E-ISSN : 2407 – 3911
P-ISSN : 2686 - 0333

IMPLEMENTASI CONTIONOUS INTEGRATION DAN CONTINOUS DEPLOYMENT PADA APLIKASI LEARNING MANAGEMENT SYSTEM DI PT. MILLENNIA SOLUSI INFORMATIKA

Indra Guna Noviantama [1], Ari Purno Wahyu W [2]

Fakultas Teknik, Prodi Informatika, Universitas Widyatama [1,2]
Jl. Cikutra No.204A, Sukapada, Cibeunying Kidul, Kota Bandung.

Email : indra.guna@widyatama.ac.id [1], ari.purno@widyatama.ac.id [2]

Abstrak

Aplikasi Learning Management System atau LMS merupakan produk aplikasi yang dikembangkan oleh PT. Millennia Solusi Informatika. Aplikasi LMS ini telah digunakan oleh salah satu jaringan sekolah swasta. Dalam pengembangannya, aplikasi ini menggunakan metode scrum dimana pendekatan metode ini bersifat *agile* dan dapat menyesuaikan kebutuhan dengan cepat. Berangkat dari hal tersebut maka dalam proses *delivery* perangkat lunak ini maka perlu menggunakan konsep *continuous integration* dan *continuous deployment* guna memenuhi alur pengembangan yang bersifat *agile* dan dapat berulang.

Continous Integration (CI) adalah pengintegrasian kode ke dalam repositori kode kemudian menjalankan penggunaan secara otomatis, cepat dan sering. Sementara *Continuous Deployment* atau *Continuous Delivery* (CD) adalah praktik yang dilakukan setelah proses CI selesai dan seluruh kode berhasil terintegrasi, sehingga aplikasi bisa dibangun dan dirilis secara otomatis.

Dengan menggunakan metode CI/CD diharapkan dalam proses penyampaian aplikasi dapat terus berlangsung otomatis, cepat dan sering walaupun aplikasi tersebut sudah digunakan oleh pengguna.

Kata Kunci : aplikasi, learning management system, Scrum, *agile*, *continuous integration*, *continuous deployment*

Abstract

Learning Management System or LMS is an application product developed by PT. Millennia Solusi Informatika. This LMS aplication has been used by a private school network. The application is developed using scrum method where this method approach is agile and can adjust requirements quickly. Based on this, in process of delivering the software, it is necessary to use the concepts of continuous integration and continuous deployment in order to fulfill an agile and iterative development flow.

Continuous integration (CI) is the integration of code into repository code then run deployment, quickly and more frequent. While Continuous Deployment or Continuous Delivery (CD) is a practice that carried out after CI process is complete and all code is successfully integrated, so that application can be build and release automatically.

By Using CI/CD, the application delivery process can continue automatically, quickly and often even though the application has been used by the user.

Keywords : application, learning management system, scrum, agile, continuous integration, continuous deployment.

E-ISSN : 2407 – 3911
P-ISSN : 2686 - 0333

I. PENDAHULUAN

Learning Management System atau LMS adalah salah satu produk aplikasi yang dikembangkan oleh PT. Millennia Solusi Informatika. Aplikasi ini sudah digunakan oleh salah satu jaringan sekolah swasta di beberapa kota. Namun, pengembangan dari aplikasi ini terus berjalan untuk peningkatan fitur-fitur dan mengadaptasi perubahan pada kebutuhan user.

Dikarenakan pengembangannya yang dilakukan terus menerus dan cepat, maka perlu suatu konsep yang dapat mendukung agar setiap perubahan dapat terimplementasi dengan cepat. Meskipun implementasi perubahannya cepat namun kualitas dari aplikasi harus juga dapat dijaga agar tidak mengganggu aplikasi yang sudah digunakan oleh pengguna.

Awalnya dalam *delivery* aplikasi LMS ini dilakukan secara manual, dimana *developer* melakukan *push* ke repository kode setelah itu dirilis ke dalam server. Namun hal ini dirasa sangat merepotkan terlebih ketika perubahan yang dirilis bersifat *minor* dan perubahan kode semakin sering.

Berangkat dari hal tersebut maka perlu diterapkan konsep *Continuous Integration/ Continuous Deployment* agar *developer* dapat fokus pada aplikasi yang dikembangkan dan setiap rilisan aplikasi dapat dirilis dengan cepat dan sudah melalui serangkaian proses yang teruji. Konsep CI/CD ini sudah banyak digunakan oleh banyak pengembang aplikasi terutama bagi tim pengembang yang menggunakan metode scrum atau *agile* dalam metodologi pengembangan perangkat lunak.

II. KAJIAN LITERATUR

Continuous Integration dan *Continuous Delivery/Continuous Deployment* merupakan bagian dari praktik dari DevOps. DevOps sendiri bisa diartikan sebagai sebuah praktik yang bisa membuat proses otomatis antara pengembangan aplikasi dan tim pengembangnya.

Continuous Integration (CI) adalah pengintegrasian kode ke dalam repositori kode kemudian menjalankan pengujian secara otomatis, cepat dan sering. Sementara *Continuous Delivery* atau *Continuous Deployment* (CD) adalah praktik yang dilakukan setelah proses CI selesai dan seluruh kode berhasil terintegrasi, sehingga aplikasi bisa dibangun lalu dirilis secara otomatis.

CI/CD *pipeline* ini sangat lazim digunakan dalam pengembangan perangkat lunak. CI/CD pipeline ini menjadi penghubung antara tim pengembang dengan tim operasional yang di dalamnya terpadat tiga fase yang berupa *continuous integration*, *continuous delivery* dan *continuous deployment*. Ketiga fase tersebut akan dilakukan secara terus menerus dan otomatis untuk mendapatkan perangkat lunak yang andal dan bebas dari *bug*.

Pada proses pengembangan perangkat lunak, penerapan praktik CI/CD telah meningkatkan efisiensi proyek. Dengan menerapkan praktik CI/CD, fitur baru aplikasi disebarkan kepada pengguna di setiap pengiriman *sprint* yang akan selalu melewati tahap uji coba sebelum dikirim ke pengguna. Otomatisasi telah memungkinkan untuk melakukan uji coba yang lebih efisien, sehingga mempercepat proses pengembangan serta menghasilkan aplikasi yang lebih handal (Arachi and Perera, 2018). Penerapan CI/CD juga menghasilkan dampak positif yaitu, mengurangi waktu pembuatan dan pengujian dalam *Continuous Integration*, meningkatkan visibilitas pada hasil pengujian, mendukung pengujian otomatis, mendeteksi kekurangan dan kesalahan, menangani masalah keamanan dan skalabilitas serta meningkatkan proses penyebaran sebuah aplikasi.

III. ANALISIS DAN PERANCANGAN

Dalam implementasi penelitian ini menggunakan metode CI/CD *Workflow Pipeline* sebagaimana pada gambar 3.1.

Gambar 1. DevOps Lifecyle

A. Code

Pada fase ini kode sumber dikembangkan oleh developer dalam membangun aplikasi Learning Management System.

B. Commit

Kode sumber yang telah dikembangkan oleh developer dilakukan *commit* untuk mengirimkan perubahan ke repositori kode. Di tahap ini proses kolaborasi sudah dimulai, dimana kode sumber yang telah dikirimkan oleh developer dapat digunakan oleh developer yang lain.

C. Build

Kode sumber dalam repositori akan diintegrasikan ke dalam sebuah *build*. Tahap ini kode sumber dikompilasi menjadi sebuah aplikasi yang utuh.

D. Tests

Setelah perubahan diintegrasikan ke dalam sebuah build, perubahan tersebut divalidasi dengan berbagai tingkatan pengujian otomatis (*automated testing*). Proses ini dilakukan untuk memastikan perubahan yang dilakukan tidak terdapat konflik atau *bug* pada baris kode yang telah dibuat.

E. Deploy

Versi terbaru yang telah diintegrasikan akan dirilis pada *environment* production.

Sebelum mengimplementasikan dari seluruh alur yang sudah dijelaskan sebelumnya. Perlu untuk mendefinisikan *tools* dan teknologi yang akan diimplementasikan. Pada tabel 3.1 di bawah adalah tools yang akan digunakan.

Tabel 1. Tools dan Teknologi CI/CD

No	*Tools*	*Software*
1.	*Versioning*	Git
2.	CI/CD Platform	Gitlab
3.	*Testing*	Enzyme

IV. IMPLEMENTASI & PENGUJIAN

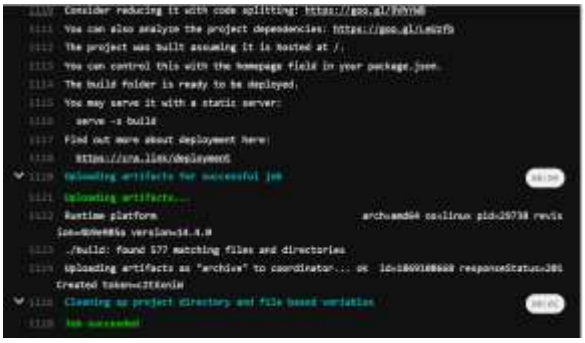
Aplikasi Learning Management System akan dibuat ke dalam bentuk Docker Container. Untuk membuat aplikasi menjadi sebuah Docker Container perlu didefinisikan Dockerfile seperti berikut.

```
FROM node:14.0-alpine
ENV NODE_ENV=production
WORKDIR /usr/src/app
COPY ["ckeditor5/","package.json", "package-lock.json*", "npm-shrinkwrap.json*", "./"]
RUN npm install --silent
RUN npm install serve --silent -g
COPY . .
RUN npm run build
EXPOSE 3000
CMD ["serve", "-s", "build"]
```

Gitlab perlu membaca .gitlab-ci.yml untuk proses otomatisasi proses CI/CD pada setiap tahapnya. Aplikasi akan melalui serangkaian tahapan yang ditulis dalam .gitlab-ci.yml.

```
stages:
  - build
  - test
  - deploy
build:
  stage: build
  script:
    - npm install
    - npm run build
  artifacts:
    paths:
      - ./build
  only:
    - test
test:
  stage: test
  script:
    - echo "Running react test suite....."
    - npm test
  only:
    - test
deploy:
  stage: deploy
  script:
    - cp ./build/ /var/www/lms --recursive
    - pm2 restart lms-web
  only:
    - test
```

Setiap perubahan yang dikirimkan ke repositori kode nantinya akan melalui rangkaian integrasi, test dan penyebaran kode ke *production environment* sesuai dengan skrip pada .gitlab-ci.yml. Pada gambar 2 menunjukkan proses CI/CD yang sedang dijalankan oleh Gitlab.



Gambar 2. Proses CI/CD berjalan di GitLab

Pada gambar 2 menunjukkan CI/CD *pipeline* mulai dari *build*, *test* hingga deploy aplikasi pada GitLab. Setiap tahapan akan berjalan berurutan yang jika satu tahapan tidak berhasil maka tahapan selanjutnya tidak akan dijalankan oleh GitLab. Dengan begitu DevOps akan mengecek pada bagian yang tidak berhasil dijalankan dan memperbaiki kesalahan yang terjadi. Sehingga hal ini dapat mencegah kode yang dikirimkan terdapat kesalahan pada saat dikirimkan ke pengguna.

Gambar 3. CI/CD Pipeline

Setelah CI/CD *pipeline* sukses dijalankan aplikasi akan dirilis otomatis di server seperti pada gambar 4.3 dimana aplikasi telah sukses dirilis setelah tahapan CI/CD.

Gambar 4. Aplikasi yang sudah dirilis

V. KESIMPULAN DAN SARAN

Dari hasil penelitian dapat disimpulkan dengan penerapan konsep CI/CD memudahkan tim pengembang dan operasional bekerja secara praktis, Dengan otomatisasi pada tahapan CI/CD ini memungkinkan kesalahan yang terjadi oleh manusia dapat terhindarkan dikarenakan seluruhnya telah otomatis dilakukan oleh mesin. Pada pengembangan aplikasi Learning Management System di PT. Millennia Solusi Informatika dimana perubahan pada kode aplikasi berjalan sering hal ini dapat membantu tim pengembang dan operasional dalam merilis aplikasi.

REFERENSI

Rony Setiawan (2021), Dicoding, Apa itu CI/CD? Developer Harus Tau, Retrieved from https://www.dicoding.com/blog/apa-itu-ci-cd/

P. Jha dan R. Khan. *A Review Paper on DevOps: Beginning and More To Know, International Journal of Computer Applications* (0975 – 0887), vol 180, no 48, pp. 16-20,2018

Arvind (Website). *DevOps Life Cycle :* Everything You Need To Know About DevOps Life cycle Phases, Retrieved From https://www.edureka.co/blog/devops-lifecycle/

Arachchi, S. S., & Perera, I. (2018). *Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management. Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management*, 1