yang hangus karena tidak dipakai dan dapat dilihat berdasarkan bulan.



Gambar 3.30. *Flowchart scenario Filter* pada Point Management

Feature Campaign Point memiliki *scenario* pencarian data dan filter sesuai dengan Gambar 3.25. Selain itu, ada juga *scenario* penambahan data yang dapat dilihat dari Gambar 3.31.

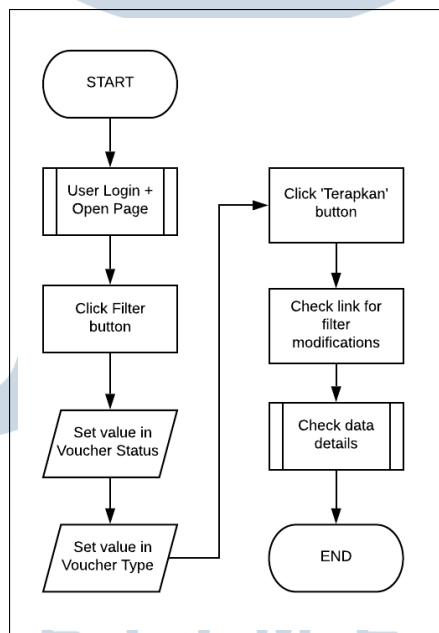


Gambar 3.31. *Flowchart scenario Campaign Point* pada Point Management

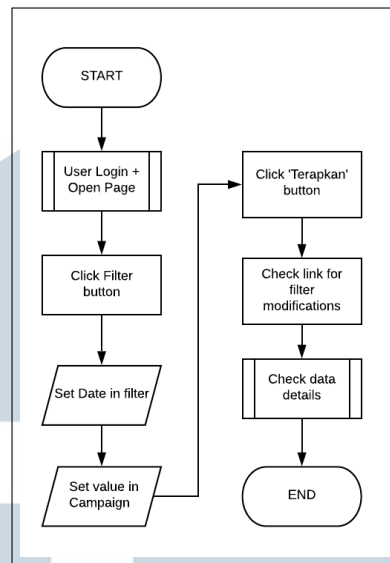
G. Voucher and Transaction Management

Voucher and Transaction Management merupakan tab yang mengatur keseluruhan alur transaksi serta mengelola *Voucher* dan *Campaign* yang ada pada MyValue. Tab ini memiliki tujuh *Feature* utama dengan beberapa *Feature* tambahan di dalamnya; *Welcome Gift*, *Voucher Bundle*, *List Voucher*, *Voucher Quota Identifier*, *Transactional Data*, *Campaign*, dan *Payment*.

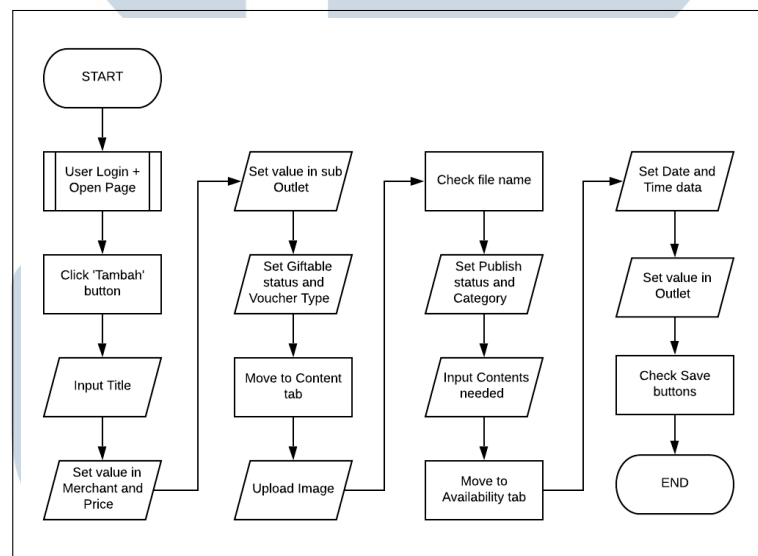
Feature List Voucher mengelola data dari *voucher* yang dibuat untuk MyValue. Pada *Feature* ini, terdapat *scenario* pencarian data, *filter* dengan beberapa *parameter*, dan penambahan data (Gambar 3.34). *Scenario* pencarian serupa dengan *scenario* pada Gambar 3.11, dan *scenario filter* dipisah menjadi dua *flowchart* untuk membedakan *parameter* yang dipilih, seperti terlihat pada Gambar 3.32 dan Gambar 3.33. Sedangkan untuk penambahan data, terdapat dua *scenario* yang serupa, dimana perbedaannya adalah status *voucher* (*reguler* atau *flash sale*).



Gambar 3.32. *Flowchart scenario filter* Voucher pada Voucher-Transaction Management (1)



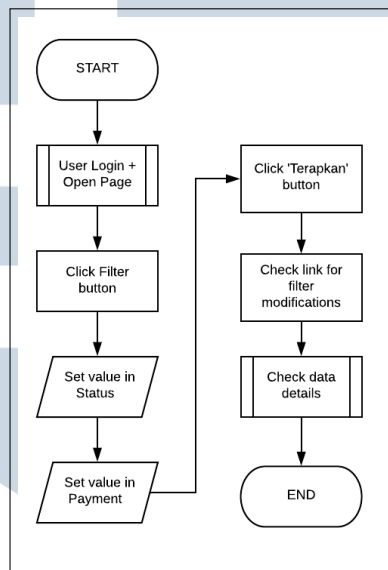
Gambar 3.33. *Flowchart scenario filter* Voucher pada Voucher-Transaction Management (2)



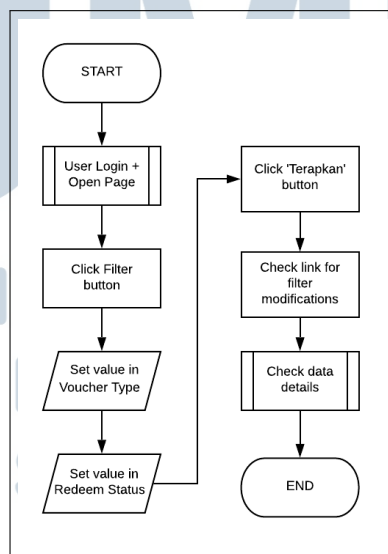
Gambar 3.34. *Flowchart scenario Add Voucher* pada Voucher-transaction Management

Feature Voucher Bundle mengelola fitur yang sesuai namanya, dimana beberapa *Voucher* akan dikumpulkan menjadi sebuah paket yang dapat dibeli dengan harga yang lebih murah. *Feature* ini terbagi menjadi tiga *Feature* lainnya, dimana terdapat *scenario* untuk *Feature* masing-masing. *List Voucher Bundle* mengelola data *Voucher Bundle* yang ada, dimana terdapat *scenario* pencarian dan penambahan data seperti pada Gambar 3.37. Pada *Voucher Bundle*

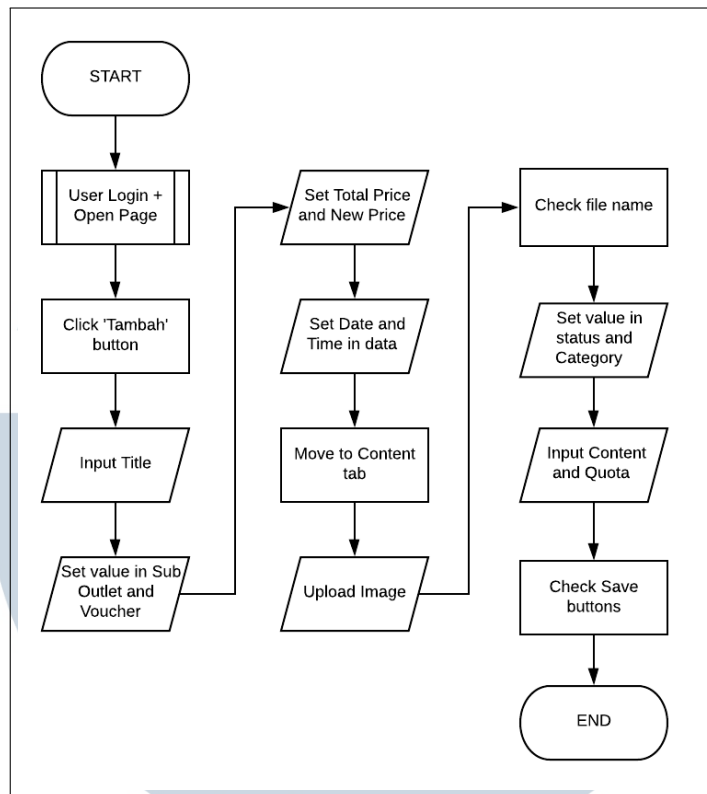
Code, data yang diolah adalah data *redeem code* untuk *voucher bundle* yang dapat dimasukkan melalui *upload file*. Sedangkan untuk *Voucher Code Sources* sesuai dengan Gambar 3.38, data voucher code yang telah diupload dapat dilihat disini. Ketiga fitur ini juga memiliki *scenario* untuk melakukan *filter* data, dengan flowchart pada Gambar 3.35 dan Gambar 3.36.



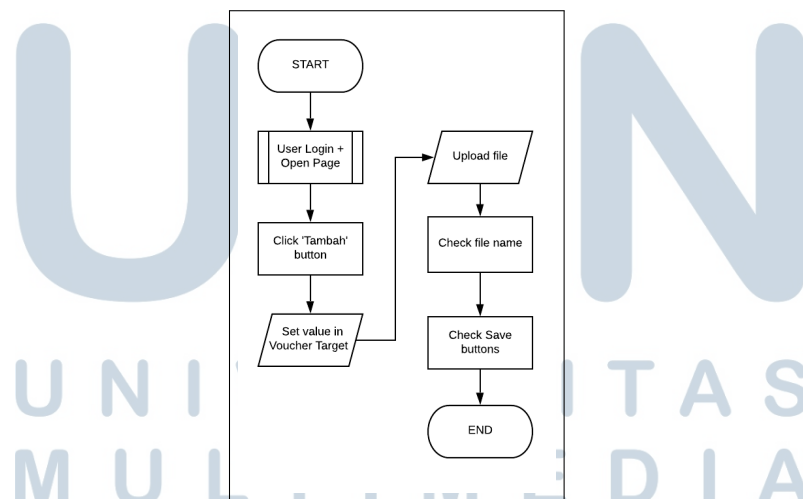
Gambar 3.35. *Flowchart scenario filter Voucher Code* pada *Voucher-Transaction Management* (1)



Gambar 3.36. *Flowchart scenario filter Voucher Code* pada *Voucher-Transaction Management* (2)

