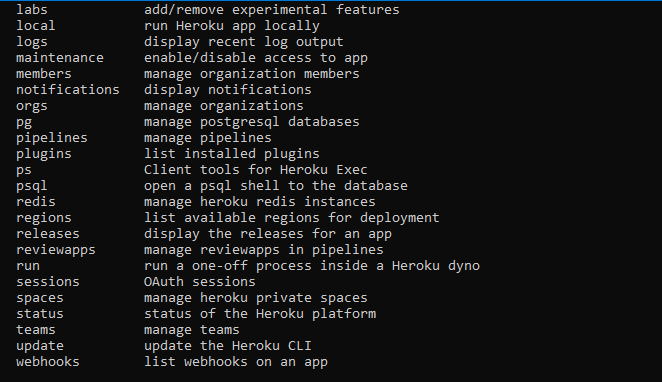
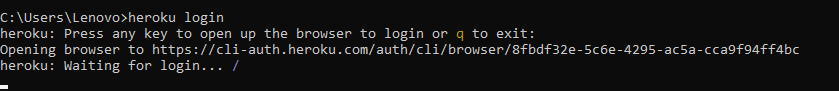
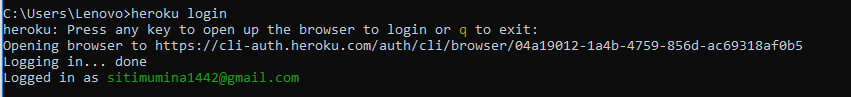
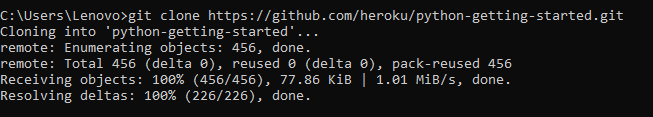
PRAKTIKUM TEKNOLOGI CLOUD COMPUTING minggu 03

1. Signup ke heroku
2. Buat aplikasi baru melalui dashboard
3. Install Python (dan PostgreSQL jika diperlukan)
4. Kerjakan [Getting Started on Heroku with Python](https://devcenter.heroku.com/articles/getting-started-with-python?singlepage=true), tidak perlu mengerjakan heroku pg:psql jika tidak ada PostgreSQL di lokal.





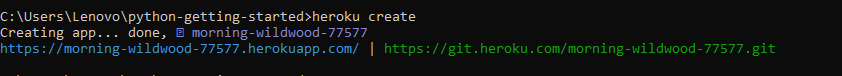




**Menyebarkan aplikasi**

Pada langkah ini Anda akan menyebarkan aplikasi ke Heroku.

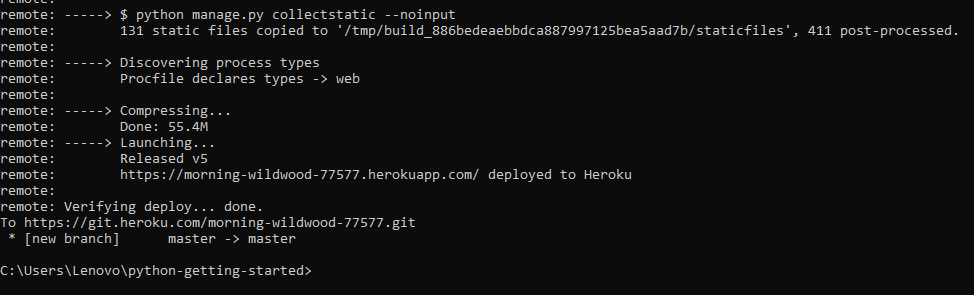
Buat aplikasi di Heroku, yang menyiapkan Heroku untuk menerima kode sumber Anda:



Saat Anda membuat aplikasi, git remote (dipanggil heroku) juga dibuat dan dikaitkan dengan repositori git lokal Anda.

Heroku menghasilkan nama acak (dalam hal ini serene-caverns-82714) untuk aplikasi Anda, atau Anda dapat memberikan parameter untuk menentukan nama aplikasi Anda sendiri.

Sekarang sebarkan kode Anda: dengan menggunakan perintah **“git push heroku master”**



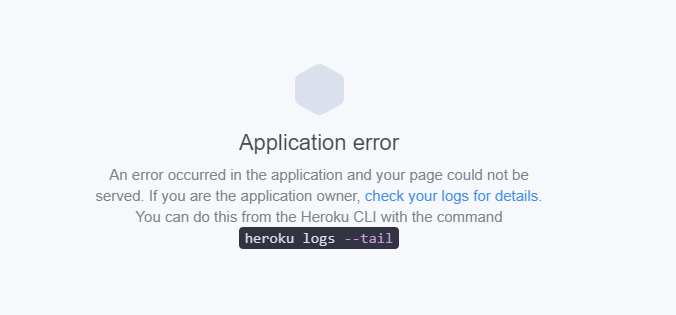
Aplikasi sekarang digunakan. Pastikan setidaknya satu instance aplikasi sedang berjalan:



Sekarang kunjungi aplikasi di URL yang dihasilkan oleh nama aplikasinya. Sebagai pintasan praktis, Anda dapat membuka situs web sebagai berikut:



Setelah di jalankan pada perintah di atas muncul tampilan seperti pada gambar di bawah ini.



Aplikasi error

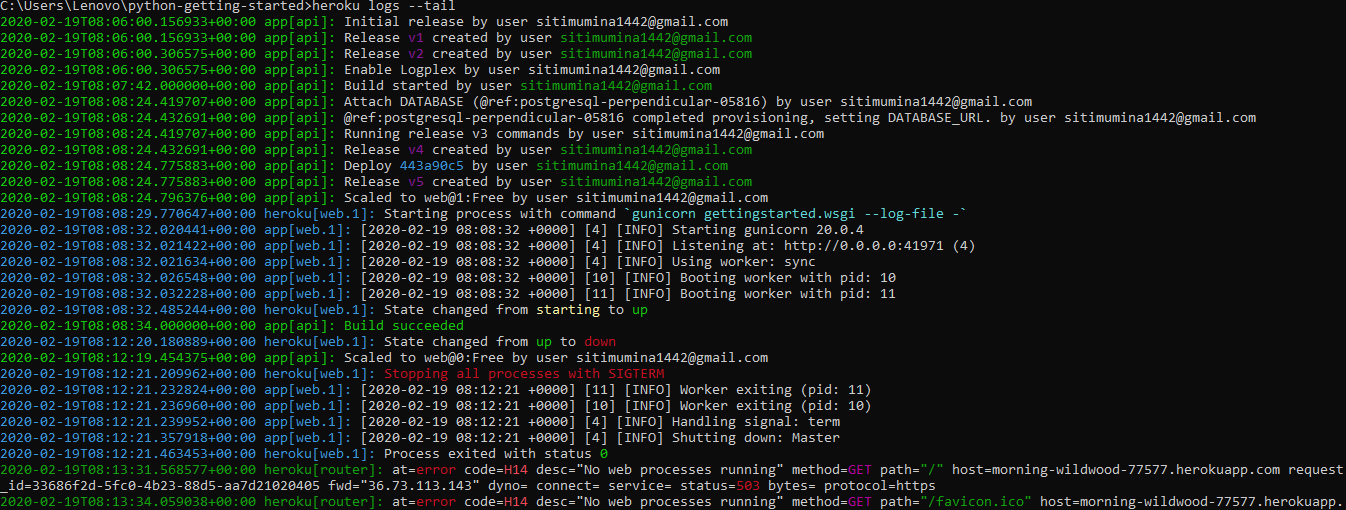
Terjadi kesalahan dalam aplikasi dan halaman Anda tidak dapat dilayani. Jika Anda adalah pemilik aplikasi, periksa rincian log Anda. Anda dapat melakukan ini dari Heroku CLI dengan perintah

**“log heroku --tail"**

## Lihat log

Heroku memperlakukan log sebagai aliran peristiwa yang dipesan waktu yang dikumpulkan dari aliran keluaran semua aplikasi Anda dan komponen Heroku, menyediakan saluran tunggal untuk semua peristiwa.

Melihat informasi tentang Anda menjalankan aplikasi menggunakan salah satu [perintah logging](https://devcenter.heroku.com/articles/logging) , heroku logs --tail:



Kunjungi aplikasi Anda di browser lagi, dan Anda akan melihat pesan log lain dibuat.

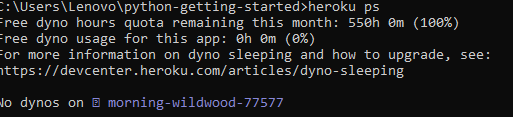
Tekan Control+Cuntuk berhenti mengalirkan log.

## Tetapkan Procfile

Gunakan [Procfile](https://devcenter.heroku.com/articles/procfile) , file teks di direktori root aplikasi Anda, untuk secara eksplisit menyatakan perintah apa yang harus dijalankan untuk memulai aplikasi Anda.

The Procfiledalam contoh aplikasi yang dikerahkan terlihat seperti ini:

web: gunicorn gettingstarted.wsgi --log-file –



Secara default, aplikasi Anda digunakan pada dyno gratis. Dyno gratis akan tidur setelah setengah jam tidak aktif (jika mereka tidak menerima lalu lintas apa pun). Ini menyebabkan penundaan beberapa detik untuk permintaan pertama saat bangun tidur. Permintaan selanjutnya akan berjalan normal. [Dyno gratis](https://devcenter.heroku.com/articles/free-dyno-hours) juga mengkonsumsi dari kuota bulanan gratis, tingkat akun tingkat [dyno](https://devcenter.heroku.com/articles/free-dyno-hours) - selama kuota tidak habis, semua aplikasi gratis dapat terus berjalan.

Untuk menghindari dino tidur, Anda dapat meningkatkan ke tipe hobi atau profesional seperti yang dijelaskan dalam artikel [Jenis Dyno](https://devcenter.heroku.com/articles/dyno-types) . Misalnya, jika Anda memigrasikan aplikasi ke dyno profesional, Anda dapat dengan mudah mengaturnya dengan menjalankan perintah yang memberitahu Heroku untuk menjalankan sejumlah dino tertentu, masing-masing menjalankan jenis proses web Anda.

Penskalaan aplikasi pada Heroku sama dengan mengubah jumlah dino yang sedang berjalan. Skala jumlah dinamika web menjadi nol:

