

BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Metode penelitian yang digunakan dalam pengembangan *dashboard Business Intelligence (BI)* untuk pemantauan kinerja operasional pada Divisi TechOps di perusahaan satelit ini adalah metode *Agile*, khususnya menggunakan *framework Scrum*. Pemilihan metode *Agile* didasarkan pada karakteristik proyek yang dinamis dan membutuhkan adaptasi cepat terhadap perubahan kebutuhan pengguna serta lingkungan bisnis yang terus berkembang. *Agile* memungkinkan proses pengembangan yang iteratif dan inkremental, sehingga setiap tahapan pengembangan dapat dievaluasi dan disesuaikan secara berkelanjutan sesuai dengan *feedback* dari pengguna dan *stakeholder*.

Dalam konteks Divisi TechOps yang berfokus pada pemantauan kinerja operasional, kebutuhan akan *dashboard BI* yang responsif dan mudah disesuaikan sangat penting. Metode *Agile* memberikan keunggulan dalam hal kolaborasi intensif antara tim pengembang dan pengguna akhir, sehingga produk yang dihasilkan dapat lebih tepat guna dan sesuai dengan kebutuhan nyata di lapangan. *Agile* mendukung pengelolaan risiko yang lebih baik melalui siklus pengembangan yang pendek dan evaluasi berkala.

Metode *Agile* dipilih sebagai pendekatan utama dalam penelitian ini untuk memastikan bahwa pengembangan *dashboard BI* dapat berjalan secara efektif, efisien, dan adaptif terhadap kebutuhan Divisi TechOps, sehingga dapat memberikan nilai tambah yang optimal bagi perusahaan.

3.2 Prosedur Penelitian

Penelitian ini merupakan penelitian pengembangan yang bertujuan untuk menghasilkan sebuah sistem berupa *dashboard Business Intelligence (BI)* yang

Rinjani Putri Djunaedi, 2025

PENGEMBANGAN DASHBOARD BUSINESS INTELLIGENT (BI) UNTUK PEMANTAUAN KINERJA OPERASIONAL PADA DIVISI TECHOPS DI PERUSAHAAN SATELIT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

dapat dimanfaatkan oleh para pelaku operasional unit TechOps pada divisi satelit untuk meningkatkan kinerjanya.

Rinjani Putri Djunaedi, 2025

PENGEMBANGAN DASHBOARD BUSINESS INTELLIGENT (BI) UNTUK PEMANTAUAN KINERJA OPERASIONAL PADA DIVISI TECHOPS DI PERUSAHAAN SATELIT

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Penelitian ini menggunakan metodologi pengembangan dengan Agile terdiri dari 6 tahapan, yakni *requierements*, *design*, *development*, *testing*, *deployment*, dan *review* yang digambarkan melalui diagram berikut:



Gambar 3.4 Tahapan *Agile*
Sumber : (PuskoMedia, 2025)

3.2.1 *Requierements*

Pada tahap *requierements*, peneliti mengidentifikasi tujuan dan ruang lingkup dari pengembangan *dashboard Business Intelligence* (BI). Proses ini melibatkan pengumpulan informasi awal mengenai kebutuhan pengguna dan ekspektasi sistem.

Analisis kebutuhan dilakukan untuk mengidentifikasi dan mendokumentasikan kebutuhan spesifik dari pengguna. Pada tahap ini, peneliti mengumpulkan data dari pengguna dan *stakeholder* untuk memahami kebutuhan spesifik yang harus dipenuhi oleh *dashboard* BI. Proses analisis ini meliputi:

3.2.1.1 Wawancara

Menggunakan teknik wawancara untuk mendapatkan informasi langsung dari

pengguna mengenai fitur yang mereka inginkan. Teknik dalam konteks penelitian ini dilakukan dalam situasi *meeting* di mana *user* dan para

Rinjani Putri Djunaedi, 2025

**PENGEMBANGAN DASHBOARD BUSINESS INTELLIGENT (BI) UNTUK PEMANTAUAN KINERJA OPERASIONAL
PADA DIVISI TECHOPS DI PERUSAHAAN SATELIT**

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

stakeholder terkait mengusulkan *requierement* sesuai kebutuhan masing- masing.

