# BAB II
# LANDASAN TEORI

## 2.1. Tugas Akhir

Tugas Akhir adalah suatu karya tulis ilmiah, berupa hasil tulisan penelitian membahas tentang masalah dalam bidang tertentu dengan menggunakan aturan-aturan yang berlaku dalam bidang ilmu tersebut. Dibuat untuk pemecahan masalah tertentu dengan menggunkan aturan-aturan yang berlaku dalam bidang tersebut. Penelitian adalah perwujudan dari metode ilmiah, yaitu usaha atau kegiatan memecahkan masalaha berdasarkan langkah-langkah berfikir ilmiah. Tujuan utama penelitian adalah untuk mengembangkan dasar-dasar pengetahuan ilmiah untuk praktek yang efektif dan efisien. Peneliti bertanggung jawab kepada masyarakat dalam hal penyediaan kualitas pelayanan dan merumuskan cara-cara untuk meningkatkan mutu hidup masyarakat.

Secara umum, penelitian skripsi atau tugas akhir bertujuan untuk mengembangkan ilmu dan pengetahuan yang sudah ada, serta adanya fakta dan temuan-temuan baru sehingga dapat disusun sebuah teori, konsep, hukum, kaidah atau metodologi baru yang dapat digunakan untuk memecahkan masalah yang ada. Tujuan Khususnya adalah :

1. Membuktikan teori-teori yang sudah ada. Begitu banyak penelitian dan teori-teori lama yang menunggu untuk dibuktikan apakah hasil penelitian dan teori-teori yang sudah ada tersebut masih releven dengan keadaan saat ini.
2. Menemukan adanya teori-teori baru atau produk yang baru, Perkembangan zaman dan kebutuhan yang ada sekarang menuntut untuk Penemuan teori atau produk baru yang akan lebih memudahkan manusia untuk memenuhi kebutuhannya. Selain berupa produk dan juga teori-teori baru, penemuan juga dapat berupa teknik atau hasil ilmu pengetahuan lainnya yang bisa dimanfaatkan oleh masyarakan untuk memperbaiki kualitas hidupnya.
3. Mengembangkan hasil penelitian sebelumnya, Tujuan dari penelitian-penelitian ini menitikberatkan untuk perkembangan ilmu pengetahuan dan teknologi melalui perkembangan dari hasil penelitian yang sudah ada kemudian dipadukan dengan penelitian yang baru.

## 2.2. *Information Retrieval*

*Information Retrieval* (IR) adalah ilmu yang mempelajari prosedur-prosedur dan metode-metode untuk menemukan kembali informasi yang terdapat pada berbagai sumber yang ada. Dengan melakukan *indexing, searching,* pemanggilan data kembali (*recalling*) dan seterusnya.

### 2.2.1 Definisi Information Retrieval

Menurut DwijaWisnu B, Anandini Hetami (2015) *Information Retrieval* merupakan sistem yang menerima *query* dari pengguna, kemudian dilakukan ranking terhadap dokumen berdasar kesesuian terhadap *query*. Hasil ranking yang diberikan pada pengguna merupakan dokumen yang menurut sistem memiliki relevansi terhadap *query*, tetapi tingkat relevansi itu sendiri merupakan hal yang subjektif tergantung dari pengguna yang dipengaruhi oleh berbagai macam faktor seperti topik, pewaktuan, sumber informasi maupun tujuan pengguna. Model sistem temu kembali menentukan detail sistem temu yaitu meliputi representasi dokumen maupun *query*, fungsi pencarian (*retrieval function*), dan notasi kesesuaian (*relevance notation*) dokumen terhadap *query*.

Menurut Smeaton (1990) memformulasikan tujuan dari Sistem Temu Kembali Informasi ialah, terambilnya dokumen berdasarkan permintaan pengguna dengan harapan bahwa *content* atau isi dari dokumen yang terambil tersebut relevan dengan kebutuhan informasi pencari informasi. Sedangkan Secara teknis, tujuan Sistem Temu Kembali Informasi adalah mencocokkan *term-term* atau istilah dari *query*, dengan *term* atau indeks yang ada dalam dokumen.

Menurut DwijaWisnu B, Anandini Hetami (2015) *Information Retrieval* terbagi dari beberapa bagian yang dijabarkan sebagai berikut:

1. *Text Operations*, yaitu meliputi pemilihan kata-kata dalam sebuah *query* maupun dokumen dalam proses *transformasi* dokumen atau *query* menjadi *term* index.
2. Pembobotan, memberikan bobot pada indeks dari *query*.
3. Perankingan, menggurutkan dokumen berdasarkan kemiripannya dengan *query*.

4. Indexing, membangung basis data indeks dari koleksi dokumen Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

*Information Retrieval* menggambarkan bagaimana suatu metode pencarian informasi yang dilakukan oleh user dari suatu gudang penyimpanan yang bersekala besar, Terkadang ketika data yang dimiliki semakin banyak sebuah media penyimpanan masalah yang muncul adalah kita akan lupa dimana kita meletakan data yang kita simpan, sehingga kita perlu melakukan proses pencarian data, menggunakan *tools* pencarian atau bisa dengan memeriksa satu persatu tempat penyimpanan data kita.

## 2.3. Klasifikasi

Menurut Eko Prasetyo (2012), klasifikasi merupakan suatu kegiatan menilai suatu objek data dengan cara memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang sudah ditentukan. Dalam klasifikasi terdapat dua proses yaitu yang pertama adalah membangun model untuk disimpan sebagai memori dan menggunakan model tersebut untuk melakukan pengenalan atau klasifikasi pada suatu data lain supaya diketahui di kelas mana objek data tersebut dimasukkan berdasarkan model yang telah disimpan dalam memori.

Klasifikasi Merupakan metode menganalisis data yang digunakan untuk membentuk suatu model yang mendeskripsikan kelas data yang penting, atau model yang memprediksikan trend dari sekelompok data. Klasifikasi digunakan untuk memprediksikan kelas data yang bersifat *categorical*, sedangkan prediksi untuk memodelkan fungsi yang mempunyai nilai continuous.

Contoh:

1. Model klasifikasi yang dibangun untuk mengkategorisasikan aplikasi-aplikasi bank sebagai aplikasi yang aman atau beresiko.
2. Model predikasi yang dibangun untuk memprediksikan pengeluaran konsumen berdasarkan pendapatan dan pekerjaannya.

## 2.4. *Clustering*

*Clustering* adalah suatu proses pembagian elemen-elemen data kedalam kelompok yang berbeda (disebut sebagai *cluster*) sedemikian rupa sehingga elemen-elemen data dalam suatu kelompok memiliki kesamaan yang tinggi dan elemen-elemen data pada kelompok tersebut berbeda dengan elemen-elemen yang berada dalam kelompok lain.

Secara umum metode *clustering* dibedakan menjadi dua yaitu *clustering* klasik dan *fuzzy clustering*. Metode *clustering* klasik (atau disebut juga sebagai *hard clustering*) didasarkan pada teori himpunan klasik yang menunjukan apakah suatu objek merupakan anggota atau bukan anggota dari suatu *cluster*. *Clustering* klasik bertujuan untuk membagi atau mempartisi (*partitioning*) data ke dalam suatu kelompok (*cluster*) secara eksklusif. Artinya apabila suatu elemen data telah menjadi anggota dari satu *cluster*, maka elemen tersebut tidak mungkin menjadi anggota dari *cluster* yang lain.

Berbeda dengan *clustering* klasik yang mempartisi data ke dalam suatu *cluster* secara eksklusif, metode *fuzzy clustering* memungkinkan suatu objek menjadi anggota dari beberapa cluster secara bersamaan dengan derajat keanggotaan yang berbeda. Setiap objek dalam suatu *cluster* tidak dibatasi secara tegas menjadi anggota *cluster* tersebut melainkan ditentukan oleh derajat keanggotaan yaitu antara 0 sampai dengan 1. Derajat keanggotaan tersebut yang akan mengindikasikan keberadaan suatu objek pada suatu *cluster*, dimana semakin besar derajat keanggotaan suatu objek dalam suatu *cluster*, maka semakin dekat objek tersebut dengan pusat *cluster*nya. Hal ini berarti suatu objek akan cenderung menjadi anggota suatu *cluster* yang memiliki derajat keanggotaan yang paling besar.

## 2.5. *Stemming*

Proses *Stemming* digunakan untuk mengubah *term* yang masih melekat dalam *term* tersebut awalan, sisipan, dan akhiran. Proses *Stemming* dilakukan dengan cara menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi

dari awalan dan akhiran) pada kata turunan, *Stemming* digunakan sebagai pengganti bentuk dari sebuah kata menjadi kata dasar dari kata tersebut yang sesuai dengan struktur *morfologi* bahasa Indonesia yang benar (Tala, 2003).

### 2.3.1. *Stemming* Nazief dan Andriani

*Stemming* adalah bagian yang tidak terpisahkan dalam *Information Retrieval* (IR). Algoritma Nazief dan Adriani sebagai algoritma *Stemming* untuk teks berbahasa Indonesia yang memiliki kemampuan prosentase keakuratan (*presisi*) lebih baik dari algoritma lainnya. Algoritma *Stemming* Nazief dan Andriani sangat berguna dan menentukan di proses IR terutama dalam pengolahan text bahasa Indonesia.

Algoritma dari Nazief dan Adriani yang disusun oleh Bobby Nazief dan Mirna Adriani ini mempunyai tahap-tahap sebagai diantaranya adalah:

1. Langkah pertama masukkan kata yang akan di proses kemudian kata tersebut akan dibandingkan dengan kata yang sudah ada di dalam *database*. Jika ditemukan maka diasumsikan kata adalah *root word*. Maka proses berhenti.
2. Setelah itu dilakukan proses *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. kemudian Jika berupa *particles* (“-lah”, “-kah”, “-tah” atau “-pun”) maka proses ini diulangi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”).
3. Langkah selanjutnya adalah mengapus *Derivation Suffixes* (“-i”, “-an” atau “-kan”). Jika kata ditemukan didalam kamus, maka proses akan berhenti sampai disini. Jika tidak maka akan dilanjutkan ke langkah 3a.
    a. Jika “-an” sudah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga dihapus. Jika kata tersebut ditemukan di kamus maka proses akan berhenti. Jika tidak ditemukan maka berlanjut ke 3b.
    b. Langkah selanjutnya adalah menghapus Akhiran (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.
4. Setelah itu adalah langkah menghapus *Derivation Prefix*. Jika pada langkah 3 ada *sufiks* yang dihapus maka dilanjutkan ke langkah 4a, tapi jika tidak ada berlanjut ke langkah 4b.

a. Periksa kata kombinasi awalan dan akhiran yang tidak diperbolehkan. Jika ditemukan maka proses berhenti, jika belum maka berlanjut ke langkah 4b.
b. For i = 1 to 3, tentukan jenis awalan kemudian hapus awalan. Jika *root word* masih belum juga ditemukan maka lakukan proses 5, jika sudah maka proses berhenti.

5. Langkah selanjutnya adalah Melakukan *Recoding*.
6. Jika semua proses telah selesai tetapi masih belum juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

Algoritma diatas mempunyai beberapa keterbatasan, Untuk mengatasi keterbatasan tersebut maka perlu untuk ditambahkan aturan-aturan dibawah ini :

1. Aturan untuk *reduplikasi*.
   a. Jika ada dua kata atau lebih yang dihubungkan oleh kata penghubung adalah mempunyai kata yang sama maka *root word* adalah bentuk kata dasarnya, contoh : "buku-buku" *root word*-nya adalah "buku".
   b. Kata lain, misalnya "bolak-balik", "berbalas-balasan, dan "seolah-olah". Untuk mendapatkan *root word*-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama maka diubah menjadi bentuk tunggal, contoh: kata "berbalas-balasan", "berbalas" dan "balasan" memiliki *root word* yang sama yaitu "balas", maka *root word* "berbalas-balasan" adalah "balas". Tapi beda halnya dengan kata "bolak-balik", "bolak" dan "balik" memiliki *root word* yang tidak sama, maka *root word*-nya adalah "bolak-balik".
2. Tambahkan di awal dan di akhir dengan aturan yang sudah ada. Untuk tipe yang berawalan "mem-", dan kata yang berawalan "memp".

### 2.6. Pembobotan *Term*

Pembobotan kata sangat berpengaruh dalam menentukan kemiripan antara dokumen dengan *query*. Apabila bobot tiap kata dapat ditentukan dengan tepat, diharapkan hasil perhitungan kemiripan teks akan menghasilkan perangkingan dokumen yang baik.

### 2.4.1 *Term* Frequency (tf)

Pembobotan lokal yang paling banyak dipakai adalah menggunakan *term frequency* (tf). Faktor ini lah yang menyatakan makin banyak munculnya suatu kata didalam dokumen. Semakin seringnya suatu kata tersebut muncul didalam dokumen, berarti semakin penting kata di dalam dokumen tersebut. Ada empat cara yang basa digunakan untuk mendapatkan nilai TF:

1. Raw Tf

   Nilai Tf sebuah *term* dihitung berdasarkan kemunculan *term* tersebut dalam dokumen.

2. Binary Tf

   Cara ini hanya akan bernilai 0 apabila *term* tidak ada pada sebuah dokumen, dan bernilai 1 apabila *term* tersebut ada dalam dokumen. Sehingga banyaknya kemunculan *term* pada sebuah dokumen tidak berpengaruh.

3. *Logarithmic* Tf

   Dalam memperoleh nilai Tf, cara ini menggunakan fungsi *logaritmik* dalam matematika.

   $Tf$ = 1+log ($Tf$) ..... (2.1)

4. *Augmented* Tf

   $$\text{tf} = 0.5 + 0.5 \times \frac{tf}{\max(tf)} \quad \text{..... (2.2)}$$

   a. Nilai Tf adalah jumlah kemunculan *term* pada sebuah dokumen
   b. Nilai max(Tf) adalah jumlah kemunculan terbanyak *term* pada dokumen yang sama.

## 2.7. *Naïve Bayes*

Menurut Eko Prasetyo (2012) Bayes merupakan teknik memprediksi berbasis *probabilistik* sederhana yang berdasar pada penerapan teorima bayes (atau aturan bayes) dengan asumsi *independence* (ketidaktergantungan) yang kuat (naif). Prediksi bayes didasarkan pada teorima bayes dengan formula umum

Menurut Bustami (2013) Naive Bayes merupakan sebuah pengklasifikasian *probabilistik* sederhana yang menghitung sekumpulan probabilitas dengan

menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Algoritma mengunakan teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas. Definisi lain mengatakan Naive Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya

$$P(H|X) = \frac{p(X|H)p(H)}{P(X)} \quad \text{..... (2.5)}$$

Keterangan :

1. X adalah data sampel dengan klas (label) yang belum diketahui.
2. H merupakan hipotesa X adalah data dengan kelas (label).
3. P(H) adalah peluang dari hipotesa H.
4. P(X) adalah peluang data sampel yang diamati.
5. P(X|H) adalah peluang data sampel X, bila diasumsikan bahwa hipotesa benar (valid).

Metode Naive Bayes ini perlu diketahui bahwa untuk proses klasifikasinya sangat membutuhkan beberapa petunjuk sebagai penentu dari kelas mana yang cocok untuk contoh yang sedang dianalisis tersebut. Karena itu, metode ini disesuaikan sebagai berikut:

$$P(C|F1..Fn) = \frac{P(C)P(F1..Fn|C)}{P(F1...Fn)} \quad \text{.....(2.6)}$$

## 2.8. *Recall dan Precision*

Menurut Purwono (2010) *Recall* (perolehan berhubungan dengan kemampuan sistem untuk memanggil dokumen yang relevan. sedangkan *precision* (ketepatan) adalah berkaitan dengan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan.

Menurut Kurniawan (2010) *Recall* adalah perbandingan jumlah dokumen relevan yang terambil sesuai dengan *query* yang diberikan dengan total kumpulan dokumen yang relevan dengan *query*. *Precision* adalah perbandingan jumlah

dokumen yang relevan terhadap *query* dengan jumlah dokumen yang terambil dari hasil pencarian. Precision dapat diartikan sebagai ketepatan atau kecocokan (antara permintaan informasi dengan jawaban terhadap permintaan itu). Sedangkan istilah *recall* dibidang sistem temu kembali informasi (information retrival) berkaitan dengan kemampuan menemukan kembali informasi yang sudah tersimpan (Pendit 2008).

**Tabel 2.1** Parameter menghitung precision dan recall

| **Keterangan** | **Relevan** | **Tidak Relevan** |
|---|---|---|
| Terambil | True positive (tp) | False positive (fp) |
| Tidak terambil | False negative (fn) | True negative (tn) |

Rumus dari Precision :

$$Precision = \frac{tp}{tp+fp} \quad \text{..... (3.1)}$$

Rumus dari Recall :

$$Recall = \frac{tp}{tp+ fn} \quad \text{..... (3.2)}$$

Rumus dari Accuracy :

$$Accuracy = \frac{tp+tn}{tp+fp+tn+fn} \quad \text{..... (3.3)}$$

Rumus untuk menghitung F-Measure :

$$F - measure = \frac{2\times precision\times recall}{precision+recall} \quad \text{..... (3.4)}$$

Nilai *precision*, *recall*, dan *accuracy* dinyatakan dalam persen. Semakin tinggi ketiga nilai tersebut menunjukkan semakin baiknya kinerja aplikasi. Evaluasi yang akan dilakukan dalam penelitian ini adalah menghitung nilai dari *precision*, *recall*, *accuracy* dan *f-measure* berdasarkan judul yang berhasil ditemukan kembali oleh sistem aplikasi yang dibuat. Sedangkan untuk menentukan nilai dari *precision*, *recall* dan *accuracy* harus didapatkan jumlah judul yang relevan terhadap suatu topik judul.

## 2.9. Penelitian Sebelumnya

1. PENERAPAN TEXT MINING DALAM KLASIFIKASI JUDUL SKRIPSI. Pada tahun 2016 Ahmad Fathan Hidayatullah. dari fakultas Teknik jurusan Teknik Informatika Universitas Islam Indonesia Yogyakarta telah melakukan penelitian Seminar Nasional Aplikasi Teknologi Informasi (SNATi), penelitian ini difungsikan untuk melakukan klasifikasi judul skripsi menggunakan metode Naive Bayes. Tingkat akurasinya mencapai 97%.
2. SISTEM DIAGNOSIS PENYAKIT HATI MENGGUNAKAN METODE NAÏVE BAYES. Pada tahun 2018 Novianto Donna Prayoga, Nurul Hidayat, Ratih Kartika Dewi. Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya. Telah melakukan penelitian dalam jurnal Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 8, Agustus 2018, hlm. 2666-2671. penelitian ini difungsikan untuk mendiaknosis penyakit hati menggunakan metode Naive Bayes. Tingkat akurasinya mencapai 87,5%.