

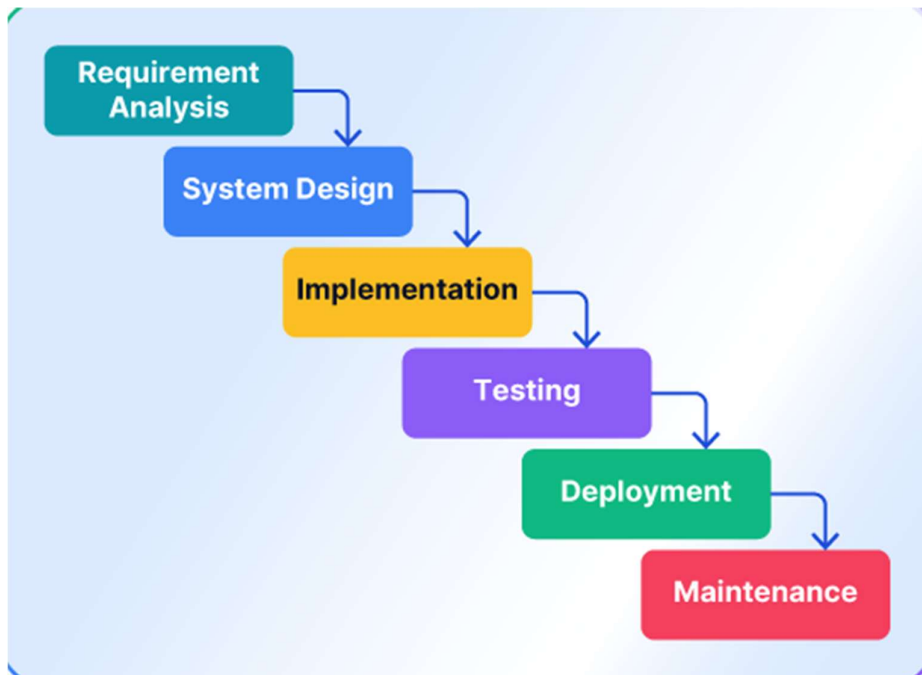
BAB III

METODOLOGI PENELITIAN

3.1 Prosedur Penelitian

Prosedur penelitian ini disusun secara sistematis untuk memastikan proses pengembangan sistem berjalan terarah, terstruktur, dan menghasilkan sistem yang sesuai dengan kebutuhan pengguna. Penelitian ini menggunakan metode pengembangan perangkat lunak Waterfall, yaitu model yang bersifat linear dan berurutan, di mana setiap tahapan harus diselesaikan sebelum melanjutkan ke tahap berikutnya [14].

Adapun tahapan dalam metode Waterfall adalah sebagai berikut:



Gambar 3. 1 Metode Waterfall

3.3.1 Requirements Analysis (Analisis Kebutuhan)

Pada tahap ini dilakukan proses analisis kebutuhan sistem secara menyeluruh berdasarkan hasil observasi dan wawancara yang dilakukan di SMK Ma'arif Talang. Analisis ini bertujuan untuk mengidentifikasi permasalahan pada sistem yang sedang berjalan serta menentukan kebutuhan sistem yang akan dikembangkan.

Berdasarkan hasil observasi, diketahui bahwa proses peminjaman alat bengkel masih dilakukan secara manual menggunakan buku catatan sebagai media pencatatan. Dalam proses tersebut, Admin mencatat setiap transaksi peminjaman dan pengembalian alat secara tertulis, termasuk data pengguna, data alat, serta tanggal transaksi.

Meskipun metode ini telah digunakan dalam jangka waktu yang cukup lama, sistem manual tersebut memiliki beberapa keterbatasan yang berdampak pada efektivitas dan efisiensi pengelolaan data. Adapun permasalahan yang ditemukan adalah sebagai berikut:

1. Tingginya potensi kesalahan pencatatan (*human error*)

Proses pencatatan manual rentan terhadap kesalahan, seperti penulisan data yang tidak lengkap, kesalahan input, serta tulisan yang sulit dibaca, sehingga dapat menyebabkan ketidaksesuaian informasi.

2. Kesulitan dalam pencarian dan pengolahan data

Data yang tersimpan dalam bentuk buku catatan mengharuskan Admin mencari data secara manual dengan membuka halaman satu per satu, sehingga membutuhkan waktu yang lama terutama ketika jumlah data semakin banyak.

3. Tidak adanya sistem monitoring stok secara *real-time*

Admin tidak dapat mengetahui jumlah stok alat bengkel secara langsung karena harus melakukan pengecekan manual, sehingga berpotensi terjadi kesalahan dalam proses peminjaman.

4. Risiko kehilangan dan kerusakan data

Data yang disimpan dalam bentuk fisik sangat rentan terhadap kerusakan, seperti sobek atau hilang, sehingga dapat mengakibatkan hilangnya data penting.

