|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NASKAH UJIAN FAKULTAS ILMU KOMPUTER UNIVERSITAS BUNG KARNO** | | | | | | | | | | |
| **UTS** | **UAS** | **Susulan UTS / UAS** | | | | **Lain-lain: …………….** | | **Ganjil / ~~Genap~~: 2024-2025** | | |
| **Program Studi** | | **:** | Sistem Informasi | | | | | **Tipe Ujian** | **:** | Essay |
| **KMK - Mata Kuliah** | | **:** | MKB7220334 - Pemrograman Web + Praktikum | | | | | **NIM** | **:** | 7201230108 |
| **Bobot SKS** | | **:** | 3 SKS | | | | |
| **Kelas - Ruang** | | **:** | 3 SIM - D/304 | | | | | **Nama**  **Mahasiswa/i** | **:** | Nur Ishmah Salsabilla |
| **NUPTK** | | **:** | 5946773674130272 | | | | |
| **Nama Dosen** | | **:** | Joko Saputro, S.Kom., M.Kom. | | | | |
| **Hari, Tanggal** | | **:** | Kamis, 16-01-2024 | | | | **Diperiksa Oleh:**  (Joko Saputro, S.Kom., M.Kom) Tanggal: 10-11-2025 | **Tanda Tangan** | **:** |  |
| **Waktu Ujian** | | **:** | 150 Menit / 17:00-19:30 | | | |
| **Sifat Ujian** | | **:** | OpenBook | CloseBook | Online | |
| **Lembar Jawaban** | | **:** | Ya / ~~Tidak~~ | | | |
| **Kalkulator** | | **:** | ~~Ya~~ / TIdak | | | |

**Petunjuk Umum:**



1. Berdoa terlebih dahulu sebelum anda memulai mengerjakan !



2. Bacalah soal dengan cermat dan pastikan anda memahami apa yang ditanyakan sebelum dijawab !

3. Berikan jawaban yang lengkap dan informatif sesuai pertanyaan yang diajukan !

4. Gunakan bahasa yang jelas dan mudah dipahami!

5. Mulailah mengerjakan soal yang menurut anda mudah terlebih dahulu !

6. Kerjakan setiap soal dengan penuh kejujuran dan rasa tanggung jawab !

7. Percaya diri dengan jawaban masing-masing !

8. Jangan lupa untuk mencantukkan nim, nama, dan tanda tangan anda sebelum menjawab soal !

9. Jawaban harus ditulis dengan tulisan yang jelas dan mudah dibaca serta tidak diperbolehkan kerja sama dengan orang lain !

10. Periksalah kembali jawaban anda sebelum mengumpulkannya !

11. Jika ada jawaban yang identik (copy paste), nilai akan dibagi sesuai dengan jumlah jawaban yang sama !

12. Ujian dilaksanakan sesuai batas waktu yang telah ditetapkan dan pengumpulan lembar jawaban ujian tidak boleh melebihi batas waktu yang telah ditetapkan !

**Petunjuk Khusus:**

… *isi disini jika ada petunjuk khusus*…

1. Buatlah salinan dari lembar jawaban yang telah anda kerjakan, kemudian kirimkan ke alamat email:

[jokosaputro@ubk.ac.id,](mailto:jokosaputro@ubk.ac.id) subject: UAS\_WP\_NIM\_NAMA LENGKAP

2. Salinan harus berupa file dengan format .doc atau .docx tidak berformat ~~.pdf~~.

3. Kumpulkan source code dan jawaban terkait melalui github.

**Soal:**

1) Jelaskan perbedaan antara HTML, CSS dan JavaScript dalam pengembangan web! Berikan contoh bagaimana ketiganya saling bekerja sama untuk membuat jalaman web yang interaktif!

2) Apa itu HTTP? Jelaskan perbedaan antara metode GET dan POST dalam HTTP, sertakan contoh kapan masing-masing metode tersebut digunakan!

3) Jelaskan konsep DOM (Document Object Model) dalam JavaScript. Bagaimana cara memanipulasi element HTML menggunakan DOM!

4) Apa perbedaan antara static website dan dynamic website? Berikan contoh teknologi atau bahasa pemrograman yang digunakan untuk masing-masing jenis website tersebut!

5) Jelaskan cara kerja client-server dalam arsitektur web. Berikan contoh bagaimana browser berkomunikasi dengan server untuk membuat halaman web!

6) Buatlah halaman HTML sederhana yang memiliki form pendaftaran dengan elemen berikut:

- Input untuk nama, email dan password

- Tombol “Submit”

Berikan atribut HTML5 yang sesuai untuk setiap elemen form.

7) Tulis kode CSS sederhana untuk membuat tombol dengan properti berikut:

- Latar belakang biru

- Teks Putih

- Membulatkan sudut tombol dengan radius 10px

- Efek hover yang mengubah latar belakang menjadi hijau

8) Anda diminta untuk membuat tabel sederhana menggunakan HTML untuk menampilkan daftar produk dengan kolom berikut: Nama Produk, Harga, dan Ketersediaan Stok. Tulis kode HTML-nya dan tambahkan sedikit css untuk mempercantik tampilan tabel.

9) Buatlah sebuah fungsi JavaScript sederhana yang akan menampilkan pesan “Selamat Datang!”

ketika sebuah tombol diklik. Sertakan HTML dan JavaScript-nya

10) Kumpulkan seluruh jawaban, source code secara rapih gunakan teknik file manajemen kemudian upload ke akun github dan kumpulkan!

\*) Cara Pengiriman:

- kirimkan link repositori uas anda kemudian kirimkan ke-email[: jokosaputro@ubk.ac.id](mailto:jokosaputro@ubk.ac.id)

- Batas waktu terakhir pengumpulan yakni: **Minggu, 18-01-2025, 23:00 WIB**

-----------------------------------------------\*~Selamat Mengerjakan~\*-----------------------------------------------

1. HTML, CSS, dan JavaScript adalah tiga teknologi utama dalam pengembangan web, masing-masing memiliki peran yang berbeda namun saling melengkapi untuk menciptakan halaman web yang fungsional, menarik, dan interaktif.

**1. HTML (HyperText Markup Language)**

**Fungsi:** HTML adalah struktur dasar dari sebuah halaman web. HTML digunakan untuk menentukan konten dan elemen-elemen seperti teks, gambar, tabel, formulir, dan lainnya.

**Contoh:** Menyusun teks paragraf, header, daftar, atau menambahkan gambar.

**2. CSS (Cascading Style Sheets)**

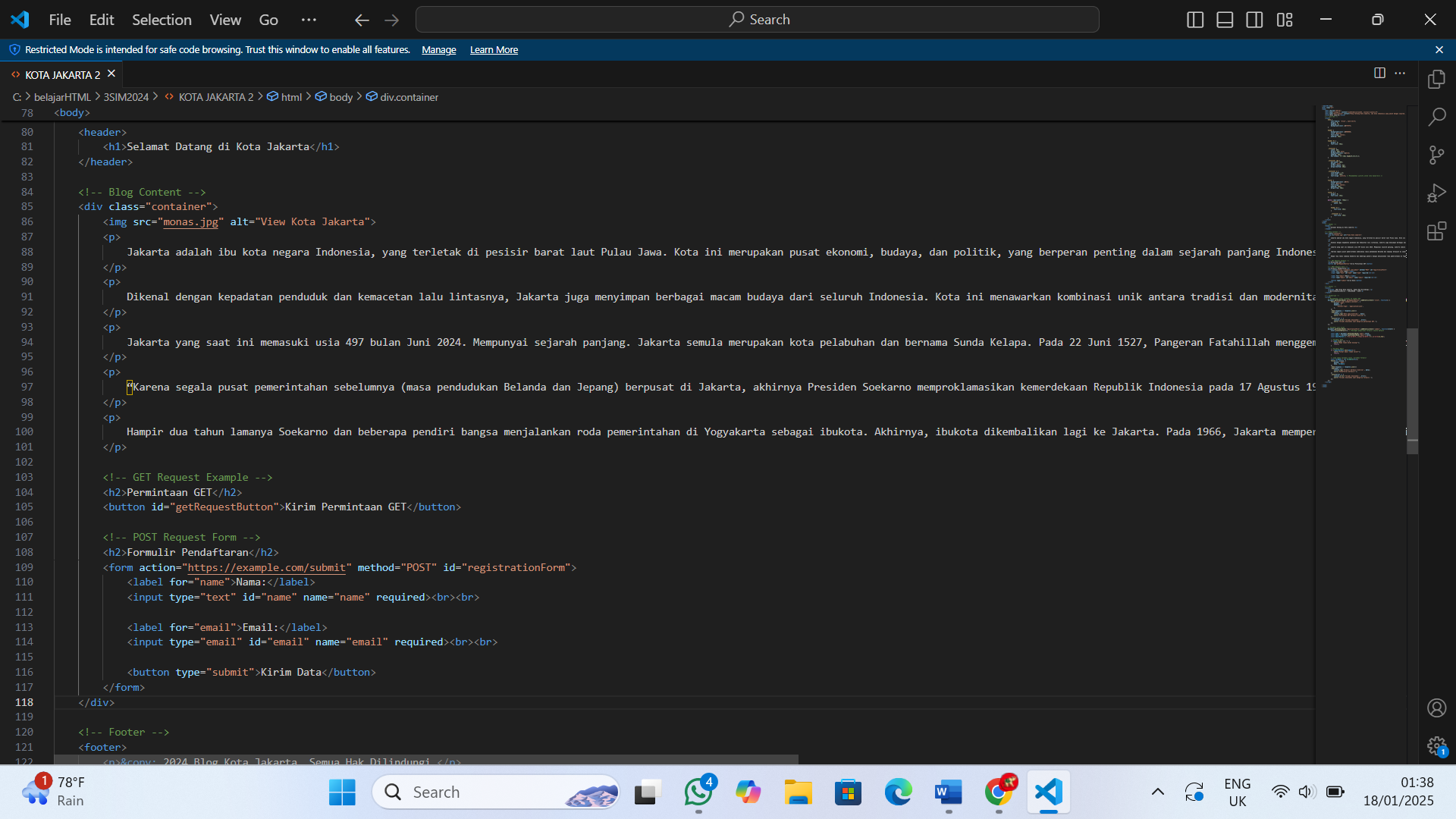
**Fungsi:** CSS digunakan untuk mengatur tampilan dan tata letak halaman web. Dengan CSS, pengembang dapat mengubah warna, ukuran, posisi, dan gaya elemen HTML agar terlihat menarik.

**Contoh:** Memberikan warna latar belakang, mengatur font, atau menyusun elemen dalam grid atau flexbox.

**3. JavaScript**

**Fungsi:** JavaScript adalah bahasa pemrograman yang membuat halaman web menjadi dinamis dan interaktif. Dengan JavaScript, pengembang dapat menambahkan animasi, validasi formulir, pengolahan data, atau elemen interaktif lainnya.

**Contoh:** Menampilkan notifikasi, memvalidasi input pengguna, atau membuat konten yang berubah tanpa perlu memuat ulang halaman.



1. **HTTP (HyperText Transfer Protocol)** adalah protokol komunikasi yang digunakan untuk mengirim dan menerima data antara klien (misalnya browser) dan server di web. HTTP adalah fondasi dari komunikasi di World Wide Web, memungkinkan pengiriman dokumen HTML, file, gambar, video, dan data lainnya. HTTP bekerja berdasarkan model **request-response**, di mana klien mengirim permintaan (request) dan server memberikan respons (response).

**Metode GET dan POST dalam HTTP**

HTTP menyediakan beberapa metode untuk mengirim permintaan ke server, dan dua metode yang paling umum adalah **GET** dan **POST**.

**1. Metode GET**

**Fungsi:** Metode GET digunakan untuk meminta data dari server tanpa memodifikasi sumber daya di server.

**Karakteristik:**

* + Data dikirim melalui URL sebagai parameter query string (misalnya, ?key=value).
  + Tidak aman untuk mengirim data sensitif karena data terlihat di URL.
  + Batasan panjang pada data yang dikirim (karena keterbatasan panjang URL).
  + Bisa di-cache oleh browser.
  + Digunakan untuk permintaan yang idempotent (tidak mengubah data di server).

**Kapan Digunakan:**

* + Saat ingin mengambil data tanpa mengubah server.
  + Contoh: Mencari artikel di mesin pencari, menampilkan produk berdasarkan filter di toko online.

**Contoh GET Request:** URL:

**2. Metode POST**

**Fungsi:** Metode POST digunakan untuk mengirim data ke server untuk memproses atau menyimpan sesuatu.

**Karakteristik:**

* + Data dikirimkan dalam badan (body) permintaan, bukan URL.
  + Lebih aman dibanding GET untuk data sensitif karena tidak terlihat di URL.
  + Tidak memiliki batas panjang pada data yang dikirim.
  + Tidak di-cache oleh browser.
  + Digunakan untuk permintaan yang tidak idempotent (mengubah data di server).

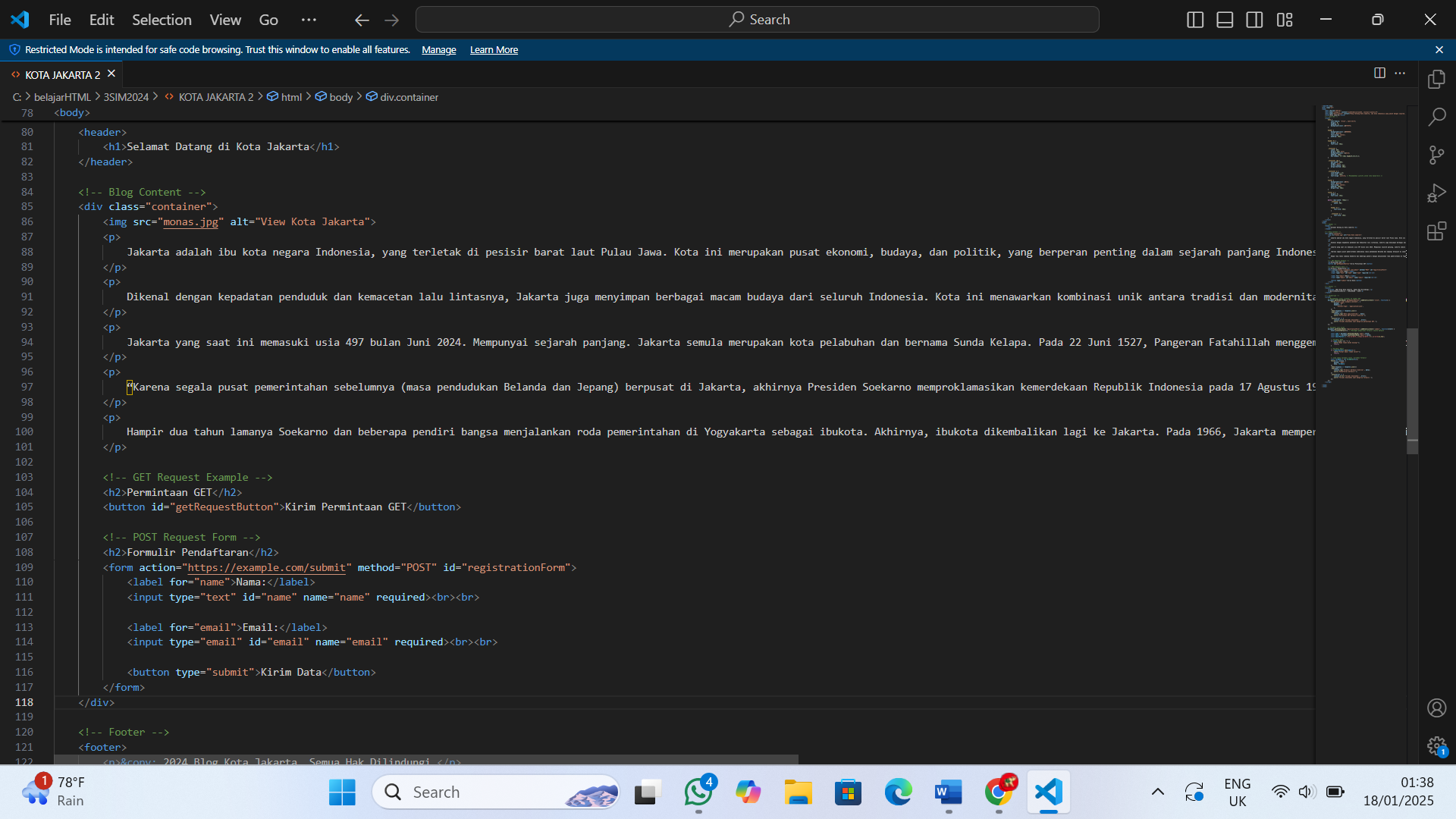
**Kapan Digunakan:**

* + Saat mengirimkan data sensitif atau melakukan perubahan pada server.
  + Contoh: Mengirim formulir pendaftaran, login, atau memproses pembelian.

**Contoh POST Request:** URL:

**Perbandingan GET vs POST**

|  |  |  |
| --- | --- | --- |
| **Aspek** | **GET** | **POST** |
| **Pengiriman Data** | Melalui URL query string | Dalam body permintaan |
| **Keamanan** | Kurang aman untuk data sensitif | Lebih aman (data tidak terlihat di URL) |
| **Batas Ukuran Data** | Terbatas panjang URL | Tidak ada batas yang ketat |
| **Caching** | Bisa di-cache oleh browser | Tidak di-cache |
| **Contoh Penggunaan** | Mencari artikel, menampilkan data | Login, registrasi, pengiriman formulir |



1. **Konsep DOM (Document Object Model)**

**Document Object Model (DOM)** adalah representasi berbasis objek dari struktur halaman web. DOM memungkinkan pengembang untuk mengakses dan memanipulasi elemen-elemen HTML dan atributnya menggunakan bahasa pemrograman seperti JavaScript.

Ketika browser memuat halaman web, ia membuat model DOM dari dokumen HTML tersebut, yang terdiri dari:

**Node**: Setiap elemen HTML, atribut, atau teks dianggap sebagai node.

**Hierarki**: Elemen-elemen HTML memiliki struktur hirarkis seperti pohon (tree structure), dengan elemen document sebagai akar dari semua node lainnya.

**Interaktivitas**: JavaScript dapat digunakan untuk mengubah elemen, atribut, gaya, atau bahkan menambahkan/menghapus elemen.

**Cara Kerja DOM**

**Akses Elemen**: DOM menyediakan metode untuk mengambil elemen HTML tertentu.

**Manipulasi**: Elemen yang diakses dapat dimanipulasi, seperti mengubah teks, gaya, atribut, atau menambahkan event listener.

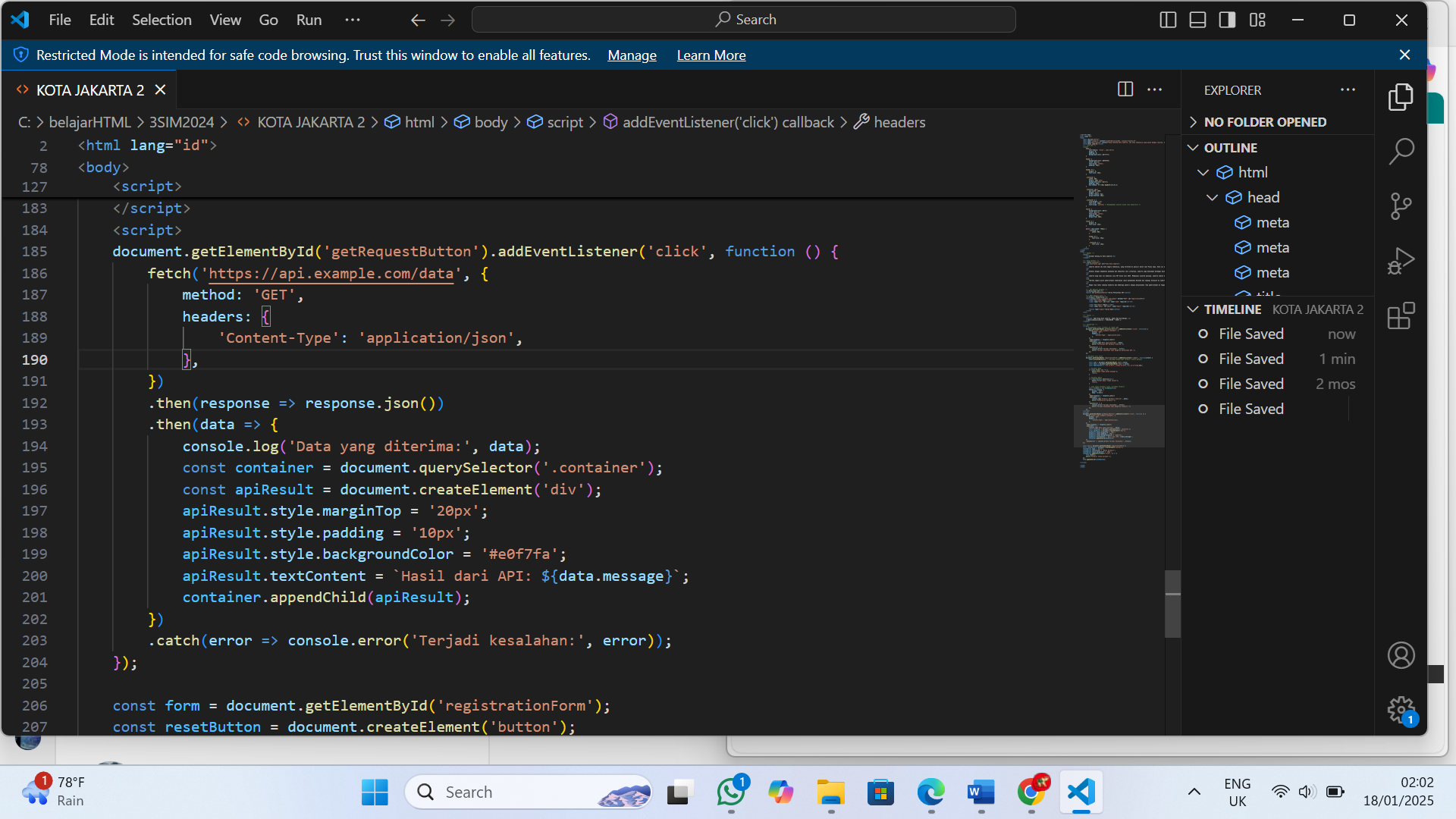
**Dinamisme:** Semua perubahan pada DOM langsung mencerminkan perubahan pada halaman web yang terlihat di browser.

**Penjelasan Contoh**

Saat tombol "Ubah Teks" diklik, teks pada elemen <h1> akan berubah dan kelas highlight akan diaktifkan atau dinonaktifkan.

Saat tombol "Tambah Paragraf" diklik, sebuah paragraf baru akan ditambahkan ke halaman.

Dengan menggunakan DOM, kita dapat membuat halaman web menjadi dinamis dan interaktif.



1. **Perbedaan antara Static Website dan Dynamic Website**

|  |  |  |
| --- | --- | --- |
| **Aspek** | **Static Website** | **Dynamic Website** |
| **Konten** | Kontennya tetap (tidak berubah) kecuali diubah secara manual oleh pengembang. | Kontennya dapat berubah berdasarkan interaksi pengguna atau data dari server. |
| **Fungsi** | Hanya menampilkan informasi statis, seperti teks dan gambar. | Memiliki fungsi yang lebih kompleks, seperti login, formulir, dashboard, dan personalisasi. |
| **Interaktivitas** | Tidak interaktif, hanya menampilkan informasi. | Interaktif, memungkinkan pengguna berinteraksi dengan halaman web. |
| **Kecepatan** | Lebih cepat karena tidak ada pengolahan data di server. | Bisa lebih lambat karena memerlukan pengolahan data di server. |
| **Pengelolaan Konten** | Konten diperbarui dengan mengedit file HTML secara manual. | Konten dikelola melalui sistem manajemen konten (CMS) atau database. |
| **Teknologi Pendukung** | Hanya memerlukan server untuk menyajikan file (tidak ada server-side processing). | Memerlukan server-side processing dan database untuk menghasilkan konten dinamis. |

Static Website

Definisi:

Static Website adalah situs web yang isinya bersifat tetap dan dihasilkan menggunakan file HTML, CSS, dan JavaScript yang disajikan langsung ke browser tanpa proses tambahan dari server.

Ciri-Ciri:

Konten disajikan persis seperti yang ada di file.

Tidak ada interaksi dengan server atau database.

Cocok untuk website sederhana seperti landing page, portofolio, atau blog kecil.

Contoh Teknologi/Bahasa Pemrograman:

* HTML, CSS, JavaScript: Untuk membuat tampilan.
* Framework Static Site Generator (SSG):
* Jekyll (berbasis Ruby)
* Hugo (berbasis Go)
* Gatsby (berbasis React)
* Hosting: GitHub Pages, Netlify, atau Vercel.

Contoh Static Website:

Halaman portofolio pengembang.

Halaman dokumentasi sederhana tanpa interaksi pengguna.

**Dynamic Website**

**Definisi:**

Dynamic Website adalah situs web yang isinya dapat berubah-ubah sesuai dengan input pengguna atau data yang diambil dari server. Kontennya sering diambil dari database dan diproses oleh server sebelum ditampilkan di browser.

**Ciri-Ciri:**

* Menggunakan server-side processing untuk menghasilkan konten.
* Kontennya dapat berbeda untuk setiap pengguna atau sesi.
* Cocok untuk aplikasi web, e-commerce, media sosial, atau platform berbasis komunitas.

**Contoh Teknologi/Bahasa Pemrograman:**

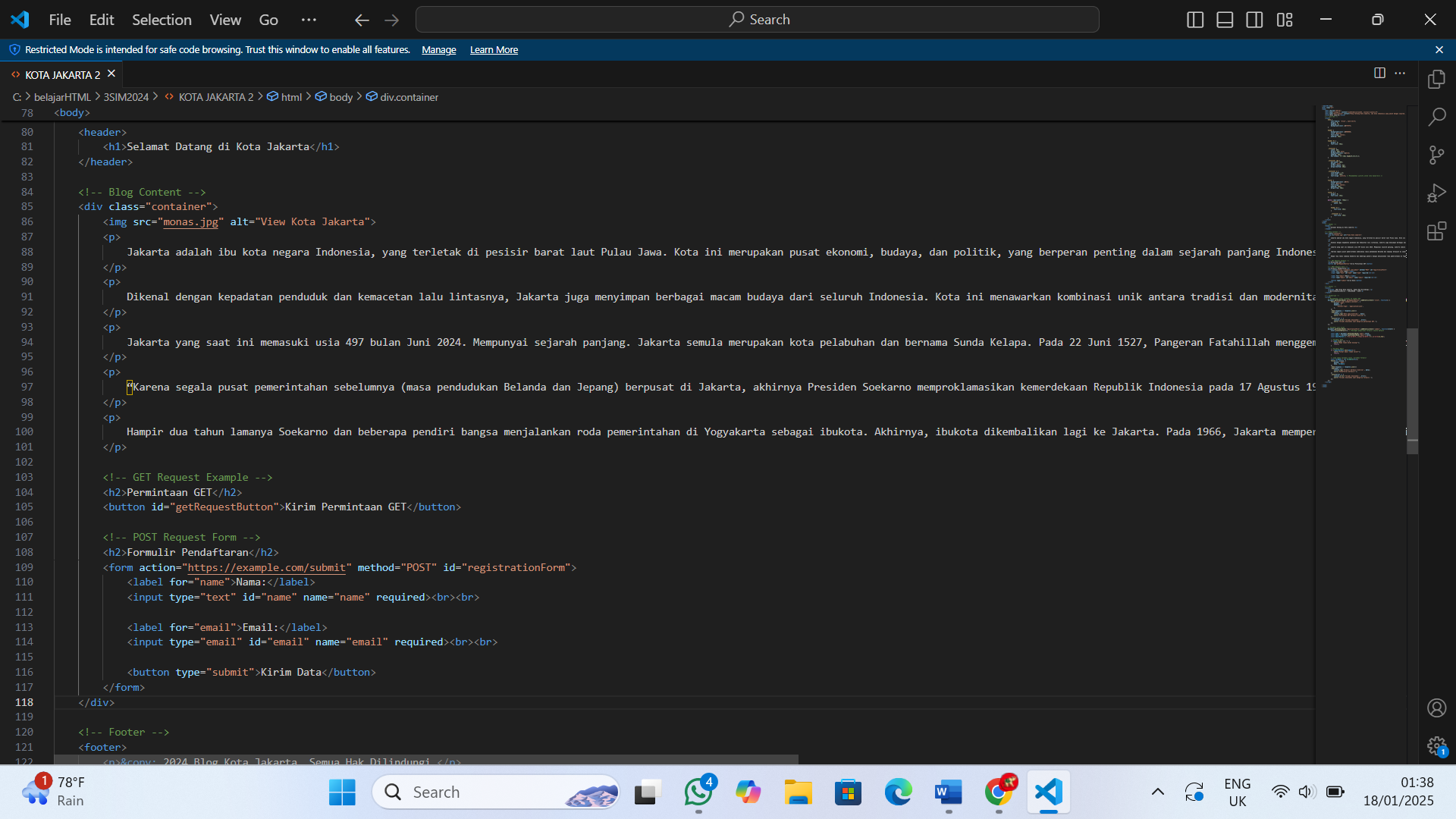
* **Server-Side Languages:**
  + PHP (digunakan oleh WordPress, Laravel)
  + Python (Framework: Django, Flask)
  + JavaScript (Runtime: Node.js)
  + Ruby (Framework: Ruby on Rails)
  + Java (Framework: Spring)
* **Database:**
  + MySQL, PostgreSQL, MongoDB, SQLite.
* **Client-Side Libraries/Frameworks:**
  + React.js, Angular, Vue.js (untuk meningkatkan interaktivitas pada sisi klien).
* **Hosting Platform:**
  + Heroku, AWS, Google Cloud, atau DigitalOcean.

**Contoh Dynamic Website:**

* E-commerce seperti Amazon atau Tokopedia.
* Media sosial seperti Facebook atau Instagram.
* Portal berita yang memperbarui konten setiap waktu.

**Static Website:** Cocok untuk situs kecil dengan konten yang jarang diperbarui.

**Dynamic Website:** Cocok untuk aplikasi web kompleks atau situs dengan konten yang sering berubah dan membutuhkan personalisasi.



1. **Cara Kerja Client-Server dalam Arsitektur Web**

Arsitektur client-server adalah model komunikasi di mana **klien** (biasanya browser) meminta layanan atau data dari **server**, dan server merespons dengan menyediakan layanan atau data tersebut. Proses ini memungkinkan pertukaran informasi dan pengiriman konten di web

**Komponen Utama dalam Arsitektur Client-Server**

1. **Klien (Client):**
   * Biasanya berupa web browser seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge.
   * Bertugas mengirim permintaan (request) ke server dan menampilkan respons (response) dari server dalam bentuk halaman web.
2. **Server:**
   * Server adalah perangkat keras atau perangkat lunak yang bertugas memproses permintaan dari klien.
   * Server mengelola data, menjalankan logika bisnis, dan mengirimkan respons (biasanya berupa file HTML, JSON, gambar, atau video).
3. **Protokol HTTP/HTTPS:**
   * HTTP (HyperText Transfer Protocol) adalah protokol komunikasi yang digunakan untuk mengatur pengiriman data antara klien dan server.
   * HTTPS adalah versi aman dari HTTP, yang mengenkripsi data menggunakan SSL/TLS.
4. **Database (Opsional):**
   * Digunakan oleh server untuk menyimpan dan mengambil data dinamis.

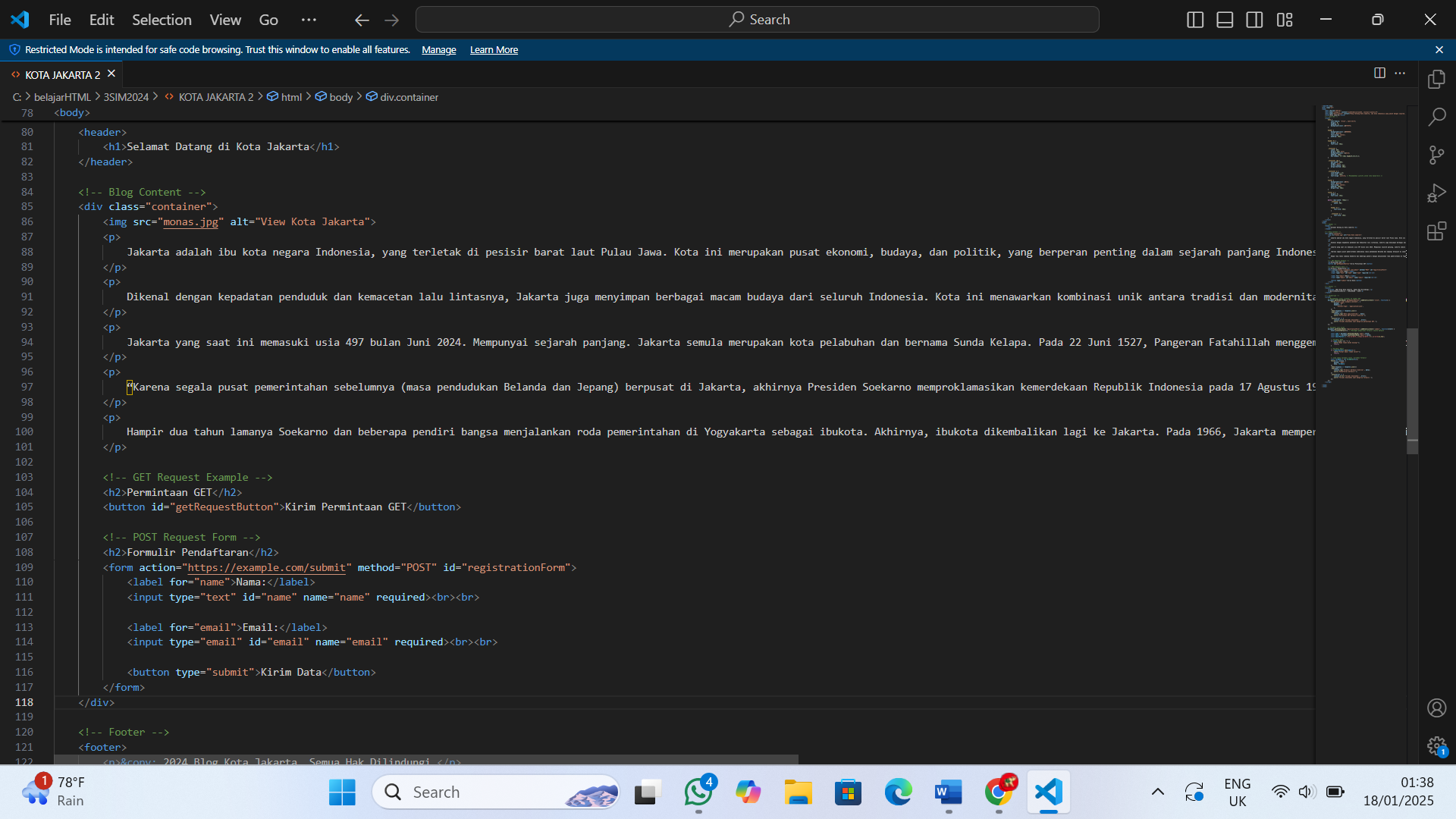
**Cara Kerja Client-Server**

Berikut adalah langkah-langkah umum bagaimana browser (klien) berkomunikasi dengan server untuk membuat halaman web:

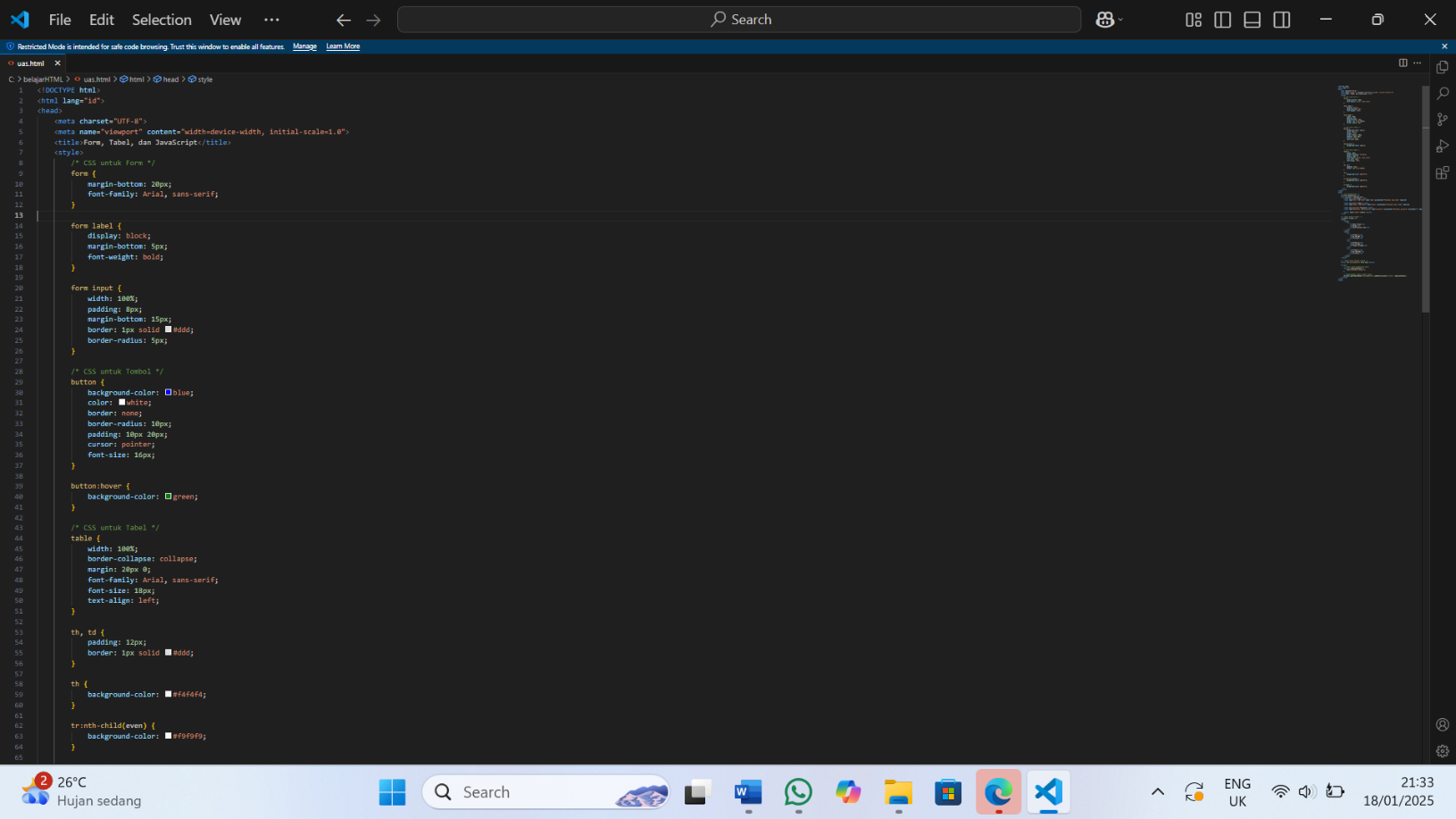
1. **DNS Request (Resolusi Domain):**
   * Browser mengubah nama domain (misalnya, www.example.com) menjadi alamat IP server melalui DNS (Domain Name System).
2. **Permintaan (Request):**
   * Browser mengirimkan permintaan HTTP/HTTPS ke server, biasanya berupa permintaan GET untuk file HTML.
   * Permintaan mencakup:
     + **Metode HTTP**: Misalnya, GET atau POST.
     + **Header**: Informasi tambahan seperti tipe browser, format data yang diterima, dll.
3. **Proses di Server:**
   * Server menerima permintaan dari klien dan memprosesnya.
   * Jika konten bersifat statis, server langsung mengirimkan file (seperti HTML, CSS, JavaScript, atau gambar).
   * Jika konten bersifat dinamis, server memproses logika bisnis (misalnya, mengambil data dari database) dan menghasilkan respons.
4. **Respons (Response):**
   * Server mengirimkan respons ke klien, biasanya berupa:
     + **Kode Status HTTP**: Misalnya, 200 (sukses), 404 (tidak ditemukan), atau 500 (kesalahan server).
     + **Konten**: File HTML, JSON, atau lainnya.
5. **Rendering di Browser:**
   * Browser menerima file HTML dan mulai memprosesnya.
   * Browser melakukan permintaan tambahan untuk file pendukung (seperti CSS, JavaScript, gambar).
   * Browser merender halaman web berdasarkan semua file yang diterima.

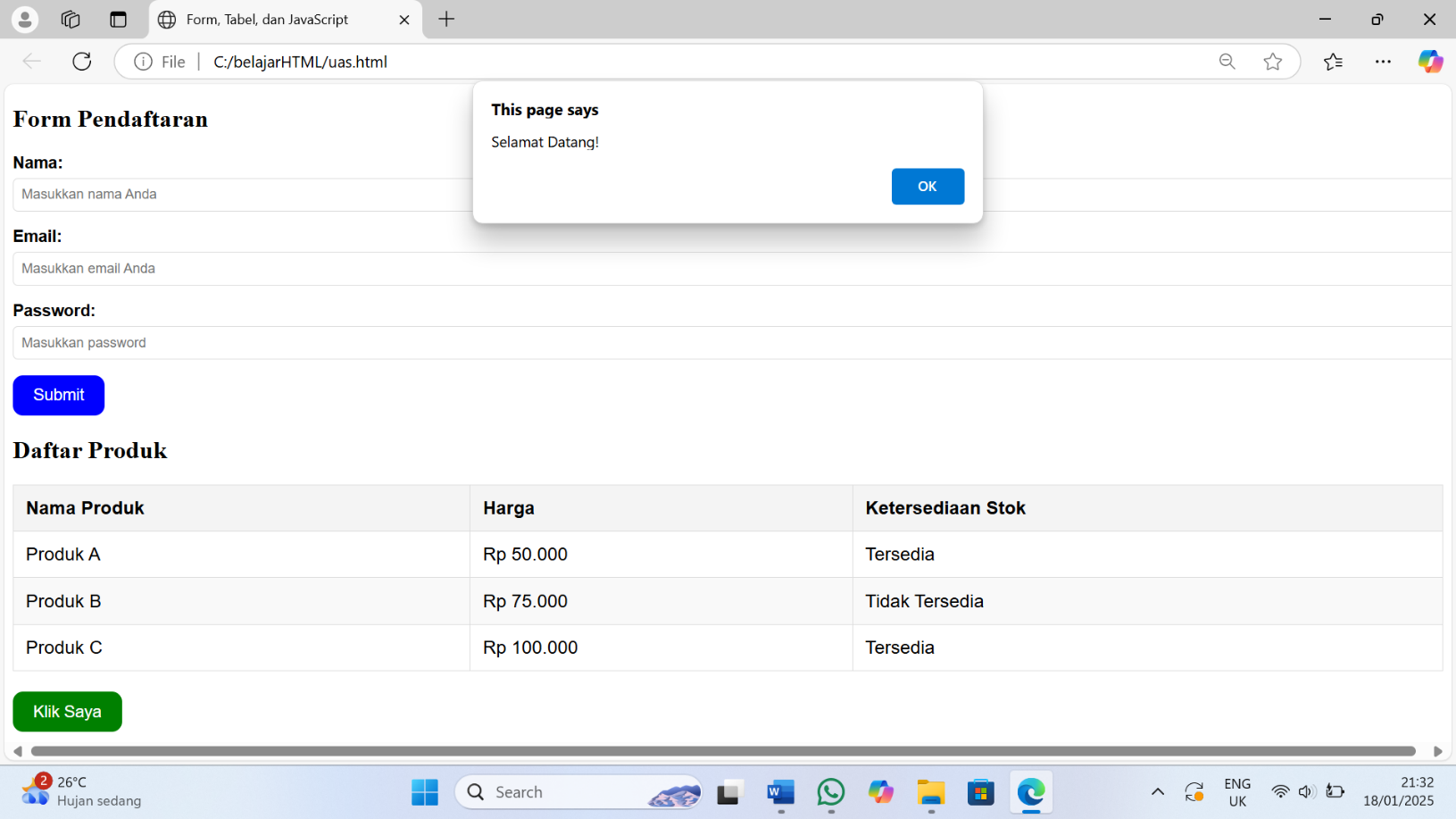
### ****Contoh Teknologi yang Terlibat:****

1. **Client-Side:**
   * **Bahasa:** HTML, CSS, JavaScript.
   * **Framework/Libraries:** React.js, Angular, Vue.js.
2. **Server-Side:**
   * **Bahasa:** PHP, Python, Ruby, Java, JavaScript (Node.js).
   * **Framework:** Laravel (PHP), Django (Python), Express.js (Node.js), Spring (Java).
3. **Database:**
   * MySQL, PostgreSQL, MongoDB.
4. **Protokol:** HTTP/HTTPS.



**Jawaban nomor 6, 7, 8, 9 dan 10**

****

****