

BAB II

LANDASAN TEORI

2.1. Tinjauan Terhadap Penelitian Terkait

Berikut beberapa penelitian yang terkait dan menjadi referensi pada penelitian ini:

Shativa Sonrisa, Adam Mukharil Bachtiar, melakukan penelitian dengan judul “IMPLEMENTASI DATA MINING TERHADAP PENENTUAN PAKET HEMAT SEMBAKO DAN KEBUTUHAN HARIAN MENGGUNAKAN ATURAN ASSOCIATION RULE DI PRIMER KOPERASI KARTIKA BAJA CILEGON DENGAN ALGORITMA FP-GROWTH”. Primer Koperasi Kartika Baja adalah sebuah usaha perdagangan barang dan jasa di Kodim 0623 Cilegon. Dalam sistem pelayanannya, Hal ini disebabkan karena saat Primer Koperasi Kartika Baja mengeluarkan paket, paket tersebut tidak terlalu diminati oleh pembeli karena paket yang disediakan tidak terlalu menarik minat pembeli. Oleh karena itu, untuk dapat mengatasi permasalahan tersebut, perlu dilakukannya implementasi data mining terhadap penentuan paket menggunakan aturan association rule dengan algoritma Fp-Growth. Algoritma Fp-Growth digunakan untuk menggali informasi yang ada guna memprediksi kombinasi barang yang dibeli secara bersamaan. Berdasarkan hasil penelitian, sistem yang dibangun dapat membantu koperasi dengan memberikan informasi barang-barang yang dapat ditentukan sebagai bentuk paket.

Selanjutnya M.Kom Taryana Suryana melakukan penelitian dengan judul “Penerapan data mining dengan metode Association rule terhadap penyusunan layout makanan dan penentuan paket makan hemat dirumah makan roso echo menggunakna algoritma fp-growth” . Rumah makan Roso Echo merupakan sebuah rumah makan khas jawa yang menerapkan konsep prasmanan dalam pelayanannya. Namun dalam proses penyusunan layout makanannya, rumah makan ini masih memiliki kesulitan dalam penyusunan layout makanan, Algoritma FP-Growth termasuk jenis aturan asosiasi pada data mining. Algoritma FP-Growth merupakan salah satu alternatif algoritma yang cukup efektif untuk mencari himpunan data yang paling sering muncul (Frequent Itemset) dalam sebuah kumpulan data yang besar. Algoritma ini dikenal juga dengan sebutan algoritma FP-Tree, dengan menggunakan struktur fp-tree algoritma fp-growth dapat langsung mengekstrak frequent itemset dari susunan fp-tree yang telah terbentuk. Untuk itulah perlu dilakukan suatu analisis menggunakan data mining guna mendapatkan informasi mengenai susunan layout makanan yang baru serta paket makanan hemat yang dapat ditawarkan. Hasil dari penelitian ini berupa suatu aplikasi yang mampu memberikan informasi kepada pihak pengelola rumah makan Roso Echo mengenai susunan layout makanan dan paket makanan hemat yang diterapkan.

Selanjutnya STMIK AMIKOM Yogyakarta, 4 Februari 2017 melakukan penelitian dengan judul “IMPLEMENTASI ALGORITMA HASH BASED TERHADAP ATURAN ASOSIASI UNTUK MENENTUKAN FREQUENT ITEMSET STUDY KASUS RUMAH MAKAN SEAFOOD “KITA” “.Dengan memanfaatkan data – data transaksi pembelian yang kemudian diolah untuk menggali informasi berharga dari kumpulan data transaksi tersebut. Dari informasi yang tergali tersebut kemudian dapat dijadikan menjadi suatu aturan untuk membuat kebijakan untuk meningkatkan penjualan. Salah satu algoritma yang sering digunakan untuk asosiasi adalah algoritma apriori.

Namun algoritma tersebut memiliki kelemahan dalam hal performa. Karena pada setiap menentukan frequent k-itemset harus melakukan scan pada database. Dan hal tersebut akan menjadi masalah serius apabila kandidat k-itemset memiliki jumlah sangat besar. Melakukan scan pada database yang besar akan memakan waktu yang banyak. Algoritma hash based bisa menjadi solusi untuk mengatasi masalah dalam menentukan frequent kitemset dari kandidat k-itemset dengan jumlah yang besar. Dengan menggunakan teknik hashing, kandidat k–itemset yang telah discan akan dimasukkan kedalam bucket pada tabel hash. Dari bucket tersebut akan digunakan untuk mencari frequent (k+1)-itemset sehingga scan database hanya dilakukan 1 kali pada iterasi pertama. Pada penilitan ini akan dijelaskan bagaimana algoritma hash bashed bekerja.

