

Metrik Kompleksitas Desain *Database* Relasional

Taufiqurrahman Idrus, Muhammad Ainul Yaqin, Layla Qomariah, Rahma Nur Ardinna Putri
Jurusan Teknik Informatika, UIN Maulana Malik Ibrahim Malang
17650088@student.uin-malang.ac.id, yaqinov@gmail.com, 17650048@student.uin-malang.ac.id,
17650076@student.uin-malang.ac.id

Info Artikel

Riwayat Artikel:

Diterima: 21 Oktober 2019
Direvisi: 18 November 2019
Diterbitkan: 15 Januari 2020

Kata Kunci:

Database
Kompleksitas
Metrik
Relasional

ABSTRAK

Desain *database* relasional semakin kompleks dari waktu ke waktu. Maka dari itu dibutuhkan sebuah formula yang dapat mengukur kompleksitas *database* relasional secara komprehensif dan sensitif terhadap perubahan kecil agar kualitas dan pemeliharannya dapat lebih terjaga. Tujuan paper ini adalah untuk mengajukan sebuah formula kompleksitas yang dapat mengukur tingkat kompleksitas sebuah *database* relasional secara komprehensif dan sensitif terhadap perubahan kecil. *Paper* ini menggunakan metode GQM untuk identifikasi metriknya. Paper ini mengidentifikasi bahwa metrik untuk sebuah *database* relasional meliputi tabel, relasi antar tabel, kolom, *primary key*, *secondary key*, *foreign key*, *index*, *cohesion of schema*, *depth referential tree*, tipe data dan tipe relasi antar tabel. Dari beberapa formula kompleksitas yang sudah pernah diajukan, kami menambahkan perhitungan terhadap jumlah tabel, jumlah relasi antar tabel, tipe data dan tipe relasi. Formula kompleksitas kami direpresentasikan dalam model *tree*. Sedangkan metode formulasinya digunakan metode AHP. Tahap terakhir adalah validasi menggunakan Weyuker *Properties*. Penelitian ini menghasilkan sebuah formula kompleksitas *database* relasional seperti pada persamaan (1). Setelah dilakukan validasi dengan Weyuker's *Properties* terdapat 6 dari 9 properti yang terpenuhi.

Copyright © 2019 SIMANIS.
All rights reserved.

Korespondensi:

Taufiqurrahman Idrus,
Jurusan Teknik Informatika,
UIN Maulana Malik Ibrahim Malang,
Jl. Gajayana No. 50 Malang, Jawa Timur, Indonesia 65144
17650088@student.uin-malang.ac.id

1. PENDAHULUAN

1.1. Latar Belakang

Database adalah kumpulan *file-file* yang saling berelasi, relasi tersebut biasa ditunjukkan dengan *key* (kunci) dari tiap *file* yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan, instansi [1]. Jenis *database* yang umum digunakan saat ini adalah *database* relasional/SQL dan *database* non-relasional/NoSQL. *Database* relasional adalah *database* yang data-datanya tersimpan dalam tabel-tabel dan terdapat relasi antar tabel. Aspek-aspek yang terdapat dalam *database* relasional meliputi: Tabel, Kolom, Tipe data, Lebar data, *Primary Key*, *Secondary Key*, *Foreign Key*, *index*, Relasi dan jenis relasi. Pengukuran pada proses pengembangan perangkat lunak dan produknya dilakukan dengan menerapkan 'metrik' perangkat lunak tertentu. Pengukuran biasanya terdiri dari beberapa metrik dan menghasilkan beberapa hasil pengukuran per metrik [2]. Pengukuran perangkat lunak dapat membantu kita memahami, mengontrol, dan mengembangkan perangkat lunak [3].

Sama seperti sebuah program, sebuah *database* relasional perlu diukur juga tingkat kompleksitasnya. Sayangnya, Hampir semua pengukuran perangkat lunak yang ada hanya terfokus pada karakteristik program

saja, contohnya *cyclomatic number* [4] dan mengabaikan *database* dari programnya [5]. Pada Sistem informasi Modern, *database* telah menjadi bagian yang krusial. Jadi, diperlukan sebuah pengajuan dan penelitian sebuah pengukuran untuk menilai kualitas suatu *database* [6]. *Database* menjadi semakin kompleks dari waktu ke waktu sehingga dibutuhkan sebuah metrik kompleksitas untuk memahami, memantau, mengontrol, memprediksi dan meningkatkan pengembangan *database* dan perawatan terhadap proyek [6].

Untuk menciptakan sebuah formula kompleksitas *database* yang komprehensif dan sensitif terhadap perubahan kecil harus menyertakan setiap metrik untuk *database* relasional. Beberapa formula kompleksitas untuk *database relasional* sudah pernah diajukan sebelumnya. Namun, ada beberapa metrik yang belum disertakan. Sehingga, saat digunakan untuk mengukur kompleksitas *database* relasional yang strukturnya hampir identik hasil yang diperoleh bisa bernilai sama. Maka, disini kami akan menambahkan metrik yang belum disertakan pada formula kompleksitas yang sudah pernah diajukan sebelumnya. Sehingga, tingkat kompleksitas *database* relasional dapat diukur secara lebih komprehensif dan lebih sensitif terhadap perubahan kecil.

Beberapa tahap dalam mewujudkan tujuan paper ini adalah identifikasi metrik, representasi metrik, formulasi metrik dan validasi metrik. Metode identifikasi yang digunakan adalah metode *Goal Question Metric* (GQM). Paper ini mengidentifikasi bahwa metrik untuk sebuah *database* relasional meliputi tabel, relasi, kolom, *primary key*, *secondary key*, *foreign key*, *index*, *cohesion of schema*, *depth referential tree*, tipe data dan tipe relasi. Dari beberapa formula kompleksitas yang sudah pernah diajukan, kami menambahkan perhitungan terhadap jumlah tabel, jumlah relasi, tipe data dan tipe relasi. Hasil paper ini kemudian direpresentasikan dalam model Tree. Untuk formulasinya, Paper ini menggunakan metode *Analytical Hierarchy Process* (AHP). Tahap terakhir adalah validasi metrik menggunakan *Weyuker's Properties*.

1.2. Tinjauan Ulang Formula Kompleksitas yang Sudah Pernah Diajukan Sebelumnya

Calero dkk. Merepresentasikan skema *database* relasional ke dalam model *graph*, kemudian nilai kompleksitasnya diukur berdasarkan: Jumlah tabel keseluruhan, panjang jalur referensial terpanjang (*Depth Referential Tree*), jumlah *foreign key*, dan *cohesion of schema* (COS), yakni jumlah keseluruhan akar kuadrat dari jumlah tabel pada setiap *subgraph* yang tidak saling terhubung [6].

Pavlic dkk. Menyebut formulasinya sebagai *Database Complexity* (DC). Pertama-tama dicari nilai kompleksitas untuk sebuah tabel dengan menjumlahkan: Jumlah atribut, jumlah *primary key* dan *secondary key*, jumlah *index*, dan jumlah *foreign key*. Setelah itu, nilai kompleksitas semua tabel dijumlahkan untuk memperoleh nilai kompleksitas *database* [7].

Perbandingan metrik yang disertakan pada formula kompleksitas yang akan kami ajukan dengan formula kompleksitas dari peneliti lain dapat dilihat pada tabel (1).

Tabel 1. Metrik-metrik yang telah disertakan dalam formula kompleksitas

	T	R	C	PK	FK	SK	I	DT	DW	RT
Calero dkk., 2001	-	-	✓	-	✓	-	-	-	-	-
Pavlic dkk., 2008	-	-	✓	✓	✓	✓	✓	-	-	-
formula kompleksitas yang akan kami ajukan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Dimana T adalah Tabel. R adalah Relasi. C adalah Kolom. PK adalah *primary key*. FK adalah *Foreign key*. SK adalah *Secondary key*. I adalah *Index*. DT adalah Tipe data. DW adalah Lebar data. RT adalah Tipe relasi. Dari tabel (1) terlihat bahwa formula kompleksitas yang akan kami ajukan menyertakan metrik yang telah disertakan dan yang tidak disertakan dari formula kompleksitas peneliti lain.

2. METODE PENELITIAN

2.1. Identifikasi Metrik Menggunakan GQM

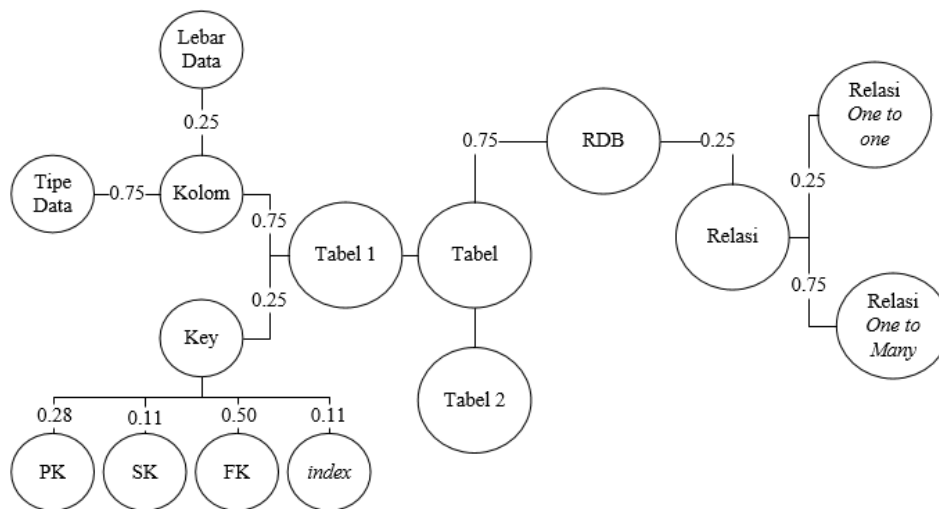
Hasil identifikasi memperoleh bahwa metrik yang akan digunakan adalah tabel, kolom, tipe data, *primary key*, *foreign key*, *secondary key*, *index*, relasi dan tipe relasi. Proses identifikasi metrik dapat dilihat pada tabel (2).

