

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

The first previous research conducted by L. Bosjak, (2018) with the title "Brute-force and dictionary attacks on real-world passwords". The research problem is to crack most user-generated passwords using simple and predictable patterns. To identify the differences between them, we performed an analysis of cracked and uncracked passwords and measured their strength. This discussion leads to the conclusion that results have shown that even one modern low to mid-range GPU can crack over 95% of passwords in just a few days, while more dedicated systems can crack all but the strongest 0.5% of them.

Penelitian terdahulu pertama yang dilakukan oleh L. Bosjak, (2018) dengan mengambil judul "**Brute-force dan serangan kamus pada kata sandi dunia nyata**". Permasalahan penelitian untuk memecahkan sebagian besar kata sandi yang dibuat pengguna menggunakan sederhana dan pola yang dapat diprediksi. Untuk mengidentifikasi perbedaan antara mereka, kami melakukan analisis retak dan *password uncracked* dan diukur kekuatan mereka. Pembahasan ini menghasilkan kesimpulan yaitu hasil telah menunjukkan bahwa bahkan satu low hingga mid-range GPU modern dapat memecahkan lebih dari 95% kata sandi hanya dalam beberapa hari, sementara sistem yang lebih berdedikasi dapat memecahkan semua kecuali terkuat 0,5% dari mereka.

The first previous research conducted by Marijn J. H. Heule (2017) with the title "Science of Bruce Force". The problem discussed in this study is a threefold Pythagorean boolean problem, which comes from the mathematical domain. This discussion led to the conclusion that the largest evidence ever produced (200 TB), had strong media echoes, and skepticism about "meaning" (or lack thereof) had been raised. We show, however, that these extracted methods and evidence do have meaning, and are indeed useful, when considering real-world applications, security and verification domains.

Penelitian terdahulu pertama yang dilakukan oleh Marijn J. H. Heule (2017) dengan mengambil judul "**Ilmu dari Bruce Force**". Permasalahan yang dibahas dalam penelitian ini adalah masalah tiga kali lipat boolean *Pythagoras*, yang berasal dari domain matematika. Pembahasan ini menghasilkan kesimpulan yaitu bukti terbesar yang pernah dibuat (200 TB), memiliki gema media yang kuat, dan *skeptisisme* tentang "makna" (atau ketiadaannya) telah dimunculkan. Kami menunjukkan, bahwa metode dan bukti yang diekstraksi ini memang memiliki makna, dan memang berguna, ketika mempertimbangkan aplikasi dunia nyata, domain keamanan dan verifikasi.

The first previous research conducted by Mark V Ivanov (2018) with the title "Brute-Force Approach for Identification of Peptide Variants Based on Mass Spectrometry in Proteogenomics without Personalized Genome Data". The problem addressed in this study is to identify peptide variants without the essential personalized genomic data for various applications. This discussion leads to the

conclusion that To increase the efficiency of this approach, we propose an algorithm to exclude false variant identification from search results involving analysis of modifications that mimic single amino acid substitutions.

Penelitian terdahulu pertama yang dilakukan oleh Mark V Ivanov (2018) dengan mengambil judul **“Pendekatan Brute-Force untuk Identifikasi Varian Peptida Berbasis Spektrometri Massa dalam Proteogenomik tanpa Data Genom yang Dipersonalisasi”**. Permasalahan yang dibahas dalam penelitian ini adalah untuk mengidentifikasi varian peptida tanpa data genom yang dipersonalisasi penting untuk berbagai aplikasi. Pembahasan ini menghasilkan kesimpulan yaitu Untuk meningkatkan efisiensi pendekatan ini, kami mengusulkan algoritma untuk mengecualikan identifikasi varian palsu dari hasil pencarian yang melibatkan analisis modifikasi yang meniru substitusi asam amino tunggal.

The first previous research conducted by Adwan Yasin, (2016) with the title "INCREASING RFID SECURITY AGAINST BRUTE FORCE ATTACKS BASED ON PASSWORD POWER AND MARKOV MODEL". The problems discussed in this study are identifying threats and countermeasures from possible attacks. This discussion resulted in the conclusion that is to increase the security in their communications and survive from brute force attacks.

Penelitian terdahulu pertama yang dilakukan oleh Adwan Yasin, (2016) dengan mengambil judul **“MENINGKATKAN KEAMANAN RFID TERHADAP SERANGAN BRUTE FORCE BERDASARKAN KEKUATAN PASSWORD DAN MODEL MARKOV”**. Permasalahan yang dibahas dalam penelitian ini adalah mengidentifikasi ancaman dan penanggulangan dari kemungkinan serangan. Pembahasan ini menghasilkan kesimpulan yaitu untuk meningkatkan keamanan dalam komunikasi mereka dan bertahan dari serangan *brute force*.

