

BAB II

TINJAUAN PUSTAKA

2.1 Penilaian

Dalam dunia pendidikan, penilaian atau asesme (*assessment*) diartikan sebagai prosedur yang digunakan untuk mendapatkan informasi untuk mengukur taraf pengetahuan dan keterampilan subjek didik yang hasilnya akan digunakan untuk keperluan evaluasi. Menurut (Pramana & Putra, 2019) mendefinisikan penilaian (*assessment*) adalah penetapan berbagai cara dan menggunakan beragam alat penilaian untuk memperoleh informasi tentang seberapa jauh hasil belajar peserta didik atau ketercapaian kompetensi (rangkaiannya kemampuan). Hasil penilaian dapat berupa nilai kualitatif (pernyataan naratif dalam kata-kata) dan kuantitatif (berupa angka).

Penilaian hasil belajar oleh pendidik adalah proses pengumpulan sebuah informasi ataupun data tentang capaian pembelajaran peserta didik dalam aspek sikap, aspek pengetahuan, dan aspek keterampilan yang dilakukan secara terencana dan sistematis yang dilakukan untuk memantau proses, kemajuan belajar, dan perbaikan hasil belajar melalui penugasan dan evaluasi hasil belajar. Sedangkan penilaian hasil belajar oleh satuan pendidikan adalah proses pengumpulan suatu informasi atau data tentang capaian pembelajaran peserta didik dalam aspek pengetahuan dan aspek keterampilan yang dilakukan secara terencana dan sistematis dalam bentuk penilaian akhir dan ujian sekolah yang dilakukan dengan cara melakukan ujian semester (Wartulus, 2020).

Penilaian berfungsi mengetahui pencapaian hasil belajar peserta didik. Selain itu penilaian dapat meningkatkan kemampuan peserta didik dalam belajar. Penilaian dapat dilakukan melalui 3 pendekatan, yaitu penilaian akhir pembelajaran (*assessment of learning*), penilaian untuk pembelajaran (*assessment for learning*), dan penilaian sebagai pembelajaran (*assessment as learning*). Penilaian akhir pembelajaran adalah penilaian yang dilaksanakan setelah proses pembelajaran selesai. Penilaian untuk pembelajaran merupakan penilaian yang dilakukan selama proses belajar mengajar berlangsung dan digunakan sebagai

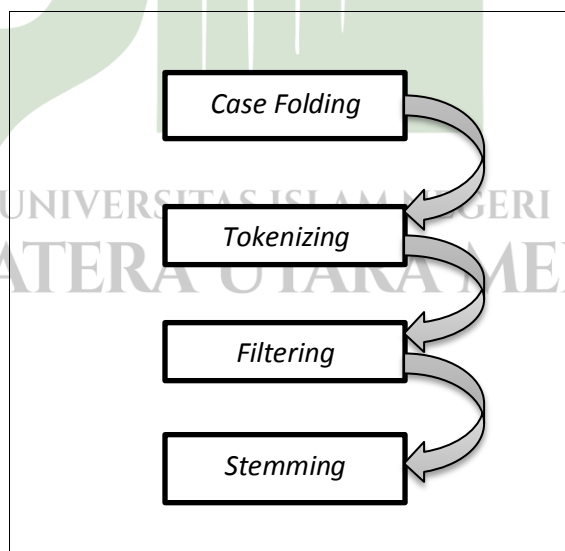
dasar untuk melakukan proses perbaikan belajar mengajar (Tune Sumar & Tune Sumar, 2020).

Penilaian pada dasarnya bertujuan untuk mengetahui perkembangan hasil belajar siswa dan hasil mengajar guru. Hasil belajar siswa digunakan untuk memotivasi siswa, dan untuk perbaikan serta peningkatan kualitas pembelajaran oleh guru, dan sekolah.

2.2 *Text Mining*

Menurut buku *The Text Mining Handbook*, *text mining* didefinisikan sebagai suatu proses menggali sebuah informasi di mana seorang pengguna berinteraksi dengan sekumpulan dokumen menggunakan alat analisis yang merupakan bagian dari data *mining*. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Sumber data yang digunakan pada *text mining* adalah sekumpulan teks.