Metode yang dilakukan merupakan wawancara semi-terstruktur untuk mendapatkan informasi mendalam terkait permasalahan sistem data operasional. Berikut merupakan daftar pertanyaan wawancara yang telah dikelompokkan menjadi beberapa topik fokus:

Table 3.1 Daftar Pertanyaan Wawancara

| No | Topik Fokus | Pertanyaan |
|----|-----------------------------------|--|
| 1 | Penyebaran Data Operasional | Bagaimana kemudahan akses data operasional yang tersebar di beberapa sistem? |
| | | Apa tantangan utama saat mengambil data dari sistem berbeda? |
| 2 | Waktu Akses dan Penggabungan Data | Berapa lama waktu rata-rata mengakses data tiap sistem? |
| | | Bagaimana proses penggabungan data dilakukan? Apa kendala yang sering terjadi? |
| 3 | Proses Analisis Manual | Jelaskan bagaimana Anda melakukan analisis kinerja secara manual? |
| | | Apa kendala dalam menjaga akurasi data selama analisis manual? |
| 4 | Koordinasi Antar Departemen | Bagaimana koordinasi antar departemen dalam menyelesaikan masalah data? |
| | | Komunikasi apa yang paling efektif menurut Anda untuk masalah data? |
| 5 | Risiko Kesalahan Manual | Sejauh mana kesalahan manual |

| | | |
|--|--|---|
| | | berdampak pada pengambilan keputusan? |
| | | Apa langkah yang sudah dilakukan untuk meminimalkan kesalahan manual? |

3.2.1.2 Dokumentasi Kebutuhan

Menyusun dokumen kebutuhan yang jelas dan terperinci untuk menjadi acuan dalam pengembangan. Pada pengembangan ini dokumentasi pengumpulan *user requirement* dicatat dalam bentuk MoM (*Minutes of Meeting*).

3.2.1.3 Grafik Performansi *Service*

Grafik performansi *service* adalah alat visual yang digunakan untuk mengukur dan memantau efektivitas penanganan gangguan pada layanan. Fokus dari grafik ini adalah durasi waktu yang dibutuhkan dari saat tiket gangguan dibuka (*open ticket*) hingga tiket tersebut ditutup setelah gangguan terselesaikan (*close ticket*).

3.2.2 *Design*

Berdasarkan kebutuhan yang telah dikumpulkan, dilakukan perancangan arsitektur sistem dan desain antarmuka *dashboard*. Pada tahap ini, peneliti merancang bagaimana data akan ditampilkan dan bagaimana pengguna akan berinteraksi dengan sistem. Spesifikasi yang akan dibuat berdasarkan pada langkah berikut:

3.2.2.1 *Flowchart*

Flowchart dibuat dengan tujuan untuk memetakan dan menggambarkan alur proses dalam sistem secara visual sehingga dapat memudahkan pemahaman dan komunikasi antar anggota tim pengembang serta *stakeholder*. Langkah membuat *flowchart* dimulai dengan mengidentifikasi proses-proses utama yang terjadi dalam sistem *dashboard* BI, seperti pengumpulan data, pemrosesan data, dan penyajian informasi kepada pengguna. Selanjutnya, langkah-langkah proses tersebut disusun

secara berurutan menggunakan simbol-simbol standar *flowchart* seperti oval untuk menandai awal dan akhir, kotak untuk menggambarkan proses, belah ketupat untuk pengambilan keputusan, dan panah untuk menunjukkan aliran proses. Setelah alur susunan langkah proses dibuat, *flowchart* tersebut perlu diverifikasi agar menggambarkan proses yang sebenarnya dan mudah dipahami oleh semua pihak terkait. Untuk memudahkan pembuatan dan penyajian *flowchart*, digunakan perangkat lunak khusus yakni *draw.io*.

3.2.2.2 Perancangan UML

Perancangan UML (*Unified Modeling Language*) adalah metode yang digunakan untuk menggambarkan sistem secara visual melalui berbagai diagram. UML membantu dalam merancang struktur dan perilaku sistem dengan cara yang terstandarisasi. Penggunaan UML pada penelitian ini terdiri dari beberapa bentuk untuk membantu memvisualisasikan gambaran sistem yang akan dibuat, yakni sebagai berikut:

3.2.2.2.1 Use Case Diagram

Pembuatan *Use Case Diagram* diawali dengan mengidentifikasi aktor-aktor yang akan berinteraksi dengan sistem *dashboard* BI, seperti pengguna utama dan administrator. Selanjutnya, peneliti menentukan fungsi utama yang diharapkan dari sistem berdasarkan analisis dokumen kebutuhan yang telah dilakukan. Fungsi-fungsi ini kemudian digambarkan sebagai *use case* yang berinteraksi dengan aktor tersebut. Proses ini bertujuan untuk memvisualisasikan kebutuhan fungsional sistem secara jelas yang selanjutnya akan menjadi dasar pengembangan fitur pada *dashboard* BI. Penggambaran *Use Case Diagram* dilakukan menggunakan perangkat lunak UML untuk menghasilkan diagram yang rapi dan mudah dipahami.

3.2.2.2.2 Activity Diagram

Pembuatan *Activity Diagram* untuk penelitian ini dilakukan dengan menelaah proses bisnis yang berjalan dalam *dashboard* BI mulai dari fase pengumpulan data,

pemrosesan, hingga penyajian data kepada pengguna. Peneliti mendefinisikan setiap aktivitas yang terlibat dan mengurutkannya agar mencerminkan alur kerja sistem yang sebenarnya. Diagram ini menggambarkan bagaimana aktivitas saling berhubungan dan aliran transisi antar aktivitas tersebut. Dalam tahap ini, simbol aktivitas dan pengambilan keputusan digunakan untuk memperjelas langkah-langkah yang bersifat kondisional dan paralel.

3.2.2.2.3 *Class Diagram*

Dalam pembuatan *Class Diagram* ini, peneliti mulai dengan mengidentifikasi objek-objek penting yang ada pada sistem *dashboard* BI beserta atribut dan metode yang relevan untuk setiap objek. Proses ini mengacu pada kebutuhan sistem yang telah diuraikan sebelumnya. Kemudian, hubungan antar objek tersebut, seperti asosiasi, agregasi, dan pewarisan, digambarkan untuk menampilkan struktur statis sistem yang akan dibangun.

3.2.2.2.4 *Data Flow Diagram*

Data Flow Diagram dalam penelitian ini, pembuatan dimulai dengan mengidentifikasi sumber data eksternal yang memasok informasi ke dalam sistem *dashboard* BI, serta proses-proses utama pengolahan data di dalam sistem. Peneliti menggambarkan aliran data antar proses serta penyimpanan data yang digunakan, sehingga memperlihatkan jalur pergerakan data secara rinci dan sistematis. DFD ini sangat berguna untuk memastikan integritas data dan memperjelas bagaimana informasi berpindah dari sumber hingga sampai ke pengguna akhir.

3.2.3 *Development*

Pada tahap pengembangan, desain yang telah disusun diimplementasikan ke dalam kode program. Peneliti menggunakan alat pengembangan Power BI yang kemudian disajikan di antarmuka pengguna dalam bentuk aplikasi website menggunakan fitur *embed*. Tahap ini dilakukan secara iteratif dengan membangun modul-modul *dashboard* sesuai desain yang telah dibuat. Pengembangan dilakukan dalam *sprint* dengan prioritas fitur yang telah ditentukan.

3.2.4 Testing

Tahap pengujian dilakukan untuk memastikan bahwa semua fitur berfungsi dengan baik dan sistem bebas dari *bug*. Pengujian ini mencakup berbagai jenis pengujian, termasuk pengujian fungsional, pengujian integrasi, dan pengujian pengguna. Bagian ini terdiri dari dua tahap yakni :

3.2.4.1 Pengujian *Alpha*

Pengujian alpha adalah tahap awal dalam proses pengujian, di mana sistem diuji oleh tim pengembang internal. Tujuan dari pengujian ini adalah untuk menemukan dan memperbaiki *bug* sebelum sistem diperkenalkan kepada pengguna akhir. Pengujian *alpha* memungkinkan pengembang untuk melakukan evaluasi awal terhadap fungsionalitas dan kinerja sistem, sehingga dapat meningkatkan kualitas produk sebelum diluncurkan ke tahap berikutnya. Pada tahap pengujian ini peneliti menggabungkan 2 metode testing yakni sebagai berikut :

3.2.4.1.1 *White Box Testing*

White Box Testing adalah metode pengujian perangkat lunak yang berfokus pada pemeriksaan struktur internal, logika, dan alur kode program. Pada tahap pengujian alpha, *white box testing* dilakukan oleh tim pengembang untuk

memastikan bahwa setiap jalur kode, kondisi, dan fungsi dalam sistem *dashboard Business Intelligence (BI)* berjalan sesuai dengan yang diharapkan. Teknik ini meliputi pengujian *path*, *branch*, dan *statement* untuk mendeteksi kesalahan logika, bug tersembunyi, serta memastikan bahwa semua bagian kode telah diuji secara menyeluruh.

3.2.4.1.2 *Black Box Testing*

Black Box Testing adalah metode pengujian yang berfokus pada pengujian fungsi dan fitur sistem tanpa memperhatikan struktur internal atau kode program. Pada pengujian *alpha*, *black box testing* dilakukan untuk memastikan bahwa dashboard BI memenuhi kebutuhan fungsional yang telah ditentukan dan berjalan sesuai dengan

spesifikasi. Pengujian ini dilakukan dengan memberikan berbagai input dan memeriksa output yang dihasilkan, serta memastikan sistem dapat menangani berbagai kondisi penggunaan secara benar.

3.2.4.2 Pengujian *Beta*

Setelah pengujian *alpha*, sistem akan memasuki tahap pengujian *beta*, di mana pengguna akhir diundang untuk menguji sistem dalam lingkungan nyata. Pengujian beta bertujuan untuk mendapatkan umpan balik dari pengguna mengenai pengalaman mereka saat menggunakan sistem. Pengujian *beta* penting karena memberikan wawasan langsung dari pengguna yang sebenarnya, pengujian ini menggabungkan 2 metode yang mana keduanya termasuk ke dalam metode *black box testing* yakni :

3.2.4.2.1 SUS (*System Usability Scale*)

Salah satu alat yang umum digunakan dalam pengujian *beta* adalah *System Usability Scale* (SUS). SUS adalah kuesioner yang dirancang untuk mengukur kegunaan sistem dari perspektif pengguna. Kuesioner ini terdiri dari 10 pertanyaan yang mencakup berbagai aspek kegunaan, seperti kemudahan penggunaan, kepuasan, dan keefektifan sistem. SUS telah terbukti efektif dalam memberikan gambaran umum tentang pengalaman pengguna dan dapat digunakan untuk membandingkan kegunaan berbagai sistem.

Adapun alasan lain penggunaan SUS baik untuk mengukur aspek *usability* suatu produk atau layanan menurut Betteridge, K (2020) dalam Kesuma (2021), yaitu :

1. SUS dapat digunakan dengan mudah, karena hasilnya berupa angka skor 0 – 100 dan sudah tersedia template yang tinggal digunakan untuk melakukan proses perhitungannya.
2. SUS sangat mudah digunakan, tidak membutuhkan perhitungan yang rumit dan tidak memerlukan banyak sumber daya untuk mengelolanya.
3. SUS tersedia secara gratis, tidak membutuhkan biaya tambahan.

4. SUS terbukti valid dan reliable, walau dengan ukuran sampel yang kecil.
5. SUS dapat membantu penyedia produk atau layanan dalam mengevaluasi apakah suatu sistem perlu diperbarui.
6. SUS dapat membantu mengevaluasi efektivitas perbaikan sistem dari waktu ke waktu.
7. SUS dapat memberikan keyakinan kepada pemilik bisnis untuk berinvestasi lebih jauh disbanding user experience dari produk atau layanan yang dimilikinya.

