

BAB 2

DASAR TEORI DAN TINJAUAN PUSTAKA

Pada bab 2 akan dibahas mengenai dasar teori dan tinjauan pustaka yang digunakan dalam penyusunan Proyek Akhir ini.

2.1 Tinjauan Pustaka

Beberapa penelitian sebelumnya yang dijadikan referensi untuk pembuatan Proyek Akhir ini meliputi:

Tabel 2.1 Tinjauan Pustaka

Peneliti	Judul Penelitian	Hasil
Felix Andreas Sutanto, Budi Hartono, Dwi Agus Diartono, Hari Murti	PELATIHAN DASAR FULL-STACK WEBSITE DEVELOPER BAGI KOMUNITAS PEMBELAJAR WEB	Hasil akhir dari penelitian ini adalah agar anggota komunitas dapat berperan sebagai full-stack website developer untuk aplikasi aplikasi yang berskala kecil. .
Jajang Winanjar, Deffy Susanti	RANCANG BANGUN SISTEM INFORMASI ADMINISTRASI DESA BERBASIS WEB MENGGUNAKAN PHP DAN MySQL	Hasil dari penelitian ini berupa Sistem Informasi Administrasi Desa yang dapat menunjang pelayanan di Desa Sukamulya. Dari hasil pengujian yang dilakukan Sistem Informasi ini dibuat menggunakan bahasa pemrograman PHP dan Mysql sebagai databasenya, Sistem Informasi ini dapat menyajikan data penduduk dan surat desa.
Taufik R. Firdaus	KEAMANAN APLIKASI WEB MELALUI PENERAPAN CROSS SITE REQUEST FORGERY(CSRF)	Hasil dari penelitian ini diharapkan agar dapat meningkatkan pertahanan aplikasi web terhadap CSRF (Cross-Site Request Forgery), sehingga pengguna aplikasi web dapat merasa aman

		dalam menggunakan internet dan berbagai fiturnya. Berkurangnya tingkat serangan terhadap aplikasi web. Sehingga trafik pengunjung pada aplikasi web dapat meningkat.
--	--	--

2.2 Dasar Teori

Dasar teori mencakup konsep atau pengetahuan, metode, dan alat yang digunakan untuk mendukung penyelesaian Proyek Akhir.

2.2.1 CSRF

CSRF (Cross-Site Request Forgery) merupakan sebuah serangan yang mana memaksa pengguna untuk mengeksekusi aksi yang tidak di inginkan pada aplikasi web, dimana dia (korban) saat ini terautentikasi. Serangan ini memanfaatkan keuntungan fungsi protokol HTTP untuk mengirim *session cookie* pada saat request ke *server* setiap kali pengguna berhasil melakukan autentikasi, yang membantu *server* dalam memastikan request tersebut berasal dari pengguna otentik [1].

Token CSRF (*Cross-Site Request Forgery*) adalah nilai unik yang dihasilkan oleh server untuk setiap sesi pengguna atau setiap permintaan yang berhubungan dengan *form*. Token ini ditambahkan ke setiap permintaan (*request*) yang dapat mengubah data di *server*, biasanya dalam bentuk tag *input hidden* didalam *form* HTML. Fungsi utama dari token CSRF adalah untuk memastikan bahwa permintaan yang diterima oleh *server* memang berasal dari pengguna yang sah dan bukan dari sumber *eksternal* yang tidak diotorisasi.

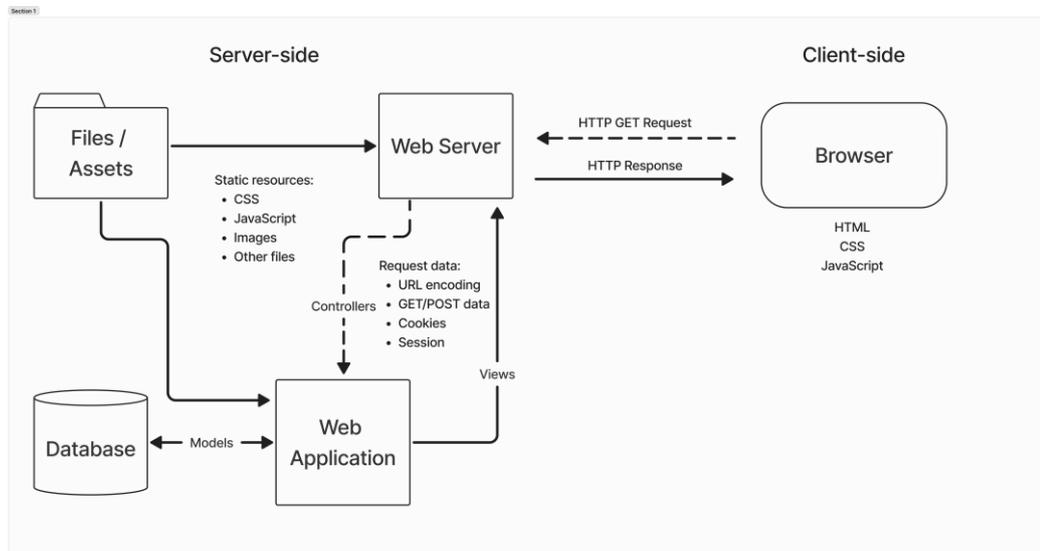
Saat pengguna mengisi dan mengirim *form* pada sebuah aplikasi web, *form* tersebut akan menyertakan token CSRF. Kemudian *server* akan memverifikasi token ini sebelum memproses *request* untuk memastikan bahwa proses tersebut berasal dari pengguna yang sah, dapat dengan memastikan bahwa token tersebut cocok dengan yang telah disimpan di sisi *server*, dapat melalui *cookie* atau *session*. Sehingga ketika *form* dikirim, token CSRF dari *form* akan dibandingkan dengan token yang ada di *cookie* atau dari *session*. Setelah *server* memverifikasi bahwa token cocok dengan yang ada di *cookie* atau *session*, *server* akan menghasilkan

token CSRF baru dan memperbarui *cookie* atau *session* dengan token yang baru juga.

2.2.2 *Full Stack Developer*

Full Stack Developer adalah sebutan untuk developer yang berurusan dengan segala sisi pengembangan website atau web development, baik *front-end* hingga *back-end*. Persyaratan dasar yang diperlukan untuk membangun web meliputi sistem operasi, *server* web, database, dan bahasa/*scripting*. Bersama dengan perangkat keras *server* yang tepat, kumpulan komponen ini memastikan bahwa informasi yang diperlukan dapat diberikan kepada klien yang meminta melalui browser internet [2].

Full Stack Developer memerlukan usaha yang lebih bagi seseorang yang ingin menekuninya. Ada beberapa keahlian yang diperlukan untuk menjadi seorang *full-stack developer*. Untuk dasarnya, programmer web harus bisa beberapa *scripting* disisi client/browser seperti HTML, CSS dan *JavaScript*. Sedangkan disisi *server/back-end*, harus menguasai *scripting* yang bisa dieksekusi oleh web *server* dan mengirimkan informasi ke client. Pengembang web juga harus mengerti *Structure Query Language* (SQL) yang digunakan untuk menyimpan dan memanipulasi data. *Relational Database Management System* (RDBMS) adalah sistem basis data yang umum digunakan oleh programmer pada saat ini. Software RDBMS yang umum digunakan dalam sistem informasi adalah Oracle, SQL Server, PostgreSQL, DB2, dan MySQL. Seringkali seorang programmer pemula kurang melakukan optimalisasi sebuah query di MySQL, padahal optimalisasi sangat penting ketika jumlah data menjadi besar. [2].



Gambar 2.1 Skema *Web Development* dengan *Design Pattern* MVC

Gambar diatas menggambarkan berbagai lapisan serta komponen yang terlibat dalam pengembangan *website*. Pada sisi *Client-side*, browser bertanggung jawab untuk mengirimkan *request* HTTP ke *web server* serta menerima respon juga dari *web server* yang kemudian hasilnya akan ditampilkan di browser. Pada sisi *Server-side*, *web server* bertanggung jawab untuk menerima permintaan HTTP dari browser dan mengembalikan respons tersebut ke browser. *Web Server* juga bertanggung jawab untuk menyajikan *Static resource*, yaitu *resource* yang tidak sering diubah kecuali diperlukan. Lalu pada *Web Application* bertanggung jawab untuk memproses *Business Logic* (aturan dan prosedur yang mengatur bagaimana data diproses, disimpan, dan diakses di aplikasi tersebut) dan berinteraksi dengan *database* [3]. *Controllers* berfungsi untuk mengelola *request data* seperti *URL encoding*, data GET/POST, *cookies*, dan *session*. *Views* bertanggung jawab untuk menghasilkan halaman HTML yang akan dikirim ke browser sebagai respon. Kemudian pada *Database* digunakan untuk menyimpan data yang diperlukan oleh aplikasi web menggunakan *model* untuk berinteraksi.

2.2.3 PHP

PHP merupakan singkatan dari *PHP Hypertext Preprocessor*. PHP merupakan bahasa pemrograman *script* yang diletakkan dalam *server* yang biasa digunakan untuk membuat aplikasi web yang bersifat dinamis [4].

Karena php merupakan scripting *server-side* maka jenis bahasa pemrograman ini nantinya script/program tersebut akan dijalankan/diproses oleh *server*. Berbeda dengan javascript yang *client-side*.

PHP adalah bahasa pemrograman umum yang berarti php dapat disematkan ke dalam kode HTML, atau dapat digunakan dalam kombinasi dengan berbagai sistem templat web, sistem manajemen konten web, dan kerangka kerja web [5].

2.2.4 MySQL

MySQL adalah salah satu jenis *database server* yang sangat populer, hal ini disebabkan karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya*. MySQL bersifat *Open Source*, Software ini dilengkapi dengan *Source code* (kode yang dipakai untuk membuat MySQL) [4].

Ada dua hal yang perlu diperhatikan untuk memahami cara kerja MySQL:

- Relational database

Banyak cara yang bisa dilakukan untuk menyimpan data dalam *database*. MySQL menyimpan data dalam format *relational database*. Format tersebut memungkinkan data dipecah dalam beberapa beberapa penyimpanan terpisah (biasa disebut tabel) dan memiliki relasi satu sama lain.

- Model *client-server*

Penyimpanan data di MySQL menggunakan model *client-server*. *Server* merupakan tempat di mana data berada. Data tersebut bisa diakses jika klien memintanya. Permintaan ini dilakukan menggunakan SQL, yaitu bahasa pemrograman yang dapat membantu klien mengirim permintaan ke *server database* untuk menampilkan informasi yang dibutuhkan.

2.2.5 Codeigniter

CodeIgniter adalah sebuah *framework* php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari [4].

Framework CodeIgniter memungkinkan *developer* bekerja secara lebih efisien dan kreatif, dengan menyediakan *library* atau kumpulan baris kode yang siap digunakan untuk berbagai macam *use case* yang paling sering dibutuhkan, seperti membangun *user interface* website yang interaktif. Selain, fitur-fitur CodeIgniter didesain agar dapat bekerja secara efektif tanpa perlu terlalu bergantung terhadap komponen-komponen atau modifikasi tambahan [6].

2.2.6 Figma

Figma adalah salah satu *design tool* yang biasanya digunakan untuk membuat tampilan aplikasi mobile, desktop, website dan lain-lain. Figma menawarkan keuntungan berupa aksesibilitas dan kemudahan kolaborasi karena dapat diakses secara *online* serta memungkinkan pengguna untuk bekerja secara *real-time* karena Figma berbasis vektor dan *cloud* [7].

Sebagai *fullstack developer*, pemahaman dan penggunaan Figma dapat membantu meningkatkan kolaborasi dengan tim desain, memastikan kesesuaian antara desain dan implementasi, serta mempercepat proses pengembangan secara keseluruhan.

2.2.7 Visual Studio Code

Visual Studio Code adalah editor source code yang ringan, gratis, dan *open source*, yang dikembangkan oleh Microsoft. Alat ini biasanya digunakan untuk pengembangan web, tetapi mendukung banyak bahasa pemrograman dan memiliki berbagai *extension* yang tersedia. VS Code dirancang supaya sederhana, cepat, dan cocok untuk *developer* individu dan tim kecil.

Di sisi lain, Visual Studio adalah *Integrated Development Environment* (IDE) berfitur lengkap yang juga dikembangkan oleh Microsoft. Visual Studio menawarkan lebih banyak alat dan fitur, termasuk *debugging* tingkat lanjut, pembuatan profil, dan kemampuan manajemen proyek.

2.2.8 NPM

Node Package Manager atau NPM adalah *package manager* dan *repository software* dalam bahasa pemrograman JavaScript yang digunakan untuk memasang, menghapus, dan mengelola *library* atau *package software* dalam proyek pengembangan aplikasi.

NPM memungkinkan developer membagikan kode yang mereka tulis dan menggunakan kode yang ditulis oleh orang lain, memfasilitasi kolaborasi dan mempercepat proses pengembangan. Repository NPM berisi lebih dari satu juta package perangkat lunak yang dapat dipakai dan dikontribusikan oleh siapa saja [8].

2.2.9 Composer

Composer adalah sebuah tools manajemen dependensi untuk bahasa pemrograman PHP. Dalam konteks pemrograman, Composer adalah sebuah aplikasi yang digunakan untuk mengelola dan mengatur dependensi seperti *library*, *framework*, atau paket lain yang diperlukan dalam proyek PHP. Dengan menggunakan Composer, dapat dengan mudah mengelola dependensi proyek dengan mendefinisikan *library* atau paket yang diperlukan dalam berkas konfigurasi *composer.json*. Composer kemudian akan mengelola unduhan, instalasi, dan pembaruan dari dependensi-dependensi ini [9].

2.2.10 Bootstrap

Bootstrap adalah salah satu *framework front-end development* yang populer untuk mengembangkan aplikasi *web responsive* dan *mobile-first*. *Framework* ini dibangun berbasis HTML, CSS, dan *JavaScript*. Komponen-komponen dasarnya dibangun menggunakan kombinasi HTML untuk struktur, CSS untuk style dan

layout dan *JavaScript* untuk interaktif. Bootstrap memanfaatkan *jQuery*, sebuah *library JavaScript* populer, untuk menangani interaksi dan animasi [10].

Tujuan dari Bootstrap adalah untuk menyederhanakan pengembangan web sehingga developer tidak perlu menulis kode CSS dari nol, melainkan bisa menggunakan kumpulan template CSS dan JavaScript yang sudah dibuat untuk komponen antarmuka.

2.2.11 JavaScript

JavaScript adalah bahasa pemrograman populer yang digunakan untuk membuat situs dengan konten website yang dinamis. Konten dinamis artinya konten dapat bergerak atau berubah di depan layar tanpa perlu me-*reload* halaman. Misal, fitur *slideshow* foto, gambar animasi, pengisian poling, dan lainnya [11].

Cara kerja JavaScript bisa dilihat dari dua sisi, yaitu dari sisi client dan dari sisi server. Dari sisi client, JavaScript dapat berperan dalam mengubah tampilan website. Sedangkan dari sisi server, JavaScript dapat digunakan untuk kebutuhan database.

2.2.12 jQuery

jQuery adalah *library JavaScript* yang khusus dirancang untuk menyederhanakan *scripting* HTML. Pada umumnya pembuatan kode JavaScript membutuhkan kode yang cukup rumit dan panjang disinilah jQuery berperan untuk menyederhanakannya dengan menggunakan fungsi yang terdapat didalam *library* tersebut [12].

Salah satu kelebihan jQuery adalah *event handling* atau penanganan event. Pembuatan *event* seperti fungsi *mouse click* (klik mouse) dan *form submission* (pengiriman form) hanya memerlukan beberapa baris kode. Metode *AJAX* (*Asynchronous JavaScript and XML*) secara khusus membuat jQuery sangat sesuai untuk mengembangkan website responsif menggunakan fungsi *AJAX*. Fitur ini meningkatkan pengalaman pengguna dengan menyederhanakan operasi permintaan *HTTP* tanpa harus reload halaman web.