

**IMPLEMENTASI DAN ADAPTASI SCRUM UNTUK
PENGEMBANGAN *DASHBOARD* VISUALISASI DATA PROYEK PLN
UP3 YOGYAKARTA**



Disusun Oleh:

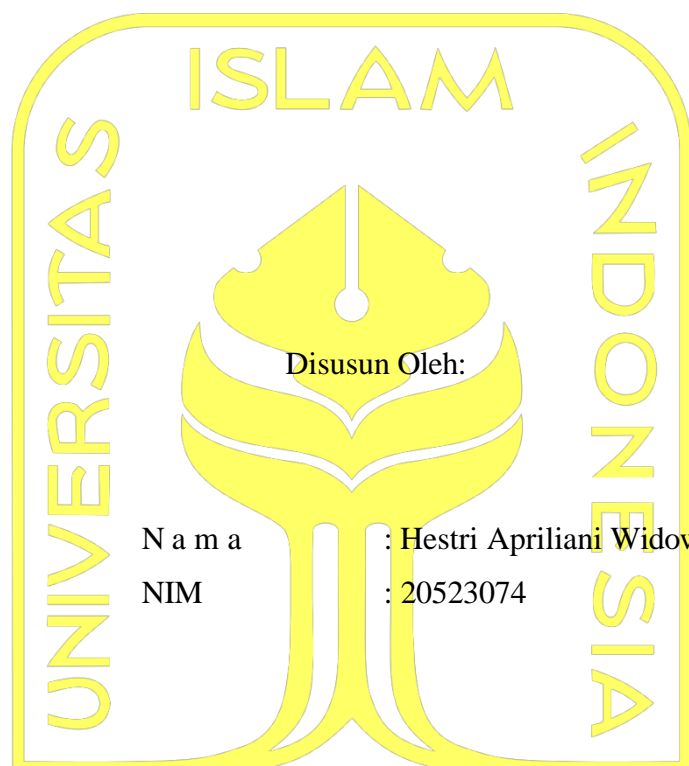
Nama : Hestri Apriliani Widowati
NIM : 20523074

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2024**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**IMPLEMENTASI DAN ADAPTASI SCRUM UNTUK
PENGEMBANGAN *DASHBOARD* VISUALISASI DATA
PROYEK PLN UP3 YOGYAKARTA**

TUGAS AKHIR



الجمهورية الإسلامية
الاستاذة الأندو

Yogyakarta, 22 Oktober 2024

Pembimbing,

(Andhik Budi Cahyono, S.T., M.T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**IMPLEMENTASI DAN ADAPTASI SCRUM UNTUK
PENGEMBANGAN DASHBOARD VISUALISASI DATA
PROYEK PLN UP3 YOGYAKARTA**

TUGAS AKHIR

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, 22 Oktober 2024

Tim Penguji

Andhik Budi Cahyono, S.T., M.T.

Anggota 1

Mukhammad Andri Setiawan, S.T., M.Sc.,
Ph.D.

Anggota 2

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

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



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

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Hestri Apriliani Widowati

NIM : 20523074

Tugas akhir dengan judul:

**IMPLEMENTASI DAN ADAPTASI *SCRUM* UNTUK
PENGEMBANGAN *DASHBOARD* VISUALISASI DATA
PROYEK PLN UP3 YOGYAKARTA**

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, 22 Oktober 2024



(Hestri Apriliani Widowati)

HALAMAN PERSEMBAHAN

Laporan Tugas Akhir ini saya persembahkan untuk Bapak, Ibu, dan Keluarga saya yang selalu mengingatkan jika saya masih memiliki kewajiban untuk menyelesaikan perkuliahan saya. Untuk teman saya yang selalu mendukung dan menemani saya selama mengerjakan laporan Tugas Akhir ini. Laporan ini saya persembahkan untuk semua orang yang akan membaca laporan ini nantinya dan menjadikannya sebagai bahan panduan ataupun referensi.

HALAMAN MOTO

مَنْ جَدَّ وَجَدَّ

Artinya: “Barang siapa yang bersungguh-sungguh, ia akan mencapai tujuannya”

إِنَّمَا أَمْرُهُ إِذَا أَرَادَ شَيْئًا أَنْ يَقُولَ لَهُ كُنْ فَيَكُونُ

Artinya: “Sesungguhnya urusan-Nya apabila Dia menghendaki sesuatu hanyalah berkata kepadanya: ‘Jadilah', maka terjadilah” QS. Yasin ayat 82

“*Just keep swimming.*” Dory, Finding Nemo

“*Life is just like that. Sometimes you're ahead, sometimes you're behind. The important thing is to be patient and keep moving forward.*” Park Saeroyi, Itaewon Class

“*Don't give up just because you're faced with a challenge. Use it as an opportunity to prove how strong you are.*” Han Ji-pyeong, Start-up

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji dan syukur penulis ucapkan kepada Allah SWT yang telah melimpahkan rahmat dan petunjuk-Nya, karena atas pertolongan-Nya penulis dapat menyelesaikan laporan ini tepat waktu. Tidak lupa sholawat dan salam penulis ucapkan kepada Nabi Muhammad SAW, yang telah membimbing umat manusia dari masa kegelapan menuju masa yang penuh cahaya dengan ajaran Islam yang mulia seperti saat ini.

Laporan Tugas Akhir ini dibuat sebagai syarat untuk menyelesaikan Penjaluran Magang Program Sarjana Studi Informatika Fakultas Teknologi Industri. Program magang ini disusun sebagai salah satu jalur kelulusan yang dapat dipilih oleh mahasiswa meraih gelar Sarjana.

Laporan ini disusun dengan tujuan untuk membagikan pengalaman serta pembelajaran yang penulis peroleh selama menjalani magang di PLN UP3 Yogyakarta sebagai *frontend developer*. Harapannya, laporan ini dapat menjadi sumber inspirasi dan pengetahuan bagi pembaca terkait metode pengembangan *Scrum* yang dapat diimplementasikan dalam pengembangan sebuah *dashboard* visualisasi data.

Banyak halangan dan rintangan yang penulis hadapi selama penyusunan Tugas Akhir ini, namun pada akhirnya semua itu dapat terlewati dan Tugas Akhir ini berhasil diselesaikan. Selama penyusunan Tugas Akhir tidak terlepas dari dukungan moral maupun spiritual dari beberapa pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua penulis yang senantiasa memberikan doa dan dukungannya kepada penulis.
2. Keluarga penulis yang selalu memberikan *support system* selama penyusunan laporan.
3. Bapak Andhik Budi Cahyono, S.T., M.T. selaku Dosen Pembimbing Tugas Akhir Magang.
4. Bapak Ryan Permana, selaku Asisten Manajer dari PLN UP3 Yogyakarta.
5. Bapak Ikhwan Cahyono, selaku *Project Leader* selama magang berlangsung.
6. Segenap karyawan PLN UP3 Yogyakarta terutama karyawan divisi Perencanaan yang telah memberikan bimbingan dan arahan selama proses magang berlangsung.
7. Teman-teman seperjuangan magang, tim pemegang PLN UP3 Yogyakarta, dan beberapa pihak yang tidak bisa saya sebutkan satu persatu yang telah memberikan

saya wawasan dan membantu saya baik selama magang maupun sebelum pelaksanaannya.

Dari laporan ini, penulis menyadari bahwa laporan ini masih jauh dari kata sempurna, karena kesempurnaan hanya milik Allah SWT. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari pembaca demi melengkapi kesempurnaan Laporan Tugas Akhir ini. Semoga laporan ini dapat memberikan kontribusi positif dalam perkembangan bangsa dan negara ini.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 22 Oktober 2024



(Hestri Apriliani Widowati)

SARI

Beberapa aplikasi di PLN dikembangkan dengan metode pengembangan *Waterfall*, bahkan tanpa metode pengembangan sehingga tidak jelas jadwal dan lini masa pengembangan dan dapat menyebabkan keterlambatan proyek. Untuk mencegah keterlambatan proyek pada pengembangan aplikasi yang baru, tim IT di PLN memilih metode *Scrum*, termasuk pada proyek aplikasi *dashboard* visualisasi data. Metode ini dipilih karena fleksibilitas, kecepatan, dan kemudahan untuk dapat diimplementasikan oleh seluruh anggota tim. Proses yang diterapkan dalam *Scrum* meliputi penulisan *Product Goal*, membuat *Product Goal*, menyusun *Sprint Planning*, melakukan *Sprint*, menghasilkan sebuah *Increment*, melakukan *Sprint Review*, melakukan *Sprint Retrospective*, dan *Daily Scrum* setiap hari. Proyek ini melibatkan Asisten Manajer Perencanaan dan Asisten Manajer Transaksi Energi sebagai *Product Owner*, *Project Leader* sebagai *Scrum Master*, dan *Development Team*. Dengan metode ini, proyek dapat berjalan sesuai dengan waktu yang telah ditetapkan dan menghasilkan semua fitur yang dibutuhkan.

Kata kunci: *Scrum*, *Dashboard*, Visualisasi Data

GLOSARIUM

<i>Dashboard</i>	Sebuah tampilan antarmuka yang menyajikan informasi secara visual berupa data tabel, gambar, diagram grafik, maupun letak geografis dengan mudah untuk dapat dipahami pengguna dengan mudah.
<i>Scrum</i>	Metode pengembangan perangkat lunak yang fokus utamanya adalah pada fleksibilitas dan dapat dikembangkan dalam waktu yang relatif cepat.
<i>Waterfall</i>	Model pengembangan pada perangkat lunak secara berurutan dari analisis kebutuhan, desain, pengkodean, pengujian, hingga perawatan aplikasi.
Visualisasi Data	Sebuah proses untuk menyajikan data mentah ke dalam bentuk diagram, sehingga informasi yang dihasilkan dapat disampaikan dengan lebih mudah untuk dipahami.

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.....	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Ruang Lingkup Magang.....	2
1.3 Tujuan.....	3
1.4 Manfaat.....	4
1.5 Sistematika Penulisan.....	4
BAB II DASAR TEORI	5
2.1 Visualisasi Data.....	5
2.2 <i>Dashboard</i>	5
2.3 <i>Scrum</i>	6
2.4 Perbandingan metode	7
2.5 Tinjauan Pustaka	8
BAB III PELAKSANAAN MAGANG.....	10
3.1 Aktivitas Magang	10
3.1.1 Tempat dan Durasi Magang	10
3.1.2 Jam Kerja dan Tugas Harian.....	11
3.1.3 Lingkungan Kerja dan Kolaborasi Tim	11
3.1.4 Manajemen Proyek.....	12
3.2 Proyek Sistem Informasi Bersama (SIBER)	13
3.2.1 Deskripsi Proyek.....	14
3.2.2 Tujuan Proyek	14
3.2.3 Analisis Kebutuhan Data	14
3.2.4 <i>Tools</i> Proyek	16
3.2.5 Pemilihan Metode yang Digunakan.....	17
3.3 Implementasi <i>Scrum</i> (Pelaksanaan Proyek Siber)	19
3.3.1 Tahapan <i>Scrum</i> dalam Pengembangan Proyek.....	19
3.4 Pelaksanaan dan Hasil Proyek.....	25
3.4.1 Proyek <i>Dashboard</i> AMR	27
3.4.2 Hasil Proyek <i>Dashboard</i> AMR.....	31
3.4.3 Proyek <i>Dashboard</i> Kinerja	33
3.4.4 Hasil Proyek <i>Dashboard</i> Kinerja.....	36
3.5 Pengujian	37
3.5.1 Metode Pengujian	37
3.5.2 Hasil Pengujian.....	37
BAB IV REFLEKSI PELAKSANAAN MAGANG	40

	xii
4.1	Teknis 40
4.1.1	Bentuk Implementasi <i>Scrum</i> 40
4.1.2	Dampak Implementasi <i>Scrum</i> 44
4.2	Nonteknis..... 45
4.2.1	Pengembangan <i>Soft skill</i> 45
4.2.2	Pengalaman Lingkungan Kerja..... 45
4.3	Refleksi dan Pembelajaran 46
4.3.1	Pelajaran Terbesar dalam Proses Pengembangan 46
4.3.2	Hal-Hal yang Dapat Diterapkan di Masa Depan..... 46
4.3.3	Saran untuk Proyek Serupa di Masa Depan..... 47
BAB V KESIMPULAN DAN SARAN..... 48	
5.1	Kesimpulan..... 48
5.2	Saran..... 49
	DAFTAR PUSTAKA..... 50
	LAMPIRAN 53

DAFTAR TABEL

Tabel 2. 1 Perbandingan Metode Pengembangan <i>Waterfall</i> dan <i>Scrum</i>	8
Tabel 3. 1 <i>Product Backlog Dashboard</i> AMR	20
Tabel 3. 2 <i>Product Backlog Dashboard</i> Kinerja	21
Tabel 3. 3 <i>Sprint Planning I Dashboard</i> AMR	22
Tabel 3. 4 <i>Sprint Planning II Dashboard</i> Kinerja	22
Tabel 3. 5 <i>Sprint Dashboard</i> AMR	23
Tabel 3. 6 <i>Sprint Dashboard</i> Kinerja	23
Tabel 3. 7 <i>Increment Dashboard</i> AMR	24
Tabel 3. 8 <i>Increment Dashboard</i> Kinerja	24
Tabel 3. 9 Pengujian <i>Dashboard</i> AMR	38
Tabel 3. 10 Pengujian <i>Dashboard</i> Kinerja	39
Tabel 4. 1 Implementasi <i>Scrum</i> Berdasarkan Peran	41
Tabel 4. 2 Implementasi <i>Scrum</i> Tiap Aktivitasnya	42

DAFTAR GAMBAR

Gambar 2. 1 Model Pengembangan <i>Scrum</i>	6
Gambar 3. 1 Contoh Aplikasi Sebelum SIBER	26
Gambar 3. 2 Tampilan Prototipe <i>Dashboard</i> Menggunakan Figma	27
Gambar 3. 3 Ikon Tambahan.....	29
Gambar 3. 4 Hasil Proyek <i>Dashboard</i> AMR 1	32
Gambar 3. 5 Hasil Proyek <i>Dashboard</i> AMR 2.....	32
Gambar 3. 6 Hasil Proyek <i>Dashboard</i> Kinerja	36
Lampiran 1 <i>Logbook</i> Magang 1	53
Lampiran 2 <i>Logbook</i> Magang 2.....	53
Lampiran 3 Dokumentasi Sewaktu Istirahat	54
Lampiran 4 Dokumentasi Perayaan HUT PLN.....	54
Lampiran 5 Dokumentasi Sewaktu Pengerjaan Proyek	55
Lampiran 6 Kunjungan Jurusan Informatika.....	55
Lampiran 7 Dokumentasi Kode <i>Dashboard</i> Kinerja.....	56
Lampiran 8 Dokumentasi Kode <i>Dashboard</i> AMR	56

BAB I

PENDAHULUAN

1.1 Latar belakang

Digitalisasi teknologi telah dilakukan oleh banyak organisasi untuk meningkatkan mutu layanan (Suharno, N., & M., 2020). Perusahaan yang mampu beradaptasi dengan perubahan ini dapat meningkatkan efisiensi, mengurangi biaya operasional, dan meningkatkan kualitas produk serta layanan. Salah satu perusahaan yang melakukan peralihan teknologi adalah PLN, sebuah perusahaan milik negara yang bergerak di bidang listrik. PLN (Persero) bertanggung jawab menyediakan listrik di seluruh Indonesia, tidak hanya dalam produksi, distribusi, dan penyaluran listrik kepada konsumen, tetapi juga dalam pengembangan infrastruktur listrik serta investasi pada teknologi terbaru.

PLN menyesuaikan diri dengan perkembangan teknologi informasi untuk meningkatkan efisiensi dan kualitas layanan bagi pegawai. Penyesuaian teknologi informasi ini dilakukan guna mempermudah pekerjaan, termasuk memfasilitasi analisis data secara lebih cepat dan akurat (Wahyudi, et al., 2023). Salah satu bentuk penyesuaian yang dilakukan adalah mengembangkan aplikasi mobile bernama “PLN Mobile” yang berfungsi sebagai sarana interaksi antara petugas PLN dan konsumen, sekaligus menjadi platform untuk menampung keluhan konsumen.

Sebagai perusahaan yang fokus pada pengembangan infrastruktur listrik dan investasi teknologi terbaru, PLN tidak hanya menyediakan listrik, tetapi juga melakukan pemantauan daya dan tegangan yang digunakan oleh pelanggan, serta monitoring dan perawatan infrastruktur kelistrikan. PLN mengembangkan aplikasi web untuk memantau daya dan tegangan listrik pelanggan secara berkala. Namun, aplikasi tersebut memiliki keterbatasan karena tidak dapat beroperasi secara *real-time*, sehingga menghambat keefektifannya. Selain itu, pemantauan dan perawatan infrastruktur kelistrikan masih berjalan secara manual menggunakan data *excel* yang tersusun secara tidak rapi.

Untuk mengatasi masalah tersebut, tim magang di PLN mengembangkan aplikasi web yang diberi nama Sistem Informasi Bersama (SIBER). Aplikasi ini dirancang memantau daya dan tegangan listrik secara *real-time* melalui integrasi dengan basis data, serta memfasilitasi koordinasi dan pemantauan infrastruktur kelistrikan yang dikelola oleh PLN. Aplikasi ini melayani dua divisi utama, yaitu divisi transaksi energi dan divisi perencanaan.

Selama proses pengembangan, aplikasi dibangun menggunakan metode pengembangan *Scrum*. *Scrum* merupakan kerangka kerja yang efektif dalam mengelola produk kompleks (Schwaber & Sutherland, 2024). Metode ini adaptif terhadap perubahan kebutuhan dan mampu berkembang sesuai dengan kemajuan teknologi (Schwaber & Sutherland, 2024). Keunggulan lain dari *Scrum* adalah kemampuannya dalam menangani masalah kompleks yang berubah-ubah serta memberikan umpan balik cepat selama proses pengembangan, sehingga menghasilkan produk berkualitas yang memenuhi ekspektasi dan kebutuhan (Wahyudi, Ramadha, Andriani, & Alfarizy, 2024). Metode ini dipilih karena sifatnya fleksibel dalam menyesuaikan dinamika proses pengembangan serta mampu menghasilkan produk dengan cepat dan tepat.

1.2 Ruang Lingkup Magang

Proses pelaksanaan magang di PLN UP3 Yogyakarta berlangsung selama 6 bulan, dari 15 September 2023 hingga 15 Maret 2024. PLN bertanggung jawab atas penyediaan listrik dan terlibat dalam pengembangan infrastruktur dan investasi teknologi terbarukan. Namun, PLN menghadapi masalah dalam hal pengelolaan dan pemantauan data listrik yang tidak efisien dan kurang akurat yang menyebabkan kesulitan dalam pengambilan keputusan yang tepat waktu dan akurat. Hal ini dapat mempengaruhi kualitas layanan yang diberikan kepada pelanggan.

Untuk mengatasi masalah ini, tim magang mengembangkan sebuah sistem informasi yang diberi nama Sistem Informasi Bersama (SIBER). Aplikasi ini dirancang untuk membantu PLN dalam mengelola dan memantau daya serta tegangan listrik (*dashboard* AMR), monitoring infrastruktur kelistrikan (*dashboard* Kinerja), serta pengelolaan dan manajemen material pelanggan (*dashboard* Simantri). Sistem ini bertujuan untuk meningkatkan efisiensi dan akurasi pemantauan, dan memudahkan koordinasi antar divisi, serta meningkatkan kualitas pengelolaan data.

