BAB II
LANDASAN TEORI

2.1 Penelitian Sebelumnya

Penelitian tentang pencarian dokumen telah banyak dilakukan. Diantaranya adalah penelitian yang telah ditulis oleh Firnas Nadirman yang berjudul "Sistem Temu Kembali Informasi dengan Metode Vector Space Model pada Pencarian File Dokumen Berbasis Teks". Dalam penelitiannya mereka menggunakan algoritma Vector Space Model (VSM), pada metode ini dokumen hasil pencarian akan diurutkan berdasarkan bobot dari kata pencarian yang terdapat di dalam dokumen tersebut. Salah satu algoritma pembobotannya adalah algoritma tf-idf yang dipengaruhi oleh frekuensi kemunculan kata pada sebuah dokumen dan frekuensi dari dokumen yang memiliki kata kata tersebut. Sebelum melakukan pencarian dokumen akan memecah isi teks dari dokumen-dokumen tersebut menjadi indeks kata. Indeks ini yang akan digunakan untuk proses pencarian. Proses pembentukan indeks dari teks yang terdapat di dalam dokumen akan melalui beberapa tahapan yaitu parsing, penghilangan stopwords penghitungan bobot. Dan juga pada proses pencarian, query dari pengguna akan melalui proses yang hampir sama pada proses pembentukan indeks. Setelah itu akan dibentuk vector dokumen dan vector query untuk diolah sehingga akan mendapatkan bobot dari dokumen hasil pencarian. Dengan metode ini dapat dicari informasi dari dokumen yang disimpan secara cepat, serta dokumen dari hasil pencarian dapat diurutkan berdasarkan bobot informasinya.

Pencarian informasi berdasar kata kunci juga ditulis oleh Nadia Damayanti berjudul "Temu Kembali Informasi Berdasarkan Lokasi pada Dokumen yang Dikelompokkan Menggunakan Metode Centroid Linkage Hierarchical" dapat membantu pengguna ketika ingin mengetahui informasi yang berhubungan dengan kata kunci yang dicari. Begitu juga dengan pencarian kelompok dokumen yang memuat lokasi tertentu yang

sama dan hanya mengambil informasi yang mempunyai tingkat kepentingan tinggi. Metode single linkage hierarchical dapat digunakan untuk mengetahui kemiripan atau kedekatan judul proyek akhir sesuai dengan inputan. Semakin atas urutan/ranking dari output judul yang dihasilkan maka semakin mendekati dengan inputan.

Penelitian yang berjudul "Implementasi Metode Single Linkage untuk Menentukan Kinerja Agent pada Call Centre Berbasis Asterisk for Java" yang ditulis oleh Beni Ilham Priyambodo, membuat sebuah sistem *Call Centre* menggunakan *Asterisk* berbasis pemrograman *Java*. Dimana digunakan suatu program Java untuk melakukan monitoring terhadap kinerja Agent. Dari hasil monitoring didapat lalu dilakukan pengelompokan menggunakan dua metode yaitu perhitungan manual dan metode Single Linkage. Pengelompokan menual dilakukan berdasar standarisasi yang telah ditetapkan, sedangkan pengelompokan menggunakan metode Single Linkage mengacu pada jarak terdekat antar parameter Agent. Dari hasil pengujian didapatkan bahwa sistem Call Centre untuk menentukan kinerja Agent berbasis Asterisk for Java telah berjalan dengan baik. Waktu eksekusi paling lama yang dibutuhkan untuk menyimpan data monitoring yaitu parameter Login Time dengan rata-rata waktu 2.2 milisekon. Sedangkan untuk waktu eksekusi program clustering manual, semakin banyak jumlah Agent maka waktu eksekusi program semakin lama.

Sedangkan pada penelitian ini dilakukan penerapan sistem temu kembali informasi untuk melakukan pencarian dokumen yang relevan dan mirip pada obyek buku berbahasa Indonesia pada Jurusan Teknik Informatika di Perpustakaan Universitas Muhammadiyah Gresik.

2.2 **Pustaka**

2.2.1 Buku

Buku adalah kumpulan kertas atau bahan lainnya yang dijilid menjadi satu pada salah satu ujungnya dan berisi tulisan atau gambar. Setiap sisi dari sebuah lembaran kertas pada buku disebut sebuah halaman.

Seiring dengan perkembangan dalam bidang dunia informatika, kini dikenal pula istilah *e-book* atau buku-e (buku elektronik), yang mengandalkan perangkat seperti komputer meja, komputer jinjing, komputer tablet, telepon seluler dan lainnya, serta menggunakan perangkat lunak tertentu untuk membacanya. (Wikipedia).

2.2.2 Perpustakaan

Dalam arti tradisional, perpustakaan adalah sebuah koleksi buku dan majalah. Walaupun dapat diartikan sebagai koleksi pribadi perseorangan, namun perpustakaan lebih umum dikenal sebagai sebuah koleksi besar yang dibiayai dan dioperasikan oleh sebuah kota atau institusi, serta dimanfaatkan oleh masyarakat yang rata-rata tidak mampu membeli sekian banyak buku atas biaya sendiri.(Wikipedia)

Tetapi, dengan koleksi dan penemuan media baru selain buku untuk menyimpan informasi, banyak perpustakaan kini juga merupakan tempat penyimpanan dan/atau akses ke map, cetak atau hasil seni lainnya, mikrofilm, mikrofiche, tape audio, CD, LP, tape video dan DVD. Selain itu, perpustakaan juga menyediakan fasilitas umum untuk mengakses gudang data CD-ROM dan internet.

Perpustakaan dapat juga diartikan sebagai kumpulan informasi yang bersifat ilmu pengetahuan, hiburan, rekreasi, dan ibadah yang merupakan kebutuhan hakiki manusia.

Oleh karena itu perpustakaan modern telah didefinisikan kembali sebagai tempat untuk mengakses informasi dalam format apa pun, apakah informasi itu disimpan dalam gedung perpustakaan tersebut ataupun tidak. Dalam perpustakaan modern ini selain kumpulan buku tercetak, sebagian buku dan koleksinya ada dalam perpustakaan digital (dalam bentuk data yang bisa diakses lewat jaringan komputer).

2.3 Sistem Temu Kembali Informasi

"Sistem Temu Kembali Informasi (Information Retrieval) digunakan untuk menemukan kembali informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis." (Wikipedia)

Sejak diperkenalkan pertama kali pada tahun 1952 dan mulai diteliti tahun 1961, banyak para ahli memaparkan pengertian sistem temu kembali informasi. Seperti yang dikutip oleh Firnas(2006:9) Menurut Lancaster (1968) di dalam Rijsbergen (1979): "sebuah sistem temu-kembali informasi tidak memberitahu (yakni tidak mengubah pengetahuan) pengguna mengenai masalah yang ditanyakannya. Sistem tersebut hanya memberitahukan keberadaan (atau ketidakberadaan) dan keterangan dokumendokumen yang berhubungan dengan permintaannya".

Sistem temu kembali informasi merupakan kegiatan yang bertujuan untuk menyediakan dan memasok informasi bagi pemakai sebagai jawaban atas permintaan atau berdasarkan kebutuhan pemakai.

Hasugian (2006: 2) mengemukakan bahwa "pada dasarnya sistem temu kembali informasi adalah suatu proses untuk mengidentifikasi, kemudian memanggil (*retrieve*) suatu dokumen dari suatu simpanan (*file*), sebagai jawaban atas permintaan informasi".

Sedangkan menurut pendapat Tague-Sutcliffe yang dikutip oleh Hasugian (2006: 3) menyatakan bahwa, "tujuan utama sistem temu kembali informasi adalah untuk menemukan dokumen yang sesuai dengan

kebutuhan informasi pengguna secara efektif dan efisien, sehingga dapat memberikan kepuasan baginya".

Menurut uraian – uraian di atas dapat disimpulkan bahwa proses mencari dan menemu kembalikan dokumen secara efektif dan efisien berhubungan dengan subjek - subjek tertentu.

Sistem temu kembali informasi merupakan sistem yang mampu melakukan pencarian informasi pada kumpulan dokumen, pencarian dokumen itu sendiri, pencarian metadata untuk dokumen tersebut, atau pencarian teks, suara, gambar, atau data dalam basis data dan pengambilan dokumen yang relevan dari sebuah koleksi dokumen sesuai dengan *query* pengguna sistem. *Input* dari suatu sistem temu balik informasi adalah *query* dari pengguna dan koleksi dokumen atau artikel, dan *output*-nya adalah dokumen atau artikel yang dianggap relevan oleh sistem. Sistem temu balik informasi ini digunakan untuk mengurangi informasi yang terlalu banyak sehingga sulit untuk dikelola.

Sistem temu kembali informasi terdiri dari komponen-komponen yang saling berkaitan satu sama lain. Menurut Chowdury 1999 dalam Zaenab, 2002: 41 "Pada intinya dalam sistem temu balik informasi terdapat tiga komponen utama yang saling mempengaruhi, yaitu (1) kumpulan dokumen; (2) kebutuhan informasi pengguna, dan (3) proses pencocokan (*matching*) antara keduanya" Pernyataan yang sama juga diuraikan Hasibuan dalam Hasugian (2006: 3) bahwa "Secara garis besar komponen sistem temu balik informasi terdiri dari pemakai (*user*), dokumen, dan *matcher-machine*".

Adapun komponen-komponen sistem temu kembali informasi menurut Hasugian (2008: 14) antara lain, (1) Pengguna; (2) Query; (3) Dokumen; (4) Indeks Dokumen dan (5) Pencocokan/ *Matcher Fungtion.*

1. Pengguna

Pengguna sistem temu kembali informasi adalah orang yang menggunakan atau memanfaatkan sistem temu kembali informasi dalam rangka kegiatan pengelolaan dan pencarian informasi.

Berdasarkan perannya, pengguna sistem temu kembali informasi dibedakakan atas 2 (dua) kelompok yaitu pengguna (*user*) dan pengguna akhir (*end user*).

2. **Query**

   Query adalah format bahasa permintaan yang di *input* (dimasukan) oleh pengguna kedalam sistem temu kembali informasi. Dalam *interface* (antar muka) sistem temu kembali informasi selalu disediakan kolom/ruas sebagai tempat bagi pengguna untuk mengetikkan (menuliskan) *query* nya.

3. **Dokumen**

   Dokumen adalah istilah yang digunakan utnuk seluruh bahan pustaka, apakah itu artikel, buku, laporan penelitian dsb. Seluruh bahan pustaka dapat disebut sebagai dokumen.

4. **Indeks Dokumen**

   Indeks adalah daftar istilah atau kata (*list of terms*). Dokumen yang dimasukkan/disimpan dalam *database* diwakili oleh indeks, Indeks itu disebut indeks dokumen.

5. **Pencocokan (Matcher Fungtion)**

   Pencocokkan istilah (*query*) yang dimasukkan oleh pengguna dengan indeks dokumen yang tersimpan dalam *database* adalah dilakukan oleh mesin komputer. Komputerlah yang melakukan proses pencocokkan itu dalam waktu yang sangat singkat sesuai dengan kecepatan *memory* dan *processing* yang dimiliki oleh komputer itu.

Dari beberapa uraian di atas disimpulkan bahwa sistem temu balik informasi memiliki komponen-komponen penyusun yang paling sedikit terdiri dari tiga bagian yaitu dokumen, pencari informasi dan proses pencocokan atau penghubung antara dokumen dan pencari informasi. Dan lebih rincinya sistem temu balik informasi terdiri atas lima komponen yaitu pengguna, query, dokumen, indeks dokumen dan pencocokan.

Proses Sistem Temu Kembali Informasi

Proses Sistem Temu Kembali Informasi. Tahapan proses sistem temu kembali informasi adalah sebagai berikut:

1. Text Operations (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam query maupun dokumen (term selection) dalam pentransformasian dokumen atau query menjadi terms index (indeks dari kata-kata).
2. Query formulation (formulasi terhadap query) yaitu memberi bobot pada indeks kata-kata query.
3. Ranking (perangkingan), mencari dokumen-dokumen yang relevan terhadap query dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan query.
4. Indexing (pengindeksan), membangun basis data indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

2.3.1 Indexing

Terdapat 5 tahap dalam proses indexing :

1. **Case Folding**

   Dalam tahap ini kalimat yang dmasukkan akan diubah menjadi huruf kecil. Hanya pada huruf ‘a’ sampai ‘z’ yang hanya diterima karakter selain huruf (angka ataupun simbol lain) akan dihilangkan dan dianggap sebagai delimeter.

2. **Pemisahan rangkaian kata (Tokenization)**

   Tokenization adalah tugas memisahkan deretan kata didalam kalimat, paragraph atau halaman menjadi token atau potongan kata tunggal atau termed word. Tahap ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token menjadi huruf kecil (lower case).

3. **Penyaringan (Filtration)**

Penyaringan atau Filtration merupakan pengambilan ‘kata’ penting dari hasil token, menggunakan algoritma stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting).

‘Stoplist’ atau ‘Wordlist’ adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag of word. Kata yang tidak penting adalah hasil parsing dicek dengan kamus (kumpulan kata) stopword. Jika parsing ada yang sama dengan stopword maka akan dibuang atau dihapus. Strategi umum penentuan stop-list adalah mengurutkan term berdasarkan frekuensi koleksi (jumlah total kemunculan setiap term di dalam koleksi dokumen) dan memasukkan term yang paling sering muncul sebagai stop-word (Manning dan Hinrich, 2008).

4. Konversi term ke bentuk akar (Stemming)

Stemming adalah pengembalian kata dasar dari kata yang telah mengalami imbuhan, sisipan dan awalan.

Kumpulan kata hasil proses sebelumnya yaitu filtering dicek 1 per 1 apakah kata tersebut termasuk dalam kata yang disimpan di dalam database atau tidak. Jika kata tersebut terdapat dalam kamus bahasa Indonesia maka kata tersebut termasuk kata dasar tetapi jika tidak terdapat didalam kamus kata yang akan dilakukan pemotongan imbuhan atau akhiran berdasarkan algoritma yang telah di tentukan.

A. Stemming Bahasa Indonesia

Stemming merupakan suatu proses untuk menemukan kata dasar dari sebuah kata. Dengan 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 untuk mengganti bentuk dari suatu kata menjadi kata dasar dari kata tersebut yang

sesuai dengan struktur morfologi Bahasa Indonesia yang baik dan benar.

Imbuhan (affixes) pada Bahasa Indonesia lebih kompleks bila dibandingkan dengan imbuhan (affixes) pada Bahasa Inggris. Karena seperti yang telah disebutkan di atas bahwa imbuhan (affixes) pada Bahasa Indonesia terdiri dari awalan (prefixes), sisipan (infixes), akhiran (suffixes), bentuk perulangan (repeated forms) dan confixes (kombinasi dari awalan dan akhiran). Imbuhan-imbuhan yang melekat pada suatu kata harus dihilangkan untuk mengubah bentuk kata tersebut menjadi bentuk kata dasarnya.

Beberapa algoritma dasar dalam steming antara lain :

1. Brute force stemming. Algoritma ini yang paling sederhana. Bermodalkan database kata dengan kata dasarnya. Computer dengan mudah mencari kata dasar. Namun metode ini mempunyai kelemahan yaitu jumlah database kata dan kata dasarnya harus besar. Kesalahan terjadi jika kata tidak di temukan di database dan kemudian dianggap kata dasar, padahal bukan.
2. Menghilangkan imbuhan (awalan, akhiran, sisipan). Untuk menggunakan metode ini harus tahu terlebih dahulu aturan bahasa. Kata akan di potong imbuhannya berdasar aturan bahasanya. Kesalahan terjadi jika kata tersebut adalah kata dasar yang dipotong. Misalnya: perawan > awan.
3. Dan masih banyak algoritma-algoritma dasar lainnya, seperti gabungan algoritma di atas, skotastik,lematasi, dll.

Untuk bahasa Indonesia beberapa algoritma yang biasanya digunakan antara lain:

1. Porter Stemmer. Algoritma ini terkenal digunakan sebagai stemmer untuk bahasa Inggris. Porter Stemmer dalam

bahasa Indonesia akan menghasilkan keambiguan karena aturan morfologi bahasa Indonesia (Tala,2003).

2. Nazief & Adriani Stemmer. Algoritma ini paling sering di bicarakan dalam stemming bahasa Indonesia. Algoritma ini hasil penelitian internal UI (Universitas Indonesia) dan tidak di publish secara umum (Nazif, 1996). Algortitma ini merupakan gabungan antara algoritma menghilangkan imbuhan dan brute force stemming. Namun algoritma ini mempunyai dua masalah, yang pertama kemampuannya tergantung dari besarnya database kata dasar, dan yang kedua, hasil stemming tidak selalu optimal untuk aplikasi information retrieval (Tala, 2003).

Bila dibandingkan, untuk teks berbahasa Indonesia, Porter stemming lebih cepat prosesnya dari pada Nazief & Adriani memiliki tingkat keakuratan lebih tinggi daripada porter stemmer (Ledy, 2009).

- **Stemming Bahasa Indonesia Algoritma Nazief & Adriani**

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 bahasa tersebut juga berbeda. Proses *stemming* pada teks berbahasa Indonesia lebih rumit/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:

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

1. Pertama cari kata yang akan diistem dalam kamus kata dasar. Jika ditemukan maka diasumsikan kata adalah *root word*. Maka algoritma berhenti.
2. *Inflection Suffixes* (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika berupa *particles* (“-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 maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b.
   b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4
4. Hilangkan *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.

Tabel 2.1 Kombinasi Awalan Akhiran Yang Tidak Diijinkan

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

b. Identifikasikan tipe awalan dan hilangkan. Awalan ada tipe:
   - Standar: “di-”, “ke-”, “se-” yang dapat langsung dihilangkan dari kata.
   - Kompleks: “me-”, “be-”, “pe”, “te-” adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada Tabel 2.1 untuk mendapatkan pemenggalan yang tepat.

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

5. Apabila setelah langkah 4 kata dasar masih belum ditemukan, maka proses *recoding* dilakukan.

   Recoding dilakukan dengan menambahkan karakter *recoding* di awal kata yang dipenggal. Pada Tabel 2.1, karakter *recoding* adalah huruf kecil setelah tanda hubung (‘-’) dan terkadang berada sebelum tanda kurung. Sebagai contoh, kata “menangkap”, setelah dipenggal menjadi “nangkap”. Karena tidak valid, maka *recoding* dilakukan dan menghasilkan kata “tangkap”.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word*. Proses selesai.

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.
3. Jika dua karakter pertama bukan “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-” maka berhenti.

4. Jika tipe awalan adalah “none” maka berhenti. Jika tipe awalan adalah bukan “none” maka awalan dapat dilihat pada Tabel 2.2 Hapus awalan jika ditemukan.

Tabel 2.2 Cara Menentukan Tipe Awalan Untuk awalan “te-”

Following Characters				**Tipe Awalan**
Set 1	**Set 2**	**Set 3**	**Set 4**	
“-r-“	“-r-“	-	-	None
“-r-“	-	-	-	Ter-luluh
“-r-“	Not (vowel or “-r-“)	“-er-“	Vowel	Ter
“-r-“	Not (vowel or “-r-“)	“-er-“	Not vowel	Ter-
“-r-“	Not (vowel or “-r-“)	Not “-er-“	-	Ter
Not (vowel or “-r-“)	“-er-”	Vowel	-	None
Not (vowel or “-r-“)	“-er-“	Not vowel	-	None

Table 2.3 Jenis awalan berdasarkan tipe awalannya

Tipe Awalan	Awalan yang harus dihapus
di-	di-
ke-	ke-
se-	se-
te-	te-
ter-	ter-
ter-luluh	Ter

Untuk mengatasi keterbatasan pada algoritma di atas, maka ditambahkan aturan-aturan dibawah ini :

1. Aturan untuk reduplikasi.
    a. Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word* adalah bentuk tunggalnya, 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". Sebaliknya, pada kata "bolak-balik", "bolak" dan "balik" memiliki *root word* yang berbeda, maka *root word*-nya adalah "bolak-balik".
2. Tambahan bentuk awalan dan akhiran serta aturannya.
    a. Untuk tipe awalan "mem-", kata yang diawali dengan awalan "memp-" memiliki tipe awalan "mem-".
    b. Tipe awalan "meng-", kata yang diawali dengan awalan "mengk-" memiliki tipe awalan "meng-".

Berikut contoh-contoh aturan yang terdapat pada awalan sebagai pembentuk kata dasar :

1. **Awalan SE-**

    Se + semua konsonan dan vokal tetap tidak berubah Contoh :

    - Se + bungkus = sebungkus
    - Se + nasib = senasib
    - Se + arah = searah
    - Se + ekor = seekor

2. **Awalan ME-**

    Me + vokal (a,i,u,e,o) menjadi sengau "meng" Contoh :

- Me + inap = menginap
- Me + asuh = mengasuh
- Me + ubah = mengubah
- Me + ekor = mengekor
- Me + oplos = mengoplos

Me + konsonan b menjadi “mem” Contoh :

- Me + beri = member
- Me + besuk = membesuk

Me + konsonan s menjadi “meny” (luluh) Contoh :

- Me + sapu = menyapu
- Me + satu = menyatu

Me + konsonan t menjadi “men” (luluh) Contoh :

- Me + tanama = menanam
- Me + tukar = menukar

Me + konsonan (l,m,n,r,w) menjadi tetap “me” Contoh :

- Me + lempar = melempar
- Me + masak = memasak
- Me + naik = menaik
- Me + rawat = merawat
- Me + warna = mewarna

3. **Awalan KE-**

   Ke + semua konsonan dan vokal tetap tidak berubah Contoh **:**

   - Ke + bawa = kebawa
   - Ke + atas = keatas

4. **Awalan PE-**

   Pe + konsonan (h,g,k) dan vokal menjadi “per” Contoh :

   - Pe + hitung + an = perhitungan
   - Pe + gelar + an = pergelaran
   - Pe + kantor + = perkantoran

Pe + konsonan “t” menjadi “pen” (luluh) Contoh :

- Pe + tukar = penukar
- Pe + tikam = penikam

Pe + konsonan (j,d,c,z) menjadi “pen” Contoh :

- Pe + jahit = penjahit
- Pe + didik = pendidik
- Pe + cuci = pencuci
- Pe + zina = penzina

Pe + konsonan (b,f,v) menjadi “pem” Contoh :

- Pe + beri = pemberi
- Pe + bunuh = pembunuh

Pe + konsonan “p” menjadi “pem” (luluh) Contoh :

- Pe + piker = pemikir
- Pe + potong = pemotong

Pe + konsonan “s” menjadi “peny” (luluh) Contoh :

- Pe + siram = penyiram
- Pe + sabar = penyabar

Pe + konsonan (l,m,n,r,w,y) tetap tidak berubah Contoh :

- Pe + lamar = pelamar
- Pe + makan = pemakan
- Pe + nanti = penanti
- Pe + wangi = pewangi

Kelebihan dan Kelemahan Algoritma Nazief dan Adriani

- Kelebihan :
  1. Memperhatikan kemungkinan adanya partikel-partikel yang mungkin mengikuti suatu kata berimbuhan.
  2. Proses stemming dokumen teks berBahasa Indonesia menggunakan Algoritma Nazief dan Adriani memiliki prosentase keakuratan (presisi) lebih besar dibandingkan dengan stemming menggunakan Algoritma Port1er.

- Kelemahan :
  1. Penyamarataan makna variasi kata.
  2. Jumlah database kata dan kata dasarnya harus besar. Kesalahan terjadi bila kata tidak ditemukan di database dan kemudian dianggap kata dasar, padahal bukan.
  3. Lamanya waktu yang diperlukan dalam proses pencarian kata di dalam kamus.

5. Pembobotan TF – IDF

Pembobotan global digunakan untuk memberikan tekanan terhadap term yang mengakibatkan perbedaan dan berdasarkan pada penyebaran dari term tertentu di seluruh dokumen. Banyak skema didasarkan pada pertimbangan bahwa semakin jarang suatu term muncul di dalam total koleksi maka term tersebut menjadi semakin berbeda.

Pendekatan terhadap pembobotan global mencakup inverse document frequency (idf), squared idf, probabilistic idf, GF-idf, entopy.Pendekatan idf merupakan pembobotan yang paling banyak digunakan saat ini. Beberapa aplikasi tidak melibatkan bobot global, hanya memperhatikan tf, yaitu ketika tf sangat kecil atau saat diperlukan penekanan terhadap frekuensi term di dalam suatu dokumen.(Polettini.2004).

Bobot local suatu term I di dalam dokumen j (tf_{ij}) dapat didefinisikan sebagai :

$$tf_{ij} = \frac{f_{ij}}{Max(f_{ij})} \qquad (2.1)$$

Dimana f_{ij} adalah jumlah berapa kali term I muncul di dalam dokumen j. frekuensi tersebut dinormalisasi dengan frekuensi dari most common term di dalam dokumen tersebut. Atau dapat dilakukan tanpa normalisasi, sehingga nilai tf merupakan jumlah kemunculan term i dalam dokumen j.

$$tf_{ij} = f_{ij} \quad (2.2)$$

Bobot global dari suatu term i pada pendekatan inverse document frequency (idf_i) dapat didefinisikan sebagai

$$Idf_i = \log \frac{(D)}{df_i}$$

Normalisasi untuk nilai Idf adalah sebgai berikut:

$$Idf_i = \log \frac{(D)}{df_i} + 1 \quad (2.3)$$

Dimana df_i adalah frekuensi dokumen dari term i dan sama dengan jumlah dokumen yang mengandung term i. Log_2digunakan untuk memperkecil pengaruhnyarelatif terhadap $tf_{ij.}$

Bobot dari term i (w_{ij}) dihitung menggunakan ukuran tf.idf yang didefinisikan sebagai berikut :

$$W_{ij} = tf_{ij} \times idf_i \quad (2.4)$$

Dimana : i = dokumen ke-i

j = kata ke-j dari kata kunci

w= bobot dokumen ke-i terhadap kata ke-j

TF-IDF akan menghasilkan nilai bobot dari term term dalam dokumen yang telah di masukkan yang akan dirangking dan menjadi acuan untuk pemilihan kata penting dalam dokumen.

2.4 Single Linkage Hierarchical Method

Metode ini dimulai dengan setiap objek dinyatakan sebagai kluster tersendiri. Kedekatan (jarak) antar kluster dihitung dan kluster yang paling terdekat digabungkan. Kedekatan pada kluster baru dihitung ulang dan kluster paling dekat digabung lagi. Proses tersebut dilakukan secara berulang sampai seluruh data (objek) menjadi kluster.

Ada berbagai metode yang digunakan untuk menghitung kedekatan (jarak) antar dua kluster, salah satunya yaitu Metode Singel Linkage (metode tetangga terdekat).

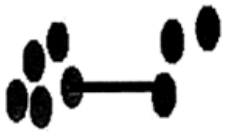
Metode Singe Linkage (metode tetangga terdekat) merupakan cara perhitungan jarak antara dua kluster dipilih dari jarak terdekat dari semua pasangan data dalam 2 kluster.

Algoritma Single Linkage Hierarchical Method pada Clustering dapat dilakukan dengan langkah-langkah sebagai berikut :

1. Diasumsikan setiap data dianggap sebagai cluster. Jika n=jumlah data dan c=jumlah cluster, berarti ada c=n.
2. Menghitung jarak antar cluster dengan Euclidian distance.
3. Mencari 2 cluster yang mempunyai jarak antar cluster yang paling minimal dan digabungkan (merge) kedalam cluster baru (sehingga c=c-1).
4. Kembali ke langkah 3, dan diulangi sampai dicapai cluster yang diinginkan.

Penghitungan jarak antar obyek, maupun antar clusternya dilakukan dengan Euclidian distance, khususnya untuk data numerik. Untuk data 2 dimensi, digunakan persamaan sebagai berikut :

$$d(x,y) = \sqrt{\sum_{i=1}^{n} |Xi - yi|^2} \qquad (2.5)$$



Gambar 2.1 single – lingkage cluster

Contoh Permasalahan :

Perhatikan data berikut:

Data	X	Y
1	1	1
2	4	1

3	1	2
4	3	4
5	5	4

Kelompokan data set tersebut dengan menggunakan metode Single Linkage.

$$D(x,y) = \sqrt{|x2 - x1|^2} + |y2 - y1|^2$$

- Langkah 1 :

  $D(1,1) = \sqrt{|1-1|^2} + |1-1|^2 = 0 + 0 = 0$

  $D(1,2) = \sqrt{|4-1|^2} + |1-1|^2 = 3 + 0 = 3$

  $D(1,3) = \sqrt{|1-1|^2} + |2-1|^2 = 0 + 1 = 1$

  $D(1,4) = \sqrt{|3-1|^2} + |4-1|^2 = 2 + 3 = 5$

  $D(1,5) = \sqrt{|5-1|^2} + |4-1|^2 = 4 + 3 = 7$

  $D(2,2) = \sqrt{|4-4|^2} + |1-1|^2 = 0 + 0 = 0$

  $D(2,3) = \sqrt{|1-4|^2} + |2-1|^2 = 3 + 1 = 4$

  $D(2,4) = \sqrt{|3-4|^2} + |4-1|^2 = 1 + 3 = 4$

  $D(2,5) = \sqrt{|5-4|^2} + |4-1|^2 = 1 + 3 = 4$

  $D(3,3) = \sqrt{|1-1|^2} + |2-2|^2 = 0 + 0 = 0$

  $D(3,4) = \sqrt{|3-1|^2} + |4-2|^2 = 2 + 2 = 4$

  $D(3,5) = \sqrt{|5-1|^2} + |4-2|^2 = 4 + 2 = 6$

  $D(4,4) = \sqrt{|3-3|^2} + |4-4|^2 = 0 + 0 = 0$

  $D(4,5) = \sqrt{|5-3|^2} + |4-4|^2 = 2 + 0 = 2$

  $D(5,5) = \sqrt{|5-5|^2} + |4-4|^2 = 0 + 0 = 0$

  Hasil matrik jarak :

D	1	2	3	4	5
1	0	3	1	5	7

2	3	0	4	4	4
3	1	4	0	4	6
4	5	4	4	0	2
5	7	4	6	2	0

Terpilih kelompok 1 dan 3 , sehingga kelompok ini digabungkan $D_{(1,3)} = 1$

Dengan menghapus baris-baris dan kolom-kolom matrik jarak yang bersesuaian dengan kelompok 1 dan 3, serta menambahkan baris dan kolom untuk kelompok (13).

- Langkah 2

  $D_{(1,3)2} = \min\{12 , 32\} = \min\{3 , 4\} = 3$

  $D_{(1,3)4} = \min\{14 , 34\} = \min\{5 , 4\} = 4$

  $D_{(1,3)5} = \min\{15 , 35\} = \min\{7 , 6\} = 6$

D	1	2	3	4	5
1	0	3	1	5	7
2	3	0	4	4	4
3	1	4	0	4	6
4	5	4	4	0	2
5	7	4	6	2	0

D	(13)	2	4	5
(13)	0	3	4	6
2	3	0	4	4
4	4	4	0	2
5	5	6	4	0

Dipilih jarak 2 kelompok terkecil → min D = min $D_{(45)} = 2$

- Langkah 3

  Menghitung jarak antar kelompok (4 dan 5) dengan kelompok lain yang tersisa, yaitu (13) dan 2.

  $D_{(45)(13)} = \min\{D_{41}, D_{43}, D_{51}, D_{53}\} = \min\{5, 4, 7, 6\} = 6$

  $D_{(45)2} = \min\{D_{42}, D_{52}\} = \min\{4, 4\} = 4$

D	(13)	2	4	5
(13)	0	3	4	6
2	3	0	4	4
4	4	4	0	2
5	5	6	4	0

D	(45)	(13)	2
(45)	0	4	4
(13)	4	0	3
2	4	3	0

Dipilih jarak terkecil → min $D_{(13)2} = 3$

- Langkah 4

  Menghitung jarak antar kelompok ((13) dan 2) dengan kelompok lain yang tersisa, yaitu (45).

  $D_{(123)(45)} = \min\{D_{14}, D_{15}, D_{34}, D_{35}, D_{24}, D_{25}\} = \{5, 7, 4, 6, 4, 4\} = 4$

Langkah 1

D	1	2	3	4	5
1	0	3	1	5	7
2	3	0	4	4	4
3	1	4	0	4	6
4	5	4	4	0	2
5	7	4	6	2	0

Langkah 2

D	(13)	2	4	5
(13)	0	3	4	6
2	3	0	4	4
4	4	4	0	2
5	5	6	4	0

Langkah 3

D	(45)	(13)	2
(45)	0	4	4
(13)	4	0	3
2	4	3	0

Langkah 4

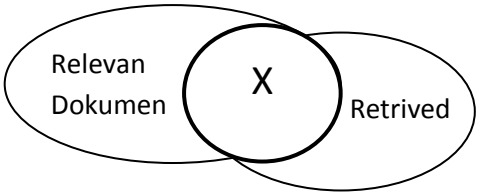
D	(132)	(45)
(132)	0	4
(45)	4	0

Jadi kelompok (132) dan (45) digabung untuk menjadi kelompok terdekat dari lima data, yaitu kelompok (13245) dengan jarak terdekat 4.

2.5 Evaluasi

Ada beberapa metode untuk mengukur evaluasi kinerja sistem temu kembali informasi, diantaranya yaitu recall dan Precission. Recall adalah rasio antara dokumen relevan yang berhasil ditemu kembalikan (diretrived) dari seluruh dokumen yang ada didalam sistem, sedangkan precision adalah rasio dokumen relevan yang berhasil ditemu kembalikan dari seluruh dokumen yang berhasil ditemu kembalikan. (Grossman, 2002).

Beberapa alasan yang berbeda mengapa tahap evaluasi IR (Information Retrival) adalah sesuatu yang penting. Sebagai contoh, penyedia sumber informasi membutuhkan informasi tentang penggunaan sumber daya oleh user, dan organisasi yang bekerja untuk meningkatkan kinerja pencarian perlu metode-metode yang efektif untuk mengevaluasi perubahan yang dilakukan untuk algoritma dan user interface. Dengan demikian, tujuan evaluasi adalah untuk menghasilkan perbaikan pada proses pengambilan informasi. Disisi lain, tujuan evaluasi biasanya tergantung pada penelitian, beberapa peneliti mungkin berpendapat bahwa tujuan utama dari evaluasi adalah untuk mengevaluasi kekuatan metodologi pengindeksan dan pencarian, namun beberapa focus lain dari penelitian evaluasi information retrival adalah proses kognitif, pengguna, antar muka manusia-komputer dan karakteristik database. Terdapat dua kategori dokumen yang dihasilkan oleh sistem IR terkait dengan pemrosesan query, yaitu relevan judul buku (ayat yang relevan dengan query) dan retrieved document (judul buku yang diterima pengguna). Hubungan antara dua kategori ini digambarkan menggunakan diagram Venn pada gambar 2.2



Gambar 2.2 Relasi antara relevant dan retrived dokumen.

Nilai precission, recall dan akurasi dapat dihitung dengan menggunakan table ketergantungan (Tabel 2.4) (Manning, 2008: 155)

Tablel 2.4 table Ketergantungan

	Relevant	nonRelevant
Retrived	True Positive (tp)	False Positive (fp)
Non retrived	False Negative (fn)	True Negative (tn)

Rumus menentukan precission :

$$\text{Precission} = \frac{tp}{(tp+fp)} \quad \ldots\ldots. (2.6)$$

Rumus menentukan nilai Recall :

$$\text{Recall} = \frac{tp}{(tp+fn)} \quad \ldots\ldots\ldots\ldots (2.7)$$

Rumus menentukan nilai Akurasi :

$$\text{Akurasi} = \frac{tp+tn}{tp+fp+tn+fn} \quad \ldots\ldots\ldots (2.8)$$

Precision adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem.

Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.

Accuracy didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai actual.

Nilai Recall, Precission dan akurasi dinyatakan dalam persen. Semakin tinggi nilai presentase ketiga nilai tersebut menunjukkan semakin baiknya kinerja sistem. Evaluasi yang akan dilakukan dalam penelitian ini adalah menghitung precission, recall dan akurasi berdasarkan nilai threshold dari

proses Single Lingkage Hierarchical Method yang dilakukan secara berulang.

Sedangkan untuk menentukan nilai recall, precision dan akurasi harus didapatkan jumlah dokumen yang relevan terhadap suatu topic informasi. Satu-satunya cara untuk mendapatkannya yaitu dengan membaca dokumen itu satu persatu.

Menurut Rijsbergen (1979) relevansi merupakan sesuatu yang sifatnya subjektif. Setiap rang mempunyai perbedaan untuk mengartikan sesuatu dokumen tersebut relevan terhadap sebuah topic informasi.

Menurut Mizzaro (1998), evaluasi pada sebuah sistem temu-kembali informasi dengan menggunakan recall dan precission sudah cukup baik untuk menjadi ukuran dari sistem tersebut.