



Otomatisasi Pembentukan *Class Diagram* dengan Pendekatan Metode Pemrosesan Teks dan Algoritma CombineTF

Rosa Delima^{#1}, Antonius Rachmat Chrismanto^{#2}

[#]Prodi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana
Jl. Dr. Wahidin Sudirohusodo 5-29, Yogyakarta, Indonesia

¹rosadelima@staff.ukdw.ac.id

²anton@ti.ukdw.ac.id

Abstrak— Spesifikasi kebutuhan merupakan bagian penting dalam proses rekayasa kebutuhan perangkat lunak. Spesifikasi kebutuhan menjadi penghubung antara *system analyst* dan *programmer* yang akan melakukan pengembangan sistem. Proses rekayasa kebutuhan merupakan pekerjaan yang bersifat *time consuming* dan membutuhkan *effort* yang besar bagi analis sistem. Pekerjaan analis untuk melakukan rekayasa kebutuhan dapat lebih efisien atau lebih cepat dengan bantuan tool untuk mengotomatisasi proses rekayasa kebutuhan. Pada penelitian ini dilakukan pengembangan spesifikasi kebutuhan berupa *class diagram* secara otomatis dari data kebutuhan. Penelitian ini bermanfaat untuk membantu analis dalam melakukan spesifikasi kebutuhan. Spesifikasi kebutuhan yang dihasilkan merupakan pengembangan dari Automatic Requirements Engineering Model (AREM). Pembentukan *class diagram* dilakukan melalui tiga tahapan yaitu pembentukan *class diagram* dari data kebutuhan, penanganan duplikasi objek pada diagram, dan *refinement class diagram*. Pembentukan diagram pada tahap pertama dilakukan dengan menggunakan pendekatan pemrosesan teks, sementara itu penanganan duplikasi objek dilakukan menggunakan pendekatan *term-frequency* (TF) dan gabungan algoritma CombineTF dan Jaro-Winkler. Penelitian ini menggunakan dataset kebutuhan untuk pengembangan sistem informasi koperasi. Penelitian berhasil mengembangkan model untuk otomatisasi pembentukan *class diagram*. Hasil penelitian menunjukkan bahwa penanganan duplikasi objek pada *class diagram* mampu mengatasi 62,5% duplikasi objek dengan nilai *precision* 0,94 dan nilai akurasi 0,97 untuk nilai *threshold* algoritma ≥ 0.8 .

Kata kunci— Requirements Engineering, CombineTF, *Class Diagram*, AREM

I. PENDAHULUAN

Dalam rekayasa perangkat lunak, rekayasa kebutuhan/*requirements engineering* (RE) adalah pekerjaan awal yang sangat penting. Beberapa pekerjaan yang tercakup dalam RE adalah pengumpulan kebutuhan, analisis, permodelan, dan validasi kebutuhan [1][2]. AREM (*Automatic Requirements Engineering Model*) merupakan model yang telah dikembangkan untuk

otomatis RE. Otomatisasi RE pada AREM dapat mempermudah analis dalam melakukan rekayasa kebutuhan. AREM dikembangkan melalui pendekatan *goal-oriented*. Pendekatan *Goal-Oriented* merupakan salah satu pendekatan yang populer untuk rekayasa kebutuhan perangkat lunak [3]. AREM mampu mengotomatisasi proses RE mulai dari elisitasi, analisis, dan validasi kebutuhan [1][4][5].

Untuk dapat menjembatani antara analisis sistem dan *programmer* dibutuhkan permodelan kebutuhan dari sistem. Salah satu bentuk spesifikasi sistem yang digunakan oleh *programmer* adalah *class diagram*. *Class diagram* merupakan spesifikasi kebutuhan yang menampilkan kelas pada sistem dan asosiasi antar kelas. Setiap kelas memiliki atribut dan operasi/*method* [6]. *Class diagram* memberikan gambaran struktur dari sebuah sistem. Spesifikasi ini dapat digunakan untuk analisis proses bisnis yang terjadi pada sebuah organisasi.

Saat ini AREM telah dapat menghasilkan *class diagram* secara otomatis, namun kualitas hasil spesifikasi kebutuhan ini masih belum baik. Masih terdapat cukup banyak duplikasi objek pada *class diagram* yang dihasilkan oleh model. Oleh karena itu perlu diterapkan metode tertentu untuk mengatasi duplikasi obyek pada *class diagram* yang dihasilkan.

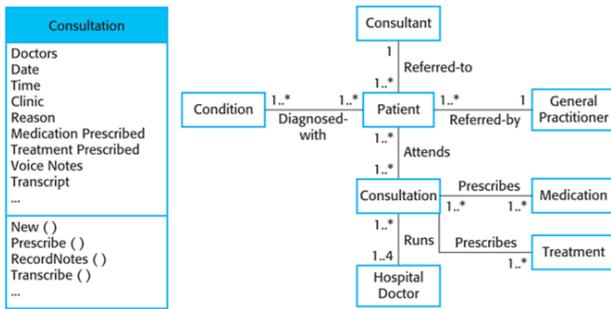
Penelitian ini berusaha memberikan kontribusi dalam mengatasi masalah yang telah disebutkan sebelumnya, dengan menggunakan gabungan metode pemrosesan teks dan metode kombinasi *term frequency* (TF). Pada penelitian ini dilakukan pembentukan *class diagram* pengembangan model AREM dengan menambahkan menggunakan pendekatan pemrosesan teks dan metode CombineTF [5]. Pendekatan pemrosesan teks digunakan untuk pembentukan awal *class diagram*. Sementara itu CombineTF digunakan untuk mengatasi duplikasi objek pada *Class Diagram*. Penelitian ini bermanfaat untuk meningkatkan efisiensi waktu bagi analis sistem untuk menghasilkan *class diagram*.