Pengembangan aplikasi dibagi menjadi tiga fase utama, masing-masing fase memerlukan dikerjakan dalam waktu dua bulan. Dalam laporan ini, yang akan dibahas hanya dua *dashboard*, yaitu *dashboard* AMR dan *dashboard* Kinerja, karena keduanya menyajikan visualisasi data yang relevan dengan judul laporan ini. Sementara itu, *dashboard* Simantri lebih berfokus pada pengelolaan permohonan material oleh Unit Layanan Pelanggan (ULP), dan tidak dibahas dalam laporan ini.

Aplikasi SIBER dikembangkan dengan menggunakan kerangka kerja *Scrum*, yang menerapkan prinsip kerja metode pengembangan *Agile*. Ruang lingkup tugas akhir ini

mencakup implementasi *Scrum* dalam aktivitas pengembangan selama magang. Pemilihan *Scrum* dilakukan berdasarkan observasi terkait metode pengembangan yang diperlukan selama pengembangan berlangsung. Beberapa aplikasi yang dikembangkan di PLN sebelumnya menggunakan metode *Waterfall*, bahkan ada yang tidak mengikuti metodologi pengembangan yang terstruktur, yang menyebabkan ketidakjelasan dalam penjadwalan dan lini masa, serta berisiko memperlambat penyelesaian proyek. Untuk menghindari masalah ini, tim IT PLN memilih untuk menerapkan metode *Scrum*, yang telah berhasil digunakan untuk proyek serupa sebelumnya. Dengan demikian, proyek dapat berjalan sesuai rencana dan selesai tepat waktu. Aktivitas yang dilakukan dalam pengerjaan pengembangan aplikasi SIBER sebagai berikut:

- a. Observasi dan analisa kebutuhan proyek SIBER
- b. Menentukan penggunaan metode pengembangan, yaitu *Scrum* yang akan digunakan selama pengembangan
- c. Menentukan framework untuk pengembangan, yaitu PHP, JavaScript, Figma, Pentaho, Tailwind, Bootstrap, dan library JS berupa Leaflet dan Chart.js
- d. Membuat dan membagi *task* untuk *programmer*
- e. Melakukan proses ETL (*Extract, Transform, Load*) data mentah dengan Pentaho
- f. Membuat rancangan desain prototipe proyek menggunakan Figma
- g. Membuat dokumen hasil pengerjaan dengan *excel*
- h. Membuat kode pemrograman sesuai dengan framework yang telah ditentukan
- i. Melakukan evaluasi dan revisi tiap minggunya

1.3 Tujuan

Tujuan pengembangan aplikasi SIBER dengan menggunakan kerangka kerja *Scrum* adalah sebagai berikut:

- a. Mengimplementasikan metode pengembangan *Scrum* dalam proyek aplikasi SIBER untuk meningkatkan efisiensi pengembangan dan pengelolaan waktu.
- b. Mengidentifikasi keuntungan dan kekurangan dari penerapan *Scrum* dalam pengembangan aplikasi SIBER sehingga bisa menjadi referensi pada proyek pengembangan selanjutnya.
- c. Meningkatkan kualitas pengelolaan dan pemantauan data dengan memastikan pengembangan dilakukan secara terencana dan terevaluasi dengan baik, agar aplikasi dapat berjalan tepat waktu dan sesuai dengan ekspektasi pengguna atau pemangku kepentingan.

1.4 Manfaat

Manfaat implementasi kerangka kerja *Scrum* dalam proses pengembangan sistem informasi SIBER adalah sebagai berikut:

- a. Kerangka kerja *Scrum* bersifat sangat fleksibel, memungkinkan penyesuaian terhadap kebutuhan selama pengembangan
- b. Melalui metode *Scrum* meningkatkan transparansi dan komunikasi dalam tim, membantu mengatasi hambatan komunikasi dan mempercepat resolusi masalah
- c. Dengan iterasi yang pendek dan evaluasi teratur, *Scrum* memungkinkan penyesuaian cepat terhadap perubahan kebutuhan atau risiko yang muncul
- d. *Product Backlog* yang dikelola secara terstruktur dan kemampuan untuk menyesuaikan prioritas setiap sprint memungkinkan pengelolaan yang lebih baik terhadap kebutuhan dan permintaan.

1.5 Sistematika Penulisan

Sistematika penulisan merupakan langkah penulisan dalam laporan tugas akhir untuk penjaluran magang. Dalam sistem ini memberikan penjelasan mengenai isi dari tiap bab dalam laporan tugas akhir. Berikut adalah runtutan dan penjelasan penyusunan laporan tugas akhir:

a. Bab I Pendahuluan

Bab ini berisi tentang gambaran umum proyek yang dikerjakan, informasi tentang perusahaan, dan ruang lingkup magang. Selain itu, bab ini juga berisi tentang tujuan dan manfaat proyek yang telah dikembangkan oleh penulis.

b. Bab II Dasar Teori

Bab ini berisi tentang pemahaman dasar teori yang relevan dengan laporan, meliputi visualisasi data, *dashboard*, *Scrum*, dan perbandingan metode pengembangan.

c. Bab III Pelaksanaan Magang

Bab ini membahas tentang kegiatan yang dilakukan selama magang, serta aktivitas yang telah diselesaikan.

d. Bab IV Refleksi Magang

Bab ini berisi tentang refleksi tentang teori yang telah dipelajari dibandingkan dengan pengalaman selama magang.

e. Bab V Kesimpulan dan Saran

Bab ini membahas tentang kesimpulan dari bab-bab sebelumnya dan saran terkait pelaksanaan magang di dunia kerja.

BAB II

DASAR TEORI

2.1 Visualisasi Data

Visualisasi data adalah proses untuk menyajikan data mentah dalam bentuk diagram agar informasi lebih mudah untuk dipahami (Nur Rizki, 2020). Berikut adalah beberapa jenis grafik visualisasi data yang umum digunakan:

a. Diagram garis (*Line Chart*)

Diagram garis digunakan untuk menggambarkan kondisi yang berlangsung secara terus menerus dalam rentang waktu yang berkelanjutan (Pratikno, Prastiwi, & Rahmawati, 2020). Misalnya, untuk menunjukkan perubahan persentase energi listrik yang hilang saat dialirkan dari pembangkit listrik ke konsumen melalui jaringan transmisi dan distribusi.

b. Diagram batang (*Bar Chart*)

Diagram ini digunakan untuk membandingkan nilai atau jumlah dalam kategori yang berbeda (Fernando, 2018). Contoh penggunaannya adalah membandingkan jumlah pelanggan yang menggunakan daya listrik tertentu setiap bulan sepanjang tahun.

c. Diagram lingkaran (*Pie Chart*)

Menunjukkan persentase dari nilai suatu item yang menunjukkan suatu rangkaian data (Fernando, 2018). Misalnya, untuk menampilkan distribusi status tegangan pelanggan dalam suatu area seperti persentase pelanggan dengan tegangan normal, perlu perhatian, dan perlu perbaikan.

d. Visualisasi Peta

Menampilkan data pada tata letak geografis (Fernando, 2018), membantu memahami distribusi listrik dan status listrik pelanggan berdasarkan lokasi. Misalnya, peta interaktif yang menunjukkan lokasi pelanggan dengan status tegangan listrik yang berbeda-beda.

2.2 Dashboard

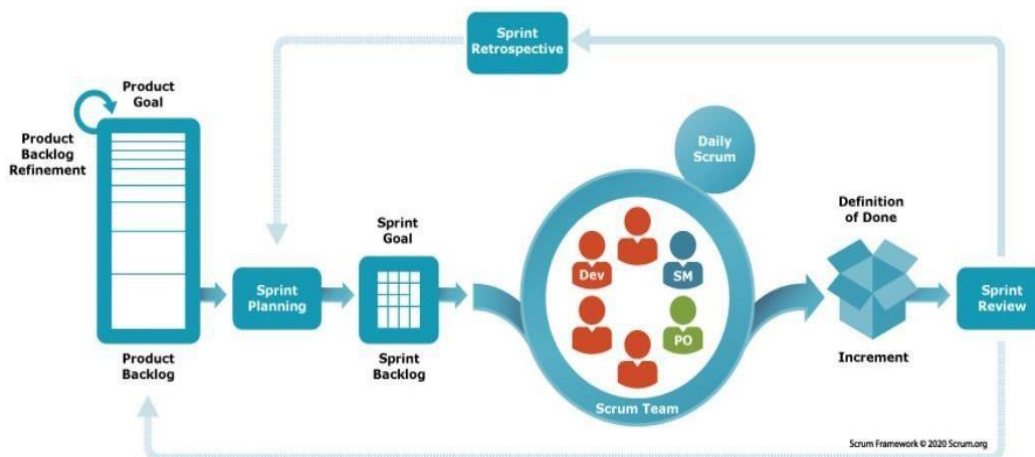
Dashboard adalah tampilan antarmuka yang menyajikan informasi secara visual berupa data tabel, gambar, diagram grafik, maupun letak geografis sehingga memudahkan pengguna untuk memahami informasi dengan cepat (Setiawan, 2023). *Dashboard* visualisasi data dikembangkan secara dinamis agar mampu untuk membaca data secara *real-time*, terbukti menjadi alat yang efektif dalam memonitoring dan menganalisis data secara berkelanjutan,

dapat membantu dalam pengambilan keputusan secara cepat dan tepat berdasarkan data yang tersedia.

2.3 Scrum

Scrum adalah metode pengembangan perangkat lunak yang menekankan fleksibilitas dan pengembangan cepat (Chasbulloh, Kautsar, Riyanto, Hapsari, & Widodo, 2023). Dalam pelaksanaannya, *Scrum* melibatkan beberapa peran kunci untuk memastikan produk berkualitas dan kerjasama yang efektif selama periode sprint. Peran dalam tim *Scrum* adalah sebagai berikut:

- Product Owner*, bertanggung jawab untuk mengelola *Product Backlog*, menetapkan prioritas fitur, memberikan umpan balik secara bertahap, dan membuat keputusan strategis tentang arah produk berdasarkan kebutuhan dan tujuan bisnis (Schwaber & Sutherland, 2024).
- Scrum Master*, bertanggung jawab dalam memimpin tim dan memastikan bahwa tim mengikuti metode *Scrum* dengan benar (Schwaber & Sutherland, 2024).
- Development Team*, bertanggung jawab untuk menghasilkan produk fungsional pada akhir setiap sprint. *Development Team* menerima tugas dari *Product Backlog* dan bekerja secara kolaboratif untuk menyelesaikannya (Schwaber & Sutherland, 2024).



Gambar 2. 1 Model Pengembangan *Scrum*

Pada Gambar 2. 1, model ini menerapkan langkah-langkah dalam pengembangan aplikasi menggunakan metode *Scrum*. Metode ini terdiri dari beberapa tahapan yang berurutan dan iteratif untuk memastikan pengembangan perangkat lunak yang efektif dan adaptif. Tahapan

tersebut meliputi: *Product Goal*, *Product Backlog*, *Sprint Planning*, *Sprint*, *Increment*, *Sprint Review*, *Sprint Retrospective*, dan *Daily Sprint* (Schwaber & Sutherland, 2024).

- a. *Product Goal*: langkah pertama adalah menetapkan tujuan produk yang ingin dicapai. *Product Goal* ditentukan berdasarkan masalah yang melatarbelakangi pengembangan aplikasi atau sistem tersebut dibuat dan disepakati bersama tim.
- b. *Product Backlog*: membuat daftar semua pekerjaan yang perlu dilakukan untuk mengembangkan produk. *Product Backlog* dikelola oleh *Product Owner* yang berisi semua fitur, perubahan, dan perbaikan yang diperlukan.
- c. *Sprint Planning*: pada tahapan ini, tim memilih item dari *Product Backlog* yang akan dikerjakan selama *Sprint* dan membuat rencana kerja untuk menyelesaikannya.
- d. *Sprint*: dilakukan selama periode tertentu untuk mengerjakan tugas yang telah direncanakan selama *Sprint Planning* bersama dengan anggota tim lainnya.
- e. *Daily Sprint*: pertemuan singkat yang diadakan setiap hari untuk melaporkan kemajuan tim, mengidentifikasi hambatan yang dihadapi, dan merencanakan pekerjaan untuk hari itu.
- f. *Increment*: pada akhir setiap *Sprint*, tim harus dapat menghasilkan *Increment* dari versi produk yang dibuat atau diperbarui. Hasil ini harus dapat diuji dan digunakan oleh pengguna akhir.
- g. *Sprint Review*: setelah *Sprint*, tahap ini dilakukan untuk memeriksa *Increment* yang telah dihasilkan dan menerima umpan balik yang membantu dalam proses perbaikan berikutnya.
- h. *Sprint Retrospective*: dilakukan untuk merefleksikan *Sprint* yang telah berlalu, mengidentifikasi aspek yang perlu ditingkatkan, dan merencanakan perbaikan untuk *Sprint* selanjutnya.

2.4 Perbandingan metode

Proses pengembangan perangkat lunak (*software development process*) adalah struktur yang diterapkan dalam pengembangan perangkat lunak untuk memastikan keberhasilan dalam proyek melalui tahapan tertentu (Widiyanto, 2018). Selama pengembangan aplikasi di PLN, dilakukan perbandingan antara dua metode pengembangan, yaitu *Waterfall* dan *Scrum*, berdasarkan pengalaman proyek sebelumnya. Perbandingan ini ditampilkan pada Tabel 2. 1.

Tabel 2. 1 Perbandingan Metode Pengembangan *Waterfall* dan *Scrum*

No.	Karakteristik Metode	Metode <i>Waterfall</i>	Metode <i>Scrum</i>	Kebutuhan
1.	Fleksibilitas metode	Kaku	Fleksibel	Membutuhkan metode yang fleksibel
2.	Pendefinisian kebutuhan	Harus sudah jelas	Fleksibel dapat berubah-ubah	Membutuhkan pendefinisian kebutuhan yang dapat berubah-ubah
3.	Perubahan kebutuhan pada saat pengembangan	Sulit untuk diubah	Mudah dalam mengubah <i>requirement</i>	Membutuhkan metode yang mudah dalam mengubah kebutuhan dalam pengembangan
4.	Waktu yang dibutuhkan dalam mengimplementasikan metode saat pengembangan	Lambat	Cepat	Membutuhkan metode pengembangan yang cepat dalam pengimplementasiannya
5.	Tingkat kemudahan dalam mengimplementasikan metode pengembangan	Mudah	Mudah	Membutuhkan metode pengembangan yang mudah untuk diimplementasikan
6.	Pemilihan metode	Metode yang dipilih di dalam makalah ini adalah metode <i>Scrum</i>		

Pada Tabel 2. 1 membandingkan metode *Waterfall* dan *Scrum*, berdasarkan fleksibilitas, pendefinisian kebutuhan, kemampuan mengubah kebutuhan, waktu implementasi, dan kemudahan penerapan. Metode *Waterfall* yang kaku dan linear, mengharuskan semua kebutuhan didefinisikan di awal dan sulit diubah selama pengembangan, sehingga prosesnya lebih lambat namun terstruktur. Sebaliknya, *Scrum* yang fleksibel dan iteratif memungkinkan perubahan yang berkelanjutan, mempercepat pengembangan melalui iterasi yang pendek (*Sprint*), dan memudahkan penyesuaian berdasarkan umpan balik. Berdasarkan lingkungan pengembangan dan kebutuhan proyek, metode *Scrum* dipilih karena fleksibilitasnya dalam menghadapi dinamika pengembangan dan kemampuannya menghasilkan produk yang sesuai dengan kebutuhan secara cepat dan tepat (Al Attar, Kusumasari, & Fauzi, 2022).

2.5 Tinjauan Pustaka

Tugas akhir ini membahas terkait penggunaan metodologi *Scrum* dalam pengembangan perangkat lunak yang sangat relevan, terutama untuk proyek berbasis visualisasi data seperti *dashboard* pada proyek PLN UP3 Yogyakarta. *Scrum* sebagai metodologi pengembangan

iteratif yang fleksibel, mendukung proses pengembangan yang adaptif sesuai kebutuhan. Beberapa penelitian serupa telah menunjukkan hasil yang positif dalam penerapan *Scrum*, khususnya untuk manajemen proyek yang melibatkan tim kolaboratif dan iterasi berkelanjutan.

Penelitian oleh (Gutama, 2020), berjudul “Implementasi *Scrum* pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP),” menunjukkan bahwa *Scrum* efektif dalam membantu pengelolaan tugas tim pada proyek berbasis monitoring dan evaluasi. Studi ini relevan untuk pengembangan *dashboard* dalam penelitian ini, karena berfokus pada pelacakan tugas secara kolaboratif. Selanjutnya, penelitian oleh (Wahyudi, Ramadha, Andriani, & Alfarizy, 2024) berjudul “Analisa Perbandingan *Software Development Model* antara Metode *Scrum* dan Metode *Incremental*,” membahas keunggulan *Scrum* dalam kecepatan iterasi, fleksibilitas perubahan kebutuhan, dan pengelolaan risiko dibandingkan model *Incremental*. Hal ini mendukung implementasi *Scrum* untuk *dashboard* visualisasi data, yang membutuhkan pembaruan dan penyempurnaan fungsi yang cepat. Selain itu, penelitian oleh (Alexander & Noranita, 2021), berjudul “Penerapan *Business Intelligence Dashboard* pada Sekolah Tinggi Pertanahan Nasional Menggunakan Metodologi *Scrum*,” menyoroti penerapan *Scrum* untuk membangun *dashboard business intelligence* (BI). Studi ini relevan karena berfokus pada pembagian *Sprint* untuk menyelesaikan fungsi-fungsi *dashboard* serta pentingnya iterasi untuk meningkatkan tampilan dan akurasi data.

BAB III

PELAKSANAAN MAGANG

3.1 Aktivitas Magang

Selama 6 bulan magang, penulis bersama tim terlibat dalam berbagai aktivitas yang meliputi pengembangan aplikasi dan tugas tambahan. Aktivitas utama melibatkan pengembangan aplikasi web untuk mendukung beberapa divisi di perusahaan, termasuk Perencanaan, Transaksi Energi, dan Konstruksi. Tugas utama ini dirancang untuk meningkatkan efisiensi dan produktivitas dalam pengelolaan data dan proses di setiap divisi.

Selain pekerjaan utama, terdapat beberapa tugas tambahan yang dilaksanakan di pertengahan periode magang, meliputi input data dari aplikasi lokal pegawai ke *excel*, yang bertujuan untuk memberikan laporan kepada beberapa divisi. Setiap tugas tambahan biasanya memiliki jangka waktu pengerjaan sekitar satu sampai dua minggu.

Selama magang, penulis berperan sebagai *frontend developer* dan bekerja sama dengan *backend developer* untuk memastikan semua aktivitas berjalan sesuai rencana dan mencapai hasil yang diharapkan. Aktivitas ini mencakup berbagai aspek teknis dan operasional, memberikan wawasan luas dalam pengembangan aplikasi web serta pengelolaan dan analisis data.

