

Mengakselerasi Keterampilan Rekayasa Perangkat Lunak: Peranan DevOps, SDLC, dan CI/CD dalam Meningkatkan Kompetensi Siswa SMK N 1 Pusomaen

Semmy Wellem Taju*¹, Stenly Richard Pungus², Rolly Junius Lontaan³,
Reymon Rotikan⁴, Marchel Thimoty Tombeng⁵

^{1,2,3,4,5} Universitas Klabat; Jl. A. Mononutu, Airmadidi, Minahasa Utara

e-mail: *¹semmy@unklab.ac.id, stenly.pungus@unklab.ac.id, rolly.lontaan@unklab.ac.id,
reymonr@unklab.ac.id, marcheltombeng@unklab.ac.id

Abstrak

Rekayasa Perangkat Lunak (software engineering) merupakan proses komprehensif yang melibatkan pengembangan program aplikasi. Pemahaman mendalam mengenai DevOps (Development Operation), Siklus Hidup Pengembangan Perangkat Lunak (Software Development Life Cycle atau SDLC), serta Integrasi Berkelanjutan/Pengiriman Berkelanjutan (Continuous Integration/Continuous Deployment atau CI/CD) adalah esensial dalam pengembangan aplikasi yang modern. DevOps diarahkan untuk meningkatkan kemudahan akses terhadap software. Model SDLC yang tradisional seringkali dipilih karena kesesuaiannya dalam mengembangkan aplikasi yang mudah dievaluasi. Namun, keterbatasan model ini mengakibatkan pola kerja yang monoton bagi pengembang, sehingga menghambat perubahan dan adaptasi yang cepat. Sebagai solusi, model CI/CD telah dikembangkan menjadi pendekatan inovatif yang mengubah proses pengembangan perangkat lunak dari manual menjadi otomatis. CI/CD tidak hanya memudahkan dan mempercepat proses pengembangan tetapi juga meningkatkan kualitas dari software yang dirilis. Survei yang dilakukan melalui pre-test di SMK N 1 Posumaen menunjukkan adanya kekurangan dalam pengetahuan siswa mengenai prinsip rekayasa perangkat lunak yang canggih ini. Setelah penyampaian materi yang dirancang khusus, terdapat peningkatan yang signifikan dalam pemahaman siswa, sebagaimana dibuktikan oleh hasil post-test.

Kata kunci— Perangkat Lunak, DevOps, SDLC, CI/CD

Abstract

Software engineering is a comprehensive process that involves the development of application programs. A deep understanding of DevOps, the Software Development Life Cycle (SDLC), and Continuous Integration/Continuous Deployment (CI/CD) is essential in modern application development. DevOps is geared towards improving the ease of access to software. The traditional SDLC model is often chosen for its suitability in developing applications that are easy to evaluate. However, the limitations of this model result in a monotonous work pattern for developers, hindering rapid change and adaptation. As a solution, the CI/CD model has been developed into an innovative approach that transforms the software development process from manual to automated. CI/CD not only facilitates and accelerates the development process but also improves the quality of the released software. A survey has been conducted through a pre-test at SMK N 1 Posumaen showed a deficiency in students' knowledge of these advanced software engineering principles. After the delivery of the presentation, there was a significant improvement in students' understanding, as evidenced by the post-test results.

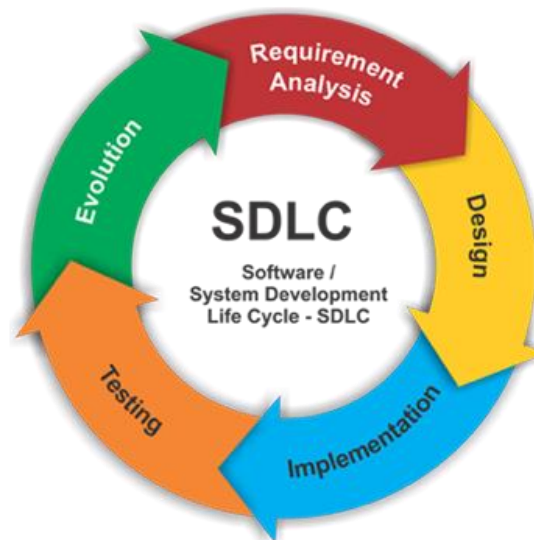
Keywords— Software Engineering, DevOps, SDLC, CI/CD