5. Tidak adanya sistem pembatasan hak akses (*access control*)

Seluruh proses pengelolaan data masih bergantung pada Admin tanpa adanya sistem kontrol berbasis teknologi, sehingga kurang efisien dan berisiko terhadap keamanan data.

Berdasarkan permasalahan tersebut, diperlukan suatu sistem informasi berbasis *website* yang mampu mengelola proses peminjaman alat bengkel secara terkomputerisasi, terstruktur, serta mampu meningkatkan efisiensi dan akurasi pengelolaan data..

Dalam sistem yang akan dikembangkan, terdapat tiga aktor utama yang berinteraksi dengan sistem, yaitu:

1. SuperAdmin

SuperAdmin merupakan pengguna yang memiliki hak akses tertinggi dalam sistem. SuperAdmin bertanggung jawab dalam mengelola data Admin, data siswa, melakukan unggah data siswa berdasarkan NIS, serta mengawasi seluruh aktivitas sistem.

2. Admin

Admin merupakan pengguna yang bertanggung jawab dalam pengelolaan operasional sistem. Admin memiliki hak akses untuk mengelola data alat bengkel, memproses transaksi peminjaman dan pengembalian alat, mengelola inventaris, mengunggah materi pembelajaran, serta membuat laporan peminjaman.

3. *User* (Siswa)

User merupakan pengguna sistem yang memiliki hak akses terbatas, yaitu melihat informasi alat bengkel yang tersedia, melakukan peminjaman alat, melakukan pengembalian alat, serta mengakses materi pembelajaran yang disediakan oleh Admin.

A. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berkaitan dengan layanan atau fitur yang harus disediakan oleh sistem. Kebutuhan ini akan menjadi dasar dalam

penyusunan *use case diagram* pada tahap perancangan sistem.

Adapun kebutuhan fungsional sistem adalah sebagai berikut:

1. Kebutuhan fungsional SuperAdmin:

- 1) SuperAdmin dapat melakukan *login* ke dalam sistem menggunakan akun yang telah terdaftar
- 2) SuperAdmin dapat mengelola data Admin
- 3) SuperAdmin dapat mengelola data siswa
- 4) SuperAdmin dapat melakukan upload data siswa berdasarkan NIS
- 5) SuperAdmin dapat melihat informasi data pengguna sistem
- 6) SuperAdmin dapat melakukan logout dari sistem.

2. Kebutuhan fungsional Admin:

- 1) Admin dapat melakukan proses *login* ke dalam sistem menggunakan akun yang telah terdaftar..
- 2) Admin dapat mengelola data alat bengkel (menambah, mengubah, dan menghapus data alat).
- 3) Admin dapat mengelola data inventaris alat bengkel.
- 4) Admin dapat mengelola materi pembelajaran.
- 5) Admin dapat melihat dan mengelola data transaksi peminjaman alat.

- 6) Admin dapat melakukan konfirmasi pengembalian alat yang telah dipinjam.
- 7) Admin dapat melihat, mengekspor, dan mengelola laporan peminjaman dalam format PDF.
- 8) Admin dapat melakukan *logout* dari sistem.

3. Kebutuhan fungsional *User* (Siswa):

- 1) *User* dapat melakukan *login* ke dalam sistem menggunakan akun SSO.
- 2) User dapat melihat daftar alat bengkel beserta informasi stok yang tersedia.
- 3) User dapat melakukan proses peminjaman alat bengkel.
- 4) User dapat melakukan pengembalian alat bengkel yang dipinjam.
- 5) User dapat melihat status peminjaman yang sedang berlangsung maupun riwayat peminjaman.
- 6) User dapat melihat dan mengunduh materi pembelajaran yang dibagikan oleh Admin.
- 7) User dapat melakukan *logout* dari sistem.

Kebutuhan fungsional ini akan direpresentasikan dalam bentuk interaksi antara aktor dan sistem pada *Use Case Diagram* di Bab IV.

B. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang berkaitan dengan kualitas sistem yang dibangun agar dapat berjalan dengan baik.

Adapun kebutuhan non-fungsional sistem adalah sebagai berikut:

1. Sistem berbasis *web* sehingga dapat diakses melalui berbagai perangkat menggunakan *browser* tanpa perlu instalasi tambahan.
2. Sistem memiliki mekanisme autentikasi dan otorisasi berbasis *Role-Based Access Control* (RBAC) yang terdiri atas SuperAdmin, Admin, dan User untuk menjaga keamanan data.
3. Sistem dirancang dengan antarmuka yang responsif (*responsive*) dan mudah digunakan (*user friendly*).
4. Sistem menggunakan *database* MySQL sebagai penyimpanan data terpusat untuk menjaga integritas dan konsistensi data.
5. Sistem memiliki performa yang stabil dan mampu mengelola data inventaris serta transaksi peminjaman secara efisien.
6. Sistem mendukung integrasi *Single Sign-On* (SSO) untuk mempermudah proses autentikasi pengguna.