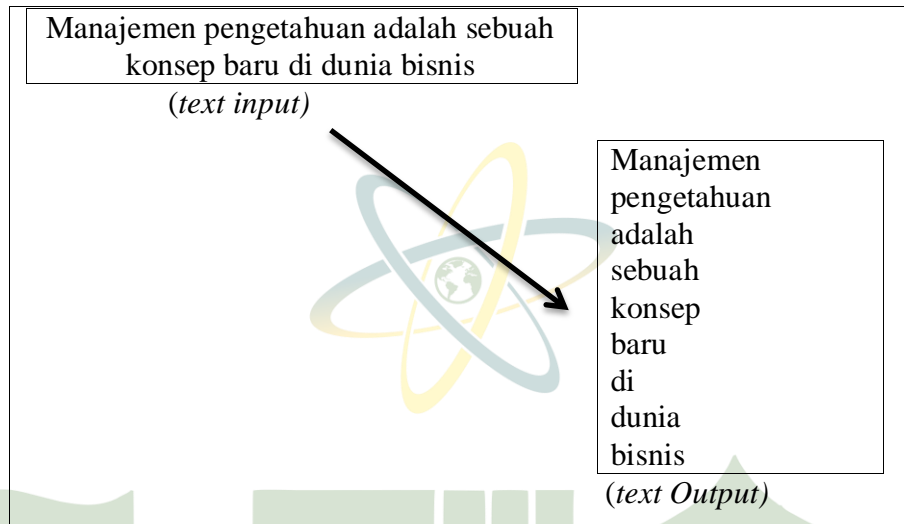
Teks yang diproses oleh *text mining* biasanya melalui pemrosesan teks dan memiliki beberapa karakteristik, seperti dimensi yang besar, noise pada data, dan struktur teks yang buruk. Sebelum mempelajari teks, proses *pre-processing* yang biasa dilakukan pada dokumen memiliki beberapa tahap yang harus dilalui, yaitu: *case folding*, *tokenizing*, *filtering*, dan *stemming*.



Gambar 2.1 Tahap *pre-processing*

2.2.1 Case Folding dan Tokenizing

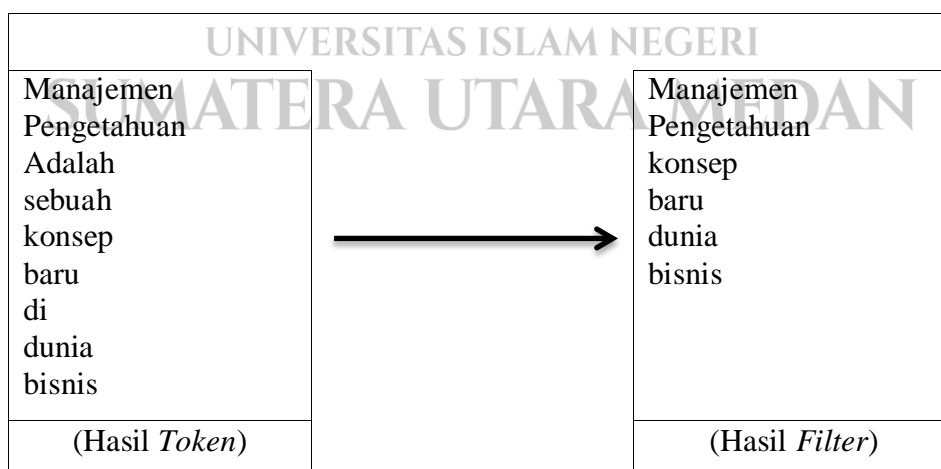
Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf “a” sampai dengan “z” diterima. Karakter selain huruf dihilangkan dan dianggap delimiter. Tahap *tokenizing*/parsing adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya.



Gambar 2.2 Contoh *Case Folding* dan *Tokenizing*

2.2.2 Filtering

Filtering merupakan bagian tahap menyimpan kata-kata yang penting dari hasil token. Bisa menggunakan algoritma *stoplist* (membuang kata-kata yang tidak penting) atau *worldlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata yang tidak deskriptif yang dihilangkan dalam pendekatan *bag-of-words*. Contoh kata dari *stopwords* adalah “Yang”, “di”, “dari”, “dan” dan seterusnya.



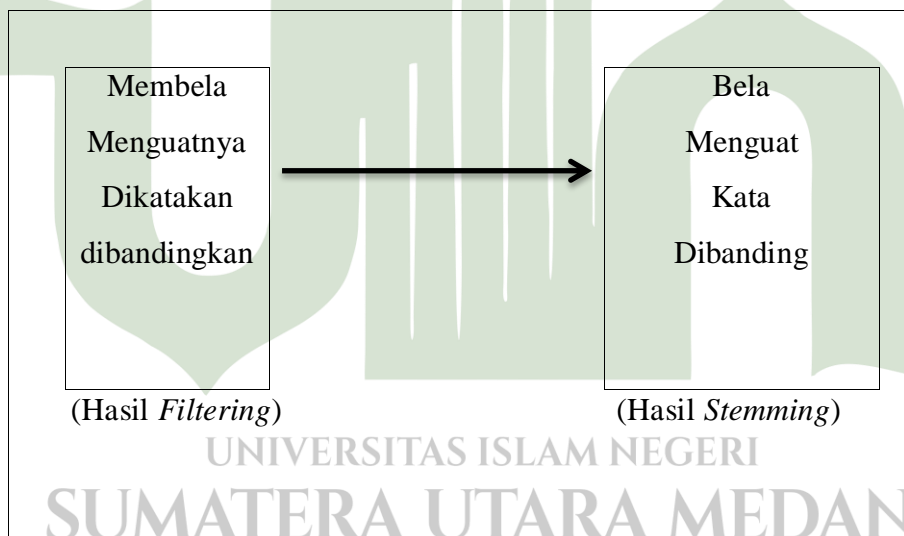
Gambar 2.3 Contoh *Filtering*

2.2.3 Stemming

Pencarian bentuk dasar sebuah kalimat dengan menghilangkan imbuhan-nya disebut sebagai *stemming*. Didalam *stemming* terdapat sebuah proses dalam sistem IR yang dapat mengubah kata-kata dalam dokumen ke kata-kata akar-nya (*root word*) dengan menggunakan fungsi aturan tertentu

