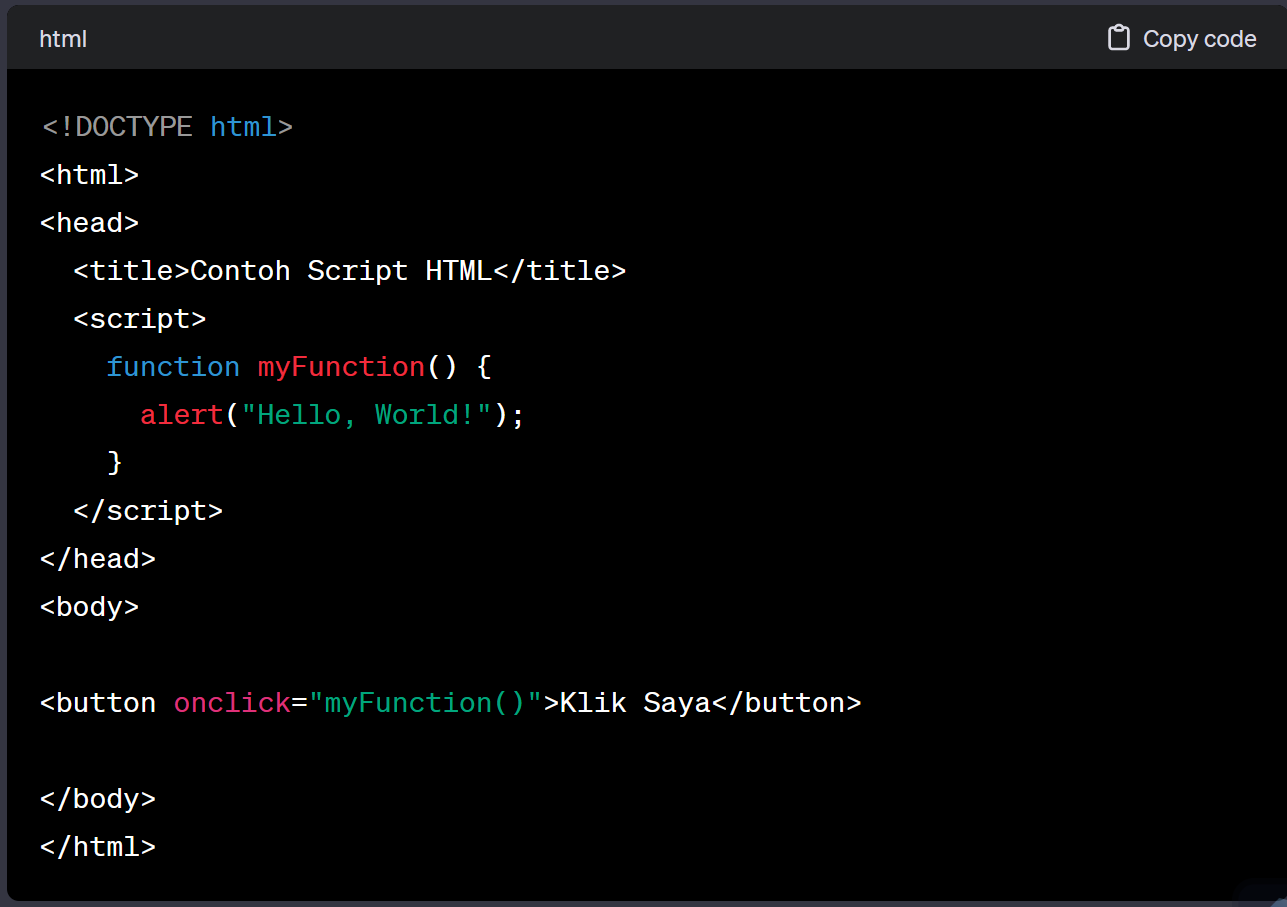
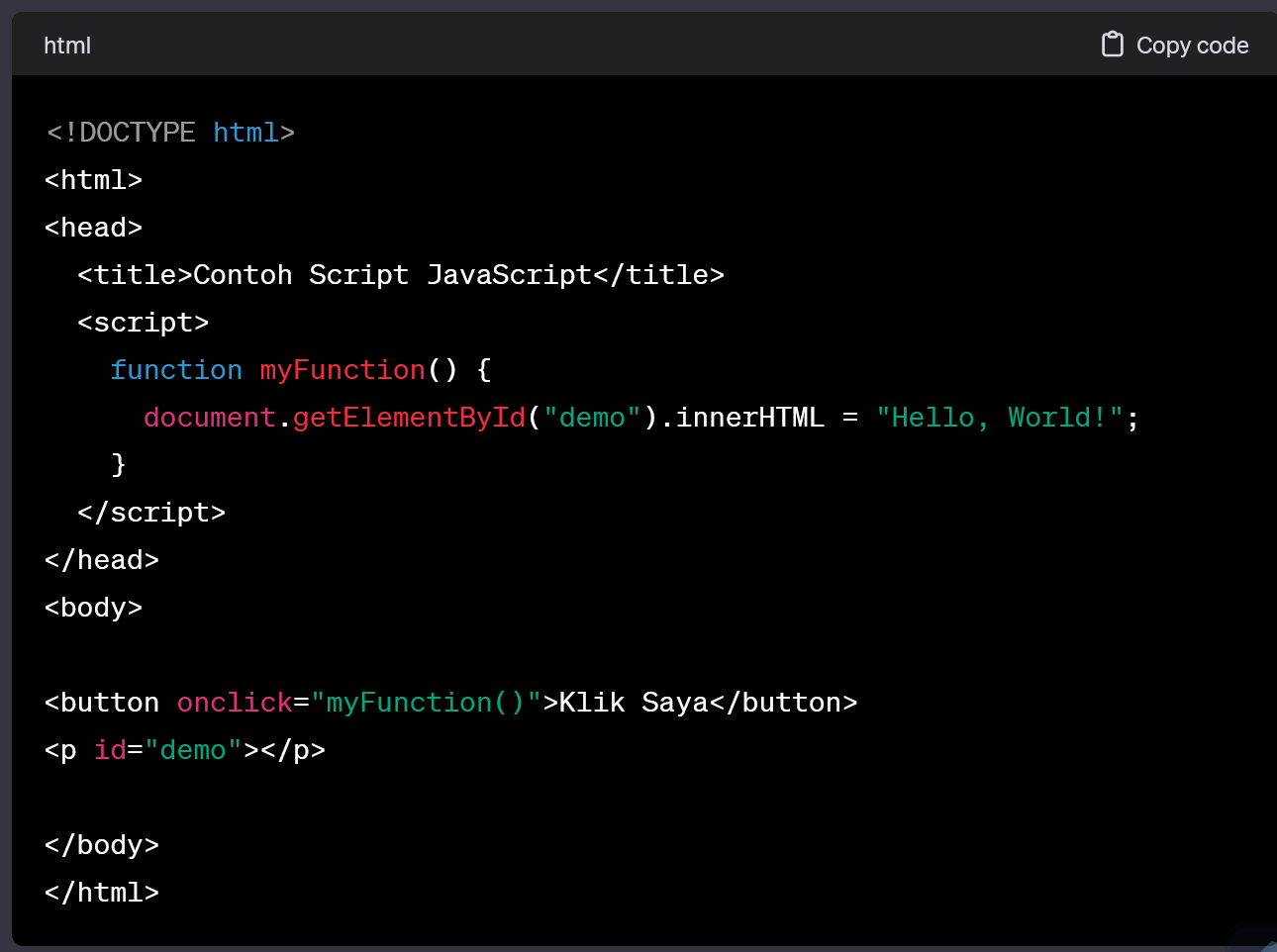
1. **Apa yang anda ketahui mengenai script pada client html, css, javascript, dan php ?**

Script pada client-side dan server-side mengacu pada kode yang dieksekusi di sisi klien (pada perangkat pengguna) dan di sisi server (pada server web), masing-masing. Berikut adalah penjelasan singkat mengenai script pada HTML, CSS, JavaScript, dan PHP:

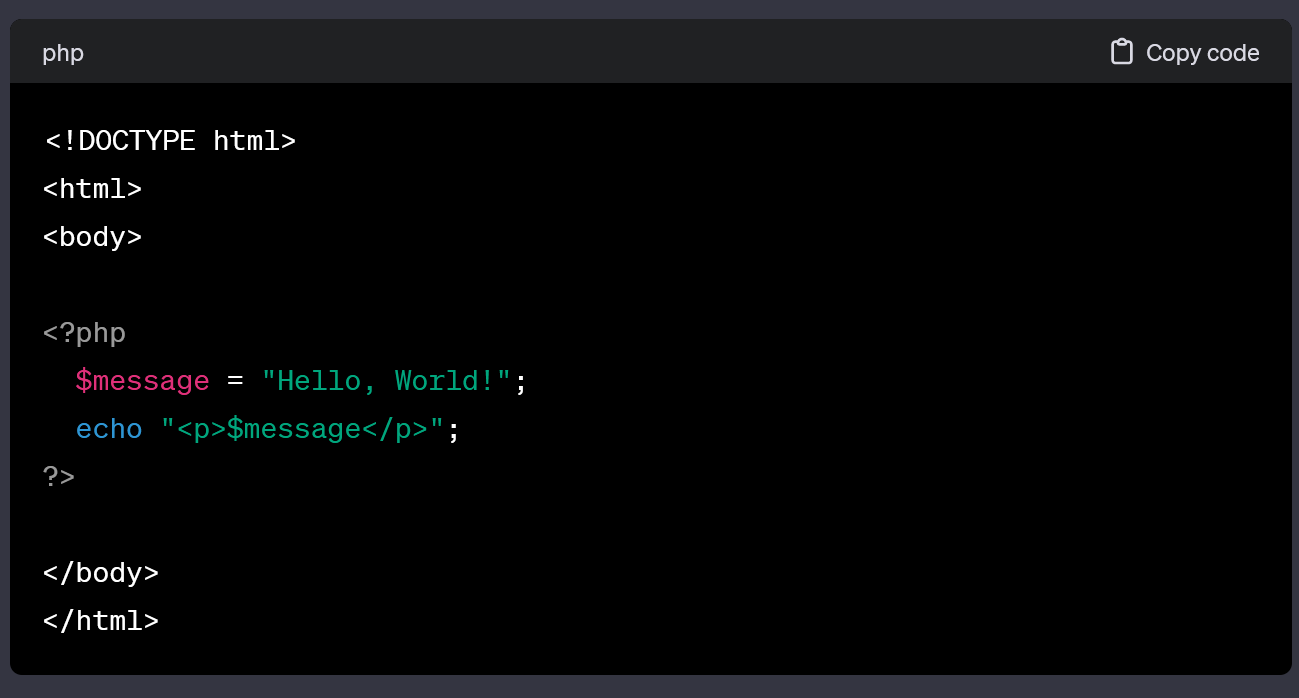
1. **HTML (HyperText Markup Language):**
   1. HTML adalah bahasa markup yang digunakan untuk membuat struktur dasar sebuah halaman web.
   2. Dalam HTML, script client-side biasanya merujuk pada penggunaan elemen <script>. Namun, peran utama HTML adalah menyediakan struktur dan konten untuk halaman web.
   3. Contoh penggunaan script HTML (JavaScript) pada client-side:



1. **CSS (Cascading Style Sheets):**
   1. CSS digunakan untuk mengatur tata letak dan gaya presentasi halaman web.
   2. Secara umum, CSS adalah bahasa deklaratif, dan tidak dianggap sebagai "script" seperti HTML atau JavaScript.
2. **JavaScript:**
   1. JavaScript adalah bahasa pemrograman yang digunakan untuk membuat interaksi dinamis pada halaman web.
   2. JavaScript dapat dieksekusi di sisi klien, yang berarti kode dieksekusi pada perangkat pengguna.
   3. Contoh penggunaan JavaScript pada client-side:



1. **PHP (Hypertext Preprocessor):**
   1. PHP adalah bahasa skrip yang umumnya digunakan di sisi server untuk membuat halaman web yang dinamis.
   2. Script PHP dieksekusi di server sebelum hasilnya dikirimkan ke perangkat pengguna.
   3. Contoh penggunaan PHP:

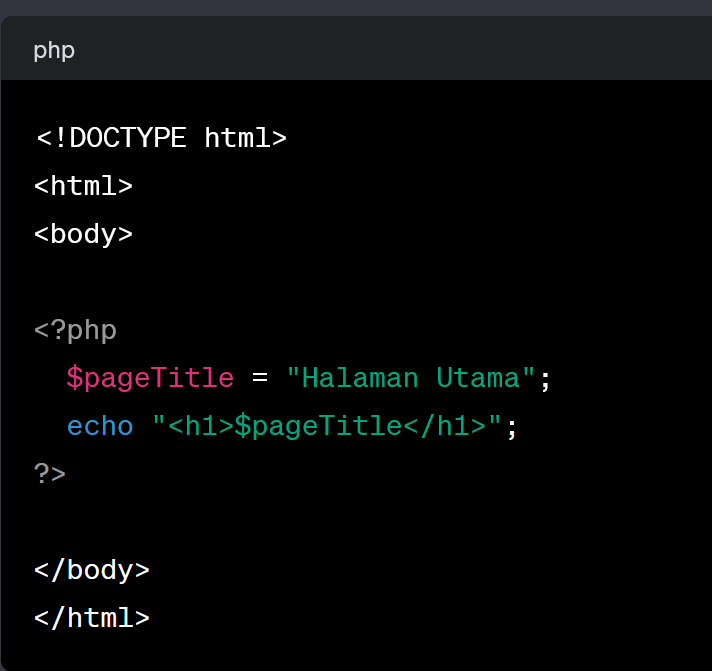


1. **Apa yang anda ketahui script pada server pada html, css, javascript, php dan database mysqldb?**

Script pada sisi server melibatkan teknologi dan bahasa pemrograman yang dijalankan di server web. Berikut adalah penjelasan singkat mengenai penggunaan script pada server-side dengan HTML, CSS, JavaScript, PHP, dan MySQL database:

1. **HTML pada Server-side:**
   * Meskipun HTML sebagian besar dieksekusi di sisi klien, beberapa teknologi server-side, seperti PHP, dapat menghasilkan HTML dinamis berdasarkan kondisi atau data dari server.

Contoh menggunakan PHP untuk membuat HTML pada server-side:



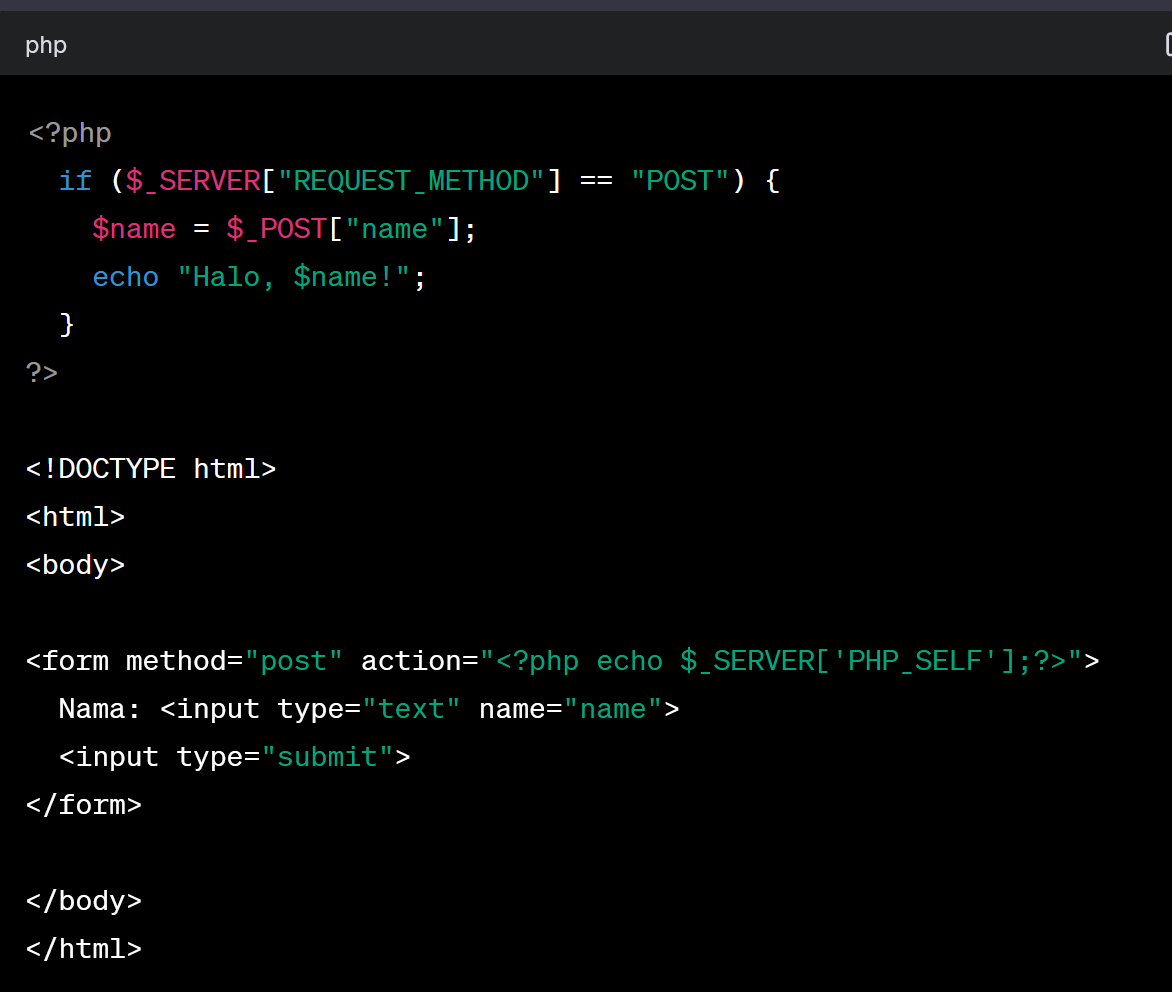
1. **CSS pada Server-side:**
   * CSS umumnya dianggap sebagai bagian dari desain tampilan dan diaplikasikan pada elemen HTML di sisi klien. Server-side scripting, seperti PHP, dapat memodifikasi nilai-nilai CSS berdasarkan logika bisnis.
2. **JavaScript pada Server-side:**
   * Node.js adalah platform JavaScript server-side yang memungkinkan eksekusi kode JavaScript di server. Ini memungkinkan pengembang untuk menulis kode JavaScript di sisi server.