C. Identifikasi Entitas Data

Berdasarkan kebutuhan sistem yang telah dianalisis, diperoleh beberapa entitas data yang akan digunakan dalam perancangan basis data. Identifikasi entitas ini menjadi dasar dalam pembuatan *Entity Relationship Diagram* (ERD) pada tahap perancangan sistem.

Adapun entitas yang teridentifikasi adalah sebagai berikut:

1. *User*

Menyimpan data pengguna sistem, seperti NIS, nama, username, password, role, dan identitas lainnya.

2. Admin

Menyimpan data Admin yang bertugas mengelola sistem.

3. SuperAdmin

Menyimpan data SuperAdmin yang memiliki hak akses tertinggi dalam sistem.

4. Barang

Menyimpan data alat bengkel, seperti nama barang, jumlah stok, kondisi barang, dan kategori barang.

5. Materi

Menyimpan data materi pembelajaran yang diunggah Admin dalam bentuk *file* PDF.

6. Peminjaman

Menyimpan data transaksi peminjaman barang yang dilakukan oleh *user*, termasuk tanggal peminjaman, tanggal pengembalian, dan status peminjaman.

7. Pengembalian

Menyimpan data pengembalian barang yang telah dipinjam beserta status konfirmasi dari Admin.

Setiap entitas tersebut memiliki hubungan (relasi) yang saling terintegrasi, yang selanjutnya akan dirancang secara lebih detail dalam bentuk ERD pada Bab IV.

3.3.2 System Design (Perancangan Sistem)

Pada tahap ini dilakukan proses perancangan sistem secara rinci berdasarkan hasil analisis kebutuhan yang telah dilakukan pada tahap sebelumnya. Perancangan sistem bertujuan untuk memberikan gambaran teknis mengenai bagaimana sistem informasi peminjaman alat bengkel akan dibangun, sehingga proses implementasi dapat dilakukan secara terarah, terstruktur, dan sesuai dengan kebutuhan pengguna di SMK Ma'arif Talang.

Sistem yang dirancang merupakan sistem informasi berbasis *website* yang dikembangkan menggunakan *framework* Laravel dengan menerapkan arsitektur *Model-View-Controller* (MVC). Perancangan dilakukan dengan memanfaatkan *Unified Modeling Language* (UML) untuk memodelkan sistem dari berbagai aspek, baik dari sisi interaksi pengguna, alur proses, maupun struktur data.

Adapun tahapan perancangan sistem yang dilakukan adalah sebagai berikut:

1. Perancangan *Use Case Diagram*

Use Case Diagram dirancang untuk menggambarkan interaksi antara aktor dengan sistem yang akan dibangun. Dalam sistem ini terdapat tiga aktor utama, yaitu SuperAdmin, Admin, dan *User* (Siswa).

SuperAdmin memiliki hak akses tertinggi dalam sistem yang meliputi pengelolaan data Admin dan siswa, serta *upload* data siswa berdasarkan NIS. Admin bertanggung jawab dalam pengelolaan data barang bengkel, materi pembelajaran, transaksi peminjaman dan pengembalian, serta pembuatan laporan. Sedangkan *User* (Siswa) memiliki akses untuk melihat informasi barang, melakukan peminjaman, pengembalian, serta mengakses materi pembelajaran yang tersedia.

Diagram ini digunakan untuk memastikan bahwa seluruh kebutuhan fungsional yang telah dianalisis sebelumnya dapat terakomodasi dengan baik dalam sistem.

2. Perancangan *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan alur proses utama dalam sistem secara detail. Proses yang dimodelkan meliputi:

- 1) Proses *login* pengguna menggunakan SSO

- 2) Proses pengelolaan data Admin dan siswa oleh SuperAdmin
- 3) Proses pengelolaan data barang oleh Admin
- 4) Proses upload materi pembelajaran oleh Admin
- 5) Proses peminjaman barang oleh *User*
- 6) Proses pengembalian barang atau buku oleh *User* dan konfirmasi pengembalian oleh Admin
- 7) Proses pembuatan dan ekspor laporan peminjaman.

Melalui *Activity Diagram*, alur kerja sistem dapat divisualisasikan secara sistematis mulai dari kondisi awal hingga akhir sehingga memudahkan dalam memahami logika proses yang akan diimplementasikan pada sistem.

3. Perancangan *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan interaksi antar komponen sistem berdasarkan urutan waktu. *Diagram* ini menunjukkan bagaimana sistem merespon setiap aksi yang dilakukan oleh pengguna.

Sebagai contoh, pada proses peminjaman:

- 1) *User* memilih alat bengkel.
- 2) Sistem memvalidasi ketersediaan stok.
- 3) Sistem menyimpan data peminjaman ke *database*.
- 4) Sistem menampilkan status peminjaman.

Dengan adanya *Sequence Diagram*, alur komunikasi antara *user interface*, *controller*, model, dan *database* pada *framework*

Laravel dapat dipahami dengan lebih jelas sesuai dengan konsep MVC.

4. Perancangan *Entity Relationship Diagram* (ERD)

ERD digunakan untuk merancang struktur *database* yang akan digunakan dalam sistem. Berdasarkan hasil analisis, terdapat beberapa entitas utama, yaitu:

- 1) *User*
- 2) Alat bengkel
- 3) Materi
- 4) Peminjaman
- 5) Pengembalian
- 6) Laporan

Entitas *User* menyimpan seluruh data pengguna sistem, baik SuperAdmin, Admin, maupun Siswa, dengan pembeda berupa atribut *role*. Setiap entitas memiliki atribut yang sesuai dengan kebutuhan sistem, seperti data identitas pengguna, stok barang dan buku, materi pembelajaran, serta informasi transaksi peminjaman dan pengembalian.

Relasi antar entitas dirancang agar data saling terhubung secara terintegrasi, misalnya relasi antara user dengan peminjaman, serta peminjaman dengan pengembalian.

Perancangan ERD ini menjadi dasar dalam pembuatan *database* menggunakan MySQL.

5. Perancangan *Database*

Perancangan *database* dilakukan dengan mengimplementasikan ERD ke dalam bentuk tabel-tabel pada MySQL. Setiap tabel dirancang dengan struktur yang sesuai, meliputi penentuan *field*, tipe data, primary key, serta foreign key. *Database* dirancang untuk:

- 1) Menyimpan data secara terpusat
- 2) Menjamin konsistensi dan integritas data
- 3) Mendukung proses transaksi secara *real-time*
- 4) Mendukung manajemen hak akses pengguna berdasarkan *role*.

Penggunaan MySQL dipilih karena memiliki performa yang stabil serta kompatibel dengan framework Laravel.

6. Perancangan Arsitektur Sistem

Sistem ini menggunakan arsitektur berbasis *web* dengan konsep *client-server*. Pengguna (SuperAdmin, Admin, dan *User*) mengakses sistem melalui *browser*, kemudian permintaan diproses oleh *server* aplikasi Laravel, dan data disimpan pada *database* MySQL.

Laravel menggunakan pola MVC yang terdiri dari:

- 1) *Model* → mengelola data dan *database*.
- 2) *View* → menampilkan antarmuka pengguna.
- 3) *Controller* → mengatur logika aplikasi.

Arsitektur ini memungkinkan sistem menjadi lebih terstruktur, mudah dikembangkan, serta memisahkan antara logika bisnis dan tampilan.

7. Perancangan Antarmuka Pengguna (*User Interface*)

Perancangan antarmuka dilakukan dengan mempertimbangkan kemudahan penggunaan (*user friendly*) dan tampilan yang responsif. Sistem dirancang agar dapat digunakan dengan mudah oleh SuperAdmin, Admin, maupun Siswa.

Antarmuka yang dirancang meliputi:

- 1) Halaman *login* SSO.
- 2) *Dashboard* SuperAdmin.
- 3) *Dashboard* Admin.
- 4) *Dashboard* User.
- 5) Halaman data siswa dan Admin.
- 6) Halaman data barang bengkel.
- 7) Halaman materi pembelajaran.
- 8) Halaman transaksi peminjaman.
- 9) Halaman pengembalian.
- 10) Halaman laporan.

Desain antarmuka dibuat sederhana, informatif, dan terstruktur agar pengguna dapat memahami sistem dengan cepat tanpa memerlukan pelatihan khusus.

3.3.3 Implementation (Implementasi)

Tahap implementasi merupakan proses penerjemahan hasil analisis kebutuhan dan perancangan sistem ke dalam bentuk aplikasi yang dapat dijalankan. Pada tahap ini dilakukan proses pengkodean (*coding*) menggunakan *framework* Laravel sebagai platform utama pengembangan sistem informasi peminjaman alat bengkel perpustakaan di SMK Ma'arif Talang.

Implementasi sistem dilakukan dengan menerapkan arsitektur *Model-View-Controller* (MVC) yang disediakan oleh Laravel. Model digunakan untuk mengelola data dan berinteraksi dengan *database* MySQL, *View* digunakan untuk menampilkan antarmuka pengguna, sedangkan *Controller* digunakan untuk mengatur logika aplikasi dan menghubungkan *Model* dengan *View*.

Proses implementasi meliputi beberapa kegiatan, yaitu:

1. Pembuatan *Database*

Tahap awal implementasi dilakukan dengan membuat struktur *database* menggunakan MySQL berdasarkan hasil perancangan *Entity Relationship Diagram* (ERD). Struktur *database* meliputi tabel pengguna, barang, materi pembelajaran, peminjaman, dan pengembalian beserta relasi antar tabel.

2. Pembuatan Model

Model dibuat untuk merepresentasikan setiap entitas yang terdapat pada *database*. Setiap model bertugas mengelola operasi

data seperti penyimpanan, pengubahan, penghapusan, dan pengambilan data dari *database*.

3. Pembuatan *Controller*

Controller digunakan untuk mengelola logika bisnis aplikasi. *Controller* bertugas menerima permintaan dari pengguna, memproses data melalui model, kemudian mengirimkan hasilnya ke halaman antarmuka yang sesuai.

4. Pembuatan Antarmuka Pengguna

Antarmuka pengguna dibangun menggunakan *Blade Template* dan Tailwind CSS yang terintegrasi dengan Laravel. Halaman yang dibuat meliputi halaman *login*, *dashboard*, data barang, peminjaman, pengembalian, materi pembelajaran, serta laporan.

5. Implementasi Hak Akses Pengguna

Sistem menerapkan mekanisme *Role Based Access Control* (RBAC) yang terdiri dari tiga jenis pengguna, yaitu SuperAdmin, Admin, dan Siswa. Setiap pengguna memiliki hak akses yang berbeda sesuai dengan tugas dan tanggung jawabnya dalam sistem.

6. Implementasi *Fitur* Sistem

Fitur-fitur utama yang diimplementasikan pada sistem meliputi:

- a. *Login* menggunakan *Single Sign-On* (SSO).

- b. Pengelolaan data Admin dan siswa oleh SuperAdmin.
- c. Pengelolaan data barang bengkel oleh Admin.
- d. Pengelolaan materi pembelajaran oleh Admin.
- e. Peminjaman alat bengkel perpustakaan oleh Siswa.
- f. Pengembalian alat oleh Siswa yang dikonfirmasi oleh Admin.
- g. Pembuatan dan ekspor laporan peminjaman dalam format PDF.

Melalui tahap implementasi ini, seluruh hasil perancangan sistem diwujudkan ke dalam bentuk aplikasi berbasis *web* yang siap untuk dilakukan pengujian pada tahap berikutnya.

3.3.4 Testing (Pengujian)

Tahap pengujian dilakukan untuk memastikan bahwa sistem informasi peminjaman berbasis *website* yang telah dikembangkan dapat berjalan sesuai dengan kebutuhan yang telah ditentukan pada tahap analisis. Pengujian bertujuan untuk menemukan kesalahan (*error*), memastikan setiap fungsi berjalan dengan baik, serta menjamin bahwa sistem dapat digunakan oleh pengguna secara optimal.

Metode pengujian yang digunakan dalam penelitian ini adalah *Black Box Testing*, yaitu metode pengujian yang berfokus pada fungsionalitas sistem tanpa melihat struktur kode program. Pengujian dilakukan dengan memberikan input pada sistem dan kemudian

mengamati *output* yang dihasilkan apakah sudah sesuai dengan yang diharapkan.

Adapun pengujian dilakukan pada fitur-fitur utama sistem sebagai berikut:

1. Pengujian Fitur *Login* dan Hak Akses

Pengujian dilakukan untuk memastikan bahwa sistem autentikasi dan otorisasi berjalan dengan baik.

Skenario pengujian:

- a. Pengguna melakukan *login* menggunakan akun yang valid.
- b. Pengguna memasukkan data *login* yang tidak valid.
- c. Pengguna yang belum *login* mencoba mengakses sistem.
- d. Pengguna mengakses menu yang tidak sesuai dengan hak aksesnya.

Hasil yang diharapkan:

Sistem mampu melakukan autentikasi pengguna dengan benar, membedakan hak akses SuperAdmin, Admin, dan Siswa, serta menjaga keamanan data dari akses yang tidak sah.

2. Pengujian Pengelolaan Data Pengguna

Pengujian dilakukan pada fitur pengelolaan data Admin dan siswa oleh SuperAdmin.

Skenario pengujian:

- a. Menambahkan data Admin atau siswa baru.
- b. Mengubah data Admin atau siswa.

- c. Menghapus data Admin atau siswa.
- d. Mengunggah data siswa berdasarkan NIS.

Hasil yang diharapkan:

Data pengguna tersimpan dengan benar di database dan dapat ditampilkan sesuai kebutuhan sistem.

3. Pengujian Pengelolaan Data Barang

Pengujian dilakukan pada fitur CRUD (*Create, Read, Update, Delete*) data barang dan buku oleh Admin.

Skenario pengujian:

- a. Menambahkan data baru.
- b. Mengubah data yang sudah ada.
- c. Menghapus data.
- d. Menampilkan data dalam tabel.

Hasil yang diharapkan:

Data tersimpan dengan benar di *database* dan ditampilkan secara akurat pada sistem.

4. Pengujian Fitur Peminjaman

Pengujian dilakukan untuk memastikan proses peminjaman berjalan sesuai logika sistem.

Skenario pengujian:

- a. Siswa melakukan peminjaman barang.
- b. Sistem mengecek ketersediaan stok.
- c. Sistem menyimpan data peminjaman.

- d. Sistem mengurangi jumlah stok.

Hasil yang diharapkan:

Proses peminjaman berjalan dengan baik dan data tercatat secara otomatis tanpa kesalahan.

5. Pengujian Fitur Pengembalian

Pengujian dilakukan untuk memastikan proses pengembalian berjalan dengan benar.

Skenario pengujian:

- a. Siswa mengajukan pengembalian barang.
- b. Admin melakukan konfirmasi pengembalian.
- c. Sistem mengubah status peminjaman.
- d. Sistem menambahkan kembali jumlah stok.

Hasil yang diharapkan:

Status transaksi berubah menjadi selesai dan stok kembali sesuai kondisi awal.

6. Pengujian Fitur Materi Pembelajaran

Pengujian dilakukan untuk memastikan fitur materi pembelajaran dapat digunakan dengan baik.

Skenario pengujian:

- a. Admin mengunggah *file* materi dalam *format* PDF.
- b. Sistem menyimpan data materi.
- c. Siswa melihat daftar materi.
- d. Siswa mengunduh materi yang tersedia.

Hasil yang diharapkan:

Materi pembelajaran dapat tersimpan, ditampilkan, dan diunduh dengan baik oleh pengguna.

7. Pengujian Fitur Laporan

Pengujian dilakukan untuk memastikan sistem dapat menampilkan dan mengekspor laporan.

Skenario pengujian:

- a. Menampilkan data laporan peminjaman.
- b. Melakukan *filter* data berdasarkan periode tertentu.
- c. Mengekspor laporan ke *format* PDF.

Hasil yang diharapkan:

Laporan ditampilkan dengan benar, lengkap, dan sesuai dengan data yang tersimpan di *database*.

8. Pengujian Antarmuka (*User Interface*)

Pengujian dilakukan untuk memastikan tampilan sistem dapat digunakan dengan baik oleh pengguna.

Skenario pengujian:

- a. Tampilan halaman pada berbagai perangkat (desktop dan *mobile*).
- b. Kemudahan navigasi menu.
- c. Konsistensi tampilan antar halaman.
- d. Kesesuaian tampilan berdasarkan hak akses pengguna.

Hasil yang diharapkan:

Sistem memiliki tampilan yang responsif, mudah digunakan, konsisten, dan tidak membingungkan pengguna.

3.3.5 Deployment (Penerapan Sistem)

Tahap *deployment* merupakan proses penerapan sistem yang telah selesai dikembangkan dan diuji ke dalam lingkungan nyata agar dapat digunakan oleh pengguna. Pada penelitian ini, sistem informasi peminjaman berbasis *website* diimplementasikan menggunakan server milik sendiri (*self-hosted server*), sehingga pengelolaan sistem dapat dilakukan secara mandiri dan lebih fleksibel sesuai kebutuhan.

Deployment dilakukan dengan mengunggah aplikasi yang telah dikembangkan menggunakan *framework* Laravel ke dalam *server*, serta melakukan konfigurasi agar sistem dapat diakses melalui jaringan lokal maupun internet.

Adapun tahapan *deployment* yang dilakukan adalah sebagai berikut:

1. Persiapan *Server*

Server yang digunakan dikonfigurasi sebagai *web server* dengan kebutuhan sebagai berikut:

- 1) *Web Server (Apache/Nginx)*
- 2) *PHP* sesuai kebutuhan *Laravel*
- 3) *Database MySQL*
- 4) *Composer* untuk *dependency* *Laravel*

Server dapat berupa komputer lokal yang dijadikan server atau VPS (*Virtual Private Server*) sesuai kebutuhan implementasi.

2. Konfigurasi Aplikasi Laravel

Setelah aplikasi diunggah ke *server*, dilakukan konfigurasi pada file `.env` untuk menyesuaikan dengan lingkungan server, meliputi:

- 1) Konfigurasi koneksi *database*
- 2) Pengaturan URL aplikasi
- 3) Konfigurasi *environment (production mode)*

Selanjutnya dilakukan perintah:

- 1) *composer install* untuk menginstal *dependency*
- 2) *php artisan migrate* untuk membuat struktur *database*
- 3) *php artisan key:generate* untuk keamanan aplikasi

3. Konfigurasi Database

Database MySQL dibuat pada server, kemudian dilakukan *import* atau migrasi tabel sesuai dengan struktur yang telah dirancang. Seluruh data sistem seperti data pengguna, barang, buku, materi pembelajaran, serta transaksi peminjaman dan pengembalian akan tersimpan secara terpusat pada *database* ini.

4. Konfigurasi Web Server

Web server dikonfigurasi agar dapat menjalankan aplikasi Laravel, termasuk:

- 1) Menentukan *root directory* ke *folder public* Laravel

- 2) Mengatur *virtual host* (jika menggunakan *domain*)
- 3) Mengaktifkan modul *rewrite* (untuk *Apache*)

Konfigurasi ini bertujuan agar sistem dapat diakses dengan URL yang sesuai.

5. Pengujian Sistem di *Server*

Setelah *deployment* selesai, dilakukan pengujian ulang untuk memastikan bahwa sistem berjalan dengan baik di lingkungan *server*, meliputi:

- 1) Akses halaman *login* SSO
- 2) Penggunaan fitur utama sistem
- 3) Koneksi ke *database*
- 4) Pengujian hak akses SuperAdmin, Admin, dan Siswa

Pengujian ini penting untuk memastikan tidak ada *error* akibat perbedaan lingkungan antara *development* dan *production*.

6. Akses Pengguna

Sistem yang telah dideploy dapat diakses oleh:

- a. SuperAdmin melalui jaringan sekolah.
- b. Admin melalui jaringan sekolah.
- c. Siswa melalui *browser* yang terhubung ke jaringan sekolah.

Jika *server* terhubung dengan *internet*, maka sistem juga dapat diakses secara *online* menggunakan *domain* atau IP *server*.

3.3.6 Maintenance (Pemeliharaan)

Tahap *maintenance* merupakan tahap pemeliharaan sistem yang dilakukan setelah sistem informasi peminjaman berbasis *website* diimplementasikan dan digunakan oleh pengguna. Tahap ini bertujuan untuk menjaga agar sistem tetap berjalan dengan baik, stabil, aman, dan sesuai dengan kebutuhan operasional di SMK Ma'arif Talang.

Pemeliharaan sistem dilakukan secara berkala untuk mengantisipasi adanya kesalahan (*bug*), meningkatkan performa sistem, serta menyesuaikan sistem dengan kebutuhan baru yang mungkin muncul seiring waktu.

Adapun kegiatan yang dilakukan pada tahap *maintenance* meliputi:

1. Perbaikan Kesalahan (*Bug Fixing*)

Setelah sistem digunakan, kemungkinan ditemukan kesalahan yang tidak terdeteksi pada tahap pengujian. Oleh karena itu, dilakukan perbaikan terhadap *bug* yang muncul, baik pada sisi fungsionalitas maupun tampilan sistem.

Perbaikan ini bertujuan untuk memastikan sistem tetap berjalan sesuai dengan kebutuhan pengguna dan tidak mengganggu proses operasional.

2. Pemeliharaan *Database*

Database sebagai pusat penyimpanan data perlu dijaga agar tetap aman dan terstruktur. Kegiatan yang dilakukan meliputi:

1) *Backup* data secara berkala.

- 2) Pembersihan data yang tidak diperlukan.
- 3) Optimasi *database* untuk menjaga performa.

Hal ini penting untuk menghindari kehilangan data serta menjaga kecepatan akses sistem.

3. Peningkatan Performa Sistem

Seiring bertambahnya jumlah data dan pengguna, performa sistem dapat mengalami penurunan. Oleh karena itu, dilakukan optimasi sistem seperti:

- 1) Optimasi *query database*.
- 2) Pengelolaan *resource server*.
- 3) Peningkatan kapasitas *server* jika diperlukan.

Langkah ini bertujuan agar sistem tetap responsif dan stabil saat digunakan.

4. Pengembangan dan Penambahan Fitur

Maintenance juga mencakup pengembangan sistem untuk menyesuaikan dengan kebutuhan baru. Contohnya:

- 1) Penambahan fitur notifikasi.
- 2) Pengembangan laporan yang lebih detail.
- 3) Pengembangan fitur manajemen pengguna.
- 4) Integrasi dengan sistem lain apabila diperlukan.

Pengembangan ini dilakukan agar sistem tetap relevan dan memberikan manfaat yang lebih optimal bagi pengguna.

5. Pemeliharaan Keamanan Sistem

Keamanan sistem menjadi aspek penting dalam pemeliharaan. Kegiatan yang dilakukan meliputi:

- 1) *Update framework* Laravel ke versi terbaru.
- 2) Pemeliharaan sistem autentikasi dan otorisasi berbasis *role*.
- 3) Pengamanan akses *login* SSO.
- 4) Pencegahan serangan seperti *SQL Injection* dan *Cross-Site Scripting* (XSS).

Hal ini bertujuan untuk melindungi data dari akses yang tidak sah.

6. *Monitoring* Sistem

Monitoring dilakukan untuk memastikan sistem berjalan dengan baik secara terus-menerus. Kegiatan ini meliputi:

- 1) Memantau penggunaan sistem.
- 2) Mengecek *error log*.
- 3) Memastikan *server* berjalan normal.
- 4) Memantau aktivitas pengguna berdasarkan hak akses.

Dengan adanya *monitoring*, permasalahan dapat dideteksi lebih cepat dan segera ditangani.

3.2 Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk memperoleh informasi yang akurat dan relevan sebagai dasar dalam pengembangan sistem. Adapun metode yang digunakan dalam penelitian ini adalah sebagai berikut:

3.3.1 Observasi

Observasi dilakukan dengan cara pengamatan langsung terhadap proses peminjaman alat bengkel di SMK Ma'arif Talang. Peneliti mengamati secara langsung bagaimana alur peminjaman dan pengembalian dilakukan oleh Admin, mulai dari proses pencatatan hingga penyimpanan data.

Dalam kegiatan observasi ini, peneliti juga mengidentifikasi sistem pencatatan yang masih menggunakan buku tulis, serta berbagai kendala yang dihadapi seperti kesalahan pencatatan, keterlambatan dalam pencarian data, dan kesulitan dalam mengetahui stok barang secara cepat. Metode ini digunakan untuk memperoleh gambaran nyata kondisi lapangan sebagai dasar dalam analisis kebutuhan sistem.

3.3.2 Wawancara

Wawancara dilakukan dengan pihak terkait seperti Admin bengkel, dan guru. Tujuan wawancara adalah untuk mendapatkan informasi secara mendalam mengenai kebutuhan sistem, kendala yang dihadapi, serta harapan pengguna terhadap sistem yang akan dikembangkan.

Wawancara dilakukan secara terstruktur dengan menggunakan panduan pertanyaan yang telah disiapkan sebelumnya. Dalam penelitian ini, wawancara dilakukan dengan:

Bapak Sholeh, selaku guru sekaligus Admin di bagian bengkel teknik, yang memberikan informasi terkait proses peminjaman alat praktik.

Dari hasil wawancara tersebut diperoleh informasi mengenai alur kerja sistem yang berjalan, permasalahan yang sering terjadi, serta kebutuhan sistem yang diharapkan oleh pengguna [14].

3.3.3 Studi Literatur

Studi literatur dilakukan dengan mengkaji berbagai sumber seperti jurnal ilmiah, buku, serta dokumentasi resmi terkait pengembangan sistem informasi, metode Waterfall, dan *framework* Laravel [15].

Metode ini bertujuan untuk memperkuat landasan teori serta mendukung proses analisis dan perancangan sistem.

3.3 Waktu dan Tempat Penelitian

3.3.1 Waktu Penelitian

Penelitian ini dilaksanakan dalam kurun waktu kurang lebih 3–6 bulan, yang mencakup seluruh rangkaian kegiatan penelitian mulai dari pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian dan penyempurnaan sistem. Rentang waktu tersebut disusun dengan mempertimbangkan kompleksitas sistem yang dikembangkan serta ketersediaan data dan sumber daya yang mendukung proses penelitian.

Selama periode penelitian, setiap proses dilakukan secara bertahap dan berkesinambungan sesuai dengan metode pengembangan sistem Waterfall, sehingga hasil dari satu tahap dapat menjadi dasar bagi tahap berikutnya. Pengumpulan data dilakukan pada awal penelitian melalui metode observasi, wawancara, dan studi literatur untuk memperoleh gambaran kondisi nyata di lapangan.

Selanjutnya, data yang diperoleh dianalisis guna menentukan kebutuhan sistem yang akan dikembangkan. Proses perancangan dan implementasi dilakukan secara sistematis berdasarkan hasil analisis, kemudian dilanjutkan dengan pengujian sistem untuk memastikan kesesuaian fungsi dengan kebutuhan pengguna. Tahap akhir dilakukan penyempurnaan sistem agar dapat berjalan dengan optimal.

Dengan alokasi waktu tersebut, diharapkan penelitian dapat berjalan secara efektif, terarah, dan menghasilkan sistem yang sesuai dengan tujuan yang telah ditetapkan.

3.3.2 Tempat Penelitian

Penelitian ini dilaksanakan di SMK Ma'arif Talang, yang merupakan salah satu institusi pendidikan menengah kejuruan yang memiliki fasilitas bengkel praktik dan perpustakaan sebagai penunjang kegiatan pembelajaran.

Dalam penelitian ini, peneliti melakukan pengumpulan data melalui observasi langsung serta wawancara dengan pihak yang terlibat dalam pengelolaan sistem peminjaman. Wawancara dilakukan

untuk memperoleh informasi yang lebih mendalam mengenai alur kerja sistem yang berjalan serta kendala yang dihadapi.

Adapun wawancara dilakukan dengan bapak Sholeh, selaku guru sekaligus Admin di bagian bengkel teknik

Selain itu, studi literatur juga dilakukan untuk mendukung pemahaman teori dan konsep yang digunakan dalam penelitian.

Fokus utama yang diamati meliputi proses peminjaman dan pengembalian alat, mekanisme pencatatan data oleh petugas, serta alur pelayanan kepada pengguna. Subjek yang terlibat dalam penelitian ini meliputi Admin bengkel, guru, serta siswa sebagai pengguna layanan.

Melalui penelitian yang dilakukan di lokasi tersebut, diharapkan dapat dikembangkan sistem informasi berbasis web yang mampu meningkatkan efisiensi, akurasi, serta kemudahan dalam pengelolaan data peminjaman, sehingga mendukung kegiatan operasional di SMK Ma'arif Talang secara lebih optimal.