

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Program kerja magang dilakukan di PT Satwa Teknologi Utama yang ditempatkan pada divisi *Website Developer* menjadi *junior front-end developer* yang dipimpin langsung oleh *Chief Technology Officer (CTO)* dari PT Satwa Teknologi Utama itu sendiri yakni James Roberto. Pada divisi *website developer* itu sendiri tentunya bertanggung jawab terhadap berjalannya *website* Petloka. Apabila ditemukan sebuah permasalahan pada *website* Petloka, divisi *Junior Front-End developer* akan langsung dicari dan diminta untuk melakukan pengecekan terhadap permasalahan tersebut. Apabila kesalahan terdapat pada *Front-End* dari *website*, maka divisi *Junior Front-End Developer* akan langsung memperbaikinya. Sedangkan, apabila kesalahan terdapat pada *Back-End* dari *Website*, maka divisi *Junior Front-End Developer* akan menghubungi divisi *Back-End Developer* dan kemudian saling berkoordinasi untuk memperbaikinya. Divisi ini juga bersama dengan divisi *Mobile Apps Developer* dan *Back-End Developer* saling bekerja sama untuk membuat adanya suatu pengembangan yang baru terhadap *website* Petloka.

32 Tugas yang Dilakukan

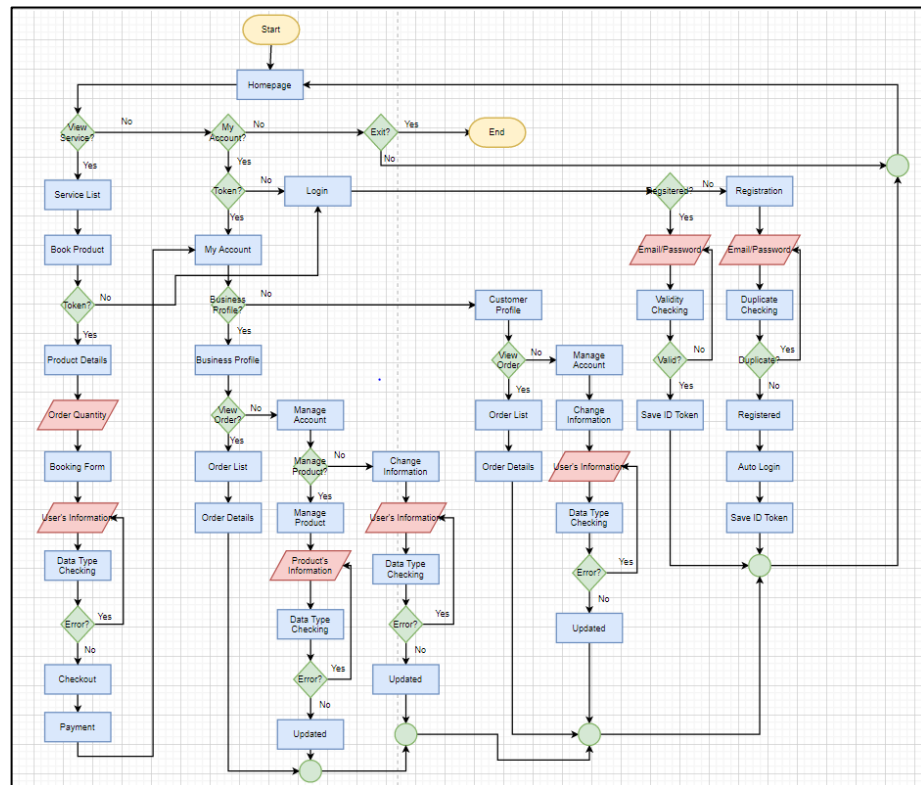
Pada program kerja magang ini, sebagai *Junior Front-End developer* memiliki tugas di PT Satwa Teknologi Utama yaitu untuk membuat sebuah pengembangan pada *website* Petloka, yang dimana pengembangan tersebut akan dibuat sesuai dengan arahan yang diberikan oleh atasan langsung (CTO). Permintaan untuk pengembangan itu sendiri dapat berupa langsung dari CTO, ataupun oleh divisi lain yang memerlukan (seperti misalnya marketing) melalui persetujuan dari CTO terkait kapabilitas dan tenggat waktu untuk menyelesaikan permintaan tersebut.

Adapun pengembangan yang dilakukan dapat bervariasi sesuai dengan persetujuan saat dilakukan wawancara dan seperti yang tercantum pada surat penerimaan kerja magang sebagai berikut:

- 1 Pembuatan dan pengaturan komponen (*view, service, model, dll*)
- 2 Integrasi API
- 3 Pembuatan dan pengaturan *view*

33 Rancangan Aplikasi

Berikut adalah *flowchart* untuk *website* petloka

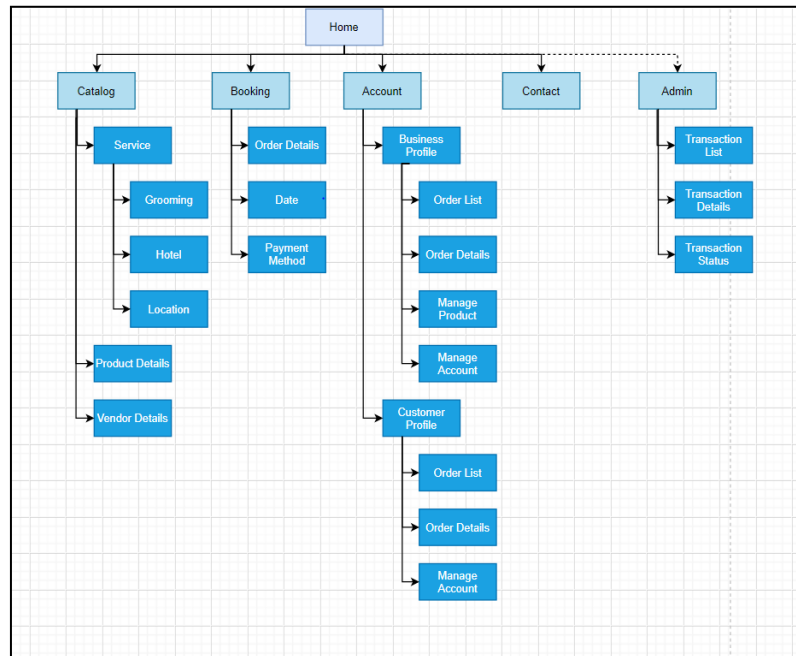


Gambar 3.1 *Flowchart* Petloka

Gambar 3.1 merupakan *flowchart* yang digunakan untuk *website* petloka, dengan trapesium merah adalah *output* dan trapesium biru adalah *input*, saat pertama mengakses *website* halaman *homepage* yang pertama kali muncul lalu bisa memilih untuk *view service* untuk melihat *list vendor* kemudian dilanjutkan ke *list product* hingga ke *booking* tetapi sebelum bisa melakukan *booking* harus dipastikan kalau sudah memiliki *token* atau tidak, *token* adalah seperti penanda kalau sudah *login* atau belum kalau tidak akan dibawa menuju halaman *login* kalau

belum memiliki *email* user dapat mendaftar dan kalau sudah tinggal *login* dan *website* akan mengecek *last url* dan apabila *booking* adalah *last url*nya maka *website* akan langsung menuju ke halaman *booking* dilanjutkan dengan *payment* apabila sudah akan menuju ke *my account* yang didalamnya berisi *order list*.