Contoh penggunaan JavaScript (Node.js) pada server-side:



1. **PHP:**
   * PHP adalah bahasa skrip server-side yang dirancang khusus untuk pengembangan web.
   * PHP memungkinkan Anda untuk melakukan berbagai tugas server-side, seperti mengakses database, memproses formulir, dan menghasilkan halaman dinamis.

Contoh penggunaan PHP untuk memproses formulir pada server-side:



1. **MySQL Database:**
   * MySQL adalah sistem manajemen basis data (DBMS) yang banyak digunakan untuk menyimpan dan mengelola data di server.
   * PHP dapat berinteraksi dengan MySQL untuk melakukan operasi database, seperti penambahan, pengambilan, pembaruan, dan penghapusan data.

Contoh penggunaan PHP untuk berinteraksi dengan MySQL:



1. **Jelaskan apa yang kalian ketahui mengenai webserver sebagai penyaji dan pengontrol ?**

Web server adalah perangkat lunak atau perangkat keras yang berfungsi sebagai penyaji (server) untuk mengakomodasi permintaan dan memberikan respons terhadap klien yang terhubung. Web server bertanggung jawab untuk menangani protokol HTTP (Hypertext Transfer Protocol) atau HTTPS (HTTP Secure) dan menyajikan halaman web kepada pengguna melalui browser. Ini berperan sebagai pengontrol dalam artian bahwa mengelola dan mengirimkan berbagai jenis konten, mulai dari halaman HTML sederhana hingga gambar, file CSS, JavaScript, dan berbagai sumber daya web lainnya.

Berikut adalah beberapa konsep dan elemen penting terkait web server:

1. **Penyaji Konten Statis vs. Dinamis:**
   * Web server dapat menyajikan konten statis dan/atau dinamis. Konten statis adalah file yang tidak berubah dan disajikan langsung tanpa pemrosesan tambahan. Konten dinamis, di sisi lain, melibatkan pemrosesan oleh server sebelum disajikan, seperti ketika menggunakan bahasa server-side scripting seperti PHP.
2. **HTTP/HTTPS:**
   * Web server berkomunikasi dengan klien menggunakan protokol HTTP atau HTTPS. HTTPS (HTTP Secure) menambahkan lapisan keamanan dengan menggunakan enkripsi SSL/TLS untuk melindungi data yang dikirimkan antara klien dan server.
3. **Apache, Nginx, dan Lainnya:**
   * Ada berbagai jenis web server yang tersedia, termasuk Apache, Nginx, Microsoft IIS, dan lainnya. Masing-masing memiliki karakteristik, kelebihan, dan kelemahan sendiri.
4. **Request dan Response:**
   * Klien (biasanya browser) mengirimkan permintaan ke server untuk mendapatkan halaman web atau sumber daya lainnya. Web server menghasilkan respons yang dikirimkan kembali kepada klien. Respons ini mungkin berisi data HTML, gambar, CSS, JavaScript, atau jenis konten lainnya.
5. **Virtual Hosting:**
   * Web server dapat dikonfigurasi untuk mendukung virtual hosting, yang memungkinkan satu server fisik untuk menyajikan banyak situs web yang terpisah dengan nama domain yang berbeda.
6. **Load Balancing:**
   * Beberapa web server mendukung teknik load balancing, di mana permintaan dari klien didistribusikan di antara beberapa server fisik atau virtual. Ini membantu meningkatkan kinerja dan ketersediaan.
7. **Log dan Analisis:**
   * Web server biasanya mencatat aktivitasnya dalam file log. File log ini mencatat informasi seperti alamat IP klien, permintaan yang dibuat, respons server, dan informasi lain yang dapat digunakan untuk analisis dan pemantauan.
8. **Keamanan:**
   * Keamanan web server penting untuk melindungi data sensitif dan mencegah serangan potensial. Ini melibatkan konfigurasi yang aman, pembaruan perangkat lunak secara teratur, dan implementasi protokol keamanan seperti SSL/TLS.
9. **Task – task pada html !**
10. **Menyusun Struktur Dasar:**

Menggunakan elemen <html>, <head>, dan <body> untuk menentukan struktur dasar halaman web.

1. **Menambahkan Teks dan Paragraf:**

Menggunakan elemen <p>, <h1>, <h2>, dll., untuk menambahkan teks dan paragraf ke halaman.

1. **Menyisipkan Gambar:**

Menggunakan elemen <img> untuk menyisipkan gambar ke halaman.

1. **Membuat Tautan:**

Menggunakan elemen <a> untuk membuat tautan (link) ke halaman lain atau ke sumber daya lainnya.

1. **Membuat Daftar:**

Menggunakan elemen <ul> untuk membuat daftar tak terurut, dan <ol> untuk membuat daftar terurut.

1. **Membuat Formulir:**

Menggunakan elemen <form> untuk membuat formulir interaktif dengan berbagai elemen input, seperti <input>, <textarea>, dan <select>.

1. **Menyisipkan Video dan Audio:**

Menggunakan elemen <video> dan <audio> untuk menyisipkan konten multimedia ke halaman.

1. **Menggunakan Elemen Semantik:**

Menggunakan elemen semantik seperti <header>, <nav>, <article>, <section>, dan <footer> untuk memberikan makna dan struktur yang lebih kaya pada halaman.

1. **Menyisipkan Script:**

Menggunakan elemen <script> untuk menyisipkan kode JavaScript ke halaman.

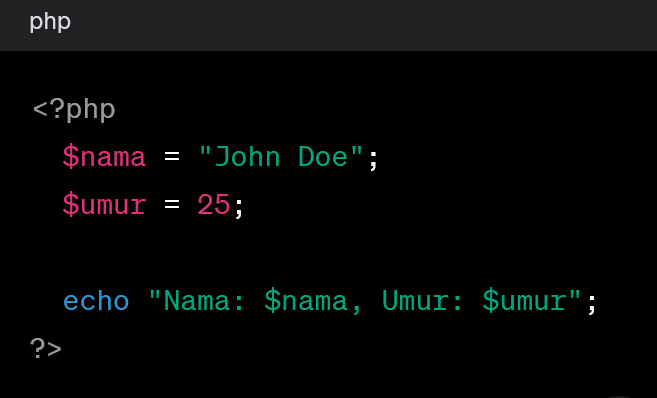
1. **Menggunakan Meta Tags:**

Menggunakan elemen <meta> untuk menyediakan informasi meta-data, seperti deskripsi halaman, karakter set, dan instruksi lainnya.

1. **Variabel pada php, $ pada php, cara menyambungkan dengan html yaitu pada bagian body !**

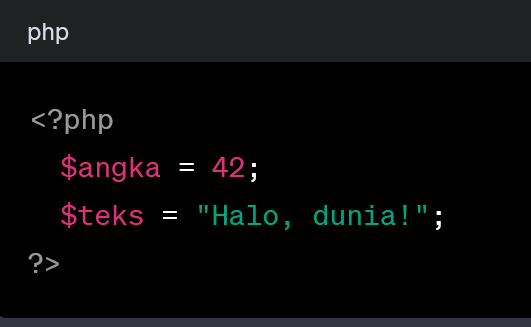
**Variabel pada PHP:** Variabel digunakan untuk menyimpan nilai dan memberikan nama pada nilai tersebut. Di PHP, variabel dimulai dengan tanda dolar $ diikuti dengan nama variabelnya. Nama variabel di PHP bersifat case-sensitive.

Contoh penggunaan variabel dalam PHP:



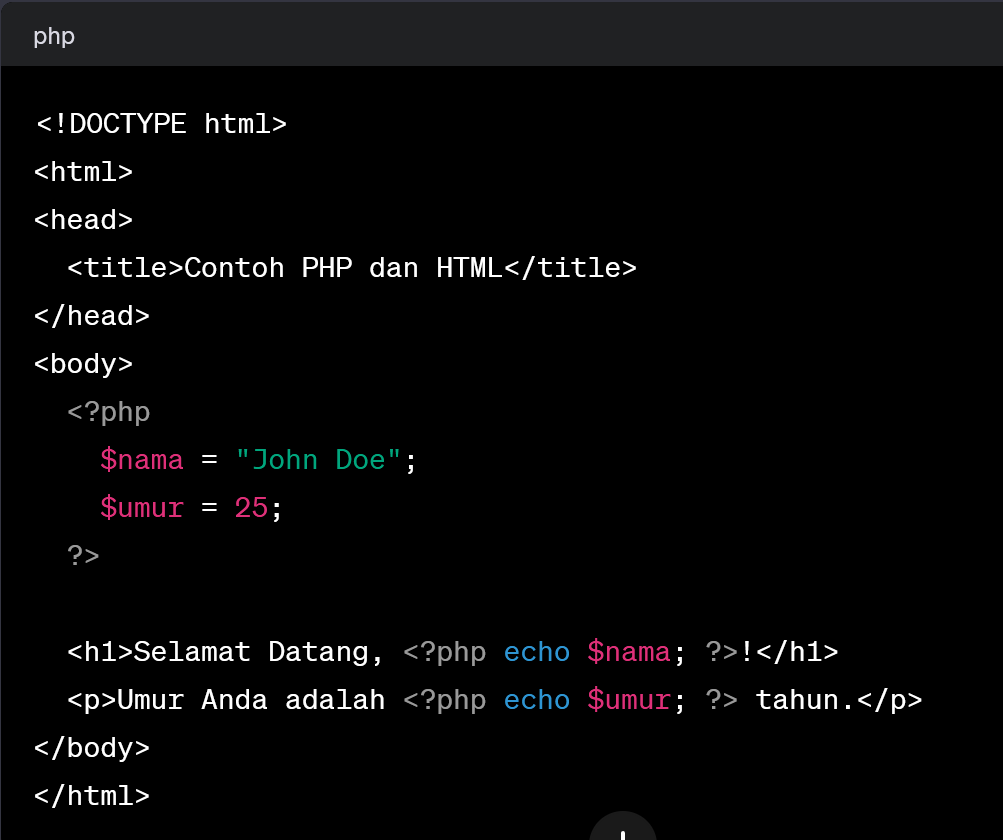
**Tanda Dolar ($) pada PHP:** Tanda dolar $ digunakan untuk mendeklarasikan dan mengakses variabel dalam PHP. Ini memberi tahu interpreter PHP bahwa kita sedang bekerja dengan variabel.

Contoh:



**Menyambungkan PHP dengan HTML dalam Bagian Body:** Untuk menyisipkan nilai dari variabel PHP ke dalam bagian body HTML, kita dapat menggunakan sintaks PHP di dalam tag HTML.

Contoh:



Pada contoh di atas, nilai dari variabel $nama dan $umur disisipkan ke dalam tag HTML di bagian body menggunakan sintaks PHP (<?php echo $nama; ?> dan <?php echo $umur; ?>). Ini memungkinkan kita untuk dinamis menampilkan nilai variabel PHP di dalam halaman HTML.

1. Unsur yang harus ada diweb logo, nama brand / perusahaan !

Beberapa unsur kunci yang sebaiknya ada di situs web, terutama pada halaman beranda, termasuk logo, nama brand/perusahaan, dan elemen lainnya. Berikut adalah penjelasan singkat mengenai unsur-unsur tersebut:

1. **Logo:**
   * **Fungsi:** Logo adalah representasi visual dari brand atau perusahaan. Ini adalah elemen identitas yang membedakan brand dari yang lain.
   * **Karakteristik:**
     + **Simpel dan Memorable:** Logo sebaiknya simpel agar mudah diingat dan dikenali.
     + **Relevan dengan Brand:** Logo sebaiknya mencerminkan nilai, visi, atau karakteristik utama dari brand.
     + **Skalabilitas:** Logo harus terlihat baik dalam berbagai ukuran, baik itu besar di halaman beranda atau kecil di bilah navigasi.
   * **Tempat di Web:** Logo sering ditempatkan di bagian kiri atas halaman web, berfungsi sebagai tautan kembali ke halaman utama, dan dapat ditempatkan di bagian kepala (header) atau di bilah navigasi.
2. **Nama Brand/Perusahaan:**
   * **Fungsi:** Nama brand atau perusahaan memberikan identitas dan mengkomunikasikan apa yang ditawarkan oleh brand tersebut.
   * **Karakteristik:**
     + **Jelas dan Mudah Dipahami:** Nama brand sebaiknya jelas dan mudah dimengerti oleh target audiens.
     + **Konsisten:** Penggunaan nama brand harus konsisten di seluruh situs web dan materi pemasaran.
   * **Tempat di Web:** Nama brand sering ditempatkan bersama logo di bagian kepala halaman web atau di bilah navigasi.
3. **Elemen Visual Pendukung:**
   * **Fungsi:** Elemen visual seperti gambar, grafik, atau elemen desain lainnya dapat digunakan untuk memperkaya tampilan halaman web dan memberikan nuansa visual yang konsisten dengan identitas brand.
   * **Karakteristik:**
     + **Relevan:** Elemen visual harus relevan dengan industri atau niche brand.
     + **Konsisten:** Penggunaan elemen visual harus konsisten dengan gaya dan warna brand.
   * **Tempat di Web:** Elemen visual dapat ditempatkan di berbagai bagian halaman web tergantung pada tujuannya, seperti slider beranda, gambar produk, atau latar belakang bagian tertentu.
4. Elemen yang harus ada diwebsite menu, layout (bottom dan top )

**Elemen yang Harus Ada di Menu Website:**

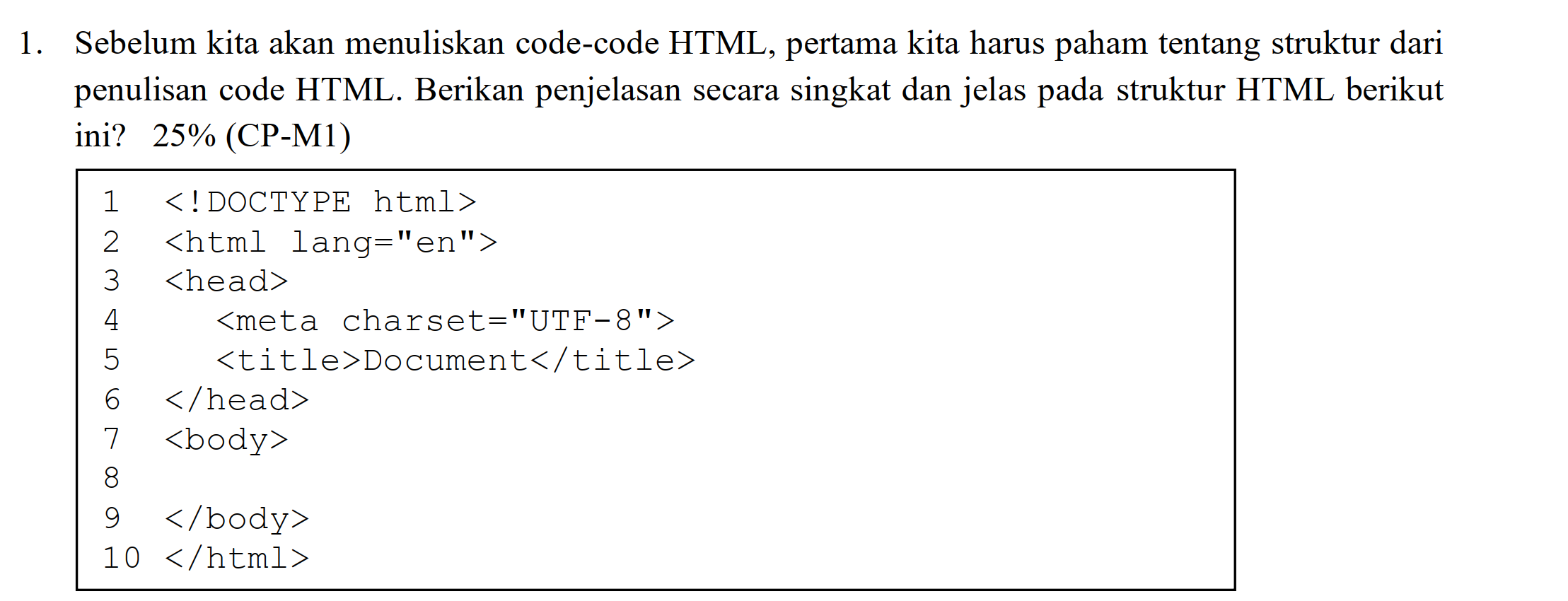
1. **Beranda (Home):** Tautan untuk kembali ke halaman beranda utama.
2. **Tentang Kami (About Us):** Informasi tentang perusahaan, sejarah, visi, dan misi.
3. **Produk atau Layanan (Products/Services):** Daftar produk atau layanan yang ditawarkan oleh perusahaan.
4. **Portofolio atau Galeri (Portfolio/Gallery):** Showcase karya atau proyek terbaru yang telah diselesaikan.
5. **Kontak (Contact):** Informasi kontak, formulir kontak, atau peta lokasi.
6. **Blog:** Jika situs web memiliki blog, tautan ke halaman blog.
7. **FAQ (Frequently Asked Questions):** Pertanyaan yang sering diajukan beserta jawabannya.
8. **Testimoni:** Ulasan atau testimonial dari pelanggan sebelumnya.
9. **Login atau Daftar (Login/Sign Up):** Jika situs memerlukan akun pengguna.
10. **Kebijakan Privasi (Privacy Policy):** Tautan ke halaman kebijakan privasi situs.
11. **Syarat dan Ketentuan (Terms and Conditions):** Tautan ke halaman syarat dan ketentuan penggunaan situs.

**Elemen yang Harus Ada di Layout Top (Header):**

1. **Logo:** Representasi visual dari brand atau perusahaan.
2. **Navigasi Menu:** Tautan menu utama yang mengarahkan ke berbagai bagian situs.
3. **Pencarian (Search):** Pencarian internal jika situs membutuhkan fungsi pencarian.
4. **Tombol Aksi (CTA - Call to Action):** Tombol yang mengarahkan pengguna untuk melakukan tindakan tertentu, seperti "Beli Sekarang" atau "Hubungi Kami."