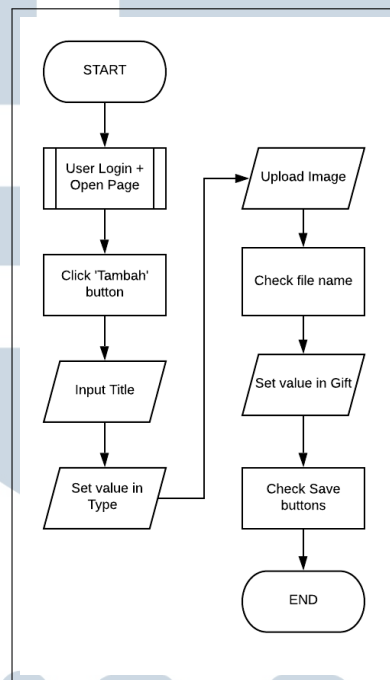


Gambar 3.37. *Flowchart scenario add Voucher Bundle pada Voucher-Transaction Management*



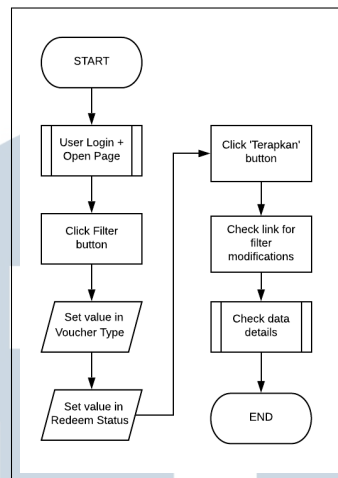
Gambar 3.38. *Flowchart scenario add Voucher Bundle Code pada Voucher-Transaction Management*

Feature Welcome Gift mengelola pemngiriman hadiah untuk anggota baru men-download dan melakukan registrasi sebagai anggota dari MyValue. Pada fitur ini, terdapat *scenario* untuk melakukan pencarian dan penambahan data, dimana *scenario* pencarian data sesuai dengan Gambar 3.11 dan penambahan *data* memiliki *flowchart* seperti pada Gambar 3.39 berikut.

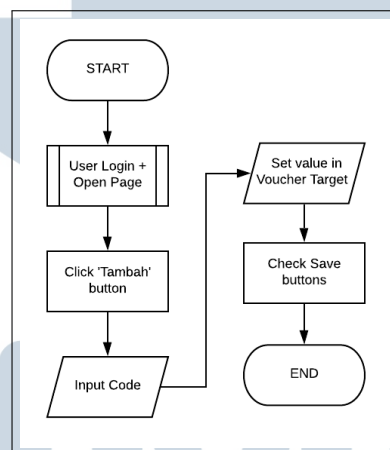


Gambar 3.39. *Flowchart scenario add Welcome Gift pada Voucher-Transaction Management*

Feature Voucher Quota Identifier terdiri atas dua fitur, *Upload Voucher Code* dan *Voucher Code*. *Upload Voucher Code* memiliki alur penambahan data sesuai dengan Gambar 3.38, dan fitur *Voucher Code* memiliki *scenario* untuk melakukan pencarian, *filter*, dan penambahan data. Pencarian memiliki alur serupa dengan Gambar 3.11, sedangkan untuk *scenario filter* dan penambahan data terdapat pada Gambar 3.40 dan Gambar 3.41 di bawah ini.



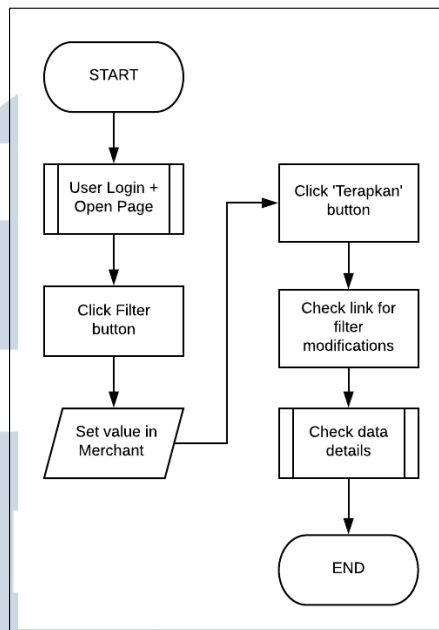
Gambar 3.40. *Flowchart scenario filter* Voucher Code pada Voucher-Transaction Management



Gambar 3.41. *Flowchart scenario add* Voucher Code pada Voucher-Transaction Management

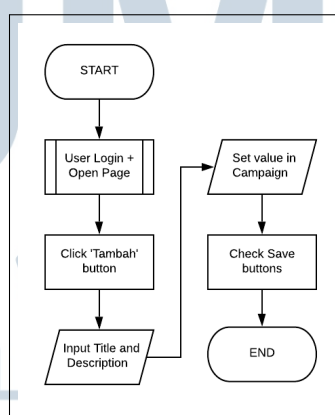
Feature Transactional Data mengelola data transaksi yang terjadi pada penggunaan MyValue. *Feature* ini terbagi atas tujuh fitur lainnya; *Pembelian Voucher*, *Redeem Voucher*, *Campaign Microsite*, *Campaign Microsite Transaction*, *Central Claim System*, *QR Code*, dan *Gift Voucher Transaction*. *Feature Campaign Microsite Transaction* dan *Gift Voucher Transaction* hanya memiliki *scenario* untuk pencarian data, dikarenakan hanya berfokus pada menampilkan data transaksi yang terjadi. Sedangkan *Feature Pembelian Voucher* dan *Redeem Voucher* terdiri atas *scenario* pencarian dan *filter* data. *Filter* data yang ada memiliki *flowchart* seperti pada Gambar 3.42 di

bawah.

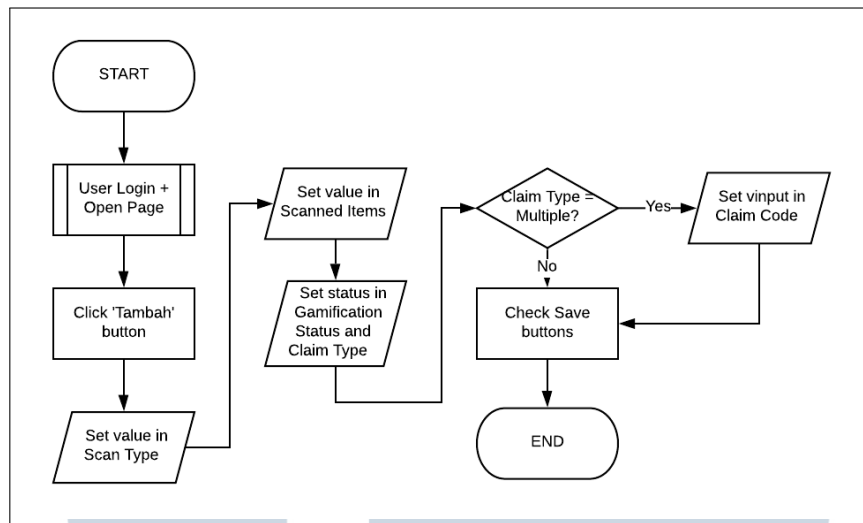


Gambar 3.42. *Flowchart scenario filter* Transactional Data pada Voucher-Transaction Management

Feature Campaign Microsite dan Central Claim System memiliki *scenario* yang serupa; pencarian dan penambahan data. Pada *scenario* penambahan data, kedua fitur memiliki langkah yang berbeda, seperti terlihat pada Gambar 3.43 dan Gambar 3.44.

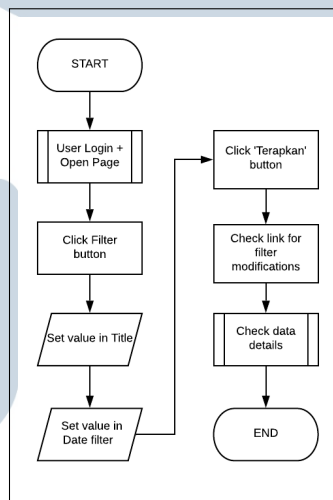


Gambar 3.43. *Flowchart scenario add* Campaign Microsite pada Voucher-Transaction Management



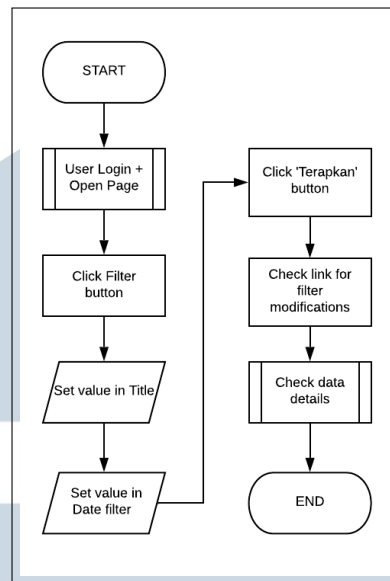
Gambar 3.44. Flowchart scenario add CCS pada Voucher-Transaction Management

QR Code memiliki scenario pencarian, filter, dan penambahan data. Gambar 3.45 dan Gambar 3.46 di bawah merupakan flowchart untuk filter dan penambahan data.



Gambar 3.45. Flowchart scenario add QR Code Voucher-Transaction Management

UNIVERSITAS
MULTIMEDIA
NUSANTARA

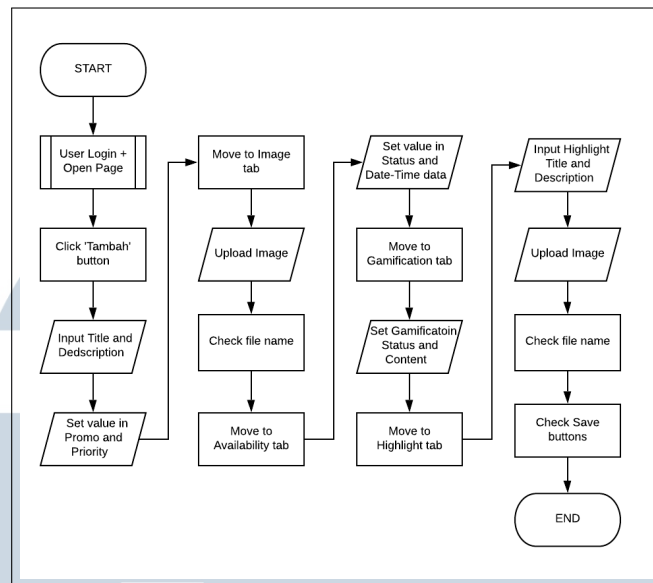


Gambar 3.46. Flowchart scenario filter QR Code pada Voucher-Transaction Management

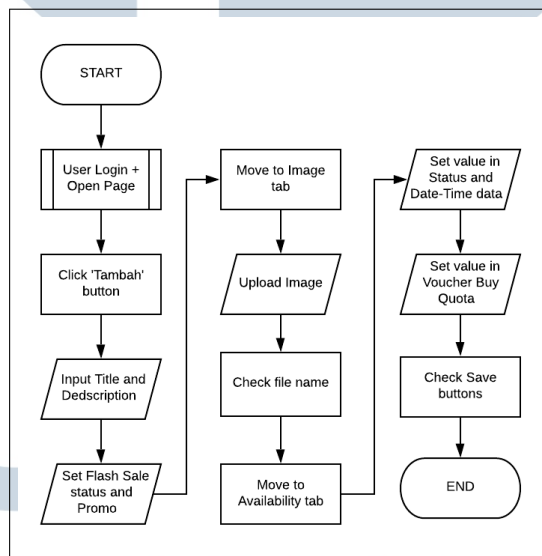
Feature Campaign mengelola data mengenai *Campaign* yang akan berjalan pada MyValue. *Campaign* sendiri terdiri atas kumpulan *Voucher* yang dipilih untuk ditampilkan secara bersamaan sesuai dengan *event* yang ingin dibuat. *Feature* ini terdiri atas fitur *List Campaign*, *Campaign Visitor*, *Action*, *Action Result*, dan *Pre Registered User Upload*. Pada *Feature Campaign Visitor*, scenario yang ada adalah pencarian dan *filter* data, dimana *filter* data dilakukan hanya berdasarkan tanggal.

