

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Program kerja magang yang dilakukan berada di posisi sebagai *IT Programmer* pada Metro Hospitals Group dengan deskripsi pekerjaan yang telah diberikan oleh *supervisor* dalam bentuk proyek untuk membuat *UI/UX* untuk *website* Metro Hospitals Group hingga ditugaskan menjadi *Front-end Engineer* yang bertugas merancang, membuat dan mengelola *website* dari Metro Hospitals Group tugas yang diberikan oleh *supervisor* diharapkan dapat membantu mengatasi masalah dan kenyamanan *user* dalam menggunakan *website* yang disajikan oleh perusahaan tersebut nantinya. *UI/UX* merupakan singkatan dari *User Interface* dan *User Experience*, *UX* adalah suatu proses mendesain produk yang sesuai dengan kebutuhan dari pengguna, ada beberapa aspek yang dibutuhkan di *UX* untuk memudahkan kebutuhan pengguna nantinya yaitu Struktur Desain, aspek Visual dan aspek interaksi dengan pengguna yang menjadikan seorang *UX* mempunyai tanggung jawab dalam jalannya produk sesuai kebutuhan dari pengguna, *UI* merupakan bagian dari *UX* dimana *UI* akan bertugas untuk membuat tampilan dari produk menjadi lebih menarik di mata pengguna lebih tepatnya akan berfokus kepada visualisasi yang diberikan kepada pengguna, visual yang diberikan oleh *UI* akan meliputi seperti Layout, animasi, warna dan tema dari sebuah produk. Sedangkan *Front – end Engineer* merupakan orang yang melakukan perancangan dari hasil *UI* yang telah dibuat menjadikan sebuah *system* melalui Bahasa pemrograman yang telah di tentukan

Dalam program magang yang dilakukan di Metro Hospitals Group ini dilakukan secara *full-time* di *office* dimulai dari hari senin hingga jumat oleh karena itu kebanyakan koordinasi yang diberikan oleh *supervisor* yaitu Bapak Franky Putra ataupun meeting yang diadakan oleh IT Corporate yaitu Bapak Yudi dilakukan full di *Office* yang bertempat di Menara Bpjamsostek, Jakarta Selatan.

3.2 Tugas dan Uraian Kerja Magang

Ada beberapa tugas yang diberikan atau yang ditargetkan selama melakukan program magang di Metro Hospitals Group berlangsung, dimulai dari tanggal 20 Februari 2023 hingga berakhir pada tanggal 19 Juli 2023. Kegiatan magang yang dimulai dari diberikan Training atau pengenalan dari *supervisor* terkait proyek yang ingin dikerjakan hingga tahap – tahap atau strategy apa yang akan di terapkan dalam merancang proyek tersebut dalam program magang yang dilakukan. Berikut gambaran kegiatan yang akan dilakukan selama program magang berlangsung di perusahaan tersebut.

Tabel 3. 1 Kegiatan magang

No	Judul Aktivitas	Mulai	Selesai
1	<i>Introduction</i>		
	<i>Introduction Workflow</i>	20 Februari 2023	28 Februari 2023
2	<i>Training</i>		
	<i>Training Laravel dan php</i>	01 Maret 2023	10 Maret 2023
	<i>Training secara mandiri</i>	13 Maret 2023	17 Maret 2023
3	<i>Prototype</i>		
	Membuat <i>Landingpage</i> sementara	20 Maret 2023	21 Maret 2023
	<i>Meeting</i> Terkait <i>Prototype</i>	24 Maret 2023	24 Maret 2023
	Membuat Final <i>Prototype</i>	27 Maret 2023	10 April 2023
	<i>Landing Page</i> website(Navbar, Banner, Spesialis, Paketmedis, Rumah sakit, artikel dan <i>footer</i>)	28 Maret 2023	28 Maret 2023

No	Judul Aktivitas	Mulai	Selesai
	Page semua <i>Specialist</i>	29 Maret 2023	31 Maret 2023
	Page Paket Medis	03 April 2023	05 April 2023
	Page Rumah Sakit, Page Artikel, Our Partner dan <i>footer</i>	06 April 2023	10 April 2023
	Page WhatsOn	10 April 2023	10 April 2023
	Presentasi Akhir dari <i>prototype</i>	10 April 2023	10 April 2023
4	<i>Development</i>		
	Menentukan Bahasa Pemogramanan	10 April 2023	11 April 2023
	Meeting	12 April 2023	12 April 2023
	Pembuatan page artikel <i>website</i>	13 April 2023	19 Mei 2023
	Pembuatan page <i>Whats'on website</i>	22 Mei 2023	15 Juni 2023

Kegiatan Program magang yang dilakukan pada perusahaan Metro Hospitals terdapat 4 bagian dalam pembuatan proyek untuk pencapaian magang yang dilakukan, diawali dengan introduction terhadap *workflow website* lama dari Metro Hospitals Group yang telah ada sebelumnya yang bertujuan agar nantinya kemana arah dari *website* yang ingin di bangun Kembali akan lebih baik dari *website* yang ada sebelumnya, setelah melakukan arahan dan pengenalan terhadap *workflow* dari *website* melanjutkan kepada tahap Training di tahap Training peran dari *Supervisor* sangatlah besar dikarenakan *supervisor* akan mengajarkan melatih

peserta magang terkait materi yang akan digunakan untuk pembangunan atau perancangan *website* nantinya, setelah melakukan beberapa minggu Training dilakukan meeting untuk melakukan atau membuat *prototype* yang bertujuan mempresentasikan terkait tampilan dan fungsional dari *website* yang akan dibangun apakah sesuai dengan keinginan dari kepala IT Corporate pada perusahaan tersebut. Setelah menentukan dan menyelesaikan beberapa revisi terkait desain *prototype* akhirnya final pada *prototype* yang akan digunakan untuk implementasi *website*. Setelah itu team IT *Programmer* meeting terkait strategi dan Bahasa pemograman apa yang akan digunakan untuk membangun *website* yang baru melalui referensi *prototype* yang telah dibuatkan.

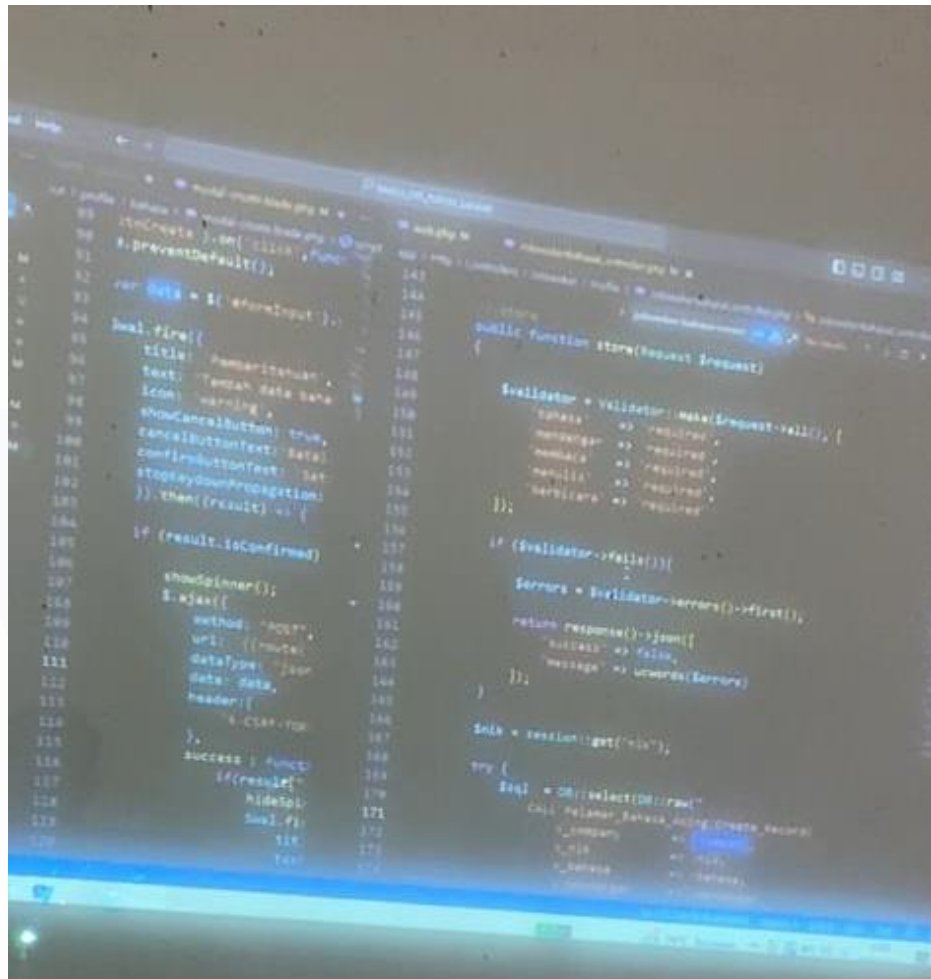
Project pada program magang 2023 ini yaitu membangun *website* yang Bernama *Website Metro Hospitals Group* dimana *website* ini merupakan *website* secara keseluruhan yang mencakup beberapa Rumah Sakit yang dimiliki oleh perusahaan, di dalam *website* ini akan menampilkan beberapa Rumah Sakit yang memiliki Fitur untuk melihat Jadwal dokter setiap rumah sakit dan melakukan *Appointment* jika *user* ingin melakukan pertemuan dengan dokter dari Rumah Sakit yang telah ditentukan, berbeda dengan *website* sebelumnya hanya menampilkan gambar dan lokasi dari rumah sakit saja tidak menampilkan layanan yang tersedia dari Rumah Sakit yang dimiliki oleh Metro Hospitals Group tersebut.

