

**PENGEMBANGAN *FRONT-END* APLIKASI BERBASIS
MOBILE ACCABSENSI MENGGUNAKAN *LOW-CODE*
PLATFORM OUTSYSTEMS**



Disusun Oleh:

N a m a : Ghifar Maulana Akbar

NIM : 18523270

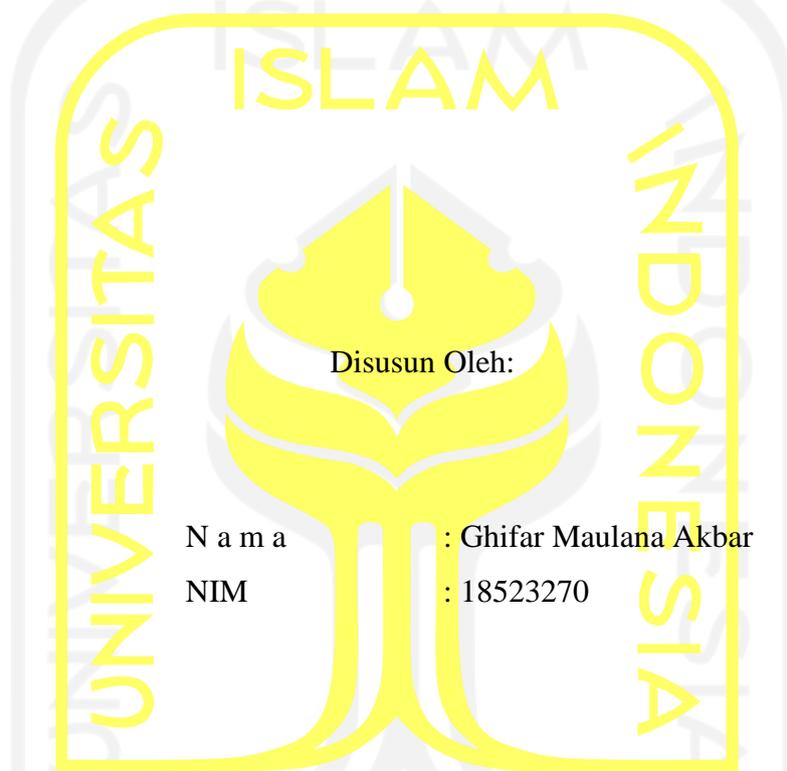
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN *FRONT-END* APLIKASI BERBASIS
MOBILE ACCABSENSI MENGGUNAKAN *LOW-CODE*
PLATFORM OUTSYSTEMS**

TUGAS AKHIR JALUR MAGANG



الجمعة المباركة الأندونيسية

Yogyakarta, 11 Oktober 2022

Pembimbing,

(Moh.Idris, S.Kom, M.Kom)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN *FRONT-END* APLIKASI BERBASIS
MOBILE ACCABSENSI MENGGUNAKAN *LOW-CODE*
PLATFORM OUTSYSTEMS**

TUGAS AKHIR JALUR MAGANG

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

Yogyakarta, 11 Oktober 2022

Tim Penguji

Moh. Idris, S. Kom. M. Kom

Anggota 1

Andhika Giri Persada S. Kom. M. Eng.

Anggota 2

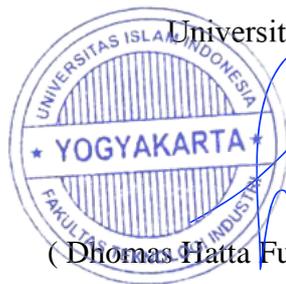
Chanifah Indah Ratnasari S. Kom. M. Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ghifar Maulana Akbar

NIM : 18523270

Tugas akhir dengan judul:

**PENGEMBANGAN *FRONT-END* APLIKASI BERBASIS
*MOBILE ACC*ABSENSI MENGGUNAKAN *LOW-CODE*
PLATFORM OUTSYSTEMS**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian 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 risiko dan konsekuensi apapun.

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

Yogyakarta, 11 Oktober 2022



(Ghifar Maulana Akbar)

HALAMAN PERSEMBAHAN

Alhamdulillah, puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufiq dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan skripsi atau tugas akhir ini dengan baik. Tidak lupa selawat serta salam kepada Nabi Muhammad SAW yang telah membimbing dan memberikan tuntunan agama yang sempurna untuk menjadi rahmat bagi seluruh alam.

Laporan Tugas Akhir ini, penulis persembahkan kepada orang tua maupun keluarga yang senantiasa telah memberikan doa, semangat, motivasi. Tanpa dukungan orang tua maupun keluarga mungkin penulis tidak akan berada pada posisi saat ini, dengan demikian penulis sangat berterimakasih terhadap apa yang telah diberikan. Terima kasih kepada dosen pembimbing saya Bapak Moh. Idris S. Kom., M. Kom. yang senantiasa memberikan arahan maupun bimbingan dalam penyelesaian tugas akhir ini.

Terima kasih juga terhadap dosen-dosen Informatika UII yang telah memberikan ilmu-ilmu selama berada pada bangku perkuliahan. Semoga ilmu yang diberikan dapat menjadi ilmu yang bermanfaat dan dapat dipertanggung jawabkan. Tidak lupa penulis ucapkan terima kasih kepada teman-teman seperjuangan yang telah memberikan kesan-kesan kehidupan selama masa perkuliahan ini, semoga segala yang dicita-citakan dapat tercapai.

HALAMAN MOTO

وَالِي رَبِّكَ فَارْغَبْ

“dan hanya kepada Tuhanmulah engkau berharap.”

QS. Al-Inshirah ayat 8.

وَلَا تَأْيِسُوا مِنْ رَوْحِ اللَّهِ إِنَّهُ لَا يَأْيِسُ مِنْ رَوْحِ اللَّهِ إِلَّا الْقَوْمُ
الْكَافِرُونَ

“dan jangan kamu berputus asa dari rahmat Allah. Sesungguhnya tiada berputus asa dari rahmat Allah, melainkan kaum yang kafir”.

QS. Yusuf ayat 87

"It's not who I am underneath, but what I do that defines me."

Bruce Wayne, Batman Begins.

الجمعة المباركة
الاستاذة الاندوية

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh,

Alhamdulillah penulis haturkan kepada Allah Swt., yang telah melimpahkan rahmat dan taufiq serta hidayat-Nya. Tidak lupa selawat serta salam senantiasa tercurahkan bagi Rasulullah Saw, yang telah membawa manusia dari zaman kegelapan menuju zaman terang benderang yakni addinul islam.

Laporan Tugas Akhir ini sengaja dibuat untuk memenuhi persyaratan kelulusan. Tujuan dibuatnya Laporan Tugas Akhir ini ialah untuk melaporkan segala sesuatu yang penulis lakukan dan kerjakan selama magang berlangsung di perusahaan Astra Credit Companies Technocenter.

Dalam pelaksanaan magang di Astra Credit Companies, tentunya banyak hambatan dan rintangan yang telah penulis lewati, namun pada akhirnya dapat terselesaikan dengan baik karena adanya bantuan dan dukungan dari orang-orang di sekitar penulis. Pada kesempatan ini, penulis ingin menyampaikan terimakasih yang sebesar-besarnya kepada:

1. Allah SWT yang selalu memberikan kenikmatan serta perlindungannya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir dengan baik.
2. Bapak Ir. Imam Akbarudin dan Ibu Ery Fatmawati selaku orang tua serta keluarga yang selalu mendukung baik secara moral maupun finansial, memberikan semangat serta doa dalam menempuh studi perkuliahan di Informatika Universitas Islam Indonesia dari awal hingga akhir.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memimpin dan membina *civitas* akademik jurusan Informatika.
4. Bapak Dhomas Hatta Fudholi, Ph.D., selaku Ketua Program Studi Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia yang sudah menerima penulis sebagai salah satu mahasiswa Informatika Universitas Islam Indonesia.
5. Bapak Moh. Idris, S.Kom., M.Kom., selaku Dosen Pembimbing tugas akhir yang sudah memberikan bimbingan dan bantuan kepada penulis dalam penyusunan Laporan Tugas Akhir ini.
6. Bapak dan Ibu selaku dosen program studi Informatika Universitas Islam Indonesia yang telah memberikan seluruh ilmu yang sangat bermanfaat selama menempuh studi perkuliahan.

7. Mutiara Caesagusta Yosadhie dan lainnya selaku supervisor di ACC Technocenter, serta teman-teman ACC Technocenter Bootcamp 2 yang telah mendukung dan memotivasi penulis.
8. Teman-teman terdekat penulis yang telah memberikan bantuan, dukungan, serta berbagi pengalaman selama penyusunan Laporan Tugas Akhir ini.
9. Tasya Alya Salsabilla, yang telah sabar menemani, memberi semangat, dukungan, serta doa sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan baik.
10. Pihak-pihak lainnya seperti teman seperjuangan yang turut mendukung dan membantu dalam penyusunan Laporan Tugas Akhir ini yang tidak dapat disebutkan satu persatu.

Segala kebaikan yang diberikan seluruh pihak kepada penulis. Maka penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan sebaik – baiknya dan semaksimal mungkin. Laporan Tugas Akhir ini memang masih jauh dari kata kesempurnaan, tapi penulis telah berusaha sebaik mungkin, sekali lagi saya ucapkan terima kasih. Semoga laporan ini bermanfaat bagi kita semua dan lebih khususnya bagi penulis.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 11 Oktober 2022



(Ghifar Maulana Akbar)

SARI

Permasalahan wabah Covid-19 menjadi permasalahan yang sangat berdampak bagi beberapa perusahaan. Berkurangnya produktivitas suatu perusahaan banyak diakibatkan dari permasalahan internal yang terjadi, seperti contoh kurangnya pengawasan atau kontrol terhadap pegawai yang diakibatkan dari penerapan sistem *work from home* yang dianjurkan oleh pemerintah untuk setiap perusahaan dikarenakan oleh pandemi. Salah satunya yaitu pengawasan terhadap presensi atau kehadiran serta pencatatan laporan kinerja yang dilakukan oleh para pegawai perusahaan itu sendiri.

Teknologi yang kian meningkat penggunaannya membuat pengembangan sebuah aplikasi diharapkan dapat menjadi target solusi untuk mengatasi permasalahan yang terjadi. Oleh sebab itu dikembangkannya ACCAbsensi yaitu aplikasi yang memiliki fungsi utama untuk melakukan pencatatan presensi serta pelaporan kinerja secara *online* hanya dengan menggunakan *smartphone* saja. Aplikasi ini dikembangkan oleh penulis pada saat magang di perusahaan Astra Credit Companies untuk mengatasi salah satu permasalahan yang terjadi pada situasi pandemi Covid-19. Dengan menggunakan *low-code* platform Outsystems aplikasi dapat dikembangkan dengan cepat tanpa melakukan *hard-coding* seperti pada umumnya dan pengembangan menjadi lebih terstruktur serta tepat dikarenakan menerapkan metode pengembangan *scrum*.

Kata kunci: Covid-19, *scrum*, *low-code* platform, outsystems.

GLOSARIUM

<i>Scrum</i>	Metode pengembangan perangkat lunak <i>agile</i> .
<i>Low-code</i>	Platform pengembangan kode rendah yang digunakan untuk membuat perangkat lunak aplikasi melalui antarmuka pengguna grafis.
<i>Front-end</i>	Pengembangan antarmuka pengguna grafis dari sebuah aplikasi.
<i>Database</i>	Kumpulan data yang disimpan dan diakses secara elektronik dari suatu sistem komputer.
<i>Framework</i>	kerangka kerja perangkat lunak.



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Ruang Lingkup.....	2
1.3 Tujuan	4
1.4 Manfaat	4
1.5 Sistematika Penulisan	4
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA.....	6
2.1 Landasan Teori.....	6
2.1.1 <i>Scrum</i>	6
2.1.2 <i>Low-code Platform</i>	8
2.1.3 <i>Outsystems</i>	9
2.1.4 <i>SIT</i>	11
2.1.5 <i>UAT</i>	12
2.2 Tinjauan Pustaka	12
BAB III PELAKSANAAN MAGANG	15
3.1 Manajemen Proyek	15
3.1.1 Platform Manajemen Proyek.....	15
3.1.2 Metode Pengembangan Aplikasi.....	16
3.1.3 Platform Pengembangan <i>Front-End</i> Aplikasi	17
3.1.4 Metode Pengujian Sistem	18
3.2 ACCAbsensi.....	18
3.2.1 Proses Pengembangan Aplikasi oleh Tim	19
3.2.2 Proses Pengembangan Aplikasi oleh Penulis.....	31
BAB IV REFLEKSI PELAKSANAAN MAGANG	49
3.3 Relevansi Akademik	49
3.3.1 Relevansi <i>Scrum</i>	49
3.3.2 Relevansi <i>Low-Code Platform</i>	50
3.4 Pembelajaran Magang.....	51
3.4.1 Platform <i>Low-Code Outsystems</i>	51
3.4.2 Metode Pengembangan <i>Scrum</i>	52
3.4.3 Platform <i>Monday</i> untuk Manajemen Proyek.....	53
BAB V PENUTUP	54
4.1 Kesimpulan	54
4.2 Saran.....	55
DAFTAR PUSTAKA	56
LAMPIRAN	58

DAFTAR TABEL

Tabel 1.1 Kegiatan program magang	3
Tabel 3.1 Detail tim pengembangan	19
Tabel 3.2 <i>Product backlog sprint</i> pertama ACCAbsensi.....	20
Tabel 3.3 <i>Product backlog sprint</i> kedua ACCAbsensi	21
Tabel 3.4 <i>Sprint backlog</i> dari <i>product backlog sprint</i> pertama	22
Tabel 3.5 <i>Sprint backlog</i> dari <i>product backlog sprint</i> kedua.....	22
Tabel 3.6 Penjelasan pelaksanaan tugas tiap bidangnya pada proses <i>implement</i>	24
Tabel 3.7 Detail pelaksanaan proses <i>review and restropect</i>	30
Tabel 3.8 Detail keterangan pembuatan fitur <i>login</i>	37
Tabel 3.9 Capaian <i>product backlog</i> pada <i>sprint</i> pertama setelah pembuatan halaman <i>login</i> ..	38
Tabel 3.10 Capaian <i>product backlog</i> pada <i>sprint</i> kedua setelah pembuatan halaman <i>login</i> ...	38
Tabel 3.11 Detail keterangan pembuatan fitur halaman utama	40
Tabel 3.12 Capaian <i>product backlog</i> pada <i>sprint</i> pertama setelah pembuatan halaman utama	41
Tabel 3.13 Detail keterangan pembuatan fitur halaman <i>to do list</i>	43
Tabel 3.14 Capaian <i>product backlog</i> pada <i>sprint</i> pertama setelah pembuatan <i>to do list</i>	44
Tabel 3.15 Detail keterangan pembuatan fitur halaman riwayat	45
Tabel 3.16 Capaian <i>product backlog</i> pada <i>sprint</i> pertama setelah pembuatan <i>to do list</i>	46
Tabel 3.17 Detail keterangan pembuatan fitur halaman <i>approval</i>	47
Tabel 3.18 Capaian <i>product backlog</i> pada <i>sprint</i> pertama setelah pembuatan <i>approval</i>	47
Tabel 3.19 Detail keterangan pembuatan fitur halaman <i>approval</i>	48
Tabel 3.20 Capaian <i>product backlog</i> pada <i>sprint</i> pertama setelah pembuatan <i>approval</i>	48

DAFTAR GAMBAR

Gambar 2.1 Fase <i>Scrum</i>	8
Gambar 2.2 <i>The Architecture Canvas</i>	10
Gambar 2.3 <i>The Architecture Canvas sub-layers</i>	11
Gambar 3.1 <i>Zoom Meeting</i> selama pelaksanaan magang.	15
Gambar 3.2 Tampilan <i>Monday</i> sebagai tempat dokumentasi pekerjaan yang dilakukan.....	16
Gambar 3.3 Tampilan <i>Monday</i> dokumentasi pelaksanaan <i>scrum</i>	17
Gambar 3.4 Tampilan <i>environment</i> Outsystems ACC	18
Gambar 3.5 Tampilan <i>Google Meet</i> saat pelaksanaan <i>daily meeting</i>	24
Gambar 3.6 Arsitektur ACCAbsensi	26
Gambar 3.7 <i>Usecase</i> diagram aplikasi ACCAbsensi.....	27
Gambar 3.8 <i>Manual Guide</i> ACCAbsensi	28
Gambar 3.9 <i>Summary System Integration Testing</i> ACCAbsensi.....	29
Gambar 3.10 <i>Summary User Acceptance Testing</i> ACCAbsensi	29
Gambar 3.11 <i>Generate file apk</i> proyek ACCAbsensi.....	31
Gambar 3.12 <i>Flowchart</i> tahapan melakukan presensi.....	33
Gambar 3.13 <i>Flowchart</i> tahapan memvalidasi hasil presensi	34
Gambar 3.14 <i>Flowchart</i> tahapan membuat daftar kerja	35
Gambar 3.15 Halaman <i>login</i> ACCAbsensi.....	36
Gambar 3.16 Alur fungsi <i>logic button</i> masuk pada halaman login	37
Gambar 3.17 Halaman Utama.....	39
Gambar 3.18 Alur fungsi <i>logic</i> presensi pada halaman utama	40
Gambar 3.19 Halaman <i>To Do List</i> ACCAbsensi.....	42
Gambar 3.20 Alur fungsi <i>Logic button</i> buat <i>task</i> pada halaman <i>to do list</i>	43
Gambar 3.21 Halaman Riwayat.....	45
Gambar 3.22 Halaman <i>Approval</i>	46
Gambar 3.23 Halaman Profil	48

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada awal tahun 2020, dunia dihebohkan dengan wabah pandemi Covid-19 yang melanda. Semakin besarnya penyebaran virus Corona yang sudah ada di Indonesia mendesak pemerintah pusat maupun pemerintah daerah mengambil langkah preventif dengan tujuan memutus rantai penularan virus tersebut. Salah satunya dengan menetapkan pembatasan penerapan kerja secara *Work from Office* (WFO) serta penerapan kerja secara *Work from Home* (WFH).

Dihadapkan dengan kondisi yang diakibatkan oleh wabah Covid-19 membuat beberapa pegawai mengalami kesulitan menerapkan sistem bekerja secara WFH. Hal yang menyulitkan dalam bekerja di rumah dapat disebabkan oleh berbagai alasan seperti gangguan dalam mengerjakan pekerjaan rumah, mengurus anak, atau aktivitas anggota keluarga lainnya yang membuat tidak fokus menjalankan WFH. Selain itu, timbulnya rasa bosan dan jenuh membuat pegawai menjadi tidak fokus terhadap hal yang seharusnya dikerjakan. Pegawai menjadi mudah terdistraksi untuk melakukan hal lainnya. Karena hal tersebut, pegawai menjadi mudah letih dan stres yang mengakibatkan berkurangnya produktivitas dalam bekerja. Batasan antara kehidupan pribadi dengan pekerjaan menjadi tidak terkontrol, isolasi dengan kehidupan sosial dan profesional serta kesempatan untuk berbagi informasi menjadi kelemahan saat bekerja dilakukan dirumah.

Penerapan WFH menjadi tantangan tersendiri bagi perusahaan untuk mengontrol dan mengawasi kinerja para karyawannya. Salah satu hal yang diperlukan yaitu terkait pengawasan kehadiran. Perlunya pengawasan dalam kehadiran dikarenakan minimnya pengawasan terhadap hal tersebut saat penerapan WFH. Selain itu, karyawan juga akan menjadi rentan terhadap rasa malas selama bekerja dari rumah karena merasa tidak diawasi langsung selayaknya ketika bekerja di kantor. Jika hal tersebut berkelanjutan maka akan berakibat fatal bagi perusahaan. Kinerja para karyawan dapat menurun sehingga proses bisnis perusahaan akan terdampak. Maka dari itu, perusahaan harus segera mencari sebuah solusi untuk membantu meminimalkan dampak WFH terhadap kinerja para karyawan terutama yang berkaitan dengan pengawasan selama penerapan WFH (Allen, Golden, & Shockley, 2015).

