**PROJEK KOPERASI ADAKITA**

Laporan ini disusun dalam rangka memenuhi Tugas Mata Kuliah, Rekayasa Perangkat Lunak, Pemrograman Berbasis Framework,

dan Pemrograman Mobile.



Nama Kelompok:

RAFI NAUFAL AMRULLOH 23416255201050

MUHAMMAD HAEKAL 23416255201066

FAUZY SUBHI NALENDRASIDI 23416255201067

SAFIN NUR IMANTORO 23416255201065

MUHAMAD ZAHRAWANI SYABAN 23416255201033

**TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BUANA PERJUANGAN KARAWANG**

**2024**

# BAB I PENDAHULUAN

## Latar Belakang

Di era digital saat ini, koperasi sebagai lembaga keuangan rakyat perlu melakukan transformasi digital agar tetap relevan dan eifisen. Banyak koperasi menghadapi tantangan dalam manajemen anggota, layanan simpan pinjam, hingga transparansi laporan keuangan. Oleh karena itu, dibutuhkan sebuah aplikasi koperasi berbasis mobile yang dapat menjawab kebutuhan tersebut secara terintegritasi.

Aplikasi AdaKita dikembangkan mengikuti tahapan Software Development Life Cycle (SDLC) untuk menjamin kualitas perangkat lunak yang tinggi. Berikut adalah tahapan-tahapan umum SDLC:

1. Perencanaan (Planning)
2. Analisis Kebutuhan (Requirement Analysis)
3. Desain Sistem (System Design)
4. Implementasi (Implementation/ Coding)
5. Pengujian (Testing)
6. Penerapan (Deployment)
7. Pemeliharaan (Maintenance)

## Tujuan

Aplikasi AdaKita bertujuan untuk memudahkan anggota koperasi dalam melakukan transaksi seperti simpanan, pinjaman, dan melihat laporan. Aplikasi ini juga memudahkan admin koperasi dalam mengelola data anggota, mencatat transaksi, serta menyusun laporan keuangan.

## Ruang Lingkup

Laporan ini mencakup beberapa aspek penting yang dibutuhkan dalam pengembangan aplikasi AdaKita berbasis mobile untuk Koperasi. Adapun ruang lingkup dari laporan ini meliputi:

1. **Analisis Kebutuhan**
   * 1. Identifikasi kebutuhan pengguna (user) dan pemilik koperasi
     2. Penentuan fitur-fitur utama yang akan dikembangkan dalam aplikasi.
2. **Desain Sistem**
   * 1. Desain arsitektur sistem yang menggabungkan framework Laravel untuk backend dan Ionic untuk frontend.
     2. Pembuatan diagram alir dan skema database untuk aplikasi.
     3. Desain antarmuka pengguna (UI) yang responsif dan user-friendly.
3. **Pengembangan Aplikasi**
   * 1. Implementasi backend menggunakan framework Laravel.
     2. Implementasi frontend mobile menggunakan framework Ionic.
     3. Integrasi antara backend dan frontend.
4. **Pengujian Sistem**
   * 1. Pengujian fungsionalitas aplikasi untuk memastikan semua fitur berjalan sesuai dengan spesifikasi.
     2. Pengujian kompatibilitas aplikasi pada berbagai perangkat mobile.
     3. Pengujian performa untuk memastikan aplikasi berjalan dengan lancar tanpa hambatan.
5. **Peluncuran dan Pemeliharaan**
   * 1. Strategi peluncuran aplikasi ke pasar.
     2. Rencana pemeliharaan dan pembaruan sistem secara berkala.
     3. Pelatihan dan dukungan teknis untuk pemilik koperasi dan pengguna.
6. **Evaluasi dan Analisis**
   * 1. Evaluasi efektivitas aplikasi dalam memudahkan peminjaman dan efisiensi penggunaan.
     2. Analisis feedback pengguna untuk perbaikan dan pengembangan lebih lanjut.

Ruang lingkup ini dibatasi pada pengembangan aplikasi Koperasi berbasis mobile untuk Koperasi AdaKita, tanpa mencakup aspek-aspek lain seperti manajemen inventaris yang kompleks, integrasi dengan sistem pembayaran eksternal yang tidak standar, atau fitur logistik dan pengiriman yang lebih rinci.

# BAB II PEMBAHASAN

## Analisis Kebutuhan

Dalam pengembangan aplikasi AdaKita berbasis mobile untuk Koperasi , terdapat beberapa kebutuhan fungsional dan non-fungsional yang harus dipenuhi. Selain itu, terdapat kebutuhan spesifik yang berkaitan dengan framework Laravel dan Ionic, serta kebutuhan khusus untuk pengembangan aplikasi mobile.

**2.1.1 Kebutuhan Fungsional**

1. Anggota Koperasi
   1. Anggota dapat masuk (login) ke aplikasi menggunakan email dan kata sandi.
   2. Anggota dapat melihat ringkasan saldo simpanan dan status pinjaman melalui halaman dashboard.
   3. Anggota dapat mengakses menu Simpanan, Pinjaman, Riwayat Transaksi, dan Ajukan Pinjaman.
   4. Di halaman Simpanan, anggota dapat melihat detail Simpanan Pokok, Simpanan Wajib, dan Simpanan Sukarela.
   5. Di halaman Pinjaman, anggota dapat melihat status pinjaman aktif, jadwal angsuran, dan denda (jika ada).
   6. Di halaman riwayat, anggota dapat melihat seluruh transaksi, seperti simpanan, pembayaran, angsuran, dan pengajuan pinjaman.
   7. Anggota dapat mengajukan pinjaman dengan mengisi nominal dan jangka waktu angsuran, lalu menekan tombol “Ajukan”.
2. Kolektor
   1. Kolektor dapat login menggunakan email dan kata sandi.
   2. Kolektor dapat melihat daftar anggota binaan beserta status pinjaman masing-masing.
   3. Kolektor dapat menginput pembayaran dari anggota dengan memasukan nominal, metode pembayaran, dan tanggal transaksi.
   4. Sistem menyediakan fitur geotagging otomatis untuk mencatat lokasi saat kolektor melakukan input pembayaran.
   5. Kolektor dapat melihat riwayat setoran dalam bentuk ringkasan harian dan bulanan.

**2.1.2 Kebutuhan Non-Fungsional**

1. Keamanan
   1. Sistem harus dapat mengamankan data pengguna dan transaksi secara terenkripsi.
   2. Proses authentikasi harus menggunakan mekanisme yang aman dan sesuai standar.
2. Kinerja
   1. Aplikasi harus dapat menampilkan data dengan cepat dan responsif.
   2. Waktu muat (loading time) aplikasi tidak boleh menggangu kenyamanan pengguna.
3. Usability
   1. Antarmuka pengguna (UI) harus mudah digunakan dan dipahami.
   2. Aplikasi harus dapat berjalan di berbagai ukuran layar dan perangkat.
4. Portabilitas
   1. Aplikasi harus dapat digunakan di berbagai perangkat mobile berbasis Android dan IOS.
   2. Proses pemeliharan dan pembaruan aplikasi harus mudah dilakukan oleh pengembang.
5. Reliabilitas
   1. Aplikasi harus stabil dan memiliki tingkat kesalahan (bug) yang rendah.
   2. Sistem harus memiliki mekanisme pemulihan jika terjadi kegagalan atau gangguan sistem.

**2.1.3 Kebutuhan Spesifik Terkait Framework Laravel dan Ionic**

1. Laravel (Backend)
   1. Menyediakan RESTful API untuk mengelola data simpanan, pinjaman, pengajuan pinjaman, transaksi, serta data kolektor dan anggota.
   2. Menggunakan middleware untuk keperluan autentikasi dan otorisasi pengguna.
   3. Membuat migrasi untuk pengelolaan struktur database yang efisien.
   4. Menyediakan validasi data yang masuk agar sesuai dengan format dan aturan yang telah ditentukan.
2. Ionic (Frontend)
   1. Menggunakan komponen antarmuka dari framework ionic untuk membangun UI yang responsif.
   2. Melakukan integrasi API dari backend agar dapat ditampilkan dan dikirim sesuai kebutuhan.
   3. Mengatur statuis data aplikasi menggunakan sistem manajemen state dari Angular atau ionic.

**2.1.4 Kebutuhan Khusus untuk Pengembangan Aplikasi Mobile**

1. Kompatibilitas Perangkat
   1. Aplikasi harus dapat dijalankan di sistem operasi Android dan IOS dengan versi yang masih layak digunakan.
   2. Aplikasi harus diuji pada berbagai resolusi dan ukuran layar.
2. Kinerja Mobile
   1. Aplikasi harus tetap berjalan dengan baik pada perangkat dengan spesifikasi rendah.
   2. Pengguna memori harus efisien untuk menghindari aplikasi tertutup secara tiba-tiba.
3. Interaksi Pengguna
   1. Navigasi antarmuka harus intuitif sesuai dengan pedoman desain aplikasi mobile.
   2. Komponen UI harus mudah diakses dan mendukung interaksi berbasis sentuhan (touchscreen).

Dengan dilakukannya analsis kebutuhan ini, pengembangan aplikasi AdaKita diharapkan dapat lebih terarah dan menghasilkan sistem yang sesuai dengan kebutuhan pengguna, baik anggota koperasi maupun kolektor. Pemenuhan kebutuhan ini juga akan membantu koperasi dalam meningkatkan efisiensi operasional serta memperkuat kepercayaan anggota melaui transparansi dan kemudahan layanan.

## Perancangan Sistem

## BAB III PENUTUP

## Kesimpulan

Ringkasan hasil proyek, pembahasan tentang pencapaian tujuan

## Saran

Saran untuk peningkatan di masa depan

# DAFTAR PUSTAKA

Referensi dan sumber daya yang digunakan dalam pengembangan proyek