3.2.1 Introduction

Untuk mengawali program magang yang dilakukan di Metro Hospitals Group dilakukan pengenalan – pengenalan dari *supervisor* terkait workflow dari *website* yang telah dimiliki oleh perusahaan sebelumnya, seperti bagaimana alur dari *website* apa saja fitur yang dimiliki oleh *website*. Proses pengenalan ini dilakukan selama 1 minggu dimana dengan tujuan agar peserta magang dapat mengerti alur yang diinginkan dari perusahaan

3.2.2 Training

Setelah melakukan tahap pengenalan terkait dari *system* yang dimiliki oleh perusahaan, pada minggu berikutnya dilakukan



Gambar 3. 1 *Training* dalam program magang

sebuah *training*, ada 2 *Training* yang dilakukan pada program magang kali ini yaitu *Training Laravel* dan *Php* dan *Training* secara mandiri, *Training* dilakukan selama 2 minggu, *Training* yang dilakukan oleh *Supervisor* kepada peserta magang bertujuan agar peserta magang dapat mengerti proses dan cara penggunaan Bahasa pemrograman dan *framework* yang digunakan oleh perusahaan tersebut.

a) *Training Laravel dan PHP*

Training pertama yang dilakukan pada program magang kali ini yaitu memahami dan mempelajari cara penggunaan dari *framework Laravel* bagaimana cara

menarik data yang diinginkan menggunakan sebuah *Routes* yang dimiliki oleh *Laravel*, *Laravel* merupakan sebuah *Framework* yang banyak digunakan hingga sekarang dengan menggunakan Bahasa pemrograman PHP dengan tujuan membangun *website* akan lebih maksimal *website* yang dinamis yang dapat dihasilkan dengan menggunakan *framework Laravel* ini, dan pada saat training di ajarkan cara menggunakan controller pada *Laravel* yaitu struktur MVC (*Model View Controller*) secara garis besar pada saat training peserta magang diajarkan cara menarik sebuah data menggunakan *Routes* dan Controller

Training ini merupakan cara penggunaan *swal fire* di *Laravel*, kemudian pada training ini diajarkan juga cara memanggil function dari *Framework Laravel*, cara penggunaan *Route*.

b) Training Secara Mandiri

Setelah melakukan Training dengan *supervisor* terkait *website* yang telah ada sebelumnya yang menggunakan *Laravel* dan *php*, peserta program magang di berikan waktu 5 hari untuk membaca dan mengenali lebih dalam terkait yang sudah di jelaskan oleh *supervisor* terkait *system* yang sudah berjalan sebelumnya di perusahaan ini. Tujuan dari diberikannya waktu pembelajaran secara mandiri kepada peserta magang di perusahaan ini peserta diharapkan dapat menambah pengetahuan terkait tentang project *Laravel* dan *php* yang ada di perusahaan ini sehingga dikemudian hari jika diberikan *jobdesc* dengan tugas yang sama dengan apa yang di pelajari peserta magang dapat mengerti apa yang di infokan dan apa yang

harus dikerjakan untuk memberikan hasil yang maksimal kepada perusahaan.

3.2.3 *Prototype*

Pada akhir bulan maret setelah melakukan *Training* mulai memasuki masa dalam pembuatan *Prototype* untuk *website* dari Mtero Hospitals Group, pengerjaan *website* ini dilakukan beberapa minggu dimulai dari tanggal 20 Maret 2023 hingga selesai melakukan pembuatan *website* tanggal 10 April 2023 tanggal tersebut sudah termasuk dari masukan hingga revisi oleh Team IT yang terdapat di dalam perusahaan, ada beberapa part yang harus dikerjakan dalam pembuatan *prototype* ini, diantaranya adalah :

1) Pembuatan *Landingpage* Sementara

Setelah melakukan beberapa minggu untuk melakukan training dan mengetahui workflow *system* yang berjalan pada perusahaan ini, peserta magang dan juga team IT *Programmer* perusahaan dikumpulkan untuk melakukan meeting terkait *design* untuk *interface* dari *website* yang baru untuk perusahaan ini pada saat ini peserta magang untuk diminta membuat suatu *prototype website* untuk Metro Hospitals Group dengan biru, kuning dan juga putih, pada tanggal 20 Maret 2023 hingga 21 Maret 2023 peserta magang mulai melakukan rancangan untuk *design* dari *website* sesuai dengan hasil meeting yang telah dilakukan.

Gambar dibawah merupakan *prototype landing page website* sementara dimana terdiri dari 3 bagian yaitu bagian Navbar yang merupakan navigasi untuk memudahkan *user* menentukan pilihan kemudian kedua yaitu body yang berisi about us untuk menampilkan sejarah dari perusahaan serta ada terdapat *button* untuk *user* melakukan *appointment* atau membuat jadwal dengan dokter yang diinginkan serta dapat

melihat jadwal dokter hingga memesan untuk jadwal penyuntikan vaksin yang diinginkan oleh *user* dan bagian terakhir sama seperti dengan *website* pada umumnya yaitu bagian *footer* sama seperti kesimpulan dari sebuah *website*



Gambar 3. 2 *Prototype website* sementara

yang menunjukkan secara inti dari *website* tersebut seperti alamat, contact dari *website*.

2) Meeting Terkait *Prototype*

Setelah membuat *protoype* sementara kemudian pada tanggal 24 Maret 2023 dilakukan meeting Kembali Bersama Team IT dan juga Team Graphic Designer, dimana pada meeting kali ini peserta magang melakukan presentasi dari *prototype landing page* secara garis besar untuk tampilan *website*, kemudian dilakukan revisi oleh Kepala IT Corporate. Meeting kali ini berisikan kepala IT Corporate yaitu bapak Yudi memberikan refrensi 2 *website* rumah sakit untuk menentukan design *website* terbaru dari Metro Hospitals Group dimana kesimpulan dari referensi tersebut kepala IT Corporate ingin design *website* Metro Hospitals Group yang terbaru bertemakan Elegan dan *User Friendly*, *website* tidak memiliki banyak warna sehingga *website* terlihat bersih dan elegan dan juga tentunya harus mudah digunakan oleh *user* untuk melakukan *appointment* dan lain – lainnya.

3) Membuat Final *Prototype*

Diambil dari hasil meeting yang dilaksanakan Bersama team IT pada tanggal 24 Maret 2023 untuk final *prototypenya* dan juga masukan dari team IT dan Team design apa saja yang perlu ditambahkan di dalam design *prototype website* yang akan dilakukan pembuatannya, dan juga terdapat masukan dari kepala IT Coorporate memberi referensi *website* yang ingin di ambil sehingga dapat memudahkan tujuan dari pembuatan design *prototype* yang dilakukan. Dalam pembuatan design *prototype* ini terdapat beberapa bagian *website* di dalam *prototype* ini yang dibuat, diantaranya adalah :

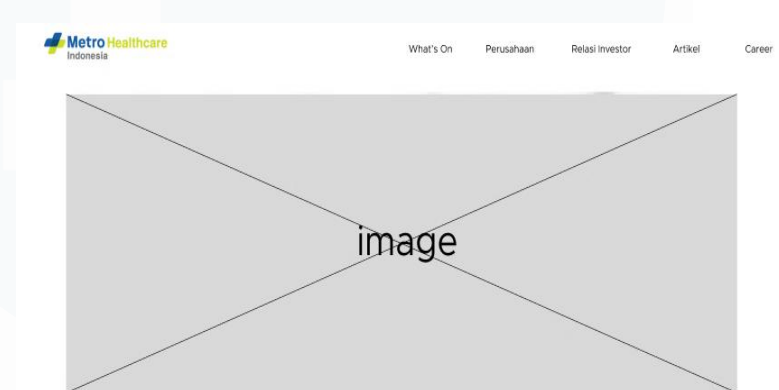
4) Landing Page

Bagian utama dari sebuah *website* yaitu landing page dimana di dalam *landing page* terdapat navigasi Navbar untuk memudahkan *user* yang berfungsi untuk mengarahkan atau pentunjuk untuk *user*,



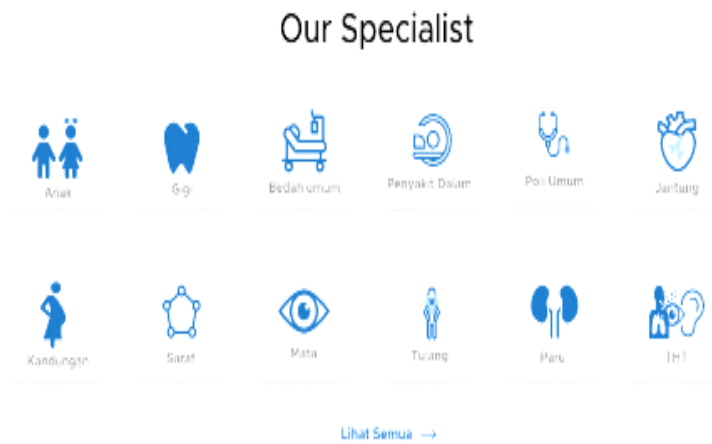
Gambar 3. 3 Prototype navbar

Setelah membuat navbar yang kedua dalam sebuah *website* untuk menunjukkan suatu informasi dan program apa yang sedang ada di dalam *website* suatu perusahaan yaitu banner.



Gambar 3. 4 Prototype Banner

Di dalam *banner* terdapat beberapa foto yang akan digunakan dengan fungsi *Caraosell* yang bertujuan gambar tersebut akan berganti setiap waktu yang telah ditentukan sesuai kesepakatan Team IT dan design di dalam perusahaan. Kemudian terdapat bagian *specialist* dimana di dalam *specialist* ini terdapat *appointmen* untuk melakukan penjadwalan dengan dokter tergantung dari *specialist* yang diinginkan oleh *user*, contohnya *specialist* anak, gigi, kandungan gingga THT (telinga, hidung dan tenggorokan), setiap *specialist Link* yang telah diarahkan kepada *specialist* yang sesuai *user* melakukan “Click” pada *image specialist* tersebut hampir sama dengan *Button*.



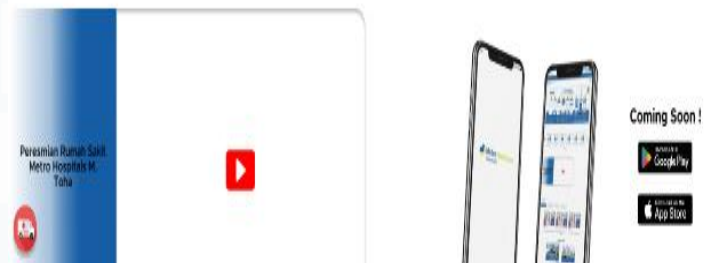
Gambar 3. 5 *Prototype menu specialist*

Kemudian terdapat *UI* untuk paket medis , dimana didalam paket medis ini terdapat beberapa paket yang disediakan oleh rumah sakit yang dimiliki oleh Metro Hospitals Group.



Gambar 3. 6 *Prototype menu paket medis*

Dibawah dari tampilan *ui* paket medis terdapat tampilan *ui* yang menampilkan rumah sakit yang dimiliki oleh perusahaan kemudian terdapat juga suatu tampilan *UI* video yang langsung diarahkan ke *Link* youtube, kemudian ada *button* untuk emergency yang akan menampilkan nomor emergency dari setiap rumah sakit yang dimiliki oleh perusahaan yaitu *button* dengan image mobil ambulance.



Gambar 3. 7 *Prototype* menu rumah sakit dan *banner* youtube

Terdapat *UI* untuk menampilkan artikel – artikel yang sedang ada di dalam perusahaan ataupun rumah sakit seperti tentang Kesehatan.



