



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Kegiatan kerja magang di PT. Viva Velove Viktori sebagai *User Interface Designer* diberikan tanggung jawab untuk membuat desain dari aplikasi *mobile* dan *website*. Di dalam kegiatan kerja magang, sudah memperoleh banyak ilmu yang sangat berguna di dunia IT terutama pada perancangan desain aplikasi *mobile* dan *website*, implementasi *payment gateway* dan juga pembuatan *website* menggunakan Laravel. Dengan berkoordinasi dengan tim IT yang terkait dalam pengembangan aplikasi dan *website* seperti dengan Pak Arif dan juga dibantu oleh Pak Reynaldi selaku pembimbing yang selalu memandu pada saat pengerjaan tugas.

3.2. Tugas yang dilakukan

Pada awal kegiatan magang dilaksanakan yaitu minggu pertama dan minggu kedua, diberikan tugas untuk melakukan pembuatan *SWOT Analysis* untuk projek pertama yang bernama Wellness Center. Wellness Center merupakan aplikasi yang menyediakan fitur konsultasi *online* menggunakan *video call* dan langsung terhubung dengan dokter yang dituju. Untuk projek Wellness Center, pembuatan diagram UML seperti *swimlane diagram* dan *use case diagram* merupakan tahap awal yang akan dilakukan sebelum melakukan pembuatan mockup pada aplikasi Wellness Center

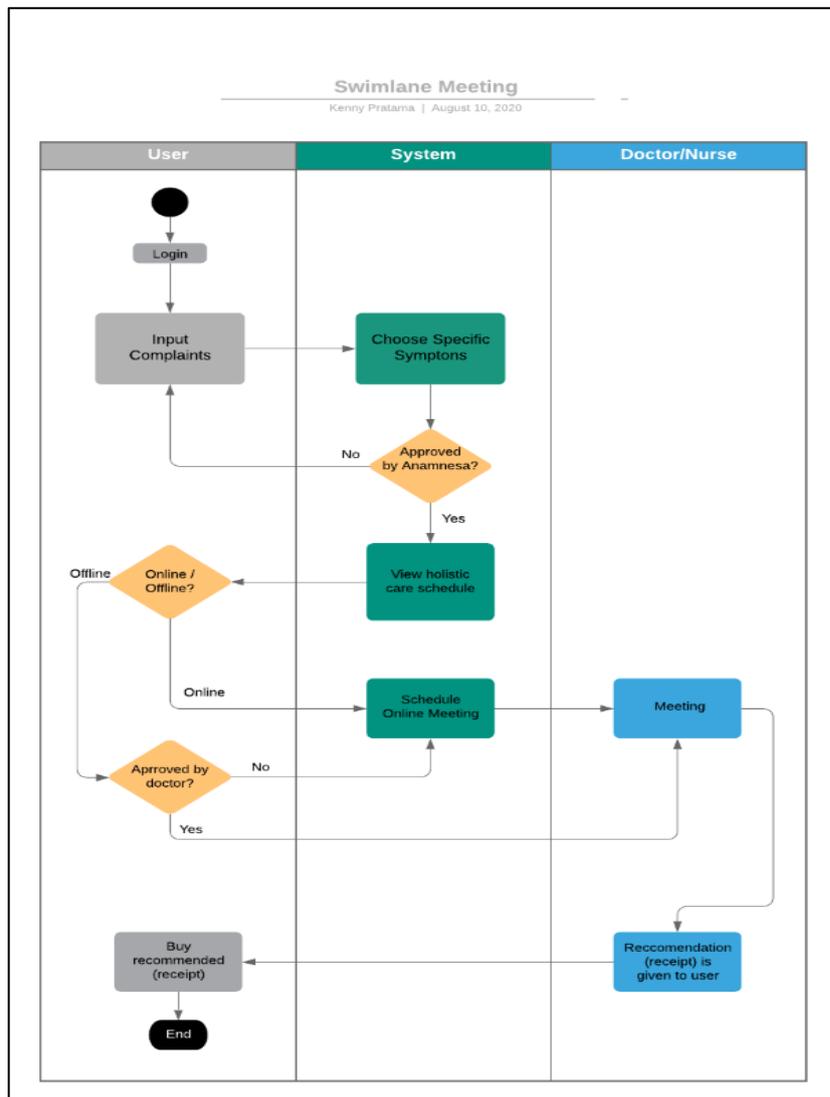
Swimlane Diagram dibuat menggunakan LucidChart. Pada saat pengguna masuk ke dalam aplikasi setelah melewati tahap *login*, pengguna dapat menjalankan fungsi *input complaints*, setelah itu dapat memilih gejala yang dirasakan.

Setelah memasukkan gejala yang dirasakan oleh pengguna, akan langsung dilakukan pengecekan oleh sistem dan pihak terkait yang dimana jika tidak memenuhi persyaratan atau tidak disetujui oleh pihak terakit, maka pengguna diharuskan untuk mengulangi tahap *input complaints* kembali. Jika sudah disetujui, maka pengguna dapat melihat jadwal yang tersedia, lalu melakukan pemilihan konsultasi akan dilakukan dengan tatap muka atau dengan metode *online*.

Jika pengguna memilih metode tatap muka, maka harus disetujui oleh dokter terkait. Pada saat dokter tidak dapat menyetujui permintaan tersebut, maka akan otomatis dijadikan sebagai pertemuan konsultasi secara *online*. Setelah proses penjadwalan konsultasi dengan metode *online*, maka *meeting* antara pengguna/pasien dan dokter pun dapat dilaksanakan.

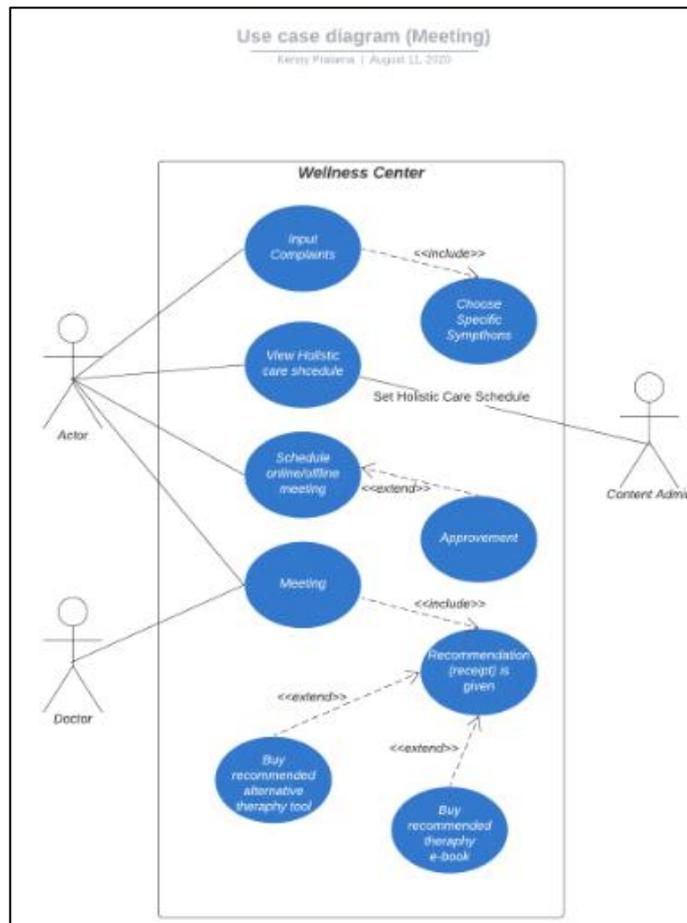
Setelah konsultasi sudah selesai dilakukan, maka pasien/pengguna langsung mendapat notifikasi di dalam aplikasi Wellness Center tentang rekomendasi resep obat yang langsung diberikan sesuai dengan penyakit atau gejala yang dialami.

