

## PERANCANGAN INFRASTRUKTUR DEVOPS SISTEM INFORMASI PUSAT HKI LPPM UNSOED

Muhammad Daniswara Sutarman<sup>1</sup>

Program Studi Teknik Elektro Universitas Jenderal Soedirman  
Email: mdannis12@gmail.com

Ari Fadli<sup>1</sup>

Program Studi Teknik Elektro Universitas Jenderal Soedirman  
Email: arifadli@unsoed.ac.id

Muhammad Syaiful Aliim<sup>1</sup>

Program Studi Teknik Elektro Universitas Jenderal Soedirman  
Email: muhammad.syaiful.aliim@unsoed.ac.id

**ABSTRACTS:** Intellectual Property Rights (IPR) are thinking activities to produce a product or process that is of value to humans. The LPPM Unsoed HKI Center is an institution tasked with managing and maintaining IPR documents. In managing documents, the Intellectual Property Center of LPPM Unsoed has used a good document management information system, but the development of the system is still conventional. This causes the system development team to find it difficult to quickly release the information system when there are requests for additional features or improvements due to bugs. Therefore, the Intellectual Property Center of LPPM Unsoed needs a technological infrastructure that can be used to accommodate these problems. DevOps is a software development method emphasizing communication, collaboration, and integration. In Addition, DevOps supports software production speed and automation, allowing software to be released more frequently as needed. This study aims to design and implement DevOps-based information system development infrastructure at the HKI Center of LPPM Unsoed. The DevOps infrastructure consists of Ubuntu 20.04 Linux, Github, Jenkins, Docker, and Apache Web Server. This study uses the Black Box Testing method to test the results of DevOps infrastructure implementation. The black box test results show that information system access is 60%, and the menu can be accessed properly. The study results show that the DevOps method has been successfully applied to the LPPM Unsoed Intellectual Property Center in building infrastructure to develop a management information system.

**Keywords:** Intellectual Property Rights, Information Systems, DevOps, Jenkins, Github, Docker.

**ABSTRAK:** Hak Kekayaan Intelektual (HKI) adalah kegiatan berpikir untuk menghasilkan suatu produk atau proses yang bernilai bagi manusia. Pusat HKI LPPM Unsoed merupakan lembaga yang bertugas mengelola dan memelihara dokumen HKI. Dalam pengelolaan dokumen Pusat HKI LPPM Unsoed telah menggunakan sistem informasi manajemen dokumen yang baik, namun pengembangan sistem tersebut masih secara konvensional. Hal ini menyebabkan tim pengembang sistem kesulitan melakukan rilis yang cepat terhadap sistem informasi, ketika ada permintaan penambahan fitur ataupun perbaikan karena adanya bug. Oleh karena itu, Pusat HKI LPPM Unsoed membutuhkan sebuah infrastruktur teknologi yang dapat digunakan untuk mengakomodasi permasalahan tersebut. DevOps merupakan sebuah metode pengembangan perangkat lunak yang menekankan pada komunikasi, kolaborasi dan integrasi. Selain itu, DevOps mendukung kecepatan produksi perangkat lunak serta otomatisasi, sehingga memungkinkan perangkat lunak dirilis lebih sering sesuai keperluan. Penelitian ini bertujuan merancang dan mengimplementasikan infrastruktur pengembangan sistem informasi berbasis DevOps pada Pusat HKI LPPM Unsoed. Infrastruktur DevOps terdiri dari Ubuntu 20.04 Linux, Github, Jenkins, Docker, dan Apache Web Server. Penelitian ini menggunakan metode Black Box Testing untuk menguji hasil implementasi infrastruktur DevOps. Hasil pengujian black box menunjukkan akses sistem informasi sebesar 60%, dan menu dapat diakses dengan baik. Hasil studi menunjukkan bahwa metode DevOps telah berhasil diterapkan pada Pusat HKI LPPM Unsoed dalam membangun infrastruktur untuk mengembangkan sistem informasi manajemen.

**Kata Kunci:** HKI, Sistem Informasi, DevOps, Jenkins, Github, Docker.

### PENDAHULUAN

Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, bersifat manajerial dan kegiatan strategi pada suatu organisasi tertentu [1]. Dalam perkembangannya sistem informasi telah di implementasikan dalam berbagai bidang, diantaranya yaitu Sistem Informasi Geografis (SIG) yang merupakan sistem berbasis komputer yang mampu menyajikan data dalam bentuk peta digital [2]. Sistem informasi yang dapat menjadi salah satu solusi yang dapat diterapkan di bidang kesehatan [3]. Sistem Informasi Akademik (SIA) akan mengintegrasikan seluruh proses inti sebuah bisnis pendidikan tinggi ke dalam sebuah sistem informasi yang didukung oleh teknologi terkini [4].

Selain implementasi tersebut, sistem informasi ini telah menjadi kebutuhan didalam sebuah organisasi/lembaga dalam melakukan pengelolaan dan pemeliharaan dokumen. Sebuah lembaga yang baik akan sangat memperhatikan hal tersebut karena pengelolaan dan pemeliharaan dokumen yang baik

---

<sup>1</sup> Program Studi Teknik Elektro Universitas Jenderal Soedirman

akan memberikan dampak positif bagi kelangsungan hidup sebuah lembaga. Setiap lembaga akan sangat kental dalam melakukan kegiatan pengelolaan dan pemeliharaan dokumen. Lembaga pada sebuah institusi pendidikan tinggi akan senantiasa menghasilkan banyak jenis dan ragam dokumen dalam menjalankan proses bisnisnya, sehingga membutuhkan pengelolaan yang baik untuk mengetahui segala macam aktivitas yang dilakukan oleh organisasi pada sebuah institusi pendidikan tinggi tersebut. aktivitas dan mutu dari lembaga perguruan tinggi tersebut.

Beberapa penelitian terkait pada pengelolaan dan pemeliharaan dokumen yaitu sistem informasi pengelolaan arsip digital berbasis web untuk mengatur sistem kearsipandi SMK Tri Karya menggunakan metode object orientasi menggunakan Unified Modelling Language (UML) [5]. Selain itu, penelitian tentang perancangan pengelolaan dokumen juga dilakukan menggunakan metode object oriented (berorientasi objek) menggunakan Unified Modelling Language (UML)[6]. Pada penelitian lainnya, perancangan sistem informasi manajemen dokumen (studi kasus STMIK Mikroskil) dilakukan dengan menggunakan Data Flow Diagram (DFD). Penelitian ini, memberikan hasil diantaranya adalah mampu mengefisienkan biaya perawatan dan penyimpanan dokumen [7].

Pembuatan Sistem informasi manajemen dokumen menggunakan metode PIECES di rumah sakit Bhayangkara Lumajang. Hasil penelitian menunjukkan bahwa prosentase kepuasan untuk variabel kinerja (performance) sebesar 81% menyatakan sangat baik, variabel informasi (information) sebesar 81% menyatakan sangat baik, variabel ekonomi (economic) sebesar 69% baik, variabel kontrol (control) sebesar 76% menyatakan sangat baik, variabel efisien (efficiency) sebesar 70% menyatakan sangat baik, dan variabel pelayanan (service) 80% responden sangat baik [8]. Implementasi DevOps digunakan pada pengembangan aplikasi e-Skrining Covid-19. Penelitian ini, mengadopsi DevOps dalam mengimplementasikan pengembangan aplikasi e-Skrining Covid-19 karena memudahkan dalam melakukan proses pengembangan perangkat lunak pada satu antarmuka web dengan perangkat GitLab [9]. Sedangkan pada peneliti lain menggunakan metode Entity Relationship Diagram dan Data Flow Diagram dalam menghasilkan Sistem Informasi Manajemen Dokumen Fakultas Teknologi Informasi Universitas Kristen Maranata [10].

Pusat HKI LPPM Unsoed merupakan lembaga yang bertugas untuk mengelola dan memelihara dokumen HKI yang dihimpun dari berbagai organisasi maupun individu. Dalam pengelolaan dokumen Pusat HKI LPPM Unsoed telah menggunakan sistem informasi manajemen dokumen yang baik, namun pengembangan sistem tersebut masih secara konvensional. Hal ini menyebabkan tim pengembang sistem kesulitan melakukan rilis yang cepat terhadap sistem informasi, ketika ada permintaan penambahan fitur ataupun perbaikan karena adanya bug. Sehingga Pusat HKI LPPM Unsoed membutuhkan sebuah infrastruktur teknologi yang dapat digunakan untuk mengakomodasi permasalahan tersebut. Devops merupakan sebuah metode pengembangan perangkat lunak yang menekankan pada komunikasi, kolaborasi dan integrasi [11]. Selain itu, DevOps mendukung kecepatan produksi perangkat lunak serta otomatisasi, sehingga memungkinkan perangkat lunak dirilis lebih sering sesuai keperluan[12]. Berdasarkan beberapa penelitian yang telah dilakukan dan permasalahan yang dihadapi tersebut, sehingga pada penelitian ini dilakukan perancangan dan pengimplementasian infrastruktur berbasis DevOps pada pengembangan sistem informasi di Pusat HKI LPPM Unsoed. Metode pengujian yang digunakan pada penelitian ini adalah metode black box.

## METODE PENELITIAN

Penelitian ini dilaksanakan di Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Unsoed, Purwokerto. Penelitian ini menggunakan beberapa alat dan bahan yaitu. Perangkat keras laptop dengan sistem operasi windows 10, yang didukung oleh perangkat lunak XAMPP versi 3.2.4, visual studio code, sistem operasi windows 10 64 bit, linux ubuntu 20.04, virtualbox, jenkins, docker, dan gitHub. Metode penelitian yang digunakan dalam penelitian ini yaitu menggunakan metode DevOps. Adapun tahapan yang digunakan dalam melaksanakan penelitian ini tampak seperti pada **Gambar 1**.

Penelitian ini memanfaatkan kolaborasi dan komunikasi antara pengembang perangkat lunak dan profesional lainnya. Tahap pertama adalah studi pustaka, hal ini bertujuan untuk menghimpun informasi mengenai penelitian terdahulu tentang implementasi DevOps, khususnya pada pengembangan sistem informasi pengelolaan dokumen. Selain itu, studi pustaka juga dilakukan pada prosedur, *file* dan *tools* yang digunakan membangun infrastruktur *DevOps* sebagai pondasi dalam mengembangkan sistem informasi. Selanjutnya tahap membangun infrastruktur *DevOps* yang digunakan untuk sistem informasi HKI. Setelah selesai tahap pembangunan maka dilakukan uji fungsionalitas dan evaluasi, dimana pada tahap ini dilakukan pengujian apakah sistem sudah berjalan

dengan baik atau tidak. Jika masih ditemukan *error* maka dilakukan perbaikan dengan cara melakukan rekonfigurasi pada infrastruktur *DevOps*. Tahapan akhir dari penelitian ini adalah pembuatan laporan yang berisikan mengenai informasi detail konfigurasi yang tepat dalam membangun infrastruktur *DevOps* untuk sistem informasi.

Pada tahap awal dalam membangun infrastruktur *DevOps*, maka dilakukan melakukan instalasi beberapa perangkat lunak yang dibutuhkan, seperti Linux Ubuntu 20.04, Jenkins, Docker, GitHub.

### Instalasi Linux Ubuntu 20.04

Sistem operasi Linux Ubuntu 20.04 pada penelitian ini dipergunakan untuk menampilkan setiap perubahan yang terjadi dari proses *DevOps*. Sistem operasi ini merupakan sistem operasi yang berjalan menggunakan aplikasi tambahan yaitu VirtualBox. Linux Ubuntu 20.04 yang diinstall merupakan sistem operasi virtual yang berjalan di atas windows 10.

### Instalasi Jenkins

Setelah tahapan instalasi sistem operasi virtual Linux Ubuntu 20.04. Langkah selanjutnya adalah instalasi jenkins pada Linux. Jenkins ini berfungsi sebagai CI/CD (*Continuous Integration/ Continuous Delivery*) pada sistem informasi Pusat HKI LPPM Unsoed.

### Instalasi Docker

Docker yang digunakan pada penelitian ini di install pada sistem operasi linux, sedangkan untuk *Docker* hub sendiri dapat diakses melalui windows karena berfungsi untuk membuat repository *Docker* yang dapat menampilkan proses *push Docker Image* dan dapat diintegrasikan dengan *Jenkins*. Setelah *Docker* terinstal di linux, proses pembuatan *image* dan *container* dilakukan didalam *Docker*. *Docker* berfungsi untuk membangun sistem informasi yang menggunakan *framework codeigniter 4* yang selanjutnya akan dirubah menjadi sebuah *Docker Image*. Selanjutnya, *Docker Image* yang sudah dibuat tadi disusun menjadi *container*.

### Instalasi GitHub

*GitHub* yang digunakan pada penelitian ini diakses melalui laman website milik Github dengan menggunakan mekanisme penyambungkan kunci ssh pada sistem operasi yang akan digunakan, yang berfungsi untuk menghubungkan antara sistem operasi yang akan melakukan perintah Git ke dalam Github. Pada penelitian ini digunakan kunci ssh pada windows yang disambungkan ke Github, sehingga oleh karena itu proses Git seperti *add*, *pull*, *commit*, *push* dilakukan pada sistem operasi windows yang nantinya akan merubah dari repository Github yang akan menjadi objek penelitian.



■ Gambar 1. Flowchart penelitian

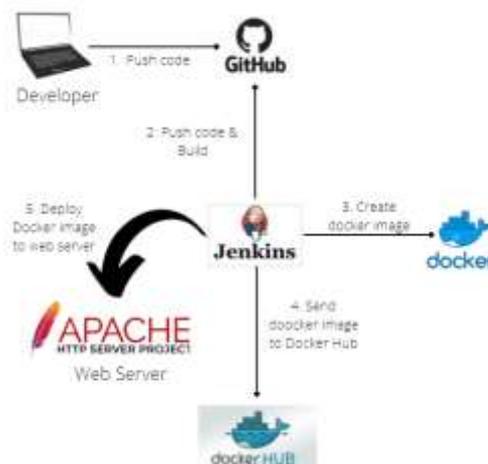
Selanjutnya, setelah proses instalasi semua perangkat lunak selesai dilakukan, maka pengembangan sistem informasi memulai siklus DevOps nya, sebagaimana tampak pada Gambar 2, yaitu *Development* dan *Operation*, sebagai berikut.

1. *Development*, tahapan ini, terdiri dari beberapa sub tahapan sebagai berikut:
  - a. *Plan*, pada tahapan ini dilakukan perencanaan kebutuhan yang digunakan dalam membangun sistem informasi yang digunakan untuk Pusat HKI LPPM Unsoed.
  - b. *Code*, pada tahapan ini dilakukan perancangan atau pembuatan sistem informasi menggunakan visual studio code, semua code yang ditulis selanjutnya push ke *repository* Github.
  - c. *Build*, pada tahapan ini dilakukan pembangunan code menggunakan Jenkins. Pada tahap ini Github, Docker, dan Jenkins diintegrasikan dengan tujuan untuk membangun infrastruktur *continuous integration/continuous deployment*, sehingga proses *build* dapat berjalan secara otomatis.
  - d. *Test*, pada tahapan ini dilakukan pengujian pada sistem informasi yang telah selesai dibangun, Pada tahap ini, peneliti melakukan pengujian fungsi *DevOps* pada sistem informasi.
2. *Operations*, tahapan ini, terdiri dari beberapa sub tahapan sebagai berikut:
  - a. *Deploy*, Sistem Informasi HKI akan *deploy* menggunakan Jenkins, sehingga prosesnya dapat berjalan secara otomatis. *Dockerfile* yang disimpan pada *repository* akan dieksekusi oleh *jenkins*, sehingga setelah kontainer berhasil dibentuk akan di *deployed* ke docker hub secara otomatis.
  - b. *Operate*, pada tahap ini pengoperasian sistem informasi pengelolaan dokumen pada Pusat HKI LPPM Unsoed dilakukan.
  - c. *Release*, pada tahap ini, sistem yang telah bekerja baik maka akan dibungkus ke dalam *docker image* menjadi *dockerfile* lalu dijalankan oleh *docker container* dan akan di-*release* ke docker hub sebagai *production server*.

## HASIL DAN PEMBAHASAN

### A. Arsitektur *DevOps* untuk Sistem Informasi HKI Unsoed

■ Gambar 2 menunjukkan arsitektur *DevOps* yang digunakan pada pengembangan sistem informasi pengelolaan dokumen pada Pusat HKI LPPM Unsoed. Berdasarkan gambar 3, maka terdapat beberapa tahapan yang dibutuhkan dalam mengembangkan arsitektur tersebut. Pada tahap pertama *developer* melakukan penulisan *source code* yang kemudian *source code* yang dibuat dipush ke *repository github*, sehingga *repository github* ter-*update* dan mengalami perubahan. Pada *Jenkins* akan menarik (*pull*) sumber kode dari repositori *Gitlab* ke server *Jenkins*. Penelitian ini, menggunakan menggunakan server *Jenkins* milik Teknik Elektro Unsoed yang ada pada tautan <https://jenkins.ee.unsoed.ac.id/>, seperti tampak pada ■ Gambar 4. Selanjutnya, kode yang diperoleh dari *repository Gitlab* tersebut dibangun (*build*) dan disusun (*compile*). Keseluruhan proses secara otomatis di dalam *Jenkins*. Apabila *repository github* mengalami perubahan akan dilakukan *pull repository github* yang berisi *source code* yang telah dibuat *developer* dan memasukkannya ke dalam folder *docker*.



■ Gambar 2. Arsitektur *DevOps* Sistem Informasi HKI Unsoed

Pada tahap berikutnya, dilakukan *build docker image* dengan menggunakan *docker engine*. Setelah *docker image* berhasil dibuat dilakukan *push* ke *repository docker hub* sehingga *repository docker hub* ter-update dan mengalami perubahan. Kemudian aplikasi tersebut akan dipindahkan ke tahap produksi (*production server*) dan ditampilkan ke *web service* menggunakan perangkat *docker*.

## B. Implementasi *DevOps* untuk Perancangan dan Pengembangan Sistem Informasi Pusat HKI

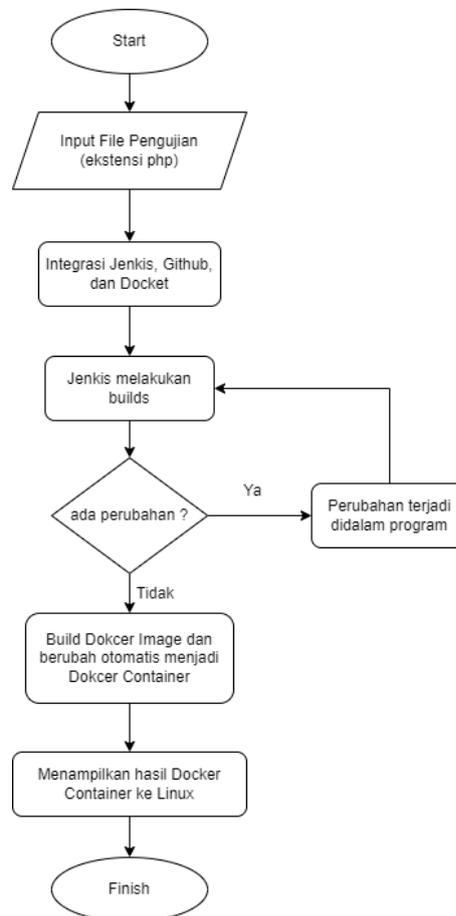
Pada penelitian ini tahapan perancangan sistem ini diimplementasikan menggunakan rilis. Rilis nantinya akan dijelaskan rancangan sebelum terjadi perubahan antara sebelum perubahan dan sesudah perubahan dari rilis tersebut. Maka dari itu penelitian ini membagi rilis ke dalam rilis A dan rilis B. Pada pengembangan sistem akan dijelaskan mengenai perubahan antara sistem rilis A dan rilis B.

### Rilis A

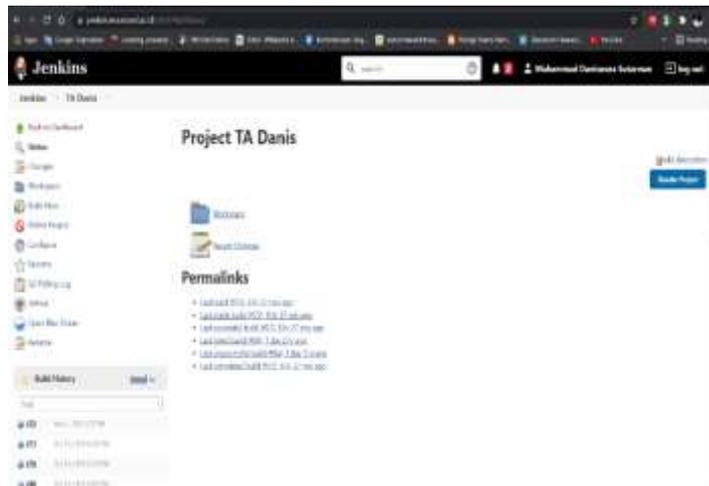
Pada ■ Gambar 3 menunjukkan *flowchart* rilis A, gambar ini menjelaskan alur kerja dari proses rilis A. Pada tahapan rilis A ini dapat dijelaskan dengan beberapa proses, yaitu *plan*, *code*, *build*, *test*, *release*, *deploy*, dan *feedback*. Berikut merupakan tahapan prosesnya.

#### 1. *Plan*

Pada tahap ini dibuat rencana mengenai bagaimana sistem *DevOps* agar dapat berjalan sesuai dengan skenario yang diinginkan. Studi pustaka mengawali tahapan ini, hal ini bertujuan untuk mencari arsitektur *DevOps* yang sesuai dengan sistem yang akan dikembangkan.



■ Gambar 3. Diagram Alir Skenario Rilis A



■ Gambar 4. Tampilan server Jenkins Unsoed

2. *Code*

Pada tahap ini dilakukan pemrograman terhadap sistem awal *DevOps* agar bisa berjalan dengan lancar sesuai dengan prinsip kerja *DevOps* yaitu *Contious Integration / Continuous Delivery (CI/CD)*. Pada program awal untuk melakukan rilis A ini digunakan kode program sederhana yang berfungsi untuk mengecek apakah sistem *DevOps* berhasil berjalan atau tidak.

3. *Build*

Pada tahap ini Github, Docker, dan Jenkins diintegrasikan dengan tujuan untuk membangun infrastruktur *continuous integration/continuous deployment*, sehingga proses *build* dapat berjalan secara otomatis. Pada tahap ini *source code* akan dipush ke GitHub dengan menggunakan Gitbash dan tersimpan di repository GitHub yang dapat diakses di alamat <https://github.com/danis506/tadanis.git>, kemudian kode tersebut akan diambil oleh jenkins sehingga otomatis untuk melakukan build dan terhadap kode tersebut di cloud sebagaimana ditunjukkan pada Gambar di bawah. Pada Gambar di bawah ini dapat dilihat bahwa proses build dan testing berhasil dalam waktu 48 detik. Dari data tersebut dapat disimpulkan bahwa proses build rilis A pada sistem ini dapat berjalan dengan baik.

4. *Test*

Pada tahap ini, dilakukan pengujian fungsi *DevOps* pada sistem. Tahap pengujian ini bertujuan untuk menguji infrastruktur *DevOps* sebagaimana rancangan yang diusulkan pada bagian metode. Hasil pengujian infrastruktur *DevOps* pada sistem rilis A dapat dilihat pada Tabel 1.

■ Tabel 1. Hasil Pengujian Rilis A

No.	Tujuan	Tools	Hasil	Waktu proses	Keterangan
1	<i>Continuous Development</i>	Github	Gambar 4.17	2 Jam	Berhasil
2	<i>Continuous Integration dan Continuous Deployment</i>	Jenkins	Gambar 4.18	48 Detik	Berhasil

Pada ■ Tabel 1 dapat dilihat bahwa seluruh pengujian yang sudah dilaksanakan sudah sesuai dengan tujuan infrastruktur *DevOps* pada sistem informasi Pusat HKI Unsoed. Pada Tabel di atas juga dapat dilihat waktu proses untuk menyelesaikan tujuan infrastruktur *DevOps* untuk *continuous integration* dan *continuous deployment* pada sistem di rilis A ini menghasilkan waktu 48 detik sehingga dapat dikatakan efektif. Jadi, dapat disimpulkan bahwa pengujian infrastruktur *DevOps* untuk *continuous integration* dan *continuous deployment* pada rilis A ini berhasil karena infrastruktur *DevOps* yang sudah dirancang dapat berjalan dengan baik dengan waktu yang efektif.



■ Gambar 5. Proses Build Rilis A di Jenkins

### 5. Release

Setelah proses *test*, selanjutnya dilanjutkan tahap *release* di GitHub menggunakan platform *Docker*. Jadi pada saat *release* sistem akan dibungkus ke dalam *Docker image* menjadi *Docker file*, lalu dijalankan oleh *docker container* dan akan di-*release* ke *docker hub* sebagai *production server*. GitHub pada proses ini sudah terintegrasi dengan Jenkins sehingga source code yang sudah berhasil dipush dapat diakses pada alamat <https://github.com/danis506/tadanis.git>. Pada GitHub dapat dilihat *release* untuk branch master rilis A.

### 6. Deploy

Proses deployment pada rilis A pada ini adalah tampilan yang telah berhasil dipasang pada *localhost*, kemudian di-*deploy* ke *production server*. Pada penelitian ini, *production server* yang digunakan adalah *docker hub*. Penerapan *continuous deployment* pada penelitian ini melibatkan beberapa *tools*, di antaranya *git*, *gitlab*, *jenkins* dan *docker*. ■ Gambar 6 berikut ini adalah tampilan dari hasil rilis A.



■ Gambar 6. Tampilan hasil DevOps rilis A

### 7. Feedback

Pada tahap ini dilakukan analisis dan evaluasi terhadap sistem yang telah dibangun. Pada tahap ini pengguna meminta perubahan terhadap sistem baik penambahan, pengurangan, ataupun perubahan fungsi dari suatu fitur yang ada di sistem tersebut. Kemudian pengembang akan melakukan perubahan terhadap sistem dengan mendesain kembali sistem agar sesuai seperti yang diharapkan pengguna, sehingga sistem tersebut terus mengalami perkembangan secara kontinu.

### Rilis B

Setelah rilis A selesai diamati, tahap selanjutnya dengan menjalankan skenario penelitian berikutnya, yaitu dengan membuat proses rilis B. ■ Gambar 7 menunjukkan *flowchart* rilis B. Pada

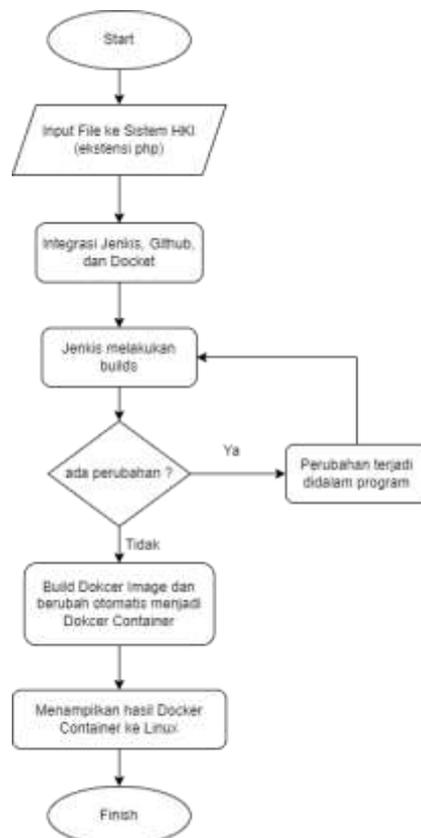
tahapan rilis B ini dapat dijelaskan dengan beberapa proses, yaitu *plan*, *code*, *build*, *test*, *release*, *deploy*, dan *feedback*. Berikut merupakan tahapan prosesnya.

1. *Plan*

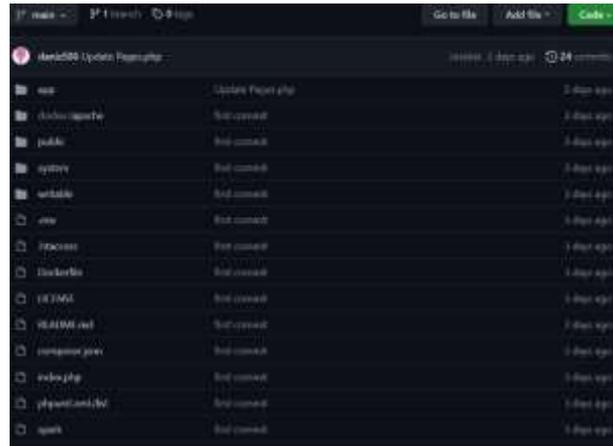
Rilis B berbeda dengan rilis A, pada rilis B ini digunakan adalah Sistem Informasi Pengelolaan Dokumen HKI LPPM Unsoed yang dikembangkan dengan menggunakan *framework* codeigniter 4.

2. *Code*

Pada tahap ini dilakukan pembuatan kode sistem informasi HKI agar dapat berjalan pada sebuah metode DevOps yaitu *Continuous Integration / Continuous Delivery (CI/CD)*. Setelah melakukan pembuatan *code* pada program sistem informasi HKI, selanjutnya program yang telah dirancang menggunakan perintah *git* untuk *add*, *pull*, *commit* dan *push* ke dalam *repository* GitHub yakni <https://github.com/danis506/tadanis.git>, seperti tampak pada ■ Gambar 8. Konfigurasi *git* dibutuhkan pada tahap ini, yaitu dengan membangun koneksi kunci *ssh* Windows ke dalam akun GitHub. Setelah itu program sederhana tadi bisa dipush ke dalam *repository* GitHub dengan *branch* *main* menggunakan perintah *git*. Berikut ini merupakan gambar dari *repository* dan *Dockerfile* rilis B.



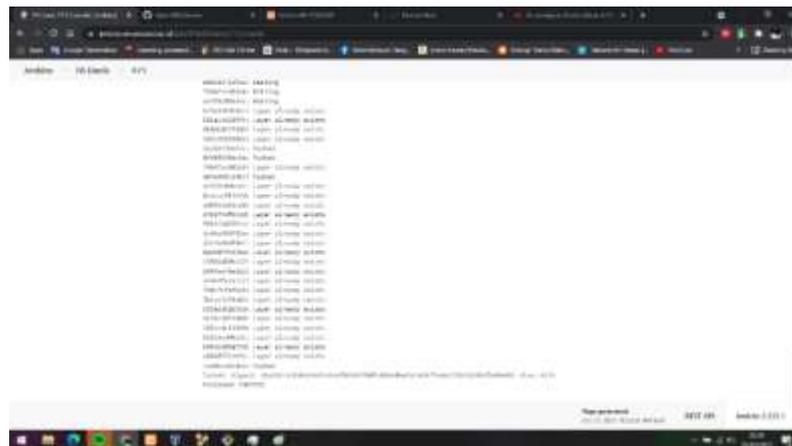
■ Gambar 7. Diagram Alir Skenario Rilis B



■ Gambar 8. Repository Github Rilis

### 3. Build

Pada tahap build, dilakukan proses *build* menggunakan Jenkins. Pada tahap ini Github, Docker, dan Jenkins diintegrasikan dengan tujuan untuk membangun infrastruktur *continuous integration / continuous deployment*, sehingga proses *build* dapat berjalan secara otomatis. Pada ■ Gambar 9 dapat dilihat bahwa proses *build* dan *testing* pada rilis B berhasil dalam waktu 88 detik. Berdasarkan data tersebut dapat disimpulkan bahwa proses build rilis B pada sistem ini dapat berjalan dengan baik.



■ Gambar 9. Proses build rilis B di Jenkins

### 4. Test

Pada tahap ini, dilakukan pengujian hasil implementasi *DevOps* pada sistem informasi. Tahap pengujian ini bertujuan untuk menguji kinerja infrastruktur *DevOps*. Hasil pengujian infrastruktur *DevOps* pada sistem rilis B dapat dilihat pada ■ Tabel 2.

■ Tabel 2. Hasil Pengujian Rilis B

No.	Tujuan	Tools	Hasil	Waktu proses	Keterangan
1	<i>Continuous Development</i>	Github	Gambar 4.27	7 Hari	Berhasil
2	Continuous Integration dan Continuous Deployment	Jenkins	Gambar 4.28	88 Detik	Berhasil

Pada ■ Tabel 2 dapat dilihat bahwa seluruh pengupengujian yang sudah dilaksanakan sudah sesuai dengan tujuan infrastruktur *DevOps* pada sistem informasi Pusat HKI Unsoed. Pada ■ Tabel 2 juga dapat dilihat waktu proses untuk menyelesaikan tujuan infrastruktur *DevOps* untuk *continuous integration* dan *continuous deployment* pada sistem di rilis B ini menghasilkan waktu 88 detik sehingga dapat dikatakan efektif. Namun pada saat pull hasil *build* ke Docker image dan Docker container

membutuhkan waktu sekitar 120 detik agar bisa ditampilkan di *localhost*. Jadi, dapat disimpulkan bahwa pengujian infrastruktur DevOps untuk *continuous integration* dan *continuous deployment* pada rilis B ini berhasil karena sistem informasi HKI menggunakan metode *DevOps* yang sudah dirancang dapat berjalan dengan baik dengan waktu yang efektif, ■ Gambar 10 menunjukkan CI/CD pada rilis B.

```
Jenkins TA Danis #71
#080972121: waiting
f064fac02a8: waiting
ee355e0b6e1: waiting
676a940d4a7: Layer already exists
581a1c20078: Layer already exists
804004172948: Layer already exists
185c9a29306a: Layer already exists
7a3d2730057: Pushed
8f48355fac1a: Pushed
f064fac02a8: Layer already exists
207c305c33b7: Pushed
ee355e0b6e1: Layer already exists
80ec1d71e48: Layer already exists
e08f1b62b08: Layer already exists
279279dfbc1d: Layer already exists
70ba24707ca: Layer already exists
7c04e950781e: Layer already exists
2377600f8d7: Layer already exists
8a58376530a: Layer already exists
13504030617: Layer already exists
0847a0f0e23: Layer already exists
732b0fc2a17: Layer already exists
72067c5040f5: Layer already exists
2bca07a794e2: Layer already exists
095a062b55d: Layer already exists
9a3b0071808: Layer already exists
382c4c15388: Layer already exists
8a252a90a41: Layer already exists
300b3208a75d: Layer already exists
e080972121: Layer already exists
7a3d2730057: Pushed
Latest: #1261: sha256:1c338a7e52c9c8f8d391707fc4884a6a5a7a007fda5721743627606a0e2 size: 6179
Finished: SUCCESS
```

■ Gambar 10. Continuous integration dan continuous deployment rilis B

### 5. Release

Setelah proses *test*, selanjutnya dilanjutkan tahap *release* di GitHub menggunakan platform *docker*. Sama seperti rilis A, rilis B juga dalam tahap *release* tidak ada perbedaan pada tahapannya. Jadi pada saat *release* sistem akan dibungkus ke dalam *Docker image* menjadi *Dockerfile*, lalu dijalankan oleh *docker container* dan akan di-*release* ke *docker hub* sebagai *production server*, prosesnya dapat dilihat pada ■ Gambar 11.

### 6. Deploy

Hasil dari proses *deployment* pada rilis B ini adalah sistem informasi HKI Unsoed yang telah berhasil dipasang pada *localhost* dengan port 8080 di Linux. Pada awal proses *Devops*, sistem tersebut disebar (*deploy*) ke *production server*. Tampilan sistem informasi HKI yang tersimpan pada *localhost* akan disebar (*deploy*) menggunakan *Jenkins*, sehingga prosesnya berjalan secara otomatis dengan memanfaatkan *execute shell* pada server *Jenkins*. *Dockerfile* yang disimpan pada *repository* akan dieksekusi oleh *jenkins*, apabila kontainer berhasil dibentuk, kemudian akan tersebar (*deployed*) ke *docker hub* yang ditunjukkan pada gambar di atas dan secara otomatis akan diupdate ke Linux dengan melakukan pengecekan *docker container* yang ada di Linux sudah terupdate. ■ Gambar 12 merupakan hasil tampilan dari rilis B.

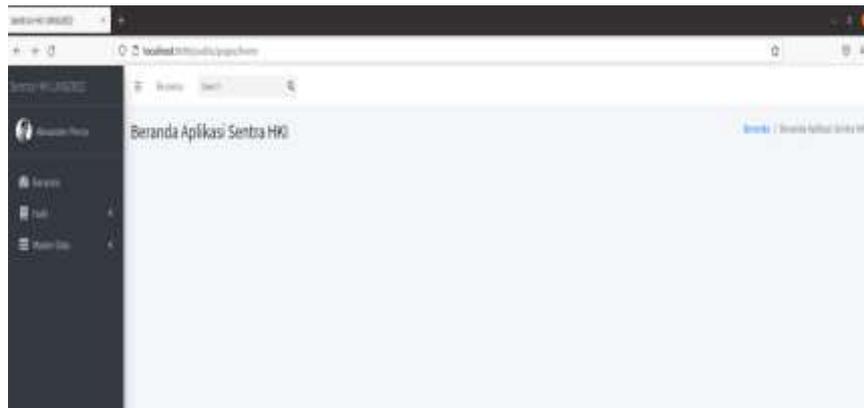
```
root@danis: /home/danis
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
a3a389408254   danis596/ta... "docker-php-entryp..." 25 seconds ago Up 23 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
d6e63c572354   gyouarobors/euroboras "ouraboras --repo-as..." 4 days ago    Up 5 hours    0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
root@danis: /home/danis# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
a3a389408254   danis596/ta... "docker-php-entryp..." 28 seconds ago Up 27 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
d6e63c572354   gyouarobors/euroboras "ouraboras --repo-as..." 4 days ago    Up 5 hours    0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
root@danis: /home/danis# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
a3a389408254   danis596/ta... "docker-php-entryp..." 29 seconds ago Up 28 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
d6e63c572354   gyouarobors/euroboras "ouraboras --repo-as..." 4 days ago    Up 5 hours    0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
root@danis: /home/danis# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
a3a389408254   danis596/ta... "docker-php-entryp..." 36 seconds ago Up 28 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
d6e63c572354   gyouarobors/euroboras "ouraboras --repo-as..." 4 days ago    Up 5 hours    0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
root@danis: /home/danis# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
a3a389408254   danis596/ta... "docker-php-entryp..." 31 seconds ago Up 29 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
d6e63c572354   gyouarobors/euroboras "ouraboras --repo-as..." 4 days ago    Up 5 hours    0.0.0.0:8080->80/tcp, :::8080->80/tcp   dazling_wing_euroboras
root@danis: /home/danis#
```

■ Gambar 11. Docker Container update bahwa sistem sudah bisa dijalankan

## Perbandingan Rilis A dan Rilis B

Perbedaan rilis A dan rilis B terdapat perbedaan yang signifikan. Pada rilis A hanya membuktikan dan menampilkan bahwa sistem *DevOps* berjalan dengan lancar tanpa ada *error*. Kemudian rilis A hanya ada program *php* dan *Dockerfile* sederhana yang menampilkan dari program tersebut. Sedangkan

pada rilis B terdapat program yang menggunakan *framework* codeigniter 4 yang berisi sistem informasi HKI. Kedua rilis tersebut berjalan lancar dan sesuai dengan harapan.



■ Gambar 12. Hasil dari rilis B

### Black Box Testing

*Black Box testing* adalah metode yang digunakan dalam pengujian yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun. Berdasarkan rencana pengujian, maka dapat dilakukan pengujian *black box* pada sistem informasi Pusat HKI Unsoed yang akan dijelaskan pada tabel di bawah ini.

■ Tabel 3. Hasil Pengujian black box

No.	Aktivitas Pengujian	Realisasi yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Masuk halaman awal	Menampilkan halaman awal sistem informasi HKI	Muncul beranda pada tampilan sistem informasi	[v] diterima [ ] ditolak
2.	Klik tombol "HaKi"	Menampilkan tampilan "Paten"	Muncul pilihan daftar dan input "Paten"	[v] diterima [ ] ditolak
3.	Klik tombol "HaKi"	Menampilkan tampilan "Cipta"	Muncul pilihan daftar dan input "Cipta"	[ ] diterima [v] ditolak
4.	Klik tombol "Master Data"	Menampilkan tampilan "Fasilitas"	Muncul tampilan "Fasilitas"	[v] diterima [ ] ditolak
5.	Klik tombol "Master Data"	Menampilkan tampilan "Civitas Akademika"	Muncul tampilan "Civitas Akademika"	[ ] diterima [v] ditolak
6.	Klik tombol "Master Data"	Menampilkan tampilan "Gelar Akademik"	Muncul tampilan "Gelar Akademik"	[ ] diterima [v] ditolak

Berdasarkan hasil pengujian *black box* menunjukkan bahwa sebesar 60% fitur pada sistem informasi dapat diakses. Pada pengujian *Blackbox testing* tersebut ditemukan beberapa fitur dari sistem informasi yang tidak bisa diakses, seperti tidak dapat memunculkan tampilan dari daftar cipta, input cipta, civitas akademika, dan gelar akademik. Pada tampilan sistem informasi yang bisa diakses yaitu beranda, daftar paten, input paten, dan fasilitas. Pada penelitian selanjutnya dapat dilakukan rekonfigurasi arsitekturnya untuk memperoleh kinerja yang lebih baik. Selain itu, penelitian selanjutnya juga dapat menggunakan *tools* yang lebih baik, seperti pemanfaatan teknologi *cloud*.

## KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasar pada penelitian yang dilakukan bahwa metode DevOps telah berhasil diterapkan pada Pusat HKI LPPM Unsoed dalam membangun infrastruktur untuk mengembangkan Sistem Informasi Pengelolaan Dokumen. Infrastruktur yang dibangun dengan menggunakan beberapa *tools* seperti Linux

Ubuntu 20.04, Github, Jenkins, Docker dan Apache Web Server, memiliki kinerja yang cukup baik. Hal ini, berdasarkan hasil dari pengujian *black box* menunjukkan bahwa sebesar 60% fitur pada sistem informasi dapat diakses.

#### B. Saran

Berdasarkan penelitian yang telah dilakukan terdapat hal yang masih dapat dikembangkan, yaitu dengan melakukan rekonfigurasi pada infrastruktur DevOps yang dibangun untuk mengantisipasi kegagalan akses pada beberapa fitur pada Sistem Informasi Pengelolaan Dokumen di Pusat HKI LPPM Unsoed.

#### DAFTAR PUSTAKA

- [1] J. Jogiyanto, "Analisis & desain : sistem informasi : pendekatan terstruktur teori dan praktik aplikasi bisnis Nasional RI.," *Andi Offset*, 2005.  
<https://opac.perpusnas.go.id/DetailOpac.aspx?id=918016> (accessed Dec. 14, 2022).
- [2] A. Fadli, G. Sugiyanto, and M. I. Zulfa, "Upaya Mereduksi Jumlah Kecelakaan Lalu Lintas Melalui Penggunaan Sistem Informasi Geografis," *Warta LPM*, vol. 23, no. 2, pp. 115–128, Apr. 2020, doi: 10.23917/WARTA.V23I2.9895.
- [3] E. Widyawati, A. Fadli, and M. S. Aliim, "Purwarupa Sistem Pakar Dengan Metode Forward Chaining Dan Certainty Factor Untuk Mendeteksi Penyakit Kanker Payudara," *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 1, no. 6, pp. 247–259, Jul. 2021, doi: 10.52436/1.jpti.53.
- [4] A. Fadli, M. Indana Zulfa, and Y. Ramadhani, "Perbandingan Unjuk Kerja Algoritme Klasifikasi Data Mining dalam Sistem Peringatan Dini Ketepatan Waktu Studi Mahasiswa," *Jurnal Teknologi dan Sistem Komputer*, vol. 6, no. 4, pp. 158–163, Oct. 2018, doi: 10.14710/JTSISKOM.6.4.2018.158-163.
- [5] I. P. Sari, I. H. Batubara, A.-K. Al-Khowarizmi, and P. P. Hariani, "Perancangan Sistem Informasi Pengelolaan Arsip Digital Berbasis Web untuk Mengatur Sistem Kearsipan di SMK Tri Karya," *Wahana Jurnal Pengabdian kepada Masyarakat*, vol. 1, no. 1, pp. 18–24, Jul. 2022, doi: 10.56211/wahana.v1i1.101.
- [6] A. Simangunsong and M. Informatika, "Sistem Informasi Pengarsipan Dokumen Berbasis Web," *Jurnal Mantik Penusa*, vol. 2, no. 1, pp. 11–19, Apr. 2018, Accessed: Dec. 14, 2022. [Online]. Available: <https://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/317>
- [7] A. Prana Utama Sembiring and K. F. Ndruru, "Perancangan Sistem Informasi Manajemen Dokumen (Studi Kasus STMIK Mikroskil)".
- [8] I. Meiyola Pradanthi, F. Erawantini, S. Farlinda, D. Setiawan Hendyca Putra, J. Kesehatan, and P. Negeri Jember, "Evaluasi Sistem Informasi Manajemen Dokumen Akreditasi (SISMADAK) Dengan Menggunakan Metode PIECES di Rumah Sakit Bhayangkara Lumajang," *J-REMI : Jurnal Rekam Medik dan Informasi Kesehatan*, vol. 2, no. 1, pp. 21–27, Dec. 2020, doi: 10.25047/j-remi.v2i1.2183.
- [9] T. Tohirin, S. F. Utami, S. R. Widiyanto, and W. Al Mauludyansah, "Implementasi DevOps Pada Pengembangan Aplikasi e-Skrining Covid-19," *MULTINETICS*, vol. 6, no. 1, pp. 15–20, May 2020, doi: 10.32722/MULTINETICS.V6I1.2764.
- [10] A. Adelia and Y. M. Djajalaksana, "PEMBUATAN SISTEM INFORMASI MANAJEMEN DOKUMEN FAKULTAS TEKNOLOGI INFORMASI," 2014.
- [11] M. A. Avila and D. Kurniadi, "Rancang Bangun Sistem Informasi E-Office pada Tata Usaha Fakultas Teknik Universitas Negeri Padang," *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, vol. 9, no. 1, p. 137, Mar. 2021, doi: 10.24036/voteteknika.v9i1.111285.

- [12] A. Taryana, A. Fadli, and S. R. Nurshiami, “Merancang Perangkat Lunak Sistem Penjaminan Mutu Internal (SPMI) Perguruan Tinggi yang Memiliki Daya Adaptasi Terhadap Perubahan Kebutuhan Pengguna secara Cepat dan Sering,” *JURNAL AI-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI*, vol. 5, no. 3, p. 121, Apr. 2020, doi: 10.36722/sst.v5i3.372.
- [13] A. Taryana, A. Fadli, and S. R. Nurshiami, “Merancang Perangkat Lunak Sistem Penjaminan Mutu Internal (SPMI) Perguruan Tinggi yang Memiliki Daya Adaptasi Terhadap Perubahan Kebutuhan Pengguna secara Cepat dan Sering,” *JURNAL AI-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI*, vol. 5, no. 3, pp. 121–129, Apr. 2020, doi: 10.36722/SST.V5I3.372.