karena bahasa Indonesia lebih kompleks dari penggunaan bahasa Inggris, *stemming* sangat penting untuk meningkatkan efisiensi pencarian informasi. Bahasa Indonesia memiliki awalan (prefiks), sisipan (infiks), akhiran (sufiks), gabungan prefiks dan sufiks (konfiks) dan gabungan imbuhan. Dengan demikian, *stemming* bahasa Indonesia harus memiliki kemampuan untuk menemukan akar kata yang sesuai dengan aturan baku bahasa.

Stemming digunakan untuk mengubah bentuk suatu kata menjadi kata dasar yang sesuai dengan morfologi Bahasa Indonesia yang tepat. Untuk mengubah bentuk suatu kata menjadi kata dasar, imbuhan-nya harus dihilangkan.



Gambar 2.4 Contoh *Stemming*

2.3 Algoritma Rabin-Karp

Algoritma Rabin-Karp adalah algoritma yang melakukan pencocokan string yang menggunakan fungsi *hash* sebagai pembanding antara string yang dicari (*m*) dengan substring pada teks (*n*). Apabila *hash value* keduanya sama maka akan dilakukan perbandingan sekali lagi terhadap karakter-karakternya (Prima Putra & Sularno, 2019).

Menurut Michael O. Rabin dan Richrad M. Karp, Algoritma Rabin-Karp dikembangkan sekitar tahun 1987 dan menggunakan fungsi *hash* untuk menemukan pola dalam sebuah *string* teks (Filcha & Hayaty, 2019).

Rabin-Karp merupakan algoritma *string matching* yang memiliki prinsip dasar yaitu mencari persamaan pola pada sebuah *text* dengan *text* pembanding dengan memanfaatkan *substring* pada sebuah *text* menggunakan *hashing* (Bahri & Wajhillah, 2020). Terdapat beberapa tahap dalam Rabin-Karp, diantaranya:

2.3.1 Parsing K-Gram.

Kalimat	:	komputer adalah perangkat Elektronik
Preprocessing	:	komputerperangkatelektronik
K-Gram	:	{komp} {ompu} {mput} {pute} {uter} {terp} {erpe} {rper} {pera} {eran} {rang} {angk} {ngka} {gkat} {kate} {atel} {tele} {elek} {lekt} {ektr} {ktro} {tron} {roni} {onik}

Gambar 2.5 Contoh K-Gram (Filcha & Hayaty, 2019)

K-Gram digunakan untuk memotong karakter huruf sejumlah K dari sebuah kalimat atau kata secara berurutan hingga keseluruhan kalimat terurai. K-Gram adalah serangkaian kata dengan panjang K yang biasanya digunakan dalam sebuah teks (Prima Putra & Sularno, 2019).

2.3.2 Rolling Hash

Rolling hash merupakan fungsi yang digunakan untuk menghasilkan nilai *hash* dari rangkaian *gram* dalam algoritma Rabin-Karp adalah dengan menggunakan fungsi *rolling hash*. Fungsi *hash* $H(c_1...c_2)$ didefinisikan sebagai berikut:

$$c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b^k + c^k \dots \dots \dots (2.1)$$

Keterangan :

c : nilai ASCII karakter

b : basis (bilangan prima)

k : banyak karakter

Tabel 2.1 Contoh Rolling Hash (Filcha & Hayaty, 2019)

Atribut	Nilai Array
Rolling	[0] => maka
hash	m=109, a=97, k=107, a=97, basis=11, mod=
pertama	10007
	$H=c_m*b^{(k-1)}+c_a*b^{(k-2)}+c_k*b^{(k-3)}$ $+c_a*b^{(k-4)}$ $H=109*11^3+97*11^2+107*11^1+97*11^0$ $H=145079+11737+1177+97$ $H=158090 \text{ Mod } 10007$ $H=7985$
Rolling	[1] => akan
hash	a = 97, k=107, a=97, n=110, basis=11, mod =
kedua	10007
	$H=c_m*b^{(k-1)}+c_a*b^{(k-2)}+c_k*b^{(k-3)}$ $+c_a*b^{(k-4)}$ $H=97*11^3+107*11^2+97*11^1+110*11^0$ $H=12907+12947+1067+110$ $H=143231 \text{ Mod } 10007$ $H= 3133$

Metode *hashing* digunakan untuk mempercepat pencarian atau pencocokan suatu *string*. Apabila tidak di-*hash*, pencarian akan dilakukan karakter per karakter. Namun pencarian akan menjadi lebih bagus setelah di-*hash* karena hanya akan membandingkan empat digit angka untuk mengetahui kesamaan suatu *substring* (Arnawa, 2019).