Berikut adalah *sitemap* dari *website* Petloka:



Gambar 3.2 *Sitemap* Petloka

Gambar 3.2 merupakan *sitemap* dari *website* petloka, *Home* adalah tampilan awal saat *website* dibuka, lalu setelah itu dipecah menjadi beberapa bagian.

Catalog terdiri dari *service*, *product details* dan *vendor details*, *service* dipecah lagi menjadi *grooming*, *hotel*, dan *location*.

Booking terdiri dari *order details*, *date*, dan *payment method*.

Account terdiri dari *Business Profile* dan *Customer Profile*, dibedakan menjadi 2 karena kalau *profile* disatukan *module profile* diperusahaan akan menjadi sangat berantakan. *Business Profile* terdiri dari *order list*, *order details*, *manage product*, dan *manage account*. *Customer Profile* terdiri dari *order list*, *order details*, dan *manage account*.

Admin digunakan sebagai alat pembantu untuk *vendor* yang mengalami kesulitan untuk menerima *orderan*. *Transaction list* untuk melihat semua transaksi yang ada diwebsite, *Transaction details* untuk melihat *details* dari semua transaksi yang ada diwebsite, *Transaction status* untuk mengganti status *orderan*.

34 Uraian Pelaksanaan Kerja Magang

Berikut merupakan *timeline* dan uraian pelaksanaan kerja selama menjalani Program Kerja Magang sebagai *Junior Front-End developer*.

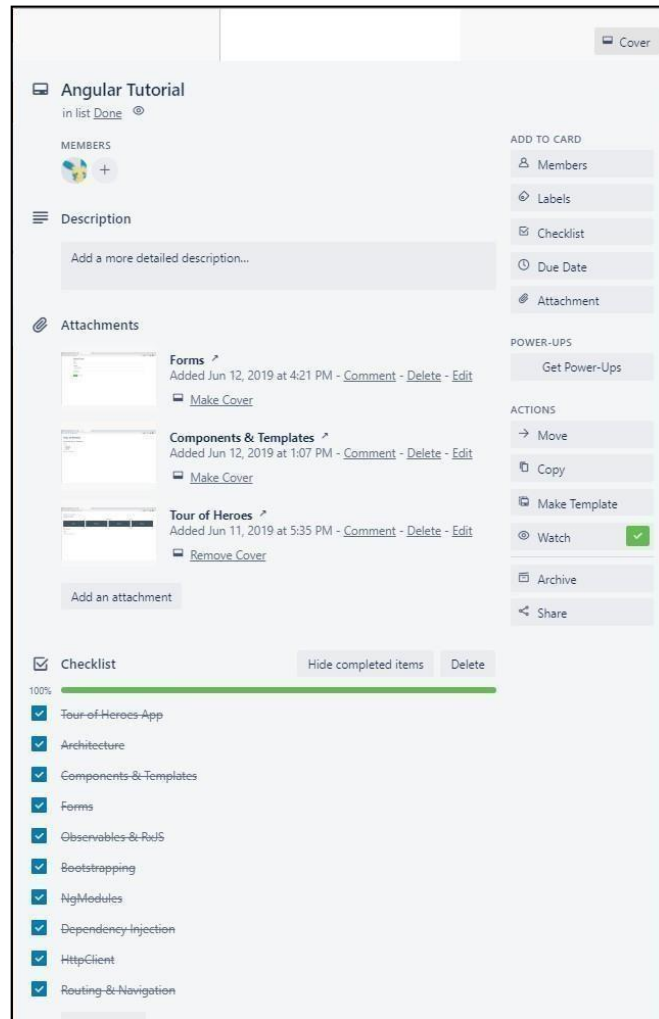
3.4.1 Instalasi *Tools* untuk Kerja Magang dan Mempelajari *Library Angular* (Minggu Pertama)

Pada minggu pertama, kegiatan yang pertama kali dilakukan tentunya adalah berkenalan dengan tim-tim dari divisi lain yang selama 3 bulan akan bekerja bersama. CTO kemudian memberikan arahan untuk melakukan instalasi aplikasi dan juga *tools* yang berguna untuk kerja magang nantinya. Aplikasi dan *tools* tersebut antara lain adalah *Visual Studio Code*, *Node*, *Angular*, *Trello*, dan *Fork*.

Angular merupakan sebuah *tools* yang digunakan PT Satwa Teknologi Utama untuk membangun dan mengembangkan *website* untuk Petloka. *Angular* itu sendiri merupakan sebuah *app-design framework* yang dikembangkan oleh *Google* untuk

membangun aplikasi *single- page* baik untuk *mobile*, *web*, dan *desktop* dengan satu basis kode. PT Satwa Teknologi Utama menggunakan *Angular* dengan alasan *Angular* dapat membangun website lebih cepat dengan komponen – komponen yang disediakan *Angular*. *Angular* sendiri menggunakan bahasa pemrograman bernama *Typescript*. *Angular* dapat diakses melalui *Visual Studio Code* dengan cara melakukan instalasi ekstensi dari *Angular* pada aplikasi tersebut.

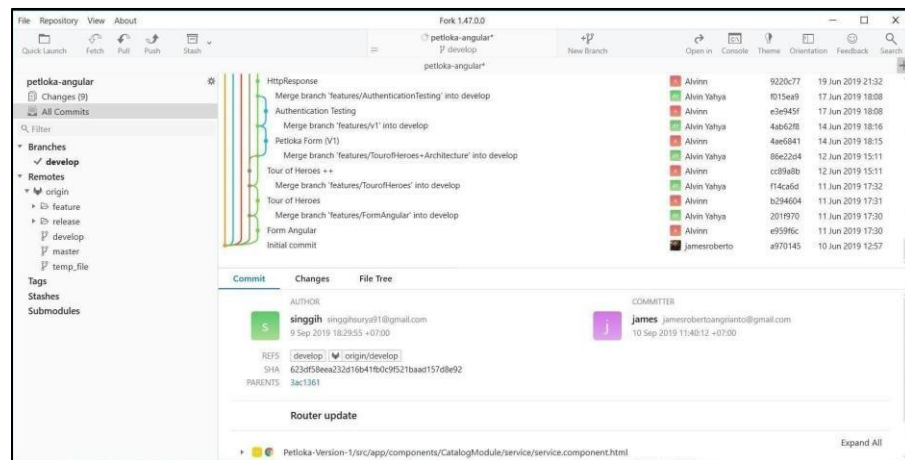
Trello merupakan sebuah *tools* yang memberikan untuk mengatur dan menyusun pekerjaan. Pada PT Satwa Teknologi Utama itu sendiri, *Trello* digunakan untuk 5 hal utama, yakni menuliskan rencana pengembangan yang akan dibuat kedepannya, pekerjaan apa yang harus dilakukan, pekerjaan yang sedang dilakukan, *review/testing* dari pekerjaan yang sudah dilakukan, dan pekerjaan yang sudah selesai dan siap untuk dinaikkan ke aplikasi. Pembuatan pekerjaan-pekerjaan tersebut berupa sebuah *card* yang apabila ditekan akan memunculkan halaman *popup* seperti pada Gambar 3.7



Gambar 3.3 Contoh isi dari *card* pada *Trello*

Untuk *tools* terakhir, terdapat *Fork* yang dimana merupakan sebuah *Git client* yang *user friendly*. PT Satwa Teknologi Utama menggunakan *GitLab* untuk mengatur repositori *Git*. *Fork* ini sendiri digunakan untuk memudahkan tim *developer* untuk mengerjakan pekerjaannya masing-masing secara paralel dan tidak

mengganggu pekerjaan yang satu dengan yang lainnya. Hal tersebut dapat dilakukan dengan membuat sebuah *branch* baru di lokal dari *branch* yang sudah ada di repository *remote*. *Branch* yang sudah ada di lokal tersebut kemudian dapat di *push* agar kemudian *branch* yang dibuat dapat muncul di repository *remote* dan dapat diakses juga oleh anggota tim *developer* yang lain.