1. Usaha Sembako

A. Pengertian Sembako

Sembako adalah singkatan dari Sembilan Bahan Pokok. Istilah sembako sangat akrab di telinga masyarakat Indonesia. Tentu saja karna hal tersebut sangat dekat dengan kebutuhan masyarakat. Secara luas sembako adalah hal yang dibutuhkan manusia dalam kegiatan pemenuhan kebutuhannya atas pangan. Secara khusus istilah sembako ini memang hanya akrab di lingkungan masyarakat Indonesia. Karena istilah ini memang diciptakan oleh Pemerintah Indonesia melalui Kementerian Industri dan perdagangan pada tahun 1998 lalu.

Kesembilan bahan pangan yang termasuk dalam kategori pokok tersebut sudah dipatenkan oleh pemerintah dalam sebuah keputusan. Keputusan tersebut dikeluarkan oleh Kementerian Industri dan Perdagangan dengan Nomor Peraturan 15/MP P/Kep/2/1998 pada 27 Februari 1998.

B. Urgensi Sembako Komoditas Lainnya

Sembako yang terdiri atas berbagai bahan-bahan makanan dan minuman yang secara umum sangat dibutuhkan masyarakat Indonesia. Di bawah ini adalah daftar bahan-bahan pokok sembako sesuai dengan keputusan Menteri Industri dan Perdagangan No. 15/MPP/Kep/2/1998 tanggal 27 Februari 1988 yaitu sebagai berikut:

a. Beras dan Sagu

b. Jagung

- c. Sayur-sayuran dan buah-buahan
- d. Daging
- e. Susu
- f. Gula pas
- g. Garam yang mengandung Yodium
- h. Minyak goreng dan margarin
- i. Minyak tanah atau gas elpiji.

Dari sisi ekonomi permintaan barang-barang sembako bersifat inelastis yaitu perubahan harga sembako tidak akan banyak mempengaruhi tingkat permintaan produk oleh konsumen selama tidak terlalu signifikan.

2.2. Datamining

Data Mining Data mining sering juga disebut knowledge discovery in database (KDD), adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar. Keluaran dari data mining ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan. Sehingga istilah pattern recognition sekarang jarang digunakan karena ia termasuk bagian dari data mining. Proses KDD secara garis besar dapat dijelaskan sebagai berikut :

Tahapan proses KDD ada 5 (Irwan Budiman, 2012), yaitu:

1. Data Selection

Pemilihan data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai.

2. Preprocessing

Sebelum proses data mining dapat dilaksanakan, perlu dilakukan proses cleaning dengan tujuan untuk membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data, seperti kesalahan cetak. Juga dilakukan proses enrichment, yaitu proses “memperkaya” data yang sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk KDD, seperti data atau informasi eksternal.

3. Transformation

Yaitu proses coding pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses data mining. Proses coding dalam KDD merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam database.

4. Data Mining

Proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu.

5. Interpretation / Evaluation

Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses KDD yang disebut dengan interpretation. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya atau tidak

Menurut Abdul Kadir (2007:30), Data mining adalah kombinasi secara logis antara pengetahuan data, dan analisa statistik yang

dikembangkan dalam pengetahuan bisnis atau suatu proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, tiruan dan *machine-learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat bagi pengetahuan yang terkait dari berbagai database besar.

2.2.1. Algoritma Fp-Growth

Algoritma FP-Growth merupakan pengembangan dari algoritma Apriori. Sehingga kekurangan dari algoritma Apriori diperbaiki oleh algoritma FPGrowth. Frequent Pattern Growth (FP- Growth) adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (frequent itemset) dalam sebuah kumpulan data. Pada algoritma Apriori diperlukan generate candidate untuk mendapatkan frequent itemsets. Akan tetapi, di algoritma FP-Growth generate candidate tidak dilakukan karena FPGrowth menggunakan konsep pembangunan tree dalam pencarian frequent itemsets

Untuk menentukan frequent itemset pada data transaksi tersebut, dapat dilakukan langkah-langkah berikut ini: 1. Menentukan Minimum Support. 2. Menentukan Header Frequent Itemset 3. Membuat FP-Tree 4. Membuat Conditional Pattern berdasarkan FP-Tree 5. Menentukan Frequent Item-set Penelitian ini melakukan tahapan dari analisa mulai dari data secara keseluruhan serta proses manual pembahasan dari pengolahan data yang akan dilakukan asosiasi menggunakan pemodelan FP-growth berdasarkan kerangka kerja penelitian yang terdapat pada bab III Metodologi Penelitian.