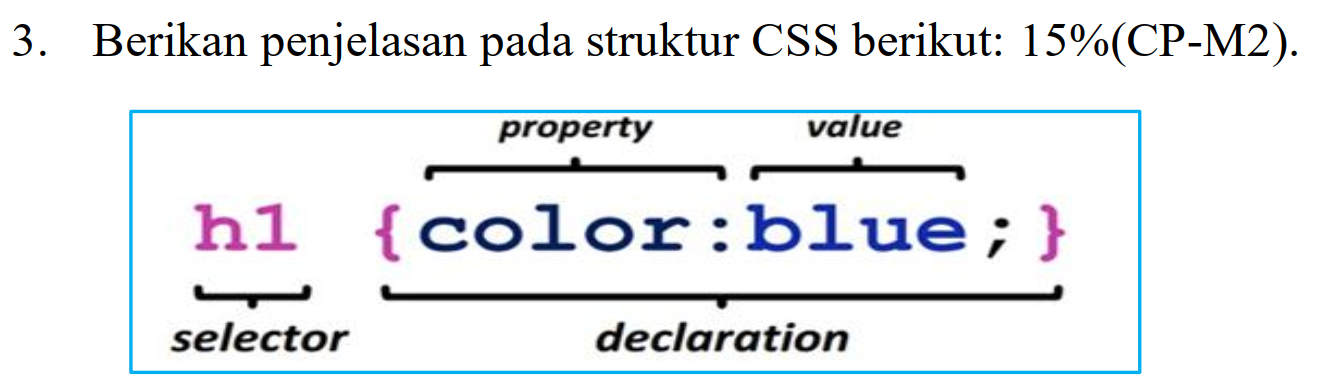
**Elemen yang Harus Ada di Layout Bottom (Footer):**

1. **Navigasi Footer:** Tautan tambahan ke halaman-halaman penting.
2. **Informasi Kontak:** Alamat, nomor telepon, dan alamat email perusahaan.
3. **Formulir Langganan (Subscription Form):** Jika situs memiliki bulletin atau langganan berita.
4. **Tautan Sosial Media:** Tautan ke halaman media sosial perusahaan.
5. **Peta Situs (Sitemap):** Daftar lengkap tautan ke semua halaman situs.
6. **Kebijakan Privasi dan Syarat & Ketentuan:** Tautan ke halaman kebijakan privasi dan syarat & ketentuan.



Script HTML di atas adalah kerangka dasar untuk membuat halaman web. Berikut adalah penjelasan dari setiap bagian:

1. **<!DOCTYPE html>:**
   * Ini adalah deklarasi tipe dokumen (Document Type Declaration atau DOCTYPE). Ini memberi tahu browser bahwa dokumen akan mengikuti standar HTML5.
2. **<html lang="en">:**
   * Tag pembuka untuk elemen root, yang menunjukkan bahwa kontennya adalah dokumen HTML. Atribut lang="en" menunjukkan bahwa bahasa yang digunakan dalam dokumen adalah bahasa Inggris (English).
3. **<head>:**
   * Bagian ini berisi informasi tentang dokumen, seperti pengaturan karakter, viewport, judul halaman, dan tautan ke stylesheet eksternal atau skrip JavaScript.
   * **<meta charset="UTF-8">:**
     + Mendeklarasikan karakter set dokumen sebagai UTF-8, yang mendukung karakter internasional.
   * **<meta name="viewport" content="width=device-width, initial-scale=1.0">:**
     + Mendefinisikan pengaturan tampilan untuk perangkat seluler. Pengaturan ini mengatakan kepada browser untuk menggunakan lebar perangkat sebagai lebar viewport awal dan mengizinkan zoom awal.
   * **<title>Document</title>:**
     + Ini adalah judul dokumen yang akan muncul di tab atau judul jendela browser.
4. **<body>:**
   * Bagian ini berisi konten utama halaman web, seperti teks, gambar, formulir, dan elemen-elemen lainnya. Semua elemen yang terlihat di halaman web ditempatkan di dalam tag <body>.



Mari kita jelaskan setiap bagian dari struktur CSS yang diberikan:

1. **h1 - Selector:**
   * Ini adalah elemen HTML yang dipilih untuk menerima gaya tertentu. Dalam contoh ini, h1 adalah elemen heading level 1.
2. **color - Property:**
   * Ini adalah properti CSS yang menentukan warna teks dari elemen yang dipilih. Properti ini mengendalikan warna teks elemen.
3. **blue - Value:**
   * Ini adalah nilai yang diberikan kepada properti color. Dalam hal ini, nilai adalah "blue", yang akan memberikan warna teks biru pada elemen h1.
4. **{color: blue;} - Declaration:**
   * Ini adalah deklarasi atau aturan CSS lengkap yang terdiri dari selector, properti, dan nilai. Setiap deklarasi diapit oleh tanda kurung kurawal {}.

**AJAX**

AJAX (Asynchronous JavaScript and XML) adalah suatu teknik pemrograman web yang memungkinkan pengambilan atau pengiriman data ke server secara asynchronous (tanpa perlu me-refresh seluruh halaman). Dengan menggunakan AJAX, halaman web dapat berkomunikasi dengan server di belakang layar dan memperbarui hanya sebagian kecil dari halaman tanpa mengharuskan pengguna memuat ulang seluruh halaman.

Beberapa konsep utama dalam AJAX melibatkan penggunaan JavaScript, objek XMLHttpRequest, dan manipulasi DOM (Document Object Model). Berikut adalah beberapa konsep dasar dalam penggunaan AJAX:

1. **Asynchronous Communication:**

AJAX memungkinkan komunikasi asynchronous antara browser dan server. Ini berarti klien (browser) dapat mengirim permintaan ke server dan melanjutkan bekerja tanpa harus menunggu respon dari server.

1. **XMLHttpRequest Object:**

Objek XMLHttpRequest adalah inti dari AJAX. Ini memungkinkan pembuatan permintaan HTTP ke server dan penanganan responsnya. Contoh penggunaan sederhana seperti berikut:



**3. Callback Functions:**AJAX menggunakan callback functions untuk menangani respons dari server setelah permintaan selesai. Callback akan dipanggil ketika status permintaan berubah.

**4. Manipulasi DOM:**Setelah mendapatkan respons dari server, JavaScript dapat digunakan untuk memanipulasi DOM dan memperbarui tampilan halaman tanpa me-refresh seluruh halaman.

**5. Data Format (XML/JSON):**Data yang dikirimkan antara browser dan server bisa berupa XML, JSON, atau format lainnya. JSON umumnya lebih populer karena lebih mudah dipahami dan diproses oleh JavaScript.

**6. Cross-Origin Resource Sharing (CORS):**Kebijakan Same-Origin Policy (SOP) mencegah permintaan AJAX ke domain yang berbeda. Untuk mengatasi ini, server perlu mengizinkan permintaan dari domain yang berbeda melalui CORS.

**7. Framework dan Library:**Ada berbagai framework dan library JavaScript yang mempermudah penggunaan AJAX, seperti jQuery, Axios, dan Fetch API yang merupakan bagian dari standar JavaScript modern.

**DOM**

DOM, atau Document Object Model, adalah representasi struktural dari dokumen HTML atau XML yang dihasilkan oleh browser ketika halaman web dimuat. DOM memungkinkan program atau skrip JavaScript berinteraksi dengan elemen-elemen halaman web dan mengubah struktur, gaya, dan kontennya secara dinamis.Berikut adalah beberapa konsep dasar yang terkait dengan DOM:

**1. Pohon Struktur:**

DOM mengorganisir elemen-elemen halaman web dalam bentuk pohon struktur, di mana setiap elemen adalah node dalam pohon tersebut. Dokumen itu sendiri adalah node root.

**2. Node:**

Semua elemen, atribut, dan teks dalam dokumen dianggap sebagai node dalam DOM.

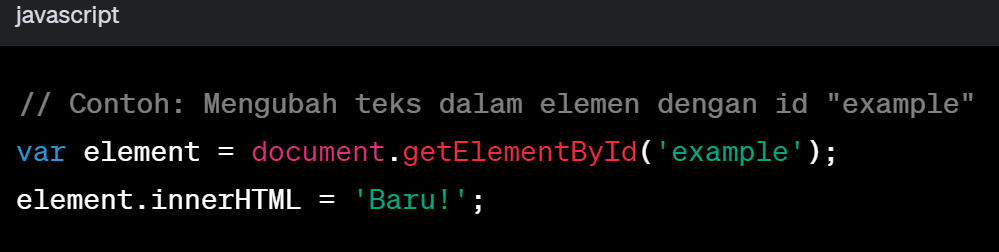
* Elemen Node: Mewakili elemen HTML (seperti <div>, <p>, dll.).
* Atribut Node: Mewakili atribut dari elemen.
* Teks Node: Mewakili teks dalam elemen.

**3. Parent, Child, dan Sibling Nodes:**

* Parent Node: Node yang berada satu tingkat di atas suatu node.
* Child Node: Node yang berada satu tingkat di bawah suatu node.
* Sibling Node: Node yang memiliki parent yang sama.

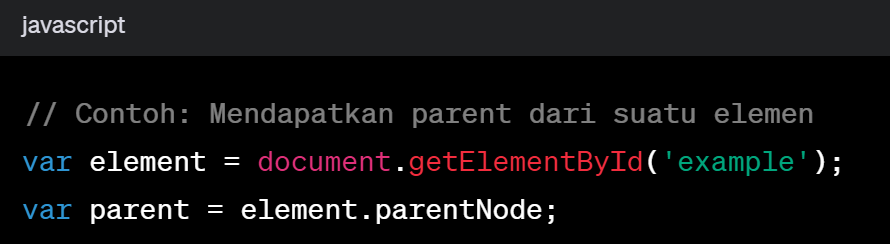
**4. Akses dan Manipulasi:**

JavaScript dapat menggunakan API DOM untuk mengakses, memanipulasi, atau mengubah elemen-elemen DOM.



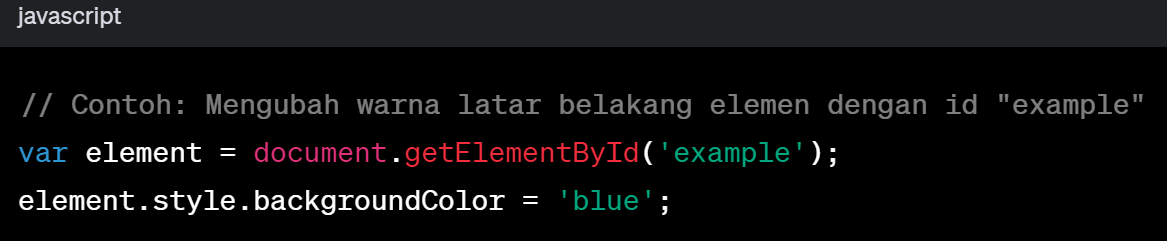
**5. Traversal (Traversing):**

Memungkinkan perjalanan melalui node-nodenya, seperti menemukan parent, child, atau sibling dari suatu node.



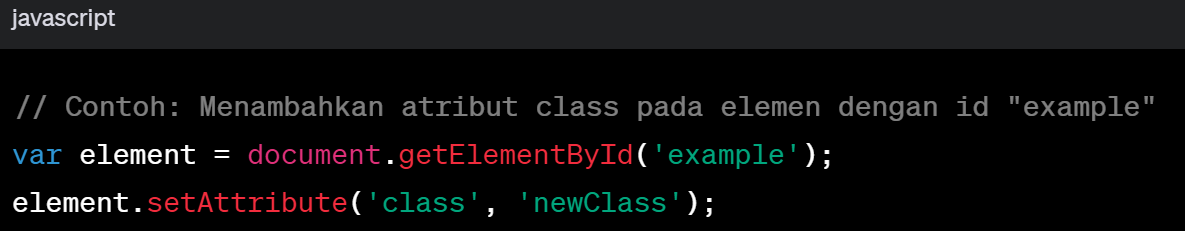
**6. Manipulasi Gaya (Styling):**

Mengubah atau memanipulasi gaya elemen, seperti warna, ukuran font, dll.



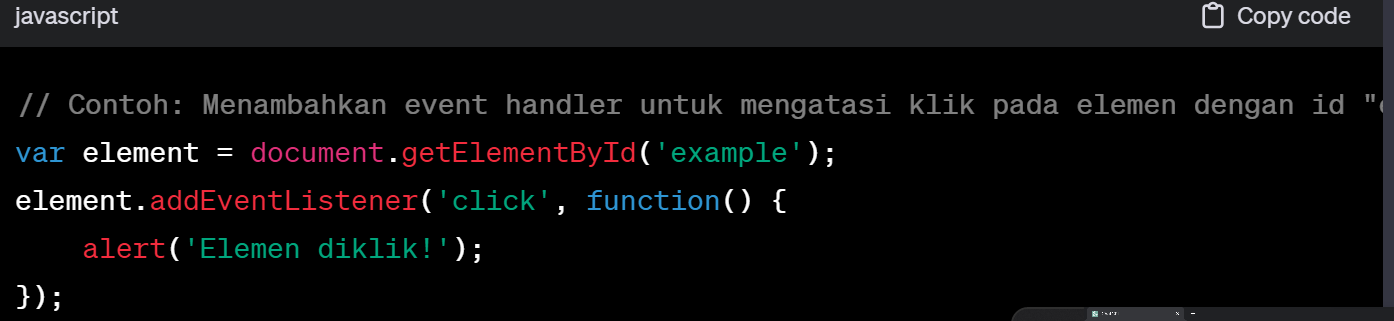
**7. Manipulasi Atribut:**

Mengubah atau memanipulasi atribut elemen.



**8. Event Handling:**

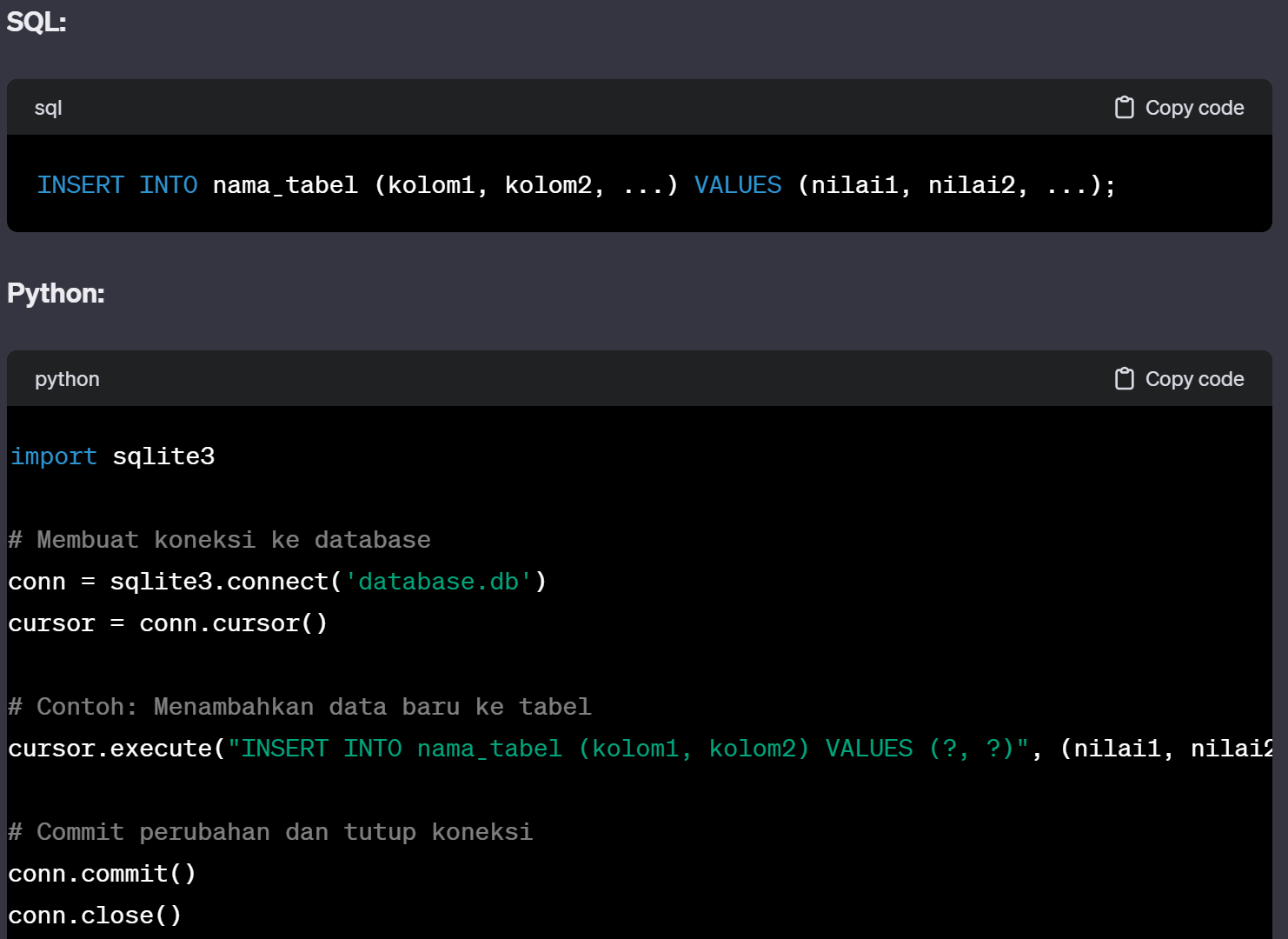
Memungkinkan menetapkan event handler untuk menanggapi interaksi pengguna.



**CRUD (SQLite3)**

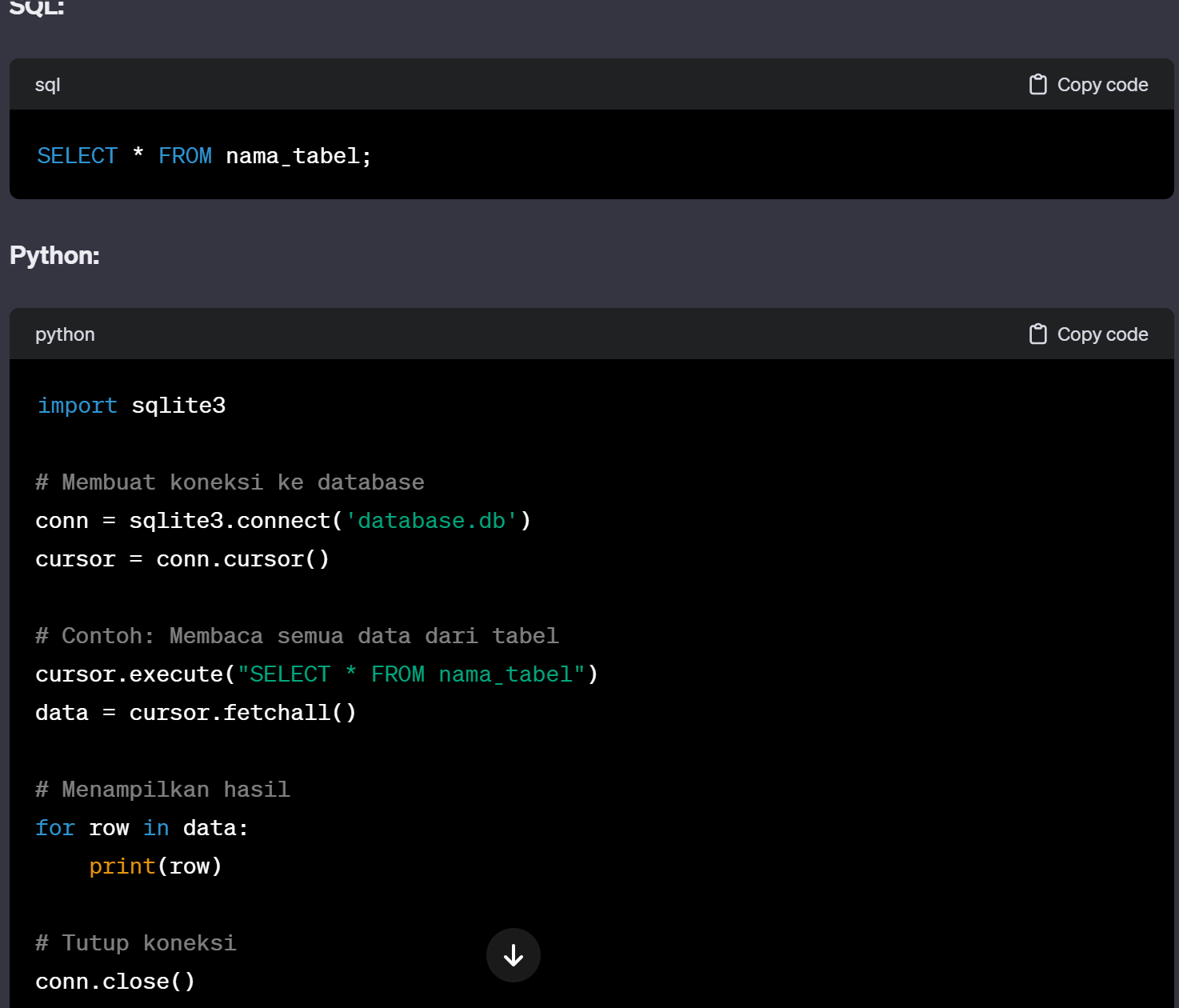
* 1. **Create**

Membuat data baru



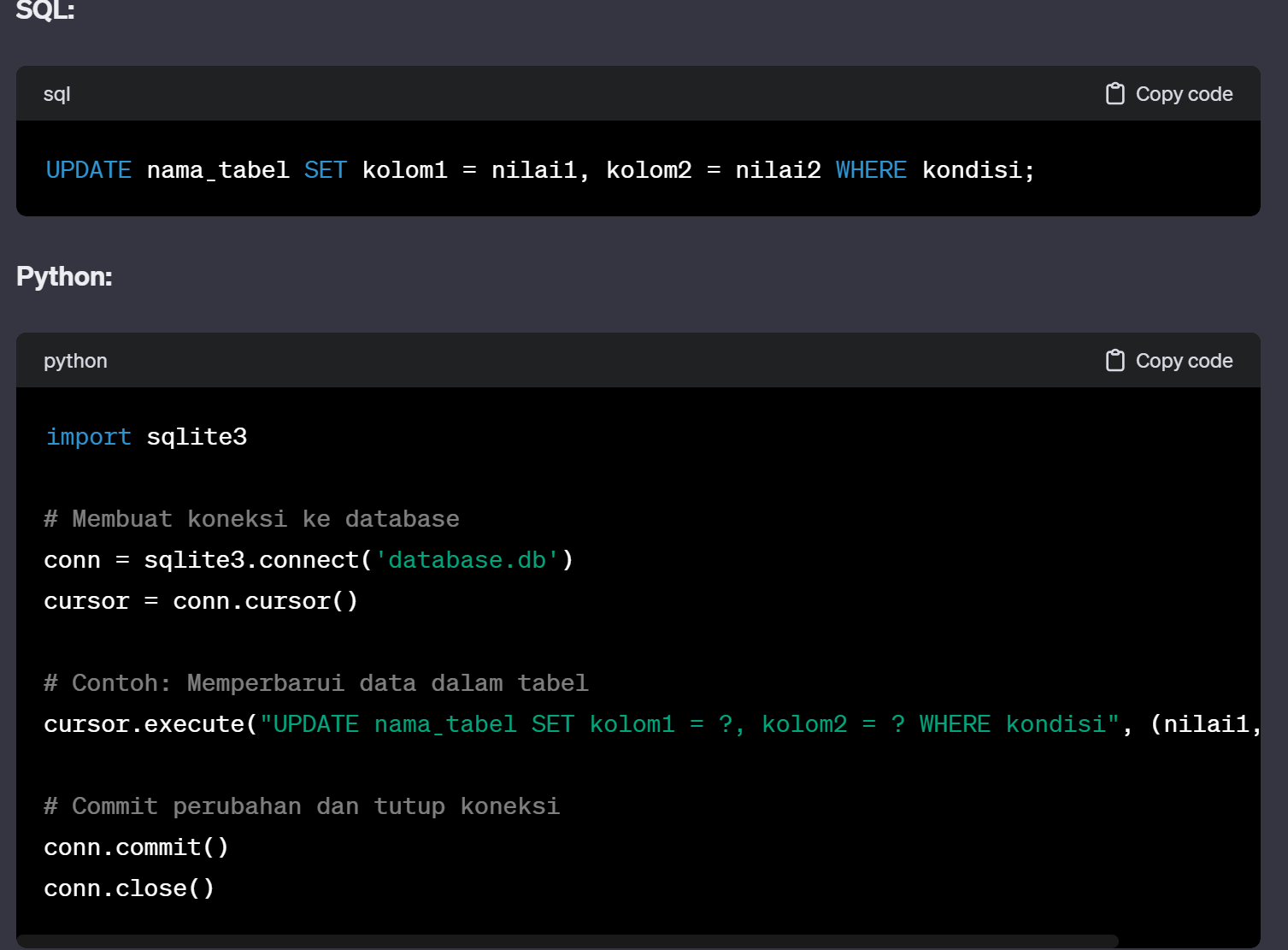
* 1. **Read**

Membaca data



* 1. **Update**

Memperbarui data

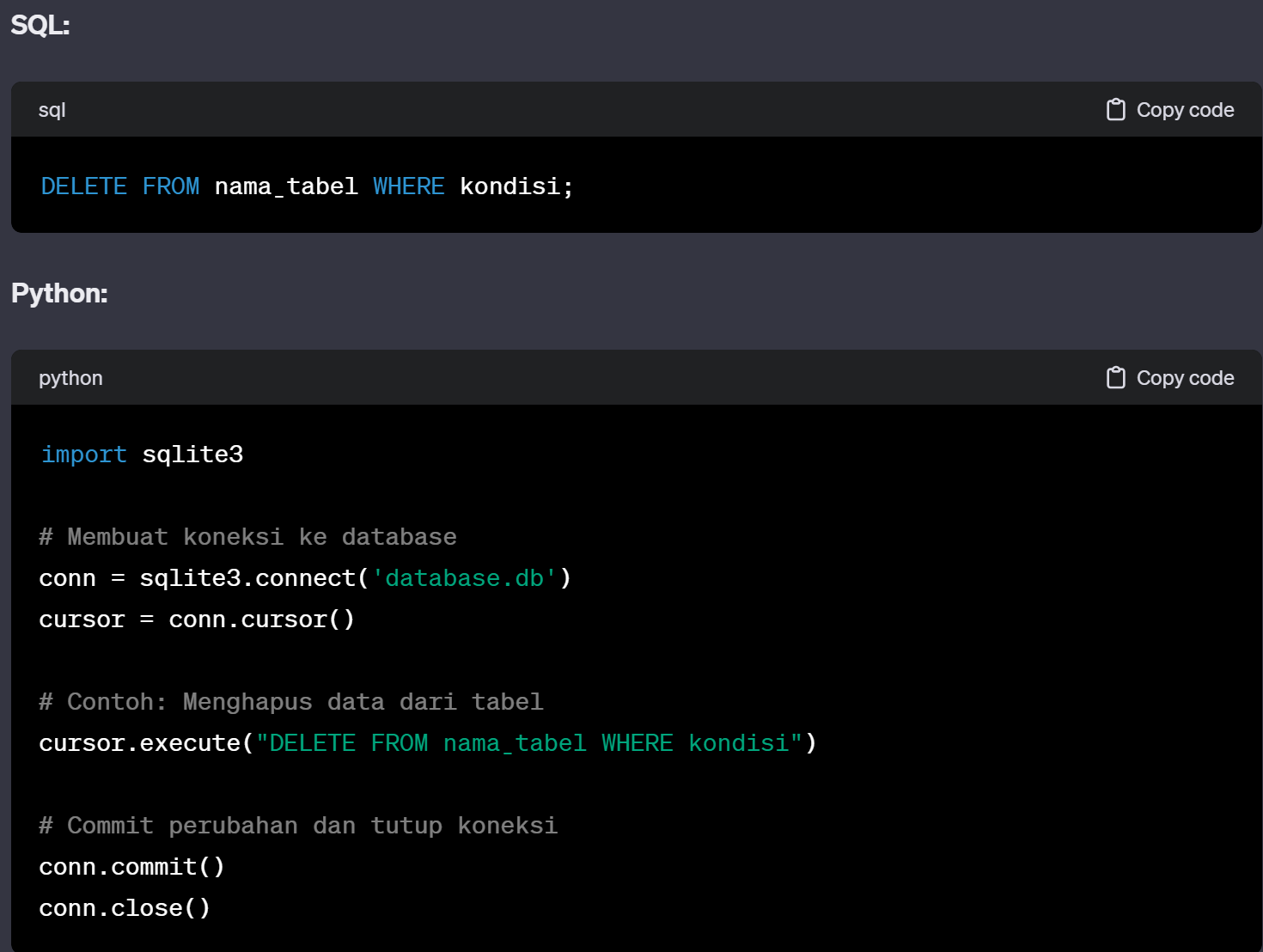
****

Skrip kepotong:

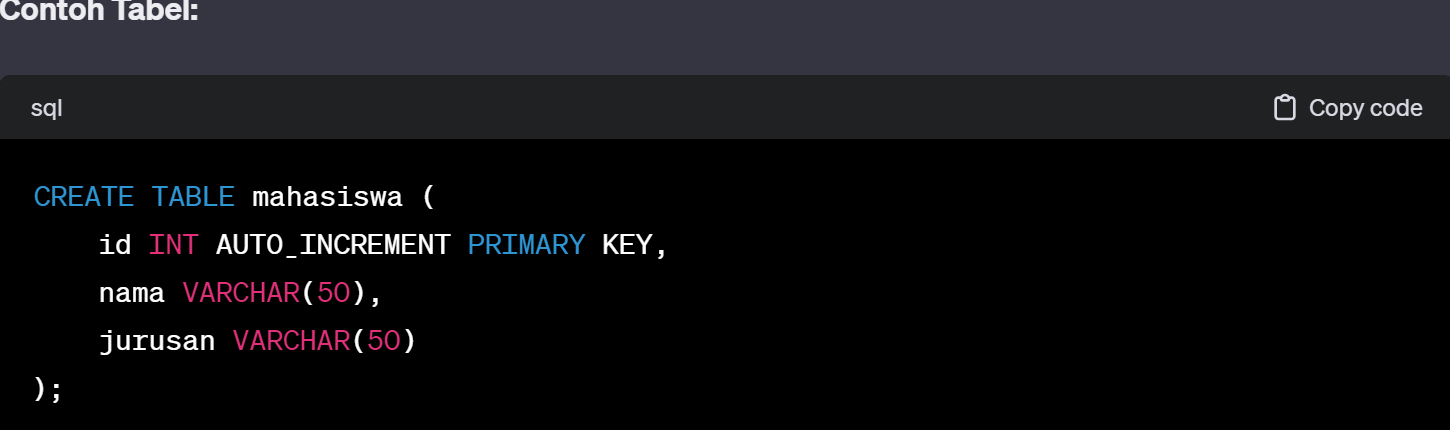
cursor.execute("UPDATE nama\_tabel SET kolom1 = ?, kolom2 = ? WHERE kondisi", (nilai1, nilai2))

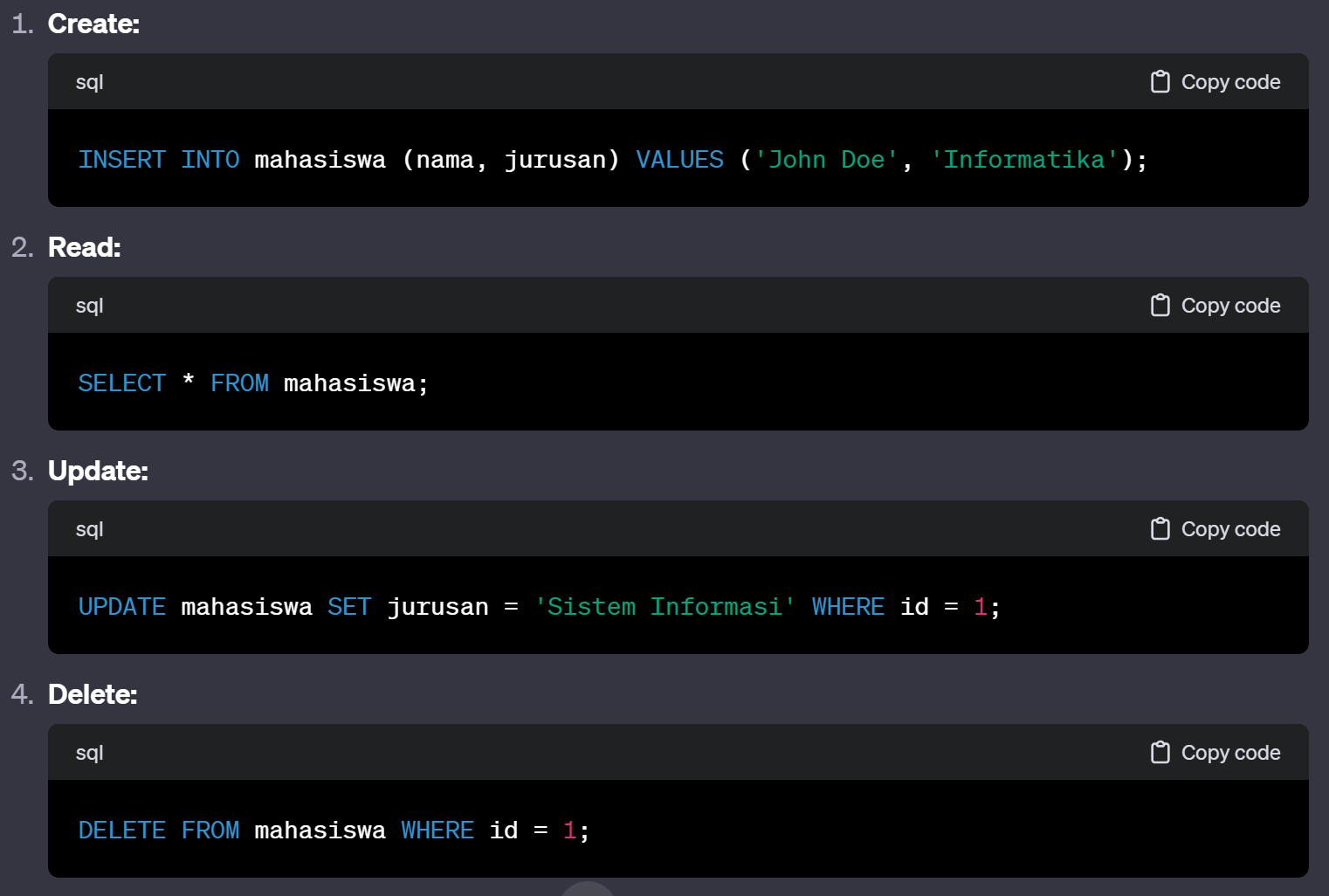
* 1. **Remove**

Menghapus data



**CRUD (My SQL)**

****

****

Implementasi

import mysql.connector

*# Koneksi ke database*

conn = mysql.connector.connect(

    host="localhost",

    user="username",

    password="password",

    database="nama\_database"

)

cursor = conn.cursor()

*# Create (Buat)*

cursor.execute("INSERT INTO mahasiswa (nama, jurusan) VALUES (%s, %s)", ("John Doe", "Informatika"))

conn.commit()

*# Read (Baca)*

cursor.execute("SELECT \* FROM mahasiswa")

result = cursor.fetchall()

for row in result:

    print(row)

*# Update (Perbarui)*

cursor.execute("UPDATE mahasiswa SET jurusan = %s WHERE id = %s", ("Sistem Informasi", 1))

conn.commit()

*# Delete (Hapus)*

cursor.execute("DELETE FROM mahasiswa WHERE id = %s", (1,))

conn.commit()

*# Menutup koneksi*

cursor.close()

conn.close()

Perbedaan Clint-side dan Server-side

