

## BAB 3

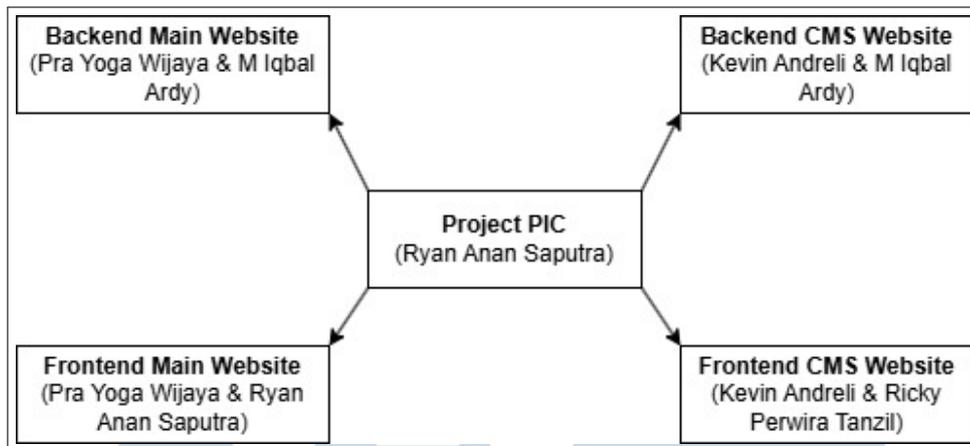
### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Organisasi

Selama program kerja magang di PT Siloam International Hospitals Tbk, Berperan sebagai *Intern Fullstack Developer* di divisi website. Selama program kerja magang dibimbing langsung oleh Stevani Lilyani Saputri selaku supervisor sekaligus Tech Lead pada divisi website. Lalu terdapat juga beberapa anggota divisi yang juga membantu dalam membimbing dan mengarahkan selama proses magang berlangsung.

Metode kerja yang digunakan oleh tim website siloam adalah *agile method* berupa *scrum*, dimana proyek diatur menjadi serangkaian sprint. Tim harus berkomitmen untuk menyelesaikan serangkaian pekerjaan yang telah direncanakan selama sprint tersebut. Sebelum sprint dimulai, *scrum master* akan membagikan dan menjelaskan *task* kepada tim, *management task* dilakukan melalui *tools* Jira. Lalu, tim diberikan waktu untuk eksplorasi terkait *task* tersebut dan setelahnya tim perlu memberikan estimasi tenggat waktu *task* tersebut dapat diselesaikan dalam 1 sprint tersebut (dalam hari). Setelah itu, akan diberikan *design dan Product Requirement Document (PRD)* lebih lanjut terkait dengan *task* yang diberikan. Tahap berikutnya adalah proses *development* dari tim. Setelah itu, dilakukan testing oleh *developer* dan tim-tim terhubung terkait dengan proyek seperti (QA, Produk, dan Marketing), serta tech lead akan memberikan *review*, jika *approved* dari tim-tim terkait maka akan diluncurkan ke website *production*.

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



Gambar 3.1. Struktural atau kedudukan pada *project Patient Centered Care*

Selama Pengerjaan *Project Development Patient Centered Care* menu Spesialisasi/SubSpesialisasi tim terdiri dari 1 orang yang berperan sebagai *Project PIC*, 2 orang yang berperan sebagai *fullstack developer*, dan 1 orang yang berperan sebagai *backend developer*, serta 1 orang yang berperan sebagai *frontend developer*. Selama proses *development* berlangsung terjadi beberapa kali perubahan tim dalam menggarap menu Spesialisasi/Subspesialisasi. Dalam pengembangan menu Spesialisasi/Subspesialisasi ini digunakan Jira sebagai alat yang membantu dalam manajemen proyek. Terdapat *parent project* yang akan di assign kepada *Project PIC* dan *child project* dari parent project yang akan di *assign* kepada developer. Komunikasi antar developer dan *project PIC* dilakukan melalui WhatsApp dan Microsoft Teams.

Setiap harinya selama periode *project* akan dilakukan *meeting*, yaitu di pagi hari dan di sore hari. Pada meeting ini, PIC melakukan *update* terkait progres dari *project* kepada scrum master, tim produk, dan Tech Lead, yang nantinya akan mendapatkan *feedback* yang dapat membantu project. Kemudian, pada beberapa hari sekali akan diadakan meeting sore hari untuk para developer memberikan *update* terkait pengerjaan pada hari tersebut kepada *project PIC*. Pada *meeting* ini PIC juga akan memberikan *feedback* kepada para developer dan juga membantu menyelesaikan permasalahan ketika developer mengalami kendala dalam pengerjaan *project*.

### 3.2 Tugas yang Dilakukan

Selama proses kegiatan magang berlangsung mengemban peran sebagai *Fullstack Developer* pada PT Siloam International Hospitals Tbk memiliki beberapa tugas utama yang perlu dilakukan, diantaranya:

1. Mempelajari lebih dalam *flow*, *logic*, dan *defined styling* pada website dari sisi *frontend* dan *backend*.
2. Melakukan pengembangan / *enhancement*, salah satunya pengembangan menu Spesialisasi/Subspesialisasi baik dari segi tampilan maupun fungsionalitas pada project tersebut.
3. Melakukan *Bug Fixing* pada project yang telah selesai dikerjakan, ataupun penemuan bug oleh QA.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1 terbagi dalam *scope* mingguan.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan research terkait end-to-end(E2E) testing script menggunakan Cypress. Melakukan pengerjaan terkait E2E testing script pada website main siloam di halaman-halaman seperti <i>Profile Form, Out Patient Department Form, Pre Regist Form, dan Search Doctor</i> .
2	Melakukan enhancement pada <i>project product change logs</i> di menu website Admin Siloam. Mengerjakan tracker di halaman OPD/cari dokter dan detail dokter. <i>Fixing bug</i> halaman-halaman pada menu Healthpedia.
3	Membuat E2E testing script menggunakan Cypress pada website Admin Siloam. Membuat Visualisasi data terhadap trackter yang telah dibuat sebelumnya menggunakan dashboard Kibana. <i>Fixing bug</i> menu Healthpedia
Lanjut pada halaman berikutnya	

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

Minggu Ke -	Pekerjaan yang dilakukan
4	<i>Enhancement UI</i> dan <i>fixing bug</i> menu Healthpedia. <i>Fixing Bug</i> pada <i>project product change logs</i> . <i>Enhancement</i> visualisasi kibana.
5	<i>Fixing issue</i> Healthpedia dan <i>development change request</i> menu Healthpedia.
6	Website <i>improvement</i> dengan tools Sonarcloud. <i>Fixing issue</i> Healthpedia. Memulai <i>development</i> menu baru pada halaman <i>Corporate Governance</i> . Layouting dan melakukan integrasi API pada menu-menu baru tersebut (Sustainability Journey, Sustainability Report, dan Materiality Topics)
7	<i>Layouting</i> dan <i>API Integratiron</i> dan halaman <i>Corporate Governance</i> menu sustainability strategy. <i>Fixing bug</i> Healthpedia. <i>Improvement</i> Sonarcloud pada website main.
8	Melakukan <i>rechecking</i> terkait <i>improvement</i> Sonarcloud yang telah dilakukan. <i>Fixing bug</i> menu baru Corporate Governance. Melakukan Refactor code untuk <i>redirect by content type</i> dihalaman informasi-siloam.
9	<i>Rechecking improvement</i> Sonarcloud, <i>fixing bug</i> menu sustainability. <i>Fixing issue production expand null reservation date</i> di Pesan janji temu dokter. Melakukan Research terkait tools Builder.io
10	Mencoba membuat simple landing page dengan tools yang ada di Builder.io dan mempelajari integrasi builder dengan component yang dimiliki di repository Siloam.
11	Mengerjakan Task untuk update submenu di Corporate Governance dan melanjutkan research builoder dan mencoba melakukan integrasi API siloam dengan Builder.io
12	<i>Fixing Analytics</i> pada artikel dan enhance kode analytics artikel. Enhance query tag pada pencarian healthpedia menjadi params tags untuk keperluan SEO. Memulai pengerjaan menu Spesialisasi/Subspesialisasi
Lanjut pada halaman berikutnya	

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

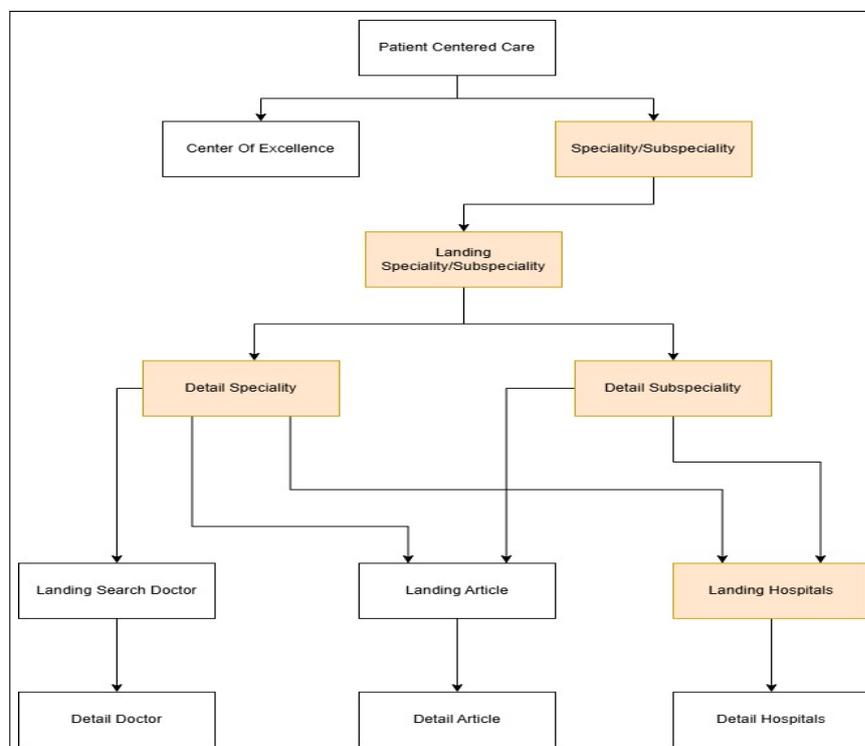
Minggu Ke -	Pekerjaan yang dilakukan
13	<i>Layouting</i> halaman Detail Spesialisasi dan SubSPesialisasi. Melakukan Enhancement GET API detail spesialisasi dan GET API hospital nearby untuk keperluan data yang akan digunakan pada halaman Detail Spesialisasi.
14	Melakukan Enhancement GET API detail subspecialisasi. Melakukan Repombakan terkait dengan pengerjaan API detail subspecialisasi dengan pivot table yang baru.
15	Menambahkan <i>mapping field</i> spesialisasi pada index Elastic Enhance API hospital nearby by multiple speciality and sub speciality. Create new API get list hospital by speciality and sub speciality and order by order di table pivot. Enhancement FE main halaman Detail Spesialisasi/Subspecialisasi. Enhancement API BE CMS Publish detail articles to return slug by current language. ( <i>SEO Improvement Task</i> )
16	Enhancement API BE CMS publish get list articles, related articles, and popular articles to return slug by current language. <i>Research and development</i> Schema Markup untuk halaman artikel Healthpedia. Create new API get list spesialisasi dengan subspecialisasi yang dimiliki. Enhance redirect detail article with new slug by current language. Enhance tags in healthpedia result deleted tag tidak berubah sesuai bahasa.
17	Memulai Pengerjaan Change Request <i>section</i> Our Doctors di halaman Detail Speciality. <i>Create GET API</i> Highlighted Doctor. <i>Layouting</i> dan melakukan <i>Integration API</i> di FE main
18	Melanjutkan Pengerjaan Change Request <i>section</i> Our Doctors di halaman Detail Speciality. <i>Create GET API</i> Highlighted Doctor. <i>Layouting</i> dan melakukan <i>Integration API</i> di FE main
19	Memulai Pengerjaan Enhancement <i>Landing Hospitals</i> . Melakukan enhancement pada <i>component</i> filter spesialisasi/subspecialisasi dan kota.
Lanjut pada halaman berikutnya	

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

Minggu Ke -	Pekerjaan yang dilakukan
20	Melanjutkan Pengerjaan Enhancement <i>Landing Hospitals</i> . Menambahkan <i>chip selected filter</i> pada tampilan mobile
21	Mengerjakan issue yang ditemukan pada menu-menu spesialisasi/subspesialisasi dan mengerjakan Tracker untuk menu-menu spesialisasi/subspesialisasi.
22	Mengerjakan issue yang ditemukan pada menu-menu spesialisasi/subspesialisasi dan mengerjakan Tracker untuk menu-menu spesialisasi/subspesialisasi.

### 3.4 Sitemap

Gambar 3.2 merupakan gambar sitemap dari Modul *Patient Centered Care*. Pada gambar tersebut, ketika pengguna masuk ke website Siloam Hospitals dan klik *navigation bar "Patient Centered Care"*(English) atau "Perawatan Pasien"(Bahasa).