Swimlane Diagram merupakan diagram alir yang terdapat proses kerja yang berjalan disertai dengan proses kerja yang dilakukan sehingga alur yang berjalan dapat terlihat dengan jelas dan apabila terjadi kesalahan dalam proses, maka langsung diketahui akar permasalahannya (Nirmalasari & Handayani, 2017). Gambar 3.1 merupakan hasil *swimlane diagram* yang dibuat untuk aplikasi Wellness Center.



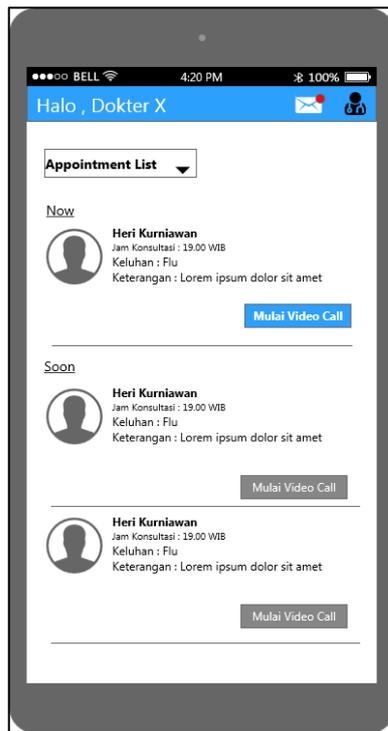
Gambar 3.1. *Swimlane Diagram* aplikasi Wellness Center

Selanjutnya dilakukan dengan pembuatan *use case diagram*. *Use case* merupakan teknik yang dapat digunakan dalam sebuah pengembangan aplikasi untuk memperoleh kebutuhan fungsional dari sistem / aplikasi yang akan dibuat (Komputer et al., 2018). Di dalam pembuatan *use case diagram*, terdapat 3 aktor yang terdiri dari pengguna, dokter, dan admin. Dimana pengguna dapat melakukan *input complaints*, pemilihan gejala, melihat penjadwalan, menjadwalkan pertemuan tatap muka dan *online*. Aktor dokter dapat menjalankan fungsi “meeting” di dalam aplikasi. Dan aktor yang berperan cukup penting yaitu Admin, dimana admin yang memasukkan jadwal konsultasi ke dalam aplikasi, serta melakukan pengawasan terhadap keseluruhan proses yang berjalan di dalam aplikasi Wellness Center. Gambar 3.2. merupakan hasil *use case diagram*



Gambar 3.2. Use case Diagram aplikasi Wellness Center

Selanjutnya dilanjutkan untuk pembuatan *mockup* Wellness Center pada *actor* dokter yang terdiri dari 4 halaman. Halaman pertama yaitu halaman utama dokter yang menampilkan *appointment list* yang sedang berlangsung, dan juga jadwal konsultasi yang akan datang. Pada halaman ini, dokter dapat langsung mengklik tombol “Mulai Video Call” agar dapat melakukan panggilan langsung ke pengguna/pasien yang sudah melakukan pembayaran dan pemilihan dokter. Gambar 3.3 merupakan *mockup* pada halaman utama dokter.

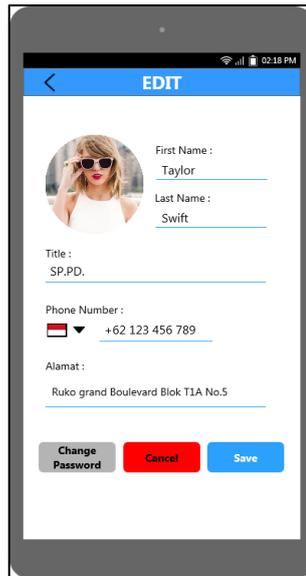


Gambar 3.3. Mockup Halaman utama Dokter

Halaman kedua yaitu halaman *edit profile* yang dapat diakses langsung oleh Dokter di dalam aplikasi Wellness Center. Didalamnya terdapat beberapa hal seperti perubahan:

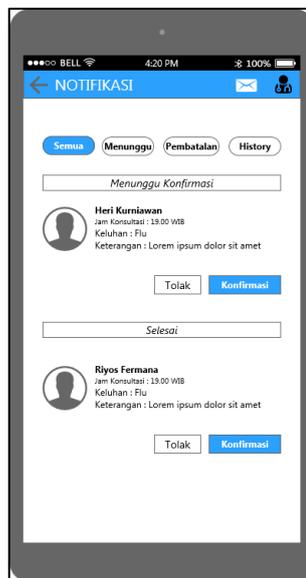
1. Nama depan dan nama belakang Dokter
2. *Title* yang dimiliki oleh Dokter
3. Nomor telepon aktif Dokter
4. Alamat Dokter

Di dalam halaman ini, dokter juga dapat mengubah *password* untuk memasuki aplikasi Wellness Center. Gambar 3.4 merupakan halaman *edit profile* :



Gambar 3.4. Halaman *Edit Profile* Dokter

Lalu untuk halaman notifikasi, terdapat beberapa pilihan ketika pengguna ingin melakukan konsultasi. Pilihan tersebut antara lain, pengguna dapat melihat notifikasi keseluruhan, notifikasi menunggu pasien, notifikasi pembatalan konsultasi yang dilakukan dengan pasien. halaman *history* konsultasi yang sudah berhasil dilakukan dengan pasien terakrit. Di halaman notifikasi ini, dokter juga dapat melihat *history* yang berisi konsultasi yang sudah berlalu. Gambar 3.5 merupakan halaman notifikasi :



Gambar 3.5. Halaman Notifikasi Dokter

Pada saat dokter menekan tombol “Konsultasi” maka akan langsung diarahkan ke halaman *video call* seperti yang tertera di Gambar 3.6, terdapat tombol “Akhiri Video Call” yang dapat ditekan ketika konsultasi/kegiatan yang berlangsung antara dokter dan pasien sudah selesai.



Gambar 3.6. Halaman Video Call Dokter

Dan selanjutnya, dilakukan pembuatan *mockup* pada *actor user phase 2* pada aplikasi Wellness Center yang pertama yaitu pembuatan halaman e-book, dimana *user* dapat melihat artikel/buku yang relevan dengan kesehatan. Gambar 3.7 merupakan hasil dari tampilan *mockup* halaman e-book.



Gambar 3.7. Halaman e-book Wellness Center

Lalu selanjutnya pada Gambar 3.8 terdapat halaman darurat yang dapat diakses oleh *user*, fungsi di dalam halaman ini yaitu *user* dapat melakukan panggilan darurat seperti pemanggilan ambulans.