Gambar 3.4 Tampilan *Fork*

Kemudian setelah selesai melakukan instalasi untuk semua *tools* tersebut, kerja magang dilanjutkan dengan mempelajari *library* yang ada pada *Angular* sesuai dengan arahan dari CTO PT Satwa Teknologi Utama. Pada Gambar 3.9, dapat dilihat daftar dari *library* yang harus dipelajari. *Library* tersebut dipelajari

secara mandiri dengan memanfaatkan internet untuk mencari berbagai macam sumber terkait dengan materi yang ingin dipelajari. Namun, dari *angular* sendiri pun menyediakan sumber berupa dokumentasi dari *component*, *library*, dan lain-lain yang ada pada <https://angular.io/>.

Sementara mempelajari, juga diminta untuk mencobanya langsung melalui *Visual Studio Code* yang digunakan.

3.4.2 Mempelajari Struktur Data *Angular* Petloka (Minggu Kedua)

Di minggu kedua, kerja magang kembali dilanjutkan dengan melihat dan mempelajari repositori *angular* Petloka yang ada di *Git*. Kerja magang ini bersifat membantu pengembangan dari *website* Petloka yang sudah ada. Pada struktur repositori *remote* dari *website* Petloka ini terdapat beberapa pembagian yang terdiri atas satu *folder* bernama *feature*, satu *folder* bernama *release*, dan satu buah *branch* bernama *develop*. Untuk *folder feature* berisikan kumpulan dari *branch* yang dimana berupa pekerjaan-pekerjaan yang dilakukan oleh tim *developer* untuk menambahkan suatu fitur pada aplikasi Petloka. Untuk *folder release* berisikan kumpulan dari *branch* yang dimana berupa pekerjaan-pekerjaan yang dilakukan oleh tim *developer* untuk *dilaunching websitenya*. Terakhir, untuk *branch develop* berupa *branch* yang

dimana kondisi aplikasinya sesuai dengan kondisi aplikasi yang sedang rilis *live server*. Sebenarnya terdapat satu buah *branch* lagi yang bernama *master*, namun selama kerja magang berlangsung *branch* tersebut tidak pernah tersentuh sama sekali dan terakhir kali *branch* tersebut *update* adalah pada bulan Januari. Kerja magang lebih sering berinteraksi dengan *folder feature* dan *folder release* karena memang kedua *folder* tersebut merupakan pengembangan dan perbaikan yang akan dilakukan pada *website* Petloka. Setelah diperkenalkan dengan repositori dari *website* Petloka, kerja magang kembali dilanjutkan dengan mempelajari isi dari *branch develop* dan kemudian melakukan *build* atau membuat *localhost website* Petloka untuk merasakan *experience* dalam menggunakan *website* Petloka.

3.4.3 Pembuatan *Authorization Module* untuk *Login, Register, Logout* dan *Json Parse response server* (Minggu Ketiga)

Pada minggu ketiga, kerja magang kembali dilanjutkan dengan pembuatan komponen *Authorization Servicer*, serta *JSON Parse* jawaban dari server. Status *User* akan dinyatakan sebagai *Logged In* apabila *user* telah melakukan *login*, serta data untuk login sama dengan data *backend*.

Pembuatan *Authorization Module* dibuat dengan maksud agar komponen tersebut yang akan mengatur segala jenis *Authorization*.

Authorization Module juga dibantu dengan membuat *Authorization Service* yang mengatur logika – logika yang digunakan untuk *Authorization*.

Authorization menggunakan variabel – variabel sebagai berikut:

Tabel 3.2 Variabel pada komponen *Authorization Module*

| Nama Variabel | Tipe Data |
|-------------------|------------------|
| <i>loginForm</i> | <i>FormGroup</i> |
| <i>Auth</i> | <i>Auth</i> |
| <i>isLoggedIn</i> | <i>boolean</i> |
| <i>token</i> | <i>String</i> |
| <i>Id_token</i> | <i>String</i> |
| <i>Step</i> | <i>Number</i> |

Tabel 3.2 adalah variabel yang digunakan pada komponen *authorization module*, *loginForm* mempunyai tipe data *FormGroup* yang diambil dari *Angular Form* sendiri, *Auth* merupakan *service* pada *Angular*, komponen *service* diberi nama *Auth*, agar lebih mudah saat mau dipanggil. *isLoggedIn* memiliki tipe data *boolean* hanya digunakan untuk mengecek kalau sudah ada *token* atau belum kalau sudah *isLoggedIn* akan menjadi *true* dan *false* untuk sebaliknya. *Token* memiliki tipe data *string* merupakan tempat tampungan *token* yang diambil dari *response server*. *Id_token* merupakan tampungan untuk nomor unik *token* yang tersimpan dalam *JSON token*. *Step* merupakan alat bantu untuk melihat *form* bagian mana yang diakses.

The image shows a login form with the following elements:

- Title:** "Please login" in a large, bold, black font.
- Email Input:** A text input field with the placeholder text "Enter your email".
- Password Input:** A text input field with the placeholder text "Enter your password".
- Login Button:** A blue button with the text "Login" in white.
- Register Button:** A blue button with the text "Register" in white.

Gambar 3.5 Contoh penggunaan *authorization module* pada halaman *login*

Gambar 3.5 adalah contoh tampilan login yang berisi *form* untuk mengisi *email* dan *password*.

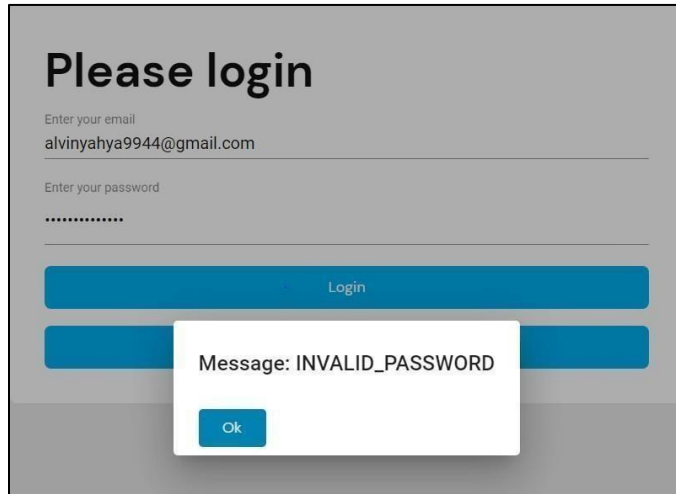
Gambar 3.6 Contoh penggunaan *authorization module* pada halaman *register*

Gambar 3.6 memiliki tampilan yang tidak jauh beda dari gambar 3.5 berisi *form* untuk memasukkan *email* dan *password*.

