PEMBANGKIT *ENTITY RELATIONSHIP DIAGRAM* DARI SPESIFIKASI KEBUTUHAN MENGGUNAKAN *NATURAL LANGUAGE PROCESSING* UNTUK BAHASA INDONESIA

Parmonangan R Togatorop[1], Rezky Prayitno Simanjuntak[2], Siti Berliana Manurung[3] dan Mega Christy Silalahi[4]

[1,2,3,4]Program Studi Sistem Informasi, Institut Teknologi Del, Jl. Sisingamangaraja, Sitoluama Laguboti, Toba Samosir Sumatera Utara, Indonesia
[1]Email:mona.togatorop@del.ac.id
[2]Email: rezkys1999@gmail.com
[3]Email:sitiberlianamanurung@gmail.com
[4]Email:christysilalahi05@gmail.com

ABSTRAK

Memodelkan *Entity Relationship Diagram* (ERD) dapat dilakukan secara manual, namun umumnya memperoleh pemodelan ERD secara manual membutuhkan waktu yang lama. Maka, dibutuhkan pembangkit ERD dari spesifikasi kebutuhan untuk mempermudah dalam melakukan pemodelan ERD. Penelitian ini bertujuan untuk mengembangkan sebuah sistem pembangkit ERD dari spesifikasi kebutuhan dalam Bahasa Indonesia dengan menerapkan beberapa tahapan-tahapan dari *Natural Language Processing* (NLP) sesuai kebutuhan penelitian. Spesifikasi kebutuhan yang digunakan tim peneliti menggunakan teknik *document analysis*. Untuk tahapan-tahapan dari NLP yang digunakan oleh peneliti yaitu: *case folding*, *sentence segmentation*, *tokenization*, POS t*agging*, *chunking* dan *parsing*. Kemudian peneliti melakukan identifikasi terhadap kata-kata dari teks yang sudah diproses pada tahapan-tahapan dari NLP dengan metode *rule-based* untuk menemukan daftar kata-kata yang memenuhi dalam komponen ERD seperti: entitas, atribut, *primary key* dan relasi. ERD kemudian digambarkan menggunakan Graphviz berdasarkan komponen ERD yang telah diperoleh Evaluasi hasil ERD yang berhasil dibangkitkan kemudian di evaluasi menggunakan metode evaluasi *expert judgement.* Dari hasil evaluasi berdasarkan beberapa studi kasus diperoleh hasil rata-rata *precision*, *recall*, F1 *score* berturut-turut dari tiap ahli yaitu: pada ahli 1 diperoleh 91%, 90%, 90%; pada ahli 2 diperoleh 90%, 90%, 90%; pada ahli 3 diperoleh 98%, 94%, 96%; pada ahli 4 diperoleh 93%, 93%, 93%; dan pada ahli 5 diperoleh 98%, 83%, 90%.
Kata kunci: *Entity Relationship Diagram*, *Natural Language Processing*, *Document Analysis*, Graphviz, *Expert Judgement*.

ABSTRACT

Modeling an ERD can be done manually, but generally obtaining an Entity Relationship (ER) Diagram modeling manually will usually take a long time. So, it takes an ERD generator automation from the requirements specification to make it easier to do ERD modeling. This study will develop a system that produces ERD from requirements specifications in Indonesian by applying several stages of Natural Language Processing (NLP) according to needs research. The requirements specification used by the research team used technical document analysis. The stages of NLP used by the research team are: case folding, sentence segmentation, tokenization, POS tagging, chunking and parsing. Then the research team will conduct the words from the text that have been studied in the stages of NLP with the Rule-Based method to find a list of words that meet the ERD components such as: entities, attributes, primary keys and relations. The research team will describe the results obtained in the previous stage using the Graphviz library. From the results of the evaluation of the ERD system design, the research team used an expert evaluation evaluation. From the evaluation results obtained based on the evaluation of several cases, the results of the average precision, recall, and F1 scores from each expert are: 91%, 90%, 90% in expert 1; in expert 2 obtained 90%, 90%, 90%; in expert 3 obtained 98%, 94%, 96%; in expert 4 obtained 93%, 93%, 93%; and in expert 5 obtained 98%, 83%, 90%.
Keywords: Entity Relationship Diagram, Natural Language Processing, Document Analysis, Graphviz, Expert Judgement.

1. PENDAHULUAN

Entity Relationship Diagram (ERD) merupakan sebuah model konseptual tingkat tinggi basis data untuk mendeskripsikan sebuah sistem maupun batasannya [1]. Pemodelan ERD dapat dilakukan secara manual, namun pemodelan ERD secara *manual* biasanya akan memakan waktu yang lama, pada tahap analisis kebutuhan [2]. Oleh karena itu dibutuhkan proses untuk membangkitkan ERD dari spesifikasi kebutuhan [1]. Spesifikasi kebutuhan adalah salah satu hasil proses *requirement engineering. Requirement engineering* adalah proses mendefinisikan, mendokumentasikan dan memelihara persyaratan dalam proses desain dan dianggap sebagai faktor keberhasilan dalam proyek sistem perangkat lunak.