The first previous research conducted by Bahaa qasim musawi, (2012) with the title "PREVENTING BRUTE FORCE ATTACKS THROUGH ANALYSIS LOGS". The problem discussed in this study is to run into an attack that tries to force a username and password combination via SSH to gain access. This discussion leads to a conclusion that is to consider whether the blocked address belonging to the user failed to be accessed or by the attacker.

Penelitian terdahulu pertama yang dilakukan oleh Bahaa qasim musawi, (2012) dengan mengambil judul **“MENCEGAH SERANGAN BRUTE FORCE MELALUI LOG ANALISIS”**. Permasalahan yang dibahas dalam penelitian ini adalah untuk mengalami serangan yang mencoba memaksa kombinasi nama pengguna dan kata sandi melalui SSH untuk mendapatkan akses. Pembahasan ini menghasilkan kesimpulan yaitu untuk mempertimbangkan apakah alamat yang diblokir milik pengguna gagal untuk diakses atau oleh penyerang.

Penelitian terdahulu pertama yang dilakukan oleh Nur Afif, (2018) dengan mengambil judul **“implementasi algoritma brute force dalam perancangan aplikasi penelusuran skripsi”**. Permasalahan yang dibahas dalam penelitian ini adalah bagaimana mahasiswa tingkat akhir menemukan informasi skripsi serta menemukan saran ide dalam pengajuan judul. Pembahasan ini

menghasilkan kesimpulan yaitu mahasiswa tidak mengajukan judul secara manual melainkan langsung menginput data ke website yang tersedia. Selain itu, mahasiswa dan dosen dapat secara langsung melakukan penelusuran skripsi dan menemukan tugas akhir mahasiswa lainnya dalam waktu relative cepat dan akurat.

Penelitian terdahulu pertama yang dilakukan oleh ismail m manto, maryam hasan (2020) dengan mengambil judul “**implementasi Algoritma Brute Force Untuk Pencarian Judul Buku Berbasis Android**”. Permasalahan yang dibahas dalam penelitian ini adalah pada pelayanan perpustakaan tidak terlepas pada aktifitas pencarian data, hal ini karena jumlah buku yang terdapat di perpustakaan sudah terlalu banyak, untuk itu penelitian ini merancang aplikasi pencarian judul buku berbasis android menggunakan algoritma *Bruce Force*. Pembahasan ini menghasilkan kesimpulan yaitu dapat melakukan fungsi pencarian judul buku. dan dapat diimplementasikan dalam pencarian judul buku.

Penelitian terdahulu pertama yang dilakukan oleh Hanif Sidiq Pratita (2016) dengan mengambil judul “**Analisa Bruce Force Attack menggunakan scanning aplikasi pada HTTP Attack**”. Permasalahan yang dibahas dalam penelitian ini adalah bagaimana penyelesaian permasalahan *password cracking* dengan menggunakan algoritma *brute force* akan menempatkan dan mencari semua kemungkinan password dengan masukan karakter dan panjang *password* tertentu tentunya dengan banyak sekali kombinasi *password*. Pembahasan ini menghasilkan kesimpulan yaitu bagaimana proses yang terjadi ketika sebuah *website* di serang dengan proses *brute force*.

Penelitian terdahulu pertama yang dilakukan oleh Sugiharto, S.Kom (2018) dengan mengambil judul “**implementasi algoritma brute force dalam pencarian kebudayaan di indonesia berbasis mobile application**”. Permasalahan yang dibahas dalam penelitian ini adalah bagaimana membangun aplikasi pencarian informasi kebudayaan yang terdapat di indonesia. Pembahasan ini menghasilkan kesimpulan yaitu untuk memudahkan untuk mencari informasi kebudayaan di indonesia dengan mudah dan cepat. Untuk membangun aplikasi pencarian kebudayaan diIndonesia yang berbasis mobile, maka dalam penelitian ini penulis akan menggunakan algoritma *brute force*.

Penelitian terdahulu pertama yang dilakukan oleh Alfin Tazkiya Ramadhoni, dkk., (2022) dengan mengambil judul “**Penerapan algoritma *Brute Force* pada aplikasi sidayko berbasis android**”. Permasalahan yang dibahas dalam penelitian ini adalah untuk memberikan gambaran tujuan dan proses yang terjadi pada aplikasi Sidayko serta menggunakan *Entity Relational Diagram* (ERD) untuk menunjukkan hubungan antar entitas yang ada pada aplikasi Sidayko. Pembahasan ini menghasilkan kesimpulan yaitu menunjukkan bahwa aplikasi Sidayko berhasil menerapkan algoritma *Brute Force* dengan tingkat akurasi mencapai 72%.