2.3.3 Dice's Similarity Coefficients

Tahapan ini merupakan tahapan terakhir dari algoritma rabin karp dengan menghitung kemiripan dokumen (Billhaqqi et al., 2020). Persamaan *dice's similarity coefficients* sebagai berikut:

$$S = \frac{2C}{A+B} * 100 \dots\dots\dots (2.2)$$

Keterangan :

S : *Similarity*

A : Jumlah *K-Gram* dari dokumen 1

B : Jumlah *K-Gram* dari dokumen 2

C : Jumlah *K-Gram* dengan nilai *hash* yang sama pada kedua dokumen

Berikut contoh menggunakan persamaan *similarity coefficients* di atas dan diketahui nilai dari jumlah dokumen dengan keterangan:

$$S = \frac{2C}{A+B} * 100$$

A = 42 *K-Gram*

B = 40 *K-Gram*

C = 32 *K-Gram*

Melakukan perhitungan:

$$S = \frac{2C}{A+B} * 100$$

$$S = \frac{2*32}{42+40} * 100$$

$$S = 78.05\%$$

Berdasarkan perhitungan diatas, dapat diketahui persentase *similarity* antara dokumen pertama dan kedua adalah 78.05% (Filcha & Hayaty, 2019).

2.4 Konsep Dasar Web

Web atau lengkapnya *www (world wide web)* adalah sebuah koleksi keterhubungan dokumen-dokumen yang disimpan diinternet dan diakses menggunakan *Protocol (Hyper Text Transfer Protocol)* (Sarwindah, 2018).

Menurut (Journal, 2020) definisi *web* adalah sebuah dokumen yang dapat ditulis dalam *Hype Text Markup Language (HTML)* dan dapat diakses melalui

Hyper Text Transfer Protocol (HTTP). Protokol adalah sebuah bahasa pemrograman untuk mentransfer data dari situs *web* pusat untuk ditampilkan kepada pengguna dari program pembaca data di situs *web*.

2.5 Website

Menurut (Journal, 2020) situs *web* terdiri dari kumpulan halaman yang saling berhubungan yang menampilkan informasi seperti teks, gambar diam atau bergerak, animasi, suara, dan atau kombinasi dari elemen-elemen ini baik statis maupun dinamis. Halaman-halaman ini terhubung satu sama lain melalui jaringan halaman, yang juga dikenal sebagai *Hyperlink*

Situs *web* merupakan sebuah halaman *web* yang saling terhubung yang pada umum-nya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, atau organisasi (Nasution, 2018).

Menurut (Hanif Nur Rosid, 2022) *web browser* adalah aplikasi untuk dapat menjelajah internet yang berfungsi untuk menampilkan dan berinteraksi dengan server.

Dilihat dari segi isi ataupun *content*, jaringan *web* dibedakan menjadi dua jenis, yaitu jaringan statis dan jaringan dinamis. Jaringan statis adalah jaringan yang isinya tidak berubah. Intinya isi dokumen tidak bisa diubah dengan cepat dan mudah. Ini dikarenakan teknologi yang digunakan untuk membuat dokumen dalam jaringan sangat tidak memungkinkan perubahan konten atau data. Teknologi yang digunakan pada *website* statis adalah jenis *client-side scripting* seperti HTML atau *Cascading style Sheet* (CSS). Contoh situs *web* statis adalah halaman profil perusahaan yang utamanya menggunakan animasi *flash* atau halaman koleksi produk animasi.

Menurut (Laugi, 2018) *web* dinamis adalah suatu *web* konten atau isinya dapat berubah-ubah setiap saat. Sebab dalam teknologi pembuatan *web* dinamis sudah dirancang semudah mungkin bagi pemakai atau *user* yang menggunakan *web* tersebut. Konten yang terdapat di *web* dinamis pun tersimpan di sebuah database, sehingga bagi orang-orang yang tidak bisa akan *coding*-pun dapat merubah isi konten tersebut tanpa harus menguasai bahasa pemrograman *web*

yang biasa disebut dengan *coding*. Sebuah *website* dinamis umumnya dibuat menggunakan *Compact Management System* (CMS) tertentu dengan penyimpanan data di database (misalnya seperti MySQL) dan biasanya juga *web* ini tersusun dari bahasa pemrograman seperti HTML, CSS, PHP, JavaScript dan berhubungan dengan database MySQL. Contoh situs *web* dinamis diantaranya adalah situs berita, situs *web e-Commerce* dan situs *e-Banking*.

2.6 *Web Server*

Menurut (Nurkamiden et al., 2017) *web server* adalah sebuah *software* dalam sebuah *server* yang berfungsi menerima permintaan (*Request*) berupa halaman *website* melalui HTTP atau HTTPS dari *Client* (*Browser*) dan mengirimkan kembali (*Response*) dalam bentuk halaman-halaman *website* yang umumnya berbentuk HTML.

