CHAPTER I

INTRODUCTION

## **I.1 Background**

DevOps is the blending of the terms development and operations, meant to represent a collaborative or shared approach to the tasks performed by a company's application development and IT operations teams. The term DevOps is used in several ways. In its most broad meaning, DevOps is an operational philosophy that promotes better communication between these teams -- and others. In its most narrow interpretation, DevOps describes the adoption of automation and programmable software development and infrastructure deployment and maintenance. The term may also label a culture that strategically looks at the entire software delivery chain, overseeing shared services and promoting the use of new development tools and best practices.

## **I.2 Writing Objective**

The reason of this essay is to make the reader understand what is Jenkins and GitHub, know Jenkins and GitHub more deeply,function of Jenkins and GitHub.

## **I.3 Problem Domain**

This essay only explain about Jenkins & GitHub and any topic related with it.

## **I.4 Writing Methodology**

Writing method applied by writer is quite simple that is, collecting materials that will be discussed and analyzed and meticulously from official source, applying with learn and do method which very influential in research, and take conclusion from all reports obtained from the research.

1.4 Writing Methodology

Writing methodology that authors were using to make this paper is authors are using literature method . Authors got some articles to make easier the authors from electronic g media, especially internet.

* 1. **Writing Framework**

To facilitate the preparation of this report, the systematics is used as follows:

**CHAPTER I INTRODUCTION**

This chapter will discuss about the background, why this article was made, problem domain or what will be discussed in this article, writing objective, writing methodology, and writing structure.

**CHAPTER II BASIC THEORY**

This chapter will discuss about

**CHAPTER III ANALYSIS**

In chapter III we will discuss about our project java in GitHub and Jenkins

**CHAPTER IV SUMMARY**

This chapter give a conclusion and suggestion for the readers who read this assignment.

**CHAPTER II**

**BASIC THEORY**

**II.1 Continuous Integration Definition**

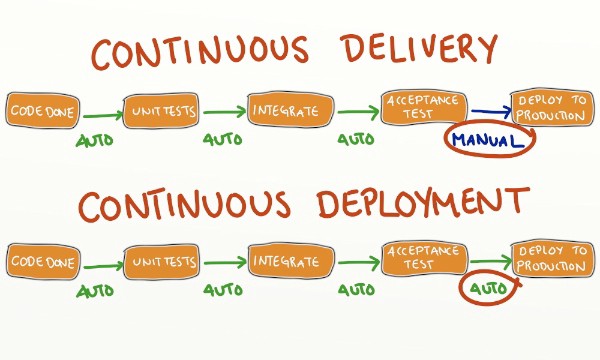


Figure 2.1 CI development ( https://medium.com/@edzob/ci-and-cd-in-the-wild-b5ca8f71fa28 )

Development teams use the continuous integration approach to software development because it allows them to develop in a quicker and more efficient manner than if they work separately for longer periods of time.

Developers using this approach submit changes to a source code repository, at which time the existing code base changes and new code can be added along with other resources that look for potential coding conflicts. There are also commercial CI servers that development teams can use to facilitate the CI process.

**II.2 Version Control**

A component of software configuration management, version control, also known as revision control or source control, is the management of changes to documents, computer programs, large web sites, and other collections of information. Changes are usually identified by a number or letter code, termed the "revision number", "revision level", or simply "revision". For example, an initial set of files is "revision 1". When the first change is made, the resulting set is "revision 2", and so on. Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and with some types of files, merged.

The need for a logical way to organize and control revisions has existed for almost as long as writing has existed, but revision control became much more important, and complicated when the era of computing began. The numbering of book editions and of specification revisions are examples that date back to the print-only era. Today, the most capable (as well as complex) revision control systems are those used in software development, where a team of people may concurrently make changes to the same files.

**II.3 Git**

Figure 2.2 git ( codepolitan,com )

Git is a version-control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source-code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision-control system, it is aimed at speed,data integrity, and support for distributed, non-linear workflows.

**II.4 GitHub**

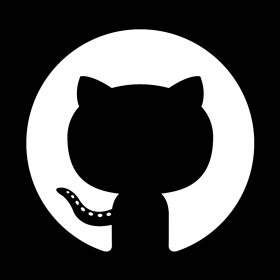


Figure 2.3 Github ( github.com)

Is a web-based [hosting service](https://en.wikipedia.org/wiki/Internet_hosting_service) for [version control](https://en.wikipedia.org/wiki/Version_control) using [Git](https://en.wikipedia.org/wiki/Git). It is mostly used for [computer code](https://en.wikipedia.org/wiki/Source_code). It offers all of the [distributed version control](https://en.wikipedia.org/wiki/Distributed_version_control) and [source code management](https://en.wikipedia.org/wiki/Source_code_management) (SCM) functionality of Git as well as adding its own features. It provides [access control](https://en.wikipedia.org/wiki/Access_control) and several collaboration features such as [bug tracking](https://en.wikipedia.org/wiki/Bug_tracking_system), [feature requests](https://en.wikipedia.org/wiki/Software_feature), [task management](https://en.wikipedia.org/wiki/Task_management), and [wikis](https://en.wikipedia.org/wiki/Wiki) for every project.

GitHub offers plans for both private repositories and free accounts which are commonly used to host [open-source](https://en.wikipedia.org/wiki/Open-source_software) software projects. As of June 2018, GitHub reports having over 28 million users and 57 million [repositories](https://en.wikipedia.org/wiki/Repository_(version_control)) (including 28 million public repositories), making it the largest host of [source code](https://en.wikipedia.org/wiki/Source_code) in the world.

**II.5 Jenkins**

****

Figure 2.4 Jenkins ( Jenkins.io)

Is an open source automation server written in Java. Jenkins helps to automate the non-human part of the software development process, with continuous integration and facilitating technical aspects of continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat. It supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, TD/OMS, ClearCase and RTC, and can execute Apache Ant, Apache Maven and sbt based projects as well as arbitrary shell scripts and Windows batch commands. The creator of Jenkins is Kohsuke Kawaguchi. Released under the MIT License, Jenkins is free software.

Builds can be triggered by various means, for example by commit in a version control system, by scheduling via a cron-like mechanism and by requesting a specific build URL. It can also be triggered after the other builds in the queue have completed. Jenkins functionality can be extended with plugins.