



Deployment & CI/CD

Gold - Chapter 8 - Topic 6

Selamat datang di **Chapter 8 Topik 6** online
course **Fullstack Web** dari Binar Academy!





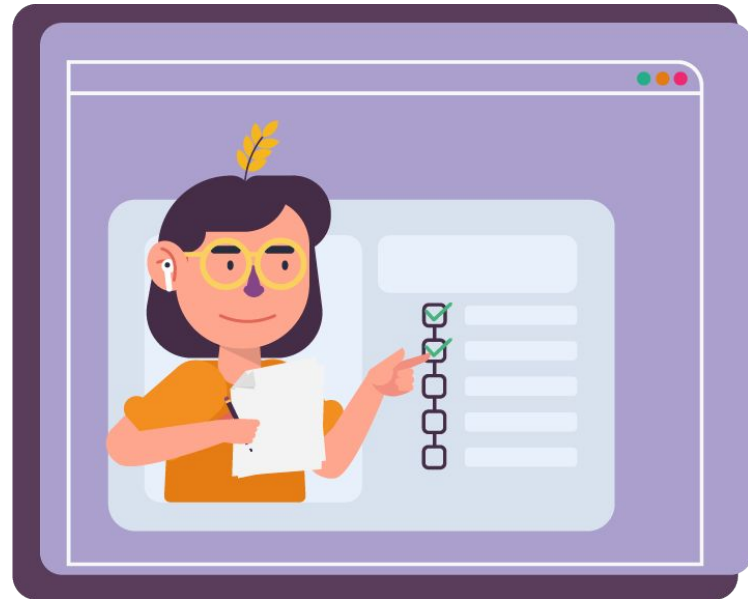
Nah, pada topik terakhir di chapter ini, kita bakal banyak bahas gimana caranya bisa melakukan **deployment dan CI/CD sehingga website yang selama ini susah-susah kita buat bisa digunakan dan dikonsumsi oleh banyak user**. Pasti makin gak sabar ya? Hehe... Yukkk kita lanjutttt ~





Detailnya, kita bakal bahas hal-hal berikut ini:

- Memahami konsep deployment dan cara melakukan deployment ke Heroku
- Memahami konsep CI/CD
- Memahami implementasi CI/CD





Nah, dari kemaren kamu udah belajar gimana caranya bikin website, tapi sayangunya belum bisa kamu pamerin ke ayang biar keliatan keren.

Di topik ini kamu bakal belajar gimana caranya melakukan **deployment** website kamu dan bisa dilihat sama ayangmu yang mungkin tinggal jauh darimu huhuhu.

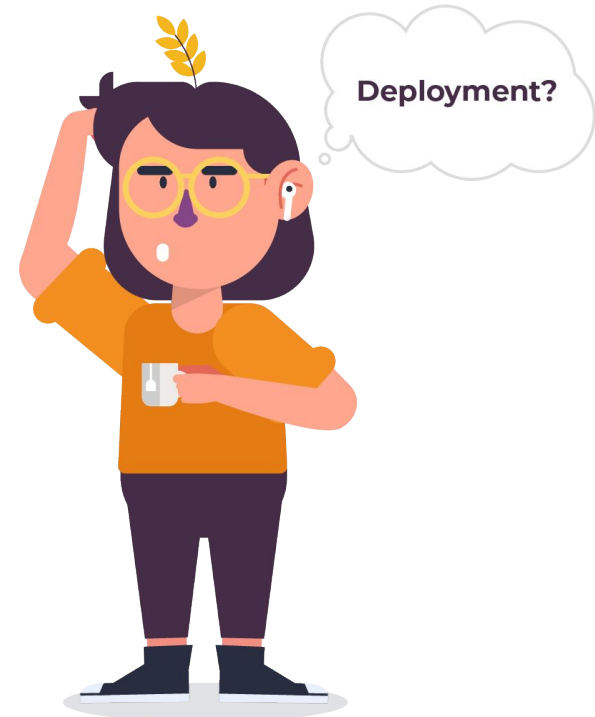




Mari kita bahas, apa sih Deployment itu?

Deployment adalah **tahapan kegiatan yang bertujuan untuk menyebarkan aplikasi yang telah dikerjakan oleh para developer pada suatu URL tertentu dalam server agar bisa dilihat untuk umum.**

Penyebarannya tergantung dari jenis aplikasinya, misal jika aplikasi web makan deployment di hosting Server, jika aplikasi Mobile maka deployment akan di PlayStore/AppStore.





Nahhh... berarti untuk Fullstack ini kita akan deploy di server ya. Hhmmm... apakah itu berarti kita harus mengeluarkan uang untuk beli server? Hmm.. kira-kira butuh berapa ya uangnya?

Eits... jangan khawatir! tunggu penjelasan berikutnya dulu. Di slide setelah ini!





Pernah denger website yang namanya Heroku? Nah **heroku ini adalah salah satu website yang menyediakan jasa hosting atau server secara gratis** lho.

Kamu bisa pake heroku buat deploy server back-end-mu sama front-end juga. Seru kan?

Nah, kamu bisa coba registrasi dulu nih ke heroku pake [link ini](#).





Ketika kamu udah register, kamu akan melihat tampilan kayak gini. Nah disini kamu bisa bikin 1 aplikasi, dengan nama terserah kamu, yang penting catchy hehehe.

Personal

New

binar-gdd-cc8	Node.js · heroku-18 · United States ☆
f-react-map-gl	Node.js · heroku-18 · United States ☆
fopency-line-detections	Node.js · heroku-20 · United States ☆
jupukaipi	Node.js · heroku-20 · United States ☆
zoom-relay	Node.js · heroku-20 · United States ☆

App name

something-catchy is available

Choose a region

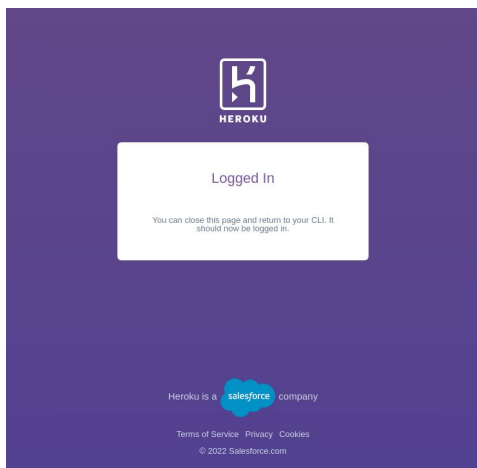
United States

Add to pipeline...

Create app



```
λ ~/ heroku login
> Warning: Our terms of service
have changed:
https://dashboard.heroku.com/terms
-of-service
heroku: Press any key to open up
the browser to login or q to exit:
```



Nah setelah kamu berhasil registrasi dan bikin aplikasi, kamu perlu instal **heroku cli**, kamu bisa ikuti tutorial [disini](#) ya!

Tools ini dipakai untuk manage aplikasi kamu di-heroku. Setelah kamu instal tools-nya, sekarang kamu login ke Heroku dulu nih lewat terminal.



Setelah itu, buka repository kalian di dalam terminal, dan jalankan perintah di samping.

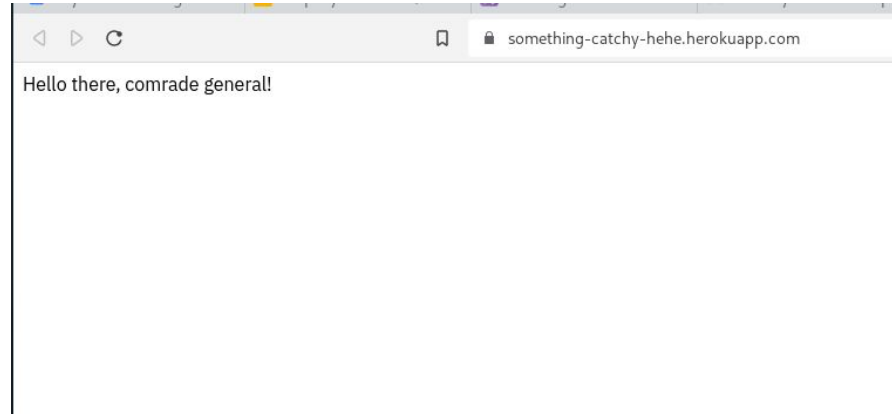
Perintah disamping akan menambahkan heroku sebagai remote repository kita dengan nama **heroku**. Dan kamu cuma perlu push aja buat deploy aplikasimu kayak perintah disamping.

```
heroku git:remote a namaproject-  
kamu  
git push heroku main
```



Setelah itu kamu bisa tunggu beberapa saat dan buka link berikut. **nama-aplikasi-mu.herokuapp.com**

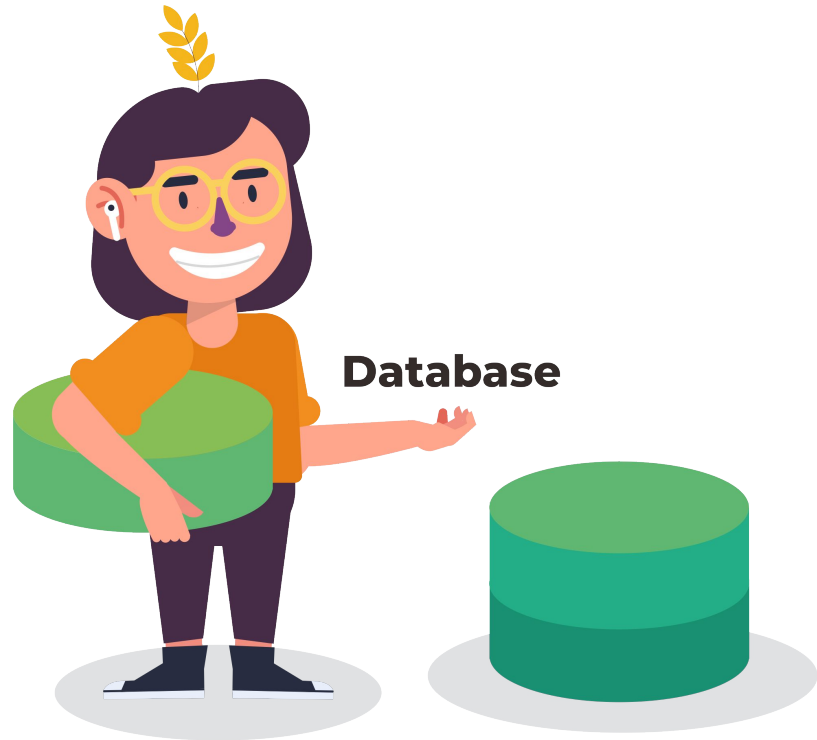
Setiap kali kamu mau update aplikasimu, tinggal push ke repository heroku tadi dan boom! versi terbaru aplikasi kamu bakal otomatis ter-deploy.





Eits, tapi masih banyak lho pekerjaanmu tentang deploy mendeploy ini. Kamu ga cuma harus mikirin aplikasi kamu doang, tapi kamu perlu mikirin aplikasi-aplikasi lain yang akan menjadi dependensi dari aplikasi kalian, sebagai contoh database, dan sebagainya.

Untuk setup hal tersebut kamu bisa ikuti [tutorial ini](#).





Gimana, simpel tapi masih males kan kalo tiap kali kamu mau men-deploy harus push ke heroku? Bisa jadi kalo kamu kerja sama temenmu, temen-mu lupa push ke heroku buat deploy versi tersebut.

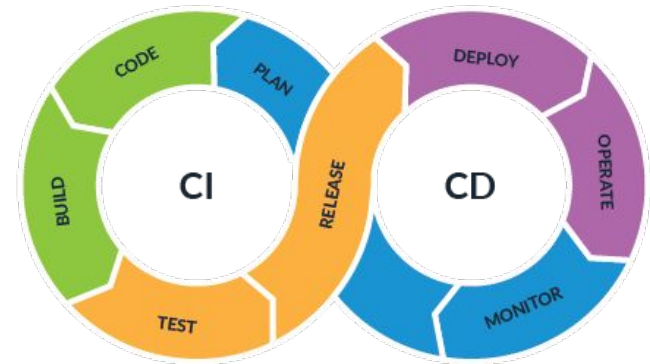
Kira-kira kamu akan kaget ga kalau Sabrina jelaskan cara **mengintegrasikan deploy** ke heroku di dalam Gitlab? Cus kita akan bahas di topik **CI/CD**





Benar sekali teman, kamu bisa setup integrasi antara Gitlab & Heroku melalui fitur CI/CD.

Apa sih CI/CD itu? CI/CD merupakan singkatan dari Continuous Integration dan Continuous Deployment. Sempelnya, dia adalah **tools yang men-automasi pekerjaan rilis aplikasi kita dengan menjalankan seperangkat perintah terminal.**





CI/CD menerapkan otomatisasi yang berkelanjutan dan dapat dipantau kelangsungan prosesnya sepanjang aplikasinya berjalan, mulai dari fase integrasi dan pengujian hingga penerapan aplikasi tersebut.

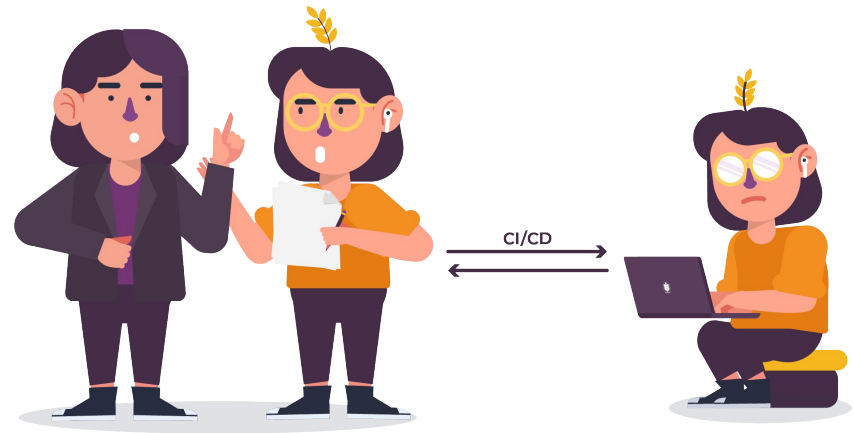
Secara keseluruhan, praktek yang berhubungan ini sering disebut dengan **CI/CD Pipeline**.





Terdapat tiga fase yang berupa **continuous integration, continuous delivery, dan continuous deployment**.

Ketiga fase tersebut akan dilakukan secara terus menerus dan otomatis untuk mendapatkan perangkat lunak yang andal dan bebas dari bug.





Nah, selanjutnya Sabrina bakal kasih tahu manfaat dari CI/CD, diantaranya:

- **Mendapat feedback lebih cepat**

Dalam penggunaan CI/CD pipeline ini kode akan diuji coba menggunakan CI tool secara bersamaan agar proses pengembangan perangkat lunak dapat berjalan dengan seimbang. Error yang terjadi juga bisa didapatkan lebih cepat sehingga tim pengembang pun dapat menindaklanjuti feedback tersebut secepat mungkin.





- **Dapat mendeteksi bug lebih cepat**

Seperti yang dikatakan sebelumnya, CI/CD ini bekerja dengan melakukan pengujian secara otomatis, sehingga jika ada bug yang muncul pada aplikasi yang dikembangkan maka akan langsung terdeteksi oleh CI tool. Pengembang juga dapat dengan mudah menemukan dan memperbaiki bug tersebut.



- **Dapat mempercepat proses rilis**

Proses rilis dari suatu aplikasi dapat mungkin dapat dipercepat. Hal itu disebabkan kode-kode yang terus digabungkan dan diterapkan ke dalam produk, sehingga aplikasi selalu dalam kondisi siap untuk dirilis kapan pun.





Nah CI/CD ini biasanya akan nempel sangat erat dengan Git repository, sebagai contoh.

- Gitlab punya Gitlab CI
- Github punya Github Action
- Dan sebagainya.





Cara setup CI/CD di Gitlab sangat sederhana, kamu hanya perlu bikin file bernama **.gitlab-ci.yaml** dengan isi seperti kode disamping. Dan verifikasi kartu kredit.

Jangan lupa setup CI/CD variable, seperti [tutorial disini](#). Dengan Heroku API key yang bisa kamu cari [disini](#).

Untuk tutorial ini, dianjurkan untuk menggunakan Repository Private yak!

```
stages:  
- deploy
```

```
Deploy to Heroku:
```

```
  stage: deploy  
  image: ruby:latest  
  only:  
  - main  
  script:  
  - gem install dpl  
  - dpl --provider=heroku --  
app=$HEROKU_APP_PRODUCTION --api-  
key=$HEROKU_API_KEY
```



Badabing badabung, kamu udah berhasil integrasiin Heroku dan Gitlab buat automatic deployment.

Reponya bisa kalian liat [disini](#) yaa!.

The screenshot shows the GitLab interface for the repository 'Express To Heroku'. The left sidebar contains navigation links: Project information, Repository, Issues (0), Merge requests (0), CI/CD, Security & Compliance, Deployments, Packages & Registries, Infrastructure, Monitor, Analytics, Wiki, Snippets, and Settings. The main content area displays the repository details, including a commit history table.

Name	Last commit	Last update
gitignore	feat: hello world	9 minutes ago
gitlab-ci.yml	ci: deploy to heroku	just now
README.md	Initial commit	12 minutes ago
index.js	feat: hello world	9 minutes ago

Saatnya kita
Quiz!





1. Tahapan kegiatan yang bertujuan untuk menyebarkan aplikasi yang telah dikembangkan pada suatu URL tertentu dalam server disebut

- A. Unit testing
- B. Deployment
- C. Integration process
- D. TDD



1. Tahapan kegiatan yang bertujuan untuk menyebarkan aplikasi yang telah dikembangkan pada suatu URL tertentu dalam server disebut

- A. Unit testing
- B. Deployment**
- C. Integration process
- D. TDD

Jadi, biar aplikasimu bisa diakses secara publik, kamu perlu melakukan deployment.



2. Berikut ini merupakan manfaat dari CI/CD, kecuali

- A. Mendapat feedback lebih cepat
- B. Dapat mendeteksi bugs lebih cepat
- C. Mengurangi jumlah developer
- D. Mempercepat proses rilis



2. Berikut ini merupakan manfaat dari CI/CD, kecuali

- A. Mendapat feedback lebih cepat
- B. Dapat mendeteksi bugs lebih cepat
- C. Mengurangi jumlah developer**
- D. Mempercepat proses rilis

Dengan menggunakan CI/CD kita bisa automasi proses-proses teknikal yang berulang, jadi dengan begini, kita ga perlu ngehire orang baru buat melakukan tugas-tugas itu.



3. Untuk memasang CI/CD di dalam sebuah repository Gitlab, kita perlu file apakah yang perlu kita buat?

- A. .gitlab-ci.yml
- B. .gitlab-ci.yaml
- C. .gitlab-ci.json



3. Untuk memasang CI/CD di dalam sebuah repository Gitlab, kita perlu file apakah yang perlu kita buat?

- A. `.gitlab-ci.yml`
- B. `.gitlab-ci.yaml`
- C. `.gitlab-ci.json`

Gitlab akan membaca file bernama `.gitlab-ci.yml` sebagai manifest CI/CD di dalam sebuah project.



4. Seperangkat script terminal yang ditulis di dalam .gitlab-ci.yml yang dijalankan jika memenuhi criteria trigger pada sebuah repository biasa disebut dengan?

- A. Pipeline
- B. Job
- C. CI/CD



4. Seperangkat script terminal yang ditulis di dalam .gitlab-ci.yml yang dijalankan jika memenuhi criteria trigger pada sebuah repository biasa disebut dengan?

- A. Pipeline
- B. Job**
- C. CI/CD

Job adalah sebuah kumpulan script yang akan dijalankan secara berurutan, sebagai contoh, job untuk menjalankan npm test dan sebagainya.



5. Sekumpulan job yang dijalankan secara berurutan berdasarkan stage di dalam sebuah manifest Gitlab disebut?

- A. Pipeline
- B. Job
- C. CI/CD



5. Sekumpulan job yang dijalankan secara berurutan berdasarkan stage di dalam sebuah manifest Gitlab disebut?

- A. Pipeline**
- B. Job
- C. CI/CD

Pipeline adalah sekumpulan job yang akan dijalankan secara berurutan pada kondisi yang ditentukan.

Terima Kasih!



Chapter ✓

completed