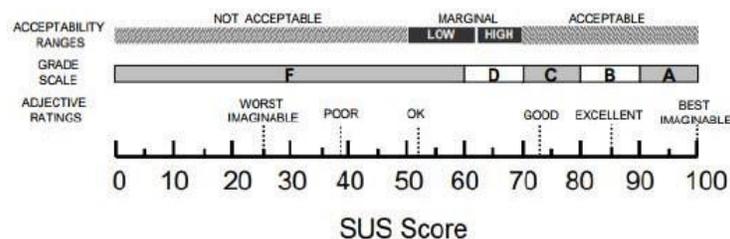
Dalam menghitung SUS terdapat aturan yang perlu diperhatikan, yakni :

1. Setiap pertanyaan dengan nomor ganjil, skor pertanyaan dikurangi 1.
2. Setiap pertanyaan dengan nomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan.
3. Skor SUS didapat dari hasil penjumlahan skor setiap pertanyaan dikali 2,5.

$$\bar{x} = \frac{\sum x}{n}$$

Selanjutnya setekah perhitungan selesai, akan didapatkan skor akhir yang berada di antara rentang 0 – 100. Jika nilai akhir sudah didapatkan, langkah terakhir yakni menentukan letak kelayakan media yang dibuat. Sebagai panduan berikut skala

acceptables range untuk dapat mengetahui kualitas sistem yang telah dibuat, apakah dapat diterima atau tidak.



Gambar 5 Skala *Acceptable Range* SUS Score

Sumber : (Kesuma, 2021) Adapun

ketentuan dari persentil rank SUS sebagai berikut:

1. Grade A : dengan skor lebih besar atau sama dengan 80,3.
2. Grade B : dengan skor lebih besar sama dengan 74 dan lebih kecil 80,3
3. Grade C : dengan skor lebih besar 68 dan lebih kecil 74
4. Grade D : dengan skor lebih besar sama dengan 51 dan lebih kecil 68
5. Grade F : dengan skor lebih kecil 51

3.2.4.2.2 *User Acceptance Testing* (UAT)

User Acceptance Testing (UAT) merupakan tahap pengujian yang dilakukan oleh pengguna akhir atau stakeholder untuk memastikan bahwa sistem *dashboard Business Intelligence* (BI) yang dikembangkan telah memenuhi kebutuhan dan ekspektasi operasional divisi TechOps di perusahaan satelit. Pada tahap ini, pengguna melakukan pengujian terhadap fitur-fitur utama *dashboard* dalam lingkungan yang menyerupai kondisi nyata penggunaan sehari-hari.

3.2.5 *Deployment*

Setelah pengujian berhasil, *dashboard* BI diluncurkan ke lingkungan operasional divisi TechOps untuk digunakan secara nyata dalam pemantauan kinerja.

3.2.6 *Review*

Sistem yang telah diluncurkan akan terus dipantau dan diperbarui sesuai dengan kebutuhan pengguna yang berubah. Pemeliharaan berkala diperlukan untuk memastikan bahwa sistem tetap berfungsi dengan baik dan dapat beradaptasi dengan perubahan lingkungan bisnis. Pemeliharaan yang efektif dapat memperpanjang umur sistem dan meningkatkan kepuasan pengguna.

3.3 Objek Penelitian

Penelitian ini dilakukan di salah satu perusahaan satelit di Indonesia, dengan penggunaan sampel yang dihitung menggunakan rumus Slovin dengan tingkat kesalahan (*error tolerance*) sebesar 5% (0,05). Perhitungan sampel dengan rumus Slovin adalah sebagai berikut:

$$n = \frac{N}{1 + N(e)^2}$$

n = jumlah sampel

N = jumlah populasi (42 karyawan)

e = tingkat kesalahan (0,05)

Berdasarkan rumus tersebut, perhitungan jumlah sampel adalah:

$$n = \frac{N}{1 + N(e)^2} = \frac{42}{1 + 42(0,05)^2} = \frac{42}{1 + 42(0,0025)} = \frac{42}{1 + 0,105} = \frac{42}{1,105} \approx 38,01$$

Dengan demikian, jumlah sampel yang dibutuhkan adalah 38 karyawan. Untuk memastikan representasi yang baik, peneliti memutuskan untuk mengambil 38 karyawan sebagai sampel penelitian.