Gambar 3.2. Flowchart Landing Speciality/Subspeciality

Pada awalnya akan disajikan 2 menu yaitu Pusat Unggulan (*Center Of Excellence*) dan Spesialisasi/Subspesialisasi (*Speciality/Subspeciality*). Ketika masuk ke menu Spesialisasi/Subspesialisasi maka akan menampilkan halaman *Landing Speciality/Subspeciality*. Kemudian, saat pengguna memilih salah satu spesialisasi/subspesialisasi, maka pengguna akan diarahkan ke halaman *Detail Speciality/Subspeciality*. Dari *Detail Speciality/Subspeciality* pengguna dapat diarahkan ke halaman-halaman lain seperti *Landing Search Doctor*, *Landing Article*, *Landing Hospitals*, *Detail Doctor*, *Detail Article*, dan *Detail Hospital*

#### **3.4.1 User Requirement**

Setiap diakhir sprint dan awal sprint baru akan diadakan meeting dari tim *product* dan *tim developer (website)*. Pada meeting tersebut akan diperkenalkan project-project yang akan dilakukan pada sprint tersebut. Setiap project sudah ditetapkan masing-masing PIC nya yang ditunjuk langsung oleh Tech Lead dan Scrum Master. Pada project pengembangan menu Spesialisasi/Subspesialisasi terdapat beberapa *requirement* yang diperlukan untuk pengembangan project. Namun, selama proses development terjadinya beberapa kali perubahan *requirement* dari sisi user dan product. Berikut merupakan *product requirement* final untuk *project development* menu Spesialisasi/Subspesialisasi (Patient Centered Care):

1. *Landing Speciality/Subspeciality*, halaman *Landing Speciality/Subspeciality* memiliki tabs "Berdasarkan Nama" dan "Berdasarkan Organ". Pada tabs "Berdasarkan Nama" terdapat fitur Search by Specialty Name yang akan menampilkan suggestion list daftar Specialties. Ketika fitur search digunakan, halaman memunculkan maksimal 4 suggestions list Specialties untuk keyword yang diketik oleh pengguna yang disusun berdasarkan abjad dan ketika user klik Enter setelah input keyword pada Search Bar, maka akan ditampilkan *section* Search Result berupa list card Specialties yang mendekati input keyword pencarian user. Filter lokasi rumah sakit memungkinkan user untuk melihat daftar Specialty yang tersedia pada unit-unit tertentu. Kemudian, terdapat *section* "Most Popular" yang menampilkan total 6 cards Specialties/Subspecialties terpopuler yang ketika di klik akan mengarahkan user ke halaman detail Specialty atau Subspecialty tersebut.

Lalu, disamping *section* "Most Popular" terdapat *section* "All Specialities/Subspecialities" yang dikategorikan berdasarkan abjad. Ketika salah satu card *Specialty* atau *Subspecialty* diklik akan mengarahkan ke halaman detail *Specialty* atau *Subspecialty* tersebut.

Pada tabs "Berdasarkan Organ" terdapat *interactive visual model body parts* dengan tombol untuk zoom in dan zoom out. Chip button pada visual model body parts clickable dan akan memunculkan list Specialties sesuai body part yang dipilih. Terdapat list chip body part pada bagian kanan. Ketika di Klik maka akan menampilkan *section list card specialities/subspecialities* yang dimiliki oleh anggota tubuh yang terpilih. Saat salah satu *card specialities/subspecialities* diklik maka akan mengarahkan pengguna ke halaman detail *Specialty* atau *Subspecialty* tersebut.

2. *Detail Speciality Page*, halaman *Detail Speciality Page* menampilkan judul dan deskripsi dari Specialty terkait. Deskripsi ditampilkan dalam 2 line. Sisanya deskripsi akan di-hide. Akan terdapat CTA "Read More" atau "Baca Selengkapnya" akan expand keseluruhan deskripsi dari Specialty, CTA "Read Less" atau "Baca Lebih Sedikit" (muncul setelah *expand speciality description*) akan mengembalikan *section description* pada tampilan default. Pada detail speciality terdapat *section* yang menampilkan list *Sub-specialties cards* dari Specialty yang sedang dipilih. *Card Sub-specialties* yang ditampilkan maksimal pada 2 lines (6 buah), sisanya akan ditampilkan sebagai CTA "Expand All" yang akan menampilkan seluruh *card sub-specialties*. Lalu, selanjutnya terdapat *Clinical Conditions section* yang akan menampilkan daftar Clinical Conditions atau Kondisi Klinis yang berkaitan dengan spesialisasi tersebut. Kemudian, terdapat *Treatments section* yang menampilkan daftar Treatments atau Tindakan yang berkaitan dengan spesialisasi.

Lalu, pada halaman Detail Spesialisasi terdapat Our Doctors section yang menampilkan dokter-dokter untuk spesialisasi tersebut, pada bagian Our Doctors juga terdapat filter bar rumah sakit untuk dokter spesialis yang ada pada rumah sakit tersebut. Saat klik salah satu *doctor card* maka akan diarahkan ke halaman detail doctor. Selanjutnya, terdapat *Available at section* yang menampilkan 4 card list rumah sakit dimana spesialisasi/subspesialisasi terkait tersedia. Ketika *Geolocation ON* maka akan mengambil 4 rumah sakit terdekat, ketika di klik maka akan mengarahkan ke halaman Detail Rumah

Sakit. Jika *Geolocation OFF* maka akan mengambil 4 *most advanced Siloam unit*. Selanjutnya terdapat *Related Articles section*, menampilkan 4 *card list* artikel kesehatan Siloam, ketika di klik maka akan mengarahkan ke halaman Detail Artikel.

3. *Detail Subspeciality Page*, halaman *Detail Subspeciality Page* menampilkan judul dan deskripsi dari subspecialisasi terkait. Deskripsi ditampilkan dalam 2 line. Sisanya deskripsi akan di-hide. Akan terdapat CTA “Read More” atau “Baca Selengkapnya” akan expand keseluruhan deskripsi dari subspecialisasi, CTA “Read Less” atau “Baca Lebih Sedikit” (muncul setelah *expand subspeciality description*) akan mengembalikan *section description* pada tampilan default. Lalu, selanjutnya terdapat *Clinical Conditions section* yang akan menampilkan daftar Clinical Conditions atau Kondisi Klinis yang berkaitan dengan subspecialisasi tersebut. Kemudian, terdapat *Treatments section* yang menampilkan daftar Treatments atau Tindakan yang berkaitan dengan subspecialisasi.

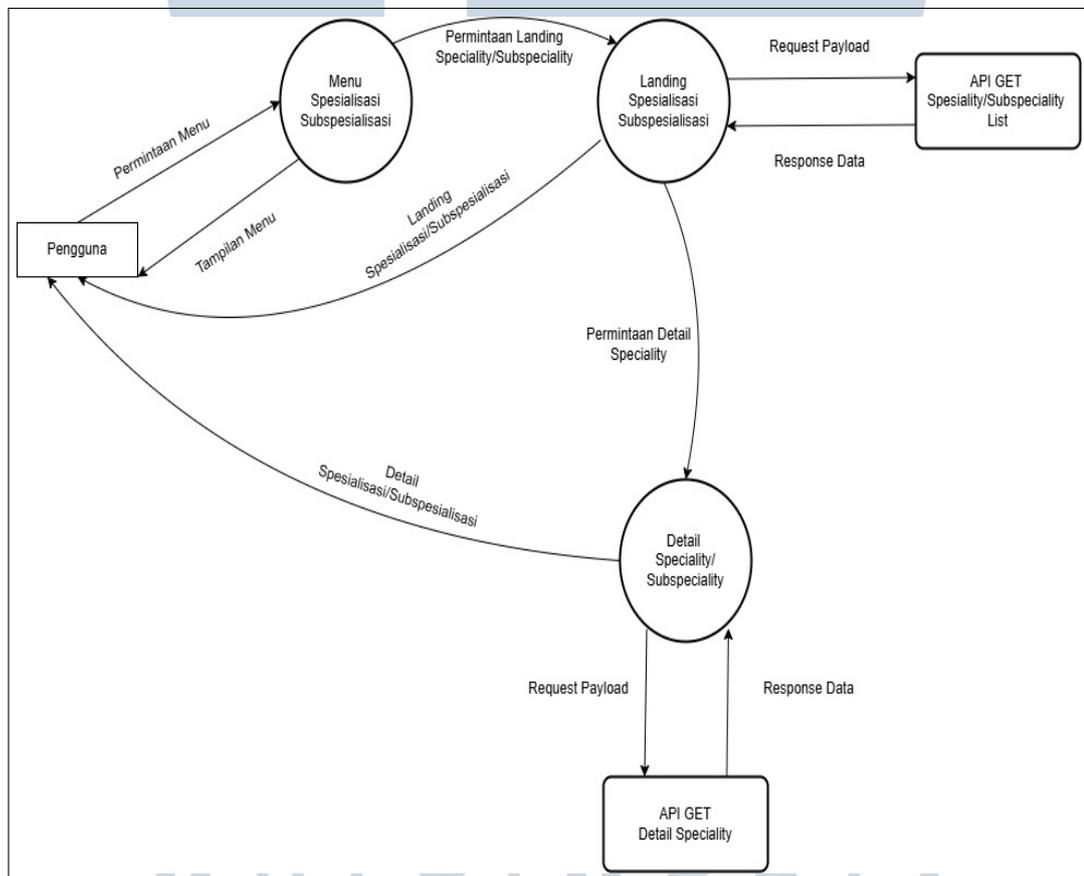
Dibawahnya terdapat *section Doctor Banner*, pada banner terdapat *button Find Doctors* yang akan mengarahkan user ke halaman Cari Dokter. Selanjutnya, terdapat *Available at section* yang menampilkan 4 *card list* rumah sakit dimana subspecialisasi terkait tersedia. Ketika *Geolocation ON* maka akan mengambil 4 rumah sakit terdekat, ketika di klik maka akan mengarahkan ke halaman Detail Rumah Sakit. Jika *Geolocation OFF* maka akan mengambil 4 *most advanced Siloam unit*. Selanjutnya terdapat *Related Articles section*, menampilkan 4 *card list* artikel kesehatan Siloam, ketika di klik maka akan mengarahkan ke halaman Detail Artikel.

4. Penyesuaian *Landing Page* Rumah Sakit, terdapat *SEO Section* yang menampilkan judul dan deskripsi halaman Rumah Sakit Kami. Kemudian, terdapat Filter *Specialty* dan *Subspecialty* akan digabung menjadi 1 filter, dengan ketentuan User dapat melakukan search di dalam filter Ketika user melakukan search di dalam filter, maka hasil search akan menampilkan hasil nama *Specialty* atau *Subspecialty* terdekat dari keyword yang diinput User dapat menghapus keyword yang diinput dengan klik *button X* pada search bar. Filter tersebut bersifat *multiple select checkbox*. Kemudian terdapat filter *city*. User dapat melakukan filter rumah sakit berdasarkan nama kota/city di mana lokasi tersebut tersedia dan hanya akan menampilkan list kota yang terdapat unit Siloam Hospitals.

Pada kedua filter tersebut terdapat CTA “Reset” untuk menghapus seluruh pilihan filter sebelumnya. Selanjutnya adalah *result section* yang menampilkan hasil dari pencarian rumah sakit dari filter yang telah di tetapkan oleh pengguna.

### 3.4.2 Data Flow Diagram Speciality/Subspeciality Menu

Gambar 3.3 merupakan *data flow diagram* atau sistem pada menu spesialisasi/subspesialisasi. Data Flow Diagram pada gambar 3.3 menggambarkan aliran data antara pengguna dan proses-proses utama dalam menu spesialisasi/subspesialisasi.



Gambar 3.3. Data Flow Diagram Menu Speciality/Subspeciality

Berikut adalah penjelasan lebih lanjut terkait ilustrasi data flow diagram menu spesialisasi/subspesialisasi. Terbagi menjadi 3 bagian, yaitu entitas eksternal, proses, dan aliran data:

### 1. Entitas Eksternal

- Pengguna, Pengguna dapat meminta data menu spesialisasi/subspesialisasi, memilih tab, dan melihat detail spesialisasi/subspesialisasi.

### 2. Proses

- Menu Spesialisasi/Subspesialisasi, memproses permintaan pengguna untuk mengakses menu utama spesialisasi/subspesialisasi.
- Landing Spesialisasi/Subspesialisasi, proses ini menampilkan daftar spesialisasi dan subspesialisasi berdasarkan tab yang dipilih pengguna (berdasarkan nama atau body part).
- Tampilkan Detail Spesialisasi/Subspesialisasi, proses ini menampilkan informasi detail spesialisasi atau subspesialisasi yang dipilih oleh pengguna.

