IMPLEMENTASI DAN PENGUJIAN APLIKASI PENILAIAN UJIAN ESSAY

Ade Bastian[1)], Harun Sujadi[2)]
Program Studi Informatika, Fakultas Teknik, Universitas Majalengka
Email : adb@ft.unma.ac.id, hns@ft.unma.ac.id

ABSTRACS

Essay Exam is one of the evaluation processes in the learning process to determine the ability of students. Essay Exams demand students' reasoning to answer every question given by the Lecturer. But the essay exam requires time in the examination of answers. Besides that, the consistency of the values given by the Lecturer in all answers is also needed. The application to assess the results of the essay exam has been designed using the Cosine Similarity method for Stemming and Nazief & Andriani Algorithms to calculate the similarity of answers with the answer key. The stages of application analysis and development using the two methods above are then implemented into an application.
This study examines the results of the essay exam assessment application using blackbox.

Keywords — Cosine Similarity, Stemming, Nazief & Andriani Algorithm, Blackbox

1. PENDAHULUAN

Umumnya proses pembelajaran di perguruan tinggi disampaikan perkuliahan secara teori dan praktek. Untuk mengetahui tingkat pemahaman mahasiswa, diselenggarakan Ujian Tengah Semester (UTS) dan Ujian Akhir Semester (UAS). Dalam penyelenggaraannya, soal yang diberikan berbentuk essay dan pilihan ganda. Untuk soal pilihan ganda, mahasiswa memilih salah satu jawaban yang sudah disediakan dalam lembar soal. Kemungkinan hasil dari soal pilihan ganda yaitu benar atau salah. Dosen menghitung jumlah jawaban benar sebagai nilai akhir dari ujian tersebut. Pada soal essay, mahasiswa dituntut untuk menjawab soal dengan pemahaman yang mereka miliki. Soal essay akan menghasillkan beragam kemungkinan jawaban sesuai dengan pemahaman masing-masing mahasiswa. Hasil dari jawaban essay bukan hanya benar atau salah tapi juga hampir benar. Sehingga skor nilai untuk jawaban essay bisa beragam. Dosen dituntut untuk konsisten dalam memberikan skor nilai jawaban essay semua mahasiswa. Selain konsistensi, Dosen juga memerlukan waktu lebih banyak untuk pemeriksaan jawaban ujian essay. (Sakti Pramukantoro, 2016)

Pengembangan aplikasi penilaian ujian essay membutuhkan beberapa alat analisa sebagai dasar dalam pemrograman pengujian jawaban esaay tersebut. Untuk proses *stemming* atau konversi kata menjadi kata dasar dapat menggunakan metode *Cosine Similarity*. Kemudian untuk menghitung tingkat kemiripan jawaban dengan kunci jawaban dapat menggunakan algoritma *Nazief & Andriani*. (Bastian, 2017)

Hasil analisa dijadikan acuan untuk mengembangkan aplikasi penilaian ujian essay. Metode pengembangan menggunakan *Extreme Programming* (*XP).*

11. METODOLOGI PENELITIAN

1. Text Mining

Text mining adalah proses menganalisis teks untuk mengekstrak informasi yangberguna untuk tujuantertentu. *Text mining* memiliki tugas yang lebih kompleks karena melibatkan *data* teks yang sifatnya tidakterstruktur dan kabur (*fuzzy*).*Text mining* merupakan bidang multidisiplin yang melibatkan*intampilanation retrieval*, analisis teks, ekstraksi informasi, *clustering*, kategorisasi, visualisasi, teknologibasis data, *machine learning*, dan *data mining*.Perbedaan mendasar antara *text mining* dan *data mining*terletak pada sumber *data* yang digunakan. Pada *data mining*, pola-pola diekstrak dari basis *data* yang terstruktur, sedangkan di *text mining*, pola-pola diekstrak dari *data* tekstual (*natural language*). Secara umum, basis data didesain untuk program dengan tujuan melakukan pemrosesan secara otomatis,sedangkan teks ditulis untuk dibaca langsung oleh manusia. *Teks Mining* adalah aplikasi yang berasal dari pencarian informasi dan pemrosesan bahasa alami. Definisi penambahan teks hanyalah metode kecil yang dapat menemukan informasi baru yang tidak jelas atau mudah diketahui dari dokumen yang ada. (Kochady, 2006)

Metode *Text Mining* diantaranya *CaseFolding* dan *Tokenizing, Filtering, Stemming, Analyzing* dan *Stop-Word. Stemming* adalah proses untuk menggabungkan atau memecahkan setiap

varian-varian suatu katamenjadi kata dasar. Proses *stemming* pada kata Bahasa Indonesia berbeda dengan *stemming* pada kataBahasa Indonesia. Proses *stemming* pada kata Bahasa Inggris adalah proses untuk mengeliminasisufiks pada kata sementara proses *stemming* pada Bahasa Indonesia adalah proses untukmengeliminasi sufiks, prefiks dan konfiks. Terdapat beberapa algoritma dalam *stemming*, antara lain algoritma Porter dan algoritma Nazief & Adriani.

2. Algoritma Nazief & Andriani

Konjungsi adalah Algoritma *stemming* Nazief dan Adriani dikembangkan berdasarkan aturan morfologi Bahasa Indonesia yang mengelompokkan imbuhan menjadi awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*) dan gabungan awalan akhiran (*confixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih. Aturan morfologi Bahasa Indonesia mengelompokkan imbuhan ke dalam beberapa kategori sebagai berikut (Firdaus, 2014):

a. *Inflection suffixes* yakni kelompok akhiran yang tidak merubah bentuk kata dasar. Sebagai contoh, kata "duduk" yang diberikan akhiran "-lah" akan menjadi "duduklah". Kelompok ini dapat dibagi menjadi dua :
   1) *Particle* (P) atau partikel yakni termaksud di dalamnya "-lah", "kah", "tah" dan "pun".
   2) *Possessive pronoun* (PP) atau kata genti kepunyaan, termaksud di dalamnya "-ku","-mu" dan "-nya".
b. *Derivation suffixes* (DS) yakni kumpulan akhiran asli Bahasa Indonesia yang secara langsung ditambahkan pada kata dasar yaitu akhiran "-i", "-kan", dan "-an".
c. *Derivation prefixes* (DP) yakni kumpulan awalan yang dapat langsung diberikan pada kata dasar murni, atau pada kata dasar yang sudah mendapatkan penambahan sampai dengan 2 awalan. Termaksud di dalamnya adalah:
   a. Awalan yang dapat bermorfologi ("me-", "be-", "pe-" dan "te").
   b. Awalan yang tidak bermorfologi ("di-", "ke-" dan "se-").

Berdasarkan pengklasifikasi imbuhan-imbuhan di atas, maka bentuk kata berimbuhan dalam Bahasa Indonesia dapat dimodelkan sebagai berikut (Firdaus, 2014) :

[DP+ [DP+ [DP+]]] Kata Dasar [[+DS[+PP]

Keterangan :
DP :*Derivation prefixes*
DS :*Derivation suffixes*
PP :*Possessive pronoun*

3. Pengujian Blackbox

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode whitebox (Ayuliana, 2009).

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya (Ayuliana, 2009):

a. Fungsi-fungsi yang salah atau hilang
b. Kesalahan interface
c. Kesalahan dalam struktur data atau akses *database* eksternal
d. Kesalahan performa
e. Kesalahan inisialisasi dan terminasi

Tidak seperti metode *whitebox* yang dilaksanakan diawal proses, ujicoba *blackbox* diaplikasikandibeberapa tahapan berikutnya. Karena ujicoba *blackbox* dengan sengaja mengabaikan struktur kontrol,sehingga perhatiannya difokuskan pada informasi *domain*. Ujicoba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut (Ayuliana, 2009) :

a. Bagaimana validitas fungsionalnya diuji?
b. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik?
c. Apakah sistem secara khusus sensitif terhadap nilai input tertentu?
d. Bagaimana batasan-batasan kelas data diisolasi?
e. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?
f. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem ?

Dengan mengaplikasikan ujicoba blackbox, diharapkan dapat menghasilkan sekumpulan kasus uji yangmemenuhi kriteria berikut (Ayuliana, 2009):

a. Kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari ujikasus tambahan harusdidesain untuk mencapai ujicoba yang cukup beralasan;
b. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan,daripada kesalahan yang terhubung hanya dengan suatu ujicoba yang spesifik.

III. HASIL DAN PEMBAHASAN

Berikut ini adalah penggalan-penggalan *source codeprogram* yang berkaitan untuk menyetem sebuah kata yang berimbuhan kembali ke kata dasar. Untuk membaca isi kamus dan memasukkan kata-kata di dalam kamus ke dalam sebuah list yaitu dengan penggalan *source code* berikut:

```
public void bacaKamus() throws
FileNotFoundException, IOException {
        readDokumen =
readDokumenTeks("kamus");
  StringTokenizer strKamus = new
StringTokenizer(readDokumen, "|");
  while (strKamus.hasMoreTokens()) {
   listKamus.add(strKamus.nextToken());
        }
      }
```

Gambar 1. Method Baca Kamus

Adapun source code untuk membaca inputan sebuah kata adalah sebagai berikut :

```
public void setKata(String kata) {
    this.kata = kata;
    this.akarKata = kata;
    bersikan = "";
  }
```

Gambar 2. Method setKata()

Setelah itu kata yang diinputkan dicek, apakah sesuai di dalam kamus, jika kata yang diinputkansesuai dengan yang ada pada kamus maka kata tersebut merupakan akar kata (*root word*), penggalan source codenya adalah sebagai berikut :

```
public boolean cekKamus(String kata) {
  if (listKamus.contains(kata)) {
    return true;
  } else {
    return false;
  }
}
```

Gambar 3. Method cekKamus()

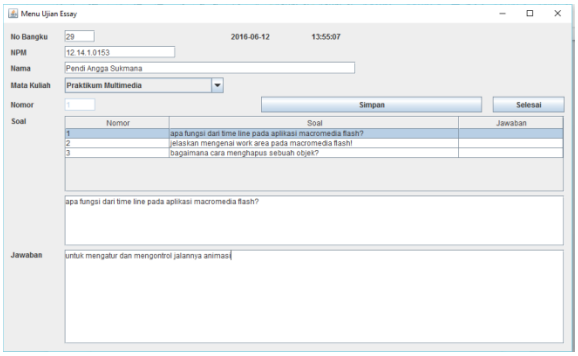
Untuk mendapatkan kata dasar setelah mengalami proses stemming dengan menggunakan algoritma Nazief-Andriani, berikut ini adalah penggalan source codenya:

```
public String KataDasar(String kata) {
    setKata(kata);
    if (cekKamus(kata)) {
      return akarKata;
    } else {
      hapusInfleksionalSuffiks();
      hapusDerivationSuffiks();
    }
    return akarKata;
  }
```

Gambar 4. Method Stemming Nazief & Andriani

Berikut ini adalah tampilan dari aplikasi Penilaian Ujian Essay yang telah dibuat, yaitu tampilan form login, form menu utama, form data mahasiswa, form data dosen atau admin, form input soal dan kunci jawaban, form ujian essay serta form daftar nilai.

Pengujin *Black Box Testing* yang dilakukan pada fungsi-fungsi seperti tombol, *textbox* dan tabel yang ada dalam aplikasi Penilaian Ujian Essay dapat dilihat sebagai berikut.



Gambar 4. Tampilan Form Ujian Essay

Berikut ini adalah tabel hasil dari pengujian *black box testing* yang dilakukan pada tampilan ujian essay.

Tabel 1. Pengujian Tampilan Ujian Essay

No	Pengujian	Keterangan
1	Menginputkan NPM mahasiswa yang sudah ada di dalam database pada textbox NPM	Sukses
2	Memilih mata kuliah pada combo box mata kuliah	Sukses
3	Mengklik baris soal pada tabel	Sukses
4	Klik tombol simpan dengan mengosongkan nomor bangku	Sukses
5	Klik tombol simpan dengan mengosongkan NPM	Sukses
6	Klik tombol simpan dengan Nama masih dalam keadaan kosong	Sukses
7	Klik tombol simpan tanpa memilih mata kuliah	Sukses
8	Mengklik simpan tanpa memilih soal pada tabel	Sukses
9	Mengklik tombol simpan tanpa mengisi textbox jawaban	Sukses
10	Mengklik tombol simpan dengan semua data telah di pilih dan di isi	Sukses
11	Klik tombol selesai	Sukses

IV. KESIMPULAN

Penelitian lanjutan dari penelitian sebelumnya ini, menghasilkan beberapa kesimpulan diantaranya :

1. Pengembangan aplikasi penilaian ujian essay dapat menggunakan metode *Cosine Similarity* dan algoritma *Nazief & Andriani*
2. Ujicoba aplikasi penilaian ujian essay menggunakan pengujian *blackbox* menunjukkan hasil pengujian jawaban essay yang sesuai dengan kunci jawaban yang telah dimasukkan terlebih dahulu.
3. Aplikasi penilaian ujian essay ini dapat dikembangkan menggunakan metode lainnya untuk mencapai hasil pengujian yang lebih tepat.

V. REFERENSI

Ayuliana. 2009. Teknik Pengujian Perangkat Lunak. *Testing Dan Implementasi*, 1-6.

Bastian, Ade, Harun Sujadi, Dan Pendi Angga Sukmana. 2017. Development Of Essay Training Application Using Nazief & Andriani Algorithm And Cosine Similarity Method. International Seminar Proceeding Msceis. Universitas Pendidikan Indonesia.

Firdaus, A., Ernawati, Vatresia, Dkk. 2014, April. Aplikasi Pendeteksi Kemiripan Pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani Dan Metode Cosine Similarity. *Jurnal Teknologi Informasi, 10 Nomor 1*, 1-14.