Pada *List Campaign*, terdapat scenario pencarian, *filter* (Gambar 3.49) dan penambahan data. Pada scenario penambahan data tersebut, terdapat dua jenis *Campaign* yang dapat dibuat, *Campaign* Reguler dan *Flash Sale*, seperti terlihat pada Gambar 3.47 dan Gambar 3.48.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

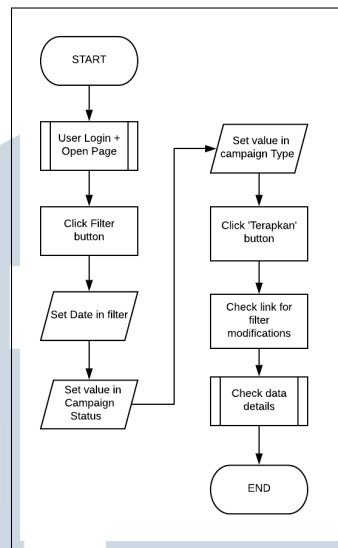


Gambar 3.47. *Flowchart scenario add campaign pada Voucher-Transaction Management (1)*



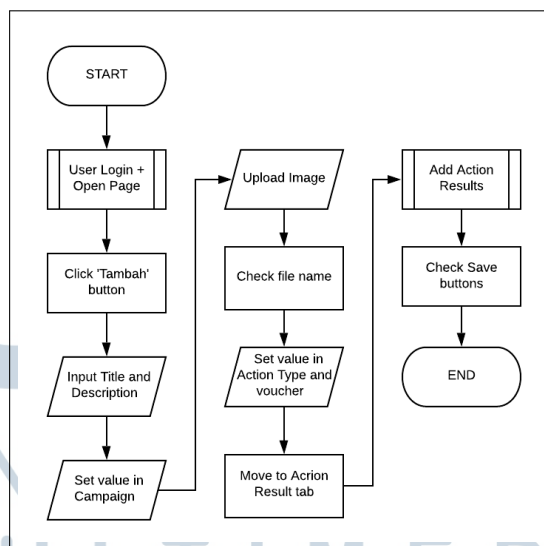
Gambar 3.48. *Flowchart scenario add Campaign pada Voucher-Transaction Management (2)*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

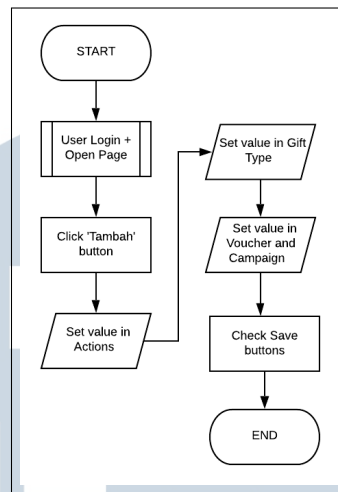


Gambar 3.49. Flowchart scenario filter Campaign pada Voucher-Transaction Management

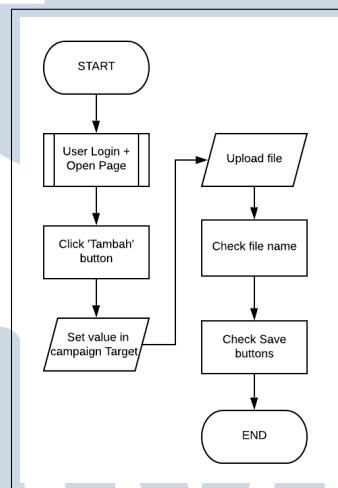
Scenario yang ada pada *Feature Action*, *Action Result*, dan *Pre Registered User Upload* adalah pencarian dan penambahan data. Gambar 3.50, Gambar 3.51, dan Gambar 3.52 berikut merupakan flowchart penambahan data dari ketiga scenario tersebut.



Gambar 3.50. Flowchart scenario add Action pada Voucher-Transaction Management

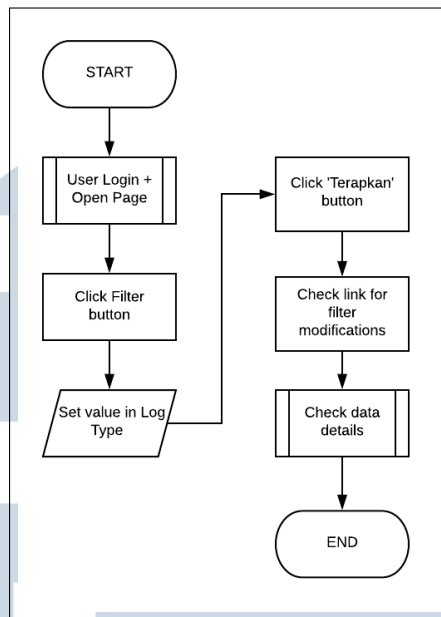


Gambar 3.51. *Flowchart scenario add Action Result pada Voucher-Transaction Management*

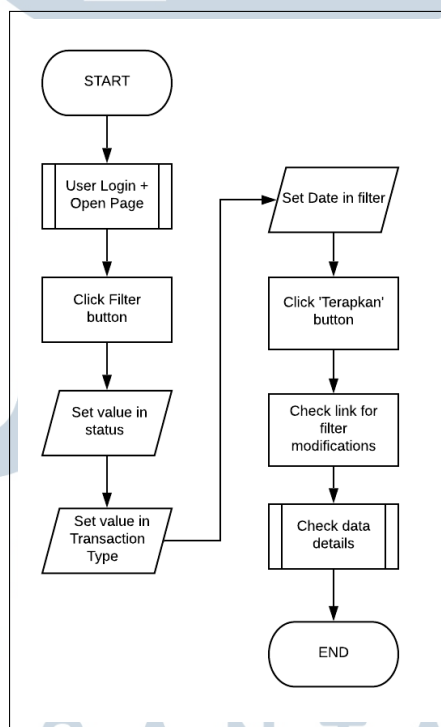


Gambar 3.52. *Flowchart scenario add Preregistered User pada Voucher-Transaction Management*

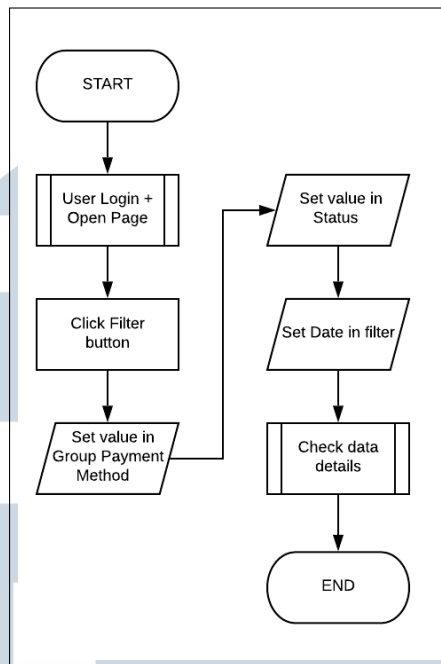
Feature Payment Method terdiri atas fitur *Group Payment Method*, *Payment Method*, *Transactions*, dan *Response Logs*. Pada *Payment Method*, *Transactions*, dan *Response Logs*, *scenario* yang dimiliki adalah *scenario* pencarian dan *filter* data. Gambar 3.53, Gambar 3.46, dan Gambar 3.55 merupakan *flowchart* dari *filter scenario* pada ketiga fitur tersebut.



Gambar 3.53. *Flowchart scenario filter Response Logs* pada Voucher-Transaction Management

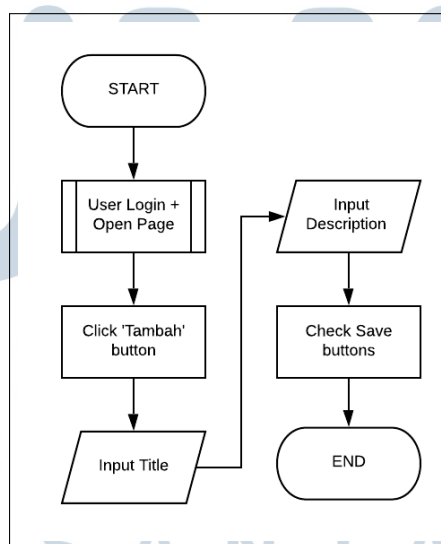


Gambar 3.54. *Flowchart scenario filter Transactions* pada Voucher-Transaction Management



Gambar 3.55. *Flowchart scenario filter* Payment Method pada Voucher-Transaction Management

Pada *Group Payment Method*, terdapat *scenario* untuk pencarian dan penambahan data. Gambar 3.56 berikut merupakan *flowchart* untuk *scenario* penambahan data.

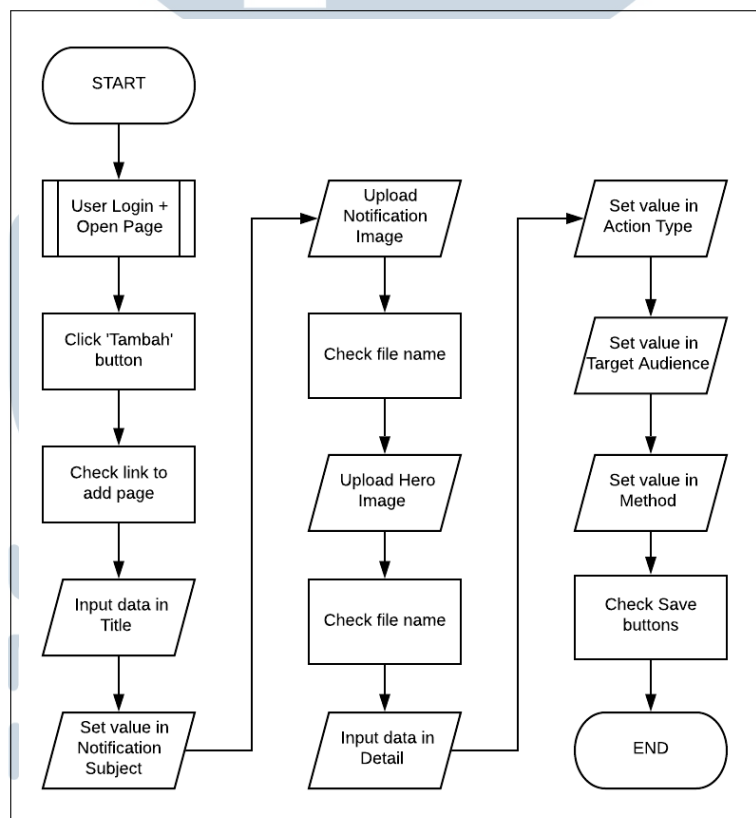


Gambar 3.56. *Flowchart scenario add* Group Payment pada Voucher-Transaction Management

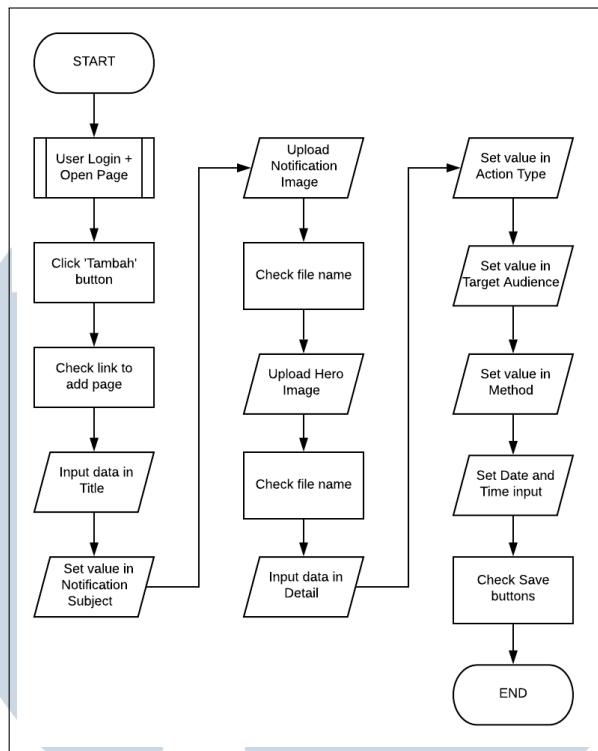
H. Messaging Management

Messaging Management merupakan tab yang diakses untuk mengatur alur dan konten untuk notifikasi yang disebar ke aplikasi ataupun *website* MyValue kepada pengguna. Bentuk notifikasi bisa berupa *inbox notification*, maupun dari *First Open Dialog* (FOD). Selain itu, tab ini juga memiliki fitur untuk membuat subjek notifikasi yang baru dan mengecek *Firestore Cloud Messaging* (FCM) *Device* dari pengguna. Fitur notifikasi dan subjek notifikasi memiliki *scenario* pencarian data seperti pada Gambar 3.11.

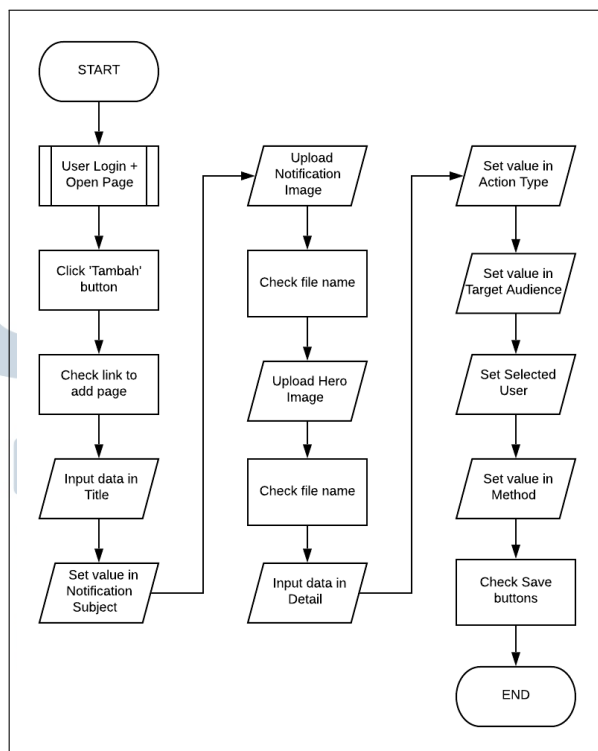
Fitur *Inbox Notification* memiliki *scenario* pembuatan notifikasi yang dapat dibagi menjadi empat jenis; notifikasi dengan *target* umum dan tanpa penjadwalan (Gambar 3.57), notifikasi dengan *target* umum dengan penjadwalan (Gambar 3.58), notifikasi dengan *target* tertentu dan tanpa penjadwalan (Gambar 3.59), dan notifikasi dengan *target* tertentu dengan penjadwalan (Gambar 3.60)



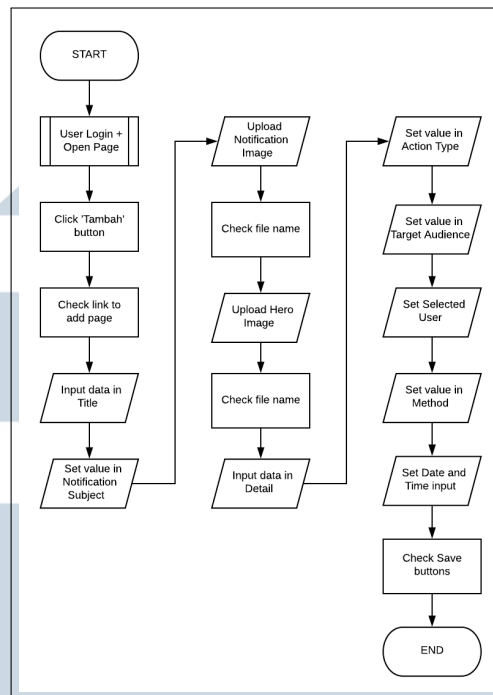
Gambar 3.57. Flowchart scenario Notification pada Member Management (1)



Gambar 3.58. Flowchart scenario Notification pada Member Management (2)

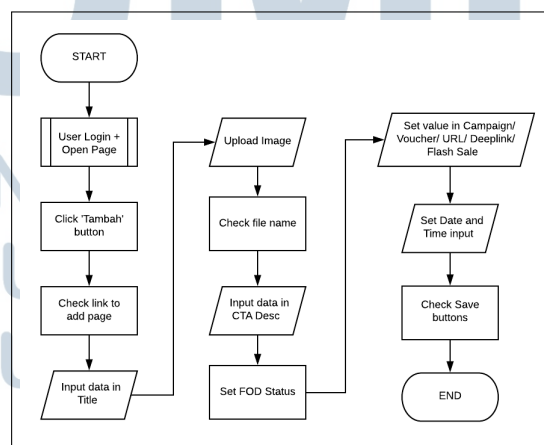


Gambar 3.59. Flowchart scenario Notification pada Member Management (3)

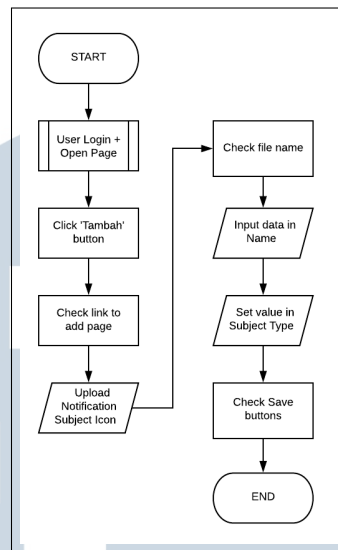


Gambar 3.60. *Flowchart scenario* Notification pada Member Management (4)

Fitur FOD pada *Messaging Management* pada Gambar 3.61 memiliki penambahan FOD. FOD tersebut kemudian akan muncul di *MyValue Apps* begitu *User* membuka aplikasi setelah notifikasi tersebut diluncurkan. Sedangkan fitur subjek notifikasi pada Gambar 3.62 memiliki *scenario* pembuatan subjek notifikasi baru, dimana nantinya dapat dipilih saat pembuatan notifikasi pada *Messaging Management*.

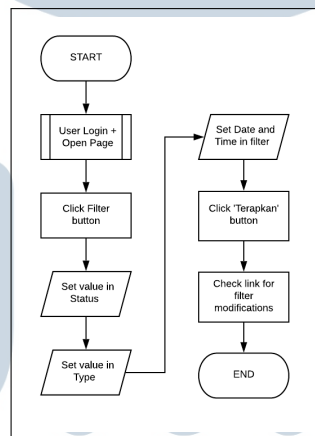


Gambar 3.61. *Flowchart scenario* FOD pada Member Management



Gambar 3.62. Flowchart scenario Notification Subject pada Member Management

Fitur FCM Device pada Gambar 3.63 memiliki *scenario* untuk melakukan *filter* data sesuai jenis *device*, *status*, dan tanggal pembuatan akun.

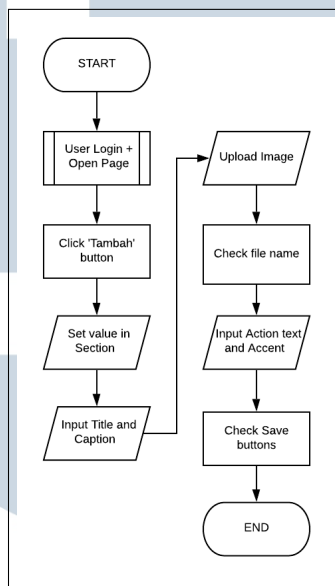


Gambar 3.63. Flowchart scenario FCM Device pada Member Management

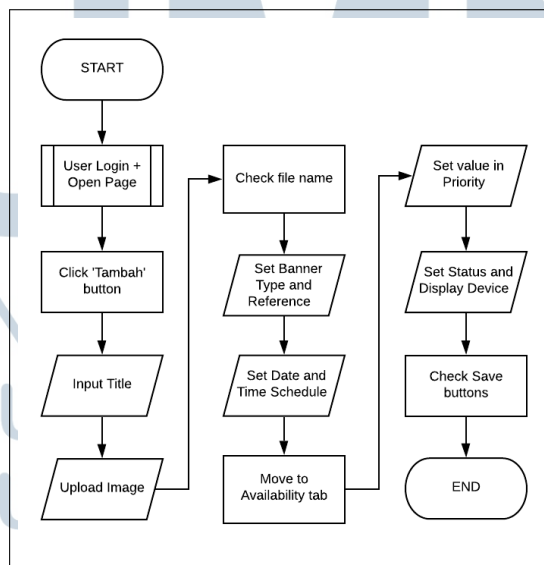
I. Content Management

Content Management merupakan tempat untuk mengolah konten pada MyValue. Tab ini terdiri dari *Feature Home App* (*Home Arrangement*, *Home Banner*, *Applinks*, dan *Point Earning Instructions*), *Badge App* (*Badge*, *Group Badge*), *Kategori*, *Custom HTML Page*, *Version*, *Task Scheduler*.