Untuk menemukan Frequent itemset dari tabel 1, maka perlu di tentukan terlebih dahulu lintasan yang berakhir dengan support count terkecil, yaitu e yang di ikuti dengan d, c, b, dan di akhiri a. Karakteristik algoritma FP-Growth adalah struktur data yang digunakan adalah tree yang disebut dengan FP-Tree. Dengan menggunakan FP-Tree, algoritma FP-growth dapat langsung mengekstrak frequent Itemset dari FP- Tree.

Metode FP-Growth dapat dibagi menjadi 3 tahapan utama yaitu sebagai berikut : (Erwin, 2009)

1. Tahapan Pembangkitan *Conditional Pattern Base*

Conditional Pattern Base merupakan subdatabase yang berisi prefix path (lintasan prefix) dan suffix pattern (pola akhiran). Pembangkitan *conditional pattern base* didapatkan melalui FP-Tree yang telah dibangun sebelumnya.

2. Tahap Pembangkitan *Conditional FP-Tree*

Pada tahap ini, *support count* dari setiap item pada setiap conditional pattern base dijumlahkan, lalu setiap item yang memiliki jumlah support count lebih besar sama dengan minimum *support count* akan dibangkitkan dengan *Conditional FP-Tree*.

3. Tahap Pencarian Frequent Itemset

Apabila *Conditional FP-Tree* merupakan lintasan tunggal (single path), maka didapatkan frequent itemset dengan melakukan kombinasi item untuk setiap *Conditional FP-Tree*.

2.2.2. Algoritma Hash-Based

Algoritma Hash Based menggunakan teknik hashing untuk menyaring keluar itemset yang tidak penting untuk pembangkitan itemset selanjutnya. Ketika support count untuk kandidat k -itemset dihitung dengan menelusuri basis data, algoritma hash based mengumpulkan informasi mengenai $(k+1)$ -itemset dengan cara seluruh kemungkinan $(k+1)$ -itemset dihash ke dalam hash table dengan menggunakan fungsi hash (menggunakan sebuah bilangan prima untuk operasi modulo) [5]. Algoritma Hash Based terbagi menjadi tiga bagian utama yang masing-masing bagian melakukan proses yang berbeda. Bagian pertama akan menghasilkan kandidat 1-itemset yang disebut C_1 dan large 1-itemset yang disebut L_1 dari basis data. Untuk kandidat 1-itemset, seluruh transaksi ditelusuri untuk menghitung support count dari itemset ini. Pada tahap ini hash tree untuk C dibangun dengan tujuan mengefisienkan penghitungan support count.

Pada bagian ini juga algoritma akan membangun hash table (dengan fungsi hash) untuk 2 itemset yang akan berguna mengurangi banyaknya kandidat 2-itemset C_2 . Pada bagian kedua, kumpulan kandidat itemset C_k dibangkitkan berdasarkan hash table yang telah dibuat pada iterasi sebelumnya. Lalu ditentukan large itemset L_k dan mengurangi ukuran basis data untuk pembangkitan itemset selanjutnya. Bagian algoritma ini terbagi menjadi dua fase. Fase

pertama untuk membangkitkan kandidat k-itemset berdasarkan hash table.

Fase kedua akan menghitung support pada kandidat itemset dan mengurangi ukuran dari setiap transaksi. Bagian ketiga sama seperti bagian kedua tetapi tidak menggunakan hash table sehingga mirip dengan algoritma apriori. Bagian kedua dilakukan selama nilai hash bucket lebih besar dari minimum support. Setelah batasan ini terlewati, algoritma hash based diganti dengan algoritma apriori.


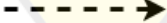

Berikut penjelasan dari rumus hash table (hash bucket) .
 $H(X,Y) = [(order\ of\ X) * penambahan\ ctr\ hash\ table + (order\ of\ y)] \mod\ prima$
Keterangan : – Order of X adalah perwakilan nilai X – Penambahan ctr hash table adalah nilai modulus bilangan prima, apabila terjadi collision nilai tersebut ditambah 1 (+ 1) sampai tidak terjadi collision. – Order of Y adalah perwakilan nilai Y – Prima adalah bilangan prima yang terdekat dan yang lebih besar dari jumlah kombinasi 2-itemet (C2).