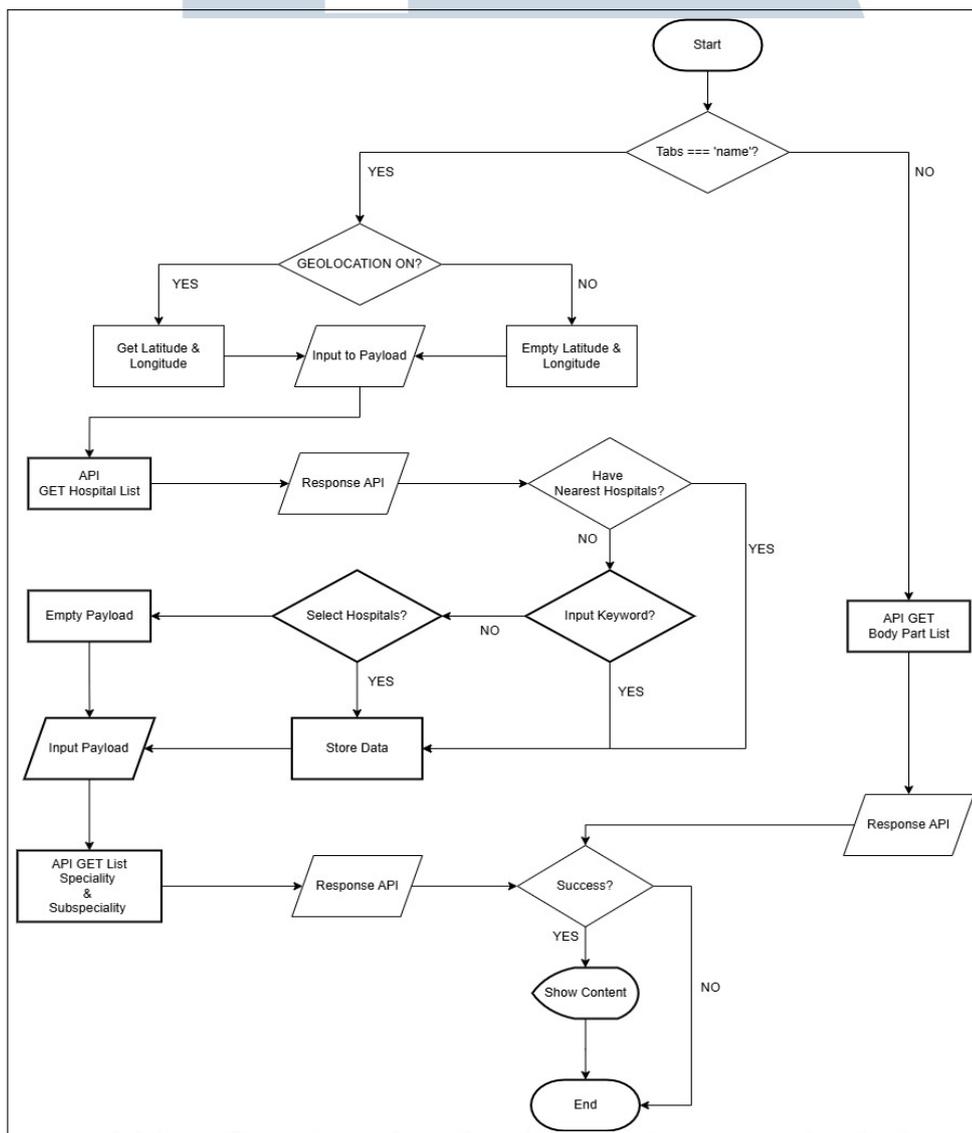
### 3. Aliran Data

- Permintaan Menu, pengguna mengirimkan permintaan untuk mengakses menu utama spesialisasi/subspesialisasi. Proses "Menu Spesialisasi/Subspesialisasi" memproses permintaan menu.
- Permintaan Landing Page, proses "Landing Spesialisasi/Subspesialisasi" meminta data spesialisasi/subspesialisasi dari API GET List Speciality/Subspeciality. API mengembalikan daftar spesialisasi/subspesialisasi untuk ditampilkan.
- Permintaan Detail Spesialisasi, saat pengguna memilih salah satu spesialisasi/subspesialisasi, proses "Detail Spesialisasi/Subspesialisasi" meminta data detail dari API Get Detail Speciality/Subspeciality. API mengembalikan informasi detail yang kemudian ditampilkan kepada pengguna.

### 3.4.3 Flowchart

#### A. Flowchart Landing Speciality/SubSpeciality

Gambar 3.4 merupakan flowchart Landing Speciality/Subspeciality yang menunjukkan proses mendapatkan data hingga menampilkan section-section yang ada pada Landing tersebut.



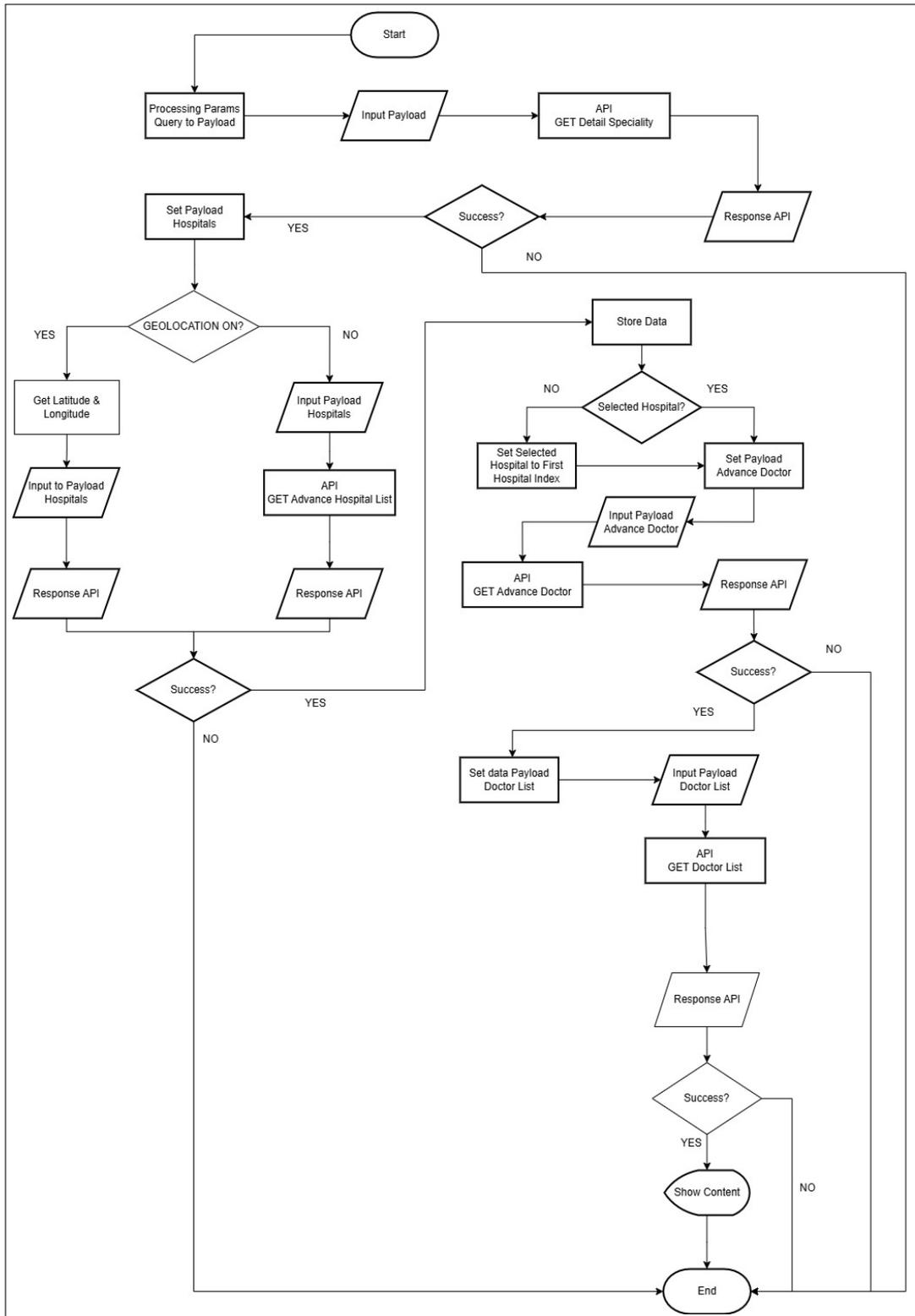
Gambar 3.4. Flowchart Landing Speciality/Subspeciality

Pada awalnya akan, akan mengecek tabs yang sedang dipilih oleh pengguna. Jika selected tabs adalah "Name" maka akan memproses data-data yang dibutuhkan untuk menampilkan konten *Landing Speciality By Name*. Pertama akan mengecek

apakah *geolocation* aktif, jika aktif akan set *latitude* dan *longitude*, lalu jika *geolocation off* maka *latitude* dan *longitude* akan bernilai kosong. Kemudian, akan set ke payload untuk dikirimkan ke API *GET Hospital List*. Lalu, setelah mendapatkan *response* API maka terdapat pengecekan apakah terdapat *nearest hospitals*, *input keyword* dan *selected hospitals* dari pengguna. Input dari user akan distore dan diinput ke payload dan akan dikirimkan ke API *GET List Speciality/Subspeciality*. Jika mendapatkan *response success* maka akan menampilkan konten-konten terkait. Kemudian, ketika *selected tabs* adalah "bodyPart" maka akan melakukan *hit API GET Speciality Body Part*, ketika data berhasil didapatkan maka akan menampilkan konten-konten di *section Landing Speciality/Subspeciality by Body Part*

## **B. Flowchart Detail Speciality**

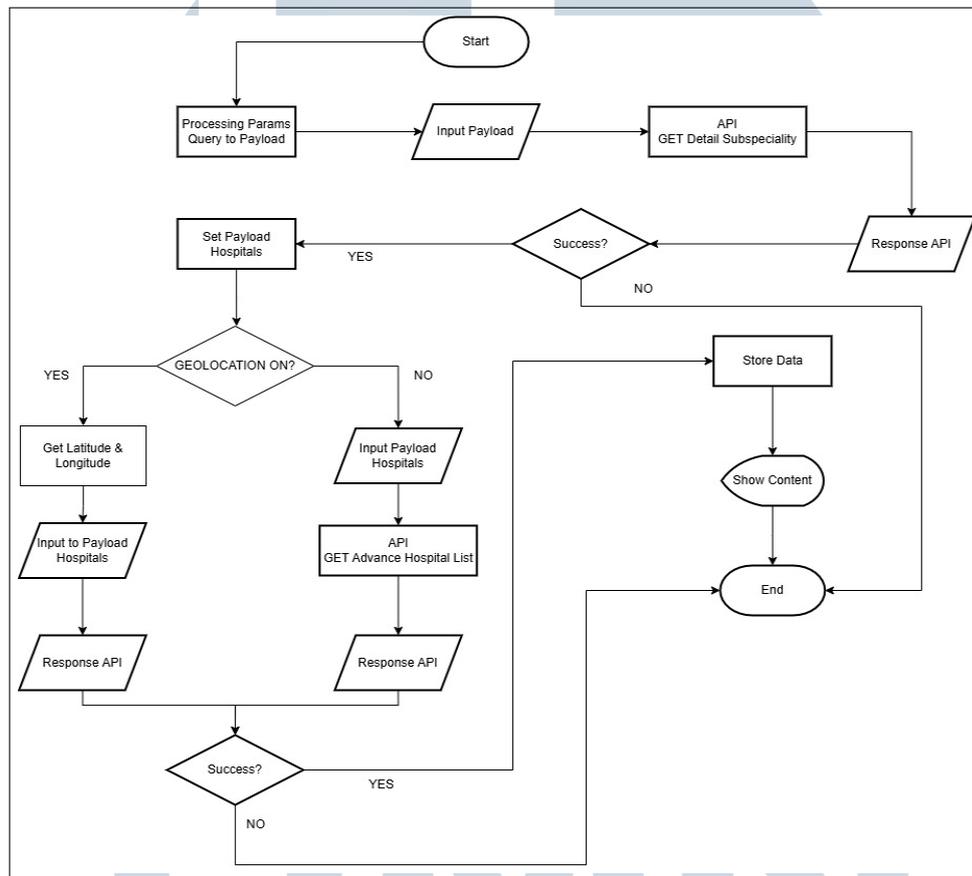
Gambar 3.5 merupakan flowchart yang menunjukkan proses pada halaman Detail Spesialisasi mulai dari proses mendapatkan data hingga menampilkan data. Pertama, akan dilakukan proses pembacaan *params query* dan memasukkannya ke payload, payload akan dikirimkan ke API *GET Detail Specialty*. Ketika data detail spesialisasi didapatkan, maka akan melanjutkan proses untuk set *payload hospitals*. Lalu akan ada pengecekan *geolocation*. Jika, *geolocation on* maka akan mengambil *latitude* dan *longitude* pengguna dan akan ditambahkan ke payload *hospitals* untuk dikirimkan ke API *GET Nearby Hospitals*. Sedangkan, ketika *geolocation off* maka akan mengirimkan payload *hospitals* yang sudah didefine sebelumnya ke API *GET Advanced Hospital List*. Selanjutnya, ketika berhasil mendapatkan datanya maka akan *store data* dan akan melakukan pengecekan *selected hospitals*, data rumah sakit yang terpilih akan dimasukan ke payload untuk dikirimkan ke API *GET Advanced Doctor*. Kemudian, data dari API *GET Advanced Doctor* digunakan untuk mengatur payload *Doctor List* dan mengirimkan payload tersebut ke API *GET Doctor List*. Jika sudah mendapatkan datanya. Maka akan menampilkan section-section yang terdapat di halaman Detail Spesialisasi. Setelah semua data didapatkan, maka akan akan menampilkan semua section pada detail speciality.



Gambar 3.5. Flowchart Detail Spesialisasi

### C. Flowchart Detail SubSpeciality

Gambar 3.6 merupakan flowchart yang menunjukkan proses pada halaman Detail Subspesialisasi mulai dari proses mendapatkan data hingga menampilkan data.