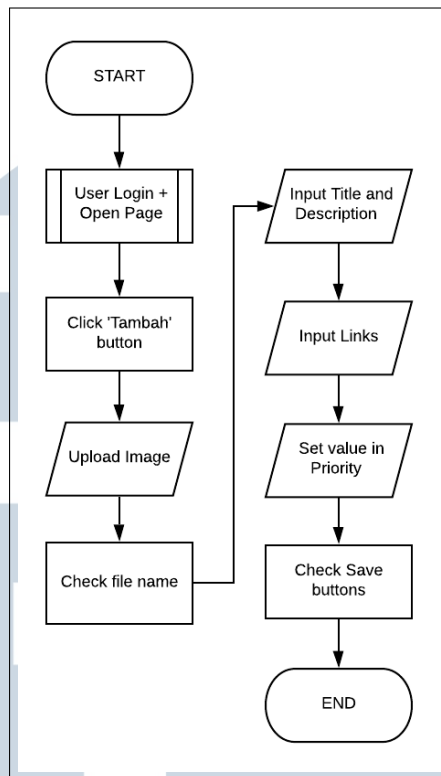
Pada *Feature Home App*, terdapat *scenario* pencarian data pada *Feature Home Banner* dan *Point Earning Instruction*. Keempat *Feature* ini juga memiliki *scenario* untuk menambahkan data pada *Feature* masing-masing, sesuai dengan langkah-langkah yang ditetapkan untuk *Feature* di Gambar 3.64, Gambar 3.65, Gambar 3.66, dan Gambar 3.67.



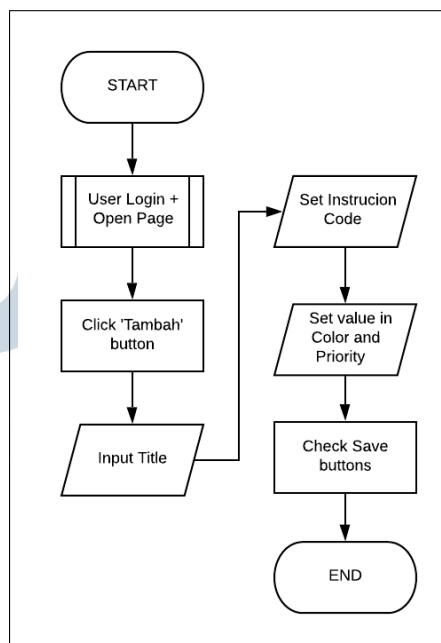
Gambar 3.64. *Flowchart scenario* Home Arrangement pada Content Management



Gambar 3.65. *Flowchart scenario* Home Banner pada Content Management

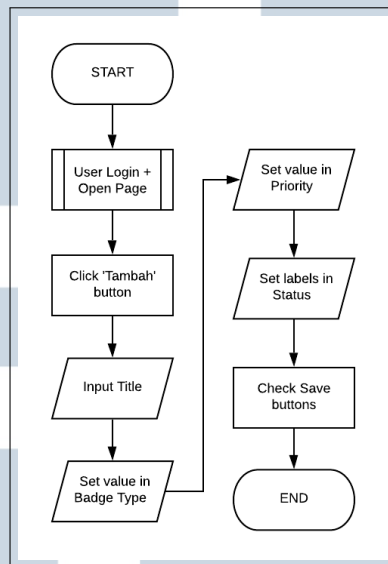


Gambar 3.66. *Flowchart scenario* Applinks pada Content Management

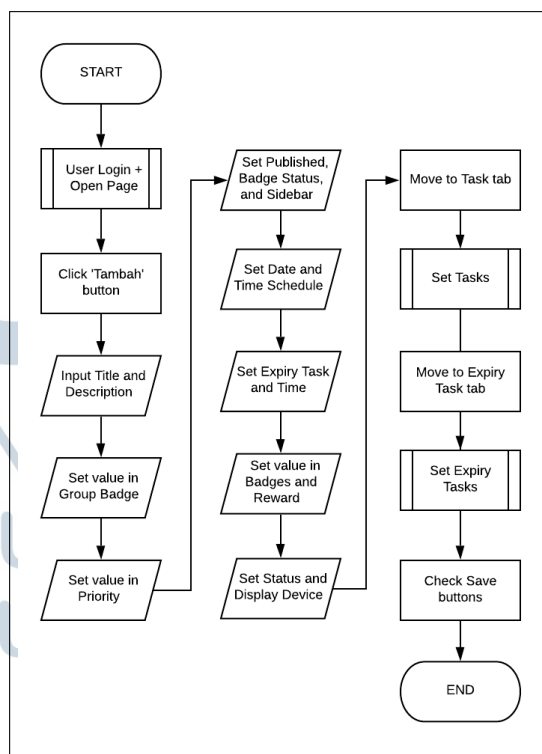


Gambar 3.67. *Flowchart scenario* Point Instructions pada Content Management

Feature Badge App, scenario yang dimiliki oleh kedua *Feature* adalah *scenario* pencarian dan penambahan data. Berikut merupakan *flowchart* dari *scenario* penambahan data pada *Badge* (Gambar 3.69) dan *Group Badge* (Gambar 3.68)

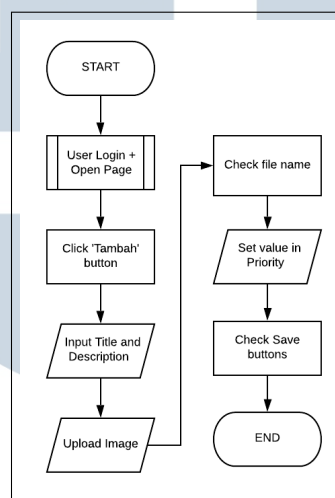


Gambar 3.68. *Flowchart scenario* Badge Group pada Content Management

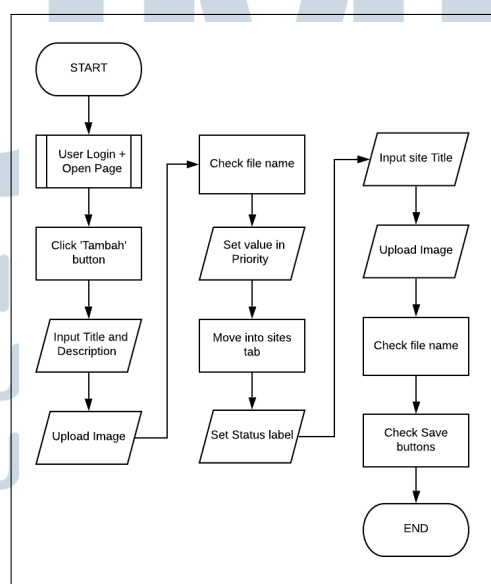


Gambar 3.69. *Flowchart scenario* Badge pada Content Management

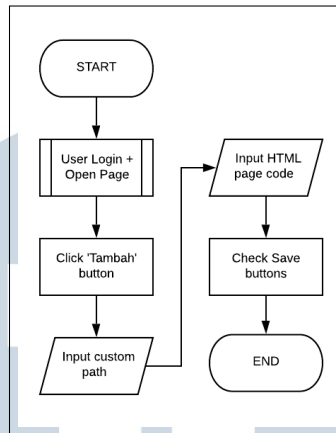
Feature Kategori, Custom HTML Page, dan Task Scheduler, terdapat scenario yang serupa; pencarian dan penambahan data, dengan langkah yang sedikit berbeda pada setiap Feature. Feature Kategori memiliki dua jenis scenario penambahan data, dimana ada tambahan kategori lainnya pada saat dilakukan penambahan (Gambar 3.70 dan Gambar 3.71). Selain itu, Custom HTML (Gambar 3.72 dan Task Scheduler juga memiliki dua scenario penambahan data (Gambar 3.73 dan Gambar 3.74), dan scenario Filter data (Gambar 3.75).



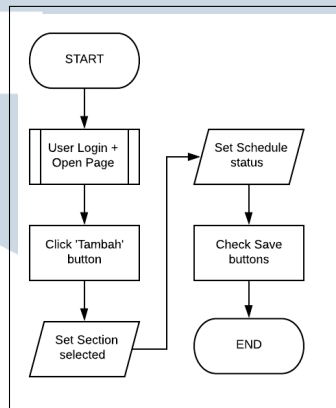
Gambar 3.70. *Flowchart scenario* Kategori pada Content Management (1)



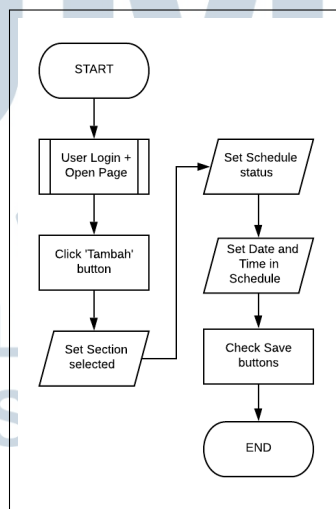
Gambar 3.71. *Flowchart scenario* Kategori pada Content Management (2)



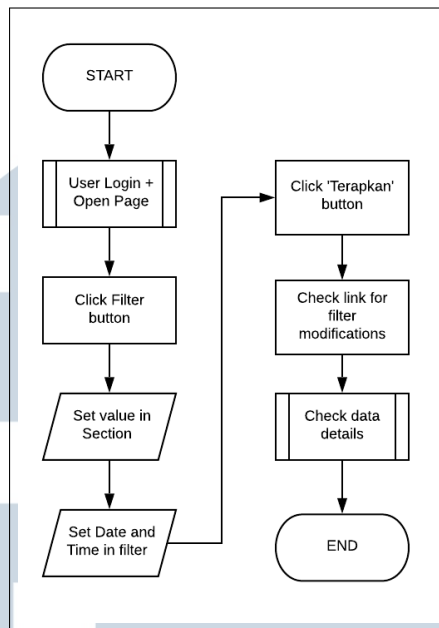
Gambar 3.72. *Flowchart scenario* Custom HTML pada Content Management



Gambar 3.73. *Flowchart scenario* Task pada Content Management (1)

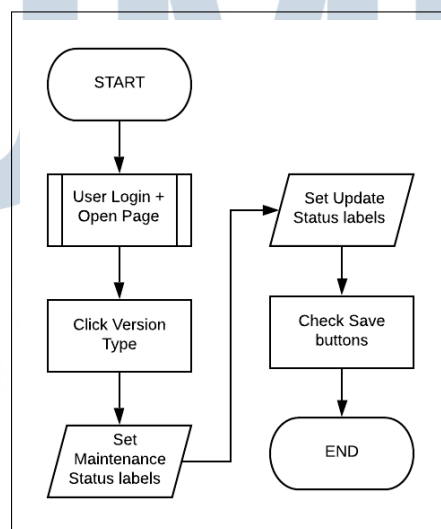


Gambar 3.74. *Flowchart scenario* Task pada Content Management (2)



Gambar 3.75. *Flowchart scenario* Task Filter pada Content Management

Pada *Feature Version*, *scenario* yang ada adalah *scenario* untuk mengubah data mengenai versi MyValue Apps, baik di Android maupun di iOS. Selain itu, *scenario* juga dapat mengubah *maintenance status* dan *update status* dari aplikasi, dan dapat memberitahukan kepada *user* mengenai *Update* yang berlaku. Gambar 3.76 ebrikut merupakan *flowchart* dari *scenario* perubahan data versi.