Saat ini sudah ada sistem yang dapat membangkitkan ERD dari spesifikasi kebutuhan dengan menerapkan *Natural Language Processing* (NLP) yang dapat mempermudah pemodelan ERD bagi *system analyst, database administrator,* dan tim pengembang perangkat lunak lainya karena pembangkit ERD dapat menggambarkan ERD berdasarkan spesifikasi kebutuhan [2]. Penelitian [3] menghasilkan ERD dari spesifikasi kebutuhan menggunakan NLP pada Bahasa Inggris pelabelan kelas kata menggunakan *Part of Speech* (POS) *Tagging, chunking* dan *parsing*. Penelitian [2] menghasilkan ERD dari spesifikasi pada Bahasa Inggris pada satu domain dengan melakukan tahapan pembangkitan ERD menggunakan NLP dengan tahapan *text preprocessing*, POS *tagging*, dan SVM *classifier*. Penelitian [4] menghasilkan ERD dari spesifikasi kebutuhan menggunakan NLP pada Bahasa Inggris menggunakan r*ule-based mapping* untuk menghasilkan ERD berdasarkan spesifikasi kebutuhan dalam bahasa Inggris. Langkah-langkah yang dilakukan adalah pengolahan kata (*text preprocessing*), pelabelan kelas kata menggunakan POS *tagging, chunking* dan *parsing*. Penelitian tersebut menghasilkan ERD dari spesifikasi kebutuhan untuk Bahasa Inggris. Tetapi, penelitian terkait pembangkit ERD dari spesifikasi kebutuhan dalam Bahasa Indonesia belum ada.

Bahasa Indonesia merupakan bahasa kesatuan rakyat Indonesia dan termasuk bahasa yang paling banyak dituturkan. Hal ini didukung dari hasil sensus penduduk pada tahun 2010 yang diterbitkan oleh Badan Pusat Statistik (BPS) yang menunjukkan jumlah penduduk Indonesia sebesar 237.641.326 jiwa dengan jumlah penutur Bahasa Indonesia lebih dari 43 juta jiwa sebagai bahasa ibu dan lebih dari 156 juta penutur Bahasa Indonesia sebagai bahasa kedua [5].

Oleh karena itu penelitian ini bertujuan untuk menghasilkan sebuah pembangkit ERD yang berasal dari spesifikasi kebutuhan menggunakan Bahasa Indonesia menggunakan pendekatan berbasis NLP.

2. MATERI DAN METODE

Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah sebuah diagram struktural yang digunakan untuk merancang sebuah basis data [6]. ERD akan mendeskripsikan data yang disimpan pada sebuah sistem maupun batasannya. ERD memiliki tiga konsep utama yaitu [7]:

1. Entitas
   Sebuah entitas dapat berupa orang, tempat, objek, atau kejadian yang dapat dianggap penting bagi sebuah organisasi atau perusahaan. Setiap entitas memiliki beberapa atribut yang mendeskripsikan karakteristik dari objek. Atribut yang ada dalam entitas harus disimpan dan dicatat dalam basis data [8]. Entitas pada komponen ERD dapat dibedakan menjadi dua macam yaitu *strong entity* dan *weak entity*. *Strong entity* merupakan entitas yang tidak bergantung pada entitas lain atau entitas yang dapat berdiri sendiri. Sedangkan untuk *weak entity* merupakan entitas yang keberadaannya tergantung pada entitas lain.
2. Atribut
   Setiap entitas memiliki karakteristik tertentu yang disebut dengan atribut. Atribut berfungsi untuk mendeskripsikan karakteristik yang ada pada entitas yang disimpan dalam basis data [8]. Berdasarkan karakteristik sifatnya, atribut dapat dibedakan menjadi beberapa jenis [9] yaitu *simple attribute* dan *composite attribute, single valued attribute* dan *multi value attribute, derived attribute, key attribute*. *Primary key* adalah nama untuk atribut yang digunakan dalam mengenali suatu entitas. Atribut dalam entitas yang merupakan *primary key* adalah kode identifikasi yang bersifat unik ditunjukkan berdasarkan masing-masing *record* pada sistem. *Primary key* bertujuan untuk memberitahu lokasi untuk tiap catatan pada suatu *file* tentang catatan-catatan yang sama [9].
3. Relasi
   Relasi adalah sebuah hubungan antara dua atau lebih entitas yang saling berkaitan [8]. Relasi pada ERD dapat digambarkan dengan menggunakan simbol belah ketupat (*diamond*). Relasi memiliki beberapa jenis relasi yaitu *unary, binary, ternary*.

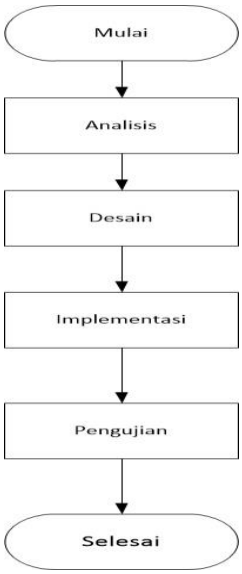
Pemodelan ERD menggunakan notasi entitas berbentuk persegi, relasi berbentuk belah ketupat dan atribut berbentuk oval seperti Gambar 1 [10].

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

Gambar 1. Bentuk entitas, relasi dan atribut dalam ERD

Kerangka Kerja Penelitian

Pada bagian ini, kami menyajikan kerangka kerja atau prosedur kami dengan pendekatan NLP. Gambar 2 menunjukkan gambaran umum dan langkah-langkah yang diikuti untuk menghasilkan Diagram ER.



Gambar 2. Kerangka kerja penelitian

Kerangka kerja penelitian pada gambar 2 terdiri dari beberapa tahapan. Tahapan pertama yang dilakukan yaitu menganalisis kebutuhan yang akan digunakan pada penelitian seperti data, proses yang akan diimplementasikan untuk mencapai tujuan penelitian. Tahapan selanjutnya adalah merancang proses yang diperlukan berdasarkan analisis kebutuhan untuk dapat diimplementasikan menjadi sebuah program. Rancangan yang dihasilkan dapat berupa *flowchart* maupun diagram. Selanjutnya akan dilakukan tahapan implementasi dengan menggunakan bahasa pemrograman *python* dan *javascript*. Kemudian hasil implementasi dievaluasi untuk mengetahui apakah rancangan ERD yang dihasilkan oleh sistem sudah sesuai dengan masukan yang diberikan.

