

# **Makalah Rekayasa Perangkat Lunak Metode Continuous Integration (CI)**



Untuk memenuhi tugas mata kuliah Metodologi Desain Perangkat Lunak Praktik

## Metode Agile disusun oleh:

- |                         |              |
|-------------------------|--------------|
| 1. Novian Yuda Permana  | (5200411317) |
| 2. Khafid Nur Maulana   | (5200411491) |
| 3. Rizky Diar Panuntun  | (5200411499) |
| 4. Alfaeni Syafa Safira | (5200411511) |

## **Jurusan Informatika**

Fakultas Sains dan Teknologi

Universitas Teknologi Yogyakarta

## DAFTAR ISI

BAB I .....	1
PENDAHULUAN .....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	1
C. Tujuan Penelitian .....	1
BAB II.....	2
PEMBAHASAN .....	2
A. Apa itu Continuous Integration .....	2
B. Tujuan Continuous Integration.....	2
C. Manfaat Penggunaan metode Continuous Integration .....	2
D. Kelebihan dan Kekurangan dari metode Continuous Integration.....	3
a. Kelebihan Continuous Integration.....	3
b. Kekurangan Continuous Integration .....	3
BAB III .....	4
METODE CONTIONOUS INTEGRATION .....	4
E. Review Jurnal.....	4
BAB IV.....	11
PENUTUP .....	11
F. Perbandingan Metode Pengembangan Perangkat Lunak .....	11

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Pada masa ini perkembangan teknologi telah berkembang sangat pesat dan salah satunya menjadi pendukung sebuah perusahaan. Perusahaan memiliki sebuah teknologi perangkat lunak sebagai rencana strategis perusahaan yang menggunakannya, dengan adanya teknologi perangkat lunak bisa membantu perusahaan untuk memecahkan permasalahan.

Metode Agile merupakan salah satu jenis metode dalam pengembangan perangkat lunak. Biasanya sering disebut dengan SDLC (Software Development Life Cycle). Metode ini sering digunakan oleh startup maupun perusahaan besar dalam proses pengembangan software.

Didalam metode agile memiliki banyak jenis-jenis metode turunan yang memiliki perbedaan pada sistem saat pengembangannya seperti kelebihan dan kekurangannya, jenis-jenis metode turunan seperti Crystal Clear, SCRUM, Extreme Programming, Continuous Integration dll. Didalam laporan ini, akan di jelaskan mengenai salah satu metode agile yaitu Continuous Integration, yang didalamnya mengenai apa itu Continuous Integration, apa tujuan Continuous Integration, manfaat dari pengembangan metode Continuous Integration dan kekurangan dari metode Continuous Integration. Mari kita pelajari mengenai metode agile dengan jenis Continuous Integration.

### **B. Rumusan Masalah**

1. Apa itu Continuous Integration?
2. Apa tujuan Continuous Integration?
3. Manfaat dari pengembangan metode Continuous Integration?
4. Kelebihan dan kekurangan dari metode Continuous Integration?

### **C. Tujuan Penelitian**

1. Mengetahui apa itu Continuous Integration?
2. Mengetahui apa tujuan Continuous Integration?
3. Mengetahui manfaat dari pengembangan metode Continuous Integration?
4. Mengetahui kelebihan dan kekurangan dari metode Continuous Integration?

## **BAB II**

### **PEMBAHASAN**

#### **A. Apa itu Continuous Integration**

Continuous integration (CI) adalah pengintegrasian kode ke dalam repositori kode kemudian menjalankan pengujian secara otomatis, cepat, dan sering. Kamu dapat melakukan CI ini dengan menggunakan perintah commit.

Continuous Integration adalah salah satu kegiatan dalam pengembangan aplikasi software dimana seluruh hasil kerja dari developer digabungkan ke dalam satu wadah. Bisa sekali sehari, bisa juga beberapa kali dalam sehari. Kata kuncinya adalah bahwa seluruh hasil kerja digabungkan di satu tempat.

Jadi, CI adalah salah satu praktik DevOps yang digunakan untuk pengembangan perangkat lunak menjadi lebih terorganisir. Penggunaan CI pada pengembangan aplikasi memberikan banyak manfaat yang akan langsung terasa oleh tim pengembang. Selain itu, ada banyak pilihan tool atau alat yang dapat digunakan untuk mendukung berjalannya proses CI ini secara otomatis dan aman.

#### **B. Tujuan Continuous Integration**

- a. Dapat menyebarkan software sesuai dengan permintaan berdasarkan persyaratan bisnis.
- b. Mengurangi risiko software yang tidak berfungsi dengan baik dalam produksi.
- c. Dapat membuat rapid iteration berdasarkan feedback pelanggan yang nyata.
- d. Dapat pulih lebih cepat ketika terjadi kegagalan.
- e. Mempermudah dalam pembuatan program kelompok / bersama-sama

#### **C. Manfaat Penggunaan metode Continuous Integration**

- a. Mendapatkan Feedback Lebih Cepat Menggunakan CI Tools  
Untuk setiap code yang ditulis, setiap tes akan berjalan secara bersamaan untuk menjaga kestabilan pengembangan. Feedback yang lebih cepat membantu untuk memeriksa kualitas dan kegunaan dari setiap code.
- b. Visibilitas yang Lebih Baik  
Menggunakan CI/CD Pipeline, keseluruhan dari pengembangan, hasil tes, dan masalah dapat lebih mudah dianalisis. Sifat CI/CD Pipeline yang transparan ini memungkinkan developer untuk mengatur perubahan dan menghindari kerusakan pada software.
- c. Deteksi Bug Lebih Awal  
CI/CD memiliki fungsi tes otomatis, melalui tes otomatis ini akan lebih mudah mendeteksi bug pada tahap pengembangan awal, jadi kamu tidak perlu khawatir dikejutkan dengan error pada detik-detik terakhir.

## **D. Kelebihan dan Kekurangan dari metode Continuous Integration**

### **a. Kelebihan Continuous Integration**

- i. Mengurangi proses manual yang berulang.
- ii. Mengurangi resiko karena mendeteksi & memperbaiki masalah integrasi yang terus menerus.
- iii. Membuat proyek lebih baik dan jelas.
- iv. Menghasilkan perangkat lunak yang dapat di-deploy kapan saja dan dimana saja.
- v. Menghemat waktu dan biaya selama proyek berlangsung.

### **b. Kekurangan Continuous Integration**

- i. Memerlukan pengaturan awal terlebih dahulu tahap demi tahap.
- ii. Memerlukan test kode untuk mencapai pengujian secara otomatis.
- iii. Refactoring (melakukan perubahan pada kode program dari perangkat lunak dengan tujuan meningkatkan kualitas dari struktur program tersebut tanpa mengubah cara program tersebut bekerja) dalam skala besar dapat mengganggu karena dapat merubah basis kode.

### BAB III

#### METODE CONTIONOUS INTEGRATION

##### E. Review Jurnal

<b>Judul</b>	IMPLEMENTASI CONTIONOUS INTEGRATION DAN CONTINOUS DEPLOYMENT PADA APLIKASI LEARNING MANAGEMENT SYSTEM DI PT. MILLENNIA SOLUSI INFORMATIKA
<b>Jurnal</b>	Jurnal Ilmiah Teknologi Informasi Terapan
<b>Volume dan Halaman</b>	Volume 8 Halaman 183-186
<b>Tahun</b>	2021
<b>Penulis</b>	1. Indra Guna Noviantama 2. Ari Purno Wahyu W
<b>Abstrak</b>	<p>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</p>

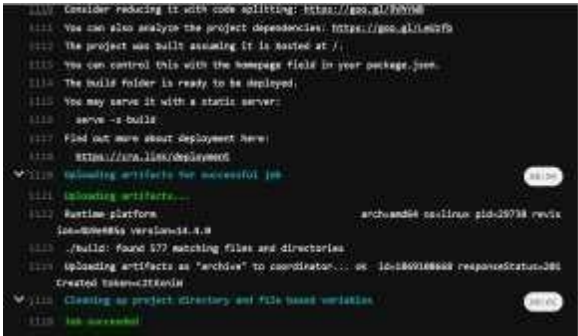
	<p>berulang.</p> <p>Continuous Integration (CI) adalah pengintegrasian kode ke dalam repositori kode kemudian menjalankan penggunaan secara otomatis, cepat dan sering.</p> <p>Sementara Continuous Deployment atau Continuous Delivery (CD) adalah praktik yang dilakukan setelah proses CI selesai dan seluruh kode berhasil terintegrasi, sehingga aplikasi bisa dibangun dan dirilis secara otomatis.</p> <p>Dengan menggunakan metode CI/CD diharapkan dalam proses penyampaian aplikasi dapat terus berlangsung otomatis, cepat dan sering walaupun aplikasi tersebut sudah digunakan oleh pengguna.</p>
<b>Pengantar</b>	<p>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.</p> <p>Dikarenakan pengembangannya yang dilakukan terus menerus dan cepat, maka perlu suatu konsep yang dapat mendukung agar setiap perubahan dapat terimplementasi dengan cepat. Meskipun</p>



	<p>implementasi perubahannya cepat namun kualitas dari aplikasi harus juga dapat dijaga agar tidak mengganggu aplikasi yang sudah digunakan oleh pengguna.</p> <p>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.</p> <p>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.</p>
<b>Analisis dan Perancangan</b>	<p>Dalam implementasi penelitian ini menggunakan metode CI/CD Workflow Pipeline.</p> <p>A. Code</p> <p>Pada fase ini kode sumber dikembangkan oleh</p>



	<p>developer dalam membangun aplikasi Learning Management System.</p> <p><b>B. Commit</b></p> <p>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.</p> <p><b>C. Build</b></p> <p>Kode sumber dalam repositori akan diintegrasikan ke dalam sebuah build. Tahap ini kode sumber dikompilasi menjadi sebuah aplikasi yang utuh.</p> <p><b>D. Tests</b></p> <p>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.</p> <p><b>E. Deploy</b></p> <p>Versi terbaru yang telah diintegrasikan akan dirilis pada environment production.</p>
--	--

	Sebelum mengimplementasikan dari seluruh alur yang sudah dijelaskan sebelumnya. Perlu untuk mendefinisikan tools dan teknologi yang akan diimplementasikan.		
	NO	Tools	Software
	1	Versioning	Git
	2	CI/CD Platform	Gitlab
	3	Testing	Enzyme
<b>Implementasi &amp; Pengujian</b>	<p>Aplikasi Learning Management System akan dibuat ke dalam bentuk Docker Container. Untuk membuat aplikasi menjadi sebuah Docker Container perlu didefinisikan Dockerfile seperti berikut.</p> <pre> 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 . . </pre> <p>Gitlab perlu membaca .gitlab-ci.yml untuk proses otomatisasi proses CI/CD pada setiap tahapannya. Aplikasi akan melalui serangkaian tahapan yang ditulis dalam .gitlab-ci.yml.</p>		

	<pre> stages:   - build   - test   - deplo ybuild:   stage: build   script:     - npm install     - npm run build artifacts:   paths:     - ./build only:   - tes ttest:   stage:   test   script:     - echo "Running react test suite....."     - npm testonly:   - tes t </pre>	<p>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.</p>  <p><b>Gambar 2. Proses CI/CD berjalan di GitLab</b></p> <p>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</p>
--	--	--

	<p>dijalankan dan memperbaiki kesalahan yang terjadi. Sehingga hal ini dapat mencegah kode yang dikirimkan terdapat kesalahan pada saat dikirimkan ke pengguna.</p>  <p><b>Gambar 3. CI/CD Pipeline</b></p> <p>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.</p>  <p><b>Gambar 4. Aplikasi yang sudah dirilis</b></p>
<p><b>Kesimpulan dan Saran</b></p>	<p>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</p>

## BAB IV

### PENUTUP

#### Perbandingan Metode Pengembangan Perangkat Lunak

##### A. Metode Agile

Perencanaan Sistem (Systems Planning)	Analisis Sistem (Systems Analysis)	Perancangan Sistem (Systems Design)	Pemeliharaan Sistem (Systems Maintenance)
Berawal dari kebutuhan	<p>Kebutuhan data dapat ditambah ataupun dikurangi sesuai dengan kebutuhan user, ketika dilakukan testing</p> <p>Perubahan dapat dilakukan tanpa mengganggu sistem yang sudah digunakan oleh pengguna.</p>	<p>Testing dilakukan ketika semua tahapan pada model sudah selesai</p> <p>Memberikan prototype sebagai gambaran sistem telah Dibangun</p> <p>Mengedepankan aspek kenyamanan user</p>	Dilakukan sesuai kesepakatan

##### B. Metode Waterfall, Prototype, dan RAD

Tahapan Pengembangan Perangkat Lunak	Waterfall	Prototype	RAD

Perencanaan Sistem (Systems Planning)	Berawal dari kebutuhan	Berawal dari kebutuhan	Berawal dari kebutuhan
Analisis Sistem (Systems Analysis)	<p>Kebutuhan data harus dianalisis diawal secara lengkap dan menyeluruh</p> <p>Perubahan data ataupun fungsional akan merubah</p>	<p>Kebutuhan data dapat ditambah ataupun dikurangi sesuai dengan kebutuhan user, ketika dilakukan testing</p> <p>Perubahan dapat dilakukan selama sistem atau perangkat</p>	<p>Kebutuhan data dapat ditambah ataupun dikurangi sesuai dengan kebutuhan user, ketika dilakukan testing .</p> <p>Kebutuhan fungsi mayor dapat dimodalkan dalam</p>

	keseluruh proses pada tahapan berikutnya.	lunak masih dalam bentuk prototype.	waktu tertentu dan dapat dibicarakan oleh tim RAD yang terpisah
Perancangan Sistem (Systems Design)	<p>Testing dilakukan ketika semua tahapan pada model sudah selesai</p> <p>Tidak dapat memberikan gambaran yang jelas telah dibangun</p> <p>Mengedepankan</p>	<p>Testing dapat dilakukan ketika prototype telah dibangun, sehingga hasil testing dapat merubah rancangan sistem.</p> <p>Memberikan prototype sebagai gambaran sistem yang akan telah dibangun.</p> <p>Mengedepankan aspek</p>	<p>Testing dapat dilakukan ketika prototype telah dibangun, sehingga hasil testing dapat merubah rancangan system</p> <p>Memberikan prototype sebagai gambaran sistem telah dibangun.</p> <p>Mengedepankan aspek</p>

	kebutuhan fungsional sistem.	kenyamanan user	kenyamanan user dan kecepatan pembangunan
Pemeliharaan Sistem (Systems Maintenance)	Dilakukan sesuai kesepakatan	Dilakukan sesuai kesepakatan	Dilakukan sesuai kesepakatan