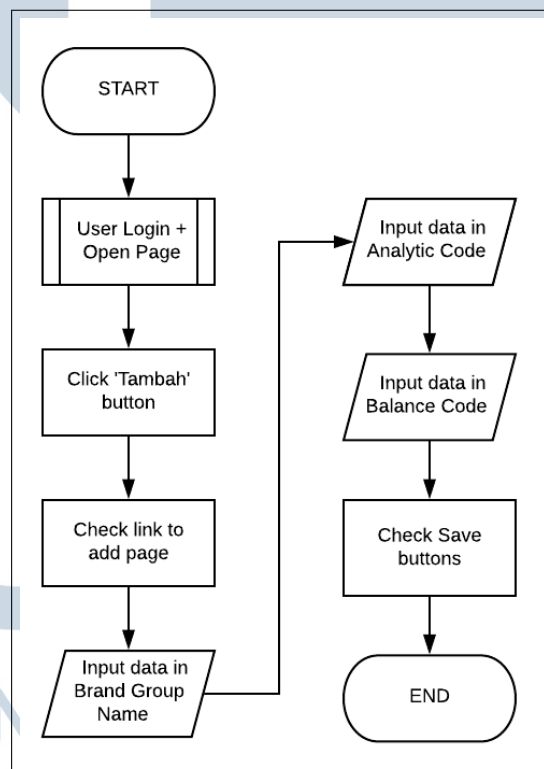


Gambar 3.76. *Flowchart scenario* Version pada Content Management

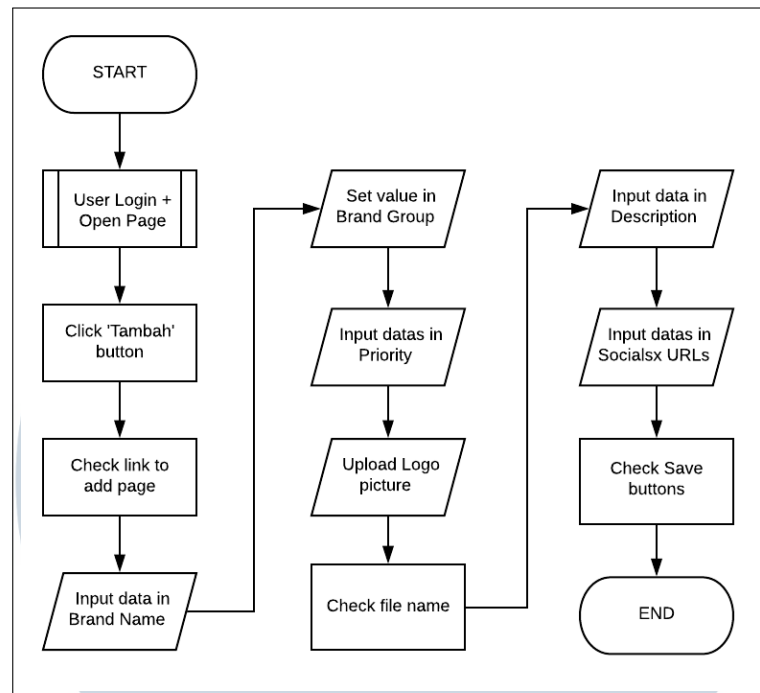
J. Merchant Management

Merchant Management, seperti namanya, merupakan tab yang digunakan untuk mengakses dan mengatur data para *merchant partner* dari MyValue. Tab ini memiliki enam *Feature* yang dibuat sesuai dengan daftar fitur yang disediakan; *Group Brand*, *Brand*, *PT*, *Outlet*, *Sub Outlet*, dan *Upload Data Outlet*.

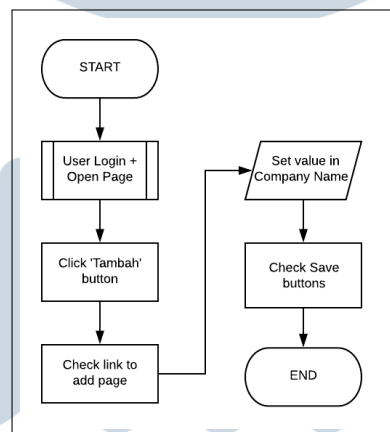
Pada *Merchant Management*, seluruh *Feature* yang ada memiliki *scenario* yang sama, yaitu *scenario* untuk melakukan pencarian data, sesuai dengan *flowchart* 3.11. Keenam *Feature* ini juga memiliki *scenario* untuk menambahkan data di masing-masing *Feature*, dengan beberapa langkah unik di dalamnya seperti yang terlihat pada Gambar 3.77 hingga Gambar 3.82 berikut.



Gambar 3.77. Flowchart scenario Add Group Brand pada Merchant Management

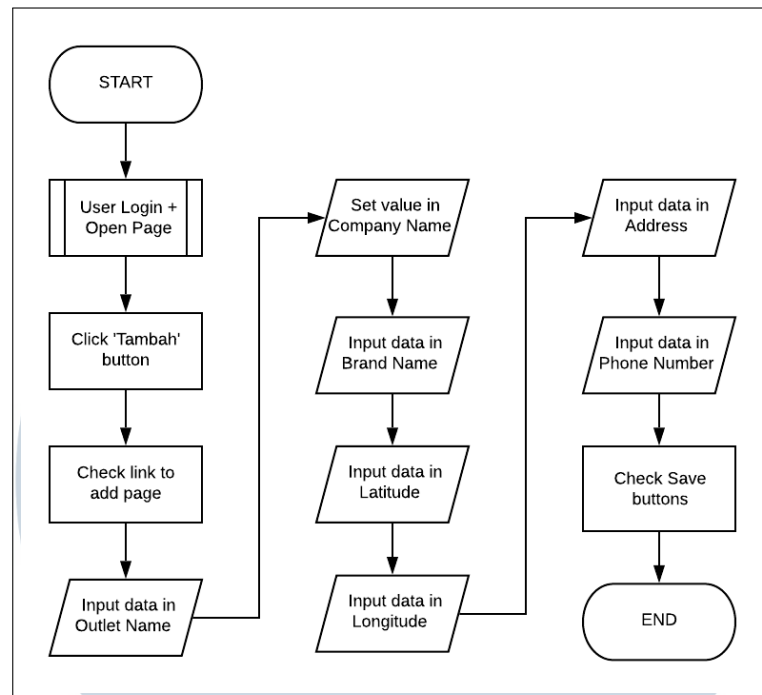


Gambar 3.78. *Flowchart scenario Add Brand pada Merchant Management*

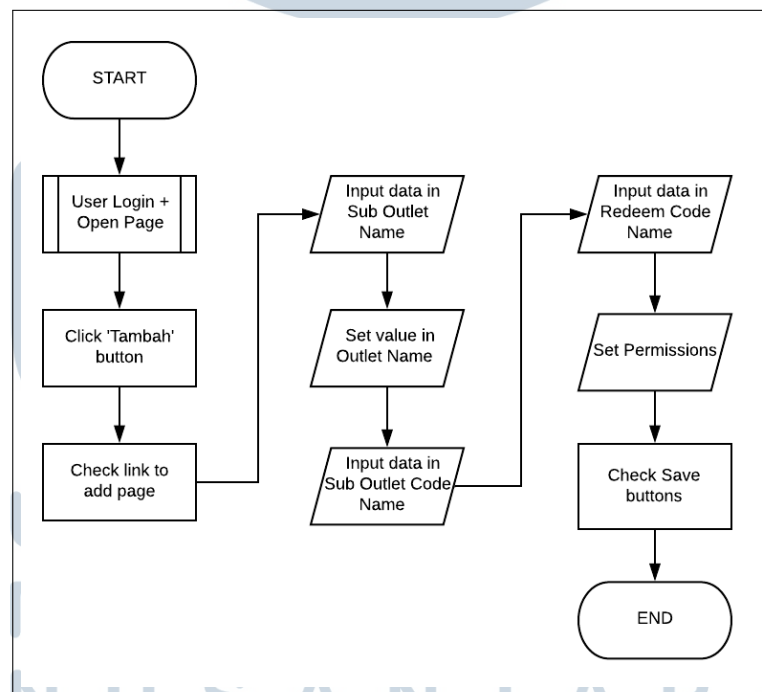


Gambar 3.79. *Flowchart scenario Add PT pada Merchant Management*

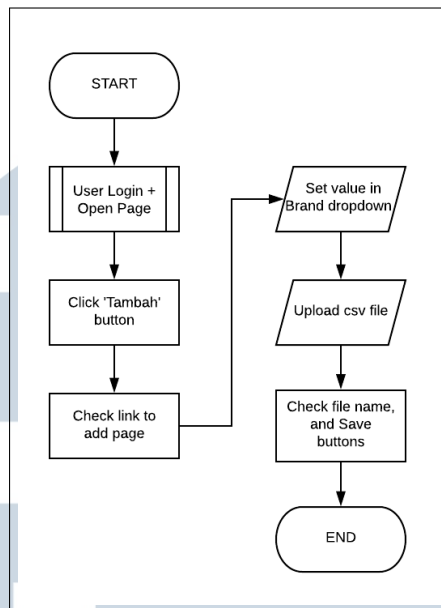
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.80. Flowchart scenario Add Outlet pada Merchant Management



Gambar 3.81. Flowchart scenario Add Sub Outlet pada Merchant Management



Gambar 3.82. Flowchart scenario Add Data Outlet pada Merchant Management

3.4.3 Test Case Scneario

Dalam perancangan *scenario* pada *Automated Testing System* yang ada, *test case scenario* dibuat langsung menggunakan *Gherkin language* ke dalam Feature tersebut. Dengan menggunakan *Gherkin*, *tester* dapat menggunakan *natural language* untuk merancang *scenario* yang mudah dimengerti dengan menggunakan *step definitions* yang telah dibuat. Berikut merupakan beberapa contoh dari *Javascript* yang dibuat, *step definitions* yang dirangkai, dan *scenario* yang diciptakan dari penggunaan *Gherkin language*.

A. Javascript Scripts

Dengan mengimplementasikan fungsi-fungsi yang disediakan *Nightwatch.js*, dapat dibuat berbagai *script* yang digunakan untuk berinteraksi dengan *DOM* pada *website*. Berikut merupakan beberapa contoh *script* interaksi pada elemen pada *Automated Testing System* yang dibuat.