Data

Penelitian ini mengunakan teknik analisis dokumen untuk mengumpulkan data. Analisis dokumen adalah teknik pengumpulan data dengan mengumpulkan dokumen dari lembaga resmi seperti buku, jurnal, artikel, *situs web* dan sebagainya. Dalam penelitian ini, bahasa yang digunakan dalam teknik pengumpulan analisis dokumen adalah spesifikasi kebutuhan pada proses pengumpulan kebutuhan dalam bahasa Indonesia. Studi kasus yang digunakan dalam penelitian ini disimpan dalam format *txt*, yang berisi teks spesifikasi kebutuhan.

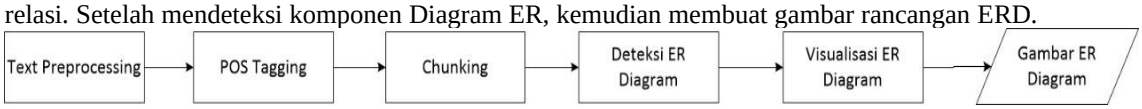
Studi Kasus 1: Pada saat mendaftar menjadi anggota perpustakaan fakultas, mahasiswa mencatat nama, nomor mahasiswa dan alamat mahasiswa. Setelah itu mereka baru bisa meminjam buku diperpustakaan. Buku-buku yang dimiliki perpustakaan banyak sekali jumlahnya. Tiap buku memiliki data nomor buku, judul, pengarang, penerbit, tahun terbit. Satu buku bisa ditulis oleh beberapa pengarang. Seorang mahasiswa boleh meminjam beberapa buku. Satu buku boleh dipinjam beberapa mahasiswa. Semua mahasiswa sangat perlu buku sehingga tidak ada yang tidak pernah meminjam ke perpustakaan. Ada buku yang sangat laris dipinjam mahasiswa, namun ada pula buku yang tidak pernah dipinjam sama sekali. Satu buku dapat memiliki beberapa *copy*, namun untuk *copy* yang sama memiliki satu nomor buku. Setiap peminjamannya. Semua mahasiswa disiplin mengembalikan buku tepat satu minggu setelah peminjaman.

Rancangan Pembangkit ERD

Rancangan pembangkit ERD dapat dilihat pada gambar 3. Tahapan dimulai dengan memasukkan spesifikasi kebutuhan teks Bahasa Indonesia kemudian melakukan *text preprocessing* yaitu *case folding*, *sentence segmentation* dan *tokenization*. Tahap selanjutnya adalah melakukan proses POS *tagging* untuk

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

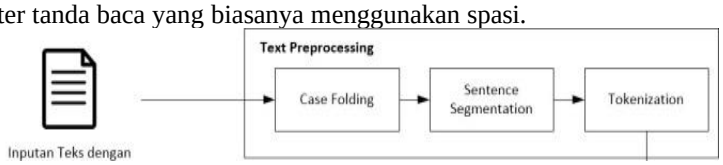
melabeli setiap kelas kata, *chunking*, dan mendeteksi komponen ERD yaitu entitas, atribut, *primary key* dan relasi. Setelah mendeteksi komponen Diagram ER, kemudian membuat gambar rancangan ERD.



Gambar 3. Proses pembangkit ERD

Text Preprocessing

Text preprocessing merupakan sebuah tahapan yang dilakukan untuk mengubah teks yang belum terstruktur menjadi teks yang sudah terstruktur dan sesuai dengan kebutuhan. Tahapan *text preprocessing* pada penelitian ini dapat dilihat pada Gambar 4. Proses pertama yang dilakukan pada *text preprocessing* yaitu *case folding* digunakan untuk mengubah semua huruf pada *file* teks menjadi huruf kecil (*lowercase*) [11]. *Case folding* dilakukan agar *file* teks yang dimasukkan dapat dideteksi atau diproses oleh sistem, jika ada kalimat yang memiliki huruf kapital maka sistem tidak dapat mendeteksi dan memproses *file* teks tersebut. *Case folding* merupakan tahap awal dalam proses pra-pemprosesan teks. Kemudian dilakukan proses *sentence segmentation* untuk memecah sebuah teks menjadi kalimat-kalimat agar lebih mudah untuk mengolah teks atau paragraf menjadi bagian-bagian yang lebih kecil [12]. *Sentence segmentation* membagi teks menjadi kalimat berdasarkan akhiran umum dalam kalimat, yaitu tanda baca titik (.), tanda seru (!), dan tanda tanya (?). Terakhir dilakukan *tokenization* untuk membagi kalimat menjadi potongan-potongan kecil atau yang disebut dengan kata atau *token*. *Tokenization* juga akan mengidentifikasi kata dan angka dalam setiap kalimat [3]. Dalam *tokenization* diperlukan pemisah kata agar tidak menimbulkan disambiguasi karakter tanda baca yang biasanya menggunakan spasi.



Gambar 4. *Text preprocessing*

POS *Tagging*

POS *tagging* adalah proses yang dilakukan untuk memberikan suatu label kelas kata untuk setiap kata dalam kalimat [13]. Hasil dari proses POS *tagging* adalah teks yang sudah memiliki *tag* untuk setiap kata. Pada POS *tagging* memiliki beberapa pendekatan yang dapat digunakan yaitu dengan menggunakan beberapa jenis pendekatan yaitu berdasarkan aturan (*rule based*), pendekatan probabilistik, dan pendekatan berbasis transformasi (*transformational based*). Dengan menggunakan pendekatan *probabilistik*, metode *Hidden Markov Model* dapat digunakan, karena proses POS *tagging* dapat dilihat sebagai sebuah proses klasifikasi suatu rangkaian *tag* pada setiap kata dalam sebuah kalimat [14].