Artikel ini ditulis dalam lima bagian, dimulai dengan pendahuluan dilanjutkan dengan studi literatur, metodologi penelitian, analisis, dan diskusi, serta diakhiri dengan kesimpulan. Studi literatur berisi tentang metode-metode penelitian sejenis yang telah digunakan pada penelitian-penelitian sebelumnya. Metode penelitian berisi tentang tahapan pengembangan model yang diusulkan oleh penulis pada penelitian ini secara terperinci. Hasil, analisis, dan diskusi dijelaskan pada bagian selanjutnya secara komprehensif. Bagian akhir dari artikel ini berisi kesimpulan yang diperoleh, limitasi penelitian, dan saran untuk penelitian selanjutnya.

II. STUDI LITERATUR

A. Class Diagram

Salah satu bentuk permodelan sistem yang banyak digunakan adalah *Class diagram*. Diagram ini umum digunakan pada sistem berorientasi objek. *Class diagram* menampilkan *class* pada sistem dan hubungan antar kelas. Setiap kelas memiliki atribut dan *method* [6]. Contoh *class diagram* pada *Hospital Information System* dapat dilihat pada Gambar 1. [6].



Gambar. 1 Contoh class diagram pada *Hospital Information System*

B. Similaritas Teks

Dalam pendekatan bahasa natural (*NLP*), *Semantic text similarity* (STS) adalah aktivitas untuk menemukan nilai similaritas makna secara semantik antara dua teks. Teks yang dimaksud bisa berupa kata, frasa, maupun kalimat. STS merupakan bagian dari bidang *computational linguistics* dan *natural language understanding* [7]. Beberapa aplikasi yang mendukung STS meliputi *question answering*, *text summarization*, *plagiarism*, *sentiment analysis*, *machine translation*, dan *text information retrieval* [8]. Ontologi klinis merupakan salah satu contoh penerapan STS di bidang kesehatan [9]. Bidang lain yang juga menerapkan STS adalah bidang biomedis [10] dan pengembangan perangkat lunak [2].

Makna semantik antara dua teks pada STS bernilai sama sehingga kedua nilai tersebut dapat saling menggantikan. [11], hal ini disebut *paradigmatic*. Terdapat berbagai pendekatan untuk menguji similaritas semantik pada teks. Pendekatan yang banyak digunakan *vector space* dan *alignment vector based*. Selain itu pendekatan yang juga banyak digunakan saat ini meliputi: *machine learning based* [12], *deep learning* dengan RNN [13], CNN [14], LSTM [15], GRU [16], [17], gabungan antara LSTM-

CNN[18], [19], [20], dan Transformer (BERT) [21], [22], [23]. Sebagian besar penelitian masih menggunakan arsitektur LSTM, CNN, atau gabungannya. Perkembangan Transformer BERT dominan digunakan pada tahun 2019 atau setelahnya.

C. CombineTF

Algoritma CombineTF merupakan algoritma yang digunakan untuk mengukur similaritas dua buah teks. Algoritma ini mengkombinasikan perhitungan *term frequency* (TF) dengan metode similaritas berbasis karakter. Metode CombineTF merupakan metode *hybrid* yang menggabungkan *token-based* dan *character-based similarity*. Metode ini menyederhanakan proses perhitungan pada SoftTFIDF, di mana pada CombineTF pengukuran pertama tidak perlu dilakukan sampai perhitungan TFIDF, tetapi cukup diukur TF dan *document frequency* (DF) dari kedua *string* yang akan dihitung similaritasnya. *Document frequency* adalah banyaknya dokumen di mana *term* muncul [5].

Metode CombineTF bekerja dalam tiga tahapan yaitu pertama, menentukan *term* yang sama pada kalimat *s* dan *t*. Langkah pertama pada tahap ini adalah menggabungkan kalimat *s* dan *t*. setelah itu dilakukan perhitungan $tf_{w,s}$ dan $tf_{w,t}$ yang dilanjutkan dengan perhitungan df_w , di mana *w* adalah banyaknya frekuensi pada *term*. Perhitungan df_w dapat dilihat pada persamaan 1. Setelah menghitung df_w selanjutnya dilakukan pembentukan matrik *dic* yang berisi daftar *term* yang sama pada *s* dan *t*. parameter $|D|$ adalah banyaknya dokumen atau kalimat yang diuji. Dalam kasus pengujian similaritas dua buah kalimat maka nilai dari $|D|$ adalah 2. Langkah terakhir pada tahap pertama adalah membentuk kalimat baru ns dan nt , di mana ns dan nt merupakan *string* yang terbentuk dari *s* dan *t* setelah dikurangi dengan *dic*. Hal ini berarti ns dan nt memiliki *term* yang berbeda. Persamaan 2 dan 3 digunakan pada tahap awal ini. Pada persamaan 2, $dic(w)$ merupakan sekumpulan *term* yang sama dari *s* dan *t*. Deteksi *term* yang sama pada $dic(w)$ dideteksi melalui nilai $df_w = 2$, karena $|D|$ adalah 2 dan $dic(w)$ merupakan himpunan *multiset* dimana anggotanya dapat berulang sesuai dengan $min(tf_{w,s}, tf_{w,t})$. Selanjutnya dilakukan pembentukan kalimat baru/new sentence (ns) untuk *s* dan *t* seperti pada persamaan 3, dimana kalimat baru merupakan kumpulan *term* yang tidak sama hasil pengurangan kalimat (*s*) dengan $dic(w)$ [5].

$$df_w = |\{s \in D | w \in s\}| \tag{1}$$

$$dic(w) = \{df_w = |D| \mid m_w(\min(tf_{w,s}, tf_{w,t})) > 0\} \tag{2}$$

$$ns = s - dic(w) \tag{3}$$

Tahap kedua, melakukan penerapan perhitungan dengan *character-based similarity*, dalam penelitian ini digunakan pendekatan yaitu CombineTF-Jaro-Winkler (TF-JW). Persamaan 4 berikut, digunakan untuk proses pemeriksaan isi kalimat ns dan nt di mana $sim2$ merupakan similaritas metode yang dikombinasikan dengan CombineTF, dalam hal ini Jaro-Winkler. Pada persamaan 4. dinyatakan bahwa

$sim2_d$ dihitung jika ns dan nt tidak \emptyset , jika kedua kalimat bernilai \emptyset maka $sim2$ bernilai 1, selain kedua kondisi diatas nilai $sim2 = 0$.

$$sim2 = \begin{cases} sim2_d & \text{if } ns \neq \emptyset \text{ and } nt \neq \emptyset \\ 1 & \text{if } ns = \emptyset \text{ and } nt = \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Tahap terakhir, adalah menghitung similaritas CombineTF adalah dengan menghitung p sebagai proporsi $term$ yang sama pada kedua $string$ dan menggabungkan dengan nilai $sim2$ dengan menggunakan persamaan 5 – 8 [5].

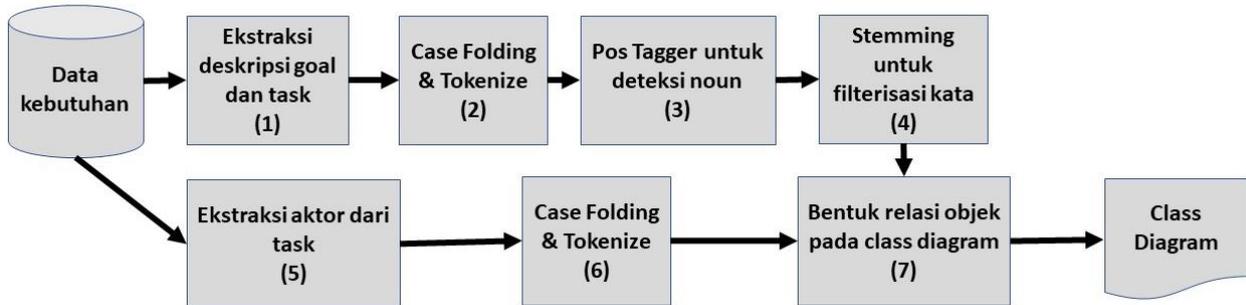
$$N = |s| + |t| \quad (5)$$

$$n = |ns| + |nt| \quad (6)$$

$$p = \frac{(N-n)}{N} \quad (7)$$

$$sim_{combineTF} = (1 \times p) + (sim2 \times (1 - p)) \quad (8)$$

Di mana, N merupakan jumlah $term$ pada kalimat s dan t (persamaan 5), dan n adalah jumlah $term$ pada ns dan nt



Gambar. 3 Tahapan pembentukan Class Diagram

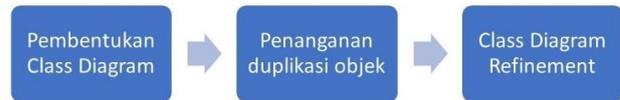
Tahapan pengembangan model untuk pembentukan $class\ diagram$ dapat dilihat pada Gambar 3. Pembentukan $class\ diagram$ dilakukan dengan menggunakan pendekatan teks. $Class\ diagram$ disusun berdasarkan data kebutuhan/requirement data. Pembentukan $class\ diagram$ diawali dengan melakukan ekstraksi objek pada data kebutuhan. Data kebutuhan merupakan data yang dihasilkan dari proses validasi kebutuhan pada AREM. Gambar 4 merupakan contoh data kebutuhan yang menjadi dataset awal penelitian.

Tercermin pada Gambar 3, ekstraksi dilakukan pada deskripsi objek $goal$ dan $task$ (tahap 1) pada data kebutuhan. Kedua jenis objek ini merupakan objek utama pada $class\ diagram$. Setelah dilakukan ekstraksi deskripsi objek selanjutnya dilakukan $case\ folding$ dan tokenisasi untuk deskripsi objek (tahap 2). $Case\ folding$ bertujuan untuk menyetarakan objek menjadi huruf kecil dan tokenisasi digunakan untuk memecah $term/kata$ dari kalimat pada deskripsi objek. Selanjutnya dilakukan $pos\ tagger$ untuk menentukan jenis kata pada objek hasil tokenisasi dan dilanjutkan dengan $stemming$ untuk memfilterisasi kata-kata umum (tahap 3 an 4).

(persamaan 6). Parameter p merupakan proporsi $term$ yang sama pada s dan t (persamaan 7), dan $sim_{combineTF}$ merupakan penjumlahan dari bagian $term$ yang sama pada kedua $string$ dan $sim2$ similaritas $term$ yang berbeda dari kedua $string$ (persamaan 8).

III. METODE PENELITIAN

Model pada penelitian ini dikembangkan melalui tiga tahapan, yaitu pembentukan $class\ diagram$ dari data kebutuhan, tahapan penanganan duplikasi objek pada $class\ diagram$, dan tahap penyusunan kembali ($refinement$) $class\ diagram$. Gambar 2 merupakan diagram untuk pengembangan sistem.



Gambar. 2 Tahapan pengembangan sistem

LIST OF SYSTEM DATA OBJECT							
ID	PROJECT ID	STAKEHOLDER	OBJECT ID	OBJECT TYPE	OBJECT DESCRIPTION	PARENT ID	PARENT TYPE
1	3	3	15	G	mempermudah menampilkan informasi mengenai koperasi	0	G
2	3	3	16	G	menampilkan anggota	15	G
3	3	3	17	G	menampilkan pinjaman anggota	15	G
4	3	3	18	G	menampilkan tabungan anggota	15	G
5	3	3	19	G	menampilkan hasil usaha koperasi	15	G
6	3	3	20	G	melakukan pendataan anggota koperasi	16	G
7	3	3	21	G	mencatat tabungan anggota koperasi	18	G
8	3	3	22	G	melakukan pendataan proses peminjaman koperasi	17	G
9	3	3	23	G	mengetahui keuangan koperasi tahun.	19	G
10	3	3	9	T	pendaftaran anggota	20	G
11	3	3	10	T	tabungan anggota	21	G
12	3	3	11	T	pinjaman anggota	22	G

Gambar. 4 Contoh requirement data

Ekstraksi kedua dilakukan terhadap aktor dari $task$, dilanjutkan dengan $case\ folding$ dan tokenisasi (tahap 5 dan 6). Kedua proses ekstraksi dipertemukan melalui pembentukan relasi objek pada $class\ diagram$ (Tahap 7),

selanjutnya ditentukan relasi *parent* dan *child* objek pada *class diagram*.

Pada tahap kedua pengembangan sistem (Gambar 2), dilakukan penanganan duplikasi objek. Pada tahap tersebut dilakukan penerapan algoritma deteksi similaritas objek. Melalui hasil pengukuran similaritas objek diketahui objek yang similar/sama. Semua objek yang sama akan dilakukan penggabungan.

Pada tahap akhir pengembangan sistem adalah melakukan *refinement* terhadap *class diagram* dengan memperbaiki relasi dari objek sebagai dampak dari penanganan duplikasi objek.

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini dikembangkan sistem yang dapat membentuk *class diagram* berdasarkan data kebutuhan yang dihasilkan AREM. Sistem ini merupakan bagian sistem yang melengkapi AREM Tool. Secara umum sistem terdiri dari tiga bagian yaitu pembentukan *class diagram* dari data kebutuhan yang dilanjutkan dengan penanganan duplikasi objek pada *class diagram*, dan terakhir dilakukan *refinement* untuk memperbaiki struktur *class diagram*.

A. Pembentukan Class Diagram

Pembentukan *class diagram* dilakukan dengan menerapkan metode pemrosesan teks. Pembentukan *class diagram* dilakukan secara otomatis berdasarkan data kebutuhan (Gambar 4) yang telah dimiliki oleh AREM. Pada penelitian ini digunakan contoh *dataset* kebutuhan dari sistem informasi koperasi. Tahapan pembentukan *class diagram* seperti terlihat pada Gambar 3. Pembentukan *class diagram* diawali dengan ekstraksi objek berdasarkan deskripsi dari elemen *goal* dan *task* pada model. Selanjutnya dilakukan pengolahan teks yang berasal dari deskripsi elemen. Pengolahan teks berupa tokenisasi, *pos tagger*, dan *stemming* (tahap 1 – 4).

Tokenisasi ditujukan untuk memisahkan kata pada deskripsi elemen. *Pos tagger* ditujukan untuk mendeteksi kata benda / *noun* pada deskripsi elemen. Kata benda menjadi objek pada *class diagram*. *Stemmer* ditujukan untuk mendapatkan kata dasar dari objek yang ekstrak. *Stemmer* juga ditujukan untuk melakukan filterisasi terhadap objek yang sama pada *class diagram*. Setelah dilakukan ekstraksi objek selanjutnya dilakukan ekstraksi terhadap aktor. Dalam proses ekstraksi aktor dilakukan tokenisasi untuk memecah jika terdapat lebih dari satu aktor untuk menjalankan satu *task* (tahap 5 dan 6). Hasil dari tahap 1 sampai 6 pada Gambar 3 adalah data objek pada *class diagram* seperti pada terlihat pada Tabel 1.

TABEL 1.

CONTOH HASIL EKSTRAKSI OBJEK DAN PEMROSESAN TEKS PADA DATA KEBUTUHAN

Objek	Tipe	ID Objek	ID Parent Objek
Pinjaman anggota	Goal	G17	G15
Tabungan anggota	Goal	G18	G15

Objek	Tipe	ID Objek	ID Parent Objek
Proses peminjaman	Goal	G22	G17
Status proses pinjaman	Goal	G33	G24
Proses pengajuan pinjaman	Task	T42	G33
Jumlah stok barang	Task	T34	G50
Simpan anggota bulan	Task	T16	T10
Anggota	Actor	A7	T16
Pegawai	Actor	A9	T16

Tahap selanjutnya pembentukan *class diagram* adalah menentukan jenis relasi dari setiap objek dan aktor pada *class diagram* (tahap 7 pada Gambar 3). Penentuan jenis relasi didasarkan pada jenis objek, jika objek adalah *actor* maka jenis relasi yang terbentuk adalah *dependency*, jika bukan *actor* maka jenis relasi adalah *generalization*.

Tahap akhir dari pembentukan *class diagram* adalah menentukan relasi antara *child* dengan *parent object*. Proses diawali dengan penentuan *child object* dari semua objek pada *class diagram*. Selanjutnya dilakukan penelusuran *parent_object* untuk *object child* yang sama. Semua *parent* ditambahkan sebagai sekumpulan *parent* pada *child object*. Proses ini berulang untuk semua objek pada *class diagram*.

Hasil dari pembentukan *class diagram* berupa objek dan relasi antar objek yang terbentuk. Informasi mencakup objek anak, tipe relasi, dan objek induk. Berdasarkan informasi ini dapat digambarkan *class diagram* dari sistem. Tabel 2 merupakan hasil pembentukan *class diagram* oleh sistem.

Tabel 2 medeskripsikan *class diagram* yang dihasilkan model untuk kasus pengembangan sistem informasi koperasi. Pada Tabel 2 dapat diketahui beberapa objek memiliki kesamaan secara semantik. Berdasarkan data pada Tabel 2 diidentifikasi terdapat delapan pasangan objek dan relasi yang sama pada *class diagram*. Daftar objek dan relasi yang similar dapat dilihat pada Tabel 3.

Tahap kedua dilakukan penanganan duplikasi objek. Penanganan duplikasi objek tahap kedua dilakukan melalui lima langkah penanganan duplikasi objek yaitu penghitungan term-frequency (TF), eliminasi kata dengan nilai TF melebihi *threshold*, pembentukan kalimat baru, eliminasi kalimat baru yang mengalami duplikasi atau bernilai null, dan uji similaritas antara dua objek yang berelasi. Tahapan penanganan duplikasi objek tahap kedua ditampilkan pada Gambar 5.

Pada tahap awal dilakukan perhitungan banyaknya frekuensi kata atau *term-frequency* (TF). Hasil perhitungan TF menjadi dasar eliminasi kata yang umum pada dataset yang dimiliki. Kata umum merupakan kata yang memiliki pengaruh minimum dari objek *class diagram*. Kata umum yang memiliki nilai *threshold* lebih dari 10% dari banyak data akan digunakan untuk melakukan *refinement* kalimat pada objek. Hasil perhitungan TF setiap kata pada objek *class diagram* dideskripsikan pada Tabel 4.

TABEL II
CLASS DIAGRAM DARI MODEL

No	Object Child	Relation Type	Object Parent
1	informasi koperasi	generalization	
2	anggota	dependency	R1 bayar angsur anggota R2 pinjam anggota R3 simpan anggota bulan R4 simpan pokok R5 daftar anggota R6 informasi koperasi
3	pinjam anggota	generalization	R1 proses pinjam koperasi R2 informasi koperasi
4	tabung anggota	generalization	R1 tabung anggota koperasi R2 informasi koperasi
5	hasil usaha koperasi	generalization	R1 informasi koperasi
6	anggota koperasi	generalization	R1 anggota
7	tabung anggota koperasi	generalization	R1 tabung anggota
8	proses pinjam koperasi	generalization	R1 pinjam anggota
9	uang koperasi tahun	generalization	R1 hasil usaha koperasi
10	daftar anggota	generalization	R1 anggota koperasi
11	hitung untung koperasi	generalization	R1 uang koperasi tahun
12	simpan pokok	generalization	R1 daftar anggota
13	simpan anggota bulan	generalization	R1 tabung anggota
14	bayar angsur anggota	generalization	R1 pinjam anggota
15	sisa pinjam	generalization	R1 pinjam anggota
16	lapor dapat koperasi	generalization	R1 hitung untung koperasi
17	lapor keluar	generalization	R1 hitung untung koperasi
18	laba koperasi	generalization	R1 hitung untung koperasi
19	besar anggota	generalization	R1 hitung untung koperasi
20	pegawai	dependency	R1 bayar angsur anggota R2 pinjam anggota R3 simpan anggota bulan R4 simpan pokok R5 anggota

TABEL III
DAFTAR OBJEK YANG SIMILAR BESERTA RELASINYA

No	Object Child	Object Parent
1	Anggota	daftar anggota
2	pinjam anggota	proses pinjam koperasi
3	tabung anggota	tabung anggota koperasi
4	anggota koperasi	anggota
5	tabung anggota koperasi	tabung anggota
6	proses pinjam koperasi	pinjam anggota
7	uang koperasi tahun	hasil usaha koperasi
8	daftar anggota	anggota koperasi

No	Kalimat	Nilai TF	Persentase
15	pegawai	2	2.82
16	pinjam	4	5.63
17	pokok	2	2.82
18	proses	2	2.82
19	simpan	3	4.23
20	Sisa	2	2.82
21	tabung	3	4.23
22	tahun	2	2.82
23	uang	2	2.82
24	untung	2	2.82
25	usaha	2	2.82
	Total	71	



Gambar. 5 Tahapan Penanganan Duplikasi Objek.

TABEL IV
HASIL PERHITUNGAN TF

No	Kalimat	Nilai TF	Persentase
1	anggota	10	14.08
2	angsur	2	2.82
3	bayar	2	2.82
4	besar	2	2.82
5	bulan	2	2.82
6	daftar	2	2.82
7	dapat	2	2.82
8	hasil	2	2.82
9	hitung	2	2.82
10	informasi	2	2.82
11	keluar	2	2.82
12	koperasi	10	14.08
13	Laba	2	2.82
14	lapor	3	4.23

Berdasarkan data pada Tabel 4, didapatkan dua kata yang memiliki nilai *threshold* lebih dari 10% yaitu “anggota” dan “koperasi”. Tahapan selanjutnya dilakukan eliminasi kedua kata tersebut dan dilakukan pembentukan kalimat/objek baru dari *class diagram*. Eliminasi kata dilanjutkan dengan proses refinement dengan menghapus objek duplikasi dan objek yang bernilai *null*. Hasil *refinement* dapat dilihat pada Tabel 5.

Penanganan duplikasi objek pada *class diagram* dilakukan melalui uji similaritas antara dua objek yang saling berelasi. Uji similaritas dilakukan dengan menerapkan kombinasi algoritma CombineTF dan Jaro-Winkler (JW) (TF-JW). Metode Jaro Winkler (JW) merupakan pengembangan dari Jaro Distance dengan penambahan perhitungan pencocokan *string* pada awalan. Metode ini memiliki akurasi yang baik untuk mendeteksi kemiripan pada dua *string* [24]. Metode ini telah diaplikasikan secara luas dan banyak dikembangkan dengan berbagai metode dan pendekatan lain. Pada

penelitian sebelumnya kombinasi CombineTF dengan JW dapat mendeteksi kemiripan data kebutuhan dari *stakeholder*, sehingga dapat menghindari duplikasi kebutuhan dan meningkatkan kualitas permodelan kebutuhan [5].

TABEL V
HASIL PEMBENTUKAN OBJEK

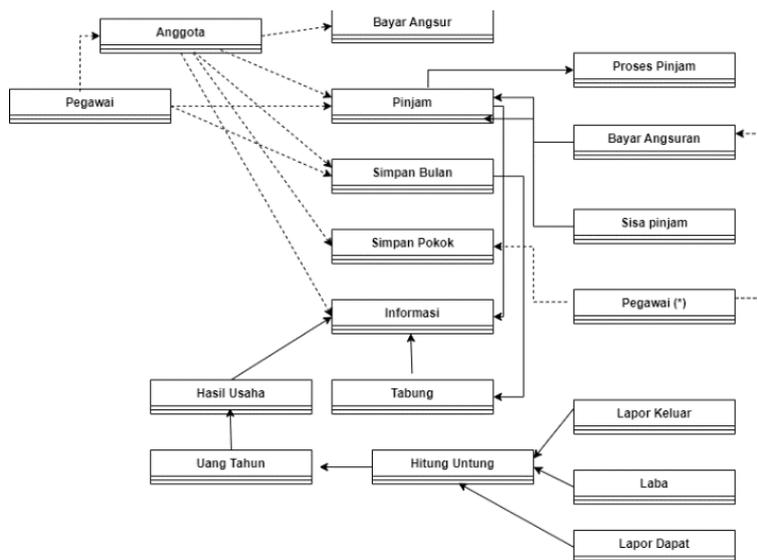
No	Objek Awal	Object Baru
1	anggota	anggota
2	bayar angsur anggota	bayar angsur
3	besar anggota	besar
4	daftar anggota	daftar
5	hasil usaha koperasi	hasil usaha
6	hitung untung koperasi	hitung untung
7	informasi koperasi	informasi
8	laba koperasi	laba
9	lapor dapat koperasi	lapor dapat
10	lapor keluar	lapor keluar
11	pegawai	pegawai
12	pinjam anggota	pinjam
13	proses pinjam koperasi	proses pinjam
14	simpan anggota bulan	simpan bulan
15	simpan pokok	simpan pokok
16	sisa pinjam	sisa pinjam
17	tabung anggota	tabung
18	uang koperasi tahun	uang tahun

Tahap kedua menghasilkan daftar objek yang similar berdasarkan perhitungan TF-JW dengan $threshold \geq 0.8$. Algoritma mampu mendeteksi dua pasangan objek yang similar dari Tabel 3, yaitu pasangan objek nomor 3,4,5,6 dan 8. Berdasarkan hasil dari tahap ini dilakukan *refinement* terhadap *class diagram* yang merupakan bagian akhir dari tahapan pembentukan *class diagram*.

Berdasarkan hasil uji coba dengan dataset kebutuhan dari sistem informasi koperasi diketahui bahwa proses penanganan duplikasi objek mampu menyelesaikan 62,5% kasus duplikasi objek. Gambar 6, merupakan bentuk grafis *class diagram* yang diterjemahkan dari hasil pembentukan *class diagram*.

B. Evaluasi Penanganan Duplikasi Obyek

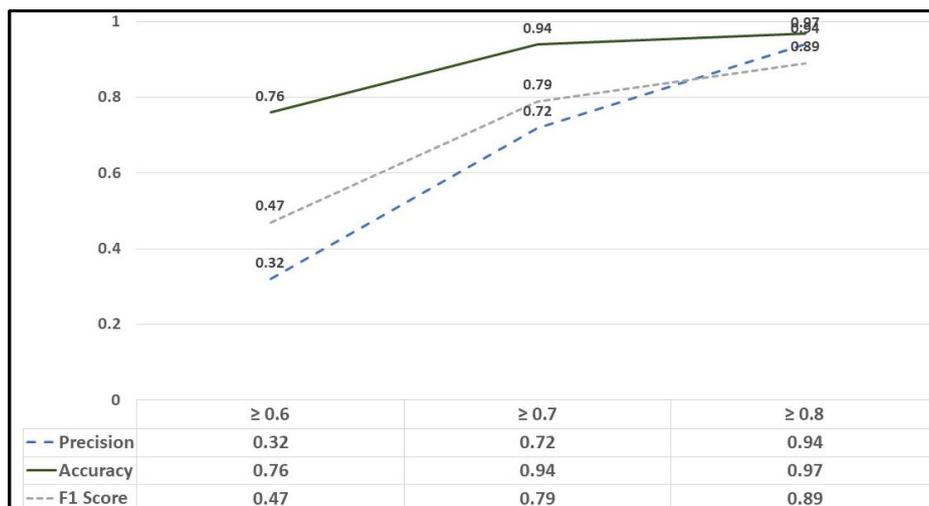
Uji similaritas menggunakan algoritma TF-JW dilakukan pada setiap relasi pada *class diagram*. Berdasarkan data pada *class diagram* pada Tabel 1 didapatkan 30 relasi objek. Setiap pasangan obyek yang berelasi diuji similitasnya dengan menggunakan tiga *threshold* yaitu 0.6, 0.7, and 0.8.



Gambar 6. Skema grafis *Class Diagram* hasil penanganan duplikasi

Hasil dari pengujian dianalisis menggunakan variabel *precision*, *accuracy*, and *F1 Score*. Gambar 6, merupakan grafik hasil pengujian penerapan algoritma. Berdasarkan hasil analisis pada Gambar 7, diketahui bahwa nilai *threshold* yang sesuai dengan kasus ini dimana *True Positive* dan *True Negative* menjadi dua variabel yang sangat penting, maka nilai *threshold* yang sesuai adalah \geq

0.8 dengan nilai *precision* 0,94 dan nilai akurasi 0,97. Hal ini berarti sistem mampu menemukan similaritas obyek dengan nilai kemiripan kedua objek tersebut ≥ 0.8 . Jika nilai kemiripan < 0.8 maka kemampuan sistem untuk mendeteksi kemiripan kedua obyek tersebut juga semakin menurun.



Gambar 7. Hasil pengujian penerapan algoritma CombineTF dengan Jaro-Winkler.

C. Keterbatasan Penelitian

Penelitian ini baru menggunakan satu dataset kebutuhan saja yaitu sistem informasi koperasi. Dibutuhkan uji coba dengan lebih banyak dataset kebutuhan agar didapat hasil penelitian yang lebih lengkap. Pada penelitian ini belum dilakukan perbandingan hasil penerapan kombinasi metode CombineTF dengan algoritma similaritas yang lain seperti *Cosine similarity* atau *Levenstain distance*. Terbuka juga penelitian dengan menggunakan metode pengukuran similaritas yang lain.

V. KESIMPULAN

Penelitian ini telah menghasilkan sebuah model yang dapat mengotomatisasi pembentukan *class diagram* pada AREM. Proses pembentukan *class diagram* terdiri dari tiga tahapan yaitu pembentukan *class diagram* dari data kebutuhan pada AREM, dilanjutkan dengan penanganan duplikasi objek pada *class diagram*, dan proses *refinement* pada *class diagram*. Penanganan duplikasi objek dilakukan dengan menerapkan kombinasi algoritma CombineTF dengan Jaro-Winkler. Penanganan duplikasi objek mampu mengatasi 62,5% duplikasi objek dengan nilai *precision* 0,94 dan nilai akurasi 0,97 untuk nilai *threshold* algoritma ≥ 0.8 .

Hasil penanganan objek masih perlu untuk ditingkatkan, namun jika dinilai dengan nilai akurasi algoritma diketahui bahwa algoritma telah berkerja dengan sangat baik. Untuk itu diperlukan pendekatan spesifik untuk meningkatkan hasil penanganan duplikasi objek. Metode yang dapat diimplementasikan pada penelitian selanjutnya adalah menggunakan pendekatan deteksi similaritas teks yang dilengkapi dengan kamus persamaan kata. Hal ini dibutuhkan karena sebagian besar kesamaan dua buah teks pada kasus yang diteliti merupakan kesamaan yang bersifat semantik.

Untuk mendapatkan hasil yang lebih lengkap dibutuhkan uji coba model pada dataset kebutuhan yang lebih banyak dan perlu dilakukan uji coba penerapan

algoritma yang berbeda untuk penanganan duplikasi objek pada *class diagram*.

UCAPAN TERIMA KASIH

Terima kasih diucapkan kepada LPPM (Lembaga Penelitian dan Pengabdian kepada Masyarakat) Universitas Kristen Duta Wacana atas dukungan dana untuk aktifitas penelitian dan publikasi artikel ini.

REFERENSI

- [1] R. Delima, R. Wardoyo, and K. Mustofa, "Automatic Requirements Engineering Model using Goal-Oriented Modelling with Text Pre-Processing Technique," in *Sixth International Conference on Informatics and Computing (ICIC)*, 2021, pp. 1–8. doi: 10.1109/icic54025.2021.9632980.
- [2] R. Delima, Khabib Mustofa, and Anny Kartika Sari, "Automatic Requirements Engineering: Activities, Methods, Tools, and Domains – A Systematic Literature Review," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 3, pp. 564–578, 2023, doi: 10.29207/resti.v7i3.4924.
- [3] R. Delima, R. Wardoyo, and K. Mustofa, "Goal-Oriented Requirements Engineering: State of the Art and Research Trend," *JUITA J. Inform.*, vol. 9, no. 1, pp. 105–114, 2021, doi: 10.30595/juita.v9i1.9827.
- [4] R. Delima and J. Purwadi, "GoEliTool for Software Requirements Elicitation using Goal-Oriented Approach," *J. Edukasi dan Penelit. Inform.*, vol. 8, no. 3, pp. 432–440, 2022.
- [5] R. Delima, R. Wardoyo, and K. Mustofa, "Combinetf for Requirements Data Similarity Detection on Arem," *ICIC Express Lett.*, vol. 16, no. 9, pp. 913–921, 2022, doi: 10.24507/icicel.16.09.913.
- [6] I. Sommerville, *Software Engineering Ninth Edition*, Ninth Edit. United States of America: Addison Wesley, 2011.
- [7] E. Agirre *et al.*, "SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, 2016, pp. 497–511. doi: 10.18653/v1/s16-1081.
- [8] R. Cheng, "Semantic Similarity Using Transformers," *Towards Data Science*, 2021.
- [9] Y. Wang *et al.*, "MedSTS: a resource for clinical semantic textual similarity," *Lang. Resour. Eval.*, vol. 54, no. 1, pp. 57–72, 2020, doi: 10.1007/s10579-018-9431-1.
- [10] G. Soğançoğlu, H. Öztürk, and A. Özgür, "BIOSSES: A semantic sentence similarity estimation system for the biomedical domain," in *Bioinformatics*, 2017, vol. 33, no. 14, pp. i49–i58. doi: 10.1093/bioinformatics/btx238.
- [11] M. A. Hadj Taieb, T. Zesch, and M. Ben Aouicha, "A survey of

- semantic relatedness evaluation datasets and procedures,” *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4407–4448, Dec. 2019, doi: 10.1007/s10462-019-09796-3.
- [12] A. R. Chrismanto, A. K. Sari, and Y. Suyanto, “Enhancing Spam Comment Detection on Social Media With Emoji Feature and Post-Comment Pairs Approach Using Ensemble Methods of Machine Learning,” *IEEE Access*, vol. 11, pp. 80246–80265, 2023, doi: 10.1109/ACCESS.2023.3299853.
- [13] B. Agarwal, H. Ramampiaro, H. Langseth, and M. Ruocco, “A deep network model for paraphrase detection in short text messages,” *Inf. Process. Manag.*, vol. 54, no. 6, pp. 922–937, 2018, doi: 10.1016/j.ipm.2018.06.005.
- [14] W. Yin, H. Schütze, B. Xiang, and B. Zhou, “ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs,” *Trans. Assoc. Comput. Linguist.*, vol. 4, pp. 566–567, Dec. 2016, doi: 10.1162/tacl_a_00244.
- [15] W. Bao, W. Bao, J. Du, Y. Yang, and X. Zhao, “Attentive Siamese LSTM Network for Semantic Textual Similarity Measure,” in *2018 International Conference on Asian Language Processing (IALP)*, Nov. 2018, pp. 312–317. doi: 10.1109/IALP.2018.8629212.
- [16] A. Godbole, A. Dalmia, and S. K. Sahu, “Siamese Neural Networks with Random Forest for detecting duplicate question pairs,” in *arXiv*, Jan. 2018, pp. 1–5.
- [17] Z. Yuan and S. Jun, “Siamese Network cooperating with Multi-head Attention for semantic sentence matching,” in *2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, Oct. 2020, pp. 215–218. doi: 10.1109/DCABES50732.2020.00068.
- [18] G. Liu and J. Guo, “Bidirectional LSTM with attention mechanism and convolutional layer for text classification,” *Neurocomputing*, vol. 337, pp. 325–338, Apr. 2019, doi: 10.1016/j.neucom.2019.01.078.
- [19] M. Mansoor, Z. ur Rehman, M. Shaheen, M. A. Khan, and M. Habib, “Deep Learning based Semantic Similarity Detection using Text Data,” *Inf. Technol. Control*, vol. 49, no. 4, pp. 495–510, Dec. 2020, doi: 10.5755/j01.itc.49.4.27118.
- [20] Y. Wang *et al.*, “An opinion spam detection method based on multi-filters convolutional neural network,” *Comput. Mater. Contin.*, vol. 65, no. 1, pp. 355–367, 2020, doi: 10.32604/cmc.2020.09835.
- [21] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3980–3990. doi: 10.18653/v1/D19-1410.
- [22] E. Fonseca and J. P. R. Alvarenga, “Wide and deep transformers applied to semantic relatedness and textual entailment,” *CEUR Workshop Proc.*, vol. 2583, no. 1, pp. 68–76, 2020.
- [23] M. Hammad, M. Al-Smadi, Q. B. Baker, and S. A. Al-Zboon, “Using deep learning models for learning semantic text similarity of Arabic questions,” *Int. J. Electr. Comput. Eng.*, vol. 11, no. 4, p. 3519, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3519-3528.
- [24] W. E. Winkler, “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage,” 1990. [Online]. Available: <http://files.eric.ed.gov/fulltext/ED325505.pdf>