A.1 Click Button

Fungsi ini akan menjalankan perintah untuk melakukan interaksi *click* ke elemen button pada *web-page*, dimana *element* yang diinteraksi harus memiliki *parameter* yang sesuai dengan variabel yang dimasukkan saat perintah dijalankan. Potongan code pada Gambar 3.83 di bawah merupakan bentuk penerapan dari fungsi tersebut.

```
export const clickButtonWithID = async (text, falseCase, elementId) => {
  if (falseCase){
    await client.useXPath();

    checkIfElementExists("//button[.//*[text()= '" + text + "'] and ancestor::*[contains(@id,'" + elementId + "')]]", false);

    await client.click("//button[.//*[text()= '" + text + "'] and ancestor::*[contains(@id,'" + elementId + "')]]");

    await client.useCss();
  } else {
    await client.useXPath();

    checkIfElementExists("//button[.//*[contains(text(), '" + text + "') and ancestor::*[contains(@id,'" + elementId + "')]]", false);

    await client.click("//button[.//*[contains(text(), '" + text + "') and ancestor::*[contains(@id,'" + elementId + "')]]");

    await client.useCss();
  }
};
```

Gambar 3.83. Javascript function untuk interaksi Click Button

A.2 Input Value

Fungsi ini akan menjalankan perintah untuk melakukan interaksi *input text* ke elemen *input* dengan *parameter* yang diinginkan (*class*, *id*, *placeholder*, dsb). Potongan code pada Gambar 3.84 di bawah merupakan bentuk penerapan dari fungsi tersebut.

```
export const inputWithParameter = async (text, parameter, parameterContent) => {
  if (parameter.toLowerCase() != "text") {
    parameter = "@" + parameter.toLowerCase();
  } else {
    parameter = parameter.toLowerCase() + "()";
  }

  await client.useXPath();

  checkIfElementExists("//*[contains('" + parameter.toLowerCase() + "', '" + parameterContent + "') and (self::input or self::textarea or self::p)]", false);

  await client.setValue("//*[contains('" + parameter.toLowerCase() + "', '" + parameterContent + "') and (self::input or self::textarea or self::p)]", text);

  await client.useCss();
};
```

Gambar 3.84. Javascript function untuk interaksi Input Text

A.3 Click dropdown

Fungsi ini akan menjalankan perintah untuk melakukan interaksi *click list* yang ada pada daftar *dropdown* yang sedang dibuka. Fungsi ini dapat dijalankan sekali ataupun beberapa kali tergantung dengan jenis *input* yang dimiliki *dropdown* yang dibuka tersebut. Potongan code pada Gambar 3.85 di bawah merupakan bentuk penerapan dari fungsi tersebut.

```
export const clickDropdownList = async (falseCase, text) => {
  if(falseCase) {
    await client.useXPath();

    await client.click("//div[contains(@style, 'absolute') or contains(@style, 'fixed')]/descendant::li[.//*[text()='\" + text + "\']]");

    await client.useCss();
  } else {
    await client.useXPath();

    await client.click("//div[contains(@style, 'absolute') or contains(@style, 'fixed')]/descendant::li[.//*[contains(text(), '\" + text + '\')]");

    await client.useCss();
  }
};
```

Gambar 3.85. Javascript function untuk interaksi Click Dropdown List

A.4 Scroll into Elements

Fungsi ini akan menjalankan perintah untuk melakukan interaksi *scroll* ke bagian elemen *webpage* yang dipilih saat perintah dijalankan. Potongan code pada Gambar 3.86 di bawah merupakan bentuk penerapan dari fungsi tersebut.

```
export const getElementInView = async (falseCase, elemenName, parameter, parameterContent) => {
  if (parameter.toLowerCase() != "text") {
    parameter = "@" + parameter.toLowerCase();
  } else {
    parameter = parameter.toLowerCase() + "()";
  }

  elemenName = elemenName.toLowerCase();

  if(falseCase) {
    await client.useXPath();

    await client.getLocationInView("//" + elemenName + "[" + parameter + "=" + parameterContent + "]");

    await client.useCss();
  } else {
    await client.useXPath();

    await client.getLocationInView("//" + elemenName + "[contains(" + parameter + ", " + parameterContent + ")]");

    await client.useCss();
  }
};
```

Gambar 3.86. Javascript function untuk interaksi Scroll to Element

A.5 Switch Browser Tab

Fungsi ini akan menjalankan perintah untuk melakukan interaksi mengganti fokus dari tab *browser* awal ke tab *browser* lainnya. Potongan code pada Gambar 3.87 di bawah merupakan bentuk penerapan dari fungsi tersebut.

```
import { client } from 'nightwatch-api';

export const switchWindows = async function (tabs) {
  client.windowHandles(function(result) {
    var handle = result.value[tabs];
    client.switchWindow(handle);
  });
};
```

Gambar 3.87. Javascript function untuk interaksi Switch Tab

B. Step Definitions

Step Definitions merupakan perancangan menggunakan *Gherkin language* saat pembuatan *scenario* dengan meng-*import javascript* yang dibuat dengan *Nightwatch.js* dan dilakukan konfigurasi pada jenis *input* dan variabel yang diterima saat baris tersebut nantinya akan dijalankan dalam *scenario*. Berikut merupakan contoh hasil perancangan *Step Definitions* berdasarkan *javascript* yang ada di contoh sebelumnya.

B.1 Click Button

Setelah fungsi *Javascript* di-*import*, dilakukan perancangan baris *step definition*, dengan meletakkan bagian penarikan variabel; variabel *string* untuk menentukan jenis tulisan pada tombol yang akan diklik, variabel *boolean* untuk menentukan ketepatan dan keakurasian dari teks yang dimasukkan di variabel sebelumnya, dan variabel *string* untuk menentukan *tab id* yang ada pada halaman website. Gambar 3.88 berikut adalah potongan *Gherkin code* untuk fungsi klik tombol.

```
When(/^Saya klik tombol bertulisan "[^"]*" ( saja)* pada tab dengan id "[^"]*"$/, clickButtonWithID);
```

Gambar 3.88. *Step Definitions* untuk interaksi *Click Button*

B.2 Input Value

Setelah fungsi *javascript* di-*import*, dilakukan perancangan baris *step definition*, dengan meletakkan bagian penarikan variabel; variabel *string* untuk menentukan *text* yang dimasukkan ke *input* tersebut, variabel *string* untuk menentukan *parameter* yang akan dicari, dan variabel *string* untuk menentukan *text* yang ada pada *parameter* yang dicari. Gambar 3.89 berikut adalah potongan *Gherkin code* untuk fungsi input nilai.

```
When(/^Saya memasukkan "[^"]*" ke input dengan parameter "[^"]*" "[^"]*"$/, inputWithParameter);
```

Gambar 3.89. *Step Definitions* untuk interaksi *Input Text*

B.3 Click dropdown

Setelah fungsi *javascript* di-*import*, dilakukan perancangan baris *step definition*, dengan meletakkan bagian penarikan variabel; variabel *boolean* untuk menentukan keakurasian dari *text* pada *dropdown list*, dan variabel *string* untuk menentukan *text* pada *list dropdown* yang akan dipilih. Gambar 3.90 berikut adalah potongan *Gherkin code* untuk fungsi klik pilihan pada dropdown.

```
When(/^Saya klik( hanya)* "[^"]*" dari dropdown$/, clickDropdownList);
```

Gambar 3.90. *Step Definitions* untuk interaksi *Click Dropdown*

M U L T I M E D I A
N U S A N T A R A

B.4 Scroll into Elements

Setelah fungsi *javascript* di-*import*, dilakukan perancangan baris *step definition*, dengan meletakkan bagian penarikan variabel; variabel *boolean* untuk menentukan keakurasian *text* pada *parameter*, variabel *string* untuk menentukan elemen yang akan menjadi tujuan *scroll*, variabel *string* yang menentukan *parameter* elemen, dan variabel *string* yang menentukan *text* pada *parameter*. Gambar 3.91 berikut adalah potongan *Gherkin code* untuk fungsi *scroll* ke elemen pada *webpage*.

```
When(/^Saya scroll( hanya)* ke elemen "[^"]*" dengan parameter "[^"]*" "[^"]*"$/, getElementInView);
```

Gambar 3.91. *Step Definitions* untuk interaksi *Scroll into Elements*

B.5 Switch Browser Tab

Setelah fungsi *Javascript* di-*import*, dilakukan perancangan baris *step definition*, dengan meletakkan bagian penarikan variabel; variabel *integer* untuk menentukan urutan tab yang akan diases. Gambar 3.92 berikut adalah potongan *Gherkin code* untuk fungsi klik tombol.

```
When(/^Saya pindah ke tab (-?\d+)$/, switchWindows);
```

Gambar 3.92. *Step Definitions* untuk interaksi *Switch Tab*

C. Contoh Test Case Scenarios

Setelah baris *Step Definition* dibuat, baris tersebut sudah bisa dipakai sebagai langkah yang dilakukan saat *scenario* dijalankan. Pada sebuah *Feature*, terdapat bagian *Background* dan *Scenario*. *Background* merupakan barisan perintah yang akan dijalankan setiap *scenario* akan dilakukan. Biasanya *Background* akan memuat perintah untuk *login* dan navigasi ke *webpage* yang diinginkan, sebelum dijalanannya perintah lain pada *scenario* yang telah dibuat. Berikut merupakan beberapa contoh baris *scenario* yang dibuat pada

Automated Testing System yang dirancang pada pelaksanaan kerja magang.

C.1 Login

Pada *Login Feature*, terdapat dua *scenario*, dimana terdapat *scenario login sukses* (Gambar 3.93) dan *login gagal* dikarenakan *invalid password* (Gambar 3.94).

```
Feature: Login CMS Page

Background:
  Given Saya menghapus semua cookie
  And Saya maximize window
  And Saya membuka halaman "login"

Scenario: Login Sukses
  When Saya memasukkan "testing@oval.id" ke input dengan parameter "Placeholder" "Email"
  And Saya memasukkan "myvalue2021" ke input dengan parameter "Placeholder" "Password"
  And Saya klik tombol bertulisan "Sign in"
  And Saya menunggu selama 15000 miliseconds
  Then Saya mengharapkan linknya akan mengandung "home"
```

Gambar 3.93. *Scenario* untuk *Success Login*

```
Feature: Login CMS Page

Background:
  Given Saya menghapus semua cookie
  And Saya maximize window
  And Saya membuka halaman "login"

Scenario: Login Gagal (Password Salah)
  When Saya memasukkan "testing@oval.id" ke input dengan parameter "Placeholder" "Email"
  Then Saya memasukkan "123" ke input dengan parameter "Placeholder" "Password"
  And Saya klik tombol bertulisan "Sign in"
  Then Saya menunggu selama 2000 miliseconds
  Then Saya mengharapkan ada elemen "div" yang mengandung tulisan "Invalid credentials"
  And Saya menunggu selama 1000 miliseconds
  # prompt untuk salah passsword antara : 'Invalid credentials' or 'Email atau password salah!'
```

Gambar 3.94. *Scenario* untuk *Fail Login*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

