

BAB I

PENDAHULUAN

A. Latar Belakang

Era perkembangan perguruan tinggi saat ini menuntut adanya pengalaman praktik magang bagi mahasiswa guna meningkatkan keterampilan dan pemahaman praktis mereka dalam dunia kerja. Dalam konteks ini, banyak mahasiswa yang mengambil kesempatan untuk menjalani magang di berbagai industri, salah satunya adalah di Bank Syariah Indonesia. Magang di lembaga keuangan ini dianggap sebagai langkah awal yang penting bagi mahasiswa untuk mendapatkan pelatihan praktis serta pemahaman mendalam tentang dunia kerja, khususnya dalam sektor keuangan syariah.

Namun, di awal proses magang, seringkali mahasiswa dihadapkan pada tantangan yang memerlukan pembelajaran baru. Salah satu contoh nyata adalah kebutuhan untuk memperluas pengetahuan dan keterampilan, yang diawali dengan pembuatan sebuah proyek kecil bernama "Digibank". Proyek ini menjadi langkah awal yang menantang namun bermanfaat bagi mahasiswa magang untuk mulai belajar, menciptakan, dan mengimplementasikan konsep-konsep baru terkait penggunaan teknologi untuk perbankan.

Pengalaman magang ini bukan hanya memberikan mahasiswa pengalaman praktis, tetapi juga menjadi kolaborasi yang menghasilkan sesuatu yang bermanfaat bagi Bank Syariah Indonesia. Ini menjadi fondasi bagi mahasiswa untuk lebih memahami pentingnya kolaborasi, inovasi, dan kontribusi dalam mengembangkan lingkungan kerja yang inklusif serta progresif. Magang ini tidak hanya mengasah keterampilan praktis, tetapi juga menanamkan nilai-nilai kolaborasi serta komitmen untuk terus belajar dan memberikan kontribusi kepada orang lain.

Permasalahan yang dihadapi di Bank Syariah Indonesia adalah kebutuhan akan aplikasi yang dapat memfasilitasi aktivitas para insan, insan BSI merupakan karyawan aktif yang berkerja di bawah BSI, di dalamnya untuk memperoleh pemahaman yang lebih mendalam tentang seluk-beluk BSI. Aplikasi BSI PINTER bukan hanya berfungsi sebagai alat pembelajaran, melainkan juga menjadi wadah

yang mengembangkan literasi bagi para Insan BSI secara menarik. Melalui fitur-fitur seperti feeds product knowledge, artikel-artikel edukatif, serta tantangan dan penghargaan, aplikasi ini mendorong pengguna untuk terus menggali pengetahuan terkait produk dan layanan yang ditawarkan oleh bank.

Insan BSI menghadapi beberapa kendala dalam proses bisnis aplikasi di PT. Bank Syariah Indonesia. Pertama, bagi yang belum mengetahui materi quiz yang dikirim melalui grup WhatsApp, pengirim quiz harus mencari kembali materi tersebut, memakan waktu dan tidak efisien. Kedua, untuk memahami produk dan jasa Bank BSI, Insan BSI harus membaca dokumen SPB yang memiliki jumlah halaman yang sangat banyak. Akses ke informasi ini juga terbatas, karena hanya dapat dilakukan melalui LAN di kantor. Ketiga, dalam keterlibatan media sosial, Insan BSI kesulitan mengetahui apakah mereka sudah melakukan like atau share terhadap konten Bank BSI, karena tidak ada indikator khusus. Keempat, pada kegiatan seperti quiz atau event, masih terjadi absensi manual di grup WhatsApp, menyulitkan pemantauan siapa yang sudah dan belum mengikuti. Kelima, meskipun informasi dan event dikirimkan melalui grup WhatsApp oleh PIC grup, risiko terlewatnya informasi meningkat saat grup ramai dengan chat. Kesemuanya menunjukkan kebutuhan akan solusi yang efektif dan terpusat untuk meningkatkan efisiensi dan keterlibatan Insan BSI. Oleh karena itu sesuai dengan judul di atas laporan ini akan membahas mengenai fitur feeds pada aplikasi BSI PINTER.

B. Identifikasi Masalah

Berdasarkan latar belakang yang telah disusun maka dapat dibuat identifikasi masalah dari praktik magang ini yaitu sebagai berikut.

1. Jika insan BSI diminta untuk like atau share media sosial Bank BSI, indikator apa yang digunakan untuk mengetahui bahwa insan BSI sudah like atau share?
2. Quiz atau event yang harus diikuti, terkadang masih dilakukan absensi manual di grup Whatsapp untuk mengetahui siapa yang sudah dan siapa yang belum mengerjakan quiz.
3. PIC grup mengirimkan informasi dan event melalui grup Whatsapp. Namun, jika banyak chat di grup tersebut, informasi event tersebut dapat terlewatkan.

C. Batasan Masalah

Adapun batasan masalah pada kegiatan praktik magang ini dengan membuat rancang bangun aplikasi backend untuk fitur Feeds dan Challenge, yaitu diantaranya.

1. Pengembangan aplikasi Aplikasi backend hanya untuk fitur feeds.
2. Praktik magang dilakukan pada bidang pengembangan IT development Group Bank Syariah Indonesia.

D. Rumusan Masalah

Berdasarkan identifikasi masalah tersebut, maka terdapat beberapa rumusan masalah yaitu diantaranya:

1. Bagaimana cara melakukan pengembangan aplikasi backend fitur feeds pada aplikasi BSI PINTER.
2. Bagaimana cara merancang aplikasi backend untuk BSI PINTER dengan efisien dan dapat digunakan.

E. Tujuan Praktik Magang

Tujuan dilaksanakannya kegiatan praktik magang yaitu sebagai berikut.

1. Mengembangkan fitur feeds untuk BSI PINTER *backend*.
2. Mengembangkan fitur pendukung feeds seperti *filtering* dan pengkategorian feeds dan fitur seperti social media *like*, komentar dalam *backend* feeds BSI PINTER

F. Manfaat Praktik magang

Adapun manfaat dilaksanakannya kegiatan praktik magang yaitu sebagai berikut.

1. Bagi Mahasiswa
 - a. Mengasah dan mengimplementasikan keterampilan dasar yang telah didapatkan selama masa perkuliahan ke dunia industri.
 - b. Menghasilkan sebuah aplikasi backend fitur feeds.
 - c. Mendapatkan pengalaman untuk dapat mempersiapkan diri ketika memasuki dunia kerja.

- d. Mengetahui dan mempelajari pekerjaan seseorang yang bekerja pada bidang pengembangan aplikasi *backend* di PT. Bank Syariah Indonesia.
 - e. Dapat mengetahui tentang cara kerja dan juga sistem kerja di dunia industri yang di terapkan di PT. Bank Syariah Indonesia.
 - f. Mealakukan perancangan dengan efisien.
 - g. Menambah *skill* dalam aspek pengembangan aplikasi *backend*.
 - h. Mengetahui cara pengembangan prerangkat lunak dengan metode Agile.
2. Bagi Instansi Perusahaan
- a. Membangun hubungan atau relasi yang baik dan adanya pertukaran informasi.
 - b. Membantu perusahaan dalam membangun aplikasi BSI PINTER.
 - c. Memperoleh bantuan tenaga kerja tambahan selama masa magang berlangsung di PT. Bank Syariah Indonesia.
3. Bagi Universitas
- a. Mendapatkan relasi untuk menyediakan tempat magang bagi para mahasiswa.
 - b. Memiliki penilaian kemampuan terhadap mahasiswa sebagai praktik dari ilmu yang didapat selama masa perkuliahan.
 - c. Sharing ilmu antar mahasiswa di Universitas.

BAB II

GAMBARAN INSTANSI

A. Umum

Pada bagian ini dijelaskan mengenai sejarah perusahaan, visi dan misi serta alamat dan kontak perusahaan pada PT. Bank Syariah Indonesia.



Gambar 2. 1 Logo PT. Bank Syariah Indonesia

1. Sejarah Perusahaan



Gambar 2.2 Gedung PT. Bank Syariah Indonesia

Industri perbankan di Indonesia mencatat tonggak sejarah baru dengan berdirinya PT Bank Syariah Indonesia Tbk (BSI), yang secara resmi didirikan pada 1 Februari 2021 atau 19 Jumadil Akhir 1442 H. Peluncuran bank syariah terbesar di Indonesia ini dipimpin langsung oleh Presiden Joko Widodo di Istana Negara

BSI merupakan hasil dari penggabungan antara PT Bank BRISyariah Tbk, PT Bank Syariah Mandiri, dan PT Bank BNI Syariah. Otoritas Jasa Keuangan (OJK) secara resmi mengeluarkan izin untuk penggabungan tiga bank syariah tersebut pada 27 Januari 2021 melalui surat Nomor SR-3/PB.1/2021. Kemudian, pada 1 Februari 2021, Presiden Joko Widodo secara resmi meresmikan keberadaan BSI.

Pemegang saham utama BSI terdiri dari PT Bank Mandiri (Persero) Tbk dengan 50,83%, PT Bank Negara Indonesia (Persero) Tbk dengan 24,85%, dan PT

Bank Rakyat Indonesia (Persero) Tbk dengan 17,25%. Sisanya dimiliki oleh pemegang saham lain dengan kepemilikan masing-masing di bawah 5%.

Melalui penggabungan ini, BSI menggabungkan kelebihan dari ketiga bank syariah tersebut, memberikan layanan yang lebih komprehensif, jangkauan yang lebih luas, dan memiliki modal yang lebih solid. Dengan dukungan sinergi perusahaan dan komitmen pemerintah melalui Kementerian BUMN, BSI didorong untuk bersaing di tingkat global.

BSI dianggap sebagai upaya menciptakan bank syariah yang menjadi kebanggaan umat, diharapkan menjadi pendorong pembangunan ekonomi nasional, serta memberikan kontribusi positif pada kesejahteraan masyarakat luas. Keberadaan BSI mencerminkan citra perbankan syariah di Indonesia yang modern, universal, dan memberikan manfaat bagi seluruh alam (Rahmatan Lil 'Aalamiin).

Potensi BSI untuk terus tumbuh dan menjadi bagian dari kelompok bank syariah terkemuka di tingkat global sangat terbuka. Selain kinerja yang positif, dukungan pemerintah Indonesia terhadap ekosistem industri halal dan bank syariah nasional yang besar dan kuat, serta fakta bahwa Indonesia memiliki populasi muslim terbesar di dunia, menjadi peluang besar bagi perkembangan BSI.

Dalam konteks ini, kehadiran BSI menjadi sangat penting, bukan hanya sebagai fasilitator utama dalam aktivitas ekonomi di ekosistem industri halal, tetapi juga sebagai upaya nyata dalam mewujudkan harapan negara.

2. Visi dan Misi Perusahaan

Adapun visi dan misi dari PT. Bank Syariah Indonesia yaitu sebagai berikut.

a) Visi

“Top 10 Global Islamic Bank”, menjadi salah satu dari 10 Bank Syariah terbesar berdasarkan kapitalisasi pasar secara global dalam waktu 5 tahun ke depan.

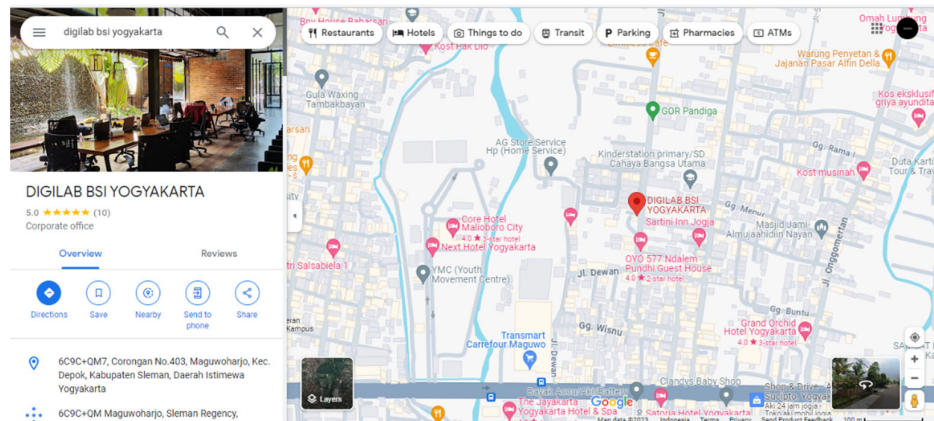
b) Misi

- 1) Memberikan akses solusi keuangan syariah di Indonesia Melayani >20 juta nasabah dan menjadi Top 5 Bank berdasarkan asset (500+T) dan nilai buku 50T di tahun 2025.
- 2) Menjadi bank besar yang memberikan nilai terbaik bagi para pemegang saham Top 5 bank yang paling profitable di Indonesia (ROE 18%) dan valuasi kuat (PB>2).
- 3) Menjadi perusahaan pilihan dan kebanggaan para talenta terbaik Indonesia Perusahaan dengan nilai yang kuat dan memberdayakan masyarakat serta berkomitmen pada pengembangan karyawan dengan budaya berbasis kinerja.

3. Alamat Perusahaan

Kantor IT Digilab PT. Bank Syariah Indonesia beralamat di Jl. KH Ahmad Munajah, Corongan No.403, Maguwoharjo, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Berikut pada Gambar 2.3 merupakan peta lokasi IT Digilab PT. Bank Syariah Indonesia di Yogyakarta, dan untuk melihat lokasi lebih detail dapat diklik pada link dibawah ini.

Peta lokasi: <https://maps.app.goo.gl/XbJy87hbmBv9qRWk6>



Gambar 2.3 Peta Lokasi PT. Bank Syariah Indonesia

4. Kontak Perusahaan

PT. PT. Bank Syariah Indonesia memiliki beberapa kontak yang dapat dihubungi untuk informasi yang lebih detail yaitu sebagai berikut.

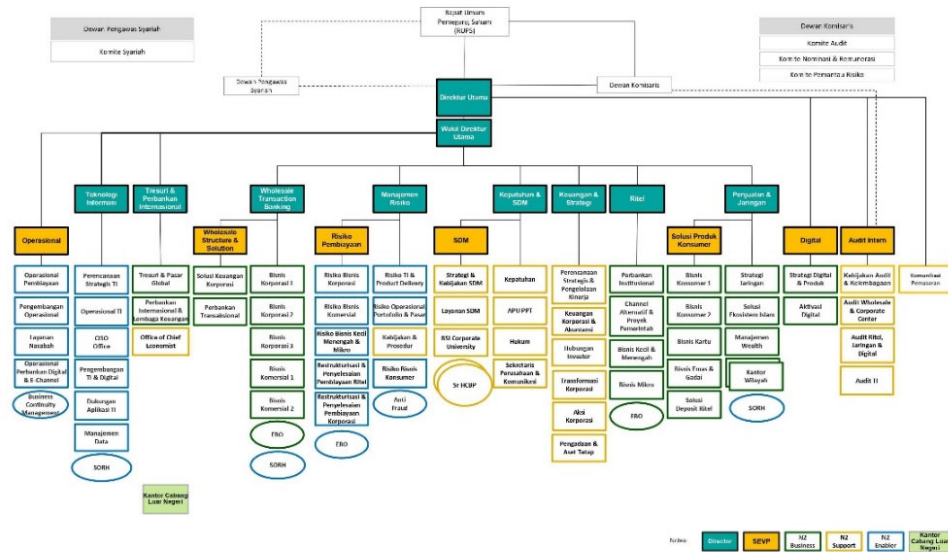
Telepon : Panggilan BSI – 14040

WhatsApp : +62 815 8411 4040

Email : hubungius@bankbsi.co.id

B. Struktur Organisasi Ditempat Magang

Struktur organisasi menunjukkan pola hubungan dari suatu organisasi yang memiliki keterkaitan antara satu dengan yang lainnya. Pada Gambar 2.4 berikut merupakan bagian struktur organisasi pada PT. Bank Syariah Indonesia.



Gambar 2.4 Struktur Organisasi PT. Bank Syariah Indonesia

C. Sumber Daya Manusia dan Sumber Daya Fisik Di Lokasi Magang

1. Sumber Daya Manusia

Sumber daya manusia sebagai salah satu sumber daya organisasi yang memegang peranan penting dalam keberhasilan pelaksanaan tujuan organisasi. Sumber daya manusia menggunakan sumber daya lain yang dimiliki oleh organisasi untuk mencapai tujuan. PT. Bank Syariah Indonesia memiliki keseluruhan staff yang berjumlah 18631 orang. Dari banyaknya staff pegawai

yang ada, mereka memiliki tugas yang berbeda dalam mengelola perusahaan yaitu sebagai berikut.

Tabel 2.1 Sumber Daya Manusia PT. Bank Syariah Indonesia

No.	Nama Bagian	Jumlah
1	Group Head, Regional Head, Project Manager	88 orang
2	Dean Area Manager, Dept Head Manager	529 orang
3	Manager, Team Leader & Branch Manager, Sec, Head	2218 orang
4	Relationship Officer, Branch Manager (Cash Outlet Manager), Officer, ODP	6088 orang
5	Staf/Pelaksana/Pegawai Dasar	9758 Orang

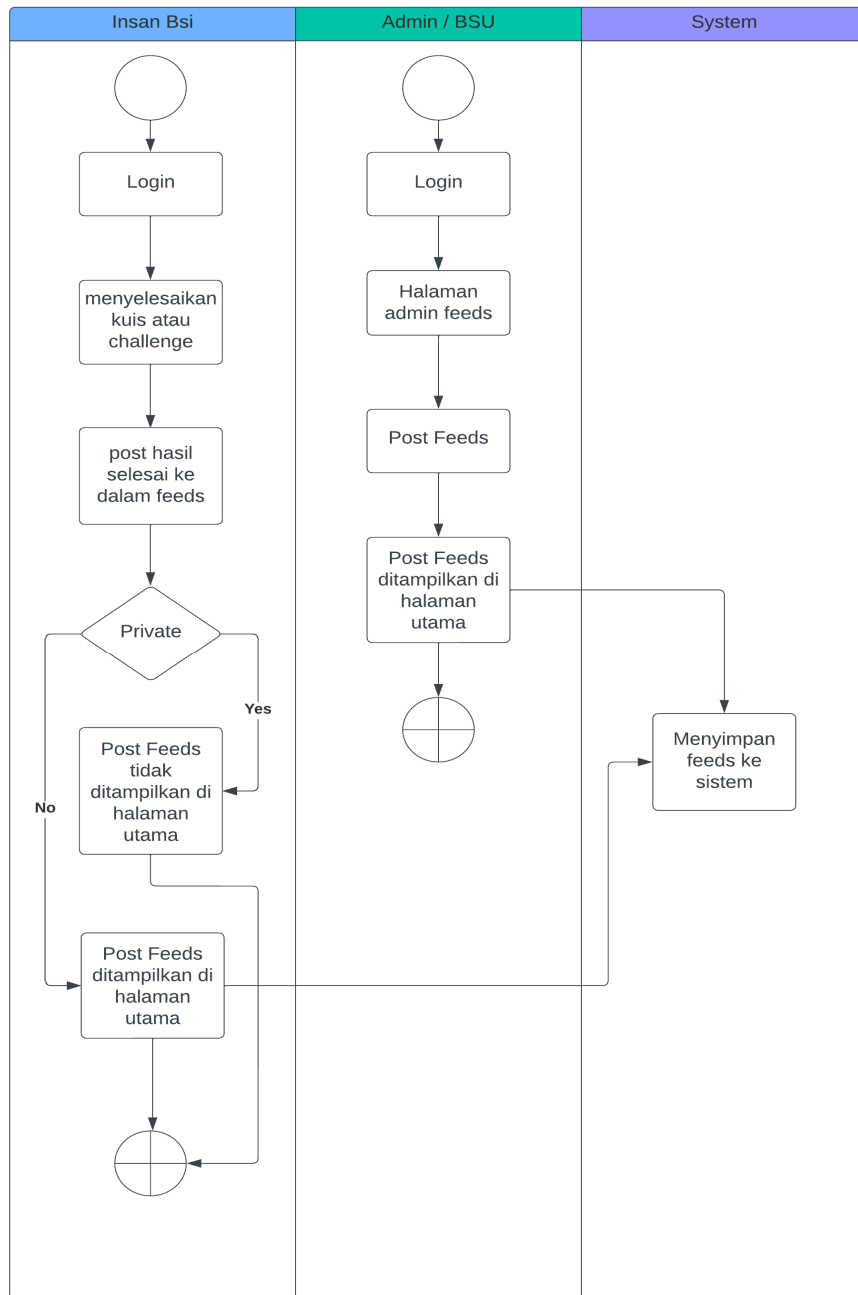
2. Sumber Daya Fisik

Sumber daya fisik merupakan sumber daya yang memiliki wujud nyata dan berguna sebagai sarana dan prasarana yang menunjang kegiatan pada lingkungan tertentu. Salah satu sumber daya fisik yang ada di PT. Bank Syariah Indonesia yaitu fasilitas material yang berada di PT. Bank Syariah Indonesia. Berikut sumber daya fisik sebagai penunjang yang terdapat ada di PT. Bank Syariah Indonesia.

- a) PC (*Personal Computer*)
- b) *Printer*
- c) Gedung
- d) Server
- e) Monitor
- f) LCD Projector
- g) Router Wi-Fi
- h) Ruang Staff
- i) Ruang Rapat
- j) *Musholla*

D. Proses Bisnis Saat Ini yang Berhubungan Dengan Tema Praktik Magang

Berikut proses bisnis yang terjadi di PT. Bank Syariah Indonesia dalam pengembangan aplikasi BSI PINTER dalam fitur feeds dalam fitur ini Insan BSI dapat mengirim feeds hanya dari setelah menyelesaikan challenge atau kuis dan untuk admin/BSU dapat mengirim feeds dengan fitur bagian admin/BSU untuk ditampilkan di halaman feeds dijelaskan pada Gambar 2.5 dijelaskan bahwa terdapat dua cara dalam fungsi fitur feeds dari insan yang dapat melakukan *share* melalui feeds atau pengalaman ketika mengerjakan kuis dan untuk BSI GROUP dapat melakukan untuk menyebar event atau berita ke dalam feeds dan dapat dilihat bagi seluruh user aplikasi BSI PINTER.



Gambar 2. 5 Proses Bisnis Presensi di PT. Bank Syariah Indonesia

BAB III

TAHAPAN KEGIATAN PRAKTIK MAGANG

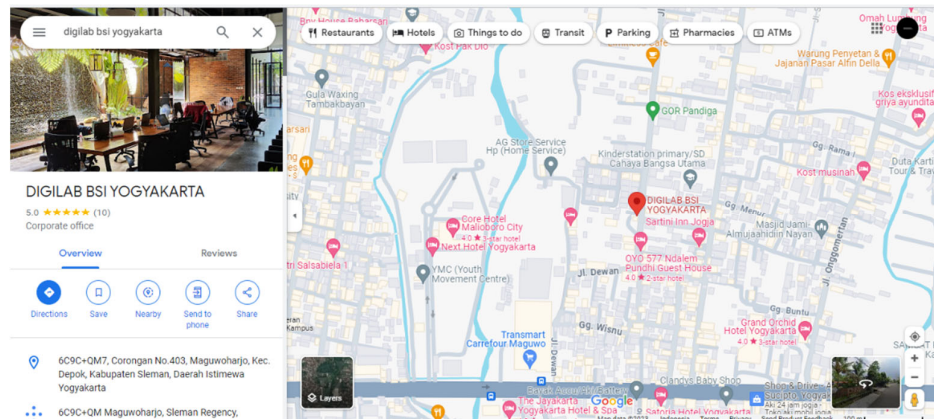
A. Lokasi Praktik Magang, Alamat, Kontak

Kantor IT Digilab PT. Bank Syariah Indonesia beralamat di Jl. KH Ahmad Munajah, Corongan No.403, Maguwoharjo, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Berikut pada Gambar 2.3 merupakan peta lokasi IT Digilab PT. Bank Syariah Indonesia di Yogyakarta, dan untuk melihat lokasi lebih detail dapat diklik pada link dibawah ini.

Kontak Telpnon : 02152997777

Email : idgsecretary1@gmail.com

Peta lokasi: <https://maps.app.goo.gl/XbJy87hbmBv9qRWk6>



Gambar 3.1 Lokasi Praktik Magang

B. Rencana Observasi

1. Perencanaan

a. Nama Instansi

Nama instansi dari tempat dilaksanakannya praktik magang adalah PT. Bank Syariah Indonesia.

b. Proses Observasi

Adapun proses observasi mahasiswa melakukan pendaftaran internship melalui link dan pamflet yang telah disebarluaskan ke seluruh mahasiswa sebelumnya. Setelah itu, dilakukan wawancara dan seleksi internal oleh Prodi Informatika Universitas Ahmad Dahlan pada tanggal 5 Agustus 2023 secara online. Selanjutnya, pada tanggal 23 Agustus 2023, mahasiswa yang telah berhasil lolos wawancara dan seleksi internal UAD akan melanjutkan tahapan berikutnya dengan wawancara online bersama HRD BSI. Taufiq Galang akan bertindak sebagai pewawancara dari BSI dalam proses ini. Pengumuman mahasiswa yang berhasil lolos untuk menjalani praktek magang di IT Digilab di PT Bank Syariah Indonesia dijadwalkan pada tanggal 7 September 2023. Tahap berikutnya melibatkan technical meeting sebagai persiapan praktek magang. Pertemuan ini akan dilaksanakan di IT Digilab BSI di Yogyakarta pada tanggal 15 September 2023. Dalam technical meeting ini, mahasiswa akan mendapatkan informasi lebih lanjut mengenai tugas dan tanggung jawab mereka selama praktek magang, serta persyaratan teknis yang perlu dipenuhi. Dengan demikian, seluruh tahapan proses observasi dari pendaftaran hingga persiapan praktek magang telah dijelaskan secara rinci. Proses ini dirancang untuk memberikan mahasiswa pengalaman magang yang bermutu dan relevan dengan bidang studi mereka di dunia industri.

c. Hasil Observasi

Praktek magang dilaksanakan selama periode 6 bulan, dimulai dari tanggal 18 September 2023 hingga 20 April 2024. Pelaksanaannya dilakukan secara Work From Office (WFO), dengan jadwal kerja selama 5 hari dalam satu minggu, yaitu Senin hingga Jumat. Jam kerja berlangsung dari pukul 08.30 WIB hingga 17.30 WIB. Dengan pengaturan waktu dan tempat yang terstruktur ini, diharapkan mahasiswa magang dapat mengoptimalkan pengalaman mereka selama masa praktek di lingkungan kerja yang sesungguhnya.

C. Rancangan Jadwal Kegiatan Magang

Kegiatan praktik magang ini dilaksanakan selama 6 bulan yang dimulai pada tanggal 18 September 2023 sampai dengan tanggal 20 April 2023. Adapun rincian

jadwal praktik magang yang telah dirancang sebelumnya di PT. Bank Syariah Indonesia diantaranya sebagai berikut pada Tabel 3.1.

Tabel 3.1 Rancangan Jadwal Kegiatan Praktik Magang

No	Nama Kegiatan	September 2023		Oktober 2023				November 2023				Desember 2023				Januari 2024		
		3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
1	pengenalan magang di BSI	■																
2	materi Class Training		■															
3	Pengenalan Bsi System Information dan inisialisai mini project Digibank			■														
4	Mini Project Digibank Start				■													
5	Pelaksanaan sprint 1 mini Project Digibank dari mulai belajar springboot hingga implementasi ke dalam aplikasi dan belajar hal hal dasar di dalam pengembangan aplikasi					■	■											
6	Pelaksanaan Spring 2 mini project Digibank implementasi pengembangan dasar mulai dengan set up environment yang dibuthkan dalam java springboot dan mulai pengembangan berdasarkan api contarct dan ERD							■	■	■								
7	pelaksanaan sprint 3 mini Project Digibank integrasi antara frontend dan backend									■	■							
8	Pelaksanaan Spring 4 mini Project Digibank melakukan fixing bug ketika dilakukan integrasi antara frontend dan backend											■	■					
9	Tahap awal pembahasan real project pembuatan apliaksi BSI PINTER													■				

10	Pelaksanaan sprint 1 Real Project BSI Pinter dimulai dari pembuatan UI/UX pendefinisian ERD dan api contract yang akan dilakukan untuk menunjang kebutuhan aplikasi dan dipandu oleh mentor																																																																																																																																																																																																
11	Pelaksanaan sprint 1 Real Project BSI Pinter melakukan implementasi code ke dalam backend springboot dari api contract dan erd																																																																																																																																																																																																

BAB IV

HASIL PELAKSANAAN PRAKTIK MAGANG

A. Hasil Observasi Magang

Hasil observasi praktik magang yaitu berisi proses bisnis dan rincian pekerjaan pada instansi pelaksanaan magang.

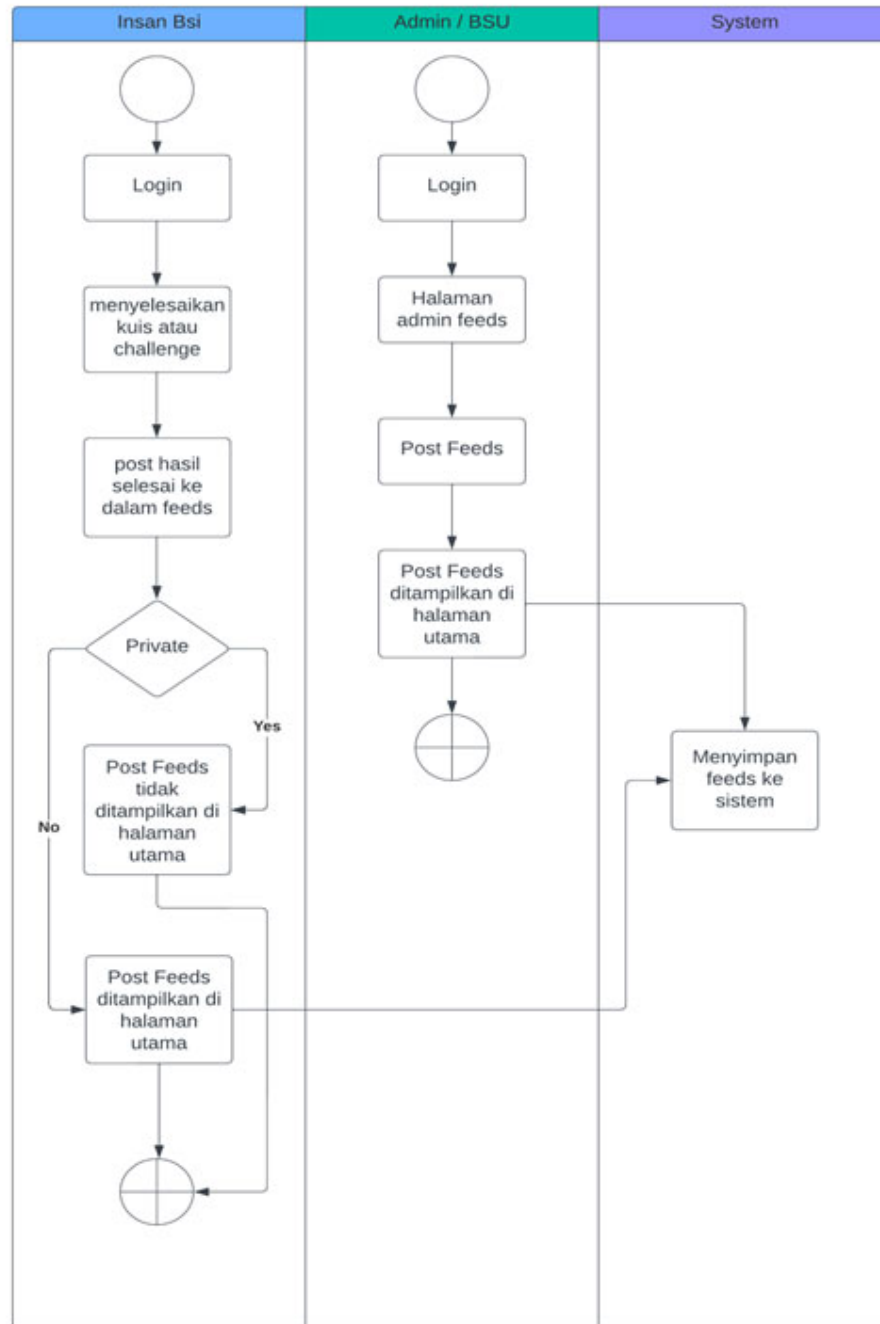
1. Proses Bisnis

Pada proses bisnis aplikasi yang berlangsung di PT. Bank Syariah Indonesia, mahasiswa diberikan tugas untuk melakukan pengembangan aplikasi BSI Pusat Informasi Terpadu (BSI PINTER) permasalahan yang terjadi adalah, insan BSI menghadapi beberapa kendala, seperti tidak mengetahui materi quiz yang dikirim di grup Whatsapp, memaksa pengirim quiz untuk mencari kembali materi tersebut. Selain itu, untuk memahami produk dan jasa Bank BSI, mereka harus membaca dokumen SPB yang berhalaman banyak dan hanya dapat diakses melalui LAN di kantor. Keterlibatan dalam media sosial juga tidak terukur dengan jelas ketika diminta untuk like atau share, mengingat tidak adanya indikator khusus. Sementara itu, absensi manual untuk quiz atau event di grup Whatsapp dan risiko terlewatnya informasi dalam chat grup yang ramai menjadi tantangan tambahan.

Solusi yang diusulkan adalah pengembangan Aplikasi Mobile yang menyediakan akses terpusat dan mudah diakses bagi Insan BSI. Aplikasi ini mencakup informasi penting dari manajemen, artikel product knowledge, quiz, serta tantangan dan hadiah. Fitur kunci meliputi materi product knowledge, sistem event dan seminar terintegrasi, quiz, challenge, dan rewards, informasi seputar Bank BSI, kemampuan untuk request ruang meeting, serta beranda informasi terbaru dari manajemen. Aplikasi ini juga dilengkapi dengan fitur pelaporan presensi untuk quiz dan seminar.

Dengan adanya Aplikasi Mobile ini, Bank BSI dapat meningkatkan efisiensi dan keterlibatan Insan BSI. Mereka dapat dengan mudah mengakses informasi, mengikuti event, dan melihat perkembangan mereka secara real-time. Penggunaan aplikasi juga menghilangkan ketergantungan pada absensi manual dan memastikan bahwa informasi penting tidak terlewatkan dalam

keramaian chat grup. Selain itu, adanya indikator pada media sosial dalam aplikasi memudahkan Insan BSI untuk berpartisipasi dengan jelas dan terukur. Selanjutnya pada Gambar 4.2 berikut juga dijelaskan bagan mengenai proses bisnis pada fitur feeds aplikasi BSI PINTER PT. Bank Syariah Indonesia.



Gambar 4.1 Proses Bisnis *backend* bagian *feeds*

2. Rincian Pekerjaan

Rincian pekerjaan selama praktek magang di PT Bank Syariah Indonesia, yaitu sebagai berikut.

- a. Melakukan analisis aplikasi BSI PINTER
 - 1) Menganalisis fitur-fitur yang akan dikembangkan dalam proses pengembangan aplikasi.
 - 2) Menganalisis kebutuhan aplikasi backend BSI PINTER.
- b. Pengembangan aplikasi *mobile banking* Digi Bank
 - 1) Mempelajari *framework* Java Springboot.
 - a) Mempelajari arsitektur dalam *framework* Java Springboot.
 - b) Mempelajari alur program sistem BSI PINTER.
 - c) Mempelajari deployment sistem backend.
 - d) Mempelajari cara kerja *request* dan *response*
 - 2) Praktek dalam pengembangan *mini project* aplikasi Digi Bank.
 - a) Membaca *requirement gathering* dari *product owner*.
 - b) Merancang ERD
 - c) Merancang API *contract*.
 - d) Merancang *logic* sistem untuk backend dan database.
 - e) Merancang database untuk backend.
 - f) Membuat *unit test* dari setiap fitur.
 - g) *Fixing bug*.
 - 3) Melakukan pengujian aplikasi
Melakukan pengujian untuk memeriksa program apakah dapat di implementasikan ke dalam *frontend* dan dilakukan pengujian melalui postman dan implementasi langsung.
 - 4) Membuat berkas dokumentasi akhir aplikasi BSI PINTER
Tahap terakhir yaitu dilakukan proses penyusunan dokumentasi pengujian akhir aplikasi BSI PINTER.

B. Pembahasan Magang

Adapun penjelasan mengenai praktik magang yang telah dilaksanakan yaitu sebagai berikut.

1. Problem yang ditemukan di tempat magang

Adapun permasalahan yang ditemukan di PT. Bank Syariah Indonesia yang dihadapi oleh insan BSI. Kendala seperti tidak mengetahui materi quiz, kesulitan memahami produk dan jasa Bank BSI, serta keterbatasan keterlibatan dalam media sosial akan diatasi dengan aplikasi ini. Solusi tersebut melibatkan pengembangan aplikasi yang memberikan akses terpusat dan mudah diakses bagi insan BSI. Aplikasi ini mencakup informasi manajemen, artikel product knowledge, quiz, serta tantangan dan hadiah. Fitur kunci termasuk materi product knowledge, sistem event terintegrasi, quiz, challenge, rewards, informasi Bank BSI, permintaan ruang meeting, serta beranda informasi terbaru.

2. Analisis terhadap hasil observasi

Proses pengembangan BSI PINTER dimulai dengan pemilihan bahasa pemrograman dan framework yang akan digunakan. Setelah melakukan observasi, diputuskan untuk mengembangkan aplikasi dengan menggunakan bahasa Java Springboot. Langkah selanjutnya melibatkan penetapan API contract dan pembuatan Entity Relationship Diagram (ERD) sebagai landasan pengembangan. Proyek ini kemudian diatur dengan mengatur set up project default untuk memastikan bahwa kode yang ditulis bersih dan rapi. Pemilihan Java Springboot juga memerlukan pemahaman yang kuat tentang konsep Object Oriented Programming (OOP) untuk memastikan konsistensi dan efisiensi dalam pengembangan. Proses implementasi backend BSI PINTER melibatkan langkah-langkah yang terstruktur, mulai dari pemilihan teknologi hingga penyusunan kontrak API dan ERD. Pilihan Java Springboot diambil untuk memastikan kesederhanaan, kebersihan, dan konsistensi dalam pengembangan aplikasi, sambil mengharuskan pengembang memiliki pemahaman mendalam tentang konsep OOP.

a) Analisis kebutuhan perangkat lunak (*software*)

Agar sistem tersebut dapat dikembangkan dan dapat digunakan sebagaimana mestinya, maka diperlukan beberapa *tools* sebagai penunjang dalam pengembangan sistem aplikasi. Adapun kebutuhan perangkat lunak (*software*) yaitu sebagai berikut.

- 1) Sistem Operasi Windows 11.

- 2) Visual Studio Code
- 3) IntelliJ IDEA Ultimate
- 4) Postman
- 5) Figma
- 6) Ngrok
- 7) PgAdmin 4
- 8) Chrome
- 9) Data Grip

b) Analisis kebutuhan perangkat keras (*hardware*)

Selain *tools* yang dibutuhkan untuk pengembangan sistem, perangkat lunak (*hardware*) juga menjadi salah satu kebutuhan dalam keberhasilan pengembangan sebuah sistem. Adapun kebutuhan perangkat lunak (*hardware*) yang digunakan dalam pengembangan sistem aplikasi *mobile* presensi ini yaitu sebagai berikut.

- 1) Laptop HP Victus
- 2) *Processor* Intel Core I7-12700H
- 3) RAM 16 GB
- 4) Penyimpanan 1 TB

c) Analisis kebutuhan sistem aplikasi

- 1) Login dan Logout aplikasi
- 2) Halaman *Dashboard*
- 3) Fitur *One Time Password (OTP)*
- 4) Fitur Register
- 5) Fitur cek kantor cabang
- 6) Fitur cek Jabatan per kantor cabang
- 7) Fitur cek profile
- 8) Fitur poin
- 9) Fitur hadiah
- 10) Fitur peringkat bulanan
- 11) Fitur peringkat *all time high*
- 12) Fitur pengetahuan
- 13) Fitur kuis

- 14) Fitur artikel
- 15) Fitur challenge
- 16) Fitur progress kuis
- 17) Fitur progress challenge
- 18) Fitur progress artikel
- 19) Fitur progress feeds
- 20) Fitur upload feeds dari challenge
- 21) Fitur upload feeds dari kuis
- 22) Fitur mencari pengetahuan
- 23) Fitur mencari feeds
- 24) Fitur tempat meeting
- 25) Fitur komentar feeds
- 26) Fitur like feeds
- 27) Fitur menambah hadiah admin
- 28) Fitur merubah profile
- 29) Fitur upload feeds admin
- 30) Fitur upload kuis admin
- 31) Fitur upload artikel admin
- 32) Fitur upload challenge admin
- 33) Fitur membuat pengetahuan admin
- 34) Fitur mengatur user dari admin

d) Analisis kebutuhan fungsional

Key Features

1. Materi Product Knowledge
2. Quiz, Challenge, Rewards seputar Bank BSI
3. Feed Informasi BSI Pinter
4. Menu E-Request
5. Request Ruang Meeting
6. Request KDO (Fasilitas Akomodasi Pegawai)

User

1. Admin BSU

2. DH Insan BSI
3. Insan BSI

Insan BSI:

1. Registrasi Akun BSI Pinter
2. Login Akun BSI Pinter
3. Melihat halaman home yang berisi container Product Knowledge BSI Pinter (Foto, Judul, Deskripsi)
4. Melihat "Artikel" Product Knowledge Bank BSI
5. Mengerjakan "Quiz" seputar Product Knowledge
6. Mengupload bukti aktivitas "Challenge" (Upload bukti share socmed Bank BSI)
7. Menerima poin dari setiap membaca artikel, quiz, dan challenge
8. Melihat halaman hadiah
9. Menukarkan poin dengan hadiah
10. Melihat history hadiah yang sudah di claim
11. Quiz dan Challenge yang sudah dikerjakan dapat di post di halaman feed jika diinginkan untuk mendapatkan like dari user, dan setiap mendapatkan like akan mendapatkan poin
12. Membuka profil user insan Bank BSI untuk melihat foto, nama, jabatan, kantor cabang / Grup Kantor Pusat (Bisa dijadikan referensi untuk input Registrasi Akun)

Admin BSU:

1. Membuat Container Product Knowledge untuk diisi dengan Artikel Product Knowledge, Quiz, dan Challenge*
2. Membuat Materi Artikel Product Knowledge*
3. Menyusun Quiz sesuai dengan Materi Artikel Product Knowledge *
4. Menyusun Challenge sesuai dengan Materi Artikel Product Knowledge*
5. Menyusun poin pada setiap aktivitas yang dilakukan oleh user BSI Pinter setelah melakukan aktivitas seperti menyelesaikan

membaca artikel, menonton video, menyelesaikan quiz dan challenge.

6. Menyusun hadiah terbatas yang dapat ditukar dengan poin dan menentukan jumlah poin yang dapat ditukar dan menentukan jumlah limit penukaran hadiah*
7. Mendapatkan notifikasi pengiriman hadiah dan mengirim hadiah ke masing masing user (Dapat melihat alamat user dari departemen atau kantor cabang user, untuk menghubunginya bisa melalui teams akan ada notifikasi "Admin akan menghubungi lewat MS Teams")
8. Mengkonfirmasi pengiriman hadiah
9. (REQUEST) Fitur tambah user insan BSI dan DH insan BSI*
 - a. *Dapat melakukan Create, Read, Update, Delete

DH Insan BSI:

1. Dapat melakukan semua aktivitas yang dilakukan oleh Insan BSI
 2. Melihat reporting dari anggotanya yang sudah melakukan aktivitas pada aplikasi (Quiz, Challenge, Membaca Materi dalam bentuk artikel atau video)
 3. Mengirim reminder kepada anggota timnya melalui aplikasi BSI Pinter yang belum mengerjakan Quiz, Challenge, Membaca Materi Product Knowledge
- e) Analisis kebutuhan non-fungsional
- 1) Sistem memiliki tampilan yang mudah dipahami.
 - 2) Sistem dapat dioperasikan dan menampilkan data yang tepat.
 - 3) Sistem dapat dioperasikan sesuai fungsinya dengan baik termasuk fitur-fitur yang diinginkan.
 - 4) Sistem dapat diakses dengan perangkat *smartphone* apapun baik android maupun iOS.

3. Capaian Magang

Setelah melaksanakan praktek magang selama 6 bulan pada tanggal 18 September 2023 sampai dengan 20 April 2024 sebagai Backend Developer

dalam tim developer di Unit Kerja Digital Laboratorium (IT DIGILAB) PT Bank Syariah Indonesia, Tbk. Adapun capaian magang yang diperoleh sebagai berikut.

a) Pengembangan Sistem Aplikasi *Backend aplikasi backend* BSI PINTER menggunakan Java Springboot

Adapun beberapa hal yang dibutuhkan dalam pengembangan Sistem Aplikasi *Mobile* Presensi Menggunakan React Native yaitu sebagai berikut.

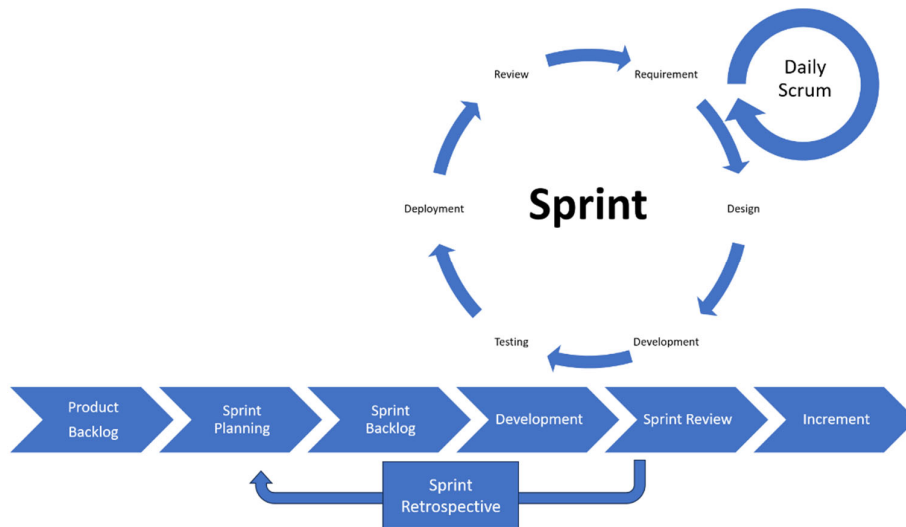
1) Metode Perancangan Aplikasi

Pengembangan sistem Aplikasi BSI PINTER pada bagian backend dilakukan menggunakan Java Springboot dengan menerapkan Metode Agile, sebagaimana tergambar dalam Gambar 4.2. Pemilihan Metode Agile dipandang sebagai pilihan yang sangat fleksibel untuk pengembangan yang dinamis. Proses pengembangan dimulai dengan tahap pertama, yaitu requirements definition. Pada tahap ini, kebutuhan-kebutuhan kunci, seperti perencanaan *Entity Relationship Diagram* (ERD), kontrak API, autentikasi pengguna, otorisasi, dan token authentication, ditentukan secara jelas. Requirements definition menjadi landasan bagi tahap berikutnya, yaitu system and software design.

Tahap system and software design, kebutuhan-kebutuhan yang telah didefinisikan digunakan sebagai dasar untuk merancang sistem dan perangkat lunak. Proses implementasi kemudian dilakukan dengan menggunakan bahasa pemrograman Java Script dan framework Java Springboot. Setelah tahap coding selesai, aplikasi yang telah dibuat akan menjalani proses unit testing menggunakan JUnit untuk memastikan kualitas dan keberlanjutan fungsi-fungsi yang diimplementasikan.

Langkah selanjutnya adalah integrasi dengan frontend, di mana bagian backend akan disatukan dengan bagian frontend untuk membentuk aplikasi secara menyeluruh. Setelah tahap integrasi, aplikasi akan menjalani pengujian lebih lanjut oleh Software Quality Assurance (SQA) untuk memastikan kualitas keseluruhan sistem. Pada tahap terakhir, dilakukan maintenance sistem untuk

menanggapi perubahan-perubahan, peningkatan, dan pemeliharaan yang mungkin diperlukan selama siklus hidup aplikasi. Dengan menggunakan Metode Agile, pengembangan Aplikasi BSI PINTER dapat dilakukan secara iteratif dan responsif terhadap perubahan kebutuhan dan perbaikan yang diperlukan.



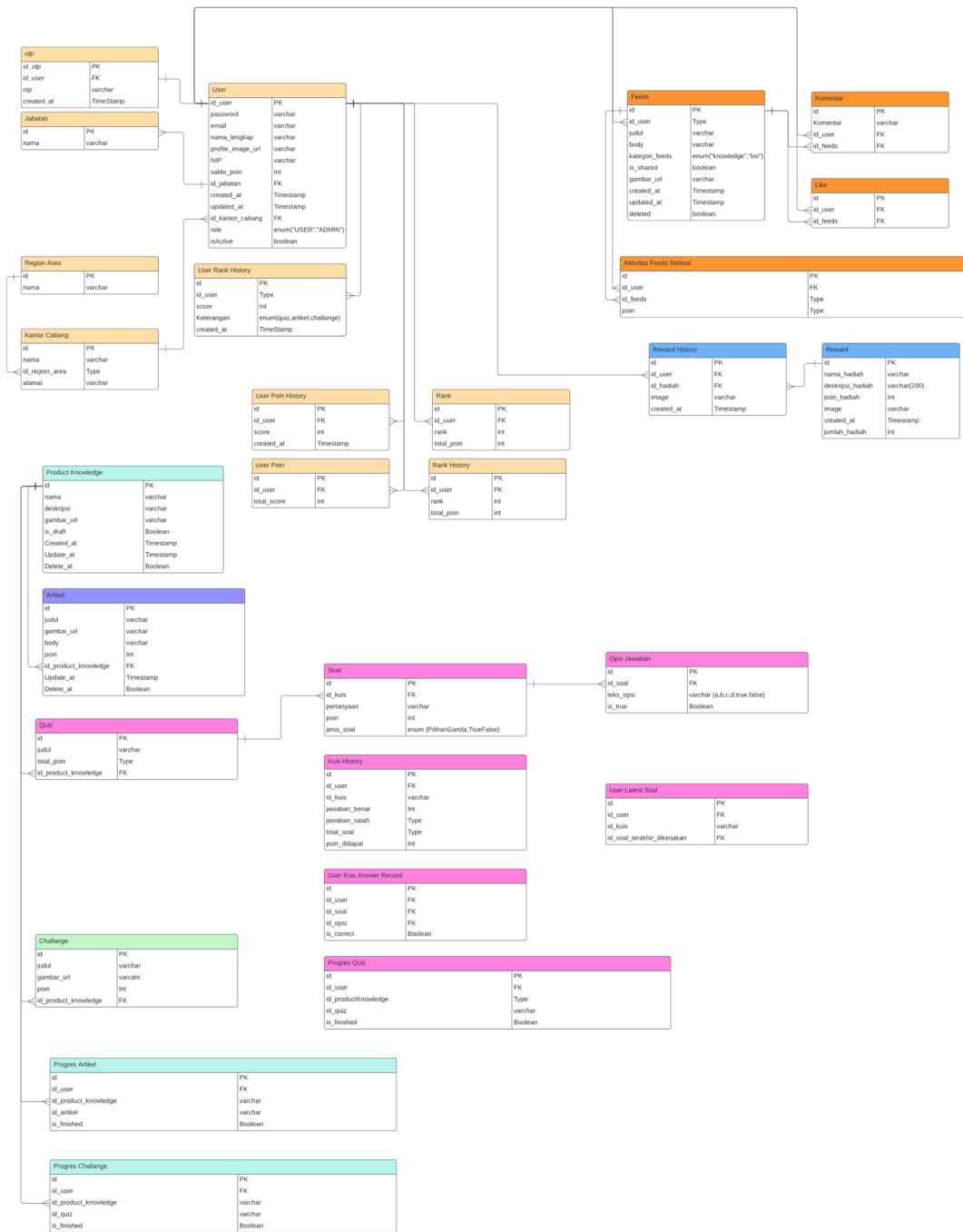
Gambar 4.2 Tahapan Metode Agile

2) Analisis Kebutuhan Sistem

Aplikasi BSI PINTER

a) *Entity Relationship Diagram*

Entity Relationship Diagram (ERD) adalah representasi grafis dari struktur data dan hubungan antar entitas dalam suatu sistem atau aplikasi. Berikut pada Gambar 4.4 merupakan ERD secara lengkap pada aplikasi *backend* BSI PINTER.

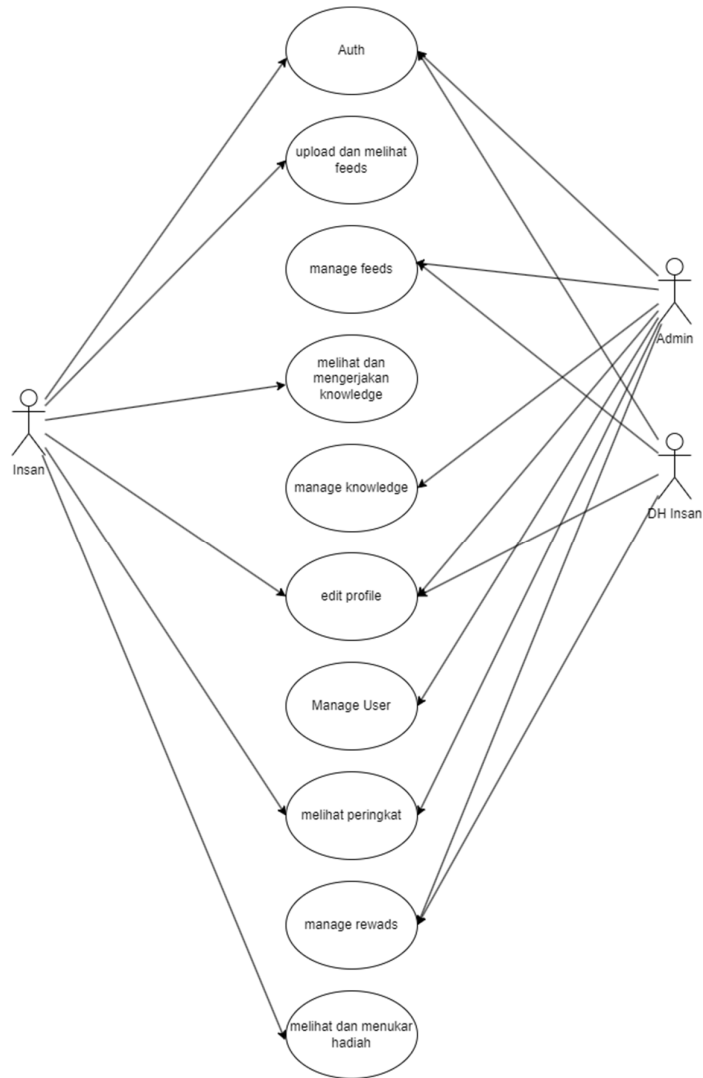


Gambar 4.3 Entity Relationship Diagram (ERD)

b) Use Case Diagram

Dalam use case diagram yang telah disusun, terdapat tiga aktor utama, yaitu Admin, Insan, dan DH Insan. Admin, sebagai otoritas tertinggi, memiliki wewenang untuk mengatur seluruh

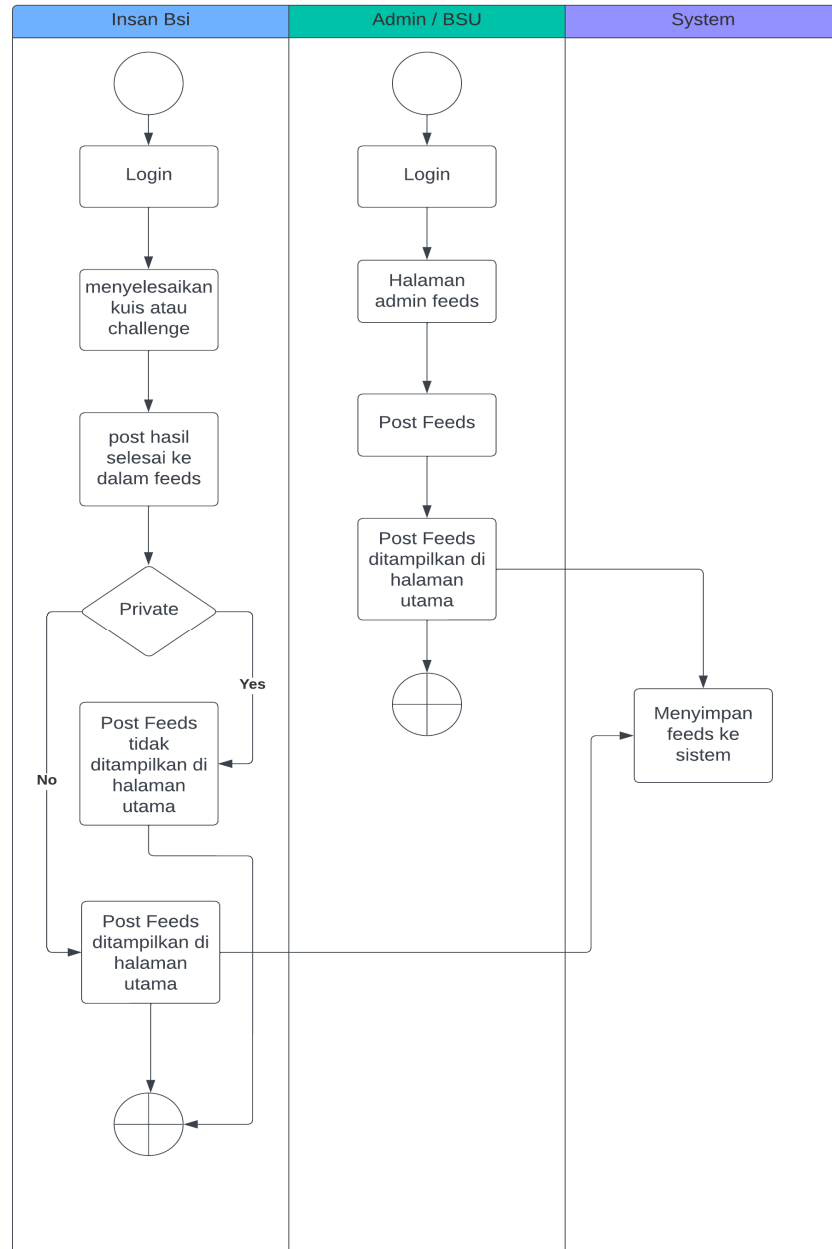
aspek yang ada di dalam aplikasi. Admin dapat melakukan pengelolaan dan kontrol penuh terhadap fitur dan fungsionalitas aplikasi. Selanjutnya, terdapat DH Insan yang merupakan aktor yang memiliki peran sebagai pengurus bagi banyak Insan. DH Insan memiliki tanggung jawab untuk mengelola dan memfasilitasi kebutuhan dari sejumlah pengguna biasa yang disebut sebagai Insan. Sebagai penghubung antara Admin dan Insan, DH Insan berperan dalam memastikan kelancaran dan efisiensi interaksi antara kedua kelompok tersebut dalam penggunaan aplikasi.



Gambar 4.4 Use Case Diagram

c) *Activity Diagram*

Diagram Aktivitas (*Activity Diagram*) dalam pengembangan aplikasi backend BSI PINTER fitur feed, Activity diagram digram dijelaskan pada Gambar 4.6 yang menunjukkan cara kerja fitur feeds dan urutanya Berikut *activity diagram* dapat dilihat pada Gambar 4.6.



Gambar 4.5 *Activity Diagram*

3) Implementasi *Coding* dan Hasil Implementasi

a) Feeds Repository

Class *FeedsRepository* pada Gambar 4.6 adalah sebuah antarmuka (interface) Spring Data JPA yang mengelola akses dan manipulasi data terkait entitas Feeds dalam aplikasi. Kelas ini menggabungkan dua antarmuka, yaitu *JpaRepository* dan *JpaSpecificationExecutor*, untuk memberikan fungsionalitas dasar dan eksekusi spesifikasi JPA. Metode-metode yang didefinisikan dalam antarmuka ini mencakup berbagai query yang memungkinkan pengaksesan data Feeds sesuai dengan berbagai kriteria, seperti pencarian berdasarkan judul atau isi Feeds, kategori Feeds, status berbagi, dan kombinasi dari kriteria-kriteria tersebut. Penggunaan anotasi `@Query` memungkinkan penentuan *query* kustom dengan menggunakan sintaks HQL (*Hibernate Query Language*). Secara keseluruhan, *FeedsRepository* memberikan fleksibilitas dalam mengakses dan memanipulasi data Feeds sesuai dengan kebutuhan aplikasi, mengoptimalkan kinerja melalui paginasi, dan memenuhi berbagai kasus penggunaan dalam pengelolaan informasi di dalam aplikasi BSI PINTER.

b) Feeds Service Interface

Interface *FeedService* pada Gambar 4.7 menyediakan sejumlah metode yang mewakili fungsionalitas utama terkait manajemen dan interaksi dengan data Feeds dalam aplikasi BSI PINTER. Metode-metode tersebut mencakup pengambilan Feeds berdasarkan berbagai kriteria, seperti pengguna, kelompok BSI, dan pencarian teks tertentu. Selain itu, antarmuka ini menangani operasi-operasi seperti memberikan like atau unlike pada suatu Feeds, mendapatkan detail informasi dari suatu Feeds, memberikan komentar pada Feeds, mengambil daftar komentar, menghapus Feeds, serta melakukan berbagai tindakan terkait pembuatan dan

pengelolaan Feeds, termasuk mengedit Feeds, membuat Feeds terkait tantangan (*challenge*), dan mendapatkan aktivitas Feeds pengguna. Dengan menyediakan antarmuka ini, FeedService membantu memfasilitasi akses dan manipulasi data Feeds secara efisien dan responsif dalam konteks aplikasi BSI PINTER.

c) Feeds Service Implementation

Class `FeedServiceImpl` pada Gambar 4.8 adalah implementasi dari antarmuka `FeedService` yang menyediakan berbagai metode untuk mengelola dan mengakses data Feeds dalam aplikasi BSI PINTER. Kelas ini memanfaatkan repositori-repositori, seperti `FeedsRepository`, `LikeFeedsRepository`, dan lainnya, untuk melakukan operasi-operasi seperti pengambilan Feeds, pencarian Feeds, dan menangani proses like pada Feeds. Metode `getFeeds` dan `getFeedsSearch` mengimplementasikan logika untuk mengambil data Feeds sesuai dengan peran pengguna dan parameter pencarian tertentu. Proses pengambilan data Feeds melibatkan mapper-mapper seperti `FeedsMapper`, `KomentarMapper`, dan `ActivityFeedsMapper` untuk mengubah data entitas menjadi DTO (Data Transfer Object) yang sesuai. Selain itu, terdapat metode `likeFeeds` yang memproses permintaan like pada suatu Feeds, dengan memastikan bahwa pengguna belum melakukan like sebelumnya. Keseluruhan implementasi ini membantu dalam menyediakan fungsionalitas lengkap dan responsif terkait fitur Feeds dalam aplikasi BSI PINTER, dan dapat disesuaikan dengan berbagai kebutuhan pengguna.

d) Feeds Mapper

Antarmuka pemetaan pada Gambar 4.9 merupakan (mapper) `FeedsMapper` menggunakan framework `MapStruct` dengan konfigurasi Spring untuk mengonversi entitas Feeds menjadi objek transfer data (DTO) dalam bentuk `ListFeedsDto`. Metode `feedsToFeedsResponses` yang didefinisikan dalam antarmuka ini menerima daftar entitas Feeds sebagai input dan

menghasilkan daftar DTO ListFeedsDto. Pemetaan ini memungkinkan transformasi data yang efisien antara struktur internal entitas dan representasi eksternal yang diperlukan oleh lapisan pengguna atau presentasi. Dengan menggabungkan anotasi @Mapper dan konfigurasi componentModel = "spring", MapStruct dapat secara otomatis menghasilkan implementasi pemetaan yang dapat diinjeksikan dan digunakan dalam konteks aplikasi Spring, menyederhanakan proses pengelolaan pemetaan antar kelas entitas dan DTO.

e) Feeds Controller

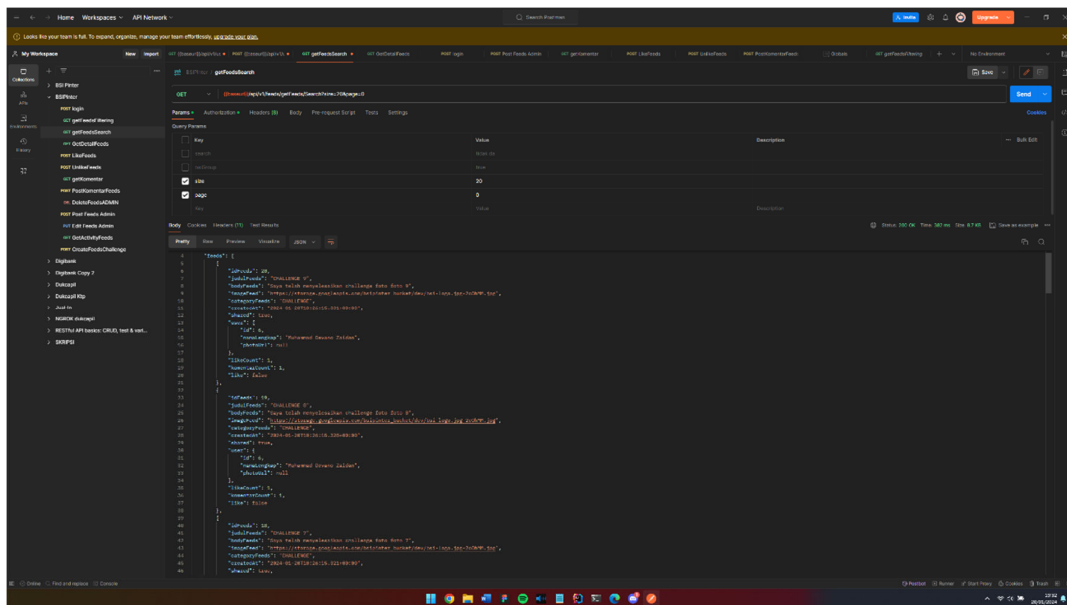
Class FeedsController pada Gambar 4.10 dalam class ini bertindak sebagai kontroler dalam aplikasi BSI PINTER yang menangani berbagai permintaan terkait operasi Feeds. Kontroler ini menggunakan anotasi @RestController untuk menandakan bahwa setiap metode dalam kelas ini mengembalikan data langsung sebagai respon HTTP. Melalui metode-metodenya, FeedsController memfasilitasi akses ke berbagai layanan yang disediakan oleh FeedService untuk pengguna aplikasi. Metode getFeedsDetail dan getFeedsSearch digunakan untuk mengambil data Feeds berdasarkan berbagai kriteria seperti halaman, ukuran, pencarian, dan kelompok BSI. Metode getDetailFeeds mengambil informasi rinci tentang suatu Feeds berdasarkan idFeeds tertentu. Metode likeFeeds dan unlikeFeeds digunakan untuk memberikan dan mencabut like pada suatu Feeds. Selain itu, terdapat metode-metode lainnya yang menangani operasi terkait komentar, pembuatan Feeds terkait tantangan, dan pengambilan aktivitas Feeds pengguna. Kontroler ini juga menggunakan anotasi seperti @RequestMapping, @GetMapping, dan @PostMapping untuk menetapkan URL endpoint yang dapat diakses oleh klien, serta melakukan manajemen otentikasi menggunakan token JWT melalui kelas JwtService.

f) Feeds Dto

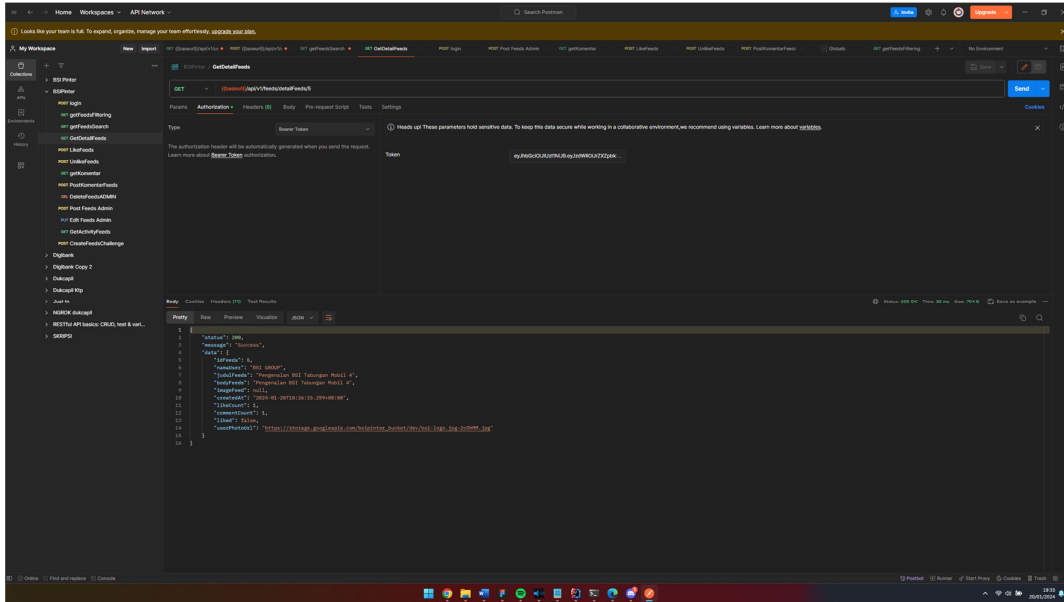
Paket `com.bsi.bsipinter.dto.Feeds.listFeeds` berisi dua kelas DTO yang digunakan untuk merepresentasikan data Feeds dalam aplikasi BSI PINTER. Pertama, kelas `FeedsDto` memiliki atribut `status`, `message`, dan `feeds` yang merupakan daftar dari DTO `ListFeedsDto`. Kelas ini digunakan untuk memberikan respon global terhadap operasi-operasi yang melibatkan Feeds, seperti pengambilan dan pencarian Feeds. Kedua, kelas `ListFeedsDto` mencakup atribut-atribut yang merepresentasikan informasi terkait suatu Feeds, seperti `idFeeds`, `judulFeeds`, `bodyFeeds`, `imageFeed`, `categoryFeeds`, `createdAt`, `shared`, `user`, `likeCount`, `komentarCount`, dan `isLike`. Kedua kelas DTO ini menggunakan anotasi dari framework Lombok seperti `@Getter`, `@Setter`, `@AllArgsConstructor`, `@NoArgsConstructor`, dan `@Builder` untuk secara otomatis menghasilkan metode getter, setter, konstruktor, dan builder, menyederhanakan proses pembuatan dan pengelolaan objek DTO. Dengan struktur ini, DTO-dto ini memfasilitasi pengiriman data terstruktur antara backend dan frontend dalam aplikasi BSI PINTER.

4) *Testing* atau Pengujian Aplikasi

Setelah sistem aplikasi backend BSI PINTER berhasil dibangun, selanjutnya dilakukan pengujian sistem dengan menggunakan postman untuk menguji API apakah sudah sesuai dengan *contract*, pada Gambar 4.12 pengujian fitur feeds dengan menggunakan *filtering* dan *pagination* dengan memasukkan *query param* untuk melakukan *request* kepada server, Gambar 4.13 untuk melihat detail menggunakan id dari feeds dalam detail feeds terdapat data yang lengkap termasuk jumlah like, jumlah komentar dan kondisi *boolean* user apakah sudah pernah menyukai postingan dari feeds,



Gambar 4.6 *Postman List Feeds*



Gambar 4.7 Postman Detail Feeds

4. Keberlanjutan

Kegiatan praktik magang di PT. Bank Syariah Indonesia ini masih berlanjut hingga April yang untuk melanjutkan proyek BSI PINTAR bagian backend dengan mengembangkan fitur yang sama atau membuat fitur lain yang memungkinkan untuk pengembangan lebih lanjut.

BAB V

PENUTUP

A. Kesimpulan

Setelah dilakukannya perancangan aplikasi mobile banking Digi Bank di BSI IT Digilab Yogyakarta, maka dapat disimpulkan sebagai berikut

1. Pembuatan sistem aplikasi *mini project mobile banking Digi Bank* sebagai latihan dan pembiasaan untuk pengembangan aplikasi *mendapatkan applause dari para mentor Developer dari BSI Pusat*.
2. Perancangan aplikasi BSI PINTER memiliki fokus utama untuk membantu *Insan BSI* dan memudahkan pihak BSI untuk mengirim berita dan kegiatan dan pengetahuan yang ada di dalam BSI.
3. Perancangan BSI PINTER masih dalam tahap *development* hingga waktu yang belum ditentukan.

B. Saran

Pengembangan aplikasi backend BSI PINTER telah dibangun dengan sesuai *user flow* yang jelas namun masih terdapat beberapa bug yang terjadi dalam proses pengembangan, untuk kedepannya mungkin perlunya melakukan dokumentasi secara menyeluruh di awal ketika proses development dan memberikan catatan apabila sebuah *api contract* akan berubah dikarenakan sering terjadi miss komunikasi antara frontend dan backend.

REFERENSI

1. https://ir.bankbsi.co.id/corporate_history.html
2. https://ir.bankbsi.co.id/annual_reports.html
3. <https://bisnis.tempo.co/read/1724572/profil-bsi-sejarah-merger-dan-cita-cita-jadi-bank-syariah-terbesar>

LAMPIRAN

A. Surat Pemberitahuan Penerimaan Program Magang IT Digital Kampus Merdeka di PT Bank Syariah Indonesia, Tbk.



5 September 2023
No. 03/055-3/IDG
Lampiran : 1(satu) Lembar

PT Bank Syariah Indonesia Tbk
Kantor Pusat
Gedung The Tower
Jl. Gatot Subroto No. 27
Jakarta Selatan 12930, Indonesia
T : +62 21-30405999
F : +62 21-30421888
www.bankbsi.co.id

Kepada
Fakultas Teknologi Industri
Universitas Ahmad Dahlan
Gedung Kampus 4 UAD
Jalan Ahmad Yani, Kel. Tamanan, Kec. Banguntapan
Bantul, D.I.Yogyakarta

U.p. Yth. Sunardi, S.T., M.T., Ph.D., *Dekan*

Perihal: **PEMBERITAHUAN MAHASISWA MAGANG**

Assalaamu'alaikum Warahmatullahi Wabarakaatuh,

Semoga Saudara beserta seluruh jajaran senantiasa dalam keadaan sehat wal'afiat dan mendapat taufik serta hidayah dari Allah SWT.

Menunjuk perihal tersebut di atas, dengan ini kami sampaikan hal-hal sebagai berikut:

1. Kami mengucapkan terimakasih atas Kerjasama yang telah terjalin dengan baik antara PT Bank Syariah Indonesia, Tbk dengan Fakultas Teknologi Industri Universitas Ahmad Dahlan (UAD).
2. Telah dilakukan penandatanganan Perjanjian Kerjasama **No.F2.1/405/H/VII/2023** tentang **Program Magang IT DIGITAL Kampus Merdeka di PT Bank Syariah Indonesia, Tbk** tanggal **4 Juli 2023**.
3. Sehubungan dengan informasi poin 2, telah dilakukan seleksi mahasiswa Program Magang IT Digital Kampus Merdeka B2B Angkatan I (pertama), dari seleksi yang telah dilakukan terpilih 15 orang mahasiswa yang berkesempatan untuk melakukan Program Magang IT Digital Kampus Merdeka di PT Bank Syariah Indonesia, Tbk di Digilab BSI Yogyakarta. Terlampir pada surat ini data 15 orang mahasiswa yang lolos seleksi.
4. Program Magang IT Digital Bank Syariah Indonesia, Tbk akan dimulai tanggal 15 September 2023 bertempat di Digilab BSI Yogyakarta.
5. Untuk koordinasi mengenai informasi lebih lanjut dapat menghubungi PIC berikut Sdr. Taufiq Galang (083867526967)

Demikian informasi ini kami sampaikan. Mohon dapat ditindaklanjuti pada kesempatan pertama. Atas perhatiannya kami ucapkan terimakasih.

Wassalaamu'alaikum Warahmatullahi Wabarakaatuh

SIGNED / Signed
PT BANK SYARIAH INDONESIA Tbk
IT DIGITAL & DEVELOPMENT GROUP

Arief Sunandar
Group Head

Lampiran 1. Daftar Nama Peserta *Eligible* Program Magang IT Digital BSI asal Universitas Ahmad Dahlan

No	Nama Lengkap	Universitas	Fakultas	Jurusan
1	Muhammad Raihandaffa Dzikrianasa	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
2	AINUNNA'IMAH	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
3	Baharudin Nur Hidayat	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
4	MUHAMMAD RIFQI ZAKARIA	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
5	Hikmatuz Zahra	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
6	Azizah Fathimattuzahro	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
7	Muhammad Ilham Maulana	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
8	Dinda Aulia Rizki	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
9	Muhamad Fadhlurrohman Hakim	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
10	Tatik Nafiaturohmah	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
11	Fahrizal Shofyan Aziz	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
12	Alifa Risiko Amalia	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
13	Muhammad Rizki Alamsyah	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
14	Herfansya Adiyoco Atmaja	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika
15	Muhammad Izza Nur Hakima	Universitas Ahmad Dahlan (UAD)	Fakultas Teknologi Industri (FTI)	S1 Teknik Informatika

FORM PENILAIAN PEMBIMBING LAPANGAN

Yang bertanda tangan di bawah ini :

Nama : Taufiq Galang Adi Putranto, S.Si
 NIK / NIP : 2193015966
 Jabatan : IT Development

Sebagai Pembimbing Lapangan Praktik Magang :
 Nama : Baharudin Nur Hidayat
 NIM : 2000018263

Setelah mengikuti pelaksanaan Praktik Magang mahasiswa tersebut, memberikan nilai sebagai berikut :

ASPEK PENILAIAN	Paramater Penilaian					Skor
	Sangat Baik 5	Baik 4	Cukup 3	Kurang 2	Buruk 1	
1. Komunikasi dan Pemahaman Instruksi Kerja		✓				4
2. Keaktifan dan Respon di Tempat Kerja		✓				4
3. Kesesuaian Materi KP dengan Keilmuan Mahasiswa	✓					5
4. Disiplin Kerja		✓				4
5. Pengerjaan Tugas dan Ketrampilan Teknis	✓					5
Total						22

* Pengesahan lapangan harus diandatangani dan dibubuhi cap



 20.
 (..)Taufiq Galang Adi P.
 NIK / NIP

B. Penilaian praktik magang sementara

C. Logbook



**LOG BOOK PRAKTIK MAGANG MAHASISWA
PROGRAM STUDI S1 INFORMATIKA T.A 2023 / 2024
(WAJIB DIISI DAN MASUK DALAM PENILAIAN)**




Nim : 2000018263
 Nama Mahasiswa : Baharudin Nur Hidayat
 Judul Praktik Magang : BSI IT Digilab Internship Programme Cycle 1
 Dosen Pembimbing : Ir. Sri Winiarti, S.T., M.Cs.
 Pembimbing Lapangan : Taufiq Galang Adi Putranto, S. Si.



Petunjuk Pengisian Log Book



1. Log book di isi per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh pembimbing lapangan/ dosen pembimbing Praktik Magang
4. Log book per minggu di paraf oleh dosen pengampu kelas Praktik Magang
5. Jumlah bimbingan minimal 4x



Logbook Minggu 4 sd 7 (sebelum UTS)



No	Kegiatan dan Lokasi Praktik Magang	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf Pembimbing Lapangan	Paraf Dosen Pembimbing Praktik Magang
		Hari/TGL	Jam Durasi				
1.	BSI UMKM Center Yogyakarta : Inclass Tranining	Senin - Selasa, 18 - 19 September 2023	08.30 – 17.00 WIB	Inclass Training memaparkan materi : - Culture & Corporate Overview, menjelaskan Visi dan Misi BSI - Industrial Overview, menjelaskan perbedaan Bank Syariah dan Bank Konvensional - Fundamental Akad Perbankan Syariah, - Branch Banking Operation, menjelaskan Tugas dan Wewenang Kantor Cabang BSI - Fundamental Financing, Menjelaskan konsep bisnis yang	Tidak ada		





	BSI KCP Kolonel Sugiyono : Observasi penggunaan IT dalam perbankan.	Rabu – Jumat, 20 – 22 September 2023		<p>adil, seimbang, dan manfaatnya.</p> <ul style="list-style-type: none"> - Fundamental Funding, menjelaskan bisnis bank yaitu investor akan menitipkan dana ke bank dengan menghimpun berupa deposito, tabungan, dan giro. <p>Hasil Observasi Lapangan :</p> <ul style="list-style-type: none"> - memahami proses bisnis end to end bagian Frontliner baik melalui Teller maupun Customer Service. - memahami proses bisnis end to end bagian Back Office dan aplikasi menunjang proses tersebut. - memahami proses bisnis end to end bagian Operasional Pencarian Pembiayaan dan aplikasi menunjang proses tersebut. <p>Hasil Observasi Lapangan :</p> <ul style="list-style-type: none"> - memahami proses bisnis end to end bagian Frontliner baik melalui Teller maupun Customer Service. - memahami proses bisnis end to end bagian Back Office dan aplikasi menunjang proses tersebut. - memahami proses bisnis end to end bagian Operasional Pencarian Pembiayaan dan aplikasi menunjang proses tersebut. 			
2	Digilib BSI Yogyakarta : System Banking	Senin – Jumat, 25 –	08.30 – 17.00 WIB	- Pemberian materi yang dijalankan dalam perbankan termasuk	Tidak ada		





	Operation	29 September 2023		<p>membahas tentang digital banking dan alur kerja mobile banking milik BSI.</p> <ul style="list-style-type: none"> - melakukan diskusi kelompok untuk menyelesaikan masalah yang dihadapi sebelumnya dalam observasi yang sudah dilakukan, dengan menentukan awal masalah kemudian mencari solusi untuk permasalahan tersebut yang sudah ada. - Mendapatkan pembelajaran tentang konsep metode SCRUM dalam project IT dan peran SCRUM dalam SDLC. - Mendapatkan pembelajaran tentang konsep SDLC Agile, dan menjelaskan cara kerja setiap posisi dalam jalanya sebuah metode agile. - melakukan diskusi kelompok untuk melakukan testing pada aplikasi dimulai dari pembuatan test script hingga implementasi dalam testing. - Mendapatkan materi tentang peranan seorang project manager serta mendapatkan Pelajaran tentang IT Strategis planning dan peran IT straregic planning dalam 			
--	-----------	-------------------	--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	---------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------





3	BSI UMKM Center Yogyakarta : IT dalam BSI	Senin – Jumat, 2 – 6 Oktober 2023	08.30 – 17.30 WIB	perkembangan Perusahaan. - Mendapatkan pembelajaran materi dari RedHat yang merupakan perusahaan perangkat lunak sumber terbuka yang menghasilkan produk dan layanan untuk komputasi cloud, manajemen konten, dan pengembangan aplikasi. Produk-produk Red Hat meliputi: Red Hat Enterprise Linux, sistem operasi Linux yang populer Red Hat OpenShift, platform kontainerisasi Red Hat Ansible Automation Platform, platform otomasi. Quarkus adalah framework Java yang dirancang untuk digunakan dengan kontainer. Quarkus memiliki waktu startup yang cepat dan penggunaan memori yang rendah, sehingga cocok untuk aplikasi kontainer. Quarkus dapat digunakan untuk berbagai jenis aplikasi, termasuk aplikasi web, aplikasi seluler, dan aplikasi backend. Cloud-native CI/CD Lifecycle Cloud-native CI/CD Lifecycle adalah proses pengembangan perangkat lunak yang dirancang untuk digunakan dengan teknologi cloud. Proses ini berfokus pada kecepatan, skalabilitas, dan keberlanjutan.	Tidak ada		
---	-------------------------------------------------	--------------------------------------------	----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------	-------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

				Cloud-native CI/CD Lifecycle dapat digunakan untuk berbagai jenis aplikasi, termasuk aplikasi web, aplikasi seluler, dan aplikasi backend. - mendapatkan pembelajaran dari Oracle tentang Banking IT Trends, Ransomware, Cloud Overview, Oracle Database, AI, Oracle Platrofm. Adaptive bank adalah bank yang mengadopsi teknologi baru untuk meningkatkan efisiensi dan efektivitasnya. Adaptive bank memiliki tiga karakteristik utama: Intelligent: menggunakan data dan analisis untuk membuat keputusan yang lebih baik Agile: dapat dengan cepat beradaptasi dengan perubahan. Integral: terintegrasi dengan berbagai sistem dan perangkat. - Ransomware adalah jenis malware yang mengenkripsi data korban dan meminta tebusan untuk membuka kuncinya. Ransomware sering menyerang sistem back up, sehingga korban tidak dapat memulihkan datanya tanpa membayar tebusan. Tiga hal yang sering dilakukan ransomware adalah:			
--	--	--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	---------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------

			<p>Mengambil data korban Menghapus kemampuan korban untuk memulihkan datanya Melumpuhkan akses korban ke datanya.</p> <p>- Oracle adalah perusahaan perangkat lunak yang mengembangkan berbagai produk database, termasuk Oracle Database. Oracle Database adalah database relasional yang mendukung berbagai jenis data, termasuk data grafis, data geospasial, dan data JSON.</p> <p>- Enterprise architecture adalah proses mendefinisikan, mendesain, dan mengimplementasikan arsitektur sistem informasi perusahaan. Arsitektur sistem informasi perusahaan adalah kumpulan elemen-elemen yang bekerja sama untuk mendukung tujuan bisnis perusahaan.</p> <p>- Cyber crime adalah kejahatan yang dilakukan dengan menggunakan komputer atau jaringan komputer. Cyber crime dapat mencakup berbagai jenis kejahatan, termasuk penipuan, pencurian data, dan</p>		 
--	--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------

				<p>serangan siber.</p> <ul style="list-style-type: none"> - Product owner adalah peran dalam pengembangan perangkat lunak yang bertanggung jawab untuk menentukan prioritas pengembangan. Product owner bekerja dengan tim pengembangan untuk memastikan bahwa produk yang dikembangkan memenuhi kebutuhan pengguna. - UI/UX designer adalah peran dalam pengembangan perangkat lunak yang bertanggung jawab untuk mendesain antarmuka pengguna (UI) dan pengalaman pengguna (UX) dari produk perangkat lunak. UI/UX designer bekerja dengan tim pengembangan untuk menciptakan produk yang mudah digunakan dan nyaman. 			
4	Digilab BSI Yogyakarta :	Senin – Jumat, 9 - 13 Oktober 2023	08.30 – 17.30 WIB	<ul style="list-style-type: none"> - Mendapatkan materi tentang Enterprise Service Bus (ESB), ESB adalah sebuah middleware yang dapat digunakan untuk berbagai tujuan, termasuk integrasi code dan pengaturan arsitektur antar sistem. ESB memiliki berbagai manfaat, antara lain: Menghubungkan antar sistem Mengurangi waktu dan biaya menghubungkan aplikasi 	Tidak ada		

				<p>Mempercepat "time to market" aplikasi Arsitektur lebih fleksibel Meningkatkan relasi dengan partner</p> <ul style="list-style-type: none"> - Mendapatkan materi tentang Pemaparan materi Neraca Bank Akuntansi banking Syariah - Mendapatkan pembagian role dalam magang, sebagai developer atau Software Quality Assurance (SQA) - mendapatkan role sebagai back end developer. - Starting pembuatan mini project yang dilakukan oleh role developer sebagai awal dalam pembelajaran sebagai developer. - melakukan pembelajaran java springboot dari dasar hingga pembuatan rest api. 			
5	Digilab BSI Yogyakarta :	Senin – Jumat, 16 – 20 Oktober 2023	08.30 – 17.30 WIB	<ul style="list-style-type: none"> - melanjutkan pembuatan mini project. - melakukan praktek membuat project untuk back end membuat mock api Dukcapil untuk mendapatkan data ktp yang di inputkan dan mendapatkan validasi menggunakan postgree sebagai database dan menggunakan java 	Tidak ada		

				springboot sebagai framework pembuatan back end api. - diskusi tentang api contract antara back end dan pengeimplementasi front end.			
6	Digilab BSI Yogyakarta :	Senin – Jumat, 23 – 27 Oktober 2023	08.30 – 17.30 WIB	- melanjutkan pembuatan mini project - melanjutkan pembuatan api contract dan finalisasi flow dari aplikasi dan melakukan pembuatan ERD untuk database yang akan dibuat berdasarkan api contract. - melakukan set up environment untuk pembuatan java springboot, mulai dari memasukkan dependency yang diperlukan di dalamnya dalam pembuatan backend sebuah aplikasi. - melakukan pembagian task per orang. - mendapatkan task untuk membuat fitur transfer. - membuat struktur code dasar dan set up file project dan folder strcture.	Tidak ada		
7	Digilab BSI Yogyakarta :	Senin – Jumat, 30 Oktober – 3 November 2023	08.30 – 17.30 WIB	- melanjutkan pembuatan mini project, - mulai pembuatan untuk fitur transfer. - membuat database untuk transfer, rekening, dan transaksi dan bank. - membuat service untuk pembuatan api. - membuat service implementasi	Tidak ada		

				untuk menheritance service. - membuat repository untuk pengambilan endpoint api untuk request dan response. - membuat exception handler untuk menerima pesan yang salah.			
--	--	--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--

Catatan Pembimbing Lapangan/Dosen Pembimbing Praktik Magang / Dosen Pengampu Kelas Praktik Magang:

.....

.....

.....

.....

.....

Yogyakarta, 08 November 2013

Mahasiswa



(Baharudin Nur Hidayat)

Dosen Pengampu Kelas Praktik Magang



(Guntur Maulana Zamroni, B.Sc., M.Kom.)
NIPM. 19840309 201810 111 1205917

LOG BOOK PRAKTIK MAGANG MAHASISWA
PROGRAM STUDI S1 INFORMATIKA T.A 2023 / 2024
(WAJIB DIISI DAN MASUK DALAM PENILAIAN)

Nim : 200018263
 Nama Mahasiswa : Baharudin Nur Hidayat
 Judul Praktik Magang : BSI IT Digilab Internship Programme Cycle 1
 Dosen Pembimbing : Ir. Sri Winiarti, S.T., M.Cs.
 Pembimbing Lapangan : Taufiq Galang Adi Putranto, S. Si.









Petunjuk Pengisian Log Book



1. Log book di isi per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh pembimbing lapangan/ dosen pembimbing Praktik Magang
4. Log book per minggu di paraf oleh dosen pengampu kelas Praktik Magang
5. Jumlah bimbingan minimal 3x

Logbook Minggu 8 sd 10 (setelah UTS)

No	Kegiatan dan Lokasi Praktik Magang	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf Pembimbing Lapangan	Paraf Dosen Pembimbing Praktik Magang
		Hari/TGL	Jam Durasi				
1	Digilab BSI Yogyakarta	Senin – Jumat, 6 November – 10 November 2023	08.30 – 17.30 WIB	- melanjutkan pembuatan mini project - melanjutkan membuat fitur transfer pada aplikasi DigiBank - membuat class repository untuk query database melalui ORM	Tidak ada		
2	Digilab BSI Yogyakarta	Senin – Jumat, 13 November – 17 November 2023	08.30 – 17.30 WIB	- melanjutkan pembuatan mini project - mengembangkan fitur untuk menampilkan list daftar bank	Tidak ada		

				- membuat repository, service, entity, controller untuk pengembang fitur list bank - review dan retrospective sprint 2 - Menyusun task dan sprint planning sprint 3			
3	Digilab BSI Yogyakarta	Senin – Jumat, 20 November – 24 November 2023	08.30 – 17.30 WIB	- melanjutkan pembuatan mini project - mengembangkan fitur transfer - membuat repository, service, entity, controller untuk pengembang fitur transfer	Tidak ada		
4	Digilab BSI Yogyakarta	Senin – Jumat, 27 November – 01 Desember 2023	08.30 – 17.30 WIB	- melanjutkan pembuatan mini project - melanjutkan fitur transfer - membuat repository, service, entity, controller untuk pengembang fitur transfer - membuat unit test untuk transfer dan list bank - review dan retrospective sprint 3 - Menyusun task dan sprint planning sprint 4	Tidak ada		
5	Digilab BSI Yogyakarta	Senin – Jumat, 4 Desember – 8 Desember 2023	08.30 – 17.30 WIB	- melanjutkan pembuatan mini project - melanjutkan fitur transfer dengan mencari kondisi exception yang memungkinkan - melanjutkan membuat unit test - Fixing bug	Tidak ada		
6	Digilab BSI Yogyakarta	Senin – Jumat, 11 Desember – 15	08.30 – 17.30 WIB	- melanjutkan pembuatan mini project - fixing bug - review dan retrospective sprint 4 - evaluasi sprint 4	Tidak ada		

		Desember 2023		- reporting mini project kepada mentor - presentasi mentor mini project			
7	Digilab BSI Yogyakarta	Senin – Jumat, 18 Desember – 22 Desember 2023	08.30 – 17.30 WIB	- Rancangan BSI PINTER untuk real project - Analisa userflow aplikasi - design ui bsi pinter - revisi redisgn BSI PINTER - menganalisa kebutuhan backend - setup project BSI PINTER	Tidak ada		
8	Digilab BSI Yogyakarta	Senin – Jumat, 27 Desember – 29 Desember 2023	08.30 – 17.30 WIB	- pengembangan BSI PINTER - meeting internal backend BSI PINTER - set up environment, dependency git dan lain lain	Tidak ada		
9	Digilab BSI Yogyakarta	Selasa – Jumat, 2 Januari – 5 Januari 2023	08.30 – 17.30 WIB	- melanjutkan pengembangan BSI PINTER - merancang ERD BSI PINTER - merancang api contract dengan frontend - Analisa UI untuk menentukan kebutuhan - review erd dan api contract BSI PINTER	Tidak ada		
10	Digilab BSI Yogyakarta	Senin – Jumat, 8 Januari – 12 Januari 2023	08.30 – 17.30 WIB	- melanjutkan pengembangan BSI PINTER - membuat requirement untuk fitur feeds - implementasi database untk feeds - pengembangan fitur serach query feeds - membuat service, controller, repository dan DTO untuk fitur feeds	Tidak ada		

				- implementasi sementara dengan frontend			
11	Digilab BSI Yogyakarta	Senin – Jumat, 15 Januari – 19 Januari 2023	08.30 – 17.30 WIB	- melanjutkan pengembangan BSI PINTER - melanjutkan implementasi database untk feeds - melanjutkan pengembangan fitur serach query feeds - melanjutkan membuat service, controller, repository dan DTO untuk fitur feeds - implementasi release fitur feeds full sesuai MVP	Tidak ada		

Catatan Pembimbing Lapangan/Dosen Pembimbing Praktik Magang / Dosen Pengampu Kelas Praktik Magang:

.....

.....

.....

.....

Dosen Pengampu Kelas Praktik Magang



(Gunter Maulana Zamroni, B.Sc., M.Kom.)
NIP. 1. 19840309 201810 111 1205917

Yogyakarta, 23 Januari 2024

Mahasiswa



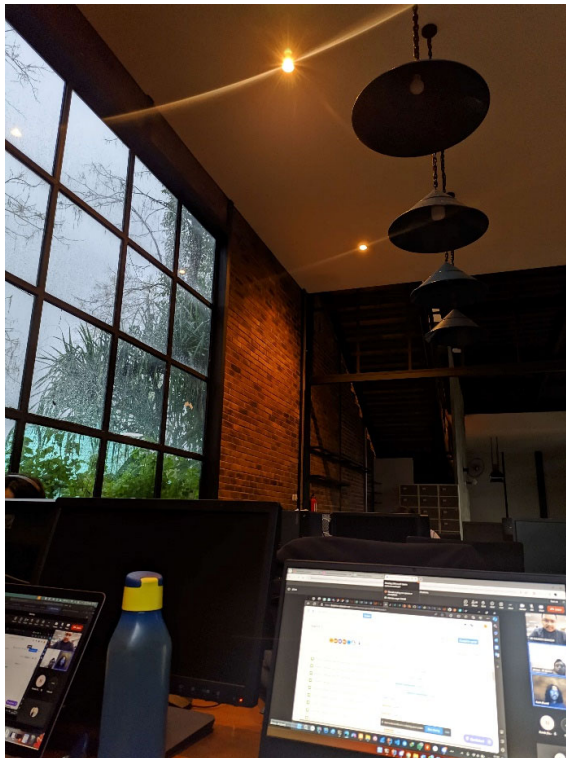
(Baharudin Nur Hidayat)

D. Dokumentasi Kegiatan Praktik Magang

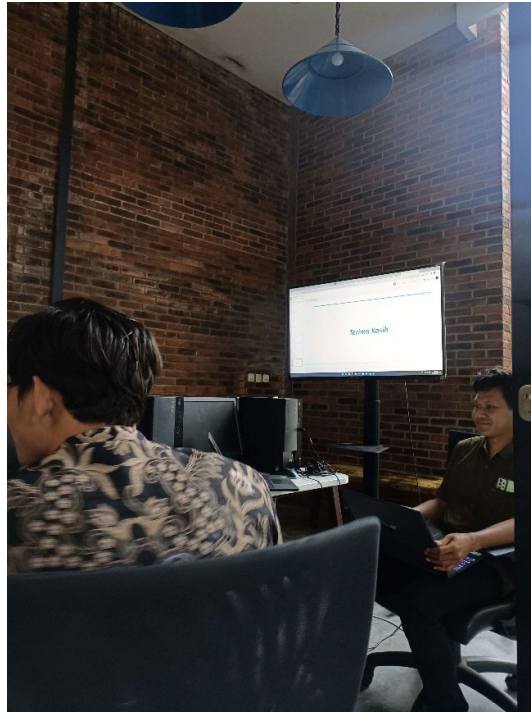
1. Suasana dan ruangan magang di PT. Bank Syariah Indonesia



2. Kegiatan Praktik Magang



3. Presentasi progress dan review sprint



4. Foto bersama praktik magang



E. Lampiran *listing code*

1. Feeds Repository

```
1. @Repository
2. public interface FeedsRepository extends JpaRepository<Feeds,
   Long>, JpaRepository<Feeds>{
3. Page<Feeds>findBySharedIsTrue(Pageable pageable);
4. Page<Feeds>findByCategoryFeeds(Category categoryFeeds, Pageable
   pageable);
5. Page<Feeds>findBySharedIsTrueAndCategoryFeeds(Category
   categoryFeeds, Pageable pageable);
6. @Query("SELECT f FROM Feeds f WHERE (LOWER(f.judulFeeds) LIKE
   LOWER(CONCAT('%', :search, '%')) OR LOWER(f.bodyFeeds) LIKE
   LOWER(CONCAT('%', :search, '%'))) AND f.shared = true")
7. Page<Feeds>searchByJudulFeedsOrAndBodyFeedsAndSharedIsTrue(String
   search, Pageable pageable);
8. @Query("SELECT f FROM Feeds f WHERE (LOWER(f.judulFeeds) LIKE
   LOWER(CONCAT('%', :search, '%')) OR LOWER(f.bodyFeeds) LIKE
   LOWER(CONCAT('%', :search, '%')))")
9. Page<Feeds>searchByJudulFeedsOrAndBodyFeeds(String search,
   Pageable pageable);
10. @Query("SELECT f FROM Feeds f WHERE (LOWER(f.judulFeeds)
   LIKE LOWER(CONCAT('%', :search, '%')) OR LOWER(f.bodyFeeds) LIKE
   LOWER(CONCAT('%', :search, '%'))) AND f.shared = true AND
   f.categoryFeeds = :categoryFeeds")
11. Page<Feeds>searchByJudulFeedsAndAndBodyFeedsAndSharedIsTrue
   eAndCategoryFeeds(String search,Category categoryFeeds, Pageable
   pageable);
12. @Query("SELECT f FROM Feeds f WHERE (LOWER(f.judulFeeds)
   LIKE LOWER(CONCAT('%', :search, '%')) OR LOWER(f.bodyFeeds) LIKE
   LOWER(CONCAT('%', :search, '%'))) AND f.categoryFeeds =
   :categoryFeeds")
13. Page<Feeds>searchByJudulFeedsAndAndBodyFeedsAndCategoryFeeds
   ds(String search,Category categoryFeeds, Pageable pageable);
14. }
```

2. Feeds Service

```
1. public interface FeedService {
2. FeedsDto getFeeds(long idUser,boolean bsiGroup,int page, int
   size);
3. FeedsDto getFeedsSearch(long idUser,boolean bsiGroup,int page, int
   size, String search);
4. ResponseLikeDto likeFeeds(RequestLikeDto requestLikeDto,long
   idUser);
5. ResponseLikeDto unlikeFeeds(RequestLikeDto requestLikeDto,long
   idUser);
6. DetailFeedsDto getDetailFeeds(long idFeeds, long idUser);
7. GlobalResponse commentFeeds(RequestCommentDto requestCommentDto,
   long idUser);
8. KomentarDto getCommentFeeds(long idFeeds, int page, int size);
9. GlobalResponse deleteFeeds( long idFeeds);
10. GlobalResponse
   createFeedsChallenge(RequestCreateChallengeFeeds
   requestCreateChallengeFeeds, long idUser, long challangeId);
11. GlobalResponse
   createFeeds(RequestFeedCreateDto
   requestFeedCreateDto, long idUser);
```

```

12.     GlobalResponse                editFeeds (RequestEditFeedsDto
requestEditFeedsDto, long idUser);
13.     ResponseActivityFeedsDtos getActivityFeeds(long idUser, int
page, int size);
14.     }

```

3. Feeds Service Impl

```

1. @Service
2. @AllArgsConstructor
3. public class FeedServiceImpl implements FeedService {
4.
5.     private final FeedsRepository feedsRepository;
6.     private final LikeFeedsRepository likeFeedsRepository;
7.     private final UserRepository userRepository;
8.     private         final         ActivityFeedsRepository
activityFeedsRepository;
9.     private final FeedsMapper feedsMapper;
10.     private final KomentarMapper komentarMapper;
11.     private         final         KomentarFeedsRepository
komentarFeedsRepository;
12.     private final GcpService gcpService;
13.     private final ActivityFeedsMapper activityFeedsMapper;
14.     private final ChallengeRepository challengeRepository;
15.     private         final         ChallengeProgressRepository
challengeProgressRepository;
16.     private         final         UserPoinHistoryRepository
userPoinHistoryRepository;
17.
18.
19.     @Override
20.     public FeedsDto getFeeds(long idUser, boolean bsiGroup,
int page, int size) {
21.         Pageable pageable = PageRequest.of(((page > 0) ?
page - 1 : page), size, Sort.by(Sort.Direction.DESC,
"createdAt"));
22.         Page<Feeds> feedsFilter;
23.
24.         Optional<User>         user         =
userRepository.findById(idUser);
25.         if (user.isEmpty()) {
26.             throw new FeedsErrorException("User not
found");
27.         }
28.         Role role = user.get().getRole();
29.
30.         if (role == Role.ADMIN || role == Role.DH) {
31.             feedsFilter         =
feedsRepository.findAll(pageable);
32.             if (bsiGroup) {
33.                 feedsFilter         =
feedsRepository.findByCategoryFeeds (Category.BSIGROUP, pageable);
34.             }
35.         } else if (role == Role.USER) {
36.             feedsFilter         =
feedsRepository.findBySharedIsTrue (pageable);
37.             if (bsiGroup) {

```

```

38.         feedsFilter =
feedsRepository.findBySharedIsTrueAndCategoryFeeds (Category.BSIG
ROUP,pageable);
39.     }
40.     } else {
41.         throw new FeedsNotFoundException("feeds not
found");
42.     }
43.
44.     return getFeedsDto(feedsFilter, user);
45. }
46.
47. @Override
48. public FeedsDto getFeedsSearch(long idUser, boolean
bsiGroup, int page, int size, String search) {
49.     Pageable pageable = PageRequest.of(((page > 0) ?
page - 1 : page), size, Sort.by(Sort.Direction.DESC,
"createdAt"));
50.     Page<Feeds> feedsFilter;
51.
52.     Optional<User> user =
userRepository.findById(idUser);
53.     if (user.isEmpty()) {
54.         throw new FeedsNotFoundException("User not
found");
55.     }
56.     Role role = user.get().getRole();
57.     if (search.isEmpty()) {
58.         if (role == Role.ADMIN || role == Role.DH) {
59.             feedsFilter =
feedsRepository.findAll(pageable);
60.             if (bsiGroup) {
61.                 feedsFilter =
feedsRepository.findByCategoryFeeds (Category.BSIGROUP,pageable);
62.             }
63.         } else if (role == Role.USER) {
64.             feedsFilter =
feedsRepository.findBySharedIsTrue (pageable);
65.             if (bsiGroup) {
66.                 feedsFilter =
feedsRepository.findBySharedIsTrueAndCategoryFeeds (Category.BSIG
ROUP,pageable);
67.             }
68.         } else {
69.             throw new FeedsNotFoundException("feeds not
found");
70.         }
71.     } else {
72.         if (role == Role.ADMIN || role == Role.DH) {
73.             feedsFilter =
feedsRepository.searchByJudulFeedsOrAndBodyFeeds (search,
pageable);
74.             if (bsiGroup) {
75.                 feedsFilter =
feedsRepository.searchByJudulFeedsAndAndBodyFeedsAndCategoryFeed
s(search, Category.BSIGROUP, pageable);
76.             }
77.         } else if (role == Role.USER) {

```

```

78.         feedsFilter =
feedsRepository.searchByJudulFeedsOrAndBodyFeedsAndSharedIsTrue(
search, pageable);
79.         if (bsiGroup) {
80.             feedsFilter =
feedsRepository.searchByJudulFeedsAndAndBodyFeedsAndSharedIsTrue
AndCategoryFeeds(search, Category.BSIGROUP, pageable);
81.         }
82.     } else {
83.         throw new FeedsNotFoundException("feeds not
found");
84.     }
85.
86.     }
87.
88.     return getFeedsDto(feedsFilter, user);
89. }
90.
91.     private FeedsDto getFeedsDto(Page<Feeds> feedsFilter,
Optional<User> user) {
92.         List<ListFeedsDto> listFeedsDto =
feedsMapper.feedsToFeedsResponses(feedsFilter.getContent());
93.         for (int i = 0; i < listFeedsDto.size(); i++) {
94.             listFeedsDto.get(i).setUser(UserFeedsDto.builder()
95.                 .id(feedsFilter.getContent().get(i).getIdUser().getIdUser())
96.                 .namaLengkap(feedsFilter.getContent().get(i).getIdUser().getNama
Lengkap())
97.                 .photoUrl(feedsFilter.getContent().get(i).getIdUser().getPhotoUr
l())
98.                 .build());
99.             Set<LikeFeeds> likeFeeds =
likeFeedsRepository.findByIdFeeds(feedsFilter.getContent().get(i
));
100.             listFeedsDto.get(i).setLikeCount(likeFeeds.size());
101.             Set<KomentarFeeds> komentarFeeds =
komentarFeedsRepository.findByIdFeeds(feedsFilter.getContent().g
et(i));
102.             listFeedsDto.get(i).setKomentarCount(komentarFeeds.size());
103.             if (user.isEmpty()){
104.                 throw new FeedsNotFoundException("User not
found");
105.             }
106.             listFeedsDto.get(i).setLike(likeFeedsRepository.findByIdFeedsAnd
IdUser(feedsFilter.getContent().get(i), user.get()).isPresent());
107.         }
108.
109.         return FeedsDto.builder()
110.             .status(HttpStatus.OK.value())
111.             .message("Success")
112.             .feeds(listFeedsDto)
113.             .build();
114.     }

```

```

115.
116.         @Override
117.         public ResponseLikeDto likeFeeds (RequestLikeDto
118. requestLikeDto, long idUser) {
119.         Optional<Feeds> feeds =
120. feedsRepository.findById(requestLikeDto.getIdFeeds());
121.         Optional<User> user =
122. userRepository.findById(idUser);
123.
124.         if (feeds.isEmpty()) {
125.             throw new FeedsNotFoundException("Feeds not
126. found");
127.         }
128.
129.         if (user.isEmpty()) {
130.             throw new FeedsNotFoundException("User not
131. found");
132.         }
133.
134.         if
135. (likeFeedsRepository.findByIdFeedsAndIdUser(feeds.get(),
136. user.get()).isPresent()) {
137.             throw new FeedsErrorException("You already like
138. this feeds");
139.         }
140.
141.         LikeFeeds likeFeeds = new LikeFeeds();
142.         likeFeeds.setIdUser(user.get());
143.         likeFeeds.setIdFeeds(feeds.get());
144.         likeFeedsRepository.save(likeFeeds);
145.
146.         ResponseLikeDto responseLikeDto = new
147. ResponseLikeDto();
148.         responseLikeDto.setStatus(HttpStatus.OK.value());
149.         responseLikeDto.setMessage("Success");
150.         responseLikeDto.setLike(true);
151.
152.         return responseLikeDto;
153.     }
154.
155.     @Override
156.     public ResponseLikeDto unlikeFeeds (RequestLikeDto
157. requestLikeDto, long idUser) {
158.         Optional<Feeds> feeds =
159. feedsRepository.findById(requestLikeDto.getIdFeeds());
160.         Optional<User> user =
161. userRepository.findById(idUser);
162.
163.         if (feeds.isEmpty()) {
164.             throw new FeedsNotFoundException("Feeds not
165. found");
166.         }
167.
168.         if (user.isEmpty()) {
169.             throw new FeedsNotFoundException("User not
170. found");
171.         }
172.
173.
174.
175.
176.
177.
178.

```

```

159.         if
            (likeFeedsRepository.findByIdFeedsAndIdUser(feeds.get(),
user.get()).isEmpty()) {
160.             throw new FeedsErrorException("You already
unlike this feeds");
161.         }
162.
163.         likeFeedsRepository.delete(likeFeedsRepository.findByIdFeedsAndI
dUser(feeds.get(), user.get()).get());
164.
165.         ResponseLikeDto responseLikeDto = new
ResponseLikeDto();
166.         responseLikeDto.setStatus(HttpStatus.OK.value());
167.         responseLikeDto.setMessage("Success");
168.         responseLikeDto.setLike(false);
169.
170.         return responseLikeDto;
171.     }
172.
173.
174.     @Override
175.     public DetailFeedsDto getDetailFeeds(long idFeeds, long
idUser) {
176.         Optional<Feeds> feeds =
feedsRepository.findById(idFeeds);
177.         if (feeds.isEmpty()) {
178.             throw new FeedsNotFoundException("Feeds not
found");
179.         }
180.         Optional<User> user =
userRepository.findById(idUser);
181.         if (user.isEmpty()) {
182.             throw new FeedsNotFoundException("User not
found");
183.         }
184.         DataDetailFeedsDto detailFeedsDto = new
DataDetailFeedsDto();
185.         detailFeedsDto.setIdFeeds(feeds.get().getIdFeeds());
186.         detailFeedsDto.setNamaUser(feeds.get().getIdUser().getNamaLengka
p());
187.         detailFeedsDto.setUserPhotoUrl(feeds.get().getIdUser().getPhotoU
rl());
188.         detailFeedsDto.setBodyFeeds(feeds.get().getBodyFeeds());
189.         detailFeedsDto.setCreatedAt(feeds.get().getCreatedAt());
190.         detailFeedsDto.setJudulFeeds(feeds.get().getJudulFeeds());
191.         detailFeedsDto.setLikeCount(likeFeedsRepository.findByIdFeeds(fe
eds.get()).size());
192.         detailFeedsDto.setCommentCount(komentarFeedsRepository.findByIdF
eeds(feeds.get()).size());

```



```

193.     detailFeedsDto.setLiked(likeFeedsRepository.findByIdFeedsAndIdUser(feeds.get(), user.get()).isPresent());
194.
195.         return DetailFeedsDto.builder()
196.             .status(HttpStatus.OK.value())
197.             .message("Success")
198.             .data(detailFeedsDto)
199.             .build();
200.
201.     }
202.
203.     @Override
204.     public GlobalResponse commentFeeds(RequestCommentDto requestCommentDto, long idUser) {
205.         Optional<Feeds> feeds =
206.             feedsRepository.findById(requestCommentDto.getIdFeeds());
207.         Optional<User> user =
208.             userRepository.findById(idUser);
209.         if (feeds.isEmpty()) {
210.             throw new FeedsNotFoundException("Feeds not found");
211.         }
212.         if (user.isEmpty()) {
213.             throw new FeedsNotFoundException("User not found");
214.         }
215.         KomentarFeeds komentarFeeds = new KomentarFeeds();
216.         komentarFeeds.setIdFeeds(feeds.get());
217.         komentarFeeds.setIdUser(user.get());
218.         komentarFeeds.setBodyKomentar(requestCommentDto.getComment());
219.         komentarFeedsRepository.save(komentarFeeds);
220.
221.         return GlobalResponse.builder()
222.             .status(HttpStatus.OK.value())
223.             .message("Success")
224.             .build();
225.     }
226.
227.     @Override
228.     public KomentarDto getCommentFeeds(long idFeeds, int page, int size) {
229.         Pageable pageable = PageRequest.of((page > 0) ? page - 1 : page, size, Sort.by(Sort.Direction.ASC, "idReply"));
230.         Page<KomentarFeeds> komentarFilter;
231.         Optional<Feeds> feeds =
232.             feedsRepository.findById(idFeeds);
233.         if (feeds.isEmpty()) {
234.             throw new FeedsNotFoundException("Feeds not found");
235.         }
236.         komentarFilter =

```

```

237.         List<ListKomentarFeedsDto>      KomentarFeeds      =
    komentarMapper.komentarToKomentarResponses(komentarFilter.getCon
    tent());
238.         for (int i = 0; i < KomentarFeeds.size(); i++) {
239.             KomentarFeeds.get(i).setUser(UserFeedsDto.builder()
240.                 .id(komentarFilter.getContent().get(i).getIdUser().getIdUser())
241.                 .namaLengkap(komentarFilter.getContent().get(i).getIdUser().getN
    amaLengkap())
242.                 .photoUrl(komentarFilter.getContent().get(i).getIdUser().getPhot
    oUrl())
243.                 .build());
244.         }
245.
246.         return KomentarDto.builder()
247.             .status(HttpStatus.OK.value())
248.             .message("Success")
249.             .komentar(KomentarFeeds)
250.             .build();
251.     }
252.
253.     @Override
254.     public GlobalResponse deleteFeeds(long idFeeds) {
255.         Optional<Feeds>      feeds      =
    feedsRepository.findById(idFeeds);
256.         if (feeds.isEmpty()) {
257.             throw new FeedsNotFoundException("Feeds not
    found");
258.         }
259.         feedsRepository.delete(feeds.get());
260.         return GlobalResponse.builder()
261.             .status(HttpStatus.OK.value())
262.             .message("Success")
263.             .build();
264.     }
265.
266.     @Override
267.     public      GlobalResponse
    createFeedsChallenge(RequestCreateChallengeFeeds
    requestCreateChallengeFeeds, long idUser, long challangeId) {
268.         Optional<User>      userOptional      =
    userRepository.findById(idUser);
269.         User user;
270.         if (userOptional.isEmpty()) {
271.             throw new FeedsNotFoundException("User not
    found");
272.         }else{
273.             user = userOptional.get();
274.         }
275.
276.         Feeds feeds = new Feeds();
277.         feeds.setIdUser(user);
278.         feeds.setJudulFeeds(requestCreateChallengeFeeds.getJudulFeeds())
    ;

```

```

279.     feeds.setBodyFeeds(requestCreateChallengeFeeds.getBodyFeeds());
280.         feeds.setCategoryFeeds(Category.CHALLENGE);
281.     feeds.setShared(requestCreateChallengeFeeds.isShared());
282.         if (requestCreateChallengeFeeds.getImage() != null)
283.         {
284.             List<FileGcpDto> images =
285.             gcpService.uploadFiles(requestCreateChallengeFeeds.getImage());
286.             feeds.setImageFeed(images.get(0).getFileUrl());
287.         } else {
288.             feeds.setImageFeed(null);
289.         }
290.         feedsRepository.save(feeds);
291.         ActivityFeeds activityFeeds = new ActivityFeeds();
292.         activityFeeds.setIdUser(user);
293.         activityFeeds.setIdFeeds(feeds);
294.         activityFeeds.setPoin(requestCreateChallengeFeeds.getPoin());
295.         activityFeedsRepository.save(activityFeeds);
296.         UserPoinHistory userPoinHistory = new
297.         UserPoinHistory();
298.         userPoinHistory.setUser(user);
299.         userPoinHistory.setScore(requestCreateChallengeFeeds.getPoin());
300.         userPoinHistory.setKeterangan(Keterangan.CHALLENGE);
301.         userRepository.save(user);
302.
303.         challengeRepository.findByIdChallenge(challengeId).ifPresentOrElse(
304.             (challenge)->{
305.                 ChallengeProgress challengeProgress = new
306.                 ChallengeProgress();
307.                 challengeProgress.setUser(user);
308.                 challengeProgress.setProductKnowledge(challenge.getProductKnowle
309.                 dge());
310.                 challengeProgress.setIsFinished(true);
311.                 challengeProgress.setChallenge(challenge);
312.                 challengeProgressRepository.save(challengeProgress);
313.             }, ()->{
314.                 //throw challenge not found exception
315.             });
316.         //ADD USER POINT
317.         Challenge challenge =
318.         challengeRepository.findById(challengeId)
319.         .orElseThrow(() -> new
320.         RuntimeException("Challenge Tidak Ditemukan"));
321.         user.setPoint(challenge.getPoinChallenge());
322.         userRepository.save(user);

```

```

321.         UserPoinHistory      userPostPoint      =      new
    UserPoinHistory();
322.         userPostPoint.setUser(user);
323.
    userPostPoint.setScore(challenge.getPoinChallenge());
324.         userPostPoint.setKeterangan(Keterangan.CHALLENGE);
325.         userPoinHistoryRepository.save(userPoinHistory);
326.
327.         return GlobalResponse.builder()
328.             .status(HttpStatus.OK.value())
329.             .message("Feeds created successfully")
330.             .build();
331.     }
332.
333.     @Override
334.     public GlobalResponse createFeeds(RequestFeedCreateDto
    requestFeedCreateDto, long idUser) {
335.         Optional<User>          user              =
    userRepository.findById(idUser);
336.         if (user.isEmpty()) {
337.             throw new FeedsNotFoundException("User not
    found");
338.         }
339.         if (user.get().getRole() == Role.USER) {
340.             throw new FeedsErrorException("User not allowed
    to create feeds");
341.         }
342.         Feeds feeds = new Feeds();
343.         feeds.setIdUser(user.get());
344.         if (requestFeedCreateDto.getJudulFeeds().isEmpty()
    || requestFeedCreateDto.getBodyFeeds().isEmpty()){
345.             throw new FeedsErrorException("Data body atau
    judul tidak boleh kosong");
346.         }
347.
    feeds.setJudulFeeds(requestFeedCreateDto.getJudulFeeds());
348.
    feeds.setBodyFeeds(requestFeedCreateDto.getBodyFeeds());
349.         feeds.setCategoryFeeds(Category.BSIGROUP);
350.         feeds.setShared(true);
351.         if (requestFeedCreateDto.getImage() != null) {
352.             List<FileGcpDto>          images          =
    gcpService.uploadFiles(requestFeedCreateDto.getImage());
353.
    feeds.setImageFeed(images.get(0).getFileUrl());
354.         } else {
355.             feeds.setImageFeed(null);
356.         }
357.         feedsRepository.save(feeds);
358.
359.         return GlobalResponse.builder()
360.             .status(HttpStatus.OK.value())
361.             .message("Feeds created successfully")
362.             .build();
363.     }
364.
365.     @Override
366.     public GlobalResponse editFeeds(RequestEditFeedsDto
    requestEditFeedsDto, long idUser) {

```

```

367.         Optional<User>                user                =
    userRepository.findById(idUser);
368.         if (user.isEmpty()) {
369.             throw new FeedsNotFoundException("User not
    found");
370.         }
371.         if (user.get().getRole() == Role.USER) {
372.             throw new FeedsErrorException("User not allowed
    to edit feeds");
373.         }
374.
375.         Optional<Feeds>                feeds                =
    feedsRepository.findById(requestEditFeedsDto.getIdFeeds());
376.         if (feeds.isEmpty()) {
377.             throw new FeedsNotFoundException("Feeds not
    found");
378.         }
379.         feeds.get().setIdFeeds(requestEditFeedsDto.getIdFeeds());
380.         feeds.orElseThrow(() ->                new
    FeedsNotFoundException("Feeds not found"));
381.         feeds.get().setJudulFeeds(requestEditFeedsDto.getJudulFeeds());
382.         feeds.get().setBodyFeeds(requestEditFeedsDto.getBodyFeeds());
383.         feeds.get().setShared(true);
384.         if (requestEditFeedsDto.getImage() != null) {
385.             List<FileGcpDto>                images                =
    gcpService.uploadFiles(requestEditFeedsDto.getImage());
386.         feeds.get().setImageFeed(images.get(0).getFileUrl());
387.         } else {
388.             feeds.get().setImageFeed(null);
389.         }
390.         feedsRepository.save(feeds.get());
391.
392.         return GlobalResponse.builder()
393.             .status(HttpStatus.OK.value())
394.             .message("Feeds edited successfully")
395.             .build();
396.     }
397.
398.     @Override
399.     public ResponseActivityFeedsDtos getActivityFeeds(long
    idUser, int page, int size) {
400.         Pageable pageable = PageRequest.of(((page > 0) ?
    page - 1 : page), size, Sort.by(Sort.Direction.DESC,
    "idActivityFeeds"));
401.         Page<ActivityFeeds> activityFeedsFilter;
402.         Optional<User>                user                =
    userRepository.findById(idUser);
403.         if (user.isEmpty()) {
404.             throw new FeedsNotFoundException("User not
    found");
405.         }
406.         activityFeedsFilter                =
    activityFeedsRepository.findByIdUser_IdUser(idUser, pageable);
407.         List<ResponseListActivityFeedsDto>
    listActivityFeedsDtos                =

```

```

    activityFeedsMapper.activityFeedsToActivityFeedsResponses(activityFeedsFilter.getContent());
408.
409.         for (int i = 0; i < listActivityFeedsDtos.size();
410.             i++) {
411.             listActivityFeedsDtos.get(i).setJudulFeeds(activityFeedsFilter.getContent().get(i).getIdFeeds().getJudulFeeds());
412.             listActivityFeedsDtos.get(i).setBodyFeeds(activityFeedsFilter.getContent().get(i).getIdFeeds().getBodyFeeds());
413.             listActivityFeedsDtos.get(i).setCreatedDate(activityFeedsFilter.getContent().get(i).getIdFeeds().getCreatedAt());
414.             listActivityFeedsDtos.get(i).setShared(activityFeedsFilter.getContent().get(i).getIdFeeds().isShared());
415.             listActivityFeedsDtos.get(i).setImageUser(activityFeedsFilter.getContent().get(i).getIdUser().getPhotoUrl());
416.         }
417.         return ResponseActivityFeedsDtos.builder()
418.             .status(HttpStatus.OK.value())
419.             .message("Success")
420.             .data(listActivityFeedsDtos)
421.             .build();
422.     }
423. }

```

4. Feeds Mapper

```

1. @Mapper(componentModel = "spring")
2. public interface FeedsMapper {
3.     List<ListFeedsDto> feedsToFeedsResponses(List<Feeds> feeds);
4. }

```

5. Feeds Controller

```

1. @RestController
2. @AllArgsConstructor
3. @RequestMapping("api/v1/feeds")
4. public class FeedsController {
5.     private FeedService feedService;
6.     private UserRepository userRepository;
7.     JwtService jwtService;
8.     @GetMapping("/hello")
9.     public Object getHelloWorlds() {
10.         return ResponseEntity.status(HttpStatus.OK).body("Hello World");
11.     }
12.     @GetMapping("/getFeeds")
13.     public ResponseEntity<?> getFeedsDetail(
14.         @RequestHeader(value = "Authorization") String
            authorizationHeader,
15.         @RequestParam(value = "page", defaultValue = "0") int page,
16.         @RequestParam(value = "size", defaultValue = "20") int size,

```

```

17.     @RequestParam(value = "bsiGroup" ,required = false) boolean
    bsiGroup
18.     ){
19.     String token = authorizationHeader.substring("Bearer
    ".length());
20.     String userName = jwtService.extractUserName(token);
21.     Optional<User> user = userRepository.findByEmail(userName);
22.     if (user.isEmpty()){
23.     return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
24.     }
25.     long idUser = user.get().getIdUser();
26.     FeedsDto listFeeds = feedService.getFeeds(idUser,bsiGroup,
    page, size);
27.     return new ResponseEntity<>(listFeeds, HttpStatus.OK);
28.     }
29.     @GetMapping("/getFeeds/Search")
30.     public ResponseEntity<?> getFeedsSearch(
31.     @RequestHeader(value = "Authorization") String
    authorizationHeader,
32.     @RequestParam(value = "search", defaultValue = "",required
    = false) String search,
33.     @RequestParam(value = "bsiGroup" ,required = false) boolean
    bsiGroup,
34.     @RequestParam(value = "page", defaultValue = "0") int page,
35.     @RequestParam(value = "size", defaultValue = "20") int size
36.     ){
37.     String token = authorizationHeader.substring("Bearer
    ".length());
38.     String userName = jwtService.extractUserName(token);
39.     Optional<User> user = userRepository.findByEmail(userName);
40.     if (user.isEmpty()){
41.     return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
42.     }
43.     long idUser = user.get().getIdUser();
44.     FeedsDto listFeeds =
    feedService.getFeedsSearch(idUser,bsiGroup, page, size, search);
45.     return new ResponseEntity<>(listFeeds, HttpStatus.OK);
46.     }
47.     @GetMapping("/detailFeeds/{idFeeds}")
48.     public ResponseEntity<?> getDetailFeeds(
49.     @RequestHeader(value = "Authorization") String
    authorizationHeader,
50.     @PathVariable(value = "idFeeds") long idFeeds
51.     ){
52.     String token = authorizationHeader.substring("Bearer
    ".length());
53.     String userName = jwtService.extractUserName(token);
54.     Optional<User> user = userRepository.findByEmail(userName);
55.     if (user.isEmpty()){
56.     return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
57.     }
58.     long idUser = user.get().getIdUser();
59.     return
    new
    ResponseEntity<>(feedService.getDetailFeeds(idFeeds,idUser),
    HttpStatus.OK);
60.     }
61.     @PostMapping("/likeFeeds")
62.     public ResponseEntity<?> likeFeeds(

```

```

63.     @RequestHeader(value      =      "Authorization")      String
      authorizationHeader,
64.     @RequestBody RequestLikeDto requestLikeDto
65.     ){
66.     String token = authorizationHeader.substring("Bearer
      ".length());
67.     String userName = jwtService.extractUserName(token);
68.     Optional<User> user = userRepository.findByEmail(userName);
69.     if (user.isEmpty()){
70.     return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
71.     }
72.     long idUser = user.get().getIdUser();
73.     return
      new
      ResponseEntity<>(feedService.likeFeeds(requestLikeDto,idUser),
      HttpStatus.OK);
74.     }
75.     @PostMapping("/unlikeFeeds")
76.     public ResponseEntity<?> unlikeFeeds(
77.     @RequestHeader(value      =      "Authorization")      String
      authorizationHeader,
78.     @RequestBody RequestLikeDto requestLikeDto
79.     ){
80.     String token = authorizationHeader.substring("Bearer
      ".length());
81.     String userName = jwtService.extractUserName(token);
82.     Optional<User> user = userRepository.findByEmail(userName);
83.     if (user.isEmpty()){
84.     return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
85.     }
86.     long idUser = user.get().getIdUser();
87.     return
      new
      ResponseEntity<>(feedService.unlikeFeeds(requestLikeDto,idUser),
      HttpStatus.OK);
88.     }
89.     @GetMapping("/komentarFeeds")
90.     public ResponseEntity<?> getCommentFeeds(
91.     @RequestParam(value = "idFeeds") String idFeeds,
92.     @RequestParam(value = "page", defaultValue = "0") int page,
93.     @RequestParam(value = "size", defaultValue = "20") int size
94.     ){
95.     return
      new
      ResponseEntity<>(feedService.getCommentFeeds(Long.parseLong(idFe
      eds),page,size), HttpStatus.OK);
96.     }
97.     @PostMapping("/postkomentarFeeds")
98.     public ResponseEntity<?> commentFeeds(
99.     @RequestHeader(value      =      "Authorization")      String
      authorizationHeader,
100.    @RequestBody RequestCommentDto requestCommentDto
101.    ){
102.    String token = authorizationHeader.substring("Bearer
      ".length());
103.    String userName = jwtService.extractUserName(token);
104.    Optional<User> user = userRepository.findByEmail(userName);
105.    if (user.isEmpty()){
106.    return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
107.    }
108.    long idUser = user.get().getIdUser();

```



```

109.     return                                                                 new
        ResponseEntity<>(feedService.commentFeeds(requestCommentDto,idUs
er), HttpStatus.OK);
110.     }
111.     @PostMapping(value =
{"{/createFeedsChallenge/challenge/{id}"}
112.     public ResponseEntity<?> createFeedsChallenge(
113.     @RequestHeader(value = "Authorization") String
authorizationHeader,
114.     @ModelAttribute RequestCreateChallengeFeeds
requestCreateChallengeFeeds,
115.     @PathVariable("id") long challengeId
116.     ){
117.     String token = authorizationHeader.substring("Bearer
".length());
118.     String userName = jwtService.extractUserName(token);
119.     Optional<User> user = userRepository.findByEmail(userName);
120.     if (user.isEmpty()){
121.     return new ResponseEntity<>(HttpStatus.UNAUTHORIZED);
122.     }
123.     long idUser = user.get().getIdUser();
124.     return                                                                 new
        ResponseEntity<>(feedService.createFeedsChallenge(requestCreateC
hallengeFeeds,idUser,challengeId), HttpStatus.OK);
125.     }
126.     @GetMapping("/getActivityFeeds")
127.     public ResponseEntity<?> getActivityFeeds(
128.     @RequestHeader(value = "Authorization") String
authorizationHeader,
129.     @RequestParam(value = "page", defaultValue = "0") int page,
130.     @RequestParam(value = "size", defaultValue = "20") int size
131.     ){
132.     String token = authorizationHeader.substring("Bearer
".length());
133.     String userName = jwtService.extractUserName(token);
134.     Optional<User> user = userRepository.findByEmail(userName);
135.     if (user.isEmpty()){
136.     return new ResponseEntity<>(HttpStatus.BAD_REQUEST);
137.     }
138.     long idUser = user.get().getIdUser();
139.     return                                                                 new
        ResponseEntity<>(feedService.getActivityFeeds(idUser,page,size),
HttpStatus.OK);
140.     }
141.     }

```

6. Feeds Dto

```

1. @Getter
2. @Setter
3. @AllArgsConstructor
4. @NoArgsConstructor
5. @Data
6. @Builder
7. public class FeedsDto {
8.     private int status;
9.     private String message;
10.     private List<ListFeedsDto> feeds;

```

```
11.     }
12.
13.     package com.bsi.bsipinter.dto.Feeds.listFeeds;
14.     import com.bsi.bsipinter.dto.Feeds.UserFeedsDto;
15.     import com.bsi.bsipinter.model.enums.Category;
16.     import lombok.*;
17.     import java.sql.Timestamp;
18.     @Getter
19.     @Setter
20.     @AllArgsConstructor
21.     @NoArgsConstructor
22.     @Data
23.     @Builder
24.     public class ListFeedsDto {
25.         private long idFeeds;
26.         private String judulFeeds;
27.         private String bodyFeeds;
28.         private String imageFeed;
29.         private Category categoryFeeds;
30.         private Timestamp createdAt;
31.         private boolean shared;
32.         private UserFeedsDto user;
33.         private int likeCount;
34.         private int komentarCount;
35.         private boolean isLike;
36.     }
```