Perusahaan Astra Credit Companies atau biasa disingkat dengan ACC adalah perusahaan pembiayaan mobil dan alat berat dalam kondisi baru ataupun bekas. Yang kemudian ACC melakukan perluasan usaha di bidang pembiayaan investasi, pembiayaan modal kerja, pembiayaan multiguna dan sewa operasi (*operating lease*), baik dengan skema konvensional maupun syariah (ACC, 2021). Perusahaan tersebut menjadi salah satu dari sekian perusahaan yang terdampak wabah Covid-19. Terpaksa menerapkan sistem bekerja secara *Work from Home* demi memutus rantai penyebaran Covid-19. Diharuskannya merubah metode bekerja secara mendadak, dari bekerja yang dilakukan di kantor menjadi dilakukan di rumah, memunculkan permasalahan internal yang terjadi pada perusahaan Astra Credit Companies dalam mengontrol kinerja para pegawainya. Dilakukannya pengontrolan oleh atasan terhadap karyawan dengan cara manual, membuat pengawasan menjadi sedikit memakan waktu banyak yang lama-kelamaan membuat para atasan menjadi tidak sanggup untuk menangani kinerja satu-persatu para karyawannya. Maka dari itu, perusahaan Astra Credit Companies membuat suatu sistem yang diharapkan dapat menangani permasalahan internal yang terjadi, sistem tersebut diberi nama ACCAbsensi.

ACCAbsensi adalah aplikasi berbasis *mobile* yang diharapkan menjadi salah satu solusi dalam memecahkan permasalahan yang terjadi di perusahaan ACC. Dari aplikasi ini nantinya pencatatan terhadap kehadiran kerja yang dilakukan secara *online* dari karyawan dapat disimpan. Selain itu, juga terdapat fitur tambahan seperti pelaporan tugas-tugas yang dikerjakan tiap karyawan setiap harinya.

Laporan Tugas Akhir ini akan menjelaskan tentang pengembangan aplikasi ACCAbsensi dalam rangka untuk mengatasi salah satu permasalahan yang terjadi di perusahaan ACC. Pengembangan aplikasi ini nantinya akan menggunakan platform *mobile* Outsystems dikarenakan sistem yang dikembangkan harus digunakan sesegera mungkin oleh perusahaan. Maka dari itu waktu pengembangan yang singkat dan penggunaan platform yang memudahkan pengembangan menjadikan platform Outsystems dipilih untuk digunakan dalam pengembangan aplikasi ACCAbsensi.

1.2 Ruang Lingkup

ACC Digital Operation Center adalah pusat operasional berbasis digital untuk memberikan pelayanan yang terbaik kepada pelanggan dan partner bisnis perusahaan Astra Credit Companies. ACC Digital Operation Center akan menjadi pusat dari 3 proses digital bisnis ACC yaitu *Operation Center* yang merupakan sentralisasi proses operasional perusahaan

pembiayaan yang efisien serta berbasis digital. *Telephony Center* yang merupakan sentralisasi kegiatan bisnis ACC berbasis telepon seperti *telemarketing*, *telesurvey* dan *telecollection*. Kemudian *Techno Center* yang merupakan kegiatan IT dan pemrograman ACC dalam melakukan seluruh development program dan aplikasi di ACC.

Dinaungi ACC Digital Operation Center *Techno Center*, seluruh *development* program dan aplikasi yang telah berjalan di ACC adalah kinerja dari ACC Technocenter tersebut. Penulis diposisikan sebagai *front-end* developer terhadap aplikasi *mobile* maupun web di dalam pengerjaan proyek ACC. Penulis diberikan tugas untuk mengerjakan seluruh tugas dibagian *front-end* yang di antaranya seperti membuat *user interface* aplikasi, *logic* aplikasi, hingga mengintegrasikannya dengan bagian *back-end*.

Hal yang belum umum dan banyak orang ketahui tentang platform yang digunakan untuk mengembangkan aplikasi *mobile* maupun web yang dikembangkan oleh ACC Technocenter, yaitu platform Outsystems, platform *low-code* yang disediakan bagi perusahaan untuk mengembangkan, menyebarkan, dan mengelola aplikasi perusahaan *omnichannel* (Outsystems, 2020). Penulis diberi tugas untuk mengembangkan aplikasi berbasis *mobile* maupun web dengan menggunakan platform tersebut. Namun tetapi, dikarenakan *framework* yang digunakan oleh perusahaan dalam mengembangkan aplikasi bermacam-macam, maka dari itu penulis tidak hanya mengembangkan dengan platform tersebut saja, melainkan juga menggunakan *framework laravel* selama periode magang berlangsung.

Kegiatan magang ACC Technocenter berlangsung selama 6 bulan pada tanggal 23 Agustus 2021 hingga tanggal 23 Februari 2021. Dimulai dari pelaksanaan *training bootcamp* yang kemudian dilanjutkan dengan *Project Challenge* atau dapat disebut dengan tantangan proyek, hingga siap terjun langsung untuk menghadapi proyek yang diberikan oleh perusahaan. Adapun pelaksanaan magang di Astra Credit Companies Technocenter dapat dilihat pada Tabel 1.1 berikut.

Tabel 1.1 Kegiatan program magang

No	Agenda	Waktu	Lokasi
1	<i>Training Technocenter Bootcamp</i> (Pelatihan)	23 Oktober 2021 – 6 Oktober 2021	<i>Online</i>
2	<i>Project Challenge ACC</i>	11 Oktober 2021 – 19 November 2021	<i>Online</i>

3	ACC Absensi (Aplikasi Presensi Karyawan)	26 November 2021 – 31 Desember 2021	<i>Online</i>
4	ACC Lead CMS	3 Januari 2022 – 23 Februari 2022	<i>Online</i>

1.3 Tujuan

Tujuan dari penulisan tugas akhir ini adalah mengembangkan aplikasi ACC Absensi dengan menggunakan *low-code* platform yang dapat memudahkan karyawan yang terdapat pada perusahaan dalam melakukan presensi secara *online* serta pengawasan terhadap kinerja yang telah dilakukan oleh setiap karyawannya. Dengan demikian proses presensi dan pengawasan terhadap kinerja karyawan pada perusahaan Astra Credit Companies dapat berjalan secara lancar dan maksimal.

1.4 Manfaat

Manfaat yang dapat diambil dari pengembangan aplikasi ACC Absensi menggunakan *low-code* platform adalah sebagai berikut:

a. Bagi Penulis

Sebagai sarana untuk menerapkan pengetahuan yang diperoleh selama menempuh studi perkuliahan, khususnya di dalam pengembangan sistem dan pemrograman.

b. Bagi Pembaca

Laporan Tugas Akhir ini dapat dijadikan sebagai pengetahuan tambahan dalam pengembangan sistem dengan menggunakan platform *low-code* Outsystems.

c. Bagi Perusahaan Astra Credit Companies

Hasil dari penulisan Tugas Akhir ini diharapkan dapat memberikan kemudahan bagi perusahaan untuk mengatasi solusi yang terjadi dengan menggunakan aplikasi ACC Absensi, sehingga proses pencatatan presensi secara *online* dan pengawasan terhadap kinerja karyawan dapat diolah dengan baik.

1.5 Sistematika Penulisan

Dalam penulisan Tugas Akhir ini, digunakan sistematika penulisan sebagai berikut:

a. BAB 1 PENDAHULUAN

Bab ini berisi mengenai latar belakang, ruang lingkup magang, tujuan, manfaat, dan sistematika penulisan Laporan Tugas Akhir.

b. **BAB 2 LANDASAN TEORI**

Bab ini berisi mengenai dasar teori yang digunakan dan mendasari penulisan tugas akhir. Meliputi metode pengembangan hingga platform yang digunakan.

c. **BAB 3 PELAKSANAAN MAGANG**

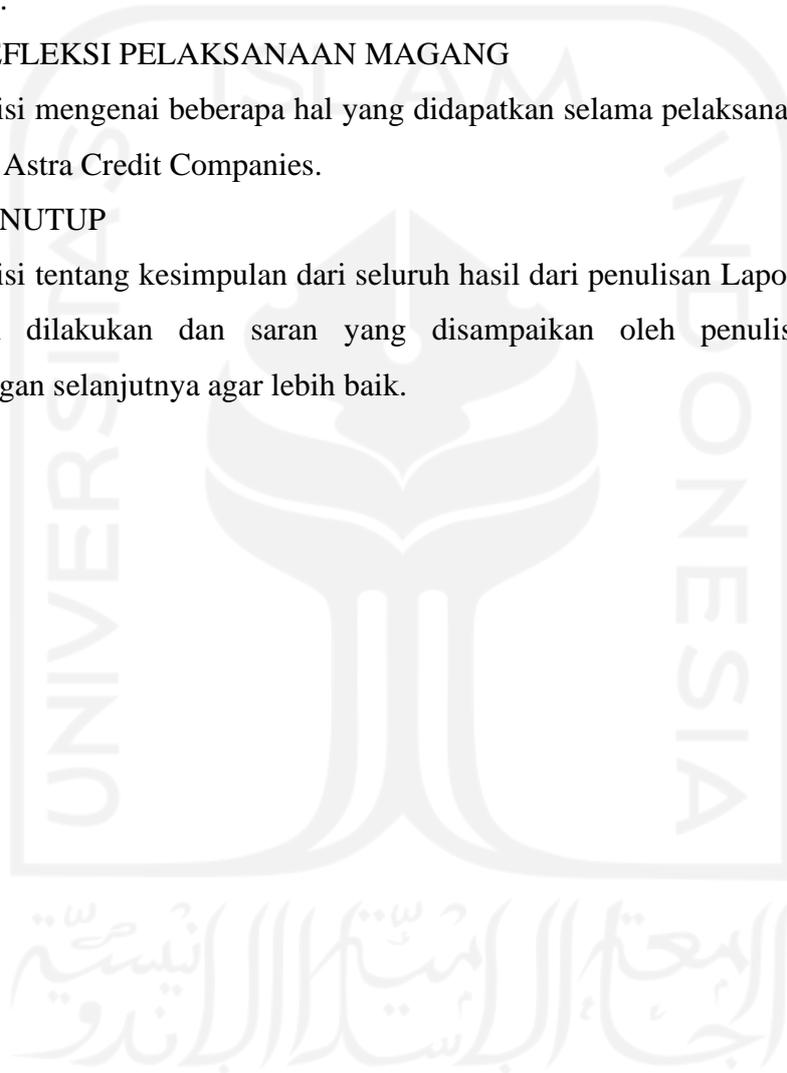
Bab ini berisi tentang tahapan pelaksanaan dalam pengembangan aplikasi ACCAbsensi yang dilakukan oleh penulis sebagai *frontend* developer menggunakan platform *low-code* Outsystems.

d. **BAB 4 REFLEKSI PELAKSANAAN MAGANG**

Bab ini berisi mengenai beberapa hal yang didapatkan selama pelaksanaan magang pada perusahaan Astra Credit Companies.

e. **BAB 5 PENUTUP**

Bab ini berisi tentang kesimpulan dari seluruh hasil dari penulisan Laporan Tugas Akhir yang telah dilakukan dan saran yang disampaikan oleh penulis untuk proses pengembangan selanjutnya agar lebih baik.



BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 *Scrum*

Scrum merupakan kerangka kerja yang digunakan untuk mengelola pengembangan produk kompleks dari awal tahun 1990-an. *Scrum* menggunakan pendekatan bertahap (inkremental) dan berkala (iterative) untuk meningkatkan prediktabilitas dan pengendalian resiko (Rubin, 2013).

Scrum pertama kali dikembangkan oleh Jeff Sutherland pada tahun 1993 yang tujuannya adalah menjadi metodologi pengembangan yang mengikuti prinsip dari metodologi *agile*. *Scrum* merupakan kerangka kerja responsif dari pengembangan perangkat lunak untuk proyek perangkat lunak dan mengelola produk atau pengembangan aplikasi. *Scrum* berfokus pada strategi, pengembangan produk holistik yang fleksibel di mana tim pengembang bekerja sebagai unit untuk mencapai tujuan bersama yang mempunyai *rival* dari pendekatan secara tradisional. *Scrum* memiliki beberapa proses yang kompleks di mana terdapat banyak faktor yang dapat mempengaruhi hasil akhir (Adi, 2015).

Proses pengembangan pada *Scrum* dapat dikelompokkan ke dalam lima fase (Hays, 2016). Fase-fase tersebut di antaranya sebagai berikut:

a. *Initiate*

Fase ini merupakan tahap awal dari proses pengembangan dengan *scrum*. Fase ini meliputi pembentukan tim, pembuatan *project vision* yang merupakan dokumen yang memuat deskripsi mengenai suatu proyek serta tujuan dan ekspektasi apa saja yang akan dicapai dari proyek tersebut, dan penentuan *product backlog* yaitu daftar pekerjaan yang diprioritaskan untuk tim pengembang sesuai *roadmap* dan *requirements* suatu produk. *Project vision* dan *product backlog* dibuat oleh *product owner* yang kemudian akan dilakukannya penjabaran deskripsinya oleh tim.

b. *Plan Estimate*

Pada fase ini dilakukan perencanaan untuk memulai pelaksanaan *sprint*, *Sprint* merupakan inti dari *Scrum*, yakni yang berarti satu fase waktu dalam satu bulan atau kurang dalam menghasilkan produk tertentu, jalannya satu *sprint* ini sudah ditentukan dan direncanakan sebelumnya oleh *product owner* beserta *team development*. Perencanaan dalam satu *sprint* meliputi penulisan *user story*, penjabaran *task* pada tiap *user story*, melakukan estimasi

nilai terhadap setiap *user story* dan *task*, serta menentukan *sprint backlog*. Disini *user story* adalah kumpulan deskripsi yang bersifat non-formal yang telah dijabarkan sebelumnya berdasarkan dari *product backlog*. Selain itu *sprint backlog* adalah perencanaan matang untuk melakukan pengembangan produk yang telah didapatkan dari deskripsi *user story*. *User story* dan *sprint backlog* ini ditentukan oleh *team development* untuk memaksimalkan kinerja mereka.

c. *Implement*

Fase ini merupakan tahap untuk mengeksekusi setiap *task* yang telah didefinisikan serta melakukan aktivitas-aktivitas untuk membentuk produk. Pada tahap ini juga terdapat *daily standup meeting*, di mana tim melakukan evaluasi apa yang telah dilakukan oleh anggota tim selama jangka waktu tertentu. Pada saat tersebut juga dapat diceritakan masalah apa saja yang ditemui selama melakukan implementasi.

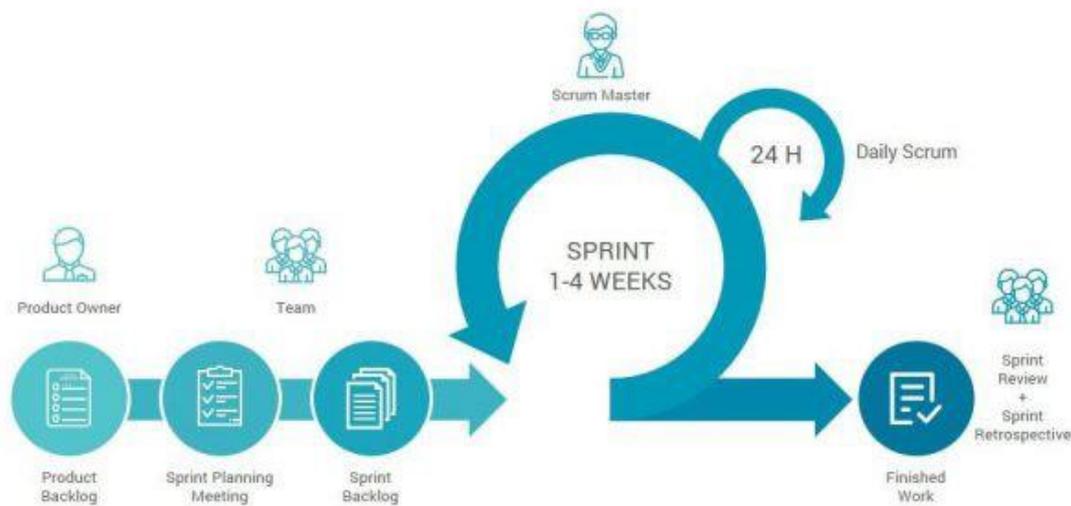
d. *Review and Restropect*

Pada tahap ini dilakukan *review* terhadap hasil pekerjaan tim (*deliverable product*) selama satu *sprint*. Tidak hanya hasilnya, proses bekerja tim juga dinilai sehingga dapat ditentukan apa saja yang perlu ditingkatkan untuk proses pengembangan pada *sprint* berikutnya. *Sprint* dianggap berhasil apabila pada *sprint* tersebut dihasilkan (*deliverable product*) yang sesuai dengan *acceptance criteria* yang telah ditentukan dan disetujui oleh *product owner*.

e. *Release*

Pada tahap ini produk yang telah memenuhi seluruh *acceptance criteria* dikirimkan ke klien. Selain itu juga dilakukan dokumentasi mengenai pelajaran apa saja yang didapat selama melakukan pengembangan.

Bukan sebuah proses, teknik, ataupun metodologi, akan tetapi *scrum* adalah sebuah kerangka kerja di mana di dalamnya dapat menggunakan beberapa proses dan teknik. *Scrum* dapat mengekspos ketidak-efektifan manajemen produk dan teknik kerja yang dilakukan, sehingga dari hal tersebut secara terus menerus akan mampu meningkatkan kinerja produk. Proses pengembangan pada *scrum* dikelompokkan ke dalam 5 fase seperti pada Gambar 2.1.



Gambar 2.1 Fase *Scrum*

2.1.2 *Low-code Platform*

Platform dengan kode rendah adalah sebuah alat untuk *programmer* dan non-*programmer*. Ini memungkinkan pembuatan dan penyampaian dengan cepat dari aplikasi bisnis dengan upaya minimal untuk menulis dalam bahasa pengkodean serta upaya yang minimal dalam instalasi, konfigurasi, dan implementasi. Dengan jumlah perusahaan *low-code* platform yang berkembang pesat, penggunaan *low-code* platform dapat menjadi solusi signifikan untuk melangkah maju dalam menciptakan aplikasi bisnis yang penting. Sejauh ini, yang tercepat dan mungkin juga metode termurah dalam mengembangkan perangkat lunak (Waszkowski, 2019).

Low-code platform merupakan salah satu alternatif pengembangan perangkat lunak. *Low-code* platform memungkinkan pengembang untuk meminimalkan pengkodean secara manual dengan bahasa pemrograman yang telah umum ditemui. Penggunaan fitur *drag and drop* merupakan salah satu fitur dapat membuat proses pengembangan menjadi lebih cepat. Pengembang dapat menggunakan *low-code* platform untuk membangun aplikasi dengan tampilan visual aplikasi secara langsung dan dapat diintegrasikan dengan *database* (Threestayanti, 2019).

Cara cepat mengembangkan aplikasi yang sekaligus dapat mendesain secara langsung dengan hanya menggunakan sedikit kode manual saja adalah keunggulan utama dari *low-code* platform. *Low-code* platform menyediakan antarmuka berbasis grafik dalam mengonfigurasi aplikasi yang akan dibuat, sehingga pengembang tidak perlu lagi melakukan implementasi dengan hanya menuliskan bahasa pengkodean seperti umumnya. Platform ini hadir dalam

serangkaian *tools* yang dapat membantu para pengembang menciptakan aplikasi dengan hanya menggunakan antarmuka *drag and drop* saja. Selain itu memungkinkan pengembang secara cepat membangun sebuah aplikasi dengan bantuan *User Interface* yang modern, integrasi, data, dan *logic*, tanpa menulis ribuan kode dan *syntax* yang kompleks.

2.1.3 Outsystems

Outsystems merupakan platform yang sering dikenal sebagai platform “*low-code*” yang memungkinkan dalam pengembangan seluruh aplikasi dilakukan secara visual, mudah diintegrasikan dengan sistem lain yang ada dan penambahan kode sesuai kebutuhan. Platform tersebut dibuat oleh para *engineer* dengan perhatian yang obsesif terhadap detail. Setiap aspek dari platform Outsystems dirancang untuk membantu pengembangan suatu aplikasi berbasis *mobile* maupun web dengan lebih baik dan cepat (Menezes, 2020) .

Banyak keunggulan yang diberikan jika menggunakan platform Outsystems, keunggulan utamanya jika dibandingkan dengan platform lain, Outsystems tidak memerlukan banyak kode dasar. Dikarenakan menggunakan *artificial intelligence* membuat para pengembang aplikasi dapat mempersingkat waktu pengembangan. Outsystems mencakup layanan pengembangan hingga peluncuran, memberikan pengguna suatu lingkungan pengembangan yang terintegrasi, meliputi seluruh *lifecycle* pengembangan. Hal tersebut didukung oleh *cloud* otomatis *DevOps* milik Outsystems (tersedia untuk semua pelanggan) yang menyematkan aplikasi *mobile* ke organisasi pelanggan (Sukindar, 2018).

