



Meningkatkan Efisiensi Pengembangan Perangkat Lunak Melalui Transparansi dan Kolaborasi dengan Ubuntu

Syifa Tri Agusriani¹, Rina^{2*}, Rakhmadi Rahman³

¹⁻³Sistem Informasi Institut Teknologi Bacharuddin Jusuf Habibie, Parepare, Indonesia

Alamat : Jalan Pemuda No.6 Kota Parepare, Sulawesi Selatan, Indonesia

Korespondensi penulis: *rinh9663@gmail.com

Abstract: Efficiency in software development is crucial to meet the increasing demands of the market today. This efficiency encompasses not only faster production times but also maintaining high quality and user satisfaction. Transparency and collaboration have been identified as critical factors in achieving this efficiency. Ubuntu, with its open-source nature, offers various features that effectively support these principles. Software development is a pivotal process in the current technological era, where innovation and development speed are keys to success. Ubuntu, as one of the popular Linux distributions based on open source, has played a significant role in facilitating this process through its transparent and collaborative approach. This article aims to explore how Ubuntu enhances software development efficiency through the principles of transparency and collaboration it upholds. This article will delve deeper into how transparency and collaboration within the Ubuntu ecosystem can significantly contribute to improving software development efficiency. By focusing on these principles, Ubuntu continues to be a top choice in the modern software development world, bringing benefits to developers, companies, and end-users worldwide.

Keywords: Efficiency, Collaboration, Ubuntu

Abstrak: Efisiensi dalam pengembangan perangkat lunak sangat penting untuk memenuhi permintaan pasar yang semakin meningkat saat ini. Efisiensi ini tidak hanya mencakup waktu produksi yang lebih cepat tetapi juga menjaga kualitas tinggi dan kepuasan pengguna. Transparansi dan kolaborasi telah diidentifikasi sebagai faktor penting dalam mencapai efisiensi ini. Ubuntu, dengan sifat open-source-nya, menawarkan berbagai fitur yang secara efektif mendukung prinsip-prinsip tersebut. Pengembangan perangkat lunak merupakan proses penting di era teknologi saat ini, dimana inovasi dan kecepatan pengembangan adalah kunci kesuksesan. Ubuntu, sebagai salah satu distribusi Linux populer berbasis open source, telah memainkan peran penting dalam memfasilitasi proses ini melalui pendekatannya yang transparan dan kolaboratif. Artikel ini bertujuan untuk mengeksplorasi bagaimana Ubuntu meningkatkan efisiensi pengembangan perangkat lunak melalui prinsip-prinsip transparansi dan kolaborasi yang dijunjungnya. Artikel ini akan mempelajari lebih dalam bagaimana transparansi dan kolaborasi dalam ekosistem Ubuntu dapat berkontribusi secara signifikan terhadap peningkatan efisiensi pengembangan perangkat lunak. Dengan berfokus pada prinsip-prinsip ini, Ubuntu terus menjadi pilihan utama dalam dunia pengembangan perangkat lunak modern, memberikan manfaat bagi pengembang, perusahaan, dan pengguna akhir di seluruh dunia.

Kata Kunci: Efisiensi, Kolaborasi, Ubuntu

1. PENDAHULUAN

Kebutuhan akan efisiensi dalam pengembangan perangkat lunak sangat penting untuk memenuhi permintaan pasar yang semakin meningkat. Efisiensi dalam konteks ini tidak hanya mengacu pada waktu produksi yang lebih cepat tetapi juga menjaga kualitas tinggi dan kepuasan pengguna. Transparansi dan kolaborasi diidentifikasi sebagai faktor kritis dalam mencapai efisiensi ini. Ubuntu, dengan sifat *open-source*-nya, menawarkan berbagai fitur yang dapat mendukung prinsip-prinsip ini secara efektif. Pengembangan perangkat lunak adalah proses yang krusial dalam era teknologi saat ini, di mana inovasi dan kecepatan pengembangan menjadi kunci keberhasilan. Ubuntu, sebagai salah satu distribusi Linux yang populer dan berbasis *open source*, telah memainkan peran penting dalam memfasilitasi proses ini dengan pendekatan transparan dan kolaboratifnya. Artikel ini bertujuan untuk mengeksplorasi bagaimana Ubuntu dapat meningkatkan efisiensi pengembangan perangkat lunak melalui prinsip transparansi dan kolaborasi yang diusungnya. (Addin *et al.*, 2024)

Penggunaan Ubuntu dalam pengembangan perangkat lunak telah dikenal karena berbagai alasan, termasuk ketersediaan kode sumber terbuka yang memungkinkan para pengembang untuk mempelajari, memodifikasi, dan memperbaiki sistem operasi sesuai kebutuhan mereka. Hal ini tidak hanya meningkatkan transparansi dalam proses pengembangan, tetapi juga membuka peluang untuk inovasi yang lebih cepat dan responsif terhadap perubahan pasar dan teknologi. Dengan adopsi model *open source*, Ubuntu mendorong kolaborasi yang luas antara komunitas pengembang. Ini menciptakan lingkungan di mana ide-ide dapat bertukar dengan bebas, meningkatkan kecepatan dan kualitas pengembangan perangkat lunak secara keseluruhan. Komunitas yang kuat di balik Ubuntu juga memfasilitasi dukungan teknis yang dapat diandalkan dan solusi-solusi *problem-solving* yang cepat terhadap tantangan-tantangan yang muncul dalam proses pengembangan. (Agusta, 2019)

Selain itu, keberhasilan Ubuntu dalam mendukung berbagai platform dan teknologi membuatnya menjadi pilihan utama bagi pengembang perangkat lunak yang menginginkan fleksibilitas dan kemampuan integrasi yang tinggi. Dengan kompatibilitas yang luas terhadap berbagai perangkat keras dan software, Ubuntu memungkinkan pengembang untuk fokus pada inovasi aplikasi mereka tanpa terkendala oleh batasan teknis dari sistem operasi. Namun demikian, meskipun Ubuntu menawarkan berbagai keunggulan, ada tantangan dan pertimbangan yang harus dihadapi dalam pengembangan perangkat lunak. Penggunaan teknologi *open source* membutuhkan pemahaman yang mendalam terhadap kode dan arsitektur sistem, serta kemampuan untuk mengelola berbagai kontribusi dari komunitas. Ini membutuhkan komitmen dari tim pengembang untuk memastikan keamanan, kualitas, dan

kehandalan aplikasi yang dikembangkan. Artikel ini akan menggali lebih dalam tentang bagaimana transparansi dan kolaborasi dalam ekosistem Ubuntu dapat memberikan kontribusi signifikan terhadap meningkatnya efisiensi pengembangan perangkat lunak. Dengan fokus pada prinsip-prinsip ini, Ubuntu terus menjadi salah satu pilihan utama dalam dunia pengembangan perangkat lunak yang modern, membawa manfaat bagi pengembang, perusahaan, dan pengguna akhir di seluruh dunia.(Волох, 2018)

Ubuntu, sebagai sistem operasi *open source* yang didukung oleh Canonical Ltd., telah membuktikan nilai transparansi dan kolaborasi dalam pengembangan perangkat lunak. Dengan menyediakan akses terbuka ke kode sumbernya, Ubuntu memungkinkan pengembang untuk mengakses, memodifikasi, dan berkontribusi pada pengembangan sistem operasi ini secara langsung. Hal ini tidak hanya meningkatkan kecepatan pengembangan fitur baru, tetapi juga memperluas cakupan kompatibilitas dan interoperabilitas Ubuntu dengan berbagai teknologi dan infrastruktur IT yang ada. Salah satu kekuatan utama Ubuntu adalah komunitas pengembang yang luas dan beragam. Komunitas ini terdiri dari individu-individu yang memiliki berbagai latar belakang dan keahlian, yang bekerja bersama untuk memperbaiki sistem, menambahkan fitur-fitur baru, serta memberikan dukungan teknis dan solusi bagi pengguna. Kolaborasi yang terjadi di dalam komunitas ini tidak hanya menghasilkan inovasi yang signifikan, tetapi juga memastikan bahwa Ubuntu tetap relevan dan mampu bersaing di pasar yang terus berubah ini (Tabassum dan Mathew, 2014).

Selain itu, Ubuntu juga menawarkan berbagai alat pengembangan yang kuat dan dokumentasi yang komprehensif. Ini termasuk *SDK (Software Development Kit)* dan perpustakaan yang mendukung pengembangan aplikasi berbasis Linux dengan lebih efisien. Dengan memanfaatkan alat-alat ini, pengembang dapat mempercepat siklus pengembangan, mengurangi waktu yang diperlukan untuk mengimplementasikan fitur-fitur baru, dan memastikan kualitas kode yang lebih tinggi. Dengan demikian, Ubuntu tidak hanya menjadi pilihan yang kuat untuk pengembang perangkat lunak, tetapi juga berperan penting dalam mendorong standar-standar baru dalam hal transparansi dan kolaborasi dalam industri IT. Dengan terus meningkatkan efisiensi pengembangan perangkat lunak melalui pendekatan ini, Ubuntu tidak hanya memberikan nilai tambah bagi pengguna akhir melalui aplikasi yang inovatif dan andal, tetapi juga memperkuat posisinya sebagai pemimpin dalam ekosistem open source global. (Tabassum dan Mathew, 2014)

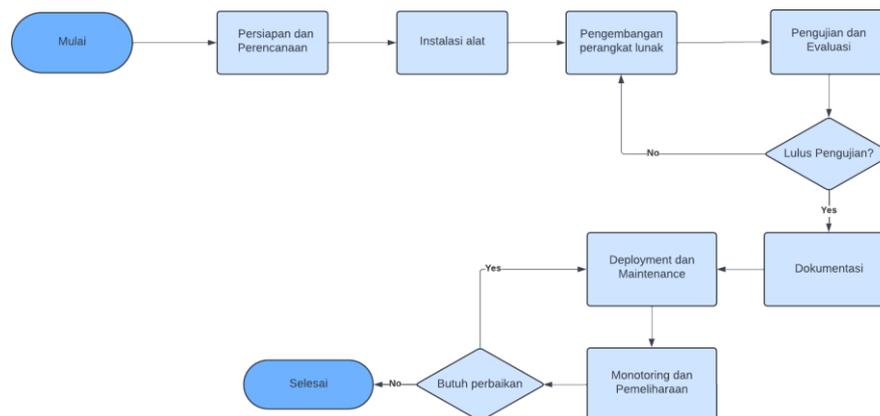
2. METODE

Studi ini menggunakan pendekatan analisis literatur untuk mengevaluasi dampak transparansi dan kolaborasi dengan Ubuntu terhadap efisiensi pengembangan perangkat lunak. Data diperoleh dari sumber-sumber seperti jurnal akademik, konferensi, dan dokumentasi teknis yang relevan. Tinjauan literatur ini membantu mengidentifikasi manfaat penggunaan Ubuntu dan alat-alat pendukungnya dalam meningkatkan produktivitas pengembangan perangkat lunak.

3. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini menggambarkan Proses pengembangan melibatkan persiapan, instalasi alat, pengembangan, pengujian, dokumentasi, *deployment*, dan pemeliharaan, menggunakan alat seperti *Git*, *GitLab*, *Docker*, dan *Visual Studio Code*. Hasil menunjukkan bahwa Ubuntu meningkatkan efisiensi dengan mengurangi waktu konfigurasi dan memberikan stabilitas. Transparansi dan kolaborasi juga meningkat melalui akses terbuka ke kode dan dokumentasi, serta alat kolaborasi yang efektif. Dukungan komunitas Ubuntu membantu menyelesaikan masalah teknis dengan cepat. Kesimpulannya, Ubuntu secara signifikan meningkatkan efisiensi pengembangan perangkat lunak. Rekomendasinya termasuk penggunaan alat yang tepat, peningkatan dokumentasi, dan pelatihan bagi tim pengembang. Berikut adalah rancangan Prototype (Husen dan Surbakti, 2020)

Tabel 1.1 Rancangan Prototype



a. Pengaturan Lingkungan Pengembangan

1. Instalasi ubuntu yang Dilakukan untuk menyediakan lingkungan yang stabil dan aman bagi pengembangan perangkat lunak. Sistem operasi Ubuntu dipilih karena sifatnya yang *open-source*, stabil, dan memiliki dukungan komunitas yang kuat.

2. Konfigurasi dasar ubuntu dengan Melakukan pembaruan sistem dan instalasi paket dasar menggunakan perintah-perintah yang relevan. Ini termasuk pemasangan alat-alat pengembangan penting seperti *Git*, *Docker*, *Visual Studio Code*, dan *GitLab* untuk kolaborasi dan manajemen proyek
3. Instalasi alat pengembangan (Kolahi *et al.*, 2015)
Git: untuk kontrol versi Memungkinkan pelacakan dan pengelolaan perubahan kode

```
sipa@sipa-virtual-machine:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.39.2-1ubuntu1.1).
0 upgraded, 0 newly installed, 0 to remove and 111 not upgraded.
```

Gambar 1

Docker: untuk kontainerisasi Mengisolasi dan mengelola lingkungan pengembangan, mempermudah pengujian dan *deployment*.

```
sipa@sipa-virtual-machine:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (24.0.5-0ubuntu1~23.04.1).
0 upgraded, 0 newly installed, 0 to remove and 111 not upgraded.
```

Gambar 2

Visual Studi Code: Sebagai IDE yang mendukung berbagai bahasa pemograman dan integrasi dengan Git

```
sipa@sipa-virtual-machine:~$ sudo snap install --classic code
Download sDownDownload snap "code" (162) from channel "stable"
code 5437499f from Visual Studio Code (vscode✓) installed
```

Gambar 3

GitLab : untuk kolaborasi dan manajemen proyek.

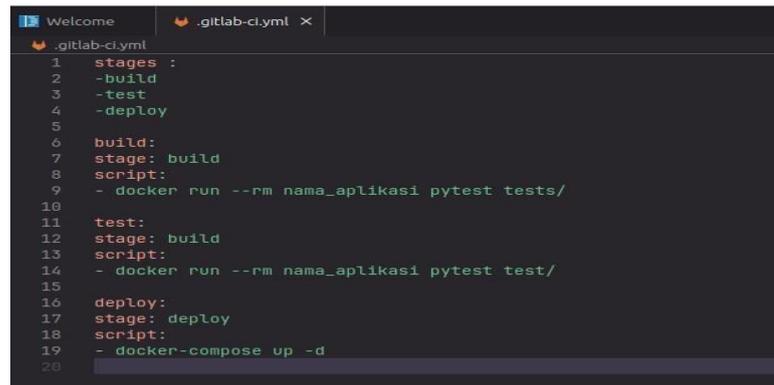
b. Manajemen Versi dan Kolaborasi

1. Membuat repostori lokal dan sinkronkan dengan GitLab, untuk mengelola kode dan mendukung kolaborasi tim.
2. Workflow *Git* (*Commit dan Push*)

c. Pengembangan dan kontainerisasi

Penggunaan *Docker* (*Build dan run kontainer*)

d. Otomatisasi pengujian dan penyebaran



Gambar 4

File `.gitlab-ci.yml` ini mengatur pipeline CI/CD di GitLab dengan tiga tahap: *build*, *test*, dan *deploy*. Pada tahap *build*, Docker menjalankan `pytest` pada direktori `tests/`. Pada tahap `test`, Docker menjalankan `pytest` pada direktori `test/` (harus diperbaiki menjadi `stage: test`). Pada tahap *deploy*, `docker-compose` memulai layanan secara terpisah. Output dari setiap tahap dapat dilihat di log pipeline GitLab untuk memeriksa keberhasilan atau kegagalan setiap perintah. (Fu, 2011)

e. Monitoring dan Pemeliharaan

Monitoring Menggunakan alat seperti Prometheus dan Grafana untuk memantau performa dan kesehatan aplikasi yang dikembangkan, sedangkan pemeliharaan memperbarui sistem dan aplikasi secara berkala. Penelitian menunjukkan bahwa penggunaan sistem operasi Ubuntu dapat secara signifikan meningkatkan efisiensi pengembangan perangkat lunak. Transparansi dan kolaborasi yang didukung oleh Ubuntu memungkinkan pengelolaan proyek yang lebih efektif dan kolaborasi tim yang lebih baik. Beberapa hasil spesifik dari penelitian ini meliputi: Efisiensi Waktu: Penggunaan Ubuntu dan alat-alat pendukungnya mengurangi waktu yang dibutuhkan untuk setup lingkungan pengembangan dan mengelola perubahan kode. Kolaborasi: Alat kolaborasi seperti Git dan GitLab memfasilitasi koordinasi tim, memungkinkan pengembang untuk bekerja secara bersamaan tanpa konflik. Stabilitas dan Keamanan: Ubuntu menyediakan platform yang stabil dan aman, yang penting untuk pengembangan perangkat lunak yang handal. Hasil dari implementasi sistem manajemen proyek menggunakan Ubuntu, Docker, dan GitLab meliputi pengurangan waktu setup dari 2-3 hari menjadi 2-3 jam, peningkatan efisiensi kolaborasi melalui penggunaan GitLab, dan pengurangan waktu rilis fitur baru dari 2-3 minggu menjadi 3-5 hari melalui pipeline CI/CD otomatis di GitLab. (Arifah *et al.*, 2022)

4. KESIMPULAN

Penggunaan Ubuntu dalam pengembangan perangkat lunak dapat secara signifikan meningkatkan efisiensi melalui transparansi yang ditingkatkan dan kolaborasi yang lebih baik. Lingkungan yang stabil dan dukungan komunitas yang luas membuat Ubuntu menjadi pilihan yang tepat bagi pengembang perangkat lunak yang mencari produktivitas yang lebih tinggi dan manajemen proyek yang efisien. Untuk penelitian lebih lanjut, disarankan melakukan evaluasi jangka panjang guna mengamati efek implementasi dalam jangka waktu yang lebih panjang, seperti menggunakan studi longitudinal dan penilaian berkala. Selain itu, penting untuk menyesuaikan alat dan metode yang digunakan agar sesuai dengan kebutuhan spesifik proyek yang berbeda melalui analisis kebutuhan, pengembangan alat khusus, dan umpan balik pengguna. Metodologi ini juga perlu diuji di berbagai lingkungan pengembangan untuk mengevaluasi konsistensi hasil, dengan melakukan studi komparatif dan mendokumentasikan penyesuaian yang diperlukan untuk mengoptimalkan metodologi.

DAFTAR PUSTAKA

- Addin, H.S., Anggraini, H., Nur, H., & Putri, R. (2024). Strategi Pengembangan Koleksi Perpustakaan Digital. *Media Informasi*, 33(1), 88–95.
- Agusta, A.S. (2019). Proses Pengembangan Koleksi Serial Elektronik. *IQRA` : Jurnal Ilmu Perpustakaan dan Informasi (e-Journal)*, 13(2), 178. <https://doi.org/10.30829/iqra.v13i2.5867>
- Arifah, S.N., Fernando, Y., & Rusliyawati, R. (2022). Upaya Meningkatkan Citra Diri Melalui Game Edukasi Pengembang Kepribadian Berbasis Mobile. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 3(3), 295–315.
- Fu, Y. (2011). Designing qualitative research. *Organization Management Journal*, 8(3), 193–195. <https://doi.org/10.1057/omj.2011.23>
- Husen, Z., & Surbakti, M.S. (2020). Membangun Server dan Jaringan Komputer dengan Linux Ubuntu. Syiah Kuala University Press.
- Kolahi, S.S., Treseangrat, K., & Sarrafpour, B. (2015). Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13. In *2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15)* (pp. 1–5). IEEE.
- Tabassum, M., & Mathew, K. (2014). Software evolution analysis of Linux (Ubuntu) OS. In *2014 International Conference on Computational Science and Technology (ICCST)* (pp. 1–7). IEEE.
- Волох, С. (2018). Ubuntu Linux с нуля. БХВ-Петербург, Kiev.