Tabel 3.3. Komponen pada pembuatan *HttpClient website* Petloka

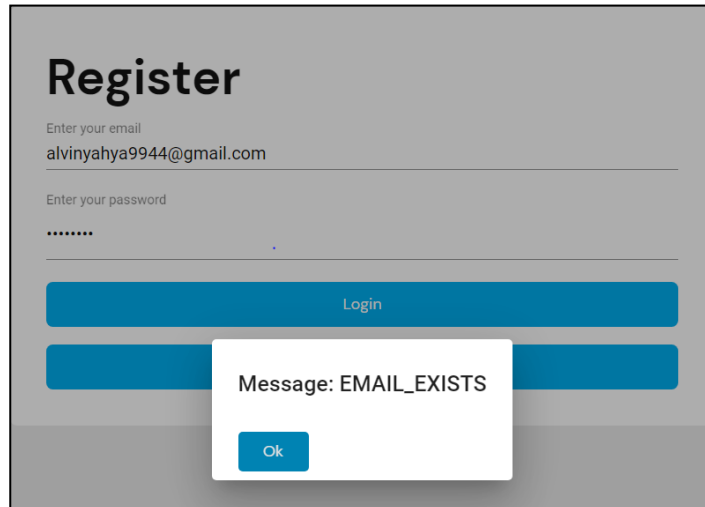
| Nama Komponen | Kegunaan |
|-------------------|---|
| <i>HttpClient</i> | Berupa model variabel penampung atribut JSON, baik untuk dikirim ke <i>endpoint</i> (POST, PUT) ataupun menarik dari <i>endpoint</i> (GET). |
| <i>HttpHeader</i> | Merupakan tambahan <i>header</i> untuk <i>HttpClient</i> |

HttpClient akan memanggil *endpoint* apabila *user* ingin melakukan *Login* atau *Register* dengan data yang diinput dihalaman *login* dan *register*, apabila data yang diinput salah *user* tidak diperbolehkan untuk *login* apabila *user* melakukan *register* *HttpClient* akan mengecek data duplikat yang ada di *endpoint* apabila tidak ada data duplikat maka *user* diperbolehkan untuk melakukan *register*.



Gambar 3.7 Contoh *password* yang salah

Gambar 3.7 merupakan tampilan dari *password* yang salah *message* tersebut merupakan *response* dari *back-endnya*.

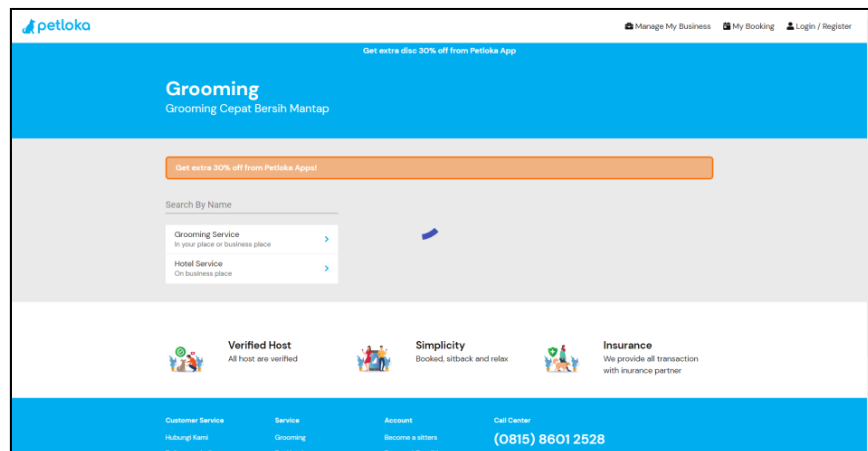


Gambar 3.8 Contoh data duplikat

Gambar 3.8 merupakan tampilan dari data yang duplikat saat ingin melakukan registrasi.

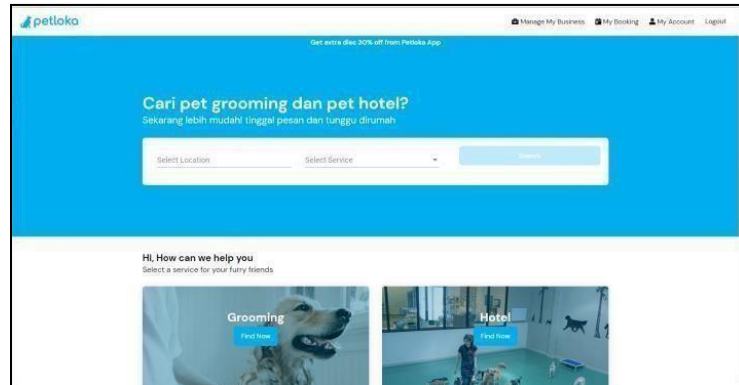
3.4.4 Pembuatan *Catalog Module* (Minggu Keempat)

Di minggu keempat, pekerjaan yang dilakukan adalah pembuatan *Catalog Module*. *Catalog Module* sendiri sebelumnya sudah pernah dibuat, namun bukan dalam format *angular*. *Catalog Module* ini berisikan *list vendor* yang *product* dan *serviceny* terdaftar di *backend*. Dan menampilkan status *vendor* apakah *vendor* sudah *terverified* atau belum *terverified*, *vendor* yang *terverified* adalah *vendor* yang data *productnya* sudah *diapprove* oleh tim operasional.



Gambar 3.9 Contoh tampilan *catalog module*

Tampilan beserta *asset* tampilan sudah diberikan oleh CTONya yang dikerjakan hanya menggabungkan tampilan dengan *endpoint*, sebelum *catalog module* dibuat tentu *website* memerlukan *homepage*.

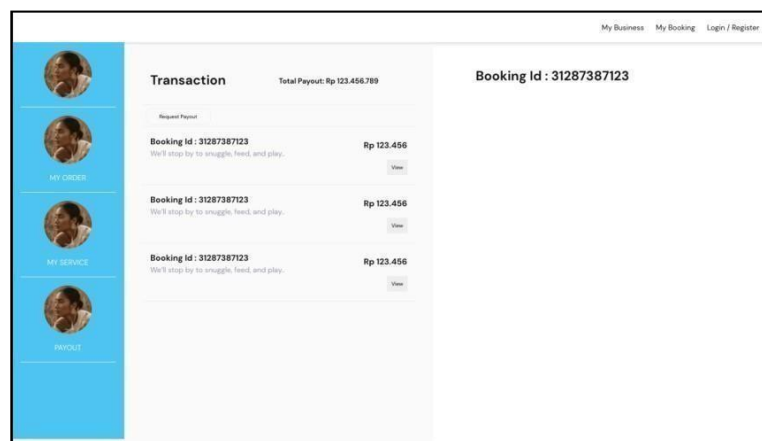


Gambar 3.10 Contoh tampilan *homepage*

Gambar 3.10 adalah tampilan petloka yang dibuat oleh CTO dan diberikan kepada tim.

3.4.5 Pembuatan *User Module* (Minggu Kelima)

Di minggu kelima, pekerjaan yang diminta adalah untuk membuat *User Module*. Dengan menerapkan *HttpClient*, pada *endpoint host* sudah terdapat atribut atribut yang dibutuhkan, atribut diambil dengan menggunakan *JSON Parse*.