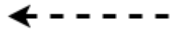





2.3. Unified Modeling Language

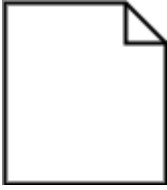
UML (Unified Modeling Language) merupakan metodologi kolaborasi antara metoda – metoda *booch*, *OMT (Object Modeling Technique)* serta *OOSE (Object Oriented Engineering)* dan beberapa metoda lainnya merupakan metodologi yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa “pemograman berorientasi objek” (OOP). (Adi Nograho; 2015 :

1. Usecase Diagram

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. Adapun notasi *use case diagram* dapat dilihat pada Gambar 2.1.

	GAMBAR	NAMA	KETERANGAN
		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>usecase</i> .
		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri(<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan



			struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
		<i>Include</i>	Menspesifikasikan bahwa <i>usecase</i> sumber secara <i>eksplisit</i> .
		<i>Extend</i>	Menspesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>usecase</i> sumber pada suatu titik yang diberikan.
		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
		<i>Usecase</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).


		<p style="text-align: center;"><i>Note</i></p>	<p style="text-align: center;">Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi</p>
--	---	--	--

Gambar 2. 1 Tipe Relasi pada Usecase Diagram (Nugroho, 2015)

b. Activity Diagram

Activity diagram atau Diagram Aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram aktivitas mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah diagram aktivitas bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Sebuah *activity diagram* memiliki :

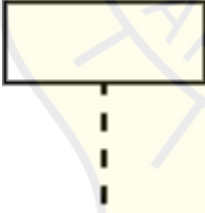


Simbol	Penjelasan
	<p style="text-align: center;"><i>Initial State</i></p> <p>Mempresentasikan dimulainya alur kerja suatu sistem dalam <i>activity diagram</i>.</p>
	<p style="text-align: center;"><i>Fork</i></p> <p>Adanya percabangan paralel dari aktivitas</p>


	<p><i>Final State</i></p> <p>Mempresentasikan bahwa telah diakhirinya alur suatu sistem dalam <i>activity diagram</i>.</p>
---	--

Gambar 2. 2 Tipe Relasi pada Activity Diagram (Nugroho, 2015)

c. *Sequence Diagram*

Merupakan diagram yang menunjukkan aliran fungsionalitas dalam *usecase*. *Sequence* diagram adalah satu dari dua interaksi diagram yang mengilustrasikan objek-objek yang berhubungan dengan *usecase* dan *message* atau pesan-pesannya. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

Simbol	Nama Elemen	Keterangan
	Objek	Komponen yang menjadikan sebuah objek dalam membuat diagram
	<i>Stimulus</i>	Untuk menandakan hubungan komunikasi antarobjek
	<i>Self Stimulus</i>	Fungsi sama dengan <i>Stimulus</i> , tetapi pesan yang

		disampaikan dikirirkan untuk objek itu sendiri
	<i>Focus Control</i>	Sebagai tempat untuk hasil input atau output dari sebuah proses yang dilakukan oleh objek ataupun aktor yang ada dalam sistem

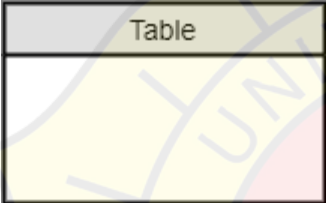


Gambar 2. 3 Komponen Sequence Diagram (Nugroho, 2015)

d. *Class Diagram*

Diagram kelas atau *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan *property*, operasi dan relasi yang sama. Disamping itu diagram kelas bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan lainnya. Itulah sebabnya diagram kelas menjadi diagram yang paling populer di *UML*.
Komponen diagram kelas adalah :

1. Asosiasi adalah *class-class* yang berhubungan satu sama lain secara konseptual. yaitu menghubungkan dua kelas menjadi satu asosiasi.
2. Atribut adalah properti dari sebuah kelas. Atribut ini menjelaskan batas nilai yang mungkin ada pada obyek dari kelas. Sebuah kelas mungkin mempunyai nol atau lebih atribut. Contoh dari atribut :
3. Operasi adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau yang anda (atau *class* yang lain) dapat lakukan untuk sebuah *class*. Sama

halnya dengan atribut, kita bisa juga memberikan tambahan informasi untuk operasi dengan menambahkan parameter yang akan dilakukan oleh operasi dengan tanda kurung.

Simbol	Nama Elemen	Keterangan
	<p>Kelas</p>	<p>Simbol untuk membangun sebuah pemrograman dengan objek</p> <p>Terdiri 3 bagian, bagian atas adalah nama kelas, bagian tengah adalah atribut, dan bagian bawah adalah metode dari kelas tersebut</p>
	<p>Garis Asosiasi</p>	<p>Simbol yang menggambarkan adanya hubungan antara satu kelas dengan kelas lainnya</p>
	<p>Generalisasi</p>	<p>Simbol yang menandakan adanya generalisasi dari kelas</p>

		input untuk menghasilkan data yang dibutuhkan
--	--	---

Gambar 2. 4 Komponen Class Diagram (Nugroho, 2015)

2.4. Internet

Internet adalah jaringan global yang menghubungkan komputer-komputer diseluruh dunia. Dengan internet, sebuah komputer bisa mengakses data yang terdapat pada komputer lain di benua yang berbeda (Hidayatullah dan Khawistara, 2015).

Internet (portmanteau dari interconnected network) adalah sistem jaringan komputer yang saling terhubung secara global dengan menggunakan paket protokol internet (TCP/IP) untuk menghubungkan perangkat di seluruh dunia. Ini adalah jaringan dari jaringan yang terdiri dari jaringan privat, publik, akademik, bisnis, dan pemerintah lokal ke lingkup global, dihubungkan oleh beragam teknologi elektronik, nirkabel, dan jaringan optik. Internet membawa beragam sumber daya dan layanan informasi, seperti dokumen hiperteks yang saling terkait dan aplikasi World Wide Web (WWW), surat elektronik, telepon, dan berbagi berkas (Wikipedia).

2.5. Web

Web merupakan suatu ruang informasi di mana sumber-sumber daya yang berguna diidentifikasi oleh pengenal global yang disebut *Uniform User Identifier* (URL). URL dapat diibaratkan suatu alamat, dimana alamat tersebut terdiri atas:

1. Protokol yang digunakan oleh suatu *browser* untuk mengambil informasi.
2. Nama computer (*server*) dimana informasi tersebut berada.
3. Jalur atau *path* serta nama *file* dari suatu informasi.

Format umum dari URL adalah sebagai berikut :

Protokol_transfer://nama_host/path/nama_file.

Contoh: <http://www.mine.com/e-journal/index.html>

Dari contoh tersebut dapat disimpulkan bahwa:

4. http adalah protocol yang digunakan.
5. www.mine.com adalah nama *host* atau *server* komputer dimana informasi yang dicari berada.
6. *e-journal* adalah jalur atau *path* dari informasi yang dicari.
7. Index.html adalah nama *file* dimana informasi tersebut berada.

Sebuah halaman *web* diakses dengan menggunakan *web browser* dengan menuliskan URL nya atau mengikuti *link* yang menuju kepadanya. *Uniform Resource Locator* (URL) akan menunjukkan lokasi dokumen yang dikelola oleh sebuah *web server*. URL diubah menjadi alamat IP *server web* yang bersangkutan. *Browser* kemudian mengirimkan request *Hypertext Transfer Protocol* (HTTP) ke *web server* dan *web server* akan menjawab dokumen yang diminta dalam format *Hypertext Markup Language* (HTML). HTTP adalah suatu protocol yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan oleh *web server* dalam menyediakan dokumen yang diminta *web*

browser. Protokol ini merupakan protocol standar yang digunakan untuk mengakses dokumen HTML. Dalam komunikasi jaringan computer diatur dengan protocol yang memungkinkan beragam jaringan computer untuk berkomunikasi. Protokol ini secara resmi dikenal dengan *Transmission Control Protocol* (TCP/IP) yang merupakan cara untuk mempacketkan sinyal elektronik sehingga data tersebut dapat dikirim ke computer lain. Gambar di bawah adalah menggambarkan cara kerja *web* yang diakses.

8. *Server-side Programming*

Perintah-perintah program atau *script* dijalankan di *web server*, kemudian hasilnya dikirimkan ke *browser* dalam bentuk HTML.

9. *Client-side Programming*

Perintah program dijalankan di *web browser* sehingga ketika *client* meminta dokumen yang mengandung *script*, maka *script* tersebut akan *download* dari *server*nya kemudian dijalankan di *browser* yang bersangkutan.

2.6. HTML

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*. HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language* (SGML) (Kadir, 2015). HTML sebenarnya adalah dokumen *ASCII* atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

Mendesain HTML berarti melakukan suatu tindakan pemrograman.

Namun *HTML* bukanlah sebuah bahasa pemrograman. Namun *HTML* hanyalah berisi perintah-perintah yang telah terstruktur berupa tag-tag penyusun. Menuliskan tag-tag *HTML* tidaklah sebatas hanya memasukkan perintah-perintah tertentu agar *HTML* kita dapat di akses oleh browser. Mendesain *HTML* adalah sebuah seni tersendiri. *Homepage* yang merupakan implementasi dari *HTML* adalah refleksi dari orang yang membuatnya. Untuk itu kita perlu mendesainnya dengan baik agar para pengunjung homepage yang kita buat merasa senang dan bermanfaat. Mendesain *HTML* dapat dilakukan dengan cara menggunakan *HTML Editor*, seperti notepad++, adobe dreamweaver dan lain-lain.

2.7. PHP

Menurut Kadir (2015) “*PHP* adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”. *PHP* disebut juga pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. *PHP* adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan *open source* yaitu pengguna data mengembangkan kode-kode fungsi sesuai kebutuhannya.

Menurut Shalahuddin (2015:22) “*PHP (Perl Hypertext Preprocessor)* adalah bahasa *server-side-scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis”. Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif dan dinamis. Kelebihan-kelebihan dari *PHP* yaitu:

1. *PHP* merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. *PHP* dapat berjalan pada *web server* yang dirilis oleh Microsoft, seperti *IIS* atau *PWS* juga pada *apache* yang bersifat *open source*.
3. 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.
4. Jika dilihat dari segi pemahaman, *PHP* memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.

PHP dapat berjalan pada 3 operating sistem, yaitu: *Linux*, *unix*, dan *windows*, dan juga dapat dijalankan secara *runtime* pada suatu *console*.

2.8. Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup (Kadir, 2015). Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan *HTML* dan *XHTML*. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen *XML* termasuk *SVG* dan *XUL*. *Spesifikasi CSS* diatur oleh *World Wide Web Consortium (W3C)*. *CSS* digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. *CSS* digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan *HTML* atau *bahasa markup* lainnya) dengan presentasi dokumen (yang ditulis dengan *CSS*). Pemisahan ini dapat

meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi.

2.9. Teks Processing

Merupakan tahapan pemrosesan data agar menjadi data yang siap untuk dianalisis. Setelah data terstruktur maka dapat diolah lebih lanjut. Beberapa proses yang dilakukan pada tahap ini adalah sebagai berikut

1. Case folding

Merupakan proses perubahan semua huruf pada dokumen tweet menjadi huruf kecil. Hanya huruf a sampai z yang diproses. Karakter selain huruf akan dibiarkan. Contoh : Sb menambah wangi segar !! Menjadi : sb menambah wangi segar.

2. Tokenizing

Pada tahap ini, kalimat dipotong atau dipecah berdasarkan tiap kata yang menyusunnya. bekerja untuk mengidentifikasi kata – kata dalam teks menjadi beberapa urutan yang terpotong oleh spasi atau karakter spesial. Berikut contoh penerapan dari tokenizing.

Data Latih	
Input	Output
Taman hutan raya djuanda taman hutan raya djuanda masih terjaga bersih dan sejuk.	[taman, hutan, raya, djuanda, taman, hutan, raya, djuanda, masih, terjaga, bersih dan sejuk]

Gambar 2. 5 Penerapan Tokenizing (Jurnal, Wilianto, Fajri rahmat umbara,2017)

3. Stopword Removal

Merupakan tahap pembuangan kata-kata yang dianggap tidak penting. Langkah ini dilakukan supaya perhitungan lebih berfokus pada kata-kata yang jauh lebih penting. berguna untuk membuang kata – kata yang sering muncul dan bersifat umum, Kurang menunjukan relevansinya dengan teks. Kata – kata yang akan dibuang tersebut Didefinisikan dalam stopwords list. Contoh beberapa kata yang sering masuk ke dalam Stopword list adalah “sebuah”, “yang”, “itu”. Berikut adalah beberapa daftar Stoplist Yang disimpan dalam database.

4. Stemming

Tahap stemming adalah tahap mencari root (bentuk dasar) dari tiap kata. Pada tahap ini, dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama.

Data Latih	
Input	Output
Menginspirasi	Inspirasi

Gambar 2. 6 Penerapan Stemming (Jurnal, Wilianto, Fajri rahmat umbara,2017)