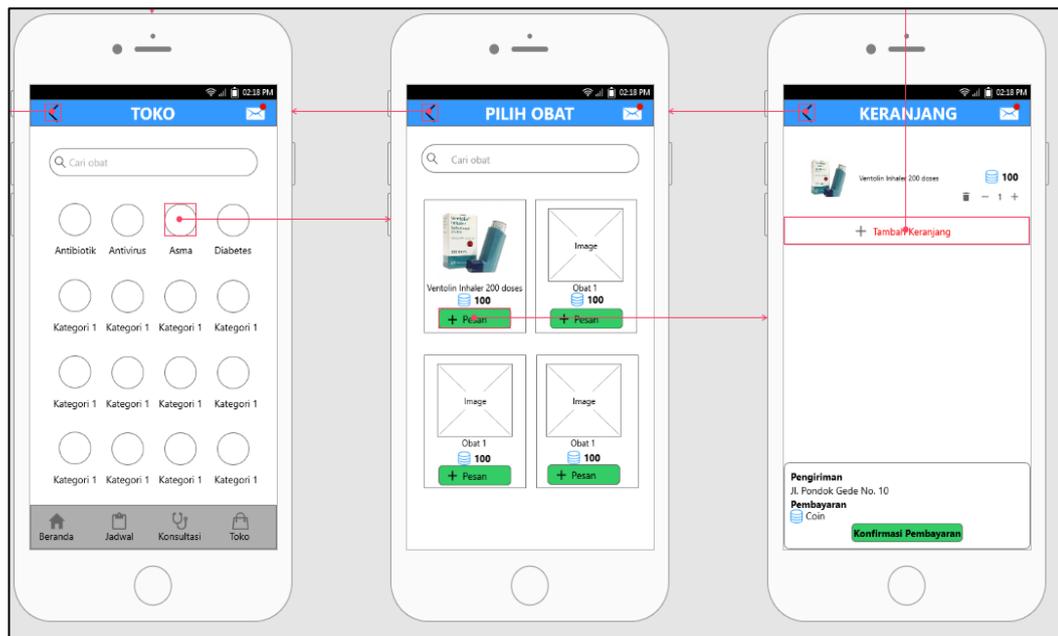


Gambar 3.8. Halaman darurat

Selanjutnya untuk halaman *user phase 2* yang terakhir yaitu halaman toko yang terdiri dari 3 halaman. Pada halaman utama ditampilkan beberapa kategori penyakit di dalam aplikasi. Pengguna juga dapat melakukan pencarian penyakit di kolom pencarian. Setelah pengguna memilih obat yang diinginkan, akan tampil halaman kedua yaitu halaman *detail* yang sudah berisi obat-obat berdasarkan penyakit yang sudah dipilih di halaman sebelumnya.

Setelah pengguna memilih produk obat, maka akan diarahkan langsung ke halaman “Keranjang” berisi obat yang sudah dipesan oleh pengguna. Di halaman ini, terdapat penjelasan *detail* seperti alamat pengiriman, metode pembayaran,

kuantitas produk, gambar produk, nama produk dan jumlah harga yang harus dibayar oleh pengguna. Gambar 3.9 dibawah ini merupakan salah satu contoh ketika pengguna memilih kategori penyakit “Asma” dan memilih salah satu obat berdasarkan kategori penyakit.



Gambar 3.9. Actor User Phase

Selanjutnya pada minggu ke 3 dan ke 4, diberikan tugas untuk mencoba mengimplementasikan *payment method* dengan cara mengambil API dari Midtrans menggunakan Flutter dan dikerjakan di Android Studio yang akan diterapkan di aplikasi Wellness Center. Untuk dapat mengambil data dari Midtrans, maka diperlukan untuk mengambil *code* Client Key yang terdapat pada website Midtrans setelah melakukan pembuatan akun. Client Key terdapat pada bagian *Settings > Access Keys*. Setelah itu *code* tersebut dapat dimasukkan kedalam Android Studio. Untuk *Hosting* menggunakan penyedia layanan *hosting* gratis yaitu Heroku. Gambar 3.10 berikut merupakan *code* yang digunakan untuk dapat mengambil data dari Midtrans.

```
void initState() {  
  super.initState();  
  flutrans.init("SB-Mid-client-3-40UGVH14m975", "https://payment1401.herokuapp.com/index.php/");  
  flutrans.setFinishCallback(_callback);  
}
```

Gambar 3.10. Code Client Key Midtrans

Gambar 3.11 dibawah ini merupakan potongan *code* untuk menjalankan fungsi pembayaran seperti harga, *item*, informasi dari pembeli (nama depan, nama belakang, email, nomor telepon)

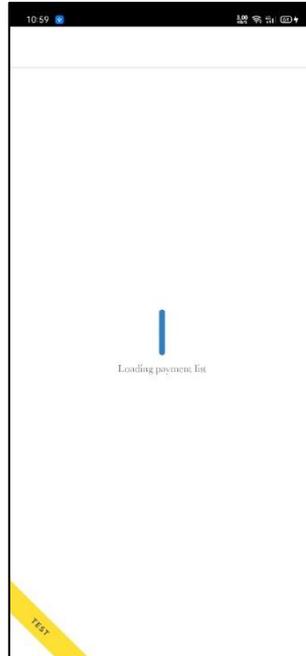
```
_makePayment() {
  setState(() {
    isMakePayment = true;
  });
  flutrans
    .makePayment(
      MidtransTransaction(
        15000,
        MidtransCustomer(
          "ken", "Pra", "kennypratama@gmail.com", "081234567890"),
        [
          MidtransItem(
            "BP1",
            7500,
            2,
            "Bakso Paket 1",
          )
        ],
        skipCustomer: true,
        // customField1: "ANYCUSTOMFIELD"
      ),
    )
    .catchError((err) => print("ERROR $err"));
}
```

Gambar 3.11. Code untuk fungsi pembayaran

Selanjutnya untuk tampilan pada aplikasi pembayaran, akan ada tombol “Make Payment” yang merupakan tombol untuk untuk menjalankan fungsi *payment method* dan mengambil API dari Midtrans.

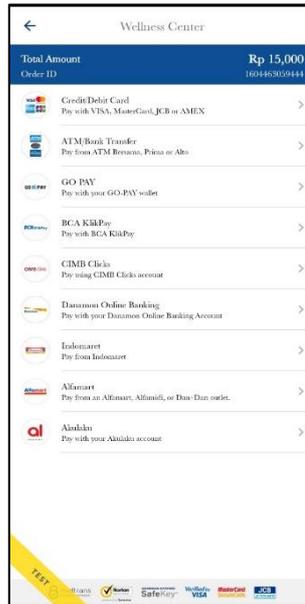
Setelah tombol “Make Payment” di klik, maka akan muncul *progress bar* yang berarti aplikasi sedang mengambil data dan menyambungkan ke server Midtrans. Estimasi waktu yang diperlukan untuk menghubungkan ke server

Midtrans setelah dilakukan uji coba yaitu sekitar 5-10 detik. Gambar 3.12 merupakan halaman *progress bar* yang dimaksud.



Gambar 3.12. Progress Bar

Setelah menunggu pengambilan data dan jika berhasil, maka akan muncul halaman detail yang berisi metode pembayaran dan jumlah pembayaran. Jika halaman ini muncul, berarti sudah berhasil mengambil API dari Midtrans. Gambar 3.13 merupakan hasil dari implementasi *payment method* menggunakan API Midtrans.



Gambar 3.13. Payment Method Flutter

Jika pengguna memilih salah satu metode pembayaran, maka akan muncul halaman seperti Gambar 3.14 dibawah ini. Halaman ini berisi instruksi yang harus dilakukan ketika ingin melakukan pembayaran melalui Go-Pay.



Gambar 3.14. Contoh halaman detail metode pembayaran Go-Pay

Ketika pengguna sudah membaca instruksi yang tertera, dan menekan tombol “Pay Now With Gopay”, maka akan muncul halaman konfirmasi seperti Gambar 3.15 dibawah ini. Di halaman ini terdapat hitungan mundur yang mengharuskan pengguna untuk menyelesaikan pembayaran. Terdapat *barcode* beserta tulisan Wellness Center untuk memberikan informasi tambahan bahwa dana yang dibayar akan langsung masuk ke dalam rekening Wellness Center.



Gambar 3.15. Halaman Konfirmasi Go-Pay

Pada minggu ke-4 dan ke-5, diberikan projek selanjutnya yaitu dengan nama SIAP. Aplikasi SIAP ini menyediakan beberapa fitur yang dapat memudahkan penggunaannya, fiturnya antara lain seperti :

1. E-Complaint = Pengguna dapat melakukan pelaporan secara *online*.
2. E-Rent = Pengguna dapat mencari lokasi dan dapat menyewa berbagai alat transportasi seperti sepeda, *scooter*, dan lain-lain.
3. E-News = Pengguna dapat melihat berita / *event* yang sedang diselenggarakan disekitarnya.
4. E-Call = Pengguna dapat melakukan *voice call*, *video call*, dan berkomunikasi langsung melalui aplikasi SIAP.
5. E-Parking = Pengguna dapat melakukan pembayaran parkir melalui aplikasi dengan cara menunjukkan *QR Code*.
6. E-Billing = Pengguna dapat melihat tagihan bulanan terbaru seperti tagihan IPL.

Tahap awal dalam pembuatan *mockup* aplikasi SIAP ini yaitu dengan melakukan pembuatan *use case diagram* tertera pada Gambar 3.16. Didalamnya terdapat 2 aktor dengan fungsi sebagai berikut :

1. User (*Logged In*) : Melihat profil, menggunakan ke-6 fitur yang ada di aplikasi SIAP ini.
2. Admin : Melakukan pengawasan terhadap aplikasi, melihat Meter Reading pada saat pengguna menggunakan fitur E-Billing, dan juga

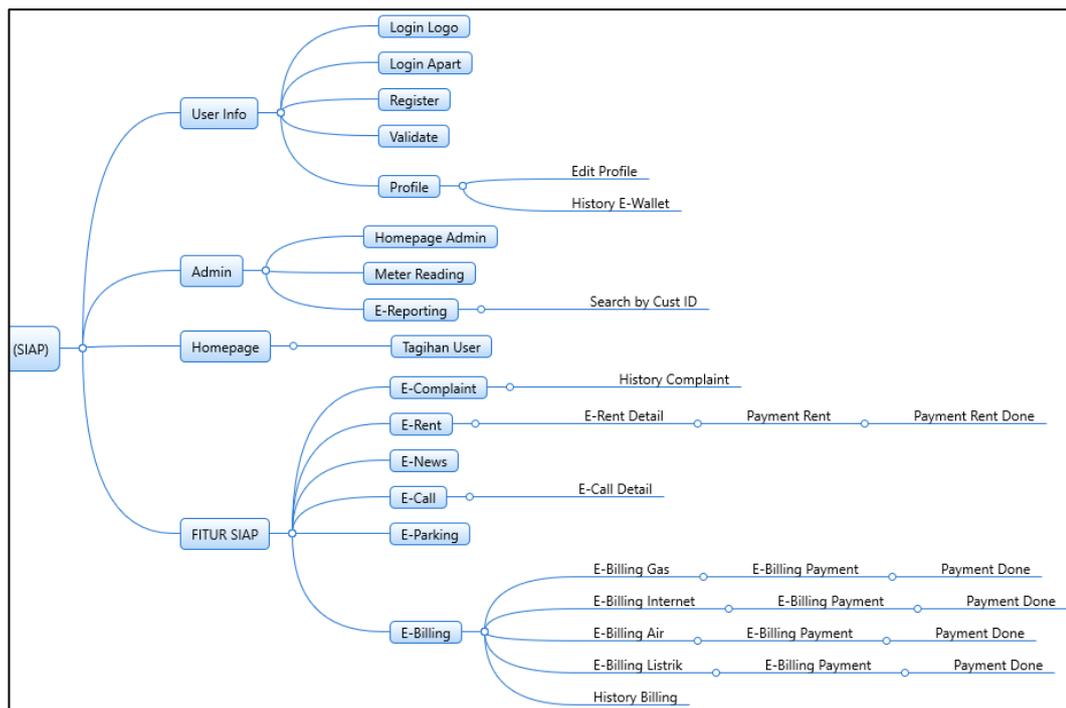
menjalankan fungsi E-Reporting yang berisi tagihan yang sudah dibayar pengguna (tanggal, metode pembayaran, dll)



Gambar 3.16. Use Case Diagram aplikasi SIAP

Untuk selanjutnya, dilanjutkan dengan pembuatan *mindmap* sehingga dapat memperjelas rincian fungsi yang terdapat dalam aplikasi . Gambar 3.17 dibawah ini merupakan *mindmap* dari pembuatan *mockup* aplikasi *mobile*. Didalam aplikasi *mobile* ini terdapat 4 bagian utama yaitu :

1. User Info yang terdiri dari halaman *login*, *register*, validasi dan *profile*.
2. Admin yang terdiri dari halaman *homepage* admin, meter *reading*, *E-reporting*.
3. Homepage terdiri dari beberapa *list*, dan tagihan pengguna.
4. Fitur SIAP

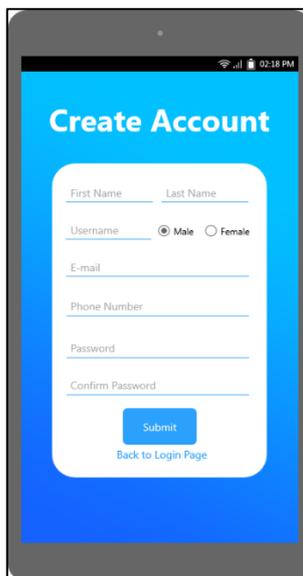


Gambar 3.17. Mindmap aplikasi SIAP

Selanjutnya, langsung dibuatkan *mockup* aplikasi SIAP yang dimulai dari halaman *login* dan *register* yang terdapat pada Gambar 3.18 dan Gambar 3.19.



Gambar 3.18. Halaman Login aplikasi SIAP



Gambar 3.19. Halaman *register* aplikasi SIAP

Saat pengguna sudah membuat akun dan masuk ke dalam aplikasi, terdapat halaman utama yang terdiri dari beberapa bagian. Pada bagian atas terdapat penjelasan mengenai tagihan yang harus segera dibayar oleh pengguna dan juga terdapat rinciannya. Tampilan tersebut akan hilang ketika pengguna melunasi semua tagihan pada bulan itu. Di tampilan *home*, terdapat fitur utama dari aplikasi SIAP. Pada bagian bawah, terdapat tampilan E-News yang menampilkan berita atau *events* yang sedang berlangsung. Gambar 3.20 menunjukkan tampilan *mockup* pada halaman *home* aplikasi SIAP.



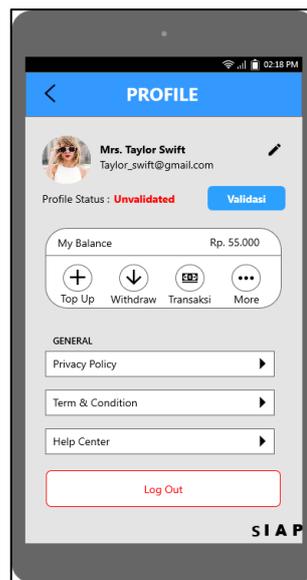
Gambar 3.20. Tampilan *home* aplikasi SIAP

Pada saat pengguna melihat halaman *profile*, terdapat beberapa hal di dalamnya yaitu:

1. Profile Status : Pada bagian ini, terdapat tulisan jika pengguna belum melakukan validasi akun seperti memasukkan foto KTP, alamat, nomor rekening dan lain-lain.

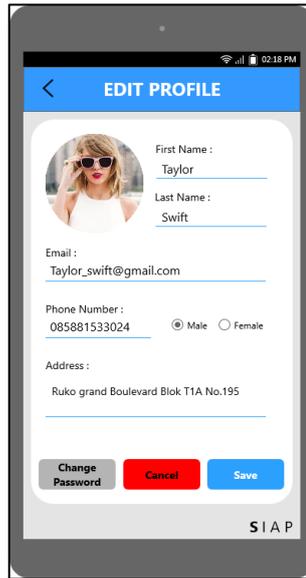
2. Balance : Keterangan mengenai sisa saldo, serta tombol untuk melakukan penambahan saldo, penarikan saldo, melihat *history* dari transaksi, dll.
3. General : Keterangan tambahan mengenai *term&condition*, *help center*, dll
4. Edit Profile : Berisi beberapa data yang bisa diubah seperti nama depan, nama belakang, *email*, nomor telepon, jenis kelamin, dan alamat pengguna.