Dalam Outsystem, terdapat komponen dan *typical infrastructure*. Berikut ini adalah komponen dari Outsystem (Outsystems, 2022) :

- a. Platform Server
Komponen yang mengatur semua aktivitas *runtime*, *deployment*, dan manajemen untuk semua aplikasi.
- b. Integration Studio
Alat pengembangan untuk membuat konektor untuk mengintegrasikan aplikasi Outsystem dengan sistem perusahaan lainnya.
- c. LifeTime
Konsol yang terpusat untuk mengelola infrastruktur, *environment*, aplikasi, IT *users*, dan *security*.
- d. Service Center
Konsol administrasi pada sebuah *environment* dari infrastruktur.

e. *Service Studio*

Alat pengembangan visual untuk membuat, mengubah, dan menggunakan aplikasi. *Service Studio* dipasang pada desktop pengembang.

Outsystem mencakup *lifecycle* aplikasi, dari pengembangan hingga penerapan. Outsystem memiliki 4 infrastruktur (Outsystems, 2022) :

a. *Development Environment*

Environment di mana aplikasi pada awalnya dikembangkan dan diuji.

b. *Quality Environment*

Environment di mana penguji dan pengguna bisnis melakukan percobaan aplikasi untuk menjamin kualitas. *Environment* ini biasanya memiliki beberapa persyaratan skalabilitas dan redundansi.

c. *Production Environment*

Environment yang menampung aplikasi versi *end-user* yang berinteraksi. *Environment* ini hanya dapat diakses oleh tim operasi.

d. *Management Environment*

Environment yang menampung aplikasi *LifeTime*, yang merupakan konsol manajemen infrastruktur.

Arsitektur yang dimiliki platform *Outsystems* adalah *The Architecture Canvas*. Desain arsitektur yang berorientasi layanan sederhana berikut memungkinkan untuk sebuah *services* dapat digunakan kembali pada modul aplikasi yang berbeda. Desain arsitektur ini memiliki tiga *layer* utama dan di dalam *layer* utama tersebut dilengkapi dengan beberapa *sub-layers*. Pada Gambar 4.1 berikut adalah layer utama dari desain arsitektur tersebut dan *sub-layers* nya terdapat pada Gambar 4.2.

<i>End-user</i>	No Services	UI and processes That provide functionality to the end users
<i>Core</i>	Reusable	Business services Services around business concepts
<i>Foundation</i>	Services	Non-functional requirements Services to connect to external systems or to extend your framework

Gambar 2.2 *The Architecture Canvas*

The Architecture Canvas

Sub Layers



Gambar 2.3 *The Architecture Canvas sub-layers*

End-user Modules merupakan modul yang berisi desain layar antarmuka yang akan digunakan sebagai perantara atau penghubung antara pengguna aplikasi dengan sistem. Modul ini tidak dapat digunakan kembali seperti dua modul lainnya dikarenakan pada modul ini merupakan layar antarmuka utama yang digunakan pada aplikasi tertentu.

Core Modules merupakan bagian terpenting dari *layers* dikarenakan semua proses bisnis berada di dalamnya. Dapat dikatakan pada modul ini adalah sebagai tempat semua data disimpan dan diolah. Dari hal tersebut, maka modul ini dapat digunakan berulang-ulang di semua modul yang terdapat pada Outsystems, tidak hanya pada aplikasi yang sama, melainkan juga dapat di lain aplikasi sekaligus.

Foundation Modules yaitu modul yang berasal dari Outsystems itu sendiri. Modul ini adalah layanan yang diberikan oleh Outsystems kepada penggunanya. Layanan tersebut berupa *plug-ins* atau fungsi tambahan yang diperlukan pada saat pengembangan aplikasi dilakukan (Abednego, 2021).

2.1.4 SIT

System Integration Testing atau biasa disingkat menjadi *SIT*. Merupakan suatu aktivitas pengujian atau tet yang dilakukan dengan tujuan untuk memastikan suatu aplikasi yang sedang diuji mampu sukses berinteroperasi dengan *software* lain yang telah ditentukan (Watkins & Simons, 2010). Dapat dipahami bahwa SIT adalah suatu aktivitas *testing* terhadap suatu sistem

secara menyeluruh yang meliputi banyak komponen subsistem. Dari aktivitas ini, dapat dipastikan bahwa aplikasi yang diuji mampu beroperasi dengan sukses. SIT menerapkan teknik *black box testing* (metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada input dan *output* aplikasi) dan menguji interoperabilitas sistem secara *high-level*. Dalam menemukan *error* dan *defect*, penggunaan teknik *negative testing* dan *error guessing* sangat umum digunakan pada fase *testing* ini.

2.1.5 UAT

Menurut William E. Perry, *User Acceptance Testing* atau yang dapat disingkat menjadi UAT adalah aktivitas pengujian yang dilakukan oleh *end-user*, di mana *user* tersebut yang secara langsung berinteraksi dengan sistem. Verifikasi juga dilakukan untuk menguji fungsi dari sistem apakah telah berjalan sesuai dengan kebutuhan ataupun fungsinya (Perry, 2007).

Tujuan tes ini adalah untuk menunjukkan bahwa sistem telah memenuhi persyaratan *user*. Fase ini bertujuan untuk membangun kepercayaan dalam kualitas perangkat lunak. *Acceptance testing* seharusnya bukan menemukan *bug* dan biasanya merupakan tahap *testing* yang terakhir sebelum produk dirilis (Black, 2009).

User Acceptance Testing umumnya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan tertentu. *User Acceptance Testing* merupakan pengujian yang dilakukan oleh *user* dengan menggunakan teknik pengujian *black box* untuk menguji sistem terhadap spesifikasinya. Hal tersebut dilakukan sebagai upaya dalam memastikan semua fungsionalitas yang relevan telah diuji.

Dari beberapa definisi dapat disimpulkan bahwa *User Acceptance Testing* adalah aktivitas pengujian yang dilakukan oleh *user* dari sistem tertentu dengan tujuan untuk memastikan fungsi yang terdapat pada sistem tersebut beroperasi dengan baik dan sesuai dengan kebutuhan dan persyaratan dari *user*.

2.2 Tinjauan Pustaka

Penelitian ini merupakan penelitian yang berfokus membahas tentang penerapan *low-code platform outsystem*, di mana nantinya diharapkan dengan menggunakan platform *low-code* yang dapat memudahkan pengawasan dan pemantauan kinerja karyawan di Astra Credit Companies dapat berjalan dengan lancar dan optimal. Selain itu, peneliti juga menggali informasi dari buku dan jurnal skripsi untuk memperoleh informasi teori yang telah ada sebelumnya terkait dengan judul untuk memperoleh tinjauan pustaka yang terkait dengan

pengembangan aplikasi berbasis *mobile* ACCAbsensi yang menggunakan *low-code* platform *Outsystem*.

Penelitian yang pertama yang telah dilakukan oleh Mew pada tahun 2018 membahas tentang penggunaan *Mendix* yang merupakan *low-code development* platform untuk mendukung proses kursus manajemen proyek. Kursus tersebut dilakukan pada SPCS (*School of Professional and Continuing Studies*). Dalam penelitiannya, Mew menyebutkan bahwa *Mendix* dipilih sebagai alat untuk pengembangan aplikasi karena fungsionalitas, kesederhanaan dan dukungan yang diberikan. Penggunaan *Mendix* yang diterapkan pada pelaksanaan kursus dapat mengembangkan pengetahuan, keterampilan, dan kemampuan siswa dalam melakukan manajemen proyek (Mew & Field, 2018)

Penelitian selanjutnya yaitu penelitian yang dilakukan Waszwoski pada tahun 2019 membahas mengenai pemanfaatan *low-code development* platform yang bernama *Auera BPM* yang berfungsi untuk melakukan otomatisasi dari proses bisnis pada bidang manufaktur. Waszwoski menyebutkan bahwa proses bisnis yang berada di pabrik masih dilakukan dengan manual, sehingga adanya proses otomatisasi dalam proses bisnis ini menjadi kunci untuk menjalankan pabrik yang menguntungkan. Penyebab utama perusahaan yang belum melakukan otomatisasi adalah dikarenakannya biaya yang mahal untuk membangun aplikasi intuitif dan perlunya pengembang ahli untuk membangunnya. Dari latar belakang tersebut kemudian terdapat solusi yaitu menggunakan *low-code platform development* yang memungkinkan pengembangan aplikasi dengan skala *enterprise* dengan lebih cepat dan murah karena tidak membutuhkan banyak *programmer* yang melakukan *hard coding*. Aplikasi *low-code* platform yang digunakan dalam penelitian ini adalah *Auera BPM*. Modul antarmuka dari *Auera BPM* mampu menyiapkan aplikasi pada berbagai tingkat kerumitan, dari alur kerja sederhana hingga solusi kompleks yang berisi basis data, dasbor pengguna dan klien, manajemen konten, otomatisasi tugas, penyeimbangan beban, seluler antarmuka, pemantauan bisnis, tinjauan pasca tindakan sampai dengan simulasi proses. Setiap diagram proses yang dibuat nantinya akan diubah menjadi aplikasi web yang dapat diakses dengan mudah dari mana saja. Dengan adanya *low-code* platform *Auera BPM* ini membuat proses otomatisasi pada bidang manufaktur menjadi lebih mudah dan efisien baik dari biaya, waktu pengembangan, dan pemeliharaan sistem karena semuanya sudah terintegrasi dengan baik (Waszkowski, 2019).

Sebuah studi tahun 2020 yang dilakukan oleh Metrolho menjelaskan bagaimana platform kode rendah dapat digunakan untuk memenuhi kebutuhan pasar pengembang aplikasi. Tujuan

dari penelitian ini adalah untuk mempersiapkan pekerja yang ingin memasuki dunia IT sebagai profesional melalui pelatihan pengembangan aplikasi menggunakan platform pengembangan kode rendah. Untuk penelitian ini, platform *low-code* yang digunakan adalah Outsystems. Outsystems ingin mempersiapkan para profesional untuk tren pekerjaan baru di dunia IT karena beberapa alasan, termasuk kurangnya pemahaman mendalam tentang bahasa pemrograman tertentu dan pengalaman bertahun-tahun, kemampuan untuk menyebarkan dan memperbarui aplikasi dengan satu klik. platform pengembangan untuk terapkan fitur baru dengan cepat di lingkungan *cloud*. Hasil pelatihan yang termasuk dalam penelitian ini menunjukkan bahwa peserta pelatihan sangat termotivasi untuk belajar, dengan tingkat ketidakhadiran yang hampir nol dan tingkat keberhasilan yang mendekati 100%. Beradaptasi dengan platform *low-code* mudah dilakukan bagi kebanyakan peserta pelatihan. Hanya ada beberapa kesulitan untuk beberapa peserta pelatihan yang tidak memiliki pengalaman dalam pengembangan web atau seluler. Beberapa aspek muncul sebagai motivasi utama untuk kursus, Salah satunya berkaitan dengan menemukan peluang karir baru di perusahaan Anda atau di pasar *low-code* (Metrólho, Ribeiro, & Araújo, 2020).

BAB III

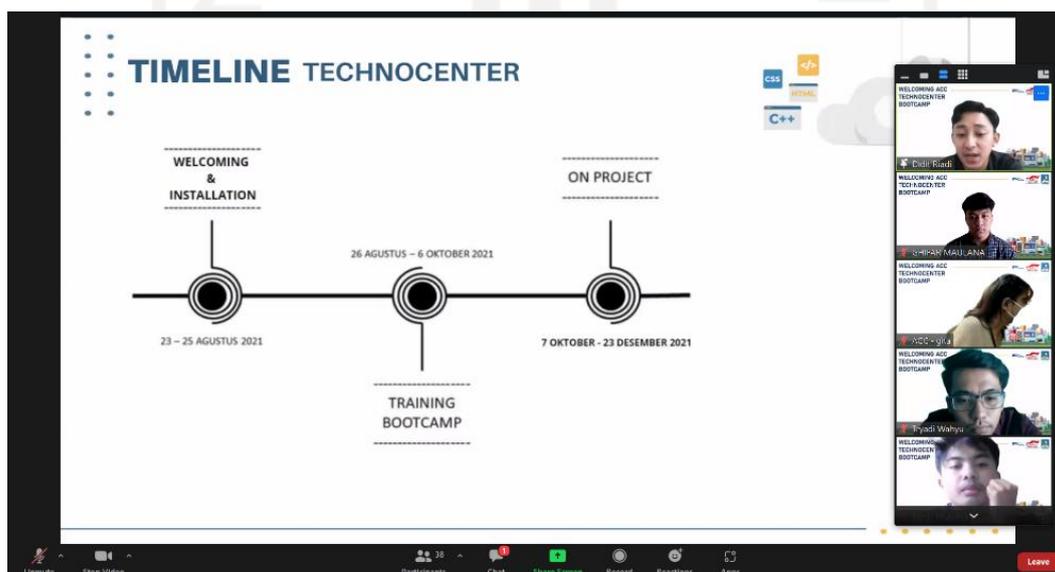
PELAKSANAAN MAGANG

3.1 Manajemen Proyek

Dalam pelaksanaan proyek, tahapan manajemen proyek menggambarkan proses untuk mencapai suatu kesuksesan dalam suatu proyek. Tahapan manajemen proyek ini menjadi suatu rangkaian yang akan dilalui dari proses awal pelaksanaan proyek hingga proses telah selesai. Tentunya salah satu tujuan dari manajemen proyek adalah mengurangi hambatan yang menyebabkan terganggunya proses pelaksanaan pengerjaan proyek.

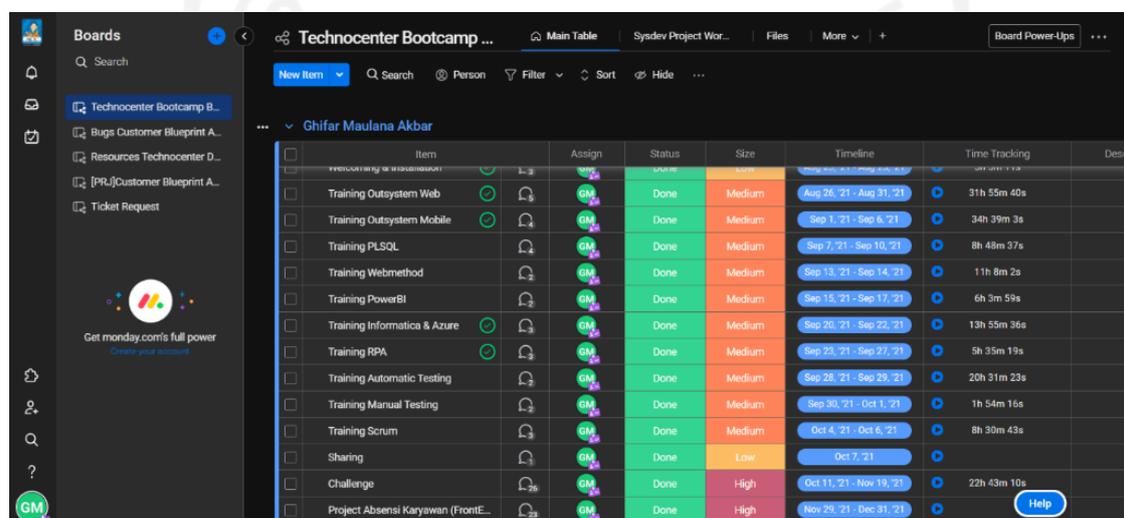
3.1.1 Platform Manajemen Proyek

Pelaksanaan magang, pengerjaan proyek, dan komunikasi banyak dilakukan secara daring atau *online*. Penggunaan platform yang cocok untuk komunikasi tentunya sangat dibutuhkan dalam pelaksanaan magang. Selama magang, penulis dapat berinteraksi dengan mentor maupun dengan rekan magang lainnya melalui bantuan platform komunikasi *online* yaitu aplikasi *Zoom Meeting* dan *Google Meet*. Aplikasi komunikasi ini tentunya juga membantu pelaksanaan metode pengembangan *scrum* yaitu sebagai sarana pelaksanaan *standup daily meeting*. Berikut adalah pelaksanaan magang yang dilaksanakan melalui platform *Zoom Meeting* pada Gambar 3.1.



Gambar 3.1 *Zoom Meeting* selama pelaksanaan magang.

Selain itu, dokumentasi harian juga diperlukan agar mentor ataupun rekan dapat melihat bagaimana dan apa saja pekerjaan penulis selama magang berlangsung. *Monday* adalah platform yang digunakan perusahaan ACC sebagai tempat dalam pencatatan dokumentasi yang ingin atau telah dikerjakan selama magang maupun pekerjaan yang berlangsung. Selain dokumentasi mengenai apa yang dikerjakan, *Monday* nantinya juga akan digunakan sebagai tempat dilakukannya *monitoring sprint* yang sedang berlangsung dikarenakan dokumentasi *sprint backlog* aplikasi yang sedang dikerjakan juga akan dicatatkan pada platform tersebut. Pada Gambar 3.2 berikut adalah tampilan *Monday* sebagai dokumentasi pelaksanaan magang.



Item	Assign	Status	Size	Timeline	Time Tracking	Desc
Training Outsystem Web	GM	Done	Medium	Aug 26, '21 - Aug 31, '21	31h 55m 40s	
Training Outsystem Mobile	GM	Done	Medium	Sep 1, '21 - Sep 6, '21	34h 39m 3s	
Training PLSQL	GM	Done	Medium	Sep 7, '21 - Sep 10, '21	8h 48m 37s	
Training Webmethod	GM	Done	Medium	Sep 13, '21 - Sep 14, '21	11h 8m 2s	
Training PowerBI	GM	Done	Medium	Sep 15, '21 - Sep 17, '21	6h 3m 59s	
Training Informatica & Azure	GM	Done	Medium	Sep 20, '21 - Sep 22, '21	13h 55m 36s	
Training RPA	GM	Done	Medium	Sep 23, '21 - Sep 27, '21	5h 35m 19s	
Training Automatic Testing	GM	Done	Medium	Sep 28, '21 - Sep 29, '21	20h 31m 23s	
Training Manual Testing	GM	Done	Medium	Sep 30, '21 - Oct 6, '21	1h 54m 16s	
Training Scrum	GM	Done	Medium	Oct 4, '21 - Oct 6, '21	8h 30m 43s	
Sharing	GM	Done	Low	Oct 7, '21		
Challenge	GM	Done	High	Oct 11, '21 - Nov 19, '21	22h 43m 10s	
Project Absensi Karyawan (FrontE...	GM	Done	High	Nov 29, '21 - Dec 31, '21		

Gambar 3.2 Tampilan *Monday* sebagai tempat dokumentasi pekerjaan yang dilakukan.

3.1.2 Metode Pengembangan Aplikasi

Selama pengerjaan proyek, penulis beserta tim menerapkan metode pengembangan *scrum* dalam mengembangkan proyek sesuai dengan metode pengembangan yang telah diterapkan oleh perusahaan ACC. Dengan metode pengembangan *scrum*, pengerjaan proyek menjadi lebih tertata dan teratur dengan baik dari segi struktur dan waktu pengerjaan. Kemudian dengan adanya *daily report* atau laporan harian membuat progres pengerjaan setiap anggota menjadi transparan dan mudah untuk dipantau oleh *scrum master* atau *project manager*. Pada Gambar 3.3 di bawah ini yaitu tampilan *Monday* selama pelaksanaan *scrum*.

