

SKRIPSI

PENGEMBANGAN FRONTEND APLIKASI *MOBILE* SIM-RS AT TURROTS, ATSOFT E-GOVERNMENT, DAN ASN MEMAYU DI CV ATSOFT TEKNOLOGI



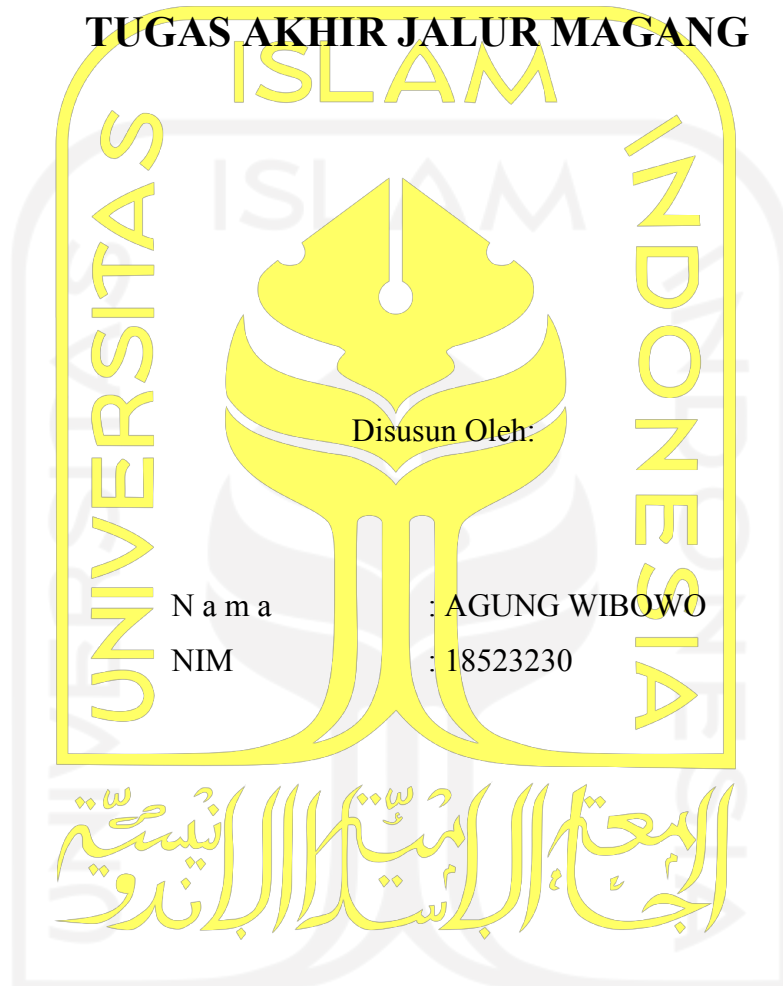
Disusun Oleh:

N a m a : AGUNG WIBOWO
NIM : 18523230

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2021

**PENGEMBANGAN FRONTEND APLIKASI *MOBILE* SIM-RS
AT TURROTS, ATSOFT E-GOVERNMENT, DAN ASN
MEMAYU DI CV ATSOFT TEKNOLOGI**

TUGAS AKHIR JALUR MAGANG



Yogyakarta, 25 Desember 2021

Pembimbing,

Rahadian Kurniawan, S.Kom., M.Kom.

**PENGEMBANGAN FRONTEND APLIKASI *MOBILE* SIM-RS
AT TURROTS, ATSOFT E-GOVERNMENT, DAN ASN
MEMAYU DI CV ATSOFT TEKNOLOGI**

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk
memperoleh gelar Sarjana Komputer dari Program Studi Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 10 Januari 2022

Tim Penguji

Ketua Penguji

Rahadian Kurniawan, S.Kom., M.Kom.

Anggota 1

Kholid Haryono, S.T., M.Kom.

Anggota 2

Andhika Giri Persada, S.Kom., M.Eng.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

Yang bertanda tangan di bawah ini:

Nama : Agung Wibowo
NIM : 18523230

Tugas akhir dengan judul:

**PENGEMBANGAN *FRONTEND* APLIKASI *MOBILE* SIM-RS
AT TURROTS, ATSOFT E-GOVERNMENT, DAN ASN
MEMAYU DI CV ATSOFT TEKNOLOGI**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 25 Desember 2021



(Agung Wibowo)

HALAMAN PERSEMBAHAN

Assalamu'alaikum Warahmatullahi Wabarakatuh. Alhamdulillah, puji syukur atas kehadiran Allah Subhanahu Wa Ta'ala Yang Maha Pengasih lagi Maha Penyayang serta tidak lupa shalawat dan salam senantiasa tercurah kepada junjungan kita Nabi Muhammad Shallallahu 'Alaihi Wasallam. Sehingga, saya dapat menyelesaikan laporan tugas akhir ini tepat pada waktunya. Laporan ini saya persembahkan kepada orang tua saya karena telah mendoakan, mendukung, menyemangati, dan memotivasi saya. Saya juga mempersembahkan laporan ini kepada teman-teman saya yang turut menyemangati saya dalam menyelesaikan laporan tugas akhir ini. *Wassalamualaikum Warahmatullahi Wabarakatuh.*



HALAMAN MOTO

“Ilmu adalah yang memberikan manfaat, bukan yang sekadar hanya dihafal.”

Imam Syafi'i

“Dirimu sendirilah yang dapat mengubah masa depan kamu”

Agung Wibowo



KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur atas kehadiran Allah Subhanahu Wa Ta'ala yang telah memberikan nikmatnya sehingga dapat menyelesaikan laporan tugas akhir yang berjudul "PENGEMBANGAN FRONTEND APLIKASI *MOBILE* SIM-RS AT TURROTS, ATSOFT E-GOVERNMENT, DAN ASN MEMAYU DI CV ATSOFT TEKNOLOGI". Tidak lupa shalawat dan salam senantiasa tercurah kepada junjungan kita Nabi Muhammad Shallallahu 'Alaihi Wasallam, yang membawa dan menerangi hati nurani kita, menjadi cahaya bagi segala perbuatan mulia. Adapun Laporan ini dibuat untuk memenuhi persyaratan tugas akhir jalur magang di Fakultas Teknologi Industri jurusan Informatika Universitas Islam Indonesia. Penulis berterima kasih kepada banyak pihak karena dalam pembuatan laporan ini mendapatkan bantuan, bimbingan, dan dorongan. Pihak-pihak tersebut diantaranya sebagai berikut:

1. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Rahadian Kurniawan, S.Kom., M.Kom., selaku Dosen Pembimbing laporan tugas akhir yang bersedia meluangkan waktunya dan bersabar untuk membimbing dalam penyusunan laporan ini.
3. CV. Atsoft Teknologi, beserta seluruh pimpinan, staf, dan karyawan yang telah memberikan kesempatan magang, dan membantu dalam menyelesaikan laporan akhir ini.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 25 Desember 2021



(Agung Wibowo)

SARI

CV. Atsoft Teknologi (Atsoft) adalah perusahaan yang bergerak dalam bidang teknologi informasi (IT). Terdapat 3 proyek yang masih membutuhkan tenaga kerja untuk menyelesaikan proyek tersebut. Proyek tersebut adalah Sistem Informasi Rumah Sakit At turrots (SIM-RS At turrots), Atsoft E-government (ATGOV), dan ASN Memayu. Proyek SIM-RS At turrot merupakan proyek dari Rumah Sakit At turrots. Permasalahan yang dialami oleh rumah sakit At turrots adalah proses pencatatan administrasinya masih manual. Rumah sakit merasa kesulitan karena mengurus data pasien yang sangat banyak dan sudah sering mendapatkan kesalahan dalam cara menghitung atau mengatur data-data pasien. Proyek kedua adalah proyek ATGOV. Proyek ini merupakan kelanjutan dari proyek SENGGUH (Sistem Evaluasi Pertanggungjawaban Pembangunan Daerah) yang sudah dibuat oleh CV. Atsoft Teknologi. Proyek SENGGUH merupakan proyek yang dapat memudahkan pencatatan dan perencanaan pembangunan daerah di Yogyakarta agar mencapai sasaran sesuai target Badan Perencanaan dan Pembangunan Daerah Daerah Istimewa Yogyakarta (BAPPEDA DIY). Sistem SENGGUH tersebut masih kaku dan *monolith* (*interface* langsung akses ke *database* serta tidak ada *Application Programming Interface* (API) atau *web service*-nya). Proyek ketiga adalah ASN Memayu. Proyek ini dikembangkan untuk meningkatkan pengelolaan manajemen kepegawaian di Badan Kepegawaian dan Diklat Daerah Provinsi Yogyakarta (BKD DIY).

Selama enam bulan magang di perusahaan CV. Atsoft Teknologi, telah dikembangkan 3 aplikasi *mobile* untuk membantu menyelesaikan permasalahan 3 proyek sebelumnya yang ada pada perusahaan tersebut, yaitu: SIM-RS At Turrots, Atsoft E-Government, dan ASN Memayu. Nama ketiga aplikasi tersebut dibuat mirip dengan nama proyeknya. Semua tugas yang diberikan oleh mentor ketika mengerjakan ketiga proyek tersebut telah dikerjakan dengan baik sehingga tujuan dan manfaat pada laporan ini terpenuhi. Tujuan pada laporan ini dapat dilihat pada **Bab 1.3** sedangkan manfaat pada laporan ini dapat dilihat pada **Bab 1.4**. Selanjutnya, hasil aplikasi *mobile* yang telah dikerjakan selama magang dapat dilihat pada **Bab 3.2.2**, **Bab 3.2.3**, dan **Bab 3.2.4**. Pada perusahaan ini, sistematika pembagian tugasnya berdasarkan fitur yang dikerjakan. Pemegang harus dapat menghasilkan satu fitur pada setiap pekannya. Terdapat hasil penilaian oleh mentor ketika mengerjakan ketiga proyek aplikasi *mobile* tersebut yang dapat memvalidasi bahwa tujuan dan manfaat pada laporan ini terpenuhi, dapat dilihat pada **Bab 4.2.2**.

Kata kunci: CV. Atsoft Teknologi, SIM-RS At Turrots, ATGOV, ASN Memayu.

GLOSARIUM

REST API	gaya arsitektur perangkat lunak untuk komunikasi antara <i>Front End</i> dengan <i>server</i>
<i>Front End Mobile</i>	Semua hal yang berhubungan dengan tampilan antarmuka pengguna dalam suatu aplikasi berbasis <i>mobile</i>
<i>Server</i>	sistem komputer yang memiliki layanan yaitu penyimpanan data
Command Palette	sebuah fitur dalam sistem operasi Window yang dapat diakses dengan menggunakan Keyboard Shortcut CTRL + Shift + P
SDK	sebuah alat untuk mengembangkan perangkat lunak
USB Debugging	suatu fitur untuk para pengembang Android
Mentor	seseorang yang membimbing selama magang berlangsung
AVD	konfigurasi perangkat yang dijalankan menggunakan emulator Android



DAFTAR ISI

Daftar Isi

HALAMAN JUDUL	1
HALAMAN PERSEMBAHAN	5
HALAMAN MOTO	6
KATA PENGANTAR	7
SARI	8
GLOSARIUM	9
DAFTAR ISI	10
DAFTAR TABEL	11
DAFTAR GAMBAR	12
BAB I PENDAHULUAN	15
1.1 Latar Belakang	15
1.2 Ruang Lingkup Magang	16
1.3 Tujuan	16
1.4 Manfaat	16
1.5 Sistematika Penulisan	16
BAB II DASAR TEORI	18
2.1. CV. Atsoft Teknologi	18
2.2 Front End Mobile Developer	21
2.3 Sistem Informasi Manajemen Rumah Sakit	21
2.4 Sistem Informasi e-Government	21
2.5 Sistem Informasi Manajemen Pegawai	22
2.6 Flutter	22
2.7 BLoC Pattern	24
BAB III PELAKSANAAN MAGANG	25
3.1 Aktivitas Magang	25
3.2 Hasil Pelaksanaan Magang	31
BAB IV REFLEKSI PELAKSANAAN MAGANG	72
4.1 Teknis	72
4.2 Non Teknis	80
BAB V KESIMPULAN DAN SARAN	83
5.1 Kesimpulan	83
5.2 Saran	83
DAFTAR PUSTAKA	85

DAFTAR TABEL

Tabel 3.1 Aktivitas Magang.....	23
Tabel 4.1 Tahap Instalasi pada Flutter dan Android Native	68



DAFTAR GAMBAR

Gambar 2.1 lokasi cabang pertama CV. Atsoft Teknologi	18
Gambar 2.2 lokasi cabang kedua CV. Atsoft Teknologi	19
Gambar 2.3 Struktur Organisasi CV. Atsoft Teknologi.....	19
Gambar 2.4 Metodologi Agile	20
Gambar 2.5. Siklus Hidup Stateful Widgets.....	23
Gambar 3.1 Tampilan framework Flutter pada Android Studio	31
Gambar 3.2 Tampilan Android Studio pada saat pertama kali memasang di komputer.....	32
Gambar 3.3 Tampilan Visual Studio Code	33
Gambar 3.4 Penambahan value C:\src\flutter\bin pada Environment Variable	33
Gambar 3.5 Command Pallette pada Visual Studio Code.	34
Gambar 3.6 Command Pallette setelah menekan Flutter: New Project.....	34
Gambar 3.7 Command Pallette setelah menekan tombol Application.....	34
Gambar 3.8 framework Flutter pada Visual Studio Code.....	35
Gambar 3.9 Tombol New Flutter Project.....	35
Gambar 3.10 Flutter SDK Path.....	36
Gambar 3.11. Formulir proyek baru SIM-RS At Turrots	36
Gambar 3.12 USB Debugging pada real device Samsung J7	36
Gambar 3.13 Pemilihan device dan tombol debugging yang berwarna hijau pada Android Studio	37
Gambar 3.14 AVD	37
Gambar 3.15 AVD	37
Gambar 3.16 Struktur folder proyek Flutter	38
Gambar 3.17 widget ListView	39
Gambar 3.18 widget WilayahItemWidget	40
Gambar 3.19 Hasil tampilan kode widget ListView.....	40
Gambar 3.20 Pemanggilan data Tipe Wilayah menggunakan aplikasi Postman.....	41
Gambar 3.21 Model TipeWilayahModel.....	41
Gambar 3.22 fungsi getListTipeWilayah.....	42
Gambar 3.23 Pemanggilan data Kenaikan Gaji menggunakan aplikasi Postman.....	42
Gambar 3.24 Model KenaikanGajiDataModel	43
Gambar 3.25 Kelas Event KenaikanGajiBlocEvent	43
Gambar 3.26 Kelas state beserta turunannya	44
Gambar 3.27 Kelas Bloc KenaikanGajiBloc	45
Gambar 3.28 Pemakaian Bloc di kelas user interface.....	45
Gambar 3.29 Struktur folder dengan menggunakan design pattern BLoC	46

Gambar 3.30 Tampilan aplikasi Postman	47
.....	47
Gambar 3.31 Tampilan aplikasi website Github	48
Gambar 3.32 Tampilan aplikasi Discord	48
Gambar 3.33 Halaman login SIM-RS At Turrots	49
Gambar 3.34 Halaman Menu Utama SIM-RS At Turrots	50
Gambar 3.35 Menu-menu utama SIM-RS At Turrots pada fitur SideBar	50
Gambar 3.36 Halaman Layanan SIM-RS At Turrots	52
Gambar 3.37 Halaman menu Kunjungan Baru SIM-RS At Turrots	52
Gambar 3.38 Halaman formulir mendaftarkan pasien baru SIM-RS At Turrots	52
Gambar 3.39 Halaman menu Daftar Kunjungan SIM-RS At Turrots	53
Gambar 3.40 Halaman splashscreen ATGOV	54
Gambar 3.41 Halaman Login ATGOV	54
Gambar 3.42 Halaman Menu Utama ATGOV	55
Gambar 3.43 Halaman Profil ATGOV	56
Gambar 3.44 Halaman tempat mengganti password pengguna aplikasi ATGOV	56
Gambar 3.45 Halaman Menu Utama ATGOV	57
Gambar 3.46 Laporan Dokumen Pelaksanaan Anggaran	57
Gambar 3.47 Halaman Wilayah	58
Gambar 3.48 Halaman Urusan	58
Gambar 3.49 Halaman Program	58
Gambar 3.50 Halaman Kegiatan	58
Gambar 3.51 Halaman Sub Kegiatan	58
Gambar 3.52 Halaman Edit Kegiatan	59
Gambar 3.53. Pop up Halaman Sub Kegiatan	59
Gambar 3.54 Halaman Tambah Program	59
Gambar 3.55 Halaman Konfigurasi	60
Gambar 3.56 Halaman Grup Menu	61
Gambar 3.57 Halaman Menu	61
Gambar 3.58 Halaman Edit Grup Menu	61
Gambar 3.59 Pop up Halaman Grup Menu	61
Gambar 3.60 Halaman Tambah Grup Menu	62
Gambar 3.61 splash screen	63
Gambar 3.62 Halaman Login	63
Gambar 3.63 Halaman Menu Utama ASN Memayu	64
Gambar 3.64 Kirang Leres	65
Gambar 3.65 Tambah Koreksi	65

Gambar 3.66 Sampun Leres.....	65
Gambar 3.67 Halaman Ubah Koreksi.....	65
Gambar 3.68 Pop up hapus Koreksi.....	63
Gambar 3.69 Halaman Profil ASN.....	66
Gambar 3.70 Halaman Data Pribadi.....	67
Gambar 3.71 Halaman Riwayat SKP.....	67
Gambar 3.72 Halaman Riwayat Keluarga.....	67
Gambar 3.73 Halaman Riwayat Prestasi.....	67
Gambar 3.74 Halaman Riwayat Hukuman.....	67
Gambar 3.75 Halaman Riwayat Pendidikan.....	67
Gambar 3.76 Halaman Riwayat Jabatan.....	67
Gambar 3.77 Halaman Riwayat Pangkat.....	67
Gambar 3.78 Halaman Pepeling.....	68
Gambar 3.79 Halaman Pensiun.....	69
Gambar 3.80 Halaman Kapan Pensiun.....	69
Gambar 3.81 Halaman Proses Pensiun.....	69
Gambar 3.82 Notifikasi Pensiun.....	69
Gambar 3.83 Halaman Kenaikan Gaji Berkala.....	70
Gambar 4.1 Penggunaan memory pada Android Studio dan Visual Studio Code.....	72
Gambar 4.2 Android Native dan Gambar 4.3 Flutter.....	73
Gambar 4.4 Widget ListView.....	73
Gambar 4.5 Komponen RecyclerView.....	74
Gambar 4.6 Adapter.....	74
Gambar 4.7 Pemanggilan data Tipe Wilayah dari REST API menggunakan Flutter.....	75
Gambar 4.8 Pemanggilan data Tipe Wilayah dari REST API menggunakan Android Native.....	75
Gambar 4.9 Struktur folder dengan konsep BLoC Pattern.....	76
Gambar 4.10 Dokumentasi REST API.....	77
Gambar 4.11. Tampilan web SIM-RS At Turrots.....	78
Gambar 4.12 Tampilan mobile e-Prima.....	78
Gambar 4.13 widgets yang ada pada halaman Kenaikan Gaji Berkala.....	79

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi pada zaman sekarang sangatlah cepat dan pesat. Hal ini membuat manusia dapat menggunakan berbagai macam teknologi untuk membantu segala bentuk aktivitas sehari-harinya. Salah satu teknologinya adalah menggunakan teknologi *smartphone*. *Smartphone* dapat digunakan sebagai teknologi yang membantu manusia dalam bidang pelayanan kesehatan. Dalam bidang pelayanan kesehatan, *smartphone* dapat digunakan sebagai media yang memberikan informasi untuk mendukung pelayanan manajemen rumah sakit.

Sebuah perusahaan yang bernama CV. Atsoft Teknologi sedang mengerjakan 10 proyek, dimana 7 proyek diantaranya sedang diselesaikan oleh beberapa orang *programmer*. Akan tetapi, terdapat 3 proyek lainnya yang masih membutuhkan tenaga kerja untuk menyelesaikan proyek tersebut. Proyek tersebut adalah Sistem Informasi Rumah Sakit At turrots (SIM-RS At turrots), Atsoft E-government, dan ASN Memayu. Proyek SIM-RS At turrot merupakan proyek dari Rumah Sakit At turrots. Permasalahan yang dialami oleh rumah sakit At turrots adalah proses pencatatan administrasinya masih manual. Rumah sakit merasa kesulitan karena mengurus data pasien yang sangat banyak dan sudah sering mendapatkan kesalahan dalam cara menghitung atau mengatur data-data pasien. Akibatnya, rumah sakit menanggung banyak kerugian. Proyek SIM-RS At turrots dikembangkan untuk membantu menangani permasalahan seputar administrasi di rumah sakit At turrots. Proyek SIM-RS At turrots dikembangkan dalam platform web dan mobile. Proyek selanjutnya adalah proyek Atsof E-Government (ATGOV). Proyek ini merupakan kelanjutan dari proyek SENGGUH (Sistem Evaluasi Pertanggungjawaban Pembangunan Daerah) yang sudah dibuat oleh CV. Atsoft Teknologi. Proyek SENGGUH merupakan proyek yang dapat memudahkan pencatatan dan perencanaan pembangunan daerah di Yogyakarta agar mencapai sasaran sesuai target Badan Perencanaan dan Pembangunan Daerah Daerah Istimewa Yogyakarta (BAPPEDA DIY). Sistem SENGGUH tersebut masih kaku dan *monolith* (*interface* langsung akses ke *database* serta tidak ada *Application Programming Interface* (API) atau web *service*-nya). Proyek ATGOV dibangun dalam platform web dan *mobile*. Proyek tersebut berguna untuk melanjutkan proyek baru yang bernama ATGOV-ANJAB di perusahaan CV Atsoft Teknologi. Proyek terakhir yang dikerjakan adalah ASN Memayu. Proyek ini dikembangkan untuk meningkatkan pengelolaan manajemen kepegawaian di Badan Kepegawaian dan Diklat Daerah Provinsi Yogyakarta (BKD DIY). Aplikasi ASN Memayu dibangun dalam platform web dan

mobile.

1.2 Ruang Lingkup Magang

Magang di CV. Atsoft Teknologi dilaksanakan selama 6 bulan dari awal Maret 2021 sampai dengan awal September 2021 yang diposisikan sebagai *Frontend Mobile Developer*. Selama magang berlangsung terdapat 3 proyek yang dikerjakan, yaitu: Sistem Informasi Rumah Sakit At turrots (SIM-RS At Turrots), Atsoft E-government (ATGOV), dan ASN Memayu.

1.3 Tujuan

Tujuan dari program magang di perusahaan CV. Atsoft Teknologi adalah sebagai berikut:

- a. Menghasilkan aplikasi *mobile* SIM-RS At turrots yang dapat memudahkan pihak Rumah Sakit At turrots dalam mengolah data pasien dengan lebih cepat dibandingkan menggunakan aplikasi webnya.
- b. Menghasilkan aplikasi *mobile* ATGOV yang dapat mempercepat pengolahan datanya dibandingkan menggunakan aplikasi webnya.
- c. Menghasilkan aplikasi *mobile* ASN Memayu. Aplikasi ini dapat menambahkan fitur yang lebih khusus dibandingkan pada saat menggunakan aplikasi webnya.

1.4 Manfaat

Manfaat dari pembuatan ketiga aplikasi tersebut adalah sebagai berikut:

- a. Pada aplikasi *mobile* SIM-RS At turrots dapat memudahkan pihak administrasi rumah sakit pada saat melakukan pendaftaran pasien yang sedang dalam kondisi darurat serta memberikan fleksibilitas dalam pengecekan data berulang.
- b. Pada aplikasi *mobile* ATGOV dapat memberikan fleksibilitas waktu pengecekan data berulang.
- c. Pada aplikasi *mobile* ASN Memayu dapat memudahkan para pegawai negeri sipil dalam mengakses sistem kepegawaiannya serta memberikan fleksibilitas waktu pengecekan data berulang.

1.5 Sistematika Penulisan

Sistematika Penulisan ini menggambarkan keseluruhan setiap materi yang ada pada laporan ini. Adapun Sistematika Penulisannya adalah sebagai berikut:

a. BAB I : PENDAHULUAN

Bab ini menggambarkan latar belakang, ruang lingkup magang, tujuan, manfaat dan sistematika penulisan.

b. BAB II : DASAR TEORI

Bab ini berupa pengertian dan definisi yang diambil dari kutipan buku dan media internet yang berkaitan dengan penyusunan laporan.

c. BAB III : PELAKSANAAN MAGANG

Bab ini menggambarkan dokumentasi dari kegiatan dan tugas yang dilaksanakan selama menjalani proses magang.

d. BAB IV : REFLEKSI PELAKSANAAN MAGANG

Bab ini menjelaskan hasil refleksi yang diperoleh selama mengembangkan aplikasi *mobile* SIM-RS At Turrots, ATGOV, dan ASN MEMAYU.

e. BAB V : KESIMPULAN DAN SARAN

Bab ini berupa kesimpulan dan saran dari seluruh materi yang sudah dijelaskan pada bab-bab sebelumnya.



