BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Pengembangan program *chatbot* multibahasa memerlukan faktor-faktor pendukung yang merupakan landasan teori yang akan digunakan dalam proses pengerjaan.

2.1.1 Sistem

2.1.1.1 Konsep Dasar Sistem

Sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapi suatu tujuan tertentu. Kedua kelompok ini benar dan tidak bertentangan. Yang membedakan adalah cara pendekatannya.

2.1.1.2 Pengertian Sistem

Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, mater atau energi. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang saling berinteraksi, di mana suatu model matematika seringkali dibuat.

Sistem juga merupakan suatu kesatuan prosedur atau komponen yang saling berkaitan satu dengan yang lainnya bersama-sama sesuai dengan aturan yang diterapkan sehingga membentuk suatu tujuan yang sama. Dimana dalam sebuah sistem bila terjadi satu bagian yang tidak bekerja atau rusak maka suatu tujuan bisa terjadi kesalahan hasilnya atau tidak sesuai dengan tujuan dari sistem tersebut.

2.1.1.3 Klasifikasi Sistem

Dari berbagai sudut pandang, sistem dapat di klasifikasikan menjadi beberapa bagian yaitu :

1. Sistem abstrak dan sistem fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara nyata. Sedangkan sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem alamiah dan sistem buatan

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin.

3. Sistem deterministic dan sistem probabilistik

   Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

4. Sistem tertutup dan sistem terbuka

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk sub sistem lainnya.

2.1.1.4 Karakteristik Sistem

Model umum sebuah sistem terdiri dari input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana, mengingat sebuah sistem dapat mempunyai beberapa masukkan dan keluaran sekaligus. Adapun karakteristik yang dimaksudkan adalah sebagai berikut :

1. Komponen sistem (*components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi.

2. Batasan sistem (*boundary*)

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya.

3. Lingkungan luar sistem (*environment*)

   Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

4. Penghubung sistem (*interface*)

   Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan sistem (*input*)

   Energi yang dimasukan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran sistem (*output*)

   Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

7. Pengolahan sistem (proses)

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran sistem (*objective*)

   Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.1.1.5 Komponen Sistem

Pemahaman sistem dengan pendekatan komponen/ elemen yaitu kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan tertentu. Suatu sistem dapat terdiri dari beberapa subsistem. Subsistem-subsistem tersebut dapat pula terdiri dari beberapa subsistem yang lebih kecil.

2.1.2 *Chatbot*

Chatbot (atau *chatterbot* atau *bots*) adalah sebuah program komputer yang dirancang untuk menstimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara audio maupun teks.

Pada mulanya, program komputer *bots* ini diuji melalui tes *Turing Test*, yaitu dengan merahasiakan identitasnya sebagai mesia sehingga dapat mengelabui orang yang bercakap-cakap dengannya. Jika pengguna tidak dapat mengidentifikasi *bots* sebagai suatu program komputer, maka *chatterbot* tersebut dikategorikan sebagai kecerdasan buatan atau *artificial intelligence*.

2.1.3 *Internet*

Internet berasal dari kata *interconnection-networking* yang secara harfiah dapat diartikan sebagai berikut:

1. Rangkaian komputer yang terhubung di dalam beberapa rangkaian.
2. Sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia.

Internet merupakan sistem global dari seluruh jaringan komputer yang saling terhubung menggunakan standar *Internet Protocol Suite* (TCP/IP) untuk melayani pengguna *internet* di seluruh dunia. *Internet* menghubungkan pengguna komputer dari satu negara ke negara lain, di mana didalamnya terdapat berbagai sumber daya informasi yang bersifat statis maupun dinamis, dan interaktif.

Layanan *internet* meliputi komunikasi langsung (email, chat), diskusi (email, milis), sumber daya yang terdistribusi (*World Wide Web*), dan lalu lintas file (Telnet, FTP), dan berbagai macam layanan lainnya.

Sebuah sistem komputer yang terhubung secara langsung ke jaringan memiliki nama domain dan alamat IP (*Internet Protocol*) dalam bentuk numerik dengan format tertentu sebagai pengenal. *Internet* juga memiliki gateway ke jaringan dan layanan yang berbasis protokol lainnya.

2.1.3.1 Sejarah *Internet*

Jaringan *internet* pertama kali dikembangkan tahun 1969 oleh Departemen Pertahanan Amerika Serikat dengan nama ARPAnet (*US Defense Advance Research Projects Agency*). ARPAnet dibangun dengan sasaran untuk membuat suatu jaringan komputer yang tersebar untuk menghindari pemusatan informasi di satu titik yang dipandang rawan untuk dihancurkan apabila terjadi peperangan. Dengan membangun ARPAnet ini diharapkan apabila satu bagian dari jaringan terputus, maka jalur yang melalui jaringan tersebut secara otomatis dipindahkan ke saluran lainnya.

Di awal 1980-an, ARPAnet terpecah menjadi dua jaringan, yaitu ARPAnet dan Milnet (sebuah jaringan militer), akan tetapi keduanya mempunyai hubungan sehingga komunikasi antarjaringan tetap dapat dilakukan. Pada mulanya jaringan interkoneksi ini disebut DARPA *Internet*, tapi lama-kelamaan disebut sebagai *internet* saja. Sesudahnya, *internet* mulai digunakan untuk kepentingan akademis dengan menghubungkan beberapa perguruan tinggi, diantaranya UCLA, University of California at St. Barbara, University of Utah, Stanford Research Institute, disusul dengan dibukanya layanan Usenet dan Bitnet yang memungkinkan *internet* diakses melalui sarana komputer pribadi (PC). Berikutnya, protokol standar TCP/IP mulai diperkenalkan pada tahun 1982, disusul dengan penggunaan sistem DNS (*Domain Name Service*) pada tahun 1984.

Tahun 1986 lahir *National Science Foundation Network* (NSFNet), yang menghubungkan para periset dengan 5 buah pusat super komputer. Jaringan ini kemudian berkembang untuk menghubungkan berbagai jaringan akademis lainnya yang terdiri atas universitas dan konsorsium-konsorsium riset. NSFNet kemudian mulai menggantikan ARPAnet sebagai jaringan riset utama di Amerika hingga pada bulan Maret 1990 ARPAnet secara resmi dibubarkan. Pada saat NSFNet dibangun, berbagai jaringan internasional didirikan dan dihubungkan ke NSFNet. Australia, negara-negara Skandinavia, Inggris, Prancis, Jerman, Kanada, dan Jepang segera bergabung kedalam jaringan NSFNet ini.

Pada awalnya, *internet* hanya menawarkan layanan berbasis teks, meliputi *remote access, email/ messaging*, maupun diskusi melalui *newsgroup*. Layanan berbasis *World Wide Web* (WWW) saat itu masih belum ada, yang ada hanya layanan yang disebut *Gopher* yang dalam beberapa hal mirip seperti web yang kita kenal saat ini, kecuali sistem kerjanya yang masih berbasis teks. Kemajuan berarti dicapai pada tahun 1990 ketika *World Wide Web* mulai dikembangkan oleh CERN (Laboratorium Fisika Partikel di Swiss) berdasarkan proposal yang dibuat oleh Tim Berners-Lee. Namun, WWW browser yang pertama baru lahir dua tahun kemudian, tepatnya pada tahun 1992 dengan nama Viola. Viola diluncurkan oleh Pei Wei dan didistribusikan bersama CERN. Tentu saja web browser yang pertama ini masih sangat sederhana, belum secanggih browser modern seperti yang ada saat ini.

Terobosan berarti lainnya terjadi pada 1993 ketika InterNIC didirikan untuk menjalankan layanan pendaftaran domain. Bersamaan dengan itu, Gedung Putih (*White House*) mulai online di *internet* dan pemerintah Amerika Serikat meloloskan *National Information Infrastructure Act*. Penggunaan *internet* secara komersial dimulai pada 1994 dipelopori oleh perusahaan Pizza Hut, dan *Internet Banking* pertama kali diaplikasikan oleh *First Virtual*. Setahun kemudian, *Compuserve, America Online*, dan *Prodigy* mulai memberikan layanan akses ke *internet* bagi masyarakat umum.

2.1.3.2 Manfaat dan Kegunaan *Internet*

Kegunaan *internet* dapat dikelompokkan menjadi beberapa kelompok, yaitu :

1. *Internet* sebagai media informasi

   Internet sebagai media informasi dan telekomunikasi yang cepat, mudah, dan murah dengan jangkauan global menyebabkan bermunculan media online. Keuntungan media online dibandingkan dengan media cetak adalah mengurangi biaya cetak dan distribusi.
2. *Internet* sebagai media promosi

Dengan mempromosikan produk di *internet* maka suatu usaha mendapat beberapa keuntungan, yaitu dapat menjangkau pelanggan di mana pun berada tanpa batasan waktu dan wilayah.

3. *Internet* sebagai alat *research and development*
4. *Internet* sebagai media perukaran data

2.1.4 *Flowmap*

2.1.4.1 Definisi *Flowmap*

Flowmap adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowmap* berguna untuk membantu analisis dan programer untuk memecahkan masalah ke dalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif pengoperasian.

2.1.4.2 Aturan Membuat *Flowmap*

Untuk membuat sebuah analisis menggunakan *flowmap*, seorang analisis dan programer membutuhkan beberapa tahapan, diantaranya :

1. *Flowmap* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefiniskan dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktivitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol flowchat yang standar.

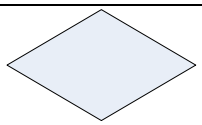
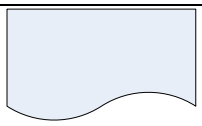
2.1.4.3 Jenis - Jenis *Flowmap*

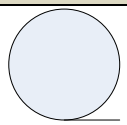
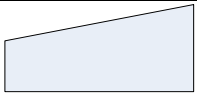
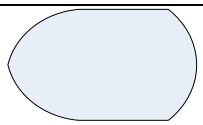
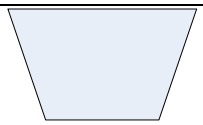
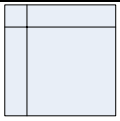
Bagan alir dokumen atau disebut juga bagan alir formulir atau *paperwork flowmap* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir dan termasuk tembusan-tembusannya. Jenis-jenis dari *flowmap* adalah sebagai berikut :

1. *Flowmap* sistem.
2. *Flowmap* paperwork atau *flowmap* dokumen.
3. *Flowmap* skematik.
4. *Flowmap* program.
5. *Flowmap* proses.

2.1.4.4 Simbol *Flowmap*

Tabel 2.1 Simbol Flowmap

Simbol	Nama
	Proses
	Keputusan
	Dokumen
	Penyimpanan data
	Data
	Proses yang belum di definisikan

Simbol	Nama
	Data sekuensial
	Penyimpanan data yang dapat di akses langsung
	Manual input
	Tampilan
	Operasi manual
	Persiapan
	Mode paralel
	Batas perulangan (awal atau akhir)
	Terminator
	Konektor halaman
	Penghubung
	Penyimpanan internal

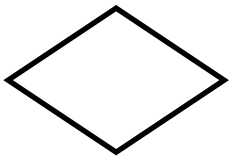
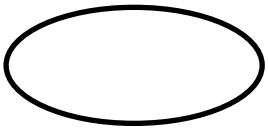
2.1.5 ERD (*Entity Relationship Diagram*)

2.1.5.1 Definisi ERD

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi (Yakub, 2008, p. 25).

2.1.5.2 Komponen - Komponen ERD

Tabel 2.2 Komponen-komponen ERD

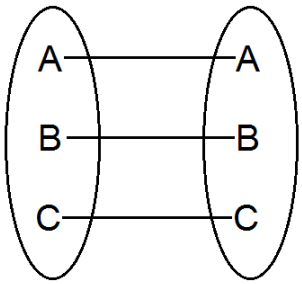
Simbol	Nama	Keterangan
	Entitas	Menggambarkan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain.
	Relasi	Menggambarkan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.
	Atribut	Menggambarkan karakteristik dari entitas yang menjelaskan secara detil tentang entitas tersebut.