Gambar 3.21 dibawah ini merupakan tampilan halaman *profile* dari pengguna aplikasi SIAP.



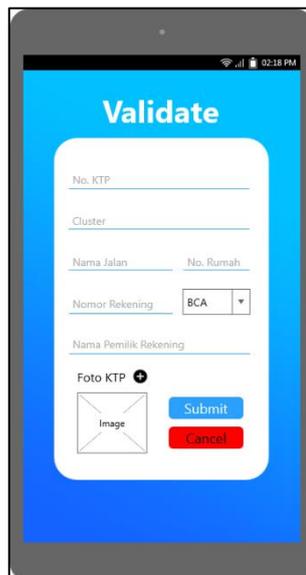
Gambar 3.21. Halaman *profile* aplikasi SIAP

Gambar 3.22 dibawah ini merupakan halaman yang dapat digunakan pengguna aplikasi SIAP untuk mengubah/menambahkan data diri pengguna.



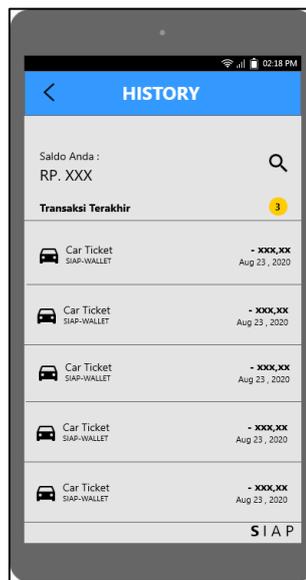
Gambar 3.22. Halaman *Edit Profile* aplikasi SIAP

Pada Gambar 3.23 tertera halaman validasi yang dapat digunakan ketika pengguna belum memvalidasi akun. Halaman validasi terdiri dari pengisian data seperti nomor KTP, alamat rumah, nomor rekening, foto KTP, dll.



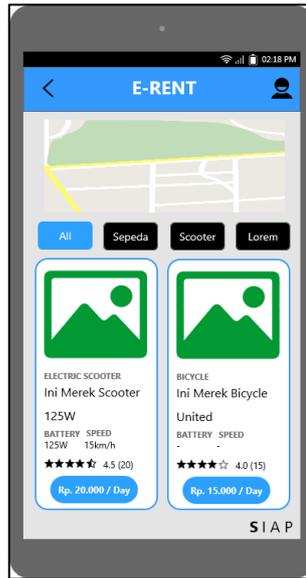
Gambar 3.23. Halaman validasi akun aplikasi SIAP

Lalu, pada saat pengguna menekan tombol “Transaksi”, maka halaman *history* dari transaksi pengguna akan muncul yang terdiri dari sisa saldo, penjelasan mengenai transaksi yang telah dilakukan oleh pengguna seperti tiket mobil, tanggal transaksi, harga tiket, dll. Gambar 3.24 merupakan halaman *history* saat pengguna ingin melihat transaksi yang sudah pernah dilakukan.



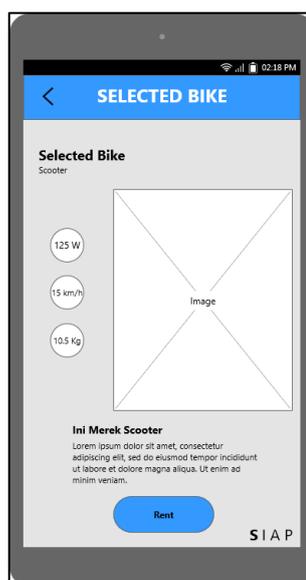
Gambar 3.24. Halaman *history* aplikasi SIAP

Salah satu fitur yaitu E-Rent, dimana pengguna dapat menyewa berbagai alat transportasi seperti sepeda, dan *scooter* terdapat 4 halaman *mockup* di dalam fitur E-Rent ini. Halaman utama yang berisi gambar sepeda / *scooter* yang dapat disewa serta penjelasan tambahan mengenai nama, jenis, rating, dan harga kendaraan tertera pada Gambar 3.25.



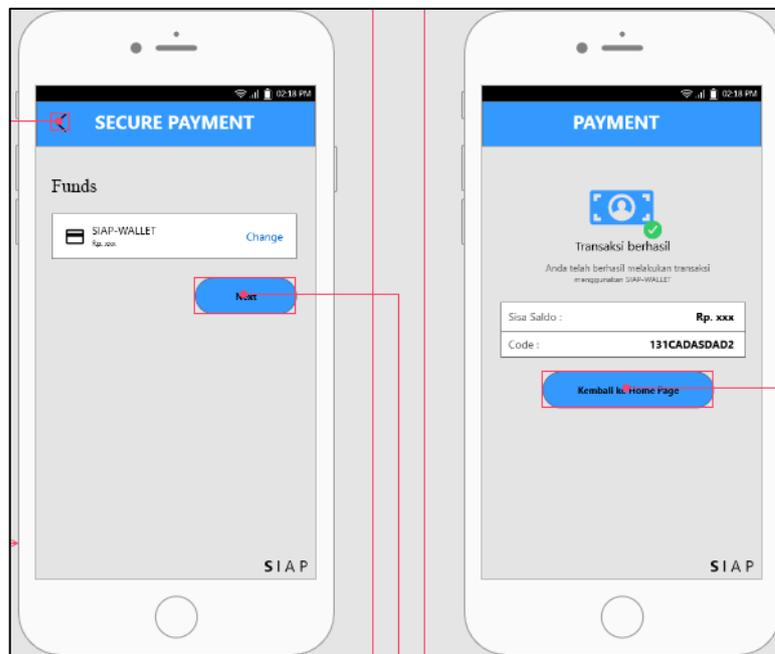
Gambar 3.25. Halaman utama dalam fitur E-Rent aplikasi SIAP

Halaman kedua menunjukkan halaman detail dari sepeda / scooter yang dipilih oleh pengguna. Beberapa hal yang dapat dilihat di halaman ini yaitu seperti kecepatan kendaraan, daya kendaraan, dan berat kendaraan. Tertera pada Gambar 3.26.



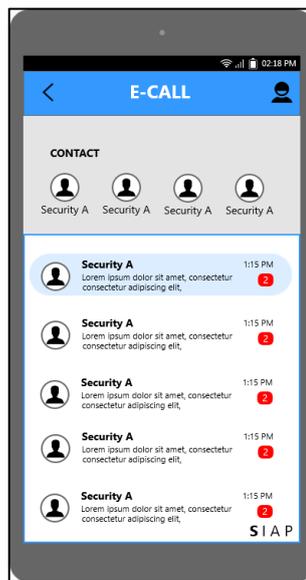
Gambar 3.26. Halaman kendaraan E-Rent aplikasi SIAP

Setelah pengguna menekan tombol “Rent”, maka akan diarahkan ke bagian pembayaran. Terdapat pilihan untuk metode pembayaran yang ingin digunakan. Halaman terakhir merupakan informasi setelah pembayaran berhasil dilakukan oleh pengguna yang juga terdiri dari sisa saldo pengguna, kode pembayaran dan informasi bahwa pembayaran sudah berhasil. Gambar 3.27 merupakan tampilan pembayaran pada fitur E-Rent.



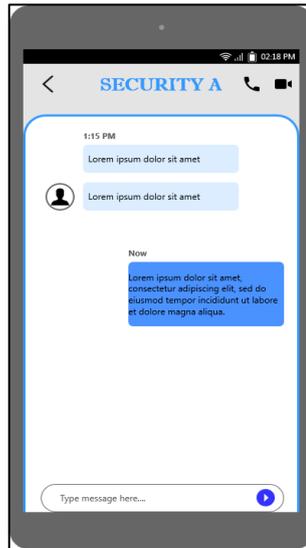
Gambar 3.27. Halaman Pembayaran dari fitur E-Rent

Fitur dari aplikasi SIAP lainnya yaitu E-Call yang dimana pengguna dapat melakukan *voice call*, *video call*, dan berkomunikasi langsung melalui aplikasi SIAP. Gambar 3.28 merupakan tampilan utama dari halaman E-Call yang berisi seperti aplikasi *instant messenger*. Terdiri dari *list chat* dari penjaga apartemen yang berada di apartemen pengguna.



Gambar 3.28. Tampilan halaman utama E-Call aplikasi SIAP

Setelah pengguna menekan salah satu kontak yang berada di halaman utama E-Call, maka akan langsung diarahkan ke halaman *chat* salah satu penjaga apartemen. Di dalam halaman ini terdapat beberapa fitur yaitu pengguna dapat langsung berkomunikasi dengan penjaga apartemen, melakukan *video call*, atau panggilan masuk. Gambar 3.29 merupakan halaman *chat* pada fitur E-Call.



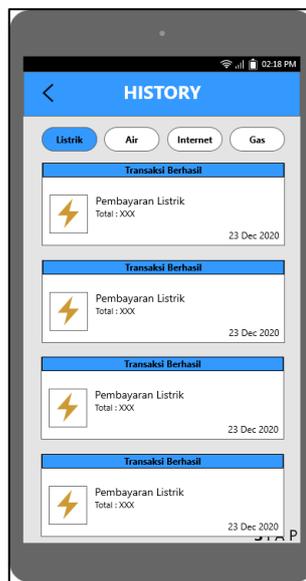
Gambar 3.29. Halaman chat E-Call aplikasi SIAP

Fitur lainnya pada aplikasi SIAP yaitu E-Billing, fitur ini memberikan kemudahan bagi pengguna aplikasi SIAP saat ingin melakukan pembayaran tagihan bulan terbaru. Pembayaran tagihan terdiri dari pembayaran tagihan air, listrik, internet dan gas. Gambar 3.30 merupakan halaman utama dari fitur E-Billing aplikasi SIAP.



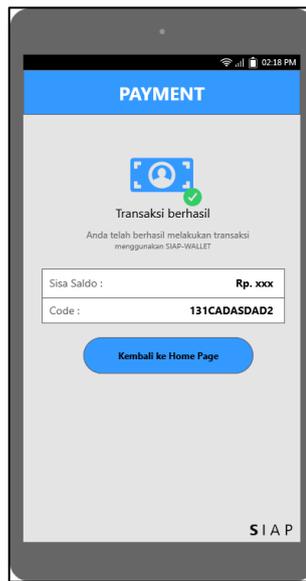
Gambar 3.30. Halaman utama E-Billing aplikasi SIAP

Pada halaman utama E-Billing, pengguna juga dapat melihat *history* transaksi pembayaran tagihan yang sudah pernah dilakukan oleh pengguna. Beberapa hal yang ada di halaman ini terdiri dari pemilihan kategori produk (Listrik, Air, Internet, Gas), informasi mengenai tanggal pembayaran transaksi, dll. Gambar 3.31 merupakan halaman *history* pada transaksi yang sudah dilakukan pengguna.



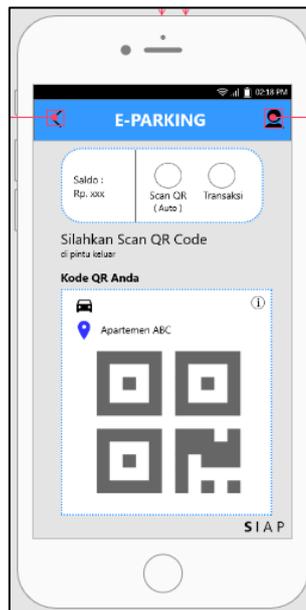
Gambar 3.31. Halaman *history* transaksi E-Billing aplikasi SIAP

Pada saat pengguna melakukan salah satu pembayaran tagihan bulanan, dan jika berhasil maka akan ditampilkan halaman pembayaran yang sudah sukses. Di dalam halaman ini, pengguna dapat melihat sisa saldo, kode pembayaran, dll. Gambar 3.32 merupakan halaman ketika pengguna sukses melakukan pembayaran tagihan bulanan.



Gambar 3.32. Halaman pembayaran berhasil E-Billing aplikasi SIAP

Fitur selanjutnya pada aplikasi SIAP yang tertera pada Gambar 3.33 yaitu E-Parking. Fitur ini memudahkan pengguna ketika ingin membayar parkir, dengan menggunakan *QR Code*, pengguna dapat langsung membayar parkir. Di halaman ini, terdapat beberapa menu yaitu sisa saldo, *scan QR*, melihat *history* transaksi, dan dapat melihat *QR Code* dari pengguna.

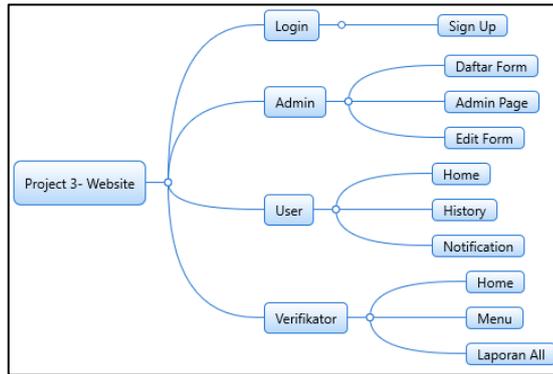


Gambar 3.33. Halaman E-Parking aplikasi SIAP

Projek yang ke-3, pembuatan *mockup* pada projek AKSAN (Aplikasi Analisis Penilaian) berbasis *website* via *desktop*. Projek ke-3 ini diberikan pada minggu ke-6. Aplikasi AKSAN merupakan aplikasi yang dapat mengakomodir para pengguna aplikasi untuk dapat melakukan *submit* laporan/kegiatan, setelah itu aktor Verifikator akan menilai laporan/kegiatan yang diberikan oleh pengguna. Untuk *mindmap* dari aplikasi ini tertera pada Gambar 3.34. Di dalam *mindmap* dapat dilihat bahwa ada 3 aktor yang berperan di dalam aplikasi, antara lain :

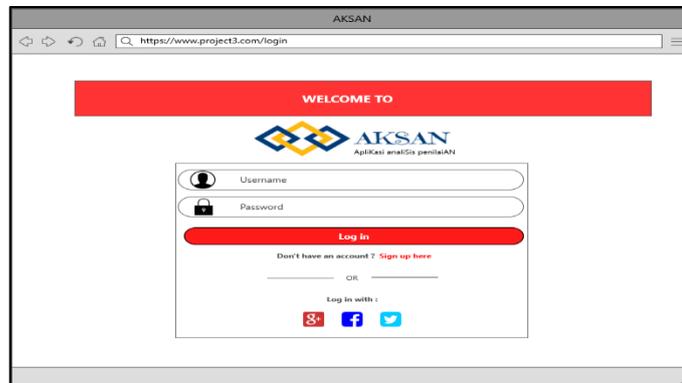
1. Admin : dapat mengakses halaman “Admin Page”, dan “Edit Form”.
2. User : dapat mengakses halaman *home*, *history*, *notification*.
3. Verifikator : dapat mengakses halaman *home*, *menu*, keseluruhan laporan.

Gambar 3.34 merupakan hasil *mindmap* pada aplikasi AKSAN.