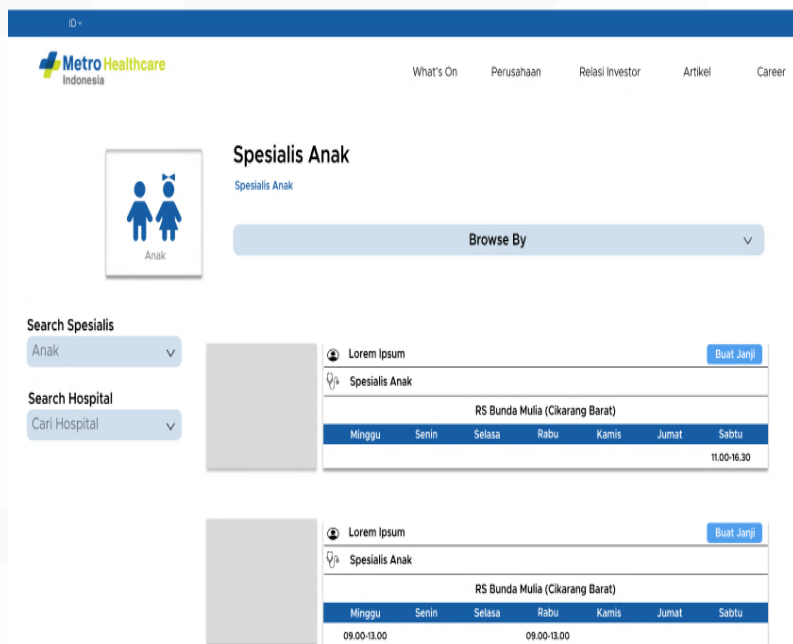
Gambar 3. 8 *Prototype* Menu artikel

Sebelum Tampilan *UI Footer* dari *website* terdapat tampilan *UI* yang menampilkan beberapa *partner* yang telah bekerja sama dengan Metro Hospitals Group

dengan berbentuk Logo masing – masing dari *partner* yang bersangkutan.

5) *Page Specialist*

Di dalam setiap *specialist* yang terdapat di dalam *website* akan mengarahkan atau terdapat *Link* yang akan menuju ke Page baru yang berisi list semua dokter dengan *specialist* yang telah dipilih sebelumnya, di dalam *specialist user* juga dapat mencari dokter yang di inginkan untuk membuat jadwal di rumah sakit, Ketika *user* sudah berada di page *appointment user* tidak perlu Kembali ke menu *home* untuk mengakses *specialist* yang lainnya dikarenakan di dalam setiap page *specialist* terdapat fitur *filter* untuk menyesuaikan rumah sakit atau *specialist*, contohnya seperti dibawah ini :



Gambar 3. 9 *Prototype detail specialist*

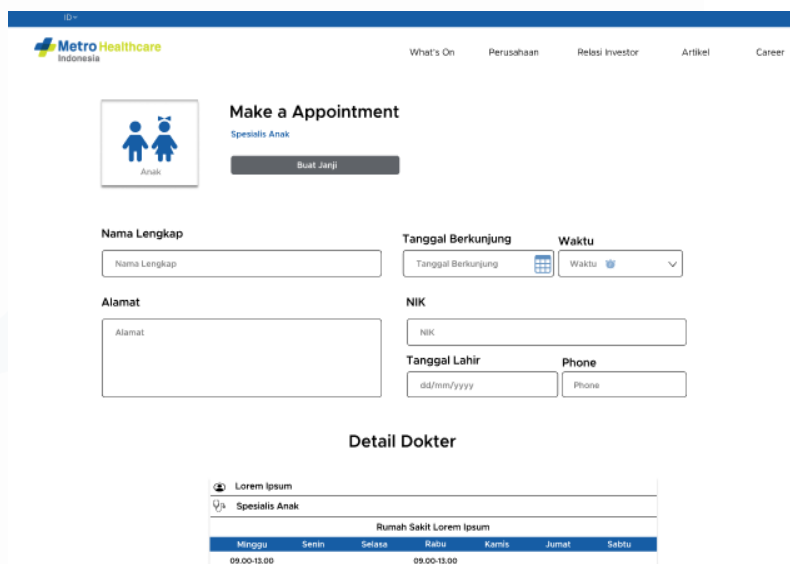
Yang akan ditampilkan di *UI* untuk pertama kali *user* memilih *specialist* adalah seluruh dokter yang bersangkutan dengan *specialist* yang dipilih, kemudian untuk memudahkan *user*

dalam pencarian *By Name*, pada *UI* ini menyediakan fitur *Browse By* abjad dimana terdapat fitur abjad A-Z untuk membantu *user* menemukan nama dokter sesuai huruf depan dari dokter tersebut



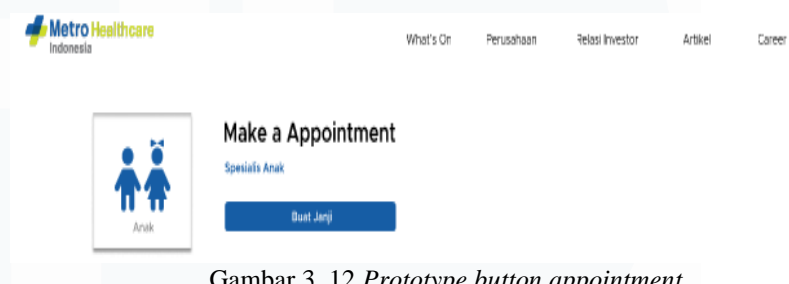
Gambar 3. 10 *Prototype fitur filter* berdasarkan nama dokter

Setelah *user* memilih dokter yang diinginkan untuk dilakukan *appointment* membuat jadwal temu dengan dokter pilihan *user*, *user* juga bisa menentukan rumah sakit yang ingin ditunjukkan dengan menggunakan menu *filter* cari rumah sakit yang terdapat di sebelah *UI* dari jadwal dokter yang tersedia.



Gambar 3. 11 *Prototype detail appointment*

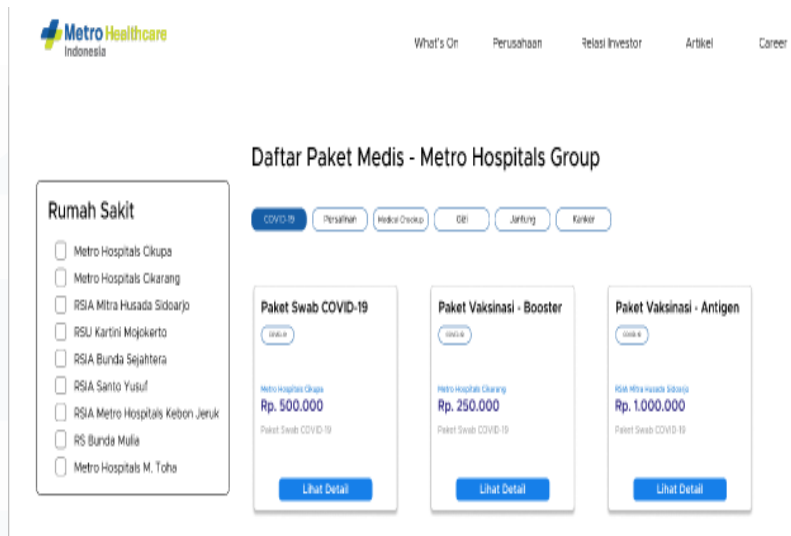
Setelah menentukan pilihannya *user* dapat masuk mengisi formular yang perlu dimasukan untuk sebelum melakukan *appointment* seperti Nama lengkap, Tanggal ingin berkunjung, waktunya, alamat dari *user* , NIK, tanggal Lahir dan terakhir Phone, setelah mengisi semuanya *button* BUAT JANJI akan aktif menjadi bewarna biru kemudian *user* dapat melanjutkan *appointment* nya.



Gambar 3. 12 *Prototype button appointment*

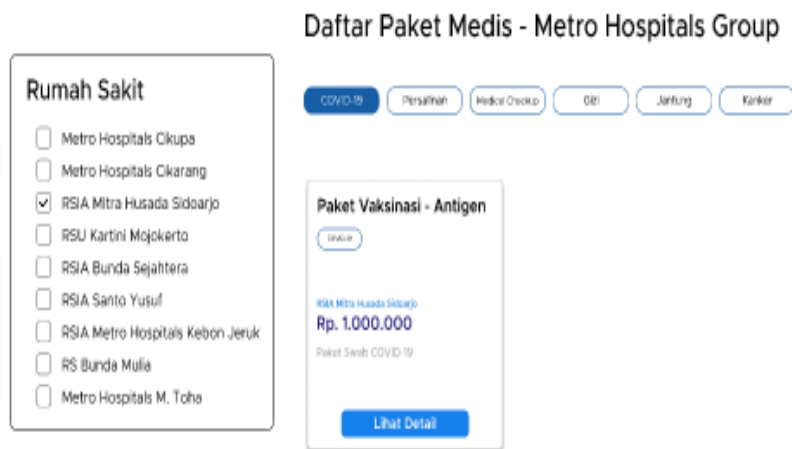
6) *Page* Paket Medis

Setelah *ui* dari *Specialist* kemudian pada *website* ini terdapat *ui* dari paket medis yang tersedia dari rumah sakit – rumah sakit yang dimiliki oleh metro hospitals group, sama halnya dengan seluruh rumah sakit yang ada semuanya pasti memiliki paket medis seperti promo untuk swab dan promo paket bersalin ibu hamil. Tujuan dari *ui* paket medis ini untuk memudahkan *user* untuk mencari paket yang ingin digunakan dengan *ui* yang mudah digunakan sehingga tidak menyulitkan *user* dalam penggunaannya.



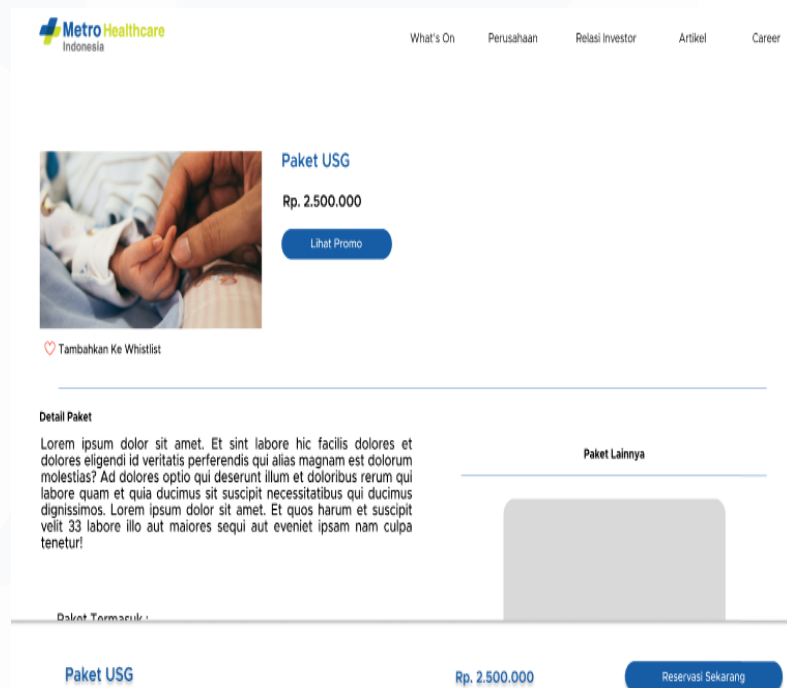
Gambar 3. 13 *Prototype* paket medis

Setelah *user* memilih paket medis pada *landing page* dari *website* maka *user* akan diarahkan kepada page yang terdapat menu seluruh paket medis yang tersedia di seluruh rumah sakit yang dimiliki oleh Metro Hospitals Group, dan untuk memudahkan *user* tidak kembali ke *landing page* lagi untuk memilih paket medis yang lainnya, disini *user* dapat menggunakan *Tag* yang telah disediakan di dalam *ui* untuk memilih paket medis lainnya, kemudian untuk memilih rumah sakit yang ditentukan *user* dapat melakukan ceklist pada menu *ui* pada kiri paket medis untuk menentukan rumah sakit yang ingin di datangi apakah terdapat paket medis yang ingin ditunjukkan, contohnya seperti gambar di bawah.