C.2 Logout

Pada *Logout Feature scenario* pada Gambar 3.95, hanya ada *positif case* untuk *logout*, dimana *logout* dilakukan dan akan dilakukan pengecekan *webpage* ketika *logout* berhasil.

```
Feature: Logout CMS Page

Background:
  # Login
  Given Saya menghapus semua cookie
  And Saya maximize window
  And Saya membuka halaman "login"
  When Saya memasukkan "testing@oval.id" ke input dengan parameter "Placeholder" "Email"
  And Saya memasukkan "myvalue2021" ke input dengan parameter "Placeholder" "Password"
  And Saya klik tombol bertulisan "Sign in"
  And Saya menunggu selama 15000 milliseconds
  Then Saya mengharapkan linknya akan mengandung "home"

Scenario: Logout Sukses
  When Saya klik nama akun
  Then Saya menunggu selama 1000 milliseconds
  And Saya klik Logout
  Then Saya menunggu selama 3000 milliseconds
  And Saya mengharapkan linknya akan mengandung "login"
```

Gambar 3.95. *Scenario* untuk *Logout*

C.3 Navigations

Berikut merupakan potongan *scenario* untuk melakukan navigasi dari *Menu* untuk mengakses *webpage* tertentu, dengan Gambar 3.96 ini menuju ke *webpage* Pencarian Member pada tab Member Management,

```
Feature: Pencarian Member di Member Management CMS

Background:
  # Login
  Given Saya menghapus semua cookie
  And Saya maximize window
  And Saya membuka halaman "login"
  When Saya memasukkan "testing@oval.id" ke input dengan parameter "Placeholder" "Email"
  And Saya memasukkan "myvalue2021" ke input dengan parameter "Placeholder" "Password"
  And Saya klik tombol bertulisan "Sign in"
  And Saya menunggu selama 15000 milliseconds
  Then Saya mengharapkan linknya akan mengandung "home"

  # Buka Tab dari Member Management
  When Saya klik "Member Management" di navigasi
  And Saya menunggu selama 1000 milliseconds
  And Saya klik "Pencarian Member" di navigasi "Member Management"
  And Saya menunggu selama 1000 milliseconds
  Then Saya mengharapkan linknya akan mengandung "member/search"
```

Gambar 3.96. *Scenario* untuk *Navigation*

C.4 Pencarian Data

Potongan *scenario* pada Gambar 3.97 ini merupakan contoh *scenario* untuk melakukan pencarian data, dengan *parameter* Nama User pada *webpage* Pencarian Member pada tab Member Management.

```
Feature: Pencarian Member di Member Management CMS

Background:
  # Login
  Given Saya menghapus semua cookie
  And Saya maximize window
  And Saya membuka halaman "login"
  When Saya memasukkan "testing@oval.id" ke input dengan parameter "Placeholder" "Email"
  And Saya memasukkan "myvalue2021" ke input dengan parameter "Placeholder" "Password"
  And Saya klik tombol bertulisan "Sign in"
  And Saya menunggu selama 15000 milliseconds
  Then Saya mengharapkan linknya akan mengandung "home"

  # Buka Tab dari Member Management
  When Saya klik "Member Management" di navigasi
  And Saya menunggu selama 1000 milliseconds
  And Saya klik "Pencarian Member" di navigasi "Member Management"
  And Saya menunggu selama 1000 milliseconds
  Then Saya mengharapkan linknya akan mengandung "member/search"

Scenario: Pencarian berdasarkan Value ID / Member ID
  When Saya memasukkan "VAFOBL" ke input dengan parameter "Placeholder" "Cari"
  And Saya menunggu selama 1000 milliseconds
  And Saya klik tombol bertulisan "Cari"
  And Saya menunggu selama 4000 milliseconds
  Then Saya mengharapkan ada elemen "span" yang mengandung tulisan "VAFOBL"
  And Saya menunggu selama 2000 milliseconds
```

Gambar 3.97. *Scenario* untuk Pencarian Data

C.5 Filter Data

Potongan *scenario* pada Gambar 3.98 merupakan contoh *scenario* untuk melakukan *filter* data pada *webpage* Fraud pada Point Management.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```
Scenario: Filter
  When Saya klik tombol bertulisan "Filter"
  And Saya menunggu selama 1000 miliseconds
  And Saya klik dropdown dengan "Div" "Nama Outlet"
  And Saya menunggu selama 1000 miliseconds
  And Saya klik "Hotel Santika Makassar" dari dropdown
  And Saya menunggu selama 1000 miliseconds
  And Saya klik input tanggal "Dari" dengan "Div" "Tanggal Transaksi"
  And Saya menunggu selama 1000 miliseconds
  And Saya memilih bulan "January" di input tanggal
  And Saya menunggu selama 1000 miliseconds
  And Saya memilih tanggal 1 di input tanggal
  And Saya klik input tanggal "Sampai" dengan "Div" "Tanggal Transaksi"
  And Saya menunggu selama 1000 miliseconds
  And Saya memilih tanggal 0 di input tanggal
  And Saya klik tombol bertulisan "Terapkan"
  And Saya menunggu selama 10000 miliseconds
  Then Saya mengharapkan linknya akan mengandung "&dateFrom="
  Then Saya mengharapkan linknya akan mengandung "&dateTo="
  Then Saya mengharapkan linknya akan mengandung "&outletId="
```

Gambar 3.98. Scenario untuk Filter

3.4.4 Implementasi

Sebelum *Automated Testing System* dapat dirancang atau dijalankan, *tester* harus melakukan instalasi *package* terlebih dahulu agar dapat mengolah sistem tersebut. *Package* yang diinstalasi dapat langsung diinstall keseluruhannya ketika *tester* melakukan *download repository* untuk *Automated Testing System* yang ada dengan cara melakukan instalasi melalui terminal dengan menjalankan perintah seperti pada Gambar 3.99.



Gambar 3.99. Langkah instalasi package melalui terminal

Setelah *package* selesai diinstal, *Automated Testing System* dapat dijalankan dengan mengetik serangkaian perintah pada terminal, sesuai dengan *script* yang sudah ditentukan pada *package.json* pada project. Terlihat pada Gambar 3.100, pada pengetesan ini, *script* diberi nama 'testing' dan memuat rangkaian *setting* untuk *environment* dari *testing*. Pada *script testing*, diberikan pengaturan dimana *browser* yang digunakan merupakan Chrome, dengan menggunakan konfigurasi dari *Cucumber BDD* dan perintah yang diatur pada bagian *Step Definitions*. Selain itu, dapat juga dibuat pengaturan untuk menampilkan hasil laporan dari *testing*.

Pada saat menuliskan perintah menjalankan *testing*, *tester* dapat menentukan *Feature file* apa saja yang ingin dijalankan secara spesifik ataupun berkelompok dengan merefer ke *folder* dan *file* pada *project* setelah *script* dimasukkan. Hal ini dilakukan dengan tujuan baik untuk melakukan testing fitur secara spesifik ataupun melakukan testing pada suatu tab atau keseluruhan secara langsung.

```
PS D:\Aisia\Kuliah\Magang\myvalue-cms\myvalue-cms\integration_test> npm run testing features/member-management/member-search.feature
> myvalue-cms@0.0.1 testing D:\Aisia\Kuliah\Magang\myvalue-cms\myvalue-cms\integration_test
> cross-env NIGHTWATCH_ENV=chrome cucumber-js --require-module @babel/register --require cucumber.conf.js --require step-definitions/**/*.js --format
@ cucumber/pretty-formatter "features/member-management/member-search.feature"

\ Connecting to localhost on port 4444...
- Connecting to localhost on port 4444...
! Connected to localhost on port 4444 (6909ms).
Using: chrome (96.0.4664.45) on Windows platform.
```

Gambar 3.100. Langkah menjalankan *testing* melalui terminal

Begitu perintah dijalankan, *browser* Chrome akan terbuka dan secara otomatis menjalankan perintah yang ada. *Tester* hanya perlu mengamati perilaku yang dilakukan sistem dan dapat melakukan modifikasi jika diperlukan pada *file* saat *testing* selesai dilakukan. Gambar 3.101 dibawah merupakan contoh proses yang berjalan saat dilakukan *scenario testing*.

```
Feature: Pencarian Member di Member Management CMS # features/member-management/member-search.feature:1
Scenario: Pencarian berdasarkan Value ID / Member ID # features/member-management/member-search.feature:22
Given Saya menghapus semua cookie
And Saya maximize window
And Saya membuka halaman "login"
When Saya memasukkan "testing@oval.id" ke input dengan parameter "Placeholder" "Email"
✓ Expected element <//*[contains(@placeholder, 'Email') and (self::input or self::textarea or self::p)]> to be present (33ms)
And Saya memasukkan "myvalue2021" ke input dengan parameter "Placeholder" "Password"
✓ Expected element <//*[contains(@placeholder, 'Password') and (self::input or self::textarea or self::p)]> to be present (13ms)
And Saya klik tombol bertulisan "Sign in"
✓ Expected element <button[contains(text(), 'Sign in') or .//*[contains(text(), 'Sign in')]]> to be present (13ms)
And Saya menunggu selama 15000 milliseconds
Then Saya mengharapkan linknya akan mengandung "home"
✓ Expected current url to contain: "home" in 4000ms (7ms)
```

Gambar 3.101. Contoh pengamatan melalui terminal saat *testing* dijalankan

Hasil *testing* seperti pada Gambar 3.102 akan ditampilkan pada terminal setelah *testing* sukses dijalankan dengan melaporkan jika *scenario* berhasil, ataupun jika ada *error* atau kesalahan penulisan saat *scenario* dirancang.

```
1 scenario (1 passed)
19 steps (19 passed)
0m33.421s (executing steps: 0m31.305s)
```

Gambar 3.102. Hasil setelah *testing* dilakukan melalui terminal

3.5 Kendala dan Solusi yang Ditemukan

Adapun kendala yang ditemukan selama pelaksanaan magang adalah sebagai berikut:

- Dikarenakan keterbatasan pengetahuan mengenai Vue JS, Nightwatch JS, dan Cucumber BDD saat pelaksanaan kerja magang berlangsung, dibutuhkan waktu lebih untuk mempelajari dan memahami penggunaannya.
- Penulis masih belum *familiar* dengan beberapa *software* dan tugas ataupun istilah yang digunakan selama pelaksanaan kerja magang sehingga pada awalnya ada beberapa bagian tugas yang masih ditemukan masalah baik secara teknis maupun dari miskomunikasi.

Sedangkan solusi yang dilakukan atas kendala yang ditemukan adalah sebagai berikut:

- Diadakannya periode waktu pembelajaran yang dilakukan di beberapa bulan pertama pelaksanaan kerja magang yang dibantu oleh supervisi lapangan sehingga penulis dapat memahami dasar-dasar dari vue JS, Nightwatch JS, dan Cucumber BDD untuk memulai membuat *testing system* yang diperlukan. Selain itu, bimbingan dan masukan masih sering diberikan oleh anggota tim untuk meningkatkan kualitas hasil *testing system* yang dibuat.
- Penulis dan supervisi lapangan melakukan *tutorial* singkat untuk aplikasi yang akan digunakan, serta tanya-jawab mengenai penggunaan ataupun pelaksanaan tugas juga dapat dilakukan sehingga penulis dapat memahami dengan lebih jelas.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A