2.1.5.3 Kardinalitas

Kardinalitas relasi menunjukkan jumlah maksimum tupelo yang dapat berelasi dengan entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antar entitas tersebut, kardinalitas relasi menunjukan kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Macam kardinalitas sebagai berikut :

1. Satu ke satu (*one to one relation*)

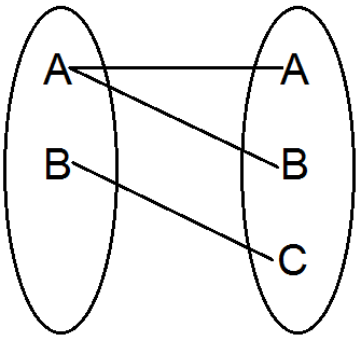
   Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B (Yakub, 2008, p. 33).



Gambar 2.1 Contoh relasi satu ke satu

2. Satu ke banyak (*one to many relation*)

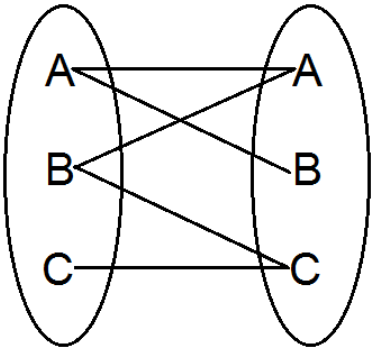
   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan paling banyak dengan satu entitas pada himpunan entitas A (Yakub, 2008, p. 34).



Gambar 2.2 Contoh relasi satu ke banyak

3. Banyak ke banyak (*many to many relation*)

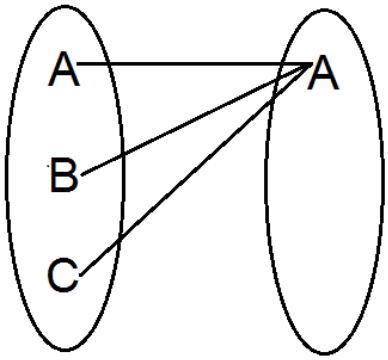
   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, begitupula sebaliknya (Yakub, 2008, p. 35).



Gambar 2.3 Contoh relasi banyak ke banyak

4. Banyak ke satu (*many to one relation*)

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B (Yakub, 2008, p. 34).



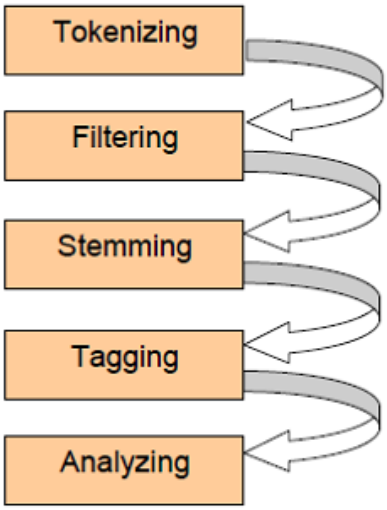
Gambar 2.4 Contoh relasi banyak ke satu

2.1.6 *Text Mining*

2.1.6.1 Teori *Text Mining*

Text Mining adalah sebuah analisa yang mengumpulkan *keywords* atau *terms* (istilah) yang sering muncul secara bersamaan dan kemudian menemukan korelasi atau hubungan asosiasi di antara *keywords* atau *terms* tersebut (Ardhani Reswari, Skripsi, 2011, p. 15). Pada sumber lain definisi *Text Mining* adalah menambang data yang berupa teks di mana sumber data biasanya didapatkan dari dokumen (bisa juga berasal dari inputan), dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar keywords yang di dapat dari dokumen tersebut.

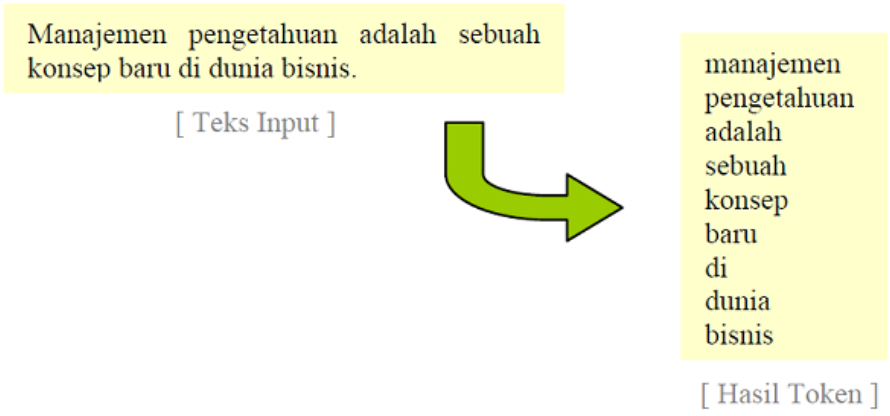
Dalam tahap *Text Mining* terdapat lima tahapan antara lain :



Gambar 2.5 Tahapan dalam Text Mining

1. Tahap *Tokenizing*

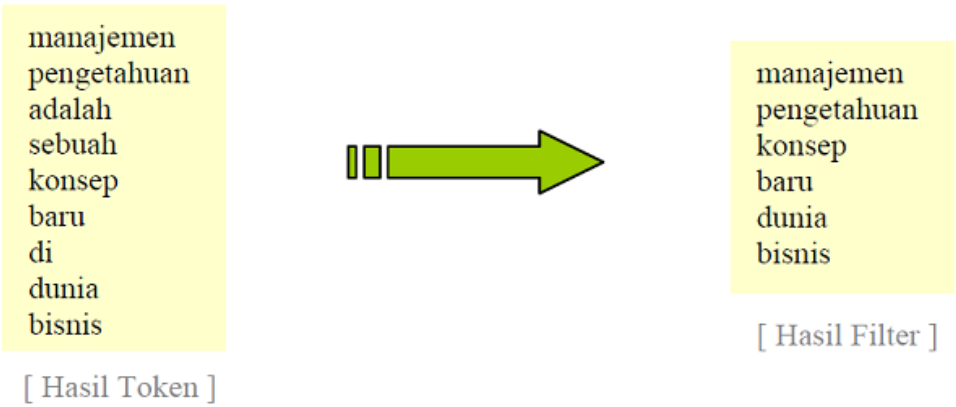
   Merupakan tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Contoh dari tahap ini adalah sebagai berikut :



Gambar 2.6 Contoh tahap Tokenizing

2. Tahap *Filtering*

   Filtering adalah tahap mengambil kata-kata penting dari hasil token. Contoh dari tahap ini adalah sebagai berikut :



Gambar 2.7 Contoh tahap Filtering

3. Tahap *Stemming*

   Stemming merupakan suatu proses yang terdapat dalam sistem IR (*Information Retrieval*) yang mentransformasi kata-kata hasil *filtering* ke kata-kata akarnya (*rood word*) dengan menggunakan aturan-aturan tertentu (Ledy Agusta, Konferensi Nasional Sistem dan Informatika 2009, KNS&I09-036, 2009, p.1).

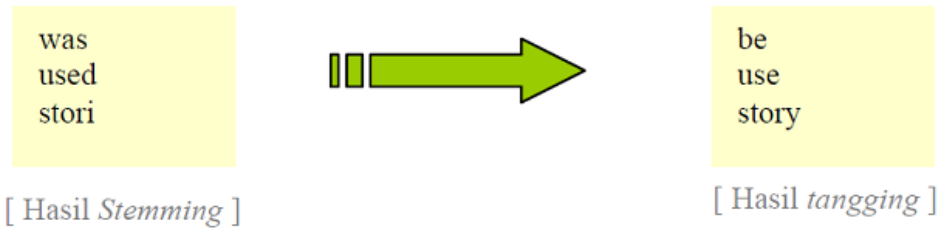
   Proses *stemming* pada teks berbahasa Indonesia berbeda dengan *stemming* pada teks berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia, selain sufiks, prefiks, dan konfiks juga dihilangkan.

   Contoh penggunaan *stemming* pada teks berbahasa Indonesia, kata bersama, kebersamaan, menyamai, jika dikenakan proses *stemming* ke bentuk kata dasarnya yaitu “sama”.

4. Tahap *Tagging*

   Tahap *tagging* adalah tahap mencari bentuk awal dari tiap kata lampau dari hasil *stemming*. Pada tahap ini dilakukan proses pengambilan berbagai bentukan kata ke dalam suatu representasi yang sama. Tahap *tagging* tidak digunakan dalam penelitian ini.

   Contoh dari tahap *tagging* :



Gambar 2.8 Contoh tahap Tagging

5. Tahap *Analyzing*

   Tahap penentuan seberapa jauh keterkaitan antar kata-kata pada dokumen/ inputan yang ada. Pada tahap *analyzing* akan digunakan rumus TF-IDF untuk mengambil sebuah informasi dari sebuah dokumen. Pada dasarnya, TF-IDF bekerja dengan menentukan frekuensi relatif dari kata-kata tertentu dalam sebuah dokumen dibandingkan dengan *inverse* dari seluruh dokumen. Kata-kata yang umum dalam sebuah dokumen cenderung memiliki nilai tinggi dalam perhitungan TF-IDF (Juan Ramos, Jurnal Rutgers University, p. 2).

Jadi, dalam penelitian ini hanya menggunakan 4 tahap, yaitu *Tokenizing*, *Filtering*, *Stemming*, dan *Analyzing*. Tahap *tagging* tidak dilakukan karena pada chatbot multitranslator ini tidak memperhatikan bentuk lampau dari suatu kata.

2.1.7 Algoritma Nazief dan Adriani

Stemming merupakan bagian yang tidak terpisahkan dalam *Information Retrieval* (IR). Tidak banyak algoritma yang dikhususkan untuk *stemming* bahasa Indonesia dengan berbagai keterbatasan didalamnya. Algoritma Porter misalnya, algoritma ini membutuhkan waktu yang relatif lebih singkat dibandingkan dengan *stemming* menggunakan algoritma Nazief dan Adriani, namun proses *stemming* menggunakan algoritma Porter memiliki persentase keakuratan lebih kecil dibandingkan dengan *stemming* menggunakan algoritma Nazief dan Adriani. Algoritma Nazief dan Adriani sebagai algoritma *stemming* untuk teks berbahasa Indonesia yang memiliki kemampuan persentase keakuratan lebih baik dari algoritma lainnya. Algoritma ini sangat dibutuhkan dan menentukan dalam proses IR dalam dokumen berbahasa Indonesia.

Stemming adalah salah satu cara yang digunakan untuk meningkatkan performa IR dengan cara mentransformasi kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya. Algoritma *stemming* untuk bahasa yang satu berbeda dengan algoritma *stemming* untuk bahasa lainnya. Sebagai contoh, bahasa inggris memiliki morfologi yang berbeda dengan bahasa Indonesia sehingga algoritma *stemming* untuk kedua kedua bahasa tersebut juga berbeda. Proses *stemming* pada teks berbahasa Indonesia lebih rumit dan kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan *root word* (kata dasar) dari sebuah kata. Pada umumnya kata dasar pada bahasa indonesia terdiri dari kombinasi.

DP + DP + DP + Kata Dasar + DS + PP + P

Keterangan :

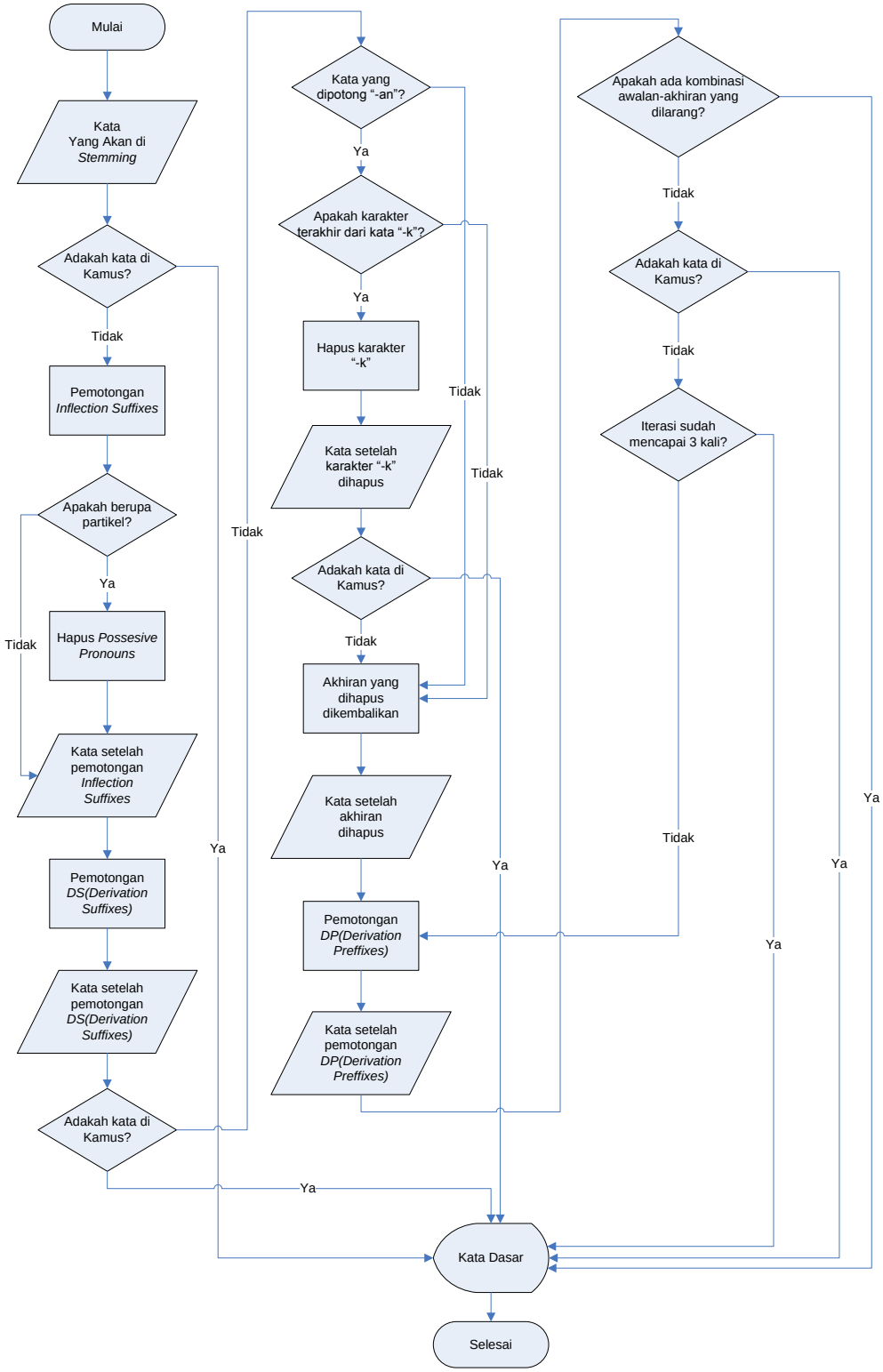
DP : *Derivation Prefix* (awalan)

DS : *Derivation Suffixes* (akhiran)

PP : *Possesive Pronouns* (kepunyaan, contoh “-ku”, “-mu”)

P : *Particels* (contoh “-lah”, “-kah”)

Sehingga dapat digambarkan menggunakan *flowchart* sebagai berikut :



Gambar 2.9 Flowchart algoritma stemming bahasa indonesia

Algoritma Nazief dan Adriani yang dibuat oleh Bobby Nazief dan Mirna Adriani ini memiliki tahap-tahap sebagai berikut :

1. Cari kata yang akan di-*stemming* dalam kamus kata dasar. Jika ditemukan maka diasumsikan kata tersebut merupakan kata dasar. Maka algoritma berhenti.
2. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa partikel (“-lah”, “-kah”, “-tah”, atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”), jika ada.
3. Hapus *Derivation Suffixes* (“-i”, “-an”, atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
    a. Jika “-an”, telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus kata dasar maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
    b. Akhiran yang dihapus (“-i”, “-an”, atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Hapus *Derivation Prefix DP* (“di-”, “ke-”, “se-”, “me-”, “be-”, “pe-”, “te-”) dengan iterasi maksimum adalah 3 kali
    a. Langkah 4 berhenti jika :
        1. Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada tabel 2.3.

Tabel 2.3 Kombinasi awalan-akhiran yang tidak diizinkan

Awalan	**Akhiran yang tidak diizinkan**
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

        2. Tiga awalan telah dihilangkan.

b. Tipe awalan ditentukan melalui langkah-langkah berikut :

1. Jika awalannya adalah : “di-”, “ke-”, atau “se-”, maka tipe awalannya secara berturut-turut adalah “di-”, “ke-”, atau “se-”.
2. Jika awalannya adalah “te-”, “me-”, “be-”, atau “pe-”, maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.

Tabel 2.4 Jenis awalan berdasarkan tipe awalan

Aturan	Format Kata	Pemenggalan
1	berV…	ber-V… | be-rV…
2	berCAP…	ber-CAP… dimana C != 'r' dan P != 'er'
3	berCAPerV…	ber-CaerV… dimana C != 'r'
4	belajar	bel-ajar
5	berC$_1$erC$_2$…	be-C$_1$erC$_2$… dimana C$_1$!= {'r'|'l'}
6	terV…	ter-V… | te-rV…
7	terCerV…	ter-CerV… dimana C != 'r'
8	terCP	ter-CP… dimana C != 'r' dan P != 'er'
9	teC$_1$erC$_2$	te-C$_1$erC$_2$… dimana C$_1$!- 'r'
10	me{l|r}w}y}V…	me-{l|r|w|y}V…
11	mem{b|f|v}…	mem-{b|f|v}…
12	mempe{r|l}…	mem-pe…
13	mem{rV|V}…	me-m{rV|V}… | me-p{rV|V}…
14	men{c|d|j|z}…	men-{c|d|j|z}…
15	menV…	me-nV… | me-tV…
16	meng{g|h|q}…	meng-{g|h|q}…
17	mengV…	meng-V… | meng-kV…
18	meny…	meny-sV…
19	mempV…	mem-pV… dimana V != 'e'
20	pe{w|y}V…	pe-{w|y}V…
21	perV…	per-V… | pe-rV…
22	perCAP	per-CAP… dimana C != 'r' dan P != 'er'

Aturan	Format Kata	Pemenggalan
23	perCAerV…	per-CAerV… dimana C != 'r'
24	pem{b|f|v}…	pem-{b|f|V}…
25	pem{rV|V}…	pe-m{rV|V}… | pe-p{rV|V}…
26	pen{c|d|j|z}…	pen-{c|d|j|z}…
27	penV…	pe-nV… | pe-tV…
28	peng{g|h|q}…	peng-{g|h|q}…
29	pengV…	peng-V… | peng-kV…
30	penyV...	peny-sV…
31	pelV…	pe-lV… kecuali "pelajar" yang menghasilkan "ajar"
32	peCerV…	per-erV… dimana C != {r|w|y|l|m|n}
33	perCP…	pe-CP… dimana C != {r|w|y|l|m|n} dan P != 'er'

3. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses berhenti.

5. Melakukan *recording* (pada chatbot multitranslator ini, proses *recording* tidak dilakukan, hal ini dilakukan untuk meminimalisir kata dasar yang tidak valid).
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai kata dasar. Proses selesai.

Pada proses *stemming* menggunakan algoritma Nazief dan Adriani, kamus yang digunakan sangat mempengaruhi hasil *stemming*. Semakin lengkap kamus yang digunakan maka semakin akurat pula hasil *stemming*.

Contoh :

Kata **mempertemukannya**

1. Cek kata "mempertemukannya" di dalam kamus kata dasar, kata "mempertemukannya" tidak ditemukan di dalam kamus kata dasar.
2. Hapus *Inflection Suffixes* ("-nya"). Sehingga kata menjadi “mempertemukan”.
3. Hapus *Derivation Suffixes* ("-kan"). Cek kata "mempertemu" di dalam kamus kata dasar, kata "mempertemu" tidak ditemukan di dalam kamus kata dasar.
4. Hapus *Derivation Preffixes* ("me-") [iterasi 1]. Periksa tabel jenis awalan pada aturan no 13, format kata ("mem-V") sehingga kata menjadi "pertemu". Cek kata "pertemu" di dalam kamus kata dasar, kata "pertemu" tidak ditemukan di dalam kamus kata dasar.
5. Hapus *Derivation Preffixes* ("pe-") [iterasi 2]. Periksa tabel jenis awalan pada aturan no 21, format kata ("per-V") sehingga kata menjadi "temu". Cek kata "temu" di dalam kamus kata dasar, kata "temu" ditemukan di dalam kamus kata dasar.
6. Proses stemming berhenti dengan menghasilkan kata dasar "temu".

Jadi kata **“mempertemukannya”** ketika menggunakan algoritma *steming* Nazief dan Adriani menjadi **“temu”**.

2.1.8 *Natural Language Processing* (NLP)

Natural Language Processing atau pemrosesan bahasa alami merupakan salah satu tujuan jangka panjang dari *Artficial Intelegence* (kecerdasan buatan) yaitu pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia. Bentuk utama representasinya adalah berupa suara/ucapan (*spoken language*), tetapi seringpula dinyatakan dalam bentuk tulisan.

Inti dari pemrosesan bahasa alami adalah penguraian kalimat atau sering disebut dengan *parser*. Parser berfungsi untuk membaca kalimat, kata demi kata dan menentukan jenis kata apa saja yang boleh mengikuti kata tersebut. Dalam

pemahaman suatu bahasa ada beberapa bidang yang harus disertakan yaitu *morfologi*, *sintaksis*, *semantik*, *pragmatik*, *fonologi*, dan pengetahuan tentang dunia sekitar.

2.1.8.1 Komponen Utama Bahasa Alami

Pengolahan bahasa alami terdiri dari dua bagian utama, yaitu : parser, sistem representasi pengetahuan dan pengolahan output.

a. Parser

   Suatu sistem yang mengambil kalimat input bahasa alami dan menguraikannya ke dalam beberapa bagian gramatikal (kata benda, kata kerja, kata sifat, dan lain-lain).

b. Sistem Representasi Pengetahuan

   Suatu sistem yang menganalisis output parser untuk menentukan maknanya.

c. Output Translator

   Suatu terjemahan yang merepresentasikan sistem pengetahuan dan melakukan langkah- langkah yang bisa berupa jawaban atas bahasa alami atau output khusus yang sesuai dengan program komputer lainnya.

2.1.8.2 Kategori Aplikasi Pengolahan Bahasa Alami

Teknologi Natural Language Processing (NLP) atau pemrosesan bahasa alami adalah teknologi yang memungkinkan untuk melakukan berbagai macam pemrosesan terhadap bahasa alami yang biasa digunakan oleh manusia. Sistem ini biasanya mempunyai masukan dan keluaram berupa bahasa tulisan (teks). *Natural Language Processing* (NLP) mempunyai aplikasi yang sangat luas. Beberapa diantara berbagai kategori aplikasi *Natural Language Processing* (NLP), sebagai berikut:

1. *Natural Language Translator*, yaitu translator dari satu bahasa alami ke bahasa alami lainnya, misalnya translator bahasa Inggris ke bahasa Indonesia, Bahasa Indonesia ke Bahasa Jawa dan sebagainya. Translator bahasa alami bukan hanya kamus yang menerjemahkan kata

per kata, tetapi harus juga mentranslasikan sintaks dari bahasa asal ke bahasa tujuannya.

2. Translator bahasa alami ke bahasa buatan, yaitu translator yang mengubah perintah-perintah dalam bahasa alami menjadi bahasa buatan yang dapat dieksekusi oleh mesin atau komputer. Sebagai contoh, translator yang memungkinkan kita memberikan perintah bahasa alami kepada komputer. Dengansistem seperti ini, pengguna sistem dapat memberikan perintah dengan bahasa sehari-hari, misalnya, untuk menghapus semua file, pengguna cukup memberikan perintah ”komputer, tolong hapus semua file !” Translator akan mentranslasikan perintah bahasa alami tersebut menjadi perintah bahasa formal yang dipahami oleh komputer.
3. Text Summarization, yaitu suatu sistem yang dapat ”membuat ringkasan” hal-hal yang penting dari suatu wacana yang diberikan.

Dalam dunia kecerdasan buatan pengolahan bahasa alami merupakan aplikasi terbesar setelah sistem pakar. Banyak para ahli Artificial Intelligence berpendapat bahwa bidang yang penting yang dapat dipecahkan oleh *Artificial Intelligence* adalah *Natural Language Processing* (Pengolahan Bahasa Alami).

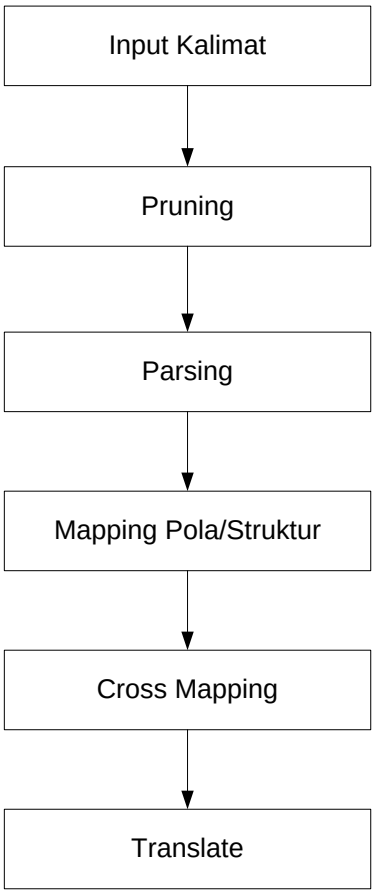
2.1.8.3 Dasar Teori

Riset ini dilaksanakan dengan tujuan untuk menerjemahkan kalimat dalam bahasa inggris ke dalam struktur kalimat berbahasa Indonesia, jadi yang ditekankan dalam riset ini lebih banyak pada struktur kalimatnya dan bukan hanya pada sintaks kalimat saja. Dalam riset ini kamu memakai beberapa referensi sebagai dasar percobaannya. Referensi tersebut memuat komponen-komponen yang digunakan dalam riset, antara lain :

1. Metode parsing kalimat.
2. Membuat link grammar.

3. Algoritma parsing dan link grammar.

Seperti yang telah diketahui bahwa susunan kalimat bahasa inggris sedikit berbeda dengan bahasa Indonesia, maka dari itu dengan riset ini kami mencoba mengatasi permasalahan tersebut.



Gambar 2.10 Prosedur Natural Language Processing (NLP)

2.1.9 HTML

2.1.9.1 Definsi HTML

HTML merupakan singkatan dari *Hyper Text Markup Language.* HTML dapat dibuat pada sembarang *editor teks*. Pembuatan teks HTML hampir sama dengan pembuatan teks-teks lainnya seperti pada *MS. Word*. Pemberian format pada suatu teks dalam sebuah dokumen akan bisa langsung terlihat hasilnya. Contohnya, jika Anda ingin membuat sebuah dokumen pada *MS. Word* dan

memformatnya sehingga salah satu kata/kalimat ingin diberikan format huruf tebal (*bold*), miring (*italic*), atau Garis bawah pada teks (*underline*), maka hasilnya segera dapat dilihat pada dokumen tersebut. Berbeda dengan dokumen HTML, format-format yang diberikan pada suatu teks tidak bisa dilihat langsung hasilnya, tetapi harus menggunakan program khusus, yaitu "*Web Browser*" atau "*Browser*".

2.1.9.2 Tag - Tag Dasar HTML

```
<html>
<head>
  <title>Tes HTML</title>
</head>

<body>
</body>
</html>
```

Gambar 2.11 Contoh tag HTML

Penjelasan kode pada bentuk umum penulisan dokumen HTML diatas :

1. Pasangan tag **<html>** dan **</html>** menandakan bahwa kode yang terdapat didalamnya adalah kode HTML sehingga browser akan menerjemahkan sebagai dokumen HTML.
2. Bagian yang terdapat dalam **<html>** dan **</html>** umumnya terbagi atas tag **<head></head>** dan **<body></body>**.
3. Pada bagian **<head>** dan **</head>** bisa ditentukan judul dokumen HTML yang dituliskan dengan pasangan **<title>** dan **</title>**.
4. Bagian **<body>** dan **</body>** dapat dituliskan teks-teks, penyisipan gambar, *link*, atau pembuatan tabel.

2.1.10 PHP

2.1.10.1 Defensi PHP

PHP adalah singkatan dari *PHP : Hypertext Preprocessor*, bahasa interpreter yang mempunyai kemiripan dengan bahasa pemrograman C dan Perl.

PHP merupakan bahasa pemrograman *server side* yang banyak digunakan untuk membuat website dinamis. Untuk hal-hal tertentu dalam membuat web,

bahasa pemrograman PHP memang diperlukan, misalnya untuk memproses data yang dikirimkan oleh pengunjung suatu website.

2.1.10.2 Sejarah PHP

Pada awalnya PHP merupakan singkatan dari *Personal Home Page* (situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data formulir dari web.

Selanjutnya, Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi *Open Source*, maka banyak programer yang tertarik untuk ikut mengembangkan PHP.

Pada tahun 1997, sebuah perusahaan bernama *Zend* menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang *PHP: Hypertext Preprocessing*.

Pada pertengahan tahun 1999, *Zend* merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

```php
<?php
  echo "Hello world!";
?>
```

Gambar 2.12 Contoh script PHP

2.1.11 Javascript

2.1.11.1 Definsi Javascript

Javascript adalah bahasa yang berbentuk kumpulan *script* yang pada fungsinya berjalan pada suatu dokumen HTML. Sepanjang sejarah internet, bahasa ini adalah bahasa *script* pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengeksekusian perintah perintah di *client side*, yang artinya di sisi browser bukan di *server side* web.

Javascript bergantung kepada *browser* (*navigator*) yang memanggil halaman web yang berisi *script* dari javascript dan tentu saja terselip di dalam dokumen HTML. Javascript juga tidak memerlukan kompilator atau penerjemah khusus untuk menjalankannya (pada kenyataannya kompilator javascript sendiri sudah termasuk di dalam browser tersebut). Lain halnya dengan bahasa pemrograman **"Java"** (dengan nama javascript selalu dibanding-bandingkan) yang memerlukan kompilator khusus untuk menerjemahkannya di *client side*.

2.1.11.2 Sejarah Javascript

Javascript diperkenalkan pertama kali oleh **Netscape** pada tahun 1995. Pada awalnya bahasa ini dinamakan *"LiveScript"* yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2. Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru-buru dan tidak ada pesan kesalahan yang di tampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara **Netscape** dan **Sun** (pengembang bahasa pemrograman *"Java"*) pada masa itu, maka **Netscape** memberikan nama *"JavaScript"* kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan **Microsoft** sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai *"Jscript"* di browser Internet Explorer 3.

```
<html>
<head>
  <title>Tes HTML</title>
</head>

<body>
  <script language="javascript">
    /* Kode Javascript */
    alert('Javascript pertamaku');
    /* Kode Javascript */
  </script>
</body>
</html>
```

Gambar 2.13 Contoh penulisan javascript pada tag HTML

2.1.12 Ajax

Ajax adalah singkatan dari *Asycronous Javascript and XML*, pengertian lebih mudah adalah menggabungkan antara javascript dan xml untuk mengakses sumber data di server. Javascript sebagai pemrograman di sisi *client* bisa digunakan untuk mengakses server secara asinkron, dan XML digunakan untuk format data hasil dari server.

1. Ajax ditulis dengan javascript, memanfaatkan object javascript yang sudah ada yaitu XMLHttpRequest.
2. Ajax tergantung dengan browser, jika browser mendukung javascript, maka bisa dipastikan mendukung ajax.
3. Berdasarkan point 2, ajax merupakan teknologi browser.
4. Ajax digunakan untuk mengakses server, dan *client*/ pengguna menerima hasil dari server tidak secara langsung, tetapi masuk ke dalam mesin ajax terlebih dahulu baru ditampilkan kemudian.

Ajax dikembangkan sekitar tahun 2006. Ajax dapat diintegrasikan dengan *server side programming* seperti PHP, ASP, JSP, dan lain sebagainya.



Gambar 2.14 Mekanisme proses ajax menggunakan PHP

2.1.13 jQuery

jQuery merupakan sebuah framework atau library JavaScript yang dapat membantu kita mempermudah dan mempercepat pengolahan DOM pada halaman web. jQuery sudah mengotomatis pekerjaan-pekerjaan yang umum dan menyederhanakan code yang kompleks. jQuery merupakan salah satu library yang membuat program web di sisi klien, tidak terlihat sebagai program JavaScript biasa yang harus secara eksplisit disisipkan pada dokumen web.

Adapun kemudahan-kemudahan yang diberikan oleh jQuery antara lain:

1. jQuery menawarkan sebuah selector yang robust dan efisien untuk mengambil bagian tertentu pada dokumen yang selanjutnya bisa dimanipulasi.
2. Dapat mengubah tampilan CSS dengan mudah.
3. Dapat menambahkan animasi ke dalam website dengan menggunakan jQuery.
4. jQuery juga mempermudah penggunaan AJAX.

2.1.13.1 jQuery *Selector*

Didalam jQuery terdapat jQuery Selector yang merupakan fungsi utama pada jQuery. Semua fungsi-fungsi pada jQuery dapat diakses melalui selector. Adapun parameter yang bisa digunakan pada jQuery adalah id element, class

element, dan element itu sendiri. Selector juga bisa mengenali hirarki sebagai parameter inputnya dan bisa memfilternya langsung dari ekpresi.

2.1.13.2 jQuery *Manipulation*

Struktur dokumen bisa dimanipulasi menggunakan jQuery Manipulation. Ada 7 kategori yang bisa dimanipulasi, yaitu :

1. Perubahan konten.

   Konten dari sebuah objek dapat diubah dengan dua fungsi yaitu HTML dan teks. Perbedaannya jika menggunakan teks maka semua tag-tag HTML akan dituliskan sebagai mana mestinya.
2. *Insert* ke dalam objek.

   Terdapat dua jenis insert, yaitu *append* dan *prepend*. *Append* adalah menambahkan objek baru setelah *value* dari objek tersebut, sedangkan *prepend* menambahkan objek baru sebelum objek.
3. *Insert* ke luar objek.

   Objek yang akan ditambahkan berada diluar dari masing-masing tag yang dipilih.
4. *Insert* di sekitar objek.
5. Mengganti objek.
6. Menghapus objek.
7. Meng-*copy* objek.

2.1.14 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread, multiuser,* dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber

dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan di sponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya.Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL memiliki beberapa ke istimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi.
2. Open Source. MySQL di distribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. Multiuser. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. Performance tuning. MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Jenis Kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti signed/ unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah **Select** dan **Where** dalam perintah (*query*).

2.1.15 *Tools* Yang Digunakan

Aplikasi / *tools* yang digunakan dalam perancangan *chatbot* multitranslator adalah Adobe Dreamweaver CS5, XAMPP, Google Chrome, MySQL.