Gambar 3. 14 *Prototype menu filter rumah sakit*

Website akan melakukan *filter* dengan rumah sakit yang telah dipilih oleh *user* dan paket medis yang tersedia untuk rumah sakit tersebut.



Gambar 3. 15 *Prototype menu paket*

Setelah itu *user* dapat melakukan reservasi untuk paket medis yang diinginkan kemudian terdapat *detail* dari paket medis yang di berikan, dan terdapat *button* Promo untuk melihat

apakah terdapat Promo yang diberikan oleh pihak rumah sakit dengan paket medis yang diinginkan oleh *user*.

The image shows a web browser window displaying a form titled "Reservasi Sekarang" (Book Now) on the Metro Healthcare Indonesia website. The form contains several input fields: "Nama Lengkap" (Full Name), "NIK" (National ID Number), "Tanggal Reservasi" (Reservation Date), "Waktu" (Time), "Alamat" (Address), "Tanggal Lahir" (Date of Birth), and "Phone". A "Reservasi Sekarang" button is located at the bottom of the form. The website header includes the Metro Healthcare Indonesia logo and navigation links for "What's On", "Perusahaan" (Company), "Relasi Investor" (Investor Relations), "Artikel" (Articles), and "Career".

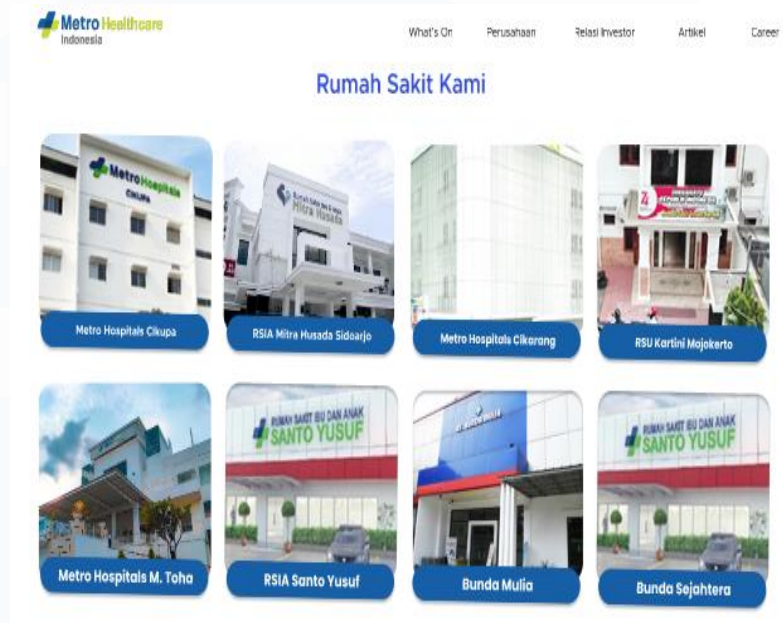
Gambar 3. 16 Prototype appointment paket

Untuk melakukan reservasi paket medis, *user* akan diarahkan ke *page* formulir untuk mengisi data diri sebelum melakukan reservasi, kemudian *user* akan dihubungi oleh pihak dari rumah sakit terkait *appointment* yang dipesan oleh *user*.

7) *Page* Rumah Sakit, *Page* Artikel dan *footer*

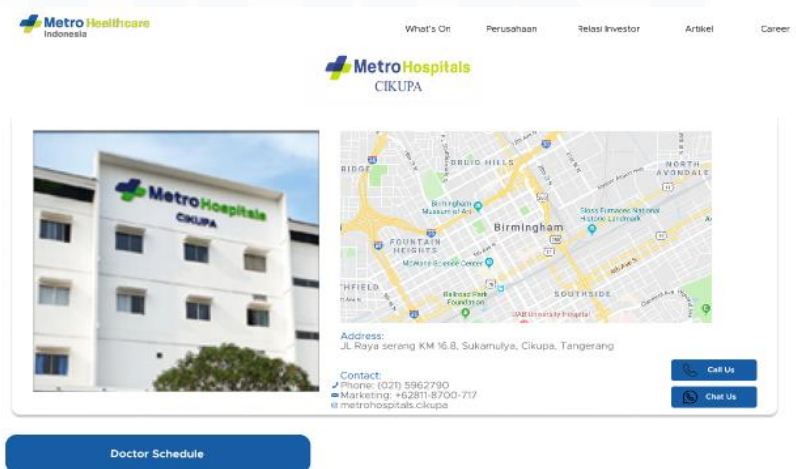
Pada *page* Rumah sakit pada *website* ini menampilkan secara garis besar dari rumah sakit yang dimiliki oleh Metro Hospitals Group kemudian terdapat *detail* dari rumah sakit hingga layanan dan fasilitas apa saja yang terdapat pada *detail* rumah sakit yang dimiliki oleh Metro Hospitals Group, sedangkan *page* artikel merupakan *page* yang berisikan sama halnya seperti berita hingga tips untuk Kesehatan dan terakhir yaitu *page footer* yaitu berisikan info garis besar dari *website* ini

merupakan sama halnya dengan *website* pada umumnya yang memiliki *footer* berisikan informasi.



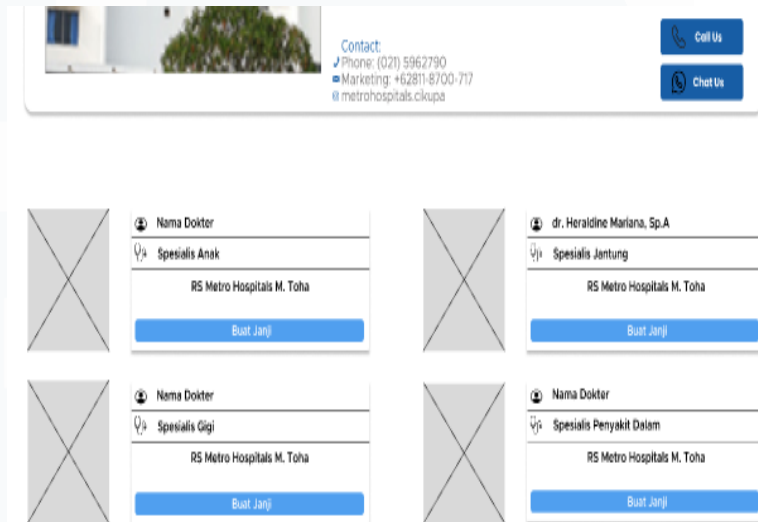
Gambar 3. 17 *Prototype* Menu rumah sakit

Ketikan *button* “Lihat Semua” yang terdapat di *landingpage* di *Click* tampilan *ui* akan diarahkan kepada page baru yang berisikan semua rumah sakit yang terdapat pada Metro Hospitals Group sehingga memudahkan *user* untuk melihat rumah sakit apa saja yang tersedia dan yang dimiliki oleh Metro Hospitals Group.



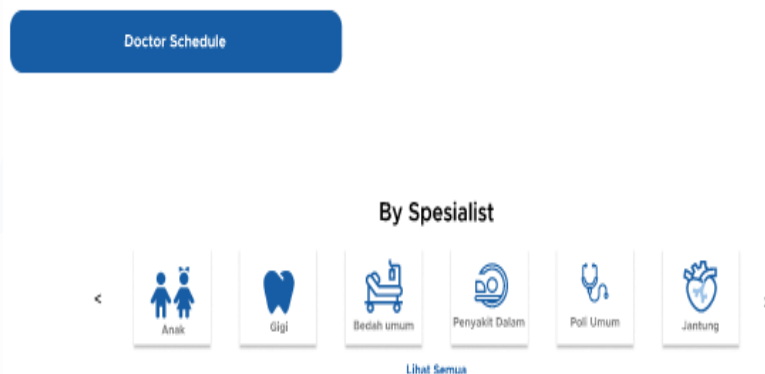
Gambar 3. 18 *Prototype* Detail Rumah Sakit

User dapat melihat *detail* dari rumah sakit yang diinginkan hingga lokasi dan contact dari rumah sakit yang diinginkan, lokasi yang dapat langsung direct ke google maps sehingga user dapat mudah mengikuti lokasi rumah sakit yang diberikan. Kemudian terdapat *Button* “*Doctor schedule*”,



Gambar 3. 19 Prototype dokter Specialist rumah sakit

Button tersebut mereload *website* sehingga menampilkan jadwal dokter yang sedang bertugas sesuai dari rumah sakit yang dipilih oleh *user*.



Gambar 3. 14 Prototype By Specialist di detail rs

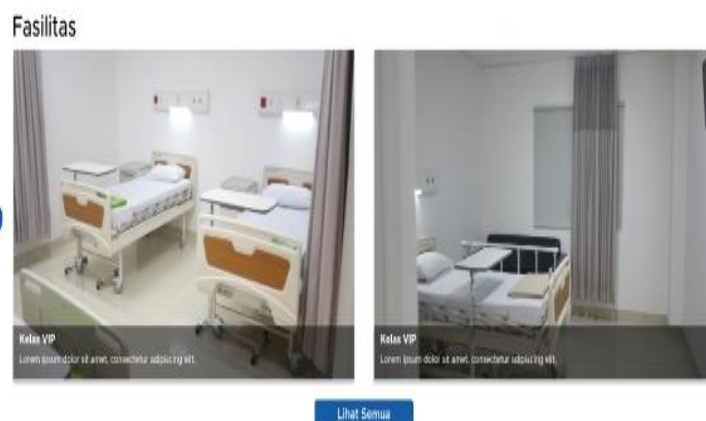
Dan terdapat *specialist* di bawah *button* yang membedakan *specialist* pada page ini ialah *specialist* akan di *filter* sehingga dokter *specialist* yang ditampilkan akan sesuai dengan rumah

sakit yang dipilih oleh *user*, sehingga dapat memudahkan *user* dalam masalah navigasi dan tidak perlu Kembali ke menu *home* atau *landing page* untuk melihat *specialist*. Isi di dalam menu *specialist* ini sama saja dengan *specialist* pada *landing page*, *filter* yang membedakan dari keduanya, untuk *specialist* yang terdapat di dalam rumah sakit akan di direct ke *specialist* sesuai dengan rumah sakit yang telah dipilih.