Tabel 2. 1.1 Daftar Referensi Jurnal Internasional

NO	AUTHOR , JUDUL JURNAL / TAHUN TERBIT	MASALAH	METODE PENYELESAIA N	HASIL
1.	L.Bosjak, <i>Brute-force</i> dan serangan kamus pada kata sandi dunia nyata, 2018	Bagaimana memecahkan sebagian besar kata sandi yang dibuat pengguna menggunakan sederhana dan pola yang dapat diprediksi. Untuk mengidentifikasi perbedaan antara mereka, kami melakukan analisis retak dan	Metode <i>Bruce Force</i>	bukti terbesar yang pernah dibuat (200 TB), memiliki gema media yang kuat, dan skeptisisme tentang "makna" (atau ketiadaannya) telah dimunculkan. Kami menunjukkan, bahwa metode dan bukti yang diekstraksi ini memang memiliki makna, dan memang berguna, ketika mempertimbangkan aplikasi dunia nyata, domain keamanan dan verifikasi

		password uncracked dan diukur kekuatan mereka		
2.	Marijn J. H. Heule, Ilmu dari <i>Bruce Force</i> , 2017	masalah tiga kali lipat boolean Pythagoras, yang berasal dari domain matematika	Metode <i>Bruce Force</i>	bukti terbesar yang pernah dibuat (200 TB), memiliki gema media yang kuat, dan skeptisisme tentang "makna" (atau ketiadaannya) telah dimunculkan. Kami menunjukkan, bahwa metode dan bukti yang diekstraksi ini memang memiliki makna, dan memang berguna, ketika mempertimbangkan aplikasi dunia nyata, domain keamanan dan verifikasi.
3.	<u>Mark V Ivanov</u> , Pendekatan <i>Brute-Force</i> untuk Identifikasi Varian Peptida Berbasis Spektrometri Massa dalam	mengidentifikasi varian peptida tanpa data genom yang dipersonalisasi penting untuk berbagai aplikasi	Metode <i>Bruce Force</i>	Untuk meningkatkan efisiensi pendekatan ini, kami mengusulkan algoritma untuk mengecualikan identifikasi varian palsu dari hasil pencarian yang melibatkan analisis modifikasi yang meniru substitusi asam amino tunggal

	Proteogenomik tanpa Data Genom yang Dipersonalisasi, 2018			
4.	Adwan Yasin, Meningkatkan keamanan rfid terhadap serangan <i>brute force</i> berdasarkan kekuatan password dan model markov, 2016.	mengidentifikasi ancaman dan penanggulangan dari kemungkinan serangan.	Metode <i>Bruce Force</i>	untuk meningkatkan keamanan dalam komunikasi mereka dan bertahan dari serangan brute force.
5.	Bahaa qasim musawi, Mencegah serangan <i>brute force</i> melalui log analisis, 2015	untuk mengalami serangan yang mencoba memaksa kombinasi nama pengguna dan kata sandi melalui SSH untuk mendapatkan akses	Metode <i>Bruce Force</i>	untuk mempertimbangkan apakah alamat yang diblokir milik pengguna gagal untuk diakses atau oleh penyerang.

Tabel 2. 1.2 Daftar Referensi Jurnal Nasional

NO	AUTHOR, JUDUL JURNAL, TAHUN TERBIT	MASALAH	METODE PENYELESAIAN	HASIL
1.	Nur Afif, implementasi algoritma <i>brute force</i> dalam perancangan aplikasi penelusuran skripsi, 2018	Bagaimana mahasiswa tingkat akhir menemukan informasi skripsi serta menemukan saran ide dalam pengajuan judul	Metode <i>Bruce Force</i>	Mahasiswa tidak mengajukan judul secara manual melainkan langsung menginput data ke website yang tersedia. Selain itu, mahasiswa dan dosen dapat secara langsung melakukan penelusuran skripsi dan menemukan tugas akhir mahasiswa lainnya dalam waktu relative cepat dan akurat.
2.	ismail m manto, maryam hasan, implementasi Algoritma <i>Brute Force</i> Untuk Pencarian Judul Buku	Pada pelayanan perpustakaan tidak terlepas pada aktifitas pencarian data, hal ini karena jumlah buku yang terdapat di perpustakaan sudah terlalu banyak, untuk itu	Metode <i>Bruce Force</i>	Dapat melakukan fungsi pencarian judul buku. dan dapat diimplementasikan dalam pencarian judul buku

	Berbasis Android, 2020	penelitian ini merancang aplikasi pencarian judul buku berbasis android menggunakan algoritma <i>Bruce Force</i>		
3.	Hanif Sidiq Pratita, Analisa <i>Bruce Force Attack</i> menggunakan <i>scanning</i> aplikasi pada <i>HTTP Attack</i> , 2016	Bagaimana penyelesaian permasalahan password cracking dengan menggunakan algoritma brute force akan menempatkan dan mencari semua kemungkinan password dengan masukan karakter dan panjang password tertentu tentunya dengan banyak sekali kombinasi password	Metode <i>Bruce Force</i>	Bagaimana proses yang terjadi ketika sebuah website di serang dengan proses brute force.

