

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Gambaran Umum Objek Penelitian**

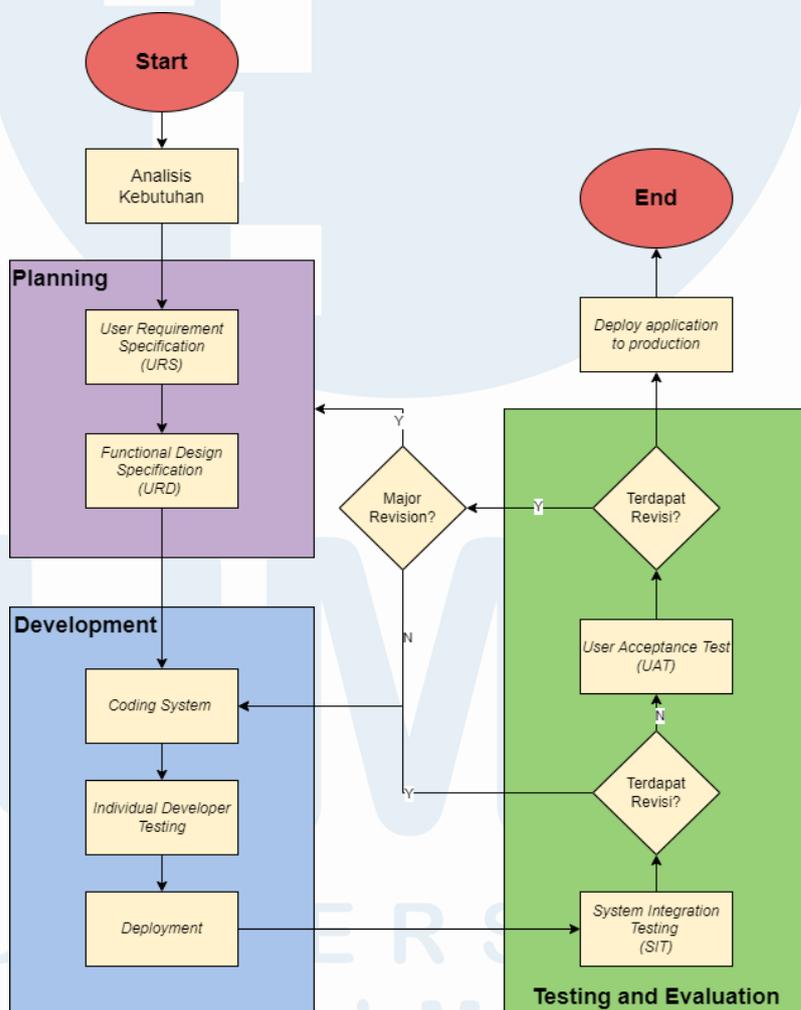
PT XYZ merupakan salah satu perusahaan farmasi terkemuka di Indonesia yang menjadi bagian integral dari industri kesehatan di Indonesia. PT XYZ telah membangun reputasi yang kuat sejak awal didirikan di Garut, Jawa Barat. Perusahaan ini bermula dengan sederhana, dengan memproduksi produk obat pertamanya untuk sakit kepala yang berhasil meredakan rasa sakit kepala dan menjadi salah satu obat terpopuler pada masanya. Dengan visi dan misi yang jelas serta komitmen terhadap kualitas, PT XYZ telah mengalami pertumbuhan yang signifikan seiring berjalannya waktu. Transformasi dari perusahaan kecil menjadi salah satu pemain utama dalam industri farmasi Indonesia terjadi seiring dengan perpindahan kantor pusat ke Jakarta, menandai langkah penting dalam perjalanan perusahaan.

PT XYZ lalu menjadi salah satu bagian dari keluarga salah satu grup perusahaan farmasi terbesar, hal ini membuka peluang baru PT XYZ untuk memperluas pangsa pasar di dalam dan luar negeri. Produk-produk nutraceutical dan *over-the-counter* yang diluncurkan berhasil meraih kesuksesan, tidak hanya di pasar domestik tetapi juga di pasar internasional dengan pengeksporannya ke berbagai negara seperti Filipina, Myanmar, Nigeria, Mongolia, Hong Kong, Sri Lanka, Kamboja, Yaman, Malaysia, India, dan Selandia Baru. Dengan upaya terus-menerus berinovasi, PT XYZ tetap setia pada akarnya dengan terus memproduksi produk obat sakit kepala pertamanya bersama dengan produk-produk lainnya yang meliputi minuman herbal, minuman energi, suplemen kesehatan dan obat-obatan lainnya, sehingga membangun portofolio perusahaan dan memperluas cakupan produknya.