1 PENDAHULUAN

Rekayasa perangkat lunak adalah pengembangan program yang dapat digunakan untuk mengelola data; berisi instruksi yang masing-masing merupakan fitur atau fungsi yang memungkinkan data diproses dan kemudian disimpan dalam format mirip maya [1]. Rekayasa perangkat lunak adalah disiplin akademis yang mencakup setiap aspek produksi lunak. Apapun produknya, itu mencakup komunikasi, analisis kebutuhan pengguna, desain, pengkodean, pengujian, dan pemeliharaan sistem [2]. Jadi rekayasa perangkat lunak adalah suatu proses yang mengembangkan suatu program aplikasi yang dapat mengolah data sehingga memberikan informasi yang dibutuhkan. Dalam pengembangan suatu perangkat lunak ada beberapa istilah yang merupakan bagian dari suatu pengembangan perangkat lunak antara lain yaitu: *DevOps*, *CI/CD*, dan *SDLC*.

DevOps (Development dan Operation) mengacu pada divisi, proses, dan pengetahuan teknologi selama fase pengembangan aplikasi. Dalam proyek ini, tim pengembangan berkomitmen penuh untuk terus menciptakan aplikasi baru dan menyempurnakan aplikasi yang sudah ada agar lebih efektif dan berkualitas tinggi [3]. Kombinasi pengembangan dan operasi disebut *DevOps*. Kombinasi yang disebutkan di atas dirancang untuk meningkatkan kemampuan perusahaan dalam menyampaikan aplikasi dengan tingkat kedatangan yang tinggi [4]. *DevOps* merupakan salah satu pendekatan yang mengintegrasikan pengembangan, pengujian, integrasi, dan pemantauan secara berkelanjutan atau real-time. Seorang pengembang software tentu ingin agar software yang dibuat bisa lebih mudah diakses oleh pelanggan bukan?

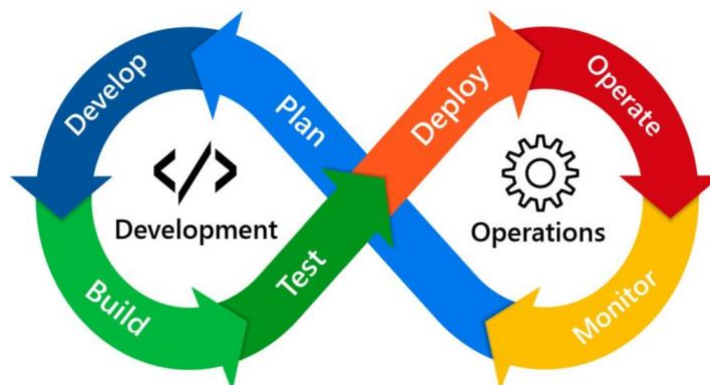
The Software Development Life Cycle (SDLC) adalah proses kerja atau kerangka manajemen proyek yang menguraikan langkah-langkah yang diperlukan untuk mengembangkan sistem TI dari awal hingga selesai [5]. SDLC ini cocok untuk mengembangkan aplikasi yang mudah dievaluasi; Namun hal ini akan mengakibatkan pengembang bekerja secara monoton dan tidak dapat melakukan perubahan dengan cepat [6].



Gambar 1 Software Development Life Cycle (SDLC)

Singkatan dari Continuous Integration/Continuous Deployment, juga dikenal sebagai Continuous Delivery, adalah *CI/CD*. Dalam konteks *DevOps*, *CI/CD* adalah metodologi yang bertujuan untuk mengotomatisasi dan mempercepat pengembangan, pengujian, dan pengiriman produk baru [7]. *CI/CD* adalah dua konsep praktik yang umum digunakan dalam pengembangan perangkat lunak di era modern. Terbukti, saat mengembangkan aplikasi online, pengembang

biasanya mengalami banyak kesulitan dalam memulai pengujian, anotasi pengkodean, dan penerapan manual. Sebaliknya, jika terdapat bug, kemungkinan besar waktu rilis aplikasi tersebut sudah ditentukan sejak peluncurannya [8]. CI/CD adalah upaya mengubah proses pengembangan software yang tadinya manual menjadi otomatis. CI/CD adalah metode pengembangan software yang memudahkan, mempercepat, sekaligus meningkatkan kualitas hasil akhir software yang dirilis.



Gambar 2 Continuous Integration/Continuous Deployment (CI/CD)

DevOps adalah kombinasi antara Development (*pengembangan*) dan Operation (*operasi*), ini merupakan cara berpikir baru dalam domain rekayasa perangkat lunak [9]. Dalam budaya DevOps, praktik ini bertujuan untuk mengurangi hambatan antara tim pengembangan dan operasi [10][11][12]. Dalam penelitian yang dilakukan oleh Welder Pinheiro Luz *dkk* [10][13][14], penelitian tersebut menggunakan kelompok fokus untuk mengumpulkan persepsi perusahaan tentang adopsi DevOps. Dengan menggunakan Grounded Theory klasik untuk membangun teori tentang 15 skenario adopsi DevOps yang sukses di perusahaan, mereka mengusulkan sebuah model alur kerja untuk adopsi DevOps. Mereka mendapati bahwa *collaboration* (kolaborasi) adalah inti dari perhatian DevOps. Mali Senapathi *dkk* [15] juga melakukan penelitian dengan wawancara pada enam insinyur perangkat lunak yang sudah berpengalaman dan memantau penerapan prinsip dan praktik DevOps secara bertahap. Mereka mendapati bahwa terdapat peningkatan komunikasi dan kolaborasi yang lebih baik antara personel pengembangan dan operasi TI. Mereka juga menemukan bahwa menerapkan *pipeline* otomatisasi dan struktur organisasi lintas fungsional dapat memberikan manfaat yang diharapkan dari DevOps.

2 METODE PELAKSANAAN

Pelaksanaan kegiatan Pengabdian Kepada Masyarakat (PKM) yang dilakukan di Universitas Klabat tepatnya di ruangan Lab Komputer GA 203 pada tanggal 27 September 2023 dengan tema yaitu “**Pengenalan dasar Rekayasa perangkat lunak, Model Proses, CI/CD dan DevOps**” yang diikuti oleh 14 siswa/i SMK N 1 Pusomaen. Kegiatan ini merupakan bagian dari kegiatan kunjungan siswa/i SMK N 1 Posumaen ke Universitas Klabat. Pada kegiatan tersebut siswa/i SMK N 1 Posumaen yang ada di berikan beberapa topik dan dibagikan dalam beberapa ruangan salah satunya adalah ruangan GA. 203 yang menjadi salah ruangan untuk pelaksanaan PKM pada kunjungan siswa/i SMK N 1 Posumaen. Pada ruangan ini materi yang diberikan Pengenalan dasar Rekayasa perangkat lunak, Model Proses, CI/CD dan DevOps.

Pada kegiatan tersebut pemateri dari materi yang ada dibawakan oleh bapak Semmy Wellem Taju, PhD yang merupakan salah dosen pada Fakultas Ilmu Komputer Universitas Klabat dan didampingi oleh bapak Stenly Richard Pungus, PhD., bapak Rolly Junius Lontaan, M.Kom., bapak Reymond Rotikan, MS., MM., dan bapak Marchel Timothy Tombeng, MS., dan juga

beberapa mahasiswa yang tergabung pada organisasi Senat dan BEM Fakultas Ilmu Komputer Universitas Klatat.

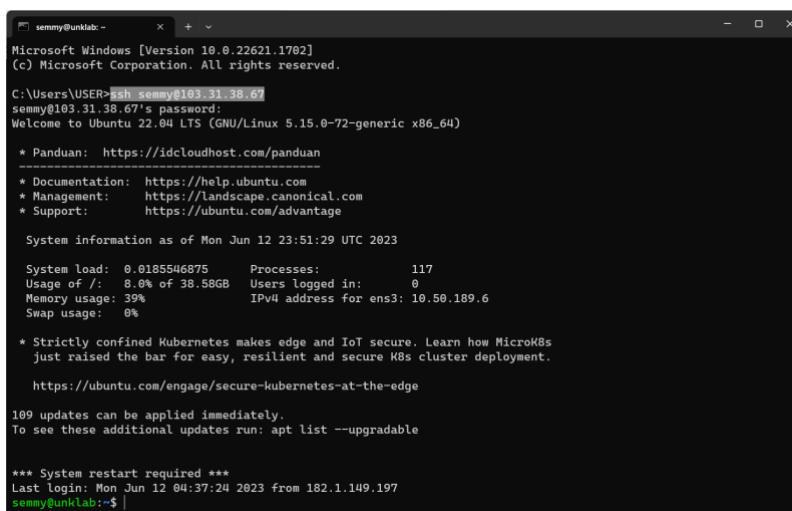
Pada awal kegiatan siswa/i diberikan pre-test untuk materi yang akan dibawakan. Hasil dari pre-test dengan 10 butir pertanyaan dan diakhir kegiatan juga diberikan dengan pertanyaan yang sama dimana pertanyaan-pertanyaan diberikan hanya meminta informasi yang mereka ketahui atau belum ketahui tentang rekayasa perangkat lunak.

3 HASIL DAN PEMBAHASAN

3.1 Evaluasi Dasar Pre-test dan Post-test

Berdasarkan hasil pre-test diketahui bahwa rata-rata pengetahuan siswa/i SMK N 1 Posumaen tentang rekayasa perangkat lunak beserta beberapa istilah maupun bagian dari rekayasa perangkat lunak adalah 8,47%. Dari hasil tersebut maka materi yang diberikan di jelaskan sesederhana mungkin sehingga siswa/i SMK N 1 Posumaen diharapkan mampu mengikuti penjelasan yang diberikan dan dapat mengerti apa yang dijelaskan oleh pemateri.

Dari hasil post-test yang dilakukan setelah selesai mengikuti pemaparan materi tentang pengenalan dasar rekayasa perangkat lunak, model proses, CI/CD dan DevOps didapati bahwa rata-rata pengetahuan siswa/i SMK N 1 Posumaen tentang materi tersebut adalah 74.74%. Dari hasil post-test yang dicapai didapati beberapa pertanyaan yang memiliki jawaban ya 100% adalah Apakah kelas ini berhasil meningkatkan pemahaman Anda tentang rekayasa perangkat lunak?, Apakah Anda merasa lebih akrab dengan konsep Continuous Integration (CI) setelah mengikuti kelas ini?, Apakah kelas ini membantu Anda memahami lebih baik tentang Continuous Deployment (CD) dalam pengembangan perangkat lunak?, Apakah Anda merasa lebih siap untuk berkontribusi dalam tim pengembangan perangkat lunak setelah mengikuti kelas ini?, Apakah Anda merasa kelas ini membantu Anda mencapai tujuan yang Anda tetapkan sebelumnya?.



```
semmy@unklab: ~
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>ssh semmy@103.31.38.67
semmy@103.31.38.67's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-72-generic x86_64)

 * Panduan: https://idcloudhost.com/panduan
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Mon Jun 12 23:51:29 UTC 2023

System load: 0.0185546875      Processes:            117
Usage of /: 8.0% of 38.58GB    Users logged in:     0
Memory usage: 39%             IPv4 address for ens3: 10.50.189.6
Swap usage: 0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

109 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Mon Jun 12 04:37:24 2023 from 182.1.149.197
semmy@unklab:~$
```

Gambar 3 Koneksi Cloud Server Ubuntu 22.04 LTS menggunakan Command Prompt

3.2 Membangun VPS di IDCloudHost

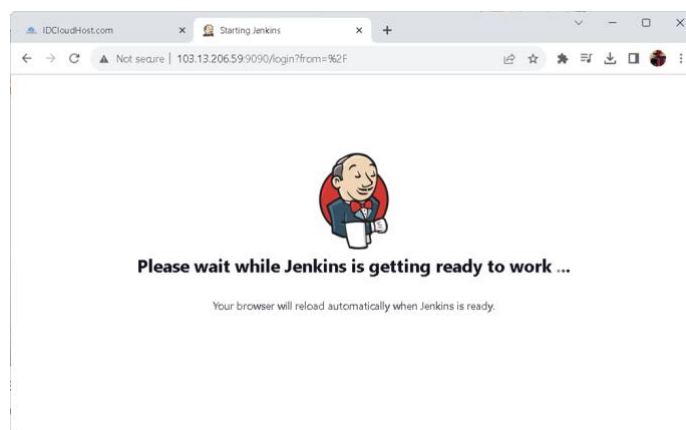
Pada bagian pertama di dalam presentasi ini, kami membahas langkah-langkah untuk membuat Virtual Private Server (VPS) melalui platform IDCloudHost, termasuk proses sewa, konfigurasi dasar, dan aktivasi lain seperti penggunaan Putty dan WinSCP tools untuk mengakses *cloud server*. Untuk membangun VPS sendiri, kami juga memberikan penjelasan tentang berbagai aspek penting di dalam layanan IDCloudHost yaitu pemilihan spesifikasi *cloud server*, sistem

operasi yang akan digunakan, dan pilihan tambahan yang dapat disesuaikan sesuai kebutuhan. Dalam mengimplementasikan CI/CD di DevOps, kami juga menjelaskan langkah-langkah dalam membangun VPS menggunakan layanan IDCloudHost kepada semua siswa/i SMK N 1 Posumaen. VPS pada IDCloudHost adalah salah satu penyedia layanan cloud hosting di Indonesia. IDCloudHost menyediakan solusi VPS dengan berbagai pilihan konfigurasi, dimana pada kesempatan tersebut kami menggunakan Apache/2.4.52 (Ubuntu) Server 22.04 dengan 2 CPU, 2 GB RAM dan 40 GB Disk standard. Dengan membahas proses ini, diharapkan siswa/i mendapatkan pemahaman yang komprehensif tentang bagaimana membangun dan menggunakan VPS dengan efisien di IDCloudHost.

Komunikasi ke cloud server juga dijelaskan agar supaya siswa/I paham bagaimana cara terhubung ke VPS. Adapun beberapa tools dapat digunakan untuk bisa terhubung ke *cloud server* diantaranya yaitu PuTTY dan WinSCP. Gambar 3.1 menunjukkan tampilan awal sesudah berhasil *signed in* menggunakan *command prompt* di *cloud server*. PuTTY adalah sebuah terminal untuk SSH dan *telnet*, sedangkan WinSCP adalah sebuah klien SCP (Secure Copy Protocol) untuk mentransfer file ataupun folder dengan aman. Untuk bisa terhubung ke cloud server Ubuntu 22.04 menggunakan Putty, pertama-tama kita dapat membuka Putty dan masukkan public IP address, username dan password dari *cloud server*. Selanjutnya, pilih protokol koneksi SSH dengan port default 22. Setelah kita terhubung dengan *cloud server*, kita dapat menggunakan terminal Putty untuk berinteraksi dengan *cloud server*. Untuk melakukan transfer file ataupun folder secara aman dan terdapat Graphical User Interface (GUI), kita bisa menggunakan WinSCP dengan memasukkan public IP address, *username*, dan *password* dari *cloud server*. Selanjutnya, kita dapat memilih protokol SCP atau SFTP untuk komunikasi antara client dan server Ubuntu 22.04, lalu kita dapat mengklik button Login. Dengan menggunakan WinSCP, kita dapat dimudahkan untuk mengelola *file* ataupun *folder* antara komputer lokal dan *cloud server* Ubuntu.

3.3 Pengenalan Automation Server di DevOps

Pada bagian pengenalan tentang *automation server*, kami menjelaskan untuk memberikan pemahaman mendalam tentang pengembangan perangkat lunak (*software*), dimana *coding* hanyalah tahap awal. Setelah proses *coding*, umumnya seorang developer software biasanya melakukan banyak hal lain sebelum aplikasi sampai ke tahap *production* di sisi *server*. Sebelum adanya *automation server*, sebagian besar pekerjaan dilakukan secara manual; namun, ini menimbulkan risiko kesalahan yang sering dan tidak konsisten. Beberapa hal yang menjadi pertimbangan dalam pengembangan software/sistem secara manual adalah diantaranya menjalankan *testing* sebuah layanan secara manual, *download dependency* aplikasi secara manual, setup database secara manual, membuat distribusi aplikasi secara manual, dan lain-lain. Setiap proses ini pastinya sangat kompleks jika harus dilakukan secara manual oleh seorang *developer* terus-menerus tanpa adanya *automation server*.

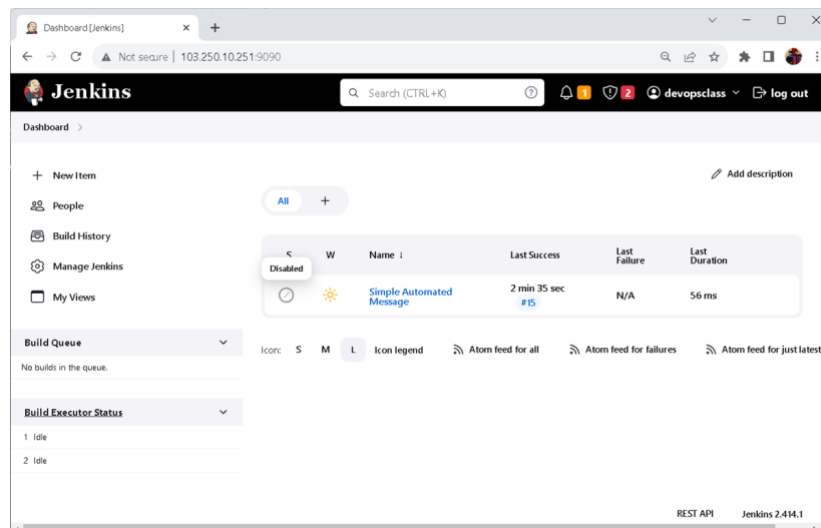


Gambar 4 Proses Inisialisasi Jenkins sebagai salah satu Automation Server Tool

Untuk tujuan demo tentang CI/CD DevOps, kami menggunakan tools atau aplikasi *automation server* di Ubuntu 22.04 untuk melakukan proses otomatisasi perintah yang diinstruksikan. Terdapat beberapa *automation server tools* yang bisa kita gunakan yaitu *Jenkins*, *Team City*, *Gitlab CI/CD*, *Github Action*, *atlassian Bamboo*, *Cricle CI*, *Travis CI* dan lain sebagainya. Gambar 3.2 menampilkan proses inialisasi Jenkins sebagai salah satu *automation server* aplikasi yang running di server ubuntu 22.04. Seperti yang sudah dijelaskan bahwa Jenkins adalah tools CI/CD yang mendukung otomatisasi pengembangan software dibagian server side, mulai dari tahapan code merging hingga sampai pada proses production atau deployment. Jenkins menyediakan additional plugin, serta mendukung berbagai jenis OS server, seperti Windows, Linux, dan MacOS. *Automation server* menggunakan Jenkins memungkinkan kita untuk mempercepat proses pada tahapan pengembangan software. *Automation server* menggunakan Jenkins juga bisa berjalan secara konsisten mengikuti pola instruksi yang sudah dibuat pada inverenment yang perangkat lunak yang berbeda-beda. *Automation server* menggunakan Jenkins dapat berjalan secara otomatis (*hampir sama seperti scheduler di sistem operasi windows*), diharapkan agar tool *automation server* ini dapat memberikan kita informasi kesalahan, ataupun lain-lain secara otomatis.

3.4 Open Source Automation Server Menggunakan Jenkins

Sedikit informasi yang bisa kami tambahkan disini yaitu Jenkins adalah salah satu *open source automation server* yang paling populer seiring dengan perkembangan DevOps pada tahun 2023. Sebelumnya Jenkins project bernama Hudson. Kita dapat mendownload Jenkins pada official website berikut <https://www.jenkins.io/>. Adapun *software* dan *hardware requirements* dalam implementasi dari *automation server* menggunakan Jenkins yaitu (1) Hardware requirements untuk instalasi Jenkins yang rekomendasikan seperti harus memiliki 4 GB+ RAM dan paling kurang terdapat 50 GB+ drive space atau storage. Selanjutnya, (2) Software requirements untuk instalasi Jenkins yang rekomendasikan seperti harus sudah terinstall Secure Shell Protocol (SSH) di Ubuntu 22.04 cloud server, Java versi 11 (<http://jdk.java.net/archive/>) dan Git/Github (<https://git-scm.com/>).



Gambar 5 Implementasi Job di Jenkins untuk Simple Automated Message

Dalam kegiatan PKM tersebut, kami menjelaskan *step by step* cara penggunaan Job di Jenkins untuk membuat *Simple Automated Message* untuk *cloud server* Ubuntu 22.04, dimana akan ada kalimat yang akan execute sebagai output pada spesifik waktu tertentu.

- 1) Dalam Step 1 sebagai step awal kita akan create sebuah Jenkins Job. Kita membuat Job Jenkins yaitu dengan cara membuka dashboard Jenkins, lalu klik "*New Item*" untuk membuat job baru. Selanjutnya, beri nama job (misalnya, "*Simple Automated Message*"). Lalu, pilih "*Freestyle project*" sebagai tipe proyek yang akan dibuat.
- 2) Pada Step 2 ini, kita akan melakukan konfigurasi Job pada Jenkins. Di halaman konfigurasi proyek Jenkins di bawah bagian "Build", tambahkan sebuah "Build step" dengan memilih "Execute shell" (pilihan ini akan menjalankan perintah shell pada sistem operasi ubuntu). Selanjutnya, kita akan melihat sebuah kotak muncul dan di dalam kotak teks yang tersebut, kita dapat mengetikkan perintah yang akan dicetak ke konsol. Misalnya, tulislah perintah seperti berikut **echo "Halo, ini adalah pesan sederhana DevOps !!!"**.
- 3) Step 3 ini masih pada tahap konfigurasi Job di Jenkins. Kita mulai dari halaman konfigurasi job, cari dan temukan bagian "*Build Triggers*". Selanjutnya Aktifkan opsi "Build periodically". Di dalam kotak teks yang tersedia, masukkan ekspresi cron dari Ubuntu 22.04 untuk menjadwalkan job setiap 2 menit. Cara untuk menuliskan ekspresi cron ubuntu untuk setiap 2 menit yaitu `*/2 * * * *`.
- 4) Step 4 adalah step yang terakhir yaitu kita akan menjalankan/run sebuah Job di Jenkins. Pastakan kita sudah kembali ke dashboard Jenkins dan temukan Job dengan nama "*Simple Automated Message*" (*sesuaikan dengan nama yang diberikan pada step awal tadi*). Selanjutnya, kita klik pada Job tersebut dan kemudian klik "Build Now" untuk menjalankan Job di Jenkins.

Pada bagian pengenalan *automation server* di DevOps ini juga, kami menginstruksikan kepada siswa/i SMK N 1 Posumaen untuk dapat memperhatikan/memantau Job pada Jenkins yang sudah berhasil dijalankan, siswa/i SMK N 1 Posumaen mengamati output/hasil dari pesan otomatis. Output tersebut dapat ditampilkan secara otomatis dengan mengklik nama Job yang telah dibuat, lalu kita click pada salah satu "*Build History*" dan melihat "*Console Output*" untuk melihat pesan yang print ke console. Perhatikan juga pada bagian Build History, dimana pada bagian history ini akan bertambah setiap 2 menit, karena perintah yang dieksekusi setiap 2 menit sekali oleh Job Jenkins tersebut.



Gambar 6 Dokumentasi Gambar Sebelum dan Sesudah Pelaksanaan Kegiatan PKM

3.5 Dokumentasi Pelaksanaan Kegiatan PKM

Dapat dilihat pada Gambar 3.4, berisi beberapa gambar umum tentang pelaksanaan kegiatan PKM dari kunjungan siswa/i SMK N 1 Posumaen ke Universitas Klabat. Sebelum pelaksanaan kegiatan PKM, salah satu gambar dibawah memperlihatkan persiapan dimulainya kegiatan PKM. Siswa/siswi SMK N 1 Posumaen dan dosen terlihat sedang mengorganisir materi presentasi atau menyiapkan tempat/lab untuk kegiatan tersebut. Saat pelaksanaan kegiatan PKM Berlangsung, Siswa/siswi SMK N 1 Posumaen terlihat sedang aktif mengikuti presentasi dari Bapak Semmy Wellem Taju, PhD. Dapat dilihat juga tangkapan layar dari presentasi di mana siswa/i terlibat dalam diskusi. Setelah kegiatan PKM selesai, salah satu gambar dibawah memperlihatkan menunjukkan suasana setelah berakhirnya kegiatan PKM. Siswa/siswi dan semua dosen yang terlibat, terlihat berinteraksi satu sama lain, dalam bentuk sesi foto bersama.

4 KESIMPULAN dan SARAN

Kegiatan Pengabdian Kepada Masyarakat (PKM) di SMK N 1 Posumaen telah memberikan bukti yang kuat mengenai efektivitas intervensi pendidikan dalam meningkatkan pengetahuan dan minat siswa/i SMK N 1 Posumaen terhadap rekayasa perangkat lunak. Evaluasi awal yang menunjukkan pemahaman yang rendah dan minat yang kurang dari siswa terhadap rekayasa perangkat lunak menjadi titik tolak yang penting. Melalui serangkaian diskusi komprehensif dan sesi edukatif yang mendalam, termasuk pengenalan terhadap konsep-konsep kunci seperti DevOps, SDLC, dan CI/CD, program ini berhasil tidak hanya meningkatkan pemahaman siswa tentang rekayasa perangkat lunak secara signifikan tetapi juga memicu ketertarikan mereka dalam bidang ini.

Lebih jauh, kegiatan ini mengungkapkan bahwa dengan pengetahuan yang tepat dan pendekatan yang terstruktur, siswa/i SMK N 1 Posumaen dapat mengembangkan visi yang lebih luas mengenai potensi mereka dalam industri teknologi informasi, khususnya rekayasa perangkat

lunak. Hasil pasca-program menunjukkan peningkatan yang substansial dalam pemahaman siswa tentang materi yang diajarkan, yang menandakan suksesnya PKM dalam menginspirasi siswa/I SMK N 1 Posumaen untuk lebih mendalami dan mungkin mengejar karir di bidang rekayasa perangkat lunak. Temuan ini menegaskan pentingnya inisiatif pendidikan yang dirancang untuk mengatasi kesenjangan pengetahuan dan memperkaya kompetensi siswa/i SMK N 1 Posumaen, yang pada akhirnya dapat berkontribusi pada pembangunan sumber daya manusia yang berkualitas di sektor rekayasa perangkat lunak.

5 UCAPAN TERIMA KASIH

Penulis ingin menyampaikan rasa terima kasih kepada SMK Negeri 1 Pusumaen atas inisiatif dan kerjasama yang erat dalam pelaksanaan Program Pengabdian kepada Masyarakat (PKM). Kepercayaan yang diberikan kepada Universitas Klabat untuk memberikan pelatihan dasar rekayasa perangkat lunak kepada siswa/i SMK N 1 Posumaen sangat dihargai. Penghargaan ini juga diperluas kepada Fakultas Ilmu Komputer Universitas Klabat serta program studi Sistem Informasi dan program studi Teknik Informatika, yang kontribusinya telah menjadi bagian integral dari keberhasilan program ini. Kerjasama dan dukungan dari semua pihak ini tidak hanya memperkaya pengalaman pendidikan siswa/i SMK N 1 Posumaen tetapi juga memperkuat hubungan antara institusi pendidikan dan komunitas yang kami layani.

DAFTAR PUSTAKA

- [1] Setiyani, L. (2019) ‘[Software Engineering] Lila Setiyani , S . T , M . Kom’, (May), pp. 20–25.
- [2] Vanya Karunia Mulia Putri, 2022, Rekayasa Perangkat Lunak: Pengertian dan Tujuannya, <https://www.kompas.com/skola/read/2022/07/18/100000169/rekayasa-perangkat-lunak--pengertian-dan-tujuannya/>, diakses tgl 13 Oktober 2023
- [3] Indra Maulana, 2022, Apa Itu DevOps? Yuk Kenalan dengan Dev Ops dan Tugas-Tugasnya!, <https://www.niagahoster.co.id/blog/devops-adalah/>, diakses tgl 13 Oktober 2023
- [4] Novi Herawati, 2022, DevOps: Definisi, Tujuan, Cara Kerja, dan Manfaatnya, <https://www.hashmicro.com/id/blog/devops-definisi-tujuan-cara-kerja-dan-manfaatnya/>, diakses tgl 13 Oktober 2023
- [5] Feradhita NKD, 2021, Pengertian, Model, dan Tahapan SDLC (Software Development Life Cycle), <https://www.logique.co.id/blog/2021/04/28/tahapan-sdlc/>, diakses tgl 13 Oktober 2023
- [6] Gilbran Maulana, 2022, Pengenalan DevOps - Pengertian, Tahapan Kerja dan Manfaat, <https://id.horangi.com/blog/pengenalan-devops-pengertian-tahapan-kerja-dan-manfaat/>, diakses tgl 13 Oktober 2023
- [7] Jessica Yunanda Bahtiar, 2023, Apa itu CI/CD? Tahapan, Tools, dan manfaatnya dalam DevOps, <https://www.sekawanmedia.co.id/blog/apa-itu-ci-cd/>, diakses tgl 13 Oktober 2023
- [8] Nurul Huda, 2022, CI/CD: Pengertian, Manfaat, dan Tools yang digunakan, <https://www.dewaweb.com/blog/pengertian-ci-cd/>, diakses tgl 13 Oktober 2023
- [9] Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016, May). What is DevOps? A systematic mapping study on definitions and practices. In Proceedings of the scientific workshop proceedings of XP2016 (pp. 1-11).
- [10] Luz, W. P., Pinto, G., & Bonifácio, R. (2019). Adopting DevOps in the real world: A theory, a model, and a case study. *Journal of Systems and Software*, 157, 110384.

- [11] Leite, L., Rocha, C., Kon, F., Milojicic, D., & Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys (CSUR)*, 52(6), 1-35.
 - [12] Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and its practices. *IEEE software*, 33(3), 32-34.
 - [13] Riungu-Kalliosaari, L., Mäkinen, S., Lwakatare, L. E., Tiihonen, J., & Männistö, T. (2016). DevOps adoption benefits and challenges in practice: A case study. In *Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22-24, 2016, Proceedings 17* (pp. 590-597). Springer International Publishing.
 - [14] Bou Ghantous, G., & Gill, A. (2017). DevOps: Concepts, practices, tools, benefits and challenges. *PACIS2017*.
 - [15] Senapathi, M., Buchan, J., & Osman, H. (2018, June). DevOps capabilities, practices, and challenges: Insights from a case study. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018* (pp. 57-67).
-