

REFERENSI CEPAT GITLAB CI/CD

Pipeline, job, stage, variabel, artifact, environment

Dasar Pipeline

Cara Kerja Pipeline

- Pipeline** Container level atas; satu per commit/trigger
- Stage** Kelompok job yang berjalan secara paralel
- Job** Tugas tunggal (script) dalam sebuah stage
- Runner** Agen yang menjalankan job

Memicu Pipeline

- Push ke branch** Otomatis (default)
- Merge request** Via workflow:rules atau only: merge_requests
- Jadwal** CI/CD → Schedules di pengaturan proyek
- API** POST /projects/:id/trigger/pipeline
- Manual** Tombol Run Pipeline di menu CI/CD

.gitlab-ci.yml

```
stages: [build, test, deploy]
build-job:
  stage: build
  script: echo "Compiling..."
```

Kata Kunci Global

- stages** Definisikan urutan stage
- default** Nilai default untuk semua job
- variables** Variabel CI/CD global
- workflow** Kontrol kapan pipeline dibuat
- include** Impor file YAML eksternal

Include Template

```
include:
- template: Auto-DevOps.gitlab-ci.yml
- local: ./ci/lint.yml
- project: 'group/shared-ci'
  file: '/templates/deploy.yml'
```

Job

```
test-unit:
  stage: test
  image: node:20
  script:
  - npm ci
  - npm test
```

Kata Kunci Job

- script** Perintah shell yang dijalankan (wajib)
- before_script** Perintah sebelum script utama
- after_script** Perintah setelahnya (bahkan jika gagal)
- image** Image Docker untuk job
- rules** Kondisi penyertaan job
- needs** Dependensi DAG (lewati urutan stage)
- allow_failure** Pipeline berlanjut meski job gagal
- retry** Jumlah retry otomatis (0-2)
- timeout** Durasi job maksimum

Rules

```
deploy:
  rules:
  - if: '$CI_COMMIT_BRANCH == "main"'
    when: manual
  - if: '$CI_PIPELINE_SOURCE == "merge_request_event"'
    when: never
  - when: on_success
```

Stage

```
urutan stage
stages:
- lint
- build
- test
- deploy
```

Stage Default

- .pre** Selalu berjalan pertama
- build** Stage default ke-1
- test** Stage default ke-2
- deploy** Stage default ke-3
- .post** Selalu berjalan terakhir

DAG dengan needs

```
test-api:
  stage: test
  needs: ["build-api"] # skip waiting for full stage
test-web:
  stage: test
  needs: ["build-web"] # runs as soon as build-web done
```

Variabel

```
Mendefinisikan Variabel
variables:
  NODE_ENV: "production"
  DB_HOST: "postgres"
job:
  variables:
    NODE_ENV: "test" # job-level override
```

Variabel Bawaan

- CI_COMMIT_SHA** Hash commit lengkap
- CI_COMMIT_BRANCH** Nama branch
- CI_COMMIT_TAG** Nama tag (jika pipeline tag)
- CI_PIPELINE_ID** ID pipeline unik
- CI_PROJECT_DIR** Path checkout repo
- CI_MERGE_REQUEST_ID** Nomor MR (hanya pipeline MR)
- CI_REGISTRY_IMAGE** Path image container registry

Protected & Masked

- Protected** Hanya tersedia di branch/tag yang dilindungi
- Masked** Disembunyikan di log job
- File** Ditulis ke file sementara; path ada di variabel

Artifact

Menyimpan Artifact

```
build:
  script: npm run build
  artifacts:
    paths: [dist/]
    expire_in: 1 week
```

Type Artifact

- paths** File/direktori yang disimpan
- exclude** Pola yang dilewati
- expire_in** Hapus otomatis setelah durasi
- reports:junit** XML JUnit untuk ringkasan test MR
- reports:coverage_report** Visualisasi coverage Cobertura

Laporan JUnit

```
test:
  script: pytest --junitxml=report.xml
  artifacts:
    reports:
      junit: report.xml
```

Cache

Cache Dependensi

```
test:
  cache:
    key: ${CI_COMMIT_REF_SLUG}
    paths: [node_modules/]
  script: npm ci && npm test
```

Cache vs Artifact

- Cache** Percepat job; tidak dijamin; gunakan ulang dengan key yang sama
- Artifact** Teruskan file antar job/stage; dijamin

Kebijakan Cache

- pull-push** Unduh + unggah (default)
- pull** Unduh saja (lebih cepat untuk konsumen)
- push** Unggah saja (untuk produsen)

Environment

Mendefinisikan Environment

```
deploy-staging:
  stage: deploy
  environment:
    name: staging
    url: https://staging.example.com
  script: ./deploy.sh staging
```

Fitur Environment

- name** Nama environment (ditampilkan di UI)
- url** Link ke app yang di-deploy
- on_stop** Job yang dijalankan saat environment dihentikan
- auto_stop_in** Hentikan otomatis setelah durasi
- action: stop** Tandai job sebagai aksi stop

Review App

```
review:
  environment:
    name: review/${CI_COMMIT_REF_SLUG}
    url: https://${CI_COMMIT_REF_SLUG}.example.com
  on_stop: stop-review
  auto_stop_in: 1 week
```

Docker

Build & Push Image

```
build-image:
  image: docker:24
  services: [docker:24-dind]
  script:
  - docker login -u $CI_REGISTRY_USER -p $CI_REGISTRY_PASSWORD
  $CI_REGISTRY
  - docker build -t $CI_REGISTRY_IMAGE:$CI_COMMIT_SHA .
  - docker push $CI_REGISTRY_IMAGE:$CI_COMMIT_SHA
```

Service (Sidecar Container)

```
test:
  image: python:3.12
  services:
  - postgres:16
  - redis:7
  variables:
    POSTGRES_DB: testdb
    POSTGRES_PASSWORD: secret
```

Docker-in-Docker

- docker:24-dind** Image service Dind
- DOCKER_TLS_CERTDIR** Set ke "/certs" atau "" untuk konfigurasi TLS
- DOCKER_HOST** tcp://docker:2376 (TLS) atau :2375

Pola Umum

Monorepo (changes)

```
test-api:
  rules:
  - changes: [api/**/*]
test-web:
  rules:
  - changes: [web/**/*]
```

Gerbang Deploy Manual

```
deploy-prod:
  stage: deploy
  when: manual
  rules:
  - if: '$CI_COMMIT_BRANCH == "main"'
```

Matrix Paralel

```
test:
  parallel:
  matrix:
  - PYTHON: ["3.10", "3.11", "3.12"]
    DB: ["postgres", "sqlite"]
  script: tox -e py${PYTHON}-${DB}
```