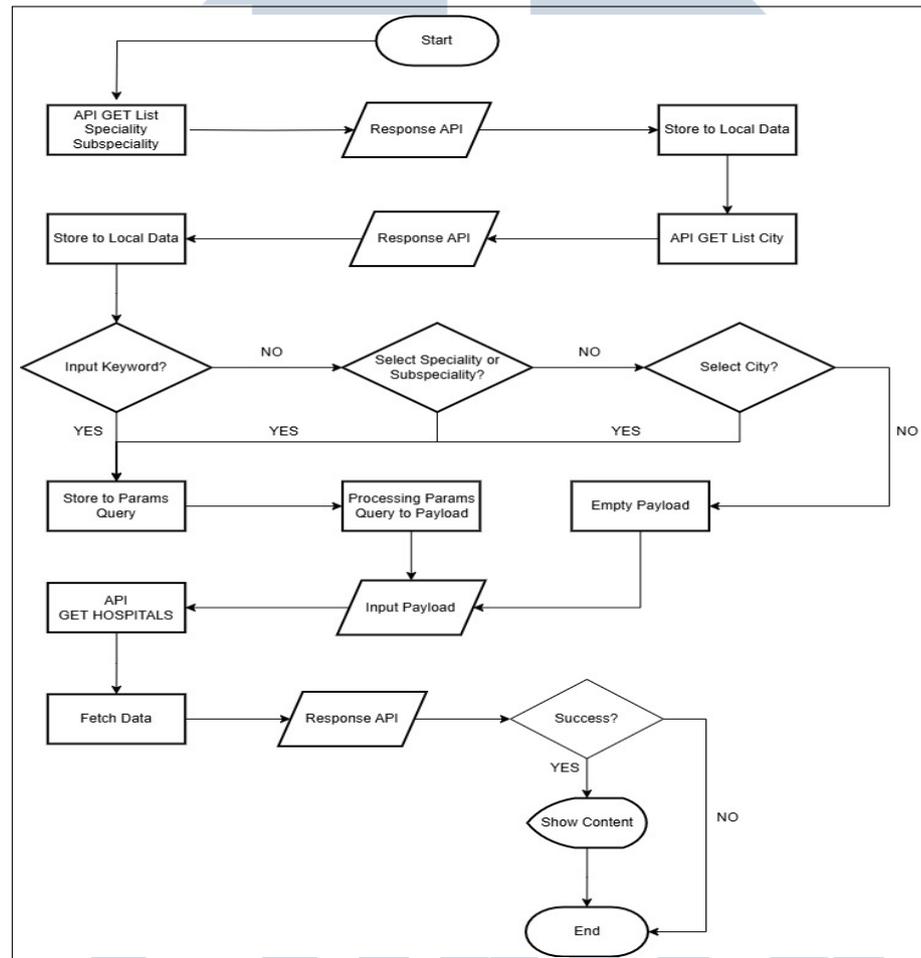


Gambar 3.6. Flowchart Detail Spesialisasi

Pertama, akan dilakukan proses pembacaan *params query* dan memasukkannya ke payload, payload akan dikirimkan ke API *GET Detail Subpecialty*. Ketika data detail subspesialisasi didapatkan, maka akan melanjutkan proses untuk *set payload hospitals*. Lalu akan ada pengecekan *geolocation*. Jika, *geolocation on* maka akan mengambil *latitude* dan *longitude* pengguna dan akan ditambahkan ke payload hospitals untuk dikirimkan ke API *GET Nearby Hospitals*. Sedangkan, ketika *geolocation off* maka akan mengirimkan payload hospitals yang sudah didefine sebelumnya ke API *GET Advanced Hospital List*. Setelah data berhasil didapatkan, maka akan menampilkan section-section pada halaman detail spesialisasi.

#### D. Flowchart Landing Hospitals

Gambar 3.7 merupakan flowchart yang menunjukkan proses mendapatkan data hingga menampilkan data pada halaman *Landing Hospitals*.



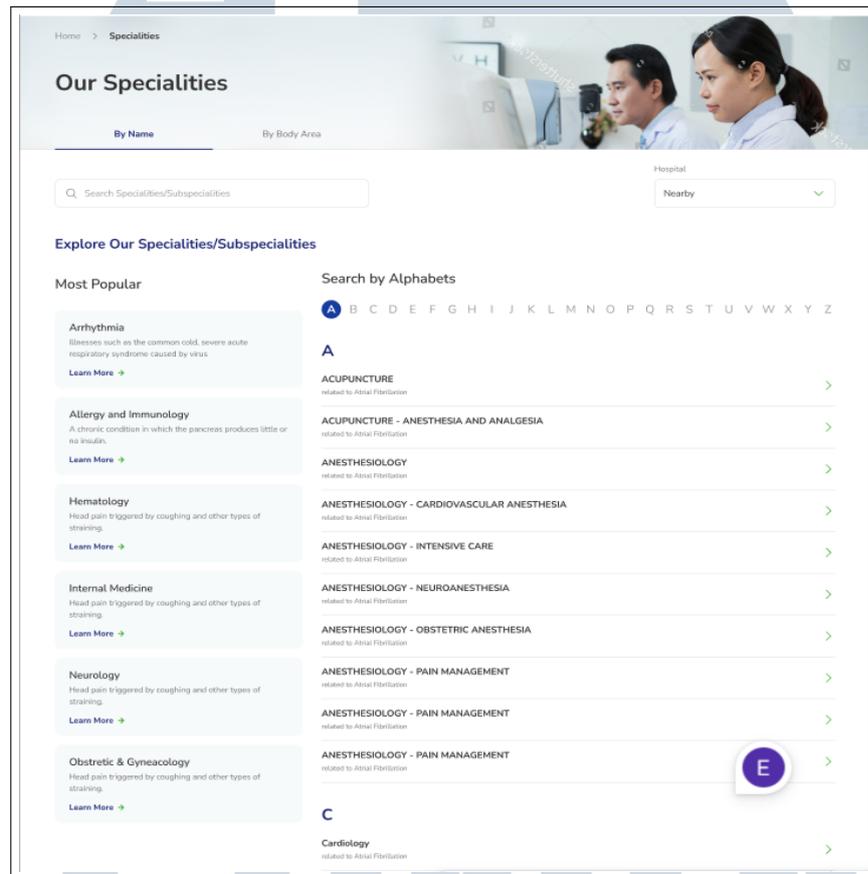
Gambar 3.7. Flowchart Landing Hospitals

Pertama, akan dilakukan *hit API GET List Speciality/Subspeciality* dan *API GET List City*. Kemudian, data-data tersebut akan distore. Kemudian, akan dilakukan pengecekan apakah terdapat input keyword, pemilihan filter spesialisasi/subspesialisasi, dan pemilihan filter kota dari pengguna. Jika ada, maka akan melakukan proses store data-data tersebut ke *params query*, lalu akan memproses *params query* tersebut untuk dimasukan kedalam payload dan akan dikirimkan ke *API GET Hospitals*. Jika Mendapatkan datanya, maka akan menampilkan hasil pencarian rumah sakit dari filter yang sudah diberikan pengguna.

### 3.4.4 Design Figma

#### A. Figma Landing Speciality/Subspeciality by Name

Gambar 3.8 merupakan design Figma untuk halaman Landing *Speciality/Subspeciality by Name*.



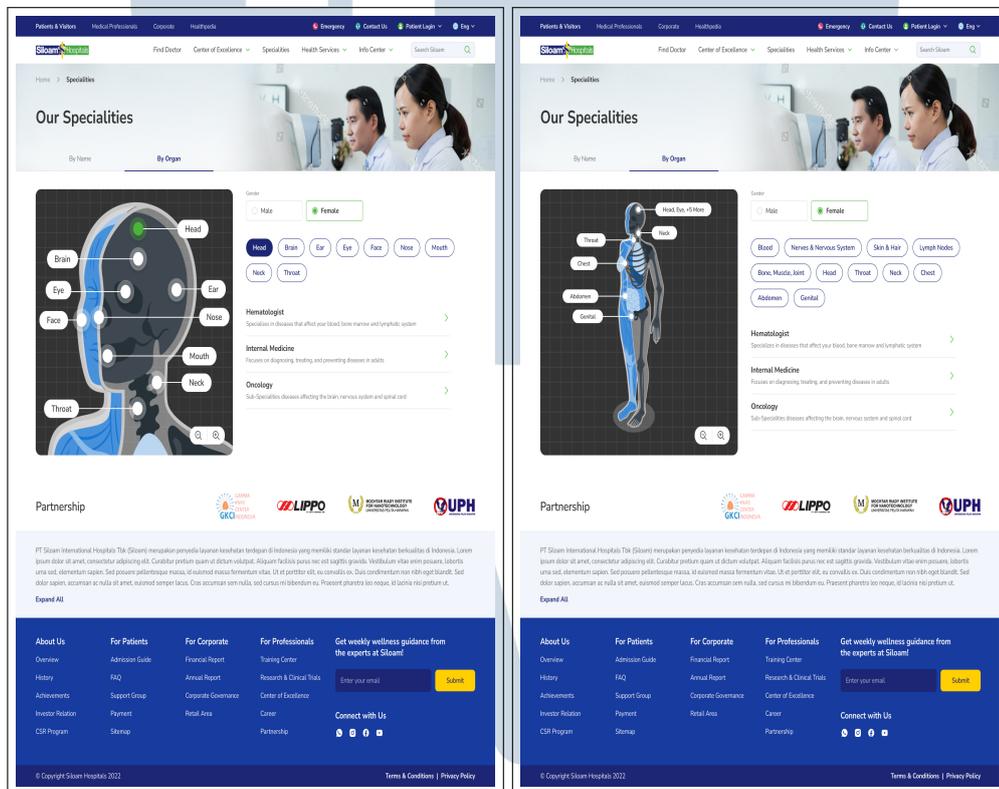
Gambar 3.8. *Landing Speciality/Subspeciality by Name*

Design *Landing Speciality/Subspeciality by Name* terdiri dari beberapa bagian, yaitu tabs untuk mengubah landing berdasarkan "Nama" dan "Organ". Lalu, terdapat *search bar* untuk melakukan pencarian *speciality/subspeciality* dari *keyword* yang diketik langsung oleh pengguna. Kemudian, terdapat dropdown filter berdasarkan rumah sakit / unit Siloam. Pengguna dapat memilih beberapa rumah sakit sekaligus yang tersedia. Ketika filter-filter tersebut digunakan maka akan menampilkan hasil *speciality/subspeciality* sesuai dengan filter yang diberikan. Kemudian, terdapat bagian *card* "Most Popular" yang menunjukkan spesialisasi dan subspecialisasi terpopuler yang dimiliki siloam. Disebelahnya terdapat bagian yang menunjukkan semua spesialisasi dan subspecialisasi dan dikategorikan berdasarkan

urutan abjad.

## B. Figma Landing Speciality/Subspeciality by Body Part

Gambar 3.9 diatas merupakan *design* Figma untuk Landing Speciality/Subspeciality by Organ. *Design* halaman Landing Speciality/Subspeciality by Body Part menampilkan *interactive visual model body parts* chip pada gambar anggota tubuh tersebut dapat diklik dan akan otomatis select chip disebelah kanannya. Saat chip diklik maka akan menampilkan list card speciality/subspeciality pada anggota tubuh tersebut.

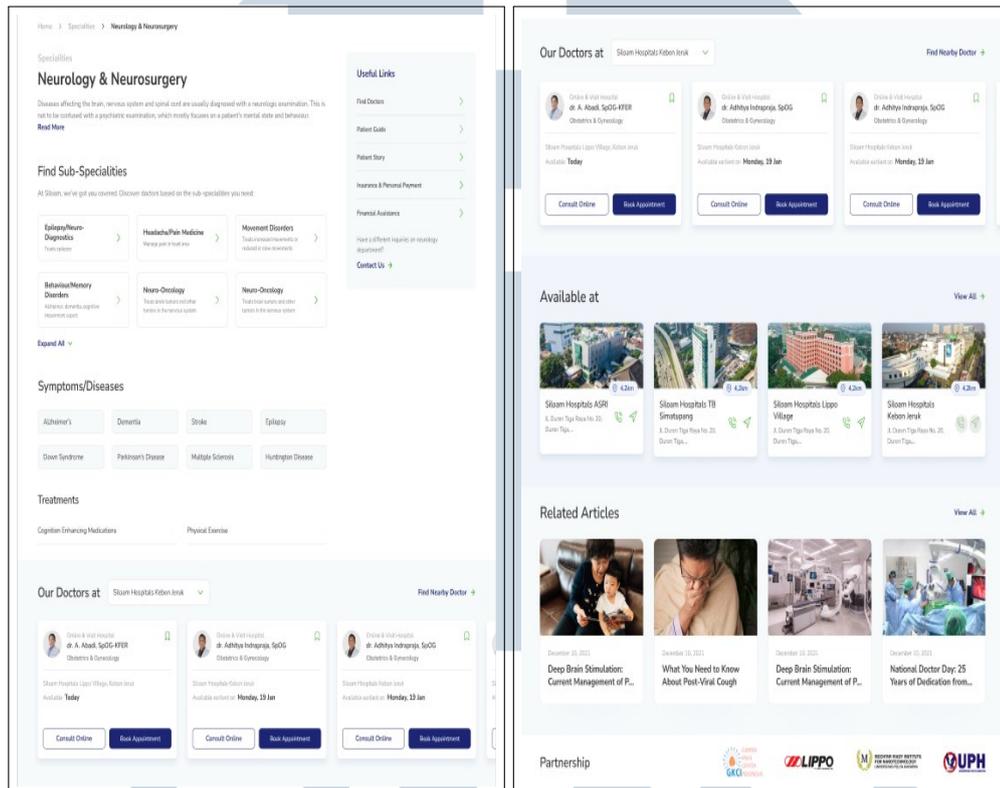


Gambar 3.9. Design Landing Speciality/Subspeciality by Organ

Interactive visual model body parts juga akan menyesuaikan sesuai dengan kelamin yang dipilih oleh pengguna. Kemudian, chip pada interactive visual model body parts tersebut juga menyesuaikan ketika pengguna *zoom in* atau *zoom out*.

### C. Figma Detail Speciality

Gambar 3.10 merupakan design halaman Detail Spesialisasi yang menunjukkan detail-detail informasi terkait dengan spesialisasi tersebut.

