

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Landasan Teori**

Landasan teori merupakan penjelasan dari rujukan teori yang relevan yang digunakan untuk menjelaskan teori apa yang berkaitan dalam pembangunan aplikasi penyedia informasi layanan Imunicare Bio Farma menggunakan Chatbot.

#### **2.2 Bio Farma**

Imunicare merupakan bagian dari Klinik Pratama Bio Farma yang memberikan pelayanan kesehatan terpercaya melalui pelayanan vaksinasi. Imunicare hadir dengan petugas yang *expert*, kualitas produk tinggi, jaminan kualitas layanan dan memberikan edukasi konsumen yang informatif. Imunicare mempunyai komitmen untuk menyediakan kebutuhan pelayanan vaksinasi untuk segala jenis usia mulai dari bayi, anak-anak, remaja, dewasa, dan lansia.

#### **2.3 Multimedia**

*Multimedia* berasal dari kata kata *multi* dan *media* yang memiliki makna multi berarti banyak dan media/medium berarti tempat untuk menyampaikan. *Multimedia* secara umum adalah kombinasi dari berbagai media seperti teks, gambar, suara, animasi, video dan lain-lain secara terpadu melalui komputer atau perangkat elektronik lain untuk mencapai tujuan tertentu. Multimedia dibagi menjadi 2 bagian yaitu elemen multimedia yang diskret (tidak berbasis waktu) dan kontinyu (berbasis waktu)

Elemen multimedia diskret diantaranya adalah teks dan gambar. Berikut adalah penjelasan dari teks dan gambar :

- 1) Teks merupakan suatu elemen multimedia yang sangat dasar, pada umumnya teks terdiri dari gabungan kata yang digunakan untuk memberikan suatu pesan/informasi. Penggunaan teks dalam multimedia sangat bermanfaat bahkan bisa dikatakan hampir dari setiap produk multimedia mengandung teks didalamnya, hal tersebut dilakukan untuk dapat memaksimalkan penyampaian informasi.

- 2) Gambar merupakan suatu citra dua dimensi yang bisa dimanipulasi oleh komputer. Beberapa contoh dari gambar diantaranya foto, grafik, ilustrasi, diagram, dan lain-lain. Gambar mempunyai berfungsi sebagai visualisasi konsep verbal atau abstrak. Gambar digunakan agar memperjelas penyampaian informasi verbal.

Elemen multimedia kontinyu diantaranya suara, animasi dan video. Berikut merupakan penjelasan suara, animasi dan video :

- 1) Suara adalah gelombang yang dibangkitkan oleh suatu benda yang bergetar dalam media seperti udara. Suara biasa dapat digunakan untuk memperjelas teks maupun gambar. Suara memiliki sebuah frekuensi dan frekuensi yang bisa didengar oleh pendengaran manusia adalah 20Hz sampai 20.000Hz.
- 2) Animasi merupakan kumpulan-kumpulan gambar yang bergerak secara urut yang menyajikan suatu proses tertentu. Animasi berguna untuk menyajikan suatu peristiwa yang sulit digambarkan, dalam multimedia animasi banyak digunakan dalam menyajikan materi pembelajaran.

Video adalah suatu rekaman peristiwa yang terdapat urutan gambar bergerak yang disertai dengan suara. Video digital menjadi suatu elemen multimedia yang cukup populer dikarenakan mudahnya dalam pengolahannya, akan tetapi video membutuhkan tempat penyimpanan yang cukup besar.

## 2.4 Text Mining

*Text Mining* yaitu adalah upaya pencarian atau penambangan data yang berupa teks dimana sumber data biasanya diperoleh dari sebuah dokumen, dengan tujuan mencari kata-kata yang bisa mewakili isi dokumen sehingga dapat melakukan analisis keterhubungan antar dokumen yang ada. *Text mining* adalah penerapan konsep dan teknik data mining untuk mencari pola dalam teks. Atau proses penganalisisan teks guna menyarikan informasi yang bermanfaat untuk tujuan tertentu yang ada. *Text mining* didefinisikan sebagai proses menggali informasi dimana seorang user dapat berinteraksi dengan sekumpulan dokumen menggunakan tools analisis yang merupakan komponen-komponen dalam data mining yang salah satunya adalah kategorisasi [12] . Secara garis besar, proses penalaran pada *chatbot* dengan menggunakan text mining mempunyai beberapa

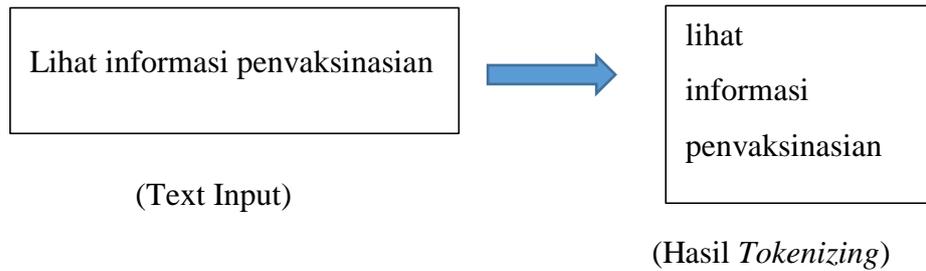
proses. Proses preprocessing ini meliputi beberapa tahap. Tahap-tahap tersebut adalah :

1. Parsing/ Tokenizing
2. Stopwords Removal/Filtering
3. Stemming
4. Analyzing

**a. Parsing/Tokenizing**

Tokenizing merupakan proses yang dilakukan seseorang untuk menjadikan sebuah kalimat mempunyai arti lebih bermakna atau berada dengan cara memecah kalimat tersebut menjadi sebuah kata-kata atau frase-frase (“Parsing”). Parsing di dalam pembuatan aplikasi Text mining ini yaitu proses penguraian deskripsi yang semula berupa kalimat yang berisi kata-kata dan tanda pemisah antara kata seperti titik(.), koma(,), spasi dan tanda pemisah lain menjadi kata-kata saja baik itu berupa kata-kata penting maupun kata-kata tak penting. Sederhananya proses parsing ini terlihat sebagai proses pengambilan kata jika ketemu tanda spasi namun pada kenyataannya tidak sesederhana itu.

Proses parsing tidak hanya bisa dilakukan pada proses information retrieval, melainkan juga dalam bidang yang lain seperti pada pembuatan sebuah compiler dan bahasa alami. Sebelumnya perlu diketahui arti dari sebuah istilah parser adalah program yang dapat melakukan proses parsing. Untuk pemrosesan, dokumen akan dipisahkan menjadi unit-unit yang lebih kecil misalnya berupa kata, frase atau kalimat. Unit pemrosesan tersebut dikatakan sebagai token. Parsing merujuk pada proses pengenalan token yang berada dalam rangkaian teks. Proses parsing (penguraian kalimat) juga merupakan proses yang dilakukan untuk menerjemahkan masukan dari user agar bisa dimengerti oleh sistem. Secara default, keseluruhan kalimat masukan pengguna yang dimasukkan akan dianggap sebagai kata-kata yang harus ada pada data yang akan dicari. Contoh tahap ini bisa dilihat pada gambar 2.1

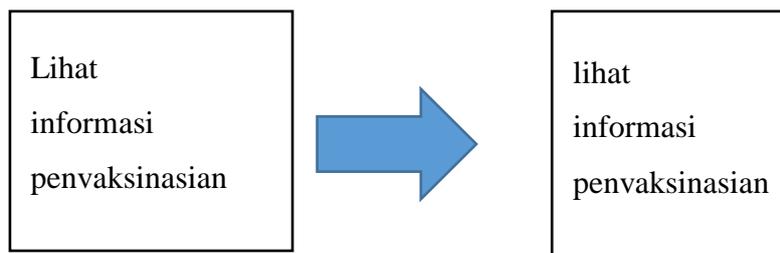


**Gambar 2.1 Tahap *Tokenizing***

### **b. Filtering**

Kebanyakan bahasa resmi di berbagai negara mempunyai kata fungsi dan kata sambung seperti artikel dan preposisi yang hampir selalu muncul pada dokumen-dokumen teks. Biasanya kata-kata ini mempunyai arti yang lebih dalam memenuhi kebutuhan seorang searcher saat mencari informasi. Di dalam bahasa Indonesia stopwords dapat disebut sebagai kata tidak penting misalnya “di”, “oleh”, “pada”, “sebuah”, “karena”. Sebelum proses stopwords removal dilakukan, lebih dulu dibuatkan daftar stopwords (stoplist). Preposisi, kata hubung dan partikel biasanya merupakan kandidat stoplist. Daftar Stoplist (stopword) dapat dilihat pada Lampiran E.

Stopwords removal merupakan proses penghilangan kata tidak penting pada deskripsi pada pengecekan kata-kata hasil parsing deskripsi apakah kata tersebut termasuk di dalam daftar kata tidak penting (stoplist) atau tidak. Jika termasuk di dalam stoplist maka kata-kata tersebut akan di-remove dari deskripsi sehingga kata-kata yang tersisa dalam deskripsi di anggap sebagai kata penting atau keywords. Tahap filtering dapat dilihat pada gambar 2.2

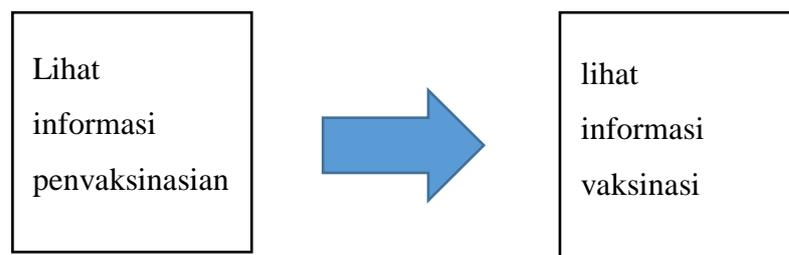


**Gambar 2.2 Tahap *Filtering***

### c. *Stemming*

*Stemming* merupakan proses dalam pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya (*stem*). Proses ini juga dikatakan sebagai *conflation*. Proses stemming secara luas sudah digunakan di dalam *Information retrieval* (pencarian informasi) untuk meningkatkan kualitas informasi yang bisa didapatkan. Kualitas informasi yang dimaksud adalah untuk mendapatkan hubungan antara varian kata yang satu dengan yang lainnya. Sebagai contoh kata “diculik”, “menculik” (melakukan tindakan menculik) dan “penculik” (orang yang menculik) yang semula mengandung arti yang berbeda dapat di-stem menjadi sebuah kata “culik” yang memiliki arti yang sama sehingga kata-kata di atas saling berhubungan satu samalain.

Selain itu juga stemming dapat digunakan untuk mengurangi ukuran dari suatu ukuran index file. Misalnya dalam suatu deskripsi terdapat varian kata “memberikan”, “diberikan”, “memberi” dan “diberi” hanya memiliki akar kata (stem) yaitu “beri”. Ukuran file daftar index yang semula berjumlah empat record akan di-reduce sehingga menjadi tiga record saja. Tahap stemming dapat dilihat pada gambar 2.3



**Gambar 2.3 Tahap *Stemming***

Algoritma stemming untuk bahasa yang satu akan berbeda dengan bahasa algoritma stemming lainnya. Sebagai contoh, bahasa Inggris memiliki morfologi yang berbeda dengan bahasa Indonesia sehingga algoritma stemming untuk kedua bahasa tersebut berbeda. Proses stemming pada teks berbahasa Indonesia lebih rumit dan kompleks karena terdapat variasi imbuhan yang harus dibuang agar mendapatkan root word (kata dasar) dari sebuah kata. Pada umumnya kata dasar pada bahasa Indonesia terdiri dari beberapa kombinasi.

[ DP + DP + DP + Kata Dasar + DS + PP+ P]

- DP : *Derivation Prefix* (awalan)  
 DS : *Derivation Suffixes* (akhiran)  
 PP : *Possesive Pronouns* (kepunyaan, contoh “-ku”, “-mu”)  
 P : *Particels* (contoh “-lah”, “-kah”)

Algoritma stemming Nazief dan Adriani ini dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan dan mengenkapsulasi imbuhan-imbuhan, termasuk di dalamnya adalah awalan (prefix), sisipan 17 (infix), akhiran (suffix) dan gabungan awalan-akhiran (confixes). Algoritma yang dikhususkan untuk stemming adalah bahasa Indonesia dengan adanya 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 juga memiliki kemampuan persentase keakuratan lebih baik dari algoritma lainnya.

Algoritma ini menggunakan kamus kata dasar dan mendukung recoding, yaitu penyusunan kembali kata-kata yang mengalami proses stemming berlebih. Kombinasi awalan dan akhiran yang dilarang dapat dilihat pada tabel 2.1.

**Tabel 2.1 Kombinasi Awalan-Akhiran yang Dilarang**

Awalan (prefix)	Akhiran (suffix) yang tidak dipebolehkan
Be-	-i
di-	-an
Ke-	-i, -kan
Me-	-an
Se-	-i, -kan
Te-	-an

Algoritma stemmer Nazief dan Adriani ini memiliki tahap-tahap sebagai berikut:

1. Cari kata yang akan di-stemming dalam kamus kata dasar. Jika ditemukan maka dapat diasumsikan kata tersebut adalah 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 agar dapat menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “- nya”), jika ada.
3. Hapus *Derivation Suffixes* (DS) (“-i”, “-an”, atau “-kan”). Jika kata tersebut ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah 3a.
  - a) Jika “-an”, sudah dihapus dan huruf terakhir dari kata tersebut adalah “- k”, maka “-k” juga akan ikut dihapus. Jika kata tersebut ditemukan dalam kamus kata dasar maka algoritma akan berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
  - b) Akhiran yang dihapus (“-i”, “-an”, atau “-kan”) akan dikembalikan, lanjut ke langkah 4
4. Hapus *Derivation Prefixes* (DP) (“di-”, “ke-”, “se-”, “me-”, “be-”, “pe-”, “te-”) dengan iterasi maksimum adalah 3 kali:
  - a) Langkah 4 berhenti jika: Terjadi kombinasi awalan dan akhiran yang terlarang seperti pada Tabel 2.1. Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya. Tiga awalan telah dihilangkan.
  - b) Identifikasikan tipe awalan dan hilangkan. Awalan ada dua tipe: Standar: “di-”, “ke-”, “se-” yang bisa langsung dihilangkan dari kata tersebut. Kompleks: “me-”, “be-”, “pe”, “te-” adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel 2.2 untuk mendapatkan pemenggalan yang tepat.
  - c) Cari kata yang sudah dihilangkan awalnya yang ada di dalam kamus. Apabila tidak ditemukan, maka langkah 4 dapat diulangi kembali. Apabila ditemukan, maka keseluruhan proses dihentikan.

**Tabel 2.2 Aturan Pemenggalan Awalan Stemmer Nazief dan Andriani**

Aturan	Format Kata	Pemenggalan
1	berV....	ber-V... ber-rV...
2	berCAP....	ber-CAP...dimana C!=’r’ & P!=’er’
3	berCAerV...	ber-CaerV...dimana C!=’r’
4	Belajar	bel-ajar
5	berC <sub>1</sub> erC <sub>2</sub> ...	berC <sub>1</sub> erC <sub>2</sub> ... dimana C <sub>1</sub> !={’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 <sub>1</sub> erC <sub>2</sub> ...	te-C <sub>1</sub> erC <sub>2</sub> ...dimana C <sub>1</sub> !=’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	menyV...	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		
23	perCAP...	per-CAP... dimana C!=’r’ dan P!=’er’
24	perCAerV...	per-CAerV... dimana C!=’r’
25	pem{b f V}...	Pem-{b f V}...
26	pem{rV V}...	Pe-m{rV V}...   pe-p{rV V}...
27	pen{c d j z}...	pen-{c d j z}...
28	penV...	pe-nV...   pe-tV...
29	peng{g h q}...	peng-{g h q}...
30	pengV...	peng-V...   peng-kV...
31	penyV...	penyV...
32	peIV...	pe-IV... kecuali “pelajar” yang menghasilkan “ajar”
33	peCerV...	per-erV... dimana C!={r w y l m n}
34	peCP...	pe-CP... dimana C!={r w y l m n} dan P!=’er’

- Namun apabila setelah langkah 4 kata dasar masih juga belum ditemukan, maka saat proses recoding dilakukan dengan mengacu pada aturan pada Tabel 2.2. Recoding dilakukan menambahkan karakter recoding di awal kata yang dipenggal. Pada Tabel 2.2, karakter recoding adalah huruf kecil setelah tanda hubung (‘-’) dan biasanya terkadang berada sebelum tanda kurung.
- Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai kata dasar. Proses selesai.

#### **d. Analyzing**

Tahap ini merupakan tahap penentuan seberapa jauh keterhubungan antara kata-kata dengan dokumen yang telah ada. Pada proses ini hasil filtering dicocokkan dengan list pertanyaan. Hasil kata yang cocok paling banyak maka akan dijadikan sebagai penentuan jawaban atas pertanyaan yang diajukan

### **2.5 Chatbot**

Chatbot yaitu salah satu program dalam kecerdasan buatan yang dirancang untuk dapat berkomunikasi secara langsung dengan manusia. Yang membedakan Chatbot dengan sistem pemrosesan bahasa alami (Natural Language Processing System) adalah kesederhanaan Algoritma yang digunakan. Meskipun banyak bots yang dapat menginterpretasikan dan menanggapi input manusia, sebenarnya bots tersebut hanya bisa mengartikan kata kunci yang diinputkan dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip dari data yang telah ada dalam database yang telah dibuat sebelumnya. Seolah-olah bot tersebut adalah seseorang yang sedang berbicara dengan pengguna padahal bot adalah sebuah kecerdasan buatan atau *Artificial Intelligence*.

Chatbot pada dasarnya memiliki 2 komponen utama yaitu Chat yang dapat diartikan sebagai pembicaraan dan Bot merupakan sebuah program yang mengandung sejumlah data, jika diberikan masukan maka akan memberikan jawaban. Chatbot dapat menjawab pertanyaan dengan membaca tulisan yang diketikkan oleh pengguna melalui keyboard[13].

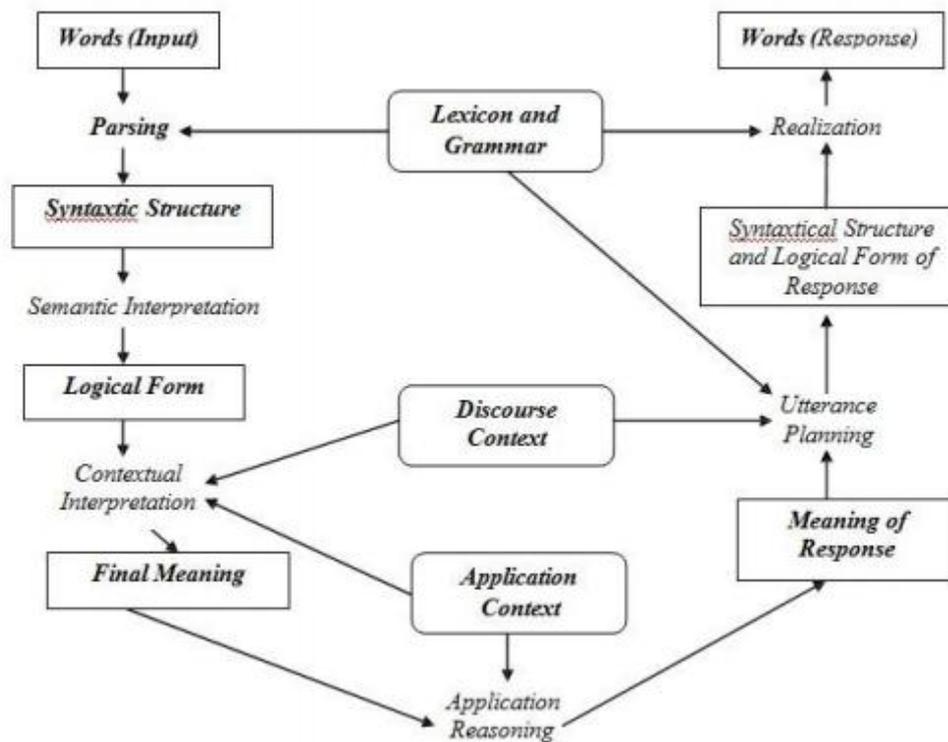
### **2.6 Natural Language Processing (NLP)**

Natural Language Processing (NLP) merupakan bertujuan untuk mengembangkan program komputer yang dapat memproses bahasa alami yang dapat dipahami oleh manusia. Tujuannya yaitu untuk memungkinkan komputer bisa berkomunikasi dengan manusia yaitu dengan cara yang sama seperti manusia berkomunikasi dengan manusia lain. Bahasa alami atau yang dapat disebut dengan natural language adalah suatu bahasa yang diucapkan, ditulis, disyaratkan oleh manusia untuk berkomunikasi umum. Jadi bahasa alami/ natural language yaitu bahasa yang dimengerti oleh manusia. Bahasa alami pada prinsipnya adalah suatu

bentuk representasi dari suatu pesan yang akan dikomunikasikan antar manusia.

Komputer dapat memahami pertanyaan dalam bahasa alami dengan cara komputer harus mempunyai pengetahuan analisis dan interpretasi input dalam data *knowledge*. Komputer harus mengerti gramatika dan definisi kata-kata. Dalam hubungan ini teknik AI digunakan untuk menampilkan pengetahuan internal dan mengolah input. Pelacakan klasik dan teknik penyocokan pola (*pattern matching*) digunakan bersama dengan basis pengetahuan agar komputer bisa mengerti apa yang pengguna masukkan dalam bahasa alami. Bila komputer sudah mengerti ucapan yang pengguna berikan, maka komputer bisa melakukan hal-hal yang pengguna harapkan/respon kembali yang dinyatakan atau diekspresikan dalam bahasa alami juga[14].

Untuk mencapai tujuan yang sudah ada tersebut dibutuhkannya tiga tahap proses. Proses yang paling pertama adalah parsing atau analisa sintaksis yang memeriksa kesetrukturan kalimat yang ada menggunakan suatu grammar (tata bahasa) dan lexion (kosa kata) tertentu. Proses kedua adalah *semantic interpretation* atau interpretasi semantik yang memiliki bertujuan untuk merepresentasikan arti dari kalimat secara *contextindependent* untuk keperluan yang lebih lanjut. Sedangkan proses ketiga adalah *contextual interpretation* atau interpretasi kontekstual yang bertujuan merepresentasikan arti secara *context-dependent* dan menentukan maksud dari penggunaan kalimat tersebut. Gambaran sebuah organisasi sistem NLP dapat dilihat pada Gambar 2.4.



**Gambar 2.4 Organisasi NLP [14]**

Jenis aplikasi yang bisa dibuat pada bidang bidang natural language adalah *text - based application* dan *dialogue - based applications*

#### 1. *Text - based application*

Mencakup segala macam aplikasi yang melakukan proses terhadap text tertulis seperti misal buku, berita di surat kabar, e-mail dan lain-lain. Contoh penggunaan dari *text - based application* ini adalah :

- a. mencari topik tertentu dari buku yang ada pada perpustakaan.
- b. memberikan respon atas input yang diberikan.
- c. mencari isi dari surat atau *e-mail*.
- d. menterjemahkan dokumen dari satu bahasa ke bahasa yang lain.

#### 2. *Dialogue - based application*

Idealnya pada pedekatan ini melibatkan bahasa lisan atau pengenalan suara, akan tetapi bidang ini juga menginputkan interaksi dengan cara memasukkan teks pertanyaan melalui keyboard. Aplikasi yang sering ditemui untuk bidang ini adalah :

- a. sistem tanya jawab, dimana *natural language* digunakan dalam mendapatkan informasi dari suatu *database*.
- b. sistem otomatis pelayanan melalui telepon.
- c. kontrol suara pada peralatan sistem.
- d. *sistem problem solving* yang membantu untuk melakukan penyelesaian masalah yang umum dihadapi.

Pada dasarnya, NLP dan text mining merupakan teknologi artificial intelligence (AI) yang dapat memungkinkan penggunaanya mengubah konten inti dari sebuah dokumen teks menjadi sebuah data kuantitatif secara lebih cepat. Data kuantitatif tersebut nantinya akan dapat digunakan atau ditindaklanjuti sesuai keinginan penggunaanya.

Dengan bantuan teknologi NLP dan text mining, proses dapat menjadi jauh lebih mudah dan efisien dibandingkan proses manual, dan juga menarik informasi atau wawasan yang mungkin tidak bisa didapatkan secara manual. Informasi tersebut kemudian bisa diolah menjadi data terstruktur untuk dianalisa dan divisualisasikan. Penggunaan NLP dan text mining bisa membantu untuk meningkatkan efisiensi analisis teks yaitu dengan memberikan kemampuan otomasi pada proses tersebut. NLP dan text mining biasanya digunakan dalam:

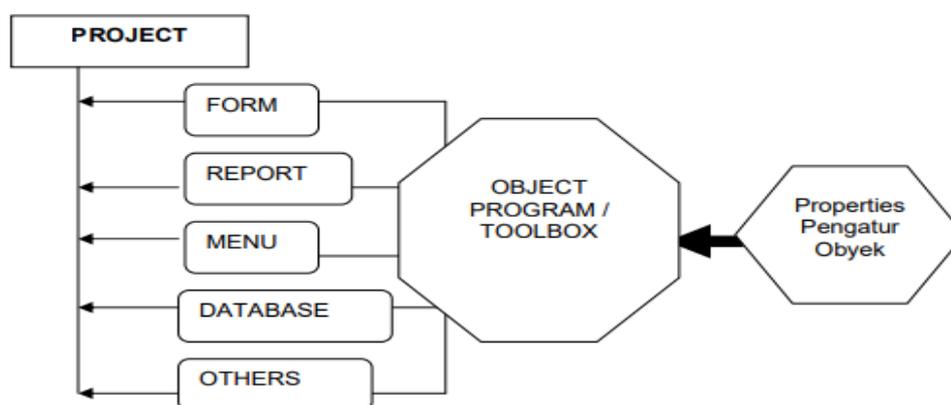
- a. Menganalisa sebuah survey dimana para peserta menuliskan komentarnya secara bebas
- b. Mengklasifikasi email sebagai spam, marketing, atau personal (contoh: Gmail)
- c. Menganalisa klaim asuransi atau garansi, wawancara diagnostik, dll
- d. Meneliti kompetitor dengan melakukan crawling dalam website merek

## **2.7 Penarikan Kesimpulan**

Kemampuan berfikir (reasoning) juga termasuk didalamnya proses penarikan kesimpulan (inferencing) berdasarkan fakta-fakta dan aturan dengan menggunakan metode heuristic atau pencarian lain didalamnya.

## 2.8 Analisis Berorientasi Objek

Pemrograman berorientasi Objek sekarang ini merupakan teknik pemrograman yang paling populer dan banyak digunakan oleh Programmer untuk menggantikan teknik pemrograman berbasis prosedur. “OOP (Object Oriented Programming) adalah sebuah pendekatan untuk pengembangan suatu software, dimana dalam struktur software tersebut didasarkan kepada interaksi objek dalam penyelesaian suatu proses atau tugas. Dan juga suatu cara baru dalam berpikir serta berlogika untuk menghadapi masalah – masalah yang akan dicoba batasi dengan bantuan Komputer [15]. Gambaran tentang pemrograman ini seperti dibawah ini :



**Gambar 2.5 Model Pemograman OOP**

Dan juga OOP memiliki beberapa konsep yaitu dapat dilihat sebagai berikut[16] :

### a. Abstrak Class

1. Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.
2. Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.
3. Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.
4. Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (method), dan sifat kelas (property).

**b. Encapsulation**

1. Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.
2. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.
3. Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.
4. Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

**c. Inheritance**

1. Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
2. Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
3. Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
4. Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu

**d. Polymorphism**

1. Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.
1. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
2. Method overriding.
3. Method name overloading.

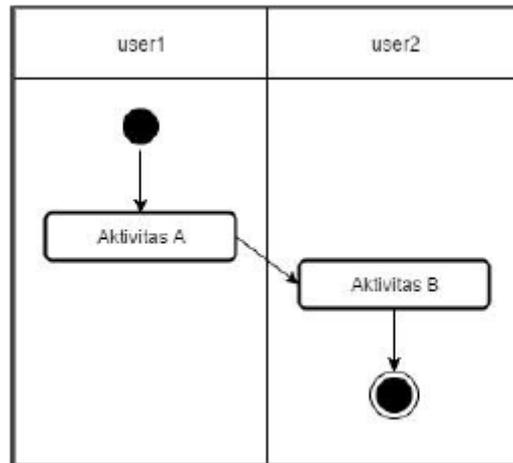
Dan tentunya UML (*Unified Modeling Language*) merupakan sebuah bahasa visual sebagai pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung[17]. UML singkatan dari Unified Modeling Language juga dapat berarti bahasa pemodelan standar. sebagai bahasa, berarti UML mempunyai sintaks dan semantik. Ketika membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang dibuat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya, Bagaimana sistem mengatasi eror yang terjadi, Bagaimana keamanan terhadap sistem yang kita buat, Dan sebagainya dapat dijawab dengan UML[18]. UML diaplikasikan untuk suatu maksud tertentu, biasanya antara lain :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisis dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML menyediakan 9 macam diagram untuk memodelkan aplikasi berorientasi objek yang 4 diantaranya adalah[18] :

1. *Activity Diagram*

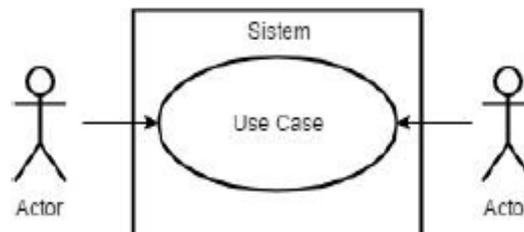
Diagram Aktivitas (*Activity Diagram*), bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini sangat penting dalam suatu sistem serta pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antarobjek. Berikut activity diagram dapat dilihat pada Gambar 2.6 *Activity Diagram* :



**Gambar 2.6 Activity Diagram**

## 2. Use Case Diagram

Diagram Use Case, bersifat statis. Diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk pengorganisasian dan memodelkan suatu perilaku sistem yang dibutuhkan serta diharapkan pengguna. Berikut use case diagram dapat dilihat pada Gambar 2.7 Use Case Diagram :



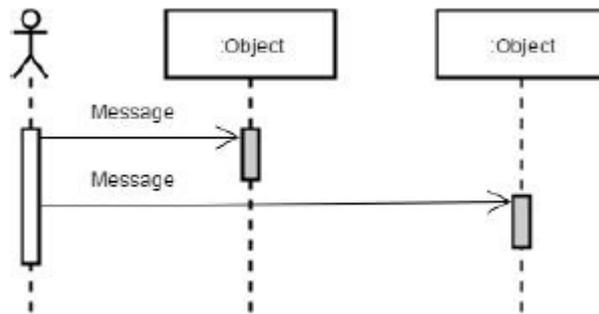
**Gambar 2.7 Use Case Diagram**

## 3. Use Case Scenario

Use case scenario merupakan urutan dari langkah-langkah yang menerangkan pengguna dengan System. Setiap scenario akan mendeskripsikan urutan dari kejadian/peristiwa. Use case scenario bisa dikatakan merupakan rincian informasi dari sebuah use case

#### 4. *Sequence Diagram*

Diagram Sequence (Urutan), bersifat dinamis. Dinamis urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu. Berikut sequence diagram dapat dilihat pada gambar 2.7 Sequence Diagram :



**Gambar 2.8 Sequence Diagram**

#### 5. *Class Diagram*

Diagram Kelas, bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

### 2.9 Perangkat Lunak Pengembangan (Tools)

Perangkat Lunak Pengembang merupakan perangkat lunak yang membantu dalam pembuatan aplikasi sistem yang akan digunakan.

#### 2.9.1 Sublime Text

*Sublime Text* merupakan teks editor berbasis *Python*, sebuah teks editor yang elegan, kaya akan fitur, cross-platform, mudah dan simpel dan dikenal penulis, dan desainer. Sublime Text Editor mendukung banyak bahasa pemrograman yang ada didalamnya dan bahasa markup, dan fungsinya dapat ditambah dengan plugin,

dan Sublime Text Editor tanpa lisensi perangkat lunak. Sublime Text mendukung operation system seperti Linux, Mac Os X, dan juga windows. Sublime Text Digunakan sebagai media Text Editor yang berfungsi sebagai Pengembangan atau perancangan halaman web.

#### **Kelebihan Sublime Text :**

- a. Aplikasi yang Ringan  
Tidak memakan banyak ram seperti text editor atom
- b. Split Mode  
dapat membuka 2-4 column 4 grid dan 2-3 row editor dalam sekali membuka aplikasi, artinya kita dapat membuka 2 atau lebih sekaligus progam dalam waktu bersamaan tanpa harus berpindah-pindah tab.
- c. Multi Selection  
kemampuan untuk dapat merubah beberapa kode di baris atau kolom yang berbeda dalam waktu bersamaan.
- d. Command Pallette  
bisa mengakses file shortcut dari sublime text. untuk menjalankannya kalian tinggal klik di keyboard **CTRL+SHIFT+P**
- e. Sudah mendukung banyak platform  
kalau yang ini mah udah pasti karena Sublime Text berdiri sejak lama dan Sudah terkenal jadi Sudah pasti juga kalau aplikasi ini Multi Platform.

#### **Kekurangan Sublime Text :**

- a. Sublime text yaitu aplikasi berbayar, jadi jika ingin memakainya dapat membeli untuk mendukung devlopnnya.
- b. ada juga beberapa plug-in Notepad++ yang belum ada di Sublime text.
- c. sidebar dari sublime tidak bisa di-hiden, maksudnya jika kita akan menghidien sidebar lumayan repot jika belum tahu shortcur dari keyboard nya, tidak seperti text editor atom yang bisa dengan mudahnya menampilkan dan menyembunyikan tanpa harus tahu shortcut keyboard.

## 2.9.2 PHP

PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda jika dibandingkan dengan bahasa pemrograman client-side yang lain seperti *JavaScript* yang diproses pada web browser (*client*).

## 2.9.3 MySQL

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

Dengan menggunakan SQL, proses akses database menjadi lebih user - friendly dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah - perintah pemrograman. MySQL merupakan software database yang paling populer yang ada di lingkungan Linux, kepopuleran ini karena ditunjang performa query dari databasenya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. MySQL ini juga sudah dapat berjalan pada lingkungan Windows.

Perintah untuk mengelola database dibagi menjadi 3 (tiga ) kelompok, diantaranya :

1. Perintah untuk mendefinisikan data/DDL (Data Definition Language).
2. Perintah untuk memanipulasi data/DML (Data Manipulation Language).
3. Perintah untuk mengendalikan data/DCL (Data Control Language).

## 2.9.4 Botman

*Botman* merupakan sebuah framework agnosik artinya botman ini bisa digunakan dalam basis kode yang menggunakan kerangka apapun yang mungkin dapat digunakan. Botman juga adalah sebuah library yang telah di sediakan dalam

bentuk framework yang dapat bekerja sama dengan platform dan web driver yang bisa di pasang kapan saja dan di mana saja. Botman juga dapat membuat chatbot real time yang tidak perlu menggunakan API atau aplikasi pihak ketiga(pendukung). Chatbot yang disediakan oleh botman dapat langsung berinteraksi dengan pengguna yang dapat meliputi tahapan tanam koding berbasis php. Yang mana dapat langsung digunakan dengan atau adanya proses hosting. Botman juga adalah kerangka kerja pustaka PHP agnostik yang mana dalam perancangannya adalah untuk menyederhanakan tugas kerja manusia dalam mengembangkan bot inovatif untuk berbagai platform pengiriman pesan, termasuk Slack Telegram Kerangka Kerja *Microsoft Bot*, *Nexmo*, *HipChat*, *Facebook Messenger*, *WeChat* dan banyak lagi. botman juga dapat terinstal dengan mudah jika pada saat proses penginstalan konfignya mengalami masalah pada saat instalasi maka beri tahu kami dengan datang langsung ke situs resmi botman dan bergabung dengan komunitas Slack developer untuk sharing tentang aplikasi chatbot yang dikerjakan dengan anggota dari seluruh dunia. Botman juga memiliki cara sederhana untuk memungkinkan user mengambil informasi yang relevan bagi pengguna yang menggunakan botman sebagai frameworknya. test uji coba botman dengan memberikan perintah dan melakukan percakapan dengan memasukan text yang sudah disediakan botman akan merespon percakapan atau text yang sudah di masukan secara realtime dengan metode `getUser` pada botman nya sendiri.

Botman juga di luncurkan dengan dukungan sejumlah saluran olahpesan yang berbeda. Setiap saluran didukung oleh drivernya sendiri yang mana dapat di install secara manual atau melalui Botman itu sendiri dengan memanfaatkan saluran pesan. jika user ingin instalasi botman secara manual ada tahapan yang harus diperhatikan yang mana tahapan tersebut sudah botman sediakan di web [botman.io](http://botman.io). botman juga mempunyai website premium dimana didalamnya mempunyai 3 paket yaitu normal, premium, gold dimana disetiap paket tersebut memiliki isi seputar video tutorial, dokumentasi dan arahan menggunakan botman secara lengkap dengan prosesi yang berbeda setiap paketnya.

### 2.9.5 API

*API* adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk dapat mengintegrasikan dua bagian pada aplikasi atau dengan aplikasi yang berbeda secara bersamaan. *API* terdiri dari berbagai elemen seperti *function*, *protocols*, dan *tools* lainnya yang memungkinkan *developer* untuk membuat aplikasi. Tujuan penggunaan *API* merupakan untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu lagi membuat fitur yang sama. Penerapan *API* akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan *payment gateway*. Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk sistem operasi, *library*, dan web.

Seperti namanya, Web *API* dalam diakses melalui protokol *HTTP*, ini merupakan konsep bukan teknologi. Kita dapat membuat Web *API* yaitu dengan menggunakan teknologi yang berbeda seperti *PHP*, *Java*, *NET*, dll.

### 2.9.6 Laravel

*Laravel* adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (model view controller). *Laravel* merupakan pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas yang ada padaperangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu penggunaan.

### 2.9.7 Webhook

*Webhook* merupakan konsep *API* yang pada saat ini semakin populer digunakan. Semakin banyak yang dapat dilakukan di web, menjadikan webhook makin banyak digunakan. Selain itu, webhook sangat berguna dan mudah untuk diterapkan. *Webhook* atau yang biasa disebut callback adalah cara bagi suatu aplikasi untuk menyediakan aplikasi lain dengan informasi secara real-time. Lebih

mudahnya, webhook adalah link URL yang ditambahkan agar data yang dikirim dapat langsung diterima di waktu sama dengan link URL yang sudah ditentukan.