



# Penerapan *Version control system* Berbasis Web Menggunakan Next.JS, Nest.JS, Node.JS, dan MongoDB Pada Proses Pengerjaan Skripsi Mahasiswa

Axsel Riando Soplanit<sup>1</sup>, Supiyanto<sup>2\*</sup>, Agung Dwi Saputro<sup>3</sup>, Remuz MB Kmurawak<sup>4</sup>, Mingsep Rante Sampebua<sup>5</sup>

<sup>1,2,3,4,5</sup>Program Studi Sistem Informasi, Universitas Cenderawasih, Indonesia  
Email: [supi6976@gmail.com](mailto:supi6976@gmail.com)

## Abstract

*In the education systems, particularly in college, thesis courses are one of the determinants of a student's graduation. However, while working on the thesis, many students are still unable to manage their own thesis files, many factors trigger this to happen, one of which is the number of revisions which causes a loss of enthusiasm in students. This can cause students forget where he/she store the files or accidentally delete their thesis files because they have not been done for a long time. To record the entire history of thesis files, a web-based version control system must be implemented which functions to record the entire history of student thesis files. The web was created using Typescript and Javascript programming languages with the Next.JS and Nest.JS frameworks that run on the Node.JS runtime, for the database using MongoDB. Based on the test results, it is concluded that this system runs well and this system is made very private so that one user cannot enter another user's repository. The system has been tested using black box testing.*

**Keywords:** *Version control system; Student Thesis; Node.JS; Next JS; MongoDB,*

## Abstrak

*Dalam dunia pendidikan khusus nya perkuliahan, mata kuliah skripsi merupakan salah satu penentu kelulusan seorang mahasiswa. Namun selama mengerjakan skripsi banyak mahasiswa yang masih belum bisa mengelola berkas skripsi mereka sendiri, banyak faktor-faktor yang menjadi pemicu ini terjadi, salah satu nya jumlah revisi banyak yang menyebabkan hilang semangat dalam diri mahasiswa. Hal ini bisa menyebabkan mahasiswa lupa dimana menyimpan atau tidak sengaja menghapus berkas skripsi nya dikarenakan sudah lama tidak dikerjakan. Untuk merekam seluruh riwayat berkas skripsi maka harus diterapkan version control system berbasis web yang berfungsi untuk merekam seluruh riwayat berkas skripsi mahasiswa. Web dibuat menggunakan bahasa pemrograman Typescript dan Javascript dengan framework Next.JS dan Nest.JS yang berjalan diatas runtime Node.JS, untuk database menggunakan MongoDB. Berdasarkan hasil pengujian diperoleh kesimpulan bahwa sistem ini berjalan dengan baik dan sistem ini dibuat sangat privat sehingga satu pengguna tidak dapat masuk ke repository pengguna lain. Sistem telah diuji menggunakan pengujian black box*

**Kata Kunci:** *Version control system; Skripsi Mahasiswa; Node.JS; Next JS; MongoDB*

## 1. PENDAHULUAN

Teknologi informasi merupakan bidang teknologi yang berkembang pesat di era globalisasi [1]. Pengguna dapat memperoleh informasi dengan cepat dan mudah. Teknologi telah merambah berbagai bidang seperti pemerintahan, swasta, dan pendidikan. Salah satu dampak perkembangan teknologi adalah dengan ditemukan *version control system*. *Version control system* merupakan suatu sistem yang dapat mengelola serta merekam perubahan dari suatu objek

[2]. *Version control system* juga sering digunakan dalam dunia pemrograman dengan keperluan merekam baris kode program, menyimpan perubahan tertentu, dan sampai dengan backup pada baris kode program. Untuk mendapatkan gelar, mahasiswa harus mengambil mata kuliah skripsi sebagai salah satu penentu kelulusan. Selama masa mengerjakan skripsi mahasiswa akan sering bertemu dengan dosen pembimbing skripsi untuk melakukan revisi tugas akhir, pertemuan ini akan menghasilkan revisi tugas akhir dan mahasiswa yang bersangkutan akan memperbaiki berkas tugas akhirnya.

Ada mahasiswa yang sering menyimpan ulang berkas hasil revisinya berulang-ulang kali dengan nama yang berbeda. Hal ini tentu sangat tidak efisien jika mahasiswa tersebut memiliki banyak sekali revisi dari dosen yang bersangkutan. Proses revisi tugas akhir dengan cara menyimpan berkas berulang-ulang kali merupakan cara lama dan masalah sering juga terjadi dari sisi mahasiswa, dimana mahasiswa tersebut tidak bisa mengolah manajemen berkas dengan baik. Masalah lainnya adalah kebutuhan tempat penyimpanan berkas, tidak memungkinkan menyimpan berkas hanya secara lokal di komputer, seorang mahasiswa juga perlu menyimpan berkas secara online menggunakan layanan yang ada seperti Google Drive, akan tetapi jika mahasiswa tersebut mendapatkan revisi dan sudah memperbaiki berkasnya, mahasiswa tersebut juga harus menyimpan ulang berkas tersebut di layanan penyimpanan online.

Berdasarkan penelitian sebelumnya yang berhubungan dengan *version control system*, tentang manajemen kontrol revisi pada baris kode program untuk merekam seluruh perubahan pada suatu objek [2]. Penelitian ini memiliki kaitan erat dengan masalah yang telah dikemukakan oleh penulis. Adapun juga penelitian lainnya membahas tentang implementasi *version control system* untuk meningkatkan kualitas manajemen proyek sistem informasi, yang menggunakan *distributed version control system* sebagai media untuk manajemen kontrol revisi pada baris kode program dan kolaborasi antar sesama mahasiswa [3], penelitian ini memiliki masalah tentang kerja sama tim dan keahlian manajemen proyek, penggunaan platform *distributed version control system* seperti gitlab, github, dan bitbucket. Dalam penelitian lainnya mengkaji *version control system* untuk mendukung kinerja developer di PT.Jawasoft memiliki masalah tentang lokasi perusahaan yang berada di berbagai negara yang menyebabkan kendala bagi developer untuk berkolaborasi, penggunaan *version control system* dapat mendukung kinerja developer untuk bekerja sama untuk melakukan kontrol revisi berkas, melakukan rollback, dan backup baris kode program [4].

## 2. METODOLOGI PENELITIAN

### 2.1. *Version control system*

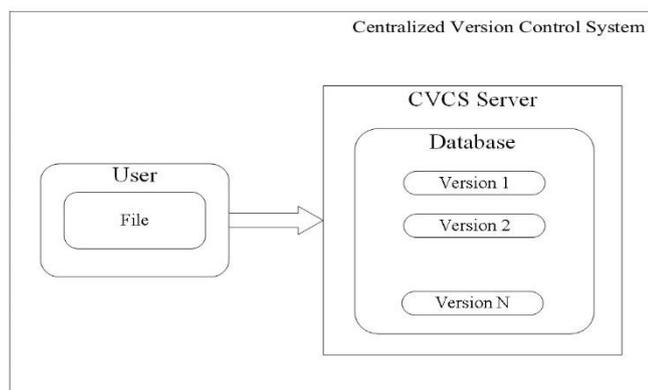
*Version control system* adalah sebuah sistem yang mengelola perubahan dari perkembangan suatu objek [5]. Dengan kata lain *version control system* juga bisa disebut sebuah sistem merekam suatu perubahan yang dilakukan pengguna. *Version control system* juga dikenal sebagai revision control system.

*Version control system* terbagi menjadi dua, *centralized version control system* (CVCS) dan *distributed version control system* (DVCS) [6]. CVCS adalah

sistem dengan satu server repository yang terpusat, sementara DVCS memiliki sistem yang tidak terpusat yang berarti setiap pengguna memiliki server repository secara lokal di komputer masing – masing.

**2.2. Centralized Version control system (CVCS)**

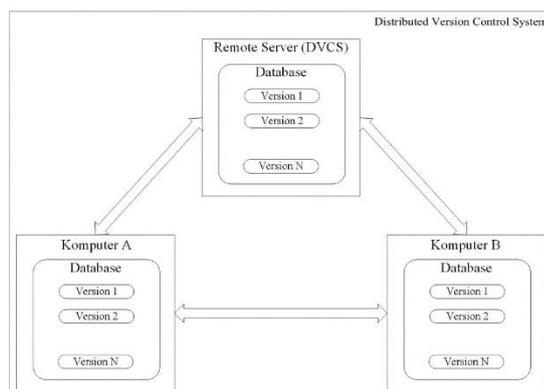
Dalam centralized version control system, server akan menyimpan repository utama dan merekam semua perubahan pada suatu objek dalam repository tersebut [7].



**Gambar 1.** Centralized version control system

**2.3. Distributed Version control system (DVCS)**

Distributed version control system merupakan version control system yang menyimpan riwayat perubahan dari suatu objek di repository lokal di komputer masing – masing, yang artinya repository tersebut tidak dapat diakses oleh orang lain, kecuali pemilik repository tersebut menggunakan layanan platform kolaborasi dan mengunggah repository tersebut ke layanan yang digunakan [8].



**Gambar 2.** Distributed version control system

Metode penelitian merupakan teknik mengumpulkan data, primer atau sekunder digunakan untuk keperluan menyusun sebuah karya ilmiah. Adapun tahapan-tahapan yang akan dilakukan oleh penulis dalam penelitian ini yaitu

wawancara, studi pustaka, analisa kebutuhan dan perancangan, implementasi, dan pembahasan hasil implementasi.

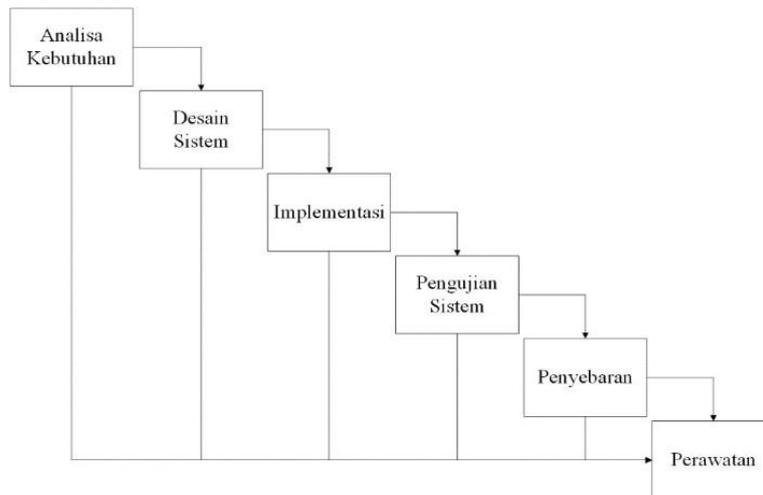


**Gambar 3. Alur Penelitian**

Metode yang digunakan pada penelitian adalah studi kasus pada suatu individu dengan pendekatan kualitatif. Penelitian kualitatif adalah penelitian yang dilakukan dengan setting tertentu yang ada di dalam kehidupan nyata bermaksud memahami suatu fenomena yang terjadi, mengapa terjadi dan bagaimana terjadinya [9].

#### 2.4. Metode pengembangan sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode waterfall. Dalam model waterfall setiap fase harus diselesaikan sebelum melanjutkan ke fase berikutnya dan tidak ada tumpang tindih di setiap fase [10].



**Gambar 4. Fase pada Metode Waterfall**

a) Analisa Kebutuhan.

Analisa kebutuhan merupakan fase dimana semua kebutuhan akan dikumpulkan dan dianalisa sehingga menghasilkan fitur – fitur yang akan diterapkan.

b) Desain Sistem.

Desain sistem merupakan fase menginterpretasikan hasil analisa kebutuhan menjadi sebuah desain sistem.

c) Implementasi.

Implementasi merupakan fase penulisan baris kode program berdasarkan hasil desain sistem yang berupa antarmuka dan server.

d) Pengujian

Pengujian sistem merupakan fase menguji dari antarmuka sampai kepada server termasuk fitur – fitur yang ada.

e) Penyebaran.

Penyebaran adalah fase publikasi sistem yang di dalam penelitian ini berbasis web. Sistem akan di unggah ke layanan hosting

f) Perawatan.

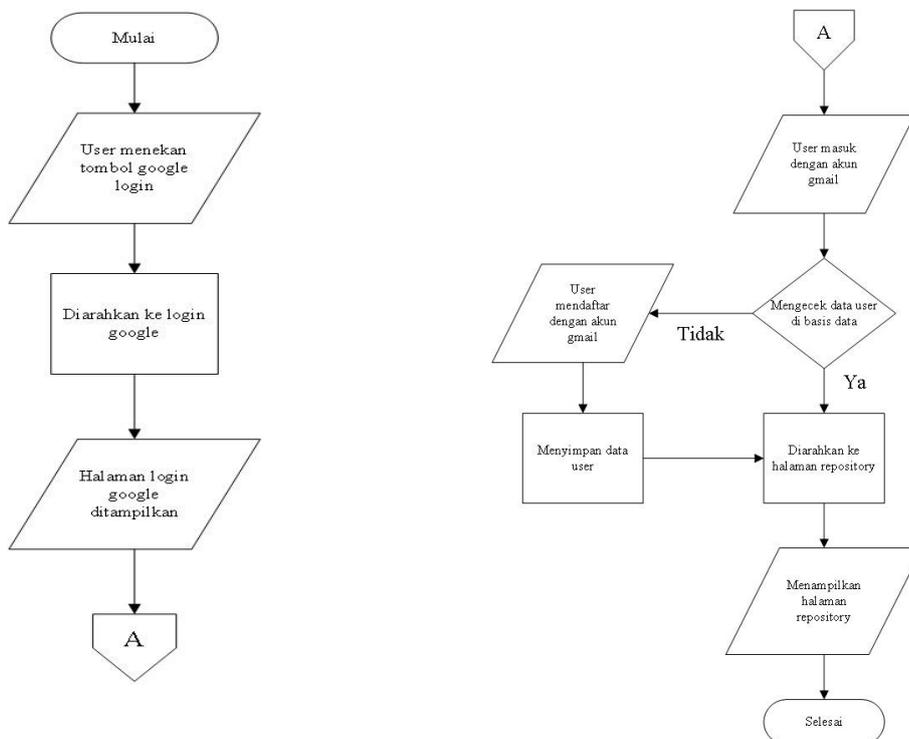
Perawatan adalah fase merawat sistem agar sisi server dan antarmuka client tetap berjalan dengan normal

**2.5. Teknik Pengujian Sistem**

Teknik pengujian sistem pada penelitian ini menggunakan black box testing. Penggunaan teknik pengujian sistem ini bertujuan untuk mengetahui fungsionalitas dari fitur yang ada tanpa harus tau struktur baris kode program dan untuk menguji kesesuaian output dari suatu input. Berdasarkan analisa yang telah dilakukan, penulis mengusulkan sebuah rancangan sistem, sistem ini menggunakan flowchart sebagai media representasi workflow yang akan terjadi di dalam sistem ini.

**2.6. Workflow Login**

Rancangan Workflow ini menjelaskan tentang proses login yang akan terjadi di dalam sistem. Detail dari workflow ini akan dijelaskan pada gambar dibawah ini.

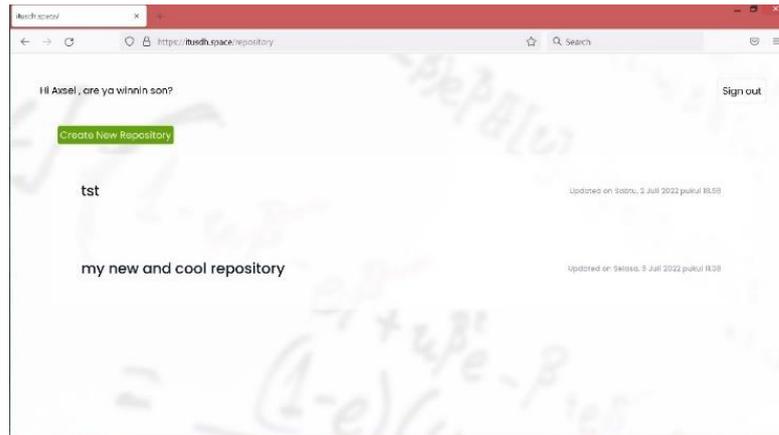


**Gambar 5.** Flowchart login



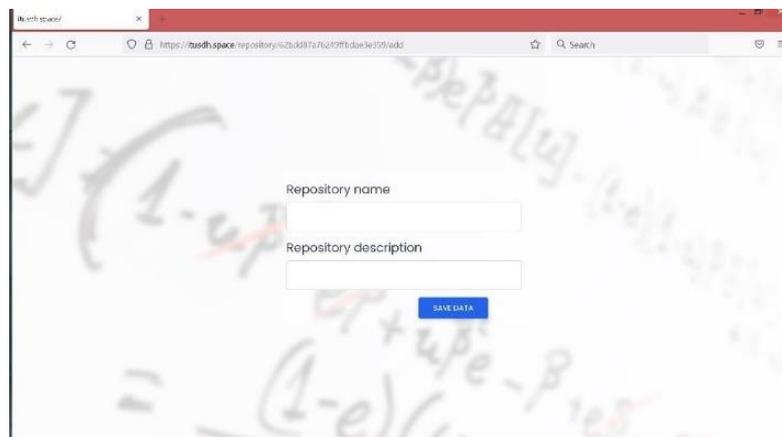
### 3.2. Halaman Repository

Halaman *repository* berisi tentang seluruh data repository yang dibuat oleh pengguna, pengguna juga bisa membuat repository baru dengan cara menekan tombol *create new repository*, dan jika pengguna ingin membuka, merubah, ataupun menghapus repository yang diinginkan pengguna hanya perlu menekan nama repository maka akan muncul beberapa tombol aksi di sudut kanan bawah.



Gambar 8. Halaman *repository*

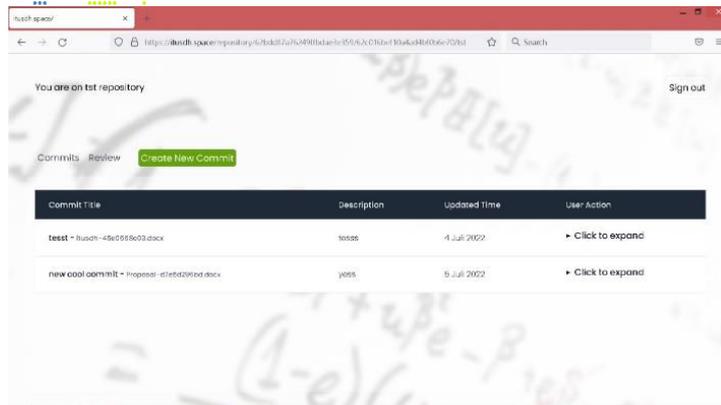
Halaman *create* dan *edit repository* ini berfungsi sebagai penerima input dari pengguna dan nantinya tersimpan di database.



Gambar 9. Halaman *create* dan *edit repository*

### 3.3. Halaman Committed File

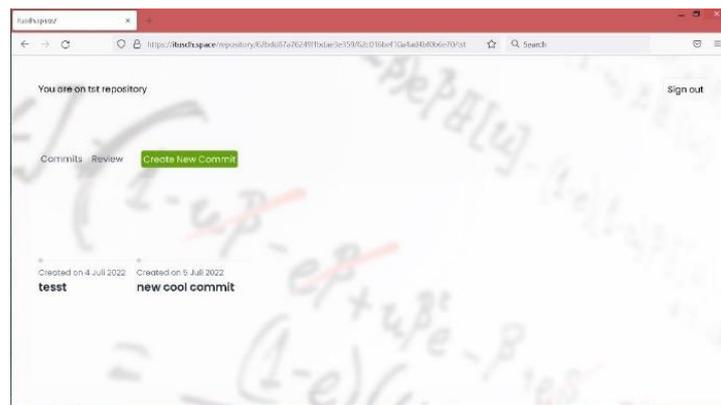
Halaman ini memiliki fungsi untuk melihat sudah berapa banyak *commit* yang dibuat oleh pengguna, dalam halaman ini pengguna juga bisa mengunduh, merubah, dan menghapus suatu *commit* dengan cara menekan tombol *click to expand*.



Gambar 10. Halaman *committed file*

### 3.4. Halaman Review

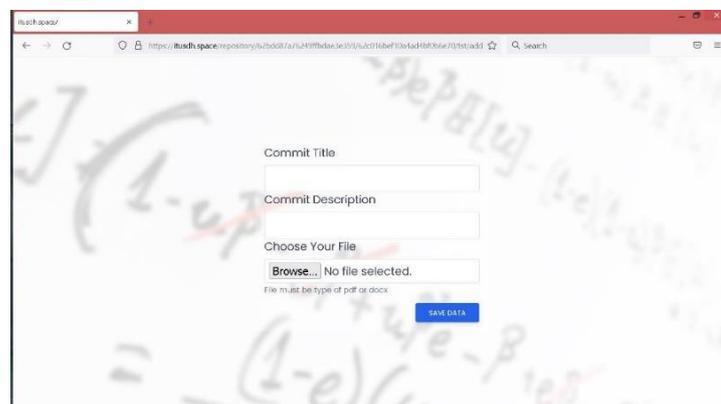
Halaman *review* berisi tentang riwayat pengguna selama menyimpan data dalam suatu *repository*.



Gambar 11. Halaman *review*

### 3.5. Halaman *create dan edit commit*

Halaman ini berfungsi untuk menerima *input* dari pengguna dan akan disimpan di *database*, untuk berkas yang diunggah akan di simpan di sisi *backend/server*.



Gambar 12. Halaman *create dan edit commit*

### 3.6. Hasil Pengujian Sistem

Pengujian pada sistem ini dilakukan secara manual menggunakan teknik pengujian *black box*, pengujian sistem ini memiliki skenario yaitu antarmuka menerima *input* dari pengguna dan akan di proses apakah *input* sudah sesuai kriteria atau belum. Pengujian ini memiliki tujuan mengetahui apakah sistem sudah sesuai dengan analisis perancangan.

### 3.7. Pengujian Fungsi Tambah dan Ubah Repository

Pengujian pada halaman *repository* bertujuan untuk menguji fungsi yang ada. Berikut adalah tabel pengujian seluruh fungsi halaman *repository*.

**Tabel 1.** Pengujian data benar fungsi tambah data repository

Field	Input	Output	Keterangan
Repository Name	My Cool Repository	New Repository Created	Sesuai
Repository Desc	My Cool Description	New Repository Created	Sesuai

**Tabel 2.** Pengujian data benar fungsi ubah data repository

Field	Input	Output	Keterangan
Repository Name	New Name Kosong	Exisisting Repository Updated	Sesuai
Repository Desc	New Desc Kosong	Exisisting Repository Updated	Sesuai

**Tabel 3.** Pengujian data salah fungsi tambah *repository*

Field	Input	Output	Keterangan
Repository Name	Kosong	Input Required	Sesuai
Repository Desc	Kosong	Input Required	Sesuai

## 4. SIMPULAN

Berdasarkan implementasi dan pengujian sistem yang telah dilakukan oleh penulis di dalam penelitian ini, dapat disimpulkan bahwa untuk menangani permasalahan mahasiswa dalam mengelola berkas skripsi, dapat diterapkan *version control system* berbasis web guna mempermudah mahasiswa dalam mengelola berkas skripsi. Dan membatasi akses pengguna terhadap repository pengguna lain.

## DAFTAR PUSTAKA

- [1] R. Komalasari, "Manfaat Aplikasi Teknologi Iot Di Masa Pandemi Covid-19 : Studi Eksploratif," *Temat. - J. Teknol. Inf. Dan Komun.*, vol. 7, no. 2, pp. 140–146, 2020, doi: 10.38204/tematik.v7i2.469.
- [2] R. Majumdar, R. Jain, S. Barthwal, and C. Choudhary, "Source code management using *version control system*," 2018, doi: 10.1109/ICRITO.2017.8342438.
- [3] Isnardi, "Implementasi *Version control system* untuk meningkatkan kualitas Manajemen Proyek Sistem Informasi," *J. Sains dan Inform.*, vol. 4, no. 1, pp. 39–43, 2019, doi: 10.22216/jsi.v4i1.

- [4] N. S. Mabror, "Kajian Version Control Dalam Mendukung Kinerja Developer Pada Pt. Jawasoft," *JIKA (Jurnal Inform.)*, vol. 2, no. 1, pp. 58–69, 2018, doi: 10.31000/jika.v2i1.1191.
- [5] N. N. Zolkifli, A. Ngah, and A. Deraman, "Version control system: A Review," 2018, doi: 10.1016/j.procs.2018.08.191.
- [6] D. Spinellis, "Version control systems," *IEEE Softw.*, vol. 22, no. 5, pp. 108–109, 2005, doi: 10.1109/MS.2005.140.
- [7] S. Vaidya, S. Torres-Arias, R. Curtmola, and J. Cappos, "Commit signatures for centralized version control systems," *IFIP Adv. Inf. Commun. Technol.*, vol. 562, pp. 359–373, 2019, doi: 10.1007/978-3-030-22312-0\_25.
- [8] K. Mulu, C. Bird, N. Nagappan, and J. Czerwonka, "Transition from centralized to decentralized version control systems: A case study on reasons, barriers, and outcomes," *Proc. - Int. Conf. Softw. Eng.*, no. 1, pp. 334–344, 2014, doi: 10.1145/2568225.2568284.
- [9] M. R. Fadli, "Memahami desain metode penelitian kualitatif," *HUMANIKA*, vol. 21, no. 1, 2021, doi: 10.21831/hum.v21i1.38075.
- [10] Y. Researchers *et al.*, "Waterative Model: an Integration of the Waterfall and Iterative Software Development Paradigms," *Database Syst. J.*, vol. X, no. 1, pp. 75–81, 2019.