Saat ini, PT XYZ telah menjadi salah satu penggerak utama dalam industri farmasi dengan jumlah karyawan sekitar 1.300 yang berkomitmen untuk terus berusaha untuk menghadirkan inovasi dalam produk obat dan suplemen kesehatan untuk masyarakat Indonesia, menegaskan posisinya sebagai salah satu perusahaan farmasi yang terpercaya dan terbesar di Indonesia.

### 3.2 Metode Penelitian

#### 3.2.1 Alur Penelitian



Gambar 3.1 Alur Penelitian

Alur Penelitian dapat dilihat dari gambar 3.1 diatas. Dimulai dengan dokumentasi semua *user requirement* yang diambil dari permasalahan dan solusi yang diinginkan oleh klien terhadap sistem *ERP Oracle* saat ini. *User requirement* tersebut akan dikembangkan kedalam bentuk URS yang berisi semua *requirement* dari klien untuk pengembangan sistem yang baru. Setelah itu, tahap berikutnya merupakan desain UML yang terdiri dari *use case* diagram yaitu diagram yang menggambarkan bagaimana *user (actor)* berinteraksi dengan sistem untuk mencapai tujuan tertentu, lalu diikuti dengan perancangan class diagram yang merupakan diagram gambaran dari semua table yang akan dibuat dalam *database* serta atribut-atribut dan relasi antar table ini. Tahap selanjutnya adalah perancangan *Entity Relationship Diagram* yang menggambarkan semua relasi antar attribute dalam *database* secara terstruktur yang akan masuk ke dalam dokumen FDS (*Functional Design Specification*).

Tahapan berikutnya adalah tahap pengembangan/development dengan proses coding *Backend* untuk merancang dan membangun API yang nantinya akan diberikan kepada rekan *Frontend* yang menyambung dengan hasil tampilan menggunakan Next.js. *Developer Frontend* dan *Backend* lalu melakukan *Testing* pribadi dari *localhost* pada sistem aplikasi yang telah dibuat. Ketika semua fitur dan *requirements* sudah dipenuhi maka sistem akan di *deploy* dalam *openshift* sebelum diberikan kepada *System Analyst* untuk memulai tahap *testing*.

Tahap berikutnya merupakan tahap *Testing* dan *Evaluation* yang dilakukan oleh *System Analyst* dalam bentuk SIT sistem jika terdapat bug atau penambahan fitur, akan dikembalikan kepada *developer*. Ketika hasil SIT sudah memuaskan, maka *Prototype* hasil sistem akan diberikan kepada *end-user*nya sendiri untuk dilakukan UAT. Jika terdapat perubahan maka sistem akan dikembangkan kembali oleh *developer*, namun ketika terdapat

perubahan dalam sistem yang mengakitbatkan fitur baru yang tidak ada dalam kedua URS dan FDS maka *developer* akan berdiskusi kembali dengan *System Analyst* terkait *requirement* sebelum melakukan revisi dari coding. Ketika sudah tidak ada kendala maka aplikasi sistem akan dimasukan dalam production (*go live*).

### 3.2.2 Perbandingan Metode Pengembangan Sistem

Dalam pengembangan apikasi, terdapat berbagai metodologi yang digunakan dalam pengembangan aplikasi tabel 3.1 dibawah merupakan perbandingan antara tiga metodologi yaitu *Agile*, *Waterfall*, dan *Prototype*

Tabel 3.1 Perbandingan Metode

<i>Comparisons</i>	<i>Agile</i>	<i>Waterfall</i>	<i>Prototype</i>
<i>Advantages</i>	<i>Agile</i> bersifat fleksibel sehingga dapat dilakukan perubahan secara berkala berdasarkan <i>feedback</i> dari klien. Proses yang dibagi ke dalam beberapa tahap dapat membantu tim untuk fokus pada pengembangan, pengujian, dan kolaborasi yang baik sehingga menghasilkan projek yang berkualitas tinggi.	Projek <i>waterfall</i> mudah dipahami dan dikelola dengan tahap-tahap yang jelas. Setiap tahap memerlukan dokumentasi yang lengkap sehingga hasil projek lebih mudah dipahami alur flow dan logic dalam codingan dan test	Klien dapat secara aktif langsung berkoordinasi dalam pembangunan projek sehingga <i>feedback</i> dari klien dapat diberikan lebih cepat yang akan mengurangi waktu pengembangan
<i>Disadvantages</i>	Tanggal <i>deadline</i> projek dapat bergerak berdasarkan <i>feedback</i> dan perubahan yang harus dilakukan sehingga projek dapat berbeda dengan perencanaan pada awalnya dikarenakan sifatnya yang	<i>Waterfall</i> tidak fleksibel sehingga perubahan-perubahan tidak dapat dilakukan secara langsung dan pengumpulan informasi pada tahap awal projek lebih sulit dikarenakan	Waktu pengembangan lebih lama dikarenakan diperlukan beberapa <i>Prototype</i> terlebih dahulu sebelum dapat dikembangkan menjadi produk final.

<i>Comparisons</i>	<i>Agile</i>	<i>Waterfall</i>	<i>Prototype</i>
	<p>fleksibel. <i>Agile</i> tidak mementingkan pada kelengkapan dokumen</p>	<p><i>waterfall</i> tidak dapat kembali pada tahap sebelumnya sehingga jika terjadi kesalahan, akan menyebabkan <i>delay</i> pada proyek</p>	<p>Hal ini dapat mengakibatkan kebingungan antar <i>developer</i> dengan klien dimana klien dapat menyukai satu fitur yang ada pada versi awal <i>Prototype</i> namun oleh <i>developer</i> fitur tersebut dapat dihapuskan dengan fitur yang lain sehingga hasil produk akhir tidak tentu akan sama dengan keinginan <i>user</i></p>

Berdasarkan perbandingan ketiga metodologi, metodologi *Agile* terpilih untuk pengembangan aplikasi *E-Smart Sparepart* Bukti Pemakaian Part Mesin. Faktor yang mempengaruhi pemilihan ini dilihat dari sifat *Agile* yang fleksibel dimana pengembangan aplikasi dapat disesuaikan secara berkala sesuai dengan permintaan *user*/klien. Hal yang tidak dapat dilakukan dengan metodologi *waterfall*. Selain itu, *Agile* memfokuskan pada pembuatan sistem yang terus berkembang seiring waktu bukan produk final sempurna dan membangun kolaborasi antar klien dengan *developer* sehingga mempercepat waktu pengembangan. Hal ini tidak dapat dilakukan dengan metodologi *Prototype*.

### 3.2.3 Perbandingan Arsitektur Pengembangan Aplikasi

Dalam pengembangan aplikasi terdapat dua arsitektur perangkat lunak utama yang menggambarkan bagaimana suatu aplikasi dibangun yaitu *monolithic* dan *Micro Services*. Tabel 3.2 dibawah merupakan perbandingan antar dua arsitektur ini:

Tabel 3.2 Perbandingan Monolithic dan Micro-Services

<i>Comparisons</i>	<i>Monolithic</i>	<i>Micro Services</i>
<i>Advantages</i>	Dengan arsitektur <i>Monolithic developer</i> hanya memerlukan 1 <i>project executable</i> yang memudahkan proses development, deployment, <i>testing</i> , dan debugging dikarenakan semua <i>code</i> untuk aplikasi serta fitur-fiturnya terdapat pada satu proyek saja.	Dengan arsitektur <i>micro services developer</i> membuat beberapa <i>project executable</i> untuk setiap modul atau fitur dalam aplikasi. Hal ini memudahkan <i>developer</i> untuk melakukan <i>unit testing</i> atau <i>testing</i> setiap modul tersendiri. Ketika terjadi masalah atau bug dalam salah satu modul, modul lainnya tidak akan terpengaruh sehingga keseluruhan aplikasi tidak perlu dikembalikan kepada <i>development</i> . Dikarenakan sifatnya yang modular, <i>micro services</i> memudahkan <i>developer</i> dalam terus menambah fitur-fitur dan modul baru sehingga cocok untuk proyek besar.
<i>Disadvantages</i>	Ketika terjadi masalah dalam satu sistem atau bug, keseluruhan aplikasi harus dikembalikan pada tahap <i>deployment</i> sehingga tidak dapat diandalkan untuk proyek-proyek besar. Selain itu, <i>Monolithic</i> mempunyai kapasitas skala yang terbatas sehingga fitur-fitur dan pengembangan modul kedepannya akan lebih sulit dan <i>maintenance</i> dari aplikasi akan semakin kompleks.	<i>Micro services</i> secara arsitekturnya sendiri lebih kompleks dibanding <i>monolithic</i> sehingga progress awal development akan lebih lambat. Dikarenakan semua fitur atau modul berada pada <i>project executable</i> yang berbeda, setiap modul mempunyai standar masing-masing untuk <i>logic</i> , <i>logging</i> , dan <i>monitoring</i> .