Main Table							
Documents Completeness							
Item	BSA	Status	Link Document	Timeline	Assign to	Last Up	
FD	[BSA Icon]			[Timeline]			NR 9 m
+ Add Item							
User Stories							
Item	BSA	Status	Link Document	Timeline	Assign to	Last Up	
1. User all login (Info segmentasi) : dapat ...	[BSA Icon]	Done		[Timeline]			RK 7 m
2. User login sales (simulasi) : Pada simul...	[BSA Icon]	Done		[Timeline]			RK 7 m
3. User login sales (simulasi) : Ketika sim...	[BSA Icon]	Done		[Timeline]			RK 7 m
4. User login sales (simulasi) : saat simula...	[BSA Icon]	Done		[Timeline]			RK 7 m
5. User login sales (simulasi) : sales dapat...	[BSA Icon]	Done		[Timeline]			RK 7 m

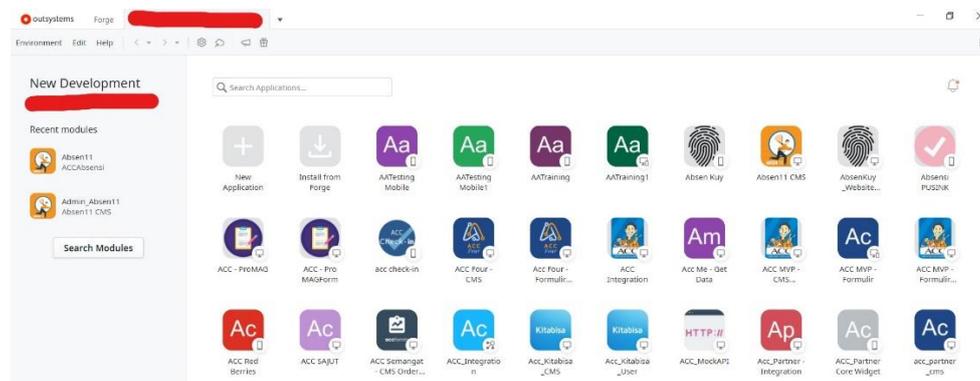
Gambar 3.3 Tampilan Monday dokumentasi pelaksanaan *scrum*.

3.1.3 Platform Pengembangan *Front-End* Aplikasi

Pada pengerjaan proyek, penulis bertanggung jawab dalam membuat antarmuka aplikasi yang disebut sebagai *front-end developer*. Sebagai *front-end developer* tentunya banyak *framework* yang dapat digunakan dalam pengembangan aplikasi, contohnya seperti Vue.js, React, Angular dan *front-end framework* lainnya. Namun kali ini perusahaan ACC tidak menggunakan *front-end framework* yang biasa digunakan, melainkan menggunakan platform *low-code* Outsystems alih-alih menggunakan *front-end framework* yang menggunakan *hard-code*. Penggunaan platform *low-code* Outsystems ini digunakan oleh penulis untuk membuat antarmuka aplikasi atau disebut dengan *front-end* dari aplikasi. Platform Outsystems ini tentunya sangat memudahkan bagi para developer, dikarenakan fitur *drag and drop* yang disuguhkan dapat meminimalkan pengkodean secara manual atau *hard-coding* yang salah satu manfaatnya dapat mempersingkat waktu pengembangan aplikasi.

Low-code development platform atau Platform *low-code* dalam pengembangan aplikasi tentunya saat ini sudah banyak diciptakan oleh para *engineer*. Di antaranya seperti platform *low-code* Creatio, Quixy, Visual LANSA, Retool dan *low-code development platform* lainnya. Di antara banyaknya platform *low-code* tersebut, perusahaan Astra Credit Companies memilih menggunakan platform dari Outsystems. Keuntungan yang didapatkan dengan menggunakan Outsystems yaitu mengembangkan dengan kecepatan tinggi dengan *low-coding* namun berkinerja tinggi, pengembangan sederhana namun meningkatkan produktivitas tim, dan hanya dengan menekan sekali klik dapat menerapkan perubahan pada aplikasi yang aman dan bebas

dari cacat. Dari hal tersebut, penulis menggunakan platform Outsystems sebagai platform dalam mengembangkan aplikasi dikarenakan kebijakan dari perusahaan Astra Credit Companies itu sendiri. Berikut adalah tampilan *environment* dari Outsystems perusahaan ACC pada Gambar 3.4.



Gambar 3.4 Tampilan *environment* Outsystems ACC

3.1.4 Metode Pengujian Sistem

Setelah pengembangan aplikasi telah selesai dilakukan, pengujian terhadap aplikasi tersebut dapat dilaksanakan. Pengujian aplikasi dilakukan oleh seorang *Quality Control* untuk memastikan kualitas dari suatu aplikasi. Terdapat jenis-jenis pengujian pada *software*, antara lain yaitu *system testing*, *unit testing*, *integration testing*, *usability testing*, *performance testing*, *smoke testing*, *user acceptance testing* dan jenis pengujian pada *software* lainnya. Pada perusahaan ACC tentunya juga sudah menerapkan jenis pengujian apa yang dibutuhkan untuk sistem-sistem yang dikembangkan di perusahaan tersebut. Menurut kebijakan dari perusahaan, *system integration testing* dan *user acceptance testing* adalah jenis pengujian yang dibutuhkan oleh perusahaan. Nantinya, aplikasi yang telah dikembangkan oleh penulis tentunya juga akan melewati fase pengujian sistem dengan menggunakan jenis pengujian yang telah diterapkan di perusahaan.

3.2 ACCAbsensi

ACCAbsensi adalah aplikasi berbasis *mobile* yang diharapkan menjadi salah satu solusi dalam memecahkan permasalahan yang terjadi di perusahaan ACC. Aplikasi tersebut diberi nama ACCAbsensi dikarenakan penulis hanya melaksanakan perintah dari perusahaan, namun fungsi aplikasi yang disuguhkan tidak seperti nama dari aplikasi tersebut, melainkan yang

dimaksud adalah aplikasi untuk presensi. Dari aplikasi ini nantinya pencatatan terhadap kehadiran kerja yang dilakukan secara *online* dari karyawan akan disimpan. Selain itu, terdapat fitur lainnya seperti pelaporan tugas-tugas yang dikerjakan oleh setiap karyawan.

Dengan menggunakan platform *low-code* Ousystems, proyek ini mulai dikembangkan oleh penulis dan tim pada tanggal 11 Oktober 2021. Pengembangan proyek ini telah melalui dua masa *sprint* yaitu *sprint* pertama pada tanggal 11 Oktober 2021 hingga 19 November 2021, dan *sprint* kedua dilaksanakan pada tanggal 29 November 2021 hingga 31 Desember 2021. Pengembangan dari proyek berikut dipantau langsung oleh *System Analyst* dari perusahaan ACC dikarenakan sistem ini nantinya akan didalami lebih lanjut dan digunakan sebagai sistem presensi bagi internal perusahaan.

3.2.1 Proses Pengembangan Aplikasi oleh Tim

Proses *Initiate*

Fase berikut merupakan tahapan awal dari proses pengembangan yang dilakukan dengan kerangka kerja *scrum*. Dalam fase ini di antaranya pembentukan tim dilakukan. Setelah tim telah terbentuk, pembuatan dan penentuan *product backlog* (daftar pekerjaan prioritas yang digunakan untuk pengembang agar sesuai *roadmap* dan *requirements*) dilakukan. Fase ini adalah fase dasar dalam mengembangkan suatu aplikasi agar selama proses pengembangan menjadi lancar.

Pada fase ini, tim dari penulis telah dibentuk berisikan dari 5 anggota. Di dalam tim tersebut terdiri dari 4 bidang yang berbeda, pada Tabel 3.1 berikut dapat dijelaskan detail dari anggota tim pengembangan berdasarkan bidang-bidangnya.

Tabel 3.1 Detail tim pengembangan

Peran	Nama	Tools
<i>Front-end Developer</i>	Ghifar Maulana	Ousystem
<i>Back-end Developer</i>	Albert Farizt dan Triyadi	Toad (PL/SQL), Postman
<i>Data Analyst</i>	Alfadeo Jeremy	Power BI, Excel
<i>Quality Control</i>	Virginia Tirza	Katalon Studio
<i>Scrum Master</i>	Virginia Tirza	Monday

Setelah tim telah dibentuk, *product backlog* dibuat guna untuk mengetahui apa saja *requirements* yang harus diselesaikan selama satu periode *sprint*. *Product Backlog* berisi ringkasan pekerjaan yang diperoleh dari permintaan pengguna. Namun dikarenakan pengembangan telah melewati 2 fase *sprint*, maka dari itu terdapat 2 *product backlog* yang telah dibentuk. Detail dari *product backlog sprint* pertama yang telah dibuat dapat dilihat pada Tabel 3.2 dan detail dari *product backlog sprint* kedua dapat dilihat pada Tabel 3.3.

Tabel 3.2 *Product backlog sprint* pertama ACCAbsensi

<i>Product Backlog</i>	<i>Sprint</i>
Sebagai <i>user</i> dapat <i>login</i> ke sistem dengan <i>username</i> dan <i>password</i>	1
Sebagai <i>user</i> dapat melakukan presensi masuk	1
Sebagai <i>user</i> dapat melakukan presensi keluar	1
Sebagai <i>user</i> dapat melihat durasi waktu bekerja	1
Sebagai <i>user</i> dapat melihat <i>to do list</i> hari ini	1
Sebagai <i>user</i> dapat melihat rangkuman presensi hari ini, durasi waktu bekerja, dan <i>to do list</i> hari ini	1
Sebagai <i>user</i> dapat melihat <i>to do list</i>	1
Sebagai <i>user</i> dapat menambah dan mengedit <i>to do list</i>	1
Sebagai <i>user</i> dapat melihat riwayat presensi dan <i>to do list</i>	1

Tabel 3.3 *Product backlog sprint* kedua ACCAbsensi

<i>Product Backlog</i>	<i>Sprint</i>
Sebagai <i>user</i> dapat melakukan <i>reset password</i>	2
Sebagai <i>user</i> dapat melihat <i>password</i> yang diisi saat <i>login</i>	2
Sebagai <i>user</i> dapat menambahkan foto profil	2
Sebagai <i>user Lead</i> dapat melakukan <i>approval</i> pada presensi <i>Staff</i>	2
Sebagai <i>user Head</i> dapat melakukan <i>approval</i> pada presensi <i>Lead</i>	2

Proses *Plan and Estimate*

Fase berikutnya adalah fase di mana tujuan utama dari dilakukannya fase ini yaitu perencanaan. Setelah *product backlog* di tentukan, kemudian pada fase ini *user story* dituliskan beserta dengan penjabaran *task-task* pada tiap *user story*. Tidak hanya itu, penilaian terhadap setiap *user story* dan *task* diperlukan untuk menentukan tingkat prioritas dalam pengembangan aplikasi yang sedang dikerjakan. Jika dirasa *user story* sudah direncanakan dengan baik, selanjutnya yaitu membuat estimasi pengerjaan dalam pengembangan yang nantinya akan masuk kedalam perencanaan *Sprint*.

Setelah mendapatkan hasil dari *product backlog*, data tersebut akan dibahas dan dirancang kembali yang kemudian dibentuk menjadi *Sprint backlog*. Dari *Sprint backlog* tersebut, semua perencanaan *task-task* dan penjabarannya ditentukan dengan tujuan untuk memperjelas masing-masing *task* yang harus dikerjakan, dimulai dari skala prioritas, hingga durasi waktu pengerjaan.

Dalam fase ini, seluruh anggota di dalam *team development* terlibat dalam proses perencanaan dan estimasi, dikarenakan seluruh anggota di dalam *team development* berhak menentukan estimasi dalam mengerjakan masing-masing *task* sesuai dengan tingkat kesulitan dari *task* tersebut. Proses perencanaan dan estimasi ini dilakukan melalui rapat yang diadakan

oleh *Scrum Master* dengan seluruh anggota *team development*. Melalui platform pertemuan daring *Zoom Meeting*, fase perencanaan dan estimasi dapat dilakukan. Tentunya hasil akhir dari fase ini membutuhkan pertunjukan dari *Scrum Master* terlebih dahulu agar *Scrum* tetap berlangsung dengan tepat dan lancar.

Pada Tabel 3.4 berikut, hasil dari *Sprint backlog* berdasarkan *product backlog sprint* pertama yang telah dibuat. Dan pada Tabel 3.5 berikut, hasil dari *Sprint backlog* berdasarkan *product backlog sprint* kedua yang telah dibuat. *Sprint backlog* ini nantinya akan dicatatkan kedalam platform *Monday* sebagai tempat *scrum master* untuk memantau pekerjaan yang belum, sedang, ataupun yang sudah dikerjakan.

Tabel 3.4 *Sprint backlog* dari *product backlog sprint* pertama

<i>Product Backlog</i>	<i>Story Point</i>	<i>Status</i>	<i>Timeline</i>
Sebagai user dapat login ke sistem dengan username dan password	3	<i>Not in development</i>	<i>Nov 19</i>
Sebagai user dapat melakukan presensi masuk	5	<i>Not in development</i>	<i>Nov 19</i>
Sebagai user dapat melakukan presensi keluar	5	<i>Not in development</i>	<i>Nov 19</i>
Sebagai user dapat melihat durasi waktu bekerja hari ini	1	<i>Not in development</i>	<i>Nov 19</i>
Sebagai user dapat melihat <i>to do list</i> hari ini	1	<i>Not in development</i>	<i>Nov 19</i>
Sebagai user dapat melihat rangkuman presensi hari ini	1	<i>Not in development</i>	<i>Nov 19</i>
Sebagai user dapat melihat <i>to do list</i>	5	<i>Not in development</i>	<i>Nov 19</i>
Sebagai user dapat menambah dan mengedit <i>to do list</i>	5	<i>Not in development</i>	<i>Nov 19</i>
Sebagai user dapat melihat riwayat presensi dan <i>to do list</i>	3	<i>Not in development</i>	<i>Nov 19</i>

Tabel 3.5 *Sprint backlog* dari *product backlog sprint* kedua

<i>Product Backlog</i>	<i>Story Point</i>	<i>Status</i>	<i>Timeline</i>
Sebagai user dapat melakukan <i>reset password</i>	5	<i>Not in development</i>	<i>Dec 31</i>
Sebagai user dapat melihat <i>password</i> yang diisi saat <i>login</i>	1	<i>Not in development</i>	<i>Dec 31</i>

Sebagai <i>user</i> dapat menambahkan foto profil	3	<i>Not in development</i>	<i>Dec 31</i>
Sebagai <i>user Lead</i> dapat melakukan <i>approval</i> pada presensi <i>Staff</i>	5	<i>Not in development</i>	<i>Dec 31</i>
Sebagai <i>user Head</i> dapat melakukan <i>approval</i> pada presensi <i>Lead</i>	5	<i>Not in development</i>	<i>Dec 31</i>

Dikarenakan *sprint backlog* ini dibuat dan diperuntukkan sebagai daftar *task* yang harus dikerjakan dalam pengembangan, maka dari itu *status* dituliskan sebagai *Not in development* pada saat awal pembuatan *sprint backlog*. Setelah per *task* nya selesai dikerjakan, maka status *Not in development* akan berubah menjadi *done*.

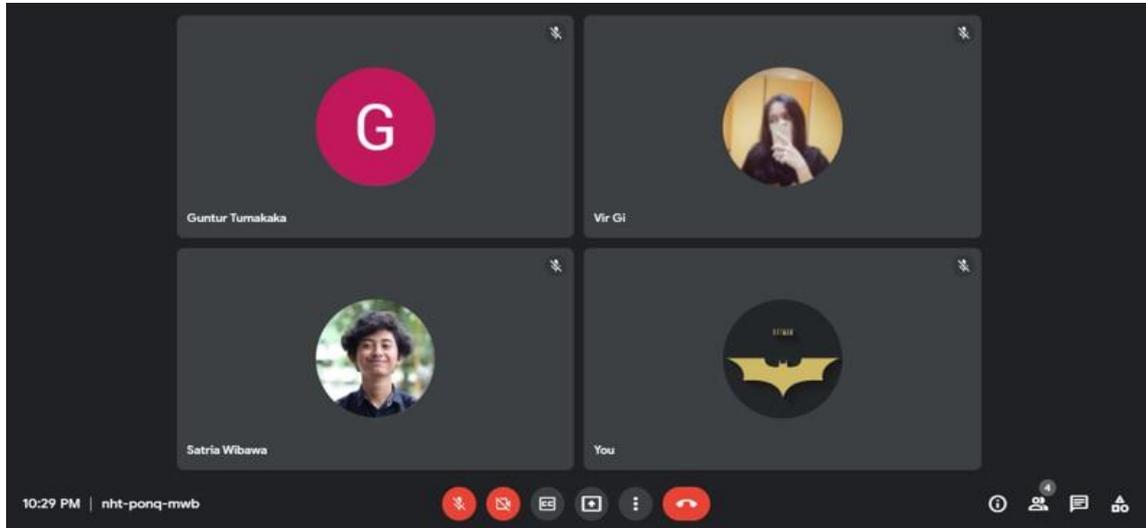
Pada tabel di atas, juga terdapat *story point* yang dimaksud sebagai ukuran estimasi untuk mengerjakan sebuah *product backlog* atau sebuah kerjaan. Estimasi terhadap rumitnya, resiko, durasi, dan banyaknya sebuah pekerjaan. Angka tersebut ditentukan dengan teknik *planning poker*, yang poinnya mengikuti pola *fibonacci*, yaitu 1,3,5,8,13,21 dst. Semakin tinggi angka *story point*, maka yang dimaksud yaitu semakin tinggi juga tingkat kesulitan dalam mengerjakan *product backlog* tersebut.

Proses Implement

Fase ini merupakan fase dalam mengeksekusi setiap *task* yang telah didefinisikan dan dibentuk sebelumnya pada fase *plan and estimate*. Pengembangan dan segala aktivitas-aktivitas dalam upaya mengembangkan suatu aplikasi dilakukan pada fase *implement*. Penulis yang bertugas sebagai *front-end developer* mengeksekusi setiap *task-task* yang telah ditentukan yaitu untuk membuat dan mengembangkan antarmuka dari aplikasi ACCAbsensi. Penjelasan lengkap mengenai proses pembuatan antarmuka aplikasi ACCAbsensi akan dijelaskan pada proses pengembangan aplikasi oleh penulis.

Pada fase ini, *daily meeting* dilaksanakan dengan tujuan untuk mengevaluasi semua pekerjaan yang telah dilakukan oleh setiap anggota tim dengan menggunakan aplikasi *Zoom Meeting*. Selain itu, seluruh masalah yang ditemui selama implementasi dapat dibicarakan dalam *daily meeting* tersebut dengan tujuan mencari solusi yang tepat agar *sprint* berjalan dengan lancar. Namun, pelaksanaan *daily meeting* tidak berjalan secara maksimal ketika dilaksanakan secara *online* atau daring. Hal tersebut diakibatkan dari ketidaksamaan aktivitas yang dilakukan oleh masing-masing anggota tim membuat pelaksanaan *daily meeting* tidak

dapat berjalan setiap harinya selama *sprint* berlangsung. Selain itu, juga berdampak pada proses evaluasi yang tidak dapat berjalan secara maksimal. Berikut pada Gambar 3.5 adalah pelaksanaan *daily meeting* yang dilakukan pada platform *Google Meeting*.



Gambar 3.5 Tampilan *Google Meet* saat pelaksanaan *daily meeting*.

Dalam proses fase *implement*, semua anggota *team development* terlibat kecuali pada bidang *Quality Control* dikarenakan bidang tersebut terlibat pada fase berikutnya yaitu fase *review and restropect*. Pada proses fase *implement* yang terlibat di dalam beserta tugasnya antara lain dapat dilihat pada Tabel 3.6 berikut.

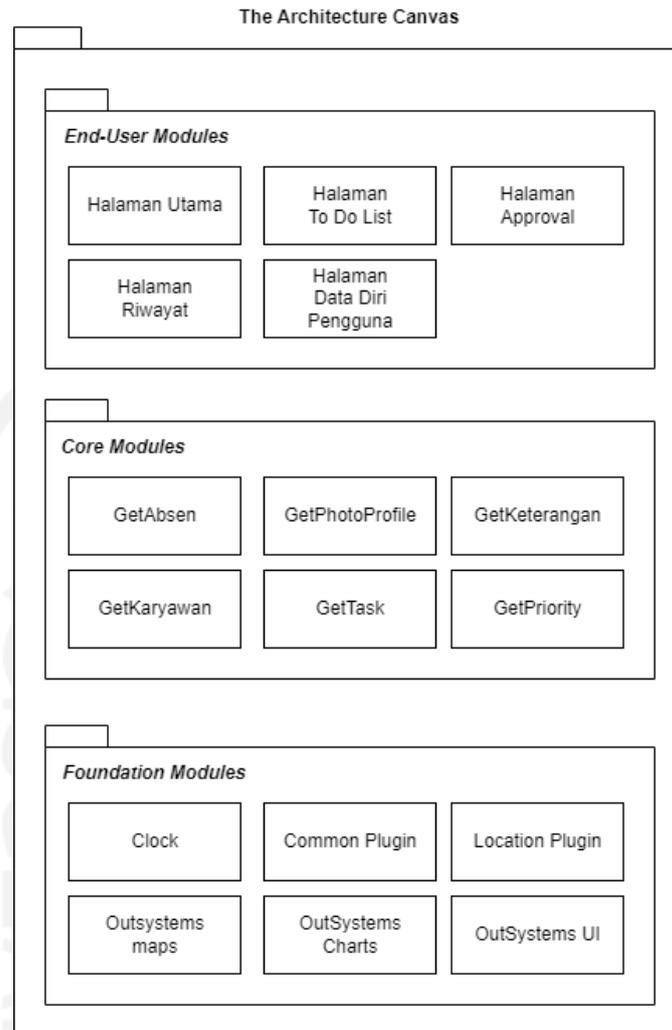
Tabel 3.6 Penjelasan pelaksanaan tugas tiap bidangnya pada proses *implement*

Siapa yang terlibat	Tugas
Front-end developer (Ghifar Maulana)	<ul style="list-style-type: none"> Antarmuka aplikasi serta <i>logic</i> di dalam aplikasi sesuai dengan <i>requirements</i>. Selain itu juga mengintegrasikan data melalui <i>API</i> yang telah dibuat oleh tim <i>Back-end</i> untuk disimpan pada <i>database</i> yang dimiliki perusahaan.