3.1.1 Tempat dan Durasi Magang

Penulis menjalani program magang di PLN UP3 Yogyakarta. PLN UP3 Yogyakarta terletak di Jl. Gedongkuning No. 3, Tegal Tandan, Banguntapan, Kec. Banguntapan, Kabupaten Bantul, Daerah Istimewa Yogyakarta. PLN UP3 atau Unit Pelaksanan Pelayanan Pelanggan, yang berada di bawah divisi distribusi PLN, dengan fungsi utama untuk mengelola distribusi listrik serta pelayanan kepada pelanggan di wilayah operasionalnya. Beberapa tugas utama PLN UP3 adalah:

- a. Distribusi Listrik: UP3 bertanggung jawab atas distribusi listrik dari gardu induk ke rumah-rumah, bisnis, dan fasilitas publik di wilayah tertentu.
- b. Pemeliharaan Jaringan: UP3 melakukan pemeliharaan rutin dan perbaikan pada jaringan distribusi listrik untuk memastikan pasokan listrik yang handal dan minim gangguan.
- c. Penyambungan Baru: UP3 mengelola dalam hal penyambungan baru bagi pelanggan yang ingin mendapatkan layanan listrik, termasuk kepada instalasi meteran dan pengujian sistem

- d. Pembacaan Meter dan Penagihan: UP3 bertanggung jawab atas pembacaan meter listrik pelanggan dan penagihan berdasarkan penggunaan listrik
- e. Layanan Pelanggan: UP3 memberikan layanan kepada pelanggan, termasuk menanggapi keluhan, memberikan informasi tentang produk dan layanan PLN, serta menangani masalah teknis.

PLN UP3 Yogyakarta memiliki 6 divisi utama, yaitu: divisi Perencanaan, divisi Konstruksi, divisi Transaksi Energi, divisi Keuangan dan Umum, divisi Jaringan, dan divisi Pemasaran (SARPP). Penulis ditempatkan di divisi perencanaan untuk merancang aplikasi yang akan dibuat dengan melibatkan beberapa divisi, yaitu divisi perencanaan, konstruksi, dan transaksi energi di PLN UP3 Yogyakarta. Penulis magang selama 6 bulan dari periode September 2023 sampai Maret 2024, dengan pengerjaan proyek tiap divisi berlangsung selama dua bulan.

3.1.2 Jam Kerja dan Tugas Harian

Penulis bekerja selama program magang dengan jadwal harian dimulai dari pukul 08.00 – 16.00 WIB. Waktu istirahat berlangsung setiap pukul 11.00 – 13.00 WIB. Penulis bekerja sebagai *frontend developer*, dengan tugas utama merancang dan mengimplementasikan kode yang akan ditampilkan kepada pengguna. Selain tugas utama tersebut, penulis juga membantu *backend developer* untuk mengekstrak data yang akan digunakan, termasuk mengolah data mentah yang diterima perusahaan menjadi data siap pakai oleh *backend developer*. Selain tanggung jawab utama, penulis juga beberapa kali dilibatkan untuk tugas tambahan dari perusahaan, yaitu menginputkan data dari aplikasi lokal pegawai ke *excel*.

3.1.3 Lingkungan Kerja dan Kolaborasi Tim

Penulis menjalani magang di PLN UP3 Yogyakarta yang menerapkan tata nilai AKHLAK dalam lingkungan kerja, yaitu:

- a. Amanah: Memegang teguh kepercayaan yang diberikan
- b. Kompeten: Terus belajar dan mengembangkan kapabilitas
- c. Harmonis: Saling peduli dan menghargai perbedaan
- d. Loyal: Berdedikasi dan mengutamakan kepentingan bangsa dan negara
- e. Adaptif: Terus berinovasi dan antusias dalam menggerakkan ataupun menghadapi perubahan
- f. Kolaboratif: Membangun kerjasama yang sinergis

Suasana kerja yang kondusif, kolaboratif, dan penuh semangat menjadikan pengalaman magang lebih bermakna dan bermfaat. Budaya kerja di PLN mendorong inovasi dan kreatifitas, memberikan kebebasan kepada penulis untuk mengeksplorasi ide dalam mengembangkan aplikasi SIBER. Karyawan dan pemegang didorong untuk berbagi ide serta memberikan umpan balik yang positif satu sama lain. Setiap anggota tim memiliki kesempatan yang sama untuk berkontribusi dan belajar dari pengalaman ini.

Penulis juga berkolaborasi dengan tim informatika lainnya dalam pengembangan aplikasi SIBER. Setiap anggota tim diberikan kesempatan untuk melaporkan progres, hambatan, dan rencana kerja secara singkat. Jika muncul hambatan, diskusi diarahkan untuk mencari solusi praktis, seperti menjadwalkan sesi tambahan untuk mengatasi *bug* atau isu teknis tertentu. Pertemuan harian dijaga agar berlangsung singkat, maksimal 15-30 menit, dengan fokus pada poin-poin utama masalah. Setelah pertemuan, ringkasan diskusi dibagikan melalui Whatsapp atau Diskord untuk memastikan semua anggota tetap sinkron.

Kolaborasi ini dilakukan secara terorganisir dengan mengutamakan komunikasi yang efektif. Untuk mendukung komunikasi yang efektif, baik di dalam maupun di luar kantor, digunakan beberapa alat komunikasi, yaitu:

- a. WhatsApp: digunakan untuk komunikasi terkait magang, seperti koordinasi antara pemegang dan *leader* tim. Grup WhatsApp juga menjadi sarana utama untuk menyampaikan pengumuman dan progres pekerjaan.
- b. Discord: digunakan untuk diskusi dan kolaborasi di luar jam kerja kantor atau ketika anggota tim berada di lokasi berbeda. Fitur seperti *chat*, *voice call*, dan *screen sharing* membantu dalam menyelesaikan masalah secara *real-time* dan meningkatkan efisiensi kerja.

3.1.4 Manajemen Proyek

Manajemen proyek yang diterapkan selama magang dalam pengembangan aplikasi SIBER mengikuti prinsip *Scrum*. Aktivitas magang terbagi menjadi beberapa fase utama dan aktivitas tambahan yang disesuaikan dengan kebutuhan perusahaan dan divisi terkait. Pembagian tugas dilakukan secara manual menggunakan *excel* sebagai alat utama untuk mendokumentasikan, mengelola, dan memantau distribusi tugas setiap anggota tim. Beberapa aktivitas utama dan tambahan tersebut meliputi:

- a. Aktivitas utama

Penulis dan tim fokus mengembangkan aplikasi web untuk monitoring dan manajemen energi, yaitu aplikasi SIBER, yang mendukung tiga divisi utama, yaitu Perencanaan, Transaksi Energi, dan Konstruksi. Pembuatan aplikasi dilaksanakan selama 6 bulan, bulan pertama dan kedua mengembangkan *dashboard* AMR, bulan ketiga dan keempat mengembangkan *dashboard* Kinerja, dan bulan kelima dan keenam mengembangkan *dashboard* Simantri.

b. Aktivitas tambahan

Menginputkan data Saluran Kabel Udara Tegangan Menengah (SKUTM) dari aplikasi lokal pegawai ke *excel*. Aktivitas ini dilakukan sebagai tugas tambahan dari divisi Perencanaan, terutama untuk pembuatan laporan evaluasi kinerja dan kondisi teknis SKUTM. Hasil dari tugas ini yaitu laporan kinerja yang membantu divisi Perencanaan dalam menyiapkan laporan yang terkait dengan evaluasi dan kinerja untuk mendukung teknis yang lebih baik.

c. Durasi dan linimasa:

Linimasa dalam proyek utama berlangsung selama 2 bulan, dengan rincian pengerjaan yang diatur melalui *Sprint Planning* di awal setiap fase. Total durasi pengembangan adalah 6 bulan, dengan aktivitas magang dimulai pada bulan September 2023 hingga Maret 2024. Durasi pengerjaan tugas tambahan, seperti input data SKUTM, biasanya diberikan waktu 1-2 minggu tergantung pada kompleksitas dan kebutuhan divisi terkait.

d. Peran dan kolaborasi:

Penulis berperan sebagai *frontend developer* yang bertanggung jawab atas desain antarmuka dan implementasi kode yang ditampilkan kepada pengguna. Selain itu, penulis juga berkolaborasi dengan *backend developer* dalam proses pengolahan dan ekstraksi data untuk tugas tambahan. Kolaborasi dilakukan secara intensif dengan rekan tim dan divisi terkait untuk memastikan semua fitur yang dikembangkan dapat memenuhi kebutuhan operasional PLN UP3 Yogyakarta.

3.2 Proyek Sistem Informasi Bersama (SIBER)

Aplikasi Sistem Informasi Bersama (SIBER) merupakan aplikasi berbasis web untuk monitoring dan manajemen energi di PLN UP3 Yogyakarta dengan mengintegrasikan data dari tiga divisi utama: perencanaan, transaksi energi, dan konstruksi. SIBER menyediakan *dashboard* Kinerja untuk monitoring infrastruktur kelistrikan, *dashboard* AMR “*Automatic Meter Reading*” untuk pemantauan daya dan tegangan listrik, dan *dashboard* Simantri untuk

mengelola permohonan pelanggan terkait pemasangan listrik baru atau perbaikan jaringan yang diinput oleh Unit Layanan Pelanggan (ULP).

Selama masa magang yang berlangsung 6 bulan, pengembangan aplikasi dibagi menjadi tiga fase utama: bulan pertama hingga kedua mengembangkan *dashboard* AMR, bulan ketiga hingga keempat mengembangkan *dashboard* Kinerja, dan bulan kelima hingga keenam mengembangkan *dashboard* Simantri, termasuk proses pengujian dan implementasi di setiap tahapannya. Laporan ini hanya menampilkan *dashboard* AMR dan *dashboard* Kinerja karena hasil akhir kedua *dashboard* tersebut berupa visualiasasi data yang sesuai dengan judul yang diangkat, sedangkan untuk *dashboard* Simantri lebih lebih fokus pada pengelolaan dan manajemen permohonan pelanggan.

3.2.1 Deskripsi Proyek

Aplikasi yang dikembangkan penulis bersama rekan pemegang informatika adalah Sistem Informasi Bersama (SIBER). SIBER adalah sebuah sistem informasi yang berfokus pada visualisasi *dashboard*, menyediakan berbagai visualisasi data seperti *line chart*, *mapping*, dan *bar chart*. Selain itu, SIBER juga memiliki fungsionalitas dapat mengelola daftar permohonan pelanggan terkait pemasangan listrik baru atau perbaikan jaringan yang diinputkan oleh unit layanan pelanggan (ULP).

3.2.2 Tujuan Proyek

Tujuan dari proyek SIBER adalah untuk mengembangkan *dashboard* visualisasi data secara *real-time* yang mencakup pemantauan daya serta tegangan listrik yang digunakan oleh pelanggan melalui *dashboard* AMR “*Automatic Meter Reading*” dari divisi Transaksi Energi. Proyek ini juga bertujuan untuk melakukan monitoring terhadap infrastruktur kelistrikan melalui *dashboard* Kinerja dari divisi Perencanaan, serta pengelolaan daftar permohonan pelanggan terkait pemasangan listrik baru atau perbaikan jaringan dari UP3, admin, maupun unit layanan pelanggan (ULP) melalui *dashboard* Simantri dari divisi Konstruksi. Proyek ini bertujuan untuk meningkatkan efisiensi, akurasi, dan koordinasi antar divisi, serta memastikan aplikasi dapat diselesaikan tepat waktu dan sesuai dengan kebutuhan pemangku kepentingan.

3.2.3 Analisis Kebutuhan Data

a. Jenis Data yang Diperlukan

Data *Dashboard* AMR: Dalam data AMR membutuhkan data dari file *monitoring_PLN* untuk dijadikan acuan yang berisi data *satuan_ap*, *namaap*, *unitup*, *satuan_up*, *namaup*, *location code*, nama pelanggan, tarif, *power*, *read date*, *voltage_l1*, *voltage_l2*, *voltage_l3*. Jenis datanya adalah numerik (dalam satuan *watt* dan *volt*) dan *string*.

Data *Dashboard* Kinerja: Dalam data Kinerja membutuhkan data dari file *to_base* untuk dijadikan acuan yang berisi data *thbl*, pelanggan, *saidi*, *saifi*, *rec_time*, *ggn 100kms*, *ggn reclosure*, *susut*, *hpl*, *kwh* jual, dan *kwh* produksi. Jenis datanya adalah numerik dan *string*.

b. Sumber Data

Dashboard AMR: Sumber data *dashboard* ini berasal dari internal PLN yang mencatat data penggunaan energi dan pemeliharanya.

Dashboard Kinerja: Data pemeliharaan dan perbaikan yang diinputkan oleh tim teknis dan operasional.

c. Pengelolaan Data

Pengelolaan data dilakukan melalui beberapa tahap yang terintegrasi. Pertama, data dikumpulkan dari internal perusahaan. Data yang diperoleh dikumpulkan secara berkala ke server pusat untuk kemudian diproses dan disimpan. Selanjutnya, proses pemrosesan dan transformasi data dilakukan menggunakan alat ETL (*Extract, Transform, Load*) seperti Pentaho. Alat ini digunakan untuk membersihkan, memformat, dan menggabungkan data yang berasal dari berbagai sumber, sehingga memastikan data berasal dalam format yang siap digunakan untuk visualisasi. Terakhir, data yang telah diproses disimpan dalam basis data terpusat yang dapat diakses secara *real-time* oleh tim pengembang untuk keperluan analisis dan visualisasi.

d. Validasi dan Verifikasi Data

Untuk memastikan bahwa data yang digunakan dalam visualisasi sudah lengkap, akurat, dan sesuai dengan kebutuhan pengguna, dilakukan serangkaian proses validasi dan verifikasi data secara menyeluruh. Validasi awal dilakukan dengan memeriksa data yang diterima dari sistem monitoring, seperti AMR dan input manual dari tim teknis. Pemeriksaan mencakup konsistensi format dan kelengkapan data, misalnya dengan memastikan kelengkapan data dan konsistensi format. Hasil validasi dan visualisasi awal kemudian ditinjau oleh pemangku kepentingan, seperti tim teknis, untuk memastikan data yang disajikan sesuai dengan ekspektasi dan kebutuhan. Proses ini memungkinkan tim untuk menerima umpan balik dan melakukan revisi jika diperlukan, sehingga *dashboard* yang dihasilkan dapat memberikan informasi yang akurat dan relevan bagi pengguna.

3.2.4 *Tools* Proyek

Proyek SIBER menggunakan beberapa *tools* untuk mendukung pengembangan dan pelaksanaan proyek. *Tools* yang digunakan antara lain:

- a. Github: Platform untuk kolaborasi pengembangan perangkat lunak secara terpusat. Kelebihan dari platform ini adalah memberikan kontrol versi yang memungkinkan tim melacak perubahan kode. Kekurangan dari platform ini adalah membutuhkan koneksi internet untuk kolaborasi *real-time*.
- b. Figma: Alat desain prototipe yang memungkinkan kolaborasi antar anggota tim dalam mendesain antarmuka aplikasi. Kelebihan alat ini adalah mendukung kolaborasi tim desain secara langsung dan dapat diakses dari mana saja. Kekurangan dari alat ini adalah bergantung pada koneksi internet.
- a. PHP: Bahasa pemrograman yang digunakan untuk membangun aplikasi sisi server, karena kestabilan dan fleksibilitasnya dalam pengembangan aplikasi web. Kelebihan bahasa ini adalah stabil dan fleksibel, serta kompatibel dengan berbagai server web. Kekurangannya adalah dibutuhkan pengembang yang berpengalaman untuk pengelolaan kode besar. Alternatif bahasa pemrograman lain adalah Python yang lebih fleksibel dan mendukung *machine learning*.
- b. JavaScript: Bahasa pemrograman untuk pengembangan interaktif di sisi klien, penting untuk visualisasi data dinamis pada *dashboard*. Kelebihan bahasa ini adalah mendukung visualisasi data dinamis dan interaktif, sangat relevan untuk *dashboard* serta didukung oleh banyak *library* seperti Chart.js untuk visualisasi data. Kekurangan dari bahasa ini adalah kompleksitas kode untuk proyek besar tanpa *framework* yang memadai.
- c. Tailwind dan Bootstrap: *Framework* CSS untuk mempercepat proses pengembangan antarmuka aplikasi dengan desain responsif dan mudah disesuaikan. Kelebihan dari *framework* ini adalah mempercepat proses *styling* tanpa memulai dari nol. Kekurangannya adalah pada *bootstrap* cenderung menghasilkan desain generik tanpa banyak penyesuaian, sedangkan pada *tailwind* membutuhkan waktu adaptasi lebih lama bagi pengembang pemula.
- d. Leaflet: *Library* JavaScript untuk integrasi peta dalam aplikasi, mendukung visualisasi lokasi pelanggan pemakai daya dan tegangan listrik. Kelebihan dari *library* adalah sifatnya yang *open-source* dan ringan ringan dibandingkan peta lain seperti Google Maps API. Kekurangannya adalah fitur lebih terbatas dibandingkan Google Maps API, seperti analisis

geospasial lanjutan. Alternatif lain adalah Google Maps API yang lebih lengkap tetapi berbayar untuk pengguna tertentu.

- e. Pentaho: Alat untuk ETL (*Extract, Transform, Load*) data. Kelebihan dari alat ini adalah mendukung pengolahan data mentah untuk menghasilkan laporan dan visualisasi yang terstruktur serta kompatibilitas dengan berbagai sumber data. Kekurangan dari alat ini adalah membutuhkan pembelajaran awal untuk memanfaatkan fitur secara maksimal.
- f. FTP Modzilla: Alat untuk mentransfer file antar server, penting untuk penyebaran aplikasi dan pengelolaan server secara efisien. Kelebihan dari alat ini adalah mudah digunakan dan mendukung transfer file yang aman. Kekurangannya adalah kurang efisien dibandingkan metode *deployment* otomatis seperti CI/CD.
- g. Discord: Platform untuk komunikasi tim untuk mendukung kolaborasi jarak jauh dengan update proyek secara *real-time*. Kelebihan platform ini adalah mendukung komunikasi dengan fitur obrolan suara, video, dan teks. Kekurangannya adalah harus terhubung jaringan internet
- h. Microsoft *Excel*: Alat bantu untuk mengelola dan memformat data mentah sesuai dengan templat *dashboard*. Data ini kemudian diproses lebih lanjut untuk menghasilkan visualisasi dalam bentuk diagram atau grafik yang akan ditampilkan pada *dashboard* SIBER. Kelebihan dari alat ini adalah memiliki antarmuka yang intuitif serta mudah digunakan oleh karyawan yang memudahkan penginputan dan pengelolaan data. Kekurangan dari alat ini adalah kurang efektif dalam menangani dataset yang sangat besar bisa membuat aplikasi menjadi lambat atau bahkan *crash*.

3.2.5 Pemilihan Metode yang Digunakan

Metode pengembangan yang dipilih untuk aplikasi SIBER adalah *Scrum* didasarkan pada karakteristik proyek dan kebutuhan pengelolaan yang fleksibel, iteratif, serta berbasis tim yang kolaboratif. Berikut beberapa alasan mengapa *Scrum* lebih cocok dibandingkan dengan metode lain seperti *Waterfall* atau *Kanban*:

- a. Fleksibilitas dan responsif terhadap perubahan: *Scrum* memungkinkan perubahan kebutuhan dan prioritas selama siklus pengembangan. Proyek SIBER, yang melibatkan beberapa divisi di PLN dengan kebutuhan yang dinamis, memerlukan metode yang dapat menyesuaikan dengan cepat terhadap umpan balik dan perubahan data teknis, sesuatu yang lebih sulit dicapai dengan metode *Waterfall* yang bersifat lebih kaku dan linier.

- b. Iterasi cepat dan pengujian yang berkelanjutan: *Scrum* mendukung iterasi cepat, di mana setiap *Sprint* menghasilkan fitur atau peningkatan yang dapat diuji dan dievaluasi langsung oleh pemangku kepentingan. Ini sangat relevan dalam proyek SIBER yang mengembangkan berbagai *dashboard* dengan data yang harus selalu diperbarui dan divalidasi secara *real-time*. *Sprint* yang singkat (biasanya 2-4 minggu) memungkinkan tim untuk mendapatkan *feedback* yang cepat dan memperbaiki kesalahan sebelum melanjutkan pengembangan lebih lanjut, meningkatkan efisiensi dan kualitas produk akhir. Jika dibandingkan dengan *Kanban* yang lebih cocok untuk proyek dengan alur kerja yang kontinu dan tidak banyak perubahan dalam *scope*, sedangkan SIBER membutuhkan perubahan dan penyesuaian cepat seiring kemajuan proyek.
- c. Kolaborasi tim yang efektif: *Scrum* mendorong kolaborasi aktif antara semua anggota tim, termasuk pengembang, *Project Leader*, dan *Product Owner*, melalui *Daily Scrum* dan sesi review. Ini sangat relevan untuk proyek SIBER yang melibatkan beberapa divisi (Transaksi Energi, Perencanaan, Konstruksi) dan memerlukan komunikasi yang intensif dan terkoordinasi dengan baik.
- d. Responsif terhadap *feedback*: *Sprint Review* dan *retrospective* memberikan umpan balik langsung dari pemangku kepentingan atau pengguna aplikasi. Ini sangat penting untuk proyek SIBER, di mana aplikasi yang dikembangkan harus terus disesuaikan dengan kebutuhan dan ekspektasi pengguna. Sedangkan pada *Waterfall* jika terdapat kesalahan atau perubahan dalam fase akhir, perubahan tersebut bisa mempengaruhi keseluruhan proyek dan jadwal secara signifikan.
- e. Transparansi dan visibilitas: *Scrum* menyediakan visibilitas tinggi terhadap progres proyek melalui artefak seperti *backlog* dan *Sprint Review* yang memungkinkan tim dan pemangku kepentingan untuk memantau perkembangan proyek. Ini memungkinkan perbaikan lebih awal jika ada kendala atau masalah yang muncul selama pengembangan.

Keuntungan yang didapatkan adalah:

- a. Penilaian dan penyesuaian berkelanjutan melalui iterasi pendek, memastikan kualitas produk yang tinggi.
- b. Fleksibilitas tinggi memungkinkan penyesuaian rencana dan prioritas berdasarkan umpan balik.
- c. Pengelolaan terstruktur dan fokus pada iterasi pendek untuk meningkatkan efisiensi tim.
- d. Kolaborasi aktif antara tim pengembang, *Project Leader*, dan *Product Owner*.

3.3 Implementasi *Scrum* (Pelaksanaan Proyek Siber)

Aplikasi SIBER menggunakan metode pengembangan *Scrum* yang menekankan fleksibilitas dan pengembangan cepat (Chasbulloh, Kautsar, Riyanto, Hapsari, & Widodo, 2023). Proses pelaksanaan *Scrum* terdiri dari beberapa tahap, seperti yang ditunjukkan dalam Gambar 2. 1, yaitu *Product Goal*, *Product Backlog*, *Sprint Planning*, *Sprint*, *Increment*, *Sprint Review*, *Sprint Retrospective*, dan *Daily Scrum*.

Tim pengembang terdiri dari dua *frontend developer*, termasuk penulis yang berfokus pada antarmuka dan pengalaman pengguna, serta dua *backend developer* yang bertanggung jawab untuk memastikan fungsionalitas dan integrasi sistem berjalan dengan lancar. Kolaborasi antara *frontend* dan *backend developer* dilakukan melalui diskusi harian untuk meminimalkan *bug* integrasi, khususnya ketika mengembangkan fitur baru yang melibatkan data dinamis.

Selain itu, proyek ini dipandu oleh seorang *Product Owner* yang bertugas mengarahkan visi produk dan menetapkan prioritas pengembangan, serta seorang *Scrum Master* yang juga berperan sebagai *Project Leader*. *Scrum Master* memastikan tim mengikuti kerangka kerja *Scrum* dengan baik meskipun menghadapi tantangan dalam membagi peran sebagai pemimpin proyek. Untuk mengatasi tantangan dualitas peran ini, *Daily Scrum* dioptimalkan sebagai forum utama untuk memperoleh pembaruan terkini. Dalam situasi di mana peran sebagai *Project Leader* lebih dominan, seperti saat membuat keputusan strategis, *Scrum Master* memprioritaskan fokus pada aspek tersebut. Sebaliknya, untuk tugas operasional seperti pengelolaan *backlog* atau memfasilitasi diskusi tim, *Scrum Master* lebih mengutamakan perannya sesuai kerangka kerja *Scrum*. Sesi *Sprint Retrospective* juga dimanfaatkan untuk mengevaluasi efektivitas pembagian peran ini, sehingga memungkinkan tim untuk melakukan penyesuaian yang diperlukan.

3.3.1 Tahapan *Scrum* dalam Pengembangan Proyek

Dalam pengembangan proyek yang menerapkan *Scrum*, peran tim dibagi menjadi tiga, yaitu *Product Owner* selaku Asisten Manajer divisi Perencanaan untuk *dashboard* AMR dan Asisten Manajer divisi Transaksi Energi untuk *dashboard* Kinerja, *Scrum Master* dari divisi Perencanaan, dan *Development Team* dari tim magang PLN.

a. *Product Goal*

Tujuan aplikasi SIBER adalah mengembangkan aplikasi *dashboard real-time* untuk visualisasi data terkait pemantauan daya dan tegangan listrik (*dashboard* AMR) dan

monitoring infrastruktur kelistrikan (*dashboard* Kinerja). Tujuan ini ditetapkan oleh *Product Owner* dari Asisten Manajer divisi Perencanaan dan Transaksi Energi PLN UP3 Yogyakarta.

b. *Product Backlog*

Product Backlog berisi fitur dan kebutuhan aplikasi SIBER dengan tingkat prioritas masing-masing. *Backlog* ini dibagi menjadi dua kelompok berdasarkan target pengembangan *dashboard*: *dashboard* AMR dan *dashboard* Kinerja. Tabel 3. 1 dan Tabel 3. 2 menunjukkan detail *Product Backlog* untuk masing-masing *dashboard*.

Tabel 3. 1 *Product Backlog Dashboard AMR*

No.	Fitur	Deskripsi	Waktu	Prioritas
1.	<i>Navbar dashboard</i>	Navigasi untuk mengakses berbagai halaman dalam <i>dashboard</i>	3 hari	Tinggi
2.	<i>Scorecard</i>	Tampilan yang menunjukkan jumlah pelanggan, daya serta tegangan perlu perbaikan dan perhatian	7 hari	Menengah
3.	Peta pelanggan	Peta yang berisi lokasi pelanggan, daya, dan tegangan terpakai	14 hari	Tinggi
4.	Data tabel pelanggan	Tabel yang berisi semua data pelanggan	7 hari	Tinggi
5.	<i>Pie chart</i>	Grafik yang berisi jumlah tiap kota, jenis tegangan, dan statusnya	14 hari	Tinggi
6.	Filterisasi ke <i>scorecard</i> , peta, dan diagram lingkaran	Filter data pada <i>scorecard</i> , peta, dan diagram	14 hari	Menengah

Tabel 3. 2 *Product Backlog Dashboard Kinerja*

No.	Fitur	Deskripsi	Waktu	Prioritas
1.	Pembagian menjadi 4 halaman: Pelanggan & Daya; <i>Susut</i> , KWH Jual, KWH Produksi; <i>Saidi</i> , <i>Saifi</i> , <i>Recovery time</i> , <i>GGN 100 Kms</i> , <i>GGN Reclosure</i> ; Hpl	Pembagian <i>dashboard</i> kinerja menjadi 4 halaman sesuai dengan fungsionalitasnya masing-masing. Pelanggan adalah individu yang menggunakan listrik dari penyedia listrik; daya adalah ukuran jumlah energi listrik yang digunakan atau dibutuhkan dalam suatu sistem dalam periode tertentu; <i>susut</i> adalah energi listrik yang hilang selama proses distribusi dan transmisi oleh resistansi dalam kabel dan peralatan listrik; kwh jual adalah jumlah energi listrik yang dijual kepada pelanggan; kwh produksi adalah jumlah energi listrik yang dihasilkan oleh pembangkit listrik; <i>saidi</i> adalah durasi padam yang dialami oleh pelanggan dalam satu tahun; <i>saifi</i> adalah frekuensi padam yang dialami oleh pelanggan dalam satu tahun; <i>recovery time</i> adalah waktu pemulihan layanan listrik setelah terjadi gangguan; <i>ggn 100 kms</i> adalah gangguan pada trafo yang terjadi sepanjang jaringan distribusi sejauh 100 kms; <i>ggn reclosure</i> adalah gangguan pada pembatas setiap bagian atau zona dalam jaringan distribusi listrik.; hpl atau halaman perluasan adalah halaman yang menyediakan informasi jumlah penambahan pemasangan trafo, baik pemasangan baru maupun tiang yang memerlukan penambahan daya.	1 hari	Tinggi
2.	<i>Navbar dashboard</i> perhalaman	Navigasi untuk setiap halaman yang ada pada <i>dashboard</i>	3 hari	Tinggi
3.	<i>Scorecard</i> perhalaman	Tampilan yang menunjukkan total data statistik dari tiap chart per tahunnya sesuai dengan filter	5 hari	Menengah
4.	Diagram batang untuk halaman pelanggan & daya, diagram garis untuk halaman sisanya	Grafik yang berisi data tertentu di setiap halamannya	35 hari	Tinggi
5.	Filterisasi ke <i>scorecard</i> dan tahun	Menyaring data berdasarkan tahun	14 hari	Menengah

Pada Tabel 3. 1 dan Tabel 3. 2, terdapat fitur dan kebutuhan yang diperlukan untuk mengembangkan aplikasi SIBER dan tingkat prioritas dari masing-masing fitur. *Product Backlog* dibagi menjadi dua bagian sesuai dengan *dashboard* yang dikembangkan untuk memenuhi kebutuhan masing-masing divisi. *Dashboard AMR* dari divisi Transaksi Energi difokuskan pada pemantauan daya serta tegangan listrik yang dipakai pelanggan,

sedangkan *dashboard* Kinerja dirancang untuk divisi Perencanaan dalam melakukan monitoring serta perawatan terhadap infrastruktur kelistrikan.

Prioritas pada *Product Backlog* ditentukan berdasarkan fungsionalitas paling penting dari setiap fitur pada masing-masing *dashboard*, seperti yang tercantum dalam Tabel 3. 1 dan Tabel 3. 2. Penetapan prioritas dilakukan oleh *Product Owner* dan dibahas lebih lanjut oleh tim dalam rapat.

Tabel 3. 2, pembagian *dashboard* menjadi empat halaman diperlukan untuk menampilkan berbagai diagram atau visualisasi data yang berbeda sesuai kategori spesifik. Halaman pertama menampilkan data pelanggan dan penggunaan daya, halaman kedua menampilkan energi yang hilang dan kwh jual dan produksinya, halaman ketiga menampilkan durasi padam dan waktu pemulihan, dan halaman keempat akan fokus kepada penambahan tiang trafo listrik per pelanggan.

Estimasi waktu pada Tabel 3. 1 dan Tabel 3. 2 ditentukan berdasarkan pengalaman tim pengembang, percobaan akan *framework* yang digunakan, serta perkiraan waktu yang dibutuhkan untuk mengatasi eror atau kesalahan yang terjadi selama pengujian aplikasi.

c. *Sprint Planning*

Tabel 3. 3 *Sprint Planning I Dashboard AMR*

<i>Sprint</i>	Fitur	Waktu	Prioritas
<i>Sprint 1</i>	<i>Navbar dashboard, scorecard</i>	10 hari	Tinggi, menengah
<i>Sprint 2</i>	Peta pelanggan, data tabel pelanggan	21 hari	Tinggi, tinggi
<i>Sprint 3</i>	<i>Pie chart, filterisasi</i>	28 hari	Tinggi, menengah

Tabel 3. 4 *Sprint Planning II Dashboard Kinerja*

<i>Sprint</i>	Fitur	Waktu	Prioritas
<i>Sprint 1</i>	Pembagian halaman, <i>navbar dashboard</i>	4 hari	Tinggi, tinggi
<i>Sprint 2</i>	<i>Scorecard</i> perhalaman, filterisasi	19 hari	Menengah, menengah
<i>Sprint 3</i>	Diagram perhalaman	35 hari	Tinggi

Tabel 3. 3 dan Tabel 3. 4, terdapat fitur dan kebutuhan yang diperlukan untuk mengembangkan aplikasi SIBER beserta tingkat prioritas masing-masing fitur. *Product Backlog* dibagi menjadi dua bagian, masing-masing untuk *dashboard* AMR dan Kinerja, sesuai dengan kebutuhan dan fungsionalitas dari setiap *dashboard*. *Dashboard* AMR dirancang untuk memenuhi kebutuhan divisi Transaksi Energi dalam hal pemantauan daya

serta tegangan listrik yang digunakan pelanggan. Sementara itu, *dashboard* Kinerja difokuskan untuk divisi Perencanaan dalam melakukan monitoring dan perawatan terhadap infrastruktur kelistrikan yang ada.

d. *Sprint*

Tabel 3. 5 *Sprint Dashboard AMR*

<i>Sprint</i>	Fitur	Waktu	Prioritas
<i>Sprint 1</i>	<i>Navbar dashboard</i>	3 hari	Tinggi
	<i>Scorecard</i>	7 hari	Menengah
<i>Sprint 2</i>	Peta pelanggan	14 hari	Tinggi
	Data tabel pelanggan	7 hari	Tinggi
<i>Sprint 3</i>	<i>Pie chart</i>	14 hari	Tinggi
	Filterisasi	14 hari	Menengah

Tabel 3. 6 *Sprint Dashboard Kinerja*

<i>Sprint</i>	Fitur	Waktu	Prioritas
<i>Sprint 1</i>	Pembagian halaman	1 hari	Tinggi
	<i>Navbar dashboard</i>	3 hari	Tinggi
<i>Sprint 2</i>	<i>Scorecard</i> perhalaman	5 hari	Menengah
	Filterisasi	14 hari	Menengah
<i>Sprint 3</i>	Diagram perhalaman	35 hari	Tinggi

Tabel 3. 5 dan Tabel 3. 6, berisi pelaksanaan *Sprint* yang dilakukan sesuai dengan *Sprint Planning*. Tim mulai mengembangkan dan menguji aplikasi sesuai dengan fitur dan target waktu yang telah ditentukan. Setiap *dashboard* terdiri dari tiga kali *Sprint* untuk membagi pekerjaan menjadi beberapa bagian yang lebih terkelola dan dapat dicapai dalam jangka waktu tertentu. Waktu dan prioritas ditetapkan berdasarkan pencatatan *Product Backlog* dan rencana yang telah disusun bersama tim di *Sprint Planning*.

e. *Daily Sprint*

Pertemuan harian biasanya diadakan di pagi hari bersama dengan *Product Owner*, *Scrum Master*, dan tim pengembang untuk berkomunikasi. Pertemuan ini juga memberi kesempatan kepada tim pengembang untuk berbagi informasi tentang kemajuan pekerjaan, serta mengidentifikasi hambatan yang dialami. Beberapa kemajuan pekerjaan yang dilaporkan meliputi keberhasilan dalam mengatasi kendala akses sumber daya server,

mencoba *framework* yang dipilih untuk pengembangan, dan mengatasi *bug* di setiap tampilan yang ada.

f. *Increment*

Tabel 3. 7 *Increment Dashboard AMR*

No.	Fitur	Hasil
1.	<i>Navbar dashboard</i>	Sukses
2.	<i>Scorecard</i>	Sukses
3.	Peta pelanggan,	Sukses
4.	Data tabel pelanggan	Sukses
5.	<i>Pie chart,</i>	Sukses
6.	Filterisasi ke <i>scorecard</i> , peta, dan diagram	Sukses

Tabel 3. 8 *Increment Dashboard Kinerja*

No.	Fitur	Hasil
1.	Pembagian menjadi 4 halaman: Pelanggan & Daya; <i>Susut</i> , KWH Jual, KWH Produksi; <i>Saidi</i> , <i>Saifi</i> , <i>Recovey time</i> , <i>GGN 100 Kms</i> , <i>GGN Reclosure</i> ; Hpl	Sukses
2.	<i>Navbar dashboard</i> perhalaman	Sukses
3.	<i>Scorecard</i> perhalaman	Sukses
4.	Diagram perhalaman	Sukses
5.	Filterisasi ke <i>scorecard</i> dan tahun	Sukses

Tabel 3. 7 dan Tabel 3. 8, terlihat hasil dari setiap *Sprint*. Hasil setiap fitur dari *dashboard* kemudian akan diuji kualitas dan fungsionalitasnya bersama dengan *Product Owner* dan *Project Leader*. Tim juga akan mencatat fitur yang telah dikembangkan untuk selanjutnya dipresentasikan saat rapat bersama dengan *Product Owner* dan *Project Leader*.

Kriteria sukses pada Tabel 3. 7 dan Tabel 3. 8 ditentukan berdasarkan fitur yang telah memenuhi kebutuhan dan keinginan sesuai dengan *Product Backlog*. Validasi dan pengujian bersama dengan *Product Owner* dan *Project Leader* akan memastikan bahwa fitur tersebut telah sesuai dan bebas dari masalah.

g. *Sprint Review*

Pada tahap *Sprint Review*, dilakukan oleh tim pengembang, *Product Owner*, dan *Project Leader* di akhir *Sprint* dengan pemilihan hari Jumat. Pemilihan waktu di hari Jumat karena biasanya merupakan hari yang lebih santai bagi karyawan. Pada tahap ini, *Product Backlog* yang telah diselesaikan disampaikan dan ditinjau kembali oleh *Product Owner*. Hasil dari tinjauan akan berupa umpan balik dan masukan untuk penyesuaian pada *Sprint*

berikutnya. Selain melakukan tinjauan, juga dilakukan pengujian unit untuk setiap fitur yang telah diselesaikan setelah *Sprint* berakhir. Hasil tinjauan dan pengujian dari Asisten Manajer Perencanaan dan Transaksi Energi PLN UP3 Yogyakarta adalah sebagai berikut:

1. Semua fitur telah berfungsi dengan baik
2. Tiap *dashboard* dapat berjalan secara *real-time* yang diintegrasikan ke *database*
3. Desain *dashboard* yang digunakan sangat jelas dan informatif

h. *Sprint Retrospective*

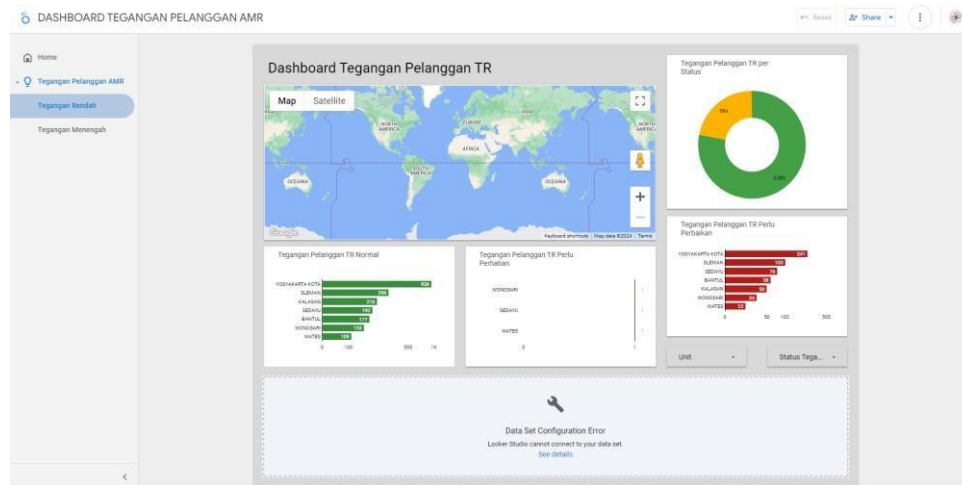
Setiap tim menyampaikan pendapat dan evaluasi terkait kinerja selama menerapkan metode *Scrum*. Tujuan dari pelaksanaan tahap ini adalah untuk mengidentifikasi hal-hal yang berjalan dengan baik serta aspek yang perlu diperbaiki. Diskusi mencakup tantangan yang dihadapi selama *Sprint* dan ide untuk meningkatkan proses *Sprint* berikutnya. Tahap ini dilakukan setiap akhir *Sprint Review*, memastikan bahwa pembelajaran dari *Sprint* sebelumnya dapat langsung diterapkan pada siklus berikutnya. Hasil dari *Sprint Retrospective* menunjukkan bahwa meskipun semua fitur yang direncanakan berhasil diselesaikan, namun terdapat masalah selama pelaksanaannya. Salah satunya adalah kurangnya efektivitas kolaborasi tim yang menyebabkan beberapa tugas tertunda. Masalah ini dapat diatasi dengan meningkatkan komunikasi, transparansi, dan kejelasan tugas. *Sprint Retrospective* dilakukan bersama dengan *Product Owner* dan *Project Leader*.

3.4 Pelaksanaan dan Hasil Proyek

Hasil akhir perancangan *dashboard* dalam penulisan ini akan menampilkan informasi berupa visualisasi data, yang terdiri dari *pie chart*, *bar chart*, dan map yang dikembangkan dengan metode pengembangan *Scrum*. Sebelum pelaksanaan, penulis dan tim melakukan observasi tampilan yang akan digunakan sebagai pedoman dalam pembuatan aplikasi SIBER. Tampilan contoh aplikasi sebelum proyek SIBER dapat dilihat pada

Gambar 3. 1. *Dashboard* tersebut, yang dikembangkan tanpa menggunakan metode pengembangan terstruktur, mengalami kendala berupa penundaan proyek terkait perubahan kebutuhan dan prioritas. Hal ini disebabkan oleh keterbatasan dari *dashboard* yang bersifat statis dan tidak dinamis, yang menyulitkan adaptasi terhadap perubahan dan menghambat

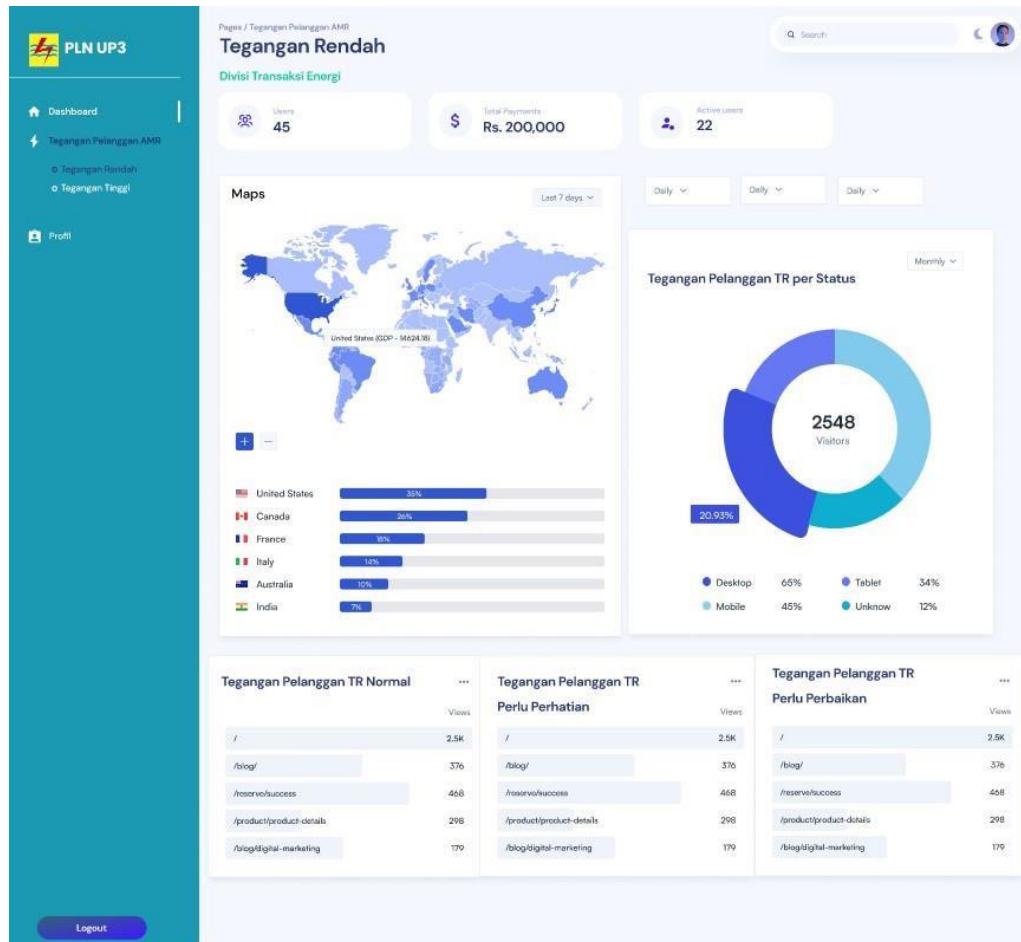
pengembangan yang efisien. Proyek SIBER diharapkan dapat mengatasi masalah ini dengan pendekatan pengembangan yang lebih terstruktur dan fleksibel.



Gambar 3. 1 Contoh Aplikasi Sebelum SIBER

Untuk membuat sebuah aplikasi, pengembang memerlukan visualisasi konsep desain atau *prototipe* untuk mempermudah pengembangan. *Prototipe* tersebut dibuat menggunakan aplikasi Figma, yang merupakan alat untuk merancang antarmuka pengguna, membuat *prototipe* secara interaktif, serta berkolaborasi secara *real-time* dengan tim. Rancangan desain sistem yang dibuat dapat dilihat pada

Gambar 3. 2.



Gambar 3. 2 Tampilan Prototipe *Dashboard* Menggunakan Figma

3.4.1 Proyek *Dashboard* AMR

a. *Sprint* 1

Sprint 1 telah berhasil diselesaikan dengan pencapaian berupa pengembangan beberapa fitur, yaitu *navbar* sebagai elemen navigasi utama dan *scorecard* untuk menyajikan data secara ringkas dan informatif.

1. Membuat *Navbar Dashboard*

Pada tahap ini, tim berhasil membuat *navbar* dan menangani *bug* minor yang ditemukan. Pertemuan *Daily Sprint* telah dilaksanakan untuk melaporkan kemajuan secara rutin dan memastikan tampilan yang konsisten di semua halaman. Hasil produk berupa *navbar* responsif yang terdiri atas logo perusahaan, nama aplikasi, *button dashboard*, dan *button* keluar. *Sprint Review* telah dilakukan, menghasilkan umpan balik positif terkait desain dan fungsionalitas *navbar*. Pengujian unit juga dilakukan pada tahap ini, dengan usulan dari *Product Owner* untuk mengubah jenis *navbar* yang awalnya dirancang saat pembuatan *prototipe*. Desain awal yang telah dibuat pada tahap

perancangan dirasa kurang konsisten dengan tampilan web PLN yang digunakan oleh perusahaan, sehingga perlu dilakukan penyesuaian agar navbar terlihat lebih seragam dengan branding dan gaya visual perusahaan termasuk warna, tata letak, dan elemen visual lainnya. Masukan ini diterima dengan baik oleh tim, dan perusahaan pada desain navbar segera dilakukan agar lebih sesuai dengan ekspektasi pengguna. Selain itu, *Sprint Retrospective* juga telah dilaksanakan, menghasilkan umpan balik positif dari *Product Owner* dan *Project Leader*. Masukan tersebut berupa saran untuk meningkatkan kolaborasi dalam tim dan menjaga komunikasi yang baik antar tim. Selain itu, pengerjaan fitur *navbar* bisa berjalan lebih cepat dari estimasi, dari 3 hari menjadi 2 hari.

2. Membuat *Scorecard*

Scorecard telah berhasil dikembangkan untuk menampilkan data secara ringkas yang ditampilkan di peta, data tabel, dan diagram. Selama *Sprint*, *Daily Sprint* dilakukan untuk membahas tantangan dalam pengolahan data dan visualisasi, terutama terkait penyesuaian tampilan *scorecard*, sehingga diskusi ini memungkinkan tim segera memperbaiki kesalahan. Hasilnya, *scorecard* mampu menampilkan data statistik yang akurat mengenai jumlah pelanggan, daya, dan tegangan secara *real-time* dengan desain sesuai arahan *Product Owner*. Pada *Sprint Review*, *Product Owner* memberikan umpan balik positif, terutama terkait tata letak dan pemilihan warna. Pengujian unit juga dilakukan untuk memastikan data sinkron dengan *database*, diagram, peta, dan data tabel. Pada *Sprint Retrospective*, tim menyadari estimasi waktu pengembangan terlampaui dari perencanaan awal. Estimasi waktu yang ditetapkan selama *Sprint Planning* adalah 7 hari, tetapi karena kompleksitas data yang lebih tinggi dari perkiraan dan proses sinkronisasi data dengan elemen lainnya, pengembangan sebenarnya memakan waktu hingga 8 hari.

b. *Sprint 2*

Sprint 2 telah berhasil diselesaikan dengan pencapaian pengembangan dua fitur, yaitu peta pelanggan untuk menampilkan data lokasi secara interaktif dan data tabel yang menyajikan informasi pelanggan dalam format terstruktur.

1. Membuat Peta Pelanggan

Fitur peta pelanggan berhasil dikembangkan dengan menampilkan data lokasi pelanggan, daya, dan tegangan secara *real-time*. Pada *Daily Sprint*, tim fokus membahas progres implementasi peta, bertukar ide mengenai API yang akan digunakan, serta berkolaborasi dengan tim *backend* untuk memperbaiki masalah sinkronisasi data yang menyebabkan penundaan pembaruan peta. Diskusi juga mencakup penanganan eror dalam memuat data besar dan strategi untuk mengoptimalkan interaktivitas peta. Hasilnya, peta pelanggan yang dirancang dengan penanda interaktif, lengkap dengan *tooltip* yang memberikan informasi detail seperti ID pelanggan, nama pelanggan, koordinat lokasi, status tegangan, daya dan voltase. Tim juga menambahkan ikon bergerak untuk mengurangi kebosanan saat pengguna menunggu pemrosesan data yang dapat dilihat pada Gambar 3. 3. Pada *Sprint Review*, dilakukan juga pengujian unit yang menunjukkan peta pelanggan berfungsi dengan baik, dengan umpan balik positif dari *Product Owner* mengenai interaktivitas dan tampilannya. Namun, terdapat saran untuk menambahkan informasi pada *tooltip* dan memperbaiki fitur zoom agar navigasi lebih baik. Dalam *Sprint Retrospective*, tim menyadari tantangan utama adalah menemukan API yang tepat dan menangani ketidakakuratan beberapa data lokasi, yang menyebabkan perpanjangan waktu pengembangan dari 14 hari menjadi 15 hari.



Gambar 3. 3 Ikon Tambahan

2. Membuat Data Tabel Pelanggan

Fitur data tabel berhasil dikembangkan untuk menampilkan semua informasi terkait pelanggan dalam format terstruktur. Pada *Daily Sprint*, diskusi berfokus pada perbaikan bug yang ditemukan selama pengujian tabel, seperti masalah pada fitur *sorting* yang tidak berfungsi pada kolom tertentu. Tim juga membahas fitur impor data agar kompatibel dengan versi PHP server yang digunakan perusahaan, serta penyesuaian desain tabel untuk memastikan informasi yang ditampilkan mudah dipahami oleh

pengguna. Hasil *Increment* menunjukkan tabel yang mencakup kolom untuk ID pelanggan, nama, unit, daya, *read date*, tegangan, voltase, status, dan *button action*. Pada kolom *action*, terdapat beberapa fungsi, seperti melihat lokasi pelanggan di peta, mengedit data pelanggan, dan menghapus data pelanggan. Data tabel mendukung ekspor dalam format CSV, PDF, dan salin ke *clipboard*, serta memungkinkan impor file dalam format *excel* yang kompatibel dengan versi server PHP perusahaan. Pada *Sprint Review*, *Product Owner* memberikan saran untuk menambahkan kolom *action* yang dapat melakukan edit, hapus, dan melihat lokasi pelangga, serta menyarankan agar posisi data tabel dipindahkan di bawah peta, di samping kiri grafik. Pengujian unit juga dilakukan pada tahap ini, untuk memastikan data yang tertampil sinkron dengan semua elemen dan *button action* dapat digunakan. Dalam *Sprint Retrospective*, tim mencatat bahwa pengembangan tabel memerlukan waktu lebih lama dari yang direncanakan karena kompleksitas dalam menyesuaikan format dan sinkronisasi antara tabel, peta, dan grafik, dengan estimasi waktu pengembangan awalnya 7 hari, berlangsung selama 8 hari.

c. *Sprint 3*

Sprint 3 telah berhasil diselesaikan dengan pencapaian pengembangan beberapa fitur, yaitu *pie chart* untuk menampilkan distribusi data secara visual dan filterisasi yang memungkinkan pengguna menyaring informasi berdasarkan kriteria tertentu.

1. Membuat *Pie Chart*

Fitur *pie chart* berhasil dikembangkan untuk menampilkan distribusi jumlah pelanggan berdasarkan kategori kota dan status tegangan. Selama pelaksanaan *Daily Sprint*, tim menangani perubahan desain yang telah dibuat, di mana desain awal menggunakan *bar chart* untuk menampilkan masing-masing status tegangan, namun kemudian diubah menjadi satu berbentuk *pie chart*. Selain itu, ditambahkan *tooltip* untuk memberikan kejelasan informasi grafik. Hasil *Increment* menunjukkan bahwa diagram lingkaran ini dirancang dengan interaktivitas, memungkinkan pengguna untuk melihat detail tambahan melalui *tooltip*. Pengujian dilakukan untuk memastikan akurasi data dan memastikan bahwa filter dapat digunakan dengan baik. Pada *Sprint Review*, *product owner* memberikan umpan balik positif mengenai kejelasan dan fungsionalitas *pie chart*, serta meminta agar posisi grafik dipindahkan di bawah peta dan di samping data tabel. Selain itu, pengujian unit dilakukan untuk memastikan bahwa semua fungsi

berjalan sesuai yang diharapkan. Dalam *Sprint Retrospective*, tim mencatat bahwa pengembangan dapat berjalan lebih cepat dari estimasi waktu 14 hari menjadi 10 hari, karena pengetahuan beberapa anggota tim mengenai *framework* yang digunakan serta pengalaman dari proyek pengembangan sebelumnya. Tantangan yang dihadapi termasuk dalam kolaborasi dan komunikasi yang berkurang antar tim, sehingga tim sepakat untuk memperbaiki dan meningkatkan koordinasi dalam tim.

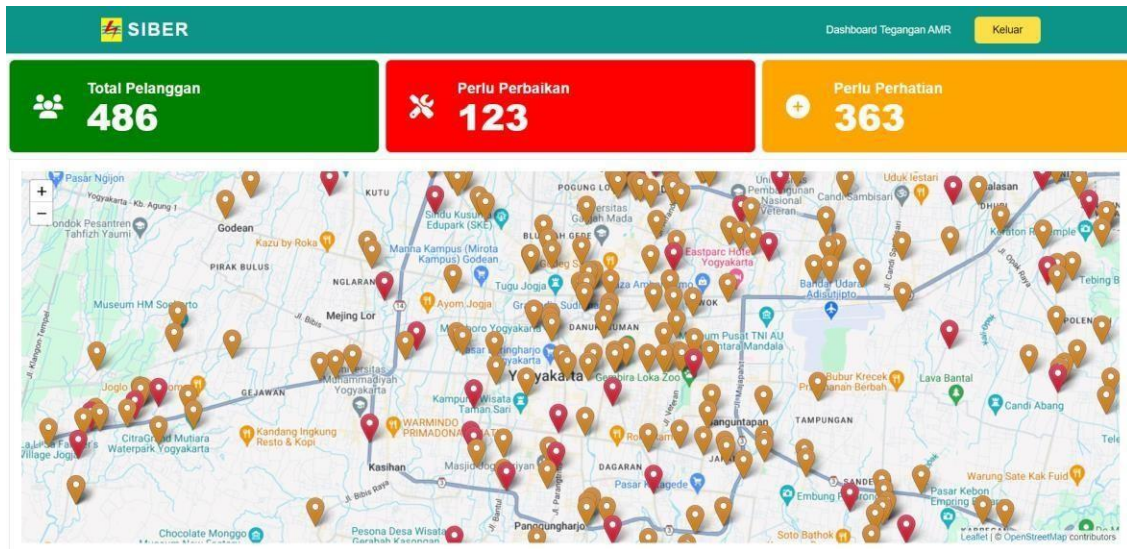
2. Membuat Filterisasi

Fitur filterisasi berhasil dikembangkan untuk memungkinkan pengguna menyaring data pada *scorecard*, peta, dan *pie chart* berdasarkan berbagai kriteria, seperti kota, jenis tegangan, dan status tegangan. Diskusi selama *Daily Sprint* berfokus pada solusi untuk *bug* yang ditemukan selama pengujian, terutama terkait ketidakakuratan data yang disaring. Hasilnya menunjukkan bahwa filterisasi ini mencakup *dropdown* dan *checkbox*, memudahkan pengguna dalam memilih parameter yang diinginkan. Uji coba dilakukan untuk memastikan bahwa filter bekerja secara efektif pada semua komponen *dashboard* dan hasilnya akurat. Pada *Sprint Review*, *Product Owner* memberikan umpan positif mengenai fungsionalitas fitur dan menyarankan agar kota pada filter lokasi diurutkan menjadi Kalasan, Wates, Bantul, Sedayu, Wonosari, Sleman, dan Yogyakarta Kota. Selain itu, pengujian unit dilakukan untuk memastikan bahwa semua fungsi filter berjalan sesuai yang diharapkan. Dalam *Sprint Retrospective*, tim mencatat bahwa pengembangan fitur berjalan lebih cepat dari perkiraan, dengan estimasi waktu 14 hari dan realisasi selama 10 hari. Kecepatan ini dicapai berkat efisiensi dalam pengelolaan data dan komunikasi efektif dalam tim selama pengembangan. Tim sepakat untuk mempertahankan strategi ini dalam pengembangan fitur mendatang dan memanfaatkan waktu yang lebih cepat untuk meningkatkan kualitas.

3.4.2 Hasil Proyek *Dashboard* AMR

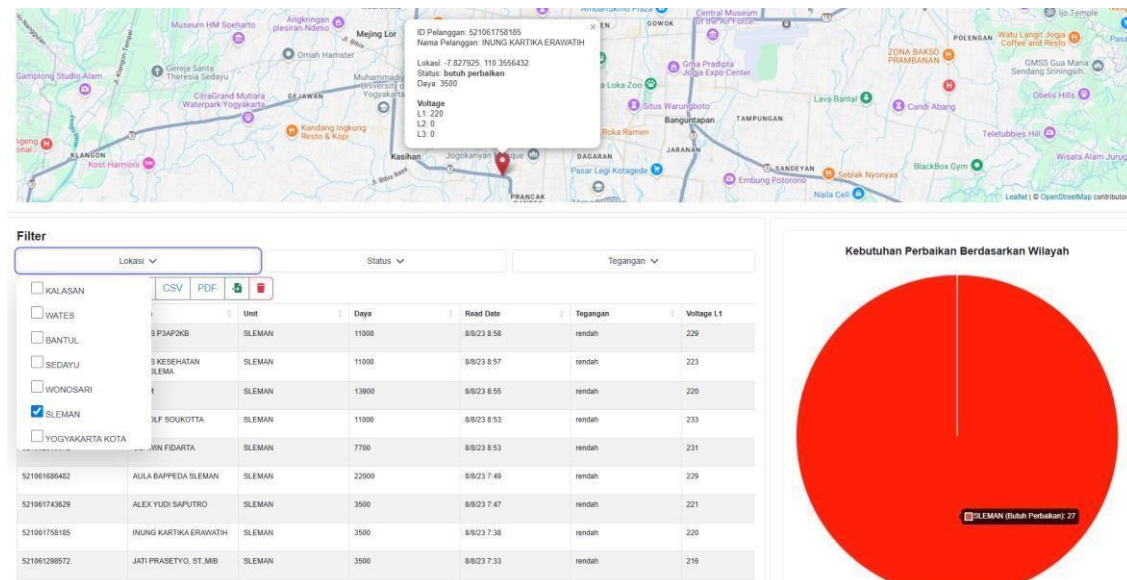
Hasil akhir dari proyek *dashboard* AMR adalah berupa *dashboard* yang berisi *scorecard*, peta, data tabel, dan diagram lingkaran. Pada Gambar 3. 4, pengguna dapat melihat *scorecard* data total pelanggan, perlu perbaikan, dan perlu perhatian. Peta

interaktif yang disajikan berisi lokasi pelanggan, daya, dan tegangan terpakai.



Gambar 3. 4 Hasil Proyek Dashboard AMR 1

Gambar 3. 5 pengguna dapat melakukan hapus, edit, dan lihat lokasi spesifik pelanggan dibagian data tabel. Selain itu, pengguna juga dapat melakukan filterisasi data yang dibutuhkan dan melakukan impor dan ekspor file excel. Pengguna juga dapat melihat jumlah tiap kota, jenis tegangan, dan statusnya di grafik lingkaran.



Gambar 3. 5 Hasil Proyek Dashboard AMR 2

3.4.3 Proyek *Dashboard Kinerja*

a. *Sprint 1*

Sprint 1 telah berhasil diselesaikan dengan pencapaian pengembangan beberapa fitur, yaitu melakukan pembagian halaman untuk mengorganisasi tampilan data dan *navbar* yang responsif.

1. Melakukan Pembagian Halaman

Pembagian halaman berhasil dilakukan dengan memisahkan informasi *dashboard Kinerja* ke dalam beberapa halaman spesifik, yaitu halaman *saidi*, *saiifi*, *recovery time*, GGN 100 km/s, dan GGN *reclosure*; serta halaman HPL. Aktivitas harian tim selama *Daily Sprint* berfokus pada pengujian antar halaman, memperbaiki bug navigasi, dan memastikan bahwa data ditampilkan dengan akurat di setiap halaman. Hasil *Increment* menunjukkan pembagian ini memudahkan pengguna dalam mengakses data spesifik sesuai kategori dan meningkatkan pengalaman navigasi. Pada *Sprint Review*, tim menerima umpan balik dari *Product Owner* untuk menambahkan halaman *home* sebagai titik awal navigasi pengguna, dan pengujian unit dilakukan untuk memastikan semua halaman berfungsi dengan baik. Dalam *Sprint Retrospective*, tim mengidentifikasi tantangan utama dalam menjaga konsistensi data antar halaman serta memastikan performa halaman tetap optimal meskipun memuat data yang besar.

2. Membuat *Navbar Dashboard* Perhalaman

Fitur *navbar* per halaman berhasil dikembangkan dengan desain yang konsisten dengan *dashboard AMR*, memastikan keseragaman tampilan dan pengalaman pengguna di semua halaman *dashboard Kinerja*. Aktivitas *Daily Sprint* difokuskan pada pengujian tampilan *navbar* dan konsistensi dalam elemen desain untuk menjaga estetika dan fungsionalitas di setiap halaman. Hasil *Increment* menunjukkan tim berhasil mengimplementasikan *navbar* per halaman dengan mengadopsi desain yang sudah ada, seperti warna, ikon, dan tata letak, proses navigasi antar halaman menjadi lebih familiar bagi pengguna yang sebelumnya sudah terbiasa dengan *dashboard AMR*. Pada *Sprint Review*, umpan balik dari *Product Owner* menunjukkan bahwa penggunaan desain *navbar* terbukti berfungsi dengan baik di setiap halaman, memastikan navigasi yang konsisten dan mudah digunakan. Selain itu, pengujian unit dilakukan untuk memastikan bahwa setiap elemen *navbar* beroperasi dengan baik. Dalam *Sprint Retrospective*, tim mencatat bahwa penggunaan desain yang sudah ada mempercepat

proses pengembangan dan mengurangi waktu yang dihabiskan untuk iterasi desain baru.

b. *Sprint 2*

Sprint 2 telah berhasil diselesaikan dengan pencapaian pengembangan beberapa fitur, yaitu *scorecard* untuk menampilkan data secara ringkas dan filterisasi yang memungkinkan pengguna menyaring data berdasarkan kriteria tertentu.

1. Membuat *Scorecard* Perhalaman

Fitur *scorecard* dengan desain yang seragam seperti *dashboard* AMR berhasil dikembangkan dan ditambahkan ke setiap halaman *dashboard* Kinerja, menggunakan elemen visual yang konsisten, termasuk ikon dan tata letak. Selama *Daily Sprint*, fokus tim adalah mengintegrasikan *scorecard* dengan data yang dinamis serta memastikan tampilan tetap konsisten dan responsif. Tim juga melakukan pengujian mandiri untuk memastikan bahwa *scorecard* berfungsi dengan baik di seluruh halaman dan mencocokkan data secara akurat dengan informasi yang tersedia. Hasil *Increment* berupa *scorecard* yang telah mencakup tampilan data yang *real-time*, serta berfungsi optimal dalam menampilkan informasi penting secara jelas dan mudah dipahami oleh pengguna di setiap halaman. Selama *Sprint Review*, *Product Owner* memberikan rekomendasi untuk menyesuaikan warna *scorecard* tertentu agar lebih menonjol dan mudah dikenali oleh pengguna. Pengujian unit juga dilakukan untuk memastikan bahwa informasi yang disajikan telah sesuai dan sinkron antara *database* dan diagram. Dalam *Sprint Retrospective*, tim mencatat bahwa penggunaan desain yang sama dapat mempercepat proses pengembangan, meskipun tantangan muncul dalam menjaga performa *scorecard* saat mengelola data dalam jumlah besar, yang berhasil diatasi melalui penyesuaian kode dan query data agar lebih efisien.

2. Membuat Filterisasi

Fitur filterisasi berdasarkan tahun berhasil dikembangkan dan diimplementasikan pada *dashboard* Kinerja, memungkinkan pengguna untuk memilih rentang tahun dari 2015 hingga 2020 guna melihat data yang relevan. Aktivitas *Daily Sprint* difokuskan pada pengujian fungsionalitas filterisasi ini agar berjalan dengan baik dan memastikan bahwa data yang ditampilkan sesuai dengan pilihan tahun yang dipilih oleh pengguna. Desain filterisasi disesuaikan dengan *dashboard* AMR untuk menjaga konsistensi

antarmuka pengguna. Hasil *Increment* mencakup filterisasi yang efektif untuk menyaring data sesuai dengan rentang tahun yang dipilih, memberikan kemudahan bagi pengguna dalam menganalisis data historis dengan cepat dan tepat. Selama *Sprint Review*, *Product Owner* memberikan umpan balik positif mengenai fungsionalitas filter ini, serta dilakukan pengujian unit untuk memastikan bahwa filterisasi bekerja secara akurat di semua rentang tahun. Pada *Sprint Retrospective*, tim mencatat bahwa menggunakan desain filterisasi yang sama seperti *dashboard* AMR berhasil mempercepat proses pengembangan dari estimasi awal 14 hari menjadi 5 hari, meskipun tantangan utama adalah memastikan filter bekerja dengan baik untuk data dari tahun 2015 hingga 2020.

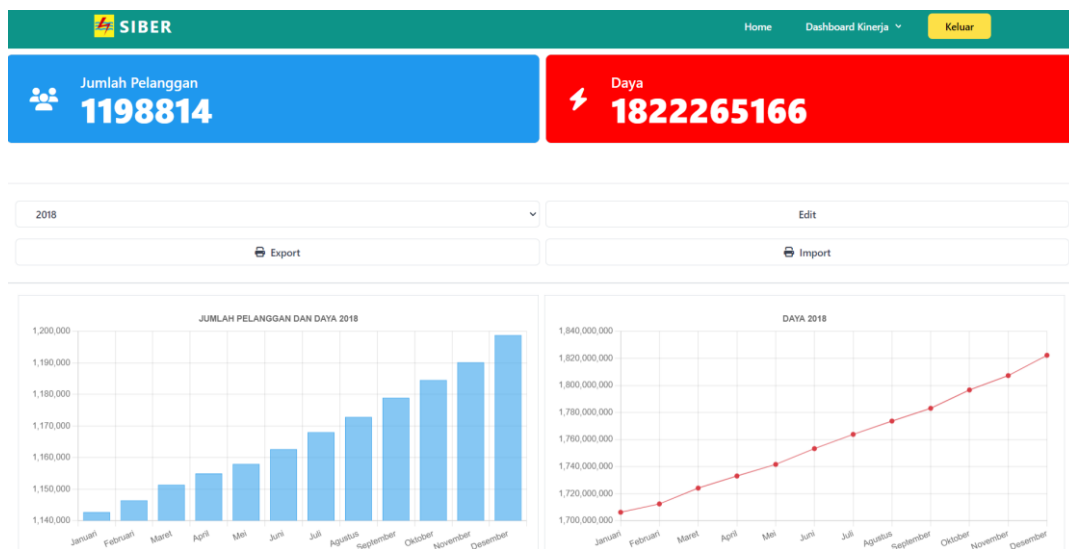
c. *Sprint 3*

Sprint 3 telah berhasil dilaksanakan dengan fokus pada pengembangan dua fitur, yaitu diagram batang dan garis yang dirancang untuk menyajikan data secara visual dan interaktif. Pada fitur diagram batang dan garis berhasil dikembangkan dan diimplementasikan pada *dashboard* Kinerja, memberikan visualisasi yang jelas dan mendalam tentang data yang ditampilkan. Diagram batang digunakan untuk menampilkan data pelanggan di halaman pelanggan dan daya, sedangkan diagram garis digunakan untuk menampilkan data daya; *susut*, KWH jual, dan KWH produksi; *saidi*, *saifi*, *recovery time*, GGN 100 km/s, dan GGN *reclosure*; dan HPL. Aktivitas *Daily Sprint* difokuskan pada pengujian integrasi data dengan diagram serta memantau umpan balik untuk memastikan grafik berfungsi dengan baik dan memberikan informasi yang bermanfaat. Hasil *Increment* berupa implementasi diagram batang dan garis yang terintegrasi dengan filterisasi tahun dan menggunakan warna konsisten dengan *scorecard*, membantu pengguna menganalisis data dengan lebih baik. Selama *Sprint Review*, *Product Owner* memberikan umpan balik positif namun menyarankan penyesuaian warna diagram agar selaras dengan *scorecard* dan menambahkan *tooltip* untuk memperjelas informasi yang ada pada diagram. Pengujian unit dilakukan selama *Sprint Review* untuk memastikan bahwa data pada diagram ditampilkan dengan akurat sesuai dengan filter yang digunakan. Pada *Sprint Retrospective*, tim mencatat bahwa pengembangan berjalan lebih cepat dari perkiraan, dengan estimasi awal 35 hari menjadi 28 hari pengembangan karena pengalaman beberapa anggota tim dengan *framework* yang digunakan. Tantangan utama yang dihadapi adalah menjaga

akurasi data pada grafik, terutama ketika memproses data dalam jumlah besar dan memastikan bahwa setiap elemen grafik mempresentasikan data yang tepat.

3.4.4 Hasil Proyek *Dashboard Kinerja*

Hasil akhir dari proyek *dashboard Kinerja* adalah berupa *dashboard* yang berisi diagram garis dan batang. Pada Gambar 3. 6, pengguna dapat memilih opsi *navbar*, yaitu yaitu halaman pelanggan & daya; *susut*, kwh jual, kwh produksi; *saidi*, *saifi*, *recovery time*, *GGN 100 Kms*, *GGN reclosure*; *hpl*. *Scorecard* yang tertampil juga menyesuaikan dengan halaman yang sedang dipilih. Selain itu, pengguna dapat menghapus, edit, dan memilih tahun yang tersedia. Pengguna juga dapat memiliki akses untuk mengimpor dan ekspor file dalam format *excel*.



Gambar 3. 6 Hasil Proyek *Dashboard Kinerja*

3.5 Pengujian

Pengujian adalah tahap krusial dalam pengembangan aplikasi untuk memastikan bahwa setiap fitur berjalan sesuai dengan ekspektasi dan tujuan yang telah ditetapkan. Setelah pengujian unit yang dilakukan pada setiap *Sprint*, pengujian akhir dilakukan untuk memverifikasi integrasi seluruh fitur aplikasi dan memastikan tidak ada konflik atau kesalahan sistem. Pengujian dilakukan melalui metode *black box testing*, yang berfokus pada fungsionalitas aplikasi tanpa mengganggu atau memperhatikan logika internal sistem (Pratama, 2021).

Pengujian aplikasi terbagi menjadi dua tahap utama: pengujian unit dan pengujian akhir. Pengujian unit dilakukan untuk memastikan setiap fitur berfungsi dengan baik dan sesuai dengan yang diharapkan. Sementara itu, pengujian akhir dilakukan untuk menguji keseluruhan fitur aplikasi secara terpadu dan memastikan bahwa sistem secara keseluruhan bekerja dengan baik dan sesuai dengan tujuan yang telah ditentukan.

3.5.1 Metode Pengujian

Proses pengujian aplikasi SIBER menggunakan pendekatan *black box testing*, yang menilai fungsionalitas aplikasi berdasarkan input dan output yang diharapkan tanpa melibatkan pemahaman tentang struktur internalnya. Pengujian dilakukan untuk memeriksa kesesuaian fitur dengan spesifikasi yang telah ditetapkan sebelumnya. Pengujian ini juga bertujuan untuk memastikan bahwa aplikasi berfungsi dengan baik di berbagai kondisi dan tidak menimbulkan kesalahan atau konflik saat digunakan.

3.5.2 Hasil Pengujian

Tabel 3. 9 dan Tabel 3. 10 menunjukkan hasil pengujian dari berbagai fitur dalam aplikasi SIBER. Setiap pengujian dilakukan berdasarkan skenario yang telah disiapkan sebelumnya untuk memeriksa fungsionalitas dari setiap fitur yang diuji. Proses pengujian melibatkan *Product Owner* dan *Project Leader* untuk memastikan fitur bekerja dengan baik dan sesuai dengan harapan.

Setelah pengujian dilakukan, umpan balik dari *Product Owner* dan tim pengembang sangat penting untuk melakukan memastikan kualitas dan kinerja aplikasi telah sesuai dengan yang diharapkan. Pada tahap pengujian akhir ini, seluruh fitur telah berfungsi dengan baik tanpa ditemukan masalah atau kebutuhan untuk penambahan maupun perbaikan. Hasil pengujian ini

juga menunjukkan bahwa aplikasi telah memenuhi spesifikasi yang ditetapkan sebelumnya dan siap digunakan sesuai tujuannya.

Tabel 3. 9 Pengujian *Dashboard* AMR

No.	Fitur	Langkah Pengujian	Skenario Pengujian Berhasil	Skenario Pengujian Gagal	Status
1.	<i>Navbar dashboard</i>	1. Klik tautan " <i>Dashboard Tegangan AMR</i> ".	Semua tautan navigasi berfungsi dengan benar.	Beberapa atau semua tautan tidak berfungsi.	Sukses
2.	<i>Scorecard</i>	1. Pilih data yang akan dilihat pada <i>scorecard</i> , lalu periksa apakah data seperti "Total Pelanggan", "Perlu Perbaikan", "Perlu Perhatian" tampil sesuai filter.	Data ditampilkan dengan benar sesuai dengan filter yang diberikan.	Data tidak sesuai dengan filter yang diterapkan.	Sukses
3.	Peta pelanggan	1. Gunakan kursor untuk zoom <i>in</i> dan zoom <i>out</i> untuk memperbesar atau memperkecil peta. 2. Klik titik koordinat pelanggan untuk menampilkan informasi lebih lanjut. 3. Gunakan fitur filter untuk menampilkan data pelanggan pada peta.	Peta berfungsi dengan baik, termasuk zoom, titik koordinat, dan kesesuaian dengan data yang tertampil.	Peta tidak menampilkan data, filter tidak berfungsi, atau fungsi zoom tidak berfungsi	Sukses
4.	Data tabel pelanggan	1. Pilih data pelanggan untuk diperbarui. 2. Gunakan fungsi " <i>Copy to Clipboard</i> ", "Ekspor PDF", "Ekspor CSV", dan "Delete Row".	Data ditampilkan secara lengkap dan akurat, dan dapat disaring sesuai dengan kategori yang benar.	Data tidak lengkap, tidak akurat, atau tidak dapat disaring.	Sukses
5.	<i>Pie chart</i>	1. Terapkan filter berdasarkan status untuk melihat data yang ditampilkan dalam diagram.	Diagram menampilkan data yang telah dikelompokkan dengan benar.	Diagram tidak menampilkan data dengan benar atau tidak sesuai filter.	Sukses
6.	Filterisasi ke <i>scorecard</i> , peta, dan diagram lingkaran	1. Terapkan filter lokasi, status, tegangan, dan fungsi pilih kolom untuk memodifikasi tampilan <i>scorecard</i> , peta, dan diagram.	Semua fitur <i>dashboard</i> bereaksi sesuai dengan filter yang diberikan.	Filter tidak mempengaruhi peta, <i>scorecard</i> , diagram dengan benar.	Sukses

Tabel 3. 10 Pengujian *Dashboard Kinerja*

No.	Fitur	Langkah Pengujian	Skenario Pengujian Berhasil	Skenario Pengujian Gagal	Status
1.	<i>Navbar dashboard</i> perhalaman	<ol style="list-style-type: none"> Klik tautan "Home". Klik tautan "<i>Dashboard Kinerja</i>". Klik tautan "Pelanggan & Daya". Klik tautan "Susut, KWH Jual, KWH Produksi". Klik tautan "Saidi, Saifi, Recovey time, GGN 100 Kms, GGN Reclosure". Klik tautan "HPL TNP, HPL Perluasan TRF, HPL TM". 	Semua navigasi berfungsi dengan benar di setiap halaman.	Navigasi tidak berfungsi di setiap halaman dengan benar.	Sukses
2.	<i>Scorecard</i> perhalaman	<ol style="list-style-type: none"> Periksa data pada <i>scorecard</i> di setiap halaman. 	Mengkonfirmasi bahwa data ditampilkan dengan benar sesuai dengan filter yang diberikan.	Data tidak sesuai dengan filter atau tidak ditampilkan di beberapa halaman.	Sukses
3.	Diagram batang untuk halaman pelanggan & daya, diagram garis untuk halaman sisanya	<ol style="list-style-type: none"> Gunakan fitur edit data untuk mengubah data. Gunakan fitur ubah tahun. Periksa diagram yang ditampilkan dan dibandingkan dengan data yang ada pada database. 	Memastikan bahwa diagram telah menampilkan data secara akurat dan mudah dipahami di setiap halaman.	Diagram tidak menampilkan data dengan benar atau sulit dipahami.	Sukses
4.	Filterisasi ke <i>scorecard</i> dan tahun	<ol style="list-style-type: none"> Terapkan filter tahun untuk melihat apakah diagram berubah sesuai dengan filter yang diterapkan. 	Memastikan bahwa semua filter di setiap halaman dapat diatur berdasarkan tahun.	Filter tidak mempengaruhi tampilan data dengan benar.	Sukses

BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Teknis

4.1.1 Bentuk Implementasi *Scrum*

Selama magang enam bulan di PLN UP3 Yogyakarta, aplikasi Sistem Informasi Bersama (SIBER) mengimplementasikan metode *Scrum* sebagai metode pengembangan yang efektif. *Scrum* dipilih karena fleksibilitas dan kemampuannya untuk beradaptasi dengan perubahan kebutuhan. Meski demikian, karena keterbatasan sumber daya dan pengalaman tim dalam *Scrum*, beberapa aspek *framework* ini diadaptasi sesuai konteks proyek. Contohnya, *Daily Sprint* sering kali dilakukan lebih dari 15 menit untuk memungkinkan diskusi mendalam, sementara *Sprint Retrospective* difokuskan pada evaluasi cepat akibat jadwal *Project Leader* yang padat. Adaptasi ini dilakukan untuk menjaga keberlanjutan proyek tanpa mengorbankan prinsip-prinsip utama *Scrum*. Namun, risiko tetap muncul, seperti kurang optimalnya peran *Scrum Master* dan kolaborasi tim, yang mempengaruhi efektivitas implementasi dan dapat mengurangi esensi *Scrum* sebagai kerangka kerja yang efisien dan terstruktur. Meskipun demikian, keberhasilan pengembangan aplikasi menunjukkan bahwa prinsip inti *Scrum*, seperti iterasi yang teratur dan kolaborasi, tetap dipertahankan.

Pada Tabel 4. 1 pengembangan aplikasi SIBER mengikuti pedoman *Scrum guide* dan realisasinya di proyek ini. Kolom peran pada Tabel 4. 1 menjelaskan posisi dalam tim, *Scrum guide* menggambarkan deskripsi ideal dari setiap peran berdasarkan panduan *Scrum*, dan kolom proyek SIBER menjelaskan tentang bagaimana masing-masing peran diimplementasikan dalam proyek. Kolom keterangan menilai kesesuaian pelaksanaan peran di proyek SIBER dengan pedoman *Scrum*, dan kolom gap mengidentifikasi perbedaan antara pelaksanaan di proyek dengan standar dalam *Scrum guide*.

Tabel 4. 1 Implementasi *Scrum* Berdasarkan Peran

No.	Peran	<i>Scrum Guide</i>	Proyek SIBER	Keterangan	Gap
1.	<i>Product Owner</i>	Bertanggung jawab untuk memaksimalkan nilai produk yang dihasilkan dan mengelola <i>Product Backlog</i> yang harus dilakukan oleh <i>Development Team</i> .	Peran ini ditangani oleh Asisten Manajer Transaksi Energi dan Perencanaan. Mereka bertanggung jawab atas pembuatan <i>dashboard</i> dan pengelolaan waktu proyek.	Peran sesuai dengan panduan <i>Scrum</i> dan melaksanakan tugas dengan baik	Tidak ada gap signifikan antara pelaksanaan dan panduan.
2.	<i>Scrum Master</i>	Memastikan <i>Development Team</i> memahami teori dan praktik <i>Scrum</i> serta implementasinya yang benar.	Peran ini dijalankan oleh <i>Project Leader</i> dari divisi Perencanaan. <i>Project Leader</i> membantu <i>Scrum</i> tim memahami <i>Product Backlog</i> dan mengatasi hambatan.	Peran tidak sepenuhnya berjalan sebagaimana mestinya.	Gap terlihat dalam komunikasi dan kurang dalam mengenalkan penggunaan kerangka kerja <i>Scrum</i> , sehingga anggota tim masih belum mengimplementasikan konsep <i>Scrum</i> dengan baik.
3.	<i>Development Team</i>	Menciptakan produk sesuai dengan desain <i>Product Owner</i> , serta membuat <i>Sprint Planning</i> .	Terdiri dari <i>frontend</i> dan <i>backend developer</i> . <i>Frontend</i> bertugas pada bagian tampilan antarmuka dan pengalaman pengguna sedangkan <i>backend</i> bertugas pada bagian server dan basis data.	<i>Development Team</i> bersifat lintas-fungsi yang dibutuhkan untuk menghasilkan produk yang siap untuk dirilis.	Peran kurang sesuai dengan <i>Scrum guide</i> , karena kurangnya komunikasi dan kolaborasi yang menyebabkan miskomunikasi.

Tabel 4. 2 Implementasi *Scrum* Tiap Aktivitasnya

No.	Peran	<i>Scrum Guide</i>	Proyek SIBER	Keterangan	Gap
1.	<i>Product Backlog</i>	<i>Product Backlog</i> adalah daftar kebutuhan dalam proyek, disusun berdasarkan prioritasnya setiap fitur.	<i>Product Backlog</i> berisi daftar kebutuhan fitur yang diprioritaskan dan diproses oleh tim <i>Scrum</i> menjadi sebuah fitur suatu produk.	<i>Product Backlog</i> diuraikan dalam bentuk tabel oleh <i>Development Team</i> dan dipantau oleh <i>Product Owner</i> .	<i>Product Backlog</i> disusun sejak awal proyek dikelola oleh <i>Product Owner</i> . <i>Product Backlog</i> disusun manual dalam bentuk <i>excel</i> .
2.	<i>Sprint Planning</i>	<i>Sprint Planning</i> memaparkan pekerjaan yang akan dilakukan selama <i>Sprint</i> .	<i>Sprint Planning</i> dilakukan di awal setiap <i>Sprint</i> , menganalisis <i>Product Backlog</i> dan penjabaran tugas masing-masing anggota.	Melibatkan <i>Product Owner</i> dan <i>Project Leader</i> untuk merencanakan pekerjaan selama proyek berlangsung.	<i>Sprint Planning</i> dilakukan di awal setiap <i>Sprint</i> dengan melibatkan <i>Product Owner</i> dan <i>Project Leader</i> .
3.	<i>Sprint</i>	<i>Sprint</i> adalah aktivitas inti dari <i>Scrum</i> , dengan durasi tertentu untuk mengubah ide menjadi nilai.	Durasi <i>Sprint</i> bervariasi tergantung pada kompleksitas fitur. Setiap <i>Sprint</i> dimulai dengan <i>Sprint Planning</i> dan diakhiri dengan <i>Sprint Review</i> dan <i>Sprint Retrospective</i> .	<i>Sprint</i> mencakup seluruh aktivitas dari perencanaan hingga pengembangan fitur.	Durasi <i>Sprint</i> bervariasi karena tim pengembang yang belum berpengalaman memerlukan waktu untuk mencari solusi.
4.	<i>Daily Sprint</i>	<i>Daily Sprint</i> adalah pertemuan singkat 15 menit untuk meningkatkan komunikasi dan pengambilan keputusan.	<i>Daily Sprint</i> dilaksanakan setiap pagi sebelum memulai aktivitas. Tim melaporkan kemajuan dan kendala yang dihadapi.	<i>Daily Sprint</i> sering berlangsung lebih dari 15 menit, dan partisipasi tim selalu tidak konsisten.	<i>Daily Sprint</i> sering melebihi 15 menit dan partisipasi tim tidak konsisten karena anggota tim sering absen atau terlambat.
5.	<i>Sprint Review</i>	<i>Sprint Review</i> adalah pertemuan di akhir <i>Sprint</i> untuk memeriksa hasil dan meninjau <i>Product Backlog</i> .	<i>Sprint Review</i> dilaksanakan setiap akhir <i>Sprint</i> pada hari Juma. Tim mempresentasikan kendala yang dihadapi dan memberikan laporan kepada <i>Product Owner</i> dan <i>Project Leader</i> .	Pertemuan ini digunakan untuk menyelaraskan tujuan dan prioritas untuk <i>Sprint</i> berikutnya.	<i>Sprint Review</i> sering hanya melibatkan <i>Product Owner</i> tanpa kehadiran <i>Project Leader</i> , sehingga umpan balik yang diterima kurang mendalam.
6.	<i>Sprint Retrospective</i>	<i>Sprint Retrospective</i> adalah kesempatan untuk merencanakan peningkatan kualitas dan efektivitas.	<i>Sprint Retrospective</i> dilaksanakan di akhir setelah <i>Sprint Review</i> . Tim mendiskusikan tantangan yang dihadapi dan langkah perbaikan.	Beberapa pertemuan <i>Sprint Retrospective</i> kurang fokus dan terstruktur.	<i>Sprint Retrospective</i> terkadang kurang fokus dan tidak semua anggota berpartisipasi aktif, sehingga evaluasi dan perbaikan tidak optimal.

Tabel 4. 2 menunjukkan dampak dari modifikasi dalam implementasi *Scrum* oleh tim pengembang aplikasi SIBER. Kolom peran merujuk pada aktivitas utama dalam *Scrum*, sedangkan kolom proyek SIBER memaparkan deskripsi standar aktivitas yang diterapkan dalam proyek. Kolom keterangan memberikan detail tambahan terkait pelaksanaan aktivitas, dan terakhir kolom gap mengidentifikasi perbedaan antara praktik ideal dan realisasi di proyek SIBER.

Beberapa tugas dalam peran *Scrum* telah dilaksanakan sesuai dengan pedoman, namun ada pula peran yang realisasinya memerlukan improvisasi. Salah satu kendala utama adalah minimnya focus dan partisipasi dalam *Sprint Retrospective*, yang mengurangi efektivitas perbaikan proses kerja di *Sprint* berikutnya. Untuk mengatasi hal ini, *retrospective* difasilitasi dengan agenda yang lebih terstruktur dan beberapa diskusi individu terutama dengan *Project Leader*.

Selain itu, *Daily Sprint* yang sebelumnya sering melebihi waktu dirancang ulang dengan agenda yang lebih jelas untuk menjaga efisiensi pertemuan. Peran *Scrum Master* juga diperkuat melalui pelatihan singkat untuk meningkatkan pemahaman tim terhadap metode *Scrum*. Pembagian tim lintas-fungsi membantu dalam menyelesaikan tugas secara efektif selama periode *Sprint*.

Terdapat sejumlah gap yang mempengaruhi proses dan hasil akhir. Gap ini memiliki dampak positif dan negatif yang dipengaruhi faktor internal dan eksternal, seperti:

- a. Salah satu anggota pengembang yang masih memiliki jadwal kelas luring di kampus yang menyebabkan keterlambatan dalam menyelesaikan tugas dan aktivitas harian. Tenggat waktu yang sering mundur dari kesepakatan anggota tim lain untuk membantu menyelesaikan tugas yang seharusnya menjadi tanggung jawab anggota tersebut.
- b. Tim pengembang mengalami kesulitan dalam menerapkan praktik, aturan, serta nilai-nilai *Scrum* karena kurangnya pemahaman. Hal ini terutama disebabkan oleh peran *Scrum Master* yang kurang dalam hal komunikasi dan kolaborasi aktif, sehingga penerapan *Scrum* tidak berjalan optimal.
- c. Proyek sering mengalami penundaan jadwal evaluasi mingguan karena sulitnya menghubungi *Project Leader* yang sibuk dengan pekerjaan internal perusahaan. Hal ini mengakibatkan mundurnya jadwal evaluasi atau seringnya perubahan hari evaluasi.

Selain itu, terdapat beberapa faktor yang dapat mempengaruhi keberhasilan dalam implementasi *Scrum* selama proyek berlangsung, seperti:

- a. *Sprint Retrospective* menjadi kunci untuk meningkatkan proses kerja di *Sprint* berikutnya. Melalui *Sprint Retrospective*, tim dapat belajar dari kesalahan sebelumnya dan menghindari pengulangan kesalahan yang sama. Agar efektif, *Sprint Retrospective* harus difasilitasi dengan baik oleh *Scrum Master*.
- b. Komunikasi yang efektif dan kolaborasi yang baik antara tim pengembang, *Scrum Master*, dan *Product Owner* sangat penting. Dengan komunikasi yang jelas, setiap anggota dapat memahami tugas dan tanggung jawabnya. Kolaborasi yang baik membantu tim bekerja lebih efisien dan mengatasi hambatan dengan lebih mudah, serta memastikan kebutuhan dan prioritas proyek terakomodasi dengan baik.
- c. Selama pengembangan aplikasi dengan metode *Scrum*, peran *Scrum Master* sangat penting. *Scrum Master* bertanggung jawab untuk menjelaskan dan mengajarkan penggunaan kerangka kerja *Scrum* secara keseluruhan. Tanpa *Scrum Master* yang efektif, penerapan *Scrum* dalam perusahaan dapat berjalan kurang optimal.

4.1.2 Dampak Implementasi *Scrum*

Penerapan kerangka kerja *Scrum* dalam pengembangan aplikasi SIBER memiliki signifikan baik positif maupun negatif. Dampak ini dipengaruhi oleh berbagai faktor internal dan eksternal yang mempengaruhi efektivitas implementasi *Scrum* dalam proses pengembangan.

Scrum memberikan kerangka kerja yang fleksibel, namun keberhasilannya sangat bergantung pada adaptasi sesuai konteks proyek. Dalam proyek ini, suksesnya implementasi *Scrum* sebagian besar disebabkan oleh pengelolaan *backlog* yang terstruktur, perencanaan *Sprint* yang jelas, dan kolaborasi lintas fungsi. Meskipun terdapat kekurangan, seperti peran *Scrum Master* yang kurang optimal, fokus pada evaluasi iteratif dan komunikasi intensif memungkinkan proyek tetap menghasilkan produk akhir sesuai kebutuhan. Hal ini menunjukkan bahwa fleksibilitas dan siklus perbaikan berkelanjutan dalam *Scrum* mampu mengatasi berbagai tantangan yang muncul selama proyek berlangsung. Berikut adalah dampak positif dan negatif dari penerapan *Scrum* dalam proyek SIBER:

- a. Dampak positif
 1. Implementasi *Scrum* memungkinkan tim untuk mudah beradaptasi dan fleksibel terhadap perubahan yang terjadi selama pengembangan.
 2. *Scrum* mendorong transparansi dan kolaborasi tinggi di antara anggota tim, yang memungkinkan kerja sama lebih efektif dan penyelesaian masalah lebih cepat.

3. Manajemen tugas dalam *Scrum* dibagi secara jelas di antara tim, meningkatkan efisiensi dan produktivitas dalam hasil produk.
 4. *Scrum* memungkinkan pengembangan produk melalui iterasi berkelanjutan, memberi kesempatan bagi tim untuk mengevaluasi dan memperbaiki produk secara berkala, sehingga meningkatkan kualitas produk akhir.
- b. Dampak negatif
1. Kegagalan dalam melaksanakan beberapa aspek *Scrum*, seperti peran *Scrum Master* yang kurang aktif dan ketelibatan *Project Leader* yang terbatas, menghambat progress proyek karena tim bergantung pada arahan dan keputusan mereka. Hal ini berdampak pada penundaan penyelesaian beberapa fitur yang telah direncanakan karena kurangnya evaluasi berkelanjutan. Implikasi dari kegagalan ini adalah perlunya pendekatan yang lebih adaptif dalam menentukan peran kunci tersebut serta pentingnya pelatihan tambahan bagi tim agar lebih memahami nilai-nilai *Scrum*.
 2. *Scrum* mendorong perubahan berdasarkan umpan balik, namun ini dapat menjadi kontraproduktif jika perubahan tersebut menyebabkan kebingungan di tim dan mempengaruhi produktivitas serta hasil akhir produk.

4.2 Nonteknis

4.2.1 Pengembangan *Soft skill*

Selama pengembangan proyek SIBER, penulis tidak hanya mendapatkan keterampilan teknis, tetapi juga keterampilan nonteknis atau *soft skill* yang penting dalam dunia kerja. Keterampilan ini meliputi:

- a. Penulis belajar bagaimana menyampaikan ide dan informasi dengan jelas dan tepat sasaran, baik secara lisan maupun secara tulisan, kepada rekan tim dan atasan.
- b. Melalui proyek dan tugas kelompok, penulis meningkatkan kemampuan kerja sama, berkontribusi secara positif untuk mencapai tujuan bersama, dan memahami kolaborasi tim.
- c. Penulis belajar mengatur waktu secara efisien untuk menyelesaikan tugas-tugas, memprioritaskan pekerjaan, dan meminimalkan penundaan untuk memenuhi *deadline*.

4.2.2 Pengalaman Lingkungan Kerja

Pengalaman langsung di lingkungan kerja memberikan banyak pelajaran berharga yang tidak hanya memperbanyak dalam ilmu pengetahuan tetapi juga menambah keterampilan lebih lanjut. Beberapa pelajaran penting meliputi:

- a. Penulis belajar menyesuaikan diri dengan cepat di lingkungan kerja, memahami dan menghormati budaya kerja di PLN, serta berinteraksi secara profesional dengan rekan kerja dari berbagai latar belakang.
- b. Penulis membangun dan memperluas jaringan profesional dengan bertemu dan berinteraksi dengan para profesional di bidangnya, yang bermanfaat bagi karir masa depan.
- c. Penulis belajar untuk mengambil tanggung jawab atas tugas dan proyek yang diberikan, bekerja secara efektif, dan menunjukkan inisiatif dalam menyelesaikan pekerjaan dengan benar.

4.3 Refleksi dan Pembelajaran

Selama enam bulan magang dan pengembangan aplikasi SIBER di PLN UP3 Yogyakarta, penulis memperoleh banyak pembelajaran yang tidak hanya berkaitan dengan keterampilan teknis, tetapi juga dengan aspek pengelolaan proyek, kolaborasi tim, dan adaptasi terhadap perubahan.

4.3.1 Pelajaran Terbesar dalam Proses Pengembangan

Salah satu pelajaran terbesar yang saya peroleh adalah pentingnya fleksibilitas dalam menghadapi perubahan kebutuhan dan hambatan yang muncul selama pengembangan. *Scrum* sebagai metodologi pengembangan yang penulis terapkan memberi banyak ruang untuk beradaptasi dengan perubahan, namun, tanpa adanya komunikasi yang efektif dan manajemen yang baik, perubahan tersebut bisa menjadi tantangan besar. Selama pengembangan aplikasi SIBER, penulis sering kali dihadapkan pada perubahan kebutuhan, baik dari sisi desain tampilan maupun fungsionalitas, yang mengharuskan tim untuk melakukan penyesuaian dengan cepat.

Hal yang penulis pelajari adalah bahwa dalam menghadapi perubahan ini, peran *Scrum Master* sangat krusial. Pengelolaan perubahan yang baik harus melibatkan pemahaman mendalam mengenai proses *Scrum* serta kemampuan untuk menjaga komunikasi terbuka antar seluruh anggota tim, termasuk *Product Owner* dan *Project Leader*.

4.3.2 Hal-Hal yang Dapat Diterapkan di Masa Depan

Berdasarkan pengalaman ini, penulis menyadari pentingnya beberapa hal yang dapat diterapkan pada proyek serupa di masa depan untuk meningkatkan kualitas dan efisiensi:

- a. Peningkatan komunikasi dan kolaborasi antar tim

Komunikasi yang lebih terkoordinasi antara tim pengembang, *Product Owner*, dan *Project Leader* sangat penting untuk efisiensi kerja. Kolaborasi yang baik memungkinkan tim memecahkan masalah lebih cepat dan lebih mudah menyesuaikan diri terhadap perubahan yang terjadi.

b. Manajemen waktu dan sumber daya yang lebih baik

Perencanaan sumber daya yang matang menjadi prioritas untuk memastikan anggota tim dapat mengalokasikan waktu secara optimal dan menyelesaikan tugas. Hal ini mencakup pertimbangan terhadap kendala waktu individu, seperti jadwal kuliah atau pekerjaan lain, yang dapat mempengaruhi progres *Sprint* secara keseluruhan.

c. Optimalisasi peran *Scrum Master* dan pemahaman tim terhadap *Scrum*

Scrum Master harus lebih aktif dalam menjaga komunikasi dan memastikan bahwa semua anggota tim memahami serta menerapkan prinsip-prinsip *Scrum*. Berdasarkan evaluasi implementasi *Scrum* dalam proyek ini, keberhasilan proyek lebih didorong oleh adaptasi yang tepat daripada kepatuhan penuh pada *framework Scrum*. Oleh karena itu, fleksibilitas dalam pengelolaan proyek harus seimbang dengan penguatan prinsip dasar *Scrum* untuk menjaga keberhasilan di masa mendatang.

4.3.3 Saran untuk Proyek Serupa di Masa Depan

Berdasarkan pengalaman dan pelajaran yang didapatkan dalam pengembangan aplikasi SIBER, beberapa saran yang bisa diterapkan pada proyek-proyek serupa di masa depan adalah sebagai berikut:

- a. Penggunaan *tools* manajemen proyek yang efektif: menggunakan alat bantu manajemen proyek seperti Jira atau Trello untuk membantu progres dan memprioritaskan pekerjaan, serta memastikan kolaborasi yang lancar antar anggota tim.
- b. Mengelola *stakeholder* dengan lebih baik: menjaga keterlibatan *Product Owner* dan *Project Leader* dalam setiap tahapan pengembangan, memastikan komunikasi yang jelas dan konsisten mengenai prioritas dan kebutuhan proyek.
- c. Pengelolaan perubahan kebutuhan dan risiko: selama pengembangan aplikasi, perubahan kebutuhan risiko dapat mempengaruhi kelancaran proyek. Untuk itu, penting untuk memiliki mekanisme yang jelas dalam mengelola perubahan, serta mengidentifikasi potensi risiko diawal. Hal ini akan membantu tim untuk lebih siap dalam menghadapi ketidakpastian dan memperkecil dampak negatif terhadap jadwal dan kualitas produk.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

- a. Metode *Scrum* berhasil diterapkan pada pengembangan *dashboard* visualisasi data proyek PLN UP3 Yogyakarta. Penerapan *Scrum* memungkinkan tim untuk bekerja secara kolaboratif dan terstruktur, melalui iterasi sprint yang jelas dan penentuan prioritas yang tepat. Dengan pengelolaan *backlog* yang baik, tim mampu mencapai target *Increment* dalam setiap sprint, yang menunjukkan keberhasilan proses pengembangan. Pencapaian setiap fase pengembangan yang sesuai dengan jadwal yang telah ditentukan juga mengindikasikan bahwa *Scrum* dapat meningkatkan efisiensi dalam pengelolaan waktu proyek.
- b. Penerapan metode *Scrum* pada proyek SIBER memiliki beberapa keunggulan, seperti fleksibilitas dalam merespons perubahan, peningkatan kolaborasi tim, serta transparansi dalam proses pengembangan. Selain itu, *Scrum* memfasilitasi evaluasi berkelanjutan melalui *Sprint Review* dan *retrospective*, yang memungkinkan perbaikan berkesinambungan. Namun, terdapat beberapa kekurangan, seperti ketergantungan pada keterlibatan *Product Owner* dan *Project Leader* yang jika kurang terlibat dapat menghambat progres proyek. Selain itu, perubahan yang sering terjadi akibat umpan balik dapat menimbulkan kebingungan di dalam tim jika tidak dikelola dengan baik. Meskipun demikian, *Scrum* tetap layak untuk diterapkan pada proyek pengembangan sistem selanjutnya karena efektivitasnya dalam memastikan pencapaian tujuan proyek secara bertahap dan adaptif.
- c. *Scrum* berhasil membantu proyek berjalan lebih tepat waktu dan terkelola dengan baik. Meskipun terdapat variasi dalam estimasi waktu selama sprint, di mana beberapa tugas memakan waktu lebih lama dan beberapa selesai lebih cepat, metode ini tetap efektif dalam menjaga proyek sesuai jadwal. Pembagian tugas yang terstruktur, sprint yang terencana, serta *Daily Sprint* memastikan setiap hambatan dapat segera diidentifikasi dan diatasi. Fleksibilitas *Scrum* dalam merespon perubahan estimasi memungkinkan tim tetap berada di jalur yang benar, sehingga risiko keterlambatan dapat diminimalkan. Hal ini mendukung tujuan pengelolaan proyek yang lebih baik dan memastikan aplikasi dapat diselesaikan tepat waktu, sesuai dengan ekspektasi pengguna.

5.2 Saran

Untuk meningkatkan hasil implementasi *Scrum* dalam pengembangan aplikasi SIBER, beberapa hal perlu diperhatikan:

- a. Perlu adanya batasan terkait permintaan dan kebutuhan yang diminta oleh *Product Owner* agar target dapat terealisasi tepat waktu.
- b. Komunikasi dan kolaborasi dengan beberapa pihak perlu dilakukan secara konsisten dan teratur pada tiap pertemuan.

DAFTAR PUSTAKA

- Al Attar, Z., Kusumasari, T. F., & Fauzi, R. (2022, Desember). Pengembangan Sistem Dashboard untuk Mengevaluasi Platform E-Marketplace dengan Metode *Scrum* pada Startup GRAVIS. *Jurnal Edukasi & Penelitian Informatika*, 536. doi:<http://dx.doi.org/10.26418/jp.v8i3.58782>
- Alexander, A., & Noranita, B. (2021, Desember). Penerapan Business Intelligence Dashboard Pada Sekolah Tinggi Pertanahan Nasional Menggunakan Metodologi *Scrum*. *Jurnal Masyarakat Informatika*, 66. doi:<https://doi.org/10.14710/jmasif.12.2.41048>
- Chasbulloh, A. R., Kautsar, M. A., Riyanto, M. R., Hapsari, R. K., & Widodo, W. (2023, Februari). Perancangan Sistem Informasi Manajemen Penjualan UMKM Warung Rujak Cingur Berbasis Web dengan Model Pengembangan Agile *Scrum*. *Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi*, 165. doi:10.31284/p.semtik.2023-2.4721
- Fernando, D. (2018, November). Visualisasi Data Menggunakan Google Data Studio. *Seminar Nasional Rekayasa Teknologi Informasi*, 71. Diambil kembali dari <http://e-jurnal.lppmunsera.org/index.php/snartisi/article/view/808>
- Gutama, R. (2020). Implementasi *Scrum* pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP). *Dspace UII*. Diambil kembali dari <https://dspace.uui.ac.id/handle/123456789/31414>
- Handayani, N. U., & Wibowo, M. A. (2023). *Revolusi Industri 4.0: Perspektif Teknologi, Manajemen, dan Edukasi*. PENERBIT ANDI. Diambil kembali dari https://sipil.ft.undip.ac.id/wp-content/uploads/2023/01/2021-Revolusi-Industri-4.0-Perspektif-Teknologi-Manajemen-dan-Edukasi_-ebook.pdf
- Kurniyanti, V. A., & Mulyadi, D. (2022, Agustus). Perbandingan Model Waterfall Dengan Prototype Pada Pengembangan System Informasi Berbasis Website. *JURNAL SYNTAX FUSION*, 632. doi:<https://doi.org/10.54543/fusion.v2i08.210>
- Mutawali, L., Fathoni, B. K., & Asyari, H. (2020, Juli). IMPLEMENTASI *SCRUM* DALAM PENGEMBANGAN SISTEM INFORMASI JASA DESAIN GRAFIS. *Jurnal Manajemen Informatika & Sistem Informasi*, 116. doi:<https://doi.org/10.36595/misi.v3i2.166>

- Nur Rizki, D. A. (2020). Visualisasi Data Sentimen Terhadap Organisasi Perangkat Daerah Pemerintah Provinsi Jawa Barat Di Jabar Digital Service. *elibrary UNIKOM*. doi:10.59141/comserva.v2i08.493
- Pratama, K. N. (2021, Juli). *Rancang bangun sistem informasi e-commerce penjualan baju fashion wanita berbasis website studi kasus : ale clothsky*. Sains dan Teknologi. Jakarta: Universitas Islam Negeri Syarif Hidayatullah. Diambil kembali dari <https://repository.uinjkt.ac.id/dspace/handle/123456789/57156>
- Pratikno, A. S., Prastiwi, A. A., & Rahmawati, S. (2020). Penyajian Data, Variasi Data, dan Jenis Data. *OSF PREPRINTS*. doi:<https://doi.org/10.31219/osf.io/7w8xp>
- Schwaber, K., & Sutherland, J. (2024, Mei 20). *Panduan Scrum*. Diambil kembali dari *Scrum Guide*.
- Setiawan, D. B. (2023, Januari). Pengembangan Dashboard Interaktif Covid-19 sebagai Alat Informasi dan Pendidikan Masyarakat dalam Penanganan Pandemi Studi Kasus di PT. Inovasi Lentera Cipta Kreasi. *Jurnal Dedikasi Pengabdian Masyarakat*, 117. doi:<https://doi.org/10.61227/inisiatif.v2i2.120>
- Suharno, H. R., N., G., & M., S. (2020). Analisis Penerapan Metode *Scrum* Pada Sistem Informasi Manajemen Proyek Dalam Industri & Organisasi Digital. *Majalah Ilmiah Teknologi*.
- Wahid, A. A. (2020, Oktober). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*. Diambil kembali dari https://www.researchgate.net/publication/346397070_Analisis_Metode_Waterfall_Untuk_Pengembangan_Sistem_Informasi
- Wahyudi, A., Assyamiri, M. B., Al Aluf, W., Fadhillah, M. R., Yolanda, S., & Anshori, M. I. (2023, Desember). Dampak Transformasi Era Digital Terhadap Manajemen Sumber Daya Manusia. *Jurnal Bintang Manajemen (JUBIMA)*, 99. doi:<https://doi.org/10.55606/jubima.v1i4.2222>
- Wahyudi, E. D., Ramadha, F. N., Andriani, E. O., & Alfarizy, A. Z. (2024, Juli). ANALISA PERBANDINGAN SOFTWARE DEVELOPMENT MODEL ANTARA METODE SCRUM DAN METODE INCREMENTAL. *Jurnal Media Informatika dan Teknologi (JUMINTEK)*, 8. Diambil kembali dari <https://himdikom.org/index.php/jumintek/article/view/14/13>
- Widiyanto, W. W. (2018, Agustus). ANALISA METODOLOGI PENGEMBANGAN SISTEM DENGAN PERBANDINGAN MODEL PERANGKAT LUNAK SISTEM

INFORMASI KEPEGAWAIAN MENGGUNAKAN WATERFALL DEVELOPMENT MODEL, MODEL PROTOTYPE, DAN MODEL RAPID APPLICATION DEVELOPMENT (RAD). *ResearchGate*, 34.

doi:<https://doi.org/10.46808/informa.v4i1.34>

Winarno, C. A. (2020). Visualisasi Data Untuk Analisis Sentimen Pada Penilaian Kepuasan Pelanggan. *elibrary UNIKOM*. Diambil kembali dari <https://elibrary.unikom.ac.id/id/eprint/4027/>

LAMPIRAN





Logbook

Home / Logbook










Logbook Harian

Export

Tambah

NO. ↕	KEGIATAN/KETERANGAN ↕	TANGGAL ↕	PARAF	AKSI
5	Pengarahan dan Pembekalan materi oleh Supervisor Pemegang Informatika mendapatkan pengarahannya dan pembekalan materi proyek oleh Supervisor	15 September 2023	Belum Diparaf	 
6	Observasi dan Penelitian terkait pembuatan Dashboard Pemegang Informatika melakukan observasi dan penelitian untuk kebutuhan pembuatan Dashboard	18 September 2023	Belum Diparaf	 
7	Membuat Desain Mockup Website Sistem Informasi Bersama (SIBER) Pemegang membuat desain mockup pada Dashboard AMR menggunakan Figma	05 Januari 2024	Belum Diparaf	 
2	Membuat Desain Mockup Dashboard AMR Pemegang membuat desain mockup pada Dashboard AMR menggunakan Figma	19 September 2023	Belum Diparaf	 
8	Pembuatan Dashboard AMR Membuat kode halaman Dashboard AMR dimulai dari Sidebar dan Scorecard menggunakan tailwind	20 September 2023	Belum Diparaf	 
9	Pembuatan Dashboard AMR - Responsif Membuat tampilan sidebar dan scorecard menjadi responsif dengan tailwind	21 September 2023	Belum Diparaf	 

Lampiran 1 Logbook Magang 1

59	Perbaikan Tampilan Pelanggan dan Susut pada Dashboard Kinerja Melakukan perbaikan tampilan berupa mengganti warna chart sesuai dengan scorecard di tiap halaman Dashboard Kinerja	30 November 2023	Belum Diparaf	 
60	Pembuatan Landing Page Sebelum Login Ke Masing-masing Divisi Melakukan pembuatan landing page yang berisi tentang deskripsi perusahaan dan deskripsi web aplikasi	01 Desember 2023	Belum Diparaf	 
61	Pembuatan Halaman Preview Setelah Landing Page Membuat halaman preview setelah landing page yaitu berupa halaman Pelanggan Daya dan HPL berdasarkan evaluasi minggu	04 Desember 2023	Belum Diparaf	 
62	Pembuatan Scorecard di Halaman Preview Pelanggan Daya dan HPL Membuat scorecard dan menghubungkannya dengan data pada halaman preview Pelanggan Daya dan HPL	05 Desember 2023	Belum Diparaf	 
63	Pembuatan Data Tertampil Sesuai Tahun Terbaru di Halaman Preview Pelanggan Daya dan HPL Membuat agar halaman preview Pelanggan Daya dan Hpl dapat menampilkan data sesuai dengan tahun terbaru	06 Desember 2023	Belum Diparaf	 
64	Perubahan Tampilan Halaman Preview Pelanggan Daya, HPL Mengedit tampilan berupa warna scorecard dan menyesuaikan thbl yang diambil	07 Desember 2023	Belum Diparaf	 
65	Perubahan Data Yang Diambil di Halaman Preview Pelanggan Daya Merubah tampilan agar chart dapat mengambil data langsung ke 800 pelanggan sesuai data yang ada	08 Desember 2023	Belum Diparaf	 
66	Penambahan dan Perubahan Pada Tampilan Halaman Preview Menambahkan judul dan memperbaiki tata letak chart pada Halaman Preview Pelanggan Daya dan HPL	11 Desember 2023	Belum Diparaf	 
67	Testing Halaman Preview Dashboard Kinerja Melakukan testing bersama dengan supervisor dan pemegang informatika lainnya terkait dengan halaman preview Dashboard Kinerja	12 Desember 2023	Belum Diparaf	 
68	Testing Halaman Preview Dashboard Kinerja Melakukan testing bersama dengan supervisor dan pemegang informatika lainnya terkait dengan halaman preview Dashboard Kinerja	12 Desember 2023	Belum Diparaf	 

Lampiran 2 Logbook Magang 2



Lampiran 3 Dokumentasi Sewatku Istirahat



Lampiran 4 Dokumentasi Perayaan HUT PLN



Lampiran 5 Dokumentasi Sewaktu Pengerjaan Proyek



Lampiran 6 Kunjungan Jurusan Informatika

```

tegangan.php 9+ x
ansaksi > tegangan.php > html > body > div#body.w-full.flex.flex-col.min-h-screen > div.w-full.flex.flex-col.lg:p-5.px-1.py-2.gap-2.lg:gap-3 > div.flex.lg:flex-row.flex-col.w-full.gap-5 > div.w-full.border.p-3
15 <html lang="en">
165 <body>
166 <div class="w-full flex flex-col min-h-screen" id="body">
196 <div class="w-full flex flex-col lg:p-5 px-1 py-2 gap-2 lg:gap-3">
222 <!-- map -->
223 <div class="flex lg:flex-row flex-col w-full gap-5">
224 <div class="w-full border p-3">
225 <div class="flex flex-col" id="mapContainer">
226 <!-- class map -->
227 <div id="map" class="lg:h-[500px] h-[350px]" style="width:100%" class="relative"></div>
228 <!-- Filtering -->
229 </div>
230 <div id="loadingMap" class="p-2 text-lg flex items-center gap-1 w-fill justify-center">
231 <i class="fa fa-map-pin fa-bounce fa-10x"></i>
232 </div>
233 </div>
234 </div>
235
236 <!-- Tabel -->
237 <div class="flex items-start justify-start w-full gap-2 lg:gap-5 lg:flex-row flex-col-reverse">
238 <!-- <div class="w-full border p-3"> punya masnya -->
239 <div class="w-full lg:w-2/3 border p-3">
240 <div id="loadingTable" class="p-2 text-lg flex items-center gap-1 w-fill justify-center">
241 <i class="fa fa-table fa-10x animate-pulse"></i>
242 </div>
243 <!-- Filtering -->
244 <div id="filter-div">
245 <div class="lg:text-xl font-bold">Filter</div>
246 <div class="flex lg:flex-row flex-col gap-2 py-2 items-start w-full lg:gap-20">
247 <div class="flex flex-col gap-1 w-full">
248 <div class="relative inline-block text-left">
249 <button id="dropdown-toggle-filter-lokasi"
250 class="inline-flex justify-center items-center gap-2 w-full rounded-md border border-gray-300 bg-white px-4 py-2 text-sm font-m
251 lokasi <i class="fa fa-chevron-down"></i>
252 </button>
253 <div id="dropdown-menu-filter-lokasi"
254 class="hidden origin-top-right absolute left-0 mt-2 w-56 rounded-md shadow-lg bg-white ring-1 ring-black ring-opacity-5 z-20">
255 </div>

```

Lampiran 8 Dokumentasi Kode *Dashboard AMR*

```

tegangan.php 9+ x  pelanggan_daya.php 9+ x
main > perencanaan > pelanggan_daya.php > body > div#body.w-full.flex.flex-col.min-h-screen > script > addEventListener("change") callback
212 <body>
213 <div class="w-full flex flex-col min-h-screen" id="body">
272 <div id="filter-div" class="w-full lg:w-5/3 border p-3 mt-5">
308 <div class="flex lg:flex-row flex-col gap-2 py-2 items-start w-full lg:gap-20">
313 <button onclick="openImportModal()" class="inline-flex justify-center items-center gap-2 w-full rounded-md border border-gray-300 b
314 <i class="fa fa-print"></i>
315 Import
316 </button>
317 </div>
318 </div>
319
320 <table class="flex lg:w-5/3 border p-3" id="chartTable">
321 <tr class="flex lg:w-5/3">
322 <td class="flex lg:w-5/3 border p-3 ml-1">
323 <canvas id="barChart1" width="700" height="350"></canvas>
324 </td>
325 <td class="flex lg:w-5/3 border p-3 ml-2">
326 <canvas id="lineChart" width="700" height="350"></canvas>
327 </td>
328 </tr>
329 <tr>
330 <td class="flex lg:w-5/3 border p-3 ml-1 mt-2">
331 <canvas id="barChart2" width="700" height="350"></canvas>
332 </td>
333 <td>
334 <!-- <canvas id="lineChart3" width="800" height="400"></canvas> -->
335 </td>
336 </tr>
337 </table>

```

Lampiran 7 Dokumentasi Kode *Dashboard Kinerja*