



# Jenkins Dasar

Eko Kurniawan Khannedy

# Eko Kurniawan Khannedy

- Technical architect at one of the biggest ecommerce company in Indonesia
- 11+ years experiences
- [www.programmerzamannow.com](http://www.programmerzamannow.com)
- [youtube.com/c/ProgrammerZamanNow](https://youtube.com/c/ProgrammerZamanNow)





# Eko Kurniawan Khannedy

- Telegram : [@khannedy](https://t.me/khannedy)
- Facebook : [fb.com/ProgrammerZamanNow](https://fb.com/ProgrammerZamanNow)
- Instagram : [instagram.com/programmerzamannow](https://instagram.com/programmerzamannow)
- Youtube : [youtube.com/c/ProgrammerZamanNow](https://youtube.com/c/ProgrammerZamanNow)
- Telegram Channel : [t.me/ProgrammerZamanNow](https://t.me/ProgrammerZamanNow)
- Email : [echo.khannedy@gmail.com](mailto:echo.khannedy@gmail.com)



# Sebelum Belajar

- Sudah Mengikuti Kelas Git dari Programmer Zaman Now
- Pernah Menggunakan Sistem Operasi Linux
- Pernah menggunakan VPS atau Cloud Compute



# Agenda

- Pengenalan Jenkins
- Menginstall Jenkins
- Job
- Integrasi dengan Git
- Credential
- Build Step
- Scheduled Job
- Dan lain-lain

---

# Pengenalan Automation Server



# Sebelum Automation Server

- Dalam pengembangan perangkat lunak, koding hanyalah tahapan awal saja
- Setelah koding, masih banyak hal yang biasanya dilakukan oleh pengembang perangkat lunak sebelum aplikasi yang dibuat sampai ke production server
- Sebelum adanya Automation Server, biasanya semuanya dilakukan secara manual, namun risikonya adalah sering terjadi kesalahan dan tidak konsisten
- Contoh menjalankan unit test, download dependency aplikasi, setup database, membuat distribusi aplikasi, dan lain-lain
- Hal-hal diatas sangat kompleks jika harus dilakukan secara manual terus menerus



# Pengenalan Automation Server

- Automation Server adalah aplikasi yang bisa kita gunakan untuk melakukan proses otomatisasi perintah yang kita instruksikan
- Ada banyak sekali keuntungan yang kita dapatkan jika menggunakan Automation Server, dibanding manual dalam tahapan pengembangan perangkat lunak
- Automation Server mempercepat proses, karena kita tidak perlu menunggu di Laptop kita.
- Automation Server bisa berjalan konsisten, di Laptop programmer mungkin perangkat lunak bisa berbeda-beda
- Automation Server bisa berjalan secara otomatis, bisa memberitahu kita ketika terjadi kesalahan, dan lain-lain





# Contoh Automation Server

- Jenkins <https://www.jenkins.io/>
- Team City <https://www.jetbrains.com/teamcity/>
- GitLab CI/CD <https://docs.gitlab.com/ee/ci/>
- Github Action <https://github.com/features/actions>
- Atlassian Bamboo <https://www.atlassian.com/software/bamboo>
- Circle CI <https://circleci.com/>
- Travis CI <https://travis-ci.org/>
- Dan lain-lain

---

# Pengenalan Jenkins



# Pengenalan Jenkins

- Jenkins adalah Open Source Automation Server yang paling populer saat ini
- Sebelum Jenkins, awalnya project ini bernama Hudson
- Hudson dibuat menggunakan Java, dan pengembang inti dari Hudson kebanyakan bekerja di Sun Microsystems (pemiliki Java)
- Tahun 2011, terjadi sengketa setelah pembelian Sun Microsystems oleh Oracle, dan karena trademark Hudson dipegang oleh Oracle, akhirnya pengembang inti dari Hudson membuat project baru bernama Jenkins dari source code Hudson
- Hudson sendiri akhirnya di donasikan ke Eclipse Foundation tahun 2012 dan akhirnya pengembangannya berhenti tahun 2017
- <https://www.jenkins.io/>



# Jenkins Plugin

- Jenkins sendiri merupakan sebuah platform untuk Automation Server
- Kebanyakan fitur di Jenkins sendiri tersedia dalam bentuk plugin, bahkan kita bisa membuat plugin sendiri jika kita mau
- <https://plugins.jenkins.io/>

---

# Menginstall Jenkins



# Hardware Requirements

## Minimum

- 256 MB of RAM
- 1 GB of Drive Space

## Recommended

- 4 GB+ of RAM
- 50 GB+ of Drive Space



# Software Requirements

- SSH
- Java 11 <http://jdk.java.net/archive/>
- Git <https://git-scm.com/>
- Web Browser



# Sistem Operasi Linux

- Pada kelas ini, kita akan menggunakan Linux sebagai sistem operasi yang menjalankan Jenkins nya
- Ini karena kebanyakan di perusahaan biasanya menggunakan Linux sebagai sistem operasi untuk Jenkins
- Tapi kita tetap bisa menggunakan sistem operasi apapun jika mau, karena Jenkins bisa berjalan di semua sistem operasi





# Setup Virtual Machine

- Pada kelas ini, kita akan setup dari awal menggunakan Virtual Machine
- Teman-teman bisa menggunakan VPS di penyedia layanan hosting, atau bisa menggunakan Virtual Machine Manager seperti VirtualBox di laptop, atau langsung menggunakan laptop sendiri jika menggunakan sistem operasi Linux



# Setup SSH

- Pastikan setup SSH terlebih dahulu di Laptop atau VM yang kita gunakan
- Gunakan perintah :  
ssh-keygen



# Menginstall Software Requirement

- Menginstall Java 11
- Menginstall Git



# Menginstall Jenkins

- Download War File Jenkins di <https://www.jenkins.io/download/>
- Jalankan Jenkins menggunakan perintah :  
`java -jar jenkins.war --httpPort=xxx`



# Jenkins Data



# Jenkins Data

- Jenkins tidak membutuhkan database untuk menyimpan seluruh konfigurasi dan datanya
- Jenkins akan secara otomatis membuat folder .jenkins di home directory sistem operasi kita
- Di folder tersebut lah, Jenkins akan menyimpan seluruh datanya
- Jika kita menghapus folder .jenkins, secara otomatis kita akan menghapus seluruh data Jenkins



# Membuat Job

- Job adalah sebutan untuk tugas automation di Jenkins
- Saat kita menginstall Jenkins, tidak ada satupun Job sama sekali
- Kita bisa membuat Job di Jenkins menggunakan web Jenkins

---

# Membuat Job



# Tampilan Membuat Job



Dashboard ▸ All ▸

Enter an item name

» Required field



**Freestyle project**

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system,

OK

---

# Menginstall Git Plugin



# Git Plugin

- Secara default, Jenkins tidak tahu bagaimana cara mengambil kode program yang terdapat di Git
- Kita perlu menginstall Git Plugin terlebih dahulu
- Setelah menginstall Git Plugin, maka secara otomatis kita bisa menambahkan sumber kode dari git repository di Job yang sudah kita buat



# Tampilan Git Plugin

## Plugin Manager

Updates

Available

Installed

Advanced

Install Name ↓

Git 4.11.3



git

Source Code Management

This plugin integrates [Git](#) with Jenkins.



# Setup SSH di Github

- Kita akan menggunakan Github sebagai Git Repository untuk menyimpan contoh project kita
- Oleh karena itu, kita perlu melakukan setup Github terlebih dahulu, agar Jenkins bisa mengambil kode program di Github
- Silahkan tambahkan SSH Public Key di VM Jenkins ke Github



# Tampilan Setup Git Key di Github



SSH

**Jenkins PZN**

SHA256:t1f2qSX3+7jFRBtFVCPNG8QrCE/eUC/7A2XkcfdwRfA

Added on May 23, 2022

Never used — Read/write

Delete

---

# Credentials



# Credentials

- Saat kita melakukan koneksi ke Git, biasanya kita perlu credential SSH
- Jenkins sendiri bisa menggunakan bawaan dari sistem operasi nya, namun hal ini tidak disarankan
- Hal ini karena kedepannya ketika kita setup beberapa server untuk Jenkins, kemungkinan credential SSH nya bisa berbeda
- Oleh karena itu, ada baiknya semua informasi credential disimpan di Jenkins
- Jenkins sendiri memiliki fitur bernama Credentials, dimana kita bisa menyimpan credential, salah satunya credential untuk SSH






# Tampilan Credential

▸ Global credentials (unrestricted) ▸



## Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name
	<a href="#">8887bec9-93ef-4c46-bdac-0b4fe8741e9b</a> khannedy

Icon: **S** **M** **L**

---

# Integrasi dengan Git



# Integrasi dengan Git

- Setelah kita setup Git dengan Github, dan juga menambahkan Credential SSH ke Jenkins, sekarang kita bisa mulai integrasi Jenkins dengan Git di Github
- Namun sebelumnya, yang perlu kita lakukan adalah, pastikan memiliki project di Github yang bisa digunakan sebagai sumber kode untuk Jenkins Job yang sudah kita buat



# Contoh Project

- <https://github.com/ProgrammerZamanNow/belajar-spring-dasar>



# Konfigurasi Git di Job

## Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

git@github.com:ProgrammerZamanNow/belajar-spring-dasar.git

Credentials ?

khannedy ▾

Add ▾

Advanced...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

main

---

# Build Step



# Build Step

- Source Code Management menggunakan Git, hanyalah awal dari tahapan Job, biasanya Job itu akan menjalankan perintah-perintah yang diperlukan, atau dalam Jenkins dinamakan Build Step
- Di Jenkins, kita bisa melakukan konfigurasi Job untuk menambahkan Build Step yang kita mau, bisa lebih dari satu



# Menambah Build Step

## Build



Execute shell



Command

```
./mvnw clean
```

See [the list of available environment variables](#)



Execute shell



Command

```
./mvnw compile test-compile
```



---

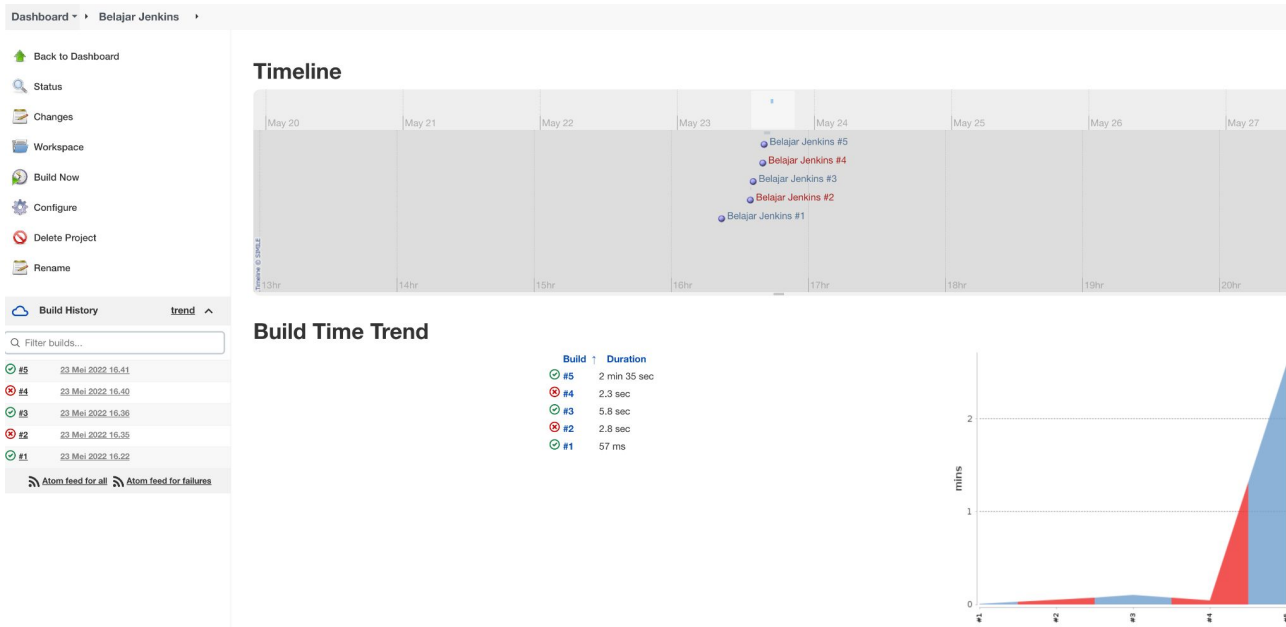
# Build History



# Build History

- Jenkins menyimpan seluruh riwayat build yang kita lakukan, baik itu di sukses ataupun gagal
- Hal ini memungkinkan kita bisa melihat hasil build kita yang lama, sehingga bisa mempermudah kita ketika proses debugging
- Selain itu, Jenkins juga memiliki fitur trend, dimana bisa memperlihatkan statistic Build History kita

# Tampilan Build History Trend



---

# Post Build Action



# Post Build Action

- Jenkins memiliki fitur bernama Post Build Action
- Fitur ini biasanya digunakan untuk memberitahu status sebuah Job ketika proses build selesai
- Misal, setelah sebuah Job selesai, kita ingin mengirim statusnya ke Email, Chat atau yang lainnya
- Contohnya, misal kita ingin mengirim status Job ke Slack, salah satu platform untuk Chat yang lumayan populer di dunia kerja



# Slack Notification Plugin

## Plugin Manager

Updates

Available

Installed

Advanced

Install Name ↓

Slack Notification 608.v19e3b\_44b\_b\_9ff



slack

Build Notifiers

Integrates Jenkins with Slack, allows publishing build statuses, messages and files to Slack channels.



# Setup Slack Jenkins Integration

[Browse apps](#) > [Jenkins CI](#) > New configuration



## Jenkins CI

An open source continuous integration server.

Jenkins CI is a customizable continuous integration server with over 600 plugins, allowing you to configure it to meet your needs.

This integration will post build notifications to a channel in Slack.

### Post to Channel

Start by choosing a channel where Jenkins notifications will be posted.

#jenkins-notification



✔ New channel created

**Add Jenkins CI integration**



# Setup Jenkins Slack Credential

› Global credentials (unrestricted) ›

Kind

Secret text

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Secret

.....

ID ?

slack-token

Description ?





# Setup Jenkins Slack Configuration

## Slack

Workspace ?

carvello

Credential ?

slack-token ▾

 Add ▾

Default channel / member id ?

jenkins-notification

☐ Custom slack app bot user ?

Advanced...

Test Connection



# Konfigurasi Post Build Action

## Post-build Actions



### Slack Notifications

- ☒ Notify Build Start
- ☒ Notify Success
- ☒ Notify Aborted
- ☒ Notify Not Built
- ☒ Notify Unstable
- ☒ Notify Regression
- ☒ Notify Every Failure
- ☒ Notify Back To Normal



Advanced...

Add post-build action ▾

# Tampilan Contoh Slack Notification



**khannedy** 12:02 AM  
joined #jenkins-notification.



**khannedy** 12:02 AM  
set the channel description: Jenkins Notification



**khannedy** 12:05 AM  
added an integration to this channel: [jenkins](#)



**jenkins** APP 12:16 AM  
Slack/Jenkins plugin: you're all set on <http://188.166.242.253:8080/>

Belajar Jenkins - #6 Started by user Programmer Zaman Now ([Open](#))

12:18 Belajar Jenkins - #6 Success after 28 sec ([Open](#))

---

# Copy Job



# Copy Job

- Saat membuat Job, kadang banyak sekali konfigurasi yang sama antara Job yang satu dengan Job yang lain, misal Build Step sama atau Post-Build Action yang sama
- Jenkins memiliki fitur yang bisa digunakan untuk membuat Job baru dengan cara copy dari Job yang sudah ada



# Tampilan Copy Job

Enter an item name

» Required field



## Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than building.

If you want to create a new item from other existing, you can use this option:



Copy from

OK

---

# Scheduled Job



## Scheduled Job

- Secara default Job di Jenkins hanya akan berjalan jika kita melakukan trigger build secara manual
- Kadang ada kasus dimana kita ingin menjalankan Jenkins Job dengan scheduled yang pasti, misal tiap jam 12 malam, atau bahkan tiap jam, tiap menit, dan sebagainya
- Contoh kasus misal, kita ingin menjalankan backup database tiap jam 12 malam misalnya





# Build Periodically

- Untuk membuat Scheduled Job, kita cukup tambahkan informasi scheduled dalam bentuk waktu Cron pada bagian Build Periodically
- Secara otomatis, Jenkins akan menjalankan Job tersebut sesuai dengan scheduled Cron
- <https://crontab.guru/>



# Tampilan Build Triggers

## Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☒ Build periodically ?

Schedule ?

\* \* \* \* \*

**⚠ Do you really mean "every minute" when you say "\* \* \* \* \*"? Perhaps you meant "H \* \* \* \* \*" to poll once per hour**

Would last have run at Monday, May 23, 2022 at 11:47:02 PM Coordinated Universal Time; would next run at Monday, May 24, 2022 at 11:47:02 PM Coordinated Universal Time

# Build History Scheduled Job



Build History

trend ^

🔍 Filter builds...

✓ #4 23 Mei 2022 23.51

✓ #3 23 Mei 2022 23.50

✓ #2 23 Mei 2022 23.49

✓ #1 23 Mei 2022 23.48

---

Poll SCM



# Poll SCM

- Saat menggunakan Scheduled Job menggunakan Cron, Job akan selalu di eksekusi ketika sudah waktunya sesuai dengan waktu Cron
- Kadang, ada kasus dimana kita ingin membuat Scheduled Job, namun hanya ingin menjalankan Job, ketika terjadi perubahan di Source Code saja
- Pada kasus ini, kita bisa menggunakan Poll SCM
- Poll SCM mirip dengan Scheduled Job, namun hanya akan menjalankan Job nya, ketika terjadi perubahan di Source Code, jika tidak ada perubahan, maka Job tidak akan dijalankan



# Tampilan Poll SCM

## Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) ?
  - ☐ Build after other projects are built ?
  - ☐ Build periodically ?
  - ☒ Poll SCM ?
- Schedule** ?

\* \* \* \* \*

**!** Do you really mean "every minute" when you say "\* \* \* \* \*"? Perhaps you meant "H \* \* \* \* \*" to poll once per hour

Would last have run at Monday, May 23, 2022 at 11:51:00 PM Coordinated Universal Time; would next run at Monday, May 24, 2022 at 11:51:00 PM Coordinated Universal Time

---

# Disabled Job



## Disabled Job

- Saat kita tidak butuh Job lagi, ada dua hal yang bisa kita lakukan
- Menghapus Job, artinya Job dan semua konfigurasi akan hilang secara permanen, atau
- Men-disabled Job, artinya Job tidak akan aktif, namun konfigurasi dan Job tetap ada, dan kita bisa mengaktifkan nya lagi jika kita mau





# Tampilan Disabled Job

General

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

## Description

[Plain text] [Preview](#)

- ☐ Discard old builds ?
- ☐ This project is parameterized ?
- ☒ Disable this project ?
- ☐ Execute concurrent builds if necessary ?

---

# Jenkins Environment Variable



# Jenkins Environment Variable

- Saat kita menggunakan Build Step berupa execute shell atau batch. Jenkins secara otomatis akan menambahkan environment variable yang berisikan informasi dari Job dan Jenkins nya itu sendiri
- Kadang pada beberapa kasus, kita sering butuh informasi dari Jenkins dan Job nya
- Kita bisa lihat semua env variable yang tersedia di Jenkins di halaman `JENKINS_URL/env-vars.html/`
- Untuk menggunakan env variable di shell atau batch, kita bisa gunakan tanda `${NAMA_ENV_VAR}`



# Contoh Menggunakan Env Variable

## Build



Execute shell

?

Command

```
echo "Execute Job ${JOB_NAME}:${BUILD_NUMBER}";
```

See [the list of available environment variables](#)

---

# Global Environment Variable



# Global Environment Variable

- Selain menggunakan env variable yang disediakan oleh Jenkins, kita juga bisa menambahkan environment variable tambahan ke Jenkins
- Environment variable yang kita tambahkan secara otomatis akan teregistrasi secara global, dan bisa digunakan di semua Job
- Fitur ini sangat cocok ketika kita butuh data konfigurasi yang bisa diubah-ubah, tanpa harus mengubah Build Step setiap Job yang kita buat
- Untuk menambahkan environment variable, kita bisa menggunakan menu Manage Jenkins -> Configure System -> Global Properties -> Environment Variables



# Menambah Environment Variable

## Global properties

---

☒ Environment variables

List of variables ?

Name

PZN\_AUTHOR

Value

Eko Kurniawan Khannedy



# Menggunakan Environment Variable

## Build



### Execute shell

?

### Command

```
echo "Execute Job ${JOB_NAME}:${BUILD_NUMBER}";  
echo "Build with Love by ${PZN_AUTHOR}";
```

See [the list of available environment variables](#)



---

# Build Parameter



# Build Parameter

- Saat membuat Job, kadang ada kasus dimana kita ingin meminta input data dari user sebelum menjalankan Job tersebut
- Misal ketika kita ingin menjalankan Job untuk deploy aplikasi, kita bisa meminta input dari user ingin di deploy kemana misalnya
- Jenkins memiliki fitur Build Parameter, dimana kita bisa meminta input data dari user sebelum menjalankan Job nya
- Terdapat banyak sekali jenis input data, seperti boolean, string, choice, secret, dan lain-lain
- Semua input data yang diberikan oleh user, secara otomatis bisa kita ambil seperti environment variable di Build Step



# Menambah Build Parameter

☐ Discard old builds ?

☒ This project is parameterized ?

## Choice Parameter

?

Name ?

SCRIPT

Choices ?

clean  
compile  
test  
package



# Menggunakan Build Parameter

## Build



Execute shell

?

Command

```
./mvnw ${SCRIPT}
```

See [the list of available environment variables](#)



# Menjalankan Job

Dashboard ▶ Belajar Build Parameter ▶



Back to Dashboard



Status



Changes



Workspace



Build with Parameters



Configure

## Project Belajar Build Parameter

This build requires parameters:

### SCRIPT

clean ▼

Build

---

**Discard Old Build**



# Discard Old Build

- Jenkins menyimpan semua history build di folder .jenkins di HOME directory
- Secara default, Jenkins akan menyimpan semua history Job selamanya, hal ini tidak baik terutama jika sampai kita lupa, maka server Jenkins bisa kehabisan ruang penyimpanan
- Jenkins memiliki fitur penghapusan riwayat build yang lama secara otomatis, baik itu menggunakan hari atau jumlah build
- Ketika kita menambahkan aturan ini, secara otomatis Jenkins akan melakukan pengecekan ketika selesai menjalankan Job



# Tampilan Discard Old Builds

☒ Discard old builds ?

Strategy

Log Rotation

**Days to keep builds**

1

if not empty, build records are only kept up to this number of days

**Max # of builds to keep**

1

if not empty, only up to this number of build records are kept



---

# Build Executor



# Build Executor

- Jenkins menjalankan semua Job dalam Executor
- Secara default Jenkins memiliki 2 Executor yang digunakan untuk menjalankan Job
- Saat semua Executor sedang sibuk, maka Job yang akan dijalankan harus menunggu terlebih dahulu sampai Executor ada yang tidak sibuk
- Kita bisa mengubah jumlah Executor dari menu Manage Jenkins -> Configure Jenkins -> # of Executor
- Harap bijak menentukan jumlah Executor, karena semakin banyak proses Job yang berjalan secara paralel, semakin banyak pula resource CPU dan Memory yang akan dibutuhkan

---

# User Management



# User Management


- Saat pertama kita menginstall Jenkins, kita telah membuat User untuk login ke web Jenkins
- Jika kita ingin menambah user baru, kita bisa menggunakan menu Manage Jenkins -> Manage Users -> Create User




# Anonymous Read Only


- Secara default, hanya user yang sudah login yang bisa melihat semua Job di Jenkins
- Pada kasus misal ketika kita menggunakan Jenkins untuk project Open Source, ada baiknya kita membolehkan akses untuk Read Only ke Guest User
- Kita bisa mengaktifkan fitur ini di menu Manage Jenkins -> Configure Global Security -> Authorization -> Logged-in -> Allow Anonymous Read Access


# Tampilan Guest User


 **Jenkins**


Dashboard ▸

 People

 Build History

 Project Relationship

 Check File Fingerprint

 Credentials

**Build Queue** ^

No builds in the queue.

**Build Executor Status** ^

1 Idle

2 Idle

All

S	W	Name ↓	Last Success	Last Failure
✓	⚙️	Belajar Build Parameter	1 hr 12 min <a href="#">#2</a>	N/A
⌚	⚙️	Belajar Copy Job	1 hr 46 min <a href="#">#34</a>	N/A
✓	⚙️	Belajar Environment Variable	1 hr 10 min <a href="#">#3</a>	N/A
✓	☁️	Belajar Jenkins	8 hr 50 min <a href="#">#6</a>	9 hr 27 min <a href="#">#4</a>
✓	⚙️	Build Remotely	1 hr 53 min <a href="#">#2</a>	N/A
✓	⚙️	Poll SCM	2 hr 3 min <a href="#">#2</a>	N/A



# Role Based Authorization Strategy

- Secara default, saat kita membuat user di Jenkins, user tersebut bisa melakukan apapun layaknya admin user
- Kadang kita ingin membatasi akses untuk user, sayangnya secara default di Jenkins kita tidak bisa melakukan ini
- Namun terdapat plugin Role Based Authorization Strategy yang bisa kita gunakan dalam kasus ini
- <https://plugins.jenkins.io/role-strategy/>



# Menginstall Role-based Authorization Strategy

## Plugin Manager

Updates

Available

Installed

Advanced

Install Name ↓

Role-based Authorization Strategy 484.v8a\_a\_e4b\_d785fd

Security

Authentication and User Management



Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes :

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information





# Mengubah Authorization

## Authorization

---

- ☐ Anyone can do anything ?
- ☐ Legacy mode ?
- ☐ Logged-in users can do anything ?
- ☒ Role-Based Strategy ?

# Manage and Assign Roles



## Manage and Assign Roles



### Manage Roles

Manage Roles



### Assign Roles

Assign Roles



### Role Strategy Macros

Provides info about macro usage and available macros



# User Management Lainnya

- <https://plugins.jenkins.io/saml/>
- <https://plugins.jenkins.io/google-login/>
- <https://plugins.jenkins.io/github-oauth/>
- <https://plugins.jenkins.io/gitlab-oauth/>
- Dan lain-lain <https://plugins.jenkins.io/ui/search/?labels=user>

---

# Trigger Build Remotely



# Trigger Build Remotely

- Jenkins memiliki fitur dimana kita bisa meminta Jenkins untuk menjalankan Job secara Remote
- Remote dalam artinya tanpa harus membuka web Jenkins
- Fitur ini cocok sekali ketika misal kita membuat Bot, sehingga Bot tersebut bisa memerintah Jenkins untuk menjalankan Job yang kita minta
- Saat kita menggunakan fitur ini, kita perlu menggunakan Authentication Token, yaitu token yang perlu kita gunakan ketika kita akan menjalankan Job secara Remote



# Tampilan Trigger Build Remotely

## Build Triggers

- ☒ Trigger builds remotely (e.g., from scripts) ?

### Authentication Token

RAHASIA12345

Use the following URL to trigger build remotely: `JENKINS_URL/job/Build%20Remotely/build?token=TOKEN_NAME`

Optionally append `&cause=Cause+Text` to provide text that will be included in the recorded build cause.

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ Poll SCM ?

---

**View**



# View

- Saat Job sudah banyak, kadang kita ingin membuat grup yang berisikan kumpulan Job tertentu, di Jenkins fitur ini disebut View
- Kita bisa membuat View dari menu (+) yang terdapat di halaman Dashboard, lalu bisa memilih Job mana yang ingin kita tambahkan ke View tersebut





# Membuat View

## New view

Name

Group Belajar

Type



**List View**

Shows items in a simple list format. You can choose which jobs are to be displayed in which view.



**My View**

This view automatically displays all the jobs that the current user has an access to.

Create









# Hasil View

All

Group Belajar



S	W	Name ↓	Last Success
		Belajar Copy Job	2 hr 5 min <a href="#">#34</a>
		Belajar Environment Variable	1 hr 28 min <a href="#">#3</a>
		Belajar Jenkins	9 hr 8 min <a href="#">#6</a>



# Nodes



# Nodes

- Saat kita menggunakan Jenkins, rekomendasinya sebenarnya tidak menjalankan Jenkins single Node, apalagi jika Job nya sudah sangat banyak
- Jenkins mendukung Nodes atau Agent, dimana kita bisa menambahkan Node Jenkins lainnya sehingga bisa digunakan untuk menjalankan Job
- Dengan menambahkan Node Jenkins, kita tidak perlu melakukan setup Job secara manual satu persatu di semua Node Jenkins, karena secara otomatis Jenkins akan mendistribusikan eksekusi Job nya ke semua Node
- Untuk menambah Node baru, kita bisa menggunakan menu Manage Jenkins -> Manage Nodes and Clouds -> New Node



# SSH Build Agent

- Saat kita menambah Node, terdapat banyak pilihan untuk terkoneksi dari Node master ke Node agent
- Salah satu yang aman adalah menggunakan SSH, namun secara default plugin nya tidak terinstall
- Jadi kita perlu menginstall plugin SSH Build Agent
- <https://plugins.jenkins.io/ssh-slaves/>



# Menginstall SSH Build Agent

## Plugin Manager

Updates

Available

Installed

Advanced

Install Name ↓

SSH Build Agents 1.814.vc82988f54b\_10



Agent Management

Allows to launch agents over SSH, using a Java implementation of the SSH protocol.

Install without restart

Download now and install after restart

Update information obtained: 11 hr ago

Check now



# Menginstall VM Baru

- Sebelum menambah Node, silahkan install VM Baru, dan jangan lupa untuk menginstall Software Requirement yang dibutuhkan, Git dan Java 11



# Menambah Node Baru

## New node

Node name

Jenkins Agent 1

Type



**Permanent Agent**

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins.

Create





# Setting SSH Agent

Launch method 

Launch agents via SSH

Host 

157.245.194.249

Credentials 

root/\*\*\*\*\*

 Add 











Host Key Verification Strategy 

Non verifying Verification Strategy



# Agent Log

Dashboard ▸ Nodes ▸ Jenkins Agent 1

-  Back to List
-  Status
-  Delete Agent
-  Configure
-  Build History
-  Load Statistics
-  Script Console
-  Log
-  System Information
-  Disconnect

```
SSHLauncher{host='157.245.194.249', port=22, credentialsId='ffcdf6eb-b6b1-4f87-91b5-60d8325c78bc', jvmOptions='', javaPath='', launchTimeoutSeconds=60, maxNumRetries=10, retryWaitTime=15, sshHostKeyVerificationStrategy=hudson.plugins.sshslaves.verifiers.trackCredentials=true}
[05/24/22 04:26:30] [SSH] Opening SSH connection to 157.245.194.249:22.
[05/24/22 04:26:30] [SSH] WARNING: SSH Host Keys are not being verified. Man-in-the-middle attacks may be possible against this
[05/24/22 04:26:31] [SSH] Authentication successful.
[05/24/22 04:26:31] [SSH] The remote user's environment is:
BASH=/usr/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:extquote:force_ignoresglobasciiranges:hostcomplete:interactive_comments:progcomp:verbose
BASH_ALIASES=()
BASH_ARGC=( [0]="0" )
BASH_ARGV=()
BASH_CMDS=()
BASH_EXECUTION_STRING=set
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=( [0]="5" [1]="0" [2]="17" [3]="1" [4]="release" [5]="x86_64-pc-linux-gnu" )
BASH_VERSION='5.0.17(1)-release'
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/0/bus
DIRSTACK=()
EUID=0
GROUPS=()
HOME=/root
```

# Agent Status

Dashboard ▸ Nodes ▸

 Back to Dashboard

 Manage Jenkins

 New Node

 Configure Clouds

 Node Monitoring

## Build Queue ^

No builds in the queue.

## Build Executor Status ^

### Built-In Node

1 Idle


2 Idle

### Jenkins Agent 1

1 Idle

2 Idle

## Manage nodes and clouds

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space
	<a href="#">Built-In Node</a>	Linux (amd64)	In sync	74.69 GB	 0 B	74.69 GB
	<a href="#">Jenkins Agent 1</a>	Linux (amd64)	In sync	75.17 GB	 0 B	75.17 GB
Data obtained		53 sec	53 sec	53 sec	52 sec	53 sec

---

# Materi Selanjutnya



# Materi Selanjutnya

- Jenkins Pipeline
- Jenkins Docker
- Jenkins Github
- Jenkins GitLab
- Jenkins Bitbucket
- Jenkins Kubernetes