	<ul style="list-style-type: none"> Platform <i>Outsystems</i> adalah platform yang digunakan untuk membuat antarmuka dari aplikasi.
<p><i>Back-end developer</i> (Albert Farizt dan Triyadi)</p>	<ul style="list-style-type: none"> Membuat perancangan <i>database</i> beserta pembuatan <i>API</i> agar dapat terintegrasi dengan antarmuka aplikasi. Platform <i>TOAD Database Developer Tools</i> adalah platform yang digunakan untuk perancangan <i>database</i> dari aplikasi.
<p><i>Data Analyst</i> (Alfadeo Jeremy)</p>	<ul style="list-style-type: none"> Mengumpulkan data yang tersimpan dalam aplikasi dan mengolah data tersebut agar menjadi lebih informatif. Platform <i>Power BI</i> adalah platform yang digunakan untuk pengolahan data agar menjadi lebih informatif.

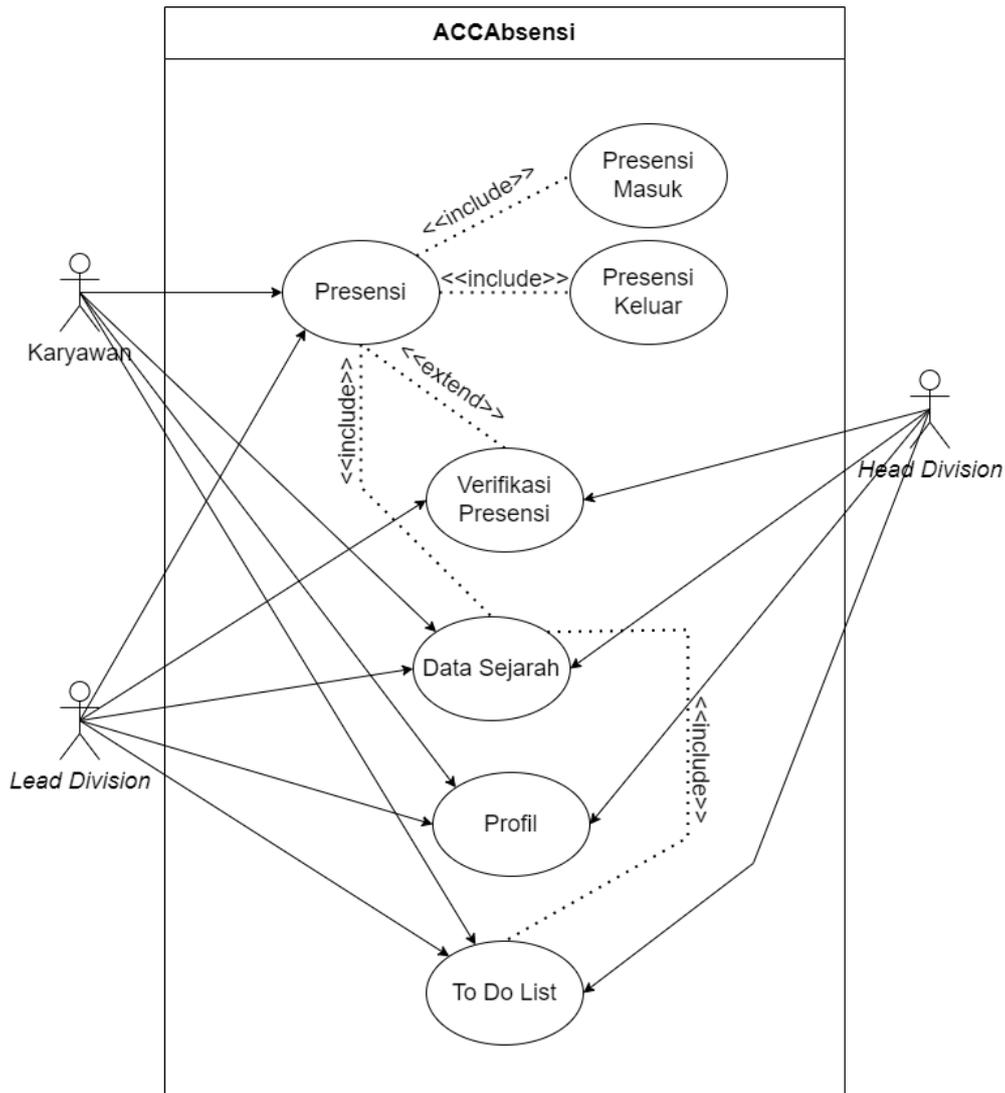
Fase *implement* merupakan fase terpenting di mana aplikasi dibuat dan dikembangkan. Semua kegiatan pengembangan aplikasi yang dilakukan harus sesuai dengan hasil dari perencanaan pada fase *plan and estimate*. *Sprint backlog* menjadi dasar dari *team development* guna menghindari pengembangan di luar dari permintaan *product owner*.

Aplikasi ACCA Absensi sendiri dikembangkan menggunakan desain arsitektur dari Outsystems yaitu *The Architecture Canvas* sehingga aplikasi terbagi menjadi *End-user Modules*, *Core Modules*, dan *Foundation Modules*. Pada *Core Modules*, data akan disimpan terlebih dahulu pada *cloud* yang telah disediakan oleh Outsystems, kemudian setelah data tersimpan pada *Core Modules*, data akan dilanjutkan menuju ke *back-end* melewati API. Berikut adalah Gambar 3.6 *the architecture canvas* yang menggambarkan arsitektur dari aplikasi ACCA Absensi berdasarkan platform pengembangan Outsystems.



Gambar 3.6 Arsitektur ACCAbsensi

Selain itu, terdapat *usecase* diagram yang menggambarkan hubungan interaksi antara sistem dengan pengguna dari aplikasi ACCAbsensi agar memudahkan dalam proses pengembangan pada fase *implement*. Pada aplikasi ACCAbsensi, terdapat 3 *role user* di dalamnya, di antaranya yaitu *role user* karyawan, *role user* head, *role user* lead. Tujuan dari dibentuknya 3 *role user* tersebut dikarenakan dibutuhkannya validasi mengenai presensi kehadiran yang telah dilakukan. *Role user* lead bertanggung jawab atas presensi yang dilakukan oleh *role user* karyawan, dan *role user* head bertanggung jawab atas presensi yang dilakukan oleh *role user* lead. Berdasarkan Gambar di bawah ini menunjukkan pengguna aplikasi ACCAbsensi oleh *role user* karyawan, *role user* lead, dan *role user* head. Pada Gambar 3.7 berikut adalah *usecase* diagram dari aplikasi ACCAbsensi.



Gambar 3.7 Usecase diagram aplikasi ACCAbsensi

Proses Review dan Restropect

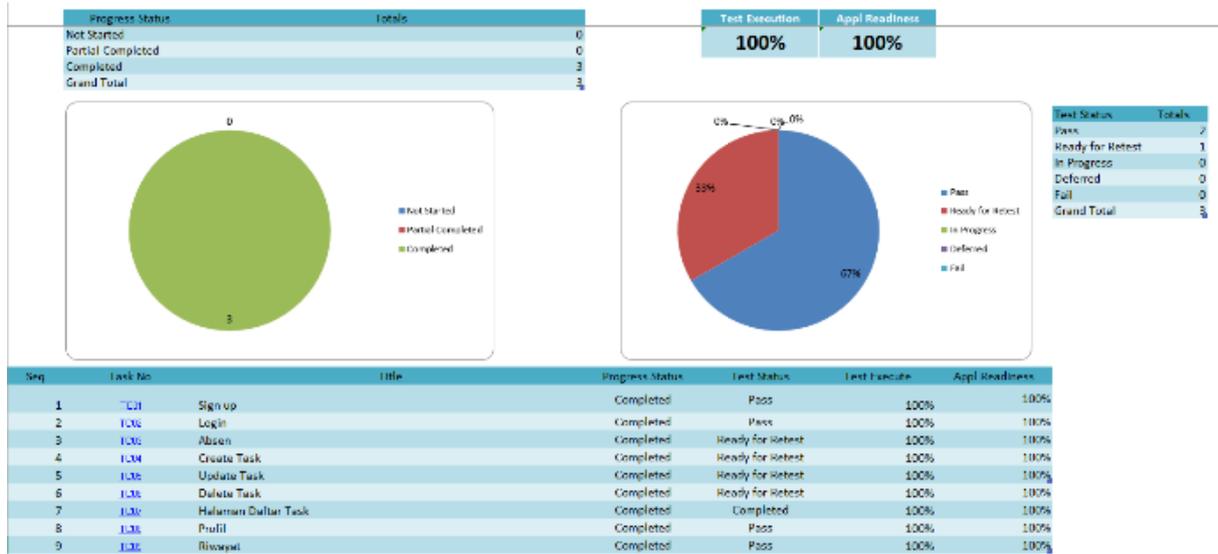
Setelah fase implemementasi dilakukan, berikut ini adalah fase dilakukannya *review* terhadap hasil pekerjaan yang telah dilakukan selama *sprint*. Namun sebelum dilakukannya *review*, *Manual Guide* harus dibuat terlebih dahulu oleh tim pengembang sebagai petunjuk untuk membantu pada saat tahap pengujian dilakukan oleh tim penguji. Berikut adalah dokumen *Manual Guide* ACCAbsensi pada Gambar 3.8.

MANUAL GUIDE <i>Absen 11</i>	
ASTRA CREDIT COMPANIES	
2021	
Versi / Version	1.1
Status / Status	Aktif
Nomor Dokumen / Document Number	
Klasifikasi / Classification	Internal
Pemilik Dokumen / Document Owner	Bilfaq Maulana

Gambar 3.8 *Manual Guide ACCAbsensi*

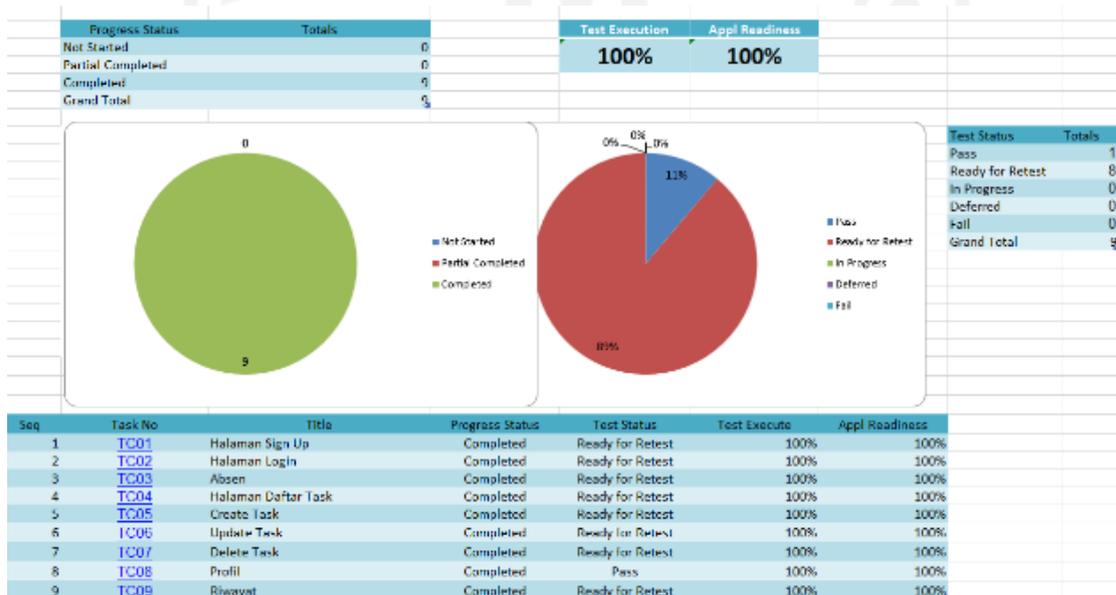
Pada fase ini, semua proses yang telah dihasilkan di-*review* kembali dengan menggunakan metode manual *testing* berdasarkan *test case* yang telah dibuat oleh tim *quality control* agar dapat ditentukan apa saja yang perlu ditingkatkan untuk proses pengembangan pada *sprint* berikutnya. *Sprint* dianggap berhasil apabila pada pengembangan yang dilakukan dapat menghasilkan *deliverable product* (Barang berwujud yang dikirimkan ke pelanggan sebagai keluaran proyek) yang sesuai dengan *user acceptance* (sistem yang dikembangkan telah sesuai dengan kebutuhan pengguna) yang telah ditentukan dan disetujui oleh *product owner*.

Hasil yang didapatkan dari proses pengujian *System Integration Testing* dari pengembangan aplikasi dapat dikatakan sukses dikarenakan mendapat 67% pengujian dianggap *pass* atau sukses. Di lain sisi mendapat 33% pengujian dianggap meragukan dan siap untuk dilakukan uji ulang selama aplikasi dijalankan. Berikut adalah ringkasan kesimpulan yang didapatkan dari hasil pengujian SIT yang dilakukan oleh tim penguji pada Gambar 3.9.



Gambar 3.9 Summary System Integration Testing ACCAbsensi

Selain itu terdapat hasil yang didapatkan dari proses pengujian *User Acceptance Testing* dari pengembangan dapat dikatakan sukses namun masih harus dilakukan pengecekan kembali dikarenakan mendapat 11% pengujian dianggap *pass* atau sukses. Dan di lain sisi mendapat 89% pengujian dianggap meragukan dan siap untuk dilakukan uji ulang selama aplikasi dijalankan. Berikut adalah kesimpulan yang didapatkan dari hasil pengujian UAT pada Gambar 3.10.



Gambar 3.10 Summary User Acceptance Testing ACCAbsensi

Berikut adalah detail mengenai keterangan pelaksanaan proses *review and restropect* yang terdapat pada Tabel 3.20.

Tabel 3.7 Detail pelaksanaan proses *review and restropect*

Siapa yang terlibat	Tugas
<i>Quality Control (Virginia Tirzha)</i>	<ul style="list-style-type: none"> • Melakukan <i>review</i> atau menguji kembali aplikasi dengan menggunakan metode manual <i>testing</i> berdasarkan <i>test case</i> yang telah dibuat. • Membuat dokumen SIT untuk menyimpulkan hasil dari pengujian aplikasi yang telah dilakukan. • Membuat dokumen UAT untuk menyimpulkan hasil dari pengujian aplikasi yang telah dilakukan.
<i>PIC Quality Control (Mutiara Caesagusta Yodashie)</i>	<ul style="list-style-type: none"> • Mendampingi dan membimbing selama pelaksanaan pengujian aplikasi.

Pada pelaksanaan pengujian *User Acceptance Testing* pada aplikasi ACCA Absensi belum terdapat *user* yang dapat menguji, karena hal tersebut UAT juga dilaksanakan oleh tim QC yang didampingi oleh *PIC* dari *Quality Control* dengan berpatokan pada *user acceptance criteria*. Hal tersebut membuat pengujian menjadi kurang maksimal dan belum terdapat temuan atau perbaikan yang diberikan atau disampaikan. Namun aplikasi tersebut dapat dilanjutkan ke fase berikutnya yaitu fase *release*.

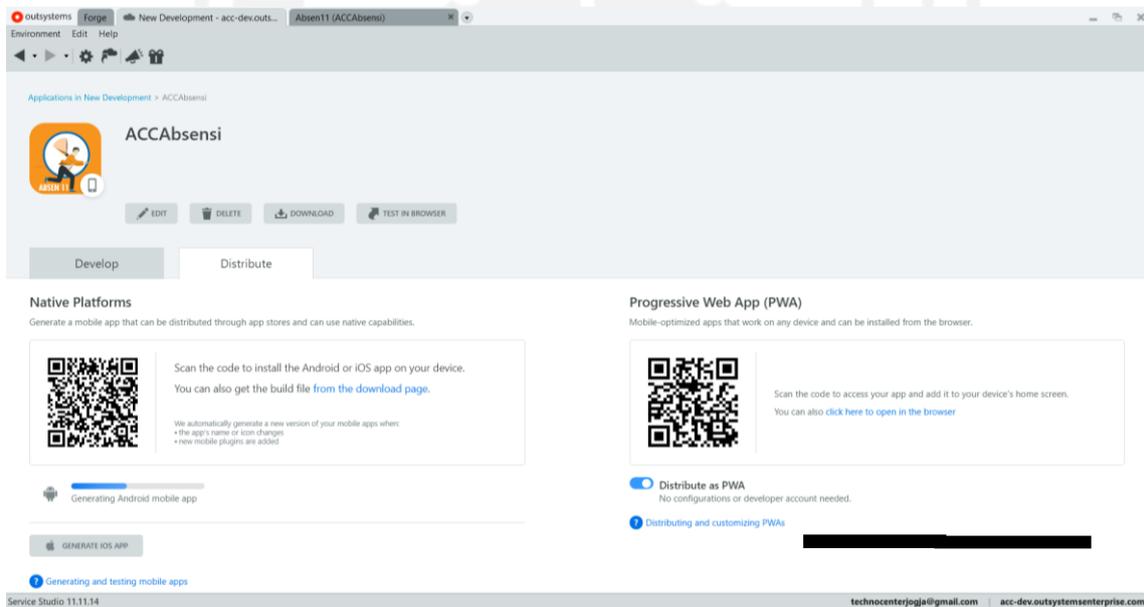
Proses Release

Fase berikut adalah fase yang terakhir dilakukan. Jika produk telah memenuhi seluruh kriteria dari *user acceptance*, maka produk aplikasi tersebut siap di *deployment* dan diberikan seluruhnya ke klien. Setelah produk sudah berhasil pada fase ini berarti *sprint* dianggap telah selesai dilaksanakan.

Deployment proyek dilakukan dengan cara membuat atau *generate debug apk* yang kemudian nantinya *file* hasil dari *generate* tersebut dapat diunduh dan dapat di *install* melalui perangkat *android* yang dimiliki. Setelah *file* telah di *install* di perangkat masing-masing pengguna, maka aplikasi dapat digunakan dengan bantuan jaringan internet.

Namun kebanyakan aplikasi yang dibuat atau di *generate* tersebut adalah untuk perangkat *android* saja, dikarenakan jika membuat atau *generate* aplikasi yang diperuntukkan bagi perangkat *ios*, maka pengembang dari aplikasi tersebut harus memiliki sertifikat khusus untuk mengembangkan aplikasi bagi perangkat *ios*, jika tidak memiliki sertifikat tersebut maka aplikasi tidak dapat di *generate*. Tetapi terdapat solusi lain bagi pengguna *ios* untuk tetap dapat mengakses aplikasi yang dikembangkan tersebut, dengan cara menggunakan PWA (*Progressive Web App*) yaitu aplikasi yang dioptimalkan untuk *smartphone* yang berfungsi di perangkat apa pun dan dapat diinstal dari browser.

Dapat dilihat pada Gambar 3.11 di bawah ini hasil dari proses *generate apk* untuk dapat diunduh *file apk* nya atau menggunakan PWA pada browser untuk mengakses aplikasi dengan cara melakukan *scan QR Code*.



Gambar 3.11 *Generate file apk* proyek ACCAbsensi

Setelah proses *generate* telah selesai dilakukan, *file apk* tersebut diberikan kepada pihak perusahaan guna nantinya akan ditindak lebih lanjut. Pada fase ini, tim dan penulis telah dinyatakan selesai dari pengerjaan proyek tersebut dikarenakan telah sampai pada tahap *release*.

3.2.2 Proses Pengembangan Aplikasi oleh Penulis

Proses pengembangan aplikasi yang dilakukan oleh penulis dapat dilakukan setelah tim *back-end* telah merancang *database* beserta *API* untuk mengintegrasikan antara *front-end*

aplikasi dan *back-end* aplikasi. Selanjutnya adalah bagian implementasi terhadap pengembangan sistem yang akan menampilkan dan menjelaskan antarmuka dan fungsi yang telah dibuat oleh tim *front-end*. Selain itu juga terdapat penjelasan pembuatan *logic* yang terdapat di dalam setiap antarmukanya.

Analisis Kebutuhan Aplikasi

Dalam mendukung kinerja aplikasi, analisis kebutuhan sistem sangat dibutuhkan apakah aplikasi yang dibuat telah sesuai dengan kebutuhan atau belum. Karena kebutuhan sistem akan mendukung tercapainya tujuan dari suatu aplikasi. Agar sebuah sistem dapat berjalan dengan baik dan mempunyai kemampuan yang memadai. Perangkat keras yang dibutuhkan dalam pembuatan aplikasi ini yaitu:

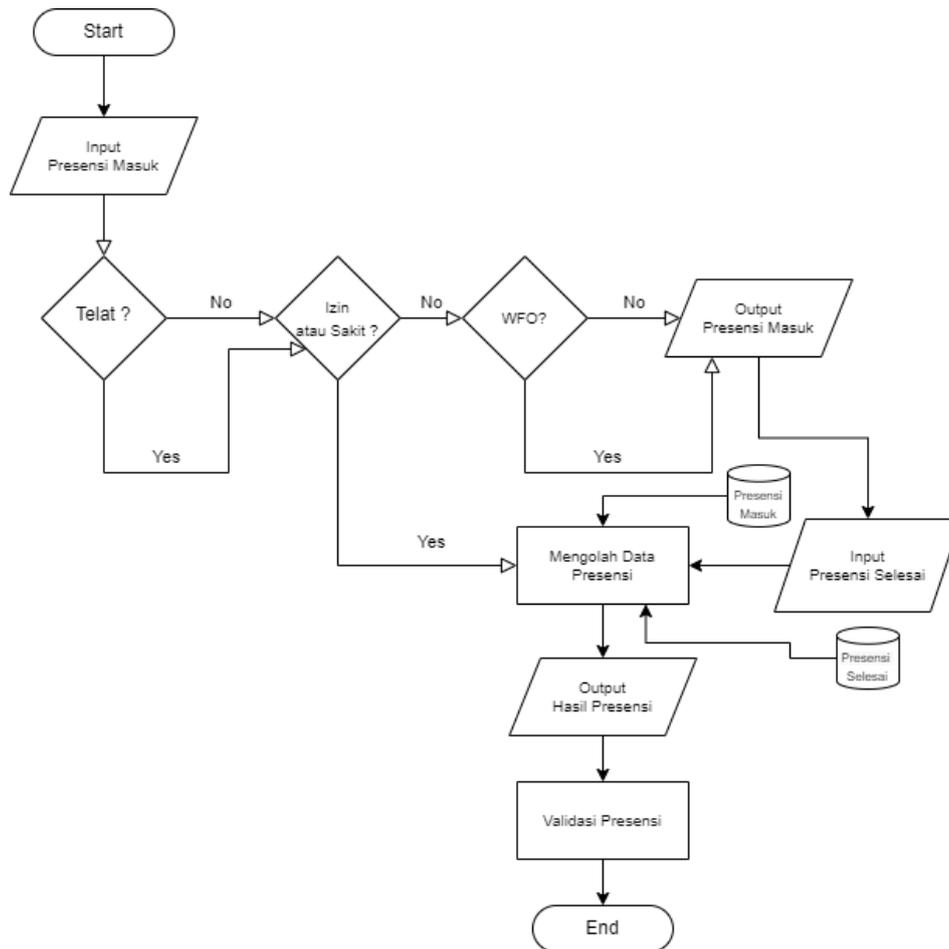
- a. Laptop
- b. Intel Core i7 11th Gen (16 CPUs) 2.30 GHz
- c. Ram 16 Gb
- d. SSD 512 GB

Perangkat lunak yang digunakan untuk mendukung dalam pembuatan dan pengopresaian aplikasi ACCAbsensi adalah sebagai berikut:

- a. Sistem Operasi Windows 11 Home Single Language
- b. Microsoft .NET Framework 4.7.2
- c. Outsystems 11

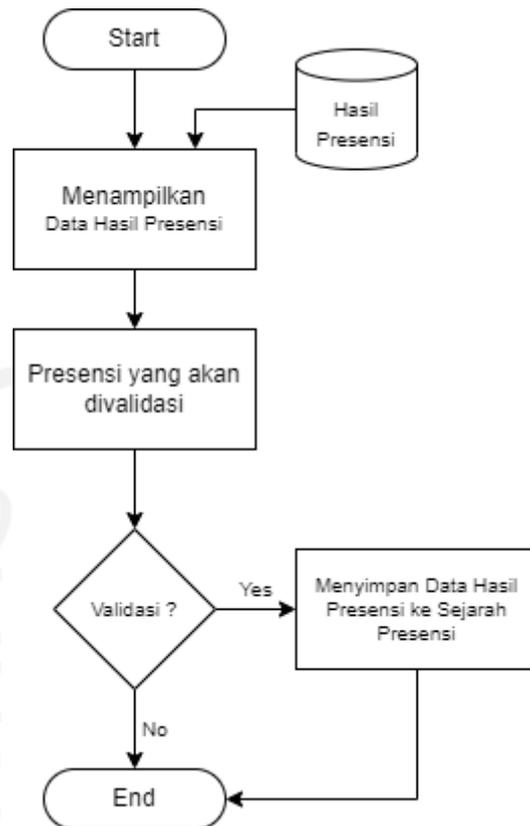
Alur Sistem

Perancangan aplikasi diawali dari proses yang digambarkan dalam diagram alur atau *flowchart*. Perancangan ini dilakukan untuk mengetahui kondisi dari sistem secara umum. Dalam perancangan aplikasi ini akan membahas mengenai kerangka aplikasi dari tahapan melakukan presensi, tahapan memvalidasi hasil presensi yang telah dilakukan, dan tahapan dalam membuat daftar kerja yang akan dilakukan. Dimulai dari tahapan melakukan presensi yang digambarkan dalam *flowchart* berikut pada Gambar 3.12.



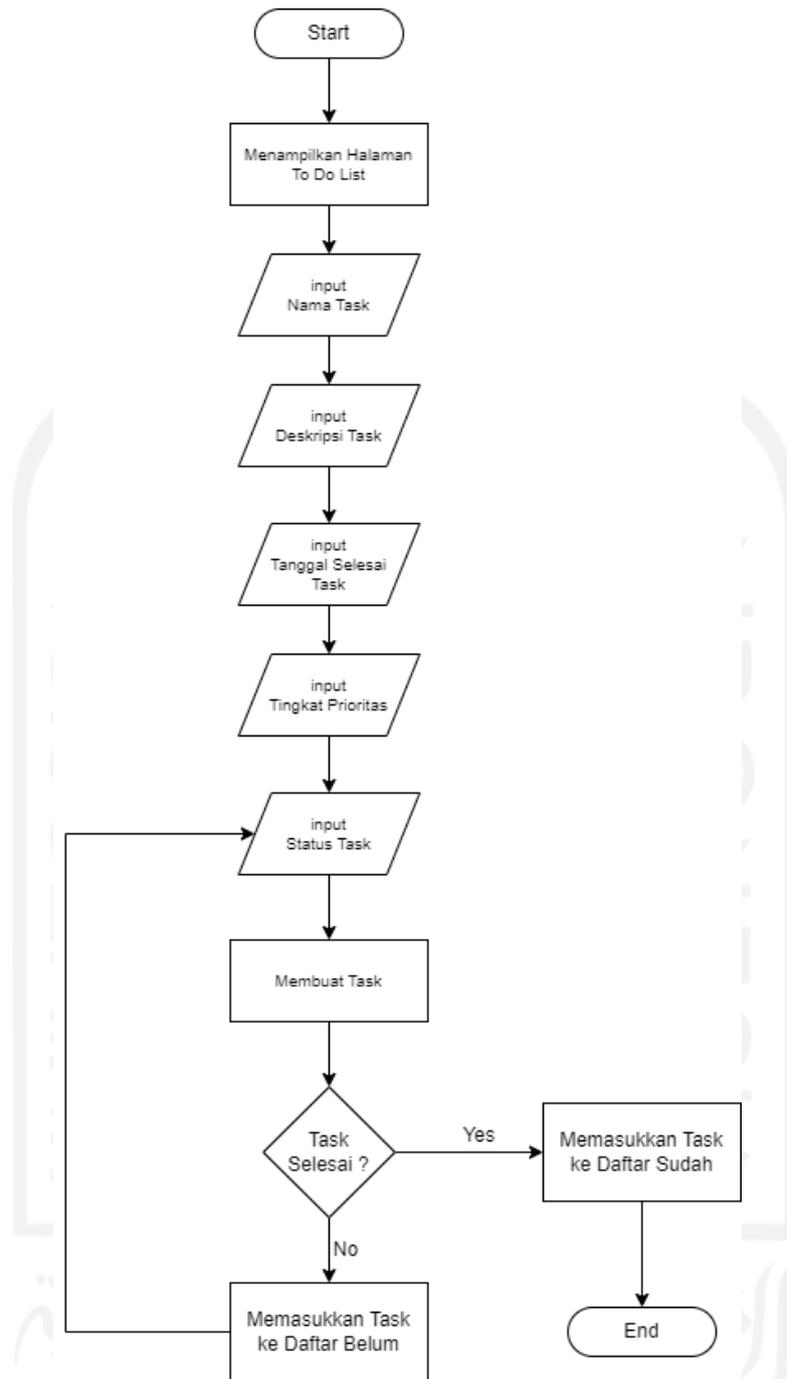
Gambar 3.12 *Flowchart* tahapan melakukan presensi

Dari Gambar 3.12 dapat dijelaskan tahapan dalam melakukan presensi yang diawali dari proses pengambilan data presensi masuk. Jika telah melakukan presensi masuk, *output* presensi masuk yang telah dilakukan akan disimpan sementara terlebih dahulu. Setelah waktu selesai kerja telah tiba, maka proses input presensi selesai kerja dapat dilakukan. Kemudian data presensi masuk dan presensi selesai akan diolah untuk menghasilkan sebuah *output* hasil presensi. Terakhir, yaitu adalah proses validasi hasil dari presensi yang telah dilakukan. Alur kerja sistem berikut hanya dapat diakses oleh *role user* karyawan dan *role user head* saja.



Gambar 3.13 *Flowchart* tahapan memvalidasi hasil presensi

Dari Gambar 3.13 berikut dapat dijelaskan tahapan dalam melakukan validasi terhadap hasil presensi yang telah dilakukan. Diawali dari menampilkan data hasil presensi yang telah dilakukan oleh karyawan atau *lead*. Kemudian pemilihan data presensi yang ingin divalidasi. Jika proses dapat divalidasi, data hasil presensi tersebut akan disimpan pada halaman riwayat presensi aplikasi. Jika hasil presensi tidak dapat divalidasi, maka alur tahapan memvalidasi hasil presensi selesai. Alur sistem ini hanya dapat dilakukan oleh *role user lead* dan *role user head*.



Gambar 3.14 *Flowchart* tahapan membuat daftar kerja

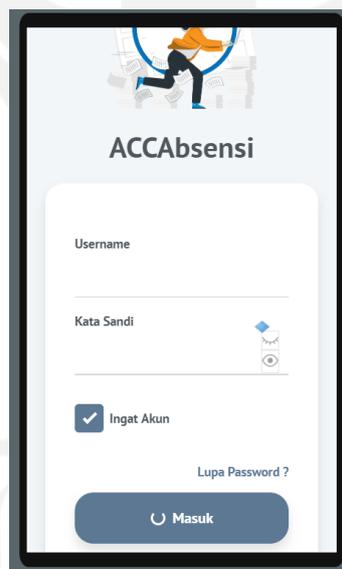
Dari Gambar 3.14 berikut dijelaskan mengenai tahapan dalam membuat daftar kerja. Dimulai dari membuka halaman *To Do List* terlebih dahulu, kemudian menginput semua data yang diperlukan dari nama *task*, deskripsi *task*, tanggal selesai *task*, tingkat prioritas, dan status *task*. Kemudian setelah itu *user* dapat menyelesaikan tugas yang diinput kan tersebut terlebih dahulu. Jika sudah menyelesaikannya, maka dapat memasukkan daftar kerja tersebut ke daftar selesai pada halaman *To Do List*.

Desain Aplikasi

Setelah alur sistem telah dibentuk, selanjutnya adalah menentukan tema yang akan digunakan untuk antarmuka yang akan dibuat. Dimulai dari menentukan pemilihan warna untuk tema dari aplikasi. ACCAbsensi menentukan tema warna berdasarkan warna dari logo perusahaan ACC. Kemudian logo yang dipakai ialah logo yang dibuat oleh penulis dengan warna dasar logo perusahaan ACC. Detail mengenai pembuatan halaman akan dijelaskan berdasarkan masing-masing halaman yang dibuat oleh penulis.

Pembuatan Halaman *Login*

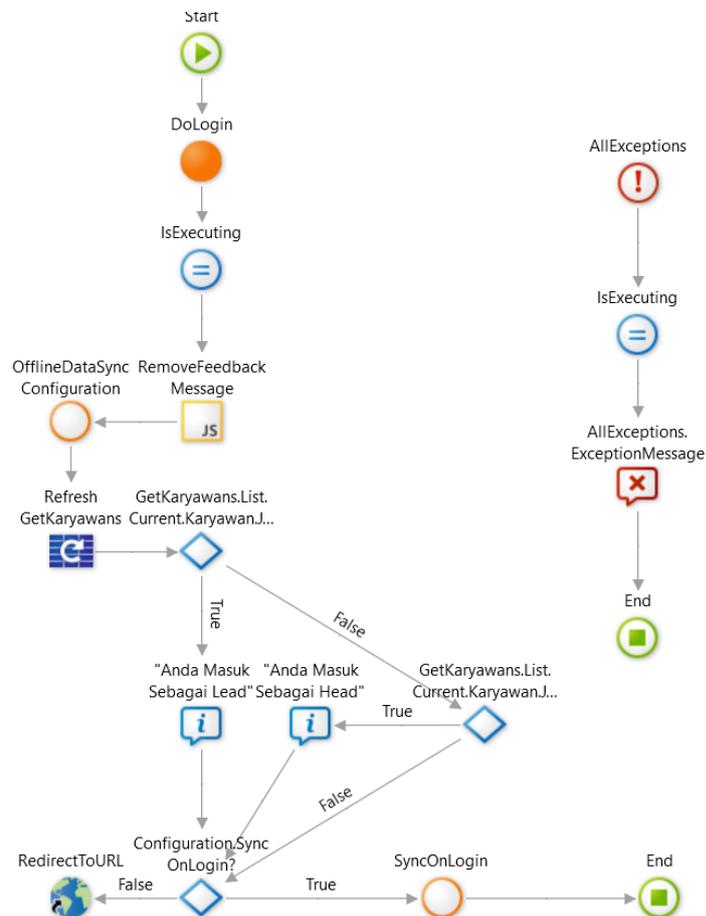
Sebelum masuk pada halaman utama, tentunya dibuatnya fitur *login* yang digunakan sebagai proses identifikasi pengguna. Proses *login* ini terdiri dari pengisian *username* pengguna dan *password*. Berikut adalah antarmuka *login* yang dibuat oleh penulis sebagai tim *front-end* pada Gambar 3.3.



Gambar 3.15 Halaman *login* ACCAbsensi

Pada *button* masuk yang terdapat pada halaman login, terdapat *function logic* yang berfungsi untuk melakukan identifikasi *username* dan *password* dari pengguna. *Logic* pada *button* masuk tersebut akan mencocokkan *username* dan *password* yang berada pada *database* melalui *API* yang terdapat pada *core modules GetKaryawan*. Namun terdapat *logic* tambahan yaitu sebagai persyaratan untuk mengetahui *role* dari pengguna yang melakukan *login*. Terdapat 3 *role* yang dapat melakukan *login* ke dalam aplikasi, yaitu *role* karyawan

(*staff*), *Lead* (*Leader* atau pemimpin dari divisi), dan *Head* (Kepala dari divisi). Selain itu terdapat *exception* atau notifikasi di atas layar bila *username* dan *password* tidak ada yang sesuai pada *database*. Detail dari pembuatan *Logic* pada *button* masuk dapat dilihat pada Gambar 3.4.



Gambar 3.16 Alur fungsi *logic button* masuk pada halaman login

Berikut adalah detail mengenai keterangan pelaksanaan pembuatan fitur *login* yang terdapat pada Tabel 3.7. Selain itu terdapat detail capaian yang telah dilakukan setelah pembuatan fitur login selesai dilakukan yang terdapat pada Tabel 3.8 dan Tabel 3.9.

Tabel 3.8 Detail keterangan pembuatan fitur *login*

Siapa yang terlibat	Keterangan Task	Modules	PIC
Front-end developer (Ghifar Maulana)	<ul style="list-style-type: none"> Membuat antarmuka <i>Login</i> serta <i>Consume API Post</i> 	Foundation Modules	Yan Pradana

	<p><i>Method Login</i> yang telah dibuat.</p> <ul style="list-style-type: none"> • Menambahkan fitur lihat <i>password</i>. • Menambahkan fitur reset <i>password</i>. 	<ul style="list-style-type: none"> • <i>Common Plugin</i> • <i>Outsystems UI</i> 	
--	--	--	--

Tabel 3.9 Capaian *product backlog* pada *sprint* pertama setelah pembuatan halaman *login*

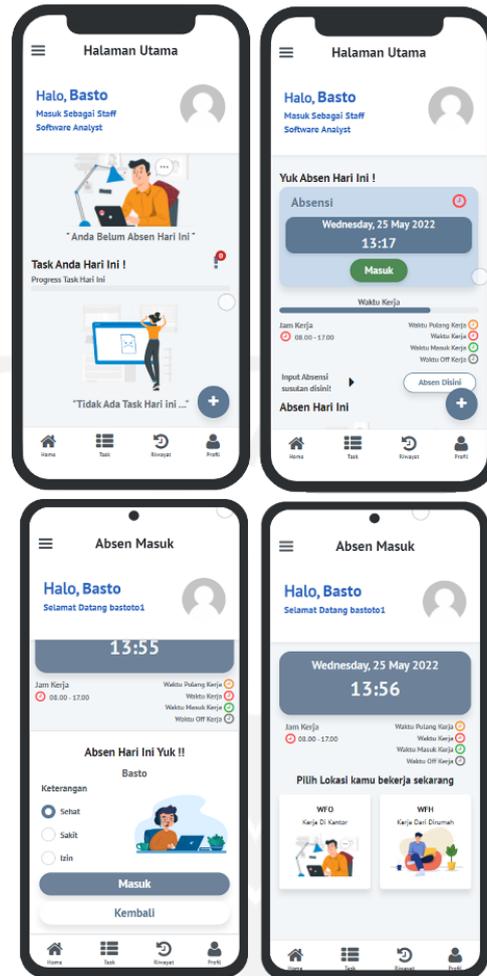
<i>Product Backlog</i>	<i>Story Point</i>	<i>Status</i>	<i>Timeline</i>
Sebagai <i>user</i> dapat <i>login</i> ke sistem dengan <i>username</i> dan <i>password</i>	3	<i>Done</i>	18 Okt 2021 – 22 Okt 2021

Tabel 3.10 Capaian *product backlog* pada *sprint* kedua setelah pembuatan halaman *login*

<i>Product Backlog</i>	<i>Story Point</i>	<i>Status</i>	<i>Timeline</i>
Sebagai <i>user</i> dapat melakukan <i>reset password</i>	5	<i>Done</i>	29 Nov 2021 – 3 Des 2021
Sebagai <i>user</i> dapat melihat <i>password</i> yang diisi saat <i>login</i>	1	<i>Done</i>	6 Des 2021

Pembuatan Halaman Utama

Halaman utama atau *homepage* akan menampilkan beberapa fitur yang di antaranya terdapat fitur presensi masuk dan keluar. Namun pada fitur ini tidak hanya mencatatkan presensi kehadiran saja, melainkan terdapat pencatatan waktu datang dan waktu keluar yang akan disimpulkan apakah karyawan tersebut tepat waktu atau tidak dalam mulai bekerja. Selain itu keterangan masuk juga terdapat pada fitur ini sebagai syarat dicatatnya presensi. Dan tentunya terdapat fitur *tag* lokasi di mana lokasi karyawan saat bekerja dapat diketahui dalam fitur ini. Pada halaman utama berikut juga terdapat fitur riwayat presensi yang dilakukan pada hari tersebut beserta *task* yang harus dikerjakan pada hari itu juga. Berikut adalah tampilan dari halaman utama pada Gambar 3.5.

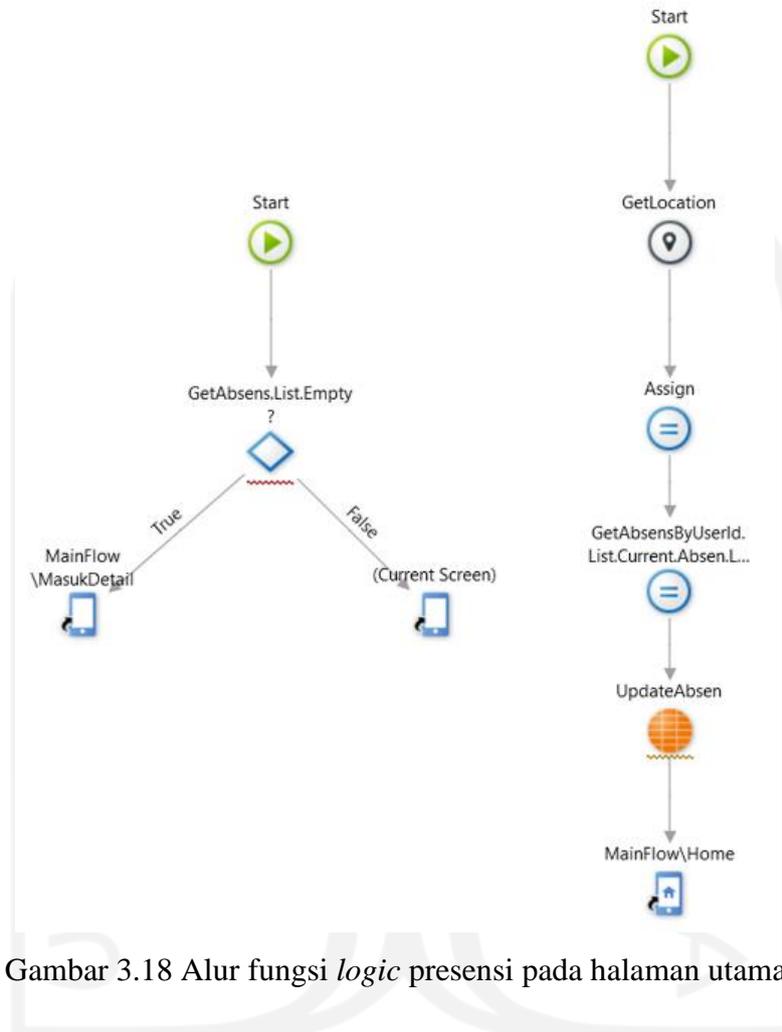


Gambar 3.17 Halaman Utama

Terdapat revisi mengenai *content* di dalam *card* yang terdapat pada tampilan antarmuka halaman utama ada Gambar., yaitu terdapat *text* bertuliskan “Absensi” dan “Absen” yang seharusnya dituliskan dengan “Presensi”.

Fungsi *logic* yang dilakukan pada tombol presensi masuk yaitu pencatatan keterangan presensi, waktu masuk dan keluar, serta lokasi pada saat bekerja. Fungsi berikut dibuat guna agar pencatatan presensi yang dilakukan dapat menjadi data yang dapat disimpan dan diolah dengan baik nantinya. Pada fungsi ini, *foundation modules* yang disediakan oleh Outsystems sebagai *plug-ins* digunakan untuk mendeteksi lokasi pada saat melakukan presensi. Dengan mengambil variabel *Location Latitude* dan *Location Longitude* dengan perintah *Assign*, maka didapatkan titik koordinat lokasi saat melakukan presensi. Kemudian semua data disimpan melalui *consume API post method* dengan menggunakan *core modules* di antaranya yaitu *GetAbsen*, *GetKeterangan*, Setelah itu ditampilkan dengan menggunakan *Google Maps* dengan

mengintegrasikannya melalui *API*. Detail dari fungsi *logic* yang dibuat terdapat pada Gambar 3.6.



Gambar 3.18 Alur fungsi *logic* presensi pada halaman utama

Berikut adalah detail mengenai keterangan pelaksanaan pembuatan halaman utama yang terdapat pada Tabel 3.10. Selain itu terdapat detail capaian yang telah dilakukan setelah pembuatan halaman utama selesai dilakukan yang terdapat pada Tabel 3.11.

Tabel 3.11 Detail keterangan pembuatan fitur halaman utama

Siapa yang terlibat	Keterangan Task	Modules	PIC
Front-end developer (Ghifar Maulana)	<ul style="list-style-type: none"> Membuat antarmuka halaman utama. 	End-User Modules	Yan Pradana

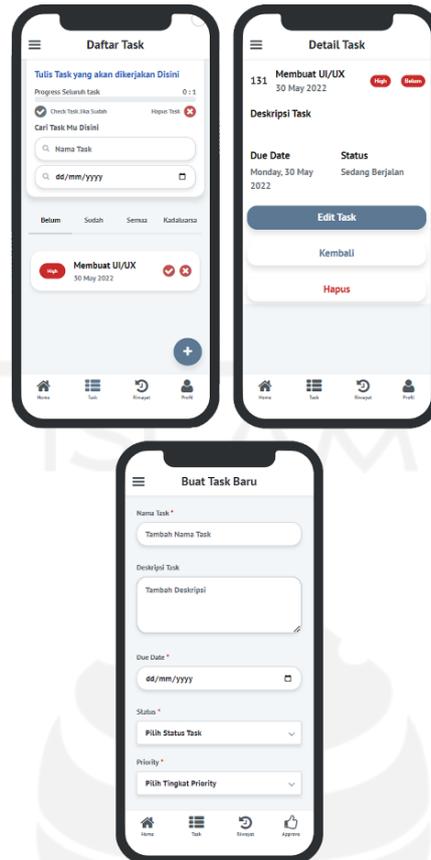
	<ul style="list-style-type: none"> • <i>Consume API Post Method</i> yang telah dibuat untuk menambah data presensi hari ini. • <i>Consume API Get Method</i> yang telah dibuat untuk mengambil data dan menampilkan hasil presensi serta daftar <i>task</i> hari ini. 	<ul style="list-style-type: none"> • Halaman Utama <p><i>Foundation</i></p> <p><i>Modules</i></p> <ul style="list-style-type: none"> • <i>Clock</i> • <i>Outsystems UI</i> • <i>Location</i> <p><i>Plugin</i></p>	
--	---	---	--

Tabel 3.12 Capaian *product backlog* pada *sprint* pertama setelah pembuatan halaman utama

<i>Product Backlog</i>	<i>Story Point</i>	<i>Status</i>	<i>Timeline</i>
Sebagai <i>user</i> dapat melakukan presensi masuk	5	<i>Done</i>	25 Okt 2021 – 29 Okt 2021
Sebagai <i>user</i> dapat melakukan presensi keluar	5	<i>Done</i>	25 Okt 2021 – 29 Okt 2021
Sebagai <i>user</i> dapat melihat durasi waktu bekerja hari ini	1	<i>Done</i>	1 Nov 2021
Sebagai <i>user</i> dapat melihat <i>to do list</i> hari ini	1	<i>Done</i>	15 Nov 2021
Sebagai <i>user</i> dapat melihat rangkuman presensi hari ini	1	<i>Done</i>	2 Nov 2021

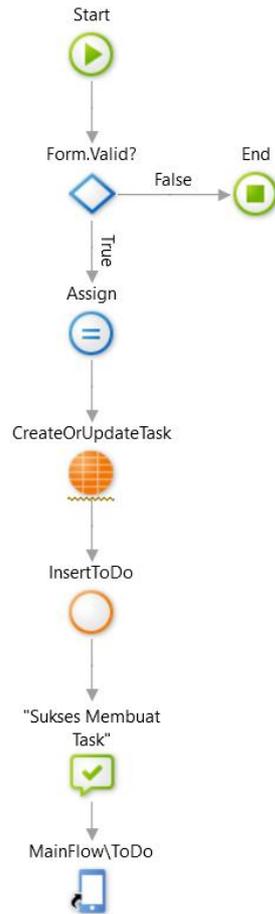
Pembuatan Halaman *To Do List*

Halaman *To Do List* merupakan halaman di mana semua pekerjaan dicatat. Fitur pencatatan tugas yang akan dilakukan selama bekerja ini nantinya akan menjadi laporan kepada atasan serta sebagai *reminder* atau pengingat *deadline* dari tugas yang harus dikerjakan tersebut. Tidak hanya menampilkan detail dari tugas yang akan dikerjakan selama bekerja, melainkan dari halaman ini pengguna juga dapat membuat dan mengubah informasi tugas yang akan dikerjakan tersebut. Halaman *To Do List* yang akan ditampilkan akan terlihat seperti pada Gambar 3.7.



Gambar 3.19 Halaman *To Do List* ACCAbsensi

Logic pada tombol buat *task* pada halaman *To Do List* merupakan fungsi yang digunakan untuk melakukan pencatatan *task* yang akan dikerjakan saat bekerja. *Form* yang berisi nama, hingga durasi dan skala prioritas *task* diisi guna sebagai pencatatan *task* yang akan dikerjakan nantinya. Jika *task* telah selesai dilaksanakan, maka terdapat juga *logic* pada tombol centang sebagai penanda bahwa tugas telah selesai dikerjakan. Berikut adalah detail dari *logic* pada tombol buat *task* dapat dilihat pada Gambar 3.8.



Gambar 3.20 Alur fungsi *Logic button* buat *task* pada halaman *to do list*

Berikut adalah detail mengenai keterangan pelaksanaan pembuatan halaman *to do list* yang terdapat pada Tabel 3.12. Selain itu terdapat detail capaian yang telah dilakukan setelah pembuatan halaman utama selesai dilakukan yang terdapat pada Tabel 3.13.

Tabel 3.13 Detail keterangan pembuatan fitur halaman *to do list*

Siapa yang terlibat	Keterangan Task	Modules	PIC
Front-end developer (Ghifar Maulana)	<ul style="list-style-type: none"> Membuat antarmuka halaman <i>to do list</i>. <i>Consume API Post Method</i> yang telah dibuat untuk menambah data <i>to do</i>. 	<i>End-User Modules</i> <ul style="list-style-type: none"> Halaman <i>To Do List</i> <i>Foundation Modules</i>	Yan Pradana

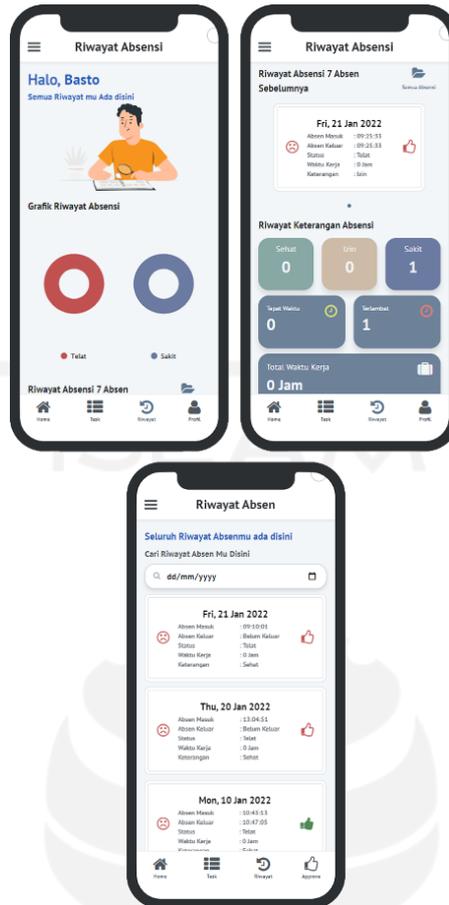
	<ul style="list-style-type: none"> • <i>Consume API Post Method</i> yang telah dibuat untuk mengubah status data <i>to do</i>. • <i>Consume API Get Method</i> yang telah dibuat untuk menampilkan data daftar <i>to do</i> di dalam <i>database</i>. 	<ul style="list-style-type: none"> • <i>Outsystems UI</i> 	
--	---	--	--

Tabel 3.14 Capaian *product backlog* pada *sprint* pertama setelah pembuatan *to do list*

<i>Product Backlog</i>	<i>Story Point</i>	<i>Status</i>	<i>Timeline</i>
Sebagai <i>user</i> dapat melihat <i>to do list</i>	5	<i>Done</i>	4 Nov 2021 – 12 Nov 2021
Sebagai <i>user</i> dapat menambah dan mengedit <i>to do list</i>	5	<i>Done</i>	4 Nov 2021 – 12 Nov 2021

Pembuatan Halaman Riwayat

Halaman riwayat akan berisikan daftar riwayat presensi yang telah dilakukan sebelumnya. Berdasarkan dari kumpulan data presensi yang telah dilakukan, *dashboard* dibuat untuk mengetahui bagaimana kinerja yang dilakukan selama bekerja. Dari *dashboard* tersebut, maka pengguna akan mengetahui informasi mengenai seluruh presensi yang pernah dilakukan. Tidak hanya itu, untuk *role Lead* dan *Head* diberikan fitur tambahan untuk dapat mengawasi kinerja dari anggota mereka. Detail dari tampilan halaman riwayat dapat dilihat pada Gambar 3.9.



Gambar 3.21 Halaman Riwayat

Berikut adalah detail mengenai keterangan pelaksanaan pembuatan halaman riwayat yang terdapat pada Tabel 3.14. Selain itu terdapat detail capaian yang telah dilakukan setelah pembuatan halaman riwayat selesai dilakukan yang terdapat pada Tabel 3.15.

Tabel 3.15 Detail keterangan pembuatan fitur halaman riwayat

Siapa yang terlibat	Keterangan Task	Modules	PIC
Front-end developer (Ghifar Maulana)	<ul style="list-style-type: none"> Membuat antarmuka halaman riwayat. Consume API Get Method yang telah dibuat untuk menampilkan data presensi dan daftar <i>to do</i>. 	<i>End-User Modules</i> <ul style="list-style-type: none"> Halaman Riwayat <i>Foundation Modules</i> <ul style="list-style-type: none"> Outsystems UI 	Yan Pradana

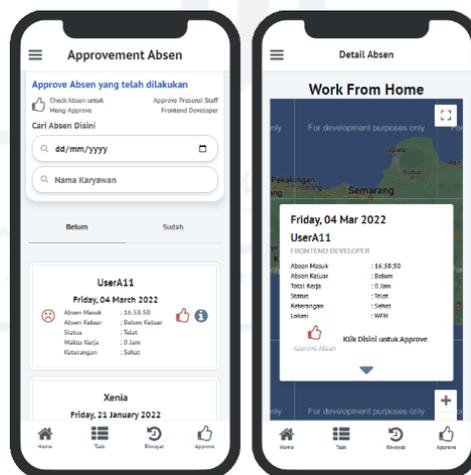
		<ul style="list-style-type: none"> • <i>Outsystems maps</i> • <i>OutSystems Charts</i> 	
--	--	--	--

Tabel 3.16 Capaian *product backlog* pada *sprint* pertama setelah pembuatan *to do list*

<i>Product Backlog</i>	<i>Story Point</i>	<i>Status</i>	<i>Timeline</i>
Sebagai <i>user</i> dapat melihat riwayat presensi dan <i>to do list</i>	3	<i>Done</i>	16 Nov 2021 – 19 Nov 2021

Pembuatan Halaman *Approval*

Halaman *Approval* merupakan halaman yang hanya tersedia bagi *user* yang memiliki *role* sebagai divisi *Lead* dan *Head*. Fitur yang terdapat pada halaman ini adalah sebagai *approval* atau persetujuan untuk riwayat presensi yang telah dilakukan oleh para karyawan. Dari halaman ini juga terdapat detail dari presensi yang telah dilakukan oleh masing-masing karyawan. Jika divisi *Lead* dan *Head* telah melakukan *approval* terhadap presensi yang dilakukan, maka data yang telah di *input* ke sistem dapat dinyatakan valid. Halaman *Approval* akan ditampilkan seperti pada Gambar 3.10.

Gambar 3.22 Halaman *Approval*

Berikut adalah detail mengenai keterangan pelaksanaan pembuatan halaman *approval* yang terdapat pada Tabel 3.16. Selain itu terdapat detail capaian yang telah dilakukan setelah pembuatan halaman *approval* selesai dilakukan yang terdapat pada Tabel 3.17.

Tabel 3.17 Detail keterangan pembuatan fitur halaman *approval*

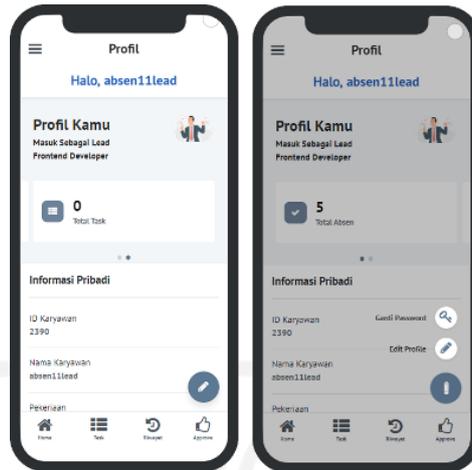
Siapa yang terlibat	Keterangan Task	Modules	PIC
Front-end developer (Ghifar Maulana)	<ul style="list-style-type: none"> Membuat antarmuka halaman <i>approval</i>. Consume API post Method yang telah dibuat untuk memperbarui <i>approval</i> pada data presensi. 	<i>End-User Modules</i> <ul style="list-style-type: none"> Halaman Riwayat <i>Foundation Modules</i> <ul style="list-style-type: none"> Outsystems UI Outsystems maps 	Yan Pradana

Tabel 3.18 Capaian *product backlog* pada *sprint* pertama setelah pembuatan *approval*

Product Backlog	Story Point	Status	Timeline
Sebagai <i>user Lead</i> dapat melakukan <i>approval</i> pada presensi <i>Staff</i>	5	Done	7 Des 2021 – 13 Des 2021
Sebagai <i>user Head</i> dapat melakukan <i>approval</i> pada presensi <i>Lead</i>	5	Done	14 Des 2021 – 20 Des 2021

Pembuatan Halaman Profil

Halaman Profil merupakan halaman yang umum ditemukan pada aplikasi *mobile* lainnya, yaitu sebagai informasi mengenai data diri beserta terdapat fitur pengubahan jika ingin memperbarui informasi data diri ataupun kata sandi yang dimiliki. Halaman profil yang akan ditampilkan seperti pada Gambar 3.11.



Gambar 3.23 Halaman Profil

Berikut adalah detail mengenai keterangan pelaksanaan pembuatan halaman *approval* yang terdapat pada Tabel 3.18. Selain itu terdapat detail capaian yang telah dilakukan setelah pembuatan halaman *approval* selesai dilakukan yang terdapat pada Tabel 3.19.

Tabel 3.19 Detail keterangan pembuatan fitur halaman *approval*

Siapa yang terlibat	Keterangan Task	Modules	PIC
Front-end developer (Ghifar Maulana)	<ul style="list-style-type: none"> Membuat antarmuka halaman profil. <i>Consume API post Method</i> yang telah dibuat untuk mengubah <i>password</i> milik <i>user</i> karyawan. <i>Consume API GetMethod</i> yang telah dibuat untuk menampilkan data <i>user</i> karyawan 	<p><i>End-User Modules</i></p> <ul style="list-style-type: none"> Halaman Data Diri Pengguna <p><i>Foundation Modules</i></p> <ul style="list-style-type: none"> <i>Outsystems UI</i> 	Yan Pradana

Tabel 3.20 Capaian *product backlog* pada *sprint* pertama setelah pembuatan *approval*

<i>Product Backlog</i>	<i>Story Point</i>	<i>Status</i>	<i>Timeline</i>
Sebagai <i>user</i> dapat melakukan <i>reset password</i>	5	<i>Done</i>	21 Des 2021 – 27 Des 2021
Sebagai <i>user</i> dapat menambahkan foto profil	3	<i>Done</i>	28 Des 2021 – 31 Des 2021

BAB IV

REFLEKSI PELAKSANAAN MAGANG

3.3 Relevansi Akademik

Terdapat beberapa relevansi terkait pelaksanaan magang di lapangan yang dilakukan oleh penulis dengan teori yang diuraikan pada bab 2 serta selama pembelajaran kuliah sebelumnya. Di antaranya sebagai berikut.

3.3.1 Relevansi *Scrum*

Pengembangan yang telah dilakukan penulis selama magang di perusahaan Astra Credit Companies yaitu menggunakan metode pengembangan *agile* dengan kerangka kerja *scrum*. *Scrum* merupakan kerangka kerja responsif dari pengembangan perangkat lunak dan mengelola produk atau pengembangan aplikasi. *Scrum* berfokus pada strategi, pengembangan produk holistik yang fleksibel di mana tim pengembang bekerja sebagai unit untuk mencapai tujuan bersama. *Scrum* dapat mengekspos ketidak-efektifan manajemen produk dan teknik kerja yang dilakukan, sehingga dari hal tersebut secara terus menerus akan mampu meningkatkan kinerja produk. Proses pengembangan pada *Scrum* dapat dikelompokkan kedalam 5 fase yang di antaranya yaitu fase *initiate*, fase *plan and estimate*, fase *implement*, fase *review and restropect*, dan fase *release*.

Pada teori dijelaskan bahwa tim pengembang bekerja sebagai unit untuk mencapai tujuan bersama. Sama dengan pelaksanaan di lapangan, karena pada pelaksanaan pengembangan yang dilakukan, satu tim berisi dari beberapa divisi yang memiliki peran masing-masing di dalamnya untuk mengembangkan aplikasi yang tujuannya untuk mencapai *requirements* yang dibutuhkan oleh *users*. Selain itu fase-fase pengembangan menggunakan metode *scrum* yang digunakan perusahaan ACC sama seperti pada penjabaran teori dikarenakan diawali dengan perencanaan, kemudian implementasi *sprint*, hingga proses *release* dilakukan. Dengan metode *scrum* yang diterapkan ini mampu mengekspos ketidakefektifan teknik bekerja pada perusahaan sehingga dapat ditingkatkan terus menerus pada pelaksanaan *sprint* berikutnya.

Terdapat perbedaan yang terletak pada pelaksanaan dalam menggunakan metode pengembangan *scrum* di lapangan. Dikarenakan dalam pengembangan aplikasi ACC Absensi dilakukan secara daring, membuat pelaksanaan metode pengembangan *scrum* menjadi kurang maksimal. Seperti pelaksanaan pada fase *impelement*, *daily meeting* tidak dapat dilaksanakan setiap harinya selama masa pengembangan. Membuat proses evaluasi menjadi terhambat dan

kurangnya transparansi yang menjadikan pengembangan sedikit terhambat dan susah dalam pencarian solusi dari masalah yang muncul.

Selain itu terdapat perbedaan lainnya yang terletak pada pelaksanaan metode pengembangan *scrum* pada pengembangan aplikasi ACCAbsensi berikut. Pada pelaksanaan fase *review and restropect*, belum terdapat *user* yang dapat menguji langsung aplikasi tersebut. Oleh sebab itu, UAT dilaksanakan oleh tim QC dan hanya berpatokan pada *user acceptance criteria* yang telah ditentukan sebelumnya. Tentunya membuat hasil pengujian menjadi kurang maksimal dengan tanpa adanya temuan atau perbaikan yang diberikan dan disampaikan kepada penulis.

3.3.2 Relevansi *Low-Code Platform*

Penggunaan platform yang digunakan oleh penulis dalam mengembangkan sistem selama magang berlangsung dapat dikatakan sebagai platform *low-code* yang tujuan utama dari platform tersebut yaitu untuk memudahkan para pengembang. Platform *low-code* atau kode rendah untuk *programmer* ini memungkinkan pembuatan aplikasi bisnis dengan cepat dikarenakan minim penulisan pengkodean serta minim dalam instalasi, konfigurasi dan pengimplementasian. Penggunaan fitur *drag and drop* membuat proses pengembangan menjadi mudah dan cepat. Selain itu pengembang dapat menggunakan *low-code* platform untuk membangun aplikasi dengan tampilan visual aplikasi secara langsung dan dapat diintegrasikan dengan *database*. Sejauh ini, yang tercepat dan mungkin murah dalam mengembangkan perangkat lunak. Penjabaran teori tersebut memiliki relevansi dengan pelaksanaan magang yang telah dilakukan oleh penulis, terutama pada bagian *tools* yang digunakan oleh penulis.

Persamaan pelaksanaan di lapangan dengan teori yang telah dijabarkan ialah platform yang digunakan oleh penulis yaitu Outsystems memiliki tujuan utama untuk mempermudah *programmer* maupun *non-programmer* dalam mengembangkan aplikasi. Outsystems merupakan platform yang memungkinkan pengembangan dilakukan secara visual, mudah diintegrasikan dengan sistem lain yang ada dan penambahan kode sesuai kebutuhan tanpa menuliskan kode secara manual dan banyak. Fitur *drag and drop* yang disuguhkan oleh Outsystems juga mempermudah developer dalam mengembangkan aplikasi dengan cepat.

Namun terdapat perbedaan relevansi yang cukup mempengaruhi dalam penggunaan platform *low-code* ini. Yaitu teori mengatakan bahwa sejauh ini metode *low-code* platform adalah metode termurah dalam mengembangkan aplikasi. Tetapi pada saat pelaksanaan di

lapangan mengungkapkan bahwa platform yang digunakan dalam mengembangkan aplikasi pada perusahaan ACC dapat dikatakan memakan biaya yang sangat banyak. Oleh karena itu, penggunaan *low-code* platform hanya diperuntukkan bagi perusahaan atau perseorangan yang memiliki dana yang cukup saja untuk mendapatkan pelayanan dari platform tersebut.

3.4 Pembelajaran Magang

Berdasarkan pelaksanaan magang yang telah dilaksanakan oleh penulis, tentunya banyak sekali pelajaran dan ilmu yang didapatkan. Dengan pembelajaran yang didapatkan selama magang tersebut, penulis menjadi mengetahui apa saja manfaat yang didapatkan dari masing-masing hal yang dipelajari hingga kendala, hambatan dan tantangan terkait teori atau pelaksanaan di lapangan yang ditemui.

3.4.1 Platform *Low-Code* Outsystems

Selama pelaksanaan magang yang dilakukan oleh penulis tentunya banyak sekali ilmu yang didapatkan dari sisi pengalaman hingga manajemen. Ilmu yang tentunya sangat berharga didapatkan oleh penulis dikarenakan diposisikan pada bidang atau divisi yang ingin ditekuni lebih dalam yaitu sebagai *front-end* developer. Sebagai seorang *front-end* developer tentunya bertugas sebagai pembuat aplikasi dari sisi *front-end* atau tampilan antarmuka yang menghubungkan antara pengguna dengan sistem. Dalam proses pengembangan yang dilakukan pada perusahaan Astra Credit Companies, penulis menemukan dan mendapatkan ilmu baru mengenai platform yang digunakan untuk pengembangan aplikasi, tidak melakukan *hard-coding* seperti pada umumnya yang telah diajarkan selama perkuliahan, melainkan dengan menggunakan platform *low-code* yang bernama Outsystems.

Outsystems adalah platform *low-code* yang digunakan untuk *front-end* developer dalam mengembangkan aplikasi berbasis web maupun *mobile* pada perusahaan Astra Credit Companies. Menurut penulis selama menjalani magang sebagai *front-end* developer dan menggunakan platform tersebut, pengembangan menjadi terasa jauh lebih mudah dibandingkan melakukan pengembangan aplikasi dengan cara *hard-code* pada umumnya. Beberapa keuntungan dan manfaat yang didapatkan dalam penggunaan platform Outsystems yang antara lain dapat memangkas durasi pengembangan aplikasi dikarenakan platform yang digunakan menggunakan sistem *drag and drop*, membuat pembuatan design pada antarmuka menjadi lebih cepat tanpa menuliskan kode kasar pada umumnya. Selain itu dapat dikatakan pengembangan dengan platform Outsystems jauh lebih efektif dan mudah dibandingkan

dengan pengembangan dengan menggunakan *hard-code* pada umumnya, karena banyak fitur-fitur tambahan yang disediakan oleh platform tersebut untuk mendukung pengembangan.

Dikarenakan platform yang digunakan tersebut dapat dikatakan awam bagi penulis, membuat hal tersebut menjadi sebuah tantangan bagi penulis untuk mempelajari dari awal penggunaan setiap sub fiturnya dalam mengembangkan aplikasi. Dijelaskan jika Outsystems adalah platform yang memudahkan penggunaannya dalam mengembangkan aplikasi namun masih terdapat beberapa kendala yang menurut penulis berpengaruh terhadap pengembangan ialah Outsystems hanya menyediakan tema dasar dari platform itu sendiri dalam mengembangkan aplikasi. Jadi pengguna tidak dapat mengubah atau memodifikasi tema yang terdapat pada aplikasi yang dikembangkan dikarenakan menggunakan platform tersebut. Selain itu terdapat kelemahan utama yang dimiliki oleh platform tersebut yaitu dapat menghabiskan dana yang cukup banyak jika ingin mendapatkan layanan penuh dari platform tersebut.

3.4.2 Metode Pengembangan *Scrum*

Pembelajaran magang selanjutnya yaitu ilmu mengenai metode pengembangan yang diterapkan dalam perusahaan Astra Credit Companies yang didapatkan dan kemudian dipelajari dengan menerapkan secara langsung pada pengembangan proyek yang dilakukan oleh penulis. Metode pengembangan *agile* dengan kerangka kerja *scrum* adalah metode yang belum pernah dipelajari oleh penulis selama mengembangkan aplikasi pada saat perkuliahan berlangsung. Lebih mengarah pada metode *waterfall* selama perkuliahan membuat bertambahnya pengalaman penulis mengenai ilmu yang didapatkan pada metode pengembangan *scrum*.

Manfaat yang didapatkan selama menerapkan metode pengembangan *scrum* ialah pengerjaan menjadi lebih terstruktur hingga tiap *task-task* nya dikarenakan berdasar pada *product backlog* yang didapatkan langsung dari *requirements* yang pengguna butuhkan. Selain itu, manfaat yang didapatkan dari segi tim yaitu adanya *daily standup meeting* di mana tim secara bersama melakukan evaluasi terkait apa yang telah dilakukan dalam jangka waktu *sprint* berlangsung. Tentunya manfaat utama yang diperoleh dalam metode ini adalah kemampuan untuk membentuk perubahan dalam pelaksanaan *sprint* berikutnya dikarenakan mengevaluasi dari pelaksanaan *sprint* sebelumnya.

Namun terdapat tantangan pada pelaksanaan metode pengembangan berikut bagi penulis, yang salah satunya adalah dikarenakan perusahaan memiliki patokan durasi waktu pengembangan hanya selama 2 minggu pada satu *sprint* membuat penulis menjadi mau tidak

mau mengorbankan waktunya secara penuh untuk mengerjakan pengembangan selama waktu pengembangan berlangsung. Mengenai kendala dan hambatan yang didapatkan selama menerapkan metode pengembangan *scrum* yaitu setiap tugas harus didefinisikan dengan baik, karena hal ini dapat mempengaruhi perkiraan dan waktu pengerjaan proyek. Selain itu, dikarenakan penulis belum berpengalaman seperti developer lainnya, membuat proyek yang dikerjakan selesai sangat mendekati dengan tenggat waktu yang diberikan.

3.4.3 Platform *Monday* untuk Manajemen Proyek

Selain pengalaman keterampilan teknis atau *hard skill* yang didapatkan selama magang. Penulis juga mendapatkan ilmu mengenai pemenejemenan proyek secara kompleks. Perusahaan Astra Credit Companies menggunakan platform *Monday* untuk mengatur setiap hal yang berhubungan dengan pelaksanaan pengembangan proyek. Platform yang digunakan tersebut tidaklah awam bagi penulis dikarenakan penulis telah mendapatkan ilmu mengenai manajemen proyek namun hanya dengan platform yang berbeda.

Banyak fitur yang tentunya sangat membantu dalam pengelolaan proyek, tugas, proses, dan alur kerja dengan menggunakan platform *Monday*. Metode pengembangan aplikasi yang digunakan perusahaan yaitu *scrum* juga menjadi sangat terbantu, hal ini dikarenakan banyak dokumen-dokumen yang diperlukan menjadi mudah untuk dibuat dan digunakan. Misalnya seperti pembuatan *sprint backlog* yang memudahkan developer untuk memperbarui laporan kinerja yang telah dilakukan, serta memudahkan *scrum master* untuk melakukan pemantauan terhadap kinerja dari developer. Selain hal itu, skala prioritas dalam sebuah proyek juga dapat ditentukan yang membantu dalam mengatur *workflow* dari proyek yang sedang dikerjakan. Dikarenakan sudah pernah dipelajari sebelumnya namun hanya berbeda platform saja, menjadikan penulis hanya perlu beradaptasi sedikit dengan platform *Monday* karena lebih kompleks mengenai fitur-fitur di dalamnya.

BAB V PENUTUP

4.1 Kesimpulan

Dalam penulisan tugas akhir ini bermaksud untuk memberikan wawasan mengenai pengembangan aplikasi yang telah dilaksanakan oleh penulis selama menjalankan magang pada perusahaan Astra Credit Companies. Selain itu juga untuk memberikan pengetahuan lebih dalam mengenai platform *low-code* Outsystems yang digunakan dalam pengembangan aplikasi. Dengan pengetahuan yang didapatkan dari penulisan tugas akhir ini diharapkan dapat menjadi dasar pengetahuan dalam mengembangkan aplikasi dengan menggunakan metode pengembangan *scrum* serta penggunaan platform *low-code* Outsystems.

Berdasarkan hasil penyusunan tugas akhir, maka dapat diambil kesimpulan:

- a. Dibuatnya aplikasi ACCAbsensi dapat mempermudah atasan untuk mendapatkan data dari hasil pencatatan yang telah di input tanpa melakukan pengecekan secara manual satu-persatu dari karyawannya. Selain itu, aplikasi dapat digunakan sesegera mungkin dikarenakan hanya memerlukan waktu singkat dalam pengembangan dikarenakan menggunakan platform Outsystems.
- b. Penerapan metode *scrum* memiliki fase-fase pengembangan yang sama dengan diawali dari perencanaan, implementasi, hingga pelepasan proyek kepada pengguna.
- c. Metode *scrum* hanya membutuhkan anggota tim yang sudah berpengalaman dikarenakan pelaksanaan *sprint* harus diselesaikan sesuai pada waktunya.
- d. Dalam pengembangan aplikasi ACCAbsensi yang dilakukan secara daring bersama tim. Terdapat kekurangan pada saat pelaksanaan fase *implement*. Yaitu mengakibatkan *daily meeting* tidak berjalan secara maksimal dan membuat proses evaluasi serta transparansi menjadi terhambat.
- e. Pengembangan yang dilakukan menggunakan metode pengembangan *agile* dengan kerangka kerja *scrum* adalah metode yang tepat untuk diterapkan di perusahaan dikarenakan dapat mudah mengatasi setiap perubahan yang terjadi. Namun masih terdapat kekurangan dalam pelaksanaan *scrum* pada pengembangan aplikasi ACCAbsensi dikarenakan belum terdapat *user* yang dapat menguji langsung aplikasi tersebut. Oleh sebab itu membuat *feedback* langsung dari *user* belum dapat diberikan atau disampaikan.

- f. Keuntungan yang didapatkan dalam penggunaan platform *low-code* Outsystems antara lain yaitu dapat memangkas durasi pengembangan aplikasi dikarenakan platform yang digunakan menggunakan sistem *drag and drop* tanpa menggunakan banyak *hard-code*, mampu mengatasi segala *exception* yang terjadi secara otomatis berkat fitur yang dimiliki Outsystems, dan banyak fitur-fitur tambahan yang disediakan oleh platform tersebut untuk mendukung pengembangan.

4.2 Saran

Berdasarkan dari hasil laporan akhir yang telah diuraikan di atas, Adapun saran yang diberikan dari penulis yang mungkin dapat bermanfaat bagi pihak perusahaan yaitu pada penamaan aplikasi ACCAbsensi yang memiliki tujuan sebagai aplikasi pencatatan presensi. Kurang tepatnya pemilihan nama aplikasi tersebut dikarenakan kata “absensi” sendiri sangat berbeda maknanya dari kata presensi. Selain itu pada proses pengembangan aplikasi ACCAbsensi, terdapat kekurangan di dalam proses pelaksanaan fase *review and restropect* dikarenakan belum terdapat pengujian sistem secara langsung dilakukan oleh *user* yang mengakibatkan kurang maksimalnya pengujian *User acceptance Testing* yang dilakukan. Dalam hal ini diharapkan menjadi perhatian dikarenakan pada saat pelaksanaan *User Acceptance Testing*, sangat dibutuhkannya *feedback* dari peran *user* aplikasi yang dikembangkan tersebut. Selain itu dapat memperlengkap pengkajian laporan akhir yang disusun oleh penulis jika proses pengembangan aplikasi dilaksanakan secara lancar.

DAFTAR PUSTAKA

- Abednego, A. (2021). PENGEMBANGAN APLIKASI LAYANAN MULTIGUNA MENGGUNAKAN LOW-CODE PLATFORM. (*Doctoral dissertation, UNIVERSITAS ATMA JAYA YOGYAKARTA*).
- ACC. (2021). ACC Lakukan Peletakan Batu Pertama Pembangunan Digital Operation Center. Retrieved from <https://web.acc.co.id/corporatenews/read/acc-lakukan-peletakan-batu-pertama-pembangunan-digital-operation-center>
- Baker, D. (2016). *Web Content Management*. O'Reilly Media, Inc.
- Black, R. (2009). *Managing the testing process : practical tools and techniques for managing hardware and software testing, 3rd edition*. Indiana : Wiley Publishing, Inc.
- Ibnu. (2021, October 4). *Content Management System: Pengertian, Jenis, Dan Contoh-contohnya*. Retrieved from Accurate: <https://accurate.id/teknologi/content-management-system/>
- Menezes, F. (2020). *Application Architecture: Best Practices for Future-Proofing Your Apps*. Retrieved from www.outsystems.com/blog/posts/application-architecture/
- Metrólho, J., Ribeiro, F., & Araújo, R. (2020). A strategy for facing new employability trends using a low-code development platform. In *14th International Technology, Education and Development Conference*, (pp. 8601-8606).
- Mew, L., & Field, D. (2018). A case study on using the mendix low code platform to support a project management course. In *Proceedings of the EDSIG Conference ISSN* (Vol. 2473, p. 3857).
- Outsystems. (2020). *Platform Review : Outsystems*. Retrieved from <https://medium.com/ecomindo-dev/outsystems-low-code-platform-untuk-mengembangkan-aplikasi-secara-cepat-tanpa-melakukan-coding-f8d932b8d91c#:~:text=Outsystems%20merupakan%20sebuah%20low%2Dcode%20platform%20untuk%20mengembangkan%20aplikasi%20mobile,aplikasi>
- Outsystems. (2022, June 29). *Setting Up OutSystems*. Retrieved from https://success.outsystems.com/Documentation/11/Setup_and_maintain_your_OutSystems_infrastructure/Setting_Up_OutSystems
- Perry, W. E. (2007). *Effective Methods for Software Testing Third Edition*. Willey.
- Rubin, K. (2013). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional.

- Sukindar. (2018, April 25). *OutSystems Berhasil Menjadi 'Pemimpin Pasar'*. Retrieved from DIGINATION.id: <https://www.digation.id/read/011230/outsystems-berhasil-menjadi-pemimpin-pasar>
- Threestayanti, L. (2019, June 28). *Apa Itu Platform Low-Code Development?* Retrieved from INFOKOMPUTER: <https://infokomputer.grid.id/read/121768787/apa-itu-platform-low-code-development>
- Waszkowski, R. (2019). *Low-code platform for automating business processes in manufacturing*. IFAC-PapersOnLine, Vol 52, Issue 10, 376-381.
- Watkins, J., & Simons, M. (2010). *Testing IT An Off-the-Shelf Software Testing Process*. Cambridge University Press.
- Widiyanto, D. (2022, March 23). *Resmi Beroperasi, Digital Operation Center ACC Dinamai 'berijalan'*. Retrieved from [krjogja.com: https://www.krjogja.com/ekonomi/bisnis/resmi-beroperasi-digital-operation-center-acc-dinamai-berijalan/](https://www.krjogja.com/ekonomi/bisnis/resmi-beroperasi-digital-operation-center-acc-dinamai-berijalan/)

LAMPIRAN