Gambar 3. 15 *Prototype* Layanan

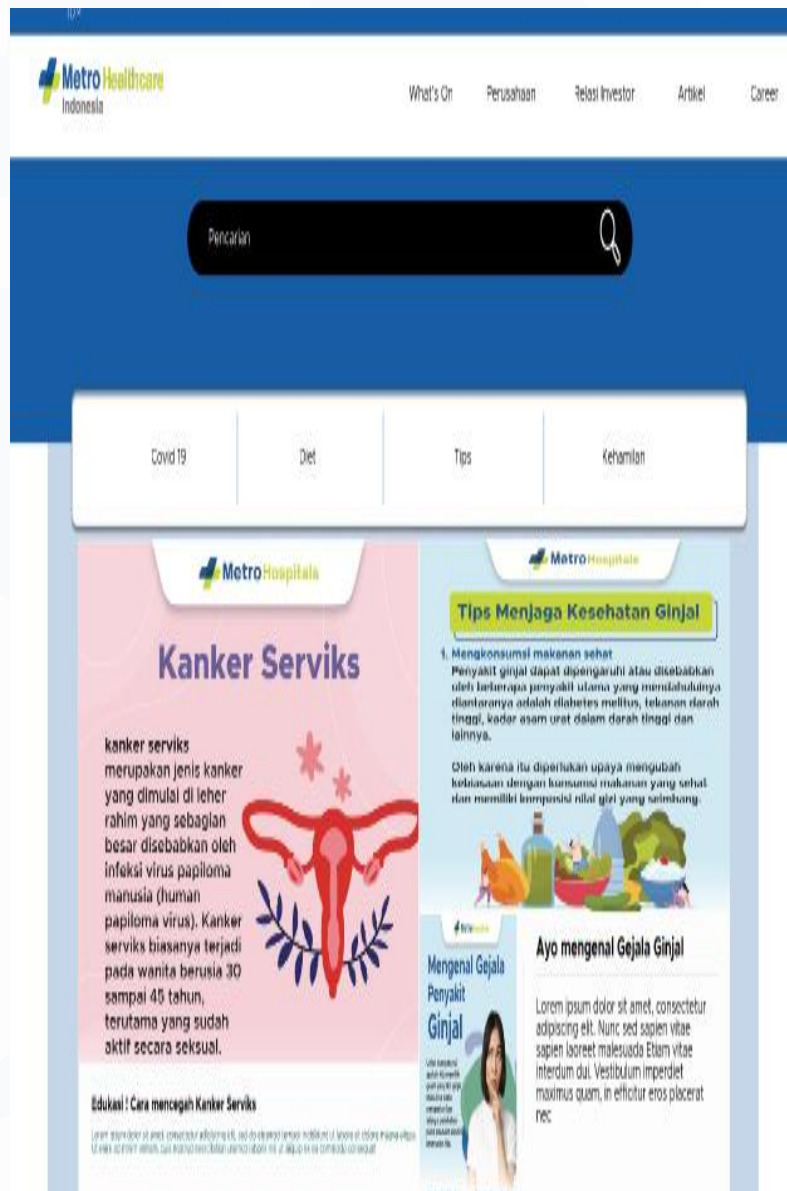
Kemudian terdapat *UI* layanan apa saja yang sedang ada pada rumah sakit tersebut yang berbentuk *carrousel* yang memiliki durasi beberapa detik kemudian layanan yang ada akan berubah dengan beberapa layanan yang tersedia.



Gambar 3. 16 *Prototype* Fasilitas rumah sakit

Kemudian tersedia *ui* yang menunjukkan fasilitas kamar dan rumah sakit yang dipilih oleh *user* dan juga dapat melihat *detail* dari rumah sakit tersebut. Kemudian sama halnya

dengan *website* rumah sakit yang ada , pada *website* ini terdapat fitur artikel yaitu berisikan info tentang Kesehatan atau jenis gejala penyakit yang sering terjadi untuk dewasa hingga sama anak – anak



Gambar 3. 17 *Prototype Landingpage* artikel

Ini merupakan tampilan *ui* dari page artikel yang berisikan empat kategori utama yang dapat diakses oleh *user*, yang dimana kategori tersebut dapat berubah dengan mengikuti

perkembangan yang sedang terjadi, Ketika salah satu gambar atau artikel di click maka akan di arahkan kehalaman *detail* artikel yang bersangkutan sehingga menampilkan secara rinci tentang artikel tersebut, contohnya seperti



Gambar 3. 18 *Prototype* detail artikel

Page ini merupakan tampilan dari *detail* salah satu artikel yang terdapat pada menu artikel di *website* Metro Hospitals Gorup, sehingga terlihat jelas dari judul hingga ke tanggal *upload* dari artikel tersebut.

Kemudian setelah halaman *page* artikel, pada *website* ini terdapat menu *ui* yaitu *Our Partner* dimana disini akan ditampilkan *partner – partner* yang bekerja sama dengan perusahaan sehingga *user* dapat melihat nya

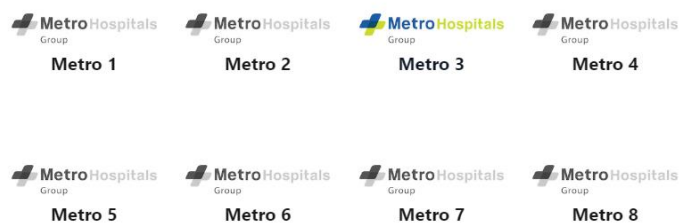
Partner Kami



Gambar 3.19 *Prototype partner*

Pada menu *ui* ini semua logo dari *partner* yang bekerja sama akan di tampilkan secara tanpa warna, kemudian Ketika *user* mengarahkan cursor atau click salah satu logo, maka akan memunculkan warna sebagai bentuk tampilan yang menarik dimata *user* yang menggunakannya

Partner Kami



Gambar 3.20 *Prototype HOVER partner*

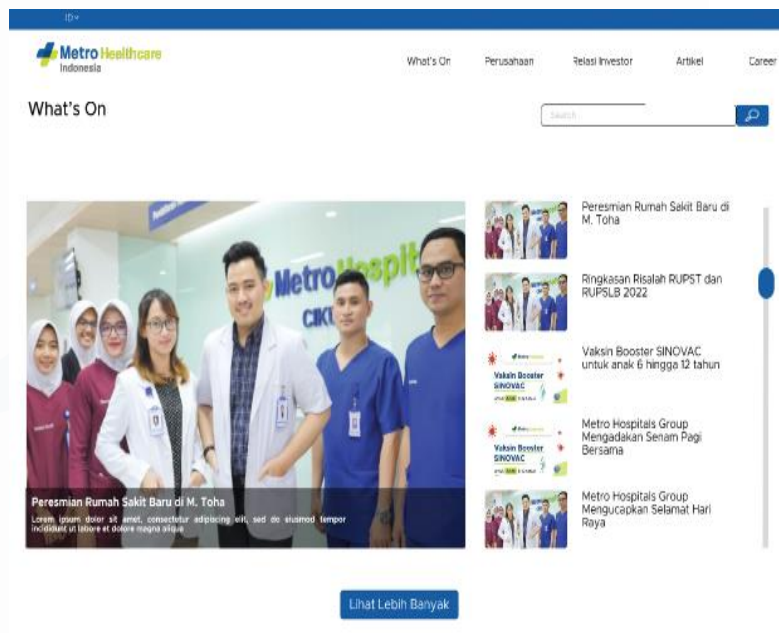
Bagian terakhir dari *prototype* yang dibuat merupakan *footer* yang memiliki info secara garis besar dari perusahaan seperti, Contact yang dapat dihubungi, Location Head *Office* dari perusahaan hingga social media dari perusahaan akan di tampilkan dalam *footer* yang merupakan bagian akhir dari suatu *website*.



Gambar 3. 21 *Prototype footer*

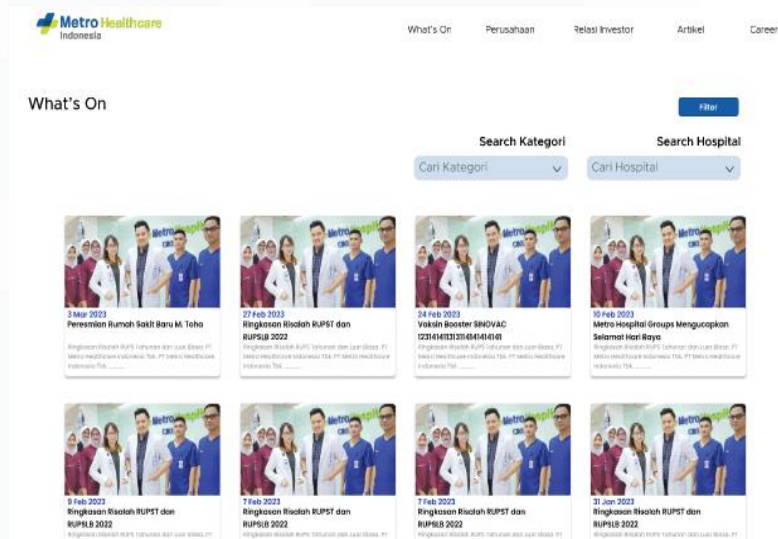
8) Page *Whats'On*

Page *whats'on* ini hampir sama halnya dengan *page* dari artikel yaitu berisi informasi, yang membedakan *page Whats'on* dengan artikel yaitu informasi pada *whats'on* berisikan informasi yang berkaitan dengan perusahaan seperti hubungan kerja dengan perusahaan lain, peresmian rumah sakit dan masih banyak hal lagi yang berkaitan dengan perusahaan.



Gambar 3. 22 *Prototype Landingpage whatson*

Kemudian untuk melihat lebih banyak dari berita terkini yang ada pada perusahaan, *user* dapat menggunakan *button* berwarna biru kemudian akan diarahkan ke *page* yang baru yaitu berisi semua dari berita



Gambar 3. 23 *Prototype tag whatson*

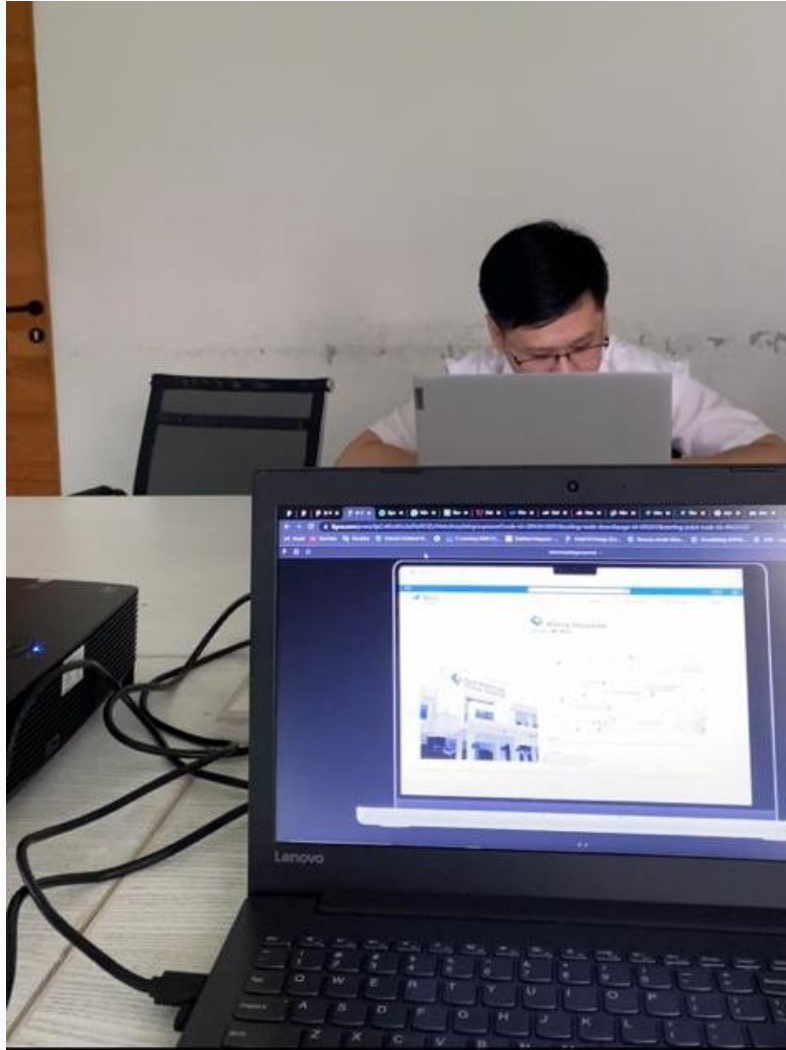
Terdapat fitur *filter* kategori dan rumah sakit yang membantu memudahkan *user* untuk mencari informasi terkait rumah sakit ataupun kategori yang diinginkan.



