

BAB III

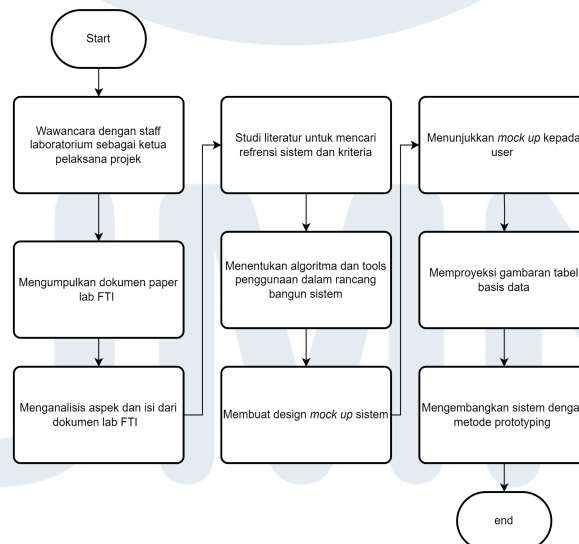
METODOLOGI PENELITIAN

3.1 Objek Penelitian

Objek penelitian ini berfokus pada administrasi laboratorium FTI UMN. Secara garis besar, laboratorium FTI UMN melakukan migrasi pencatatan permintaan instalasi dan proses pendaftaran maupun penjadwalan asisten laboratorium kedalam bentuk satu wadah sistem one stop solution berbasis web. Saat ini, informasi-informasi terkait pencatatan permintaan instalasi dan pendaftaran maupun penjadwalan asisten laboratorium masih belum tersimpan secara terorganisir dan menyeluruh dengan baik. Hal ini dikarenakan belum adanya sistem penampung yang menyimpan fitur yang diperlukan, sehingga diperlukan berbagai alat perangkat lunak yang berbeda-beda untuk menyesuaikan kebutuhan. Dengan hal ini, diharapkan melalui perancangan aplikasi *one stop solution* pada penelitian ini dapat menunjang aktivitas administrasi laboratorium FTI UMN.

3.2 Metode Penelitian

3.2.1 Alur Penelitian



Gambar 3. 1 Alur Penelitian

Tahapan-tahapan yang diuraikan merupakan proses yang sistematis dan berkesinambungan dalam mengembangkan sebuah sistem laboratorium di FTI (Fakultas Teknologi Industri). Pertama-tama, wawancara dengan staf laboratorium yaitu Rio Raymundus Theodora selaku ketua pelaksana proyek dilakukan untuk memperoleh pemahaman yang mendalam tentang kebutuhan dan tantangan yang dihadapi dalam pengelolaan laboratorium. Kemudian dilakukan mengumpulkan dokumen kertas laboratorium yang ada di FTI untuk menganalisis aspek dan isi dari dokumen tersebut. Setelah itu, dilakukan studi literatur untuk mencari referensi sistem dan kriteria yang relevan dengan kebutuhan laboratorium. Berdasarkan hasil analisis dan studi literatur, algoritma dan tools yang sesuai dipilih untuk digunakan dalam rancang bangun sistem laboratorium.

Selanjutnya, tahapan berikutnya adalah membuat desain mock up sistem yang akan dikembangkan. Desain mock up ini kemudian ditunjukkan kepada pengguna (staf laboratorium) untuk mendapatkan masukan dan validasi terhadap fitur-fitur yang dirancang. Setelah mendapatkan umpan balik dari pengguna, gambaran fitur sistem dan tabel basis data diproyeksikan untuk memberikan visi yang jelas tentang bagaimana sistem akan berfungsi dan bagaimana data akan diatur.

Langkah terakhir adalah mengembangkan sistem menggunakan metode *prototype*. Dalam proses pengembangan ini, iterasi dilakukan secara berulang untuk memastikan bahwa sistem yang dibangun sesuai dengan kebutuhan dan harapan pengguna. Dengan demikian, tahapan-tahapan tersebut membentuk alur penelitian yang berkelanjutan dan memastikan bahwa pengembangan sistem laboratorium di FTI dilakukan dengan efisien dan efektif.