POS *tagging* diperlukan untuk mengelompokkan setiap kata terhadap suatu kelas kata, antara lain berupa kata kerja (*verb*), kata benda (*noun*), dan kata sifat (*adjective*). Pengelompokan setiap kata ke dalam kelas kata bertujuan untuk memudahkan sistem dalam mengidentifikasi entitas, atribut dan relasi. Dalam penelitian ini tim peneliti menggunakan POS *tagger* yang ada dengan menggunakan metode Unigram. Dalam proses POS t*agging* diperlukan sebuah korpus untuk dilatih menggunakan metode yang digunakan dalam penelitian.

Tim peneliti melakukan studi literatur terhadap beberapa korpus Bahasa Indonesia yang berisi label untuk setiap kata dan berdasarkan hasil studi literatur yang dilakukan peneliti menggunakan *tagset Indonesian Manually Tagged*. Pada korpus tersebut terdapat ± 300.000 kata dan banyak digunakan oleh banyak peneliti yang membutuhkan sebuah korpus Bahasa Indonesia dalam penelitiannya. Namun isi dalam korpus jika dibandingkan dengan jumlah kata pada *vocabulary* Bahasa Indonesia masih tergolong sedikit sehingga akan berkemungkinan akan terdapat kata yang tidak terdaftar dalam korpus atau biasa disebut dengan *out of vocabulary* (OOV). Untuk menangani hal tersebut tim peneliti menggunakan teknik *backoff* yaitu memberikan *tag default* yaitu *tag* 'NN' untuk menangani OOV tersebut sehingga kata yang tidak terdapat pada korpus dapat ditangani dalam proses POS *tagging*.

Chunking

Chunking adalah proses mengambil unit informasi individu (*chunks*) dan mengelompokkannya menjadi unit yang lebih besar. Untuk memecah kalimat menjadi potongan yang lebih besar, setiap potongan sesuai dengan unit sintaksis seperti frasa kata benda/*noun phrase* (NP) atau kata kerja/*verb phrase* (VP) [15]. Pada penelitian ini tim peneliti menggunakan tahapan *chunking* karena proses ini akan membantu untuk menggabungkan dua kata atau lebih menjadi sebuah *frase* yang bermakna. Proses *chunking* didapatkan dari hasil proses POS *tagging*. Pada proses *chunking* (*phrase chunker*), kelas kata POS t*agger* akan diekstraksi menjadi enam level frasa seperti NP, VP, *Preposition Phrase* (PP), *Adverb Phrase* (ADVP), *Adjective Phrase* (AP), dan *Numerical Phrase* (NUMP) berdasarkan aturan buatan tangan. Aturan

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

buatan tangan digunakan di *regexpParser* untuk memecah kalimat berdasarkan Aturan *handmade* yang dibuat.

Deteksi Komponen ERD

Dalam perancangan ERD terdapat komponen-komponen ERD seperti: entitas, atribut, relasi, dan *primary key*. Untuk itu diperlukan suatu proses yang dapat digunakan dalam menentukan komponen-komponen ERD. Oleh karena itu, proses *parsing* dilakukan untuk mengidentifikasi bagian-bagian utama dalam kalimat seperti objek, subjek dan sebagainya. *Parsing* adalah cara untuk memetakan kalimat menjadi *parse tree*.

Pada penelitian ini, tim peneliti melakukan *parsing* dengan menggunakan *library Natural Language Toolkit* (NLTK). *Library* NLTK digunakan untuk melakukan pengolahan bahasa atau *parsing* dengan mendefinisikan infrastruktur yang dapat digunakan untuk membangun sebuah program *Natural Language Processing* (NLP) dengan menggunakan *python*. Pada proses *parsing* terdapat beberapa formula yang digunakan yaitu:

S	->	SUB PRE | SUB PRE OBJ | SUB PRE PEL | SUB PRE OBJ PEL | SUB PRE KET | SUB PRE OBJ KET | SUB PRE PEL KET | SUB PRE OBJ PEL KET
SUB	->	FNOM | PRP | NP | NNP | FNP
PRE	->	JJ | FVERB | FVB
OBJ	->	FNOM | FNP | FADVP
PEL	->	JJ | FADJE | FPREP
KET	->	FPREP
FNP	->	NP
FNP	->	VB
FNOM	->	NP VB | NP PR | NP PRP | NP NP PR | NP NP | NP JJ | CD NP | NP JJ PR | NNP NNP | CD CD | CD NND | NND NP | CD NND NP | NP CC NP | FNOM PRP | NNP CC NNP | NP ADVP | SC PR NP
FVERB	->	SC VB | MD VB | VB VB | MD RB VB | NEG VB | VB JJ | NP VB | ADVP VB
FADJE	->	JJ RB | JJ NP | NP JJ| JJ JJ | JJ VB | NEG JJ | RB JJ
FADVP	->	ADVP NP
FPREP	->	IN FPREP |IN NP | IN NP NP | IN NP IN | NP CD NP | NNP NNP | IN NNP | IN CD NP | PP

Hasil dari proses *parsing* akan menghasilkan *parse tree* menggunakan *library* NLTK, yaitu *ChartParser*. Hasil *chunking* akan digunakan untuk proses parsing. Hasil *parsing* digunakan untuk mendeteksi kata-kata yang mungkin berupa entitas, atribut, kunci utama dan relasi. Pada penelitian ini entitas yang akan di deteksi adalah entitas kuat sedangkan entitas yang lemah tidak dilakukan identifikasi oleh sistem. Dalam mendeteksi komponen-komponen ERD, peneliti menggunakan aturan atau metode berbasis aturan untuk mendeteksi kata-kata yang kemungkinan menjadi komponen ERD. *Rule based* adalah metode dimana aturan-aturan dalam sistem dibuat sendiri berdasarkan pengetahuan linguistik. Aturan untuk mendeteksi komponen ERD dalam makalah ini adalah sebagai berikut:

Aturan 1: Identifikasi Entitas

1. Kata yang memiliki *tag* NP atau kelas kata kemungkinan besar merupakan entitas.
2. Dari hasil *parsing* kata yang merupakan subjek dan objek yang memiliki *tag* NP diidentifikasi sebagai entitas.