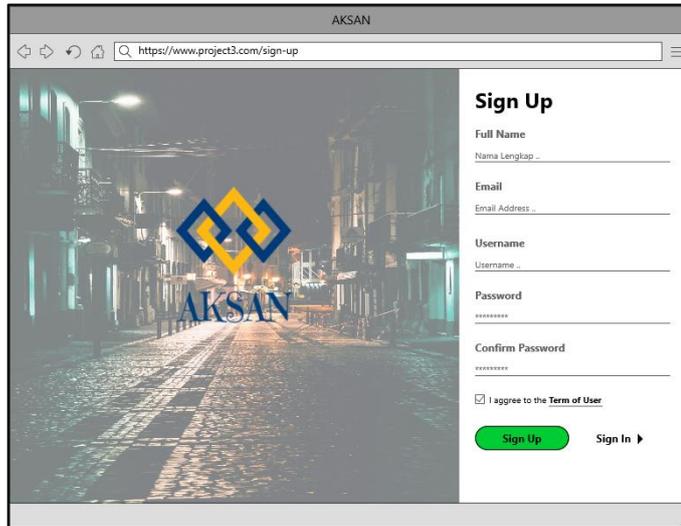


Gambar 3.34. Mindmap aplikasi AKSAN

Selanjutnya, dilakukan pembuatan *mockup* untuk halaman *login* dan *register* pada *mockup website* proyek AKSAN yang dibuat tertera pada Gambar 3.35 dan Gambar 3.36.

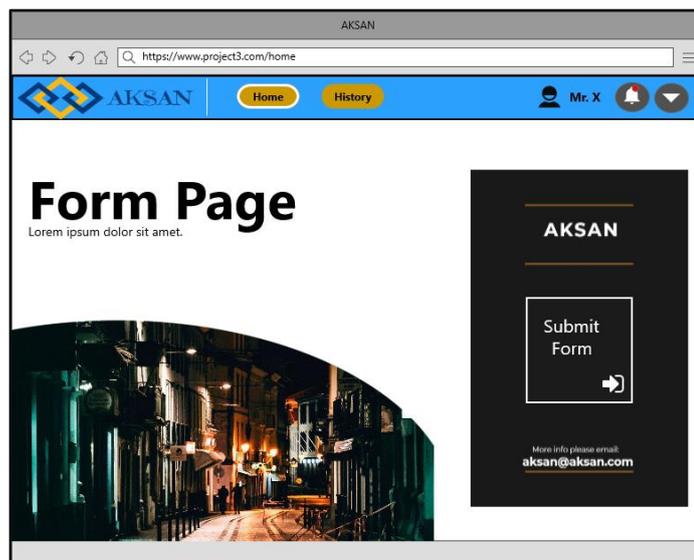


Gambar 3.35. Halaman *Login* AKSAN



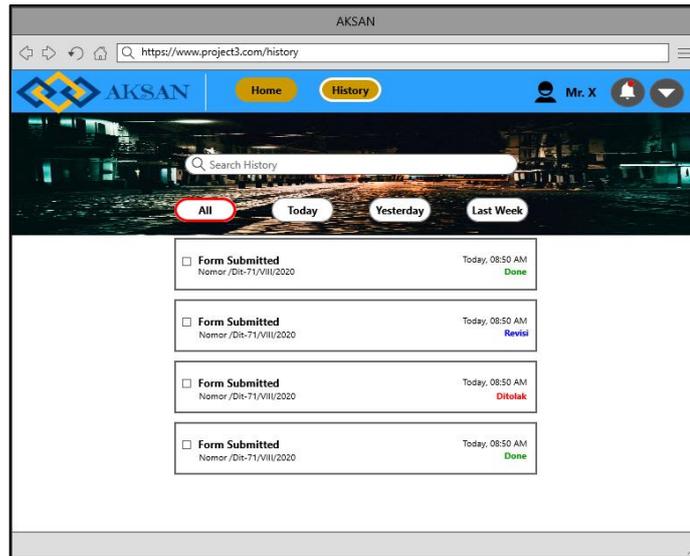
Gambar 3.36. Halaman *Sign up* AKSAN

Halaman utama ketika pengguna mengakses *website* dan setelah melakukan *login* tertera pada Gambar 3.37. Halaman utama memiliki beberapa tombol, yaitu pengguna dapat langsung *submit form* yang berisi kegiatan yang ingin diajukan, serta melihat halaman *history* dari *form* yang sudah pernah diajukan.



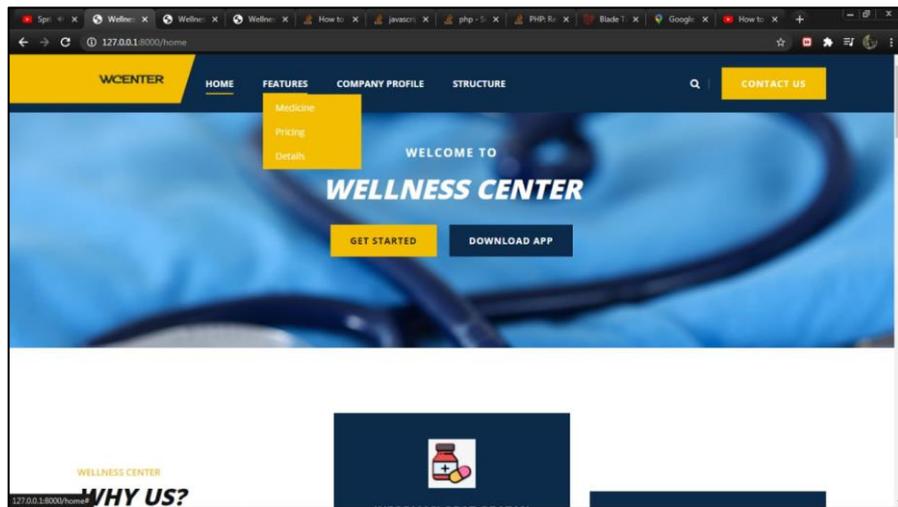
Gambar 3.37. Halaman utama *user* AKSAN

Gambar 3.38 merupakan halaman *history* pengguna dari *website* AKSAN. Di dalam halaman ini, terdapat beberapa tombol yang dapat digunakan oleh pengguna untuk melihat keseluruhan *history*, atau pada waktu tertentu.



Gambar 3.38. Halaman *history* website AKSAN

Pada minggu ke-7, dilakukan pembuatan *template* untuk *website company profile* untuk projek Wellness Center. Proses pembuatan *website* ini menggunakan Laravel. Gambar 3.39 merupakan halaman utama dari *website* Wellness Center. Di dalam halaman ini terdiri dari 4 bagian yaitu *home*, *features*, *company profile*, dan struktur organisasi.



Gambar 3.39. Halaman utama *template website* Wellness Center

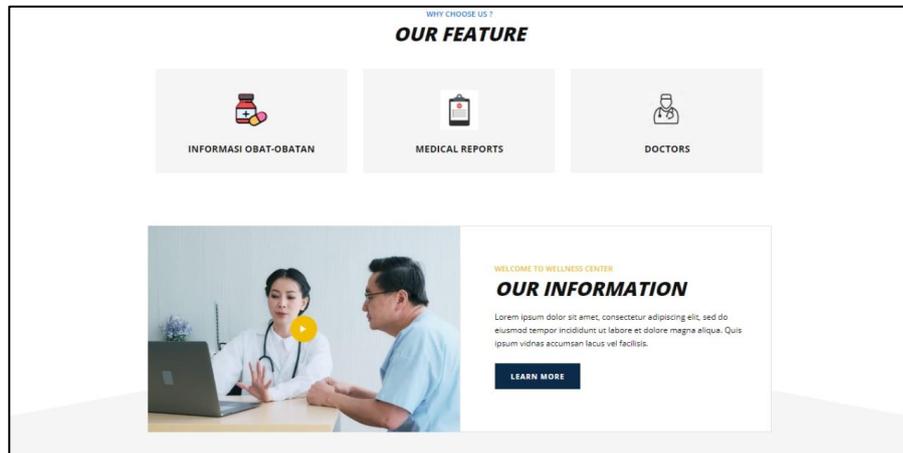
Di dalam halaman ini, terdapat keterangan mengenai beberapa paket konsultasi yang dapat dipilih oleh pengguna. Paket terdiri dari 3 bagian yang dapat dipilih langsung oleh pengguna. Gambar 3.40 merupakan bagian *pricing* pada *website* Wellness Center.



