IMPLEMENTASI ALGORITMA DEPTH FIRST SEARCH PADA SISTEM PENCARIAN DOKUMEN

Amelia Yusnita[1)], Siti Lailiyah[2)], Twom Ali Panotogomo[3)]

Program Studi Teknik Informatika, STMIK Widya Cipta Dharma
Jl. M. Yamin No.25, Samarinda, 75123
E-mail : lia_ameliay@yahoo.co.id [1)] , lail.59a@gmail.com [2)], mase.dinda@gmail.com [3)]

ABSTRAK

Metode yang digunakan pada penelitian ini meliputi metode pengumpulan data yaitu penelitian lapangan dan penelitian kepustakaan, sedangkan metode pengembangan sistem yang digunakan dalam pengembangan sistem pakar ini adalah metode Waterfall yang terdiri dari Sistem *Enginering*, *Analysis*, *Design*, *Code*, *Testing*, dan *Maintenance*. Dengan. Dengan menggunakan alat bantu *flowchart* untuk membuat rancangan sistem yang akan dibangun.

Penelitian ini mengambil tempat di Dinas Perindustrian, Perdagangan, Koperasi dan UMKM Provinsi Kalimantan Timur, yang merupakan d pada provinsi Kaltim. Pencarian kembali dokumen yang relevan dengan kata kunci untuk mencari data/informasi yang dibutuhkan pengguna yang berhubungan langsung dengan dokumen-dokumen yang terdapat di *harddisk* atau media penyimpanan lain merupakan suatu hal yang menjadi perhatian dari pihak-pihak terkait yang membutuhkan data di dalam dokumen tersebut. Pencarian Kembali terhadat data di dalam dokumen menggunakan Algoritma *Depth First Search* menjadi fokus dalam penelitian .

Penelitian ini berfokus untuk menghasilkan aplikasi pencarian yang relevan terhadap kata kunci pencarian yang diharapkan peningkatan efisien waktu bagi pengguna. Alat bantu pengembangan sistem yang digunakan *Flowchart* dan *Waterfall*, dengan menggunakan bahasa pemrograman *Xojo.*

Kata Kunci: *Sistem, Pakar, Diagnosa Penyakit, Tanaman Padi, Fuzzy Tsukamoto*

1. PENDAHULUAN

Perkembangan teknologi dewasa ini khususnya internet berkembang sangat pesat. Hal ini diiringi juga dengan semakin berkembangnya Teknologi Informasi yang dibutuhkan oleh pengguna sehingga mengakibatkan munculnya suatu cabang ilmu baru dalam teknologi informasi, yaitu pencarian informasi (*information retrieval*).

Sejalan dengan perkembangan teknologi, maka perkembangan peralatan juga ikut mengambil andil yang besar untuk mendukung perkembangan pertukaran informasi yang semakin hari semakin canggih. Hampir semua kecanggihan teknologi yang memberikan kemudahan kepada *user* dalam menemukan apa yang diinginkannya. Salah satu kemudahan yang diberikan adalah menyediakan fasilitas *searching* yang digunakan untuk memberikan kemudahan mendapatkan lebih banyak informasi yang berbentuk dokumen.

Pencarian informasi yang tepat dan sesuai kebutuhan menjadi sangat penting dengan semakin mudahnya memperoleh informasi dari seluruh dunia sebagai akibat perkembangan teknologi informasi dan komunikasi yang semakin pesat. Oleh karena itu, teknik untuk memperoleh dokumen dengan isi yang sesuai dengan kebutuhan informasi sangat diperlukan.

Dalam hal ini penulis menggunakan sebuah cara/metode untuk melakukan pencarian kembali dokumen dengan menggunakan Algoritma Depth First Search

2. RUANG LINGKUP PENELITIAN

2.1. Rumusan Masalah

Sehubungan dengan latar belakang masalah di atas, maka rumusan masalah yang di kaji adalah "Bagaimana menghadirkan sebuah sistem pencarian dokumen yang dapat memberikan kemudahan untuk menghasilkan dokumen yang relevan dengan kata kunci menggunakan metode Algoritma Depth First Search?"

2.2 Batasan Masalah

Untuk mencegah perluasan masalah dan pembahasan yang terlalu kompleks, maka dibutuhkan pembatasan masalah. Adapun batasan masalah pada penulisan tugas akhir ini adalah:

1. *Searching* dokumen ini tidak membuat pencarian berdasarkan kombinasi kata-kata penggolongan AND, OR, dan NOT serta penggunaannya yang tidak menggunakan *case sensitive*
2. Dokumen yang digunakan sebagai data dalam proses pencarian ini berupa beberapa dokumen teks yaitu rtf, doc, pdf dan txt
3. Tidak membahas *semantic* kata
4. Bahasa Pemrograman yang digunakan ialah Xojo/Realbasic.
5. Jumlah kata merupakan penjumlahan kata yang

relevan dengan kata kunci di dalam dokumen

3. BAHAN DAN METODE

3.1 Information Retrieval

Menurut Bunyamin (2008*), Information Retrieval* (IR) adalah suatu sistem yang digunakan untuk menemukan kembali (*retrieve*) informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis.

Menurut Erwin (2005), salah satu aplikasi dari IR adalah mesin pencari yang dapat diterapkan di berbagai bidang. Pada mesin pencari dengan IR pengguna dapat memasukkan *query* yang bebas dalam arti kata *query* yang sesuai dengan bahasa manusia dan sistem dapat menemukan dokumen yang sesuai dengan *query* yang ditulis oleh user.

Model sistem IR menentukan detail sistem IR yaitu, meliputi:

1 Representasi dokumen dan *query.*
2 Fungsi pencarian.
3 Notasi kesesuaian (*relevance notation*) dokumen terhadap *query.*

Menurut Mandala (2003), yang dimaksud dengan representasi dokumen adalah kumpulan kalimat yang menyatu menjadi paragraf kemudian menjadi bab dan menjadi satu buku, atau disebut juga dengan kumpulan kata yang menyusun menjadi kalimat. Sedangkan yang dimaksud dengan fungsi pencarian adalah bagaimana mesin mengolah pertanyaan untuk dicocokkan dengan dokumen, lalu mengambil dokumen yang relevan.

Bagian ini terbagi menjadi beberapa bagian, yaitu:

1 Cara memilih kata (*term*) untuk indeks.
2 Cara mengindeks kata.
3 Cara membobot kata.

Bagian yang paling signifikan pengaruhnya adalah pembobotan kata. Cara pembobotan kata mencirikan bagaimana sebuah sistem temu kembali informasi di bangun. Notasi kesesuaian adalah hubungan yang terjadi antara pertanyaan dengan hasil pencarian. Sistem temu kembali informasi digunakan untuk menemukan kembali informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis.

Kerangka dari sistem temu-kembali informasi sederhana terbagi menjadi dua bagian. Bagian yang pertama adalah bagian si pencari informasi atau pengguna dari sistem. Pengguna dari sistem temu-kembali informasi harus menerjemahkan informasi yang dicarinya agar dapat diproses oleh sistem dengan cara memasukkan kata kunci. Kata kunci tersebut nanti diproses menjadi sebuah *query* yang dapat dimengerti oleh komputer. Bagian yang kedua adalah bagian dari dokumen. Pada bagian ini dokumen-dokumen direpresentasikan dalam bentuk indeks. Nanti *query* dari pengguna akan diproses melalui fungsi kesamaan untuk membandingkan *query* dengan indeks dari dokumen untuk mendapatkan dokumen yang relevan.

Menurut Baeza-Yates dan Rierio-Neto yang dikutip oleh Firnas (2006) sistem temu-kembali informasi berbeda dengan sistem temu-kembali data. Sistem temu-kembali data tujuan utamanya untuk menentukan dokumen yang memiliki kata kunci yang sesuai dengan *query* yang diberikan oleh pengguna di dalam sekumpulan dokumen.

Gambar 2.1 Bagian-Bagian Sistem Temu Kembali Informasi (*Information Retrieval*)
Sumber : Mandala, 2003. *Query Expansion using HeterogeneousThesauri*. International Journal of Information Processing and Management

Menurut Mandala (2003), pada gambar 2.1 memperlihatkan bahwa terdapat dua buah alur operasi pada sistem temu kembali informasi. Alur pertama dimulai dari koleksi dokumen dan alur kedua dimulai dari *query* pengguna. Alur pertama yaitu pemrosesan terhadap koleksi dokumen menjadi pangkalan data indeks dan tidak tergantung pada alur kedua. Sedangkan alur kedua tergantung dari keberadaan pangkalan data indeks yang dihasilkan pada alur pertama .

Bagian-bagian dari sistem IR menurut gambar 2.1 meliputi:

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

Sistem Temu Kembali Informasi menerima *query* dari pengguna, kemudian melakukan perangkingan terhadap dokumen pada koleksi berdasarkan kesesuaiannya dengan *query*. Hasil perangkingan yang diberikan kepada pengguna merupakan dokumen yang menurut sistem relevan dengan *query*. Namun relevansi dokumen terhadap suatu query merupakan penilaian pengguna yang subjektif dan dipengaruhi banyak faktor seperti topik, pewaktuan, sumber informasi maupun tujuan pengguna.

Menurut Zafikri (2008), bahwa sistem temu kembali informasi terutama berhubungan dengan pencarian informasi yang isinya tidak memiliki struktur. Demikian pula ekspresi kebutuhan pengguna yang disebut *query*, juga tidak memiliki struktur. Hal ini yang membedakan sistem temu kembali informasi dengan sistem basis data. Dokumen adalah contoh informasi yang tidak terstruktur. Isi dari suatu dokumen sangat tergantung pada pembuat dokumen tersebut .

Sistem Temu Kembali Informasi sebagai sistem yang

berfungsi untuk menemukan informasi yang relevan dengan kebutuhan pemakai, merupakan salah satu tipe sistem informasi. Salah satu hal yang perlu diingat adalah bahwa informasi yang diproses terkandung dalam sebuah dokumen yang bersifat tekstual.

Dalam konteks ini, temu kembali informasi berkaitan dengan representasi, penyimpanan dan akses terhadap dokumen representasi dokumen. Dokumen yang ditemukan tidak dapat dipastikan apakah relevan dengan kebutuhan informasi pengguna yang dinyatakan dalam *query*. Pengguna Sistem Temu Kembali Informasi sangat bervariasi dengan kebutuhan informasi yang berbeda-beda.

Tujuan dari sistem IR adalah:

1. Menemukan seluruh dokumen yang relevan terhadap suatu *query.*
2. Hanya menemukan dokumen relevan saja, artinya tidak terdapat dokumen yang tidak relevan pada dokumen hasil pencarian.

3.2 Pakar

Menurut Bunyamin (2008), *Searching* adalah program komputer yang digunakan untuk menemukan dokumen-dokumen yang ada dalam komputer pribadi. *Searching* berusaha meminta *content* yang akan menjadi patokan pencarian sebuah dokumen dengan kriteria yang spesifik (biasanya yang berisi kata atau frasa yang kita tentukan) dan memperoleh daftar *file* yang memenuhi kriteria tersebut. *Searching* biasanya menggunakan indeks untuk mencari dokumen setelah pengguna memasukkan kriteria pencarian .

Menurut Erwin (2005) *Searching* dapat dilakukan dengan 2 cara, yaitu dengan cara umum dan canggih. Perbedaannya pada informasi yang Anda masukkan tempat pencarian, semakin banyak dan akurat kata yang bersangkutan dan parameter pencariannya akan semakin akurat pula hasilnya. Pada pencarian canggih Anda dapat memasukkan cukup banyak parameter pencarian. Metode *searching* umum akan mencari kata kunci yang Anda masukkan ke semua informasi dalam katalog dan naskah data. Karenanya cenderung hasil pencarian cukup banyak dan kurang akurat, namun semua data yang berhubungan dengan kata kunci tersebut akan ditampilkan .

Pencarian dapat dibagi 2 bagian, yaitu:

1. Pencarian internal adalah pencarian terhadap sekumpulan data yang disimpan di dalam memori utama.
2. Pencarian eksternal adalah pencarian terhadap sekumpulan data yang disimpan di dalam memori sekunder, seperti disk.

3.3 TextInputStream

Menurut Ford (2006), Untuk membaca teks dari sebuah file, maka perlu dibuat objek *TextInputStream*. *TextInputStreams* memiliki metode yang memungkinkan untuk membaca dari sebuah file yang dimiliki Xojo dan fitur *close* setelah selesai pada akhir pembacaan file. Metode ini diciptakan bersama ketika memanggil metode *Open*. *File* yang tidak mendukung UTF-8, maka harus diatur nilai properti *Encoding*.

Ketika membaca sebuah *file* teks yang berasal dari sistem operasi lain atau dalam bahasa lain (atau campuran bahasa), mungkin perlu untuk menetapkan pengkodean teks yang digunakan ketika *file* tersebut ditulis. Jika tahu pengkodean, menggunakan modul *Encodings* untuk mendapatkan *encoding* dan menggunakannya untuk mengatur nilai properti.

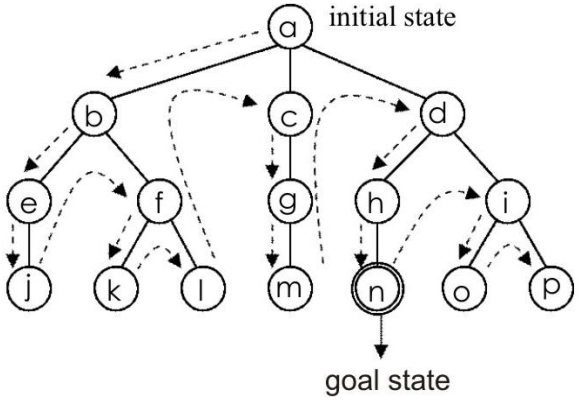
Berikut adalah contoh *Encoding* objek *TextInputStream* untuk membaca sebuah *file* teks yang menggunakan pengkodean *MacRoman*:

```
Dim f As FolderItem = GetOpenFolderItem("text")
 If f <> Nil Then
  If f.Exists Then
      Dim t As TextInputStream
       Try
              t = TextInputStream.Open(f)
              t.Encoding = Encodings.MacRoman
              TextArea1.Text = t.ReadAll
       Catch e As IOException
              t.Close
              MsgBox("Error accessing file.")
       End Try
  End If
 End If
```

Seperti kita lihat, variabel f dideklarasikan ke dalam tipe *FolderItem* dan t sebagai *TextInputStream*. "t = *TextInputStream.Open*(f)" menunjukkan bahwa modul *TextInputStream* untuk membuka file *text* yang telah disimpan di variabel f. Kemudian di *encoding* : *t.Encoding = Encodings.MacRoman*.

3.4 *Depth Depth Search*

Menurut Erwin (2005), Pencarian dengan metode ini dilakukan dari *node* awal secara mendalam hingga yang paling akhir *(dead-end)* atau sampai ditemukan. Dengan kata lain, simpul cabang atau anak yang terlebih dahulu dikunjungi. Sebagai ilustrasinya dapat dilihat gambar 2.2



Gambar 2.2 Teknik pencarian DFS

Sumber : Sumber: Erwin, Muhammad Ashari Hariyono dan Wahyudi. 2005. *Cutomer Information Gathering Menggunakan Metode Temu Kembali Informasi dengan Model Ruang Vektor.* Yogyakarta: UII

Berdasarkan gambar 2.2, proses pencarian dilakukan dengan mengunjungi cabang terlebih dahulu hingga tiba di simpul terakhir. Jika tujuan yang diinginkan belum tercapai maka pencarian dilanjutkan ke cabang sebelumnya, turun ke bawah jika memang masih ada cabangnya. Begitu seterusnya hingga diperoleh tujuan akhir *(goal).*

Depth First Search juga memiliki kelebihan di

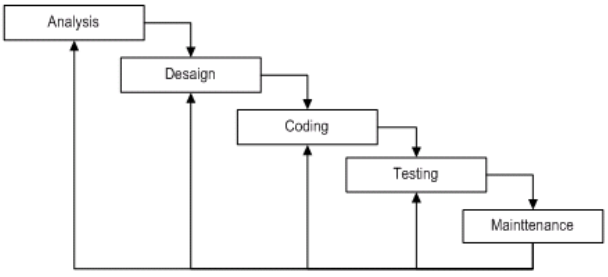
antaranya adalah cepat mencapai kedalaman ruang pencarian. Jika diketahui bahwa lintasan solusi permasalahan akan panjang maka *Depth First Search* tidak akan memboroskan waktu untuk melakukan sejumlah besar keadaan dangkal dalam permasalahan graf. *Depth First Search* jauh lebih efisien untuk ruang pencarian dengan banyak cabang karena tidak perlu mengeksekusi semua simpul pada suatu level tertentu pada daftar *open.* Selain itu, *Depth First Search* memerlukan memori yang relatif kecil karena banyak *node* pada lintasan yang aktif saja yang disimpan.

Selain kelebihan, *Depth First Search* juga memiliki kelemahan di antaranya adalah memungkinkan tidak ditemukannya tujuan yang diharapkan dan hanya akan mendapatkan satu solusi pada setiap pencarian.

3.5 Waterfall

Metode ini merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode waterfall adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2,3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.

Metode pengembangan sistem yang digunakan dalam perancangan sistem ini menggunakan menggunakan Model Waterfall, dikarenakan metode ini mempunyai tahapan-tahapan yang jelas, nyata dan praktis. Setiap tahapan harus diselesaikan terlebih dahulu untuk menghindari terjadinya pengulangan dalam tahapan sehingga pengembangan sistem yang dilakukan dapat memperoleh hasil yang diinginkan.



Gambar 1. Metode Pengembangan Waterfall

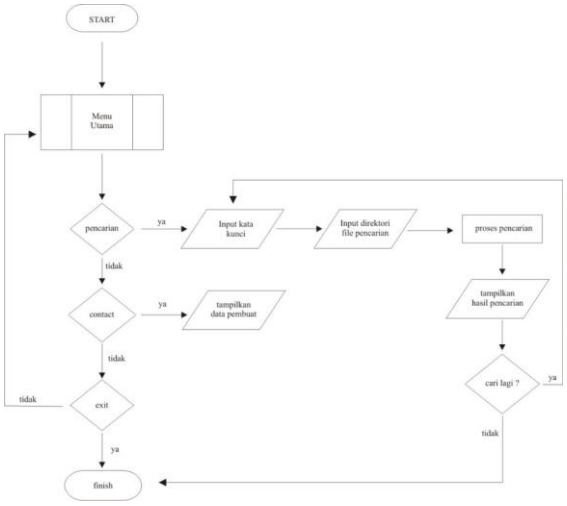
3.6 Flowchart

flowchart adalah suatu bagan alir yang digunakan untuk menunjukkan arus pekerjaan atau proses secara menyeluruh dari bagian sistem dimana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem.

4. RANCANGAN SISTEM/APLIKASI

Analisis dan perancangan sistem pakar untuk mendiagnosa penyakit tanaman padi adalah sebagai berikut :

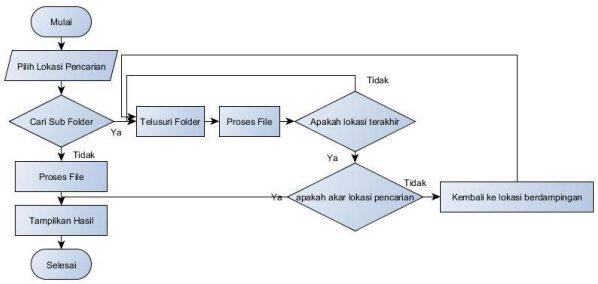
1. Flowchart Program



Gambar 2. Flowchart Program

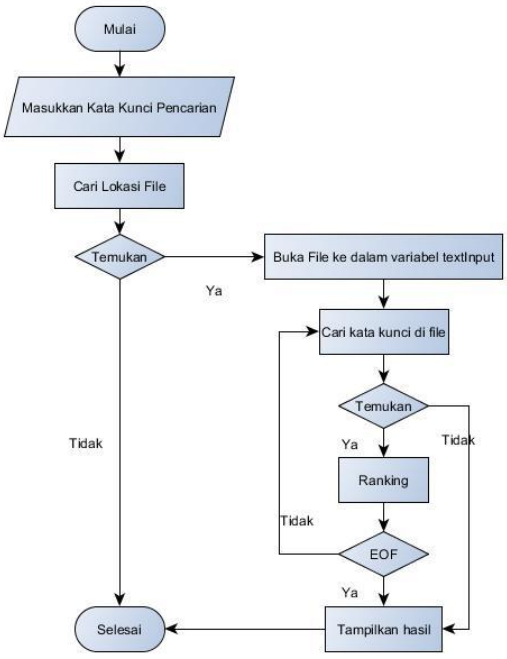
Merupakan diagaram alir Program debgab menu pilihan Pencarian, Kontak, dan Keluar. Pada menu pencarian terdapat pilihan memasukkan kata kunci yang selanjutnya dilakukan pencarian DFS pada tiap sub folder dan pencarian teks pada tiap dokumen yang ditemukan.

2. Flowchart DFS



Gambar 3. Flowchart DFS

3. Flowchart *REGEX*



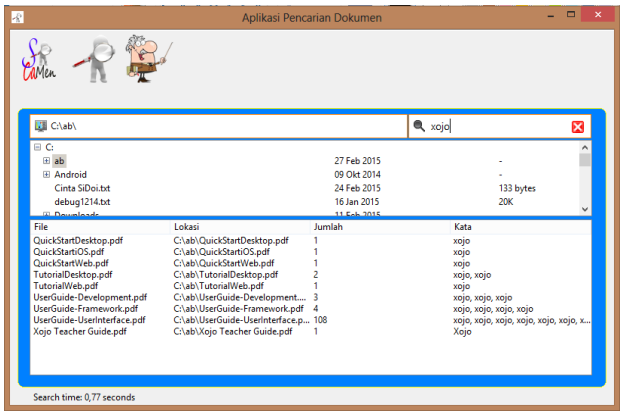
Gambar 4. Flowchart Pencarian Kata menggunakan Regex

Pada gambar 4 merupakan *flowchart* dari proses pencarian kata menggunakan Regex. Teks yang terdapat di dalam dokumen dimasukkan ke dalam variabel *textInput* menggunakan perintah *TextInputStream*, yaitu *syntac* dari Xojo yang berfungsi membaca *string*/teks dari dokumen. Untuk menghindari kesalahan bentuk tulisan maka diperlukan textInput.*encoding* yaitu fungsi untuk merubah ke format *MacRoman*. Teks yang ditemukan di*rangking* berdasar jumlah teks yang ditemukan. Pencarian teks berhenti sampai akhir dokumen dan diakhiri dengan menampilkan hasil.

5. IMPLEMENTASI

Hasil implementasi berdasarkan analisis dan perancangan adalah sebagai berikut :

1. Form Pencarian



Gambar 5 Tampilan Pencarian

Pada *form* pencarian terdapat satu *textfiled* untuk memasukkan kata kunci, sebuah tombol pencarian, dan sebuah *listbox* untuk menampilkan hasil.

Kata kunci dimasukkan di *textfield*, kata kunci yang dimasukkan berupa kata tunggal atau kalimat. Untuk mengoptimalkan pencarian dapat digunakan fungsi Regex.

Pencarian akan dilakukan di dalam folder terpilih, jika di dalam folder terpilih terdapat folder lagi maka tidak dilakukan pencarian ke dalam folder tersebut.

Hasil pencarian kata berupa dokumen teks akan ditampilkan pada *listbox*. Dapat kita lihat pada gambar 5, pada *listbox* terdiri dari kolom *file*, lokasi, jumlah dan kata. Kolom File yaitu berisi nama *file* beserta ekstensinya. Sedangkan Lokasi file menunjukkan posisi keberadaan file tersebut di dalam *folder*. Untuk Jumlah, yaitu hasil *rangking* kata yang ditemukan di dalam dokumen.

2. *Form About*

Pada *form* ini berisikan tentang pembuat aplikasi dan beberapa informasi. Pada gambar 6 dapat dilihat *form* terdiri atas judul, info aplikasi, info pembuat, serta beberapa tombol. Tombol About akan menampilkan tampilan About yaitu tentang aplikasi, versi serta informasi siapa yang membuat. Pada informasi pembuat akan ditampilkan informasi yang dapat dihubungi agar aplikasi dapat berkembang. Tombol EULA akan menampilkan Info Peraturan Pemakaian Aplikasi, syarat pemakaian, batasan-batasan dan apa-apa yang diperbolehkan jika ingin menggunakan aplikasi. Tombol Thanks To menampilkan Informasi tentang ucapan terimakasih pembuat aplikasi kepada pihak-pihak terkait yaitu STMIK WIDIYA CIPTA DARMA SAMARINDA dan XOJO, inc.

Gambar 6 *Form About*

6. KESIMPULAN

Dengan hasil penelitian yang telah dilakukan dan berdasarkan uraian-uraian yang dibahas dalam bab-bab sebelumnya, maka dapat ditarik eksimpulan bahwa:

1. Dalam pencarian kembali dokumen dapat digunakan *RegEx* yang begitu familiar di kalangan *programmer* sehingga fungsi-fungsi *syntax* RegEx dapat digunakan untuk pilihan bantuan agar pencarian kata lebih relevan,
2. Aplikasi pencarian sangat diperlukan terutama instansi pemerintah yang selalu berhubungan dengan arsip dokumen,
3. Kecepatan pecarian tergantung jumlah dokumen, besar dukumen, dan spesifikasi komputer,
4. Pencarian menggunakan *RegEx* sehingga fungsi-fungsi *syntax RegEx* dapat digunakan untuk pilihan bantuan agar pencarian kata lebih relevan,
5. Aplikasi dapat digunakan oleh orang yang awam fungsi *RegEx* sekalipun.

7. SARAN

Untuk mendapatkan manfaat yang maksimal, maka penulis mengajukan beberapa saran sebagai pertimbangan antara lain :

1. Ketika pencarian sedang berlangsung supaya tidak menggunakan aplikasi yang memakan RAM dan penggunaan *Harddisk* tinggi, karena menggangu kecepatan baca *file* dan proses pencarian.
2. Nantinya aplikasi dapat terintegrasi dengan *Explorer* atau dibuat *widget*
3. Menambahkan pilihan-pilihan khusus RegEx seperti *Replace*.

8. DAFTAR PUSTAKA

Buku:

Eldira, Hervilorra dan Entin Martiana. 2004. *Web Mining untuk Pencarian Dokumen Bahasa Inggris Menggunakan Hill Climbing Automatic Cluster*. Surabaya: ITS.

Erwin, Muhammad Ashari Hariyono dan Wahyudi. 2005. *Cutomer Information Gathering Menggunakan Metode Temu Kembali Informasi dengan Model Ruang Vektor*. Yogyakarta: UII

Fatansyah, 2004. *Basis Data*. Informatika : Bandung.

Jerry Lee Ford, Jr. 2006, *Beginning REALbasic : From Novice to Professional*. Springer-Verlag, New York.

Jogiyanto, Hartono. 2008. *Analisis dan Desain Sistem Informasi*, Edisi III. Yogyakarta:ANDI.

Pressman, Roger S. 2005. *Software Engineering A Practitioner's Approach.* New York : McGraw-Hill Inc.

Jurnal Ilmiah:

Huda, Nuqson Masykur. 2010. *Aplikasi Data Mining untuk Menampilkan Informasi Tingkat Kellusan Mahasiswa*. Skripsi: Semarang, Indonesia: Universitas Diponegoro.

Muliantara, Agus. *Penerapan Regular ExpresiondalamMelindungiAlamat Email dari Spam Robot padaKontenWordpress.*Skripsi. Bali, Indonesia :UniversitasUdayana.

Zafikri, Atika. 2008. *Implementasi Metode Term Frequency Inverse Document Frequency (TF-IDF) pada Sistem Temu Kembali Informasi.* Skripsi. Medan. Universitas Sumatra Utara