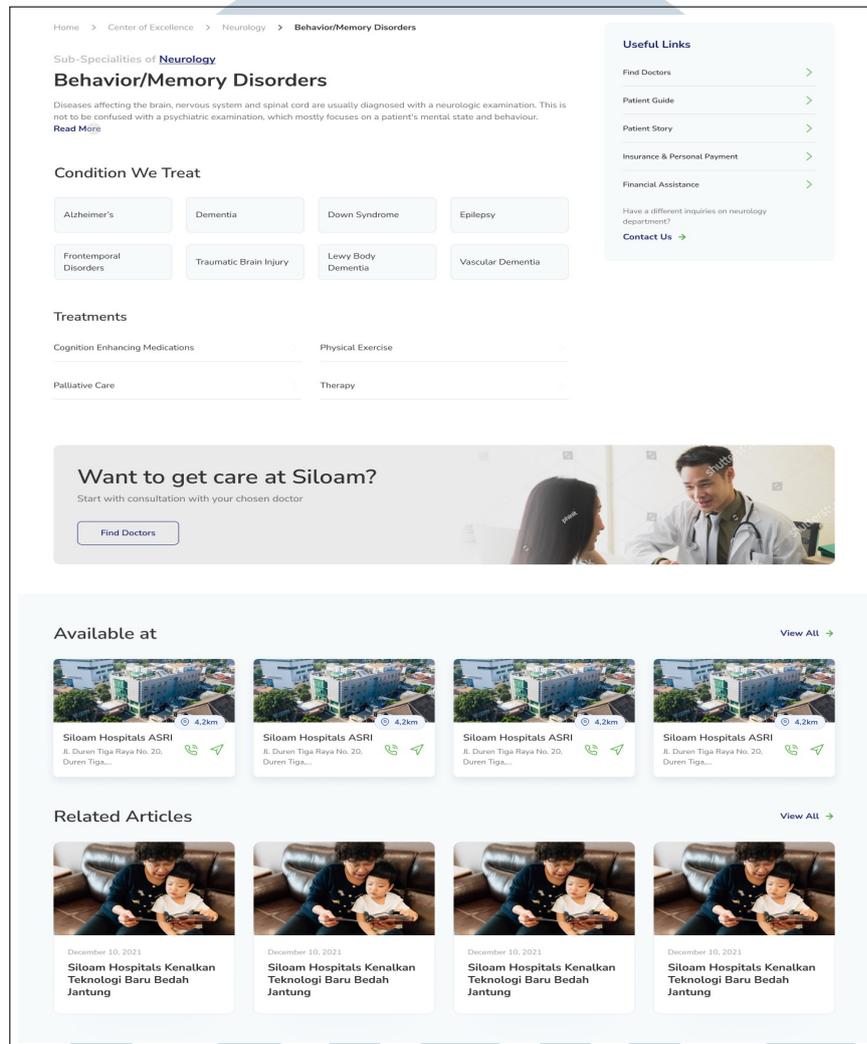


Gambar 3.10. Design Detail Speciality

Terdapat bagian yang menampilkan judul dan deskripsi terkait spesialisasi. Lalu, dibawahnya terdapat bagian yang menampilkan subspecialisasi dari spesialisasi tersebut. Kemudian, terdapat bagian yang menampilkan kondisi klinis dan perawatan yang berhubungan dengan spesialisasi tersebut. Dibawahnya, terdapat bagian-bagian yang menampilkan banner dokter, dokter spesialis, rumah sakit, dan artikel kesehatan yang terkait dengan spesialisasi tersebut. Kemudian, terdapat *section sticky Useful Links* yang berikan list buton yang dapat mengarahkan pengguna langsung menuju halaman terkait.

## D. Figma Detail Subspeciality

Pada Gambar 3.11 menunjukkan design pada halaman Detail Subspesialisasi.

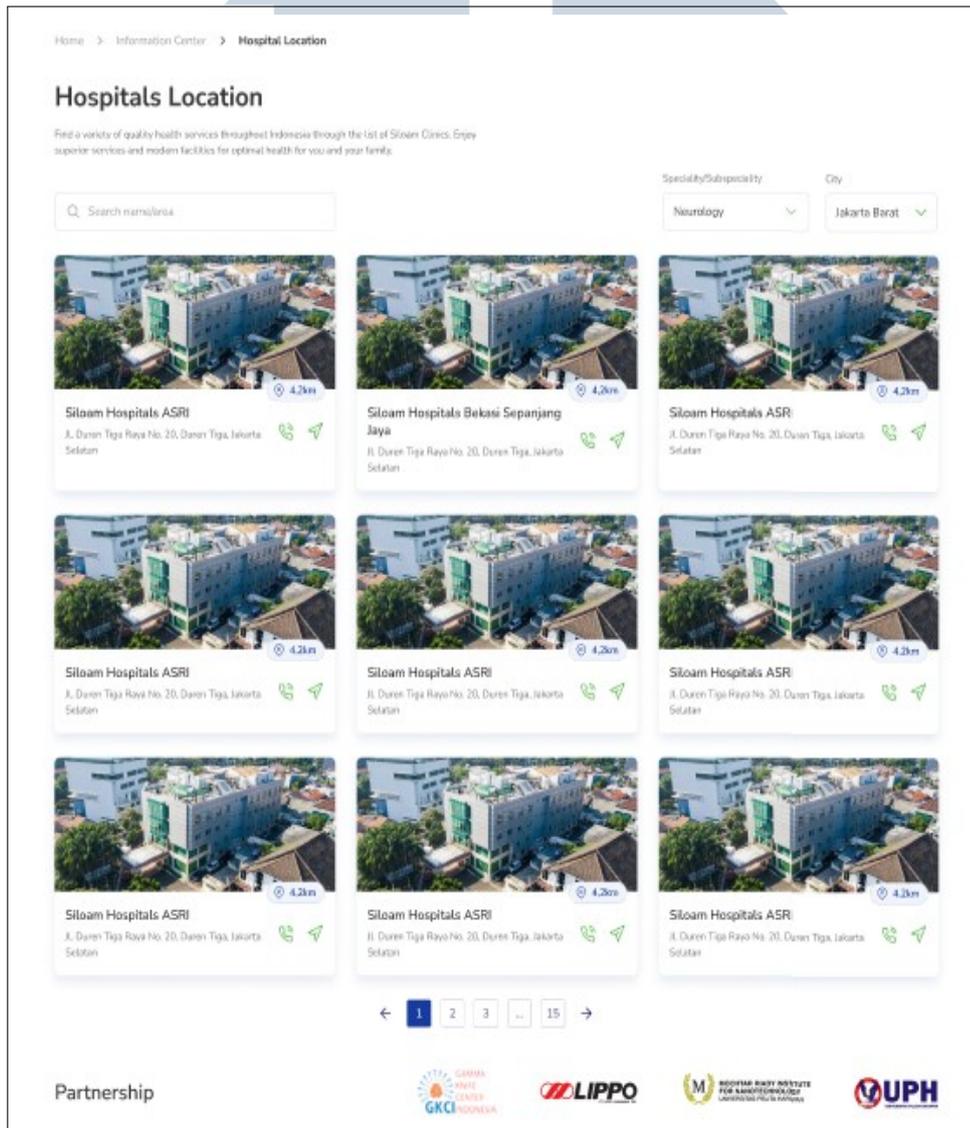


Gambar 3.11. Figma Detail Subspeciality

Pada *design* halaman *Detail Subspeciality* menampilkan informasi-informasi terkait dengan subspecialisasi yang dipilih oleh pengguna. Seperti informasi parent dari subspecialisasi tersebut dan deskripsi dari subspecialisasi. Secara keseluruhan, *section* yang ditampilkan mirip dengan detail spesialisasi. Namun, terdapat beberapa *section* yang tidak ada pada halaman *Detail Subspeciality*, yaitu "subsialisasi terkait" dan "Our Doctors".

## E. Figma Landing Hospitals

Gambar 3.12 menampilkan list-list rumah sakit yang dimiliki oleh Siloam. Ditampilkan dalam bentuk *card list* dan terdapat *pagination* dihalaman tersebut.



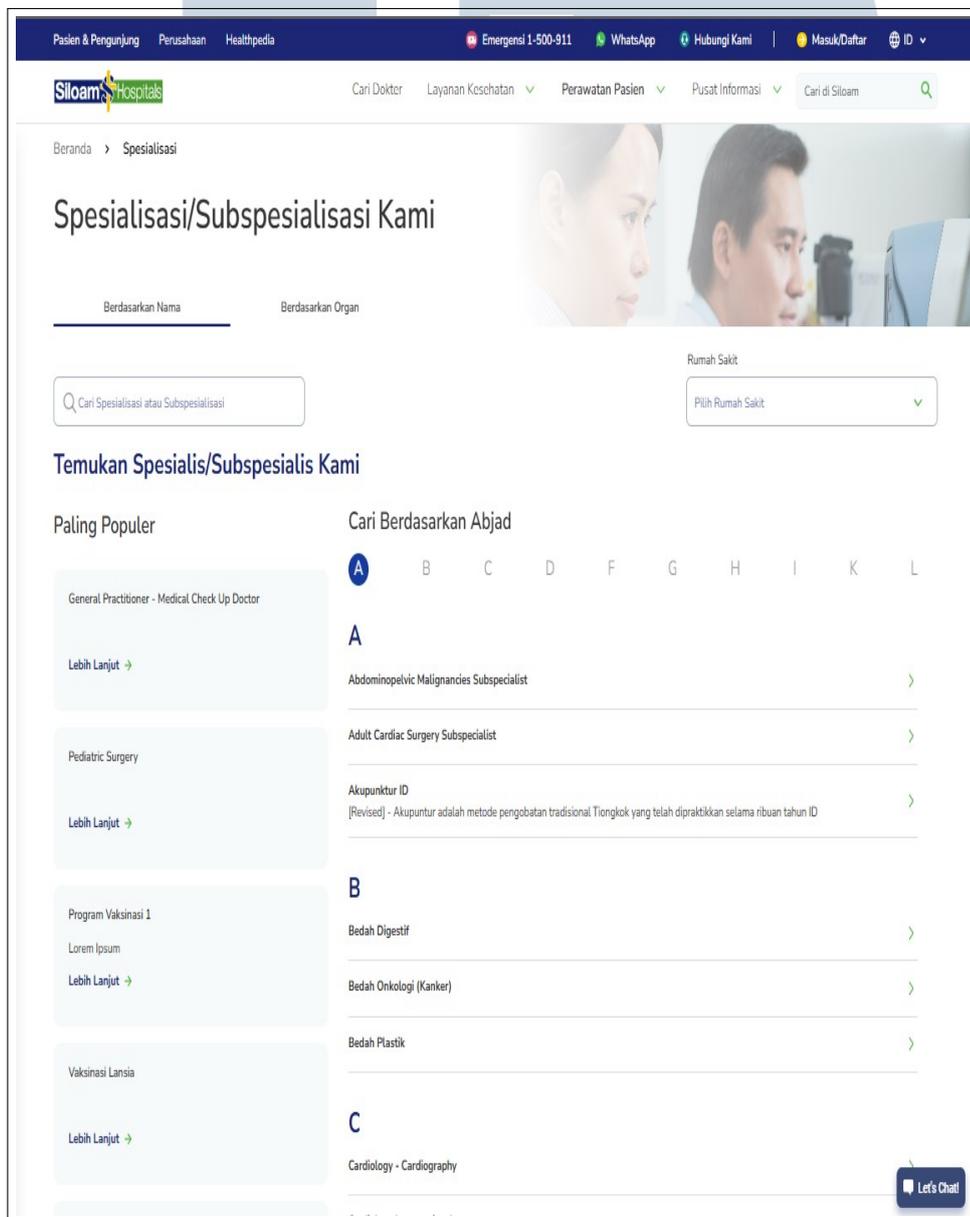
Gambar 3.12. Figma Landing Hospitals

Halaman *Landing Hospitals* memiliki search bar yang dapat digunakan untuk *enter keyword* untuk mencari rumah sakit yang sesuai atau terdekat dengan *keyword* yang dimasukkan oleh pengguna. Kemudian terdapat juga filter dropdown berdasarkan spesialisasi/subspesialisasi dan kota. Filter tersebut bersifat *multiple selection* sehingga pengguna dapat memilih beberapa pilihan sekaligus. Halaman akan menampilkan hasil sesuai dengan *input* keyword dan filter dari pengguna.

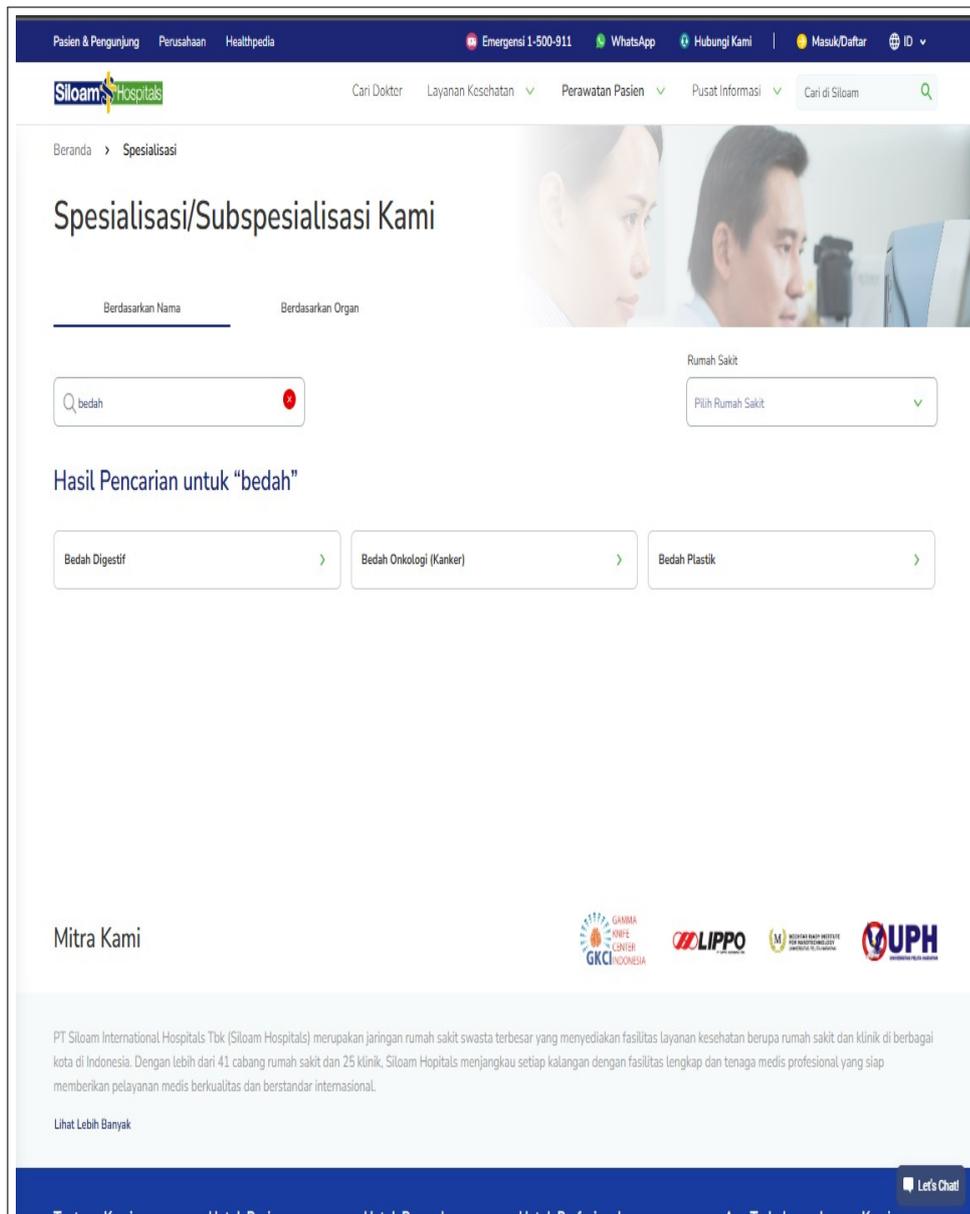
## 3.5 Implementasi

### 3.5.1 Landing Speciality/Subspeciality by Name

Pada gambar 3.13 dan 3.14 merupakan hasil pengembangan halaman *Landing Speciality by Name* berdasarkan Figma dan *user requirement*. Pengerjaan halaman *Landing Speciality/Subspeciality by Name* dimulai dengan *layouting* dan *API integration* (ilustrasi pada gambar 3.4).



