

Laporan Praktikum Pembuatan Website Peminjaman Laptop

Mata Kuliah : Pemograman Lanjutan



Disusun oleh:

Nama : Alyaa Mahiraah Ramadhani

NPM : 2423101997

Kelas : II Rekayasa Keamanan Siber A

**REKAYASA KEAMANAN SIBER
POLITEKNIK SIBER DAN SANDI NEGARA**

2026

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa, karena atas limpahan rahmat dan karunia-Nya, kami dapat menyelesaikan penyusunan laporan praktikum yang berjudul "Laporan Praktikum Pembuatan Website Peminjaman Laptop" dengan baik dan tepat waktu. Penyusunan laporan praktikum ini kami maksudkan untuk memenuhi nilai tugas Mata Kuliah Pemograman Lanjutan di Politeknik Siber dan Sandi Negara.

Dalam proses pelaksanaan praktikum hingga penyusunan laporan ini, kami mendapatkan banyak bimbingan, bantuan, dan dukungan dari berbagai pihak. Oleh karena itu, kami ingin menyampaikan ucapan terima kasih yang tulus kepada:

1. Bapak Rahmat Purwoko, S.T., M.T., selaku Dosen Pengampu Mata Kuliah Pemograman Lanjutan, yang telah memberikan bimbingan dan arahan yang sangat berharga selama proses pembelajaran.
2. Orang tua dan keluarga tercinta, yang senantiasa memberikan dukungan morel, materiil, dan doa yang tidak terhingga.

Saya menyadari sepenuhnya bahwa laporan praktikum ini masih jauh dari kata sempurna karena keterbatasan pengetahuan dan pengalaman yang dimiliki. Oleh karena itu, segala bentuk kritik dan saran yang membangun akan saya terima dengan terbuka demi perbaikan di masa mendatang.

Akhir kata, saya berharap semoga laporan praktikum ini dapat memberikan manfaat dan menambah wawasan, khususnya mengenai penggunaan osiloskop dan function generator, bagi para pembaca.

Bogor, 23 Februari 2026

Penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat telah mendorong pemanfaatan sistem berbasis web dalam berbagai bidang, termasuk dalam pengelolaan sarana dan prasarana. Website menjadi salah satu media yang efektif karena dapat diakses dengan mudah, bersifat fleksibel, serta mampu menyajikan dan mengelola data secara terstruktur.

Proses peminjaman dan pengembalian laptop yang masih dilakukan secara manual sering menimbulkan berbagai permasalahan, seperti pencatatan yang tidak rapi, risiko kehilangan data, serta kesulitan dalam memantau status ketersediaan perangkat. Dengan adanya aplikasi peminjaman dan pengembalian laptop berbasis website, proses pengelolaan data dapat dilakukan secara lebih efektif, efisien, dan akurat. Sistem ini memungkinkan admin dan pengguna untuk melakukan pencatatan, pemantauan, serta pelaporan data secara real-time.

Melalui kegiatan praktikum pembuatan website aplikasi peminjaman dan pengembalian laptop ini, penulis diharapkan mampu mengimplementasikan konsep-konsep dasar pemrograman web yang telah dipelajari, seperti perancangan antarmuka, pengelolaan basis data, serta logika pemrograman. Selain itu, praktikum ini juga bertujuan untuk melatih kemampuan analisis, pemecahan masalah, dan kerja mandiri mahasiswa dalam mengembangkan sebuah sistem informasi sederhana namun fungsional.

1.2 Tujuan

1. Melatih dalam merancang dan membangun aplikasi berbasis website.
2. Mengimplementasikan konsep dasar pemrograman web.
3. Menerapkan penggunaan database untuk menyimpan dan mengelola data peminjaman.

1.3 Ruang Lingkup

- Aplikasi yang dibuat berupa website peminjaman dan pengembalian laptop.
- Fitur yang dibahas meliputi:
 - Pengelolaan data pengguna (user/admin)
 - Pengelolaan data laptop
 - Proses peminjaman laptop
 - Proses pengembalian laptop

- Status ketersediaan laptop
 - Riwayat peminjaman dan pengembalian
- Sistem memiliki 2 jenis akses:
 - Admin (mengelola data dan memantau transaksi)
 - User (melakukan peminjaman dan pengembalian)

BAB II

TEKNOLOGI DAN ARSITEKTUR APLIKASI WEB

2.1 Pengantar Teknologi Web

Teknologi web adalah sekumpulan teknologi yang digunakan untuk membangun dan menjalankan situs maupun aplikasi web di internet. Pada dasarnya, teknologi ini bertumpu pada tiga komponen utama di sisi klien (front-end), yaitu HTML untuk menyusun struktur konten, CSS untuk mengatur tampilan, dan JavaScript untuk menambahkan interaktivitas pada halaman yang diakses melalui browser. Di sisi server (back-end), pengembangan web memanfaatkan bahasa pemrograman dan basis data untuk memproses logika aplikasi serta menyimpan data, sehingga halaman yang ditampilkan tidak lagi statis tetapi dinamis dan dapat menyesuaikan dengan kebutuhan pengguna. Pemisahan peran antara front-end dan back-end ini menjadi konsep dasar penting yang perlu dipahami dalam praktikum pemrograman web.

Seiring perkembangan, web berevolusi dari Web 1.0 yang bersifat statis dan hanya “read-only” menuju Web 2.0 yang mendukung kolaborasi dan konten buatan pengguna (read–write), dan kini mengarah ke Web 3.0 yang lebih cerdas dan terdesentralisasi dengan dukungan AI, IoT, dan blockchain . Perkembangan ini menuntut pengembang untuk tidak hanya menguasai HTML dan CSS, tetapi juga memahami framework modern dan konsep aplikasi web interaktif yang responsif di berbagai perangkat .

2.2 Arsitektur Web Modern

Arsitektur web modern dirancang untuk membuat aplikasi web yang modular, skalabel, dan mudah dipelihara. Secara umum, aplikasi web dibangun dengan pendekatan multi-lapis (multi-tier), misalnya pemisahan lapisan presentasi (front-end), layanan/logika (service atau controller), dan persistensi data (database) agar tiap lapisan dapat dikembangkan dan diuji secara terpisah. Pemisahan ini didukung pola pengembangan full-stack modern yang menghubungkan framework front-end (seperti React/Angular) dengan back-end berbasis RESTful API dan database relasional maupun NoSQL, sehingga tercipta solusi end-to-end yang konsisten dan mudah

diintegrasikan dengan sistem lain. Dalam konteks praktikum, pemahaman struktur berlapis ini membantu mahasiswa melihat dengan jelas alur data: dari antarmuka pengguna, ke logika server, hingga penyimpanan di basis data.

Perkembangan kebutuhan skala besar mendorong pergeseran dari arsitektur monolitik ke microservices dan sistem terdistribusi. Microservices memecah aplikasi menjadi layanan-layanan kecil yang dapat dikembangkan, di-deploy, dan diskalakan secara independen, biasanya berkomunikasi melalui API stateless dan pola seperti API Gateway, service discovery, serta circuit breaker untuk menjaga ketersediaan dan ketahanan sistem. Pendekatan ini sering dikombinasikan dengan komputasi awan, container (misalnya Docker), dan DevOps/CI-CD untuk mendukung penskalaan horizontal, deployment cepat, dan pemantauan berkelanjutan.

2.3 Keamanan pada Aplikasi Web

Keamanan pada aplikasi web merupakan aspek penting yang harus diperhatikan sejak tahap perancangan hingga deployment, karena aplikasi web sering menangani data sensitif seperti informasi pribadi, kredensial login, dan data keuangan. Serangan umum yang mengancam aplikasi web meliputi SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), broken authentication, dan broken access control. Kerentanan ini biasanya muncul karena validasi input yang lemah, praktik pengkodean yang tidak aman, serta konfigurasi server atau database yang salah. Dampaknya dapat berupa pencurian data, perubahan data tanpa izin, pengambilalihan akun, hingga kerugian finansial dan reputasi bagi organisasi. Oleh karena itu, pengembang dan administrator perlu memahami jenis-jenis serangan ini dan bagaimana cara kerjanya sebagai dasar untuk merancang mekanisme perlindungan yang efektif.

Untuk meminimalkan risiko tersebut, penelitian terbaru menekankan pentingnya secure coding dan penerapan prinsip keamanan di seluruh Software Development Lifecycle (SDLC). Praktik yang disarankan antara lain validasi dan sanitasi input, penggunaan prepared statement untuk mencegah SQL Injection, pengaturan autentikasi dan manajemen sesi yang kuat (misalnya 2FA dan session timeout), penggunaan enkripsi yang tepat untuk data sensitif, serta penerapan kontrol akses berbasis peran (RBAC). Kerangka kerja seperti OWASP Top 10 banyak digunakan sebagai panduan utama untuk

mengidentifikasi risiko paling kritis dan menentukan langkah mitigasinya. Selain itu, pengujian keamanan seperti code review, static analysis, dan penetration testing perlu dilakukan secara berkala untuk menemukan dan menutup celah keamanan sebelum disalahgunakan.

BAB III

TEKNOLOGI DAN ARSITEKTUR APLIKASI MOBILE

3.1 Pengantar Teknologi Mobile

Teknologi mobile didefinisikan sebagai perangkat teknologi komunikasi seluler yang memungkinkan pengguna untuk mengakses informasi dan berkomunikasi tanpa terbatas oleh ruang dan waktu. Perangkat ini telah berevolusi secara signifikan dari generasi ke generasi, dimulai dari teknologi analog (1G) yang hanya mampu mengirimkan suara, hingga teknologi digital seperti 2G yang memperkenalkan layanan pesan singkat (SMS), dan 3G yang memungkinkan akses data seperti video call dan internet cepat. Perkembangan terkini pada generasi 4G dan 5G semakin mengukuhkan posisi perangkat mobile sebagai perangkat komputasi pribadi yang canggih, mampu melakukan berbagai fungsi mulai dari navigasi GPS, browsing internet, hingga menjadi pusat hiburan genggam dengan kecepatan akses data yang sangat tinggi .

Dalam konteks pendidikan dan praktikum, pemanfaatan teknologi mobile telah membuka paradigma baru dalam metode pembelajaran. Dengan spesifikasi perangkat yang semakin mumpuni, smartphone tidak lagi sekadar alat komunikasi, melainkan telah bertransformasi menjadi laboratorium digital portabel. Berbagai sensor yang tertanam dalam perangkat mobile, seperti akselerometer, giroskop, dan magnetometer, dapat dimanfaatkan untuk melakukan eksperimen ilmiah, seperti praktikum fisika, yang sebelumnya hanya bisa dilakukan di laboratorium konvensional. Inovasi ini memungkinkan proses pembelajaran dan penyelidikan ilmiah (inkuiri) dapat dilakukan secara lebih fleksibel, interaktif, dan menarik, sehingga mampu meningkatkan pemahaman konsep mahasiswa terhadap materi yang abstrak menjadi lebih konkret.

3.2 Arsitektur Aplikasi Mobile

Arsitektur aplikasi mobile merupakan kerangka kerja struktural yang mendefinisikan komponen-komponen perangkat lunak, relasi antar komponen, serta prinsip-prinsip yang mengatur desain dan evolusinya seiring waktu. Dalam pengembangan aplikasi modern, pemilihan arsitektur yang tepat menjadi faktor krusial karena berdampak langsung pada performa, skalabilitas, dan maintainability aplikasi, terutama mengingat keterbatasan sumber daya pada perangkat mobile seperti

memori dan daya proses . Beberapa pola arsitektur yang umum diadopsi antara lain Model-View-ViewModel (MVVM) dan Model-View-Presenter (MVP) untuk sisi frontend, serta arsitektur Microservices untuk sisi backend. Penelitian menunjukkan bahwa MVVM cenderung unggul dalam efisiensi memori dan waktu eksekusi dibandingkan MVP karena mekanisme binding dan siklus hidup yang lebih terkelola, meskipun MVP dapat lebih unggul dalam efisiensi penggunaan CPU pada skenario tertentu.

Di sisi lain, arsitektur Microservices memisahkan backend ke dalam layanan-layanan independen yang saling terhubung melalui API Gateway, seperti yang diimplementasikan pada aplikasi pencari kos dengan layanan terpisah untuk User Service, Kos Service, dan Notifikasi Service . Pendekatan ini memungkinkan pengembangan, deployment, dan scaling secara independen, sehingga meningkatkan fleksibilitas dan ketahanan sistem secara keseluruhan. Lebih lanjut, penerapan arsitektur modular seperti HMVC (Hierarchical Model-View-Controller) pada backend juga terbukti menghasilkan maintainability yang tinggi, dengan skor kepuasan mencapai 94% berdasarkan evaluasi menggunakan McCall Software Quality Model, karena memudahkan pengembangan dan pemeliharaan modul secara terpisah.

3.3 Integrasi dengan API

Integrasi dengan Application Programming Interface (API) merupakan aspek fundamental dalam pengembangan aplikasi mobile modern, yang memungkinkan aplikasi untuk berkomunikasi dan bertukar data dengan layanan backend. Dalam konteks proyek akhir, setiap aplikasi mobile diharapkan mampu berinteraksi dengan layanan web melalui API, di mana penggunaan RESTful API dengan format JSON sangat direkomendasikan. Arsitektur REST (Representational State Transfer) memanfaatkan metode HTTP standar seperti GET, POST, PUT, dan DELETE untuk melakukan operasi data, sehingga menawarkan kesederhanaan, skalabilitas, serta interoperabilitas yang tinggi antara aplikasi web dan mobile. Penelitian dalam pengembangan aplikasi seperti sistem pemesanan travel dan aplikasi pencari kost membuktikan bahwa implementasi RESTful API memungkinkan sinkronisasi data secara real-time dan efisien antara frontend mobile dan backend server.

Penggunaan format JSON (JavaScript Object Notation) sebagai media pertukaran data semakin memperkuat efisiensi komunikasi tersebut karena sifatnya yang ringan, mudah dibaca oleh manusia, dan mudah diurai oleh mesin dibandingkan

format lainnya seperti XML. Dalam praktiknya, aplikasi mobile akan mengirimkan permintaan HTTP ke endpoint API yang telah disediakan oleh backend, kemudian server akan memproses permintaan tersebut dan mengembalikan respons dalam bentuk JSON. Data ini selanjutnya akan diuraikan (parsing) oleh aplikasi mobile untuk ditampilkan kepada pengguna atau disimpan secara lokal. Arsitektur ini tidak hanya memisahkan logika bisnis di backend dari antarmuka pengguna di mobile, tetapi juga memungkinkan pengembangan yang lebih terstruktur dan maintainable, di mana perubahan pada satu sisi tidak selalu mengharuskan perubahan besar pada sisi lainnya.

BAB IV

MANAJEMEN PROYEK DAN SIKLUS PENGEMBANGAN PERANGKAT LUNAK (SDLC)

4.1 Pengantar Manajemen Proyek

Manajemen proyek pengembangan website peminjaman laptop ini menerapkan pendekatan terstruktur dengan menggabungkan perencanaan sistematis dan metodologi tangkas (Agile) yang terdokumentasi dengan baik. Proyek diawali dengan fase perencanaan dan analisis yang mendalam, di mana kebutuhan pengguna (Taruna dan Admin) diidentifikasi melalui user stories (seperti AS-01 hingga AS-06) dan kemudian dispesifikasikan menjadi persyaratan fungsional dan non-fungsional yang jelas dalam dokumen SRS. Fondasi ini menjadi acuan untuk fase desain sistem, yang mencakup pembuatan diagram UML (Use Case, Sequence, Class) untuk memodelkan arsitektur serta desain antarmuka pengguna dengan skema warna merah-biru-putih. Proses pengembangan kemudian dijalankan dalam tiga sprint selama delapan minggu, sebagaimana digambarkan dalam timeline proyek: Sprint 1 fokus pada fitur inti (form pinjam, riwayat, notifikasi), Sprint 2 pada fitur tambahan (modal jam malam, pengembalian), dan Sprint 3 pada integrasi REST API serta optimasi. Setiap sprint menghasilkan inkremen kode sumber yang fungsional (menggunakan Laravel), yang kemudian memasuki fase pengujian terpadu, meliputi uji coba form, validasi notifikasi SweetAlert, dan audit keamanan dasar. Seluruh alur ini dikelola dengan keluaran terdokumentasi di setiap fase—mulai dari PRD, cerita pengguna, laporan bug, hingga dokumentasi API—yang mencerminkan peran mahasiswa tidak hanya sebagai pengembang, tetapi juga sebagai manajer proyek yang memastikan setiap tahapan SDLC berjalan sesuai waktu, ruang lingkup, dan kualitas yang ditargetkan.

4.2 Perencanaan Proyek

Tahap perencanaan proyek merupakan fondasi awal yang menentukan arah dan keberhasilan seluruh siklus pengembangan perangkat lunak, di mana aktivitas yang dilakukan mencakup identifikasi tujuan proyek secara komprehensif, seperti fungsi yang akan dibangun, manfaat yang dihasilkan, serta siapa target pengguna dari aplikasi tersebut. Selanjutnya, dilakukan penyusunan *Work Breakdown Structure* (WBS) untuk memecah pekerjaan besar menjadi tugas-tugas kecil yang lebih terkelola, yang kemudian dilanjutkan dengan penjadwalan menggunakan alat

bantu seperti *Gantt Chart* untuk visualisasi timeline atau *Kanban Board* untuk memantau alur kerja. Selain aspek teknis, perencanaan yang matang juga mencakup analisis risiko untuk mengidentifikasi potensi hambatan sejak dini serta menyusun strategi mitigasi yang tepat. Dengan perencanaan yang baik, pengembang dapat meminimalkan ketidakpastian dan meningkatkan efisiensi selama proses implementasi.

| SDLC (Siklus Hidup Pengembangan Perangkat Lunak) | | |
|--|--|---------------------------|
| Fase | Aktivitas | Keluaran |
| 1. Perencanaan | Pengumpulan syarat (Nama, NPM, Keterangan) | PRD, Cerita Pengguna |
| 2. Analisis | Spesifikasi sistem dan skema database | Daftar Fitur, Dokumen SRS |
| 3. Desain | Desain UI merah-biru dan ERD Database | UML, ERD, Maket UI |
| 4. Pengembangan | Coding Laravel | Kode Sumber |
| 5. Pengujian | Uji coba form dan notifikasi SweetAlert | Laporan Bug, Kasus Uji |

4.3 Model SDLC yang Direkomendasikan

Pemilihan model *Software Development Life Cycle* (SDLC) yang tepat bergantung pada karakteristik proyek, kompleksitas kebutuhan, serta dinamika tim pengembang, sehingga mahasiswa perlu memahami kelebihan dan kekurangan masing-masing model sebelum menerapkannya.

Model *Waterfall* direkomendasikan untuk proyek dengan kebutuhan yang stabil, jelas sejak awal, dan perubahan minimal selama pengembangan, karena pendekatannya yang berurutan dan terstruktur. Sementara itu, model *Iterative* cocok untuk proyek yang memerlukan penyempurnaan bertahap, di mana setiap iterasi menghasilkan versi prototipe yang dapat dievaluasi sebelum lanjut ke tahap berikutnya. Adapun model *Agile* atau *Scrum* sangat sesuai untuk proyek tim kecil yang membutuhkan fleksibilitas tinggi, kolaborasi dinamis, serta respons cepat terhadap perubahan kebutuhan pengguna, sehingga memungkinkan pengiriman fitur secara bertahap dalam jangka waktu pendek (*sprint*).

4.4 Dokumentasi Proyek

| 📖 Kisah Pengguna | | |
|------------------|--|-----------|
| PENGENAL | Kisah Pengguna | Prioritas |
| AS-01 | Sebagai Taruna, saya ingin menginput data NPM agar sistem mengenali identitas peminjam. | Tinggi |
| AS-02 | Sebagai Taruna, saya ingin memilih kategori laptop agar sesuai dengan kebutuhan tugas/lab. | Tinggi |
| AS-03 | Sebagai Taruna, saya ingin menerima notifikasi "Berhasil" agar yakin data sudah tersimpan. | Tinggi |
| AS-04 | Sebagai Taruna, saya ingin melihat status "Dipinjam" atau "Kembali" untuk memantau tanggungan. | Sedang |
| AS-05 | Sebagai Taruna, saya ingin mengajukan jam malam agar bisa menggunakan laptop lebih lama untuk kegiatan seperti akademik maupun organisasi. | Sedang |
| AS-06 | Sebagai Admin, saya ingin melihat semua riwayat peminjaman dalam satu tabel yang rapi. | Tinggi |

| 📄 SRS (Spesifikasi Persyaratan Perangkat Lunak) | | | |
|---|--------------|---|-----------|
| Persyaratan Fungsional | | | |
| PENGENAL | Fitur | Deskripsi | Status |
| FR-01 | Form Pinjam | Input Nama, NPM, dan Keterangan laptop secara dinamis. | ✅ Selesai |
| FR-02 | Riwayat | Tabel yang menampilkan Tgl Pinjam, Tgl Kembali, dan Status. | ✅ Selesai |
| FR-03 | Jam Malam | Modal popup khusus untuk mengatur rentang waktu jam malam. | ✅ Selesai |
| FR-04 | Pengembalian | Fitur satu klik untuk mengubah status laptop menjadi "Sudah Kembali". | ✅ Selesai |
| FR-05 | Notifikasi | Pop-up SweetAlert untuk konfirmasi sukses transaksi. | ✅ Selesai |
| FR-06 | REST API | Endpoint untuk integrasi data dengan aplikasi pihak ketiga. | ✅ Selesai |

| Persyaratan Non-Fungsional | | |
|----------------------------|-------------------|--|
| PENGENAL | Persyaratan | Deskripsi |
| NFR-01 | Keamanan | Validasi input data Taruna dan perlindungan CSRF pada Laravel. |
| NFR-02 | Kegunaan | Desain responsif dengan skema warna Merah, Biru, dan Putih. |
| NFR-03 | Keandalan | Sinkronisasi waktu otomatis menggunakan timezone Asia/Jakarta. |
| NFR-04 | Interoperabilitas | Dukungan REST API untuk pertukaran data format JSON. |

BAB V

INTEGRASI DAN DEPLOYMENT BERKELANJUTAN (CI/CD) DENGAN GITLAB

5.1 Konsep CI/CD

CI/CD (*Continuous Integration/Continuous Deployment*) merupakan praktik modern dalam pengembangan perangkat lunak yang bertujuan mengotomatisasi proses integrasi kode, pengujian, dan penerapan aplikasi. Pada proyek Web Peminjaman Laptop, CI/CD membantu memastikan setiap perubahan kode yang diunggah ke repositori akan otomatis diuji dan diproses sebelum diterapkan ke server. Dengan pendekatan ini, kesalahan dapat terdeteksi lebih awal, kualitas aplikasi lebih terjaga, serta proses pengembangan menjadi lebih cepat, terstruktur, dan minim risiko dibandingkan deployment manual.

5.2 CI/CD Pipeline di GitLab

Dalam implementasinya, GitLab menyediakan fitur pipeline yang terdiri dari beberapa tahapan utama yaitu *build*, *test*, dan *deploy*. Pada proyek ini, tahap *build* berfungsi untuk menginstal seluruh dependensi aplikasi seperti menjalankan composer install. Tahap *test* digunakan untuk menjalankan pengujian sistem, misalnya dengan perintah `php artisan test`, guna memastikan tidak ada error setelah perubahan kode. Sedangkan tahap *deploy* bertugas menerapkan aplikasi ke server produksi dengan menarik kode terbaru dari branch utama dan menjalankan migrasi database, sehingga aplikasi selalu dalam kondisi terbaru dan stabil.

5.3 Contoh Konfigurasi

Konfigurasi pipeline CI/CD dituliskan dalam file `.gitlab-ci.yml` yang berisi definisi tahapan dan perintah yang dijalankan secara otomatis. Pada proyek Web Peminjaman Laptop, file ini mendefinisikan tiga stage yaitu *build*, *test*, dan *deploy*. Masing-masing stage memiliki script tertentu, seperti instalasi dependensi pada tahap *build*, pengujian aplikasi pada tahap *test*, serta proses deployment menggunakan SSH ke server pada tahap *deploy*. Dengan konfigurasi ini, setiap perubahan pada repository akan langsung memicu pipeline secara otomatis tanpa perlu intervensi manual.

5.4 Keamanan Pipeline

Keamanan dalam pipeline CI/CD sangat penting untuk mencegah kebocoran data dan akses tidak sah. Pada proyek ini, kredensial seperti password server atau token akses tidak boleh disimpan langsung di dalam file konfigurasi, melainkan menggunakan fitur GitLab CI/CD Variables agar data sensitif tetap aman. Selain itu,

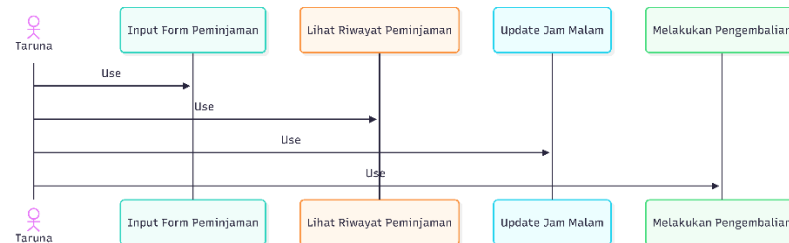
penerapan mekanisme *code review* dan *merge request approval* juga diperlukan untuk memastikan setiap perubahan kode telah ditinjau sebelum digabungkan ke branch utama, sehingga stabilitas dan keamanan aplikasi tetap terjaga.

BAB VI

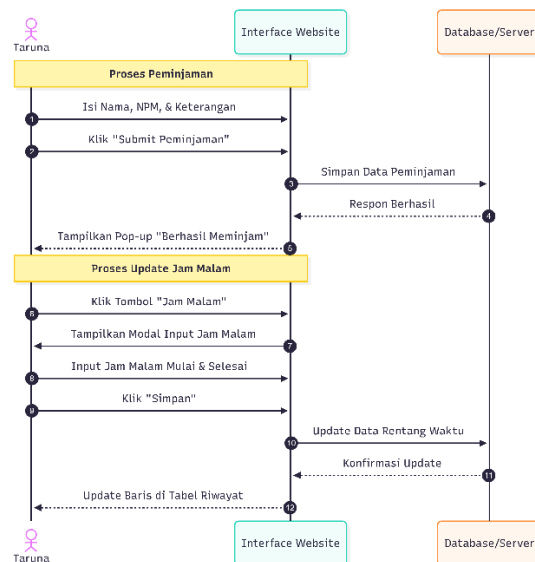
DESAIN SISTEM DAN TOOLS PENDUKUNG

6.1 Desain Sistem

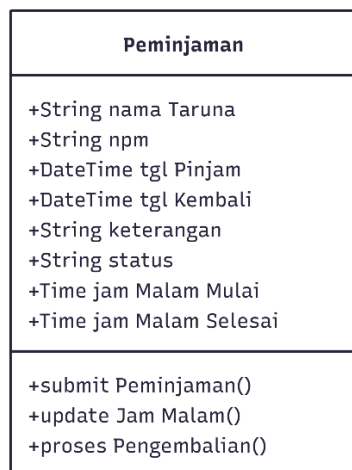
a. Usecase Diagram



b. Sequence Diagram



c. Class Diagram



6.2 Tools Desain Visual dan Diagram

Dalam pengembangan dan perancangan sistem Web Peminjaman Laptop, beberapa tools digunakan untuk mendukung pembuatan desain dan dokumentasi. Diagram UML dan ERD dapat dibuat menggunakan tools seperti Visual Paradigm atau Draw.io untuk memvisualisasikan struktur dan alur sistem. Untuk desain antarmuka pengguna (UI/UX), tools seperti Figma digunakan guna merancang tampilan halaman login, dashboard admin, halaman peminjaman, serta form pengajuan secara interaktif sebelum diimplementasikan ke dalam kode program. Selain itu, Lucidchart juga dapat dimanfaatkan untuk membuat visualisasi arsitektur sistem secara keseluruhan agar mempermudah pemahaman struktur aplikasi berbasis web ini.

6.3 Prinsip Desain Aman

Dalam perancangan sistem Web Peminjaman Laptop, aspek keamanan menjadi bagian penting agar data pengguna dan transaksi tetap terlindungi. Sistem menerapkan mekanisme autentikasi login untuk membatasi akses antara admin dan user. Data sensitif seperti password pengguna disimpan dalam bentuk terenkripsi untuk mencegah penyalahgunaan. Selain itu, sistem membatasi hak akses berdasarkan peran pengguna sehingga hanya admin yang dapat mengelola data laptop dan menyetujui peminjaman. Pengelolaan sesi (*session management*) juga diterapkan untuk menjaga keamanan selama pengguna mengakses sistem, serta meminimalkan risiko akses tidak sah terhadap data peminjaman dan informasi pengguna.

BAB VII

LEARNING PATH UNTUK PROYEK AKHIR

7.1 Tujuan Learning Path

Learning Path ini disusun untuk memastikan mahasiswa menguasai seluruh kompetensi yang dibutuhkan dalam mengembangkan aplikasi berbasis web seperti proyek Web Peminjaman Laptop. Tujuannya adalah agar mahasiswa memahami konsep arsitektur aplikasi, mampu mengimplementasikan sistem menggunakan framework backend, mengelola database, menerapkan keamanan aplikasi, serta memahami proses deployment dan pengelolaan proyek menggunakan version control. Dengan alur pembelajaran yang terstruktur, mahasiswa diharapkan mampu membangun aplikasi yang fungsional, aman, dan siap digunakan.

7.2 Struktur Pembelajaran

1) Dasar-dasar Arsitektur Arsitektur

Mahasiswa perlu memahami konsep dasar arsitektur client-server, di mana browser sebagai client berinteraksi dengan server untuk memproses data. Selain itu, pemahaman tentang pola arsitektur seperti MVC (Model-View-Controller) sangat penting karena proyek Web Peminjaman Laptop menggunakan struktur pemisahan antara logika bisnis, tampilan, dan pengelolaan data. Pemahaman mengenai alur request-response serta komunikasi dengan database juga menjadi bagian utama dalam tahap ini.

2) Penguasaan Development Environment

Tahap ini mencakup instalasi dan konfigurasi lingkungan pengembangan seperti PHP, Laravel (jika digunakan), dan MySQL sebagai sistem manajemen basis data. Mahasiswa juga perlu memahami penggunaan Composer untuk manajemen dependensi serta konfigurasi server lokal seperti XAMPP atau Laragon. Penguasaan environment ini penting agar aplikasi dapat dijalankan, diuji, dan dikembangkan secara optimal.

3) Pemograman Web Lanjutan

Pada tahap ini mahasiswa mempelajari pengelolaan database relasional menggunakan MySQL, termasuk pembuatan tabel, relasi, dan query. Selain itu, pemahaman tentang validasi form, pengelolaan session login, serta pengolahan data CRUD (Create, Read, Update, Delete) menjadi kompetensi

utama karena fitur-fitur tersebut digunakan dalam sistem peminjaman laptop seperti pengajuan, persetujuan, dan pengelolaan data laptop.

4) Penerapan SDLC dan CI/CD

Mahasiswa perlu memahami tahapan Software Development Life Cycle (SDLC) mulai dari analisis, desain, implementasi, hingga pemeliharaan sistem. Selain itu, penggunaan Git sebagai version control sangat penting untuk mengelola perubahan kode. Implementasi CI/CD pipeline juga menjadi bagian pembelajaran agar proses build, testing, dan deployment aplikasi dapat dilakukan secara otomatis dan terstruktur.

5) Keamanan Aplikasi

Dalam pengembangan Web Peminjaman Laptop, aspek keamanan meliputi validasi input untuk mencegah kesalahan atau serangan seperti SQL Injection, pengelolaan autentikasi login, enkripsi password, serta manajemen hak akses antara admin dan user. Mahasiswa juga perlu memahami dasar analisis kerentanan aplikasi agar sistem yang dibangun lebih aman dan stabil.

7.3 Referensi Pembelajaran Online

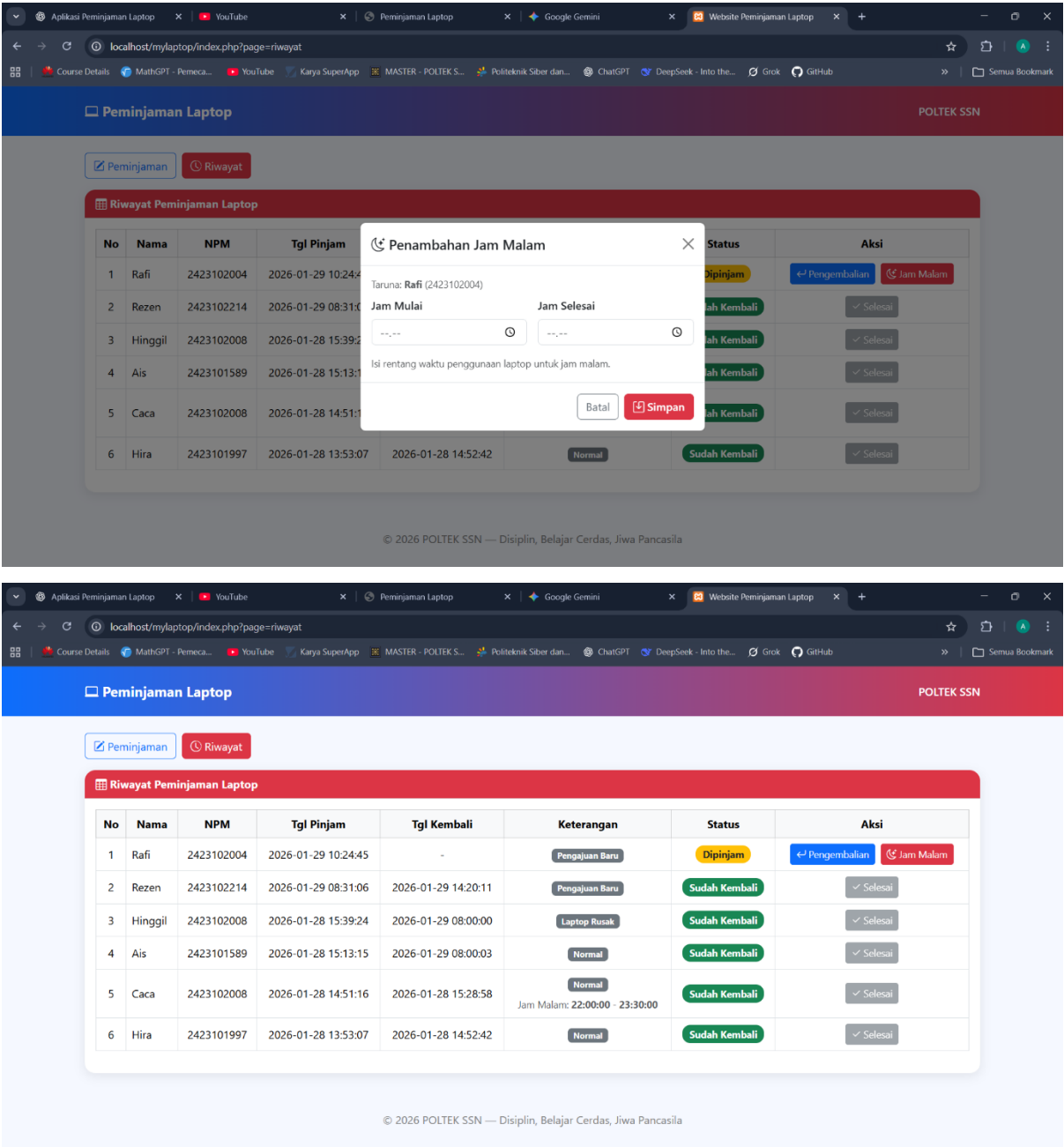
Beberapa referensi yang dapat digunakan untuk mendukung pengembangan proyek ini antara lain:

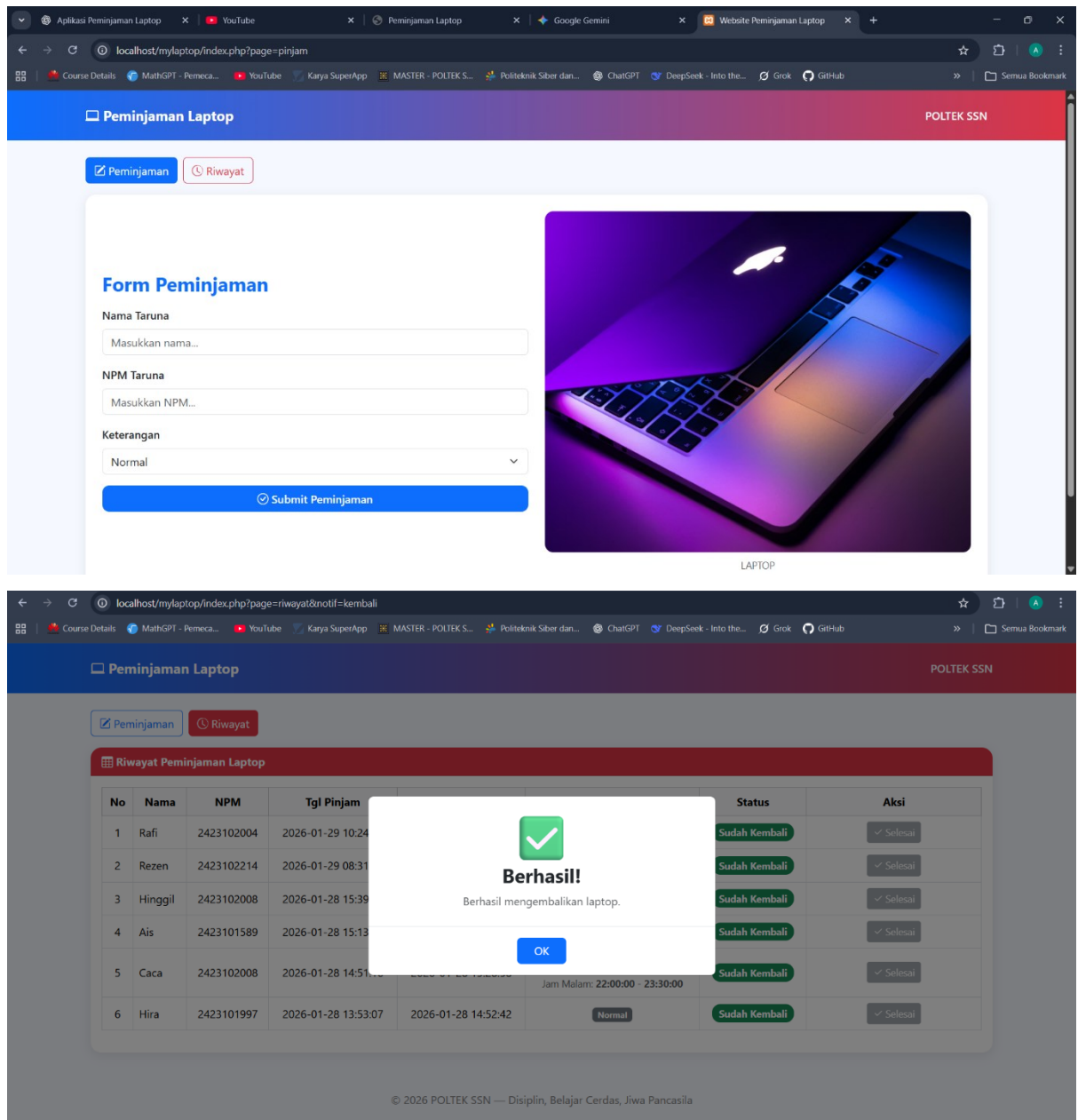
- Laravel Documentation atau dokumentasi resmi framework yang digunakan sebagai panduan backend.
- PHP Manual untuk memahami sintaks dan konsep dasar bahasa pemrograman PHP.
- W3Schools sebagai referensi dasar HTML, CSS, dan JavaScript untuk tampilan antarmuka.
- MySQL Documentation untuk pengelolaan database relasional.
- Git Documentation untuk memahami version control dan kolaborasi tim.
- GitLab Docs sebagai panduan implementasi CI/CD pipeline.

BAB VIII

IMPLEMENTASI DAN EVALUASI

8.1 Implementasi

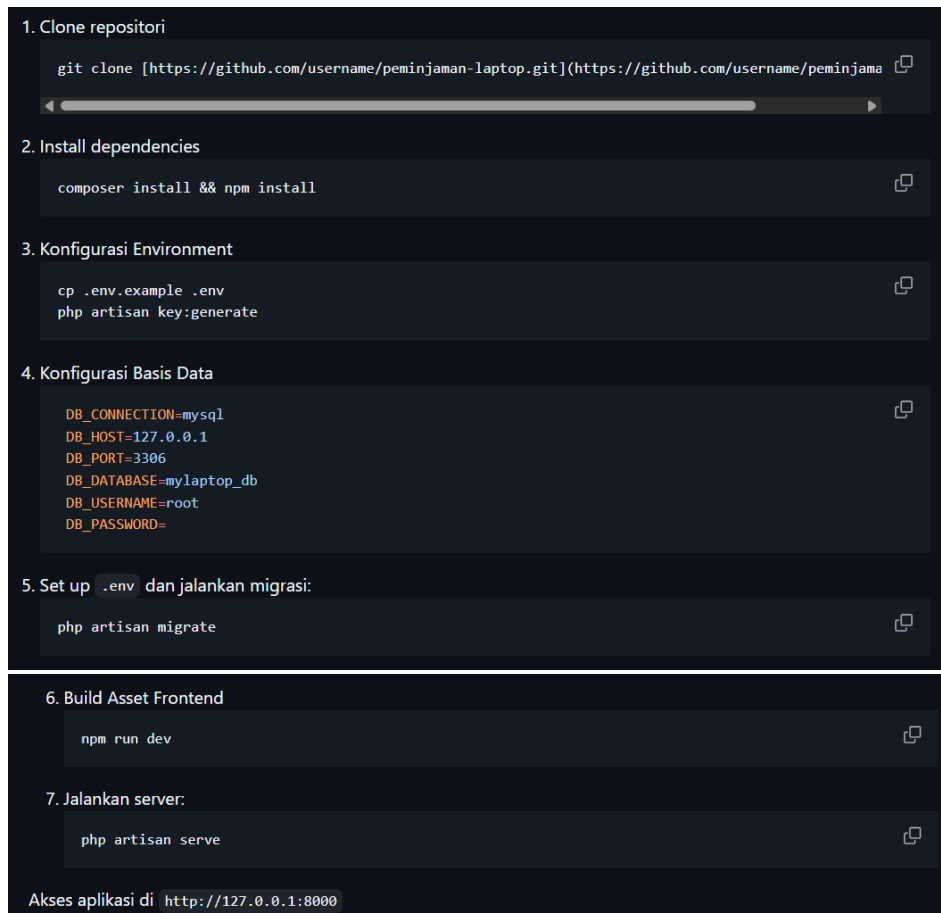




8.2 Pengujian

Pengujian dengan

- Framework: Laravel 10/11
- Frontend: Tailwind CSS / Bootstrap
- Database: MySQL
- Icons & Pop-ups: SweetAlert2 (Untuk notifikasi "Berhasil!") dan FontAwesome/Heroicons.



8.3 Evaluasi

Perlu adanya pengembangan lebih lanjut dan perbaruan sehingga bisa dilakukan dan dibuat menjadi aplikasi yang dapat dikases dan didownload.

DAFTAR PUSTAKA DAN REFERENSI ONLINE

- [1] Arcuri, A. (2019). RESTful API Automated Test Case Generation with EvoMaster. *ACM Transactions on Software Engineering and Methodology*, 28(1), 1–37.
<https://doi.org/10.1145/3293455>
- [2] Adiputra, F., & Mustofa, K. (2015). Purwarupa framework aplikasi desktop menggunakan teknologi web. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 9(1), 23. <https://doi.org/10.22146/ijccs.6637>
- [3] Primawanti, E. P., & Ali, H. (2022). PENGARUH TEKNOLOGI INFORMASI, SISTEM INFORMASI BERBASIS WEB DAN KNOWLEDGE MANAGEMENT TERHADAP KINERJA KARYAWAN (LITERATURE REVIEW EXECUTIVE SUPPORT SISTEM (ESS) FOR BUSINESS). *Jurnal Ekonomi Manajemen Sistem Informasi*, 3(3), 267–285. <https://doi.org/10.31933/jems.v3i3.818>
- [4] Somya, R., & Nathanael, T. M. E. (2019). PENGEMBANGAN SISTEM INFORMASI PELATIHAN BERBASIS WEB MENGGUNAKAN TEKNOLOGI WEB SERVICE DAN FRAMEWORK LARAVEL. *Jurnal Techno Nusa Mandiri*, 16(1), 51–58. <https://doi.org/10.33480/techno.v16i1.164>
- [5] Sondakh, D. E., Pungus, S. R., Runtukahu, P., & Saroinsong, R. (2016). IMPLEMENTASI TEKNOLOGI WEB SERVICES PADA APLIKASI PENCARIAN TAKSI. *CogITO Smart Journal*, 2(1), 1–13.
<https://doi.org/10.31154/cogito.v2i1.10.1-13>
- [6] Supriyono, H., Sutopo, A., Nursyahid, H., Kurniawan, B. A., Fahrudin, I. N., Handoko, D., Rivai, I., & Kurniawan, D. C. (2016). PENERAPAN TEKNOLOGI WEB SEKOLAH BAGI SMP DAN SMA MUHAMMADIYAH KARTASURA. *Warta LPM*, 19(1), 39–52. <https://doi.org/10.23917/warta.v19i1.1983>
- [7] <https://fip.unesa.ac.id/pengembangan-mobile-learning-tingkatkan-pemahaman-konsep-sejarah-indonesia-di-smk/>