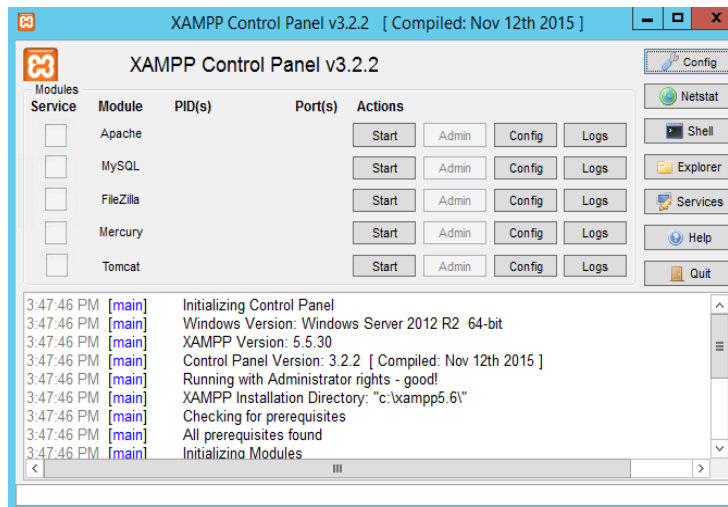
Contoh *web server* yang dipakai adalah:

2.6.1 *XAMPP for Windows*

Menurut (Cahyono et al., 2019) XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. XAMPP berperan sebagai *server web* pada komputer. XAMPP juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu untuk melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses *internet*.



Gambar 2.6 Aplikasi XAMPP



Gambar 2.7 Control Panel XAMPP

Pengertian bagian-bagian XAMPP yang biasa digunakan pada umumnya, sebagai berikut:

- a. *Htdocs* adalah tempat penyimpanan berkas-berkas yang akan dijalankan, seperti *file* PHP, HTML, dan *script* lain dalam sebuah folder.
- b. *PhpMyadmin* digunakan untuk mengelola basis data *Mysql* di komputer.
- c. *Control panel* berfungsi untuk mengelola layanan (*service*) XAMPP, seperti menghentikan (*stop*) layanan, ataupun memulai (*start*) layanan.

2.6.2 IIS (*Internet Information Services*)

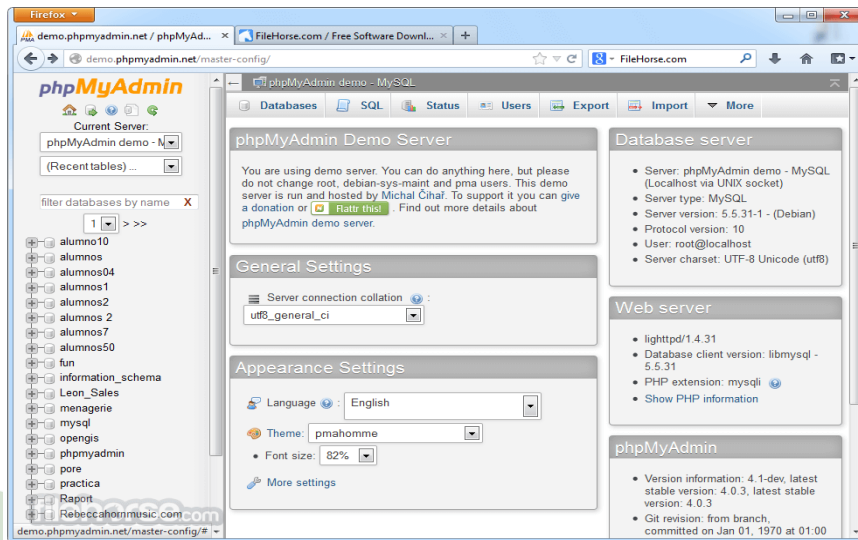
IIS atau juga biasa dikenal dengan *Internet Information Service*, adalah sebuah HTTP *web server* seperti *APACHE* yang digunakan pada *server Windowsi*, seperti *Windows NT 4*, *Oserver*, *Windows 2000*, dan *Windows 2003*.

2.6.3 PWS (*Personal Web Server*)

Komputer memiliki program server, maka komputer tersebut dianggap sebagai *web server*. PWS (*Personal Web Server*) digunakan untuk mengakses komputer user dibagian halaman *web* yang ada di dalam komputer server *web*.

2.7 PHP (Personal Home Page)

PHP atau juga dikenal sebagai *Hypertext PreProcessor* adalah bahasa pemrograman *interpreter* yang memungkinkan penerjemah melakukan penerjemahan baris kode sumber ke kode mesin *side programing*, hal ini dikarenakan seluruh prosesnya dijalankan pada server (Ayu et al., 2020).



Gambar 2.8 Contoh Tampilan PHP MyAdmin

PHP adalah bahasa pemrograman *script* yang umum digunakan oleh kalangan programmer untuk membuat situs *web* dinamis. Namanya adalah *Interpreter Form Hypertext Preprocessing* (Cipta et al., 2017).

PHP merupakan bahasa berbentuk *script* yang ditempatkan didalam server baru kemudian diproses. Lalu, hasil pemrosesan dikirim kepada *web browser* klien. Bahasa pemrograman ini dirancang khusus untuk membentuk *web* dinamis. Artinya, pemrograman PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini, misalnya halaman-halaman menampilkan daftar tamu. Halaman tersebut akan selalu mengalami perubahan mengikuti jumlah data tamu yang telah mengisi buku tamu (Rubiati, 2018).