Gambar 3.13. *Landing Speciality/Subspeciality by Name*



Gambar 3.14. Result By Keyword Landing Speciality/Subspeciality

Pada halaman *Landing Speciality/Subspeciality by Name* menampilkan *section* banner judul. Lalu, terdapat *tabs section* dengan tabs "By Name" merupakan tabs yang terpilih, ditandai dengan tulisan biru dan underline. Kemudian, jika tabs "By Organ" diklik maka akan mengganti tampilan ke *Landing Speciality by Body Part*. Lalu dibawahnya merupakan *section* konten utama dari halaman *Landing Speciality/Subspeciality by Name*, terdapat *search bar* dan filter pemilihan rumah sakit. Kemudian dibawahnya terdapat *section* hasil dari pencarian terdapat *section* Most Popular disebelah kiri dan All Specialities/Subspecialities disebelah kanan.

Saat pengguna klik / memasukan *keyword* pada *search bar* maka akan muncul *dropdown list Suggestion* yang sesuai dengan *keyword* yang diketik. Lalu ketika pengguna klik salah satu *suggestion* maka akan diarahkan ke halaman Detail Spesialisasi/Subspesialisasi. Namun, jika pengguna mengetik *keyword* dan klik enter maka akan menampilkan *section result specialities/subspecialities by keyword* yang berbentuk *simple card list* yang berisikan judul dan tombol panah. Ketika *card* tersebut diklik maka akan diarahkan ke halaman Detail Spesialisasi/Subspesialisasi.

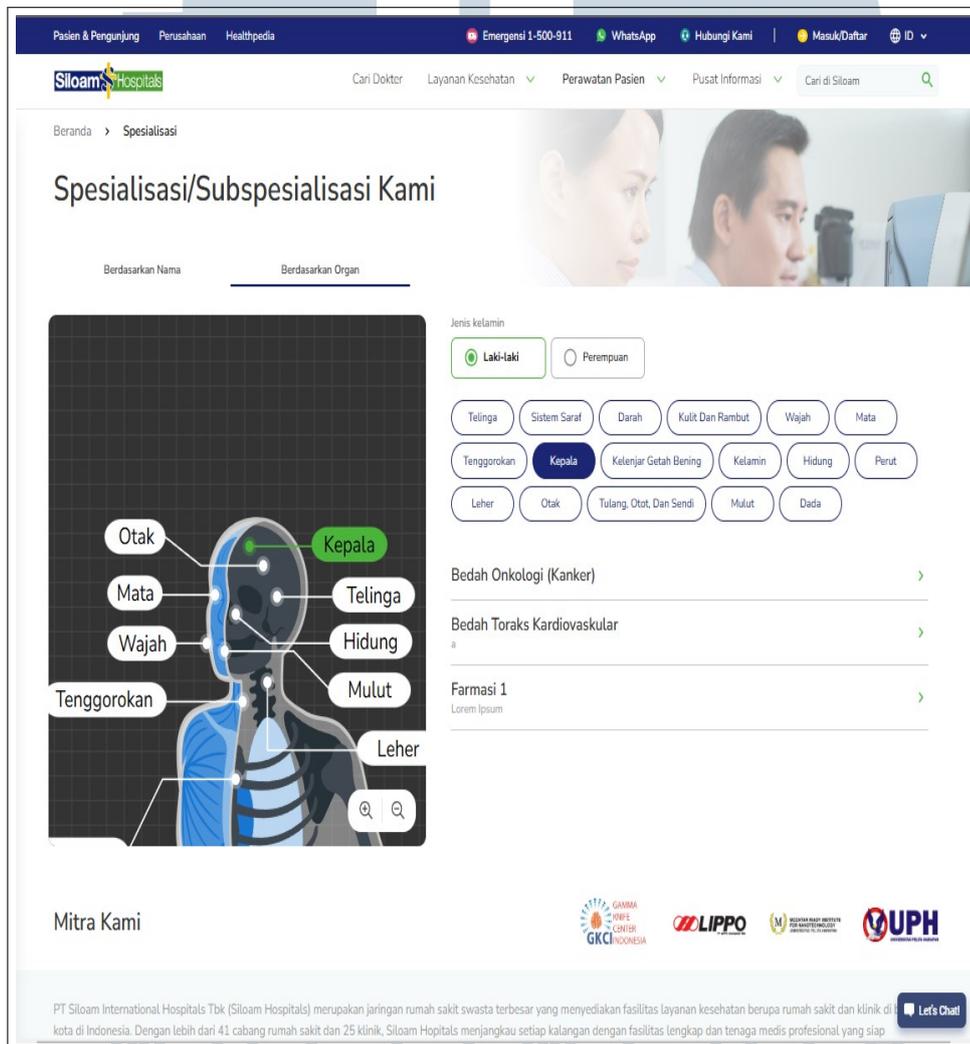
Saat pengguna mengaktifkan filter hospital / memilih rumah sakit maka akan *trigger fetching data GET specialities/subspecialities* dengan mengoper payload rumah sakit yang telah dipilih oleh pengguna ke API tersebut. Setelah mendapatkan datanya, maka akan menampilkan kembali *section* Most Popular dan All Specialities/Subspecialities.

Most Popular menampilkan 3 spesialisasi dan 3 subspesialisasi terpopuler dari Siloam dalam bentuk *card list*. *Card* tersebut berisikan judul, deskripsi singkat, dan sebuah tombol "learn more". Ketika pengguna klik tombol learn more (pelajari lebih lanjut) di salah satu cardnya maka akan diarahkan ke halaman Detail Spesialisasi/Subspesialisasi. *Section* tersebut bersifat sticky terhadap *section* disebaliknya, yaitu All Specialities/Subspecialities.

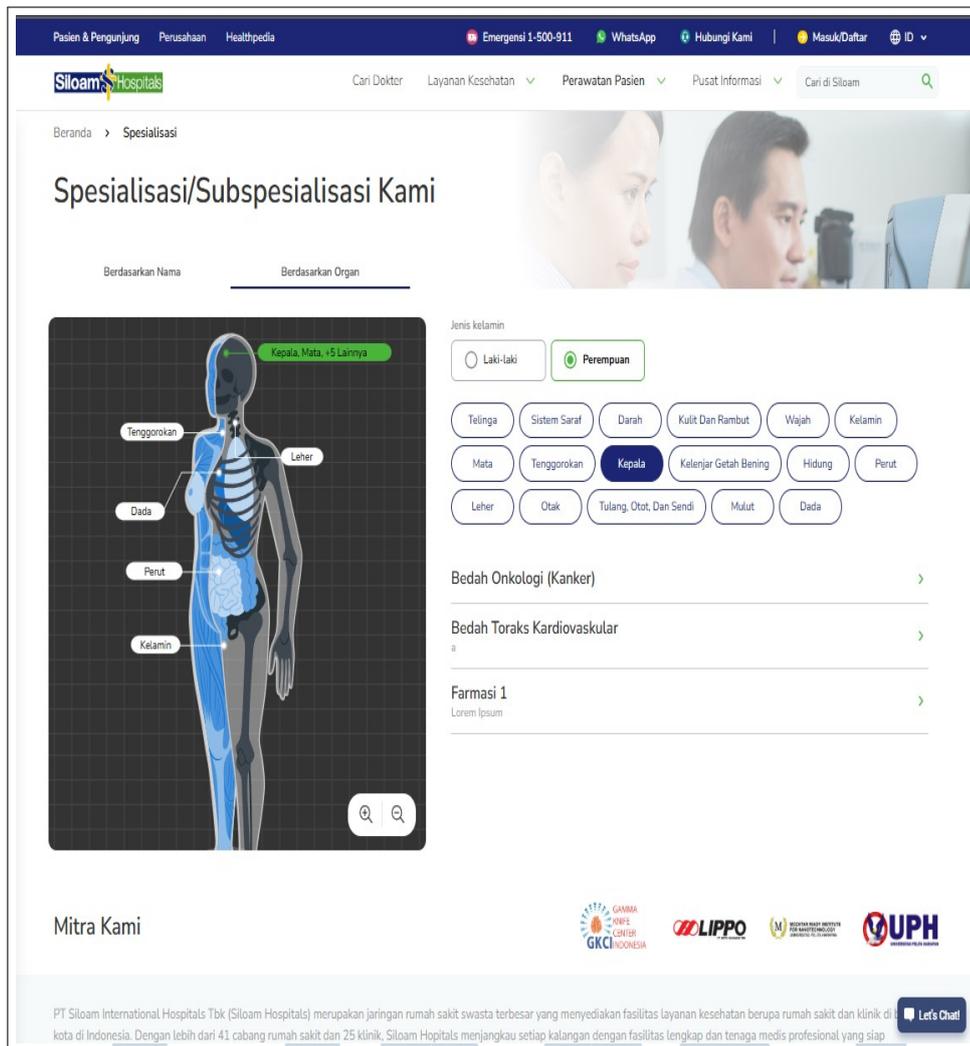
All Specialities/Subspecialities merupakan *section* yang menampilkan semua spesialisasi dan subspesialisasi dalam bentuk *simple list* yang berisikan judul, deskripsi singkat dan tombol panah. List-list tersebut dikategorikan berdasarkan urutan *alphabet*. *Section alphabet* diatas menunjukkan daftar *alphabet* yang memiliki spesialisasi/subspesialisasi. Saat diklik, maka akan auto scroll dan mengarahkan pengguna ke *section* spesialisasi/subspesialisasi *alphabet* tersebut. *Alphabet* yang terpilih ditandai dengan lingkaran berwarna biru. Kemudian, ketika pengguna melakukan scroll maka *alphabet* yang terpilih akan mengikuti *section alphabet* spesialisasi/subspesialisasi yang sedang dilihat pengguna. Kemudian, ketika salah satu list spesialisasi/subspesialisasi diklik maka akan diarahkan ke halaman Detail Spesialisasi/Subspesialisasi.

### 3.5.2 Landing Speciality by Body Part

Pada gambar 3.15 dan 3.16 merupakan hasil pengembangan halaman *Landing Speciality by Body Part* berdasarkan Figma dan *user requirement*. *Landing Speciality by Body Part* merupakan satu kesatuan dengan *Landing Speciality by Name*. *Landing Speciality by Body Part* muncul ketika pengguna memilih tabs *By Organ/Berdasarkan Organ*. Pengerjaan halaman *Landing Speciality by Body Part* dimulai dengan *layouting* dan *API integration* (ilustrasi pada gambar 3.4).



Gambar 3.15. *Landing Speciality/Subspeciality by Body Part (Zoom In)*



Gambar 3.16. *Landing Speciality/Subspeciality by Body Part (Zoom Out)*

Landing *Landing Speciality by Body Part* menampilkan *interactive visual model body parts* dengan tombol untuk *zoom in* dan *zoom out* disebalah kiri dan disebalah kanan terdapat *section radio button* untuk memilih kelamin. Kemudian dibawah *radio button* tersebut terdapat *body parts list* dalam bentuk *chip*.

Pada *section chip body parts list* saat pengguna klik salah satu chipnya maka chip yang dipilih akan ditandai dengan berwarna biru dan menampilkan daftar spesialisasi/subspesialisasi pada *body part* tersebut dalam bentuk *simple-list card* yang berisikan judul, deskripsi, dan tombol panah. Lalu, Saat salah satu *card* spesialisasi/subspesialisasi tersebut diklik maka akan diarahkan ke halaman Detail Spesialisasi/Subspesialisasi.

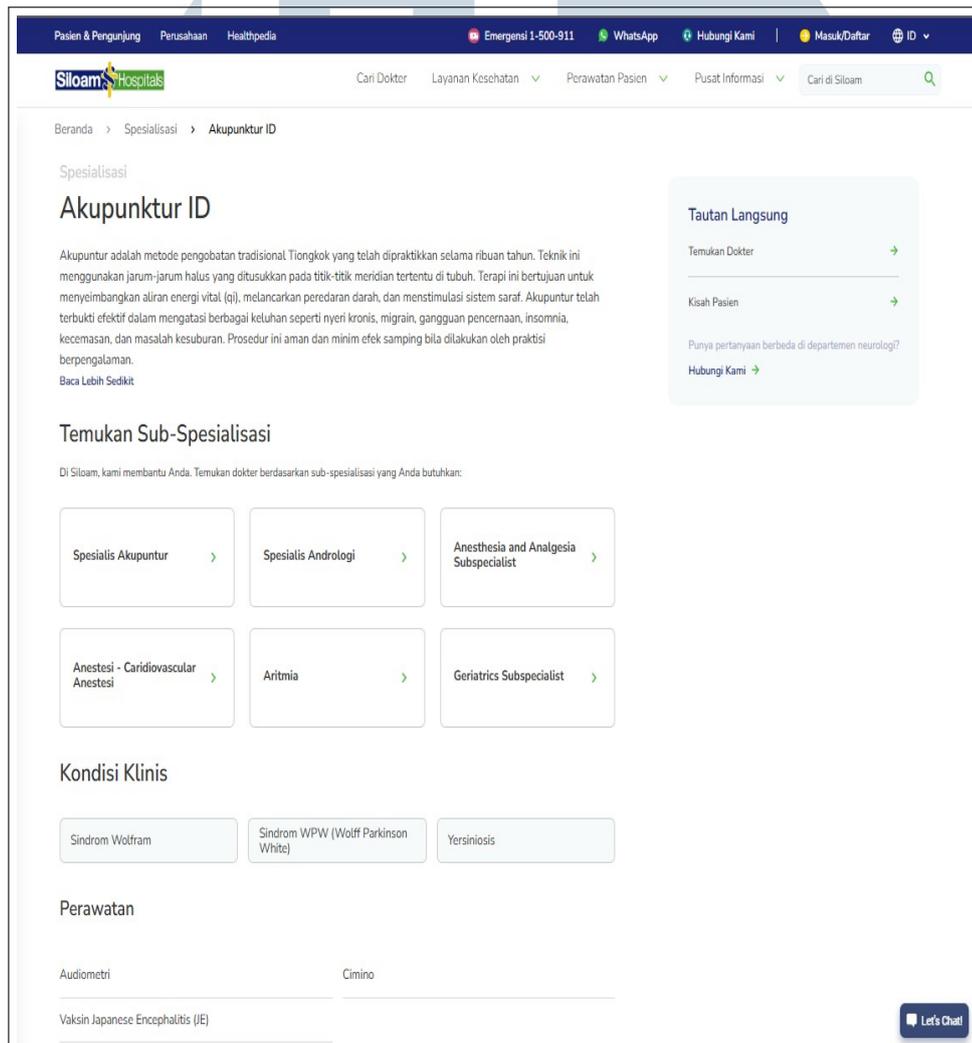
Pada *section visual model body parts* juga terdapat *body part chips* yang sudah diatur titik-titik nya sesuai dengan tempat body part tersebut berada. Chip button pada *visual model body parts clickable* dan saat diklik akan *auto select chip* disebelah kanan dan menampilkan list Specialties sesuai body part yang dipilih dan chip pada *visual model body parts* yang terpilih ditandai dengan berwarna hijau. Saat pengguna berada pada *zoom out level* tertentu di *visual model body parts* maka mappingan body part chip dibagian kepala akan berubah yang awalnya terdapat 'head', 'mouth', 'brain', 'ear', 'eye', 'nose', 'face' akan berubah menjadi "Head, Eye, +5 More". Pada kondisi ini, ketika pengguna klik chip "head, eye, +5 More" maka akan zoom in kebagian kepala dan memunculkan kembali body part chip 'head', 'mouth', 'brain', 'ear', 'eye', 'nose', 'face' dan chip yang aktif di *visual model body parts* dan *list chip* disebelah kanan akan terpilih sebagai kepala (*head*).

Kemudian, terdapat *hidden body part* yang terdapat pada body part chips disebelah kanan, namun tidak ditampilkan pada *visual model body parts*. Sehingga ketika pengguna klik salah satu *chip hidden body part* tersebut maka akan melakukan zoom out (terjadi ketika zoom in berada pada level yang memunculkan chip "Head, Eye, +5 More") tanpa adanya penanda body part yang dipilih pada *visual model body parts*.

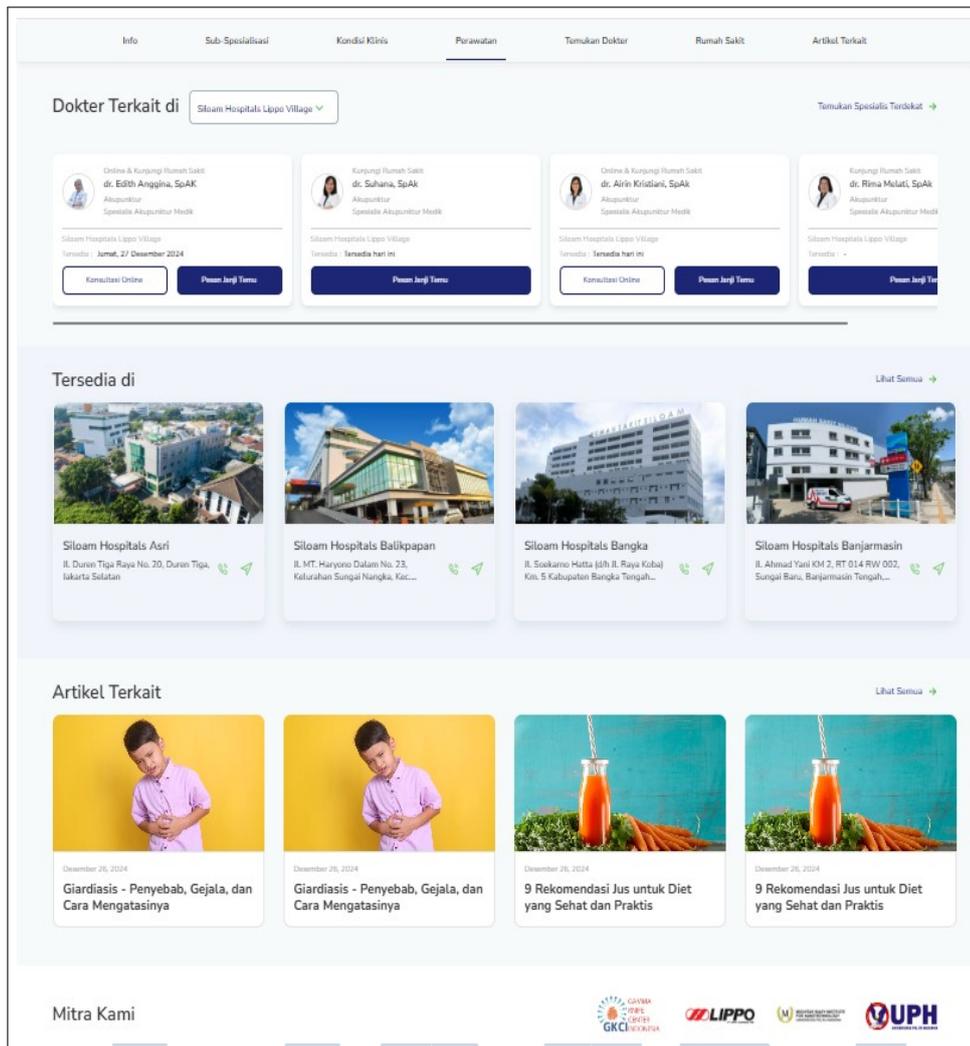


### 3.5.3 Detail Speciality

Gambar 3.17 merupakan hasil dari pengerjaan Detail Speciality. Pengerjaan halaman *Detail Speciality* dimulai dengan *layouting* dan *API integration* (ilustrasi pada gambar 3.5).



UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.17. Detail Speciality

Pada halaman *Detail Speciality* menampilkan section detail spesialisasi, seperti nama spesialisasi dan deskripsi tentang spesialisasi tersebut. Pada bagian deskripsi singkat terdapat tombol untuk membaca lebih lanjut deskripsi tentang spesialisasi tersebut, dan tombol akan berganti ketika diklik menjadi baca lebih sedikit (*read less*). Kemudian, dibawahnya terdapat *section Find Sub-Specialities* yang akan menampilkan subspecialisasi dari spesialisasi tersebut. *Section section Find Sub-Specialities* ditampilkan dalam bentuk *card list* yang berisikan judul subspecialisasi, deskripsi singkat dan *arrow icon*. Ketika salah satu *card* diklik maka akan mengarahkan pengguna ke halaman Detail subspecialisasi yang dipilih. Selanjutnya, terdapat *section Symptoms/Diseases* yang menampilkan gejala/penyakit terkait dengan spesialisasi tersebut. Ketika salah satu gejala/penyakit diklik maka akan mengarahkan pengguna ke halaman Detail

Gejala/Penyakit tersebut. Kemudian, dibawahnya terdapat *section Treatments* yang menampilkan list perawatan yang dapat dilakukan terkait dengan gejala dan penyakit spesialisasi tersebut. Disebelah kanan, terdapat *sticky section Useful Links / Quick Links* yang merupakan *card* berisi list tombol tautan untuk mengarahkan pengguna ke halaman-halaman tersebut.

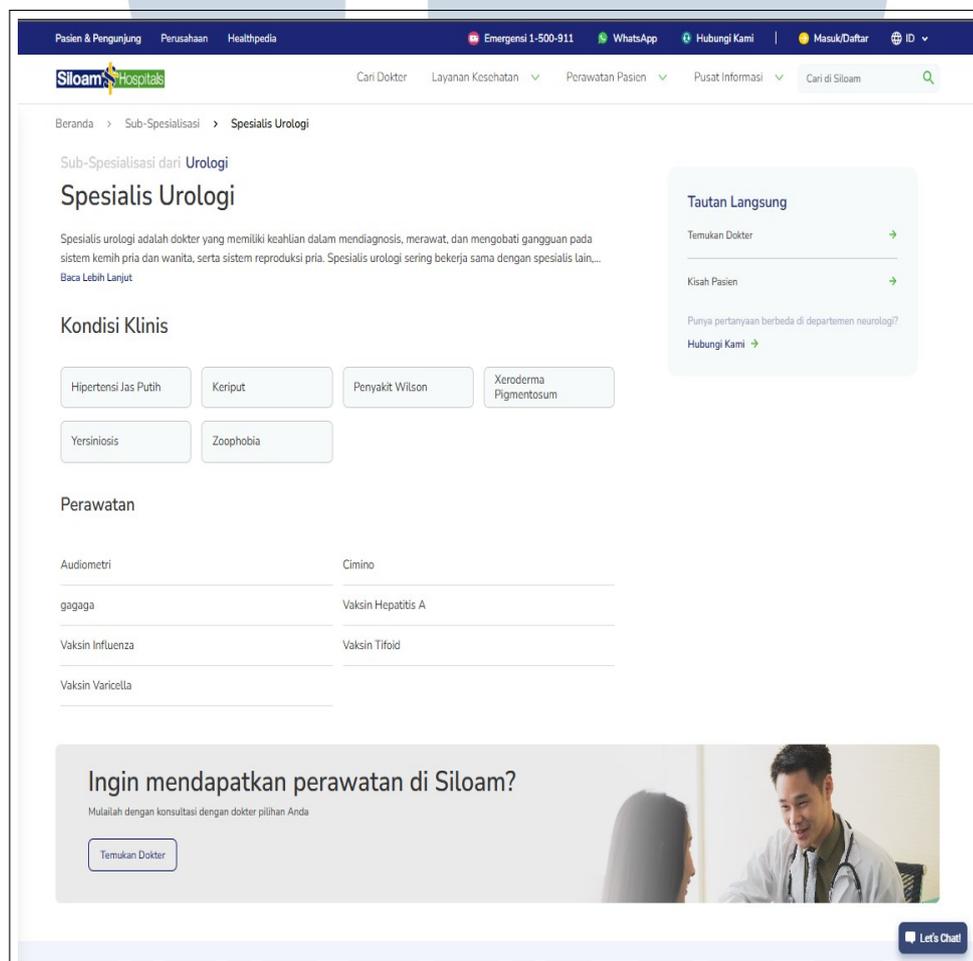
Dibawah *section* detail spesialisasi, terdapat *section Our Doctors At* yang menampilkan daftar dokter dalam bentuk *doctor card* dengan maksimal menampilkan 8 *doctor card*. Disamping judul *Our Doctors At* terdapat *filter dropdown hospital* dimana pengguna dapat memilih salah satu rumah sakit dari list yang didapatkan dari *API GET Nearby Hospitals / API GET Advanced Hospitals*. Saat pengguna mengubah filter rumah sakitnya maka daftar dokter juga akan berubah. *Doctor Card* berisi informasi tentang dokter seperti nama, spesialisasi, *primary hospital*, dan *available at* dari dokter tersebut, serta tombol untuk konsultasi online dan *book appointment*. Ketika nama, atau gambar dokter tersebut diklik, maka akan diarahkan ke halaman Detail Dokter tersebut. Ketika pengguna klik tombol konsultasi online, maka akan mengarahkan pengguna ke aplikasi MySiloam. Sedangkan, ketika pengguna klik tombol *book appointment* maka akan diarahkan ke halaman Detail Dokter untuk melakukan janji temu. Kemudian, diujung kanan terdapat tombol "find nearby doctor" ketika diklik akan mengarahkan pengguna ke halaman Cari Dokter dengan mengoper params, rumah sakit dan spesialisasi pilihan pengguna.

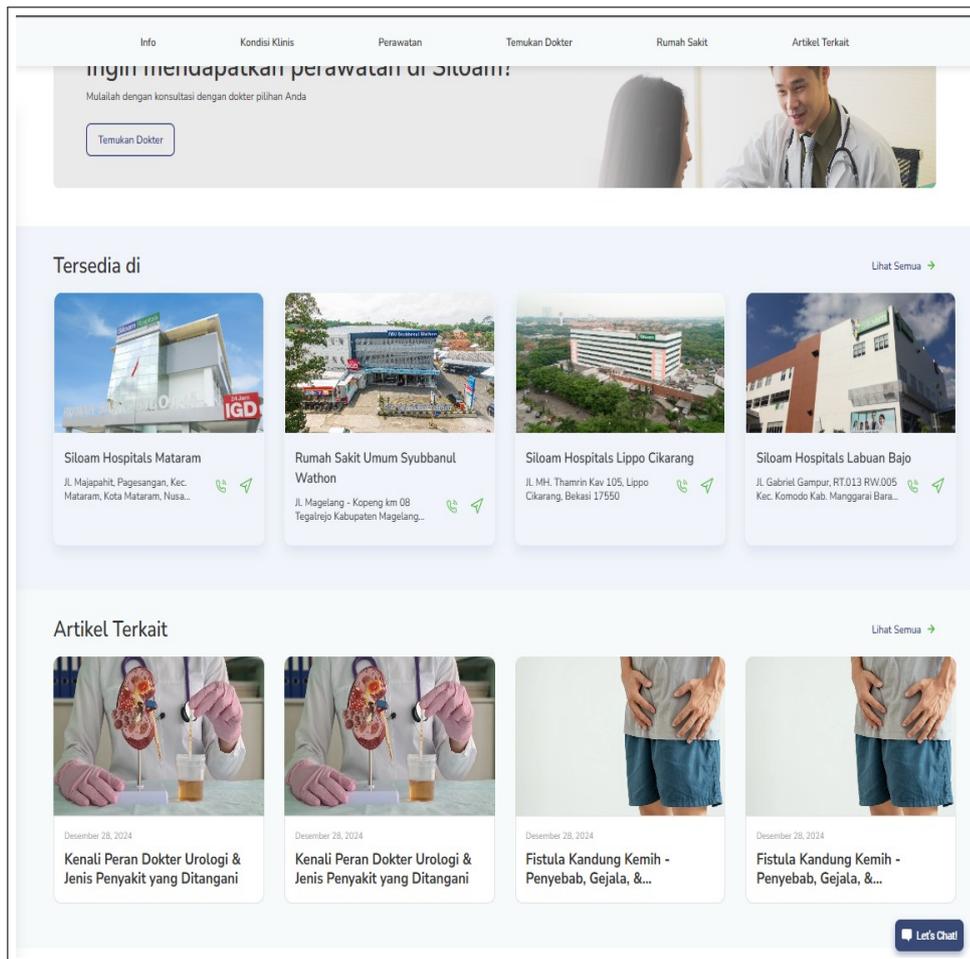
Setelah *section Our Doctors At*, terdapat *section Available At* yang menampilkan daftar rumah sakit yang didapatkan dari *API GET Nearby Hospitals / API GET Advanced Hospitals*. Ditampilkan dalam bentuk *location card* (ditampilkan maksimal 4 *card*), *location card* berisi informasi-informasi terkait daftar rumah sakit yang didapatkan seperti gambar, nama, dan alamat. Saat salah satu *location card* diklik maka akan mengarahkan pengguna ke halaman Detail Rumah Sakit. Terdapat juga *icon phone* dan *direct*. Saat pengguna klik *icon phone* maka akan mengarahkan pengguna ke WhatsApp rumah sakit tersebut / memunculkan pop up nomor telepon rumah sakit. Kemudian, ketika pengguna klik *icon direct* maka akan mengarahkan pengguna ke Google Maps lokasi rumah sakit berada. Diujung kanan juga terdapat tombol *view all* yang ketika diklik akan mengarahkan pengguna ke halaman Landing Hospitals dengan params spesialisasi dan subspecialisasi yang dimiliki oleh spesialisasi tersebut.

Dibawahnya terdapat *section Our Articles*, menampilkan 4 *Article Card* yang berisikan gambar, judul artikel, dan tanggal update artikel tersebut. Saat salah satu *card* diklik maka akan mengarahkan pengguna ke halaman Detail Article tersebut. Lalu, pada ujung kanan terdapat tombol *view all*, ketika diklik maka akan mengarahkan pengguna ke halaman Landing Healthpedia.

### 3.5.4 *Detail Subspeciality*

Gambar 3.18 merupakan hasil dari pengembangan Detail Subspeciality. Pengembangan halaman *Detail Subspeciality* dimulai dengan *layouting* dan *API integration* (ilustrasi pada gambar 3.6).





Gambar 3.18. *Detail Subspeciality*

Pada halaman *Detail Subspeciality* menampilkan *section* detail subspecialisasi, seperti nama subspecialisasi, *parent* spesialisasi, dan deskripsi tentang subspecialisasi tersebut. Pada bagian deskripsi singkat terdapat tombol untuk membaca lebih lanjut deskripsi tentang spesialisasi tersebut, dan tombol akan berganti ketika diklik menjadi baca lebih sedikit (*read less*). Kemudian, dibawahnya terdapat *section Symptoms/Diseases* yang menampilkan gejala/penyakit terkait dengan spesialisasi tersebut. Ketika salah satu gejala/penyakit diklik maka akan mengarahkan pengguna ke halaman Detail Gejala/Penyakit tersebut. Kemudian, dibawahnya terdapat *section Treatments* yang menampilkan list perawatan yang dapat dilakukan terkait dengan gejala dan penyakit spesialisasi tersebut. Disebelah kanan, terdapat *sticky section Useful Links / Quick Links* yang merupakan *card* berisi list tombol tautan untuk mengarahkan pengguna kehalaman-halaman tersebut.

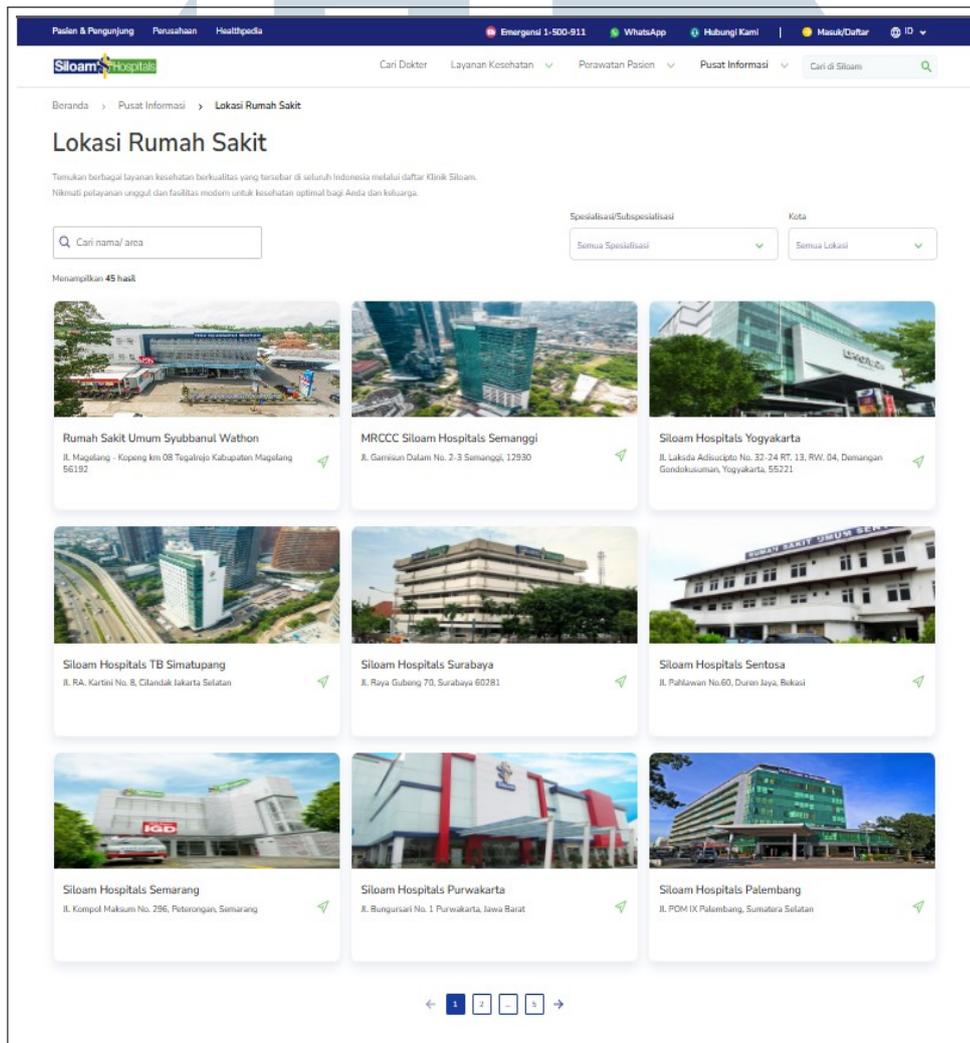
Dibawah *section* detail subspecialisasi, terdapat *section Find Doctor Banner* yang didalamnya terdapat tombol untuk mengarahkan pengguna ke halaman Cari Dokter. Kemudian, terdapat *section Available At* yang menampilkan daftar rumah sakit yang didapatkan dari *API GET Nearby Hospitals / API GET Advanced Hospitals*. Ditampilkan dalam bentuk *location card* (ditampilkan maksimal 4 *card*), *location card* berisi informasi-informasi terkait daftar rumah sakit yang didapatkan seperti gambar, nama, dan alamat. Saat salah satu *location card* diklik maka akan mengarahkan pengguna ke halaman Detail Rumah Sakit. Terdapat juga *icon phone* dan *direct*. Saat pengguna klik *icon phone* maka akan mengarahkan pengguna ke WhatsApp rumah sakit tersebut / memunculkan pop up nomor telepon rumah sakit. Kemudian, ketika pengguna klik *icon direct* maka akan mengarahkan pengguna ke Google Maps lokasi rumah sakit berada. Diujung kanan juga terdapat tombol *view all* yang ketika diklik akan mengarahkan pengguna ke halaman Landing Hospitals dengan params spesialisasi dan subspecialisasi yang dimiliki oleh spesialisasi tersebut.

Dibawahnya terdapat *section Our Articles*, menampilkan 4 *Article Card* yang berisikan gambar, judul artikel, dan tanggal update artikel tersebut. Saat salah satu *card* diklik maka akan mengarahkan pengguna ke halaman Detail Article tersebut. Lalu, pada ujung kanan terdapat tombol *view all*, ketika diklik maka akan mengarahkan pengguna ke halaman Landing Healthpedia.



### 3.5.5 Landing Hospitals

Gambar 3.19 merupakan hasil dari pengembangan *Landing Hospitals*. Pengembangan halaman *Landing Hospitals* dimulai dengan *layouting* dan *API integration* (ilustrasi pada gambar 3.7).



Gambar 3.19. *Landing Hospitals*

Pada Halaman *Landing Hospitals* menampilkan judul halaman dan deskripsi halaman. Dibawahnya terdapat *section search bar*, filter *dropdown* spesialisasi/subspesialisasi, dan kota. Kemudian, dibawahnya terdapat *section result* yang menampilkan hasil pencarian rumah sakit berdasarkan keyword, filter spesialisasi/subspesialisasi, dan kota. Hasil ditampilkan dalam bentuk *location card* yang sama pada halaman *Detail Spesialisasi*.

Pada bagian filter spesialisasi/subspesialisasi. Warna biru menandakan spesialisasi dan dibawahnya merupakan subspesialisasinya. Saat klik *checkbox* spesialisasi maka akan otomatis select semua subspesialisasinya, sebaliknya ketika klik *checkbox* spesialisasi yang sudah tercentang maka akan melakukan *uncheck* pada semua spesialisasi dibawahnya. Kemudian, pada filter juga terdapat search bar yang dapat digunakan untuk mencari spesialisasi/subspesialisasi berdasarkan *keyword*. Kemudian, terdapat *section* yang menampilkan jumlah spesialisasi/subspesialisasi yang sudah dipilih pengguna. Disampingnya terdapat tombol *reset* untuk menghapus semua spesialisasi/subspesialisasi yang sudah dipilih oleh pengguna.

Pada filter kota pengguna dapat melakukan *checked/unchecked* pada list kota yang ada. Sama halnya dengan filter spesialisasi, terdapat search bar yang dapat digunakan untuk mencari kota berdasarkan *keyword*. Kemudian, terdapat *section* yang menampilkan jumlah kota yang sudah dipilih pengguna. Disampingnya terdapat tombol *reset* untuk menghapus semua kota yang sudah dipilih oleh pengguna.

### 3.6 Pengujian

Pengujian terhadap menu Spesialisasi/Subspesialisasi telah dialokasikan kepada tim Quality Assurance (QA) untuk memastikan fungsionalitas dan performa produk sesuai dengan spesifikasi yang ditentukan per awal Desember 2024. Namun, proses pengujian belum dimulai dari tim QA menjelang akhir tahun 2024.

UIMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 3.7 Kendala dan Solusi yang Ditemukan

Kendala yang ditemukan selama mengembangkan project menu Spesialisasi/Subspesialisasi modul *Patient Centered Care*, yaitu:

1. Kurangnya kemampuan dalam manajemen waktu saat mengerjakan sebuah tugas atau *issue* yang diberikan. Sehingga menghabiskan waktu berlebih untuk mengerjakan sebuah *issue* atau *task*.
2. *User requirement* yang berubah-ubah ditengah proses *development*. *User requirement* yang berubah-ubah ditengah proses *development* mengganggu *timeline* dari proyek dan mengurangi efektivitas *developer* dalam proses pengembangan, ketika sebuah fitur telah selesai dibuat, namun diminta untuk dirubah karena *user requirement* yang berubah.

Solusi yang dapat dilakukan untuk mengatasi kendala diatas adalah sebagai berikut:

1. Memberikan batas waktu saat mengerjakan *task* atau *issue*. Sehingga, jika sudah melewati batas waktu yang ditentukan dan belum selesai mengerjakan tugas atau belum menemukan solusi dari *issue* yang dikerjakan maka akan berpindah ke task lain.
2. Meminta *user requirement* yang jelas dan bersifat final agar mengurangi adanya permintaan perubahan website ketika masih dalam tahap pengembangan. Kemudian, perlunya untuk membuat catatan setiap adanya meeting terkait penyampaian *requirement* sebuah *task*.

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