AKADEMI
CLOUD CONNECT

KURSUS TREK KEPAKARAN: LATIHAN PROSES MODEN

- Hari Ketiga

Dibawakan kepada anda oleh:

cloudconnect

Pautan Latihan

https://code.cloud-connect.asia/Hanafiah/proses-moden

Agenda – Hari 03

Hari Ketiga	
Masa / Sesi	Topik
Sesi 1	Ringkasan Latihan Hari Kedua
Sesi 2	• Latihan 6: GitLab CI/CD
Sesi 3	• Latihan 7: Stress Test Laman Web
Sesi 4	• Latihan 8: Pengujian Keselamatan
Sesi 5	• Latihan 9: Pengujian Aplikasi menggunakan Cypress.io
Sesi 6	• Pengenalan kepada CD (Continuous Delivery / Deployment)
Sesi 7	• Scenario 1: Aplikasi API Node.js
Sesi 8	• Scenario 2: Aplikasi PHP Laravel
Penutup	
Sesi 9	• Rumusan Latihan
Sesi 10	• Penilaian Latihan

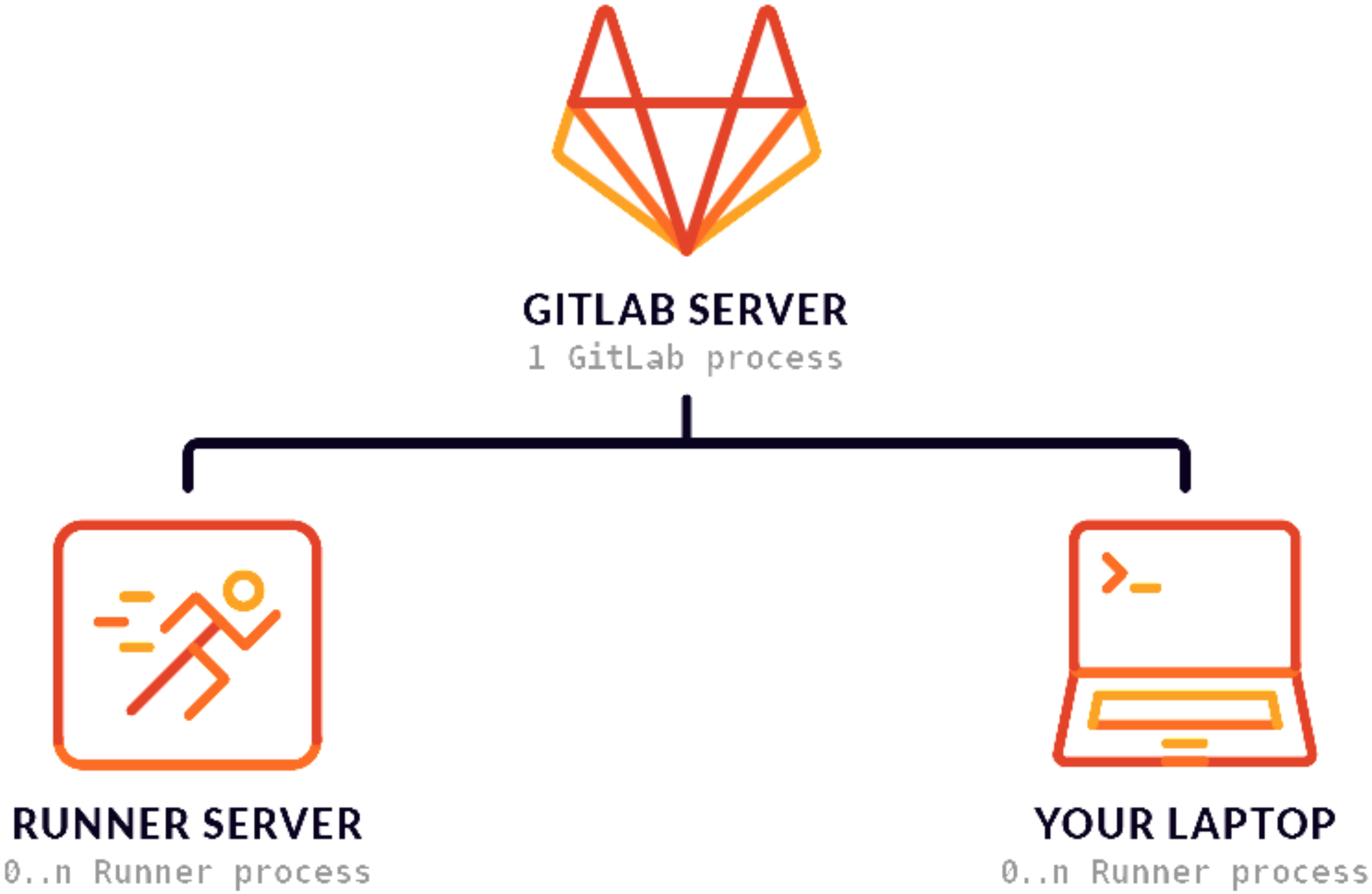
Ringkasan Hari 02

Hari Kedua	
Masa / Sesi	Topik
Sesi 1	Ringkasan Latihan Hari Pertama
Latihan Proses Moden	
Sesi 1	• Pengenalan kepada Git
Sesi 2	• Latihan 3: Pemasangan & Konfigurasi Git
Sesi 3	• Pengenalan kepada code repository
Sesi 4	• Latihan 4: Penggunaan Git & Code Repository
Sesi 5	• Git Flow & Branching
Sesi 6	• Latihan 5: Git Flow & Branching
Sesi 7	• Pengenalan kepada CI (Continuous Integration)
Sesi 8	• Latihan 6: GitLab CI/CD
Sesi 9	• Latihan 7: Stress Test Laman Web
Sesi 10	• Latihan 8: Pengujian Keselamatan

GitLab Ci/CD

- To use GitLab CI/CD:
  1. Runners available to run your jobs.
     - GitLab SaaS provides runners
     - Self-hosted GitLab – you need to install runner
  2. Create a .gitlab-ci.yml file at the root of your repository. This file is where you define your CI/CD jobs.

GitLab Runners



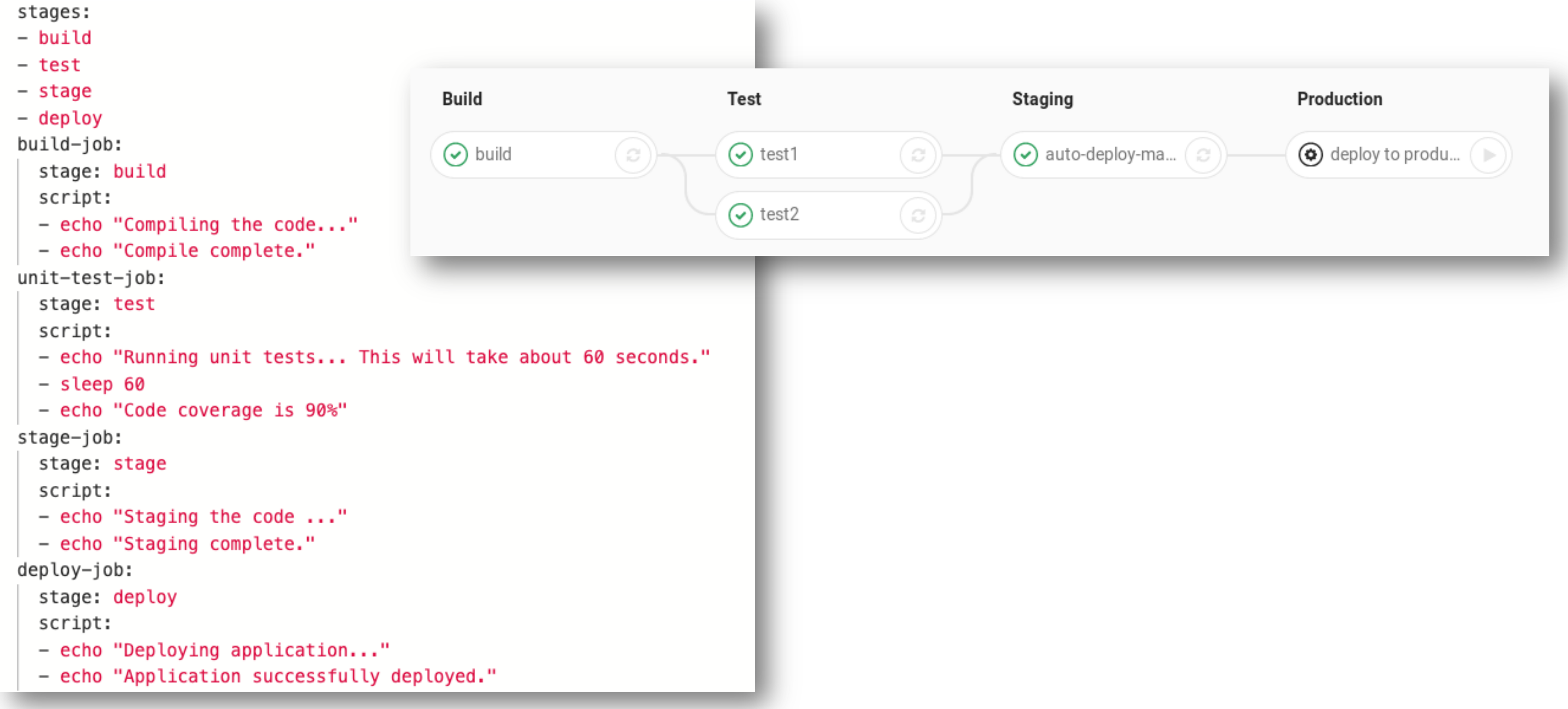
.gitlab-ci.yml file

- is a **YAML** file where you configure specific instructions for GitLab CI/CD:
  - The structure and order of jobs that the runner should execute.
  - The decisions the runner should make when specific conditions are encountered

CI/CD Pipelines

- Pipelines are the top-level component of continuous integration, delivery, and deployment.
- Pipelines comprise:
  - Jobs (executed by runners), which define what to do. For example, jobs that compile or test code.
  - Stages, which define when to run the jobs.
    - A **build** stage, with a job called compile.
    - A **test** stage, with two jobs called test1 and test2.
    - A **staging** stage, with a job called deploy-to-stage.
    - A **production** stage, with a job called deploy-to-prod.

CI/CD Pipelines



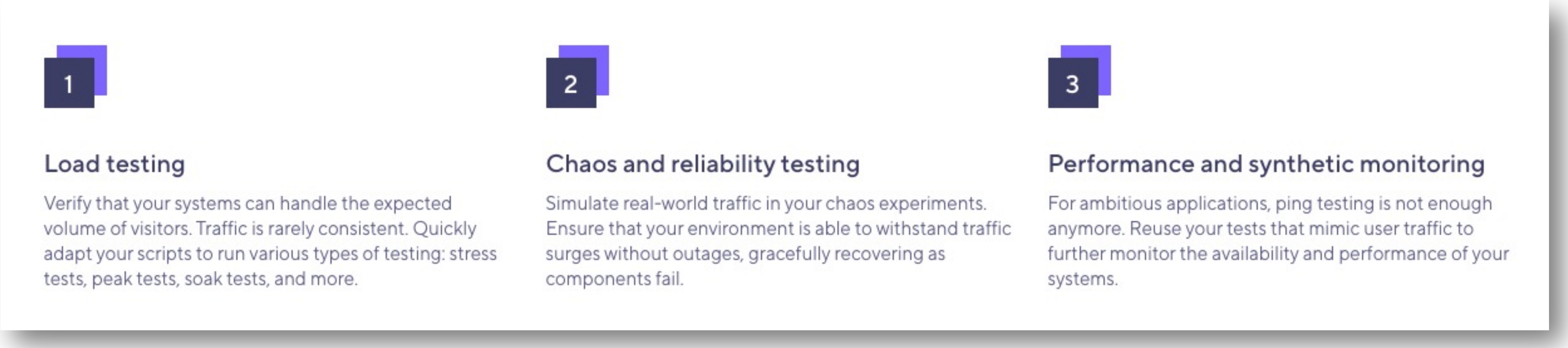
```
stages:
- build
- test
- stage
- deploy
build-job:
  stage: build
  script:
  - echo "Compiling the code..."
  - echo "Compile complete."
unit-test-job:
  stage: test
  script:
  - echo "Running unit tests... This will take about 60 seconds."
  - sleep 60
  - echo "Code coverage is 90%"
stage-job:
  stage: stage
  script:
  - echo "Staging the code ..."
  - echo "Staging complete."
deploy-job:
  stage: deploy
  script:
  - echo "Deploying application..."
  - echo "Application successfully deployed."
```

{ Latihan 6: *GitLab CI/CD* }

Performance Testing with K6

- *Performance Testing is the process of analyzing the quality and capability of a product.*
- *Load testing is a subset of performance testing where we analyze the behavior of the application under normal or peak load conditions.*
- ***k6** is an open-source load testing tool for testing the performance of APIs, microservices, and websites.*

K6 Features



1

Load testing

Verify that your systems can handle the expected volume of visitors. Traffic is rarely consistent. Quickly adapt your scripts to run various types of testing: stress tests, peak tests, soak tests, and more.

2

Chaos and reliability testing

Simulate real-world traffic in your chaos experiments. Ensure that your environment is able to withstand traffic surges without outages, gracefully recovering as components fail.

3

Performance and synthetic monitoring

For ambitious applications, ping testing is not enough anymore. Reuse your tests that mimic user traffic to further monitor the availability and performance of your systems.

{ Latihan 7: *Stress Test* Laman Web }

GitLab CI - Security

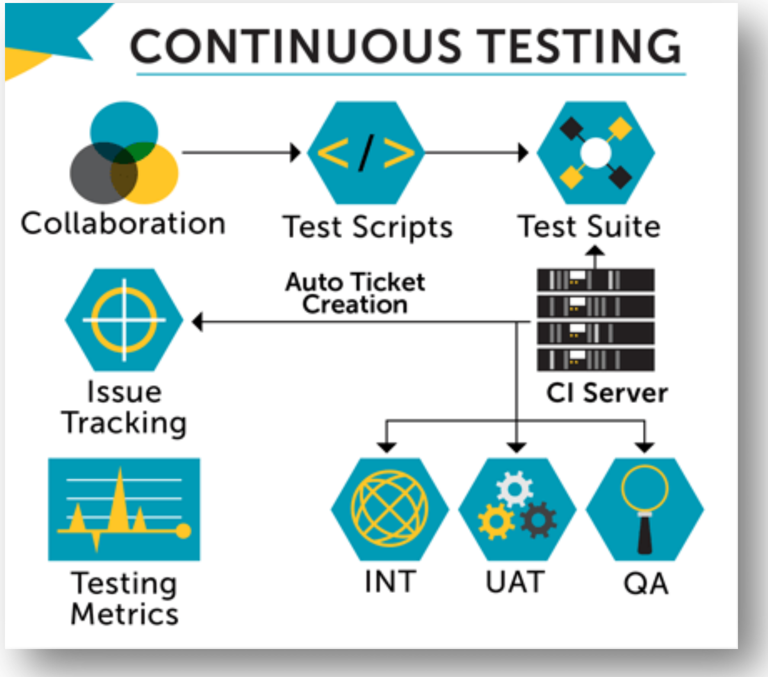
- Static Application Security Testing (SAST),
- Dynamic Application Security Testing (DAST),
- Container Scanning,
- Dependency Scanning,
- Code Quality,
- Secret Scanning,
- Fuzz Testing

{ Latihan 8: *Pengujian Keselamatan* }

{ Pengenalan kepada *Continuous Testing* }

Continuous Testing

- the process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release candidate as rapidly as possible.

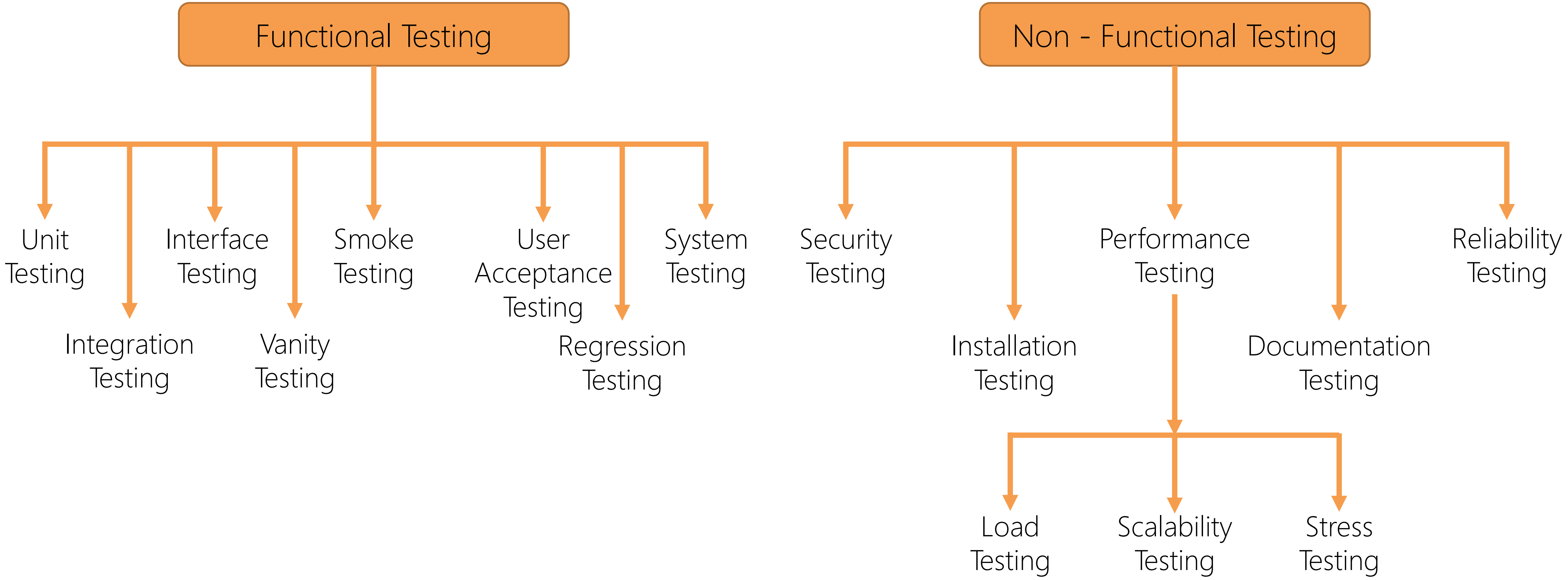


AKADEMI CLOUD CONNECT

cloudconnect

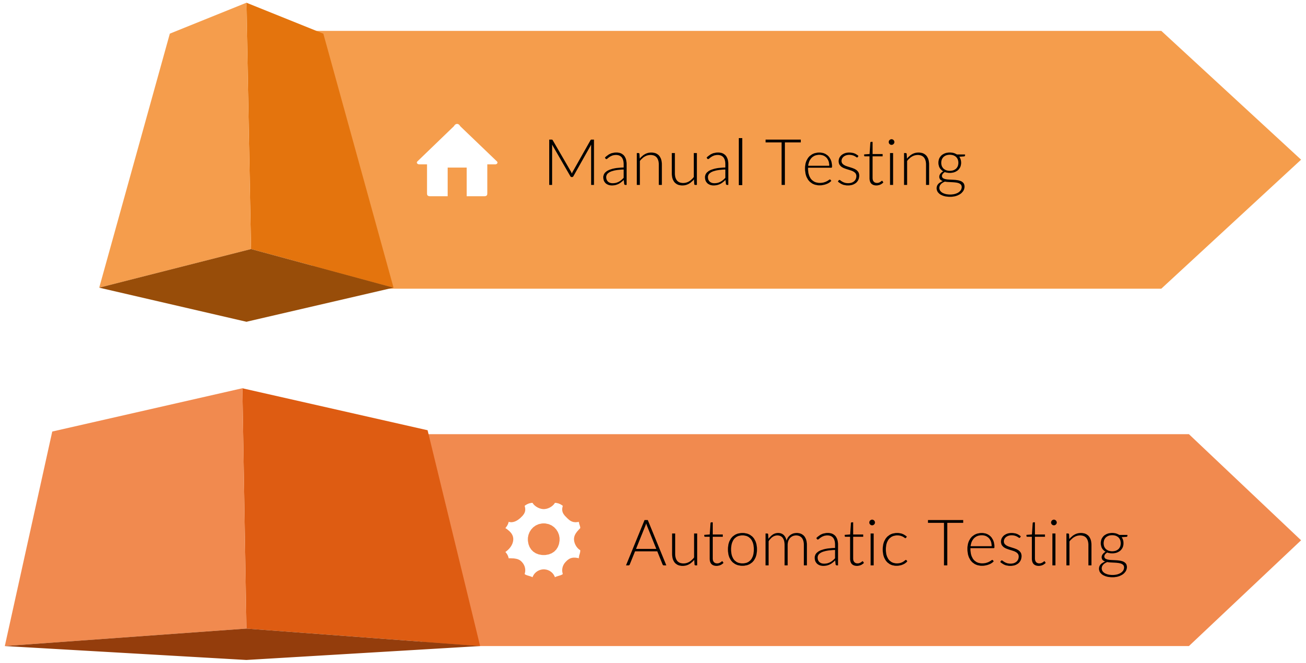
Continuous Testing

Software Testing Types



Continuous Testing

Testing Execution Approach



Manual Testing

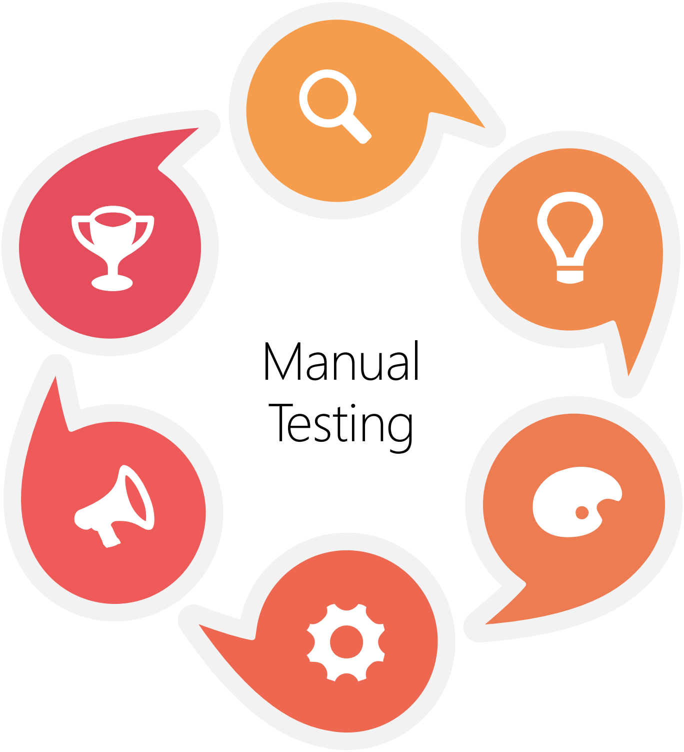
Automatic Testing

AKADEMI CLOUD CONNECT

cloudconnect

Continuous Testing

Manual Testing Process



◉ Step 01: Requirement Analysis

◉ Step 02: Test Plan Creation

◉ Step 03: Test Case Creation

◉ Step 04: Test Case Execution

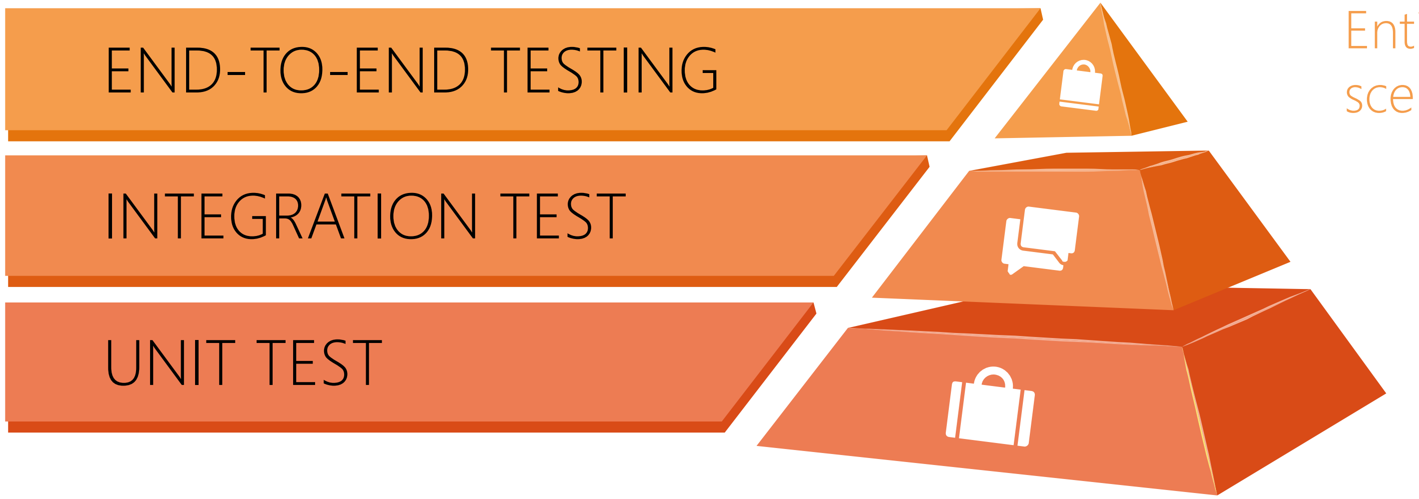
◉ Step 05: Defect Logging

◉ Step 06: Defect Fix

Automatic Testing using cypress.io

- What is cypress.io?
  - Cypress is a next generation front end testing tool built for the modern web applications.
- Type of testings:
  - End-to-end tests
  - Integration tests
  - Unit tests

cypress.io



END-TO-END TESTING

INTEGRATION TEST

UNIT TEST

Entire application is tested n a real-world scenario

Program units are combined and tested as groups in multiple ways

Testing code in isolated small pieces

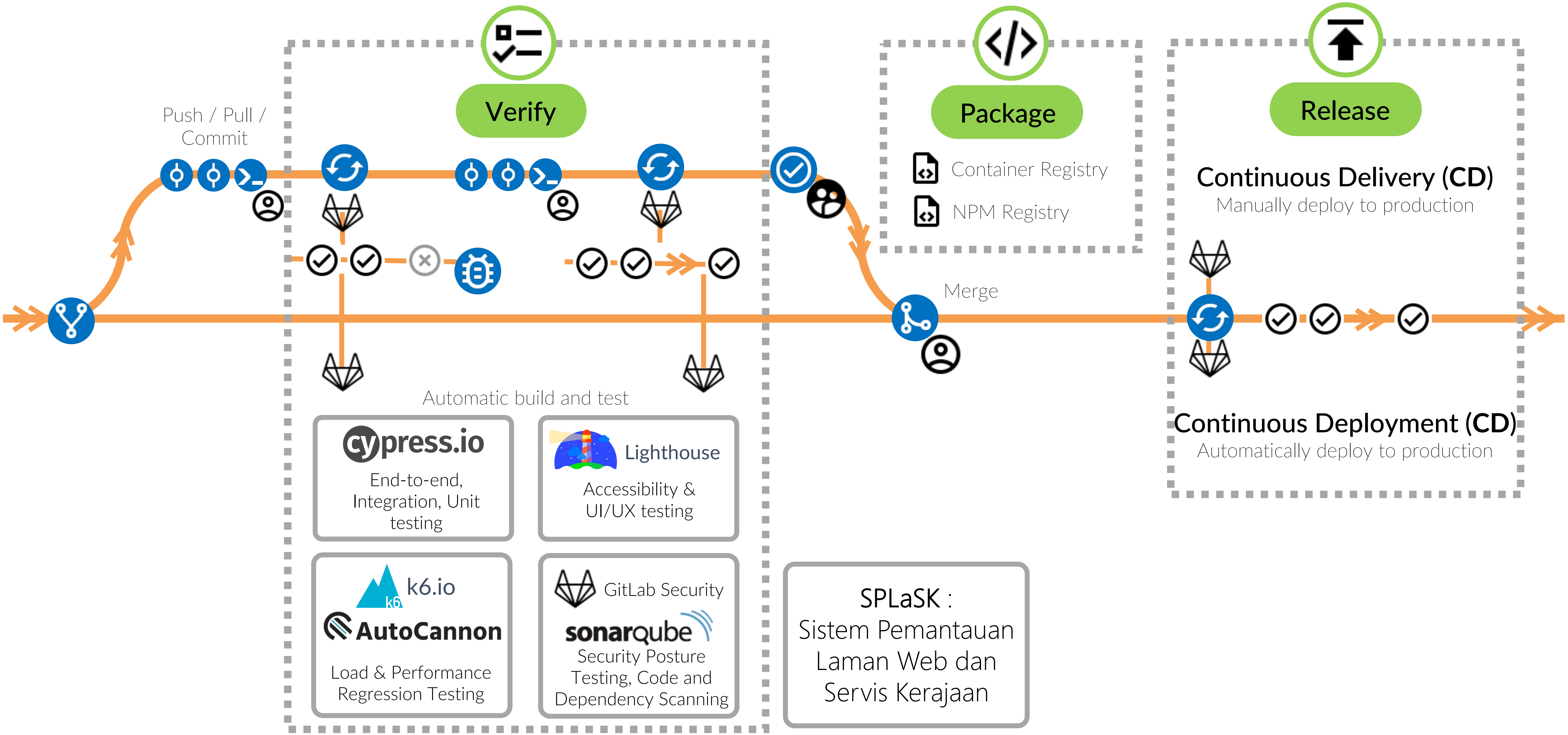
{ Latihan 9: Pengujian Aplikasi menggunakan *Cypress.io* }

{ Pengenalan kepada CD (*Continuous Delivery / Deployment*) }

AKADEMI CLOUD CONNECT

cloud connect

Modern Process using GitLab



Push / Pull / Commit

Verify

Automatic build and test

cypress.io

End-to-end, Integration, Unit testing

Lighthouse

Accessibility & UI/UX testing

k6.io

AutoCannon

Load & Performance Regression Testing

GitLab Security

sonarqube

Security Posture Testing, Code and Dependency Scanning

Package

Container Registry

NPM Registry

Merge

Release

Continuous Delivery (CD)

Manually deploy to production

Continuous Deployment (CD)

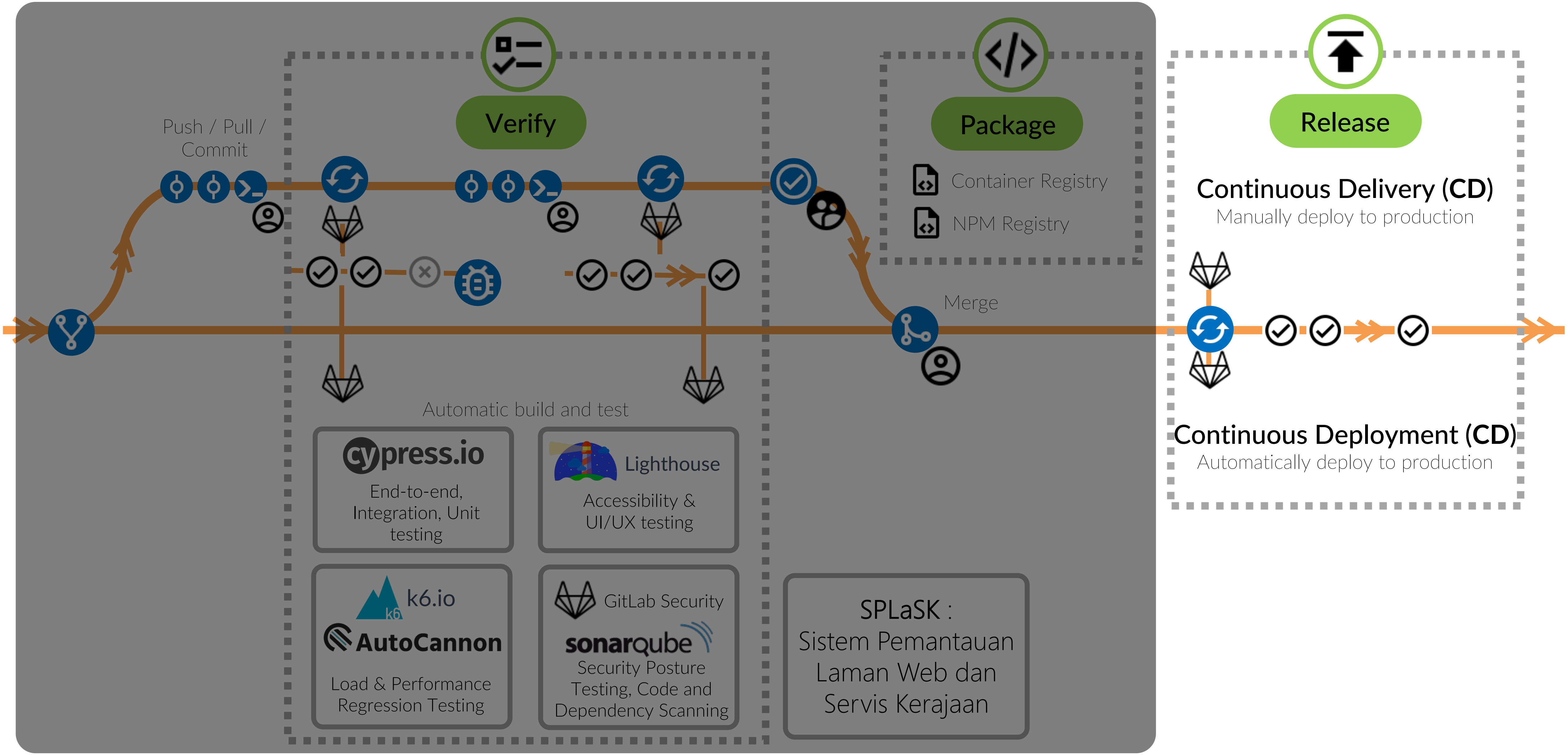
Automatically deploy to production

SPLaSK :
Sistem Pemantauan Laman Web dan Servis Kerajaan

AKADEMI CLOUD CONNECT

cloudconnect

Modern Process using GitLab



cypress.io
End-to-end, Integration, Unit testing

Lighthouse
Accessibility & UI/UX testing

k6.io
AutoCannon
Load & Performance Regression Testing

GitLab Security
sonarqube
Security Posture Testing, Code and Dependency Scanning

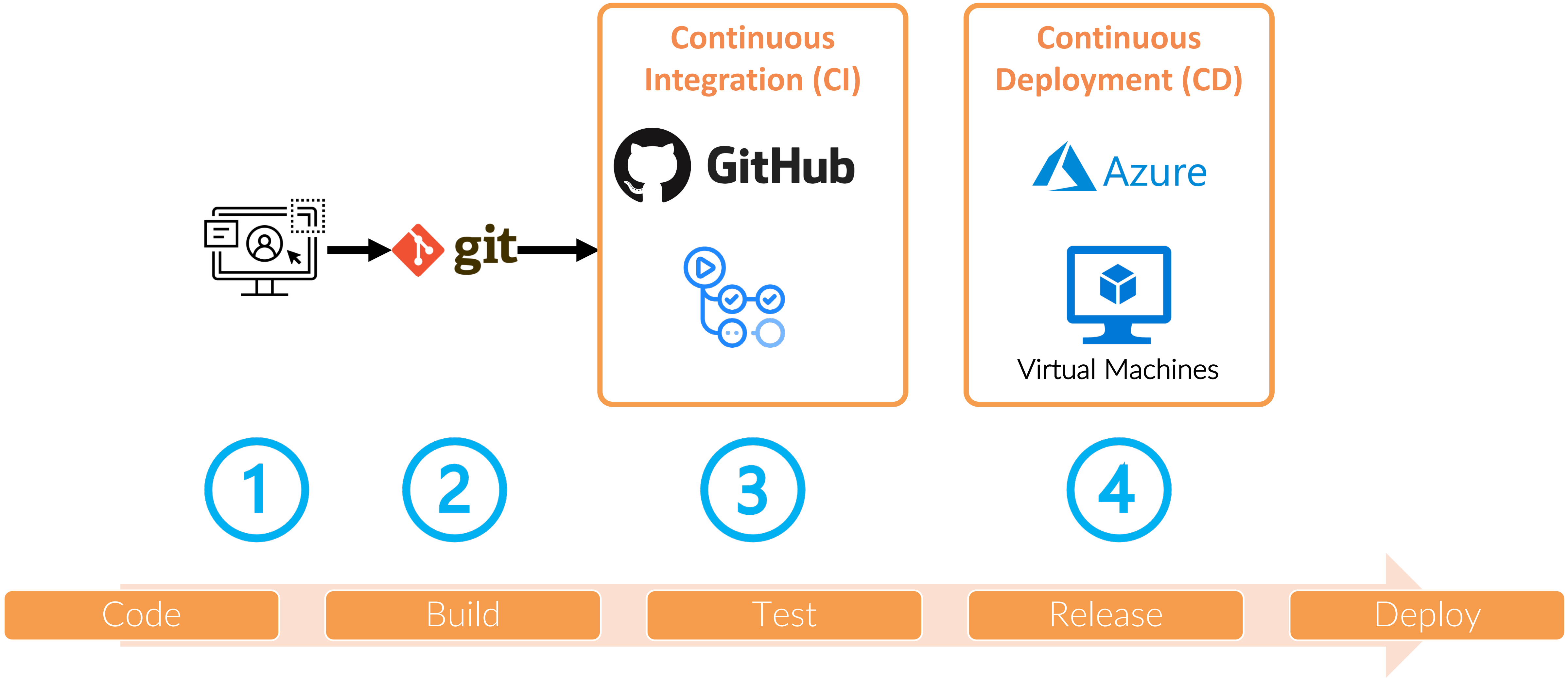
SPLaSK :
Sistem Pemantauan Laman Web dan Servis Kerajaan

Continuous Delivery (CD)
Manually deploy to production

Continuous Deployment (CD)
Automatically deploy to production

{ Latihan 9: *CI / CD* }

CD Demo – API App



git

Continuous Integration (CI)

GitHub

Continuous Deployment (CD)

Azure

Virtual Machines

1 2 3 4

Code

Build

Test

Release

Deploy

{ Latihan 9.1: *CI / CD*
Scenario 1: Aplikasi API Node.js }

{ Latihan 9.2: *CI / CD*
Scenario 2: Aplikasi PHP Laravel }

Tamat