

CI / CD

Introduction & Process Overview



CI / CD

- ☐ Pengenalan
- ☐ Integrasi Berkelanjutan (CI)
- ☐ Penerapan Berkelanjutan (CD)
- ☐ Gambaran CI / CD
- ☐ Case Study Sederhana

Objektif sesi

- Memahami Konsep CI/CD: Memperkenalkan dan memahami konsep Continuous Integration (CI) dan Continuous Deployment/Delivery (CD) sebagai bagian penting dari siklus pengembangan perangkat lunak modern.
- Menerapkan CI/CD pada Proyek React Native dengan Expo: Memberikan pengetahuan dan langkah-langkah praktis untuk mengimplementasikan pipeline CI/CD dalam proyek React Native yang menggunakan Expo.
- Automasi Proses Pengembangan: Menunjukkan cara mengotomatiskan proses build, pengujian, dan penerapan aplikasi untuk meningkatkan efisiensi, mengurangi kemungkinan kesalahan, dan mempercepat siklus rilis.
- Penggunaan Alat CI/CD (GitHub Actions): Memperkenalkan dan menggunakan GitHub Actions sebagai alat untuk mengatur dan menjalankan workflow CI/CD.

Pengenalan

CI/CD, singkatan dari Continuous Integration dan Continuous Deployment (atau Continuous Delivery), adalah metode pengembangan perangkat lunak yang dirancang untuk meningkatkan dan menyederhanakan proses integrasi perubahan dan penerapan aplikasi. Ini merupakan komponen penting dalam praktik DevOps modern dan bertujuan untuk membuat pengembangan perangkat lunak lebih efisien, andal, dan cepat.

Integrasi Berkelanjutan (CI)

Definisi: CI adalah praktik mengotomatiskan integrasi perubahan kode dari berbagai kontributor ke dalam satu proyek perangkat lunak. Ini terutama digunakan untuk mendeteksi masalah lebih awal dalam siklus pengembangan.

Integrasi Berkelanjutan (CI)

Komponen Utama:

Pengujian Otomatis: Saat kode diintegrasikan, tes otomatis dijalankan untuk memastikan bahwa kode baru tidak merusak fungsionalitas yang ada.

Kontrol Versi: Pengembang sering melakukan commit kode mereka ke repositori bersama (seperti Git) yang secara otomatis memicu proses CI.

Umpan Balik Segera: Pengembang menerima umpan balik segera tentang kode mereka, memungkinkan mereka untuk mengidentifikasi dan memperbaiki masalah dengan cepat.

Penerapan Berkelanjutan (CD)

Definisi: CD memperluas CI dengan secara otomatis menerapkan semua perubahan kode ke lingkungan pengujian dan/atau produksi setelah tahap build. Ada dua jenis:

- Penerapan Berkelanjutan: Setiap perubahan yang lulus tes otomatis diterapkan ke lingkungan produksi secara otomatis.
- Pengiriman Berkelanjutan: Proses penerapan diotomatiskan, tetapi ada langkah manual untuk memicu penerapan akhir ke produksi.

Penerapan Berkelanjutan (CD)

- **Komponen Utama:**

Penerapan Otomatis: Mengotomatiskan proses penerapan, mengurangi kemungkinan kesalahan manusia dan mempercepat proses rilis.

Manajemen Lingkungan: Memastikan konsistensi di berbagai lingkungan (pengembangan, staging, produksi) untuk pengujian dan penerapan.

Kemampuan Rollback: Kemampuan untuk cepat kembali ke versi sebelumnya jika penerapan baru gagal.

Gambaran Umum Proses CI/CD

Commit Kode: Pengembang melakukan commit kode ke sistem kontrol versi (seperti Git), yang berfungsi sebagai pemicu untuk pipeline CI/CD.

Build Otomatis: Server CI memantau repositori dan melakukan checkout perubahan saat terjadi. Kemudian mengkompilasi kode menjadi build.

Pengujian Otomatis: Build diuji menggunakan tes otomatis. Ini bisa mencakup unit test, integrasi test, dan jenis pengujian lainnya untuk memastikan kualitas.

Gambaran Umum Proses CI/CD

Tinjauan Hasil: Hasil tes ini kemudian ditinjau. Jika tes gagal, tim diberitahu, dan masalahnya perlu diatasi sebelum melanjutkan.

Penerapan ke Staging: Setelah build lulus semua tes, itu diterapkan ke lingkungan staging untuk pengujian dan validasi lebih lanjut.

Tinjauan Manual: Dalam Pengiriman Berkelanjutan, biasanya ada proses tinjauan manual sebelum penerapan akhir ke produksi.

Gambaran Umum Proses CI/CD

Penerapan ke Produksi: Dalam Penerapan Berkelanjutan, build yang sukses secara otomatis diterapkan ke produksi.

Pemantauan dan Umpan Balik: Setelah penerapan, pemantauan berkelanjutan sangat penting untuk mengidentifikasi masalah apa pun di lingkungan produksi.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

1. Setup Repositori GitHub

Pastikan proyek React Native Anda sudah berada di repositori GitHub.

2. Membuat Workflow CI/CD

Di dalam repositori GitHub Anda, buat direktori `.github/workflows`. Di dalam direktori ini, buat file YAML untuk mendefinisikan workflow CI/CD Anda. Misalnya, `ci-cd.yml`.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

3. Konfigurasi Workflow

Berikut adalah contoh dasar dari file ci-cd.yml yang melakukan build dan test:

```

1  name: React Native CI/CD
2
3  on:
4    push:
5      branches: [ main ]
6    pull_request:
7      branches: [ main ]
8
9  jobs:
10   build-and-test:
11     runs-on: ubuntu-latest
12     strategy:
13       matrix:
14         node-version: [14.x]
15
16     steps:
17       - uses: actions/checkout@v2
18
19       - name: Use Node.js ${{ matrix.node-version }}
20         uses: actions/setup-node@v2
21         with:
22           node-version: ${{ matrix.node-version }}
23
24       - name: Install dependencies
25         run: npm install
26
27       - name: Run tests
28         run: npm test
29
30       - name: Install Expo CLI
31         run: npm install -g expo-cli
32
33       - name: Expo Login
34         run: expo login --non-interactive -u ${{ secrets.EXPO_CLI_USERNAME }} -p ${{ secrets.EXPO_CLI_PASSWORD }}
35
36       - name: Expo Publish
37         run: expo publish
38

```

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

Nama Workflow

- name: React Native CI/CD: Menetapkan nama untuk workflow ini, yaitu "React Native CI/CD".

Trigger Workflow

- on: Menentukan kapan workflow harus dijalankan.
- push: Workflow dijalankan saat ada push ke branch main.
- pull_request: Workflow juga dijalankan saat ada pull request ke branch main.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

Pekerjaan (Jobs)

- build-and-test: Nama pekerjaan yang didefinisikan untuk workflow ini.
- runs-on: ubuntu-latest: Menentukan bahwa pekerjaan ini akan dijalankan pada runner terbaru dari Ubuntu.
- strategy: Mendefinisikan strategi untuk pekerjaan.
 - matrix: Memungkinkan Anda menjalankan pekerjaan pada berbagai versi environment. Dalam kasus ini, digunakan node-version: [14.x], artinya pekerjaan akan dijalankan pada versi Node.js 14.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

Langkah-langkah (Steps)

- `uses: actions/checkout@v2`: Menggunakan action checkout untuk mengambil repositori GitHub yang workflow-nya sedang dijalankan.
- `name: Use Node.js ${{ matrix.node-version }}`: Menyiapkan Node.js versi yang ditentukan dalam matrix.
 - `uses: actions/setup-node@v2`: Menggunakan action setup-node untuk mengatur Node.js.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

Langkah-langkah (Steps)

- name: Install dependencies: Langkah untuk menginstal dependensi NPM.
 - run: npm install: Menjalankan npm install untuk menginstal dependensi yang diperlukan.
- name: Run tests: Menjalankan tes yang ada di proyek.
 - run: npm test: Menjalankan perintah untuk menjalankan tes.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

Langkah-langkah (Steps)

- name: Install Expo CLI: Menginstal Expo CLI.
 - run: `npm install -g expo-cli`: Menjalankan perintah untuk menginstal Expo CLI secara global.
- name: Expo Login: Melakukan login ke Expo menggunakan akun Expo.
 - run: `expo login --non-interactive -u ${{secrets.EXPO_CLI_USERNAME}} -p ${{secrets.EXPO_CLI_PASSWORD}}`: Melakukan login ke Expo secara non-interaktif dengan menggunakan username dan password yang disimpan sebagai secrets di GitHub.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

Langkah-langkah (Steps)

- name: Expo Publish: Menerbitkan proyek ke Expo.
 - run: expo publish: Menjalankan perintah untuk menerbitkan proyek ke Expo.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

4. Konfigurasi Secrets

Anda harus mengatur EXPO_CLI_USERNAME dan EXPO_CLI_PASSWORD sebagai secrets di pengaturan repositori GitHub Anda untuk mengotomatiskan proses login Expo.

5. Penjelasan Workflow

Trigger: Workflow dijalankan pada push atau pull request ke branch main.

Build dan Test: Menginstal dependensi, menjalankan tes, dan memastikan bahwa build tidak memiliki error.

Expo CLI: Menginstal Expo CLI dan menggunakan credential untuk login dan publish proyek.

Contoh Pengaturan GitHub Actions untuk Proyek React Native dengan Expo

6. Testing

Setelah workflow CI/CD dikonfigurasi, setiap push ke branch main akan memicu proses CI/CD. Anda bisa melihat log dan status di tab 'Actions' di repositori GitHub Anda.



Thank you