

“berikan saya sebuah langkah apa saja yang dibutuhkan untuk membuat sebuah website wisata pantai yang ada di lombok,yang saya harapkan dari sebuah website ini adalah website ini langsung terhubung langsung dengan API google map ketika kita mengklik terbetuliskan lokasi maka akan langsung muncul lokasinya di google map”

Berikut adalah langkah-langkah komprehensif untuk membuat website wisata pantai Lombok dengan fitur integrasi Google Maps yang Anda inginkan.

Saya juga telah membuatkan dokumen rencana detailnya di Rencana Pengembangan.

Langkah-Langkah Pembuatan:

1. Perencanaan & Pengumpulan Data
 - Kumpulkan data pantai (Nama, Deskripsi, Foto).
 - Penting: Dapatkan titik koordinat (Latitude, Longitude) atau Link Google Maps untuk setiap pantai. Contoh: Pantai Kuta Mandalika.
2. Persiapan Google Maps (Pilih Salah Satu)
 - Opsi A (Google Maps API - Recommended untuk fitur "Embed"):
 1. Buka Google Cloud Console.
 2. Buat Project baru.
 3. Aktifkan Maps JavaScript API.
 4. Buat API Key. (Perlu kartu kredit/debit untuk verifikasi, meski ada kuota gratis).
 - Opsi B (Google Maps Links - Lebih Mudah & Gratis):
 1. Gunakan format link standar:

<https://www.google.com/maps/search/?api=1&query>Nama+Pantai>

.

1. Saat user klik, akan membuka aplikasi Maps/Web Maps langsung ke lokasi tersebut.
1. Struktur Web (Coding)
 - HTML: Membuat kerangka halaman (Header, Hero Section, Daftar Wisata).
 - CSS: Mendesain tampilan agar estetik (tema pantai, biru/putih, modern).
 - JavaScript: Membuat logika agar ketika tombol diklik, fungsi Google Maps dipanggil.
2. Implementasi Fitur Peta
 - Jika menggunakan API: Buat elemen

<div>

untuk peta dan gunakan script JS untuk merender peta saat tombol diklik.

- Jika menggunakan Link: Buat tag

<a>

dengan

href

dinamis sesuai lokasi.

Apakah Anda ingin saya mulakan pembuatan kerangkanya?

Saya bisa langsung membuatkan file dasar (

index.html

,

style.css

,

script.js

) di folder project Anda sekarang. Saya akan siapkan dengan contoh 1-2 pantai di Lombok.

“oke tapi sebentar dulu,seblum itu aku juga ingin websitnya menggunakan fitur login jadi ada data basenya”

Merencanakan Fitur Login & Database

User meminta fitur login dan database. Saya memperbarui dokumen rencana untuk mencakup arsitektur Fullstack (Node.js + SQLite) untuk menangani autentikasi pengguna secara lokal.

Files Edited

Implementation PlanTask

Progress Updates

1

Memperbarui rencana implementasi untuk mencakup backend dan autentikasi

Baik, untuk fitur login dan database, saya menyarankan menggunakan Node.js + SQLite. Alasannya:

1. Mudah & Ringan: SQLite tidak perlu install aplikasi database berat (seperti MySQL), datanya hanya berupa satu file di dalam folder proyek.
2. Modern: Node.js adalah standar industri saat ini untuk web development.

“konfirmasi dulu website ini saya buat untuk wisatawan bule jadi memakai bahasa inggris dan kita bisa mengubah bahasanya ke 10 negara yang sering berwisata ke lombok”

Merencanakan Fitur Multibahasa

User menambahkan kebutuhan multibahasa (English + 10 negara lain). Saya memperbarui Roadmap dan Task List untuk mencakup implementasi sistem I18n (Internationalization) sederhana menggunakan JavaScript dan JSON dictionary.

baik, Rencana telah diperbarui. Fitur Utama:

1. Backend Login: Node.js + SQLite.
2. Multilingual: Default Bahasa Inggris + 10 Bahasa (Jerman, Prancis, Belanda, Italia, Spanyol, Mandarin, Jepang, Korea, Rusia, Arab).
3. Google Maps: Integrasi lokasi langsung.

Boleh saya mulai coding sekarang? Saya akan mulai dengan setup folder dan backend server dulu.

“oke sekarang buat kan saya, pastikan jumlah wisatanya 15 sesuai yang saya minta”

Project Initialization & Backend Setup

Starting execution phase. Initializing Node.js project, installing dependencies (express, sqlite3, bcrypt, etc.), and setting up the basic server structure with SQLite database.

Progress Updates

1

Initializing Node.js project and installing dependencies

2

Checking for Node.js availability to proceed with backend setup

Saya mendeteksi bahwa Node.js tidak terinstall di komputer Anda. Tanpa Node.js, kita tidak bisa menjalankan server backend dan database SQLite seperti rencana awal.

Bagaimana jika kita menggunakan Teknologi Browser (LocalStorage) sebagai pengganti Database? Kelebihan:

1. Tidak perlu install apapun. Langsung jalan di browser (klik

index.html

saja).

1. Tetap bisa Login, Register, dan Logout (Data user tersimpan di browser).
2. Tampilan dan fitur Google Maps tetap 100% sama.
3. Sangat cocok untuk presentasi UAS karena praktis.

Jika Anda setuju, saya akan langsung buat kan kodenya sekarang. Apakah boleh?

”jangan jalankan ssuai rencana saja”

Researching Lombok Beach Data

User decided to stick with the Node.js/SQLite plan despite missing dependencies. I am now proceeding to research 15 beach locations in Lombok to satisfy the content requirement before generating the code logic.

Proyek Website Wisata Pantai Lombok telah selesai DIBUAT! 🎉

Status:

1. Backend Node.js + SQLite: Siap (termasuk 15 data pantai & sistem login).
2. Frontend Estetik: Sudah jadi dengan desain premium dan fitur multibahasa (11 Bahasa).
3. Panduan Install: Karena komputer Anda belum ada Node.js, saya sudah buat kan file

PANDUAN_INSTALL.txt


yang isinya langkah-langkah mudah untuk menginstallnya agar website bisa jalan.

Silakan cek file

walkthrough.md

untuk ringkasan lengkapnya. Apakah ada yang ingin ditanyakan lagi?

“Saya ingin upload website UAS saya ke GitHub Pages. Karena GitHub Pages bersifat statis, tolong tulis ulang kode saya dengan aturan berikut: Hapus Node.js & Backend: Hilangkan semua ketergantungan pada server.js, require, dan database.js. Ganti Database ke LocalStorage: Ubah fitur Login, Register, dan penyimpanan data agar tersimpan di memori browser (LocalStorage) saja, bukan di database MySQL. Satukan (atau rapikan) File: Buat agar semuanya berjalan hanya dengan membuka file index.html saja tanpa perlu install apapun.”

Konversi ke Static Site selesai! 

Apa yang sudah saya lakukan:

1. Hapus Node.js: Backend dan file database lama (

server.js

,

database.js

) sudah dihapus.

1. Struktur Folder: Semua file sekarang ada di folder utama, tidak ada lagi folder

public


.

1. LocalStorage: Login & Register sekarang jalan 100% di browser tanpa server.

2. Siap Upload: Anda tinggal upload semua file ini ke GitHub Repository untuk jadi website (GitHub Pages).

Silakan coba buka file

index.html

langsung. Klik dua kali saja. 

”buat perubahan coba, ketika pengunjung mengganti bahasa maka semua deskripsi juga berubah sesuai dengan bahasa yang dipilih seperti perubahan deskripsi pada paces_data ”

Siap. Rencana saya untuk fitur "Deskripsi Berubah Sesuai Bahasa":

1. Pindahkan Teks: Teks deskripsi di

main.js

akan saya hapus dan pindahkan ke

languages.js

(Kamus Bahasa).

1. Ganti dengan Kunci: Di

main.js


, saya hanya akan simpan "Kode" (contoh:

beach_1_desc


), bukan teks aslinya.

1. Update Logika: Saat Anda ganti bahasa di dropdown, kode akan otomatis mencari teks yang pas di kamus.

Saya akan terapkan ini sekarang. Setuju?

Sudah selesai! 

Apa yang baru?

1. **Deskripsi Dinamis:** Sekarang, jika Anda mengganti bahasa (misal: ke Jerman ) , semua Nama Pantai dan Deskripsinya akan ikut berubah ke Bahasa Jerman.
2. **11 Bahasa:** Saya sudah memasukkan terjemahan lengkap untuk 15 pantai ke dalam 11 bahasa.


Silakan **refresh** halaman

index.html


Anda dan coba ganti bahasanya. Lihat bagaimana kontennya berubah seketika.

Ada lagi yang bisa saya bantu untuk menyempurnakan project UAS ini?

”coba tambahkan juga bahasa Indonesia”

Berhasil! Bahasa Indonesia  sudah ditambahkan dan saya letakkan di urutan kedua (tepat di bawah English).

Sekarang website Anda mendukung 12 Bahasa! Coba refresh dan pilih "Indonesia" dari menu, deskripsi pantai akan tampil dalam Bahasa Indonesia yang lengkap.

Ada lagi tambahan lain? 

“sekarng saya mau ada perubahan lagi 1) ketika pengunjung mendaftar akun,pada tampilan pendaftaran akun sertakan juga nama pengguna,asal negara, dan nama pengguna ini akan di pakai untuk login 2) ketika user sudah di dalam dashboard sertakan nama pengguna di samping logout 3) tampilan mode malam dan mode terang”

Rencana Fitur Tambahan sudah siap:

1. Registrasi: Saya tambahkan kolom "Nama Lengkap" dan "Asal Negara".
2. Dashboard: Nama Anda akan muncul di samping tombol Logout.
3. Tampilan: Ada tombol "Bulan/Matahari" untuk ganti Dark Mode.

“ganti nama lengkap dengan username,dan pastikann juga pemilihan bahasa pada luar dashboard atau pada tampilan login maupun daftar setelah itu komfirmasi dulu ke saya sebelum buat”

Baik, saya sudah perbaiki rencananya sesuai permintaan Anda:

1. Pendaftaran: Input yang diminta adalah Username, Password, dan Asal Negara (Nama Lengkap dihapus).
2. Dashboard: Akan menampilkan Username Anda di samping tombol Logout.
3. Bahasa: Pilihan bahasa dipastikan ada di halaman Login, Daftar, dan Dashboard (Sudah ada, akan saya pastikan fungsinya lancar).
4. Dark Mode: Fitur ganti tema gelap/terang.

Apakah ini sudah pas? Jika ya, saya langsung kerjakan kodenya.

“ya sekarang kerjakan”

Pembaruan Selesai! ✨

Semua permintaan Anda sudah saya terapkan:

1. Pendaftaran: Sekarang ada kolom "Country / Negara Asal".
 - Catatan: Saya tidak lagi meminta "Full Name", tapi menggunakan "Username" untuk login seperti yang Anda minta.
2. Dashboard:
 - Di pojok kanan atas akan muncul sapaan: "Hi, [Username]".
 - Ada tombol Bulan (🌙) untuk mengaktifkan Dark Mode yang estetik.
 - Tombol bahasa tetap ada dan berfungsi 100%.

Silakan dicoba daftarkan akun baru untuk melihat semua perubahannya. Ada revisi lagi?

“hapus saja mode malam dan buat website ini juga responsibel”

Siap, Revisi Selesai! ✅

1. Dark Mode Dihapus: Tombol bulan/matahari sudah hilang. Website sekarang bersih dengan tampilan terang/glassmorphism saja.
2. Responsif (Mobile-Friendly): Saya sudah merombak CSS agar tampilan di HP lebih rapi:
 - Menu navigasi bertumpuk rapi.
 - Grid kartu pantai menjadi 1 kolom di layar kecil agar gambar terlihat jelas.
 - Padding dan font size disesuaikan agar enak dibaca di layar sentuh.

Silakan cek kembali tampilannya, coba kecilkan ukuran window browser Anda untuk melihat efek responsifnya. Masih ada lagi?