Gambar 3. 24 *Prototype Detail Artikel*

Untuk *detail* dari *whats 'on* tampilan *ui* yang ditampilkan sama dengan *article* sehingga memberi kesan dari segi *simple* dalam penggunaan dari *website*.

9) Presentasi Final *Website*



Gambar 3. 25 *Presentasi final prototype*

Setelah menyelesaikan pembuatan *prototype* yang dilakukan beberapa minggu, kemudian melakukan presentasi dengan Kepala IT Corporate dan di dampingin oleh *supervisor* yaitu Bapak Franky, setelah melakukan presentasi dan tidak memiliki revisi lagi, kepala IT mesetujui untuk melanjutkan ke

progress berikutnya yaitu pembuatan *website* dari Metro Hospitals Group yang telah dibuat *prototype* nya.

3.2.4 *Development*

Setelah menyelesaikan design *Prototype* untuk *website* kemudian masuk kepada project keempat yaitu menyiapkan strategi hingga menggunakan bahasa pemrograman apa dalam membangun *website* untuk Metro Hospitals Group, untuk menentukan apa yang selanjutnya akan dilakukan team it melakukan pertemuan untuk menentukan pilihan cara membangun *website* tersebut.

A. Menentukan Bahasa Pemrograman

Team IT *Programmer* yang terdapat pada Metro Hospitals Group terdiri dari 3 orang, kemudian melakukan meeting untuk menentukan *Framework* dan *style* apa yang akan digunakan untuk membangun *website* tersebut, setelah melakukan research team setuju untuk menggunakan *Framework* NextJs yaitu *framework* Javascript dari ReactJs semacam penyempurnaan dari ReactJs dan untuk *style* team setuju untuk menggunakan *Tailwind* CSS.

1. NextJs



Gambar 3. 26 Prototype Logo Nextjs

NextJs merupakan sebuah *Framework* JavaScript yang dibuat secara garis besar untuk ReactJs untuk menyempurnakan ReactJs [3]. terutama dalam hal *Rendering* NextJs terbilang sangat bagus. NextJs dapat dikatakan merupakan *framework* full-stack yang berguna untuk memabangun *framework* dari segi *front-end* hingga *back-end*. Terdapat dua cara kerja dari NextJs yaitu :

- *Server Side Rendering*, berfungsi ketika *rendering* server akan mengubah package React dan JavaScript menjadi HTML setiap tampil dibuka di dalam browser
- *Static Site Generator*, setiap browser memanggil halaman tersebut, server tidak perlu men-generate HTML. dikarenakan, server melakukan generate halaman HTML pada tahap *build*, hal tersebut memungkinkan *programmer* menggunakan jenis *pre-rendering* yang berbeda tiap halaman *website* sesuai dengan kebutuhan.\

2. *Tailwind* CSS



Gambar 3. 27 Prototype Logo Tailwind

Tailwind CSS merupakan suatu *framework* CSS yang memiliki basis *utility* untuk melakukan pembuatan *UI* atau tampilan dari *website* [4]. Keuntungan menggunakan *tailwind* css yaitu memiliki proses pembangunan *UI* yang lebih cepat sehingga memudahkan modifikasi terhadap *UI* sesuai kebutuhan dari *user* dan juga memudahkan membuat *website* menjadi *responsive* sehingga dapat digunakan diberbagai media seperti *smartphone*, *tablet*, *laptop* hingga *pc*.

B. Meeting

Setelah menentukan *Framework* dan *style* yang akan digunakan dalam pembangunan *website* ini, team IT dikumpulkan untuk melakukan Meeting dimana di dalam meeting ini berisikan *deadline* yang diberikan kepala IT dalam pembangunan *website* ini hingga *live*, pada awalnya team *front – end* mengajukan untuk waktu pembuatan 3 bulan tetapi menurut *supervisor* waktu 3 bulan tidak mencukupi untuk membangun *website* hingga *live* digunakan, setelah meeting dilakukan waktu *deadline* yang diberikan oleh kepala IT yaitu selama 5 hingga 6 bulan pengerjaan.

C. Pembuatan page artikel *website*

Dalam project pertama yang diberikan kepada saya yaitu, *supervisor* memberikan tanggung jawab kepada saya untuk membangun page dari artikel dengan menggunakan data dami terlebih dahulu.

Dimulai dari membuat *landing page* dari article yang menampilkan semua artikel dari semua kategori.

```
const inter = Inter({ subsets: ["latin"] });

export async function getStaticProps() {
  var articles = []
  return {
    props: {
      articles: articleList,
    },
  };
}
```

Gambar 3. 28 Code index utama landingpage

Pada *index articles* langsung dipanggil dari *json* yang dibuat untuk secara dami terlebih dahulu.

```
export default function IndexPage({articles}) {
  return (
    <div>
      <LanguageProvider>
      <Head>
        <title>Metro Healthcare Indonesia</title>
        <meta name="description" content="Generated by create next" />
        <link rel="icon" href="/favicon.ico" />
      </Head>
      <Header />
      <Articles dataArticle={articles}/>
      { /* .sort((a, b) => 0.5 - Math.random()) */ }
      <Footer />
    </LanguageProvider>
  </div>
);
}
```

Gambar 3. 29 code dataArticle

Pada awalnya ingin menggunakan javascript sort random untuk menampilkan data secara random untuk *landing page* dari wesbite, tetapi untuk sementara

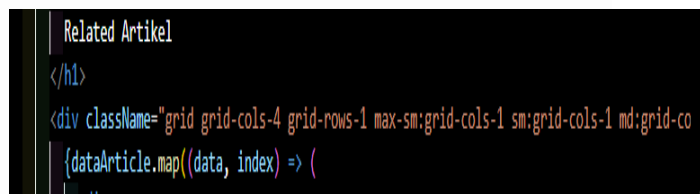
menggunakan data dari *Json* yang dibuat secara sendiri terlebih dahulu. `dataArticle={articles}` dimana berfungsi merubah variabel `dataArticle` yang mempunyai isi `articles` sedangkan `articles` sudah terdapat isi data *json* di dalamnya yang bernama `articlelist`.



```
articles.jsx x [slug].js [tag].js index.js
src > modules > articles > articles.jsx > Articles
1 import React from "react";
2 import Image from "next/image";
3 import Link from "next/link";
4
5 export const Articles = ({ dataArticle }) => {
6   return (
7     <div>
```

Gambar 3. 30 Code Pemanggilan `dataArticle` dari *Index Js*

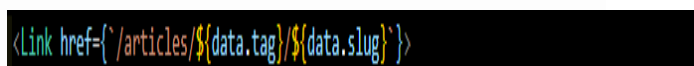
Pada *landing page* `articles.jsx` memanggil `dataArticle` yang sudah diinisialisasi yang dalamnya memiliki data *json* dari `articlelist`.



```
Related Artikel
</h1>
<div className="grid grid-cols-4 grid-rows-1 max-sm:grid-cols-1 sm:grid-cols-1 md:grid-co
  {dataArticle.map((data, index) => (
```

Gambar 3. 31 Code Loopingan data

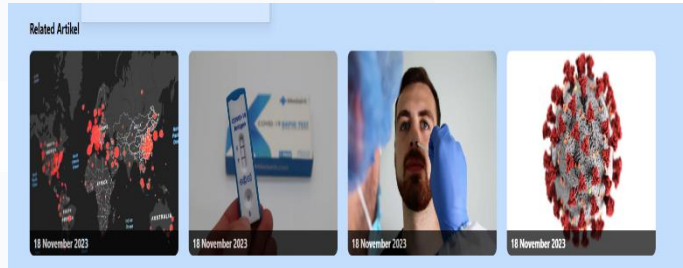
Kemudian mengatur layout untuk menampilkan data dari `dataArticle`, (`.map`) merupakan looping sehingga `dataArticles` yang di panggil pada line ini yaitu akan di looping sebanyak data yang tersedia di *JSON* dengan membentuk 4 kolom dan 1 baris



```
<Link href={` /articles/${data.tag}/${data.slug}`} >
```

Gambar 3. 32 Code penarikan data Tag dan Slug

Link tersebut mengambil data dari *tag* dan *slug* yang sesuai dengan apa yang dipilih *user* dan ditarik data dari data *json* yang dimiliki.



Gambar 3. 33 Menu Tag Artikel

Merupakan *output* dari looping yang diambil dari *dataArticle* yang diloop menggunakan *.map*. kemudian untuk menggunakan *tag* pada menu articles seperti *tag covid, kehamilan* dan lainnya disini menggunakan *tag.js* atau sebagai id.

```
26 export async function getStaticProps({ params }) {
27   var dataTagArticle = articleList.filter((data) => data.tag == params.tag );
28
29   return {
30     props: {
31       dataTagArticle,
32     },
33   };
34 }
```

Gambar 3. 34 Code DataTagArtikel

Melakukan inisialisasi pada variabel *dataTagArticle* untuk memanggil data dari data *json* yang bernama *articleList* kemudian dilakukan *filter* data sesuai *tag* yang di *click* oleh *user*, kemudian *mereturn* hasil dari *filter* yang bernama *dataTagArticle*.

```

export default function TagPage({ dataTagArticle, }) {
  return (
    <>
    <LanguageProvider>
      <Header/>
      <ArticleTag dataTagArticle={dataTagArticle}/>
      <Footer />
    </LanguageProvider>
  </>
)
}

```

Gambar 3. 35 Menampilkan dataTagArtikel

Setelah itu melakukan export kepada *dataTagArticle* sehingga dapat di tampilkan di index website dengan cara *dataTagArticles={dataTagArticle}*.

```

[slug].js [tag].js article-tag.jsx X
src > modules > articles > tag > article-tag.jsx > ArticleTag
1 import React from "react";
2 import Image from "next/image";
3 import Link from "next/link";
4
5 export const ArticleTag = ({ dataTagArticle }) => {
6 return (