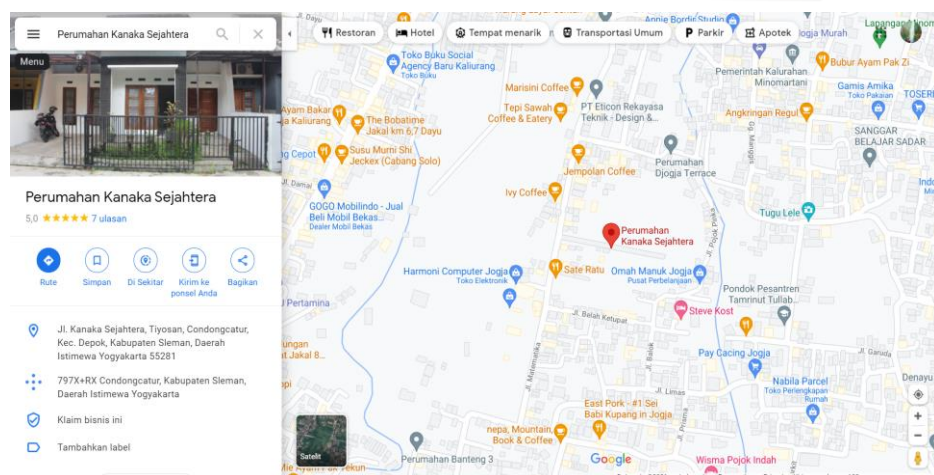
BAB II

DASAR TEORI

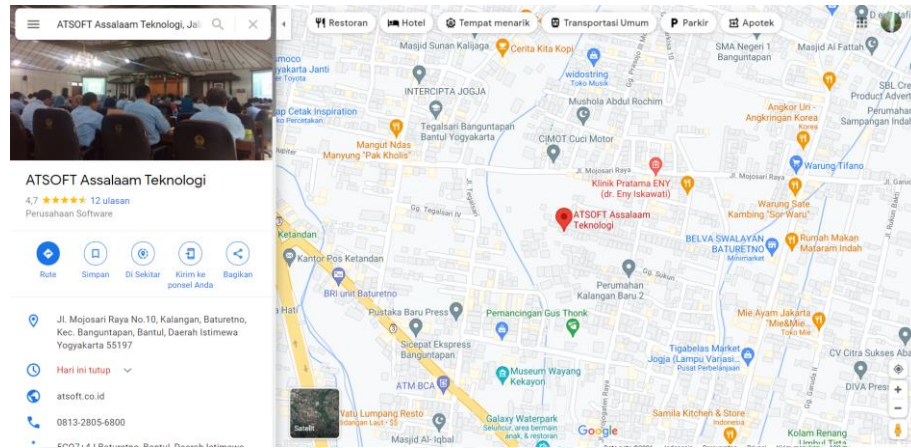
2.1. CV. Atsoft Teknologi

CV. Atsoft Teknologi (Atsoft) adalah perusahaan yang bergerak dalam bidang teknologi informasi. Perusahaan ini mengembangkan teknologi informasi pada berbagai bidang, seperti: pemerintahan, pendidikan, industri, dan kesehatan. Perusahaan Atsoft telah berdiri sejak pertengahan tahun 2006 dari beberapa Sarjana Informatika yang melakukan riset dan eksplorasi sains dan teknologi dengan konsep otomasi kerja dan integritas jaringan kinerja perusahaan. Perusahaan ini memberikan solusi otomasi proses - proses bisnis dari manual menjadi *computerized* dan mengedepankan efektifitas serta efisiensi kinerja sumber daya perusahaan karena fokus pada konsumen sehingga dapat terwujud *long term innovation*. Perusahaan ini bertekad untuk mengembangkan teknologi informasi pada berbagai bidang, seperti: pendidikan, pemerintahan, industri, dan kesehatan.

Perusahaan Atsoft memiliki 8 karyawan dimana 5 diantaranya bekerja sebagai *programmer* dan terdapat 3 karyawan lainnya yang bekerja sebagai pembicara maupun administrator. Setiap bulannya, perusahaan ini dapat mengerjakan hingga 10 proyek secara bersamaan. Terdapat dua lokasi kantor CV. Atsoft Teknologi di Yogyakarta. Lokasi kantor CV. Atsoft Teknologi yang pertama dapat dilihat pada **Gambar 2.1** yang beralamat di Perumahan Kanaka Sejahtera No.2-j, Tiyasan, Condongcatur, Depok, Sleman, Daerah Istimewa Yogyakarta. Alamat kantor CV. Atsoft Teknologi yang kedua dapat dilihat pada **Gambar 2.2** yang beralamat di Jl. Mojosari Raya No.10, Kalangan, Baturetno, Kec, Banguntapan, Bantul, Daerah Istimewa Yogyakarta. 55197.

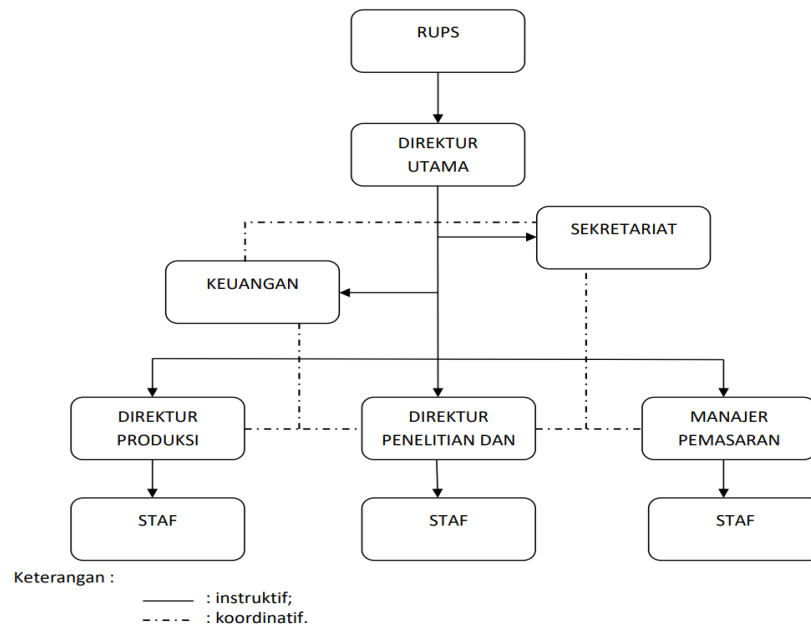


Gambar 2.1 lokasi cabang pertama CV. Atsoft Teknologi



Gambar 2.2 lokasi cabang kedua CV. Atsoft Teknologi

Struktur organisasi CV. Atsoft Teknologi terdiri dari beberapa bagian yang dapat dilihat di **Gambar 2.3**. Terdapat Rapat Umum Pemegang Saham (RUPS) yang merupakan pimpinan tertinggi pada struktur organisasi. Di bawah Rapat Umum Pemegang Saham (RUPS) terdapat Direktur Utama yang membawa lima bidang utama, yaitu: Keuangan, Sekretariat, Direktur Produksi, Direktur Penelitian, dan Manajer Pemasaran.



Gambar 2.3 Struktur Organisasi CV. Atsoft Teknologi.

CV. Atsoft Teknologi memiliki visi, yaitu: “Solusi Teknologi Informasi yang berkualitas dan mempunyai daya saing tinggi.”. Misi, yaitu: “Memberikan solusi Teknologi Informasi yang berkualitas di berbagai bidang untuk produktivitas dengan efektif dan efisien.”. CV. Atsoft Teknologi memiliki nilai sebagai berikut:

- a. *Clean*. Dikelola secara profesional, menjunjung tinggi kepercayaan dan integritas, menolak kecurangan dan sejenisnya, transparan, dan akuntabel.
- b. *Capable*. Dikelola oleh Sumber Daya Manusia yang profesional, mempunyai talenta dan kemampuan teknis tinggi, berkomitmen pada kualitas dengan penuh tanggung jawab.
- c. *Customer Focused*. Berorientasi pada kepentingan konsumen, berkomitmen dan mencurahkan segala daya untuk memberikan hanya yang terbaik kepada konsumen.
- d. *Competitive*. Berani dan mampu berkompetisi secara sehat, mengembangkan budaya menghargai kejujuran, kerja keras dan kinerja.

Perusahaan Atsoft ini menggunakan metodologi Agile untuk memudahkan dan mempercepat dalam pengembangan *software*. Menurut Ajoku pada tahun 2021, metodologi Agile digunakan untuk memecah seluruh proyek menjadi bagian-bagian yang lebih kecil. Langkah-langkah pengerjaan yang menjadi persyaratan pada metodologi ini adalah *brainstorm*, *design*, *development*, *quality assurance*, dan *deployment*. Pada **Gambar 2.4** terdapat langkah-langkah pengerjaan metodologi Agile yang dapat dikerjakan ulang hingga semua tim developer dan klien sepakat untuk mengakhiri pengerjaan proyeknya.



Gambar 2.4 Metodologi Agile (Sumber: K. K. Ajoku et al. / Journal of Advances in Computing, Communications and Information Technology 2 (2021) 1 – 13)

Perusahaan Atsoft menggunakan Phalcon sebagai *framework* yang dapat membantu untuk mengomunikasikan *database* informasi kepada *browser*. Selain itu, Phalcon dijadikan sebagai *framework* yang mengimplementasikan desain yang diberikan dan disarankan oleh perusahaan tersebut menjadi tampilan berbasis web. Terakhir, Atsoft juga menggunakan Flutter sebagai *framework* yang mengimplementasikan desain yang diberikan dan disarankan oleh perusahaan tersebut menjadi tampilan berbasis *mobile*.

2.2 Front End Mobile Developer

Front End Mobile Developer adalah bidang kerja yang mengembangkan aplikasi mobile sehingga pengguna dapat melihat dan berinteraksi dengan aplikasi tersebut. Front End Mobile Developer harus dapat membuat perubahan pada tampilan aplikasi *mobile* dan perlu terus-menerus melakukan perkembangan menyesuaikan dengan keadaan dilapangan (Cody Lindley, 2019). Dalam pekerjaan ini biasanya pengembang diharuskan untuk mengimplementasikan tampilan desain ke dalam aplikasi *mobile*. Selain itu, pekerjaan ini juga harus dapat memanfaatkan basis data dari server yang akan menjadi konten pada aplikasi *mobile*.

2.3 Sistem Informasi Manajemen Rumah Sakit

Sistem Informasi Manajemen Rumah Sakit (SIMRS) adalah kumpulan dari sub sistem yang saling berhubungan dan bekerja untuk mencapai suatu tujuan, yaitu: mengolah data menjadi informasi yang dibutuhkan untuk mendukung pelayanan manajemen rumah sakit (Muhyarsyah, 2007 dalam DIEN APRILIA SALIHA, 2018). SIMRS merupakan sebuah sistem yang dapat mempunyai nilai tambah dan sangat memiliki peran penting bagi rumah sakit agar dapat memberikan pelayanan terbaik. Dalam membangun SIMRS, rumah sakit perlu mempertimbangkan dari berbagai sudut pandang. Terdapat banyak faktor yang perlu dipertimbangkan (Handiwidjojo, 2015 dalam DIEN APRILIA SALIHA, 2018), diantaranya:

1. Sistem diharapkan dapat memenuhi kebutuhan pasien untuk mendukung pelayanan yang cepat, nyaman dan berkualitas.
2. Sistem diharapkan dapat mengelola segala transaksi secara akurat, efisien, dan cepat sehingga dapat melaporkan pelayanan medik secara tepat waktu.
3. Pada pembuatan SIMRS, sistem analis sangat diperlukan sebagai penghubung antara pihak pengembang dan rumah sakit agar sistem dapat dibuat sesuai dengan kebutuhan rumah sakit.

2.4 Sistem Informasi e-Government

Sistem informasi merupakan kumpulan dari setiap unit sistem yang dikelola oleh orang, *hardware*, *software*, jaringan komputer, dan jaringan komunikasi data. Kumpulan tersebut akan diubah dan disebarkan menjadi sebuah informasi mengenai bentuk organisasi (T. Sutabri, 2012 dalam Ahmad Sobri, 2019). E-Government adalah pemakaian teknologi informasi oleh pemerintah untuk memberikan informasi dan pelayanan yang berkenaan dengan pemerintahan (S. Adhim dan A. Baraja, 2012 dalam Ahmad Sobri, 2019). Sehingga, Sistem Informasi e-

Government merupakan sebuah sistem teknologi informasi yang akan dijadikan sebuah informasi untuk meningkatkan pelayanan yang berkenaan dengan pemerintahan.

2.5 Sistem Informasi Manajemen Pegawai

Sistem Informasi Manajemen Pegawai adalah sistem yang dapat memberikan informasi mengenai data-data pegawai pada suatu instansi yang saling berinteraksi untuk mencapai suatu tujuan. Sistem Informasi Manajemen Pegawai dapat menangani dan mengelola suatu data kepegawaian, khususnya meliputi: pendataan pegawai, Badan Kepegawaian dan Diklat Daerah, proses perencanaan, dan formasi kepegawaian. Pada intinya, Sistem Informasi Manajemen merupakan suatu sistem kepegawaian yang dapat mempercepat proses pencatatan, pengelolaan data, dan mampu menyajikan informasi kepegawaian (Universitas Andalas, 2016).

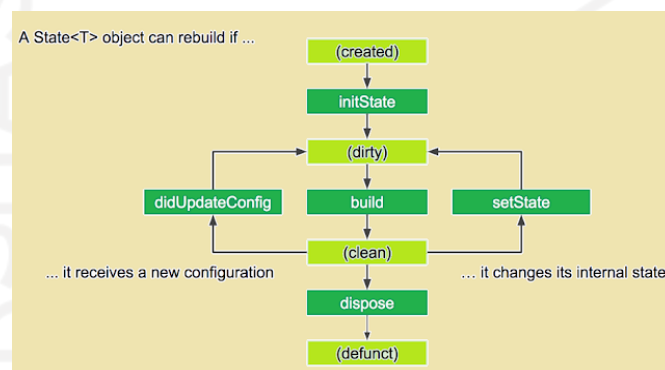
2.6 Flutter

Flutter telah diresmikan sejak tahun 2015 oleh developer Dart yaitu Sky. Eric Seidel (Direktur untuk Flutter di Google). Flutter adalah *cross-platform framework* yang dibuat oleh Google untuk membangun berbagai aplikasi seperti *android mobile*, *iOS mobile*, *web*, dan *desktop*. *Framework* ini menggunakan *widgets* untuk membuat UI dan Dart sebagai bahasa pemrograman untuk mengembangkan aplikasinya. *Widget* dapat diibaratkan dengan permainan Lego yang bisa menambahkan berbagai macam jenis bongkahan plastik kecil dan mengubah tampilan UI sesuai dengan yang diinginkan oleh developer. *Widget* adalah komponen UI yang membangun aplikasi Flutter. Setiap membuat UI dalam Flutter akan menciptakan berbagai macam pohon *widgets* (John Wiley, 2020). Berikut beberapa informasi mengenai *widget*:

1. *Widget* digunakan untuk membuat UI, seperti: baris, kolom, *stack*, *card*, *form*, dan *padding*.
2. Pada *Widget*, developer dapat melakukan *styling*, seperti: tipe *font*, ukuran, berat, warna, batas, dan bayangan.
3. *Widget* dapat berupa kumpulan bentuk dan formulir.

Dalam *framework* Flutter terdapat fitur yang bernama *hot-reload*. Dengan fitur ini, developer membangun *User Interface (UI)*, menambahkan fitur, dan memperbaiki bug dapat menjadi lebih cepat dan mudah. *Hot-reload* bekerja dengan cara menanamkan kode *file* baru ke dalam Dart Virtual Machine (VM). Setelah VM diperbaharui, kerangka kerja pada Flutter membuat ulang semua pohon *widget* secara otomatis. Hal ini membuat developer untuk melihat perubahan langsung pada UI aplikasi yang dikembangkannya (Flutter Dev, 2021).

Terdapat dua kombinasi *class* dalam mengembangkan aplikasi menggunakan *framework* Flutter, yaitu: *Stateless Widgets* dan *Stateful Widgets*. *Class* tersebut merupakan *widget* yang akan menggambarkan tampilan *User Interface* dengan membangun konstelasi *widget* lainnya agar menjadi lebih konkret. *Stateless Widget* hanya dapat menggambar satu kali saat *widget* dimuat sehingga tidak dapat menggunakan fitur *hot-reload* karena state tersebut sifatnya statis. Sebaliknya, *Statefull Widgets* dapat menggunakan *hot-reload* karena sifatnya yang dinamis. Dalam *Statefull Widgets* terdapat siklus hidup yang penting untuk diketahui oleh developer Flutter, dapat dilihat pada **Gambar 2.1**. Pada gambar tersebut terdapat fungsi `setState` dan `didUpdateConfig` untuk melakukan perubahan tampilan antarmukanya secara langsung.



Gambar 2.5. Siklus Hidup *Stateful Widgets* (Sumber:

<https://www.developerlibs.com/2019/12/flutter-lifecycle-widgets.html>)

Pada **Gambar 2.5** terdapat fungsi yang bisa dieksekusi ketika menggunakan *Statefull Widgets* (Developer Libs, 2019).

1. *createState*: Saat developer membuat *Stateful Widgets*, *framework* Flutter akan menginstruksikan metode *createState*. Metode ini mengembalikan sebuah *instance* yang dapat dilihat pada gambar sebelumnya.
2. *initState*: metode yang pertama kali dipanggil ketika *Stateful Widgets* telah dibuat. Metode ini dipanggil hanya sekali. Developer dapat menginisialisasi data, properti, dan objek lainnya untuk mengubah data pada *widget* ini.
3. *build*: Metode ini menunjukkan bagian dari *User Interface* yang diwakili oleh *widget*.
4. *setState*: Metode ini dipanggil dari *framework* oleh developer. Developer dapat mengubah status internal objek dan membuat perubahan dalam fungsi yang akan diteruskan ke metode ini. Metode *setState* akan memberitahu kerangka kerja bahwa terjadi perubahan yang akan mempengaruhi *User Interface*.

5. *didUpdateWidget*: Jika terdapat induk *widget* untuk mengubah konfigurasi dan harus membangun kembali *widget* yang ada. *Widget* tersebut akan dibangun bersamaan *runtimeType*. Setelah tahap ini, metode *didUpdateWidget* akan dipanggil untuk memperbaharui properti *widget* yang telah ada menjadi *widget* baru.
6. *dispose*: Metode ini dipanggil ketika objek *state* akan dihapus secara permanen.

2.7 BLoC Pattern

Business Logic Component design pattern (BloC Pattern) adalah *design pattern* yang membantu memisahkan antara *User Interface* dengan *business logic* sehingga komponen pada proyek dapat dibagi menjadi *presentational componen*, *BloC*, dan *backend*. *Design Pattern* ini memperbolehkan developer untuk dapat fokus dalam mengkonversikan *event* menjadi *state*. Terdapat beberapa istilah yang penting jika developer melakukan penerapan *BloC* (Nurul Faza, 2019), yaitu:

1. *Event* merupakan *input* untuk *BLoC*.
2. *States* merupakan *output* dari *BloC*.
3. *Transitions* merupakan perubahan dari satu *state* ke *state* lain.
4. *Blocs* merupakan komponen yang akan mengonversi *Stream* dari *event* ke *Stream* yang akan mengubah *state*.

Hal ini menjadikan developer lain dapat dengan mudah memahami kode yang telah dikembangkan sebelumnya karena *design pattern* ini dibuat khusus oleh Flutter. Hal ini juga dapat membantu membuat aplikasi menjadi lebih kompleks karena *pattern* tersebut tersusun dari komponen-komponen yang lebih kecil dan mudah di tes.

BAB III PELAKSANAAN MAGANG

3.1 Aktivitas Magang

Kegiatan magang telah dilakukan selama enam bulan dimulai dari awal bulan Maret 2021 hingga akhir bulan Agustus 2021 sebagai *Front End Mobile Developer* di perusahaan CV Atsoft Teknologi. Terdapat tiga aktivitas yang telah dilalui selama kurun waktu tersebut. Aktivitas-aktivitas tersebut dapat dilihat pada **Tabel 3.1**. Adapun penjelasan terkait detail aktivitas magang pada **Tabel 3.1** dijelaskan pada sub bab berikutnya.

Table 3.1 Aktivitas Magang

No.	Aktivitas	Waktu Mulai	Durasi
1	Aktivitas di bulan Maret	1 Maret	1.5 Bulan
2	Aktivitas di bulan April, Mei, dan Juni	15 April	2 Bulan 7 hari
3	Aktivitas di bulan Juli dan Agustus	22 Juni	2 Bulan

3.1.1 Aktivitas di bulan Maret

Pada awal pertama masuk magang, *Project Manager* menjelaskan tentang lingkungan perusahaan mulai dari jam kerja untuk daring dan luring, *workflow* kerja dan beberapa *tools* yang akan dipakai selama kegiatan berlangsung. *Tools* tersebut sangat penting untuk mendukung kelancaran proses manajemen proyek. Beberapa *tool* yang perlu disiapkan selama magang, yaitu:

1. Android Studio / Visual Studio Code untuk penyedia semua utilitas yang diperlukan untuk membangun aplikasi *mobile*.
2. Memasang Flutter.
3. Postman untuk pengujian *Application Programming Interface* (API).
4. Github untuk penyimpanan dan pengecekan berkas *source code*.
5. Discord untuk komunikasi.

Project Manager memberikan rekomendasi untuk melakukan pembelajaran selama magang di salah satu akun Youtube yang bernama bapak Erico Darmawan Handoyo. Dalam akun tersebut banyak pembelajaran terkait pengembangan aplikasi *mobile* menggunakan Flutter.

Pembelajaran di akun tersebut akan terus berlanjut bersamaan dengan pengerjaan proyek yang diberikan oleh perusahaan hingga waktu kontrak magang berakhir.

Kegiatan berlanjut dengan memasuki proyek yang bernama Sistem Informasi Rumah Sakit At Turrots (SIM-RS At Turrots). Proyek SIM-RS At Turrots merupakan proyek berbasis web dan *mobile*. Proyek ini berfungsi sebagai media yang membantu proses pencatatan administrasi di Rumah Sakit At Turrots. Aktivitas magang yang dilakukan dalam proyek SIM-RS At Turrots berada pada divisi *mobile* lebih spesifik sebagai *Frontend Mobile Developer*. Proses pengembangan proyek SIM-RS At Turrots melibatkan peran lainnya dalam tim selain *Frontend Mobile Developer*, diantaranya: bapak Andrey Ferriyan sebagai *Project Manager*, mas Vandy sebagai *Backend Developer*, mba Reisza Novi sebagai *Frontend Web Developer*, dan bapak Sofyan sebagai *Lead Developer*.

Pada proyek SIM-RS At Turrots, mentor memberikan tugas untuk membuat aplikasi *mobile* berbasis Flutter. Nama proyek aplikasi tersebut adalah aplikasi *mobile* SIM-RS At Turrots. Pertama kali hal yang harus dilakukan ketika membuat aplikasi tersebut adalah melakukan instalasi proyek. Contoh tahap-tahap dan detail instalasi dapat dilihat pada **Bab 3.2.1.1**, yaitu: **Tahap Instalasi Flutter**. Sebelum melakukan instalasi, folder pada komputer perlu dihubungkan dengan aplikasi Github. Penjelasan Github dapat dilihat pada **Bab 3.2.1.3**. Dengan dihubungkannya hal tersebut, mentor dapat memonitoring kode yang sudah dibuat oleh pemegang. Setelah melakukan instalasi, mentor akan memberikan tugas dengan jangka waktu satu minggu per fiturnya. Setelah mengerjakan per fitur, mentor dan sebagian karyawan CV. Atsoft Teknologi akan melakukan pengecekan terhadap fitur yang sudah dibuat dengan cara mengunduh aplikasi SIM-RS At Turrots pada *smartphone*-nya. Jika fitur berjalan dengan baik, pemegang akan diberikan *task* baru oleh mentor. Sebaliknya, pemegang harus terus mengerjakan hingga fitur tersebut berjalan dengan baik. Pemegang yang sudah mengerjakan tugas dan mengalami kendala harus melaporkan kepada mentor melalui aplikasi Discord. Detail penjelasan aplikasi Discord dapat dilihat pada **Bab 3.2.1.4**. Selanjutnya, pemegang perlu membuka aplikasi Postman untuk melakukan pengujian pada EndPoint REST API yang diberikan oleh *Backend Developer*. Penjelasan Postman dapat dilihat pada **Bab 3.2.1.2**. Ketika melakukan pengujian, pemegang perlu berkomunikasi menggunakan aplikasi Discord dengan *Backend Developer*. *Backend Developer* akan memberi dan menjelaskan cara pemakaian EndPoint REST API pada aplikasi Postman. Setelah berhasil melakukan pengujian pada aplikasi Postman, pemegang atau developer dapat langsung memasukkan data-data tersebut ke dalam proyek aplikasi

menggunakan Flutter. Data-data tersebut berfungsi sebagai konten pada proyek aplikasi *mobile*-nya. Selanjutnya, mentor akan memberikan tugas per fitur. Ketika membuat fitur tersebut, developer perlu membuat kelas tampilan *User Interface*-nya. Dalam kelas tersebut developer akan mengimplementasikan desain yang diberikan dan disarankan oleh perusahaan CV. Atsoft Teknologi. Desain tersebut akan diimplementasikan menjadi kode *widget-widget* pada Flutter. Salah satu penjelasan dan penggunaan *widget* dapat dilihat pada **Bab 3.2.1.1**, yaitu: Penggunaan Widget ListView. Setelah mengimplementasikan desain, developer akan banyak melakukan pengiriman dan pemanggilan data ke *server* dengan menggunakan REST API. Cara pengiriman dan pemanggilan data dari REST API dapat dilihat pada **Bab 3.2.1.1**, yaitu: **Pemanggilan Data dari REST API**. Dengan komunikasi ke *server* menggunakan REST API tersebut, developer dapat membuat banyak fitur aplikasi, seperti: pencarian data, penambahan data, dan lain sebagainya.

Tugas pertama dalam membuat proyek aplikasi *mobile* SIM-RS At Turrots adalah membuat halaman Login untuk aplikasi *mobile*. Dalam halaman tersebut terdapat fitur *login*. Tugas dilanjutkan dengan membuat halaman Menu Utama yang memiliki sebuah fitur, yaitu *logout*. Tugas terakhir pada aktivitas ini adalah membuat halaman Layanan untuk aplikasi *mobile* SIM-RS At Turrots. Dalam halaman Layanan terdapat fitur tertentu, yaitu: pencarian, menampilkan data, dan menambah data. Halaman-halaman tersebut diselesaikan dalam kurung waktu 1 bulan 15 hari. Adapun penjelasan detail mengenai fitur tersebut yang ada pada halaman Login, halaman Menu Utama, dan Layanan dapat dilihat pada **Bab 3.2.2**. Proyek aplikasi *mobile* tersebut dihentikan sementara dan *source code file*-nya digunakan untuk keperluan proyek perusahaan di masa yang akan datang.

