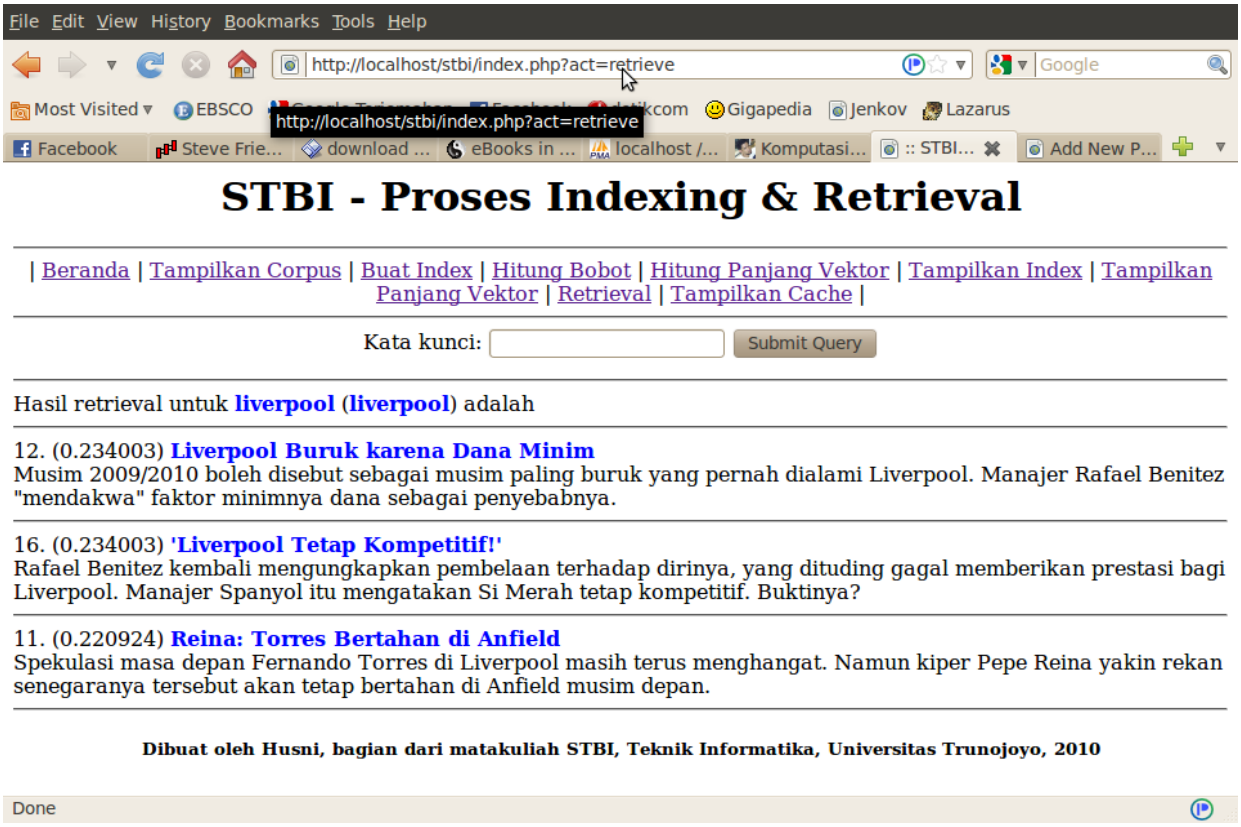
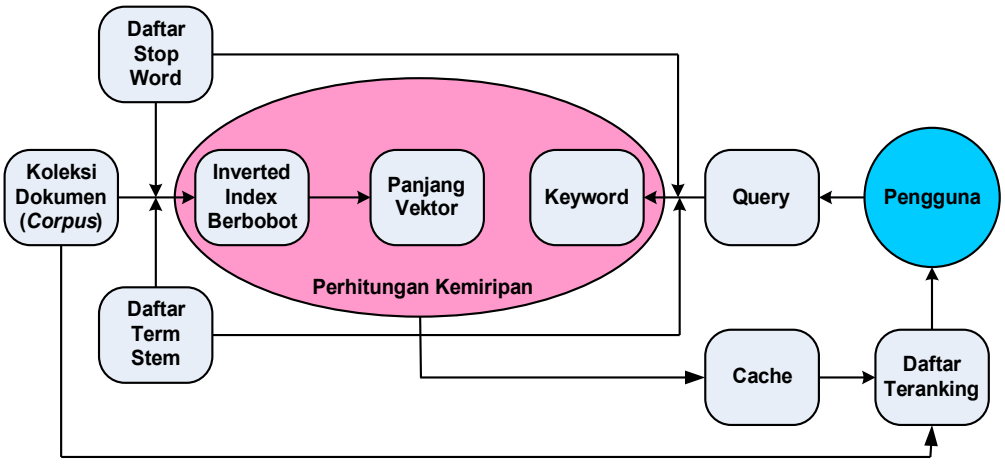
Sistem Temu-Balik Informasi

Sebuah Contoh Implementasi

Husni, husni@if.trunojoyo.ac.id, husni.trunojoyo.ac.id



Tulisan 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

Sistem ini terdiri dari 3 file PHP (index.php untuk menampilkan antarmuka kepada pengguna, fungsi.php menyimpan fungsi-fungsi yang diperlukan selama sistem IR bekerja dan koneksi.php untuk membangun koneksi ke database dbstbi pada server MySQL). Pada server MySQL terdapat beberapa tabel, yaitu: tbberita (menyimpan koleksi berita sebagaimana aslinya, *raw data*), tbstem (menyimpan daftar stem dari suatu term), tbindex (menyimpan inverted index dari setiap term dan bobot dari term tersebut), tbvektor (penyimpan panjang vektor dari setiap dokumen) dan tbcache (menyimpan nilai kemiripan antara suatu query dengan daftar dokumen dalam koleksi).

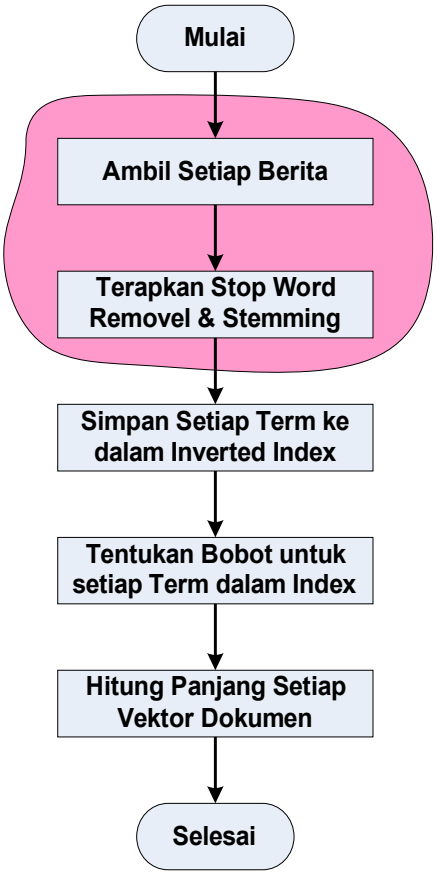
Arsitektur dasar dari IR yang dibangun diperlihatkan pada gambar 1.

Sistem IR secara umum terdiri dari dua tahapan besar, yaitu *indexing* yang dijalankan secara *offline* dan *retrieval* yang bekerja secara *online* (*real-time*). Setelah menghimpun dokumen (dalam hal ini berita) segera dilakukan *indexing*, yaitu membangun suatu daftar index (*inverted index*). Proses *indexing* mencakup penentuan *token*, *stop word removal* dan *stemming*. Kemudian dilanjutkan dengan menyimpan setiap term yang penting ke dalam suatu index (dalam kasus ini menggunakan suatu tabel pada database MySQL). Sistem ini berusaha memberikan kinerja yang baik karena itu proses pembobotan terhadap setiap term dan perhitungan panjang vektor dari setiap berita dilakukan secara *offline*, tepat setelah pembuatan index. Proses ini diperlihatkan pada gambar 2.

Pada tahapan *preprocessing*, yaitu penghapusan *stop word* dan *stemming*, akan dihasilkan daftar kata atau term yang lebih *compaq* tetapi tetap mewakili dokumen yang sedang diproses. Daftar *stop word* disimpan di dalam suatu array sedangkan daftar stem dari suatu term disimpan di dalam sebuah tabel. Agar proses stemming berjalan cepat, terutama pada kondisi tidak terdapat perubahan daftar stem, lebih baik daftar stem diletakkan di dalam array. Tutorial mengenai pemanfaatan array untuk stemming dapat dilihat di http://komputasi.wordpress.com.

Pembobotan untuk setiap term menggunakan skema *tf.idf* dimana *idf* adalah *log(n/N)* dimana N adalah jumlah dokumen yang mengandung term tertentu sedangkan n adalah jumlah total koleksi dokumen (*corpus*). Panjang dari vektor memanfaatkan pendekatan jarak *euclidean*, yaitu akar dari jumlah kuadrat sebuah term dari suatu dokumen.

Tahapan *indexing* di atas dikerjakan oleh 4 fungsi dalam kode program. Fungsi preproses() melakukan penghapusan *stop word* dan *stemming*. Fungsi buatindex() menyimpan setiap term hasil *stop word removal* dan *stemming* ke dalam tabel index. Fungsi hitungbobot() menentukan bobot dari setiap term yang terdapat di dalam tabel index. Fungsi panjangvektor() menghitung panjang vektor dari setiap dokumen yang diwakili oleh term-term dalam tabel index.



Gambar 2. *Preprocessing*, *indexing*, penentuan bobot term dan perhitungan panjang vektor dari setiap dokumen (berita)

Berikut ini adalah kode program dari 4 fungsi yang dijalankan secara *offline* tersebut:

```
//=============== 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()
//---------------------------------------------------------------------------

//---------------------------------------------------------------------------
//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']
                                    *  $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
//---------------------------------------------------------------------------
```

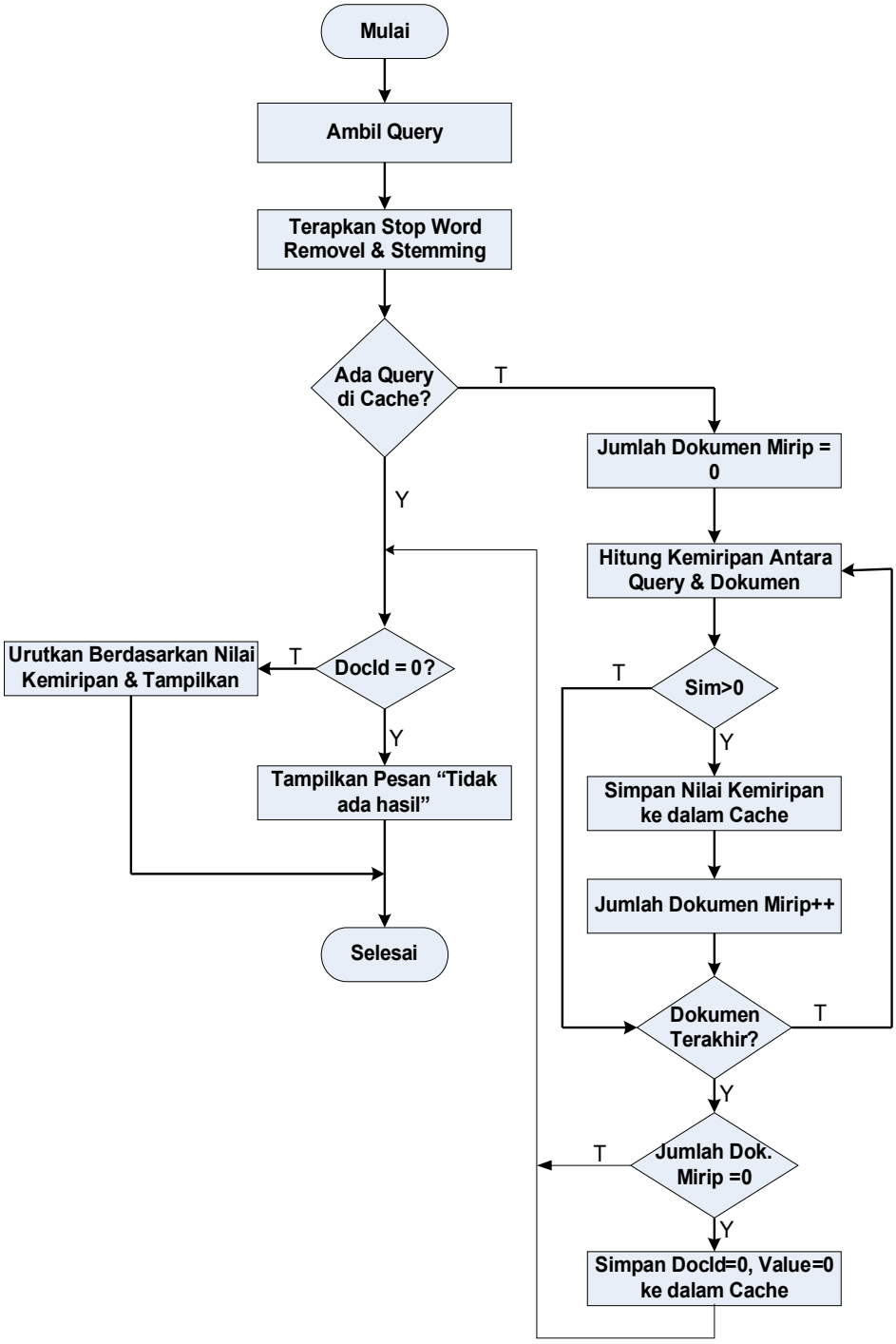
Kode untuk membangun koneksi (koneksi.php) ke database MySQL dbstbi adalah

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

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

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

Tahapan *retrieval* dimulai dengan mengambil query dari pengguna, menerapkan s*top word removal* dan *stemming* sehingga dihasilkan *keyword* yang *compaq* tetapi dapat mewakili query tersebut. Berdasarkan keyword ini, sistem melihat ke dalam cache. Jika di sana telah ada keyword tersebut maka sistem langsung mengurutkan dokumen-dokumen yang mirip dengan keyword dalam mengembalikannya kepada pengguna. Jika keyword belum terdapat di dalam cache, artinya belum pernah ada perhitungan kemiripan berdasarkan keyword tersebut, sistem menghitung kemiripan antara keyword dengan daftar dokumen yang diwakili oleh term-term di dalam index. Hasil perhitungan ini disimpan ke dalam tabel cache. Diagram alir dari tahapan ini diperlihatkan pada gambar 3.



Gambar 3. Diagram alir dari proses *retrieval* (*online*)

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=latin1;
```

Kode program dan contoh database, secara lengkap dapat didownload dari husni.trunojoyo.ac.id. Setelah download, buatlah sebuah database bernama dbstbi kemudian import isi dari file dbstbi-data.sql. Letakkan 3 file PHP yang didownload ke dalam direktori web, misalnya C:\XAMPP\htdocs (jika menggunakan XAMPP under Windows) atau /var/www jika menggunakan Apache2 pada Linux Ubuntu 10.04 LTS. Selanjutnya adalah memanggil file index.php melalui web browser, http://localhost/index.php (saya lebih suka meletakkan file-file PHP di atas dalam direktori stbi sehingga URL tersebut menjadi http://localhost/stbi/index.php).

Klik Tampilkan Corpus untuk melihat koleksi dokumen yang ada dalam database. Klik Buat Index untuk membuat Index. Kemudian klik Hitung Bobot dan Hitung Panjang Vektor. Daftar Index yang dihasilkan dapat dilihat dengan klik Tampilkan Index. Tampilkan Panjang Vektor digunakan untuk menampilkan daftar panjang vektor dari setiap dokumen. Klik link Retrieval untuk menampilkan antarmuka retrieval. Di sini pengguna dapat memasukkan query dan menerima daftar dokumen yang relevan dengan query tersebut. Link Tampilkan Cache dapat digunakan untuk melihat daftar cache (nilai kemiripan dokumen dengan query) berdasarkan query yang telah dikirimkan ke sistem.

Semoga tutorial singkat ini bermanfaat.