4.	Sugiharto, S.Kom, implementasi algoritma <i>brute force</i> dalam pencarian kebudayaan di indonesia berbasis <i>mobile application</i> , 2018	Bagaimana membangun aplikasi pencarian informasi kebudayaan yang terdapat di indonesia	Metode <i>Bruce Force</i>	untuk memudahkan untuk mencari informasi kebudayaan di indonesia dengan mudah dan cepat. Untuk membangun aplikasi pencarian kebudayaan diIndonesia yang berbasis mobile, maka dalam penelitian ini penulis akan menggunakan algoritma brute force
5.	Alfin Tazkiya Ramadhoni, dkk, Penerapan algoritma <i>Brute Force</i> pada aplikasi sidayko berbasis android, 2022	Untuk memberikan gambaran tujuan dan proses yang terjadi pada aplikasi Sidayko serta menggunakan <i>Entity Relational Diagram (ERD)</i> untuk menunjukkan hubungan antar entitas yang ada pada aplikasi Sidayko	Metode <i>Bruce Force</i>	Yaitu menunjukkan bahwa aplikasi Sidayko berhasil menerapkan algoritma <i>Brute Force</i> dengan tingkat akurasi mencapai 72%.

2.2 Aplikasi

Aplikasi disebut piranti atau sistem perangkat lunak dengan memanfaatkan kecanggihan komputer dan digunakan untuk melakukan tugas yang dibutuhkan dan

diinginkan oleh pengguna sehingga dapat memudahkan pengguna dalam memaksimalkan apa yang diinginkan. Dengan adanya aplikasi maka pekerjaan manusia bisa menjadi lebih ringan sehingga dapat menghemat waktu menjadi lebih efektif dan bisa untuk menghemat tenaga, sehingga dengan adanya aplikasi itu sangat mengubah kehidupan manusia karena dulunya hal atau pekerjaan yang berat sekarang bisa dilakukan dengan mudah karena adanya aplikasi. Jadi bisa diartikan aplikasi adalah program perangkat lunak yang dibuat dan digunakan untuk menjembatani antara manusia dengan komputer agar dapat melakukan dan memudahkan tugas ataupun kegiatan yang diinginkan oleh manusia.

2.3 Website

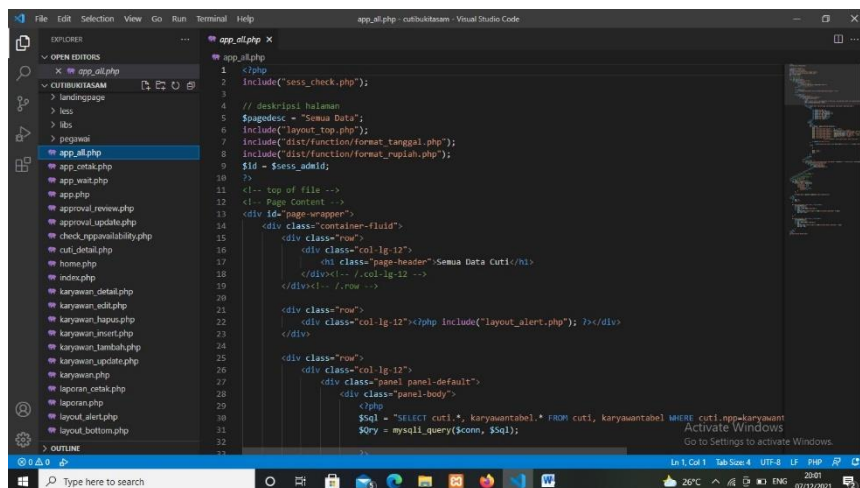
Website adalah kumpulan kumpulan halaman *web* yang di dalamnya terdapat sebuah domain mengandung informasi. Sebuah *website* biasanya dibangun atas banyak halaman *web* yang saling berhubungan. Jadi bisa dikatakan bahwa pengertian *website* adalah kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan melalui jaringan-jaringan halaman. Hubungan antara satu halaman *website* dengan halaman *website* lainnya disebut *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext* (Isa & Hartawan, 2017).

2.4 Visual Studio Code

Yaitu *Software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan *built-in* dukungan untuk *JavaScript*, naskah dan *Node.js* dan memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C ++, C # , *Python*, dan *PHP*. Hal ini didasarkan sekitar *Github* ini Elektron, yang merupakan versi *cross-platform* dari Atom komponen kode-editing, berdasarkan *JavaScript* dan *HTML*. Editor ini adalah fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja dengan teknologi cloud yang terbuka Microsoft. *Visual Studio Code*

menggunakan open source NET perkakas untuk memberikan dukungan untuk ASP.NET C # kode, membangun alat pengembang Omnisharp NET dan compiler Roslyn. Antarmuka yang mudah untuk bekerja dengan, karena didasarkan pada gaya explorer umum, dengan panel di sebelah kiri, yang menunjukkan semua file dan folder Anda memiliki akses ke panel editor di sebelah kanan, yang menunjukkan isi dari file yang telah dibuka. Dalam hal ini, editor telah dikembangkan dengan baik, dan menyenangkan pada mata. Ia juga memiliki fungsi yang baik, dengan intellisense dan autocomplete bekerja dengan baik untuk *JSON*, *CSS*, *HTML*, {kurang}, dan *Node.js*.

Visual Studio Code telah dirancang untuk bekerja dengan alat-alat yang ada, dan *Microsoft* menyediakan dokumentasi untuk membantu pengembang bersama dengan bantuan untuk bekerja dengan *ASP.NET 5*, *Node.js*, dan *Microsoft* naskah, serta alat-alat yang dapat digunakan untuk membantu membangun dan mengelola aplikasi *Node.js*. *Visual Studio Code* benar-benar sedang ditargetkan pada pengembang *JavaScript* yang ingin alat pengembangannya lengkap untuk *scripting server-side* mereka dan yang mungkin ingin usaha dari *Node.js* untuk kerangka berbasis NET. *Visual Studio Code*, adalah belum solid, lintas *platform* kode Editor ringan, yang dapat digunakan oleh siapa saja untuk membangun aplikasi untuk Web (Arief, 2017).



Gambar 2.1 Tampilan *Visual Studio Code*

2.5 Database

Menurut Adi Nugroho (2011;5) dalam jurnal Fery Wongso (*ISSN*: 1829 – 9822), basis data sebagai kumpulan terorganisasi dari data-data yang berhubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi serta dipanggil oleh pengguna. Terminologi hubungan berarti data mendeskripsikan *domain* (ranah) tertentu sehingga pengguna mudah untuk mendapatkan jawaban atas pertanyaan yang diajukan ke basis data tersebut. Sedangkan pengertian sistem basis data adalah sebagai koleksi dari data-data yang terorganisasi sedemikian rupa sehingga data mudah disimpan dan dimanipulasi (diperbarui, dicari, diolah dengan perhitungan-perhitungan tertentu, serta dihapus).

Manfaat Basis Data

Menurut Kadir (2003:254) dalam jurnal Toibah Umi Kalsum dan Siswanto Database juga memiliki tujuan-tujuan lain seperti berikut ini:

1. Kecepatan dan kemudahan (*speed*)

Pemanfaat database memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah.

2. Efisiensi ruang penyimpanan (*space*)

Dapat melakukan penekanan jumlah pengulangan data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi- relasi dalam bentuk file antar kelompok data yang saling berhubungan.

3. Keakuratan (*accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan atau batasan tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidak akuratan pemasukan atau penyimpanan data.

4. Ketersediaan (*availability*)

Pertumbuhan data baik dari sisi jumlah, maupun jenisnya sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal

tidak semua data itu selalu di butuhkan, karena itu kita dapat memilih-milih adanya data utama atau master, data transaksi, data historis, hingga data kadarluasa.

5. Kelengkapan (*completeness*)

Untuk mengakomodasikan kebutuhan kelengkapan data yang semakin berkembang, yaitu dengan menambah record-record data dan melakukan perubahan struktur dalam basis data, baik dalam bentuk penambahan objek baru tabel atau dengan penambahan file-file baru pada suatu tabel.

6. Keamanan (*security*)

Mencegah pengaksesan data oleh orang yang tidak berwenang. Kebersamaan pemakaian database dikelola oleh sistem aplikasi yang mendukung lingkungan multiuser.

2.6 Pengertian MySQL

Pada perkembangannya, *MYSQL* disebut juga *SQL* yang merupakan singkatan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. *SQL* pertama kali didefinisikan oleh *American National Standards Institute (ANSI)* pada tahun 1986. *MYSQL* adalah sebuah sistem manajemen database yang bersifat *open source*.

MYSQL merupakan sistem manajemen *database* yang bersifat *relational*. Artinya, data yang dikelola dalam *database* yang akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat. *MYSQL* dapat digunakan untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar. (Andi, Wahana Komputer, 2014).

SQL juga dapat diartikan sebagai antar muka standar untuk sistem manajemen *relasional*, termasuk sistem yang beroperasi pada komputer pribadi. *SQL* memungkinkan seorang pengguna untuk mengetahui dimana lokasinya, atau bagaimana informasi tersebut disusun. *SQL* lebih mudah digunakan dibandingkan dengan bahasa pemrograman, tetapi rumit dibandingkan *software* lembar kerja dan pengolah data. Sebuah pernyataan *SQL* yang sederhana dapat menghasilkan set permintaan untuk informasi yang tersimpan pada komputer yang berbeda

diberbagai lokasi yang tersebar, sehingga membutuhkan waktu dan sumber daya komputasi yang banyak. *SQL* dapat digunakan untuk *investigasi interaktif*, atau pembuatan laporan ad hoc atau disisipkan dalam program aplikasi.

Jadi, *SQL* adalah permintaan yang melekat pada suatu *database* atau *SMBD* tertentu. Dengan kata lain, *SQL* adalah perintah atau bahasa yang melekat di dalam *SMBD*. Sebagai suatu bahasa permintaan, *SQL* didukung oleh *SMBD*, seperti *MySQL Server*, *MySQL*, *PostgreSQL*, *Interbase*, dan *Oracle*. Selain itu *SQL* juga didukung oleh *database* bukan server, seperti *MS Access* maupun *Paradox*.(Surniawan, Wahana Eri Mardiani, 2014)



Gambar 2.2 Lambang *MySQL*

2.7 PHP

Menurut Luthfi (2017) *PHP* adalah bahasa yang dirancang secara khusus untuk penggunaan pada Web. *PHP* adalah tool untuk pembuatan halaman *web* dinamis. Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page* (Situs Personal). *PHP* pertama kali dibuat oleh *Rasmus Lerdorf* pada tahun 1995. Pada waktu itu *PHP* masih bernama *FI* (Form Interpreted), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*. Saat ini *PHP* adalah singkatan dari *PHP: Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: *PHP: Hypertext Preprocessor*.

Beberapa kelebihan *PHP* menurut Fridayanthie & Mahdiati (2016) yaitu sebagai berikut:

- a. *PHP* merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.

- b. *PHP* dapat berjalan pada *web server* yang dirilis oleh *Microsoft*, seperti *IIS* atau *PWS* juga pada *apache* yang bersifat *open source*.
- c. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada *PHP* lebih cepat dan mudah, karena banyak milis-milis dan *developer* yang siap membantu pengembangannya.
- d. Jika dilihat dari segi pemahaman, *PHP* memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
- e. *PHP* bersifat bebas dipakai (*free*).

```
<body>
  <?php
    // Tulis disini kode PHP
  ?>
</body>
```

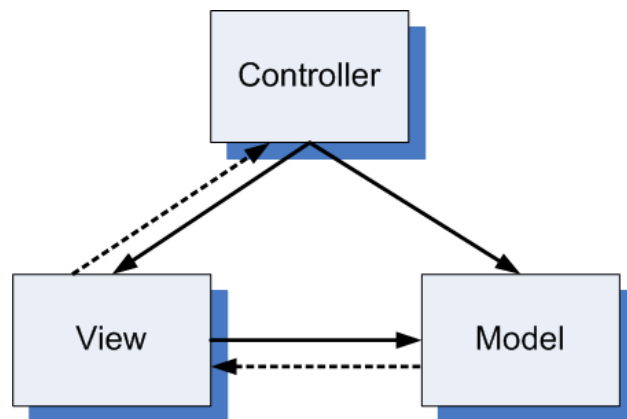
Gambar 2.3 *Syntax* bahasa *PHP*

2.8 *CodeIgniter*

Menurut Destriningrum M & Andrian (2017) “*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 awal”. Dalam situs resmi *CodeIgniter (Official Website CodeIgniter, 2002)* menyebutkan bahwa *codeigniter* merupakan *framework PHP* yang kuat dan sedikit *bug*. *CodeIgniter* ini dibangun untuk para pengembang dengan bahasa pemrograman *PHP* yang membutuhkan alat untuk membuat *web* dengan fitur lengkap. *Framework CodeIgniter* dikembangkan oleh Rick Ellis, CEO Ellislab, Inc. Kelebihan dari *framework CodeIgniter* jika dibandingkan dengan *framework* lain adalah sebagai berikut :

1. Gratis (*Open-Source*) Kerangka kerja *Codeigniter* memiliki lisensi dibawah *Apache/BSD open-source* sehingga bersifat bebas atau gratis.

2. Berukuran kecil. Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan *framework* lain yang berukuran besar dan membutuhkan *resource* yang besar dan juga dalam eksekusi maupun penyimpanannya.
3. Menggunakan konsep M-V-C. *Codeigniter* merupakan konsep M-V-C (*Model-View-Controller*) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Dengan konsep ini kode PHP, *query MySQL*, *JavaScript* dan CSS dapat saling dipisah-pisahkan sehingga ukuran *file* menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau *maintenance*.
 - a. *Model* merupakan kode program (berupa OOP *class*) yang digunakan untuk berhubungan dengan *database MySQL* sekaligus untuk memanipulasinya (*input-edit-delete*).
 - b. *View* merupakan kode program berupa *template* atau PHP untuk menampilkan data pada *browser*.
 - c. *Controller* merupakan kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol model dan *view*. Adapun alur dari program aplikasi berbasis *CodeIgniter* yang menggunakan konsep M-V-C ditunjukkan pada gambar Gambar 3.7.