Gambar 3.40. Bagian *pricing website* Wellness Center

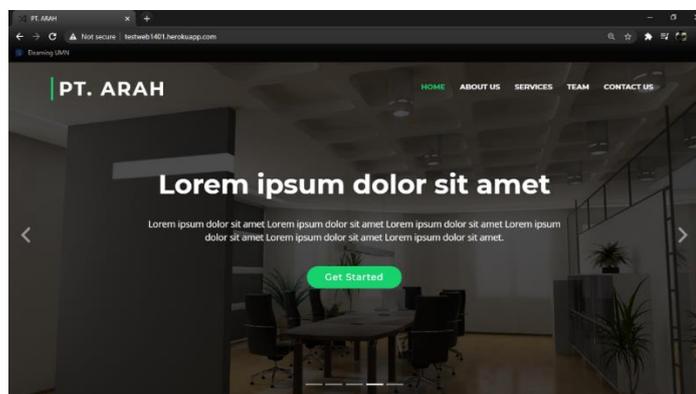
Pada halaman *feature* ini, pengguna dapat melihat beberapa produk yang ditawarkan oleh aplikasi Wellness Center. Serta terdapat informasi tambahan

terkait aplikasi dan *video* yang dapat diputar melalui Youtube. Gambar 3.41 merupakan tampilan dari halaman *feature*.



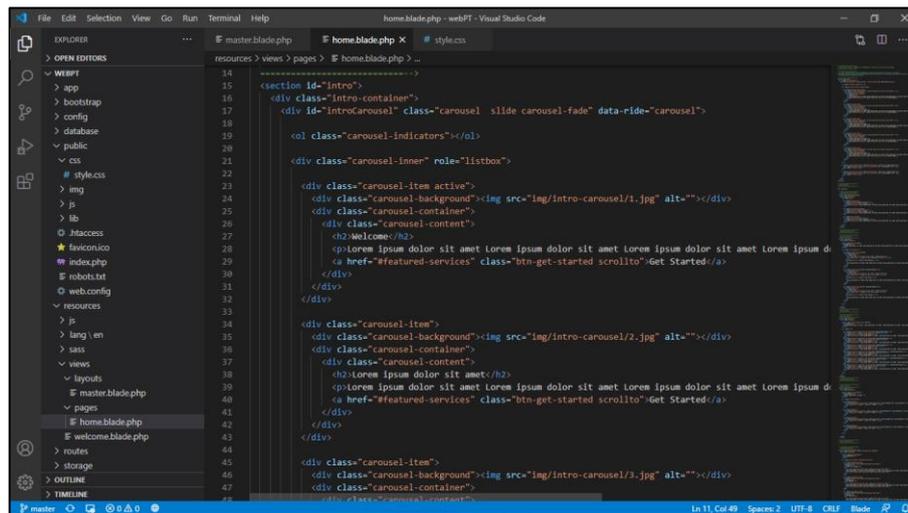
Gambar 3.41. Halaman *feature* pada *website* Wellness Center

Dan pada minggu ke-8 hingga minggu ke-10, diberikan tugas terakhir yaitu membuat *template* pada *website company profile* untuk salah satu PT yaitu PT. Arah Akakia Teknologi yang juga merupakan bagian dari PT. Viva Velove Viktori. Proses pembuatan *website* ini menggunakan Laravel. Gambar 3.42 berikut merupakan hasil *template website* yang dibuat menggunakan Laravel di Visual Studio.



Gambar 3.42. *Template website* PT.Arah Akakia Teknologi

Dan untuk hasil potongan *code* pembuatan *template website* untuk PT. Arah Akakia Teknologi dibuat dengan menggunakan Visual Studio, tertera pada Gambar 3.43.



```
14 .....
15 <section id="intro">
16 <div class="intro-container">
17 <div id="introCarousel" class="carousel slide carousel-fade" data-ride="carousel">
18
19 <ol class="carousel-indicators"></ol>
20
21 <div class="carousel-inner" role="listbox">
22
23 <div class="carousel-item active">
24 <div class="carousel-background"></div>
25 <div class="carousel-container">
26 <div class="carousel-content">
27 <h2>Welcome!</h2>
28 <p>Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum d
29 <a href="#featured-services" class="btn-get-started scrollto">Get Started</a>
30 </div>
31 </div>
32 </div>
33
34 <div class="carousel-item">
35 <div class="carousel-background"></div>
36 <div class="carousel-container">
37 <div class="carousel-content">
38 <h2>Lorem ipsum dolor sit amet</h2>
39 <p>Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum dolor sit amet Lorem ipsum d
40 <a href="#featured-services" class="btn-get-started scrollto">Get Started</a>
41 </div>
42 </div>
43 </div>
44
45 <div class="carousel-item">
46 <div class="carousel-background"></div>
47 <div class="carousel-container">
48 <div class="carousel-content">
```

Gambar 3.43. Hasil potongan *code* pembuatan *website* Laravel

Dalam pengerjaan *website* menggunakan Laravel, untuk dapat memanggil setiap halaman, diperlukan penambahan *code* di dalam folder “route > web.php”. Sebagai contoh jika ingin memanggil halaman *home*, maka perlu ditambahkan *code* seperti pada Gambar 3.41. Selain itu, di dalam pengerjaan menggunakan Laravel juga lebih dimudahkan dalam hal penyusunan *code*. Sebagai contoh, untuk *code* yang digunakan pada *header* dan *footer* dapat ditampung ke dalam 1 file contoh dengan nama *master.php*. Pada setiap halaman *home*, *about*, *profile*, dll, tidak perlu untuk menambahkan beberapa baris *code* untuk melengkapi bagian *header* dan *footer*. Cukup dengan menambahkan *code* seperti “`@extends('layouts.master')`” “. Dimana “*layouts*” adalah nama folder di dalam folder Views, dan “*master*” merupakan nama file yang diisi untuk menampung *code* pada *header* dan *footer*.

```
Route::get('/home', function () {  
    return view('pages.home');  
});
```

Gambar 3.44. Potongan *code* Laravel

Selanjutnya, untuk minggu-minggu terakhir dilakukan pembuatan dokumen-dokumen yang diperlukan pada *project* yang sedang berjalan seperti pembuatan dokumen untuk Wellness Center yang terdiri dari fungsi aplikasi, *flow*, dan lain-lain. Serta juga pembuatan *functional specification* untuk projek AKSAN. Didalam *functional specification*, terdapat beberapa penjelasan seperti latar belakang, *timescales and priorities*, penjelasan mengenai *business requirements, policy, functional requirement* dan *UI*.

3.3. Kendala yang dihadapi

Beberapa kendala yang dihadapi saat dilakukannya kegiatan kerja magang antara lain seperti mendapatkan tugas yang sebelumnya tidak diketahui. Seperti penggunaan Flutter , Laravel yang belum pernah dipelajari sebelumnya. Pada awalnya, diperoleh banyak *error* pada saat tahap implementasi. *Error* yang didapatkan ketika pada saat mencoba untuk “run” aplikasi menggunakan Android Studio. *Error* lainnya pada saat penggunaan Laravel, yaitu pada penggunaan fungsi “extends” , “section” dan lain-lain karena belum terbiasa dalam penggunaannya.

3.4. Solusi atas kendala

Dengan cukup banyaknya kendala yang sudah dihadapi, dapat diatasi dengan cara berikut ini :

1. Bertanya langsung kepada pembimbing lapangan dan memperoleh arahan serta masukan sehingga kendala yang dihadapi dapat diatasi dengan baik.
2. Mencari *tutorial* yang ada di Internet mengenai penggunaan Flutter, Laravel dan lainnya yang belum diketahui lebih lanjut.