

Ikhwan Alfath NF Aplikasi Sistem Web dan Seluler (Mobile)

Pengantar dan Sejarah Pemrograman Web

Prodi Teknologi Informasi | Fakultas Teknik | Universitas Tidar

Aplikasi Sistem Web

#1 Materi Sebelum UTS

Pengantar dan Sejarah
Pemrograman Web



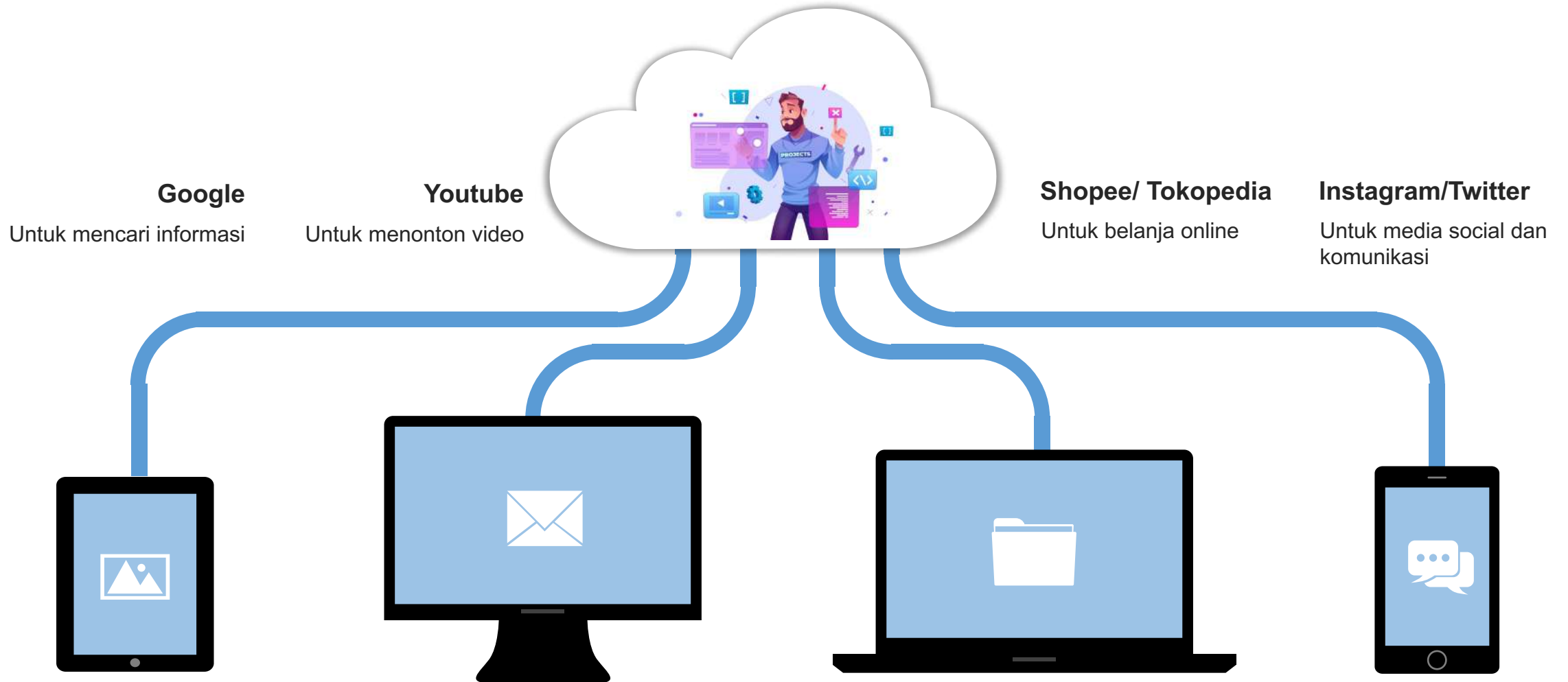


Teknologi di era digital

PEMROGRAMAN WEB
CLOUD UNTIDAR

Aplikasi Website?

Contoh Website



Pengertian

Website?

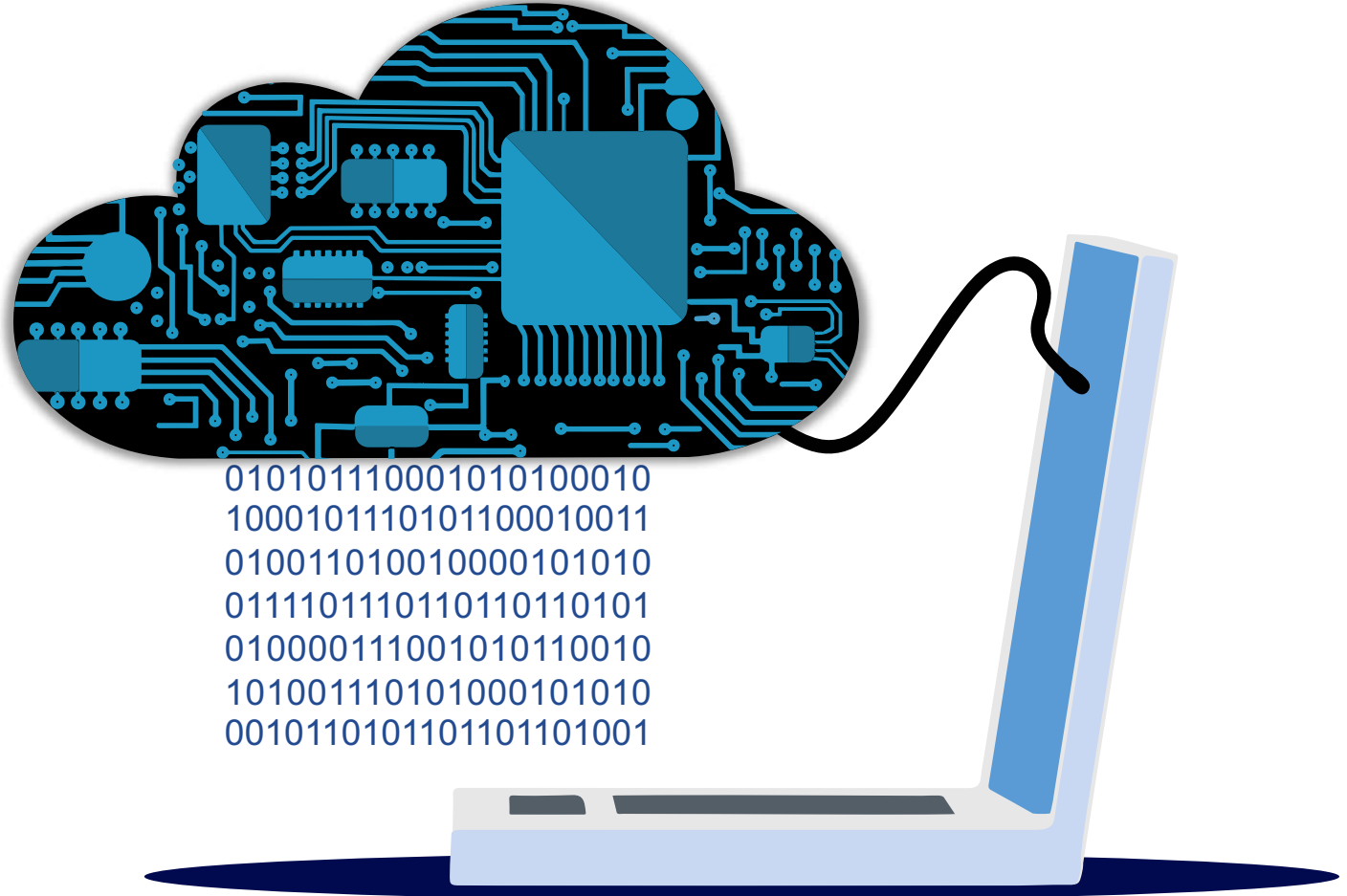
Apa itu Aplikasi Sistem Web?

Pengertian

Sederhananya, **aplikasi sistem web** merupakan **aplikasi yang berjalan melalui browser** dan bisa **diakses dari berbagai perangkat melalui internet**.

Tidak perlu diinstal seperti aplikasi mobile atau desktop!

- ◆ **Mengirim permintaan** ke server Google.
- ◆ **Server memproses permintaan** dan mencari hasil yang sesuai.
- ◆ **Server mengirimkan kembali** hasil pencarian yang ditampilkan di browser pengguna.



Sejarah Perkembangan



Era Web Statis (1990-an)

Pada era ini, website masih bersifat **statis**, artinya halaman web hanya menampilkan **teks dan gambar** tanpa adanya interaksi dari pengguna. Website dibuat menggunakan **HTML murni** tanpa adanya pemrosesan dinamis di server.

Perkembangan Penting:

- **1989:** Tim Berners-Lee menciptakan **World Wide Web (WWW)** di CERN.
- **1991:** Situs web pertama dibuat menggunakan **HTML** sederhana.
- **1993:** **Mosaic**, salah satu browser pertama, dirilis untuk memudahkan akses web.
- **1995:** **HTML 2.0** diperkenalkan sebagai standar pertama HTML.

Ciri Khas Era Web Statis:

Konten hanya bisa diubah oleh pengembang (bukan pengguna).
Tidak ada elemen interaktif seperti formulir atau database.
Desain dan tata letak terbatas karena belum ada **CSS**.
Website hanya menampilkan informasi satu arah, seperti brosur digital.

Sejarah Perkembangan



Era Web Dinamis (1990-an – Awal 2000an)

Pada era ini, website mulai berkembang dari statis menjadi dinamis, memungkinkan interaksi antara pengguna dan sistem. Dengan bantuan teknologi server-side, website biasanya menampilkan konten yang berubah berdasarkan input pengguna.

Perkembangan Penting:

- **1995:**
 - **JavaScript** diperkenalkan oleh Netscape untuk menambahkan interaktivitas di sisi klien.
 - **PHP** dikembangkan sebagai bahasa pemrograman server-side untuk memproses data dinamis.
 - **MySQL** hadir sebagai database untuk menyimpan data website.
- **1996:** **CSS** diperkenalkan untuk memisahkan desain dari HTML.

Ciri Khas Era Web Dinamis:

Interaktif → Pengguna bisa menginput data melalui formulir, komentar, atau login.

Menggunakan database → Data bisa disimpan dan ditampilkan kembali tanpa perlu mengedit kode HTML secara manual.

Pemrosesan di sisi server → PHP dan MySQL memungkinkan pembuatan aplikasi web berbasis data.

Desain lebih fleksibel → Berkat CSS, tampilan website bisa diatur lebih baik.

Sejarah Perkembangan



Era Web 2.0 (2000-an-Awal 2010-an)

Era **Web 2.0** menandai perubahan besar dalam dunia web, di mana website tidak lagi hanya menyajikan informasi secara satu arah, tetapi juga memungkinkan **pengguna berkontribusi dalam menciptakan konten**. Website menjadi lebih **interaktif, dinamis, dan berbasis komunitas**.

Perkembangan Penting:

- **2003: WordPress** diluncurkan, mempermudah pembuatan website tanpa perlu coding.
- **2004: Ajax** mulai populer, memungkinkan interaksi tanpa perlu reload halaman.
- **2005: Lahirnya YouTube, Facebook, dan platform media sosial lainnya.**

Ciri Khas Era Web 2.0:

User-generated content → Pengguna bisa membuat dan berbagi konten (blog, media sosial, forum).
Interaksi real-time → Teknologi **Ajax** memungkinkan halaman diperbarui tanpa reload.
Munculnya platform berbasis komunitas → Facebook, YouTube, Wikipedia berkembang pesat.
E-commerce dan digital marketing → Website mulai digunakan untuk bisnis online.

Sejarah Perkembangan



Era Web Modern (2010-an-Sekarang)

Era **Web Modern** ditandai dengan perkembangan teknologi yang semakin canggih untuk menciptakan **website yang lebih cepat, responsif, dan interaktif**.

Fokus utama era ini adalah **pengalaman pengguna (UX)**, keamanan, dan efisiensi dalam pengembangan web.

Perkembangan Penting:

- **2010: HTML5 & CSS3** hadir, mendukung multimedia tanpa perlu plugin tambahan.
- **2011: Node.js** memperluas penggunaan JavaScript ke sisi server.
- **2013:** Framework modern seperti **React.js, Angular, dan Vue.js** muncul, membuat pengembangan web lebih modular.
- **2015: REST API & GraphQL** berkembang untuk komunikasi frontend-backend yang lebih efisien.
- **Sekarang:** Teknologi seperti **Progressive Web Apps (PWA), WebAssembly, dan AI-powered web** terus berkembang.

Ciri Khas Era Web 2.0:

Full-stack JavaScript → JavaScript digunakan di frontend (React, Vue) & backend (Node.js).
Responsif & mobile-friendly → Website otomatis menyesuaikan tampilan di berbagai perangkat.
API-driven development → REST API & GraphQL memungkinkan komunikasi data lebih cepat.
Keamanan lebih baik → HTTPS, enkripsi, dan autentikasi berbasis OAuth semakin dioptimalkan.
PWA (Progressive Web Apps) → Website bisa bekerja seperti aplikasi native tanpa harus diinstal.

TINGKAT PENETRASI INTERNET



Tingkat Penetrasi
Internet di Indonesia

79.5%

Jumlah Penduduk Terkoneksi
Internet Tahun 2024

221.563.479 Jiwa dari total populasi
278.696.200 jiwa penduduk Indonesia
tahun 2023



Pertumbuhan Tingkat
Penetrasi Indonesia



Perbedaan *Coding* dan *Programming*



What Different?

A close-up photograph of a person's hand, palm up, holding a small, white, fluffy cloud against a dark blue background.

Coding

- Menulis kode dalam bahasa pemrograman untuk menjalankan instruksi tertentu.
- Hanya pada **penulisan kode** sesuai sintaks bahasa pemrograman.
- Menguasai sintaks bahasa pemrograman (seperti Python, Java, PHP).
- Menulis script atau kode untuk menyelesaikan tugas tertentu.
- Menulis fungsi Python untuk menghitung luas segitiga.

Programming

- Proses lengkap dalam membuat aplikasi, termasuk perencanaan, penulisan kode, debugging, dan pengujian.
- Membuat solusi lengkap, termasuk logika bisnis, arsitektur aplikasi, dan debugging.
- Memahami algoritma, struktur data, desain sistem, debugging, dan deployment.
- Merancang, mengembangkan, menguji, dan memelihara perangkat lunak.
- Membuat sistem pemesanan tiket online, termasuk database, UI, dan backend.

Coding merupakan bagian kecil dari programming yang berfokus pada menulis kode. **Programming** mencakup seluruh proses pengembangan perangkat lunak, dari perencanaan hingga implementasi.

Jadi, seorang programmer adalah seorang coder, tetapi seorang coder belum tentu seorang programmer. 🚀



Tren Teknologi Web: Inovasi dan Perkembangan Terbaru

Progressive **WEB** Apps

Progressive Web Apps (PWA) semakin populer di tahun 2024. PWA menggabungkan keunggulan aplikasi web dan aplikasi native, memungkinkan pengguna untuk mengakses situs web dengan cepat dan offline setelah pertama kali diakses.

Dengan PWA, website dapat memberikan pengalaman yang lebih seamless, mirip dengan aplikasi mobile, tanpa perlu diunduh dari app store.



Serverless Architecture



Arsitektur tanpa server (*serverless architecture*) merupakan pendekatan baru dalam pengembangan web yang semakin populer. Ini memungkinkan pengembang untuk fokus pada pengkodean tanpa harus mengelola infrastruktur server. Dengan memanfaatkan layanan cloud seperti AWS Lambda dan Google Cloud Functions, aplikasi dapat dikembangkan dengan lebih cepat dan lebih skalabel, serta mengurangi biaya operasional.





KEAMANAN

Website dengan Zero Trust



Keamanan web menjadi perhatian utama di tahun 2024, terutama dengan pendekatan Zero Trust. Zero Trust adalah model keamanan yang tidak mempercayai perangkat atau pengguna, bahkan jika mereka sudah berada di dalam jaringan perusahaan. Pendekatan ini meningkatkan keamanan web dengan autentikasi terus-menerus dan perlindungan terhadap serangan siber.

MOTION UI/UX

A

Pengalaman pengguna (UI/UX) yang dinamis semakin penting

B

Motion UI menjadi tren baru untuk memberikan pengalaman visual yang interaktif

C

Dengan menggunakan animasi dan transisi yang halus

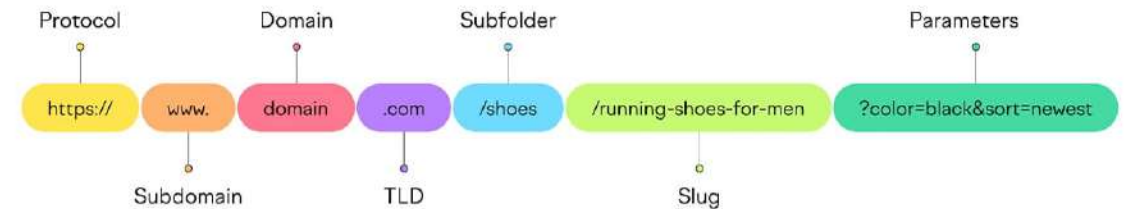
D

desain web menjadi lebih menarik dan intuitif, meningkatkan keterlibatan pengguna.



Uniform Resource Locator

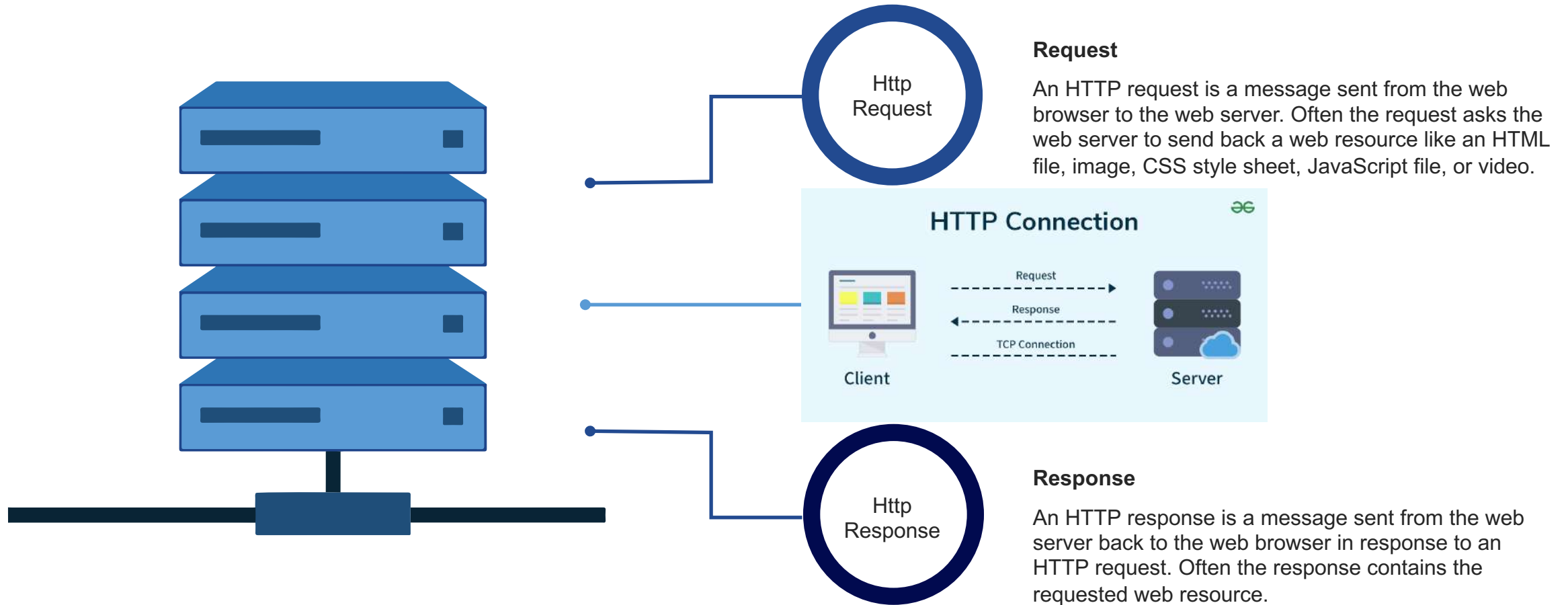
<https://untidar.ac.id>



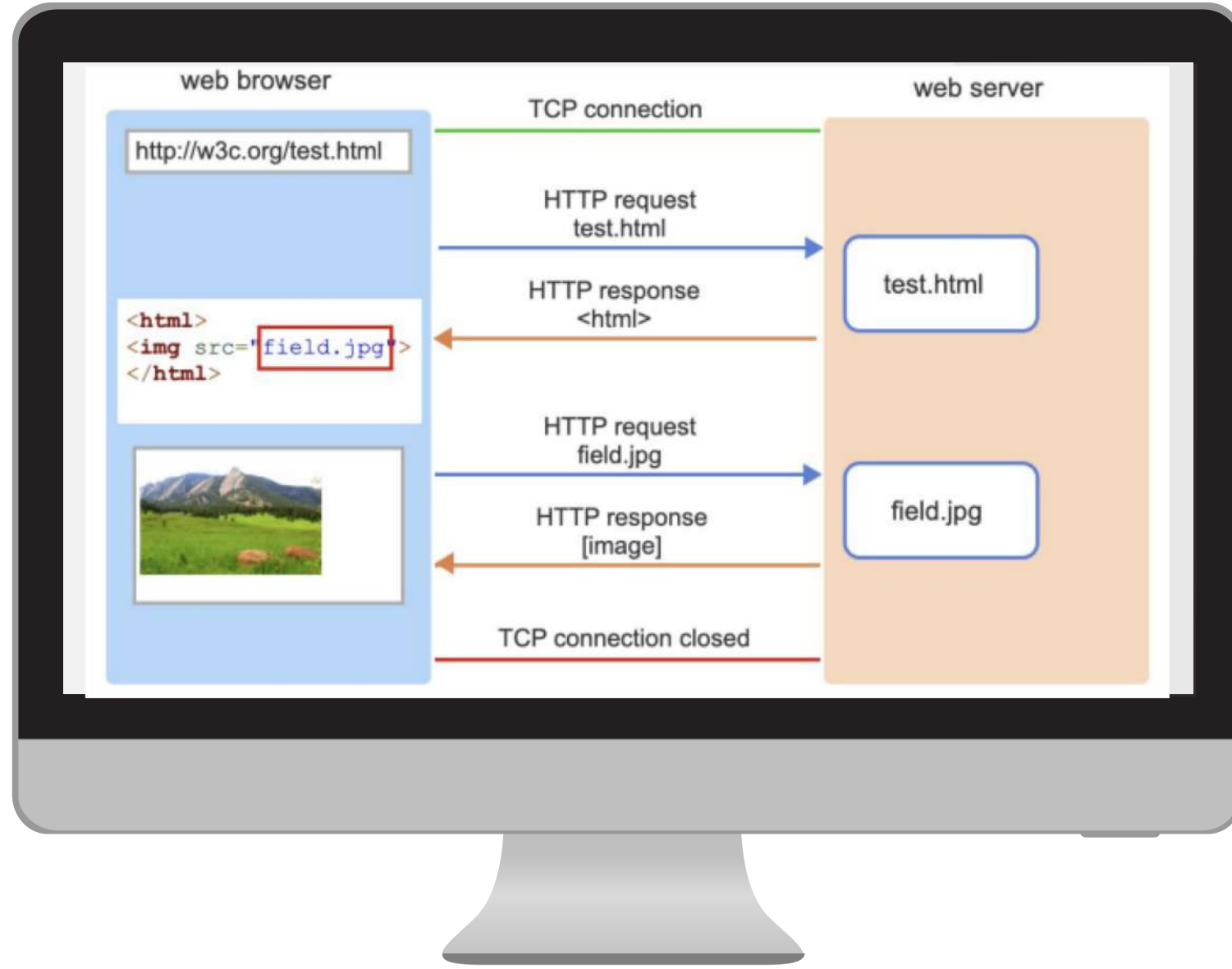
URLs are location dependent

It contains **four** distinct parts: the protocol type, the web server domain name, the directory path, the file name

HTTP (Hypertext Transfer Protocol)



HTTP (Hypertext Transfer Protocol)



HTTP Response Status Code

An HTTP response **status code** is a three digit number that indicates the status of the requested resource

HTTP Status Codes		
Level 200	Level 400	Level 500
200: OK 201: Created 202: Accepted 203: Non-Authoritative Information 204: No content	400: Bad Request 401: Unauthorized 403: Forbidden 404: Not Found 409: Conflict	500: Internal Server Error 501: Not Implemented 502: Bad Gateway 503: Service Unavailable 504: Gateway Timeout 599: Network Timeout



Client

Client-side code runs on the user's browser and is untrusted.



Responsible for:

- ✓ Displaying the UI
- ✓ Interactivity

Server

Server-side code runs on a server and is unseen by users.



Responsible for:

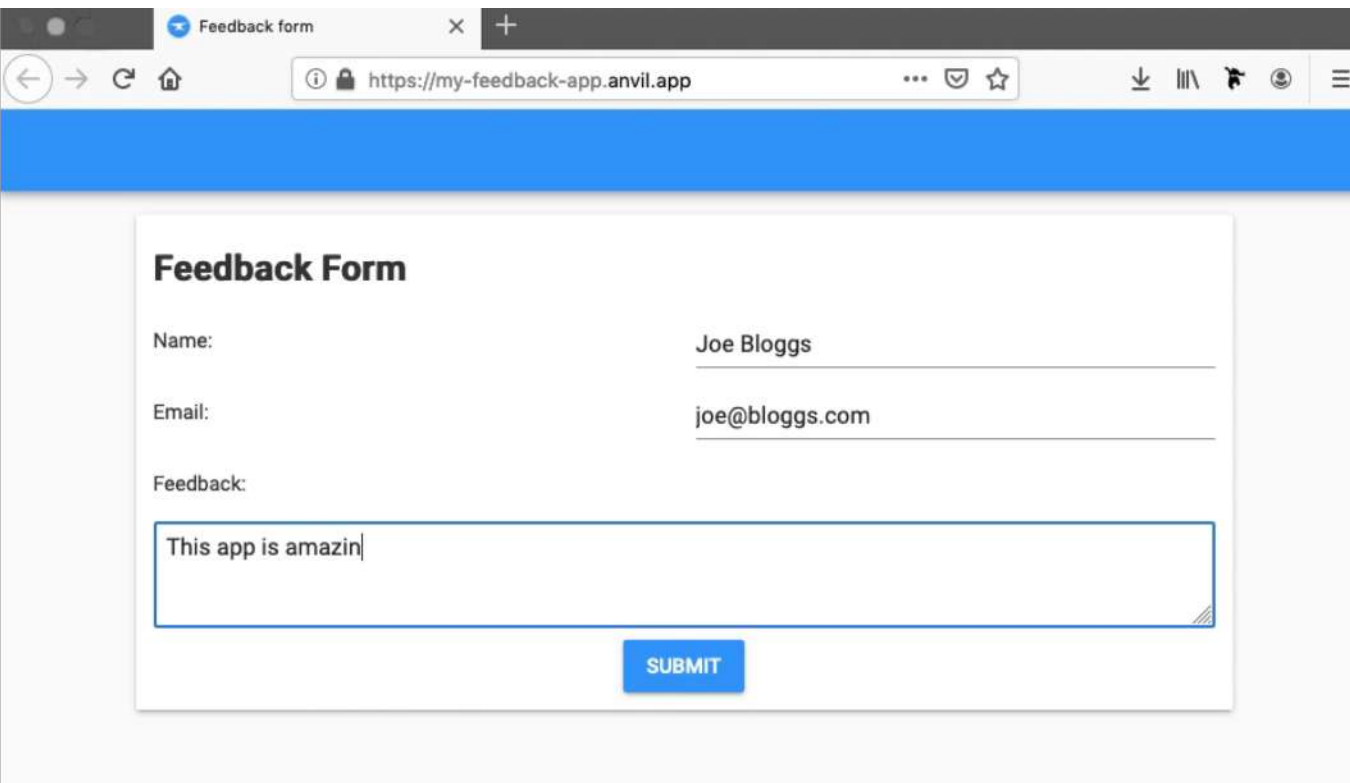
- ✓ Controlling central resources
- ✓ Permissions checks
- ✓ Handling secret information

Client-side Some process needs to be “executed” by the browser, either to form the request for the dynamic Web page or to create or display the dynamic Web Page



Server-side

- Receives the dynamic Web page request
- Performs all of the processing necessary to create the dynamic Web page
- Sends the finished Web page to the client for display in the client's browser



Feedback form

https://my-feedback-app.anvil.app

Feedback Form

Name: Joe Bloggs

Email: joe@bloggs.com

Feedback: This app is amazing

SUBMIT

Client-side

The front-end of the app is the user interface and the client-side code that collects user input. This code lives on the client because the user is interacting with it. After the user enters input and clicks the “submit” button, we use `anvil.server.call()` to call a server function to add this input to the Data Table:

```
def submit_button_click(self, **event_args):
    name = self.name_box.text
    email = self.email_box.text
    feedback = self.feedback_box.text
    # Call the 'add_feedback' server function
    # pass in name, email and feedback as arguments
    anvil.server.call('add_feedback', name, email, feedback)
```

Server-side

The server function and the Data Table are the back-end of our app. Because the server is separate from the user’s device, we can use it to run code we don’t want to expose to the user. In the case of the Feedback Form, we don’t want users to have access to code that modifies the Data Table since it contains other users’ data. By default, only server-side code has permission to access a Data Table. We can pass the user input to a server function that then updates the Data Table:

```
@anvil.server.callable
def add_feedback(name, email, feedback):
    app_tables.feedback.add_row(
        name=name,
        email=email,
        feedback=feedback,
        created=datetime.now()
    )
```


FRONT-END



Web App

Mobile App

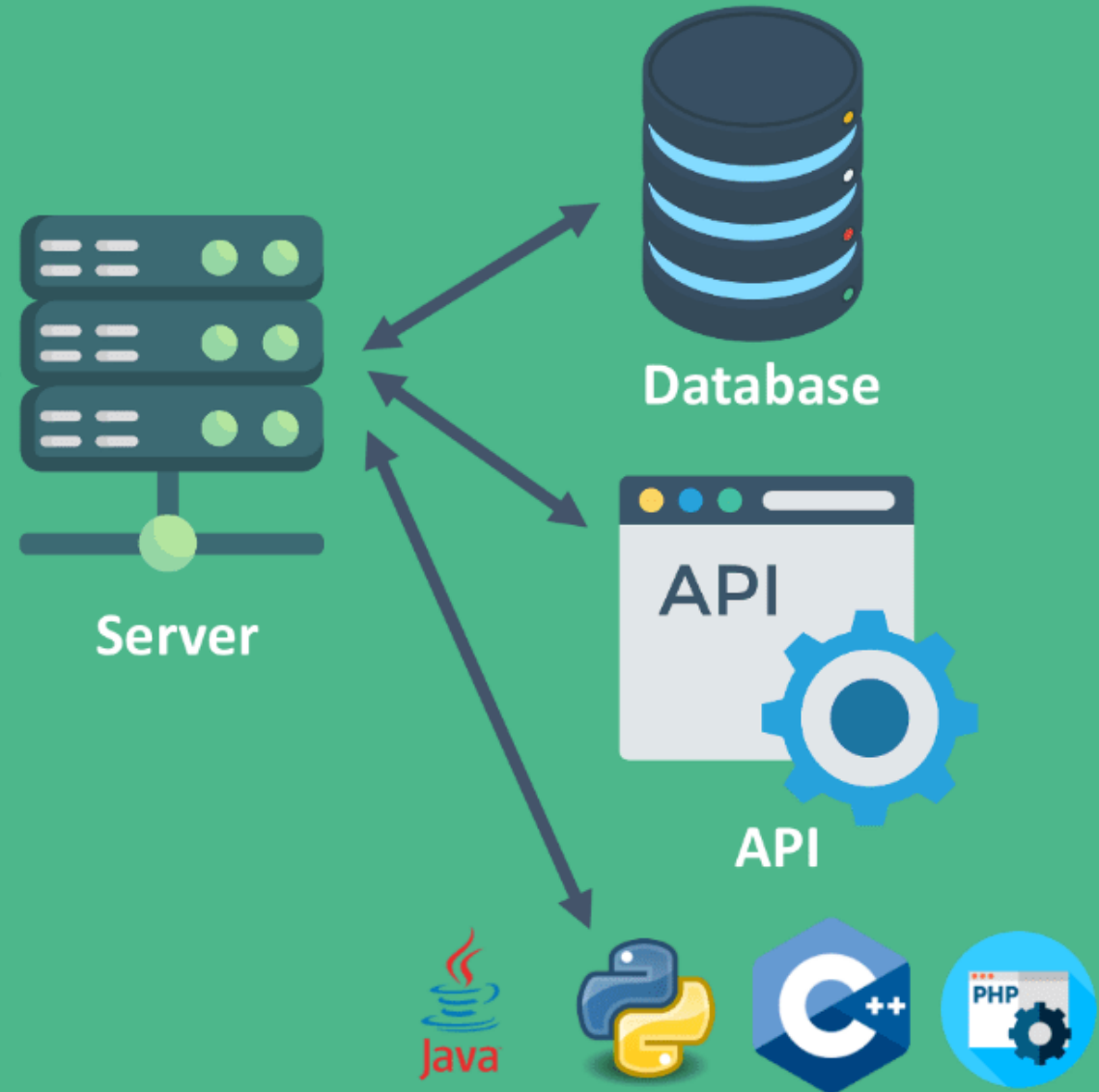
HTML

CSS

JS

JS

BACK-END



Server

Database

API

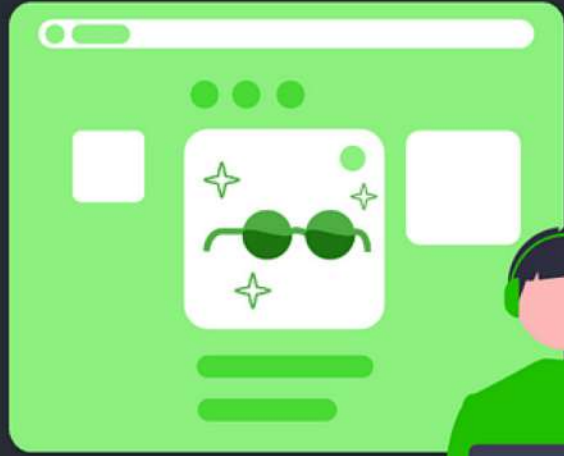
API

Java

Python

C++

PHP



Frontend



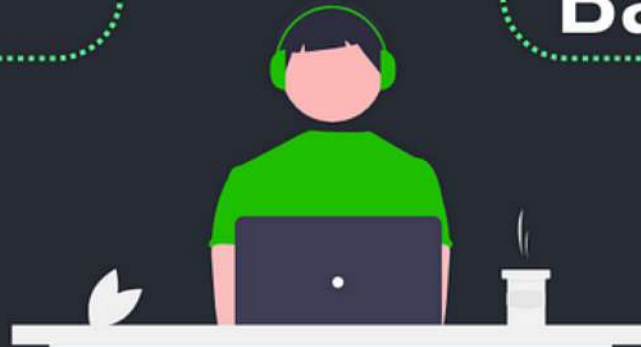
Backend



Database



Full Stack





WEB Framework

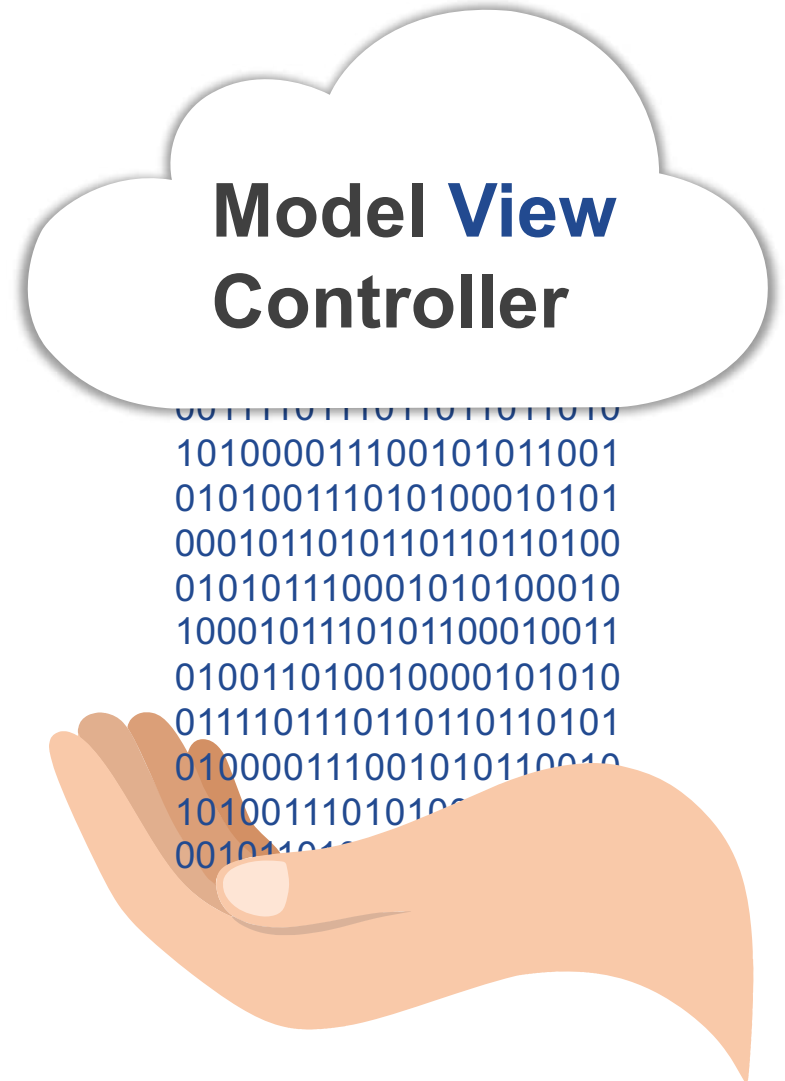
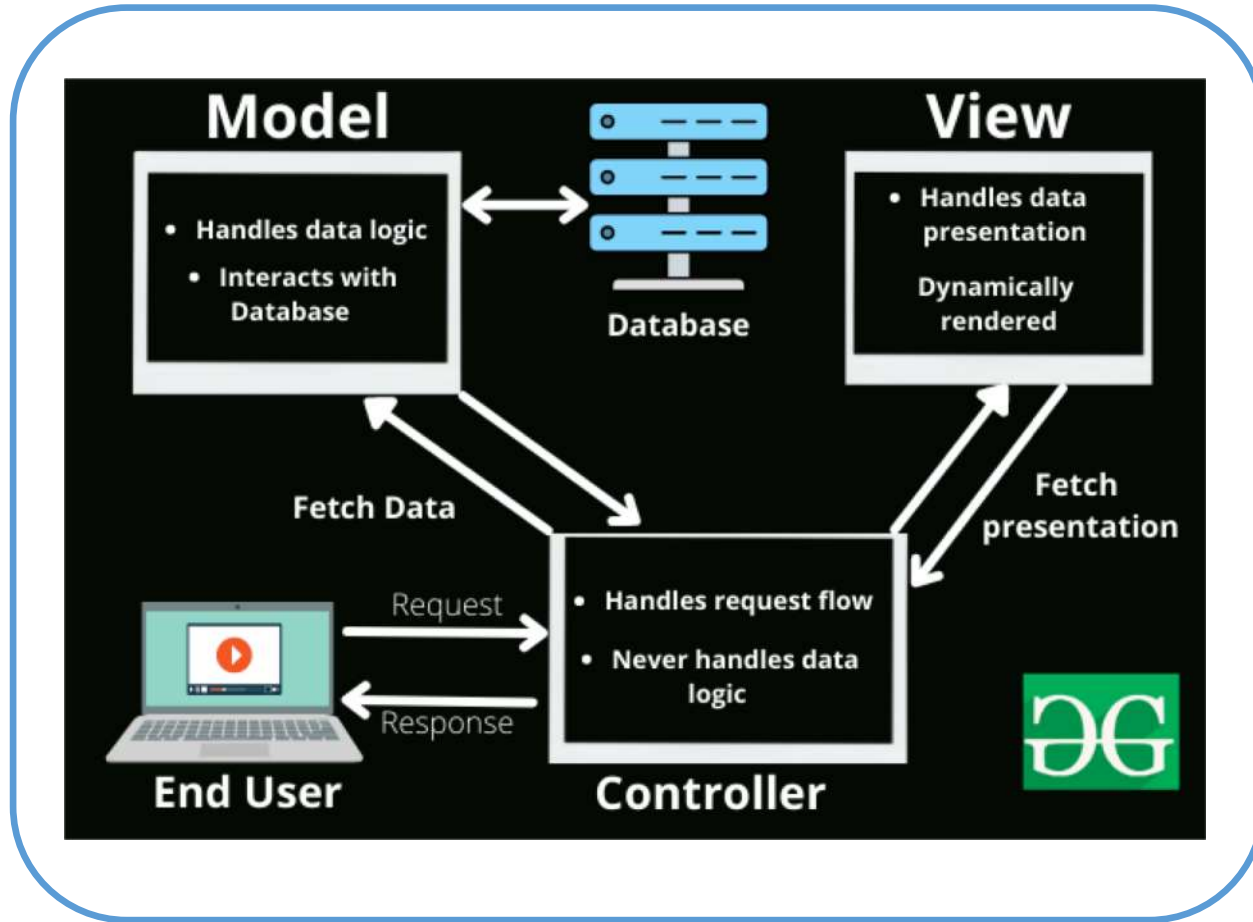
Types of web
Framework Architecture

Three-tier architecture:
client, application, database

Model View Controller (MVC) pattern

”

Introduction





THANK YOU

Universitas Tidar