Adapun kelebihan-kelebihan dari PHP yaitu:

1. PHP adalah bahasa *scripting* yang tidak dikompilasi dengan PHP. Tidak seperti bahasa pemrograman aplikasi lainnya.

2. PHP dapat berjalan pada *web server Microsoft*, seperti IIS atau PWS, dan juga pada *apache* yang bersifat *open source*.
3. Karena bersifat *open source*, perubahan dan peningkatan interpreter PHP lebih cepat dan mudah dikarenakan banyak pengembang yang bersedia untuk membantu mengembangkan PHP.
4. Dalam hal pemahaman, PHP sangat mudah untuk dipahami dikarenakan memiliki banyak referensi sebagai bahan acuan.
5. PHP bersifat bebas pakai (*free*). PHP juga dapat digunakan pada tiga sistem operasi, yaitu: Linux, Unix, dan *Windows*. Selain itu, PHP juga dapat dijalankan secara *runtime* pada konsol

2.8 MySQL

Menurut (Hermiati et al., 2021) MySQL merupakan suatu jenis database server yang sangat terkenal. MySQL termasuk jenis RDMS (*Relational Database Management System*). MySQL mendukung bahasa pemrograman PH, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI. MySQL merupakan RDMS (*Relational Database Management System*) server. RDMS adalah program yang memungkinkan pengguna *database* untuk membuat, Mengelola, dan menggunakan data pada suatu model *relational*.

MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung atau *support* dengan *database* MySQL.

2.8.1 Perintah Dasar MySQL

1. *Create Database*, perintah untuk membuat *database*.
2. *Use*, perintah menggunakan *database*.
3. *Create Table*, perintah untuk membuat tabel.
4. *Drob Database*, perintah untuk menghapus *database*.

2.8.2 Perintah MySQL pada sub DML

1. DML (*Data Manipulation Language*) adalah paket bahasa yang digunakan untuk melakukan manipulasi terhadap data pada sebuah basis

data. Manipulasi data meliputi kegiatan menambah, mengedit, mengambil, dan menghapus data.

2. *Insert*, perintah untuk memasukkan data.
3. *Select*, perintah untuk menampilkan data.
4. *Update*, perintah untuk mengubah atau mengedit data.
5. *Delete*, perintah untuk menghapus data.

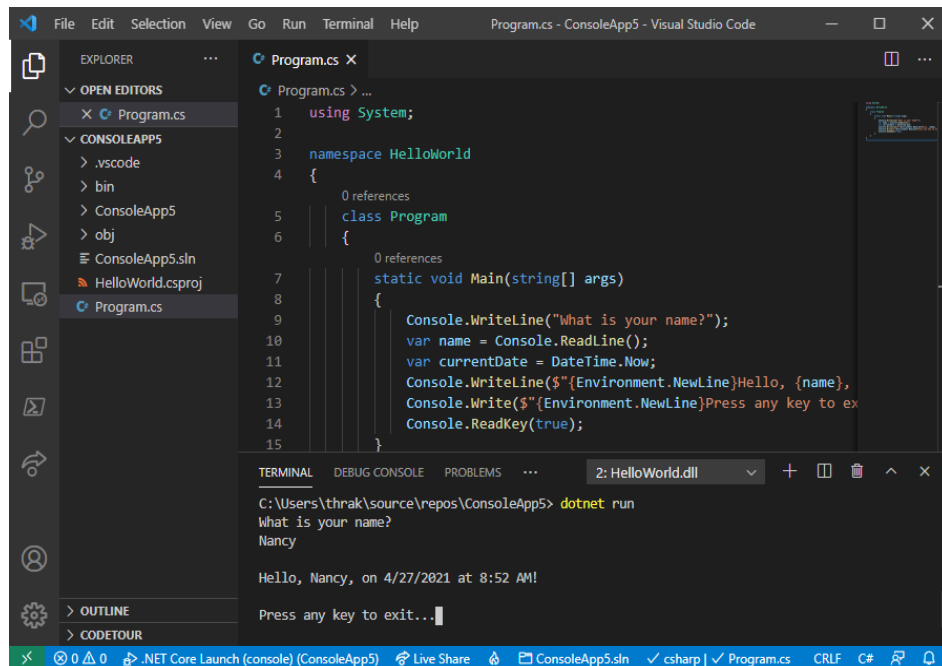
2.9 Visual Studio Code

Visual studio code adalah sebuah aplikasi editor *code open source* yang dikembangkan oleh Microsoft untuk sistem operasi Windows, Linux, MacOS. Visual code memudahkan dalam menuliskan *code* yang mendukung beberapa jenis pemrograman, seperti C++, C#, Java, Python, PHP, GO. Visual code memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberikan sebuah variasi warna sesuai dengan fungsi dalam rangkaian *code* tersebut. Visual Studio Code juga telah terintegrasi ke Github. Selain itu fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di Visual Studio Code (Ramdhan & Nufriana, 2019).