```

Gambar 3. 36 Code pemanggilan artikel tag dataTagArtikel

Kemudian setelah melakukan inialisasi pada id atau *tag.js dataTagArticle* di export atau di panggil pada *module article-tag.jsx*.

```

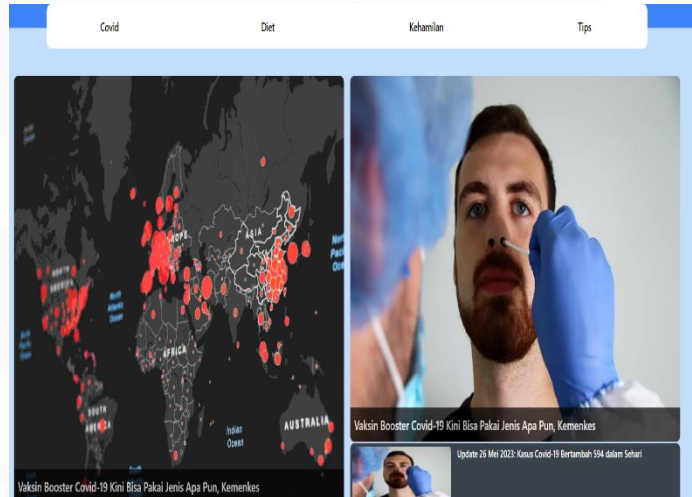
<Link
  href={` /articles/${dataTagArticle[3].tag}/${dataTagArticle[3].slug}`}

```

Gambar 3. 37 Code pemanggilan data Tag dan Slug

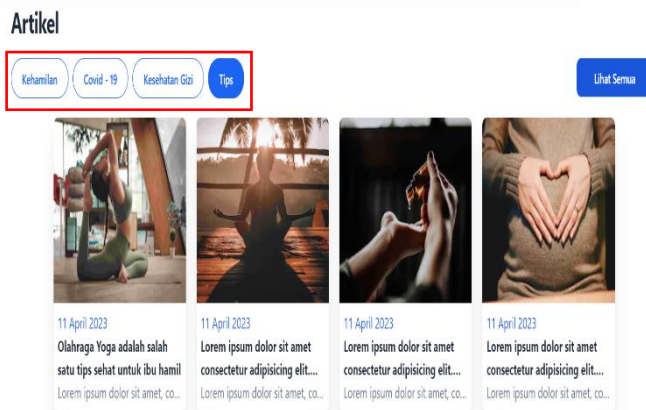
Sama halnya dengan *dataArticle.map* tetapi pada *tag* tidak menggunakan looping melainkan artikel yang ditampilkan di awal hanya data yang pasti dan tidak bersifat temporary sehingga memutuskan langsung

memanggil data dengan *array* sesuai data yang digunakan ataupun yang ingin di tampilkan.



Gambar 3. 38 Menu *Tag*

Gambar diatas merupakan ouput dari modul *article-tag.jsx* yang memanggil data menggunakan variabel yang telah diinisialisasi pada *tag.js* yaitu *dataTagArticle*.



Gambar 3. 39 Menu *Filter* menu *home* artikel

Untuk melakukan *rendering tag* yang terdapat di dalam kolom merah yaitu menggunakan *useState*.

```
export default function HomeArticle({ articles }) {  
  const { language } = useContext(LanguageContext);  
  
  const [filterCategory, setFilterCategory] = useState([]);  
  .....
```

Gambar 3. 40 code Home artikel

Code diatas berfungsi untuk menampung data.

D. Pembuatan Page *Whats'on*

Setelah pengerjaan page artikel untuk perusahaan dari mulai *landing page* awal hingga sampai ke *detail* dari *website*, tahap kedua pekerjaan yang diberikan kepada peserta yaitu membuat *page* hingga fungsi dari *page Whats'on*. *Page* atau menu ini merupakan menu yang menampilkan informasi tetapi bukan terkait tentang dunia Kesehatan melainkan *page* atau menu ini menampilkan berita yang berkaitan dengan perusahaan ataupun unit rumah sakit yang bersangkutan dengan kegiatan yang ada bagian di dalam perusahaan Metro Hospitals Group, di dalam menu *page Whats'On* ini hampir sama dengan menu artikel konsep pengerjaannya seperti terdapat *landing page* awal yang menampilkan seluruh berita dari perusahaan hingga unit rumah sakit yang berkaitan dengan perusahaan, kemudian terdapat nantinya yang membedakan *page* ini dengan artikel yaitu menu *filter* sesuai dengan kebutuhan dari *user* dan juga tentunya terdapat menu *detail* dari berita yang akan di tampilkan di dalam *page*

Whats'on ini. Terdapat 3 bagian dalam pengerjaan ini yaitu pertama membuat data *json* sesuai dengan instruksi pembimbing lapangan menggunakan data *json* sementara sehingga data yang ada akan digunakan dalam penampilan di dalam *website* di localhost dalam pengerjaan *website* lebih tepatnya menu *Whats'On* untuk *website* Metro Hospitals Group.

```
export async function getStaticProps() {
  var whatson = []
  return {
    props: {
      whatson: whatsonList,
    },
  };
}
```

Gambar 3. 41 Code *Whats'on*

Sama halnya dengan menu dari article pada menu *landing page whatson* ini pertama menggunakan *getStaticProps* yaitu sebuah function, yang pertama dilakukan membuat variabel kemudian fungsi dari variabel ini untuk memanggil atau menampung parameter dari data *json* yang bernama *whatsonList*, *whatson* list sendiri berisi data yang nantinya berguna untuk isi dari menu *whatson* ini.

```

4 export default function IndexPage({whatson}) {
5   return (
6     <div>
7       <LanguageProvider>
8         <Head>
9           <title>Metro Healthcare Indonesia</title>
10          <meta name="description" content="Generated by create next
11          <link rel="icon" href="/favicon.ico" />
12        </Head>
13        <Header />
14        <Whatson datawhatson={whatson}/>
15      </LanguageProvider>
16    </div>
17  );
18 }

```

Gambar 3. 42 Menampilkan *data* di *website*

Setelah dilakukan inisialisasi terhadap variabel tersebut, kemudian variabel tersebut dipanggil Kembali yang dimana untuk ditampilkan pada halaman *landing page* atau index dari menu *whatson* ini, dan varibel tersebut di deklarasi dalam bentuk yang baru untuk di panggil di *module* yang terdapat di *landing page*.

```

export const Whatson = ({ datawhatson }) => {
  return (
    <div>
      <div className="grid grid-rows-1 grid-cols-2 mx-20 pt-32">
        <h2 className="font-semibold text-2xl">Whats On</h2>
      </div>
    </div>
  );
}

```

Gambar 3. 43 *code* menarik *data* ke *module*

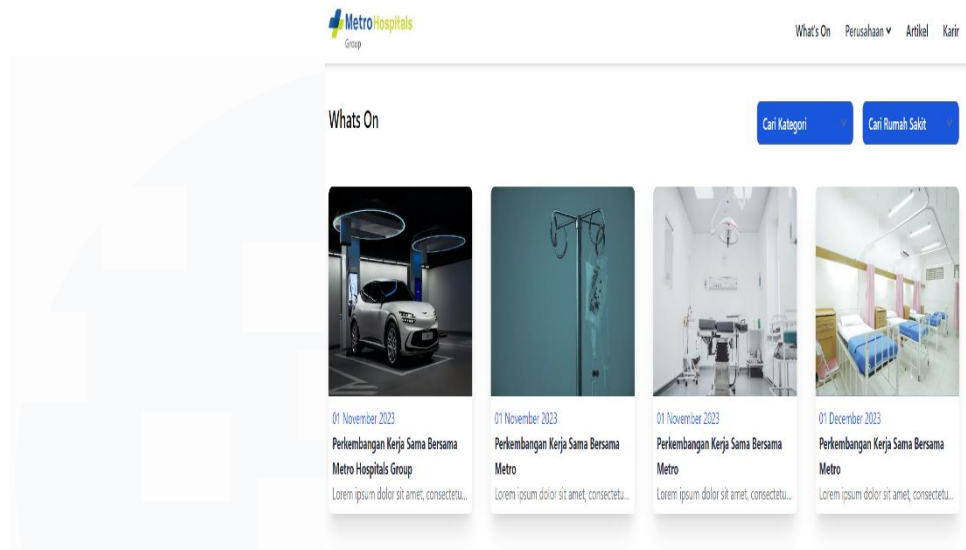
Kemudian *datawhatson* tersebut di panggil ke dalam *module* yang terdapat di *landing page* untuk *whatson* nantinya yang berguna untuk memanggil data dari

json untuk di tampilkan di halaman depan landingpage dari menu ini.

```
<div className="lg:flex flex-row md:flex-col px-20 pt-20">
  <div>
    <Link href={` /whats-on/${datawhatson[0].tag}/${datawhatson[0].tag}`}>
      <Image
        className="lg:h-[32rem] md:h-[25rem] max-md:h-[16rem]"
        src={` /metro/whatson/${datawhatson[0].src}`}
        width={500}
        height={500}
        alt=""
      />
    </Link>
  </div>
</div>
```

Gambar 3. 44 *code* memanggil *data*

Code diatas merupakan salah satu contoh pemanggilan data dari *json* yang sudah dinamai menjadi *datawhatson* yang sudah dirubah pada *index js* sebelumnya. Setelah pembuatan *landing page* untuk *whats'on* selesai kemudian dilanjutkan untuk pembuatan *tag* dari *whats'on* yang dimana berguna untuk menampilkan *data what's on* secara meluruh, seperti



Gambar 3. 45 Menu tag *Whats 'On*

Tag yang dimaksud adalah *tag* per judul ataupun per item berita yang ditampilkan contoh gambar diatas yaitu 4 *tag* sesuai dengan *tag* berbeda beda ada dengan tema masing – masing contohnya *tag* bisnis perusahaan, *tag* Kesehatan rumah sakit, *tag* peresmian rumah sakit.

```
export async function getStaticProps() {
  var dataTagWhatson = whatsonList.reverse();

  return {
    props: {
      dataTagWhatson,
    },
  };
}
```

Gambar 3. 46 Code *dataTagWhatson*

Sama halnya dengan yang dilakukan di *landing page* yaitu membuat dengan menggunakan *function*

getStaticProps kemudian membuat variabel yang bernama *dataTagWhatson* untuk mendapatkan *data* dari *Json* yaitu *whatsonList* kemudian variabel tersebut *direturn* Kembali.

```
export default function TagPage({ dataTagWhatson, }) {  
  
  return (  
    <>  
    <LanguageProvider>  
      <Header />  
      <WhatsonTag dataTagWhatson={dataTagWhatson} />  
      <Footer />  
    </LanguageProvider>  
  </>  
)  
}
```

Gambar 3. 47 Code menarik untuk di tampilkan di *module*

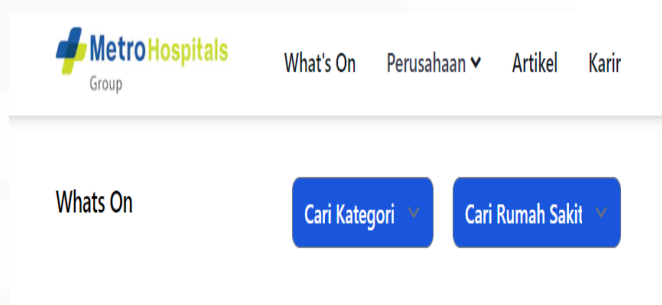
Kemudian dipanggil lagi *dataTagWhatson* tersebut untuk di tampilkan di halaman index dari *tag* yang akan tampil di *websitenya*.

```
export const WhatsonTag = ({ dataTagWhatson }) => {
```