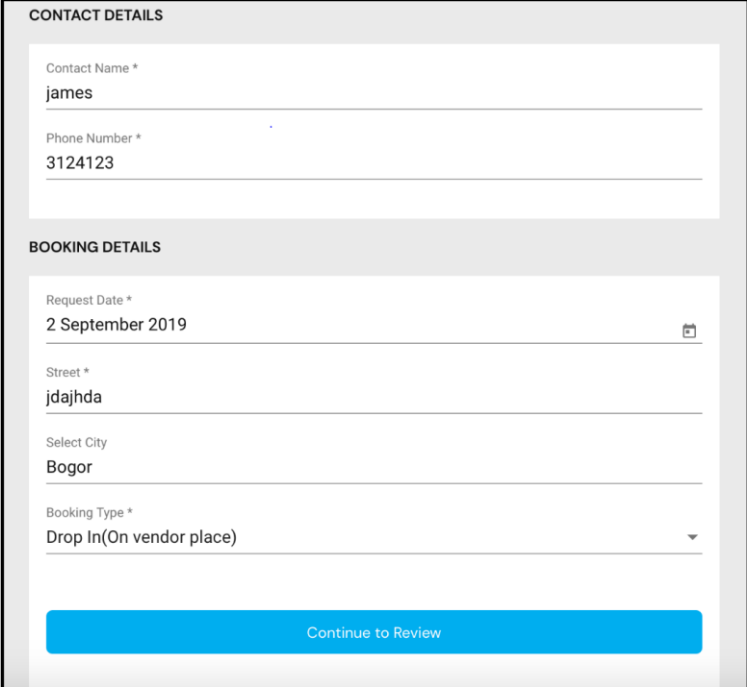


Gambar 3.11 Contoh tampilan *user module*

Gambar 3.11 memperlihatkan sekilas dari *User Module*, gambar yang digunakan merupakan gambar *mock-up*.

3.4.6 Pembuatan *Booking Module* (Minggu Keenam)

Di minggu keenam, terdapat pekerjaan untuk membuat *Booking Module*. *Booking Module* mengatur segala aktivitas *booking product* dan *booking service* antara *vendor* dan *user*.

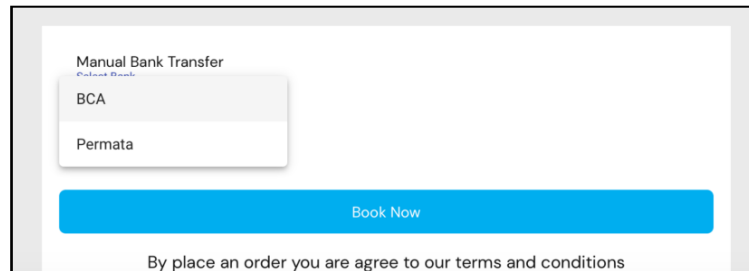


The image shows a web form titled "CONTACT DETAILS" and "BOOKING DETAILS". The "CONTACT DETAILS" section includes fields for "Contact Name *" (filled with "james") and "Phone Number *" (filled with "3124123"). The "BOOKING DETAILS" section includes fields for "Request Date *" (filled with "2 September 2019"), "Street *" (filled with "jdajhda"), "Select City" (filled with "Bogor"), and "Booking Type *" (filled with "Drop In(On vendor place)"). A blue button labeled "Continue to Review" is located at the bottom of the form.

Gambar 3.12 Contoh tampilan *booking form*

Gambar 3.12 adalah tampilan *booking form* yang akan digunakan saat ingin memesan *product* dari *vendor* yang dibutuhkan adalah tanggal pemesanan, alamat jalan rumah, alamat kota rumah, dan tipe *booking*.

Fitur pembayaran sudah disiapkan juga tetapi masih belum berjalan dan belum terhubung ke *endpoint*.



Gambar 3.13 Contoh tampilan fitur pembayaran

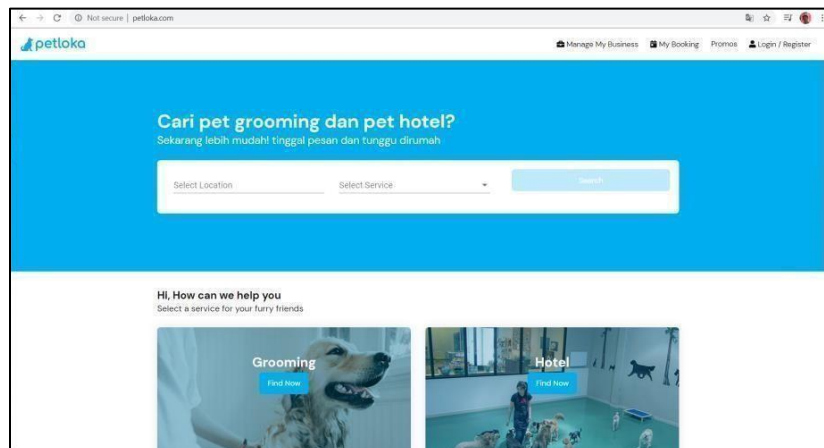
Setelah *booking* dilakukan *user* bisa mengecek status *booking* di *Myaccount*.

3.4.7 Pembaharuan *UI* dan *Dependencies* (Minggu Ketujuh & Minggu Kedelapan)

Di minggu ketujuh dan kedelapan, pekerjaan yang dilakukan adalah memperbaharui *Booking Module*. CTO memberikan *asset css* yang baru untuk diupdate beserta *dependencies*nya dan juga memperbaharui *routing*. Update *Angular* ke versi yang lebih baru.

3.4.8 *Production Build* (Minggu Kesembilan)

Di minggu kesembilan, pekerjaan yang dilakukan adalah mempersiapkan *website* untuk *production build*. *Production Build* dilakukan setelah repositori di *Gitlab* sudah dirapihkan dan *dimerge* ke branch *develop*. Tujuan *production build* adalah untuk memasukkan *website* yang *localhost* kedalam *hostingan website* menjadi *live server*. Sebelum *Production Build*, CTO mengetes *websitenya* terlebih dahulu dan menyesuaikannya dengan *mockup* dan apabila sudah beres, *Production Build* baru bisa dilakukan tentu dengan memperbaiki *bug* yang ada di *websitenya*, di minggu kesembilan juga diberikan tugas untuk membuat tampilan *mobile* untuk *websitenya*.

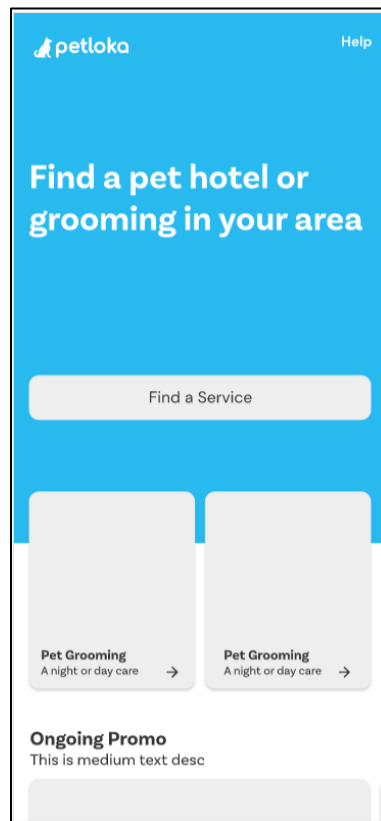


Gambar 3.14 Contoh tampilan *live server*

Gambar 3.14 adalah tampilan *live server* yang menyatakan bahwa *website* sudah masuk disalah satu *hosting* yang bekerja sama dengan perusahaan.

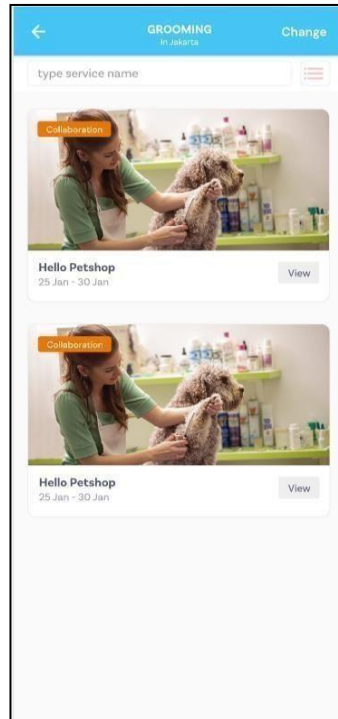
3.49 Pembuatan *Mobile Page* dan *Manage Product* (Minggu Kesepuluh)

Di minggu kesepuluh, pekerjaan yang dilakukan adalah pembuatan *Mobile Page* dan *Manage Product*. *Mobile Page* dibuat untuk memudahkan akses *mobile* ke *website* dengan cara membuat *view/tampilan* baru untuk *website*, jadi membuat 2 *view* untuk 1 *module*, 1 untuk *mobile* 1 untuk *website*, *Angular* menyediakan *plug-in* yang bisa mendeteksi apakah *user* menggunakan *mobile*, atau *desktop*.



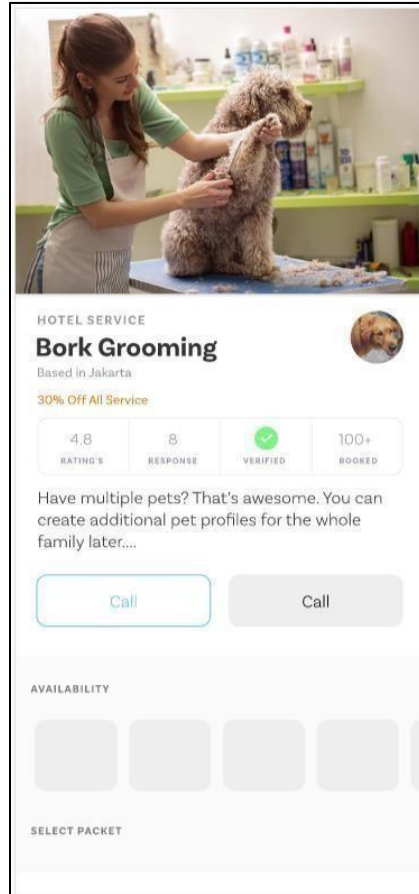
Gambar 3.15 Contoh tampilan *homepage mobile*

Gambar 3.15 merupakan tampilan *homepage* yang digunakan pada bagian *mobile view website* Petloka.



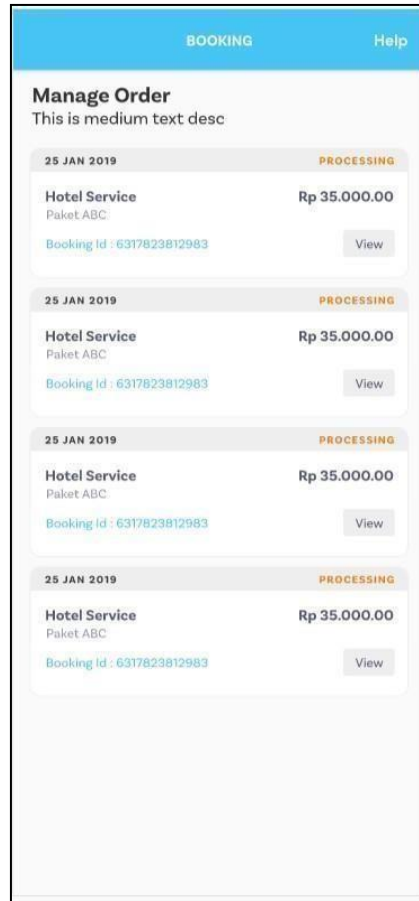
Gambar 3.16 Contoh tampilan *catalog mobile*

Catalog mobile memiliki *card* untuk setiap *list vendor* dan saat *view* diklik akan berpindah ketampilan *list product*.



Gambar 3.17 Contoh tampilan *product mobile*

Product mobile memiliki fungsi untuk *call* dan *chat* ke *vendor* meskipun belum berfungsi secara penuh tetapi sudah bisa digunakan.



Gambar 3.18 Contoh tampilan *order mobile*

Gambar 3.18 adalah tampilan *order list* yang digunakan pada *mobile view website* Petloka sama seperti *order list* yang ada di *website* hanya lebih disesuaikan untuk *mobile view*.

3.4.10 Pemanggilan *API Google Maps* (Minggu Kesebelas)

Di minggu kesebelas, pekerjaan yang dilakukan adalah memanggil *API* dari *google maps*. *Google maps* digunakan untuk mempermudah mencari lokasi pengiriman dengan memilih titik yang ada di *google maps*

```

ngOnInit() {
  this.loggedIn = this.authService.isLoggedIn();
  this.auth = JSON.parse(localStorage.getItem('customerSession'));
  this.pageTitle.setTitle('Create Booking');
  this.pId = this.cartSession.pId;
  this.serviceType = this.servService.getServiceType();

  this.mapsAPILoader.load().then(() => {
    this.setCurrentLocation();
    this.geoCoder = new google.maps.Geocoder;

    let autocomplete = new google.maps.places.Autocomplete(this.searchElementRef.nativeElement, {
      types: ["address"]
    });

    autocomplete.addListener("place_changed", () => {
      this.ngZone.run(() => {
        let place: google.maps.places.PlaceResult = autocomplete.getPlace();

        if (place.geometry === undefined || place.geometry === null) {
          return;
        }

        this.latitude = place.geometry.location.lat();
        this.longitude = place.geometry.location.lng();
        this.getAddress(this.latitude, this.longitude);
        this.zoom = 12;
      });
    });
  });
});

```

Gambar 3.19 Tampilan pemanggilan *API Google Maps*

Gambar 3.23 adalah kodingan yang dilakukan untuk memanggil *API Google Maps*, dikarenakan masalah pada *backend google maps*, *website* tidak berjalan seperti yang seharusnya.

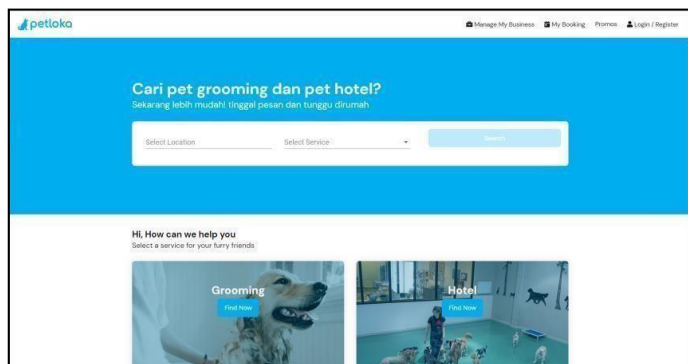
3.4.11 *Production Build II (Minggu Keduabelas)*

Di minggu keduabelas dilakukan kembali *production build* untuk memperbaharui *live server*, *production build* dilakukan setelah *merging branch* ke *develop* dan mempunyai fitur yang lebih banyak daripada *production build* pertama, serta memperbaiki *bug/error* yang ada pada *production build* pertama.

3.4.12 *Finalize Build* untuk dokumentasi dan *maintain website* (Minggu Ketigabelas dan Keempatbelas)

Di minggu ketigabelas dan keempatbelas adalah serah terima *website* kepada CTO setelah *website* sudah sesuai dengan *mockup* yang diberikan dan perbaikan *bug/error* yang ada di *website* selama 2 minggu *website* berjalan apabila ada *bug/error* tim operasional akan memberikan masukan kepada CTO yang kemudian diberikan kepada tim *front-end developer* untuk diperbaiki secepatnya. Selama 2 minggu ini pekerjaan yang diberikan CTO hanya mengawasi *website* untuk *bug testing*, ataupun kendala yang menghalangi tim operasional.

3.5 *Runtime website*



Gambar 3.20 Contoh tampilan *Homepage live server*

Gambar 3.20 merupakan tampilan awal saat *website* petloka dibuka pertama kali, *customer* bisa memilih *grooming* atau *hotel* atau mencari melalui lokasi.

The image shows a login form with the title "Please login" in bold black text. Below the title are two input fields: "Enter your email" and "Enter your password", each with a horizontal line underneath. At the bottom of the form are two blue buttons: "Login" and "Register".

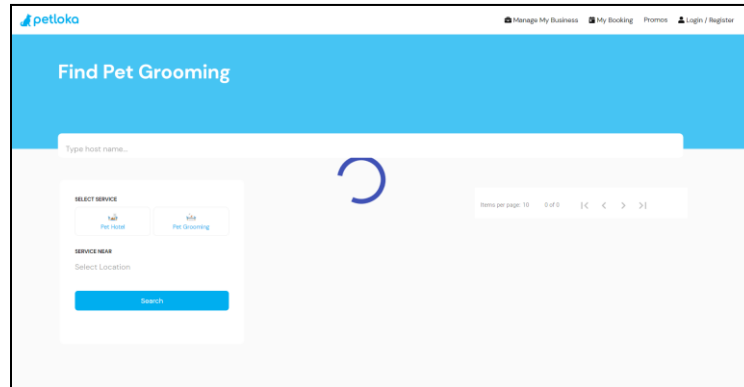
Gambar 3.21 Contoh tampilan *login live server*

Gambar 3.21 pada *website petloka* berisi *email* dan *password* dan tombol *login* dan *register*.

The image shows a register form with the title "Register" in bold black text. Below the title are two input fields: "Enter your email" and "Enter your password", each with a horizontal line underneath. At the bottom of the form are two blue buttons: "Login" and "Register".

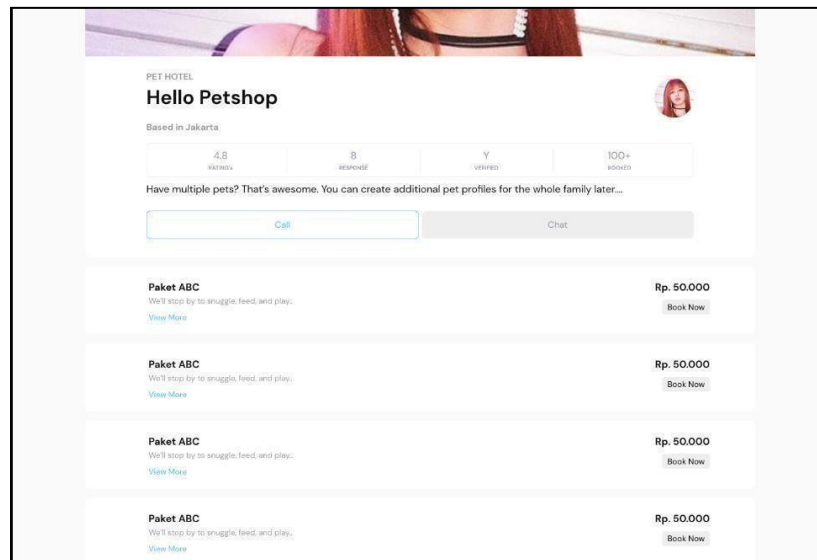
Gambar 3.22 Contoh tampilan *register live server*

Gambar 3.22 pada *website petloka* berisi *email* dan *password* dan tombol *login* dan *register*.



Gambar 3.23 Contoh tampilan *view service live server*

Gambar 3.23 berisi *list vendor* yang menjual *productnya*, *customer* juga bisa melakukan filter daerah, nama, tipe *service*.



Gambar 3.24 Contoh tampilan *list product mock-up*

Gambar 3.24 berisi *list product* yang ditawarkan oleh *vendor* dan sekilas *details* dari *vendor*.

The image shows a web form with two main sections: 'CONTACT DETAILS' and 'BOOKING DETAILS'. The 'CONTACT DETAILS' section includes fields for 'Contact Name *' (filled with 'james') and 'Phone Number *' (filled with '3124123'). The 'BOOKING DETAILS' section includes 'Request Date *' (filled with '2 September 2019'), 'Street *' (filled with 'jdajhda'), 'Select City' (filled with 'Bogor'), and 'Booking Type *' (filled with 'Drop In(On vendor place)'). A blue button labeled 'Continue to Review' is located at the bottom of the form.

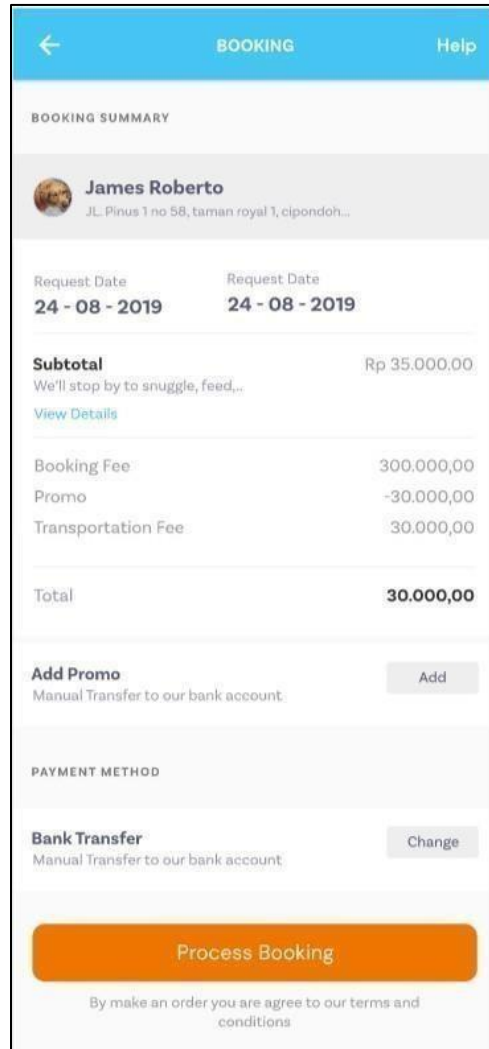
| CONTACT DETAILS | |
|-----------------|---------|
| Contact Name * | james |
| Phone Number * | 3124123 |

| BOOKING DETAILS | |
|-----------------|--------------------------|
| Request Date * | 2 September 2019 |
| Street * | jdajhda |
| Select City | Bogor |
| Booking Type * | Drop In(On vendor place) |

Continue to Review

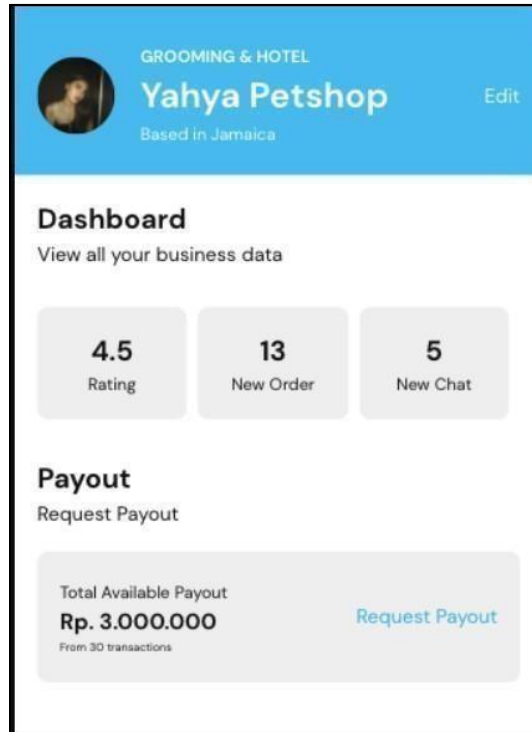
Gambar 3.25 Contoh tampilan *booking live server*

Gambar 3.25 berisi *form booking* untuk memasukkan informasi *customer* seperti alamat rumah, kota tempat tinggal, dan tanggal pemesanan.



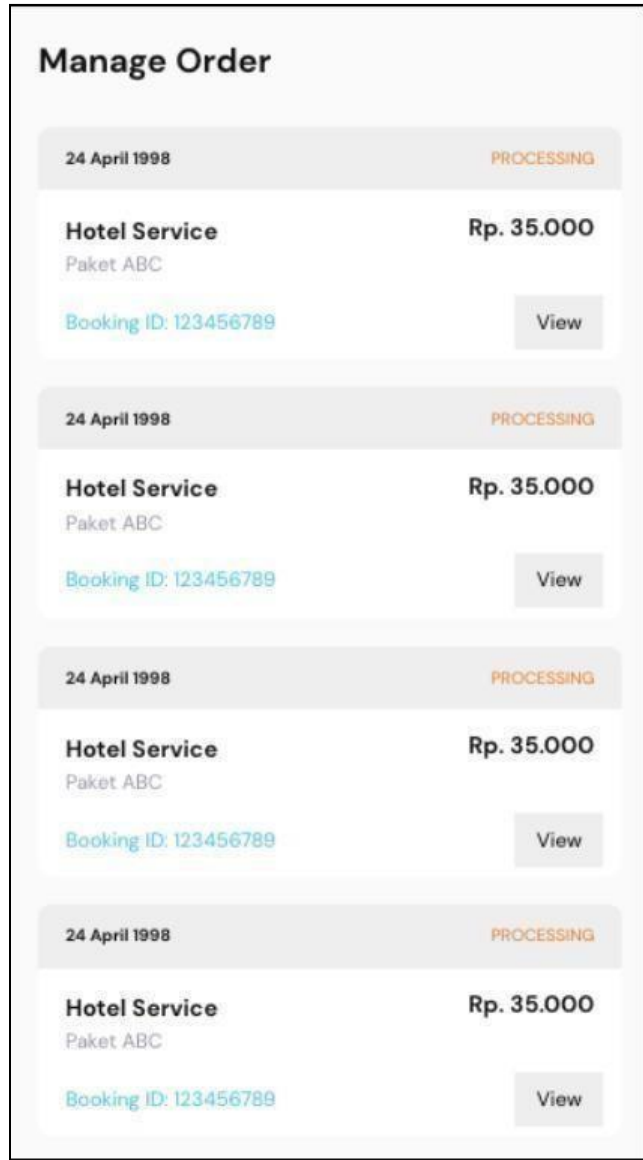
Gambar 3.26 Contoh tampilan *payment mobile mock-up*

Gambar 3.26 berisi total harga dari *orderan product* dari *vendor* tetapi tampilan diatas menggunakan *mockup* karena *backend* dari *website* sudah *dishut off* dan tidak bisa memanggil *API* lagi.



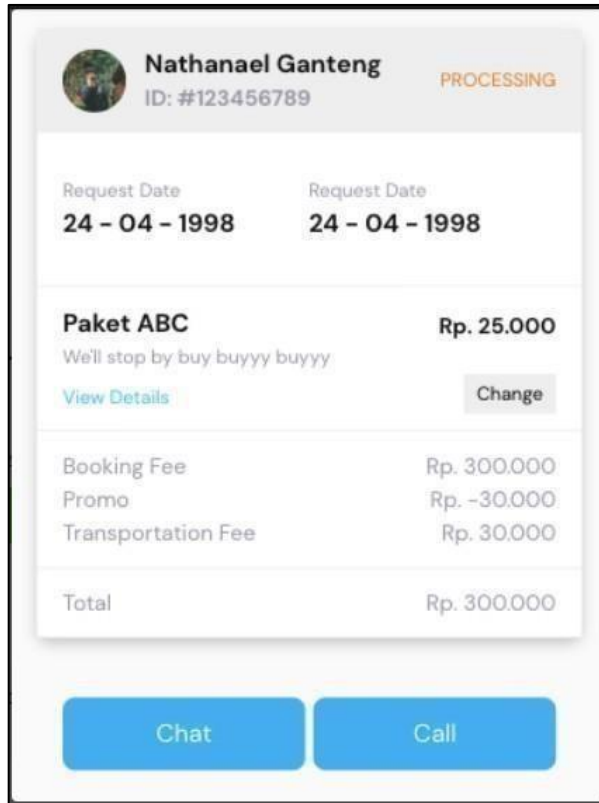
Gambar 3.27 Contoh tampilan *my account mock-up*

Gambar 3.27 berisi informasi tentang *rating*, jumlah *order*, *chat*, dan jumlah uang yang bisa ditarik ke rekening BCA dari *customer* yang memesan jasanya.



Gambar 3.28 Contoh tampilan *order list mock-up*

Gambar 3.28 berisi informasi jumlah *order* yang diterima dari *customer* yang memesan jasa *vendor*, *customer* juga memiliki tampilan *order list* yang sama untuk melihat *status order customer*.



Gambar 3.29 Contoh tampilan *order details mock-up*

Gambar 3.29 berisi tanggal pemesanan untuk *pet grooming* dan lama penginapan untuk *pet hotel*, dan juga menampilkan harga masing masing paket yang sudah dipesan.

3.6 Kendala yang dihadapi

Kendala yang dihadapi selama menjalankan program kerja magang di PT Satwa Teknologi Utama adalah sebagai berikut:

1. Belum pernah menggunakan *angular* dan juga menggunakan bahasa pemrograman *typescript* sehingga harus mempelajari terlebih dahulu saat awal kerja magang dilakukan.
2. Tidak adanya mentor untuk dapat mengajari secara langsung di tempat kerja menyebabkan pembelajaran terhadap *angular* menjadi lebih lama.

3.7 Solusi atas Kendala

Solusi yang dapat dilakukan dari kendala yang dihadapi selama menjalankan program kerja magang:

1. Berusaha mencari sumber-sumber untuk mempelajari tentang *angular* dan *typescript*, juga bertanya kepada CTO mengenai apa saja yang harus dipelajari sehingga lebih cepat dalam mempelajari.
2. Melihat secara langsung *code* yang sudah dibuat dan kemudian mempelajari secara langsung untuk penggunaan setiap komponen yang ada pada repositori tersebut.