Aturan 2: Identifikasi Atribut

1. Atribut memiliki *tag* kata NP tetapi tidak ada dalam *entity list*, sehingga kemungkinan besar merupakan atribut.
2. Kata-kata dengan *tag* NP yang terdapat dalam daftar entitas diikuti oleh VB. Untuk setiap *frase* kata benda (NP) yang tidak ada dalam daftar entitas diidentifikasi sebagai atribut.

Aturan 3: Identifikasi *Primary Key*

1. Atribut setiap entitas yang berisi kata-kata dalam daftar *primary key*, di mana daftar *primary key* berisi kata-kata unik (misalnya: no, id, kode) diidentifikasi sebagai *primary key*.

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

2. Untuk entitas yang tidak memiliki atribut atau memiliki atribut dimana atribut tersebut tidak memenuhi *primary key*, maka entitas tersebut akan memiliki atribut baru sebagai *primary key* dengan format nama atribut "id+*entity name*".

Aturan 4: Identifikasi Relasi

1. Kata-kata yang memiliki *tag* kata VB berkemungkinan menjadi relasi.
2. Kata kerja yang menghubungkan dua entitas dalam sebuah kalimat diidentifikasikan sebagai relasi.

Visualisasi ERD

Untuk memvisualisasikan komponen ERD yaitu entitas, atribut, relasi dan *primary* dibutuhkan sebuah *tools* untuk menggabungkan komponen ERD tersebut agar dapat menghasilkan model ERD dalam bentuk visual atau gambar. *Tools* yang digunakan yaitu dengan Graphviz. Graphviz merupakan perangkat lunak yang berfungsi dalam membuat visualisasi data struktural [16]. Dalam implementasi Graphviz dapat menghasilkan keluaran dengan berbagai macam format yaitu JPEG, PNG, SVG, GIF dan lainnya. Format yang menjadi keluaran sistem pembangkit ERD yaitu format PNG.

Evaluasi

Confusion matrix adalah sebuah teknik yang digunakan dalam menghitung kinerja atau tingkat kebenaran (akurasi) dari proses klasifikasi. *Confusion matrix* dapat dibuat dengan sebuah tabel yang mencakup jumlah data uji benar dan jumlah data uji yang salah diklasifikasikan. Nilai *true positive* (TP) dan *true negative* (TN) memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan untuk *false positive* (FP) dan *false negative* (FN) memberikan informasi bahwa ketika *classifier* salah dalam melakukan klasifikasi data. Terdapat beberapa formula umum yang digunakan untuk menghitung performa klasifikasi, yaitu *accuracy*, *precision*, dan *recall*, *specificity*, dan F1 *score*:

Precision adalah perbandingan dari prediksi benar positif dibagi dengan keseluruhan hasil yang diprediksi positif. Nilai dari *precision* dapat dilihat pada persamaan 1.

Precision=TP/(TP+FP) ..... (1)

Recall dapat diartikan sebagai keberhasilan model dalam mendapatkan sebuah informasi. *Recall* adalah rasio prediksi benar positif dibagi dengan keseluruhan data yang benar positif. Nilai dari *recall* dapat dilihat pada persamaan 2.

Recall=TP/(TP+FN).....(2)

F1 *Score* adalah perbandingan rata-rata *precision* dan *recall*, atau keseimbangan antara nilai precision dan recall. Nilai dari *F1* Score dapat dilihat pada persamaan 3.

F1 *Score*=2 X (*Recall* x *Precision*)/(*Recall*+*Precision*) ..... (3)

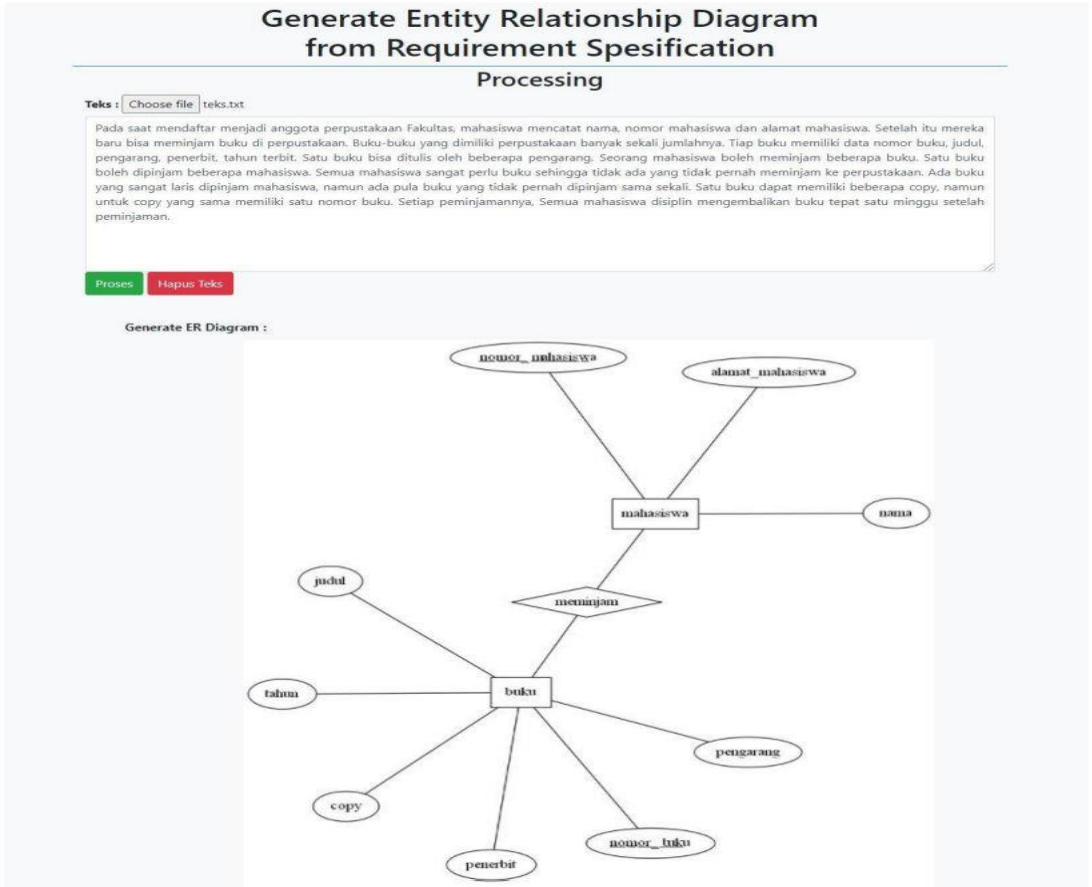
3. HASIL DAN PEMBAHASAN

Setelah mengimplementasikan metode penelitian maka hasil dari deteksi komponen ERD (entitas, atribut, *primary key*, dan relasi) diproses agar dapat menghasilkan gambar ERD untuk setiap studi kasus yang diberikan. Penelitian ini menggunakan 3 studi kasus. Hasil gambar ERD dibuat dalam bentuk berdasarkan notasi ERD pada gambar1, di mana entitas dalam bentuk persegi panjang, atribut dalam bentuk *elips*, *primary key* dalam bentuk garis bawah, serta relasi dalam bentuk belah ketupat. Gambar 5 adalah tampilan antar muka sistem untuk memasukkan inputan dari teks yang akan dibangkitkan menjadi ERD.

Gambar 5. Tampilan *interface* masukkan data

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

Hasil ERD untuk studi kasus Bahasa Indonesia dapat dilihat pada Gambar 6. Pada Gambar 6 dapat dilihat bahwa sistem telah dapat menghasilkan entitas, relasi, atribut dan *primary key*. Notasi yang dihasilkan juga telah sesuai dengan aturan notasi ERD.



Gambar 6. Hasil rancangan ERD

Pengujian

Pengujian yang digunakan yaitu pengujian *rule based* dan *expert judgement*. Berikut ini akan dibahas bagaimana hasil dari pengujian tersebut.

Pengujian *Rule Based* Deteksi Komponen ERD

Evaluasi *rule based* deteksi komponen ERD dilakukan untuk mengetahui apakah *rule based* untuk mendeteksi komponen ERD sesuai dengan hasil dari sistem. Hasil evaluasi *rule* entitas terhadap studi kasus Bahasa Indonesia dapat dilihat pada Tabel 1.

Berdasarkan tabel 1 pengujian dilakukan dengan menggunakan tiga studi kasus. Setelah melakukan evaluasi terhadap tiga studi kasus tersebut hasil yang diperoleh sudah sesuai dengan aturan atau *rule based* yang telah dibuat. ERD sudah memiliki komponen ERD yaitu entitas, attribut, *primary key* dan relasi untuk studi kasus yang sederhana sehingga semua indikator pada tabel evaluasi terpenuhi dan dapat dijadikan sebagai tolak ukur dalam penelitian ini.

Tabel 1. Evaluasi *rule* entitas studi kasus Bahasa Indonesia

Komponen	*Rule*	Studi Kasus 1	Studi Kasus 2	Studi Kasus 3
Entitas	Kata yang memiliki *tag* NP atau kelas kata kemungkinan besar merupakan entitas.	√	√	√
	Dari hasil *parsing* kata yang merupakan subjek dan objek yang memiliki *tag* NP diidentifikasi sebagai entitas.	√	√	√
Atribut	Atribut memiliki *tag* kata yaitu NP, tetapi tidak ada dalam *entity list*, sehingga kemungkinan besar merupakan atribut.	√	√	√
	Kata-kata dengan *tag* NP yang terdapat dalam daftar entitas diikuti oleh VB untuk setiap *frase* NP yang	√	√	√

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

Komponen	*Rule*	Studi Kasus 1	Studi Kasus 2	Studi Kasus 3
	tidak ada dalam daftar entitas diidentifikasi sebagai atribut.			
Primary Key	Atribut setiap entitas yang berisi kata-kata dalam daftar pk, di mana daftar *primary key* berisi kata-kata unik (misalnya: no, id, kode) diidentifikasi sebagai *primary key*.	√	√	√
	Untuk entitas yang tidak memiliki atribut atau memiliki atribut dimana atribut tersebut tidak memenuhi *primary key*, maka entitas tersebut akan memiliki atribut baru sebagai *primary key* dengan format nama atribut "id+*entity name*".	√	√	√
Relasi	Kata-kata yang memiliki *tag* kata VB (kata kerja) berkemungkinan menjadi relasi.	√	√	√
	Kata kerja yang menghubungkan dua entitas dalam sebuah kalimat diidentifikasikan sebagai relasi.	√	√	√

Pengujian *Expert Judgement* terhadap Rancangan ERD

Pada evaluasi *expert judgement* dilakukan dengan cara membandingkan antara hasil ERD yang dihasilkan oleh sistem dengan hasil rancangan oleh ahli. Dalam melakukan analisis, peneliti akan memberikan 3 studi kasus kepada para ahli untuk merancang ERD dan pada sistem studi kasus akan digunakan sebagai inputan untuk menghasilkan ERD. Kemudian dari hasil tersebut akan dilakukan analisis terhadap semua komponen ERD yaitu entitas, atribut, relasi dan *primary key*. Untuk mengukur keefektifan rancangan ERD berdasarkan studi kasus, digunakan metode *confusion matrix*. Pada tabel 2 diberikan hasil evaluasi untuk studi kasus 1, tabel 3 hasil evaluasi untuk studi kasus 2 dan tabel 4 untuk evaluasi studi kasus 3**.**

Tabel 2. Hasil evaluasi studi kasus 1

Ahli	Komponen	TP	FP	FN	*Precision*	*Recall*	F1-*Score*
Ahli 1	Entitas	2	0	2	100%	50%	67%
	Atribut	5	2	0	71%	100%	83%
	Primary Key	2	0	0	100%	100%	100%
	Relasi	1	0	2	100%	33%	50%
	Keseluruhan	10	2	4	83%	71%	77%
Ahli 2	Entitas	2	0	0	100%	100%	100%
	Atribut	7	0	0	100%	100%	100%
	Primary Key	2	0	0	100%	100%	100%
	Relasi	0	1	1	0%	0%	0%
	Keseluruhan	11	1	1	92%	92%	92%
Ahli 3	Entitas	2	0	0	100%	100%	100%
	Atribut	7	0	0	100%	100%	100%
	Primary Key	2	0	0	100%	100%	100%
	Relasi	1	0	0	100%	100%	100%
	Keseluruhan	12	0	0	100%	100%	100%
Ahli 4	Entitas	2	0	0	100%	100%	100%
	Atribut	7	0	2	100%	78%	88%
	Primary Key	2	0	0	100%	100%	100%
	Relasi	1	0	0	100%	100%	100%
	Keseluruhan	12	0	2	100%	86%	92%
Ahli 5	Entitas	2	0	2	100%	50%	67%
	Atribut	7	0	1	100%	88%	93%
	*Primary **Key***	2	0	0	100%	100%	100%
	Relasi	1	0	1	100%	50%	67%
	Keseluruhan	12	0	4	100%	75%	86%

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

Tabel 3. Hasil evaluasi studi kasus 2

Ahli	Komponen	TP	FP	FN	*Precision*	*Recall*	F1-*Score*
Ahli 1	Entitas	3	0	0	100%	100%	100%
	Atribut	12	0	0	100%	100%	100%
	Primary Key	3	0	0	100%	100%	100%
	Relasi	2	0	0	100%	100%	100%
	Keseluruhan	20	0	0	100%	100%	100%
Ahli 2	Entitas	3	0	0	100%	100%	100%
	Atribut	12	0	0	100%	100%	100%
	Primary Key	3	0	0	100%	100%	100%
	Relasi	0	2	2	0%	0%	0%
	Keseluruhan	18	2	2	90%	90%	90%
Ahli 3	Entitas	3	0	1	100%	75%	86%
	Atribut	11	1	0	92%	100%	96%
	Primary Key	3	0	0	100%	100%	100%
	Relasi	2	0	1	100%	67%	80%
	Keseluruhan	19	1	2	95%	90%	93%
Ahli 4	Entitas	3	0	0	100%	100%	100%
	Atribut	12	0	0	100%	100%	100%
	Primary Key	1	2	0	33%	100%	50%
	Relasi	2	0	0	100%	100%	100%
	Keseluruhan	18	2	0	90%	100%	95%
Ahli 5	Entitas	3	0	1	100%	75%	86%
	Atribut	12	0	0	100%	100%	100%
	Primary Key	3	0	0	100%	100%	100%
	Entitas	2	0	0	100%	100%	100%
	Atribut	20	0	1	100%	95%	98%

Tabel 4. Hasil evaluasi studi kasus 3

Ahli	Komponen	TP	FP	FN	*Precision*	*Recall*	F1-*Score*
Ahli 1	Entitas	5	0	0	100%	100%	100%
	Atribut	19	1	0	95%	100%	97%
	Primary Key	2	3	0	40%	100%	57%
	Relasi	4	0	0	100%	100%	100%
	Keseluruhan	30	4	0	88%	100%	94%
Ahli 2	Entitas	5	0	0	100%	100%	100%
	Atribut	20	0	0	100%	100%	100%
	Primary Key	5	0	0	100%	100%	100%
	Relasi	0	4	4	0%	0%	0%
	Keseluruhan	30	4	4	88%	88%	88%
Ahli 3	Entitas	5	0	1	100%	83%	91%
	Atribut	20	0	1	100%	95%	98%
	Primary Key	5	0	0	100%	100%	100%
	Relasi	4	0	1	100%	80%	89%
	Keseluruhan	34	0	3	100%	92%	96%
Ahli 4	Entitas	5	0	0	100%	100%	100%
	Atribut	19	1	2	95%	90%	93%
	Primary Key	2	3	0	40%	100%	57%
	Relasi	4	0	0	100%	100%	100%
	Keseluruhan	30	4	2	88%	94%	91%
Ahli 5	Entitas	5	0	2	100%	71%	83%
	Atribut	18	2	5	90%	78%	84%
	Primary ***Key***	5	0	0	100%	100%	100%
	Entitas	4	0	1	100%	80%	89%
	Atribut	32	2	8	94%	80%	86%

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

Pada tabel 5 dapat dilihat bahwa persentase dari rata-rata *precision*, *recall*, dan F1 *score* dari hasil perbandingan hasil antara sistem dan setiap ahli sudah memperoleh hasil baik hal ini dapat dibuktikan dari hasil masing-masing rata-rata *precision*, *recall*, dan F1 *score* hasil terendah diperoleh diatas 80%. Untuk perbedaan hasil tiap ahli berdasarkan evaluasi terhadap sistem dikarenakan masih terdapatnya entitas yang belum dapat dideteksi oleh sistem dikarenakan sistem belum menangani *generalization* pada sistem pembangkit ERD. Hal lain adalah masih terdapatnya atribut yang belum terdeteksi oleh sistem serta terdapatnya kesalahan deteksi sistem dimana kata yang seharusnya menjadi entitas namun terdeteksi sebagai atribut dalam sistem, dikarenakan sistem belum dapat menangani deteksi entitas dalam kalimat majemuk atau kompleks pada sistem pembangkit ERD. Hasil rata-rata *precision*, *recall*, dan F1 *Score* untuk setiap ahli dapa pada tabel 5.

Tabel 5. Rata-rata *precision*, *recall* dan F1 *score*

	Ahli 1	***Ahli 2***	***Ahli 3***	***Ahli 4***	***Ahli 5***	***Rata rata***
Rata-rata *Precision*	91%	90%	98%	93%	98%	93%
Rata-rata *Recall*	90%	90%	94%	93%	83%	90%
Rata-rata F1 *Score*	90%	90%	96%	93%	90%	91.8%

4. KESIMPULAN DAN SARAN

Kesimpulan

Pembangkit ERD sangat membantu *system analyst* dalam memodelkan ERD karena dapat mengurangi waktu dan beban pada fase perancangan basis data. Penelitian ini sudah berhasil membangun sistem untuk memodelkan komponen ERD yaitu entitas, atribut, *primary key* dan relasi. Berdasarkan hasil pengujian disimpulkan bahwa *rule* yang dihasilkan sudah dapat menghasilkan ERD untun studi kasus sederhana. Pengujian berdasarkan metode evaluasi *expert judgement* memperoleh hasil rata-rata *precision*, *recall*, F1 *score* berturut-turut dari tiap ahli yaitu: pada ahli 1 diperoleh 91%, 90%, 90%; pada ahli 2 diperoleh 90%, 90%, 90%; pada ahli 3 diperoleh 98%, 94%, 96%; pada ahli 4 diperoleh 93%, 93%, 93%; dan pada ahli 5 diperoleh 98%, 83%, 90%.

Saran

Untuk pengembangan selanjutnya untuk dapat melengkapi setiap komponen ERD yang belum terdapat pada penelitian ini. Selain itu perlu dilakukan pada pengembangan selanjutnya untuk dapat mendeteksi semua komponen pada ERD dan agar dapat mengatasi kalimat majemuk atau kalimat kompleks.

DAFTAR PUSTAKA

[1] P. G. T. H. Kashmira and S. Sumathipala, "Generating Entity Relationship Diagram from Requirement Specification based on NLP," *2018 3rd Int. Conf. Inf. Technol. Res. ICITR 2018*, pp. 1–4, 2018, doi: 10.1109/ICITR.2018.8736146.

[2] S. Ghosh, R. Bashar, P. Mukherjee, and B. Chakraborty, "Automated generation of e-r diagram from a given text in natural language," *Proc. - Int. Conf. Mach. Learn. Data Eng. iCMLDE 2018*, pp. 97–102, 2019, doi: 10.1109/iCMLDE.2018.00026.

[3] E. S. Btoush and M. M. Hammad, "Generating ER Diagrams from Requirement Specifications Based On Natural Language Processing," *Int. J. Database Theory Appl.*, vol. XIII, no. 2, pp. 61–70, 2015, doi: 10.14257/ijdta.2015.8.2.07.

[4] M. K. Habib, "On the Automated Entity-Relationship and Schema Design by Natural Language Processing," *Int. J. Eng. Sci.*, vol. VIII, no. 11, pp. 42–48, 2019, doi: 10.9790/1813-0811034248.

[5] "Penduduk Indonesia," 2012. https://www.bps.go.id/publication/2012/05/23/9cd01b5265c6988245eca87a/migrasi-internal-penduduk-indonesia-hasil-sensus-penduduk-2010.

[6] M. L. A. Latukolan, A. Arwan, and M. T. Ananta, "Pengembangan Sistem Pemetaan Otomatis Entity Relationship Diagram Ke Dalam Database," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. III, no. 4, pp. 4058–4065, 2019, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5117.

[7] A. S. H. F. K. S. Sudarshan, *Database System Concepts*, 6th ed. United States: Raghothaman Srinivasan, 2011.

[8] Y. A. Pratama and E. Junianto, "Sistem Pakar Diagnosa Penyakit Ginjal Dan Saluran Kemih Dengan Metode Breadth First Search," *J. Inform.*, vol. II, no. 1, 2015, doi: 10.31311/ji.v2i1.69.

[9] A. Munif, *Basis Data*. Jakarta, 2013.

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)

[10] E. Darmanto, "Analisa Perbandingan Pemodelan Basis Data Menggunakan Er- Diagram Dan Eer-Diagram Pada Kasus Sistem Asistensi Perkuliahan Praktikum," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. VII, no. 1, p. 405, 2016, doi: 10.24176/simet.v7i1.532.
[11] D. S. Indraloka and B. Santosa, "Penerapan Text Mining untuk Melakukan Clustering Data Tweet Shopee Indonesia," *J. Sains dan Seni ITS*, vol. VI, no. 2, pp. 6–11, 2017, doi: 10.12962/j23373520.v6i2.24419.
[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An introduction to natural language processing*, Third Edit. New Jersey: Prentice Hall, 2020.
[13] K. Widhiyanti and A. Harjoko, "POS Tagging Bahasa IndonesiaDengan HMM dan Rule Based," *Informatika*, vol. VIII, no. 2, 2012.
[14] I. F. Rozi, "Implementasi Rule-Based Document Subjectivity Pada Sistem Opinion Mining," *ELTEK*, vol. XI, no. 1, pp. 29–41, 2013.
[15] A. W. Syahroni and Harsono, "Aplikasi Penentuan Kategori dan Fungsi Sintaksis Kalimat Bahasa Indonesia," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. I, no. 1, 2019.
[16] Y. Setyawan, "VISUALISASI GRAF DAN ALGORITMA-ALGORITMA DALAM TEORI GRAF MENGGUNAKAN BEBERAPA PAKET SOFTWARE," *SNAST*, no. November, pp. 211–216, 2014.

ISSN:2337-7631 (printed)
ISSN: 2654-4091 (Online)