Gambar 2.4 Konsep Aliran M-V-C

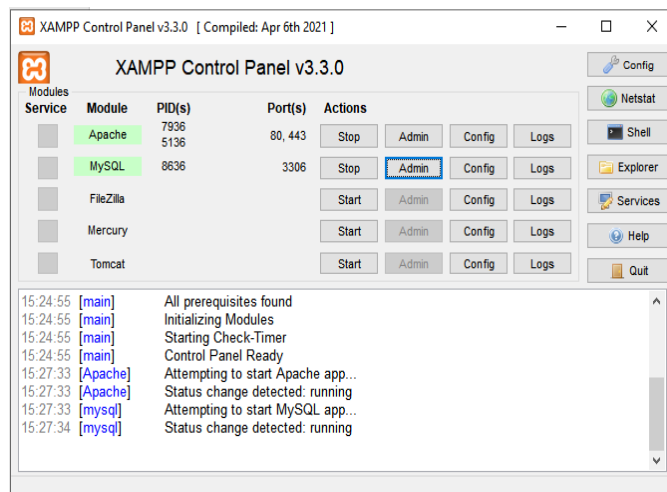
2.9 CSS

CSS atau singkatan dari *Cascading Style Sheet* adalah suatu aturan untuk mengatur tampilan dari website sehingga tampilan dalam web lebih terstruktur.

CSS sendiri bukanlah bahasa pemrograman, CSS lebih seperti konfigurasi tampilan dari suatu *tag* pada *website*. CSS dapat merubah text, warna, *background* dan posisi dari suatu *tag* (Marlina & Masnur, 2021).

2.10 XAMPP

Menurut Umagapi, D., & Ambarita, A. (2018) *XAMPP* adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi *XAMPP* sendiri adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama *XAMPP* sendiri merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah untuk digunakan yang dapat menampilkan halaman *web* yang dinamis.



Gambar 2.5 Tampilan *XAMPP*

2.11 HTML

Hypertext Markup Language (HTML) adalah suatu system untuk menambahkan dokumen dengan table yang mennandakan bagaimana teks di dokumen harus disajikan dan bagaimana dokumen dihubungkan bersama-sama. Di dalam skema *HTML* terdapat kekuatan untuk membuat aplikasi client-server, multimedia, form,

interaktif. *HTML* sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu system operasi tertentu. Secara teknis, *HTML* didefinisikan sebagai *Standard Generalized Markup Language (SGML)*. Sebuah dokumen *HTML* dapat dikatakan contoh sebuah dokumen SGML (Sulhan, 2006).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Document</title>
  </head>
  <body>
    <!-- Tulis disini -->
  </body>
</html>
```

Gambar 2.6 *Syntax* dasar bahasa pemrograman *HTML*

2.12 *JavaScript*

Menurut Marlina & Masnur (2021) *Javascript* adalah bahasa pemrograman yang mendekati bahasa manusia atau bisa dikatakan bahasa tingkat tinggi, maka dari itu *javascript* mudah di pelajari. *Javascript* sendiri tujuannya di buat untuk memperkaya fitur pada website agar lebih dinamis, seperti untuk menampilkan dan menghilangkan objek-objek pada website kemudian dengan fungsi *javascript* dapat memanggil kembali objek yang di hilangkan tersebut.



Gambar 2.7 Lambang *Javascript*

2.13 *Metode Brute Force*

Algoritma yang dapat digunakan untuk String Matching salah satunya adalah algoritma *Brute Force*. Algoritma *brute force* memecahkan masalah dengan sangat sederhana, langsung, dan jelas. Algoritma *brute-force* merupakan suatu teknik yang biasa digunakan bila si penyusun algoritma lebih mempertimbangkan

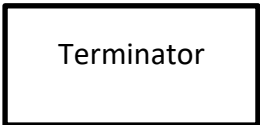
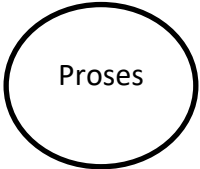
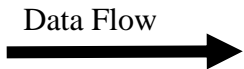

memperoleh solusi dari problem secara langsung. Cara kerja yang dilakukan algoritma *brute force* adalah membandingkan karakter demi karakter antar kata atau string yang dicari dengan string sumber. Apabila tidak sesuai maka akan dilakukan penggeseran posisi dari kiri ke kanan. Demikian seterusnya sampai ditemukan string yang dicari (Danuri, 2016). Secara rinci langkah-langkah yang digunakan algoritma *Brute Force* untuk mencocokkan string adalah sebagai berikut:

1. Pencocokan pattern Algoritma *Brute Force* dimulai dari awal teks.
2. Algoritma *Brute Force* akan mencocokkan karakter per karakter pattern dengan karakter pada teks yang bersesuaian dari kiri ke kanan, sampai salah satu kondisi berikut terpenuhi:
 - a. Karakter di pattern yang dibandingkan cocok maka pencarian selesai.
 - b. Apabila dijumpai ketidak-cocokan antara pattern dengan teks, maka pencarian tidak cocok dan belum berhasil.
3. Kemudian algoritma *Brute Force* akan melakukan penggeseran pattern sebesar satu ke kanan dan mengulangi langkah ke2 sampai pattern berada di ujung teks.

2.14 Data Flow Diagram (DFD)

Menurut (Muhamad Muslihudin, Oktafianto, 2016:46) dalam jurnal Agus Ramdhani Nugraha dan Gati Pramukasari (*ISSN : 2338-1477*) *Data Flow Diagram (DFD)* adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan ke mana tujuan data yang keluar dari sistem, di mana data tersimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data tersimpan dan proses yang dikenakan pada data tersebut.

Tabel 2.1 Data Flow Diagram



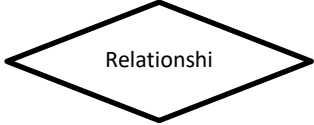
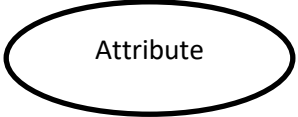

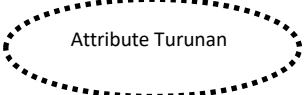
NO	SIMBOL	KETERANGAN
1.		Kesatuan diluar sistem (<i>external entity</i>) yang memberikan input kesistem / menerima output dari sistem berupa oraganisai, orang, atau sistem lain.
2.		Aktivitas yang mengolah input menjadi output.
3.		Aliran data pada sistem (antar proses, antara proses & terminator, serta antara proses& data store).
4.		Penyimpanan data pada <i>database</i> , biasayanya berupa tabel.


(Sumber : Saputra, 2014)

2.15 Entity Relational Diagram (ERD)

Menurut (Oetomo, 2002) dalam jurnal Rohayati dan Agus Irwandi HJ (*ISSN:1412-5609*), ERD adalah suatu diagram yang digunakan untuk memodelkan struktur data dan hubungan antar data. *ERD* berfungsi untuk menggambarkan relasi dari dua file atau dua tabel yang dapat di golongankan dalam tiga macam bentuk relasi, yaitu satu-satu, satu-banyak dan banyak-banyak.

Tabel 2.2 Entity Relational Diagram (ERD)

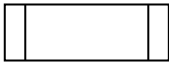
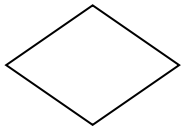


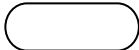
NO	SIMBOL	KETERANGAN
1.		<p><i>Entitas</i> ialah suatu objek yang dapat dibedakan dengan objek lainnya. <i>Entitas</i> berfungsi untuk memberikan identitas pada <i>entitas</i> yang memiliki label dan nama.</p>
2.		<p><i>Entitas</i> yang lemah adalah entitas yang harus ditentukan oleh hubungan kunci asing dengan entitas lain karena tidak dapat diidentifikasi secara unik dengan atributnya sendiri.</p>
3.		<p>Relasi ialah hubungan yang terjadi antara 1 <i>entitas</i> atau lebih yang tidak mempunyai fisik tetapi hanya sebagai konseptual. Dan berfungsi untuk mengetahui jenis hubungan yang ada antara 2 file.</p>
4.		<p>Atribut ialah karakteristik dari entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut. Dan berfungsi untuk memperjelas atribut yang dimiliki oleh sebuah entitas</p>
5.		<p>Atribut multivalued dapat memiliki lebih dari satu nilai. Misalnya, entitas karyawan dapat memiliki beberapa nilai keterampilan.</p>
6.		<p>Atribut turunan didasarkan pada atribut lain. Misalnya, gaji bulanan karyawan didasarkan pada gaji tahunan karyawan.</p>

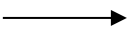



7.	<p style="text-align: center;">Alur</p> 	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi. Dan berbentuk garis.
----	---	---

(Sumber : Firman, 2014)

2.16 Pengertian *Flowchart*

Menurut Wahyudi (2020) *flowchart* adalah suatu gambaran urutan logika dari suatu prosedur pemecah masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian masalah yang di tuliskan dalam simbol-simbol tertentu. diagram alir ini selain dibutuhkan sebagai alat komunikasi, juga diperlukan sebagai dokumentasi. Berdasarkan pengertian tersebut dapat disimpulkan bahwa pengertian *flowchart* adalah sebuah bagan yang terdiri dari alur atau urutan serta simbol-simbol tertentu untuk menggambarkan urutan logika dari sebuah permasalahan.

Simbol	Nama	Fungsi
	<i>Predefined Process</i> (Sub Program)	Permulaan sub program/proses menjalankan sub program
	<i>Decision</i>	Perbandingan, pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
	<i>On Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman
	<i>Off page Connectoor</i>	Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda
	<i>Terminator</i>	Permulaan/akhir program

	Garis Alir (<i>Flow Line</i>)	Arah aliran program
	Proses	Proses penghitung/ proses pengolahan data
	<i>Input/output data</i>	Proses <i>input/output</i> data, parameter, informasi
	<i>Preparation</i>	Arah aliran Program

(Sumber : Anharku, 2009)