Banyak sekali fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya *Intellisense*, *Git Integration*, *Debugging*, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VS Code dengan teks editor yang lain (Permana & Romadlon, 2019).



Gambar 2.9 Icon Visual Studio Code



Gambar 2.10 Tampilan Program Visual Studio Code

2.10 Aliran Sistem Informasi

Menurut (Novendri, 2019) bahwa aliran sistem informasi sangat berguna untuk mengetahui permasalahan yang ada pada suatu sistem. Dari sini dapat diketahui bahwa apakah sistem informasi tersebut masih layak dipakai atau tidak, masih manual atau komputerisasi. Jika sistem informasinya tidak layak lagi maka perlu adanya perubahan dalam pengolahan datanya sehingga menghasilkan informasi yang cepat dan akurat serta keputusan yang lebih baik.

2.10.1 Flowchart

Flowchart, juga dikenal sebagai diagram alir. *Flowchart* adalah jenis diagram yang menunjukkan algoritma ataupun langkah-langkah instruksi yang berurutan dalam sebuah sistem. Diagram alir digunakan sebagai bukti dokumentasi untuk analisis sebuah sistem, yang menawarkan kepada seorang programmer bagaimana gambaran logis dari sistem yang akan dibangun. Oleh karena itu, diagram *flow* dapat membantu menyelesaikan masalah yang mungkin muncul selama proses pembangunan sistem. Pada dasarnya, *flowchart* menggunakan simbol-simbol untuk menunjukkan proses tertentu dan garis yang menghubungkan proses satu ke proses yang lainnya.

Bagan alir atau *flowchart* adalah bagan yang menunjukkan alir didalam program atau prosedur sistem secara logika (Cipta et al., 2017).

Dengan adanya *flowchart*, *flowchart* dapat membantu menjelaskan setiap urusan proses dan memudahkan penambahana proses baru. Setelah proses selesai, programmer menerjemahkan desain yang logis ke dalam program dengan berbagai macam bahasa pemrograman yang telah di rancang dan disepakati pengguna(Rosaly & Prasetyo, 2019).

2.10.2 Fungsi *Flowchart*


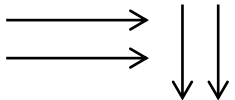

Pada dasarnya, *flowchart* dibuat dengan tujuan untuk menunjukkan setiap proses yang harus dilalui dalam sebuah proses sistem. Namun juga terdapat beberapa fungsi yang terdapat dalam *flowchart*, yaitu:



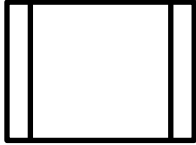
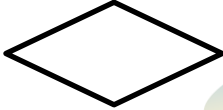
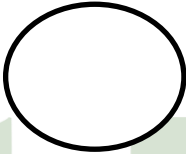



- a. Merancang proyek baru
- b. Mengelola alur kerja
- c. Memodelkan proses bisnis
- d. Mendokumentasikan hasil setiap proses
- e. Mempresentasikan algoritma
- f. Mengaudit proses

2.10.3 Simbol-Simbol *Flowchart*

Berikut ini adalah simbol-simbol yang dapat dipergunakan dalam sebuah flowchart:

Tabel 2.2 Aliran Sistem Informasi (Rosaly & Prasetyo, 2019)

No.	Simbol	Nama	Keterangan
1		terminator	Terminator untuk permulaan program dan akhir program
2		Garis alur	Garis alur untuk aliran program
3		Simbol persiapan program	Untuk proses instalasi dan pemberian nilai awal

4		Simbol proses	Proses perhitungan atau proses pengolahan data
5		<i>Input dan output</i>	Untuk proses <i>input</i> dan <i>output</i> data, parameter, dan informasi
6		Simbol sub proses	Untuk permulaan sub-program dan proses menjalankan sub-program
7		Simbol keputusan	Untuk memilih aliran syarat
8		<i>Connector</i>	Untuk menghubungkan flowchart yang berada pada satu halaman yang sama
9		Simbol penghubung	Untuk menghubungkan flowchart yang berbeda
10		<i>Document</i>	format <i>Input/Output</i> yang dapat di cetak
11		<i>Disk Storage</i>	<i>Input/Output</i> berfungsi untuk menyimpan akses <i>flowchart</i>

2.11 Penelitian Terdahulu

Penelitian terdahulu menjadi salah satu acuan dasar dalam melakukan penelitian, sehingga dapat memperkaya teori-teori yang digunakan dalam mengkaji penelitian ini. Penelitian sebelumnya yang telah dilakukan oleh peneliti terdahulu dapat ditunjukkan dengan tabel sebagai berikut:

Tabel 2.3 Penelitian Terdahulu

No	Penelitian	Asal & Tahun	Judul	Kesimpulan
1.	Ni Nyoman Emang Murti, Aggry Saputra, Zulfachmi, Ricak Agus Setiawan	STMIK Bandung Bali, 2019	Aplikasi Penilaian Otomatis Jawaban Soal Essay dengan Menggunakan Metode <i>K-Nearst Neighbors</i>	Hasil dari pembobotan dokumen TF-IDF tergantung dari hasil proses <i>text</i> mining dan kata kunci jawaban soal yang digunakan dalam dokumen tersebut. Bilamana sistem tidak menggunakan kata kunci jawaban maka akurasi penilaian turun hingga 2,5%.
2.	Ade Bastian, Harun Sujadi, Pendi Angga Sukmana	Universitas Majalengka, 2018	Rancang Bangun Aplikasi Penilaian Ujian Essay Dengan Menggunakan Algoritma Nazief & Andriani dan Metode <i>Cossine Similarity</i>	Dalam penelitian hasil dari ujian essay mahasiswa waktu yang digunakan seorang dosen menjadi lebih efisien. Dan tersedianya panduan bagi dosen yang digunakan dalam penelitian terhadap hasil dari jawaban mahasiswanya menjadi lebih konsisten.

3.	Herriyance, Handrizal, Siti Dara Fadilla	Universitas Sumatera Utara, AMIK Tunas Bangsa Pematang Siantar, 2017	Analisis Algoritma Rabin- Karp pada Kamus Umum Berbasis Android	Untuk membuat kamus berbasis android dapat menggunakan algoritma <i>string matching</i> , salah satu algoritma <i>string matching</i> adalah algoritma Rabin-Karp, algoritma Rabin-Karp melakukan pencocokan <i>string</i> berdasarkan nilai <i>hash</i> pada teks dan nilai <i>hash</i> pada <i>pattern</i> . Penelitian ini menghasilkan sebuah aplikasi kamus berbasis android yang mana bilangan basis yang digunakan untuk membangkitkan nilai <i>hash</i> sangat mempengaruhi kecepatan pencarian kata. Rata-rata <i>running time</i> untuk 10 kali percobaan pencarian kata adalah 14,9 ms.
4.	Saeful Bahri, Rusda Wajhillah	STMIK Nusa Mandiri, 2020	Optimalisasi Algoritma Rabin Karp Menggunakan TF-IDF dalam Pencocokan <i>Text</i> pada Penilaian	Pada penelitian ini diterapkan sebuah algoritma rabin karp, yang bekerja secara langsung dalam mencocokkan teks berdasarkan <i>Hash</i> yang ditambahkan TF-IDF yang

			Ujian Essay Otomatis	berguna untuk melakukan pengindekan dan menghitung frekuensi kemunculan teks pada sebuah dokumen, kedua metode ini terbukti mampu meningkatkan hasil pencocokan sebesar 11,81%.
5.	Joko Priambodo	Universitas Pamulang, 2018	Pendeteksian Plagiarisme Menggunakan Algoritma Rabin-Karp dengan Metode <i>Rolling Hash</i>	Pendeteksian plagiarisme menggunakan algoritma Rabin-Karp dengan metode <i>Rolling Hash</i> dari hasil pengujian 30 dokumen teks yang sudah dijelaskan pada bab sebelumnya menghasilkan tingkat akurasi yang terbesar yaitu 47.58%. hasil persentase tersebut termasuk dalam kategori tingkat plagiat 15-50% berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang. Sedangkan tingkat akurasi yang terkecil yaitu 19,28%, berarti menandakan dokumen tersebut termasuk plagiat tingkat sedang.

6.	Ruslan, Gunawan, Suhatati Tjandra	Sekolah Tinggu Teknik Surabaya, 2018	Sistem Penilaian Otomatis Jawaban Essay Menggunakan GLSA(<i>Generalize d Latent Semantic Analysis</i>)	Berdasarkan hasil pengujian terhadap tingkat akurasi sistem dapat disimpulkan dari dua parameter pengujian yang dilakukan bahwa pengujian dengan menggunakan <i>training data</i> sebanyak 30% dari jumlah dataset akan menghasilkan tingkat akurasi yang lebih baik dibandingkan tidak menggunakan <i>data training</i> . Dari penilaian akurasi terhadap 3 mata pelajaran dengan jumlah dataset total mencapai 864 data set jawaban, GLSA dengan menggunakan <i>training data</i> mengalami peningkatan akurasi sebesar 1,18%.
7.	Febrilia Endah Kurniawati, Windha Mega Pradnya	Universitas Amikom Yogyakarta, 2020	Implementasi Algoritma <i>Winnwing</i> Pada Sistem Penilaian Otomatis Jawaban Esai Pada Ujian Online Berbasis <i>Web</i>	Algoritma <i>Winowwing</i> sangat berguna dalam mengoreksi soal-soal essay yang terdiri dari isian singkat dengan jumlah <i>hash</i> yang tidak lebih besar dari nilai awal (<i>value</i>). Hasil <i>similarity</i>

				<p>bernilai 0 jika jumlah <i>hash</i> lebih kecil dari nilai awal (<i>value</i>). Nilai K-Gram sangat mempengaruhi hasil dari nilai <i>similarity</i> dengan hasil semakin kecil nilai K-Gram maka semakin banyak nilai yang diterima siswa dan semakin besar nilai <i>similarity</i>-nya.</p>
--	--	--	--	--



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN