

## 2.2. Kegiatan yang Dilakukan Sesuai *Learning Plan*

Kegiatan yang saya lakukan saat 6 bulan di RSSC Adalah, mengerjakan projek-projek development software untuk membuat pelayan RSSC lebih efisien dengan bantuan teknologi berbasis software atau aplikasi. Biasanya untuk projek baru dimulai dengan mapping user requirements di sebuah word document terus merancang dan visualisasikan pembuatan aplikasi. Menggunakan diagram-diagram seperti flow chart untuk mapping flow proses aplikasi, ERD untuk mapping pembuatan database dan backend dari aplikasi, terus memilih Bahasa coding apa yang akan dipakai untuk membuat aplikasi biasanya dibagi 2 Front End dan Backend.

Kemudian kami mulai coding projek kami secara full stack setelah merancang dan memvisualisasikan projek yang kami buat, biasanya kami coding melalui Visual Studio Code, environment backend kami memakai MySQL dari Laragon dan visualisasi databasenya dari TablePlus. Dari backend kami memakai Model View Controller design pattern, biasanya kami membuat query-query untuk aplikasi di model, yang di validasi oleh controller dan dijadikan route untuk nanti dipakai di bagian frontend, Dimana dibikin menjadi route statis yang menghubungkan frontend ke backend.

Kami juga bisa memakai resources dari eRSSC seperti API, dan services seperti untuk send whats app text, validasi kode booking atau NOP, dll. Frontend kami biasanya melakukan component based architecture, Dimana page kami tersusun dari components-components yang bersifat mandiri dan melengkapi halaman tersebut, tetapi untuk halaman-halaman yang tidak kompleks kami tidak pakai components untuk melengkapi halamannya.

Setelah melakukan coding kami biasanya melakukan functional testing untuk mencari bugs di aplikasi secara local dengan membuat google sheets dengan semua fitur-fitur cara kerjanya dan expected resultnya. Kemudian kami akan upload aplikasi kami ke server melalui docker linux menggunakan CLI. Terakhirnya kami melakukan live testing untuk aplikasi-aplikasi yang memerlukan pelayanan pasien seperti antrian untuk mencari bug atau saran untuk direvisikan

### **2.2.1. Project**

#### **a) Bulan I**

untuk bulan pertama kami membantu karyawan magang dengan projeknya untuk bisa tanda tangan pdf Standard Operating Procedure (SPO) secara digital dan di save, melalui pdfJS untuk library pdfnya dan yii2 untuk aplikasinya, kami mencari tau di pdfjs bagaimana untuk draw, terus modifikasi fitur savenya supaya bisa save coretannya dan save ke database dan local, untuk nanti ditampilkan. Kemudian disambungkan pdfJS yang sudah dimodifikasi ke aplikasi yii2nya supaya bisa CRUD pdfnya.

Kemudian kami belajar cara membuat RBAC website, saya membuat backendnya via Laravel Scramble, disana kami mempelajari cara memakai JWT untuk authentikasi dan menjaga keamanan aplikasi dan prinsip prinsip RBAC.

#### **b) Bulan II**

kami melanjutkan website RBAC kami, dan mendapati projek baru yaitu sebuah form Medical Check Up untuk IGD rumah sakit, disana saya membuat figma dari medical checkup formnya, terus flowchart prosesnya dan mapping databasenya melalui ERD. Formnya bersifat website tetapi bisa dipakai untuk mobile dan tablet, setelah perancangan aplikasinya kami memakai NextJs dan bootstrap sebagai Frontend, dan Laravel PHP untuk backend aplikasi.

Form MCUnya dibagi-bagi menjadi halaman-halaman berbeda berdasarkan berapa banyak halaman di form tertulisnya yang dibagi berdasarkan hal-hal seperti identitas diri, Riwayat Kesehatan pribadi, keluarga dan keluarganya pasien tersendiri, disitu secara frontend pagesnya terdiri dari komponen-komponen terus karena beberapa pertanyaan cara jawabnya sama misalnya yes or no, kami jadi membuat sebuah form yang bisa diisi dengan pertanyaan jadinya mapping satu komponen untuk banyak pertanyaan mengurangi

jumlah komponen di aplikasi. Kemudian memakai google OAuth untuk mencoba login ke form MCU dengan gmail sekalian mengambil email dari pasiennya. Setelah membuat semua pertanyaan di form menjadi module dan bisa diisi, kami mulai membuat validasi data yang diinput di modul pertanyaannya dan kemudian membuat sebuah page setelah login untuk cek apakah user pernah mengisi formnya melalui pengecekan NIK nama dan tanggal lahir. Bila user sudah pernah mengisi form tersebut nanti semua jawaban dari database untuk user tersebut akan terisi di formnya tinggal update dan Ganti jawaban yang perlu digantikan. Kemudian kami membuat api untuk save jawaban form tersebut, jadinya databasenya dibagikan menjadi berbagai table berdasarkan halaman yang disediakan jadi ada table untuk response form identitas, Riwayat Kesehatan pribadi, dll. Kemudian kami lengkapi dengan API dan bridging ke frontend

Kemudian kami juga memulai setup untuk melanjutkan projek-projek dari karyawan magang sebelumnya seperti Tensi Mandiri dan Antrian farmasi.

c) **Bulan III**

Melanjutkan setup antrian farmasi dan tensi mandiri, Carlobot Chatbot WA, dan juga MCU.

Untuk MCU diperlukan fitur save jawaban sebagai pdf jadi memakai library jsPDF untuk membuat pdf sendiri secara custom yang dapat di download secara local setelah menjawab semua form MCU, yang kemudian kami melakukan bug fixing dan mempercantik frontend dan optimize backend sebisa mungkin untuk disimulasikan di rapat Bersama dokter-dokter.

Untuk antrian farmasi ada berbagai hal-hal penting yang terjadi selain setup farmasinya supaya bisa run di local dari karyawan magang sebelumnya, seperti bug fixing dan juga setup websocket supaya tidak perlu timer tiap kali untuk call API mengambil data ke frontend

tetapi melalui emitter dari websocket, mengutamakan frontend dinamis dan efisien secara realtime. Kemudian membuat beberapa modifikasi QOL seperti, mengambil data berdasarkan timestamp dia terbuat di database atau diupdate statusnya, membat queue untuk speech announce pasien yang dapat ambil obat, pembuatan antrian baru melalui kode booking baru FAXXXX.

Untuk tensi madiri saya mempelajari RUST karena backendnya memakai rust untuk bisa terhubung ke alat tensi mandirinya, setelah setup dan bugfixing kami langsung test di lantai 2 antrian jantung dan klinik penyakit dalam lainnya untuk melihat apakah ada saran dan masukan.

Untuk carlobot, saya mempelajari cara memakai docker CLI, cara up dan build aplikasi di docker serta mempelajari rust untuk backend carlobot.

d) **Bulan IV**

Bulan ini saya lebih focus ke projek antrian farmasi dan sedikit carlobot, untuk farmasi kami melakukan testing secara langsung di farmasi BPJS, membuat page baru disebut logs untuk melihat data heuristic antrian farmasi dan yang paling penting Kumpulan semua logging data yang masuk ke antrian farmasi dari verifikasi ke pembuatan ke pemanggilan obat ke pasien, kemudian di verifikasi mulai mengganti sistemnya dari service menjadi proses sehingga tidak memerlukan pasien datang ke loket untuk verifikasi tetapi tinggal nunggu diverifikasi. Caranya Adalah mengubah system antrian farmasi yang tadi berdasarkan kiosk BPJS menjadi seperti sebuah service ticketing, Dimana setiap antrian BPJS yang masuk akan di print keluar tracer, kemudian dapat di scan melalui barcode scanner setalah verifikasi untuk nanti disiapkan obatnya sesuai resep. Disini mulai munculnya system notifikasi WA farmasi yang memberi tau pasien status obat dan juga nomor antriannya yang sekarang diganti berdasarkan tipe obat yakni RC untuk Racikan dan NR non racikan

Kemudian kami melakukan bugfixing, optimisasi backend dan frontend communication aplikasi, mengubah display admin dan display untuk pasien sesuai dengan revisi farmasi.

Kemudian untuk carlobot kami hanya menambahkan hal-hal seperti pertanyaan baru atau teks deskripsi baru dan up ke docker.

e) **Bulan V**

Bulan ini masih melanjutkan farmasi berdasarkan revisi yang diberikan. Dan fixing kiosk bpjs.

Untuk farmasi kami mengubah booking id menjadi NOP baik di database, backend dan frontend jadi primary key barunya Adalah NOP bukan booking id, juga menambahkan berbagai visualisasi data untuk table logs, kemudian menambahkan fitur-fitur QOL di admin verif untuk mempermudah mencari antrian atau data yang diperlukan seperti search bar. Filter berdasarkan tanggal untuk page verif, proses obat dan juga admin obat untuk pemanggilan pasien mengambil obatnya. Kemudian melanjut menggantikan display farmasi supaya lebih simple dan mudah dibaca, dan juga menambahkan data-data penting seperti status obat beserta nama pemilik antrian, colour coding statusnya, misalnya hijau menunggu/dipanggil, dan merah untuk dilewatkan. Dan menambahkan fitur di admin verif untuk insert antrian baru berdasarkan NOP pasien, yang nantinya dapat pilih tipe obat dan send WA ke pasien untuk notifikasi beserta print di tracer BPJS.

Dilanjutkan dengan simulasi antrian farmasi lagi untuk mencari revisi dan masukan, setalah simulasi melakukan revisi yakni membuat website lebih realtime dan sinkron datanya saat pindah pindah halaman dan status melalui socket js, mempermudah UI baik untuk admin dan pasien, menambah fitur edit nomor telepon di admin verif sekalian send WA ke nomor yang baru diupdate, retry send WA

dan print jika terjadi error saat eksekusi functionnya, dan membuat dokumentasi progress farmasinya dengan screenshot UI dan flow successnya di word.

Kemudian untuk KIOSK BPJS kami mencari tau kenapa terjadi error crash atau force close setiap kali mau fingerprint scan atau print antrian setelah verifikasi selesai dan berhasil. Sayangnya tanggal 30 Juni untuk 1 minggu saya tidak bisa masuk kerja karena sakit penyakit dalam di bagian lambung dan pencernaan.

#### f) **Bulan VI**

Saya balik kerja pada minggu ke 2 Juli di tanggal 6 Juni. Menyelesaikan farmasi BPJS untuk simulasi terakhir. Dimulai dengan menyamakan semua komponen dari antdesign karena berbagai komponen seperti datepicker diambil dari library lain atau bikin sendiri, dan mempelajari slug di nextjs untuk mempersiapkan pemakaian antrian farmasi di tempat lain seperti LT3 dan LT1 GMCB RSSC. Sehingga diperlukan sebuah page setelah login untuk pilih ma uke farmasi bagian mana, dan menambahkan back button di sidebar untuk balik ke page Lokasi farmasi tersebut. Kemudian melengkapi dokumentasi farmasi dengan update ERD, flowchart dan dokumentasi API beserta functional testing. Lanjut bugfixing, optimisasi API call farmasi dan SocketJS, membuat backend function untuk ambil semua data atau data-data tertentu per hari atau berdasarkan tanggal tertentu untuk ditampilkan di front end, kemudian menambahkan fitur untuk send ke wa pasien atau send ke wa sendiri di sidebar dalam bentuk toggle switch untuk safe testing, kemudian kami mempelajari docker dan cara up aplikasi ke linux docker melalui CLI, setelah itu kami melakukan Simulasi Antrian Farmasi dan sampai sekarang berjalan lancar sambil revisi revisi hal-hal minor.

Kami juga mempelajari senam untuk pasien pasien di healing garden dan update deskripsi dan gambar di carlobot.

### **2.2.2. Technical Competency (TC)**

#### **a) Bulan I**

Untuk projek penandatangan SPO secara digital hubungannya terhadap kompetensi teknis saya tidak hanya dalam pengertian saya dalam Bahasa coding Javascript atau Laravel tetapi juga dalam Analisa code dan refaktor coding supaya sesuai dengan yang diperlukan oleh projek kami. Umumnya kami memakai dokumentasi dan bantuan AI untuk mengertikan code dan mengubah cara kerjanya.

Untuk program RBAC kami hubungannya terhadap skill kami dalam coding Full Stack dengan prinsip-prinsip coding yang telah dipelajari. Untuk backend kami memakai Laravel dengan memakai Scramble dan kemudian untuk projek-projek lainnya Postman untuk dokumentasi API yang merupakan hal yang penting, sebuah jurnal dari 2019 menjelaskan dampak-dampak positif API Documentation yakni, untuk mempraktekan langsung kerja dan hasil dari API yang terkumpul di dokumentasi, dan mempermudah developer untuk membaca, mengumpul, mengertikan, dan testing API yang dipakai di sebuah projek [1]. Dengan kompetensi teknis API Documentation kami bisa mengumpul dan juga mengertikan segala koneksi API di projek kami, dan juga mempermudah mempelajari segala kode error saat testing API seperti 404, 500, 400, dll.

Kemudian saya mendalami konsep Model View Controller design pattern yang menjadi design pattern umum di Laravel. Alasannya kami tetap lanjut memakai MVC meskipun ada design pattern lain di backend dapat diambil dari jurnal ini dari tahun 2022, karena sangat mudah untuk dimengertikan, memecahkan aplikasi menjadi Model, View, dan Controller, setiap bagian di MVC mempunya

tujuannya sendiri dan tidak saling campur tetapi terhubung untuk melakukan tujuan tertentu. Sehingga aplikasi mudah untuk di modifikasi, dan menjaga keamanan dari backend aplikasi dengan abstraction. [2]

Dari segi frontend hubungannya terhadap kompetensi teknis yang saya miliki lebih cenderung ke menghubungkan Frontend dengan Backend melalui API route, karena saya di projek ini lebih cenderung menjadi backend developer. Saya memakai Axios untuk membantu hal tersebut, jadinya ada sebuah folder yang mengumpul semua API routes dari backend ke frontend, nanti jika sebuah web atau komponen memerlukannya tinggal panggil function yang diperlukan untuk trigger Axios call. Alasan kami memakai Axios karena dia mempermudah komunikasi dengan REST API yang telah kami buat dan dokumentasi, dan bisa dipakai di backend atau frontend, axios juga bisa membuat operasi CRUD, menerima dan merubah request maupun response ke dalam bentuk JSON untuk mempermudah pembacaan data, disini kami mempelajari cara-cara lebih efisien untuk melakukan API call, yakni membuat folder untuk mengumpulkan routes dari backend yang telah disambung dengan axios untuk dipakai di component frontend, dll. [3]

Projek RBAC juga memiliki hubungan erat dengan Software Design melalui UML. Merupakan salah satu tahap dari SDLC yakni design kami membisualisasikan aplikasi kami melalui diagram seperti UML. Tahap ini sangat penting karena dapat memvisualisasikan requirement dari aplikasi kami untuk mempermudah development, kami biasanya mapping flow dari aplikasinya lewat flowchart, UI dari Figma dan database backend melalui ERD, disini kami diajari oleh karyawan magang system analyst cara mudah

untuk proses requirements jadi UML, dan cara efektif mapping database ke dalam ERD, dll. [4]

b) **Bulan II**

Untuk bulan kedua kompetensi teknis yang terhubung dengan projek baru saya di MCU ada pembuatan UI/UX, dan visualisasi software memakai UML untuk bagian perancangan. Untuk pembuatan UI/UX saya mengutamakan membuat sebuah form yang mudah dibaca dan dimengertikan pasien, dan yang paling penting Adalah menjaga supaya pasien tidak melewati pertanyaan-pertanyaan penting dan dapat mencari tau di pertanyaan mana yang perlu diganti, alias formnya harus responsive dan mudah dibaca, mengikuti prinsip-prinsip utama dari UI/UX design, peningkatan yang saya alami Adalah lebih ke pergantian prinsip dari membuat website yang bagus dan eye catching ke website yang simple dan mudah dibaca, karena user-user kami tidak terlalu paham teknologi, dan tidak terlalu memikirkan design tetapi yang penting Adalah mudah untuk dimengertikan. Jadinya kami mengadaptasi cara kami membuat aplikasi mengutamakan UX disbanding UI yang unik dan cakep. [5].

Dari segi development salah satu kompetensi teknis yang terlatih Adalah Component Driven Design. Alasannya saya memakai Component Driven Design untuk MCU karena formnya memiliki banyak pertanyaan dan jikalau tidak dibagi dikomponen maka kodennya akan menjadi “bloated” jadinya kami memakai komponen yang bersifat “reusable” untuk menjadi modul pertanyaanya formnya mempermudah efisiensi dan memudahkan implementasi karena fungsinya tertera di komponennya sendiri, peningkatan yang saya lihat Adalah lebih pintarnya saya memakai komponen secara efektif dan jumlah dikit untuk membuat aplikasinya dan

bagaimana saya bridging antara PHP Laravel dan NextJS frontend untuk melakukannya[6].

c) **Bulan III**

Untuk bulan ketiga, saya sudah mulai develop projek-projek banyak seperti MCU, Antrian Farmasi, Tensi Mandiri dan Carlobot, yang setiap projek memiliki hubungan dengan kompetensi teknis tertentu.

Untuk Antrian Farmasi seperti biasa mengasah ke kompetensi teknis saya dalam Component Driven Development, MVC design pattern dan juga satu hal penting yaitu cara membuat website kami dinamis atau dapat mengupdate data secara realtime.

Kenapa kami memilih Antrian Farmasi kami bersifat realtime dengan Websocket? Supaya petugas dan Masyarakat dapat memantau dan mengelola antrian secara langsung tanpa latensi atau jeda waktu lama antara informasi baru dengan informasi lama. Sehingga, system ini memungkinkan pengguna mendapatkan antrian dan melihat perkembangan secara langsung, dan mempermudah petugas dalam memanggil, dan memproses obat meningkatkan efisiensi layanan, peningkatan saya di kompetensi ini menurut saya sangat drastic, tak hanya saya mempelajari prinsip baru yaitu memakai Websocket untuk update data secara realtime, tetapi saya mempelajari banyak tentang Websocket, dari cara efektif memakainya dan pula ke setupnya [7].

Cara kami melakukan hal tersebut Adalah memakai Websocket, alasannya kami memakai websocket karena “Websocket, dengan koneksi mereka yang bersifat 2 arah, selalu menyala, lebih sesuai untuk menjaga respon yang konsisten dan low latency setelah koneksi terbuka,

membuat mereka pas untuk aplikasi real time yang mengutamakan daya tanggap.” Mengutip dari jurnal membandingkan Websocket dengan HTTP polling dari Ibom State Polytechnic. Jadinya kami belajar cara setup Websocket di frontend dan backend, dan cara mereka melakukan realtime pollingnya yakni dengan 2 step penting, emit dan receive. Jikalau dari backend sebuah socket sedang emit data atau pesan, dari frontend dapat menerima emit tersebut dengan socket.on dan mematikan dengan socket.off jadinya tidak perlu timer untuk cek dan ambil data terus dari API tetapi hanya taruh emitter socket js di tempat yang tertentu dan di receive di tempat tertentu [7].

Untuk Tensi Mandiri, ada hubungan dengan kompetensi teknis IoT Integration, ini karena saya develop software menggunakan RUST dan React untuk membaca tensi pasien saat diukur dan disimpan di dalam database.

Alasan kenapa kami memakai RUST untuk backend ini adalah karena RUST tidak memakai banyak memory CPU, sehingga bersifat efisien, ini sangat penting karena computer-computer untuk pelayanan rumah sakit tidak bersifat “high-end”, sebuah riset yang membandingkan berbagai Bahasa coding yang dipakai untuk Backend development dari Tampere University Of Applied Sciences di Finlandia, menyatakan bahwa RUST memakai hanya 0.5% CPU dan performanya sangat efisien dan bagus meskipun pembuatnya tidak mengerti RUST dan menurut dia dapat ditingkatkan lagi dengan pengalaman lebih banyak dalam RUST, sehingga RUST disarankan jikalau perlu melestarikan memory computer, tetapi susah untuk dipelajari dan perlu pengalaman yang dalam terhadap RUST untuk mendapatkan hasil yang sangat efisien, peningkatan kompetensi teknis yang saya alami Adalah di

cara menghubungkan alat tensi mandiri Omron ke software dengan cara memakai USB-B atau usb untuk menghubungkan printer ke PC dan RUST backend mengambil port device dan membuat fungsi untuk koneksi dan mengambil kode error dan hasil tensinya [8].

Untuk projek carlobot chatbot, kompetensi teknis yang memiliki hubungan dengan projeknya lebih kearah penggunaan linus dan docker untuk deploy aplikasi, tetapi hanya untuk update projek karena di projek ini saya hanya edit-edit deskripsi tidak deploy projek atau membuat projeknya, sehingga teori tentang kompetensi teknisnya saya jelaskan di bulan ke 6.

Untuk MCU kompetensi masih tetap sama seperti bulan sebelumnya, hanya sekarang menambahkan library pdfJS untuk dokumentasikan jawaban dari formnya menjadi sebuah PDF. Alasan kami memakai pdfJS adalah karena bersifat flexible dan modular, PDFnya bisa diganti sesuai dengan yang diinginkan dan memiliki metode-metode untuk integrasi dengan aplikasi web di browser-browser seperti chrome, safari, edge, dll. pdfJS juga mudah untuk melihat dokumen pdf tanpa software external, dapat edit dan anotasi pdfnya dan bersifat efisien dalam rendering pdfnya. Kompetensi teknis baru yang saya pelajari Adalah cara customize PDF di pdfJS [9].

#### d) **Bulan IV**

Untuk kompetensi teknis bulan keempat saya mulai lebih fokus ke Antrian Farmasi, disini kami melakukan testing dan jugal development seperti biasa.

Dari segi development ada tambahan baru dalam kompetensi teknis yaitu pembuatan halaman untuk display dan pengumpulan data heuristik atau disebut Data Visualization. Data visualization ini bersifat penting karena dapat menarik informasi dari data yang mentahnya adalah angka-angka menjadi sesuatu yang kami bisa mengerti dengan mudah, mencari anomali, melihat trend-trend seperti rata-rata jam kerja layanan, total antrian per hari dan dapat membuat solusi atau rencana untuk improve layanan secara heuristik, sehingga kami membuat dashboard yang isinya merupakan visualisasi data-data penting terhadap layanan farmasi dan data semua antrian yang masuk ke dalam antrian per hari, dan dapat melihat data dari jangka waktu tertentu., disini saya mempelajari cara efektif membuat dashboard untuk menampilkan data secara efektif dan cara membuat query database dan menghubungkan Frontend dengan Backend untuk mengambil data berdasarkan tanggal tertentu atau jangka waktu tertentu [10,11,12].

Kemudian, ada kompetensi teknis dalam integrasi print automatis saat masuknya antrian baru yang disebut Event Driven Architecture, ini karena aplikasi kami yang terdiri dari berbagai komponen memerlukan event-event tertentu untuk beraksi seperti pergantian status ke verifikasi obat ke penyiapan obat yang nanti disertakan whats app yang menunjukkan bahwa obat sedang diproses, event dimana obat ganti status menjadi terpanggil dan log waktu dipanggilnya beserta diumumkan melalui text to speech, dll. Disini event driven architecture dipakai, karena EDA merupakan prinsip dimana komponen-komponen sistem dapat mengeluarkan event dan memproses event sebagai signal perubahan di sistem, sistem ini mengutamakan

komunikasi secara asinkron artinya komponen dapat beroperasi secara mandiri dan responsif, alhasilnya adalah software yang fleksibel, modular, dapat memproses event secara bersamaan mengurangi bottleneck dan respons realtime terhadap perubahan dan, akhirnya berkaitan dengan pentingnya real time updates melalui Websocket di bulan kemarin [13].

e) **Bulan V**

Kompetensi teknis yang terhubung dengan projek saya melanjutkan Antrian Farmasi lebih ke development fitur-fitur Quality Of Life atau Human Computer Interaction, seperti search, filter, retry print, menambah antrian berdasarkan NOP, mengedit nomor telpon secara langsung, dll. Peningkatan yang saya rasakan Adalah cara untuk membuat fitur-fitur tersebut dengan NodeJS dan NextJS dan diintegrasikan melalui websocket bila ada perubahan terhadap data seperti menambah antrian, update nomor telpon supaya bisa update real time.

Alasan kami menambah fitur-fitur tersebut Adalah berdasarkan revisi dari testing yang kami lakukan, yakni diperlukan cara untuk memudahkan mencari pasien di admin, cara untuk melihat data di hari-hari sebelumnya,, sebagai Solusi terhadap masalah nomor telepon pasien yang tidak sesuai dengan database, dan mengatasi antrian yang belum masuk atau tidak memakai BPJS.

Pentingnya penambahan fitur-fitur ini bisa diambil dari prinsip-prinsipnya Human Computer Interaction dan UI/UX design. Yakni, software anda harus focus terhadap keperluan user, dan kebiasaan user, design-designnya harus konsisten dan interaksinya harus jelas, respons terhadap input user harus jelas dan langsung, fitur-fiturnya harus jelas

dan membuat user lebih efisien dalam penggerjaan dan bisa dipakai siapapun, dan yang paling pentng yakni untuk mengatasi error dan cari cara untuk mengembalikkan user dari fase error ke normal, makannya ada print lagi, edit nomor telepon dan tambah antrian yang berguna sebagai Solusi untuk masalah-masalah dari revisi sebelumnya [14].

f) **Bulan VI**

Pada bulan keenam sudah mulai deploy antrian farmasi saya di Docker, disini kompetensi teknis baru yang saya pelajarin adalah cara deploy aplikasi dengan docker melalui linux, saya mempelajari linux, syntax docker dan cara deploy lewat docker. Disini saya memakai Putty menghubung ke server dan lewat github CI/CD clone dan pull projek saya ke server, dan setelah konfigurasi project saya dengan docker yaml, dockerfile, dll saya berhasil membuat container untuk backend dan frontendnya melalui build compose up dan aplikasinya berjalan dengan sangat cepat dibanding lewat local machine saya.

Alasan kami memakai docker untuk deploy software kami adalah, karena docker dapat menjalankan aplikasi secara konsisten di environment berbeda misalnya testing, development, production, tanpa modifikasi ke infrastruktur aplikasinya, dan karena kontainer memakai OS yang sama jadi lebih ringan dan cepat dibanding virtual machine mempercepat aplikasi dan menjaga efisiensi, dan yang paling penting adalah CI/Cdnya artinya aplikasi bersifat modular dan fleksibel [15]

### **2.2.3. *Employability and Entrepreneurial Skill (EES)***

#### **a) Bulan I**

Soft skills yang dipakai untuk projek saya pada bulan ini Adalah critical and creative thinking untuk project menyimpan tanda tangan pada pdf Standard Operating Procedure, ini karena saya harus integrasi sebuah library javascript ke dalam program yang berbahasa PHP yakni Yii yang memerlukan Tingkat problem solving yang tinggi, dan pengalaman dalam berbahasa javascript dan PHP, dan juga disaat saya mencoba untuk mengubah library pdfJS supaya bisa save coretan di viewernya dan mengubah function saveny menjadi save ke database secara Base64, ini memerlukan keahlian refactoring dan critical thinking supaya dapat mencari tau function mana yang perlu di ubah dan kreatifitas Dimana cara kami mengubah fungsinya yang dari menyimpan ke local menjadi ke database. Sehingga terjadi peningkatan di keahlian saya dalam code refactoring yang sekaligus meningkatkan keahlian saya di soft skill critical thinking dan creative thinking.

Kemudian untuk project RBAC, soft skill yang diperlukan Adalah collaboration karena di projek ini tugas saya adalah Backend development, dan teman saya Kennard sebagai Frontend development sehingga perlu komunikasi yang kuat antar kedua pihak supaya projek berjalan dengan benar, ini karena dari ramotion, kerjanya front end dan backend saling berhubungan, frontend memerlukan backend untuk data dan fungsionalitas, dan backend memerlukan requirementsnya dari front end. Sehingga komunikasi dan kolaborasi yang efektif sangat diperlukan supaya saling mengerti requirements projectnya. Peningkatan yang saya mengalami Adalah Pelajaran terhadap pentingnya membuat dokumentasi Bersama, code review Bersama dan sharing ide Bersama untuk mencari tau manakah metode paling pas untuk projek kami.

b) **Bulan II**

Soft skill yang terpakai di bulan ini adalah adaptibilitas dan digital dan technology fluency karena di bulan ini saya memegang 2 project yang memakai bahasa-bahasa coding yang berbeda dan konteksnya berbeda.

Dari segi adaptability, ini penting karena saya harus mengertikan cara membuat sebuah website yang bertujuannya untuk melakukan role based access control, dan pembuatan formulir medical check up dimana kedua project tidak sama konteksnya, satu merupakan projek formulir simple yang bersifat web dan satunya merupakan website untuk manage user-user punya hak melalui role based access control, sehingga perlu adaptasi di cara saya pertamanya membuat projek RBAC ke membuat formulir yang simple. Peningkatan yang saya rasakan adalah cepatnya saya merancang dan mendevelop kedua app sekaligus, jadinya saya develop kedua projek secara paralel karena yang RBAC sudah mendekati penyelesaian, dan merancang UML beserta codingkan formulir medical check up sekaligus.

Dari segi Digital dan Techonology Fluency, ini lebih ke pengalaman saya develop kedua app yang tidak seiring bahasa codingnya, mereka kedua memakai PHP laravel backend tetapi tidak untuk frontend, memakai NextJS dan React untuk frontendnya, beserta itu dengan disarankan memakai beberapa library dari Manager saya dan karyawan magang sesama seperti Ant Design, JWTAuth, dll. Saya merasa mendapat peningkatan dalam pengetahuan saya dalam integrasi berbagai library dalam aplikasi saya dan prinsip-prinsip penting seperti JWTAuth, RBAC, Component Driven Design.

c) **Bulan III**

Untuk bulan ini saya mulai mengerjakan banyak projek, ini dikarenakan karyawan magang yang develop aplikasinya sudah tidak lagi bekerja di SIRS sehingga perlu ada yang menggantikannya. Disini soft skill initiative sangat penting di bulan ini karena harus ada

yang melanjutkan projeknya dan setup projeknya lagi sehingga kami berinitiatif untuk setup projeknya dan melanjutkan yang sudah ada meskipun kami juga memiliki projek-projek sendiri. Peningkatan skill yang saya rasa lebih ke hasil dari initiatifnya, disini saya mempelajari banyak hal seperti cara update app di docker, lebih cepat untuk analisa coding dan mencari tujuannya, dan akhirnya soft skill saya tentang Adaptability dan Digital Fluency lebih meningkat karena mempelajari hal-hal baru seperti Node JS backend, strukturnya bagaimana sama cara setupnya bagaimana, mempelajari library whatsappwebJS untuk carlobot, sekaligus melanjutkan projek kami yang sebelumnya.

d) **Bulan IV**

Untuk bulan ini karena saya diprioritaskan untuk melanjutkan Antrian Farmasi, disini soft skill yang sangat terpakai adalah collaboration, disini saya komunikasi ketat bersama pihak farmasi terutama Dokter Natania, dan manager saya Pak Martinus, untuk melakukan problem solving dan menggantikan requirements antrian farmasi supaya lebih sesuai dengan yang diinginkan pasien dan farmasi. Sehingga, adanya perubahan sistem antrian dan nomor antrian, dll. Disini saya merasa lebih tidak tegang untuk berkomunikasi dengan kedua pihak dan berani memberi solusi dan mempresentasikan software kami yang sedang dikerjakan.

Soft skill kedua yang penting adalah adaptability, karena sering terjadinya perombakan atau perubahan sistem-sistem penting di software kami sehingga perlu adaptabilitas yang kuat dan ide-ide yang kreatif untuk menyelesaikan perubahannya. Peningkatan soft skill yang saya alami adalah lebih cepatnya saya mencari solusi-solusi untuk integrasi sistem-sistem baru dan mudahnya saya untuk merubah sistem yang saya ada untuk mengakomodasi requirement-requirement baru.

e) **Bulan V**

Soft skill yang dipakai di bulan ini masih sama seperti bulan sebelumnya dengan penambahan critical thinking and creative thinking, karena di bulan ini lebih mementingkan penambahan fitur-fitur Quality of Life untuk antrian farmasi baik dari pihak Farmasi atau Pasien. Jadi munculnya usulan dari Pak Martin untuk sebuah dashboard yang memvisualiskan data di bulan sebelumnya yang dilanjutkan, dan fitur-fitur seperti filter by calendar, search by name, layout baru untuk pasien supaya mudah untuk dibaca. Semua ini memerlukan critical dan creative thinking untuk mencari tau bagaimana menunjukkan informasi penting dalam ruang sedikit mungkin atau cara mudah untuk mencari data mempercepat layanan.

Sehingga peningkatan yang saya merasakan lebih ke area kreatifitas, bagaimana saya mempermudah pasien atau farmasi untuk mengertikan UI/UX antrian farmasi, baik melalui colour coding berdasarkan status, menunjukkan nama, dll.

f) **Bulan VI**

Soft skills pada bulan keenam sama seperti bulan kelima dengan penambahan social awareness, social awareness disini lebih ke mengajarkan pihak-pihak farmasi cara memakai aplikasi farmasinya dan bekerja sama dengan farmasi saat live testing, dan juga membantu mengedukasi pasien-pasien BPJS cara kerja sistem Wanya. Peningkatan yang saya rasa adalah lebih lancar dan rileks saya dalam membantu dan menjelaskan aplikasinya ke orang lain.