3.1.2 Aktivitas di bulan April, Mei, dan Juni

Proyek selanjutnya yang dikembangkan pada aktivitas ini adalah Atsoft E-Government (ATGOV). Proyek ini merupakan kelanjutan dari proyek SENGGUH (Sistem Evaluasi Pertanggungjawaban Pembangunan Daerah) yang sudah dibuat oleh CV. Atsoft Teknologi. Aktivitas magang yang dilakukan dalam proyek ATGOV berada pada divisi *mobile* lebih spesifik sebagai *FrontEnd Mobile Developer*. Proses pengembangan proyek ATGOV melibatkan peran lainnya dalam tim selain *FrontEnd Mobile Developer*, diantaranya: bapak Andrey Ferriyan sebagai *Project Manager*, mba Reisza Novi sebagai *Backend Developer*, mas Hafidh sebagai *Frontend Web Developer*, mas Fadhlán sebagai *Frontend Web Developer* dan bapak Sofyan sebagai *Lead Developer*.

Pada proyek ATGOV, mentor memberikan tugas untuk membuat aplikasi *mobile* berbasis Flutter. Nama proyek aplikasi tersebut adalah aplikasi *mobile* ATGOV. Pertama kali hal yang harus dilakukan ketika membuat aplikasi tersebut adalah melakukan instalasi proyek. Contoh tahap-tahap dan detail instalasi dapat dilihat pada **Bab 3.2.1.1**, yaitu: **Tahap Instalasi Flutter**. Sebelum melakukan instalasi, folder pada komputer perlu dihubungkan dengan aplikasi Github. Penjelasan Github dapat dilihat pada **Bab 3.2.1.3**. Dengan dihubungkannya hal tersebut, mentor dapat memonitoring kode yang sudah dibuat oleh pemegang. Setelah melakukan instalasi, mentor akan memberikan tugas dengan jangka waktu satu minggu per fiturnya. Setelah mengerjakan per fitur, mentor dan sebagian karyawan CV. Atsoft Teknologi akan melakukan pengecekan terhadap fitur yang sudah dibuat dengan cara mengunduh aplikasi Atsoft E-Government pada *smartphone*-nya. Jika fitur berjalan dengan baik, pemegang akan diberikan *task* baru oleh mentor. Sebaliknya, pemegang harus terus mengerjakan hingga fitur tersebut berjalan dengan baik. Pemegang yang sudah mengerjakan tugas dan mengalami kendala harus melaporkan kepada mentor melalui aplikasi Discord. Detail penjelasan aplikasi Discord dapat dilihat pada **Bab 3.2.1.4**. Selanjutnya, pemegang perlu membuka aplikasi Postman untuk melakukan pengujian pada EndPoint REST API yang diberikan oleh *Backend Developer*. Penjelasan Postman dapat dilihat pada **Bab 3.2.1.2**. Ketika melakukan pengujian, pemegang perlu berkomunikasi menggunakan aplikasi Discord dengan *Backend Developer*. *Backend Developer* akan memberi dan menjelaskan cara pemakaian EndPoint REST API pada aplikasi Postman. Setelah berhasil melakukan pengujian pada aplikasi Postman, pemegang atau developer dapat langsung memasukkan data-data tersebut ke dalam proyek aplikasi menggunakan Flutter. Data-data tersebut berfungsi sebagai konten pada proyek aplikasi *mobile*-nya. Selanjutnya, mentor akan memberikan tugas per fitur. Ketika membuat fitur tersebut, developer perlu membuat kelas tampilan *User Interface*-nya. Dalam kelas tersebut developer akan mengimplementasikan desain yang diberikan dan disarankan oleh perusahaan CV. Atsoft Teknologi. Desain tersebut akan diimplementasikan menjadi kode *widget-widget* pada Flutter. Salah satu penjelasan dan penggunaan *widget* dapat dilihat pada **Bab 3.2.1.1**, yaitu: Penggunaan Widget ListView. Setelah mengimplementasikan desain, developer akan banyak melakukan pengiriman dan pemanggilan data ke *server* dengan menggunakan REST API. Cara pengiriman dan pemanggilan data dari REST API dapat dilihat pada **Bab 3.2.1.1**, yaitu: **Pemanggilan Data dari REST API**. Dengan komunikasi ke *server* menggunakan REST API tersebut, developer dapat membuat banyak fitur aplikasi, seperti: pencarian data, penambahan data, dan lain sebagainya. Pada aktivitas ini juga, developer akan menggunakan *design pattern* BLoC.

Penjelasan *design pattern* tersebut dapat dilihat pada **Bab 2.6**. Dengan *design pattern* tersebut, developer lain yang akan memperbaharui proyek aplikasinya akan menjadi lebih mudah karena *design pattern* tersebut dibuat oleh profesional Flutter. *Design pattern* dapat dijadikan untuk memisahkan logika pemanggilan data dari *server*. Pemanggilan data menggunakan *design pattern* tersebut dapat dilihat pada **Bab 3.2.1.1**, yaitu **Pemanggilan Data dari REST API dengan Konsep BLoC Pattern**.

Terdapat 5 tugas yang telah dikerjakan dalam pembuatan aplikasi *mobile* ATGOV berbasis *mobile* ini, yaitu: halaman Login, halaman Menu Utama, halaman Profil, halaman Master Data, dan halaman Konfigurasi. Dalam halaman Login terdapat fitur *splash screen* dan *login*. Pada halaman Profil terdapat fitur mengubah data dan *logout*. Pada halaman Master Data dan Konfigurasi terdapat fitur yang mirip, yaitu: pencarian, menampilkan data, menambahkan data, mengubah data, dan menghapus data. Semua halaman dikerjakan dalam kurung waktu 2 bulan 7 hari. Adapun penjelasan detail mengenai fitur tersebut yang ada pada halaman Login, halaman Menu Utama, halaman Profil, halaman Master Data, dan halaman Konfigurasi, dapat dilihat pada **Bab 3.2.3**. Dalam beberapa halaman tersebut, developer perlu melakukan pemanggilan data dari REST API dengan konsep BLoC Pattern. Proyek aplikasi *mobile* tersebut dihentikan sementara dan *source code file*-nya digunakan untuk keperluan proyek perusahaan di masa yang akan datang.

3.1.3 Aktivitas di bulan Juli dan Agustus

Aktivitas pertama yang dilakukan pada bulan ini adalah mempelajari kode standar pengembangan *mobile* dengan menggunakan Flutter yang telah dibuat oleh perusahaan. Pada kode standar perusahaan masih belum menggunakan *design pattern* BLoC. Selanjutnya, aktivitas dilanjutkan dengan mengembangkan proyek ASN Memayu. Proyek ASN Memayu adalah proyek pengembangan pengembangan aplikasi berbasis web dan *mobile* yang berfungsi untuk meningkatkan pengelolaan manajemen kepegawaian di Badan Kepegawaian dan Diklat Daerah Provinsi Yogyakarta (BKDD Yogyakarta). Aktivitas magang yang dilakukan dalam proyek ASN Memayu berada pada divisi *mobile* lebih spesifik sebagai *FrontEnd Mobile Developer*. Proses pengembangan proyek ASN Memayu melibatkan peran lainnya dalam tim selain *FrontEnd Mobile Developer*, diantaranya: bapak Andrey Ferriyan sebagai *Project Manager*, mba Reisza Novi sebagai *Backend Developer*, mas Hafidh sebagai *Frontend Web Developer*, dan bapak Sofyan sebagai *Lead Developer*.

Pada proyek ASN Memayu, mentor memberikan tugas untuk membuat aplikasi *mobile* berbasis Flutter. Nama proyek aplikasi tersebut adalah aplikasi *mobile* ASN Memayu. Pertama kali hal yang harus dilakukan ketika membuat aplikasi tersebut adalah melakukan instalasi proyek. Contoh tahap-tahap dan detail instalasi dapat dilihat pada **Bab 3.2.1.1**, yaitu: **Tahap Instalasi Flutter**. Sebelum melakukan instalasi, folder pada komputer perlu dihubungkan dengan aplikasi Github. Penjelasan Github dapat dilihat pada **Bab 3.2.1.3**. Dengan dihubungkannya hal tersebut, mentor dapat memonitoring kode yang sudah dibuat oleh pemegang. Setelah melakukan instalasi, mentor akan memberikan tugas dengan jangka waktu satu minggu per fiturnya. Setelah mengerjakan per fitur, mentor dan sebagian karyawan CV. Atsoft Teknologi akan melakukan pengecekan terhadap fitur yang sudah dibuat dengan cara mengunduh aplikasi ASN Memayu pada *smartphone*-nya. Jika fitur berjalan dengan baik, pemegang akan diberikan *task* baru oleh mentor. Sebaliknya, pemegang harus terus mengerjakan hingga fitur tersebut berjalan dengan baik. Pemegang yang sudah mengerjakan tugas dan mengalami kendala harus melaporkan kepada mentor melalui aplikasi Discord. Detail penjelasan aplikasi Discord dapat dilihat pada **Bab 3.2.1.4**. Selanjutnya, pemegang perlu membuka aplikasi Postman untuk melakukan pengujian pada EndPoint REST API yang diberikan oleh *Backend Developer*. Penjelasan Postman dapat dilihat pada **Bab 3.2.1.2**. Ketika melakukan pengujian, pemegang perlu berkomunikasi menggunakan aplikasi Discord dengan *Backend Developer*. *Backend Developer* akan memberi *file* Excel yang berisi EndPoint-EndPoint REST API dan menjelaskan cara pemakaian EndPoint tersebut pada aplikasi Postman. Setelah berhasil melakukan pengujian pada aplikasi Postman, pemegang atau developer dapat langsung memasukkan data-data tersebut ke dalam proyek aplikasi menggunakan Flutter. Data-data tersebut berfungsi sebagai konten pada proyek aplikasi *mobile*-nya. Selanjutnya, mentor akan memberikan tugas per fitur. Ketika membuat fitur tersebut, developer perlu membuat kelas tampilan *User Interface*-nya. Dalam kelas tersebut developer akan mengimplementasikan desain yang diberikan dan disarankan oleh perusahaan CV. Atsoft Teknologi. Berbagai revisi desain telah dilakukan pada aplikasi ASN Memayu. Penyebab banyaknya revisi desain dapat dilihat pada **Bab 4.1.4** Desain tersebut akan diimplementasikan menjadi kode *widget-widget* pada Flutter. Salah satu penjelasan dan penggunaan *widget* dapat dilihat pada **Bab 3.2.1.1**, yaitu: Penggunaan Widget ListView. Setelah mengimplementasikan desain, developer akan banyak melakukan pengiriman dan pemanggilan data ke *server* dengan menggunakan REST API. Cara pengiriman dan pemanggilan data dari REST API dapat dilihat pada **Bab 3.2.1.1**, yaitu: **Pemanggilan Data dari REST API**. Dengan komunikasi ke *server* menggunakan REST API

tersebut, developer dapat membuat banyak fitur aplikasi, seperti: pencarian data, penambahan data, dan lain sebagainya. Pada aktivitas ini juga, developer akan menggunakan *design pattern* BLoC. Penjelasan *design pattern* tersebut dapat dilihat pada **Bab 2.6**. Dengan *design pattern* tersebut, developer lain yang akan memperbaharui proyek aplikasinya akan menjadi lebih mudah karena *design pattern* tersebut dibuat oleh profesional Flutter. *Design pattern* dapat dijadikan untuk memisahkan logika pemanggilan data dari *server*. Pemanggilan data menggunakan *design pattern* tersebut dapat dilihat pada **Bab 3.2.1.1**, yaitu **Pemanggilan Data dari REST API dengan Konsep BLoC Pattern**.

Terdapat 4 tugas yang telah dikerjakan dalam pembuatan aplikasi ASN Memayu berbasis *mobile* ini, yaitu: halaman Login, halaman Menu Utama, halaman Profil ASN, dan halaman Ngecek Data. Dalam halaman Login terdapat fitur *splash screen* dan *login*. Pada halaman Profil ASN terdapat fitur yang menampilkan data sedangkan pada halaman Ngecek Data terdapat beberapa fitur, seperti: mencari data, menampilkan data, mengubah data, dan mengedit data. Adapun penjelasan detail mengenai fitur tersebut yang ada pada halaman Login, halaman Menu Utama, halaman Profil ASN, dan halaman Ngecek Data dapat dilihat pada **Bab 3.2.4**. Dalam beberapa halaman tersebut, developer perlu melakukan pemanggilan data dari REST API dengan konsep BLoC Pattern. Proyek ASN Memayu hingga bulan Desember 2021 ini masih dikerjakan karena akan segera dipakai oleh pihak BKD Yogyakarta.

3.2 Hasil Pelaksanaan Magang

Selama kegiatan magang berlangsung, terdapat tiga proyek aplikasi berbasis *mobile* yang ikut dikembangkan, yaitu: aplikasi Sistem Informasi Rumah Sakit At Turrots (SIM-RS At Turrots), aplikasi ATSOFT E-Government (ATGOV), dan aplikasi ASN Memayu. Teknologi yang digunakan pada saat kegiatan magang dijelaskan pada **Bab 3.2.1**. Adapun produk yang dihasilkan selama proses magang dijelaskan pada **Bab 3.2.2**, **3.2.3**, dan **3.2.4**.

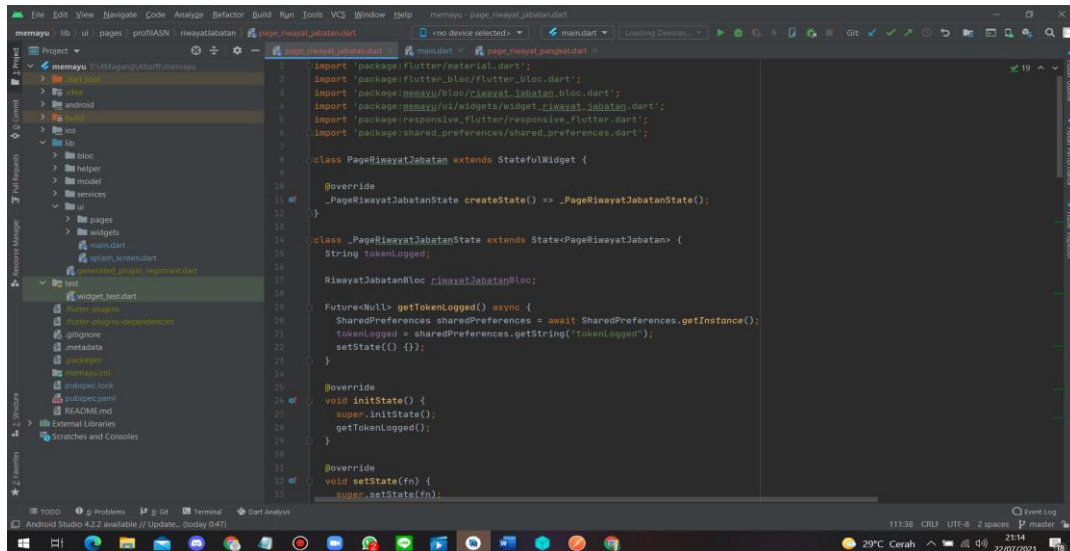
3.2.1 Teknologi Yang Digunakan

Selama kegiatan magang berlangsung, terdapat empat teknologi yang digunakan untuk mengembangkan aplikasi *mobile*, yaitu: Flutter, Postman, Discord, dan Github. Semua tiga proyek pada saat magang menggunakan teknologi tersebut. Detail setiap teknologi akan dijelaskan pada sub bab ini.

3.2.1.1. Flutter

Flutter adalah *cross-platform framework* yang dibuat oleh Google untuk membangun berbagai aplikasi seperti *android mobile*, *iOS mobile*, *web*, dan *desktop*. *Framework* Flutter yang

ditampilkan pada *Integrated Development Environment (IDE)* Android Studio dapat lihat pada **Gambar 3.1**. Flutter digunakan sebagai *framework* untuk membangun aplikasi *mobile* pada proyek SIM-RS At Turrots, ATGOV, dan ASN Memayu.

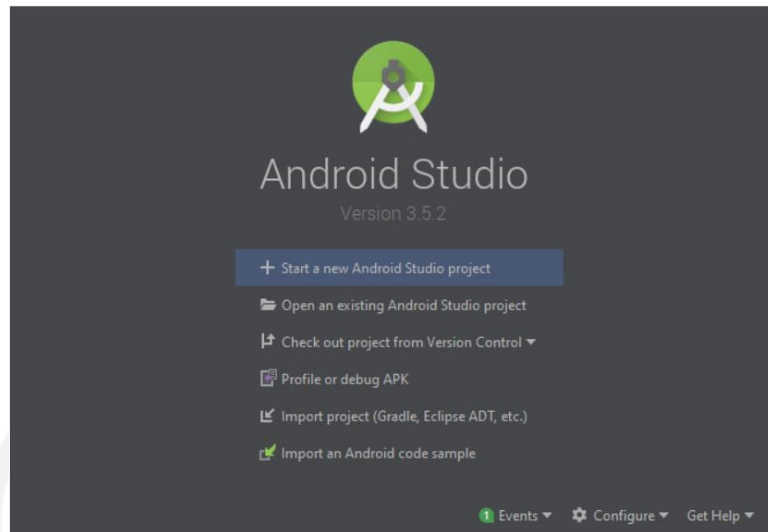


Gambar 3.1 Tampilan *framework* Flutter pada Android Studio.

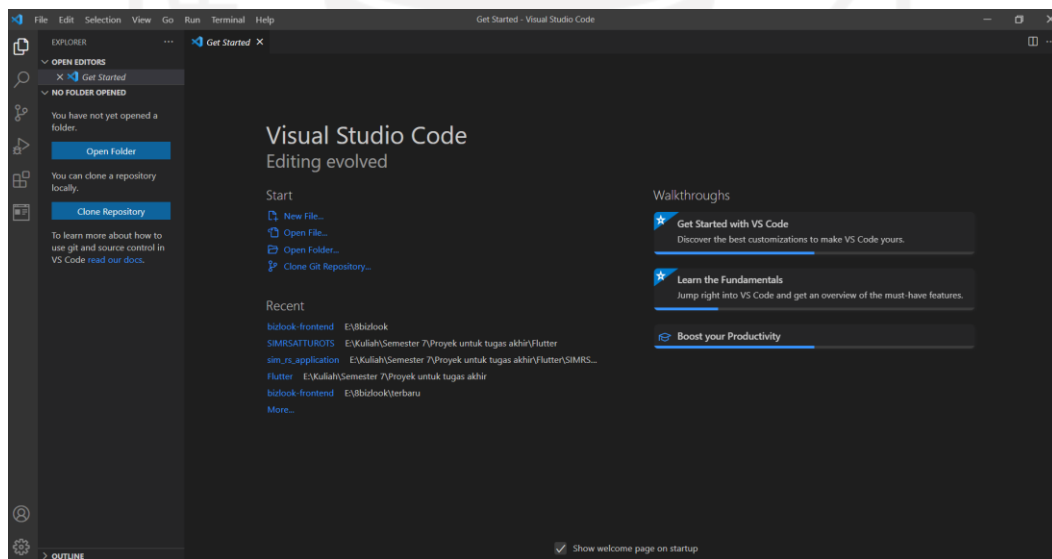
Pada sub bab ini juga akan dijelaskan mengenai Tahap Instalasi Flutter, Struktur Folder Proyek Flutter, Penggunaan *widget* ListView, Contoh Pemanggilan Data dari REST API, dan Contoh Pemanggilan Data dari REST API dengan Konsep BLoC Pattern.

Tahap Instalasi Flutter

Pertama, developer perlu memasang Java Development Kit (JDK) pada komputer. JDK merupakan sebuah perangkat lunak yang digunakan untuk memproses kode program Java dan Kotlin menjadi *bytecode*. Developer dapat mengunduh JDK pada web <https://www.oracle.com/java/technologies/downloads/> dan mengikuti *step* instalasi hingga akhir yang sudah disediakan pada saat proses instalasinya. **Kedua**, developer perlu memasang Android Studio. Android Studio dapat diunduh pada web <https://developer.android.com/studio?hl=id/>. Android Studio adalah *Integrated Development Environment (IDE)* untuk mengembangkan aplikasi android. IDE ini dapat digunakan secara gratis dan dapat menjadi *tools* tercepat yang digunakan untuk membangun aplikasi android. Dalam Android Studio terdapat *plugins* untuk membangun aplikasi android menggunakan *framework* Flutter. Dengan *plugins* tersebut, pengembangan aplikasi android dapat menjadi lebih cepat dan memungkinkan untuk memperbaiki permasalahan kode tertentu secara otomatis. Developer dapat mengikuti *step* instalasi yang sudah disediakan pada saat proses instalasinya. Tampilan Android Studio yang sudah dipasang dapat dilihat pada **Gambar 3.2**.

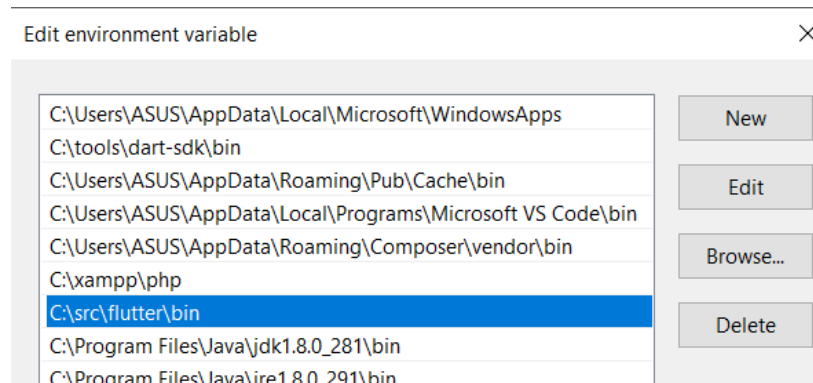


Gambar 3.2 Tampilan Android Studio pada saat pertama kali memasang di komputer **Ketiga**, developer perlu memasang VSCode sebagai aplikasi pengganti IDE Android Studio ketika komputer mengalami kendala pada saat membuka Android Studio. Developer dapat mengunduh VSCode pada web <https://code.visualstudio.com/download>. Tampilan VSCode yang sudah dipasang dapat dilihat pada **Gambar 3.3**.



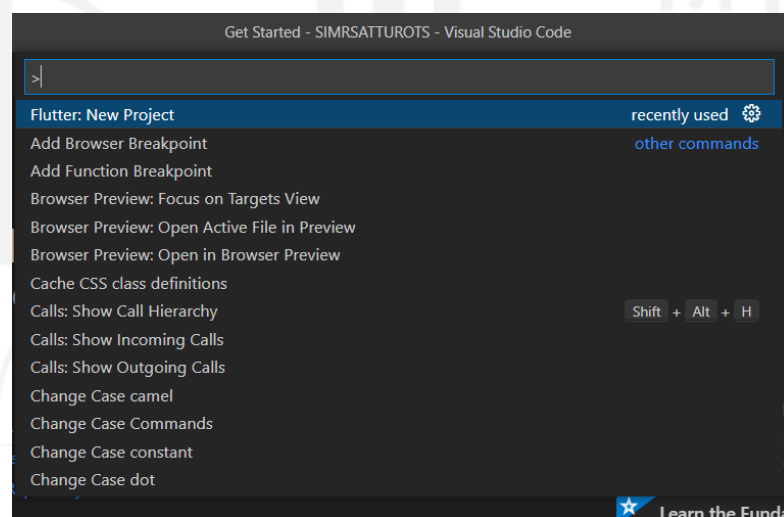
Gambar 3.3 Tampilan Visual Studio Code

Keempat, developer perlu mengunduh Flutter SDK yang ada pada *website* Docs Flutter yang terdapat pada web <https://docs.flutter.dev/get-started/install/windows>. Setelah mengunduh, developer dapat menyimpan pada folder C:\src. Jika folder tersebut belum ada, developer dapat membuat terlebih dahulu lalu *extract file* yang telah diunduh sebelumnya pada folder tersebut. Setelah itu, developer harus menambahkan *value* C:\src\flutter\bin di Environment Variable pada bagan *variable* yang bernama Path yang dapat dilihat pada **Gambar 3.4**.



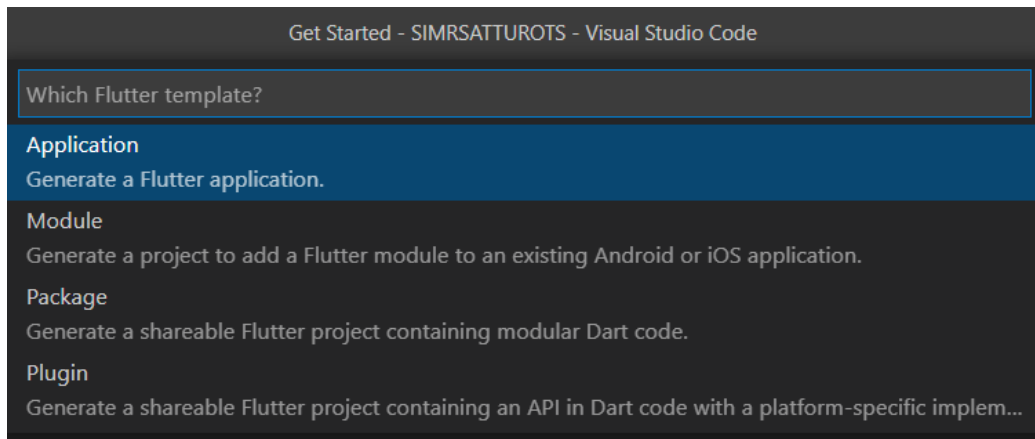
Gambar 3.4 Penambahan value `C:\src\flutter\bin` pada Environment Variable

Kelima, developer perlu memasang *extensions* Flutter di Visual Studio Code dan *plugins* Flutter di Android Studio. Developer dapat memilih salah satu *code editor* antara Visual Studio Code atau Android Studio. Visual Studio Code dapat mengurangi penggunaan memori pada komputer sehingga dapat dijadikan solusi terbaik bagi developer yang memiliki kendala kualitas performa komputernya. **Keenam**, developer dapat membuat proyek SIM-RS At Turrots pada Visual Studio Code atau Android Studio. Developer dapat membuat proyek SIM-RS At Turrots pada Visual Studio diawali dengan membuka *Command Pallete* pada aplikasi tersebut. *Command Pallete* dapat dibuka dengan menekan tombol CTRL+Shift+P. Selanjutnya, Developer harus menekan Flutter: New Project yang dapat dilihat pada **Gambar 3.5**.



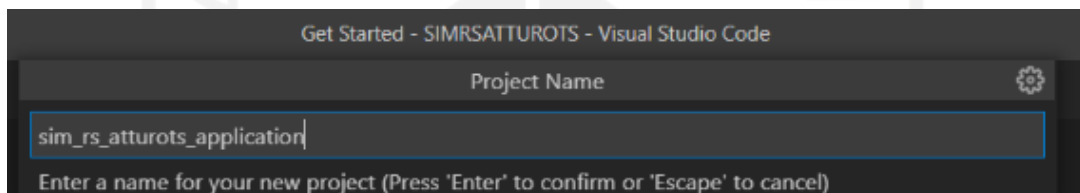
Gambar 3.5 *Command Pallete* pada Visual Studio Code

Developer dapat melanjutkan prosesnya dengan memilih Application, dapat dilihat pada **Gambar 3.6**.



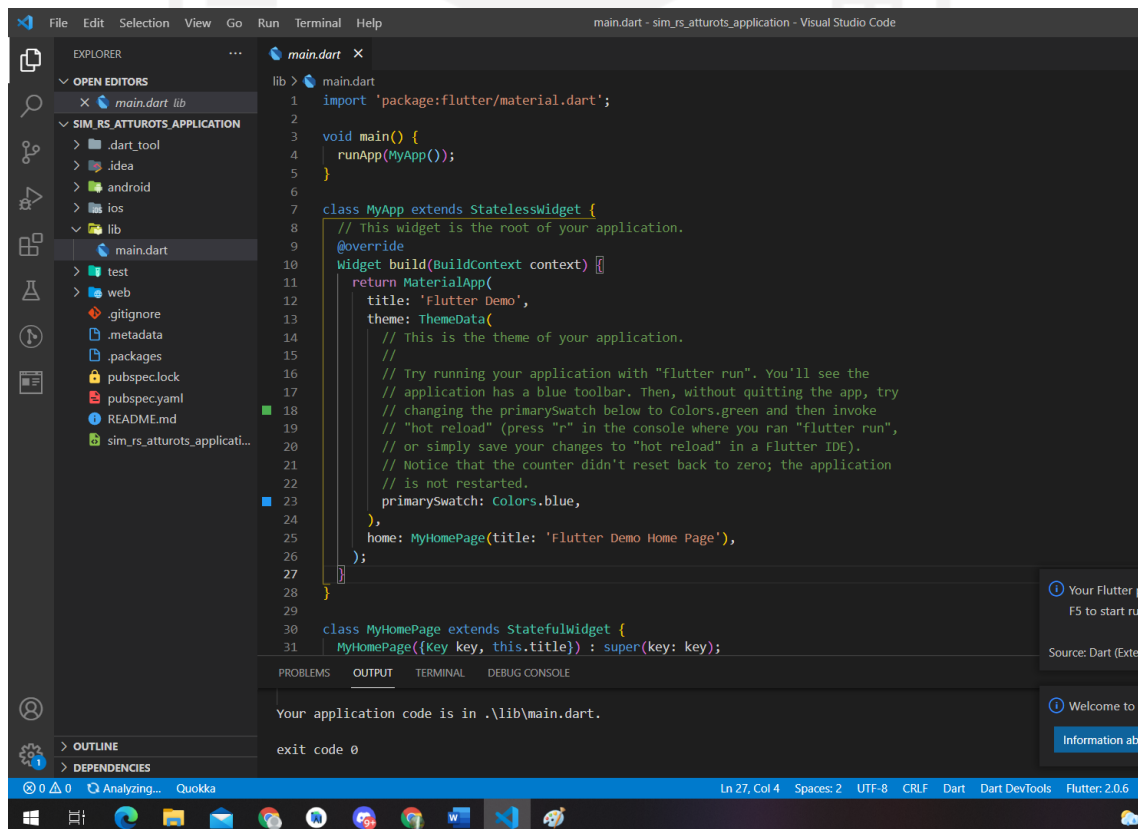
Gambar 3.6 *Command Palletete* setelah menekan Flutter: New Project.

Setelah itu, developer harus memilih *folder* sebagai tempat penyimpanan dan memberi nama pada proyek tersebut, dapat dilihat pada **Gambar 3.7**.



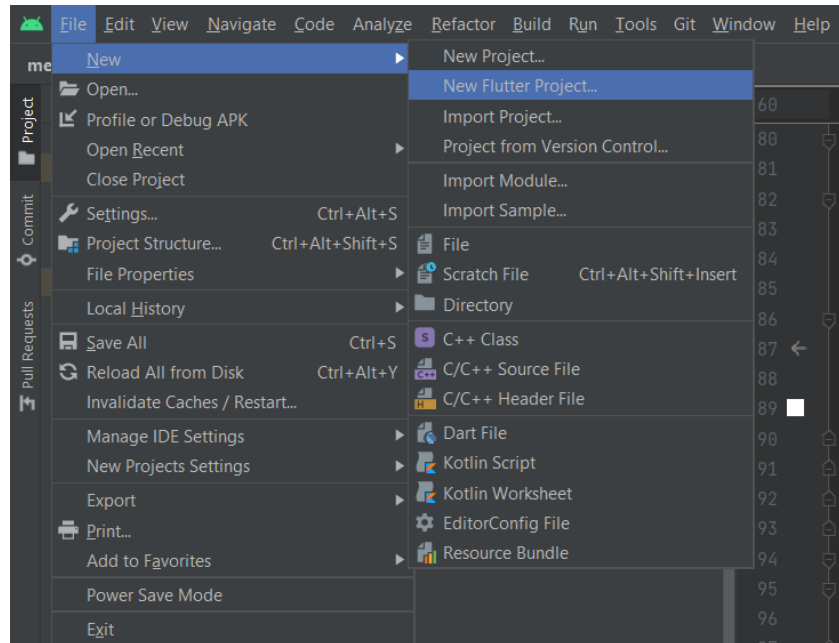
Gambar 3.7 *Command Palletete* setelah menekan tombol Application.

Proyek yang berhasil dibuat pada Visual Studio dapat dilihat pada **Gambar 3.8**.



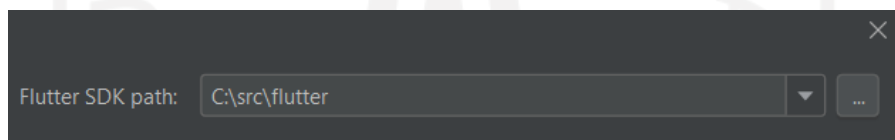
Gambar 3.8 *framework* Flutter pada Visual Studio Code

Developer juga dapat membuat proyek Flutter pada Android Studio yang menggunakan banyak memori komputer dibandingkan dengan Visual Studio Code. Developer harus menekan tombol New Flutter Project seperti yang ada pada **Gambar 3.9**.

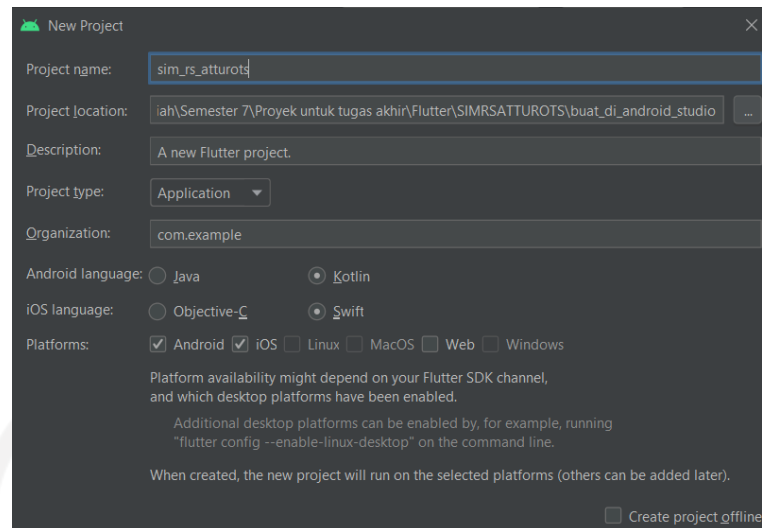


Gambar 3.9 Tombol New Flutter Project

Setelah itu, developer harus mengisi Flutter SDK Path yang dapat dilihat pada **Gambar 3.10**. Flutter SDK Path terdapat pada folder `C:\src\flutter`. Terakhir, developer SIM-RS At Turrots dapat mengisi formulir seperti yang ada pada **Gambar 3.11**. Developer dapat melihat gambar *framework* Flutter di Android Studio pada **Gambar 3.1**.

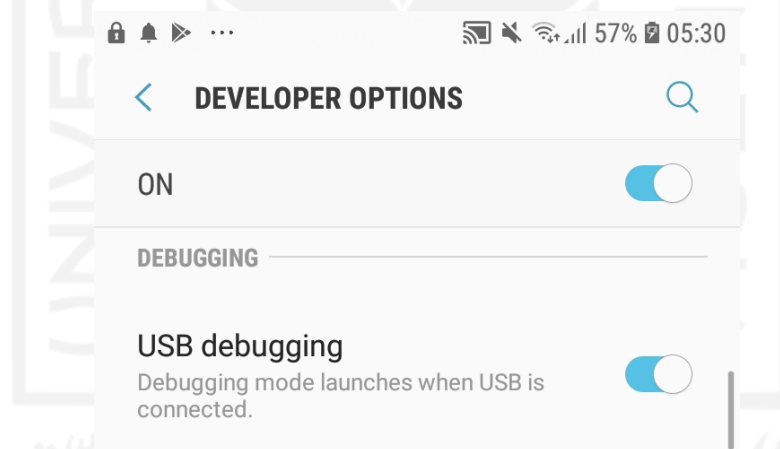


Gambar 3.10 Flutter SDK Path



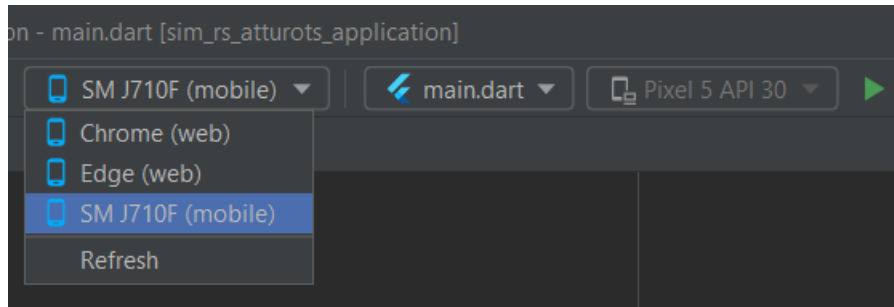
Gambar 3.11. Formulir proyek baru SIM-RS At Turrots

Terakhir, developer perlu menjalankan proyek aplikasi Flutter dapat menggunakan *real device* maupun Emulator. Developer yang ingin menjalankan aplikasinya melalui *real device* perlu mengaktifkan USB Debugging yang ada pada *device*. USB Debugging dapat dilihat pada **Gambar 3.12**.

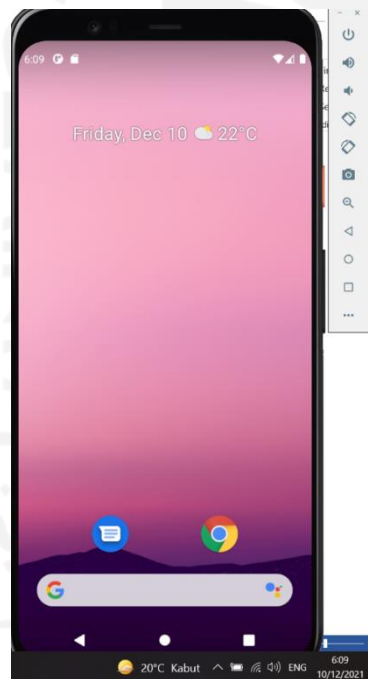


Gambar 3.12 USB Debugging pada *real device* Samsung J7

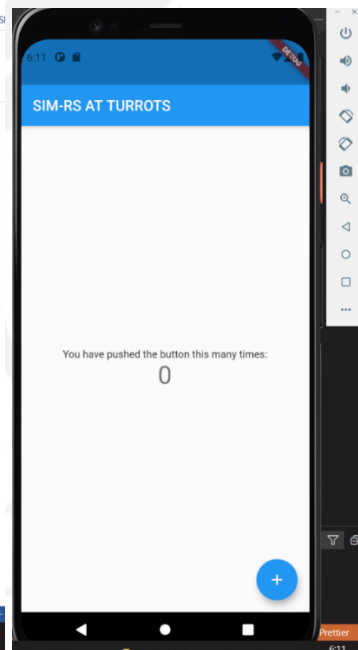
Setelah itu, developer dapat menjalankan aplikasinya pada Visual Studio Code dengan cara menekan tombol F5. Sedangkan pada Android Studio, developer perlu memilih *device* yang menjalankan aplikasinya dan menekan tombol *debugging* yang berwarna hijau, dapat dilihat pada **Gambar 3.13**.



Gambar 3.13 Pemilihan *device* dan tombol *debugging* yang berwarna hijau pada Android Studio Developer yang ingin menjalankan aplikasi melalui Emulator perlu mengunduh Android Virtual Device (AVD) yang ada pada Android Studio. Selanjutnya, developer harus menjalankan AVD yang ada pada Android Studio. AVD yang sudah dijalankan dapat dilihat pada **Gambar 3.14**. Setelah menjalankan AVD, developer dapat menjalankan proyek aplikasinya melalui Visual Studio Code dengan cara menekan tombol **F5** dan melalui Android Studio dengan cara memilih *device* yang akan dijalankan lalu menekan tombol *debugging*. Proyek yang sudah dijalankan melalui AVD dapat dilihat pada **Gambar 3.15**.



Gambar 3.14 AVD

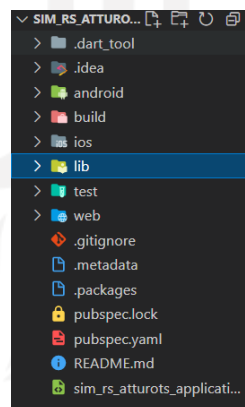


Gambar 3.15 AVD

Struktur Folder Proyek Flutter

Struktur folder proyek Flutter dapat dilihat pada **Gambar 3.16**. Pada Flutter terdapat modul-modul aplikasi, yaitu: Dart Tool, Idea, Android, Build, Ios, Lib, Test, dan Web. Modul Dart Tool digunakan untuk menggantikan direktori *pub*. Pada modul tersebut developer tidak perlu mengubah isi dari folder-folder yang ada pada modul tersebut. Idea merupakan modul yang

berisi semua pengaturan proyek yang akan disimpan secara spesifik dalam bentuk XML. Developer tidak perlu mengubah isi modul tersebut karena modul tersebut akan berubah secara otomatis oleh sistem Flutter. Selanjutnya Android, Android merupakan modul yang sangat penting untuk memproses *build* menjadi aplikasi Android Native. Dalam modul Android, developer dapat mengkonfigurasi pengembangan proyeknya untuk aplikasi Android Native. Selanjutnya Ios, modul tersebut sangat penting untuk memproses *build* menjadi aplikasi iOS Native dan developer dapat mengkonfigurasi pengembangan proyeknya dalam modul tersebut. Selanjutnya modul Lib, modul ini berisi *source code* untuk aplikasi Flutter. Flutter akan secara otomatis membuat *file* main.dart untuk diarahkan ke *file-file* yang akan baru dibuat oleh developer. Selanjutnya modul Test, modul ini digunakan untuk melakukan *testing* pada proyek aplikasi Flutter. Selanjutnya modul Web, modul ini berfungsi untuk memproses *build* menjadi aplikasi Web. Pada Flutter juga terdapat *file-file* lainnya, yaitu: .gitignore (berisi *file* Git), .metadata (berisi *metadata* proyek yang akan di *Generate* secara otomatis), packages (*file* yang berisi alamat *path package* yang dibuat oleh pub), pubspec.lock (yang berfungsi untuk mengunci versi *package* dan berisi versi-versi *library* yang dibuat oleh pub), pubspec.yaml (*file* yang berisi informasi mengenai proyek Flutter dan menambahkan *library* yang dibutuhkan), README.md (*file markdown* yang berisi mengenai penjelasan *source code* proyek Flutter), dan nama_proyek.iml (*file* yang tidak bisa diubah secara manual dan berisi pengaturan dari proyek Flutter).



Gambar 3.16 Struktur folder proyek Flutter

Penggunaan *Widget ListView*

Pada Flutter terdapat *widgets* yang membentuk tampilan antarmuka penggunanya. Developer dapat menampilkan kumpulan data dalam bentuk sebuah *variable list* menggunakan *widget* ListView yang sudah disediakan oleh Flutter. Pada aplikasi ATGOV, *widget* ListView dapat dilihat pada **Gambar 3.17**.


```

|ListView.builder(
padding: EdgeInsets.zero,
scrollDirection: Axis.vertical,
shrinkWrap: true,
itemCount: searchWilayahLoaded.listWilayahModel.length,
itemBuilder: (context, index) => (index <
    searchWilayahLoaded.listWilayahModel.length)
    ? Container(
        child: WilayahItemWidget(
            wilayahModel:
                searchWilayahLoaded.listWilayahModel[index],
            backgroundColor: Colors.green,
            token: tokenLogged,
        )) // WilayahItemWidget, Container

```

Gambar 3.17 widget ListView

Dalam *widget* ListView terdapat banyak parameter, seperti: padding, scrollDirection, shrinkWrap, itemCount, itemBuilder. Parameter yang sangat ditekankan disini adalah itemCount dan itemBuilder. Dalam itemCount, developer perlu menghitung jumlah data yang ada pada *list* Wilayah menjadi *variable integer*. Terakhir, itemBuilder akan menampilkan *widget* pada tampilan antarmuka pengguna. Dalam parameter itemBuilder terdapat *widget* WilayahItemWidget yang sudah dibuat oleh developer ATGOV. WilayahItemWidget berisi data kode, wilayah, tipe, wilayah lengkap, dan aksinya. Kode WilayahItemWidget dapat dilihat pada **Gambar 3.18**. Hasil implementasi kode *widget* WilayahItemWidget dapat dilihat pada **Gambar 3.19**.

```

import 'package:atgov/ui/pages/home/mastendata/wilayah/edit_wilayah_page.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:get/get.dart';

class WilayahItemWidget extends StatelessWidget {
  Color backgroundColor;
  WilayahModel wilayahModel;
  String token;

  WilayahItemWidget({this.backgroundColor, this.wilayahModel, this.token});

  @override
  Widget build(BuildContext context) {
    return Container(
      child: Card(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(15.0),
        ), // RoundedRectangleBorder
        color: backgroundColor,
        elevation: 5,
        child: Container(
          padding: EdgeInsets.all(10),
          child: Column(
            children: <Widget>[
              Container(
                child: Table(
                  columnWidths: {
                    0: FlexColumnWidth(3),
                    1: FlexColumnWidth(0.5),
                    2: FlexColumnWidth(6),

```

Gambar 3.18 widget WilayahItemWidget



Gambar 3.19 Hasil tampilan kode *widget* ListView

Pemanggilan Data dari REST API

Gambar 3.20 adalah pemanggilan data dari REST API pada EndPoint Tipe Wilayah menggunakan aplikasi Postman. EndPoint Tipe Wilayah merupakan EndPoint yang dipakai pada dua halaman yang bernama Tambah Wilayah dan Edit Wilayah di aplikasi *mobile* ATGOV. Pada gambar tersebut terdapat objek JSON dan array JSON yang akan dideklarasikan pada Flutter maupun Android Native. Terdapat data-data pada objek JSON, yaitu kode, deskripsi, dan *list* data.

```

{
  "kode": 0,
  "deskripsi": "Ok",
  "data": [
    {
      "tipe": "PROV",
      "tipe_label": "Provinsi"
    },
    {
      "tipe": "KAB",
      "tipe_label": "Kabupaten / Kota"
    },
    {
      "tipe": "KEC",
      "tipe_label": "Kecamatan / Kapanewon / Kemantien"
    }
  ]
}

```

Gambar 3.20 Pemanggilan data Tipe Wilayah menggunakan aplikasi Postman

Pemanggilan pada Flutter perlu melakukan beberapa persiapan, seperti: menyiapkan *library* http, membuat Model, dan membuat fungsi `getListTipeWilayah` di kode halaman *User Interface* Flutter. `TipeWilayahModel` yang sudah dibuat dapat dilihat pada **Gambar 3.21**.

```

class TipeWilayahModel {
  String tipe;
  String tipe_label;

  TipeWilayahModel({this.tipe, this.tipe_label});

  factory TipeWilayahModel.createdTipeWilayah(Map<String, dynamic> jsonObject) {
    return TipeWilayahModel(
      tipe: jsonObject['tipe'].toString(),
      tipe_label: jsonObject['tipe_label'].toString()
    );
  }
}

```

Gambar 3.21 Model TipeWilayahModel

Gambar 3.22 adalah fungsi `getListTipeWilayah` yang sudah dibuat untuk memanggil data dari REST API. Dalam fungsi tersebut terdapat *variable* `url` yang nantinya akan dideklarasikan pada *variable* `response`. *Variable* `response` akan memanggil data pada server melalui perantara REST API dan *library* `http`. Setelah memanggil pada *variable* `response`, developer perlu mengubah objek JSON menjadi objek Dart. Kemudian, developer dapat memanggil *list* dari objek Dart yang bernama 'data' pada *variable* `listTipeWilayah`. Selanjutnya, developer dapat memasukkan *list* tersebut ke dalam *variable* yang bertipe *list*. *List* tersebut berisi kumpulan data-data TipeWilayahModel. Terakhir, data-data yang berada pada *variable* `list` dapat ditampilkan pada tampilan antarmuka pengguna. Data yang ada pada *variable* bertipe *list* tersebut dapat ditampilkan dengan *widget* `ListView`. *Widget* `ListView` dapat dilihat pada **Penggunaan Widget ListView**.

```

Future<Null> getListWilayah() async {
  String url = 'https://dev.atsoft.co.id/atgov-api-sumsel/cr_wilayah/get_list_tipe?token=a928393129312312';
  var response = await http.get(url);
  var jsonObject = json.decode(response.body);

  List<dynamic> listTipeWilayah = (jsonObject as Map<String, dynamic>) ['data'];
  List<TipeWilayahModel> listTipeWilayahModel = [];

  for(int i = 0; i < listTipeWilayah.length; i++)
    listTipeWilayahModel.add(TipeWilayahModel.createdTipeWilayah(listTipeWilayah[i]));

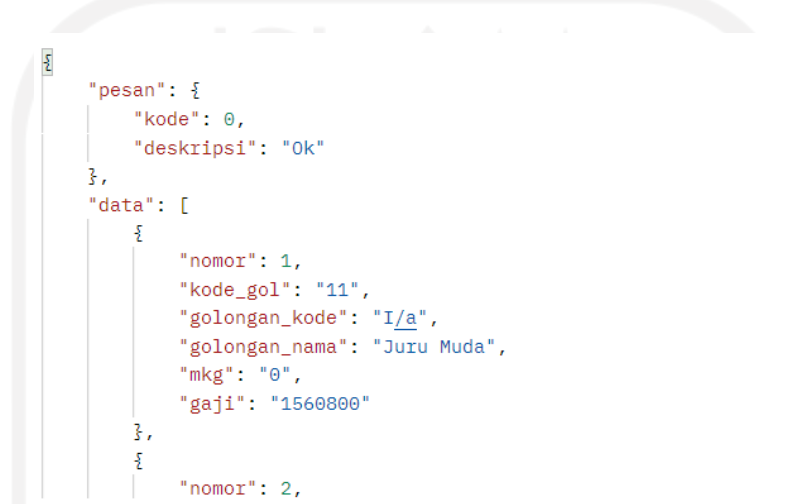
  setState(() {});
}

```

Gambar 3.22 fungsi `getListTipeWilayah`

Pemanggilan Data dari REST API dengan Konsep BLoC Pattern

Gambar 3.23 adalah pemanggilan data dari REST API pada Endpoint Kenaikan Gaji menggunakan aplikasi Postman. EndPoint Kenaikan Gaji merupakan EndPoint yang dipakai pada salah satu halaman yang bernama Kenaikan Gaji Berkala di aplikasi *mobile* ASN Memayu. Pada gambar tersebut terdapat objek JSON dan array JSON. Terdapat data-data pada objek JSON, yaitu pesan dan data. Dalam data terdapat array yang berbentuk JSON. Setelah itu, pemanggilan data dari REST API tersebut akan dideklarasikan dengan konsep BLoC Pattern.



```

"pesan": {
  "kode": 0,
  "deskripsi": "Ok"
},
"data": [
  {
    "nomor": 1,
    "kode_gol": "11",
    "golongan_kode": "I/a",
    "golongan_nama": "Juru Muda",
    "mkg": "0",
    "gaji": "1560800"
  },
  {
    "nomor": 2,

```

Gambar 3.23 Pemanggilan data Kenaikan Gaji menggunakan aplikasi Postman

Pemanggilan data dari REST API pada End Point Kenaikan Gaji dengan konsep BLoC Pattern untuk Flutter perlu melakukan beberapa persiapan, seperti: menyiapkan *library* http dan flutter_bloc, membuat Model Kenaikan Gaji, membuat BLoC dan kelas *User Interface*. Model Kenaikan Gaji dapat dilihat pada **Gambar 3.24**.

```

class KenaikanGajiDataModel {
  String nomor;
  String kodeGol;
  String mkg;
  String gaji;

  KenaikanGajiDataModel({
    this.nomor,
    this.gaji,
    this.kodeGol,
    this.mkg,
  });

  factory KenaikanGajiDataModel.fromJson(Map<String, dynamic> json) {
    return KenaikanGajiDataModel(
      nomor: json['nomor'].toString(),
      kodeGol: json['kode_gol'].toString(),
      mkg: json['mkg'].toString(),
      gaji: json['gaji'].toString());
  }
}

```

Gambar 3.24 Model KenaikanGajiDataModel

Pada Model tersebut terdapat variable nomor, kodeGol, mkg, dan gaji. *Variable* tersebut akan digunakan untuk pemanggilan data dari End Point. Pada Model tersebut juga berfungsi untuk kelas *User Interface* sebagai tempat pemanggilan data. Selanjutnya, developer perlu membuat BLoC pada proyek Flutternya. BLoC tersebut terdiri dari tiga kelas, yaitu: Event, State, dan Bloc. Kelas Event merupakan kelas yang memasukkan data token pengguna agar Endpoint tersebut dapat diakses oleh aplikasinya. Kelas Event dapat dilihat pada **Gambar 3.25**.

```

class KenaikanGajiBlocEvent {
  String token;

  KenaikanGajiBlocEvent({this.token});
}

```

Gambar 3.25 Kelas Event KenaikanGajiBlocEvent

Selanjutnya kelas State, developer perlu membuat kelas State beserta kelas turunannya, yaitu: Loading, Error, Uninitialized, dan Loaded. Kelas State beserta kelas turunannya dapat dilihat pada **Gambar 3.26**.

```

class KenaikanGajiBlockState {}

class KenaikanGajiBlocLoading extends KenaikanGajiBlockState {}

class KenaikanGajiBlocError extends KenaikanGajiBlockState {}

class KenaikanGajiBlocUninitialized extends KenaikanGajiBlockState {}

class KenaikanGajiBlocLoaded extends KenaikanGajiBlockState {
  List<KenaikanGajiDataModel> listKenaikanGajiDataModel;

  KenaikanGajiBlocLoaded({this.listKenaikanGajiDataModel});

  KenaikanGajiBlocLoaded copyWith(
    {List<KenaikanGajiBlocLoaded> listKenaikanGajiDataModel}) {
    return KenaikanGajiBlocLoaded(
      listKenaikanGajiDataModel:
        listKenaikanGajiDataModel ?? this.listKenaikanGajiDataModel);
  }
}

```

Gambar 3.26 Kelas *state* beserta turunannya

Kelas Loading berfungsi untuk menampilkan *widget loading* saat data sedang di proses. Kelas Error berfungsi untuk menampilkan *widget error* ketika terdapat kesalahan pada saat mengakses Endpointnya. Kelas Uninitialized berfungsi sebagai kelas yang menampilkan *widget* pada saat *User Interface* di akses oleh pengguna. Kelas Loaded berfungsi ketika pemanggilan data pada End Point berhasil. Ketika berhasil, Data yang ada Endpoint tersebut akan dilanjutkan kepada kelas *User Interface*. Setelah itu, developer perlu membuat kelas Bloc yang dapat dilihat pada **Gambar 3.27**.


```

class KenaikanGajiBloc extends Bloc<KenaikanGajiBlocEvent, KenaikanGajiBlockState> {
  @override
  KenaikanGajiBlockState get initialState => KenaikanGajiBlocUninitialized();

  @override
  Stream<KenaikanGajiBlockState> mapEventToState(
    KenaikanGajiBlocEvent event) async* {
    List<KenaikanGajiDataModel> listKenaikanGajiDataModel;

    yield KenaikanGajiBlocLoading();

    try {
      if (state is KenaikanGajiBlocUninitialized) {
        listKenaikanGajiDataModel =
          await KenaikanGajiServices.getKenaikanGaji(event.token);
        yield KenaikanGajiBlocLoaded(
          listKenaikanGajiDataModel: listKenaikanGajiDataModel);
      }
    }
  }
}

```

Gambar 3.27 Kelas Bloc KenaikanGajiBloc

Pada kelas tersebut terdapat logika untuk menampilkan data dan *state*. Terdapat *yield* untuk mendelegasikan semua turunan *state*. Terakhir, developer perlu mendaftarkan BLoC yang sudah dibuat ke kelas Main yang digunakan sebagai fungsi utama *starting point* untuk aplikasi Flutter. Setelah BLoC didaftarkan, BLoC dapat dipakai oleh kelas yang sudah dibuat oleh developer. Salah satu contoh pemakaian BLoC pada *User Interface* dapat dilihat pada **Gambar 3.28**.

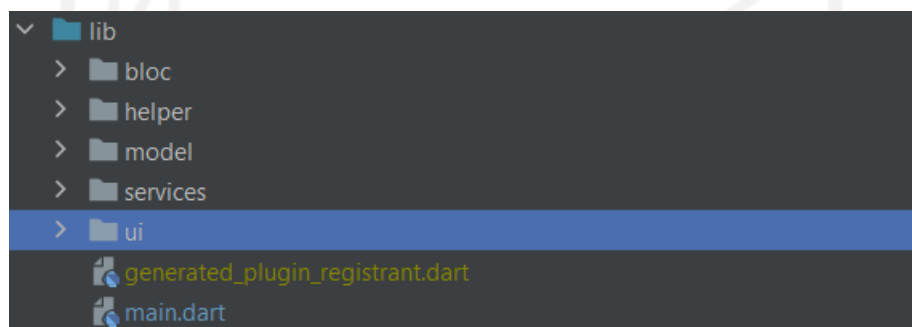
```

BlocBuilder<KenaikanGajiBloc, KenaikanGajiBlockState>(
  builder: (context, state) {
    if (state is KenaikanGajiBlocUninitialized) {
      print("1");
    }
    return Expanded(
      child: Container(
        margin: EdgeInsets.only(
          bottom: ResponsiveFlutter.of(context).hp(30)), // B
        child: Center(
          child: SizedBox(
            width: 30,
            height: 30,
            child: CircularProgressIndicator(),
          ), // SizedBox
        ), // Center
      ), // Container
    ); // Expanded
  }
)

```

Gambar 3.28 Pemakaian Bloc di kelas *user interface*

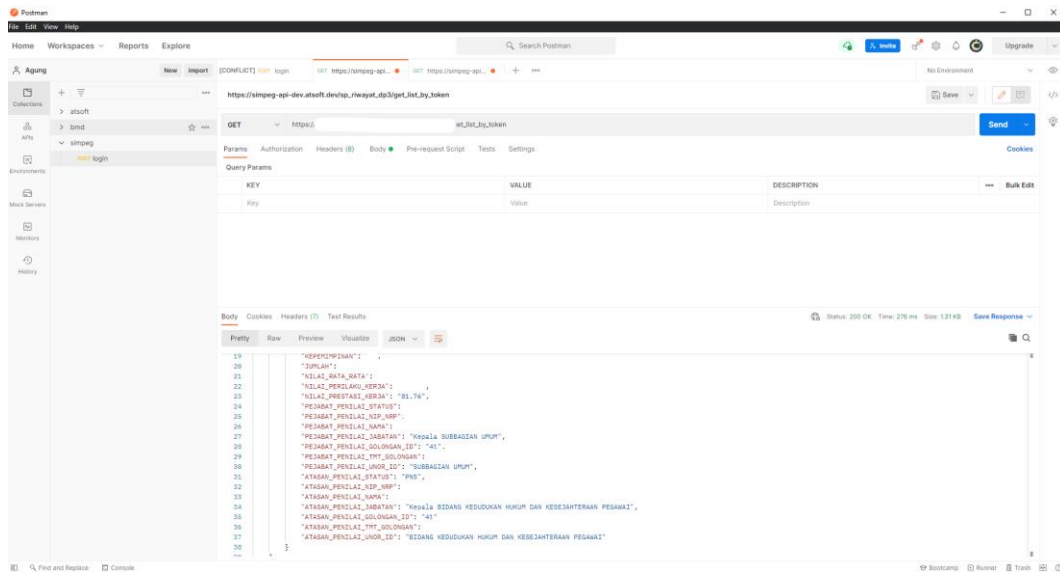
Struktur folder yang dibuat ketika developer memakai konsep BLoC Pattern, yaitu: Model, Bloc, dan *User Interface* (UI). Folder Model berisi kelas model-model proyek ASN Memayu yang digunakan untuk memproses dan menyediakan data. Folder Bloc berisi kelas Bloc, State, dan Event yang berfungsi untuk memisahkan logika memanggil data dari kelas yang akan menampilkan tampilan *User Interface*. Bloc tersebut dapat digunakan oleh kelas *User Interface* yang ada pada proyek ASN Memayu. Terakhir *User Interface*, folder ini berisi kelas *User Interface* yang akan menampilkan dan membuat tampilan antarmuka pengguna. Struktur tersebut dapat dilihat pada **Gambar 3.29**.



Gambar 3.29 Struktur folder dengan menggunakan *design pattern* BLoC

3.2.1.2 Postman

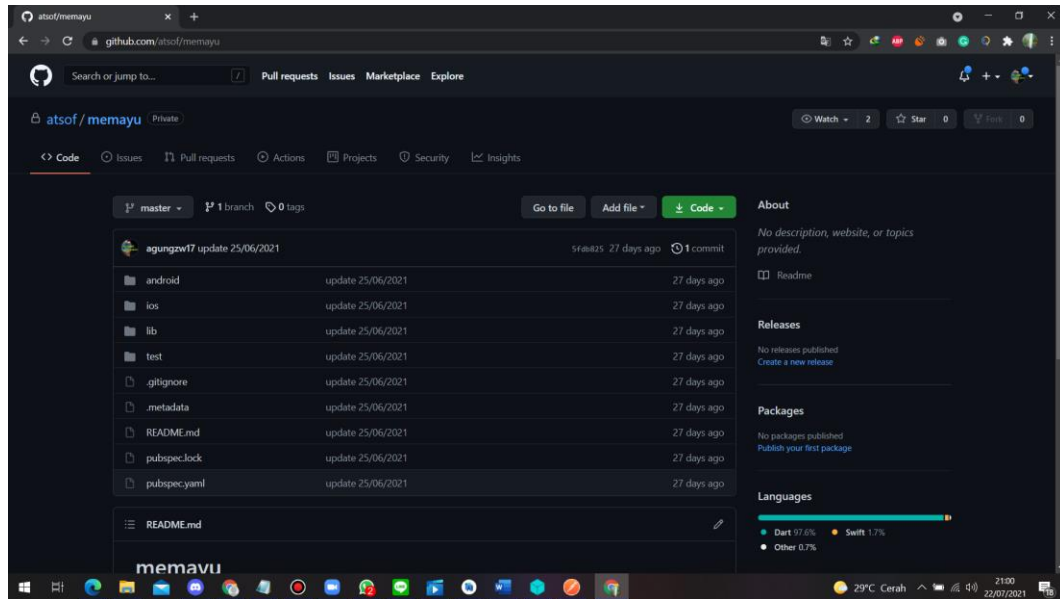
Postman adalah platform *Application Programming Interface* (API) untuk membuat dan menggunakan API. Postman ini dapat membantu untuk mempercepat proses pengujian, dokumentasi, dan desain. Postman digunakan untuk melakukan pengujian data Endpoint API sebelum mensimulasikannya kedalam di platform yang akan dikembangkannya pada proyek SIM-RS At Turrots, ATGOV, dan ASN Memayu. Gambar tampilan Postman dapat dilihat pada **Gambar 3.30**.



Gambar 3.30 Tampilan aplikasi Postman.

3.2.1.3 Github

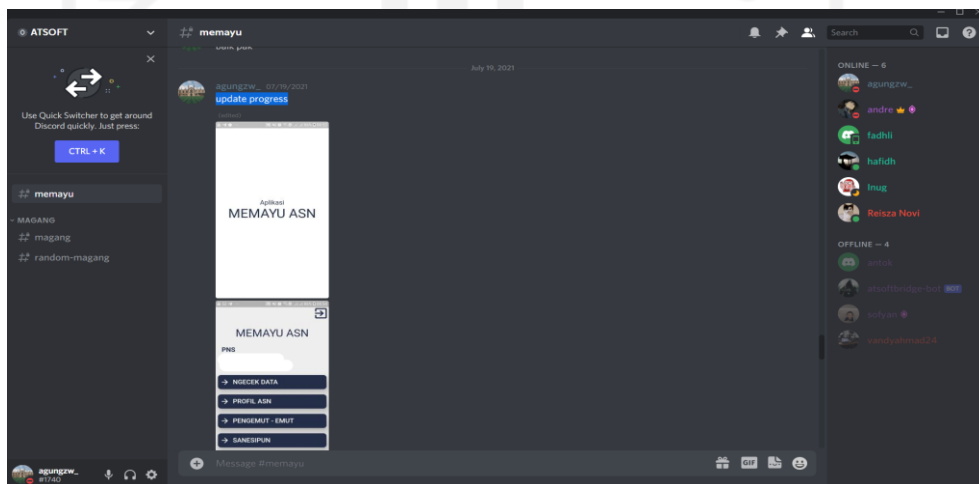
Github merupakan platform web dan layanan berbasis *cloud* untuk menyimpan dan mengelola kode para developer. Dalam platform ini, developer dapat mendokumentasikan dan mengontrol perubahan kode mereka sesuai dengan kebutuhan di masa yang akan datang. Selain itu, platform ini merupakan bagian dari *storehouse online* terbesar di dunia. Dengan platform ini, semua orang di berbagai tempat dapat bekerja sama untuk membuat aplikasi secara bersama-sama (Faradilla A, 2021). Github digunakan pada proyek SIM-RS At Turrots, ATGOV, dan ASN Memayu untuk berbagi dan menyimpan *source code* sebagai berkas perusahaan CV. Atsoft Teknologi. Tampilan aplikasi *website* github yang terdapat salah satu proyek perusahaan dapat dilihat pada **Gambar 3.31**.



Gambar 3.31 Tampilan aplikasi website Github.

3.2.1.4 Discord

Discord adalah platform yang dirancang untuk membuat komunitas karena memungkinkan penggunaannya dalam melakukan aktivitas sosial. Dalam platform ini, pengguna melakukan komunikasi melalui media panggilan suara, video, file, dan pesan. Discord digunakan untuk berkomunikasi dengan seluruh karyawan yang ada di perusahaan CV. Atsoft Teknologi. Discord juga digunakan untuk memberikan tugas oleh *Project Manager* dan menyampaikan tugas yang telah diselesaikan oleh mahasiswa magang. Tampilan Discord dapat dilihat pada **Gambar 3.32**



Gambar 3.32 Tampilan aplikasi Discord.

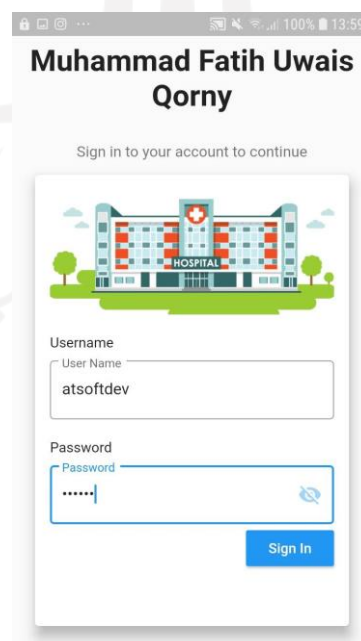
3.2.2 Sistem Informasi Rumah Sakit At Turrots (SIM-RS At Turrots)

Proyek SIM-RS At turrot merupakan proyek dari Rumah Sakit At turrots. Permasalahan yang dialami oleh rumah sakit At turrots adalah proses pencatatan administrasinya masih manual.

Rumah sakit merasa kesulitan karena mengurus data pasien yang sangat banyak dan sudah sering mendapatkan kesalahan dalam cara menghitung atau mengatur data-data pasien. Akibatnya, rumah sakit menanggung banyak kerugian. Proyek SIM-RS At turrots dikembangkan untuk membantu menangani permasalahan seputar administrasi di rumah sakit At turrots. Proyek SIM-RS At turrots dikembangkan dalam platform web dan mobile. Adapun halaman aplikasi *mobile* yang sudah dibuat pada saat magang dapat dilihat pada **Bab 3.2.2.1**, **Bab 3.2.2.2**, dan **Bab 3.2.2.3**.

3.2.2.1 Halaman Login SIM-RS At Turrots

Halaman Login adalah halaman yang mempunyai fitur *login*. Fitur *login* tersebut merupakan fitur yang mengatur proses masuknya pengguna ke dalam halaman Menu Utama aplikasi SIM-RS At Turrots. Dalam halaman tersebut terdapat formulir yang harus di isi pengguna, yaitu: *username* dan *password*. Pengguna dapat menekan tombol Sign In ketika sudah mengisi semua formulir dan formulir tersebut akan di kirim ke *database server* melalui perantara API. Formulir yang diisi pengguna harus sama dengan yang ada pada data pengguna di *database server* SIM-RS At Turrots. Dalam hal ini, jika pengguna memasukkan formulir yang berbeda, pengguna tidak dapat masuk ke halaman Menu Utama aplikasi *mobile* SIM-RS At Turrots. Setelah melakukan *login*, data-data pengguna akan di simpan ke *database local* yang bernama Shared Preferences. Dengan Shared Preferences, Pengguna yang sudah melakukan *login* tidak perlu melakukannya kembali hingga menekan fitur *logout*. Fitur *logout* berfungsi untuk menghapus data-data pengguna ketika *login*. Halaman Login dapat dilihat pada **Gambar 3.33**.

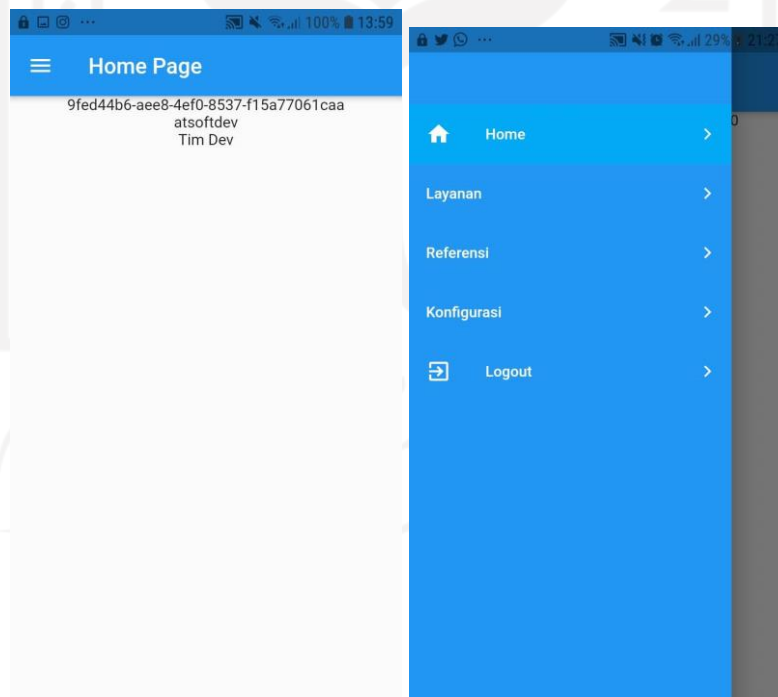


Gambar 3.33 Halaman login SIM-RS At Turrots

3.2.2.2 Halaman Menu Utama SIM-RS At Turrots

Pengguna yang sudah melakukan *login* dapat langsung masuk ke halaman Menu Utama SIM-RS At Turrots ketika membuka aplikasi di *smartphone*-nya. Halaman tersebut menampilkan data pengguna yang sudah melakukan *login*, dapat dilihat pada **Gambar 3.34**. Data yang ditampilkan adalah token pengguna, *username* pengguna, dan nama pengguna. Data tersebut berasal dari *database server* SIM-RS At Turrots. Selanjutnya, terdapat tombol yang ada pada kiri atas **Gambar 3.35**. Tombol tersebut ketika ditekan akan menampilkan fitur *sidebar*. Fitur ini akan menampilkan menu – menu utama yang ada pada aplikasi SIM-RS At Turrots, dapat dilihat pada **Gambar 3.35**. Terdapat lima menu utama yang ada pada fitur *sidebar*, yaitu:

1. Home ketika ditekan maka akan kembali ke halaman Menu Utama.
2. Layanan ketika ditekan maka akan masuk ke halaman Layanan.
3. Referensi ketika ditekan maka akan masuk ke halaman Referensi.
4. Konfigurasi ketika ditekan maka akan masuk ke halaman Konfigurasi.
5. Logout ketika ditekan maka akan masuk ke halaman Login. Menu ini mempunyai fitur khusus yang bernama *logout*.



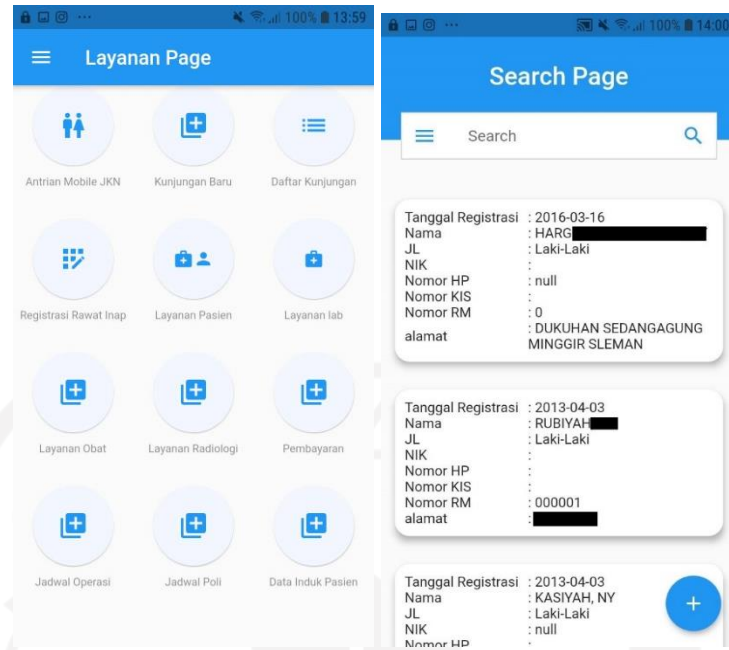
Gambar 3.34 Halaman Menu Utama SIM-RS At Turrots (Kiri) dan **Gambar 3.35** Menu-menu utama SIM-RS At Turrots pada fitur *SideBar* (Kanan)

3.2.2.3 Halaman Layanan SIM-RS At Turrots

Halaman Layanan akan muncul ketika menekan menu Layanan yang ada pada fitur *sidebar* di halaman Menu Utama. Dalam halaman ini terdapat menu-menu Layanan, yaitu:

1. Antrian Mobile JKN
2. Kunjungan Baru
3. Daftar Kunjungan
4. Registrasi Rawat Inap
5. Layanan Pasien
6. Layanan Lab
7. Layanan Obat
8. Layanan Radiologi
9. Pembayaran
10. Jadwal Operasi
11. Jadwal Poli
12. Data Induk Pasien.

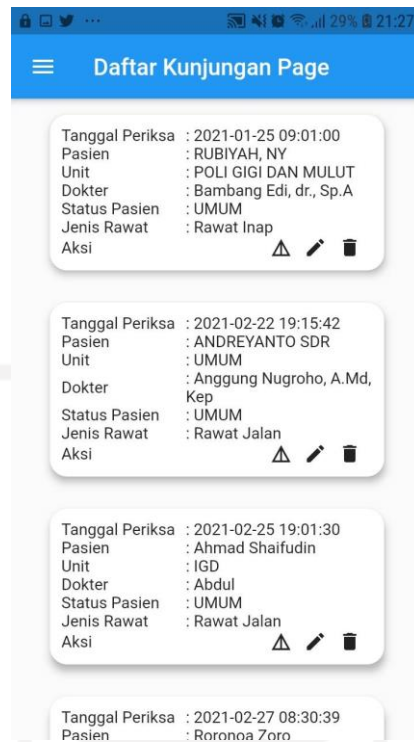
Halaman Layanan dapat dilihat pada **Gambar 3.36**. Terdapat beberapa menu Layanan yang sudah dibuat pada saat magang, yaitu: Kunjungan Baru dan Daftar Kunjungan. Dalam menu Kunjungan Baru terdapat fitur mencari data pasien yang baru mendaftar dan melihat data pasien yang baru mendaftar, dapat dilihat pada **Gambar 3.37**. Dalam menu tersebut terdapat tombol yang mempunyai *icon add* untuk masuk ke dalam formulir. Formulir tersebut digunakan untuk mendaftarkan pasien baru di Rumah Sakit At Turrots, dapat dilihat pada **Gambar 3.38**. Pengguna dapat menekan tombol Submit pada formulir tersebut dan datanya akan di kirim ke *database server* SIM-RS At Turrots. Selanjutnya, terdapat menu Layanan yang bernama Daftar Kunjungan. Dalam menu tersebut menampilkan data-data pasien yang sudah di proses oleh pihak Rumah Sakit At Turrots, dapat dilihat pada **Gambar 3.39**.



Gambar 3.36 Halaman Layanan SIM-RS At Turrots (Kiri) dan **Gambar 3.37** Halaman menu Kunjungan Baru SIM-RS At Turrots (Kanan)

The image displays two screenshots of the 'Kunjungan Pasien Baru' form in the SIM-RS At Turrots mobile application. The left screenshot shows the form with pre-filled data: Tanggal (2021-06-16), No Rekam Medis (Masukkan no rekam medis), Nama (Masukkan nama), Jenis Kelamin (Pilih Jenis Kelamin), Unit (Pilih Unit), and Jenis Rawat. The right screenshot shows the form with input fields for No Rekam Medis, Nama, Jenis Kelamin, Unit, and Jenis Rawat, along with a Submit button.

Gambar 3.38 Halaman formulir mendaftarkan pasien baru SIM-RS At Turrots



Gambar 3.39 Halaman menu Daftar Kunjungan SIM-RS At Turrots

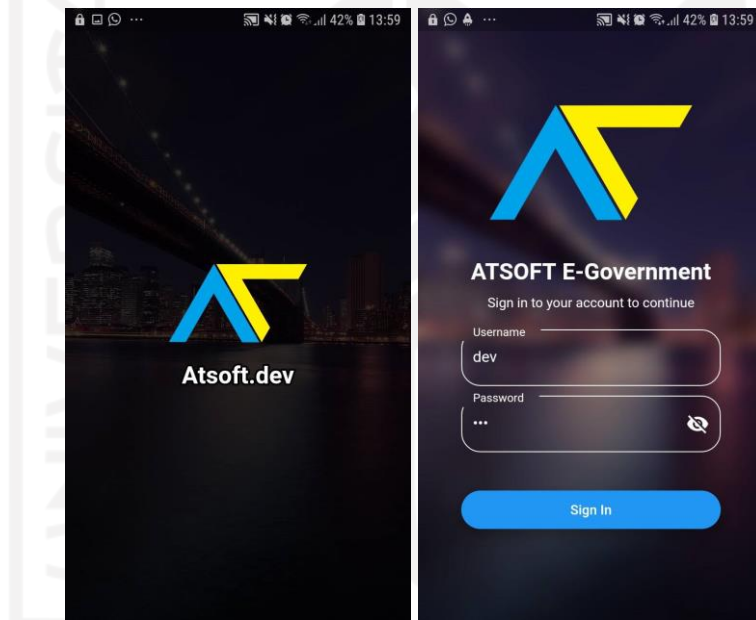
3.2.3 Atsoft E-Government (ATGOV)

Proyek selanjutnya adalah proyek Atsof E-Government. Proyek ini merupakan kelanjutan dari proyek SENGGUH (Sistem Evaluasi Pertanggungjawaban Pembangunan Daerah) yang sudah dibuat oleh CV. Atsoft Teknologi, dapat dilihat pada <http://sengguh.jogiaprov.go.id/intro/>. Proyek SENGGUH merupakan proyek yang dapat memudahkan pencatatan dan perencanaan pembangunan daerah di Yogyakarta agar mencapai sasaran sesuai target Badan Perencanaan dan Pembangunan Daerah Daerah Istimewa Yogyakarta (BAPPEDA DIY). Sistem SENGGUH tersebut masih kaku dan *monolith* (*interface* langsung akses ke database, tidak ada *Application Programming Interface* (API) atau *web service*-nya). Proyek ATGOV dibangun dalam platform web dan mobile. Proyek tersebut berguna untuk melanjutkan proyek baru yang bernama ATGOV-ANJAB di perusahaan CV Atsoft Teknologi. Adapun halaman aplikasi *mobile* ATGOV yang sudah dibuat pada saat magang dapat dilihat pada **Bab 3.2.3.1**, **Bab 3.2.3.2**, **Bab 3.2.3.3**, **Bab 3.2.3.4**, dan **Bab 3.2.3.5**.

3.2.3.1 Halaman Login ATGOV

Halaman Login adalah halaman yang mempunyai fitur *splashscreen* dan *login*. Fitur *splashscreen* merupakan fitur yang bermanfaat sebagai penanda unik sebuah aplikasi. Fitur ini akan menghilang selama waktu yang ditentukan oleh developer dan akan langsung menuju ke fitur *login*, dapat dilihat pada **Gambar 3.40**. Fitur *login* tersebut merupakan fitur yang mengatur

proses masuknya pengguna ke dalam halaman Menu Utama aplikasi ATGOV, dapat dilihat pada **Gambar 3.41**. Dalam halaman tersebut terdapat formulir yang harus diisi pengguna, yaitu: *username* dan *password*. Pengguna dapat menekan tombol Sign In ketika sudah mengisi semua formulir dan formulir tersebut akan di kirim ke *database server* melalui perantara API. Formulir yang diisi pengguna harus sama dengan yang ada pada data pengguna di *database server* ATGOV. Dalam hal ini, jika pengguna memasukkan formulir yang berbeda, pengguna tidak dapat masuk ke halaman Menu Utama aplikasi *mobile* ATGOV. Setelah melakukan *login*, data-data pengguna akan di simpan ke *database local* yang bernama Shared Preferences. Dengan Shared Preferences, Pengguna yang sudah melakukan *login* tidak perlu melakukannya kembali hingga menekan fitur *logout*. Fitur *logout* berfungsi untuk menghapus data-data pengguna ketika *login*.



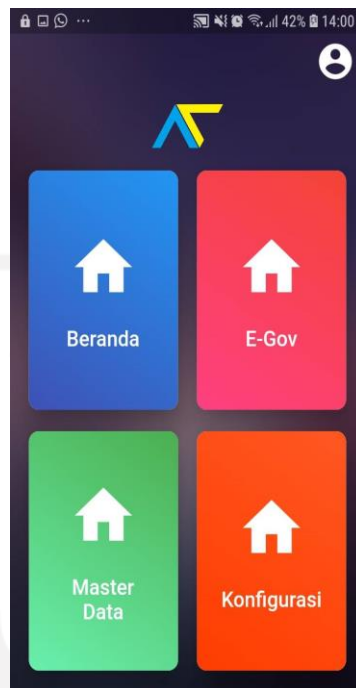
Gambar 3.40 Halaman *splashscreen* ATGOV **Gambar 3.41** Halaman Login ATGOV

3.2.2.2 Halaman Menu Utama ATGOV

Ketika pengguna sudah melakukan *login* maka akan masuk ke halaman Menu Utama ATGOV. Dalam halaman ini terdapat menu-menu utama aplikasi ATGOV. Halaman tersebut dapat dilihat pada **Gambar 3.42**. Terdapat empat menu utama yang ada pada halaman Menu Utama, yaitu:

1. E-Gov
2. Master Data
3. Konfigurasi
4. Profil. Menu ini terdapat pada kanan atas halaman Menu Utama ATGOV yang

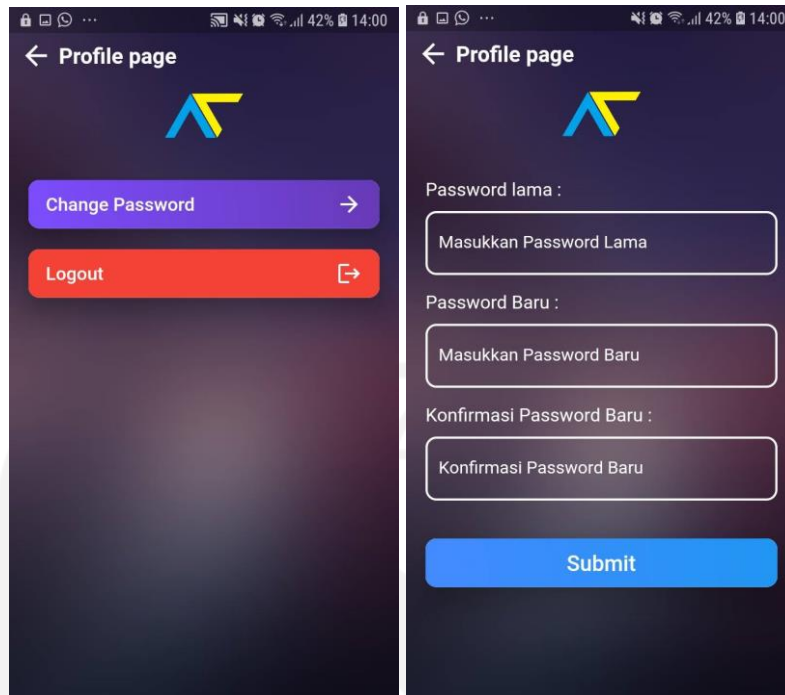
mempunyai *icon person*.



Gambar 3.42 Halaman Menu Utama ATGOV

3.2.2.3 Halaman Profil ATGOV

Halaman Profil akan muncul ketika menekan menu Profil di halaman Utama ATGOV. Dalam halaman ini terdapat 2 fitur khusus, yaitu: *logout* dan mengubah kata sandi pengguna, dapat dilihat pada **Gambar 3.43**. Fitur *logout* akan aktif ketika menekan tombol Logout. Dengan menekan fitur tersebut, semua data *login* yang ada pada *database local* akan terhapus sehingga ketika membuka aplikasi ATGOV pengguna perlu melakukan *login* terlebih dahulu sebelum masuk ke halaman Menu Utama ATGOV. Selanjutnya, terdapat fitur mengubah kata sandi pengguna. Pengguna perlu menekan tombol Change Password sehingga dapat masuk ke tempat yang mempunyai formulir untuk mengganti password, dapat dilihat pada **Gambar 3.44**. Dalam formulir tersebut terdapat tempat pemasukan Password Lama, Password Baru, dan Konfirmasi Password Baru. Setelah mengisi formulir tersebut dengan benar pengguna dapat menekan tombol Submit. Dengan menekan tombol Submit, data formulir tersebut akan dikirim ke *database server* ATGOV melalui perantara API. Ketika Submit berhasil, pengguna diharapkan memasukkan *password* yang terbaru pada saat melakukan *login*.



Gambar 3.43 Halaman Profil ATGOV (Kiri) dan **Gambar 3.44** Halaman tempat mengganti *password* pengguna aplikasi ATGOV (Kanan)

3.2.3.4 Halaman Master Data ATGOV

Halaman Master Data akan muncul ketika menekan menu Master Data yang ada pada halaman Menu Utama ATGOV. Halaman tersebut dapat dilihat pada **Gambar 3.45**. Halaman ini berupa menu-menu Master Data yang akan dijadikan sebagai salah satu data di laporan Dokumen Pelaksanaan Anggaran. Salah satu laporan tersebut dapat dilihat pada **Gambar 3.46**. Menu-menu yang ada pada halaman Menu Utama ATGOV, yaitu:

1. Wilayah
2. Urusan
3. Program
4. Kegiatan
5. Sub Kegiatan



Gambar 3.45 Halaman Menu Utama ATGOV

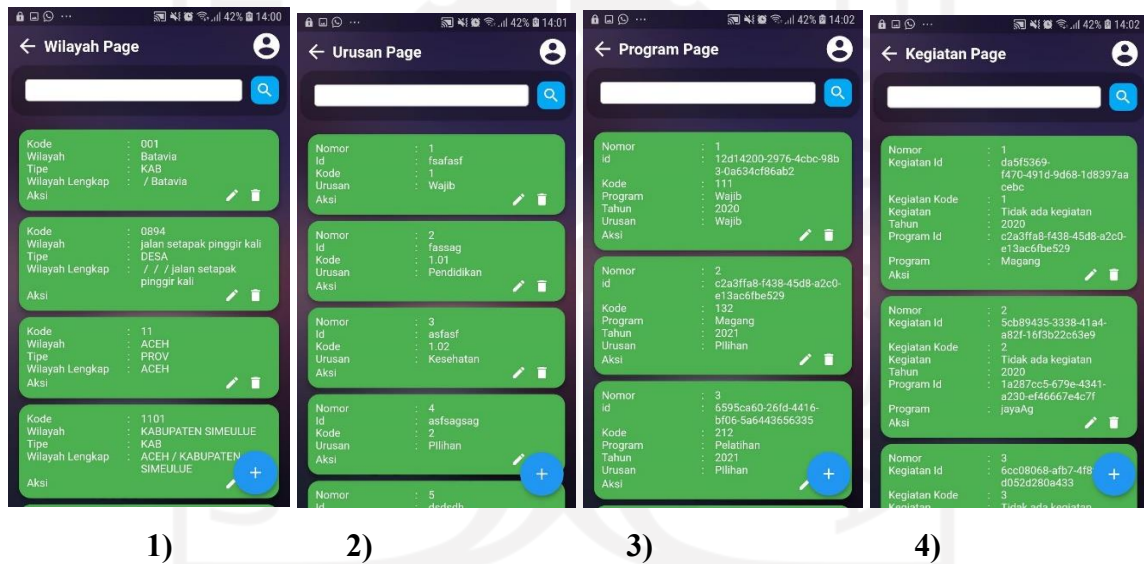
Kode	Program / Sub Kegiatan	Lokasi	Keluaran	Hasil	Pagu	Bobot (%)	Sub Unit Organisasi SKPD
Jumlah DPA					42		
Total Pagu					31.053.890.479		
4.01.10	PROGRAM PENYELENGGARAAN KEISTIMAWAAN URUSAN KELEMBAGAAN DAN KETATALAKSANAAN				330.076.000		
4.01.10.1.02.07	Pembekalan Keistimewaan bagi PNS Mutasi Luar Daerah		Jumlah ASN Mutasi yang Memahami Keistimewaan DIY : 60 orang		158.976.000	48,16	-
4.01.10.1.07.01	Pengadaan Sarana dan Prasarana Lembaga Pelaksana Urusan Keistimewaan		Camera Web Cam : 3 unit Laptop : 3 unit Scanner : 2 unit		171.100.000	51,84	-
5.03.01 PROGRAM PENUNJANG URUSAN PEMERINTAHAN DAERAH PROVINSI					19.711.489.229		
5.03.01.1.01.01	Penyusunan Dokumen Perencanaan Perangkat Daerah		Dokumen Perencanaan OPD Tahun Anggaran 2022 dan Perubahan Tahun Anggaran 2021 : 10 dokumen Kesepakatan Hasil Forum Perencanaan OPD : 1 BA		73.906.000	0,37	-
5.03.01.1.01.07	Evaluasi Kinerja Perangkat Daerah		Laporan Kinerja Instansi Pemerintah (LKjIP) yang Capaian Kinerja sesuai dengan Dokumen Perencanaan (RPJMD, Renstra dan IKU) (Nilai LKjIP A) : 1 laporan Realisasi Capaian Fisik dan Keuangan sesuai Tahapan ROPK : 12 laporan		28.868.000	0,15	-

Gambar 3.46 Laporan Dokumen Pelaksanaan Anggaran

Halaman Wilayah, Urusan, Program, Kegiatan, dan Sub Kegiatan

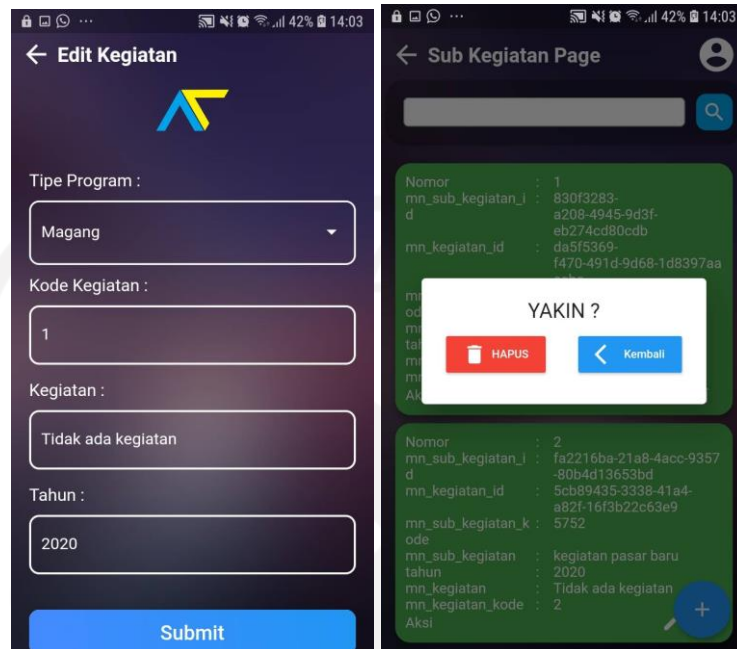
Halaman yang sudah dibuat, yaitu: Halaman Wilayah yang dapat dilihat pada **Gambar 3.47**, Urusan yang dapat dilihat pada **Gambar 3.48**, Program yang dapat dilihat pada **Gambar 3.49**, Kegiatan yang dapat dilihat pada **Gambar 3.50**, dan Sub Kegiatan yang dapat dilihat pada **Gambar 3.51**. Dalam halaman tersebut terdapat fitur-fitur untuk mencari, menampilkan, menambah, mengubah, dan menghapus data sesuai dengan judul halaman. Pengguna dapat

mencari data tersebut dengan cara mengisi formulir yang berada pada sebelah tombol pencarian. Selanjut, pengguna dapat mengubah data dengan cara menekan tombol yang ber-*icon* pen sehingga pengguna akan dibawa menuju halaman untuk mengubah data. Salah satu halaman untuk mengubah data dapat dilihat pada **Gambar 3.52**. Pengguna juga dapat menghapus salah satu data dengan cara menekan tombol yang ber-*icon* *trash*. Sehingga, pengguna akan ditampilkan *pop up* yang berisi tombol Hapus dan Kembali. Salah satu *Pop up* tersebut dapat dilihat pada **Gambar 3.53**. Selanjutnya, pengguna yang ingin menambahkan data diharuskan untuk menekan tombol tambah yang akan membawa pengguna ke halaman menambahkan data. Salah satu halaman yang dapat menambahkan data dapat dilihat pada **Gambar 3.54**. Data yang dikirim seperti data untuk mencari, dan menambahkan, mengedit, dan menghapus juga akan dikirim ke server ATGOV. Data yang ditampilkan juga berasal dari server ATGOV melalui perantara REST API.



5)

1) merupakan **Gambar 3.47** yaitu Halaman Wilayah, 2) merupakan **Gambar 3.48** yaitu Halaman Urusan, 3) merupakan **Gambar 3.49** yaitu Halaman Program, 4) merupakan **Gambar 3.50** yaitu Halaman Kegiatan, 5) merupakan **Gambar 3.51** yaitu Halaman Sub Kegiatan



Gambar 3.52 Halaman Edit Kegiatan (Kiri) dan **Gambar 3.53.** *Pop up* Halaman Sub Kegiatan (Kanan).

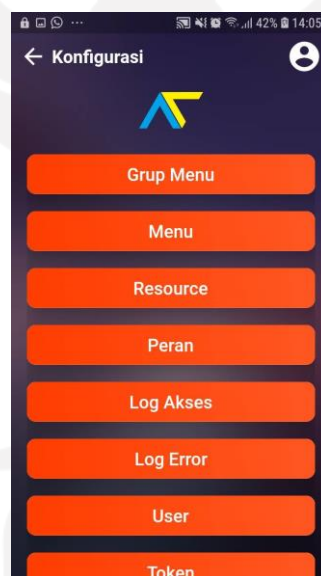


Gambar 3.54 Halaman Tambah Program

3.2.3.5 Halaman Konfigurasi ATGOV

Halaman Konfigurasi akan muncul ketika menekan menu Konfigurasi yang ada pada halaman Menu Utama ATGOV. Menu Konfigurasi ini berfungsi sebagai menu bantuan dari sisi aplikasi ATGOV, seperti: mengatur data user, melihat akses, melihat *error*, dan lain sebagainya. Halaman tersebut dapat dilihat pada **Gambar 3.55**. Halaman ini berupa menu-menu Konfigurasi, yaitu:

1. Grup Menu
2. Menu
3. Resource
4. Peran
5. Log Akses
6. Log Error
7. User
8. Token

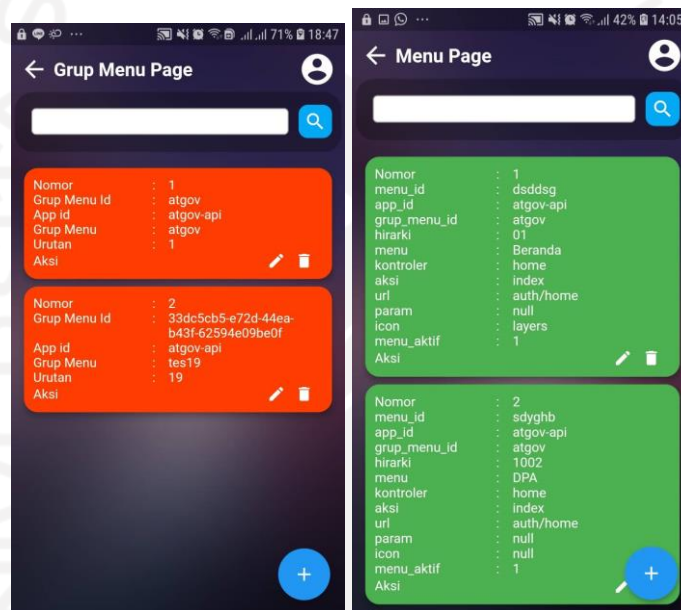


Gambar 3.55 Halaman Konfigurasi

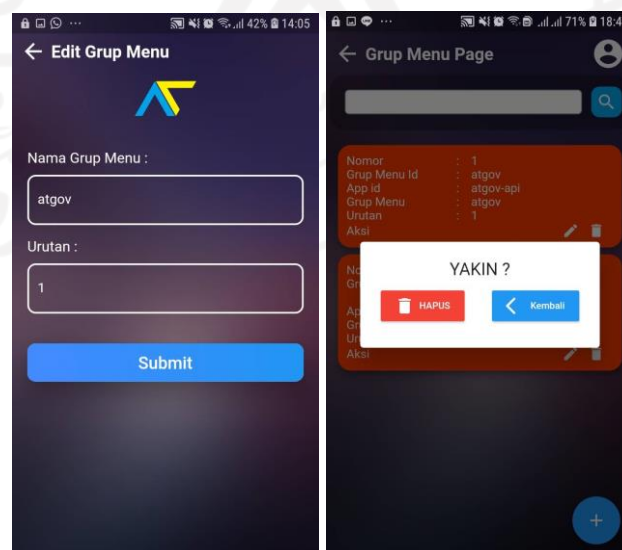
Halaman Grup Menu, Menu

Halaman yang sudah dibuat, yaitu: Halaman Grup Menu yang dapat dilihat pada **Gambar 3.56** dan Menu yang dapat dilihat pada **Gambar 3.57**. Dalam halaman tersebut terdapat fitur-fitur untuk mencari, menampilkan, menambah, mengubah, dan menghapus data sesuai dengan judul halaman. Pengguna dapat mencari data tersebut dengan cara mengisi formulir yang berada pada sebelah tombol pencarian. Selanjut, pengguna dapat mengubah data dengan cara menekan tombol yang ber-*icon* pen sehingga pengguna akan dibawa menuju halaman untuk

mengubah data. Salah satu halaman untuk mengubah data dapat dilihat pada **Gambar 3.58**. Pengguna juga dapat menghapus salah satu data dengan cara menekan tombol yang ber-*icon trash*. Sehingga, pengguna akan ditampilkan *pop up* yang berisi tombol Hapus dan Kembali. Salah satu *Pop up* tersebut dapat dilihat pada **Gambar 3.59**. Selanjutnya, pengguna yang ingin menambahkan data diharuskan untuk menekan tombol tambah yang akan membawa pengguna ke halaman menambahkan data. Salah satu halaman yang dapat menambahkan data dapat dilihat pada **Gambar 3.60**. Data yang dikirim seperti data untuk mencari, dan menambahkan, mengedit, dan menghapus juga akan dikirim ke server ATGOV. Data yang ditampilkan juga berasal dari server ATGOV melalui perantara REST API.



Gambar 3.56 Halaman Grup Menu (Kiri) dan **Gambar 3.57** Halaman Menu (Kanan).



Gambar 3.58 Halaman Edit Grup Menu (Kiri) dan **Gambar 3.59** *Pop up* Halaman Grup Menu (Kanan).

Gambar 3.60 Halaman Tambah Grup Menu

3.2.4 ASN Memayu

Proyek terakhir yang dikerjakan adalah ASN Memayu. Proyek ini dikembangkan untuk meningkatkan pengelolaan manajemen kepegawaian di Badan Kepegawaian dan Diklat Daerah Provinsi Yogyakarta (BKDD Yogyakarta). Aplikasi ASN Memayu dibangun dalam platform web dan mobile. Adapun halaman aplikasi *mobile* ASN Memayu yang sudah dibuat pada saat magang dapat dilihat pada **Bab 3.2.4.1**, **Bab 3.2.4.2**, **Bab 3.2.4.3**, **Bab 3.2.4.4**, dan **Bab 3.2.4.5**.

3.2.4.1 Halaman Login ASN Memayu

Halaman Login adalah halaman yang mempunyai fitur *splashscreen* dan *login*. Fitur *splashscreen* merupakan fitur yang bermanfaat sebagai penanda unik sebuah aplikasi. Fitur ini akan menghilang selama waktu yang ditentukan oleh developer dan akan langsung menuju ke fitur *login*, dapat dilihat pada **Gambar 3.61**. Fitur *login* tersebut merupakan fitur yang mengatur proses masuknya pengguna ke dalam halaman Menu Utama aplikasi ASN Memayu, dapat dilihat pada **Gambar 3.62**. Dalam halaman tersebut terdapat formulir yang harus di isi pengguna, yaitu: *username* dan *password*. Pengguna dapat menekan tombol Sign In ketika sudah mengisi semua formulir dan formulir tersebut akan di kirim ke *database server* melalui perantara REST API. Formulir yang diisi pengguna harus sama dengan yang ada pada data pengguna di *database server* ASN Memayu. Dalam hal ini, jika pengguna memasukkan formulir yang berbeda,

pengguna tidak dapat masuk ke halaman Menu Utama aplikasi *mobile* ASN Memayu. Setelah melakukan *login*, data-data pengguna akan di simpan ke *database local* yang bernama Shared Preferences. Dengan Shared Preferences, Pengguna yang sudah melakukan *login* tidak perlu melakukannya kembali hingga menekan fitur *logout*. Fitur *logout* berfungsi untuk menghapus data-data pengguna ketika *login*.

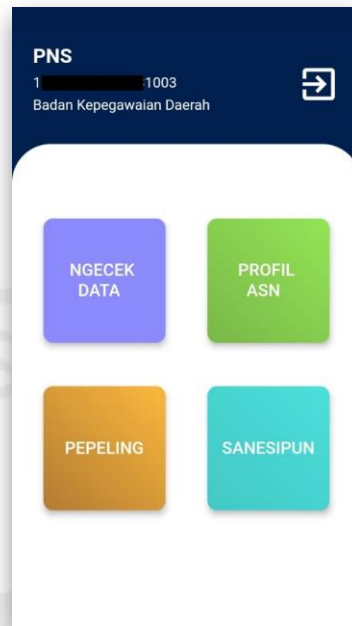


Gambar 3.61 *splash* screen (Kiri) dan **Gambar 3.62** Halaman Login (Kanan)

3.2.4.2 Halaman Menu Utama

Ketika pengguna sudah melakukan *login* maka akan masuk ke halaman Menu Utama ASN Memayu, dapat dilihat pada **Gambar 3.63**. Dalam halaman ini terdapat data *login* pengguna dan menu-menu utama aplikasi ASN Memayu. Data yang ditampilkan adalah Status Kepegawaian pengguna, Nomor Identitas Pegawai Negeri Sipil pengguna, dan Unit Organisasi pengguna. Data tersebut berasal dari *database server* ASN Memayu. Pengguna juga dapat melakukan *logout* dengan cara menekan *icon exit_to_app*. Dengan melakukan *logout*, pengguna harus melakukan *login* kembali untuk menuju halaman Menu Utama. Selanjutnya, Terdapat empat menu utama yang ada pada halaman Menu Utama, yaitu:

1. Ngecek Data
2. Profil ASN
3. Pepeling
4. Sanesipun

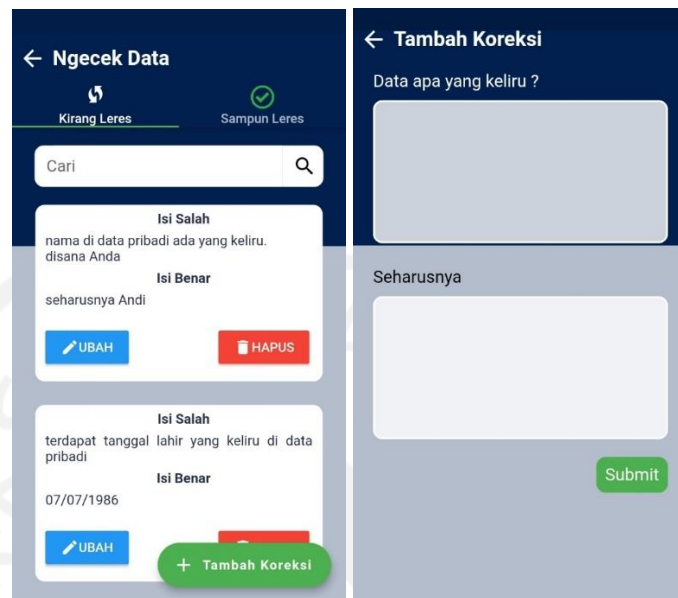


Gambar 3.63 Halaman Menu Utama ASN Memayu

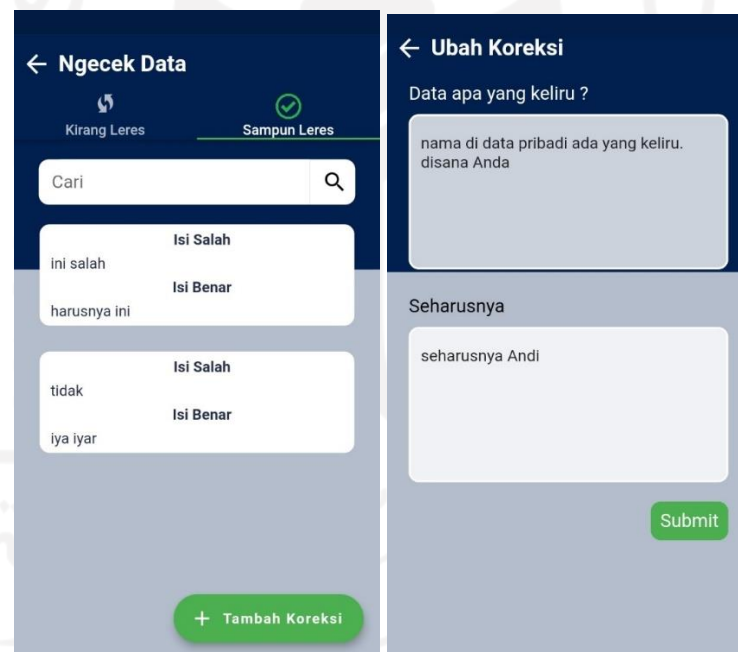
3.2.4.3 Halaman Ngecek Data

Halaman Ngecek Data akan muncul ketika menekan menu Ngecek Data di halaman Menu Utama ASN Memayu. Terdapat tiga menu di halaman Ngecek Data, yaitu: Kirang Leres, Sampun Leres, dan Tambah Koreksi. Ketika menekan menu Kirang Leres maka tampilan akan seperti pada **Gambar 3.64**. Menu Kirang Leres merupakan menu yang menampilkan data yang kurang benar. Data tersebut bisa berupa kesalahan penulisan data pengguna yang ada pada aplikasi. Pengguna dapat menambahkan data yang kurang benar dengan cara menekan menu Tambah Koreksi, dapat dilihat pada **Gambar 3.65**. Selanjutnya, terdapat menu Sampun Leres. Dalam menu tersebut data yang kurang benar sudah diperbaiki oleh pihak Admin ASN Memayu, dapat dilihat pada **Gambar 3.66**. Perbedaan yang paling menonjol antara menu Kirang Leres dan menu Sampun leres adalah tidak adanya tombol Ubah dan Hapus. Tombol Ubah berfungsi untuk mengubah koreksi yang sudah dikirim pengguna. Setelah menekan tombol Ubah, pengguna akan masuk ke dalam halaman Ubah Koreksi, dapat dilihat pada **Gambar 3.67**. Selanjutnya, Tombol Hapus berfungsi untuk menghapus koreksi yang sudah dikirim oleh pengguna. Setelah menekan tombol Hapus, aplikasi akan menampilkan *pop up* yang dapat dilihat pada **Gambar 3.68**. *Pop up* tersebut bertujuan untuk meyakinkan pengguna dalam menghapus data koreksinya. Semua data yang ditampilkan berasal dari *database server* ASN Memayu. Data yang ada pada formulir

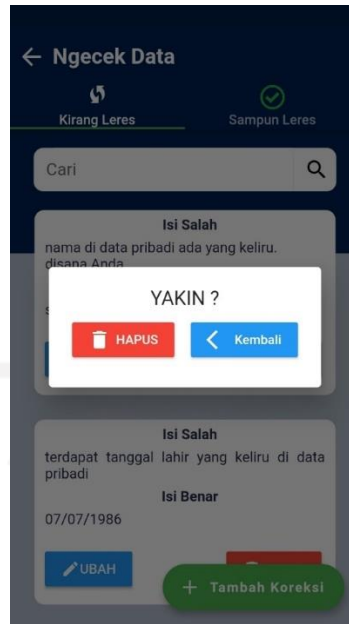
halaman Tambah Koreksi dan Ubah Koreksi dapat dikirimkan juga oleh pengguna ke *database server* ASN Memayu.



Gambar 3.64 Kirang Leres (Kiri) dan **Gambar 3.65** Tambah Koreksi (Kanan)



Gambar 3.66 Sampun Leres (Kiri) dan **Gambar 3.67** Halaman Ubah Koreksi (Kanan)



Gambar 3.68 *Pop up* hapus Koreksi

3.2.4.4 Halaman Profil ASN

Halaman Profil ASN akan muncul ketika menekan menu Profil ASN di halaman Menu Utama ASN Memayu. Halaman ini dapat dilihat pada **Gambar 3.69**. Dalam halaman Profil ASN ini terdapat tujuh menu Profil, yaitu:

1. Data Pribadi yang berisi data-data pribadi, dapat dilihat pada **Gambar 3.70**.
2. Riwayat Uji Kompetensi.
3. Riwayat SKP (Sasaran Kerja Pegawai), dapat dilihat pada **Gambar 3.71**
4. Riwayat Keluarga/Pribadi, dapat dilihat pada **Gambar 3.72**.
5. Riwayat Prestasi dapat dilihat pada **Gambar 3.73**.
6. Riwayat Hukuman, dapat dilihat pada **Gambar 3.74**.
7. Riwayat Pendidikan, dapat dilihat pada **Gambar 3.75**.
8. Riwayat Jabatan, dapat dilihat pada **Gambar 3.76**.
9. Riwayat Pangkat, dapat dilihat pada **Gambar 3.77**.

Semua data yang ditampilkan berasal dari *database server* ASN Memayu dan sebagian gambar terdapat kotak hitam untuk menutupi identitas data pengguna.



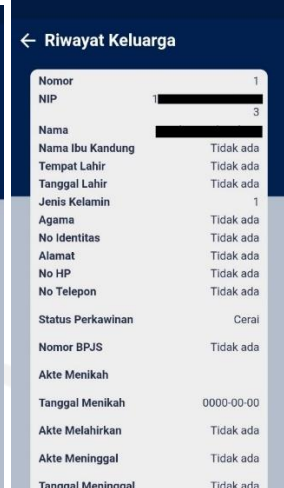
1)



2)



3)



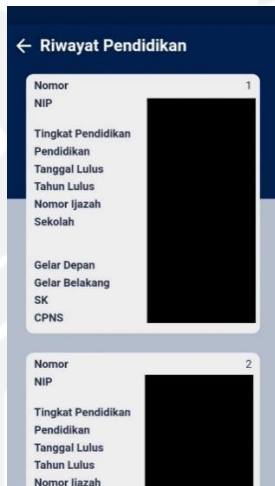
4)



5)



6)



7)



8)



9)

1) merupakan **Gambar 3.69** yaitu Halaman Profil ASN, 2) merupakan **Gambar 3.70** yaitu Halaman Data Pribadi, 3) merupakan **Gambar 3.71** yaitu Halaman Riwayat SKP, 4) merupakan **Gambar 3.72** yaitu Halaman Riwayat Keluarga, 5) merupakan **Gambar 3.73** yaitu Halaman Riwayat Prestasi, 6) merupakan **Gambar 3.74** yaitu Halaman Riwayat Hukuman, 7) merupakan **Gambar 3.75** yaitu Halaman Riwayat Pendidikan, 8) merupakan **Gambar 3.76** Halaman Riwayat Jabatan, dan 9) merupakan **Gambar 3.77** yaitu Halaman Riwayat Pangkat

3.2.4.5 Halaman Pepeling

Halaman Pepeling akan muncul ketika menekan menu Pepeling di halaman Menu Utama ASN Memayu. Dalam halaman Pepeling ini terdapat enam menu Pepeling, yaitu:

1. Notifikasi Kehadiran
2. Kenaikan Pangkat
3. Hukuman Disiplin
4. Kenaikan Gaji Berkala
5. Pensiun
6. Pengembangan Kompetensi

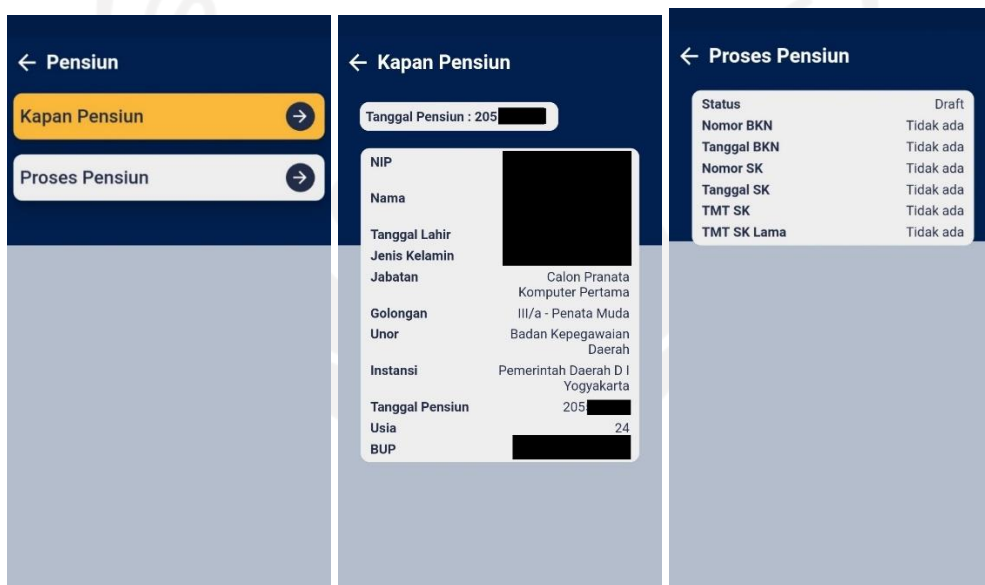
Halaman ini dapat dilihat pada **Gambar 3.78**. Menu yang sudah dibuat di halaman Pepeling ini adalah menu Pensiun dan Kenaikan Gaji Berkala.



Gambar 3.78 Halaman Pepeling

Halaman Pensiun

Halaman Pensiun akan muncul ketika menekan menu Pensiun di halaman Pepeling ASN Memayu, dapat dilihat pada **Gambar 3.79**. Terdapat dua menu pada halaman ini, yaitu Kapan Pensiun dan Proses Pensiun. Menu Pensiun merupakan menu yang akan memindahkan pengguna ke halaman Kapan Pensiun, dapat dilihat pada **Gambar 3.80**. Selanjut menu Proses Pensiun, menu ini akan memindahkan pengguna ke halaman Proses Pensiun yang dapat dilihat pada **Gambar 3.81**. Pengguna juga dapat menerima notif Pensiun ketika membuka halaman Menu Utama dengan waktu yang ditentukan oleh developer, yaitu: 2 tahun sebelum pensiun dan 1 tahun sebelum pensiun. Notif tersebut dapat dilihat pada **Gambar 3.82**.



Gambar 3.79 Halaman Pensiun (Kiri), **Gambar 3.80** Halaman Kapan Pensiun (Tengah), dan **Gambar 3.81** Halaman Proses Pensiun (Kanan)



Gambar 3.82 Notifikasi Pensiun

Halaman Kenaikan Gaji Berkala

Halaman Kenaikan Gaji Berkala akan muncul ketika menekan menu Pensiun di halaman Pepeling ASN Memayu, dapat dilihat pada **Gambar 3.83**. Dalam halaman ini terdapat fitur-fitur, yaitu: jumlah gaji, pencarian gaji, dan menampilkan data gaji. Jumlah gaji ditampilkan dengan cara menghitung semua data gaji pengguna. Selanjutnya terdapat fitur pencarian, pengguna dapat melakukan pencarian berdasarkan Kode Gol dengan cara mengisi formulir lalu menekan *icon search*. Formulir yang diisi oleh pengguna akan dikirimkan ke *database server*

ASN Memayu. Data yang ditampilkan pada halaman ini juga merupakan data yang diperoleh dari *database server* ASN Memayu.

← Kenaikan Gaji Berkala

Jumlah Gaji : Rp 1.743.676.500

Cari

Kode Gol	11
Masa Kerja	0
Golongan	
Gaji	Rp 1.560.800

Kode Gol	11
Masa Kerja	1
Golongan	
Gaji	Rp 1.560.800

Kode Gol	11
Masa Kerja	2
Golongan	
Gaji	Rp 1.610.000

Kode Gol	11
Masa Kerja	3
Golongan	

Gambar 3.83 Halaman Kenaikan Gaji Berkala

BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Teknis

Selama enam bulan magang terdapat hal yang dapat dipelajari pada Evaluasi Teknis, yaitu: Perbedaan antara Flutter dengan Android Native, Struktur folder dengan konsep BLoC Pattern, Dokumentasi REST API yang diberikan oleh perusahaan, Desain yang diberikan dan disarankan oleh perusahaan, dan Penggabungan widgets dalam Flutter. Dengan hal tersebut, pembaca dapat mengetahui sebagian teknis yang ada pada dunia kerja di perusahaan CV. Atsoft Teknologi yang berposisi sebagai *Front End Mobile Developer*. Hal tersebut akan dijelaskan pada sub bab berikutnya.

4.1.1 Perbedaan antara Flutter dengan Android Native

Pada sub bab ini akan dijelaskan perbedaan antara Flutter dengan Android Native. Flutter telah dipelajari selama magang bersamaan dengan mengerjakan proyek yang ada pada perusahaan CV. Atsoft Teknologi. Sementara, Android Native telah dipelajari pada mata kuliah Pengembangan Aplikasi Bergerak di kampus. Perbedaan tersebut dapat diketahui dari cara pengembangannya, yaitu:

Tahap instalasi pada Flutter dan Android Native

Tabel 4.1 merupakan tabel yang memiliki tahap-tahap untuk instalasi pada Flutter dan Android Native. Pada tabel tersebut terlihat bahwa tahap Flutter lebih banyak dibandingkan dengan Android Native. Pada tahap Flutter terdapat tahap untuk memasang Visual Studio Code (VSC) dan mengunduh Flutter serta *extensions* atau *plugins*. Flutter dapat dikembangkan melalui VSC dan Android Studio tergantung yang diinginkan oleh developernya. Penggunaan memori fisik atau biasa disebut *Random Access Memory (RAM)* pada Visual Studio Code sebanyak 199,4 MB dengan membuka 9 *sourcecode* dan Android Studio sebanyak 1441,8 MB dengan membuka 7 *sourcecode* yang dapat dilihat pada **Gambar 4.1** Penggunaan RAM dapat berpengaruh terhadap kinerja komputer. RAM dengan spesifikasi yang bagus akan mendukung kelancaran pembuatan video (Purnama, 2013 dalam Taufiq Kurniawan, 2018). Pembuatan video dapat dikaitkan dengan pembuatan aplikasi *mobile* sehingga dapat disimpulkan bahwa Flutter mendukung developer untuk memudahkan pembuatan proyek aplikasinya.

Table 4.1 Tahap Instalasi pada Flutter dan Android Native

No.	Tahap instalasi pada Flutter	Tahap instalasi pada Android Native
1.	Memasang <i>Java Development Kit (JDK)</i> pada komputer	Memasang <i>Java Development Kit (JDK)</i> pada komputer
2.	Memasang Android Studio	Memasang Android Studio
3.	Memasang Visual Studio Code	Membuat proyek aplikasi <i>mobile</i> pada Android Studio
4.	Mengunduh Flutter	Menjalankan aplikasi dengan <i>device</i> atau Emulator
5.	Memasang Extensions Flutter di Visual Studio Code atau Plugins Flutter di Android Studio	Menjalankan aplikasi dengan <i>device</i> atau Emulator
6.	Membuat proyek SIM-RS At Turrots pada Visual Code atau Android Studio	
7.	Menjalankan aplikasi dengan <i>device</i> atau Emulator	

The screenshot shows the Windows Task Manager Performance tab. The 'Memory' section is expanded, showing a total usage of 61%. Below this, a table lists the memory usage for several applications:

Name	Status	CPU	Memory	Disk	Network
> Android Studio (7)		1,3%	1.441,8 MB	0,1 MB/s	0 Mbps
> Google Chrome (32)		3,0%	796,0 MB	0,1 MB/s	0,4 Mbps
Google Chrome		38,5%	212,0 MB	0 MB/s	0 Mbps
> Visual Studio Code (9)		0,1%	199,4 MB	0 MB/s	0 Mbps

Gambar 4.1 Penggunaan *memory* RAM pada Android Studio dan Visual Studio Code

Penggunaan memori *hardisk* dan *device* pada proyek Flutter dan Android Native

Penggunaan tempat menyimpan data komputen pada proyek Flutter lebih besar dibandingkan dengan Android Native. Dalam folder flutter terdapat modul untuk mengkonfigurasi aplikasi *cross-platform*, yaitu: Android, iOS, dan Web. Pada **Gambar 4.2 dan 4.3** adalah gambar yang menunjukkan perbedaan memori yang dipakai pada saat mengunduh Android Native dan Flutter di komputer. Android Native memakan memori sebesar 31,2 MB sedangkan untuk Flutter memakan memori sebesar 521 MB. Hal ini dapat berpengaruh terhadap aplikasi pada saat di *install* pada *device* pengguna. Developer perlu memikirkan kembali terhadap performa aplikasinya yang dapat berpengaruh pada memori internal *device* pengguna.

Location:	E:\Kuliah\Semester 7\Proyek untuk tugas akhir\Android
Size:	30.2 MB (31.668.804 bytes)
Size on disk:	31.2 MB (32.727.040 bytes)
Contains:	682 Files, 359 Folders

Gambar 4.2 Android Native

Location:	E:\Kuliah\Semester 7\Proyek untuk tugas akhirFlutter
Size:	520 MB (545.796.582 bytes)
Size on disk:	521 MB (546.635.776 bytes)
Contains:	1.057 Files, 396 Folders

Gambar 4.3 Flutter

Komponen dasar untuk menampilkan kumpulan data dalam bentuk *list*

Komponen dasar tersebut sangatlah berbeda antara Flutter dengan Android Native. Flutter hanya perlu memanggil *widget* `ListView` pada kelas untuk menampilkan antarmuka pengguna, dapat dilihat pada **Gambar 4.4**. Setelah itu, developer dapat mengisi *variable* bertipe *list* dan mengkonfigurasikannya terhadap parameter yang ada pada *widget* tersebut.

```

|ListView.builder(
padding: EdgeInsets.zero,
scrollDirection: Axis.vertical,
shrinkWrap: true,
itemCount: searchWilayahLoaded.listWilayahModel.length,
itemBuilder: (context, index) => (index <
    searchWilayahLoaded.listWilayahModel.length)
    ? Container(
      child: WilayahItemWidget(
        wilayahModel:
          searchWilayahLoaded.listWilayahModel[index],
        backgroundColor: Colors.green,
        token: tokenLogged,
      )) // WilayahItemWidget, Container

```

Gambar 4.4 *Widget* `ListView`

Sedangkan pada Android Native, developer perlu membuat XML (*Extensible Markup Language*), Activity, dan Adapter. XML adalah bahasa *markup* seperti HTML untuk mendesain antarmuka pengguna. Dalam XML ini, developer perlu memanggil komponen *RecyclerView* untuk menampilkan *variable* yang bertipe *list*. Komponen *RecyclerView* dapat dilihat pada **Gambar 4.5**. Activity berfungsi sebagai tempat untuk menampilkan *RecyclerView* pada antarmuka pengguna. Adapter berfungsi untuk menentukan data yang akan ditampilkan oleh *RecyclerView*, dapat dilihat pada **Gambar 4.6**. Sehingga dapat disimpulkan, developer dapat mengembangkan aplikasi *mobile* dengan Flutter menjadi lebih cepat dibandingkan dengan Android Native.

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rv_heroes"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:listitem="@layout/item_cardview_wilayah" />

```

Gambar 4.5 Komponen *RecyclerView*

```

class CardViewWilayahAdapter(private val listWilayah: ArrayList<Hero>) : RecyclerView.Adapter<CardViewWilayahAdapter.CardViewViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): CardViewViewHolder {
        val view: View = LayoutInflater.from(parent.context).inflate(R.layout.item_cardview_wilayah, parent, attachToRoot: false)
        return CardViewViewHolder(view)
    }
    override fun onBindViewHolder(holder: CardViewViewHolder, position: Int) {
    }
    override fun getItemCount(): Int {
        return listWilayah.size
    }
    inner class CardViewViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        var tvName: TextView = itemView.findViewById(R.id.tv_item_name)
    }
}

```

Gambar 4.6 Adapter

Pemanggilan data dari REST API

Pemanggilan data Tipe Wilayah dari REST API menggunakan Flutter dapat dilihat pada **Gambar 4.7** sedangkan pada Android Native dapat dilihat pada **Gambar 4.8**. Pada Android Native terdapat pemanggilan data yang lebih kompleks dibandingkan dengan Flutter. Pemanggilan yang lebih kompleks tersebut dapat dilihat pada fungsi `onSuccess` dan `onFailure`. Hal ini memungkinkan untuk menjaga keamanan ketika mendapatkan beberapa kekeliruan oleh pihak developer. Developer Flutter perlu mendeklarasikan secara manual apabila terdapat kesalahan pada *server*. Sedangkan pada Android Native, developer sudah disediakan dua fungsi ketika melakukan pemanggilan data dari REST API, yaitu: `onSuccess` dan `onFailure`. Fungsi `onSuccess` merupakan fungsi yang dipanggil ketika data dapat diakses melalui *server* dan sebaliknya juga pada fungsi `onFailure`.

```

Future<Null> getListWilayah() async {
  String url = 'https://dev.atsoft.co.id/atgov-api-sumsel/cr_wilayah/get_list_tipe?token=a928393129312312';
  var response = await http.get(url);
  var jsonObject = json.decode(response.body);

  List<dynamic> listTipeWilayah = (jsonObject as Map<String, dynamic>) ['data'];
  List<TipeWilayahModel> listTipeWilayahModel = [];

  for(int i = 0; i < listTipeWilayah.length; i++)
    listTipeWilayahModel.add(TipeWilayahModel.createdTipeWilayah(listTipeWilayah[i]));

  setState(() {});
}

```

Gambar 4.7 Pemanggilan data Tipe Wilayah dari REST API menggunakan Flutter

```

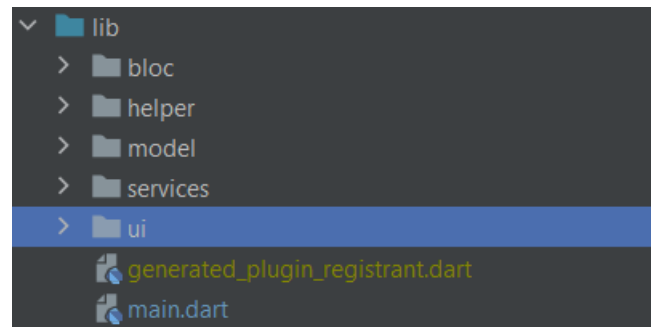
private fun getListWilayah() {
  val client = AsyncHttpClient()
  val url = "https://dev.atsoft.co.id/atgov-api-sumsel/cr_wilayah/get_list_tipe?token=a928393129312312";
  client.get(url, object : AsyncHttpResponseHandler() {
    override fun onSuccess(statusCode: Int, headers: Array<Header>, responseBody: ByteArray) {
      val listWilayah = ArrayList<TipeWilayahModel>()
      val result = String(responseBody)
      try {
        val jsonArray = JSONObject(json: "data").getJSONArray(result)
        var tipeWilayahModel = TipeWilayahModel();
        for (i in 0 until jsonArray.length()) {
          val jsonObject = jsonArray.getJSONObject(i)
          tipeWilayahModel.tipe = jsonObject.getString(name: "tipe")
          tipeWilayahModel.tipe_label = jsonObject.getString(name: "tipe_label")
          listWilayah.add(tipeWilayahModel)
        }
      } catch (e: Exception) {
        e.printStackTrace()
      }
    }
    override fun onFailure(statusCode: Int, headers: Array<Header>, responseBody: ByteArray, error: Throwable) {
      val errorMessage = when (statusCode) {
        401 -> "$statusCode : Bad Request"
        403 -> "$statusCode : Forbidden"
        404 -> "$statusCode : Not Found"
        else -> "$statusCode : ${error.message}"
      }
    }
  })
}

```

Gambar 4.8 Pemanggilan data Tipe Wilayah dari REST API menggunakan Android Native

4.1.2 Struktur folder dengan konsep BLoC Pattern

Struktur folder yang dibuat ketika developer memakai konsep BLoC Pattern, yaitu: Bloc, dan *User Interface* (UI). Folder Bloc berisi kelas Bloc, State, dan Event yang berfungsi untuk memisahkan logika dari kelas yang akan menampilkan tampilan *User Interface*. Bloc tersebut dapat digunakan oleh kelas *User Interface* yang ada pada proyek ASN Memayu. Terakhir *User Interface*, folder ini berisi kelas *User Interface* yang akan menampilkan dan membuat tampilan antarmuka pengguna. Struktur tersebut dapat dilihat pada **Gambar 4.9**.



Gambar 4.9 Struktur folder dengan konsep BLoC Pattern

BLoC Pattern membantu memudahkan para developer dalam memperbaharui proyek aplikasinya. Struktur folder yang ada pada *pattern* tersebut sudah dikenal oleh developer yang mempelajarinya. Selain itu, developer dapat mudah dalam melakukan *testing* pada fungsi yang ada di folder Bloc untuk memastikan bahwa fungsi tersebut tidak terdapat *bug* dan *error*. Terdapat beberapa keuntungan yang diketahui ketika memakai *design pattern*. Pertama, *Design pattern* dapat mengurangi waktu pengembangan karena *design pattern* tersebut meningkatkan kecepatan pengembangan maupun pengujian yang dikembangkan dari pengalaman ahlinya (CH Chang, 2011 dalam Subburaj Ramasamy, 2015). Terakhir, *Design Pattern* dapat meningkatkan struktur perangkat lunak, kecepatan meningkatkan implementasi, menyederhanakan perawatan, dan membantu menghindari pergeseran arsitektur (W. Tichy, 2011 dalam Subburaj Ramasamy, 2015).

4.1.3 Dokumentasi REST API yang diberikan oleh perusahaan

Dokumentasi REST API yang diberikan oleh perusahaan dapat dilihat pada **Gambar 4.10**. Dokumentasi tersebut disajikan dalam *file* yang berbentuk *Excel*. Pada *file* tersebut, terdapat tabel dokumentasi REST API. *Field* yang ada pada tabel tersebut, yaitu: Method, Fungsi, Input Var, Input Var Optional, dan Keterangan. Dalam dokumentasi tersebut tidak ada petunjuk secara mendetail untuk pemakaian *EndPoint API*, seperti: penggunaan POST atau GET atau PUT, penggunaan Params, dan lain sebagainya. Sehingga, developer *Front End* perlu lebih banyak waktu untuk melakukan komunikasi dengan pihak *Back End*. Menurut Rahardan Apriadji pada tahun 2017, dokumentasi yang baik bagi *user* dan *developer* secara sederhana perlu berisi:

1. Keterangan yang jelas pada *method/resource*.
2. Parameter pada *method/resource* tersebut perlu menyertakan tipe, *special formatting*, *rules*, dan keterangan.
3. *Sample call* yang disertakan dengan sambungan ke media *type body*.
4. *Sample body* yang disertakan dengan sambungan ke media *type body*.

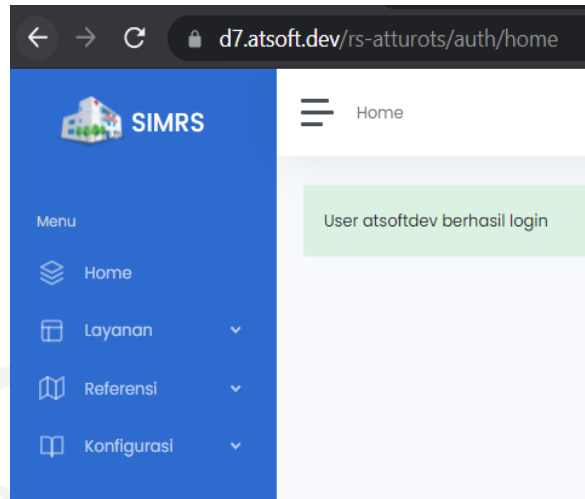
5. Contoh *code* untuk beberapa beberapa pemrograman.
6. Contoh SDK yang memperlihatkan cara untuk mengakses *method/resource*.
7. Tambahkan *link* ke *resource*.
8. *Comment section* agar *user* dapat membagikan dan mendiskusikan *code* yang ada Di dalamnya.

Method	Fungsi	Input Var	Input Var Optional	Keterangan
	ambil data unor dinas balai badan sekolah utp			
	input data device baru			
		code, vbaseos, hard, host, id, sup, tags, type,		Variabel yg diisi (kalo platform Android)
		model, iident, machine		Variabel yg diisi (kalo platform IOS)
	token	riwayat skn		

Gambar 4.10 Dokumentasi REST API

4.1.4 Desain yang diberikan dan disarankan oleh perusahaan

Desain yang diberikan oleh perusahaan masih dalam bentuk saran dan rujukan. Perusahaan belum mempunyai pekerja yang berposisi sebagai *UI/UX*. Sehingga pada saat mengerjakan proyek aplikasi, pekerja yang berposisi sebagai *FrontEnd Mobile Developer* harus terus mengimplementasikan dan merevisi tampilan proyeknya sampai klien menyetujui tampilan tersebut. Pekerja *UI/UX* sangatlah penting bagi perusahaan. Dengan adanya pekerja tersebut, pekerja yang berposisi sebagai *FrontEnd Mobile Developer* dapat fokus untuk menentukan struktur *file* aplikasi, mengembangkan berbagai macam *fitur*, memastikan tampilan aplikasi tersebut dibuat *responsive* agar dapat digunakan pada ukuran *device* manapun, membuat kode yang dapat dikembangkan oleh developer lain, dan memastikan tidak adanya *error* dan *bug* pada proyek aplikasinya. Dengan adanya pekerja tersebut juga, tampilan desain akan lebih mendekati yang diharapkan oleh klien karena dibuat menggunakan metode *design thinking* untuk mengerjakan tampilan desain tersebut. Dalam mengerjakan desain terdapat lima tahap *design thinking*, yaitu: *Empathize*, *Define*, *Ideate*, *Prototype*, dan *Testing* (Aditya Raka Pradana, 2021). Tampilan aplikasi *mobile* SIM-RS At Turrots mengikuti saran dari perusahaan dan desain tampilan *web* SIM-RS At Turrots yang sudah dibuat oleh perusahaan, dapat dilihat pada **Gambar 4.11**. Tampilan aplikasi *mobile* Atsoft E-Government dan ASN Memayu mengikuti saran dari perusahaan dan desain tampilan aplikasi *mobile* e-Prima yang sudah dibuat oleh perusahaan, dapat dilihat pada **Gambar 4.12**.



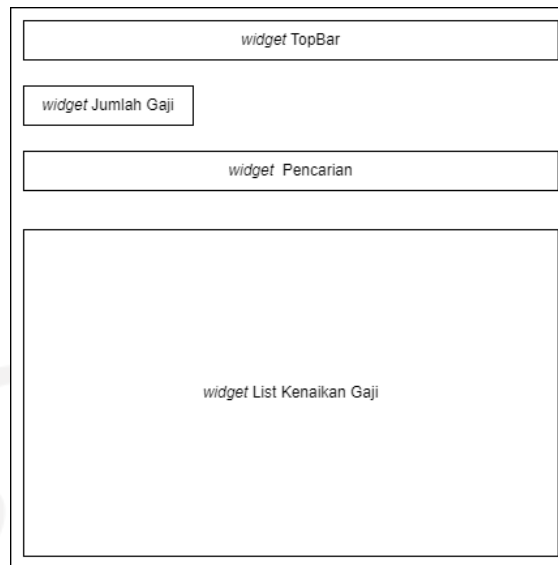
Gambar 4.11. Tampilan *web* SIM-RS At Turrots



Gambar 4.12 Tampilan *mobile* e-Prima

4.1.5 Penggabungan widgets dalam Flutter

Dalam sub bab ini akan diberikan contoh penggabungan *widgets* dalam halaman Kenaikan Gaji Berkala. *Widget-widget* tersebut dapat dilihat pada **Gambar 4.13**. Pada gambar tersebut terdapat *widget* TopBar, *widget* Jumlah gaji, *widget* Pencarian, dan *widget* List Kenaikan Gaji. *Widgets* tersebut lebih baik dari *widget* Build karena dapat memudahkan developer lain yang akan memperbaharui tampilan Kenaikan Gaji Berkala. *Widget* Build merupakan *widget* yang akan menampilkan semua *widgets* pada tampilan antarmuka pengguna halaman Kenaikan Gaji Berkala.



Gambar 4.13 *widgets* yang ada pada halaman Kenaikan Gaji Berkala

4.2 Non Teknis

Terdapat banyak hal non-teknis yang didapatkan selama enam bulan magang di perusahaan CV. Atsoft Teknologi (Atsoft), yaitu: mengetahui lingkungan perusahaan Atsoft dan hasil penilaian oleh mentor selama magang. Hal non-teknis ini bermanfaat sebagai bekal ketika sudah masuk dunia kerja. Penjelasan detail terkait hal non-teknis ini akan dijelaskan pada sub bab berikutnya.

4.2.1 Mengetahui lingkungan perusahaan

Lingkungan perusahaan Atsoft diketahui pada saat mentor menjelaskan alur pengerjaan tugas yang diberikan oleh mentor, pengalaman komunikasi dengan karyawan Atsoft, dan pengerjaan sebuah proyek pada perusahaan Atsoft. Sebelum mengikuti magang, pengetahuan alur pengerjaan tugas ketika mengerjakan sebuah proyek aplikasi *mobile* hanya terbatas pada fungsi tugas yang dikerjakan berjalan. Tetapi dalam sebuah perusahaan, pengerjaan tugas ketika mengerjakan sebuah proyek aplikasi *mobile* harus melewati tahap pengecekan yang dilakukan oleh mentor. Mentor akan melakukan pengecekan tugas dengan cara memastikan tugas yang dikerjakan berjalan dengan baik pada aplikasi di *smartphone*-nya. Selama magang berlangsung, komunikasi sangatlah penting untuk keberhasilan mengerjakan suatu proyek. Komunikasi bermanfaat untuk menjaga keharmonisan dan menghindari *misunderstanding*. Pemegang harus berkomunikasi dengan anggota lainnya maupun dengan mentor melalui aplikasi Discord. Sebagai contoh, pemegang harus menceritakan semua ketidaktahuan dan kesalahan ketika mengerjakan sebuah tugas yang diberikan oleh mentor. Dunia kerja memanglah berbeda dengan dunia kampus. Pengerjaan sebuah proyek tidak dapat dikerjakan secara langsung sebelum

mendapatkan tugas dari mentor. Pengerjaan sebuah proyek juga harus dikerjakan sesuai dengan yang diharapkan oleh perusahaan. Pemegang harus dapat mengatur waktu dengan baik karena tugas yang dikerjakan tidak hanya tugas dari perusahaan melainkan tugas dari kampus juga.

4.2.2 Hasil penilaian mentor selama magang

Dari hasil penilaian oleh mentor, proses magang telah dilalui dengan baik karena dapat mengerjakan *task* sesuai dengan yang diharapkan oleh perusahaan. Terdapat evaluasi oleh mentor mengenai proyek yang dikerjakan pada saat magang, yaitu: SIM-RS At Turrots, Atsoft E-Government, dan ASN Memayu. Evaluasi tersebut akan dijabarkan pada sub bab ini.

Proyek SIM-RS At Turrots

Mentor melakukan evaluasi terhadap aplikasi *mobile* SIM-RS At Turrots yang telah dikembangkan. Mentor menyimpulkan bahwa aplikasi *mobile* tersebut telah memenuhi tujuan yang diinginkan oleh perusahaan. Tujuan utama yang diinginkan oleh perusahaan Atsoft adalah mengembangkan aplikasi yang dapat mengolah data pasien dengan lebih cepat dibandingkan menggunakan aplikasi webnya. Dengan aplikasi *mobile*, manusia dapat membuka lebih cepat dibandingkan pada saat membuka aplikasi *web*-nya. Hal ini dibuktikan dengan aktivitas manusia yang lebih sering memakai *smartphone* pada aktivitas sehari-harinya. Ketika membuka *smartphone*, pertama kali yang manusia lihat adalah aplikasi yang telah diunduh olehnya. Aplikasi *mobile* SIM-RS At Turrots akan lebih cepat dibandingkan dengan aplikasi web SIM-RS At Turrots yang harus membuka aplikasi *browser* lalu membuka link dan melakukan konfigurasi seperti *zoom out/in* pada tampilan layarnya. Sedangkan pada aplikasi *mobile*, manusia tidak perlu melakukan hal tersebut karena aplikasi *mobile* sudah diatur agar fiturnya sesuai dengan aplikasi yang ada pada *smartphone*-nya. Manfaat pada aplikasi *mobile* SIM-RS At Turrots telah disimpulkan memudahkan pihak administrasi rumah sakit pada saat melakukan pendaftaran pasien yang sedang dalam kondisi darurat. Karena pada saat proses pembuatan aplikasi *mobile* tersebut, mentor melakukan pengecekan Halaman Kunjungan baru berjalan dengan baik pada *smartphone* mentor tersebut.

Proyek Atsoft E-Government (ATGOV)

Mentor melakukan evaluasi terhadap aplikasi *mobile* ATGOV yang telah dikembangkan. Mentor menyimpulkan bahwa aplikasi *mobile* tersebut telah memenuhi tujuan yang diinginkan oleh perusahaan. Tujuan utama yang diinginkan oleh perusahaan Atsoft adalah mengembangkan aplikasi yang dapat mempercepat pengolahan datanya dibandingkan menggunakan aplikasi webnya. Dengan aplikasi *mobile*, manusia dapat membuka lebih cepat dibandingkan pada saat

membuka aplikasi *web*-nya. Hal ini dibuktikan dengan aktivitas manusia yang lebih sering memakai *smartphone* pada aktivitas sehari-harinya. Ketika membuka *smartphone*, pertama kali yang manusia lihat adalah aplikasi yang telah diunduh olehnya. Aplikasi *mobile* ATGOV akan lebih cepat dibandingkan dengan aplikasi web ATGOV yang harus membuka aplikasi *browser* lalu membuka link dan melakukan konfigurasi seperti *zoom out/in* pada tampilan layarnya. Sedangkan pada aplikasi *mobile*, manusia tidak perlu melakukan hal tersebut karena aplikasi *mobile* sudah diatur agar fiturnya sesuai dengan aplikasi yang ada pada *smartphone*-nya. Manfaat pada aplikasi *mobile* ATGOV telah disimpulkan memberikan fleksibilitas waktu pengecekan data berulang. Karena pada saat proses pembuatan aplikasi *mobile* tersebut, mentor dan sebagian karyawan lainnya melakukan pengecekan Halaman Master Data berjalan dengan baik pada *smartphone*-nya.

Proyek ASN Memayu

Mentor melakukan evaluasi terhadap aplikasi *mobile* ASN Memayu yang telah dikembangkan. Mentor menyimpulkan bahwa aplikasi *mobile* tersebut telah memenuhi tujuan yang diinginkan oleh perusahaan. Tujuan utama yang diinginkan oleh perusahaan Atsoft adalah menambahkan fitur yang lebih khusus dibandingkan pada saat menggunakan aplikasi webnya. Pada aplikasi *mobile* ASN Memayu terdapat fitur khusus, yaitu: melaporkan data-data yang kurang tepat kepada Admin BKD DIY dan memberikan notifikasi pensiun pada saat 2 tahun dan 1 tahun sebelum masa pensiunnya pegawai. Aplikasi web ASN Memayu tidak mempunyai fitur-fitur tersebut. Manfaat pada aplikasi *mobile* ASN Memayu telah disimpulkan memudahkan para pegawai negeri sipil dalam mengakses kepegawaiannya serta memberikan fleksibilitas waktu pengecekan data berulang. Karena pada saat proses pembuatan aplikasi *mobile* tersebut, mentor dan sebagian karyawan lainnya melakukan pengecekan halaman-halaman yang sudah dikerjakan oleh pemegang berjalan dengan baik pada *smartphon*-nya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Selama enam bulan magang di perusahaan CV. Atsoft Teknologi, telah dikembangkan 3 aplikasi *mobile* untuk membantu menyelesaikan permasalahan 3 proyek yang ada pada perusahaan tersebut, yaitu: SIM-RS At Turrots, Atsoft E-Government, dan ASN Memayu. Nama ketiga aplikasi tersebut dibuat mirip dengan nama proyeknya. Semua tugas yang diberikan oleh mentor ketika mengerjakan ketiga proyek tersebut telah dikerjakan dengan baik sehingga tujuan dan manfaat pada laporan ini terpenuhi. Tujuan pada laporan ini dapat dilihat pada **Bab 1.3** sedangkan manfaat pada laporan ini dapat dilihat pada **Bab 1.4**. Selanjutnya, hasil aplikasi *mobile* yang telah dikerjakan selama magang dapat dilihat pada **Bab 3.2.2**, **Bab 3.2.3**, dan, **Bab 3.2.4**. Pada perusahaan ini, sistematika pembagian tugasnya berdasarkan fitur yang dikerjakan. Pemagang harus dapat menghasilkan satu fitur pada setiap pekannya. Terdapat hasil penilaian oleh mentor ketika mengerjakan ketiga proyek aplikasi *mobile* tersebut yang dapat memvalidasi bahwa tujuan dan manfaat pada laporan ini terpenuhi, dapat dilihat pada **Bab 4.2.2**.

5.2 Saran

Adapun saran terkait pelaksanaan magang yang ingin disampaikan kepada:

a. Perusahaan Magang

1. Dalam membuat dokumentasi REST API mungkin perlu dibuat lebih jelas. Terdapat sedikit penjelasan mengenai dokumentasi yang baik pada **Bab 4.1.3**. Dengan dokumentasi tersebut mungkin akan mengurangi komunikasi antara *frontend* dengan *backend* yang akan mempercepat pengerjaan proyeknya.
2. Dalam mengembangkan desain pada setiap proyek perusahaan mungkin lebih baik jika ada pekerja yang fokus dalam pekerjaan *UI / UX*. Dengan pekerjaan tersebut mungkin hasil yang ditampilkan pada setiap aplikasi akan lebih cepat dan baik. Pekerjaan tersebut telah dijelaskan pada **Bab 4.1.4**
3. Dalam mengembangkan proyek perusahaan, pemagang mungkin lebih senang ketika mendapatkan proyek yang berhubungan langsung dengan klien. Pemagang juga lebih senang ketika mengerjakan proyek tersebut hingga selesai dan dapat melihat klien memakai aplikasi yang dikerjakan oleh pemagang.
4. Bagi perusahaan mungkin kedepannya perlu terus melibatkan pemagang ketika terdapat *meeting* perusahaan. Dengan mengikuti *meeting*, pemagang dapat ikut serta dalam menganalisis masalah yang terdapat pada *meeting* tersebut.

b. Kampus

1. Saran untuk kampus agar menginformasikan mengenai hal yang harus dilakukan pada saat magang, seperti: harus turut aktif dalam menyelesaikan tugas yang diberikan oleh mentor dan mengaitkan pengerjaan proyek dengan mata kuliah. Dengan hal tersebut, mahasiswa dapat membuat laporan dan mengerjakan proyek dengan semaksimal mungkin.
2. Saran untuk kampus agar diinformasikan lebih cepat terkait *deadline* pengumpulan *file* yang harus dikumpulkan mahasiswa kepada kampus. Dengan hal tersebut, mahasiswa yang mengambil jalur magang dapat menyiapkan laporan tersebut dengan lebih baik lagi.

c. Mahasiswa yang akan magang baik dari UII maupun non UII

1. Mahasiswa yang mengikuti magang lebih mempersiapkan *skill* yang akan difokuskan untuk kerja sebagai pemegang nanti. Karena dengan lebih mempersiapkan *skill*, pemegang dapat mengajukan kepada mentor agar ikut terjun langsung ke proyek yang berhubungan langsung dengan klien. Sehingga, perusahaan akan mempertimbangkan kembali *skill* yang dimiliki oleh pemegang.
2. Mahasiswa yang ingin bermagang sebagai developer khususnya aplikasi *mobile* perlu lebih mendalami *design pattern* pada mata kuliah Arsitek Perangkat Lunak karena dengan mendalami hal tersebut mahasiswa dapat lebih luas wawasannya ketika mempelajari *design pattern* yang dipakai pada setiap aplikasi.

d. Proyek aplikasi menggunakan Flutter

1. Fitur khusus seperti menampilkan *list*, pencarian, login, dan pendaftaran dalam aplikasi Flutter perlu dibuatkan *widget* khusus yang dapat dipasang dan tidak membuat kode *widget* sepenuhnya pada *widget* build sehingga dapat memudahkan developer lain dalam melakukan perubahan *widget*-nya.
2. Semua fungsi logika seperti melakukan pemanggilan data dari REST API, perlu dipisah dari kelas *User Interface* seperti dengan menggunakan *design pattern* BLoC sehingga fungsi pemanggilan data tersebut dapat memudahkan untuk melakukan *unit test* di folder Test yang sudah disediakan oleh Flutter.

DAFTAR PUSTAKA

- Pradana, A.R. (2021). *IMPLEMENTASI USER EXPERIENCE PADA PERANCANGAN USER INTERFACE APLIKASI MOBILE E-LEARNING DENGAN PENDEKATAN DESIGN THINKING* (Skripsi, Universitas Islam Indonesia, 2021). Yogyakarta.
- Sobri, A. & Aan. (2019). SISTEM INFORMASI E-GOVERNMENT PADA KANTOR BADAN PENDIDIKAN DAN PELATIHAN KOTA LUBUKLINGGAU BERBASIS WEBSITE MENGGUNAKAN METODE OBJECT ORIENTED PROGRAMMING. *Jurnal Teknologi Informasi Mura*, 49.
- Lindley, C. (2019). *Frontend Developer Handbook 2017. Handbook*. Dapat diakses di <https://frontendmasters.com/guides/front-end-handbook/2019/>
- Developer Libs. (2019). Flutter – Lifecycle of Widgets. Situs Web. Diakses dari <https://www.developerlibs.com/2019/12/flutter-lifecycle-widgets.html>
- SALIHA, D.A. (2018). *EVALUASI SISTEM INFORMASI MANAJEMEN RUMAH SAKIT DENGAN MENGGUNAKAN MODEL HUMAN ORGANIZATION AND TECHNOLOGY FIT* (Skripsi, Universitas Islam Negeri Syarif Hidayatullah, 2018). Jakarta.
- Ayunindya, F. (2021). Apa Itu GitHub? Kenali Pengertian dan Fungsinya. Situs Web. Diakses dari <https://www.hostinger.co.id/tutorial/apa-itu-github>
- Flutter. (2021). Hot Reload. Situs Web. Diakses dari <https://docs.flutter.dev/development/tools/hot-reload>
- Napoli, M.L. (2020). *Beginning Flutter: A Hands On Guide To App Development*. Buku. Indiana: John Wiley, S. I. Diakses dari [https://books.google.co.id/books?hl=en&lr=&id=extDwAAQBAJ&oi=fnd&pg=PR21&dq=John+Wiley+%26+Sons,+Inc+\(2020\).+INTRODUCING+FLUTTER.+Indianapolis,+Indiana+in+Canada&ots=YVMnd-8fEe&sig=fo8TryL1AGOyi1mn4WmFQ4wl9nU&redir_esc=y#v=onepage&q&f=false](https://books.google.co.id/books?hl=en&lr=&id=extDwAAQBAJ&oi=fnd&pg=PR21&dq=John+Wiley+%26+Sons,+Inc+(2020).+INTRODUCING+FLUTTER.+Indianapolis,+Indiana+in+Canada&ots=YVMnd-8fEe&sig=fo8TryL1AGOyi1mn4WmFQ4wl9nU&redir_esc=y#v=onepage&q&f=false)
- Faza, N. (2019). BLoC Pattern pada Flutter. Situs Web. Diakses dari <https://medium.com/learnfazz/bloc-pattern-pada-flutter-a13523af8484>
- Apriadji, R. (2017). 8 Hal yang Perlu Ada dalam Penulisan API yang Baik. Situs Web. Diakses dari <https://dailysocial.id/post/8-hal-yang-perlu-ada-dalam-penulisan-api-yang-baik>
- Ramasamy, S., Hwata, C., & Jekese, G. (2015). Impact of Object Oriented Design Patterns on Software Development. *International Journal of Scientific and Engineering Research*. 963-964.

Kurniawan, T. (2018). *ANALISIS PENGGUNAAN MEMORI INTERNAL PADA PERANGKAT LUNAK VIDEO EDITING* (Skripsi, Universitas Islam Negeri Sunan Kalijaga, 2018). Yogyakarta.

Universitas Andalas. (2016). Sistem Informasi Manajemen Kepegawaian Universitas Andalas. Situs Web. Diakses dari <https://lptik.unand.ac.id/berita-dan-panduan/item/149-sistem-informasi-manajemen-kepegawaian-simpeg-universitas-andalas>

Ajoku, K. K., Nwokonkwo, O.C., John-Otumu, A.M., & Oleji, C.O. (2021) A Model for Stock Market Value Forecasting using Ensemble Artificial Neural Network. *Journal of Advances in Computing, Communications and Information Technology* 2. 5.

CV. Atsoft Teknologi (2021). DATA ORGANISASI PERUSAHAAN. *Handbook*. Yogyakarta.