3.2.2 Perbandingan Metode Penelitian

SDLC (Systems Development Life Cycle) merupakan metode yang dapat menggambarkan kerangka kerja dalam proses pembuatan hingga modifikasi suatu sistem dalam ranah pengembangan perangkat lunak, pemilihan model

pengembangan yang tepat menjadi langkah krusial untuk mencapai keberhasilan suatu proyek [29]. Tiga model yang sering digunakan dalam proyek kecil-menengah adalah model *waterfall*, *prototype*, dan *Rapid Application Development (RAD)*. Model *waterfall* menetapkan proses pengembangan secara linier, seperti aliran air yang mengalir turun melalui tahapan perencanaan, pemodelan, implementasi, pengujian, hingga pemeliharaan. Model ini memberikan kejelasan structural namun memiliki keterbatasan dalam fleksibilitas terhadap perubahan yang mungkin diperlukan sepanjang jalannya proyek. Sebaliknya, model *prototype* menyajikan pendekatan yang berfokus pada interaksi aktif dengan pengguna. Dengan menciptakan sistem yang dapat diuji oleh pengguna, model ini memungkinkan perbaikan berkelanjutan berdasarkan umpan balik pengguna [11]. Kecepatan pengembangan yang relatif tinggi dan kemampuan adaptasi terhadap perubahan kebutuhan pengguna menjadi keunggulan utama *prototype*. Hal ini memberikan responsibilitas yang lebih baik terhadap dinamika yang mungkin terjadi selama siklus pengembangan. Sementara itu, *Rapid Application Development (RAD)* menonjolkan percepatan dalam durasi pembuatan sistem. Walaupun menawarkan pengembangan yang lebih cepat dan keterlibatan aktif pengguna, model ini memiliki kekurangan dalam struktur yang mungkin mengakibatkan kesulitan pemeliharaan sistem dalam jangka panjang.

Dalam evaluasi ketiga model ini, *prototype* sering kali dianggap sebagai pilihan yang lebih efektif dan responsif. Fleksibilitas tinggi dan kemampuan adaptasi terhadap perubahan kebutuhan pengguna meminimalkan risiko pengembangan [13]. Kecepatan pengembangan yang relatif tinggi dan iterasi berulang memungkinkan deteksi dini kesalahan serta perbaikan yang cepat. Dengan demikian, dalam konteks penelitian ini yang menitikberatkan pada pengembangan sistem, model *prototype* menjadi pendekatan yang lebih menguntungkan, memastikan bahwa hasil akhir sesuai dengan harapan dan kebutuhan pengguna.

Tabel 3. 1 Perbandingan metode SDLC

Fitur	Waterfall	RAD	Prototyping
<i>System planning</i>	Perencanaan proyek dan sumber daya yang detail	Perencanaan tingkat tinggi dengan kemampuan adaptasi terhadap perubahan	Berfokus pada identifikasi kebutuhan dan kendala utama untuk <i>prototype</i>
<i>System analysis</i>	Pengumpulan kebutuhan yang komprehensif di awal	Penyempurnaan kebutuhan secara terus menerus melalui iterasi	Analisis eksploratif untuk menentukan ruang lingkup prototipe
<i>System design</i>	Desain lengkap sebelum pengembangan	Desain iteratif, disempurnakan berdasarkan umpan balik	Desain berfokus pada fungsi prototipe dan antarmuka pengguna
<i>System implementation</i>	Fase pengembangan dan pengujian berurutan	Pengembangan cepat prototipe yang berfungsi dalam siklus	Prototyping dan iterasi sampai kebutuhan divalidasi
<i>System maintenance</i>	Fleksibilitas terbatas untuk perubahan setelah selesai	Pemeliharaan lebih mudah karena desain modular dan keterlibatan pengguna yang berkelanjutan	Pemeliharaan produk akhir tergantung pada model pengembangan yang digunakan

3.2.3 Tahap Pengembangan Sistem

Pengembangan sistem yang sesuai dengan penelitian ini adalah *prototype*. Hal ini dikarenakan penelitian ini memiliki skala pengerjaan yang umumnya dikerjakan dengan *prototype* dan perubahan yang dimiliki tetap sistematis dengan kurun waktu yang tidak terlalu cepat. Berikut tahapan-tahapan yang akan dilakukan dari perancangan aplikasi menggunakan metode *prototype* pada penelitian ini yakni antara lain [13]:

1. Pengumpulan dan Analisis Persyaratan / *Requirement Gathering and Analysis:*

Pada tahap awal ini, tim pengembangan bekerja sama dengan pengguna dan pemangku kepentingan untuk mengumpulkan dan menganalisis persyaratan sistem. Tujuannya adalah memahami dengan jelas tujuan dan cakupan proyek.

2. Desain Cepat / *Quick Design:*

Setelah memahami persyaratan, dilakukan desain cepat untuk memberikan gambaran dasar tentang bagaimana sistem tersebut akan terlihat dan berfungsi. Desain ini tidak terlalu rinci tetapi berfungsi sebagai representasi visual untuk memberikan pemahaman cepat tentang solusi yang diusulkan.

3. Membangun Prototipe / *Build a Prototype:*

Menggunakan desain cepat sebagai referensi, tim pengembangan membangun prototipe aktual dari sistem. Prototipe adalah model fungsional yang mungkin tidak memiliki semua fitur produk final tetapi menunjukkan fungsionalitas inti.

4. Evaluasi Awal Pengguna / *Initial User Evaluation:*

Desain *prototype* disajikan kepada pengguna untuk pengujian dan evaluasi awal. Pengguna memberikan umpan balik tentang kelebihan dan kelemahan prototipe, memungkinkan pengembang memahami sejauh mana desain awal memenuhi harapan pengguna.

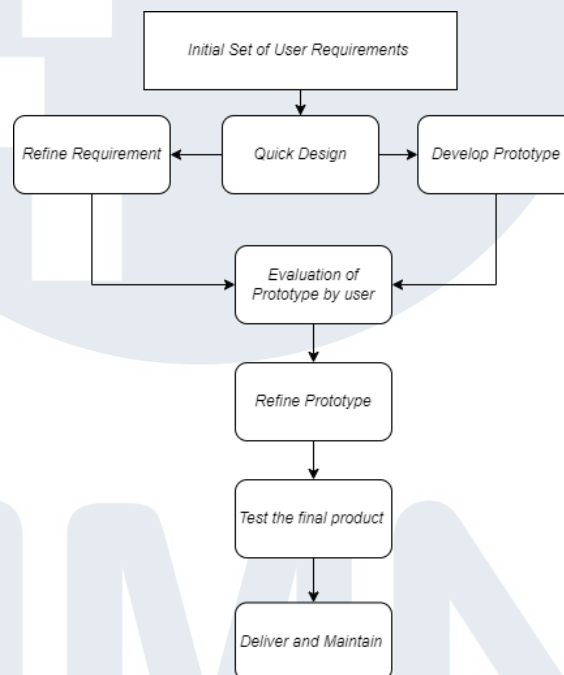
5. Penyempurnaan Prototipe / *Refining Prototype:*

Berdasarkan umpan balik pengguna, prototipe disempurnakan dan diperbaiki. Perubahan dilakukan untuk menanggapi masalah, meningkatkan fungsionalitas, dan

menggabungkan fitur atau modifikasi tambahan yang diusulkan oleh pengguna.

6. Implementasi Produk dan Pemeliharaan / *Implement Product and Maintain:*

Setelah prototipe disempurnakan dan disetujui, sistem final diimplementasikan dan dikerahkan untuk produksi. Pemeliharaan rutin dilakukan untuk memastikan kinerja sistem yang berkelanjutan dan menangani masalah yang mungkin timbul. Proses pengembangan dapat berlanjut dengan iterasi atau pembaruan lebih lanjut berdasarkan kritik saran pengguna dan perubahan persyaratan.



Gambar 3. 2 Tahap Prototype

3.3 Pengumpulan dan Analisis Persyaratan / *Requirement Gathering and Analysis*

Penelitian ini menggunakan teknik pengumpulan data campuran yang berarti menggunakan kedua pendekatan yaitu kualitatif dan kuantitatif. Dengan pendekatan kualitatif yaitu dengan metode studi pustaka dan

wawancara sebagai tahapan awal pengumpulan data penelitian/*preliminary research*.

3.3.1 Studi Pustaka

Studi pustaka dilakukan untuk mengumpulkan data pendukung dengan mencari jurnal penelitian terdahulu yang dapat menjadi acuan untuk melakukan penelitian rancang bangun *framework one stop solution* berbasis web laboratorium FTI UMN dengan menggunakan bahasa pemrograman PHP dan SDLC sebagai pendukung siklus perancangan pengembangan aplikasi.

3.3.2 Wawancara

Penelitian ini juga melibatkan wawancara untuk mengumpulkan data yang diperlukan dalam pengembangan aplikasi. Pertemuan tersebut dihadiri oleh individu terkait, yaitu ketua pelaksana proyek yang akan dikembangkan juga sebagai perwakilan pegawai laboratorium FTI. Tujuan dari pertemuan adalah untuk menetapkan konsep dan kebutuhan, serta berbagi perkembangan pekerjaan dari setiap tim. Wawancara dilaksanakan selama 15 menit. Hal ini bertujuan agar penelitian ini dapat berjalan sesuai dengan perencanaan yang telah dibuat dan memenuhi kebutuhan administrasi laboratorium FTI UMN.

3.4 Desain Cepat / *Quick Design*

Desain cepat dengan pemanfaatan wireframe dalam metode *prototype* menekankan empat aspek krusial untuk memastikan kesuksesan pengembangan prototipe. Pertama, penekanan utama terletak pada identifikasi fungsionalitas inti dari antarmuka pengguna atau aplikasi. Wireframe digunakan untuk mengidentifikasi dan menyoroti fitur-fitur utama yang harus ada dalam prototipe, memastikan bahwa pengembangan

berfokus pada memenuhi kebutuhan pengguna dan memberikan nilai tambah yang diinginkan [16].

Aspek kedua menitikberatkan pada sederhana dan kejelasan dalam desain wireframe. Wireframe dirancang agar menyajikan antarmuka dengan cara yang jelas dan dapat dimengerti oleh berbagai pemangku kepentingan, termasuk pengembang dan pengguna. Menghindari kompleksitas dalam detail desain membantu menjaga fokus pada disposisi elemen, navigasi, dan interaksi dasar, sehingga pemahaman terhadap wireframe seragam di kalangan tim pengembangan dan pengguna.

Aspek ketiga yang penting adalah perancangan navigasi dasar. Desainer wireframe perlu mempertimbangkan struktur dasar navigasi antarmuka, memikirkan bagaimana pengguna akan berpindah antar halaman atau fitur. Ini memastikan bahwa pengalaman pengguna dalam menggunakan prototipe menjadi lebih intuitif dan efisien.

Keempat, penggunaan placeholder untuk konten berperan krusial dalam merancang wireframe. Penempatan konten semacam ini mencakup penggunaan placeholder untuk teks, gambar, dan data lainnya, memvisualisasikan bagaimana antarmuka mendukung konten yang akan ditampilkan. Ini memungkinkan pemangku kepentingan untuk memahami secara konkret bagaimana prototipe akan berinteraksi dengan konten aktual.

Penggunaan *UML*, *use case*, dan *activity diagram* melibatkan suatu esensialitas dalam mengembangkan wireframe yang baik. UML, dengan use case diagram, membantu identifikasi kebutuhan pengguna yang esensial untuk tercermin dalam wireframe. Sementara itu, *activity diagram* membantu merencanakan alur kerja dan navigasi antarmuka, menciptakan pemahaman mendalam tentang tata letak dan interaksi elemen-elemen. Kombinasi ketiganya membentuk fondasi yang kokoh untuk pengembangan wireframe yang mengakomodasi kebutuhan pengguna dengan optimal dan memudahkan proses iterasi dan perbaikan. Dengan demikian, pemanfaatan

UML, use case, dan activity diagram dalam proses desain cepat menjadi krusial untuk mencapai hasil yang berkualitas.

3.5 Membangun Prototipe / *Build a Prototype*

Saat kita membangun prototipe dalam konteks pengembangan *web* dengan menggunakan bahasa PHP dan *framework* CodeIgniter, dua poin utama yang perlu diperhatikan adalah integrasi fungsionalitas utama dan ketepatan representasi *wireframe*. Dalam mengimplementasikan fungsionalitas utama, kita dapat menggunakan fitur-fitur khas PHP untuk memastikan integrasi yang efektif. Sebagai contoh, kita dapat menggunakan model-model CodeIgniter untuk mengelola akses *database* dan mengambil data yang diperlukan untuk membangun prototipe. Selain itu, penggunaan kontroler dalam CodeIgniter dapat membantu menyusun logika bisnis yang terkait dengan fungsionalitas tersebut[29].

Selain itu, ketepatan representasi *wireframe* harus menjadi fokus utama dalam pengembangan dengan CodeIgniter. Framework ini memungkinkan kita untuk memisahkan logika presentasi dan bisnis dengan baik. Dengan menggunakan fitur *templating* CodeIgniter, kita dapat dengan mudah menyusun tata letak dan struktur halaman sesuai dengan representasi *wireframe*. Pastikan bahwa setiap elemen dan interaksi yang dijelaskan dalam *wireframe* terwujud dengan baik dalam tampilan yang dibangun dengan CodeIgniter.

Dengan menggabungkan kedua aspek ini, pengembang dapat menciptakan prototipe *web* yang memadukan kehandalan fungsionalitas PHP dengan struktur kerja yang efisien dari *framework* CodeIgniter. Hal ini memungkinkan prototipe yang sesuai dengan harapan pengguna dan mempermudah proses iterasi serta peningkatan berkelanjutan berdasarkan umpan balik yang diterima dari pengguna dan pemangku kepentingan.

3.6 Evaluasi Awal Pengguna / *Initial User Evaluation*

Dalam fase *prototype*, evaluasi awal pengguna (*Initial User Evaluation*) memainkan peran yang sangat penting dalam menilai kesesuaian prototipe dengan harapan pengguna. Pada tahap ini, pengguna berpartisipasi dalam interaksi awal dengan prototipe, memberikan umpan balik mengenai fungsionalitas, antarmuka, dan kemudahan penggunaan. Penelitian sebelumnya menunjukkan bahwa evaluasi pengguna pada tahap awal memungkinkan pengembang untuk mengidentifikasi secara efektif kelebihan dan kelemahan prototipe. Metode evaluasi melibatkan sesi pengujian langsung, wawancara, atau pengisian kuesioner oleh para pengguna. Hasil dari evaluasi ini menjadi dasar untuk perbaikan segera, memastikan bahwa pengembangan selanjutnya dapat memenuhi harapan pengguna. Dengan memanfaatkan data kualitatif dan kuantitatif dari evaluasi awal, tim pengembangan dapat merinci langkah-langkah perbaikan dan iterasi selanjutnya, membentuk landasan yang kuat untuk pengembangan prototipe yang sukses.

3.7 Penyempurnaan Prototipe / *Refining Prototype*

Dalam tahapan Penyempurnaan Prototipe (*Refining Prototype*) dalam proses *prototype*, fokus utama terletak pada debug dan penyempurnaan untuk memastikan bahwa prototipe mencapai tingkat kualitas yang optimal. Penelitian sebelumnya menunjukkan bahwa proses debug merupakan langkah kritis untuk mengidentifikasi dan memperbaiki potensi kesalahan atau bug yang mungkin muncul selama pengembangan prototipe. Metode penyempurnaan juga melibatkan evaluasi lebih lanjut terhadap fungsionalitas dan respons prototipe oleh pengguna. Pengumpulan umpan balik dari pengguna dan pemangku kepentingan membantu mengidentifikasi area-area yang memerlukan peningkatan atau penyesuaian. Selain itu, analisis kualitatif dan kuantitatif dari performa prototipe memungkinkan tim pengembangan untuk membuat keputusan yang terinformasi dalam merinci perbaikan dan iterasi. Penekanan pada penyempurnaan prototipe bukan hanya pada aspek teknis, melainkan juga

pada aspek pengalaman pengguna secara keseluruhan, sehingga memastikan bahwa prototipe tidak hanya bekerja dengan baik tetapi juga memenuhi harapan dan kebutuhan pengguna.

