

PERANCANGAN KOREKSI UJIAN ESSAY OTOMATIS DENGAN TEXT MINING PADA E-LEARNING STIKOM BALI

Ida Bagus Ketut Surya Arnawa
STMIK STIKOM Bali
arnawa@stikom-bali.ac.id

Abstrak. Dalam dunia pendidikan tahap evaluasi pembelajaran sangat penting dilakukan untuk menentukan sukses atau tidaknya proses pembelajaran yang dilakukan selama ini. Salah satu jenis soal ujian yang dapat dilakukan dalam proses evaluasi pembelajaran adalah ujian *essay*. Sistem *e-learning* yang dibangun di STIKOM Bali merupakan sebuah sistem yang digunakan untuk menyampaikan materi pembelajaran, forum diskusi, pengumpulan tugas serta ujian *online*. Salah satu jenis soal ujian yang diberikan *e-learning* adalah jenis soal dalam bentuk *essay*. Namun dalam ujian soal bentuk *essay*, dosen membutuhkan waktu yang lebih dalam memeriksa jawaban setiap mahasiswa, semakin banyak ujian semakin banyak pula waktu yang dibutuhkan dalam memeriksa jawaban mahasiswa. Hal ini menyebabkan objektivitas dalam penilaian menurun. Untuk mengatasi permasalahan kurang objektivitasnya penilaian terdapat evaluasi dalam ujian *essay*, maka perlu dikembangkan sebuah cara untuk mengoreksi jawaban *essay* secara otomatis pada sistem *e-learning* dengan memanfaatkan *Text Mining* dengan algoritma *Rabin Karp*. Hasil perancangan berupa *data flow diagram*, konseptual *database* dan struktur tabel. Perancangan tersebut menampilkan aliran data yang digunakan untuk kebutuhan *input*, proses dan *output*. Perancangan ini dapat diimplementasikan sehingga dapat mengatasi permasalahan kurang objektivitasnya penilaian terdapat evaluasi dalam ujian *essay*.

Kata Kunci: *Data mining, Rabin Karp, e-learning, koreksi essay*

PENDAHULUAN

Dalam dunia pendidikan tahap evaluasi pembelajaran sangat penting dilakukan untuk menentukan sukses atau tidaknya proses pembelajaran yang dilakukan selama ini. Evaluasi pembelajaran juga dapat digunakan untuk mengetahui tingkat kemampuan mahasiswa sebagai peserta didik dalam memahami suatu bidang studi yang sedang ditempuh. Salah satu cara yang dapat dilakukan untuk mengevaluasi pembelajaran adalah dengan melakukan ujian. Ujian mempunyai manfaat untuk mengetahui seberapa paham peserta didik terhadap materi yang disampaikan oleh seorang pengajar. Salah satu jenis soal ujian yang dapat dilakukan dalam proses evaluasi pembelajaran adalah ujian *essay*. Ujian dengan sistem *essay* merupakan bentuk evaluasi dimana pilihan jawaban tidak disediakan, dan siswa harus menjawab dengan kalimat (Putri Ratna, et al., 2007).

STIKOM Bali merupakan salah satu perguruan tinggi swasta di bidang ICT (*Information and Communication Technology*) yang berada di Bali. STIKOM Bali didirikan pada tanggal 10 Agustus 2002 dengan izin dari mendiknas RI No. 157/D/O/2002. Sebagai salah satu perguruan tinggi di bidang ICT STIKOM Bali selalu melakukan inovasi dalam meningkatkan kualitas pendidikan. Banyak inovasi yang dilakukan diantaranya inovasi dalam pemanfaatan Teknologi Informasi dalam menunjang proses belajar dan mengajar. Pemanfaatan teknologi informasi dalam menunjang kegiatan belajar dan mengajar dilakukan dengan memanfaatkan sistem *e-learning*. Sistem *e-learning* yang dibangun merupakan sebuah sistem yang digunakan untuk menyampaikan materi pembelajaran, forum diskusi, pengumpulan tugas serta ujian *online*.

Salah satu jenis soal ujian yang diberikan *e-learning* adalah jenis soal dalam bentuk *essay*. Ujian dengan jenis soal *essay* ini tetap menjadi pilihan bagi para dosen dalam melakukan evaluasi pembelajaran untuk mengetahui tingkat pemahaman peserta didiknya dalam menerima materi yang telah diberikan. Namun dalam ujian soal bentuk *essay*, dosen membutuhkan waktu yang lebih dalam memeriksa jawaban setiap mahasiswa, semakin banyak ujian semakin banyak

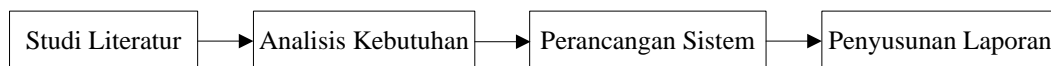
pula waktu yang dibutuhkan dalam memeriksa jawaban mahasiswa. Hal ini menyebabkan objektivitas dalam penilaian menurun.

Untuk mengatasi permasalahan kurang objektivitasnya penilaian terdapat evaluasi dalam ujian *essay*, maka perlu dikembangkan sebuah cara untuk mengoreksi jawaban *essay* secara otomatis pada sistem *e-learning* dengan memanfaatkan *Text Mining*. Dalam *Text Mining* ada beberapa metode yang dapat digunakan untuk melakukan koreksi otomatis pada jawaban soal *essay* salah satunya adalah metode *Rabin Karp*. Metode *Rabin Karp* adalah metode pencarian kata yang mencari sebuah pola berupa *substring* dalam sebuah teks menggunakan *hashing*. Metode ini sangat efektif untuk pencocokan kata dengan pola banyak. Dengan dikembangkan cara koreksi otomatis jawaban ujian *essay* pada *e-learning* diharapkan mampu menyelesaikan permasalahan subjektivitas dosen yang terjadi pada proses penilaian jawaban ujian *essay* secara manual.

METODE PENELITIAN

1. Sistematis Penelitian

Sistematis penelitian yang digunakan untuk Perancangan Koreksi Ujian *Essay* Otomatis Dengan *Text Mining* Pada *E-learning* STIKOM Bali dapat dilihat pada Gambar 1 berikut.



Gambar 1. Sistematis Penelitian

Pada sistematis penelitian yang digunakan terdapat empat tahapan yang dilakukan. Adapun penjelasan masing-masing tahapan dari sistematis penelitian yaitu sebagai berikut :

a. Studi Literatur

Pada tahapan studi literatur dilakukan pengumpulan sumber-sumber bacaan baik dari buku jurnal ilmiah dan sumber lainnya yang berhubungan dengan *string match* menggunakan algoritma *rabin karp*.

b. Analisis Kebutuhan

Pada tahapan ini dilakukan analisis kebutuhan yang diperlukan oleh pihak STIKOM dalam perancangan sistem koreksi ujian *essay* otomatis.

c. Perancangan Sistem

Setelah mengetahui kebutuhan sistem yang akan dirancang maka dilakukan perancangan sistem dengan menggunakan *Data Flow Diagram*, *Konseptual Database* dan *Struktur Tabel*.

d. Penyusunan Laporan

Tahapan selanjutnya yaitu membuat laporan dari hasil perancangan sistem yang sudah dilakukan.

2. Algoritma Rabin Karp

Algoritma *Rabin-Karp* diciptakan oleh Michael O. Rabin dan Richard M. Karp pada tahun 1987 yang menggunakan fungsi *hashing* untuk menemukan *pattern* di dalam *string* teks (Gipp, B. and Meuschke, N. 2011). Algoritma *Rabin-Karp* digunakan sebagai algoritma deteksi kesamaan pada dokumen tugas akhir. Algoritma ini memiliki beberapa keunggulan dalam penerapannya, salah satunya adalah algoritma ini sangat cocok digunakan untuk *string* yang panjang (Raharja, I. Putu Dharma Ade, and I. Putu Dharma Ade Raharja. 2015). Karakteristik algoritma *Rabin-Karp*, yaitu :

e. Menggunakan sebuah fungsi *hashing*.

f. Tahap *preprocessing* menggunakan kompleksitas waktu $O(m)$

g. Untuk tahap pencarian kompleksitasnya : $O(mn)$

h. Waktu yang diperlukan $O(n+m)$

Dalam algoritma *Rabin-Karp*, ada beberapa tahap yang harus dilelalui dalam implementasi algoritma tersebut.

a. Preprocessing

Tahap ini melakukan analisis semantik (kebenaran arti) dan sintaktik (kebenaran susunan) teks. Tujuan dari pemrosesan awal adalah untuk mempersiapkan teks menjadi data yang mengalami pengolahan lebih lanjut. Pembersihan yang dilakukan adalah pembersihan suatu teks dari karakter-karakter yang bersifat sebagai stoplist, dimana karakter tersebut banyak dan pasti muncul dalam setiap teks.

b. Rolling Hash

Fungsi yang digunakan untuk menghasilkan nilai *hash* dari rangkaian *gram* dalam Algoritma *Rabin-Karp* adalah dengan menggunakan fungsi *rolling hash*. Fungsi $hashH(c_1 \dots c_k)$ didefinisikan sebagai berikut,

$$c_1 * b^{(k-1)} + c_2 * b^{(k-2)} + \dots + c_{(k-1)} * b^k + c^k \quad (1)$$

Keterangan :

c : nilai ASCII karakter

b : basis (bilangan prima)

k : banyak karakter

Metode *hashing* digunakan untuk mempercepat pencarian atau pencocokan suatu *string*. Apabila tidak di-*hash*, pencarian akan dilakukan karakter per karakter. Namun pencarian akan menjadi lebih bagus setelah di-*hash* karena hanya akan membandingkan empat digit angka untuk mengetahui kesamaan suatu *substring*. Berikut ini merupakan contoh penghitungan *rolling hash* dengan $k\text{-grams} = 6$ dan basis = 101:

Diketahui

Teks : abracadabra

ASCII : a = 97, b = 98, r = 114

Maka :

$hash("abr") = (97 \times 101^2) + (98 \times 101) + (114 \times 1) = 999.509$ Untuk menghitung nilai *hash* dari *substring* selanjutnya, dapat dilakukan dengan mengurangi *hash* dengan nilai dari karakter yang hilang kemudian mengalikannya dengan basis dan menambahkannya dengan nilai karakter yang ditambahkan. $hash("bra") = [101 \times (999.509 - (97 \times 101^2))] + (97 \times 101) = 1.011.309$.

c. Pencocokan

Proses pencocokan dalam Algoritma Rabin-Karp dilakukan dengan menggunakan sebuah teorema yaitu:

Sebuah *stringA* identik dengan *stringB*, jika (syarat perlu) *stringA* memiliki *hash key* yang sama dengan *hash key* yang dimiliki oleh *stringB*

d. Pengukuran Nilai Kemiripan

Mengukur *similarity* (kemiripan) dan jarak antara dua entitas informasi adalah syarat inti pada semua kasus penemuan informasi, seperti pada *Information Retrieval* dan *Data Mining* yang kemudian dikembangkan dalam bentuk perangkat lunak, salah satunya adalah sistem deteksi kesamaan (Raharja, 2015). Penggunaan ukuran *similarity* yang tepat tidak hanya meningkatkan kualitas pilihan informasi tetapi juga membantu mengurangi waktu dan biaya proses sehingga memperangkat lunakkan *Dice's Similarity Coefficient* dalam penghitungan nilai *similarity* yang menggunakan pendekatan *k-gram*.

$$S = \frac{K * C}{A + B} \quad (2)$$

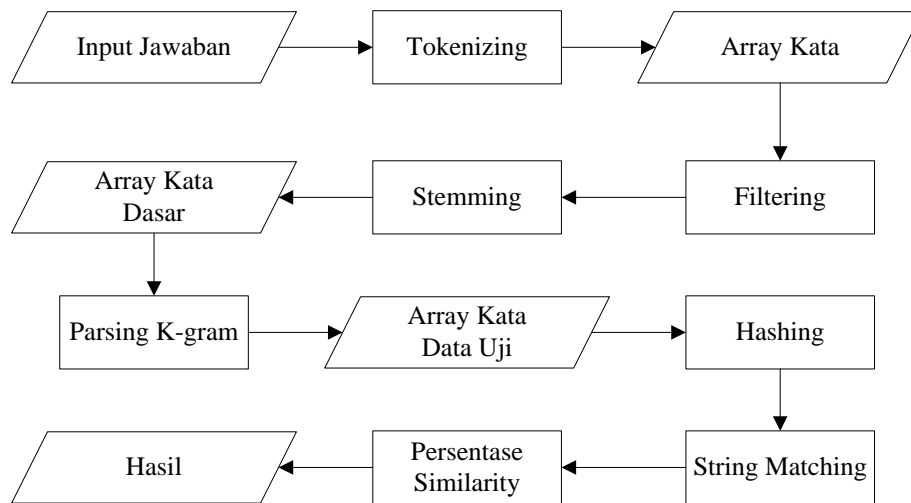
Dimana S adalah nilai *similarity*, A dan B adalah jumlah dari kumpulan *k-grams* dalam teks 1 dan teks 2. C adalah jumlah dari *k-grams* yang sama dari teks yang dibandingkan. Berikut adalah contoh dari penggunaan rumus tersebut. Terdapat dua buah dokumen teks (dok A dan dok B) dengan nilai *k-gram* masing-masing dokumen secara berturut-turut adalah 2608 dan 3040, sedangkan nilai *k-gram* yang sama adalah sebesar 1203. Maka hasil nilai dari kemiripan kedua dokumen tersebut adalah :

$$S = K \times C / (A+B) = 2 \times 1203 / (2608 + 3040) \\ = 2406 / 5648 = 0,4259 \times 100\% = 42,59\%$$

HASIL DAN PEMBAHASAN

1. Perancangan Proses

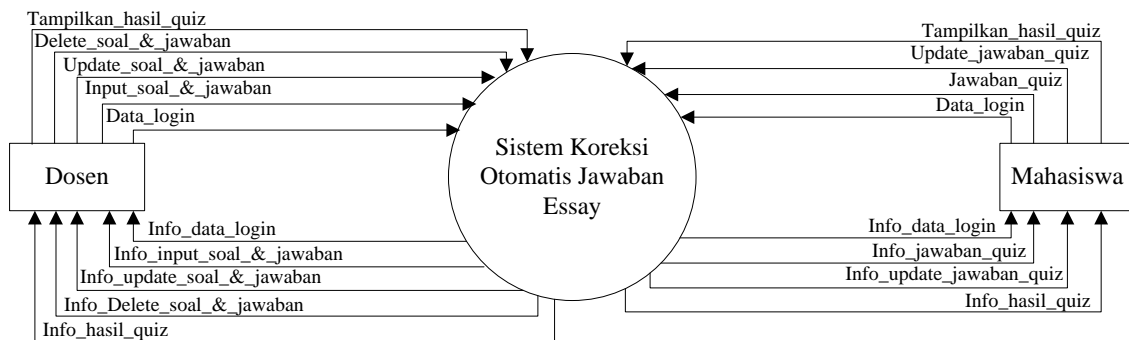
Perancangan sistem yang dibuat merupakan sistem untuk mencocokkan jawaban ujian *essay* dengan kunci jawaban. Sistem ini dapat menerima inputan berupa *text* yang merupakan jawaban dari ujian *essay*. Tahap awal sistem akan mencari informasi tentang jumlah paragraf, jumlah kalimat dan jumlah kata dari jawaban yang diinputkan, kemudian sistem akan melakukan *tokenizing*, *filtering* dan dilanjutkan dengan *stemming* setelah itu dilakukan *string matching* untuk mendapatkan hasil kesamaan jawaban (Gipp, B. and Meuschke, N. 2011). Gambar 2 merupakan gambaran arsitektur sistem koreksi ujian *essay* otomatis.



Gambar 2. Arsitektur Sistem

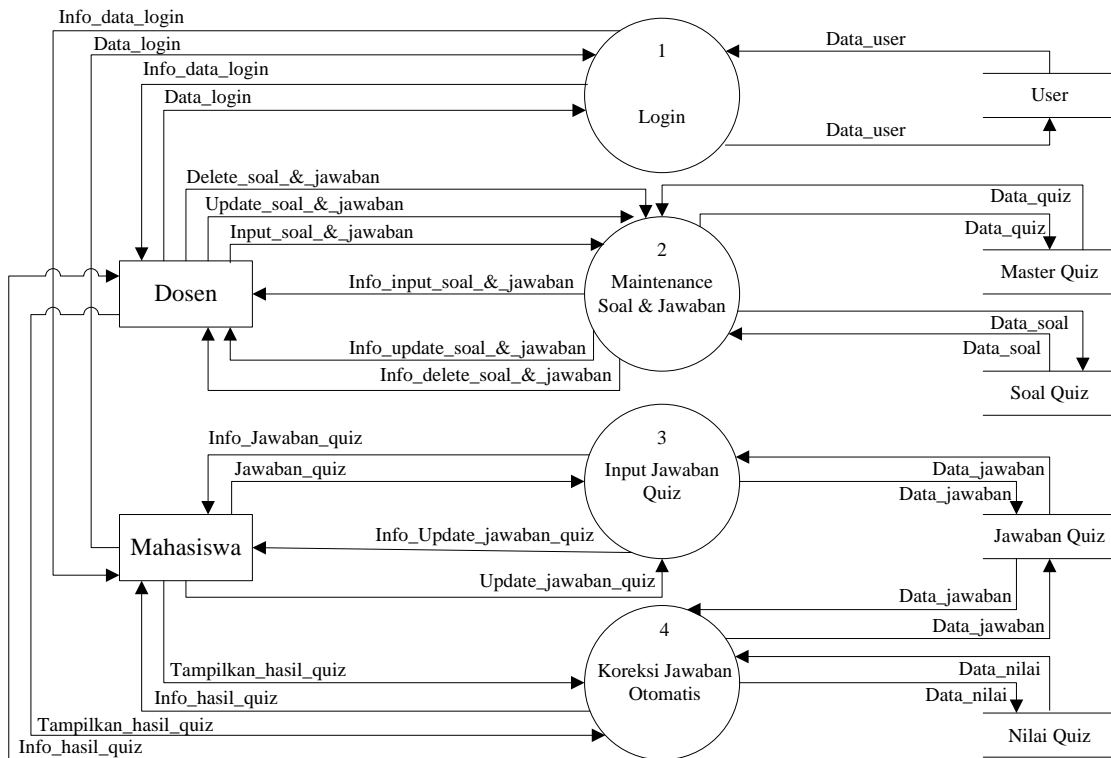
2. Data Flow Diagram (DFD)

Diagram konteks merupakan diagram yang menggambarkan kegiatan keseluruhan pada sistem (S. Pressman, Roger. 2012). Dalam diagram konteks terdapat dua aktor yaitu dosen dan mahasiswa yang masing-masing memiliki hak akses yang berbeda. Gambar 3 merupakan diagram konteks dari sistem koreksi jawaban *essay* secara otomatis.



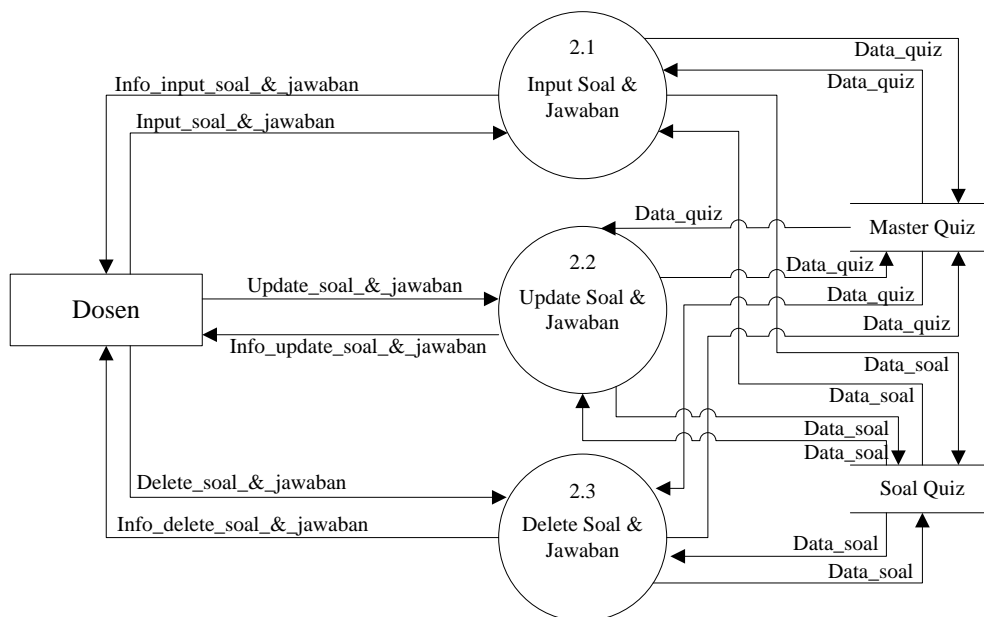
Gambar 3. Diagram Konteks

DFD Level 0 merupakan hasil pemecahan dari diagram konteks. Pada DFD level 0 terdapat empat proses yaitu proses *login*, proses *maintenance* soal & jawaban, input jawaban quiz dan koreksi jawaban otomatis. DFD level 0 dapat dilihat pada Gambar 4.



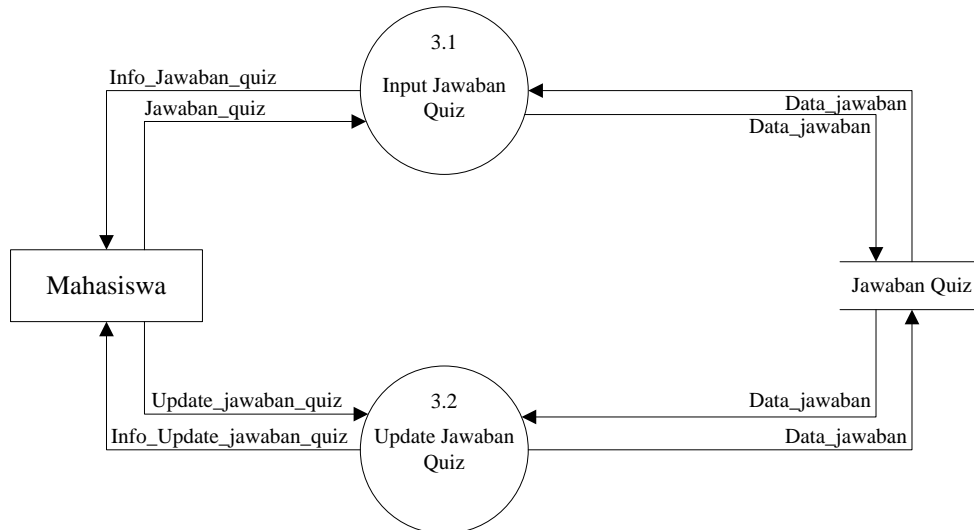
Gambar 4.DFD Level 0

DFD Level 1 ini merupakan pengembangan dari proses *maintenance* soal & jawaban. Pada DFL level 1 ini terdapat tiga proses yaitu *input* soal & jawaban, *update* soal & jawaban dan *delete* soal & jawaban. Proses ini digunakan oleh dosen untuk mengelola soal quiz. Gambar 5 merupakan gambaran dari DFD level 1 proses *maintenance* soal dan jawaban.



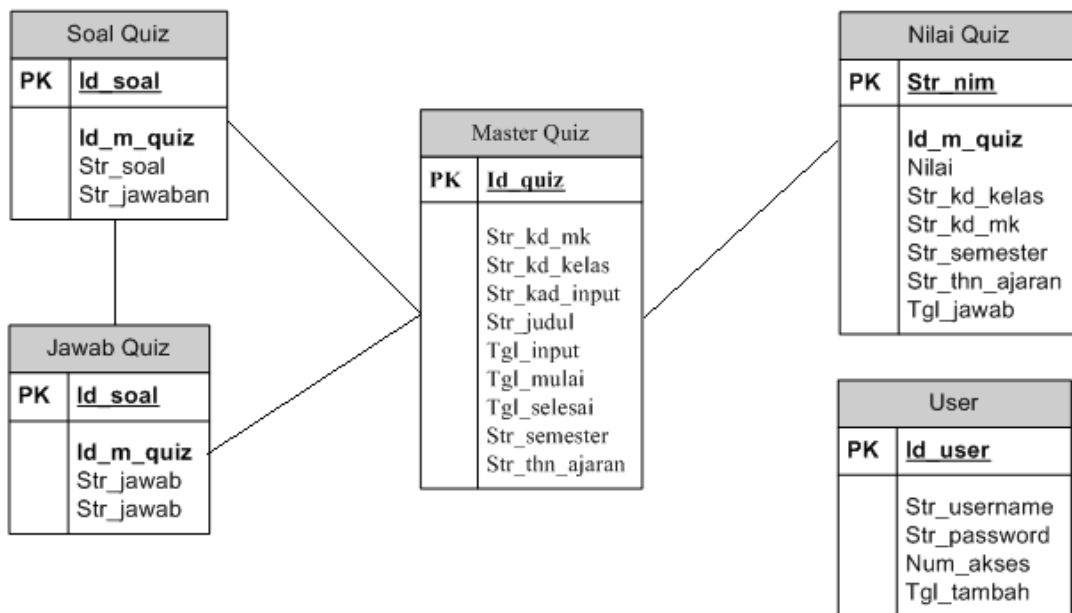
Gambar 5.DFD Level 1 Proses Maintenance Soal & Jawaban

DFD level 1 ini merupakan pengembangan dari proses *input jawaban quiz*. Pada DFD level 1 ini terdapat dua proses yaitu *input jawaban quiz* dan *update jawaban quiz*. Proses ini digunakan oleh mahasiswa untuk menginput jawaban *quiz*. Selain menginput jawaban *quiz* mahasiswa dapat melakukan perbaikan atau *update jawaban quiz* selama sisa waktu *quiz* masih tersedia. Gambar 6 merupakan gambaran dari DFD Level 1 proses *input jawaban quiz*.



Gambar 6.DFD Level 1 Proses Input Jawaban Quiz

Konseptual *database* merupakan gambaran rancangan tabel yang digunakan pada sistem koreksi ujian *essay* otomatis. Pada sistem ini terdapat lima tabel yaitu tabel *master quiz*, tabel soal *quiz*, tabel jawaban *quiz*, tabel nilai *quiz* dan tabel *user*. Gambar 7 merupakan gambaran dari konseptual *database* yang digunakan.



Gambar 7.Konseptual Database

Struktur tabel merupakan rincian dari konseptual *database*. Masing-masing tabel menampilkan *field-field* yang ada pada tabel tersebut. Selain menampilkan *field* yang ada pada

tabel, struktur tabel juga menampilkan tipe data, ukuran dan keterangan dari masing-masing *field* yang ada pada tabel. Struktur tabel yang digunakan pada sistem koreksi ujian *essay* otomatis dapat dilihat pada Tabel 1, Tabel 2, Tabel 3, Tabel 4 dan Tabel 5.

Tabel 1. Master Quiz

<i>No</i>	<i>Nama Field</i>	<i>Tipe</i>	<i>Ukuran</i>	<i>Keterangan</i>
1	Id_quiz	Integer	4	Id quiz
2	Str_kd_mk	Varchar	6	Kode matakuliah
3	Str_kd_kelas	Varchar	10	Kode kelas
4	Str_kad_input	Varchar	5	Kode dosen input
5	Str_judul	Varchar	100	Judul quiz
6	Tgl_input	Datetime	-	Tanggal input quiz
7	Tgl_mulai	Datetime	-	Tanggal mulai quiz
8	Tgl_selesai	Datetime	-	Tanggal selesai quiz
9	Str_semester	Varchar	10	Semester quiz
10	Str_thn_ajaran	Varchar	10	Tahun ajaran quiz

Tabel 2. Soal Quiz

<i>No</i>	<i>Nama Field</i>	<i>Tipe</i>	<i>Ukuran</i>	<i>Keterangan</i>
1	Id_soal	Integer	4	Id soal quiz
2	Id_m_quiz	Integer	4	Id master quiz
3	Str_soal	Text	-	Soal quiz
4	Str_jawaban	Text	-	Jawaban quiz

Tabel 3. Jawab Quiz

<i>No</i>	<i>Nama Field</i>	<i>Tipe</i>	<i>Ukuran</i>	<i>Keterangan</i>
1	Id_soal	Integer	4	Id soal quiz
2	Id_m_quiz	Integer	4	Id master quiz
3	Str_jawab	Text	-	Jawaban quiz
4	Str_nim	Varchar	10	Nim mahasiswa

Tabel 4. Nilai Quiz

<i>No</i>	<i>Nama Field</i>	<i>Tipe</i>	<i>Ukuran</i>	<i>Keterangan</i>
1	Str_nim	Varchar	10	Nim mahasiswa
2	Id_m_quiz	Integer	4	Id quiz
3	Nilai	Float	4	Nilai quiz
4	Str_kd_kelas	Varchar	6	Kode kelas
5	Str_kd_mk	Varchar	10	Kode matakuliah
6	Str_semester	Varchar	10	Semester quiz
7	Str_thn_ajaran	Varchar	10	Tahun ajaran quiz
8	Tgl_jawab	Datetime	-	Tanggal jawab quiz

Tabel 5. User

<i>No</i>	<i>Nama Field</i>	<i>Tipe</i>	<i>Ukuran</i>	<i>Keterangan</i>
1	Id_user	Integer	4	Id user
2	Str_username	Varchar	50	Username
3	Str_password	Varchar	50	Password
4	Num_akses	Integer	4	Akses user
5	Tgl_tambah	Datetime	-	Tanggal tambah user

PENUTUP

1. Kesimpulan

Berdasarkan uraian dari perancangan sistem koreksi ujian *essay* otomatis dapat ditarik kesimpulan bahwa *Data Flow Diagram*, Konseptual *database* dan Struktur tabel dapat digunakan untuk merancang sistem koreksi ujian *essay* otomatis yang diterapkan pada sistem *e-learning* STIKOM Bali. Pada perancangan yang sudah dilakukan terlihat aliran data yang digunakan untuk

kebutuhan *input*, proses dan *output* sistem. Perancangan ini dapat diimplementasikan sehingga dapat mengatasi permasalahan kurang objektivitasnya penilaian terdapat evaluasi dalam ujian *essay*.

2. Saran

Adapun saran yang dapat penulis berikan dari perancangan sistem ini yaitu ditambahkan desain interface sistem sehingga lebih mudah dalam mengimplementasikan perancangan sistem ini.

DAFTAR RUJUKAN

- Gipp, B. and Meuschke, N., (2011). Citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence. In Proceedings of the 11th ACM symposium on Document engineering 249-258.
- Putri Ratna, A. A., Budiardjo, B., & Hartanto, D. (2007). SIMPLE : Sistem Penilaian Esei Otomatis Untuk Menilai Ujian Dalam Bahasa Indonesia. Makara, *Teknologi*, 11(1). 5-11.
- Raharja, I. Putu Dharma Ade, and I. Putu Dharma Ade Raharja. (2015). Perancangan dan Implementasi Sistem Penilaian Jawaban Esai Otomatis Menggunakan Algoritma Rabin-Karp. *PhD diss.*, Universitas Udayana.
- S. Pressman, Roger. (2012). Rekayasa Perangkat Lunak Pendekatan Praktisi Buku 1 dan 2. Yogyakarta: Andi.