

LECTURE NOTES

Specialized Platform Development

Deployment and Maintenance of Web
and Mobile Applications

Pualam Dipa Nusantara, S.Kom., M.Kom.



Learning Outcomes

- LO 4 : Evaluate security, performance, and regulatory challenges in the development and deployment of specialized platforms, applying best practices for risk mitigation.

Outline Materi

1. Introduction
2. CI/CD for Web and Mobile (GitHub Actions, CircleCI, Bitrise)
3. Build Optimization and Packaging
4. App Store Deployment (iOS & Android Guidelines)
5. Web App Deployment (Vercel, Netlify, Heroku)
6. Maintenance, Monitoring, and Analytics
7. Kesimpulan

10.1

Introduction

Dalam pengembangan perangkat lunak modern, proses deployment dan pemeliharaan (maintenance) memainkan peran krusial dalam memastikan aplikasi tetap berjalan secara optimal, aman, dan dapat diakses oleh pengguna akhir. Fokus utama dalam topik minggu ini adalah pada praktik terbaik dalam menerapkan aplikasi ke lingkungan produksi serta menjaga kinerjanya secara berkelanjutan.

Topik yang akan dibahas:

- CI/CD Workflows (Continuous Integration / Continuous Deployment)

CI/CD adalah pendekatan otomatisasi dalam proses integrasi dan distribusi kode. Dengan CI, setiap perubahan kode yang dikirimkan oleh tim pengembang akan diuji secara otomatis. CD melanjutkan proses ini hingga aplikasi di-deploy ke server atau cloud secara otomatis dan aman. Ini meningkatkan kecepatan pengembangan dan meminimalkan risiko kesalahan manual.

- Build Optimization

Optimisasi build bertujuan untuk mempercepat waktu pembuatan aplikasi dan mengurangi ukuran file akhir (bundle size), yang berdampak pada kecepatan akses pengguna. Teknik seperti tree shaking, code splitting, lazy loading, dan image compression akan dibahas untuk memastikan aplikasi ringan dan cepat dimuat.

- Deployment (App/Web)

Proses deployment mencakup pemindahan aplikasi dari lingkungan pengembangan ke produksi. Untuk aplikasi web, hal ini bisa dilakukan melalui layanan seperti Vercel, Netlify, atau Firebase Hosting. Untuk aplikasi mobile, deployment dilakukan ke Google Play Store dan Apple App Store dengan mematuhi standar yang ketat.

- Maintenance and Monitoring Tools

Setelah aplikasi berjalan di lingkungan produksi, penting untuk melakukan pemeliharaan secara berkala dan memantau kinerjanya. Tools seperti Firebase Performance Monitoring, Sentry, dan LogRocket akan diperkenalkan untuk mendeteksi error, mengukur performa, serta memastikan aplikasi bebas dari masalah yang mengganggu pengalaman pengguna.

Melalui topik-topik ini, mahasiswa diharapkan dapat memahami pentingnya automasi proses deployment, strategi optimasi performa, serta penggunaan alat bantu untuk mendukung keberlangsungan aplikasi di dunia nyata.

Belajar Mandiri

Pelajari secara mandiri terkait dengan materi ini pada PPT dan video yang ada di LMS pada sesi 10. Selain itu, rekan-rekan juga dapat menonton video berikut ini :

<https://www.youtube.com/watch?v=RIX4ufelA58>

CI/CD for Web and Mobile (GitHub Actions, CircleCI, Bitrise)

Continuous Integration (CI) dan Continuous Deployment (CD) adalah praktik penting dalam DevOps modern yang membantu tim pengembang melakukan otomatisasi dalam pengujian, build, dan penyebaran aplikasi. Dalam konteks pengembangan aplikasi web dan mobile, CI/CD tidak hanya meningkatkan efisiensi tim tetapi juga mengurangi risiko error di lingkungan produksi.

1. GitHub Actions

GitHub Actions memungkinkan otomatisasi berbagai proses seperti testing, build, dan deployment menggunakan file konfigurasi berbasis YAML. Dengan mengatur workflow, pengembang dapat menjalankan serangkaian perintah setiap kali ada perubahan pada kode, misalnya :

- Menjalankan unit test saat ada push ke repositori.
- Build aplikasi secara otomatis.
- Deploy ke server Firebase atau layanan hosting lainnya.

Keunggulan GitHub Actions :

- Native ke platform GitHub.
- Konfigurasi fleksibel menggunakan YAML.
- Mendukung lintas platform dan lintas bahasa pemrograman.

2. CircleCI

CircleCI adalah platform CI/CD yang sangat skalabel dan mendukung caching pipeline untuk mempercepat proses build. Pipeline ini memungkinkan pengembang membuat job paralel dan mengelola workflow yang kompleks dengan efisien.

Fitur utama :

- Eksekusi paralel dan caching yang mempercepat build.
- Integrasi dengan berbagai VCS seperti GitHub dan Bitbucket.
- Dukungan untuk Docker, macOS, dan Linux.

3. Bitrise

Bitrise adalah layanan CI/CD yang dirancang khusus untuk aplikasi mobile, baik Android maupun iOS. Bitrise menyediakan berbagai step template siap pakai untuk testing, build, dan deployment ke App Store dan Google Play.

Kelebihan Bitrise :

- Support native Android & iOS, React Native, Flutter.
- Mendukung automatic provisioning untuk iOS.
- Integrasi dengan Firebase App Distribution, Slack, dan lainnya.

Studi Kasus : Deployment Otomatis

Seorang developer melakukan perubahan pada fitur login di repositori GitHub. GitHub Actions secara otomatis menjalankan test, lalu melakukan build, dan jika lulus, CircleCI atau Bitrise akan men-deploy versi terbaru ke Firebase Hosting atau mengirim ke Play Store/App Store untuk pengujian.

Bahan Diskusi



Silahkan diskusikan bersama dosen/tutor/rekan mahasiswa mengenai:

- Bagaimana GitHub Actions dapat mengotomasi proses build dan testing aplikasi web dengan workflow YAML?
- Apa kelebihan CircleCI dalam menjalankan pipeline CI/CD yang scalable untuk proyek skala besar?
- Mengapa Bitrise menjadi pilihan populer untuk CI/CD di pengembangan aplikasi mobile iOS dan Android?

Build Optimization and Packaging

Optimasi build adalah proses penting dalam pengembangan perangkat lunak untuk memastikan aplikasi berjalan lebih cepat, efisien, dan memiliki ukuran distribusi yang minimal. Hal ini tidak hanya memengaruhi performa aplikasi di sisi klien, tetapi juga berdampak pada waktu unduh, konsumsi bandwidth, dan pengalaman pengguna secara keseluruhan.

1. Optimasi Bundle Menggunakan Webpack

Webpack adalah module bundler yang umum digunakan dalam pengembangan aplikasi berbasis JavaScript seperti React atau Vue. Dua teknik utama yang digunakan untuk optimasi adalah :

- Tree Shaking : Menghapus kode JavaScript yang tidak digunakan dari bundle. Ini membantu mengecilkan ukuran file akhir.
- Minification : Menghapus whitespace, komentar, dan memperpendek nama variabel untuk mengurangi ukuran file.

Contoh : Tanpa optimasi, aplikasi React bisa berukuran 3MB. Dengan tree shaking dan minification, ukurannya dapat dikurangi hingga 800KB.

2. Optimasi Build pada Aplikasi Mobile

Untuk aplikasi Android dan iOS, ukuran file APK (Android) dan IPA (iOS) sangat penting karena memengaruhi :

- Waktu instalasi
- Konsumsi penyimpanan pengguna
- Kecepatan loading

Langkah-langkah yang umum digunakan :

- Mengompres gambar menggunakan format WebP atau alat seperti ImageOptim.
- Menghapus resource yang tidak terpakai, seperti layout, style, dan font yang tidak digunakan.
- Mengaktifkan ProGuard atau R8 untuk mengurangi dan mengaburkan kode dalam aplikasi Android.

3. Lazy Loading dan Code Splitting

- Lazy Loading memungkinkan modul hanya dimuat saat dibutuhkan oleh pengguna. Ini mengurangi waktu loading awal.
- Code Splitting membagi kode aplikasi menjadi beberapa file (chunk) yang lebih kecil. Artinya, pengguna hanya akan mengunduh bagian dari aplikasi yang benar-benar diperlukan saat itu.

Keuntungan : Meningkatkan First Contentful Paint (FCP) dan mengurangi Time to Interactive (TTI).

Studi Kasus :

Sebuah aplikasi React yang awalnya memiliki bundle utama sebesar 3MB dapat dioptimalkan menggunakan tree shaking, code splitting, dan image compression menjadi hanya 800KB. Hasilnya :

- Waktu loading menurun hingga 60%.
- Skor Lighthouse Performance meningkat secara signifikan.
- Pengguna merasakan pengalaman yang lebih cepat dan mulus.

Bahan Diskusi



Silahkan diskusikan bersama dosen/tutor/rekan mahasiswa mengenai:

- Bagaimana teknik tree-shaking dan code splitting dapat mengurangi ukuran bundle aplikasi web secara signifikan?
- Apa peran Webpack, Babel, dan Metro bundler dalam proses packaging aplikasi modern?
- Mengapa penting melakukan minifikasi, kompresi, dan caching saat men-deploy aplikasi ke production?

10.4

App Store Deployment (iOS and Android Guidelines)

Proses deployment aplikasi ke App Store (Google Play Store untuk Android dan App Store untuk iOS) adalah tahapan akhir dari pengembangan aplikasi mobile. Langkah ini tidak hanya berkaitan dengan mengunggah file aplikasi, tetapi juga memastikan bahwa aplikasi memenuhi seluruh persyaratan teknis, konten, dan legal yang telah ditetapkan oleh masing-masing platform.

1. Deployment untuk Android

Untuk mengunggah aplikasi Android ke Play Store, pengembang harus menggunakan :

- Google Play Console

Platform resmi dari Google untuk mengelola aplikasi Android, mulai dari upload APK/AAB, pengujian beta, hingga pelaporan performa.

- Format File

- APK (Android Package): Format file instalasi lama.

- AAB (Android App Bundle): Format baru yang lebih efisien dan direkomendasikan oleh Google.

Langkah umum :

1. Buat akun developer di Google Play Console.
2. Siapkan file AAB/APK, ikon, deskripsi, dan screenshot aplikasi.
3. Lengkapi form privasi dan kebijakan konten.
4. Submit aplikasi untuk ditinjau oleh Google.

2. Deployment untuk iOS

Untuk platform iOS, proses deployment sedikit lebih ketat dan harus dilakukan melalui :

- Xcode : Digunakan untuk membangun dan menandatangani aplikasi sebelum dikirim.
- App Store Connect : Portal resmi Apple untuk mengunggah build, menambahkan metadata aplikasi, dan mengatur distribusi.

Langkah umum :

1. Daftar sebagai Apple Developer.
2. Buat provisioning profile dan sertifikat distribusi.
3. Build aplikasi menggunakan Xcode.
4. Upload ke App Store Connect.
5. Lengkapi metadata, screenshot, dan informasi kebijakan privasi.
6. Submit untuk review oleh Apple.

3. Panduan Umum (Guidelines)

Baik Google maupun Apple menetapkan sejumlah kebijakan dan panduan teknis seperti :

1. Kebijakan Konten : Tidak boleh mengandung konten berbahaya, kebencian, atau pornografi.
2. Privasi : Harus menyertakan kebijakan privasi dan menginformasikan pengguna terkait penggunaan data.
3. Screenshot dan Metadata : Harus akurat dan mencerminkan fungsi sebenarnya dari aplikasi.
4. Pengujian (Testing) : Apple sangat menekankan kualitas dan kestabilan. Crash saat proses review bisa menyebabkan penolakan langsung.

Bahan Diskusi



Silahkan diskusikan bersama dosen/tutor/rekan mahasiswa mengenai:

- Apa saja persyaratan utama dari App Store dan Google Play Store yang harus dipenuhi sebelum mengirim aplikasi untuk ditinjau?
- Bagaimana proses peninjauan dan persetujuan aplikasi berbeda antara platform iOS dan Android?
- Apa tantangan umum yang dihadapi pengembang saat melakukan deployment dan bagaimana cara menghindari penolakan aplikasi oleh Apple atau Google?

Web App Deployment (Vercel, Netlify, Heroku)

Dalam pengembangan aplikasi web modern, deployment adalah proses akhir untuk membuat aplikasi dapat diakses oleh pengguna secara publik. Terdapat berbagai layanan yang dapat digunakan untuk proses ini, tergantung pada kebutuhan proyek frontend dan backend.

1. Vercel

Vercel adalah platform deployment yang dirancang khusus untuk aplikasi frontend berbasis framework JavaScript modern, seperti :

- Next.js (didukung langsung oleh Vercel)
- React, Vue, Angular, Svelte

Keunggulan Vercel :

- Deployment sangat cepat dan otomatis melalui GitHub.
- Mendukung fitur Preview URL untuk setiap pull request.
- Optimasi performa otomatis (caching, CDN global).
- Cocok untuk proyek frontend-only atau aplikasi statis dan SSR.

2. Netlify

Netlify adalah platform serupa Vercel, digunakan untuk aplikasi statis dan frontend.

Fitur unggulan Netlify :

Static Hosting : ideal untuk proyek HTML/CSS/JS biasa maupun framework frontend modern.

Form support : menangani form HTML tanpa perlu backend.

Preview URL : mirip dengan Vercel, untuk setiap commit/pull request.

Plugin & Build Automation : integrasi otomatis dengan pipeline Git.

Keunggulan :

- Sederhana untuk pemula.
- Gratis dengan batasan yang cukup besar untuk proyek kecil hingga menengah.

3. Heroku

Heroku adalah Platform-as-a-Service (PaaS) yang mendukung berbagai bahasa pemrograman, terutama :

- Node.js (Express.js)
- Python, Ruby, PHP, Java, Go, dll.

Cocok untuk :

- Men-deploy backend (REST API, GraphQL, service handler).
- Menjalankan aplikasi server-side yang memerlukan database, autentikasi, dsb.

Keunggulan :

- Instalasi mudah.
- Mendukung PostgreSQL, Redis, dan add-ons lainnya.
- Git-based deployment.

Contoh Arsitektur Deployment

Untuk aplikasi fullstack, pendekatan umum adalah memisahkan frontend dan backend :

- Frontend : dideploy di Vercel atau Netlify
- Backend : dideploy di Heroku atau platform server lain
- Komunikasi : dilakukan melalui REST API atau GraphQL

Memilih platform deployment yang sesuai dengan kebutuhan aplikasi adalah langkah strategis untuk :

- Mempersingkat waktu rilis (time-to-market)
- Menyederhanakan proses integrasi dan pengujian
- Mengurangi biaya pengembangan dan pemeliharaan
- Meningkatkan performa dan pengalaman pengguna

Bahan Diskusi



Silahkan diskusikan bersama dosen/tutor/rekan mahasiswa mengenai:

- Apa kelebihan dan kekurangan masing-masing platform (Vercel, Netlify, Heroku) dalam proses deployment aplikasi web modern?
- Bagaimana cara kerja pipeline deployment otomatis di Vercel dan Netlify dibandingkan dengan pendekatan manual atau semi-otomatis di Heroku?
- Kapan sebaiknya memilih Heroku untuk backend deployment dibandingkan menggunakan layanan serverless seperti Netlify Functions atau Vercel Edge Functions?

Maintenance, Monitoring and Analytics

Setelah aplikasi berhasil dideploy, tanggung jawab pengembang tidak berhenti. Pemeliharaan, pemantauan (monitoring), dan analitik (analytics) merupakan langkah penting untuk memastikan aplikasi tetap berjalan dengan baik, stabil, dan memenuhi kebutuhan pengguna.

1. Error Tracking : Sentry dan Firebase Crashlytics

Tujuan : Mendeteksi dan menganalisis kesalahan (errors atau crash) yang terjadi pada aplikasi secara real-time.

Sentry

- Mendukung berbagai platform: JavaScript, React, Node.js, Python, dan lainnya.
- Menyediakan stack trace, log pengguna, dan waktu kejadian.
- Memungkinkan pelacakan kesalahan hingga baris kode.

Firebase Crashlytics

- Dikhususkan untuk aplikasi mobile (Android/iOS).
- Menampilkan crash report lengkap dengan device info, versi aplikasi, dan user sessions.
- Terintegrasi dengan Firebase Analytics dan Notifikasi.

Manfaat :

- Mempercepat proses identifikasi dan perbaikan bug.
- Meningkatkan kualitas dan keandalan aplikasi.

2. Logging : Pemantauan Aktivitas Sistem

Logging adalah proses mencatat aktivitas yang terjadi dalam aplikasi, termasuk :

- Timestamps : waktu kejadian
- Level error : info, warning, error, critical
- Aktivitas pengguna atau server : login, pembelian, kesalahan koneksi, dll.

Tools Logging Populer :

- Logcat (Android)
- LogRocket
- Winston (Node.js)
- Papertrail, Datadog, atau ELK Stack

Manfaat Logging :

- Audit trail untuk debugging dan keamanan.
- Melacak perilaku aplikasi secara historis.

3. Analytics: Melacak Perilaku Pengguna

Analytics membantu memahami bagaimana pengguna berinteraksi dengan aplikasi.

Tools yang Umum Digunakan :

- Google Analytics
 - Pelacakan halaman, sumber traffic, lokasi pengguna.
- Mixpanel
 - Fokus pada user flow dan event-driven tracking.
 - Menyediakan visualisasi funnel dan retensi.

Manfaat Analytics :

- Memahami fitur yang paling sering digunakan.
- Mengukur konversi dan efektivitas UI/UX.
- Mendukung pengambilan keputusan berbasis data.

| Aktivitas | Tools Utama | Tujuan |
|-----------------------|------------------------------|---|
| Error Tracking | Sentry, Firebase Crashlytics | Mendeteksi dan memperbaiki error secara proaktif |
| Logging Sistem | Logcat, LogRocket, Winston | Melacak aktivitas aplikasi dan sistem |
| Analisis Pengguna | Google Analytics, Mixpanel | Menganalisis interaksi pengguna untuk pengembangan produk |
| Monitoring Notifikasi | Real-time alert, dashboard | Menanggapi insiden dengan cepat |

Gambar 10.1 Rangkuman Best Practices

Pemeliharaan aplikasi bukan hanya tentang memperbaiki bug, tapi juga tentang memastikan performa optimal secara berkelanjutan. Dengan memanfaatkan alat pemantauan dan analitik yang tepat, pengembang dapat :

- Mendeteksi masalah lebih awal
- Menjaga pengalaman pengguna
- Membuat keputusan pengembangan yang lebih cerdas

Belajar Mandiri



Pelajari secara mandiri terkait dengan ERP pada:

- Buku Chapter 7 berikut ini:

Mobile DevOps Playbook: A practical guide for delivering high-quality mobile applications like a pro

Bahan Diskusi



Silahkan diskusikan bersama dosen/tutor/rekan mahasiswa mengenai:

- Bagaimana peran pemantauan performa dan error logging dalam menjaga stabilitas aplikasi setelah deployment?
- Apa saja metrik penting yang perlu dianalisis secara rutin untuk memahami perilaku pengguna dan meningkatkan UX?
- Strategi apa yang paling efektif untuk merencanakan maintenance berkala tanpa mengganggu pengalaman pengguna aktif?

Evaluasi Individu



Kerjakan individual assignment di LMS BINUS ONLINE pada sesi-22; nomor 3

Evaluasi Kelompok



Kerjakan team assignment di LMS BINUS ONLINE pada sesi-21; nomor 4

Post-Test

Setelah menyelesaikan materi ini, silahkan kerjakan soal berikut ini:

1. Tujuan utama dari penerapan CI/CD workflows dalam proses deployment adalah ...
 - a. Menghapus kebutuhan dokumentasi
 - b. Mengotomatiskan proses build, testing, dan deployment aplikasi
 - c. Mengurangi ukuran file source code
 - d. Membatasi aplikasi hanya berjalan di satu platform
2. Platform berikut ini sering digunakan untuk deployment aplikasi frontend modern (Next.js, React, dsb.), kecuali ...
 - a. Vercel
 - b. Netlify
 - c. Heroku
 - d. Photoshop
3. Teknik optimasi build seperti tree shaking dan code splitting bertujuan untuk ...
 - a. Meningkatkan keamanan API
 - b. Mengurangi ukuran file bundle dan mempercepat loading aplikasi
 - c. Membatasi penggunaan library eksternal
 - d. Menambahkan fitur enkripsi otomatis
4. Alat yang dapat digunakan untuk monitoring error dan logging aplikasi di tahap maintenance adalah ...
 - a. Sentry
 - b. Expo
 - c. Appium
 - d. Redux
5. Mengapa proses maintenance penting setelah aplikasi dideploy?
 - a. Agar aplikasi tidak perlu diuji ulang
 - b. Untuk memastikan aplikasi tetap stabil, aman, dan relevan seiring waktu
 - c. Supaya aplikasi hanya bisa diakses di perangkat tertentu
 - d. Untuk mengurangi biaya server sepenuhnya

Kesimpulan

Proses deployment dan maintenance merupakan tahap krusial untuk memastikan aplikasi web dan mobile dapat berjalan optimal, aman, dan berkelanjutan di lingkungan produksi. Dengan menerapkan CI/CD workflows (GitHub Actions, CircleCI, Bitrise), pengembang dapat mengotomatiskan pengujian, build, dan distribusi aplikasi sehingga lebih cepat dan minim risiko. Strategi build optimization seperti tree shaking, code splitting, dan kompresi sumber daya meningkatkan performa aplikasi, sementara deployment ke platform seperti Vercel, Netlify, Heroku, Google Play Store, dan Apple App Store memastikan aplikasi dapat diakses luas oleh pengguna. Selanjutnya, proses maintenance yang mencakup monitoring, error tracking, logging, dan analytics menggunakan alat seperti Firebase Performance Monitoring, Sentry, dan Google Analytics membantu menjaga kualitas aplikasi serta mendukung pengambilan keputusan berbasis data. Dengan pendekatan ini, pengembang dapat menghadirkan aplikasi yang stabil, efisien, dan relevan dalam jangka panjang.

Daftar Pustaka

1. Daniel Bugl.(2024).Modern Full-Stack React Projects: Build, maintain, and deploy modern web apps using MongoDB, Express, React, and Node.js
2. Moataz Nabil.(2023).Mobile DevOps Playbook: A practical guide for delivering high-quality mobile applications like a pro
3. <https://www.youtube.com/watch?v=RIX4ufelA58>

Kunci: b, d, b, a, b