3.4 Instrumen Penelitian

Dalam penelitian ini, instrumen yang digunakan untuk mengumpulkan data terdiri dari angket/kuesioner. Instrumen ini dirancang untuk mendapatkan informasi yang komprehensif mengenai penggunaan dan efektivitas *dashboard Business Intelligence* (BI) yang telah diimplementasikan. Penggunaan instrumen ini dapat

menjadi acuan validitas dan reliabilitas keberhasilan pengembangan, sehingga memberikan gambaran yang akurat tentang pengalaman pengguna dan tercapainya tujuan utama penelitian.

3.4.1 Angket/Kuisisioner

Angket atau kuesioner digunakan untuk mengukur *feedback* dari pengguna setelah implementasi *dashboard*. Kuesioner ini dirancang untuk mengevaluasi berbagai aspek, seperti kemudahan penggunaan, kepuasan, dan efektivitas dashboard dalam memenuhi kebutuhan pengguna. Angket merupakan alat yang efisien untuk mengumpulkan data dari sejumlah besar responden, sehingga memungkinkan peneliti untuk menganalisis pola dan tren dalam *feedback* pengguna. Dengan menggunakan angket, peneliti dapat mengidentifikasi area yang perlu diperbaiki dan mengukur dampak dari *dashboard* BI terhadap kinerja operasional.

Dalam pembuatan validasi pertanyaan angket, peneliti menggunakan acuan standar ISO 9241-110 yang memuat aspek-aspek kritikal kegunaan (*usability*) sistem, meliputi efektivitas, efisiensi, dan kepuasan pengguna. Pertanyaan dirancang untuk mengukur bagaimana *dashboard* BI memenuhi kebutuhan pengguna, apakah penggunaan sistem mudah dan intuitif, serta tingkat kepuasan yang dirasakan setelah menggunakan *dashboard*.

Pertanyaan dikelompokkan menjadi 5 aspek utama dengan standar ISO 9241-110 sebagai acuan yakni kemudahan pengguna, keakuratan data, visualisasi data, dan dukungan pengambilan keputusan.

3.4.2 Analisis Data

Setelah data dikumpulkan melalui angket, langkah selanjutnya adalah melakukan analisis data. Analisis data bertujuan untuk menginterpretasikan informasi yang telah diperoleh dan menarik kesimpulan yang relevan mengenai penggunaan *dashboard Business Intelligence* (BI). Analisis data yang sistematis sangat penting untuk memastikan bahwa hasil penelitian dapat diandalkan dan memberikan wawasan yang berguna bagi pengembangan lebih lanjut.

Dalam analisis data angket, peneliti akan menggunakan skala Likert untuk mengukur tingkat kepuasan pengguna terhadap berbagai aspek dashboard BI. Skala ini biasanya terdiri dari lima poin, yaitu :

Tabel 3.2 Kategori Skala Likert

| Skor | Kategori | Keterangan |
|------|---------------------|--|
| 1 | Sangat Tidak Setuju | Responden sangat tidak setuju dengan pernyataan yang diajukan. |
| 2 | Tidak Setuju | Responden tidak setuju dengan pernyataan yang diajukan. |
| 3 | Netral | Responden tidak memiliki pendapat yang jelas (tidak setuju maupun tidak setuju). |
| 4 | Setuju | Responden setuju dengan pernyataan yang diajukan. |
| 5 | Sangat Setuju | Responden sangat setuju dengan pernyataan yang diajukan. |

Dengan menggunakan skala ini, peneliti dapat menghitung skor rata-rata untuk setiap pertanyaan, yang akan memberikan gambaran umum tentang persepsi pengguna terhadap dashboard. Untuk mendapatkan rata-rata skor dari setiap item, bagi total skor dengan jumlah responden. Rumusnya adalah:

$$\text{Rata-rata Skor} = \frac{\text{Total Skor}}{\text{Jumlah Responden}}$$

Menurut Arikunto (2019), analisis data kuantitatif seperti ini memungkinkan peneliti untuk mengidentifikasi tren dan pola yang dapat digunakan untuk meningkatkan sistem yang ada.