Berdasarkan tabel 3.2 diatas, *Micro Services* terpilih untuk pengembangan aplikasi *E-Smart Sparepart* Bukti Pemakaian Part Mesin. Faktor yang mempengaruhi pemilihan ini adalah kemampuan *micro services* untuk pengembangan selanjutnya sehingga setelah aplikasi sudah *deploy* dalam

*production*, aplikasi *E-Smart Sparepart* Bukti Pemakaian Part Mesin dapat terus ditingkatkan.

### **3.3 Teknik Pengumpulan Data**

Penelitian ini menggunakan 2 jenis teknik pengumpulan data sebagai berikut:

#### **3.3.1 Studi Pustaka**

Studi Pustaka dilakukan dalam penelitian ini untuk membantu dalam pencarian referensi jurnal, buku, dan artikel penelitian sebelumnya yang diambil dari Google Scholar, Sinta, Scimago, dan ReasearchGate. Semua informasi dalam referensi-referensi ini akan digunakan untuk membantu dalam rancang bangun sistem aplikasi smart *sparepart* BPPM dari awal sampai akhir serta memberikan bantuan dalam pembuatan penulisan penelitian ini.

#### **3.3.2 Dokumentasi**

Dokumentasi merupakan salah satu cara mengumpulkan data diambil dalam bentuk dokumen atau buku yang berisi semua informasi dan keterangan yang mendukung penelitian. Data seperti keperluan business *operational*, *user requirements*, *data flow*, serta *testing requirements* diberikan dalam bentuk URS (*User Requirement Specification*) dan FDS (*Functional Design Specification*) dari hasil wawancara *System Analyst* dengan klien. Kedua Dokumen ini akan digunakan sebagai gambaran perancangan sistem yang akan dibangun, *requirement* apa saja yang harus dilakukan, serta deskripsi *testing* yang akan dilakukan setelah sistem aplikasi siap diberikan kepada klien. Dokumen FDS dan URS diberikan pada tanggal 4 September 2023 oleh *System Analyst Business Operation*.

### 3.4 Metode Evaluasi Sistem

#### 3.4.1 *System Integration testing (SIT)*

*System Integration testing* atau SIT merupakan proses QA dimana hasil sistem diuji oleh tim *developer* dan *System Analyst* untuk menilai sistem secara keseluruhan. Tujuan dari pengujian ini adalah untuk mengetahui apakah aplikasi dapat memenuhi *user requirements* yang sudah ditentukan pada awalnya. Pengujian ini meliputi semua modul dan fitur untuk mengetahui apakah semua modul dan fitur ini bekerja dengan baik tanpa ada bug atau masalah. Ketika hasil pengujian sudah memuaskan, sistem diberikan kepada *end user* untuk melakukan *User Acceptance Testing (UAT)*.

#### 3.4.2 *UAT*

*User Acceptance Testing* atau UAT merupakan tahap dalam pengembangan sistem aplikasi dimana dilakukan *testing* oleh *end-user*/pengguna yang bertujuan untuk mengecek apakah sistem yang sudah dibangun dapat menyelesaikan kebutuhan-kebutuhan bisnis yang sudah ditentukan sebelumnya. Pengguna sistem E-SS BPM akan menguji semua fitur sistem dimulai dari fitur Login, view *data sparepart* BPPM, *edit data sparepart*, melihat *catalog sparepart* yang dapat diambil, masukan *item* BPPM ke dalam *cart*, melakukan *issue* pada *sparepart* dan *delete issue*, dan monitoring *transaction* setelah *issue*. *Testing* dan validasi sistem akan sesuai dengan ketentuan validasi sistem komputer untuk *custom application* GAMP5 kategori

5.