3.8 Implementasi Produk dan Pemeliharaan / *Implement Product and Maintain*

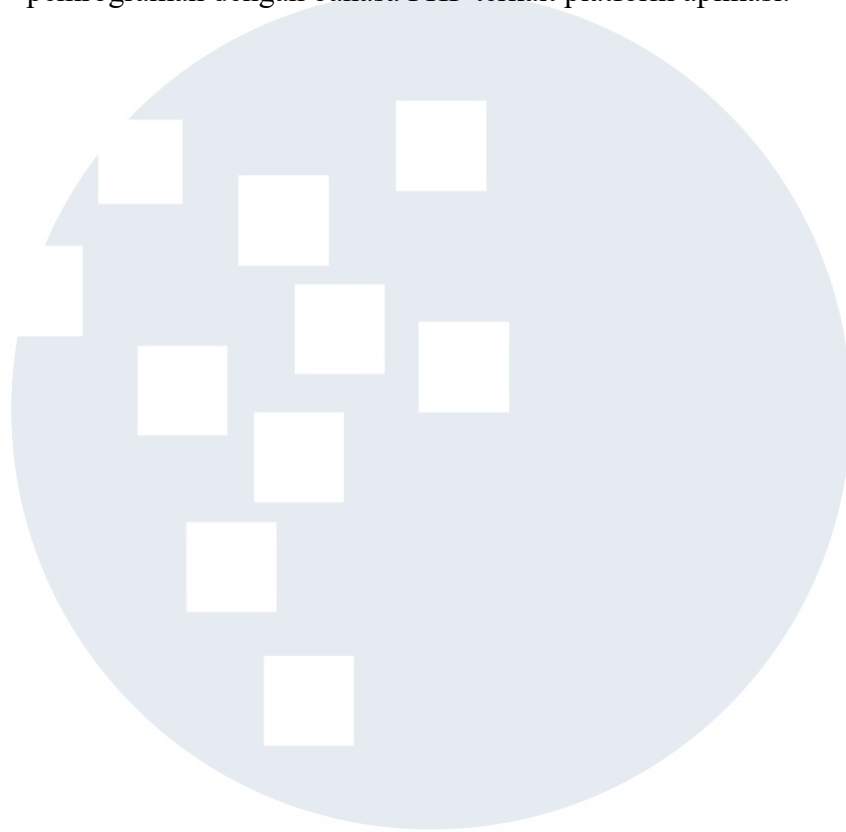
Dalam tahapan Implementasi Produk dan Pemeliharaan (*Implement Product and Maintain*) pada proses *prototype*, terdapat pergeseran fokus menuju penyerahan aplikasi ke Lab FTI untuk pengembangan lebih lanjut. Penelitian sebelumnya menunjukkan bahwa saat memasuki tahap ini, prototipe yang telah disempurnakan dan disetujui dapat diintegrasikan ke dalam lingkungan produksi dan diserahkan kepada tim pengembang lab FTI. Sementara itu, permintaan *feedback* dari pengguna selama implementasi menjadi kunci untuk terus meningkatkan kualitas dan cara kerja. Langkah ini melibatkan proses distribusi aplikasi, serta implementasi rutin untuk mencegah kegagalan dan memastikan kinerja sistem yang berkelanjutan. Pemeliharaan yang teratur diperlukan untuk memonitor dan menanggapi potensi kegagalan atau perubahan kebutuhan. Proses ini merupakan langkah kritis dalam memastikan bahwa aplikasi yang diserahkan tidak hanya memenuhi standar kualitas, tetapi juga dapat dikembangkan lebih lanjut sesuai dengan perkembangan kebutuhan dan teknologi.

3.9 Tools Penelitian

Terdapat beberapa alat perangkat lunak atau *software* pendukung yang digunakan untuk proses perancangan platform *framework one stop solution* berbasis web terkait administrasi lab FTI UMN yakni antara lain:

1. Perangkat lunak Google *tools*, seperti *google docs* dan *google form*, yang berguna untuk mendukung produktivitas jalannya penelitian secara internal maupun eksternal.
2. Perangkat lunak Draw.io untuk membuat rancangan basis data maupun UML terkait platform aplikasi.

3. Perangkat lunak Figma digunakan untuk membuat gambaran kasar antar muka pengguna dan *wireframe*.
4. Perangkat lunak Visual Studio Code digunakan untuk melakukan pemrograman dengan bahasa PHP terkait platform aplikasi.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA