Sistem Temu Balik Informasi (TEORI DANPRAKTEK)

DISUSUN OLEH:
Anna Hendri Soleliza Jones S.Kom., M.Cs
Arfiani Nur Khusna S.T., M.Kom.

UNIVERSITAS AHMAD DAHLAN
PRODI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INDUST

KATA PENGANTAR

Kata Pengantar

Puji dan syukur kita ucapkan kepada Allah SWT yang telah memberi rahmat dan hidayah-Nya sehingga penyusunan revisi petunjuk praktikum Sistem Temu Balik Informasi akhirnya bisa diselesaikan. Petunjuk praktikum ini disusun sebagai panduan pelaksanaan kegiatan praktikum Sistem Temu Balik Informasi di lingkungan Program Studi Teknik Informatika Universitas Ahmad Dahlan.

Materi yang disajikan sudah diurutkan disesuaikan dengan RPM Kurikulum baru, sehingga insya Allah mahasiswa dapat dengan mudah memahami. Pada setiap pertemuan diberikan penjelasan tentang teori terkait materi yang diberikan dan langkah praktikum berisi tahapan kegiatan yang harus dilakukan mahasiswa/praktikan pada saat praktikum.

Ucapan terimakasih untuk TIM penyusun yang telah membantu dalam penyelesaikan petunjuk praktikum ini. Penulis menyadari masih banyak ketidaksempurnaan pada penulisan ini, baik isi maupun redaksinya, oleh karenanya kritik dan saran yang membangun diharapkan dapat memperbaiki untuk tahun-tahun berikutnya.

Terima kasih kepada semua pihak yang telah membantu baik secara langsung ataupun tidak terhadap terselesaikannya petunjuk praktikum ini.

Agustus, 2019

Tim Penyusun

DAFTAR ISI

Kata Pengantar

Modul 1. Import Data……

Modul 2. Perhitungan Bobot……

Modul 3 Membangun Program Kemiripan Query ……

Modul 4 TF IDF ……

Modul 5 Vector Space Model ……

Modul 6 Rochio……

Modul 7 Recall & Precission……

PRAKTIKUM 1
IMPORT DATA

Waktu :1.5 jam

Kompetensi Dasar:

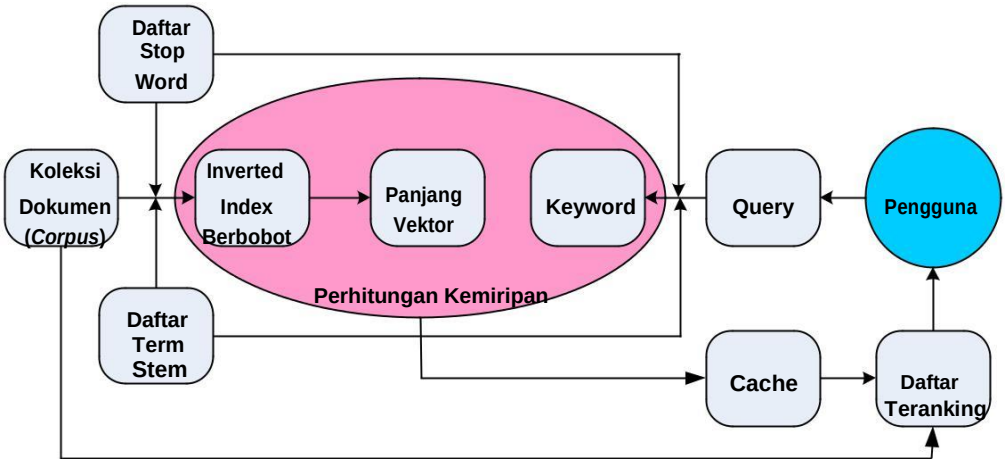
Setelah mengikuti praktikum ini, mahasiswa mampu memahami konsep system temu balik informasi

Indikator: Setelah mengikuti praktikum ini, mahasiswa dapat melakukan import data dengan baik.

A. Import data dari file excel

Saat memulai susatu proyek tentang data *science*, kemungkinan besar kita akan sering mengambil data melalui *web scrapping*, dan tidak menutup kemungkinan juga data tersebut dapat diunduh dalam format file excel dalam extensi *.csv*.

Modul ini memperlihatkan langkah-langkah membangun sebuah sistem temu-balik informasi (STBI) atau *information retrieval system* sederhana berbasis web. Sistem ini mengkoleksi beberapa berita. Berita-berita ini akan diberikan kepada pengguna sesuai dengan query yang dimasukkan. Sistem akan melakukan perhitungan kemiripan antara query dengan daftar berita yang tersedia.



Gambar 1. Arsitektur Sistem IR dari Contoh Aplikasi

B. Langkah praktikum
Ikuti langkah praktikum berikut ini:

```
//=============== koleksi fungsi ===================
//fungsi untuk melakukan preprocessing terhadap teks
//terutama stopword removal dan stemming
//-------------------------------------------------------------------------

function preproses($teks) {

	//bersihkan tanda baca, ganti dengan space
	$teks = str_replace("'", " ", $teks);
	$teks = str_replace("-", " ", $teks);
	$teks = str_replace(")", " ", $teks);
	$teks = str_replace("(", " ", $teks);
	$teks = str_replace("\"", " ", $teks);
	$teks = str_replace("/", " ", $teks);
	$teks = str_replace("=", " ", $teks);
	$teks = str_replace(".", " ", $teks);
	$teks = str_replace(",", " ", $teks);
	$teks = str_replace(":", " ", $teks);
	$teks = str_replace(";", " ", $teks);
	$teks = str_replace("!", " ", $teks);
	$teks = str_replace("?", " ", $teks);

	//ubah ke huruf kecil
	$teks = strtolower(trim($teks));

	//terapkan stop word removal
	$astoplist = array ("yang", "juga", "dari", "dia", "kami",
			"kamu", "ini", "itu", "atau", "dan", "tersebut",
			"pada", "dengan", "adalah", "yaitu", "ke");

	foreach ($astoplist as $i => $value) {
		$teks = str_replace($astoplist[$i], "", $teks);
	}

	//terapkan stemming
	//buka tabel tbstem dan bandingkan dengan berita
	$restem = mysql_query("SELECT * FROM tbstem ORDER BY Id");

	while($rowstem = mysql_fetch_array($restem)) {
	     $teks = str_replace($rowstem['Term'],
	             $rowstem['Stem'], $teks);
	}

	//kembalikan teks yang telah dipreproses
	$teks = strtolower(trim($teks));
	return $teks;
} //end function preproses
//-------------------------------------------------------------------------
```

```
//---------------------------------------------------------------------------
//fungsi untuk membuat index
function buatindex() {
       //hapus index sebelumnya
       mysql_query("TRUNCATE TABLE tbindex");

       //ambil semua berita (teks)
       $resBerita = mysql_query("SELECT * FROM tbberita ORDER BY
       Id"); $num_rows = mysql_num_rows($resBerita);
       print("Mengindeks sebanyak " . $num_rows . " berita. <br />");

       while($row = mysql_fetch_array($resBerita))
              { $docId = $row['Id'];
              $berita = $row['Berita'];

              //terapkan preprocessing
              $berita = preproses($berita);

              //simpan ke inverted index (tbindex)
              $aberita = explode(" ", trim($berita));

              foreach ($aberita as $j => $value) {
                 //hanya jika Term tidak null, tidak kosong

                 if ($aberita[$j] != "") {

                     //berapa baris hasil yang dikembalikan
                     //query tersebut?
                     $rescount = mysql_query("SELECT Count FROM tbindex
                    WHERE Term = '$aberita[$j]' AND DocId = $docId");

                     $num_rows = mysql_num_rows($rescount);

                     //jika sudah ada DocId dan Term tersebut      ,
                     //naikkan Count (+1)
                     if ($num_rows > 0) {
                        $rowcount = mysql_fetch_array($rescount);

                        $count = $rowcount['Count'];
                        $count++;

                        mysql_query("UPDATE tbindex SET Count =
                                   $count WHERE Term = '$aberita[$j]'
                                   AND DocId = $docId");
                     }

                     //jika belum ada, langsung simpan ke tbindex
                     else {
                        mysql_query("INSERT INTO tbindex (Term,
                                   DocId, Count) VALUES
                                   ('$aberita[$j]', $docId, 1)");
                     }
                 } //end if
              } //end foreach
       } //end while
} //end function buatindex()
//---------------------------------------------------------------------------
```

C. POSTTEST

1. Jelaskan pengertian STBI !

PRAKTIKUM 2
HITUNG BOBOT

Waktu :1.5 jam

Kompetensi Dasar:

Setelah mengikuti praktikum ini, mahasiswa mampu memahami konsep pembobotan dalam system temu balik informasi

Indikator:

Setelah mengikuti praktikum ini, mahasiswa dapat melakukan penghitungan bobot terhadap term yang diberikan dengan baik.

A. Langkah Praktikum

1. Lanjutkan Tahapan membangun sistem temukembali informasi dengan source code sebagai berikut:

```
//---------------------------------------------------------------------
//fungsi hitungbobot, menggunakan pendekatan tf.idf
function hitungbobot() {
	//berapa jumlah DocId total?, n
	$resn = mysql_query("SELECT DISTINCT DocId FROM
	tbindex"); $n = mysql_num_rows($resn);
	//ambil setiap record dalam tabel tbindex
	//hitung bobot untuk setiap Term dalam setiap DocId
	$resBobot = mysql_query("SELECT * FROM tbindex ORDER BY Id");
	$num_rows = mysql_num_rows($resBobot); print("Terdapat " .
	$num_rows .
		" Term yang diberikan bobot. <br />");

	while($rowbobot = mysql_fetch_array($resBobot)) {
		//$w = tf * log (n/N)
		$term = $rowbobot['Term'];
		$tf = $rowbobot['Count'];
		$id = $rowbobot['Id'];

		//berapa jumlah dokumen yang mengandung term itu?, N
		$resNTerm = mysql_query("SELECT Count(*) as N
				FROM tbindex WHERE Term = '$term'");
		$rowNTerm = mysql_fetch_array($resNTerm);
		$NTerm = $rowNTerm['N'];
		$w = $tf * log($n/$NTerm);

		//update bobot dari term tersebut
		$resUpdateBobot = mysql_query("UPDATE tbindex
				SET Bobot = $w WHERE Id = $id");
	} //end while $rowbobot
} //end function hitungbobot
//---------------------------------------------------------------------

//---------------------------------------------------------------------
//fungsi panjangvektor, jarak euclidean
//akar(penjumlahan kuadrat dari bobot setiap
Term) function panjangvektor() {
	//hapus isi tabel tbvektor
	mysql_query("TRUNCATE TABLE tbvektor");
```

```
        //ambil setiap DocId dalam tbindex
        //hitung panjang vektor untuk setiap DocId tersebut
        //simpan ke dalam tabel tbvektor
        $resDocId = mysql_query("SELECT DISTINCT DocId FROM tbindex");


        $num_rows = mysql_num_rows($resDocId);




        print("Terdapat " . $num_rows .
              " dokumen yang dihitung panjang vektornya. <br />");

        while($rowDocId = mysql_fetch_array($resDocId)) {
               $docId = $rowDocId['DocId'];
               $resVektor = mysql_query("SELECT Bobot
                            FROM tbindex WHERE DocId = $docId");

               //jumlahkan semua bobot kuadrat
               $panjangVektor = 0;
               while($rowVektor = mysql_fetch_array($resVektor)) {
                    $panjangVektor = $panjangVektor + $rowVektor['Bobot']
                                     A. $rowVektor['Bobot'];
               }

               //hitung akarnya

$panjangVektor = sqrt($panjangVektor);

               //masukkan ke dalam tbvektor
               $resInsertVektor = mysql_query("INSERT INTO tbvektor (DocId,
                                  Panjang) VALUES ($docId, $panjangVektor)");
        } //end while $rowDocId }
   //end function panjangvektor
   //---------------------------------------------------------------------------
```

B. Posttest

1. Buatlah flowchart perhitungan TF-IDF
2. Hitunglah bobot dokumen dari permasalahan yang ada dengan kata kunci yang telah ditentukam !

PRAKTIKUM 3
MEMBANGUN PROGRAM KEMIRIPAN QUERY

Waktu :1.5 jam

Kompetensi Dasar:

Setelah mengikuti praktikum ini, mahasiswa mampu memahami konsep query dalam system temu balik informasi

Indikator:

Setelah mengikuti praktikum ini, mahasiswa dapat membangun program kemiripan query dengan baik.

A. Langkah Praktikum

Kode untuk membangun koneksi (koneksi.php) ke database MySQL dbstbi adalah

```
//membangun koneksi ke database
$con = mysql_connect("localhost","root","abc123");

if (!$con)  {
       die('Koneksi ke database gagal: ' . mysql_error());
}

mysql_select_db("dbstbi", $con);
```

Kode program untuk tugas-tugas di atas diwakili oleh fungsi hitungsim($query) untuk menghitung tingkat kemiripan antara query dengan setiap daftar dokumen dan menyimpan nilai kemiripannya ke dalam tabel cache, dan fungsi ambilcache($keyword) untuk mengambil daftar dokumen yang sesuai dengan keyword tertentu. Berikut ini adalah kode program dari dua fungsi tersebut:

```
//---------------------------------------------------------------------------
- //fungsi hitungsim - kemiripan antara query
//setiap dokumen dalam database (berdasarkan bobot di tbindex)
function hitungsim($query) {
       //ambil jumlah total dokumen yang telah diindex
       //(tbindex atau tbvektor), n
       $resn = mysql_query("SELECT Count(*) as n FROM
       tbvektor"); $rown = mysql_fetch_array($resn); $n =
       $rown['n'];
       //terapkan preprocessing terhadap $query
       $aquery = explode(" ", $query);

       //hitung panjang vektor query
       $panjangQuery = 0;
       $aBobotQuery = array();

       for ($i=0; $i<count($aquery); $i++) {
              //hitung bobot untuk term ke-i pada query, log(n/N);
              //hitung jumlah dokumen yang mengandung term tersebut
              $resNTerm = mysql_query("SELECT Count(*) as N FROM tbindex
                           WHERE Term = '$aquery[$i]'");
              $rowNTerm = mysql_fetch_array($resNTerm);
              $NTerm = $rowNTerm['N'];

              $idf = log($n/$NTerm);
```

```
            //simpan di array
            $aBobotQuery[] = $idf;

            $panjangQuery = $panjangQuery + $idf * $idf;
        }


        $panjangQuery = sqrt($panjangQuery);
        $jumlahmirip = 0;

        //ambil setiap term dari DocId, bandingkan dengan Query
        $resDocId = mysql_query("SELECT * FROM tbvektor ORDER BY
        DocId"); while ($rowDocId = mysql_fetch_array($resDocId)) {
            $dotproduct = 0;

            $docId = $rowDocId['DocId'];
            $panjangDocId = $rowDocId['Panjang'];

            $resTerm = mysql_query("SELECT * FROM
                        tbindex WHERE DocId = $docId");
            while ($rowTerm = mysql_fetch_array($resTerm))
                  { for ($i=0; $i<count($aquery); $i++) {
                        //jika term sama
                        if ($rowTerm['Term'] == $aquery[$i]) {
                              $dotproduct = $dotproduct +
                                    $rowTerm['Bobot']  *  $aBobotQuery[$i];

                        } //end if
                  } //end for $i
            } //end while ($rowTerm)

            if ($dotproduct > 0) {
                  $sim = $dotproduct / ($panjangQuery * $panjangDocId);

                  //simpan kemiripan > 0  ke dalam tbcache
                  $resInsertCache = mysql_query("INSERT INTO tbcache
                  (Query, DocId, Value) VALUES ('$query', $docId, $sim)");


                  $jumlahmirip++;
            }

        } //end while $rowDocId

        if ($jumlahmirip == 0) {
              $resInsertCache = mysql_query("INSERT INTO tbcache
                              (Query, DocId, Value) VALUES ('$query', 0, 0)");
        }

    } //end hitungSim()
    //---------------------------------------------------------------------------

    //---------------------------------------------------------------------------
    function ambilcache($keyword) {
        $resCache = mysql_query("SELECT *  FROM tbcache
                    WHERE Query = '$keyword' ORDER BY Value DESC");
        $num_rows = mysql_num_rows($resCache);
        if ($num_rows >0) {
              //tampilkan semua berita yang telah terurut
              while ($rowCache = mysql_fetch_array($resCache))
                    { $docId = $rowCache['DocId'];
                    $sim = $rowCache['Value'];
                    if ($docId != 0) {
                          //ambil berita dari tabel tbberita, tampilkan
                          $resBerita = mysql_query("SELECT * FROM tbberita
                                        WHERE Id = $docId");
                          $rowBerita = mysql_fetch_array($resBerita);
```

```
                        $judul = $rowBerita['Judul'];
                        $berita = $rowBerita['Berita'];

                        print($docId . ". (" . $sim .
                              ") <font color=blue><b>"
                              . $judul . "</b></font><br />");
                        print($berita . "<hr />");
                  \} } else {
                        \} print("<b>Tidak ada... </b><hr />");
                  \} }
            \} }//end while (rowCache = mysql_fetch_array($resCache))
      \} }//end if $num_rows>0
      \} else {
            \} hitungsim($keyword);
\}
            \} //pasti telah ada dalam tbcache
            \}                                $resCache = mysql_query("SELECT
            *                                 FROM tbcache
                        \}   WHERE Query = '$keyword' ORDER BY Value
            DESC"); $num_rows = mysql_num_rows($resCache);
\}
            \}    while ($rowCache =
                  mysql_fetch_array($resCache)) { $docId =
                  $rowCache['DocId'];
                  \} $sim = $rowCache['Value'];
\}
                  \} if ($docId != 0) {
                        \} //ambil berita dari tabel tbberita, tampilkan
                       \}$resBerita = mysql_query("SELECT * FROM tbberita
                                   \} WHERE Id = $docId");
                        \} $rowBerita = mysql_fetch_array($resBerita);
\}
                        \} $judul = $rowBerita['Judul'];
                        \} $berita = $rowBerita['Berita'];
\}
                        \} print($docId . ". (" . $sim .

                              ") <font color=blue><b>"
                              . $judul . "</b></font><br />");
                        print($berita . "<hr />");
                  } else {
                        print("<b>Tidak ada... </b><hr />");
                  }
            } //end while
      }
} //end function ambilcache
//---------------------------------------------------------------------------
```

Keenam fungsi yang telah diuraikan di atas disimpan di dalam file fungsi.php. Berikut ini adalah kode program dari file index.php untuk menampilkan antarmuka retrieval kepada pengguna (juga antarmuka untuk membangun index dan memantau data yang dikelola oleh sistem IR):

```
<head>
<title>:: STBI - Indexing & Retrieval

::</title> </head>

<body>
<h1 align=center>STBI - Proses Indexing & Retrieval</h1>

<hr>

<div align=center>
| <a href="index.php">Beranda</a> |
<a href="index.php?act=corpus">Tampilkan Corpus</a>
| <a href="index.php?act=indexing">Buat Index</a> |
<a href="index.php?act=bobot">Hitung Bobot</a> |
<a href="index.php?act=panjangvektor">Hitung Panjang Vektor</a>
| <a href="index.php?act=showindex">Tampilkan Index</a> |
```

```
<a href="index.php?act=showvektor">Tampilkan Panjang Vektor</a>
| <a href="index.php?act=retrieve">Retrieval</a> |
<a href="index.php?act=cache">Tampilkan Cache</a> |
</div>
<hr />



<?php
include 'koneksi.php';
include 'fungsi.php';

//periksa apa yang diinginkan pengguna (variabel act)
$apa = $_GET["act"];

//jika "corpus"
if ($apa == "corpus") {
	$result = mysql_query("SELECT * FROM tbberita ORDER BY Id");

	while($row = mysql_fetch_array($result)) { echo
		$row['Id'] . ". <font color =blue>"
			. $row['Judul'] . "</font><br />" . $row['Berita'];
		echo "<hr />";
	}
}

//jika "indexing"
else if ($apa == "indexing") {
	buatindex();
	print("<hr />");
}

else if ($apa == "bobot") {
	hitungbobot();
	print("<hr />");
}

else if ($apa == "panjangvektor") {
	panjangvektor();
	print("<hr />");
}

else if ($apa == "showvektor") {
	print("<table>");
	print("<tr><td>Doc-ID</td><td>Panjang Vektor</td></tr>");
	$result = mysql_query("SELECT * FROM tbvektor");

	while($row = mysql_fetch_array($result)) {

		print("<tr>");
		print("<td>" . $row['DocId'] . "</td><td>"
			. $row['Panjang'] . "</td>");

		print("</tr>");

	}
	print("</table><hr />");
}

//jika "showindex"
 else if ($apa == "showindex") {

	print("<table>");
	print("<tr><td>#</td><td>Term</td><td>Doc-ID</td>
		<td>Count</td><td>Bobot</td></tr>");
	$result = mysql_query("SELECT * FROM tbindex ORDER BY Id");

	while($row = mysql_fetch_array($result)) {

		print("<tr>");
		print("<td>" . $row['Id'] . "</td><td>"
			. $row['Term'] . "</td><td>" . $row['DocId'] .
```

```
                        "</td><td>" . $row['Count']
                        . "</td><td>" . $row['Bobot'] . "</td>");
                print("</tr>");
        }
        print("</table><hr />");
    }

    //jika "retrieve"
    else if ($apa == "retrieve") {
        print('<center><form action="index.php?act=retrieve" method="post">
            Kata kunci: <input type="text" name="keyword" />
            <input name = "Cari!" type="submit" />
            </form></center><hr />');

        $keyword = $_POST["keyword"];

        if ($keyword)  {
            $keyword = preproses($keyword);

            print('Hasil retrieval untuk <font color=blue><b>' .
                $_POST["keyword"]  . '</b></font> (<font color=blue><b>'
                . $keyword . '</b></font>) adalah <hr
            />'); ambilcache($keyword);
        }
    } //end retrieve

//jika "cache"
    else if ($apa == "cache") {
        print("<table>");
        print("<tr><td>#</td><td>Query</td><td>Doc-ID</td>
            <td>Value</td></tr>");
        $result = mysql_query("SELECT * FROM tbcache ORDER BY Query ASC");

        while($row = mysql_fetch_array($result)) {

            print("<tr>");
            print("<td>" . $row['Id'] . "</td><td>"
                . $row['Query'] . "</td><td>" . $row['DocId'] .
                "</td><td>" . $row['Value'] . "</td>");
            print("</tr>");
        }

        print("</table><hr />");
    }

    //jika beranda atau tidak memilih apapun
    else {
        print("<p align=center>Pilih salah satu link di atas!</p><hr />");
    }
    ?>
    <h5 align=center>Dibuat oleh Husni, bagian dari matakuliah STBI, Teknik
    Informatika, Universitas Trunojoyo, 2010</h5> </body>
```

Berikut ini adalah daftar tabel yang digunakan dalam database dbstbi:

```
CREATE TABLE IF NOT EXISTS `tbberita` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `Judul` varchar(100) NOT NULL,
  `Berita` varchar(255) NOT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=17 ;

CREATE TABLE IF NOT EXISTS `tbcache` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `Query` varchar(100) NOT NULL,
  `DocId` int(11) NOT NULL,
  `Value` float NOT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=23 ;

CREATE TABLE IF NOT EXISTS `tbindex` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `Term` varchar(30) NOT NULL,
  `DocId` int(11) NOT NULL,
  `Count` int(11) NOT NULL,
  `Bobot` float NOT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=364 ;

CREATE TABLE IF NOT EXISTS `tbstem` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `Term` varchar(30) NOT NULL,
  `Stem` varchar(30) NOT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

CREATE TABLE IF NOT EXISTS `tbvektor` (
  `DocId` int(11) NOT NULL,
  `Panjang` float NOT NULL,
  PRIMARY KEY (`DocId`)
) ENGINE=MyISAM DEFAULT CHARSET=l
```

D. Post Test

1. Jelaskan Tahapan *retrieval* dimulai dengan mengambil query dari pengguna, menerapkan s*top word removal* dan *stemming* sehingga dihasilkan *keyword* dengan diagram alir

PRAKTIKUM 4
PENERAPAN TF-IDF

Waktu :1.5 jam

Kompetensi Dasar:

Setelah mengikuti praktikum ini, mahasiswa mampu memahami konsep *term frequency* dan *index frequency* dalam system temu balik informasi

Indikator:

Setelah mengikuti praktikum ini, mahasiswa dapat memahami penerapan dan perhitungan TF-IDF dalam STBI

A. Metode TF-IDF

TF-IDF merupakan salah satu metode pembobotan yang diunggulkan dalam beberpa penelitian. Tf-Idf adalah perhitungan yang menggambarkan seberapa pentingnya kata (term) dalam sebuah dokumen dan korpus. Proses ini digunakan untuk menilai bobot relevansi term dari sebuah dokumen terhadap seluruh dokumen dalam korpus. Termfrequency adalah ukuran seringnya kemunculan sebuah term dalam sebuah dokumen dan juga dalam seluruh dokumen di dalam korpus. Term frequency ini dihitung menggunakan persamaan (1) dengan term frequency ke-i dan adalah frekuensi kemunculan term ke-i dalam dokumen ke-j.

Sedangkan inverse document frequency adalah logaritma dari rasio jumlah seluruh dokumen dalam korpus dengan jumlah dokumen yang memiliki term yang dimaksud seperti yang dituliskan secara matematis pada persamaan (2). Nilai didapatkan dengan mengalikan keduanya yang diformulasikan pada persamaan (3).

Pembobotan kata dijabarkan oleh persamaan (2.1)

$$\boldsymbol{tf.idf}(\boldsymbol{t_i}, \boldsymbol{d_j}) = \boldsymbol{tf}(\boldsymbol{t_i}, \boldsymbol{d_j}) \times \log \frac{\mathbf{N}}{\mathbf{N}\,(\boldsymbol{t_i})} \qquad (2.1)$$

Keterangan :

Tf = Term Frequency

Idf = Index document frequency

N = Banyaknya Dokumen

N (t_i) = Banyaknya dokemen yang mengandung *term* t

B. Langkah Praktikum
   1. Berdasarkan system yang telah dibangun, cobalah untuk memasukan *keyword* dan lihat hasil pembobotan yang ditampilkan system
   2. Lakukan berulang dengan *keyword* berbeda
C. Posttest
   1. Lakukan perhitungan manual dari kata kunci yang ditentukan.
   2. Lakukan analisis dari hasil peritungan manual dengan hasil yang ditampilkan system

PRAKTIKUM 5
PENERAPAN *VECTOR SPACE MODEL*

Waktu :1.5 jam

Kompetensi Dasar:

Setelah mengikuti praktikum ini, mahasiswa mampu memahami konsep *VSM* dalam system temu balik informasi

Indikator:

Setelah mengikuti praktikum ini, mahasiswa dapat memahami penerapan dan perhitungan *Vector Spce Model* dalam STBI

A. Metode *Vector Space Model*

Vektor Space Model adalah model sistem temu balik informasi yang mengibaratkan masing-masing *query* dan dokumen sebagai sebuah vektor n-dimensi [10]. Tiap dimensi pada vektor tersebut diwakili oleh satu *term.Term* yang digunakan biasanya berpatokan kepada *term* yang ada pada *query* atau *keyword*, sehingga *term* yang ada pada dokumen tetapi tidak ada pada *query* biasanya diabaikan.

Model Ruang Vektor Dalam Information Retrieval

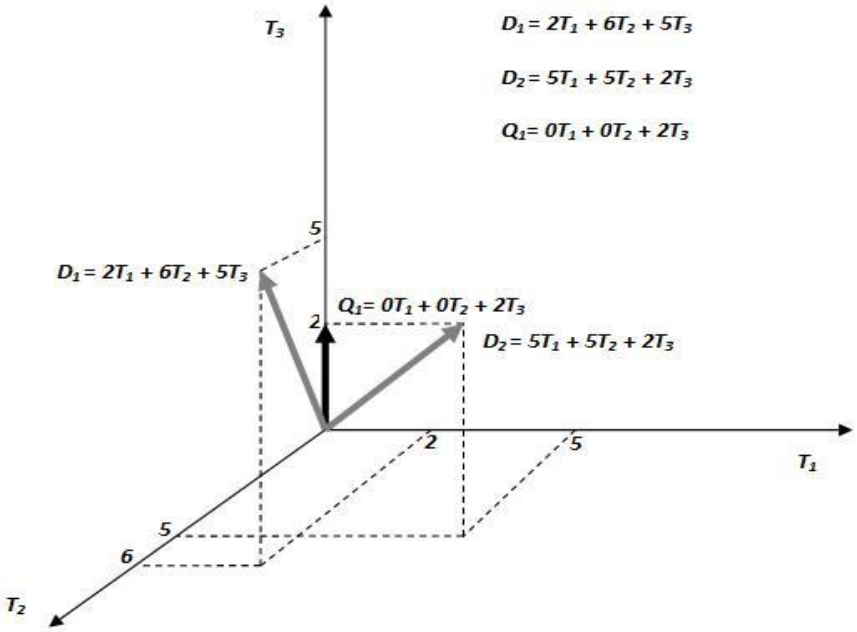
Pada *Information Retrieval System* terdapat beberapa metode yang digunakan dalam *Searching* salah satunya adalah dengan merepresentasikan proses *Searching* menggunakan Model Ruang Vektor. Model ruang vektor dibuat berdasarkan pemikiran bahwa isi dari dokumen ditentukan oleh kata-kata yang digunakan dalam dokumen tersebut. Model ini menentukan kemiripan (*similarity)* antara dokumen dengan *query* dengan cara merepresentasikan dokumen dan *query* masing-masing ke dalam bentuk

vektor. Tiap kata yang ditemukan pada dokumen dan *query* diberi bobot dan disimpan sebagai salah satu elemen vektor.

Kemiripan antar dokumen didefinisikan berdasarkan representasi *bag-of-words* dan dikonversi ke suatu model ruang vektor (*vector space model*, VSM). Model ini diperkenalkan oleh Salton [12] dan telah digunakan secara luas. Pada VSM, setiap dokumen di dalam database dan query pengguna direpresentasikan oleh suatu vektor multi-dimensi [1, 13]. Dimensi sesuai dengan jumlah term dalam dokumen yang terlibat Pada model ini:

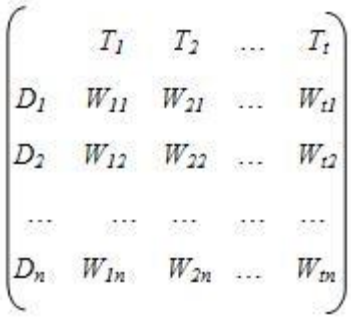
a. *Vocabulary* merupakan kumpulan semua *term* berbeda yang tersisa dari dokumen setelah *preprocessing* dan mengandung *t term index*. *Term-term* ini membentuk suatu ruang vektor.
b. Setiap *term i* di dalam dokumen atau *query j*, diberikan suatu bobot (*weight*) bernilai *real* W*ij*.
c. Dokumen dan *query* diekspresikan sebagai vektor *t* dimensi $d_j = (W_1, W_2, ..., Wtj)$ dan terdapat *n* dokumen di dalam koleksi, yaitu $j = 1, 2, ..., n$.

Contoh dari model ruang vektor tiga dimensi untuk dua dokumen D_1 dan D_2, satu query pengguna Q_1, dan tiga term T_1, T_2 dan T_3 diperlihatkan pada gambar 2.5.



Gambar 2.5 Contoh Model Ruang Vektor dengan dua dokumun D_1 dan D_2, serta *query* Q_1. [1]

Dalam model ruang vektor, koleksi dokumen direpresentasikan oleh matriks *term-document* (atau matriks *term-frequency*). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu term dalam dokumen yang ditentukan. Nilai nol berarti bahwa *term* tersebut tidak hadir di dalam dokumen. [1]



Gambar. Contoh matriks *term-document* untuk *database* dengan *n* document dan *t term*. [1]

B. Langkah Praktikum

1. Berdasarkan system yang telah dibangun, cobalah untuk memasukan *keyword* dan lihat hasil pembobotan yang ditampilkan system.
2. Lakukan berulang dengan *keyword* berbeda

C. Posttest

3. Lakukan perhitungan manual dari kata kunci yang ditentukan.
4. Lakukan analisis dari hasil peritungan manual dengan hasil yang ditampilkan system

PRAKTIKUM 6
ASOSIASI ROCHIO

Waktu :1.5 jam

Kompetensi Dasar:

Setelah mengikuti praktikum ini, mahasiswa mampu memahami konsep *asosiasi rochio* dalam system temu balik informasi

Indikator:

Setelah mengikuti praktikum ini, mahasiswa dapat memahami penerapan dan perhitungan *asosiasi Rocchiol* dalam STBI

A. Metode *Rochio*

Algoritma *Rocchio* merupakan yang mampu membandingkan setiap dokumen terhapad suatu *term* tertentu dalam klasifikasi kategori sesuai degan bobot dari *term* yang ada (Badriz,2008). Pada dasarnya *Rocchio* menggunakan Vector Spce Model, pada *Rocchio* proses klasifikasi di lakukan berdasarkan asumsi manusia (Mandowara, 2016). *Rocchio* memiliki 2 hal dalam pengerjaan yang harus di perhatikan yaitu sebagai berikut (Badriz,2008):

a. *Term* frequency and weighting

Berdasarkan frekuensi munculnya *term* yang sesuai dengan *query Term* yang sama akan dijumlah semua frekuensinya, akan tetapi metode ingin masih kurang bagus karena :
jika dokumen yang digunakan adalah dokumen yang lebih besar, maka dokumen tersebut memiliki *term* yang lebih banyak sehingga *score*-nya pun lebih besar.
Langkah-langkah dalam menggunakan metode ini adalah

1) Tiap-tiap dokumen dipecah menjadi *term-term.*
2) Kemudian *term* yang sudah ada diurutkan menjadi sebuah kamus di dalam sebuah kolom (catatan : jika ada beberapa *term* yang sama, maka hanya ditulis sekali).
3) Di sebelah kanan kolom *term*, tambahkan 2 kolom lagi. Kolom yang pertama untuk frekuensi *term* (tf).
4) Hitung jumlah *term* sama.

Untuk kolom yang kedua untuk kolom idft.

Rumus :

Idft=$log Ndft$.....1

Keterangan:

N : jumlah dokumen yang ada dalam *corpus*

dft : frekuensi dari sebuah *term*

5) Hitung jumlah *term* sama.

Tambahkan kolom lagi untuk bobot untuk menentukan dokumen yang paling sering muncul atau tidak dalam seuah *term*

Rumus :

$\beta=(tf)*(idft)$.....2

B. Langkah Praktikum
   1. Berdasarkan system yang telah dibangun, cobalah untuk memasukan *keyword* dan lihat hasil pembobotan yang ditampilkan system
   2. Lakukan berulang dengan *keyword* berbeda

C. Posttest
   1. Lakukan perhitungan manual dari kata kunci yang ditentukan.
   2. Lakukan analisis dari hasil peritungan manual dengan hasil yang ditampilkan system

PRAKTIKUM 7
PENGUJIAN RECALL AND PRECISION

Waktu :1.5 jam

Kompetensi Dasar:

Setelah mengikuti praktikum ini, mahasiswa mampu memahami konsep *recall and precision* dalam system temu balik informasi

Indikator:

Setelah mengikuti praktikum ini, mahasiswa dapat memahami penerapan dan perhitungan *recall and precision* dalam STBI

A. *Recall and precission*

Recall yaitu rasio antara jumlah dokumen relevan yang ditemukan dengan jumlah total dokumen relevan yang terdapat dalam basis data. Dari pendapat diatas dapat dinyatakan bahwa *recall* atau terpanggil adalah dokumen yang terpanggil dari sistem temu balik. informasi sesuai dengan permintaan pemakai yang mengikuti pola dari sistem tersebut. Nilai *recall* berkisar dari 0 s.d 1. Nilai *recall* makin besar belum cukup untuk menilai suatu sistem temu balik informasi apakah baik atau tidak.

Recall sebenarnya sulit diukur karena jumlah seluruh dokumen yang relevan dalam database sangat besar. Oleh karena itu presisi (*precision*) yang biasanya menjadi salah satu ukuran yang digunakan untuk menilai keefektivan suatu sistem temu balik informasi. *Precision* adalah jumlah kelompok dokumen relevan dari total jumlah dokumen yang ditemukan oleh system. Presisi juga merupakan cara mengukur tingkat efektivitas sistem temu balik informasi. Pengukuran tingkat ketepatan (*precision*) dalam kegiatan penelusuran.

Perhitungan *Recall* dan *Precision*

	Relevant	Not Relevant	Total
Retrieved	A	b	a+b
Not Retrieved	C	d	c+d
Total	a+c	b+d	a+b+c+d

Rumus untuk menghitung *Recall :*

$$R = \frac{a}{a+c}$$

Rumus untuk *Precission :*

$$P = \frac{a}{a+b}$$

Keterangan :

R = recall

P = Precision

a= Dokumen Relevan

b= Dokumen yang tidak relevan

c = Dokumen Relevan yang tidak ditemukan

d= Dokumen yang tidak terambil

B. Langkah Praktikum
   1. Memahami pengujian *recall and precission*

C. Posttest
   1. Lakukan perhitungan manual dari studi kasus yang terdapat pada sistem