

2.1.7 *Deploy with Netlify*

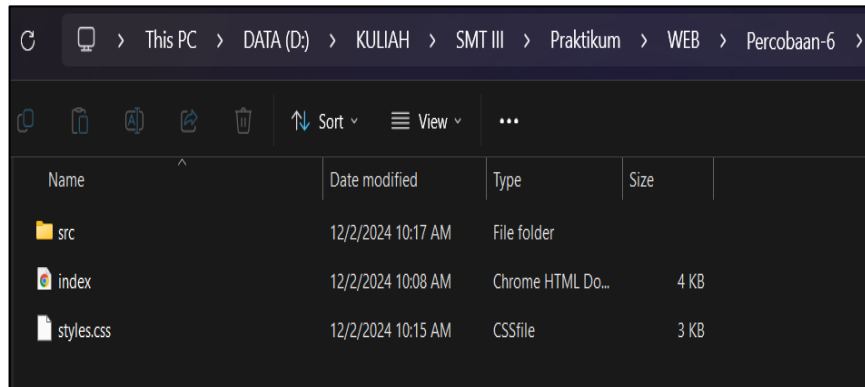
Netlify adalah platform *cloud* yang menyediakan layanan untuk meng-*host* dan men-*deploy* situs *web* dan aplikasi. Berikut adalah beberapa fitur dan fungsi utama dari Netlify:

1. *Hosting Statis*: Netlify terutama dikenal sebagai platform *hosting* statis. Ini berarti Netlify sangat cocok untuk meng-*host* situs *web* yang terdiri dari berkas HTML, CSS, JavaScript, dan aset statis lainnya. Netlify secara otomatis menyajikan situs *web* dari *cache* di lokasi geografis yang dekat dengan pengguna, memberikan kinerja yang lebih baik.
2. Integrasi Git: Netlify menyediakan integrasi yang kuat dengan sistem kontrol versi Git seperti GitHub, GitLab, dan Bitbucket. dapat menghubungkan repositori Git ke Netlify, dan setiap kali ada perubahan pada repositori, Netlify akan secara otomatis *deploy* situs baru.
3. Pengelolaan Alur Kerja (*Workflow*): Netlify memfasilitasi penerapan alur kerja (*workflow*) yang lancar. dapat mengkonfigurasi berbagai tahap dalam alur kerja pengembangan, pengujian, dan penerbitan.
4. Pembaruan Berbasis Pengguliran (*Rolling Releases*): Ketika ada pembaruan situs, Netlify mendukung pembaruan berbasis pengguliran, yang berarti pembaruan diterapkan secara bertahap ke server.
5. SSL Otomatis dan HTTPS Kuat: Netlify menyediakan SSL otomatis dan HTTPS gratis untuk semua situs yang di-*host*, bahkan untuk situs dengan domain kustom. Ini penting untuk keamanan dan SEO serta membuat situ menjadi lebih aman.
6. Fungsi *Serverless* (*Serverless Functions*): Netlify mendukung fungsi *serverless*, yang memungkinkan menulis dan menjalankan kode *backend* tanpa perlu mengelola server. Fungsi *serverless* ini dapat digunakan untuk menangani logika *server-side* pada situs *web*.
7. Pembaruan Otomatis dari *Branch* atau *Pull Request*: dapat mengaktifkan penerapan otomatis untuk setiap perubahan pada *branch* tertentu atau *pull request*. Ini mempermudah uji coba dan pengujian sebelum penerbitan ke lingkungan produksi.

3.1.7 *Deploy with Netlify*

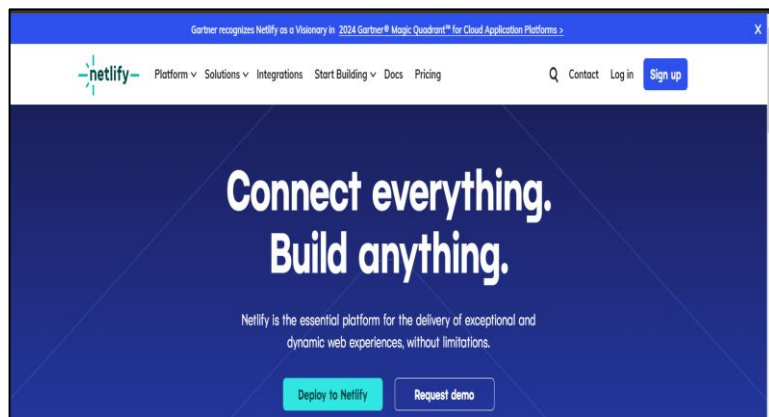
1. *Project*

Duplicate project percobaan ke-6 dan buat folder baru “Percobaan-7”

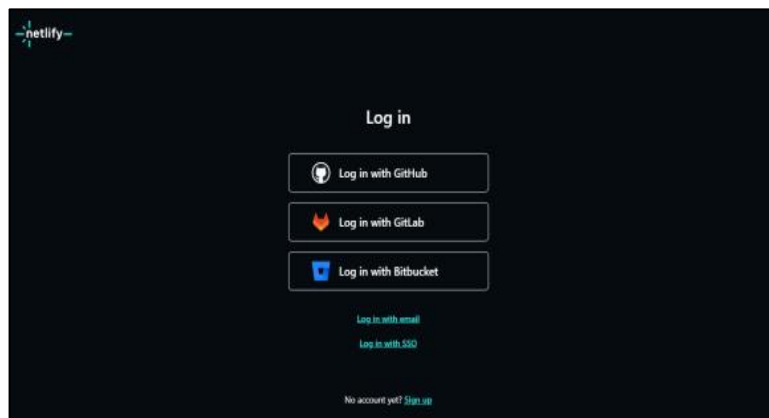


2. *Deploy*

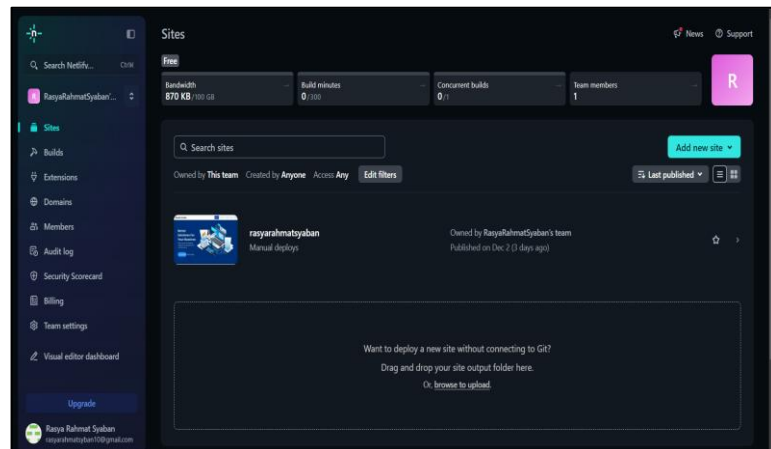
a. Buka <https://www.netlify.com/> dan pilih *Sign Up/Log In*



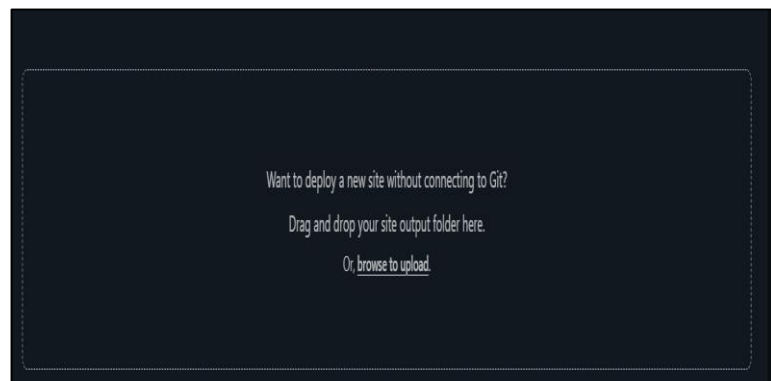
b. Silakan pilih *login* menggunakan apa saja atau daftar jika ingin menggunakan *email*.



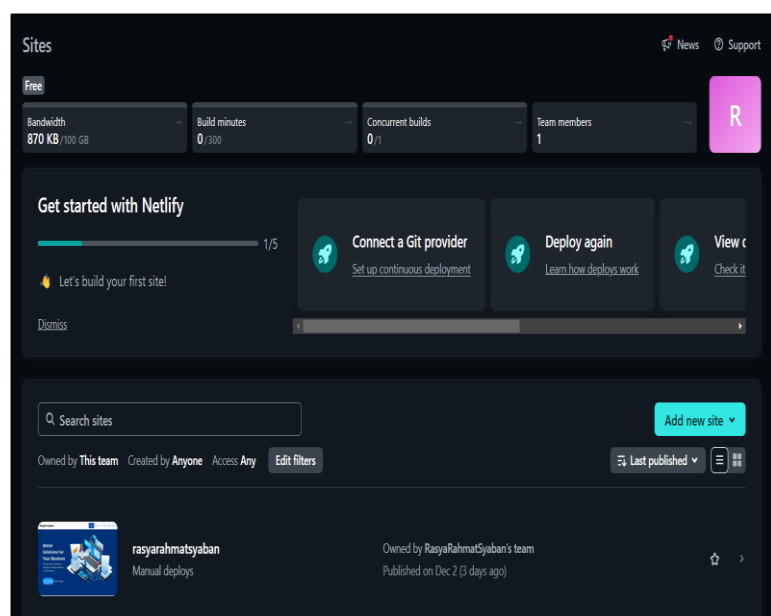
- c. Ikuti semua prosesnya sampai tampil halaman ini.



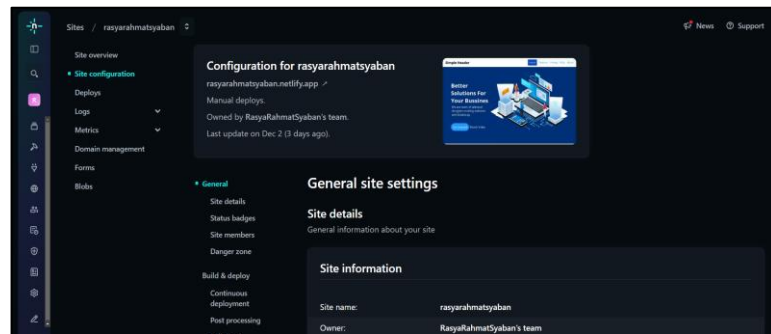
- d. Pilih *site*, kemudian *drag n drop* folder “Percobaan-7”.



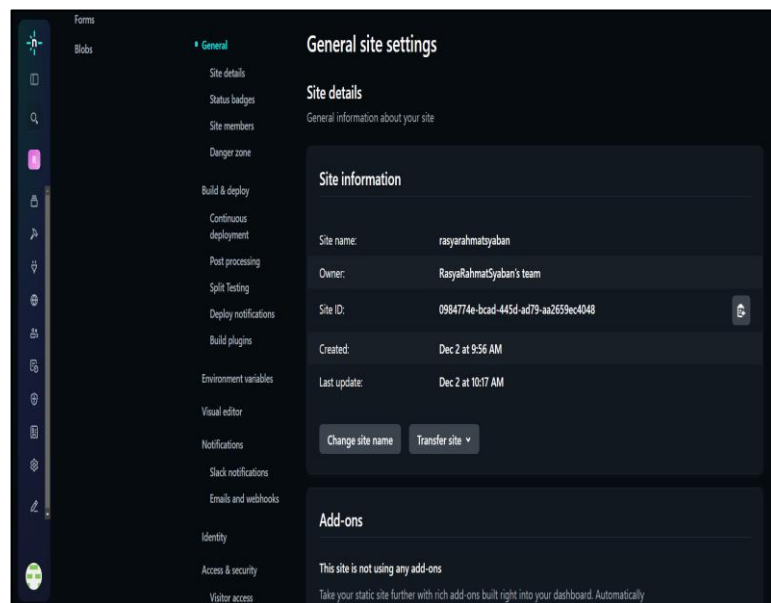
- e. Pada menu utama, “Team Overview” *scroll* dan pilih *site* tadi.



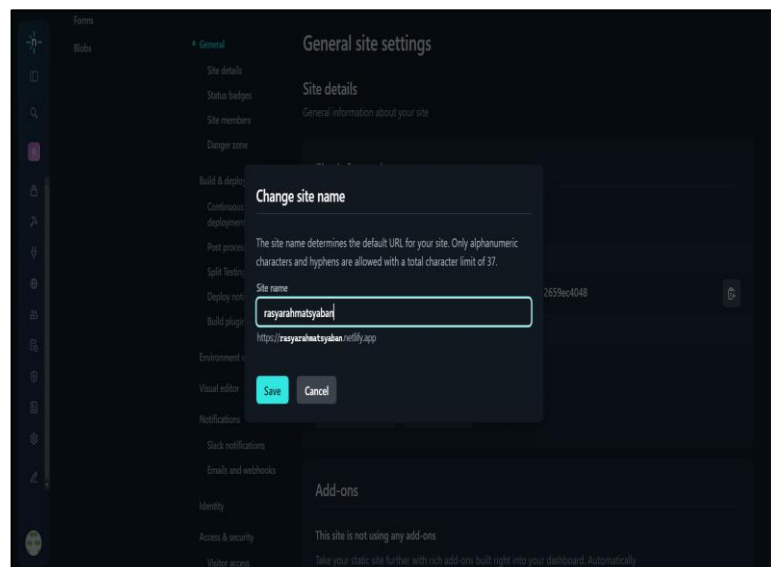
f. Kemudian pilih *site configuration*.



g. Pilih *change site name*.

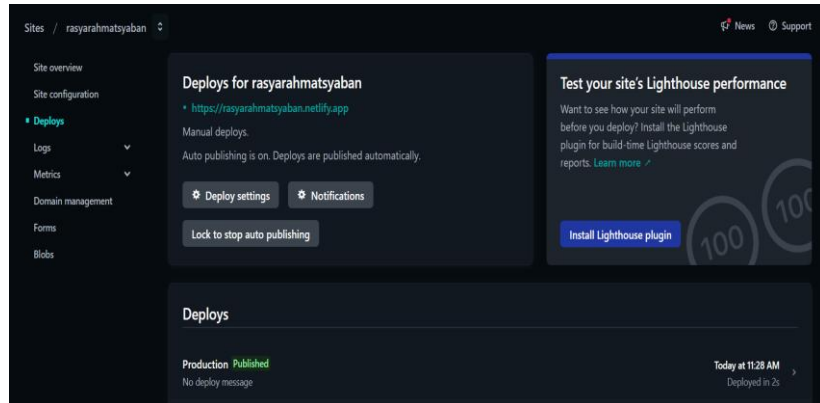


h. Ganti dengan nama kalian.



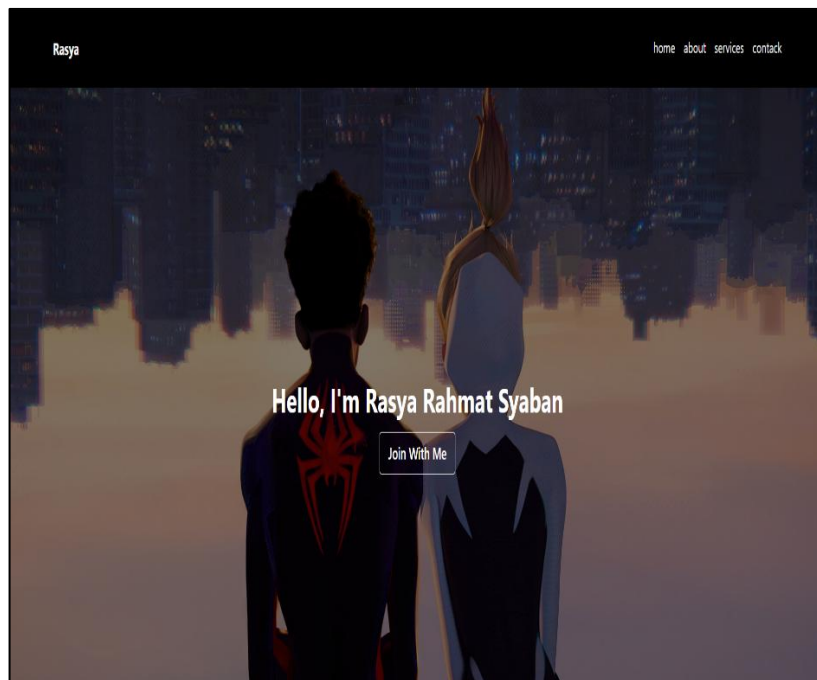
4.1.7 Deploy with Netlify


1. Deploy Succes



Gambar ini menunjukkan halaman *dashboard* Netlify untuk situs "rasyarahmatsyaban.netlify.app". Di bagian atas, ada opsi untuk mengatur pengaturan *deploy* dan notifikasi. *Auto publishing* diaktifkan, jadi setiap perubahan otomatis diterapkan ke situs. Status terakhir menunjukkan situs sudah dipublikasikan pada 9 Desember pukul 11:28 AM, dengan proses *deploy* yang selesai dalam 2 detik, menunjukkan proses yang cepat dan berhasil.

2. Web Deploy






About Me


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.


My Services

**Service 1**

Description of Service 1 goes here.

**Service 2**


Description of Service 2 goes here.


**Service 3**

Description of Service 3 goes here.

Contact Me

Contact Information

 Email: rasyaganteng@email.com

 Facebook: facebook.com/rasya

Name:



Email:

Message:

[Send Message](#)

Rasya

Copyright © Rasya Rahmat Syaban 2024

 Email: rasyaganteng@email.com
 Phone: +62 823 4678 9100

Halaman di atas merupakan antarmuka aplikasi web yang dimulai dengan header sederhana. Header tersebut berisi logo bertuliskan “Rasya” yang berada di sebelah kiri dan menu navigasi seperti *Home*, *About Me*, dan *My Services* di sebelah kanan. Bagian utama halaman menampilkan tulisan "*Hello, I'm Rasya Rahmat Syaban*", serta tombol aksi seperti "*Join With Me*". Selanjutnya, bagian *About Me* menampilkan informasi tentang diri. Pada bagian *My Services*, terdapat penjelasan tentang layanan yang ditawarkan. Di bagian selanjutnya, terdapat kolom yang menampilkan informasi kontak yaitu *email* dan *facebook*, serta *field* “nama”, “*email*”, dan “*message*” yang perlu diisi untuk dapat menekan tombol “message me” yang berguna untuk mengirim pesan. Pada bagian *footer*, terdapat lagi logo yang bertuliskan “Rasya” dan juga informasi mengenai *copyright* serta informasi kontak.

BAB V

PENUTUP

5.1 Kesimpulan

5.1.1 *Basic Tag HTML Part 1*

Dari hasil percobaan sebelumnya, menunjukkan bahwa struktur dasar HTML terdiri dari berbagai elemen seperti *tag* HTML, *head*, dan *body* yang menentukan tampilan halaman *web*. Penggunaan *tag heading*, paragraf, dan *formatting* memberikan variasi dan penekanan pada teks, sedangkan *tag* komentar membantu pengembang untuk mencatat informasi penting dalam kode tanpa memengaruhi tampilan. *Hyperlink* memungkinkan navigasi antar halaman dan ke URL *eksternal*, sedangkan *tag* gambar menyediakan cara untuk menampilkan media visual dengan atribut yang mendukung. Semua elemen ini berkontribusi pada pembentukan halaman *web* yang interaktif dan informatif.

5.1.2 *Basic Tag HTML Part 2*

Dari percobaan sebelumnya, dapat disimpulkan bahwa HTML memiliki elemen-elemen dasar seperti *list*, tabel, dan *form input*. *List* terbagi menjadi *unordered list* dan *ordered list*, yang masing-masing menggunakan *tag* “” dan “” serta diisi dengan elemen “”. Tabel dibangun dengan baris dan kolom menggunakan *tag* “<tr>”, “<td>”, dan “<th>”, yang dapat diatur ukurannya. *Form input* menyediakan berbagai tipe *input* untuk data pengguna, seperti *text*, *email*, *number* dan lainnya, serta atribut *id* dan *name* untuk identifikasi dan pengumpulan data. Dengan elemen-elemen ini, dapat memungkinkan pembuatan struktur halaman yang interaktif dan fungsional.

5.1.3 *HTML Layout*

Dari percobaan ini, dapat disimpulkan bahwa materi HTML Layout ini adalah penggunaan *semantic tag* seperti “<nav>”, “<section>”, “<article>”, dan “<footer>” membantu memberi makna yang jelas pada struktur halaman. Setiap elemen dibungkus dalam *div* dengan atribut *id* untuk identifikasi yang spesifik, yang memudahkan penerapan gaya atau

interaksi dengan CSS atau JavaScript. Elemen-elemen seperti “<h1>”, “<h2>”, “”, dan “<p>” membantu membentuk konten dan navigasi dengan lebih terstruktur. Struktur ini membuat halaman lebih rapi, mudah dipahami, dan meningkatkan aksesibilitas bagi pengguna maupun mesin pencari.

5.1.4 CSS (*Cascading Style Sheets*)

Dari percobaan sebelumnya, dijelaskan penggunaan HTML untuk menyusun elemen-elemen teks dan CSS untuk mengatur tampilannya. Melalui *class* dan *selector*, setiap elemen dapat diberi gaya khusus seperti warna latar, ukuran teks, dan tata letak fleksibel menggunakan *flexbox*. Properti seperti *padding*, *margin*, dan *border* menambah struktur visual yang rapi, sementara *background-image* digunakan untuk memperindah tampilan dengan gambar latar.

5.1.5 CSS *Part 2 Flex & Grid*

Dari percobaan sebelumnya, dapat disimpulkan bahwa materi HTML *Layout* ini mengajarkan cara menggunakan *layout flexbox* dan *grid* untuk mengatur elemen-elemen di halaman *web* dengan cara yang lebih efisien dan responsif. Penggunaan elemen-elemen seperti “<h2>”, “<div class=’grid-container’>”, dan “<div class=’grid-item’>” membantu dalam menyusun konten dengan tata letak yang rapi dan terstruktur. Penerapan properti CSS seperti “*display: flex;*”, “*grid-template-columns;*”, dan “*grid-gap;*”. Struktur ini membuat halaman lebih teratur, mudah dinavigasi, dan responsif terhadap berbagai ukuran layar.

5.1.6 CSS *Framework TailwindCSS*

Dari percobaan sebelumnya, dapat disimpulkan bahwa modul CSS *Framework TailwindCSS* ini mengajarkan cara memanfaatkan kelas-kelas bawaan untuk mempercepat proses pengembangan *web* tanpa harus menulis CSS secara manual. Penggunaan kelas seperti “*bg-black;*”, “*flex;*”, “*grid;*”, “*text-white;*”, dan “*rounded-lg;*” mempermudah pembuatan elemen yang responsif, terstruktur, dan estetis. Modul ini juga menunjukkan fleksibilitas TailwindCSS dalam mengatur *layout*, warna, dan elemen interaktif seperti

tombol dan formulir. Dengan pendekatan berbasis *utility class*, TailwindCSS memungkinkan pembuatan halaman *web* yang modern, responsif, dan efisien.

5.1.7 Deploy with Netlify

Dari percobaan ini, dapat disimpulkan bahwa materi tentang *deploy* dengan Netlify mengajarkan cara mempublikasikan situs secara otomatis. Situs dapat diakses melalui URL yang ditentukan, seperti "rasyarahmatsyaban.netlify.app". Melalui dashboard Netlify, pengaturan dan notifikasi *deploy* dapat diatur, dan proses *deploy* berjalan cepat, seperti yang terlihat pada contoh ini, di mana situs berhasil dipublikasikan dalam 3 detik.