

## Implementasi CI/CD Pipeline Pada Framework Androbase Menggunakan Jenkins (Studi Kasus: PT. Andromedia)

Ahmad Farid<sup>1\*</sup>, Indra Gita Anugrah<sup>2</sup>  
<sup>1,2</sup> Universitas Muhammadiyah Gresik

Jl. Sumatera No.101, Kabupaten Gresik, Jawa Timur 61121

Corresponding author's e-mail: ahmadfarid.gresik@gmail.com<sup>1</sup>, indragitaanugrah@umg.ac.id<sup>2</sup>

**Abstrak**— Proses *deployment* dalam pembangunan sebuah aplikasi sangatlah penting karena menentukan hasil akhir sebuah aplikasi namun sayangnya proses ini kerap kali bermasalah dikarenakan cara *deploy* yang masih manual yaitu dengan cara pull aplikasi dari server produksi, cara ini dirasa kurang praktis dikarenakan *developer* harus berulang kali melakukan pull dari server apabila ada update pada aplikasi, masalah serupa terjadi pada sebuah perusahaan jasa system informasi PT.Andromedia dimana proses *deploy* yang dilakukan masih secara manual, berangkat dari permasalahan tersebut maka penulis menawarkan solusi dengan mengintegrasikan proses *deployment* aplikasi dengan menggunakan konsep *continuous integration and deployment* atau yang biasa disingkat CI/CD. CI atau *Continuous integration* Merupakan proses dimana aplikasi akan di buat dan di uji secara otomatis setelah repositori aplikasi terintegrasi pada server CI. Sedangkan CD atau *Continuous deployment* adalah proses dimana aplikasi yang telah dibuat dan di uji akan *deploy* pada server produksi secara otomatis. dengan mengimplementasikan CI/CD dapat membantu dalam proses *deployment*.

Kata kunci: *Deployment, Continuous Integration, Continuous deployment, Jenkins, Aplikasi*

**Abstrak**— The deployment process in the development of an application is very important because it determines the final result of an application, but unfortunately this process is often problematic because the deployment method is still manual, namely by pulling the application from the production server, this method is considered less practical because the developer must repeatedly pull from the server if there is one. updating the application, a similar problem occurred in an information system service company PT. Andromedia where the deployment process was still done manually, departing from these problems, the author offered a solution by integrating the application deployment process using the concept of continuous integration and deployment or commonly abbreviated as CI./CD. CI or Continuous integration Is a process where applications will be created and tested automatically after the application repository is integrated on the CI server. While CD or Continuous deployment is a process where applications that have been created and tested will be deployed on the production server automatically. implementing CI/CD can help in the deployment process.

**Keywords:** *Deployment, Continuous Integration, Continuous deployment, Jenkins, Application*

### 1. Pendahuluan

Dalam proses pengembangan (*development*) sebuah aplikasi memiliki tahapan-tahapan yang harus dilakukan oleh pengembang aplikasi (*developer*) seperti analisis, design, *develop* dan *release* atau *deploy*. *Deploy* adalah tahap dimana aplikasi yang telah jadi akan dipublikasi atau disebar tergantung aplikasi yang dibuat. Dalam pengembangan sebuah aplikasi tahapan *deploy* adalah tahap yang sangat penting karena menentukan hasil akhir sebuah aplikasi, namun sayangnya tahapan ini kerap kali bermasalah, dikarenakan sistem yang masih bersifat manual. Cara yang dilakukan yaitu dengan memasukan aplikasi ke server kemudian *developer* menjalankan aplikasi tersebut. Apabila ada update pada aplikasi maka langkah tadi harus diulang. Sehingga menggunakan cara ini dirasa kurang efisien dan efektif karena *developer* harus *deploy* berulang kali apabila terdapat pengembangan pada aplikasi yang telah dibuat.

CI/CD merupakan salah satu dari bagian *DevOps*. *DevOps* sendiri merupakan salah satu praktik dalam proses pengembangan aplikasi dimana pada bagian *development and Operations* dilakukan secara otomatis dan dan saling terintegrasi, dengan menggunakan *DevOps* proses pengembangan aplikasi dapat dilakukan secara efektif dan efisien.

PT. Andromedia adalah sebuah perusahaan yang bergerak di bidang jasa system informasi. PT.Andromedia juga memiliki *framework* sendiri yang bernama androbase, yang digunakan untuk mempermudah proses *development*. Dalam proses *development* aplikasi pertama-tama para *developer* akan dibagikan fitur-fitur aplikasi yang akan dikerjakan pada computer masing-masing setelah fitur yang dikerjakan tidak ada masalah maka code disimpan pada tempat penyimpanan atau *repository*, setelah semua fitur yang dikerjakan selesai maka fitur-fitur tadi akan digabung atau *merge* sehingga menghasilkan aplikasi yang utuh, setelah itu aplikasi akan di test apabila tidak ada masalah maka aplikasi akan di *deploy* ke server, apabila setelah di *deploy* terdapat

masalah maka para *developer* akan memperbaiki aplikasi dan melakukan *deploy* ulang secara manual, pada proses *deploy* ini dirasa kurang efektif karena para *developer* harus melakukan *deploy* manual secara teratur apabila ada masalah pada aplikasi.

Solusi yang dapat ditawarkan untuk permasalahan di atas adalah mengintegrasikan dengan sebuah sistem atau aplikasi yang dapat melakukan *deploy* dan sekaligus testing aplikasi yang dibuat secara otomatis apabila ada perubahan pada tempat penyimpanan atau *repository* aplikasi, dalam hal ini sistem atau aplikasi yang digunakan adalah *jenkins*. *Jenkins* sendiri merupakan aplikasi *open-source* yang digunakan untuk melakukan otomatisasi tugas-tugas di dalam proses *continuous integration and delivery (CI/CD)* sebuah aplikasi, solusi ini diharapkan dapat mempermudah *developer* PT. Andromedia dalam melakukan proses *deploy* aplikasi.

## 2. Tinjauan Pustaka

### 2.1. CI/CD Pipeline

*Continuous integration and deployment* atau yang biasa disingkat *CI/CD* merupakan proses otomatisasi pada tahap *testing* dan *deploy* pada sebuah aplikasi dengan mengintegrasikan aplikasi pada layanan pihak ketiga, *CI/CD* dapat memastikan kode aplikasi selalu diuji, dan aman untuk di *deploy*. *CI* atau *Continuous integration* merupakan proses dimana aplikasi akan di buat dan di uji secara otomatis setelah *repository* aplikasi terintegrasi pada server *CI*. Sedangkan *CD* atau *Continuous deployment* adalah proses dimana aplikasi yang telah dibuat dan di uji akan *dideploy* pada server produksi secara otomatis [1].

### 2.2. Jenkins

*Jenkins* adalah aplikasi *open-source* yang digunakan untuk melakukan otomatisasi tugas-tugas di dalam proses *continuous integration and delivery (CI/CD)* sebuah aplikasi. *Jenkins* di pilih karena memiliki banyak fitur dan plugin yang mendukung semua *repository* seperti Github, Gitlab, Bitbucket dan lainnya [9].

### 2.3. Framework

*Framework* adalah sebuah kerangka kerja atau dasar dari suatu pembuatan aplikasi yang menyediakan fitur-fitur untuk mempermudah dan mempercepat proses pembuatan aplikasi, dengan menggunakan *framework* para *developer* tidak lagi harus membuat aplikasi dari awal [15].

### 2.4. Androbase

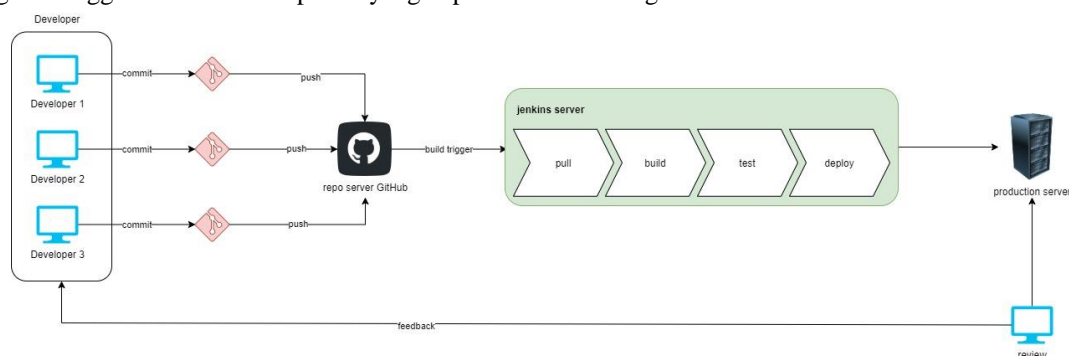
*Androbase* adalah *framework* yang dibuat oleh PT. Andromedia untuk mempermudah para *developernya* dalam *mendevolop* aplikasi, *Androbase* sendiri dibuat dengan menggunakan *javascript* sebagai Bahasa pemrogramannya, *react js* untuk *frontend* dan *prisma js* untuk *object relational model (ORM)*.

### 2.5. Git

*Git* merupakan *version control* yang digunakan untuk memelihara versi dari code, sehingga setiap *developer* dapat membuat *branch* mereka masing-masing, *git* juga dapat dikombinasikan dengan *git repository* yang dapat memudahkan para *developer* berkolaborasi dan memenejemen code [3][8].

## 3. Metode Penelitian

Dalam metode penelitian ini memiliki tahapan-tahapan yang dibuat dalam acuan dalam pembuatan penelitian. Tahapan pertama adalah identifikasi masalah, pada tahapan ini penulis mengidentifikasi masalah pada *development* aplikasi di PT. Andromedia dimana pada saat *development* aplikasi masih menggunakan cara manual. Tahapan kedua adalah rumusan masalah dimana pada tahapan ini penulis mengajukan solusi dengan menggunakan *CI/CD Pipeline* yang dapat di lihat dalam gambar berikut.



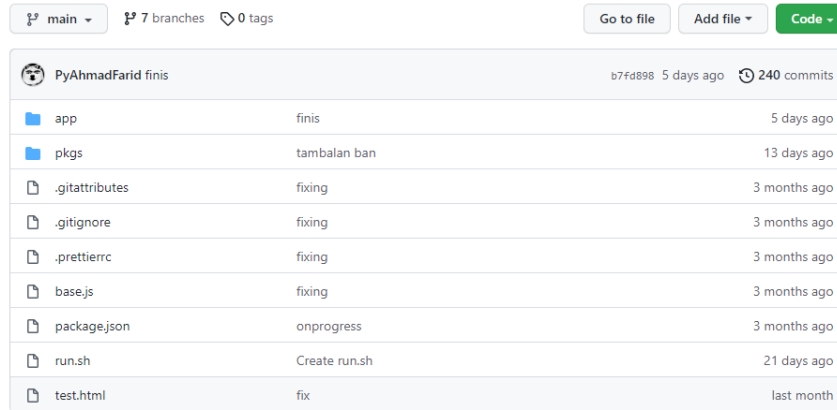
Gambar 1. Alur CI/CD pipeline

Tahapan selanjutnya adalah mengimplementasikan *CI/CD Pipeline* dengan menggunakan *jenkins*, Dan untuk tahap terakhir adalah pengujian dan hasil kesimpulan dimana aplikasi yang telah terintegrasi dengan *jenkins*

akan di uji dan dari hasil pengujian tersebut dapat di Tarik hasil kesimpulan.

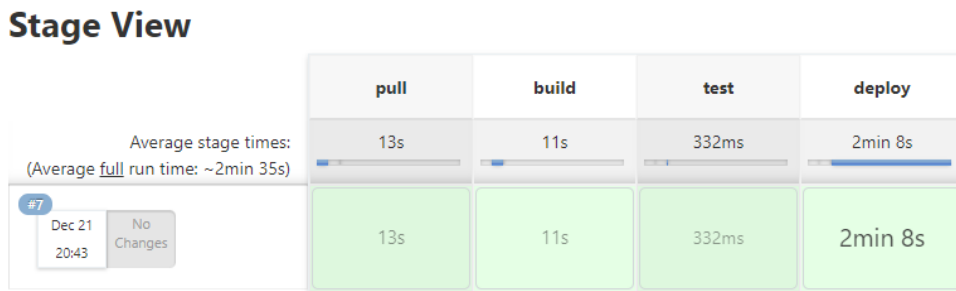
#### 4. Hasil dan Pembahasan

Penelitian ini di awali dengan melakukan instalasi *git* pada computer masing-masing *developer* dan *jenkins* pada *server* lokal, setelah aplikasi yang di butuhkan terinstall maka langkah selanjutnya adalah membuat *repository* baru di *github* kemudian *framework* androbase di masukan (*push*) ke *repository* tersebut, *repository* yang dibuat dapat dilihat pada gambar berikut.



Gambar 2. Repositori GitHub

Untuk langkah berikutnya adalah membuat *job pipeline* pada *jenkins* yang kemudian *job* tersebut diintegrasikan dengan *repository github* yang telah dibuat, Setelah *job* terintegrasi dengan *github* maka langkah selanjutnya adalah mendefinisikan *stage* *stage* yang akan di eksekusi apabila terdapat perubahan pada *repository github* *stage* *stage* yang dibuat dapat dilihat pada gambar berikut



Gambar 3. Stage success

Pada gambar 3 dapat di lihat *pipeline* memiliki 4 *stage* yang di eksekusi secara berurutan,

##### 1. Stage Pull

*Stage* pertama adalah *stage pull* dimana *jenkins* akan mengambil aplikasi dari *repository github* (*pull*) yang kemudian disimpan ke *directory* lokal.

##### 2. Stage Build

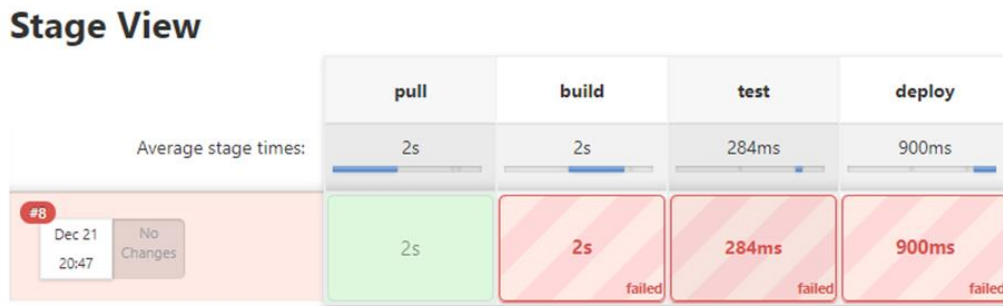
*Stage* ke dua adalah *stage build* dimana pada *stage* ini aplikasi yang telah di *pull* akan dibangun (*build*) sehingga menghasilkan aplikasi yang seutuhnya.

##### 3. Stage Test

*stage* selanjutnya adalah *stage test* dimana pada *stage* ini aplikasi akan di uji apabila pada *stage* ini tidak terdapat masalah atau *error* maka aplikasi akan masuk ke selanjutnya.

##### 4. Stage Deploy

*Stage* terakhir adalah *stage deploy* dimana aplikasi akan di *deploy* ke *server* produksi. Apabila terdapat *error* pada salah satu *stage* maka *stage* berikutnya tidak akan di jalankan sehingga hal ini dapat mencegah *deployment* aplikasi yang terdapat masalah.



Gambar 4. Stage failed

Pada gambar 4 dapat dilihat saat *stage build* terdapat masalah maka *stage test* dan *stage deploy* tidak tereksekusi, untuk log dari pipline Jenkins dapat dilihat pada gambar berikut.

```
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\androbse
[Pipeline] {
[Pipeline] stage
[Pipeline] { (pull)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\androbse\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/fadollunmun-im/eRent.git # timeout=10
Fetching upstream changes from https://github.com/fadollunmun-im/eRent.git
> git.exe --version # timeout=10
> git --version # 'git version 2.32.0.windows.2'
> git.exe fetch --tags --force --progress -- https://github.com/fadollunmun-im/eRent.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse 'refs/remotes/origin/main^{commit}' # timeout=10
Checking out Revision b7fd8980926cb7ff591fb7ced053176545ed9e13 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f b7fd8980926cb7ff591fb7ced053176545ed9e13 # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D main # timeout=10
> git.exe checkout -b main b7fd8980926cb7ff591fb7ced053176545ed9e13 # timeout=10
Commit message: "finis"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (build)
[Pipeline] bat
C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\androbse>npm install
> base@1.0.0 postinstall
> patch-package && node base clean
patch-package 6.4.7
Applying patches...
No patch files found
[ 14:59:20 | base ] Clearing build folder...
up to date, audited 1726 packages in 2m
147 packages are looking for funding
run 'npm fund' for details
found 0 vulnerabilities
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (test)
[Pipeline] echo
test
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deploy)
[Pipeline] sleep
[Pipeline] }
```

Gambar 5. Log Jenkins

Pada penelitian ini aplikasi yang di *deploy* adalah contoh aplikasi yang dibangun dengan menggunakan *framework* androbse. Berikut adalah tampilan *framework* androbse yang sukses di *deploy* menggunakan *jenkins*.

ID	Type	Title	Layout	Domain	Slug
14213	Layout	parent-layout	—	All Domain	
49484	Layout	layout-parent	—	All Domain	
80739	Layout	mobile-layout	—	All Domain	
00300	Layout	layout-mobile	—	All Domain	
11531	API	tes-notif	—	All Domain	/api/notif
12123	API	forgot-password	—	All Domain	/api/customers/forgot-password
15243	API	list-admin	—	All Domain	/api/admin
15342	API	order-detail	—	All Domain	/api/order-detail/{id}
15680	API	upload-bukti-pembayaran	—	All Domain	/api/transaksi/{id}/upload-bukti
15835	API	list-track-template	—	All Domain	/api/track-template
16155	API	hash-pass	—	All Domain	/api/hash-pass
17287	API	update-barang	—	All Domain	/api/barang-admin/{id}/update
21707	API	delete-admin	—	All Domain	/api/admin/{id}/delete
22927	API	update-tracking	—	All Domain	/api/transaksi/{id}/update-tracking
29961	API	ganti-password-admin	—	All Domain	/api/admin/{id}/change-password

Gambar 6. Androbase

Dengan mengintegrasikan aplikasi androbase dengan *CI/CD pipeline* dapat mempermudah proses *deployment* aplikasi pada PT. Andromedia yang mana pada sistem sebelumnya masih menggunakan cara manual namun dengan menggunakan *CI/CD* proses *deployment* menjadi lebih cepat dan otomatis, untuk hasil pengujian *CI/CD* penulis melakukan perbandingan antara system *CI/CD* dengan system manual, hasil perbandingan dapat dilihat pada table berikut

No	Sebelum	Sesudah
1	Proses deploy menggunakan cara manual dengan memasukkan aplikasi ke server secara langsung.	Proses deploy menggunakan cara otomatis dengan menggunakan jenkins.
2	Update aplikasi produksi dilakukan secara rutin.	Update aplikasi produksi dilakukan apabila terdapat update pada repository.
3	Testing secara manual.	Testing dilakukan secara otomatis.

Tabel 1. Pengujian

## 5. Kesimpulan

Dari hasil penelitian dapat disimpulkan bahwa dengan menerapkan sistem *CI/CD pipeline* dengan menggunakan aplikasi *jenkins* dapat mempermudah proses *deployment* aplikasi dimana proses *deployment* yang awalnya menggunakan cara manual dengan *CI/CD* dapat dilakukan secara otomatis cara ini juga dapat menghemat waktu dan juga dapat meminimalisir kesalahan. Dengan menggunakan sistem ini semoga dapat membantu *developer* PT.Andromedia dalam mendevlop aplikasi.

## Daftar Pustaka

- [1] Indra Guna Noviantama and Ari Purno Wahyu W, “IMPLEMENTASI CONTIONOUS INTEGRATION DAN CONTINUOUS DEPLOYMENT PADA APLIKASI LEARNING MANAGEMENT SYSTEM DI PT. MILLENNIA SOLUSI INFORMATIKA,” *Jurnal Ilmiah Teknologi Informasi Terapan*, vol 8, no 1, 2021.
- [2] Fileni Zalukhu and Veri Arinal, “IMPLEMENTASI SISTEM PERSEDIAAN BARANG BERBASIS WEB DENGAN METODE DevOps PADA PT. HEINZ ABC INDONESIA,” *SOSTECH (Jurnal Sosial dan Teknologi)*, vol 1, no 1, 2021.
- [3] Danur Wijayanto, Arizona Firdonsyah and Faisal Dharma Adhinata, “Implementasi Continous Integration/Continous Delivery Menggunakan Process Manager 2 (Studi Kasus: SIAKAD Akademi Keperawatan Bina Insan),” *TEKNIKA*, vol 10, no 3, 2021.
- [4] Umi Sa’adah, et al., “FRAMEWORK TESTING OTOMATIS BERBASIS SERENITY DAN JENKINS AUTOMATED BUILD,” *JUTI (Jurnal Ilmiah Teknologi Informasi)*, vol 19, no 2, 2021.

- 
- [5] Rendy Wijaya, Andri Prasetyo and Dian Wahyuningsih, "IMPLEMENTASI CI/CD UNTUK BUIDDAN DEPLOY WEBSITE DENGAN DOCKER RUNNER PADA ORGANISASI BELAJAR LINUX ID," *SeNTIK (Seminar Nasional Teknologi Informasi dan Komunikasi)*, 2021.
- [6] Sendy Ferdian, et al., "Continuous Integration and Continuous Delivery Platform Development of Software Engineering and Software Project Management in Higher Education," *JuTISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol 7, no 1, 2021.
- [7] Abriza Mahandis Shama and Dian W. Chandra, "Implementasi Static Application Security Testing Menggunakan Jenkins CI/CD Berbasis Docker Container Pada PT. Emporia Digital Raya," *Jurnal Ilmiah Informatika (JIF)*, vol 9, no 2, 2021.
- [8] Rahmad Ade Putra, "ANALISA IMPLEMENTASI ARSITEKTUR MICROSERVICES BERBASIS KONTAINER PADA KOMUNITAS PENGEMBANG PERANGKAT LUNAK SUMBER TERBUKA (OPENDAYLIGHT DEVOPS COMMUNITY)," *Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*, vol 9, no 2.
- [9] Jaga My Prieria and Robi Tanzil Ganefi, "AUTOMATIC DEPLOYMENT SYSTEM DENGAN MENGGUNAKAN METODE CONTINUOUS INTEGRATION DI KAKATU," *Jurnal Ilmiah Komputer dan Informatika*.
- [10] Andrian Alpery and Muhammad Arif Fadhly Ridha, "IMPLEMENTASI CI/CD DALAM PENGEMBANGAN APLIKASI WEB MENGGUNAKAN DOCKER DAN JENKINS," *9th Applied Business and Engineering Conference*, 2021.
- [11] Tohirin, et al., "Implementasi DevOps pada Pengembangan Aplikasi e-Skrining Covid-1," *JURNAL MULTINETICS*, vol 6, no 1, 2020.
- [12] Fransiskus Andika Setiawan, et al., "PENGUJIAN PROYEK WEBSITE OTOMATISASI DENGAN PENDEKATAN INTEGRASI ANTARA SELENIUM DAN TESTNG PADA ENVIROMENT JENKINS," pp. 1-14, 2015.
- [13] Syafrial fachri pane, et al., "Implementasi Middleware Pada Evomo Dengan Metode Web Service Restfull Dan Pengujian CI/CD, Coverage Serta Simulasi Protokol Grafana," *Jurnal Tekno Insentif*, vol 15, no 2, 2021.
- [14] Wahyuni Puji Lestari and Ari Sujarwo, "DevOps: Disrupsi pengelolaan ICT pendidikan tinggi," *SNATI*, 2018.
- [15] Mara Destiningrum and Qadhli Jafar Adrian, "SISTEM INFORMASI PENJADWALAN DOKTER BERBASIS WEB DENGAN MENGGUNAKAN FRAMEWORK CODEIGNITER (STUDI KASUS: RUMAH SAKIT YUKUM MEDICAL CENTRE)," *Jurnal TEKNOINFO*, vol 10, no 2, 2017.