

CI/CD

Perhatikan perintah-perintah berwarna merah di bawah, karena terkait pengumpulan jawaban. Anda dapat bekerja pada satu repositori tugas besar bersama, namun screenshot dan pengumpulan dilakukan secara individu.

Pendahuluan

Berdasarkan definisi dari GitHub¹, CI/CD didefinisikan sebagai:

- Continuous Integration: secara otomatis membangun, menguji, dan mengintegrasikan perubahan pada kode pada repositori bersama.
- Continuous Delivery: secara otomatis mengirimkan perubahan kode ke lingkungan *production* untuk proses persetujuan manual; atau
- Continuous Deployment: secara otomatis mengirimkan perubahan kode langsung ke *customer* (seluruhnya otomatis)

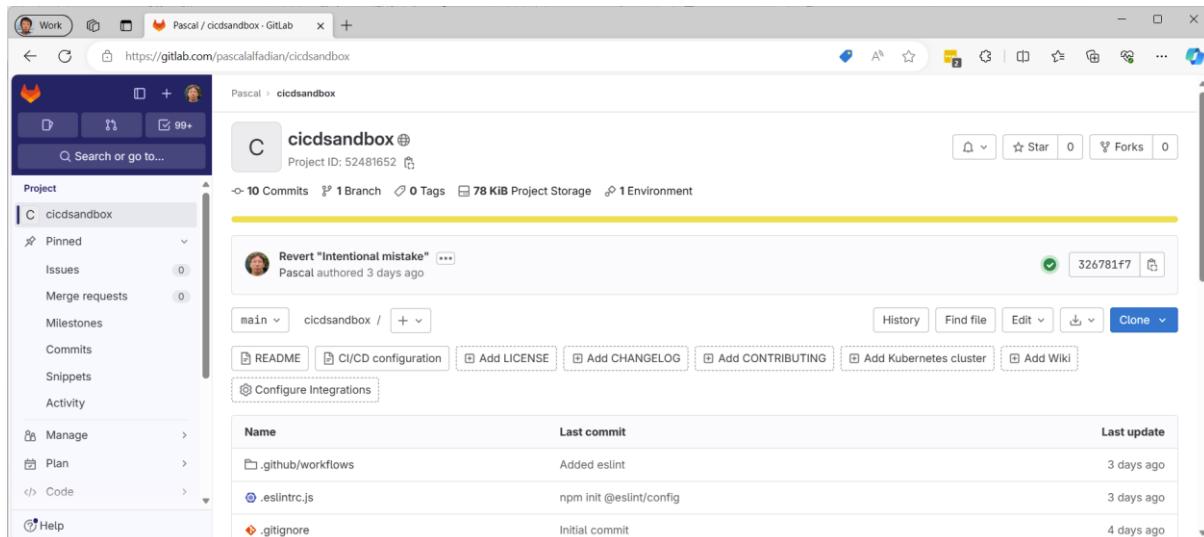
Seluruh proses CI/CD cukup kompleks dan membutuhkan pemahaman baik teknis yang digunakan maupun proses dalam organisasi. Walaupun demikian, semuanya berbasis pada hal sederhana: *proses ("action") yang otomatis dijalankan di server ("runner") di saat adanya rangsangan (trigger), biasanya berupa git push.*

Di modul ini kita akan belajar CI sederhana, yaitu mengintegrasikan git push dengan aksi memeriksa kebersihan kode program JavaScript Anda. Contoh pada GitLab dan GitHub diberikan di sini, silakan disesuaikan dengan tempat di mana Anda menaruh repositori Anda.

Jika Anda menemui kesulitan saat mengerjakan modul, silakan melihat referensi yang sudah saya siapkan di sini.

Referensi GitLab

Kode program dapat dilihat di <https://gitlab.com/pascalalfadian/cicdsandbox>, sedangkan berikut adalah contoh tangkapan layar repositori dengan CI yang berhasil, ditandai dengan simbol “cek” hijau.

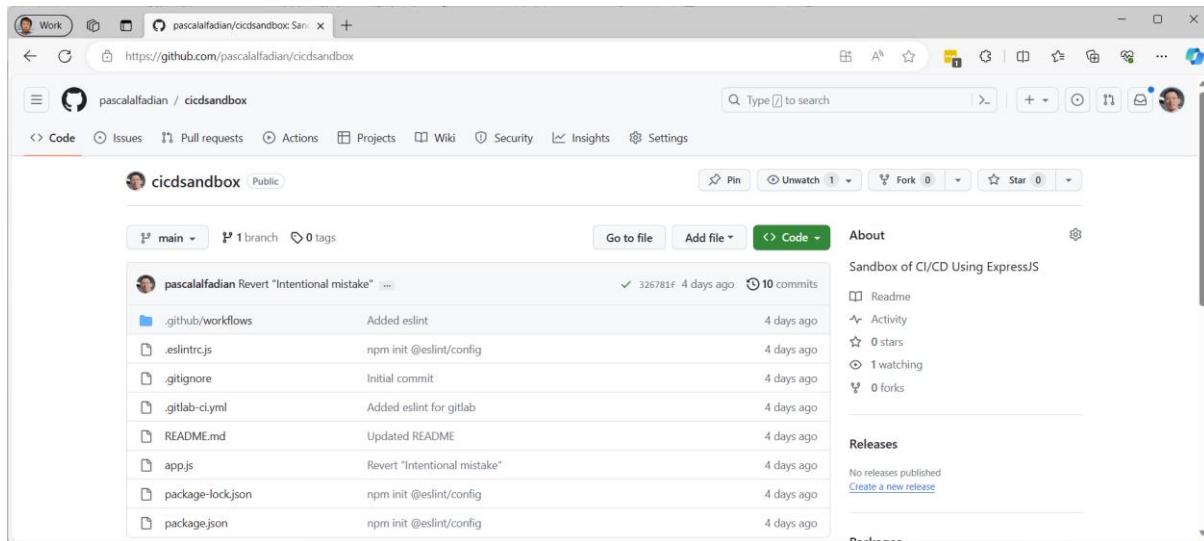


The screenshot shows a GitLab project named "cicdsandbox". The top navigation bar includes links for Work, Home, and the current project. The project summary indicates 10 Commits, 1 Branch, 0 Tags, 78 KiB Project Storage, and 1 Environment. A recent commit by Pascal is shown, with a green checkmark icon indicating successful CI status. The commit hash is 326781f7. Below the commit, there's a "Configure Integrations" section with several options like README, CI/CD configuration, Add LICENSE, Add CHANGELOG, Add CONTRIBUTING, Add Kubernetes cluster, and Add Wiki. The activity feed lists three commits: ".github/workflows" (Added eslint, 3 days ago), ".eslintrc.js" (npm init @eslint/config, 3 days ago), and ".gitignore" (Initial commit, 4 days ago). The sidebar on the left provides navigation links for Pinned, Issues, Merge requests, Milestones, Commits, Snippets, Activity, Manage, Plan, Code, and Help.

¹ <https://resources.github.com/ci-cd/>

Referensi GitHub

Kode program dapat dilihat di <https://github.com/pascalalfadian/cicdsandbox>, sedangkan berikut adalah contoh tangkapan layar CI yang berhasil, juga ditandai dengan simbol “cek” hijau.



Setup ESLint

Sebelum memasang CI di GitLab/GitHub, kita pahami dulu apa yang perlu dilakukan, dan kita lakukan dulu di komputer sendiri. ESLint² adalah sebuah program yang memeriksa kode program yang Anda buat serta melaporkan kesalahan (contoh: kurang titik koma) maupun “ketidak sempurnaan” (contoh: membaca variabel sebelum diinisialisasi) pada kode Anda.

Untuk melakukannya, masuk ke direktori proyek tugas besar Anda di komputer di mana terdapat file package.json, dan lakukan perintah ini (simbol “\$” menandakan shell dan tidak perlu diketik):

```
$ npm init @eslint/config
```

Anda akan diberikan beberapa pertanyaan, dan berikut adalah jawaban yang direkomendasikan untuk project Anda. Anda dapat menyesuaikannya jika Anda tahu apa yang Anda lakukan.

- How would you like to use ESLint? **To check syntax and find problems**
- What type of modules does your project use? **CommonJS (require(exports))**
- Which framework does your project use? **None of these**
- Does your project use TypeScript? **No**
- Where does your code run? **Node**
- What format do you want your config file to be in? **JavaScript**
- Would you like to install them (ESLint) now? **Yes**
- Which package manager do you want to use? **npm**

Setelah itu, jalankan perintah untuk melakukan lint pada setiap file .js yang Anda miliki di direktori tersebut. Jika file .js Anda berada di direktori berbeda, sesuaikan path nya di sini.

```
$ npx eslint *.js
```

² <https://eslint.org/>

Pastikan tidak ada error setelah menjalankan perintah tersebut. Jika ada error, silakan diperbaiki sesuai pesannya.

1. Screenshot jendela terminal / command prompt Anda yang menampilkan eksekusi dari eslint tanpa error dan kumpulkan dengan nama cicd_1.png.

Coba juga untuk membuat error secara sengaja. Misalnya, ubah kata kunci “const” menjadi “aconst” dan jalankan kembali perintah di atas. Hasilnya kurang lebih seperti di bawah ini:

```
npx eslint *.js

D:\workspace\RPL\AH-tubes_rpl\index.js
  3:8  error  Parsing error: Unexpected token app

✖ 1 problem (1 error, 0 warnings)
```

2. Screenshot jendela terminal / command prompt Anda yang menampilkan eksekusi dari eslint dengan error dan kumpulkan dengan nama cicd_2.png.

Jika Anda sudah berhasil menjalankan dengan ataupun tanpa error, artinya Anda sudah berhasil menyiapkan ESLint dengan baik. git add, git commit, dan git push perubahan Anda sebelum maju ke bagian berikutnya.

Setup CI/CD

Saat CI/CD tersetup dengan baik, Anda akan memanfaatkan *resource* dari *runner* GitLab/GitHub. Keduanya menyediakan gratis resource untuk *running time* tertentu setiap bulannya, cukup untuk mengerjakan modul ini. Baik di GitLab maupun GitHub, konfigurasi CI/CD menggunakan sintaks YAML. Sintaks ini sangat sederhana dan biasanya tidak membutuhkan tutorial tertentu. Namun, jika Anda membutuhkan referensi, bisa mengacu ke sini: <https://yaml.org/>.

Silakan memilih subbab di bawah ini, bergantung pada server repositori yang anda gunakan.

GitLab

Referensi: https://docs.gitlab.com/ee/ci/quick_start/

Pada GitLab, konfigurasi CI/CD dituliskan pada file .gitlab-ci.yml. Silakan ikuti langkah yang diberikan pada referensi di atas hingga berhasil, kemudian tambahkan baris-baris berikut di akhir file tersebut:

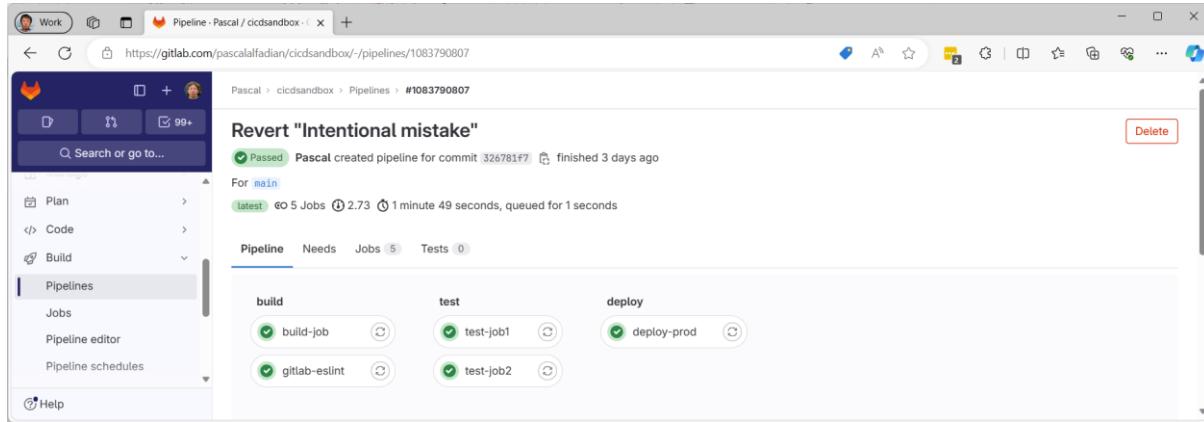
```
25 gitlab-eslint:
26   stage: build
27   image: node
28   script:
29     - npm install
30     - npx eslint *.js
```

Berikut penjelasan masing-masing baris:

25. Kita tambahkan job baru bernama “gitlab-eslint”
26. Job ini masuk ke dalam stage “build”, bersamaan dengan stage “build-job”

27. Job ini menggunakan docker image “node”³, di mana nodejs sudah disiapkan.
28. Kita definisikan skrip yang perlu dijalankan di baris-baris berikut.
29. Skrip pertama yang perlu dijalankan adalah “npm install”⁴. Skrip ini otomatis dijalankan di dalam direktori project Anda.
30. Skrip kedua “npx eslint *.js” melakukan lint terhadap file-file js Anda. Silakan disesuaikan dengan project Anda jika perlu.

Git push dan pantau hasilnya pada menu “Pipelines”. Jika berhasil, Anda akan mendapatkan tangkapan layar seperti di bawah⁵:



GitHub

Referensi: <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>

Pada GitHub, konfigurasi CI/CD dituliskan pada file-file yang berada di dalam direktori .github/workflows. Silakan ikuti langkah yang diberikan pada referensi di atas hingga berhasil, kemudian tambahkan file github-eslint.yml di direktori tersebut, dengan isi seperti berikut:

```

1 name: github-eslint
2 run-name: Running ESLint against source code
3 on: [push]
4 jobs:
5   run-eslint:
6     runs-on: ubuntu-latest
7     steps:
8       - uses: actions/checkout@v4
9       - uses: actions/setup-node@v3
10      with:
11        node-version: '20'
12      - run: npm install
13      - run: npx eslint *.js
14

```

Berikut penjelasan masing-masing baris:

1. Kita namakan workflow ini “github-eslint”.
2. Kita berikan deskripsi workflow ini.
3. Workflow ini akan dilakukan setiap ada aksi “git push”.
4. Kita definisikan job nya di baris-baris berikutnya.

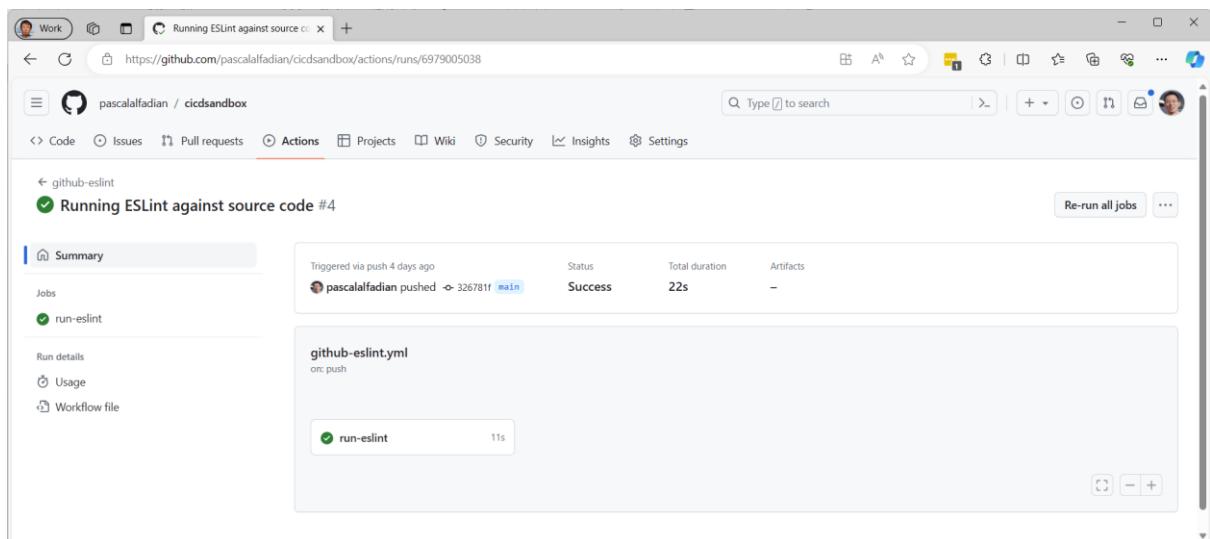
³ https://hub.docker.com/_/node/

⁴ Jika node_modules/ sudah masuk ke repositori Anda, langkah ini dapat diabaikan.

⁵ <https://gitlab.com/pascalalfadian/cicdsandbox/-/pipelines/1083790807>

5. Kita definisikan job “run-eslint” di baris-baris berikutnya.
6. Job ini menggunakan runner dengan sistem operasi dan versi “ubuntu-latest”⁶.
7. Kita definisikan langkah-langkah yang perlu dilakukan di baris-baris berikutnya.
8. Kita perintahkan runner untuk checkout dari repositori.
9. Kita perintahkan runner untuk setup node.
10. Parameter untuk setup node diberikan di baris berikutnya.
11. Parameter tersebut adalah menggunakan node versi 20.
12. Kita perintahkan untuk menjalankan “npm install”⁷.
13. Kita perintahkan untuk menjalankan “npx eslint *.js” untuk melakukan lint terhadap file-file js Anda. Silakan disesuaikan dengan project Anda jika perlu.

Git push dan pantau hasilnya pada menu “Actions”. Jika berhasil, Anda akan mendapatkan tangkapan layar seperti di bawah⁸:



Penutup

Setelah Anda berhasil memasang CI, kelompok Anda dapat mengerjakan project ini seperti biasa. Namun bedanya, setiap anggota kelompok yang melakukan git push, kebenaran dan kesempurnaan diperiksa oleh GitLab/GitHub. Jika ada hal yang tidak sesuai, maka cek hijau akan berubah menjadi X merah. Jika hal ini terjadi, seseorang pada kelompok Anda sebaiknya mencari tahu masalahnya dan memperbaikinya, sebelum (pada pekerjaan profesional sesungguhnya) masalah ini muncul di *production*.

Modul ini dirancang supaya Anda bisa menyelesaiannya di jam responsi, sehingga banyak hal yang tidak dicakup di sini. Silakan bereksplorasi dengan berbagai fitur-fitur yang ada di luar jam responsi.

3. Screenshot jendela browser yang menunjukkan bahwa job/workflow sudah berhasil dijalankan dan kumpulkan dengan nama cicd_3.png.

⁶ <https://docs.github.com/en/actions/using-workflows/workflow-syntax-for-github-actions#choosing-github-hosted-runners>

⁷ Jika node_modules/ sudah masuk ke repositori Anda, langkah ini dapat diabaikan.

⁸ <https://github.com/pascalalfadian/cicdsandbox/actions/runs/6979005038>