Gambar 3. 48 Memanggil *data Tag Whats'on*

Sama pada sebelumnya *dataTagWhatson* yang sudah berisi *data Json* tersebut dipanggil ke *module* yaitu *whatson-tag.jsx* yang berfungsi untuk menjalankan pemanggilan *data – data* yang terdapat di dalam *whatsonList*. Tetapi yang membedakan *tag whatson* dengan *tag* artikel yaitu pada *tag whatson* terdapat

filter sesuai kategori dan rumah sakit dari berita mana yang akan di tampilkan seperti contohnya,



Gambar 3. 49 *Filter By Kategori dan rumah sakit*

Pada halaman *tag whatson* ini terdapat menu *filter* yang dimana akan *filter* sesuai dengan kategori dan rumah sakit yang memiliki atau terkait dengan berita yang akan di tampilkan.

```
export const WhatsonTag = ({ dataTagWhatson }) => {
  const [filterCategory, setByFilterCategory] = useState([]);

  const filterByCategory = (tagCategory) => {
    setByFilterCategory(
      dataTagWhatson.filter((item) => item.tag === tagCategory)
    );
  };
};
```

Gambar 3. 50 *code useState*

Pada menu ini menggunakan konsep dari *Hooks*, *Hooks* berfungsi untuk melakukan state management dan *side effects* di dalam *function component* yang bertujuan untuk memudahkan *programmer* dalam penulisa *code*. Ada banyak macam hooks seperti *useState*, *useEffect*, *useRef*, *useMemo* dan juga *useCallback*. Tetapi pada menu *tag whatson* ini hanya menggunakan dua *Hooks* yaitu *useState* dan

useEffect. *Filtercategory* yang terdapat dalam codingan tersebut merupakan item pertama yang merupakan statenya kemudian item kedua yaitu *setByFilterCategory* yang berfungsi untuk mengubah state tersebut, kemudian membuat *function* untuk *filter* data *whatson* dimana ouputnya adalah data dengan *tag* yang sesuai pada parameter yang bernama *tagCategory*

Sebelum menggunakan *useState* dan yang lainnya harus melakukan import seperti

```
import React from "react";
import Image from "next/image";
import Link from "next/link";
import { useContext, useEffect, useState } from "react";
```

Gambar 3. 51 Code *useState* dan *useEffect*

```
useEffect(() => {
  if (filterCategory.length === 0) {
    setByFilterCategory(dataTag\whatson);
  }
}, []);
```

Gambar 3. 52 code *useEffect*

Hooks kedua yang digunakan yaitu *useEffect* dimana fungsi dari *useEffect* yaitu dijalankan Ketika ada perubahan di dalam *array*, tetapi untuk *tag whatson* ini *array* dikosongkan dimana ia akan memanggil data sebelum data dilakukan perubahan oleh *filter*.

01 November 2023

Perkembangan Kerja Sama Bersama

Metro



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer fringilla libero a turpis viverra vehicula. Sed ac pellentesque ligula, ac pharetra justo. Donec ut erat vitae tortor accumsan convallis. Aenean ornare commodo purus sed semper. Sed fermentum et mi ac condimentum. Etiam sed sagittis ex, in imperdiet urna. Cras iaculis ante et purus molestie lacinia. Mauris id dolor et velit tempus imperdiet sit amet vel arcu. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per

Kategori

Bisnis

Hospitals

Related Post



Gambar 3. 53 *related*

Berita yang terdapat pada menu *Related Post* merupakan berita yang memiliki kesamaan yang di ambil dari *tag* seperti contohnya Ketika *user* membuka detail berita yang dengan kategori bisnis maka pada menu detail artikel *Related Post*nya akan menampilkan *tag* yang berkaitan dengan bisnis yang di ambil data melalui halaman *tag*.

```

export async function getStaticProps({ params }) {
  var whatson = whatsonList.find(
    (data) => data.slug == params.slug
  );

  var Releated = whatsonList.filter(
    (data) => data.tag == whatson.tag
  );
}

```

Gambar 3. 54 *var releated*

Berikut merupakan cara pemanggilan data yang di ambil dari data dengan variabel *whatson*, variabel *whatson* memiliki *data json* dengan nama *whatsonList*, kemudian membuat variabel baru yang Bernama *releated* dengan memanggil hal yang sama yaitu data *json* dari *whatsonList* dengan *function filter*, *filter* akan mencari sesuai perintah seperti contohnya yaitu *data.tag* yang mengambil data dari variabel *whatson*.

```

export default function PostPage({ whatson, Releated }) {
  const router = useRouter();
  -----

  return (
    <>
    <LanguageProvider>
    <Header />
    <WhatsonDetail
      Date={whatson.Date}
      title={whatson.title}
      src={whatson.src}
      description={whatson.description}
      Releated={Releated}
    />
  )
}

```

Gambar 3 .55 *releated* tampilan ke *website*

Kemudian di panggil lagi untuk di tampilkan pada tampilan *website*.

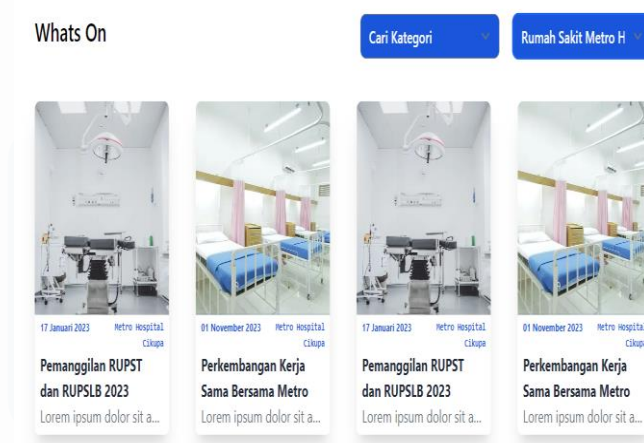
```

<div className=" overflow-y-scroll max-h-[28rem] grid grid-rows-none grid-cols-1 max-sm:grid-rows-1" style={{ overflow: 'auto' }}>
  {related.map((data, index) => (
    <div
      className="block overflow-hidden rounded-lg mx-2"
      key={index}
    >
      <Link href={` /whats-on/${data.slug}`} >
        <Image
          className="shadow rounded-lg hover:scale-110 duration-300 max-md:max-h-20"
          src={` /metro/whatson/${data.src}`}
          width={500}
          height={500}
          alt=""
        />
      </Link>
    </div>
  ))}
</div>

```

Gambar 3. 56 looping data terhadap var related

Sama seperti yang sebelumnya menggunakan looping dari variabel *related* sebelumnya yang melakukan *filter* terhadap data yang terdapat di *json*. Dan menggunakan *style overflow* untuk menjadi *scroll*.



Gambar 3. 57 filter byrs

Untuk melakukan *filter* terhadap rumah sakit untuk memunculkan rumah sakit yang berkaitan yaitu menggunakan *usestate*.

```
const filterByRS = (RS) => {
  setByFilterCategory(dataTagWatson.filter((item) => item.id === RS));
};
```

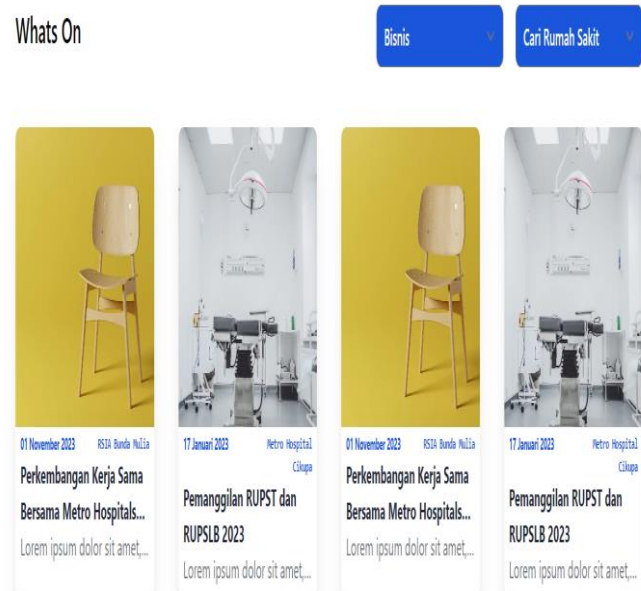
Gambar 3. 58 code by rs

menggunakan *filter* untuk mencari *data* yang dengan id yang terdapat pada json.

```
<select
  className="text-white font-semibold bg-blue-700 rounded-lg lg:w-52 md:w-44 sm:w-32 max-s
  onChange={(e) => filterByRS(e.target.value)}
  >
  <option defaultValue>Cari Rumah Sakit</option>
```

Gambar 3. 59 ubah *value*

filterByRS akan di lempar kembali kedalam menu *filter by rs* dimana dia akan merubah target value sesuai dengan id yang ditunjukkan.



Gambar 3. 60 filter kategori

Sama halnya dengan *filter* yang dilakukan pada *filter by* rumah sakit dimana sama sama menggunakan *filter* dan memanggil *data whatsonList* di dalamnya yaitu *data Json* yang ada di dalamnya.

3.3 Kendala yang Ditemukan

Terdapat beberapa kendala yang ditemukan pada saat mengikuti program magang untuk memabangun *system Website* untuk Metro Hospitals Group, diantaranya adalah :

- A. Penggunaan *framework* baru yaitu *nextjs* merupakan sebuah *framework* javascript yang sebelumnya saya belum pernah mengetahui ataupun menggunakannya sehingga membuat saya sedikit kesulitan dikarenakan harus mempelajari *framework* dari awal
- B. Dikarenakan pembangun *website* yang dilakukan hanye dengan 3 orang IT hal itu membuat kesulitan dikarenakan masalah pembagian yang extra akan diberikan pada setiap it yang ada dalam pembangunan *website* ini

3.4 Solusi atas Kendala yang Ditemukan

Terdapat solusi untuk mengatasi yang dialami selama mengikuti Pembuatan *website* untuk Metro Hospitals Group, telah dilakukan cara untuk mengatasi beberapa masalah yang dihadapi, diantaranya yaitu :

- A. Mempelajari *framework javascript* yang telah di tetapkan oleh Team IT perusahaan yaitu *Next Js* untuk pembuatan *website* ini, seperti membaca dari dokumentasi dari *framework javascript* yaitu *nextjs* yang digunakan kemudian membaca *function – function* apa saja yang terdapat di dalam pemograman *Next Js* ini.
- B. Dikarenakan hanya terdapat 3 Team IT *programmer* termasuk saya dan 1 Kepala IT Corporate sehingga untuk mengatasi masalah yang ada, Team melakukan pembagian tugas apa saja yang harus dikerjakan dan menjadi

tanggung jawab per orang sehingga membuat masalah yang ada sebelumnya dapat teratasi