Tabel 2. Identifikasi Metrik Menggunakan GQM

<i>Goal</i>	Tujuan	Mengajukan
	Isu	Formula yang komprehensif dan sensitif terhadap perubahan kecil
	Objek	Formula kompleksitas <i>database</i> relasional
	Sudut Pandang	Manajer Proyek
<i>Question Metrics</i>		Apa yang membentuk sebuah <i>database</i> relasional? Tabel, relasi
<i>Question Metrics</i>		Apa yang membentuk sebuah tabel? Baris, kolom
<i>Question Metrics</i>		Apa yang ada pada kolom? tipe data, lebar data, <i>primary key</i> , <i>foreign key</i> , <i>secondary key</i> , <i>index</i> ,
<i>Question Metrics</i>		Apa saja jenis relasi? <i>One to one</i> , <i>one to many</i>
<i>Question Metrics</i>		Apa saja metrik <i>database</i> relasional yang pernah diajukan?
<i>Metrics</i>		Kolom, <i>primary key</i> , <i>foreign key</i> , <i>secondary key</i> , <i>index</i> , <i>Cohesion of Schema</i> , <i>Depth Referential Tree</i>
<i>Question Metrics</i>		Apa saja metrik <i>database</i> relasional yang belum diajukan?
<i>Metrics</i>		Tabel, tipe data, lebar data, relasi, jenis relasi

2.2. Representasi Metrik Dalam Model Tree

Metrik yang telah diidentifikasi selanjutnya direpresentasikan menggunakan model *tree*. *Node* yang pada *tree* merepresentasikan metrik. Sedangkan *edge* merepresentasikan bobot dari perhitungan AHP. Hasil representasi disajikan pada gambar (1).



Gambar 1. Representasi metrik dalam model *tree*

2.3. Formulasi Metrik Menggunakan metode AHP

Dengan menggunakan metode AHP, diperoleh bobot untuk setiap metrik dan tipe data yang ada pada *database* relasional. Data yang dipakai dalam perhitungan AHP adalah penilaian subjektif penulis terhadap tingkat kepentingan pada setiap metrik *database* relasional. Bobot untuk metrik disajikan pada tabel (3) dan bobot untuk tipe data disajikan pada tabel (4).

Tabel 3. Bobot setiap metrik *database* relasional

	Metrik	Bobot
A.	Tabel	0.75
B.	Relasi	0.25
C.	Kolom	0.75
D.	Key	0.25
E.	Tipe Data	0.75
F.	Lebar Data	0.25
G.	Primary Key	0.28
H.	Secondary Key	0.11
I.	Foreign Key	0.50
J.	Index	0.11
K.	Relasi one to one	0.25
L.	Relasi one to Many	0.75

Tabel 4. Bobot untuk tipe data numerik dan tanggal/waktu

Tipe Data	Bobot
<i>Bit</i>	0.01
<i>Tinyint, year</i>	0.06
<i>Smallint</i>	0.11
<i>Char, binary, tinytext, tinyblob</i>	0.12
<i>Text, blob, varchar, varbinary</i>	0.16
<i>Mediumint, time</i>	0.17
<i>Int, float, timestamp</i>	0.22
<i>Mediumtext, mediumblob</i>	0.24
<i>Bigint, double, decimal, real, numeric, datetime, time</i>	0.44
<i>Longtext, longblob</i>	0.48

2.4. Validasi Metrik dengan *Weyuker's Properties*

Weyuker [8] mengajukan sembilan properti untuk validasi pengukuran perangkat lunak pada tahun 1988, sembilan properti ini dinamakan *Weyuker's Properties*. Pada paper ini kami akan melakukan validasi terhadap formula kompleksitas kami dengan *Weyuker's Properties*. Adapun untuk studi kasus dilakukan pengukuran terhadap tiga *database* relasional.

3. HASIL DAN PEMBAHASAN

3.1. Bentuk Formula Kompleksitas

$$DC = 0.75. \Sigma TC + 0.25. \Sigma RC \quad (1)$$

$$TC = 0.75. \Sigma ColC + 0.25. \Sigma KeyC \quad (2)$$

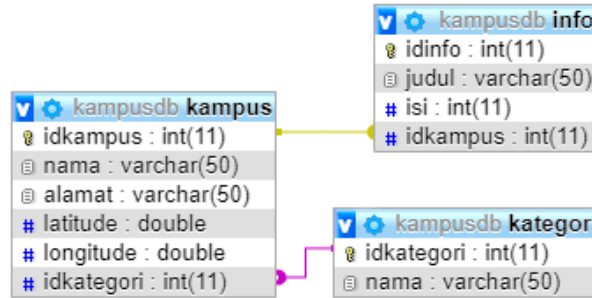
$$ColC = 0.75. DT + 0.25. DW \quad (3)$$

$$KeyC = (0.28. PK + 0.11. SK + 0.50. FK + 0.11. Index) \quad (4)$$

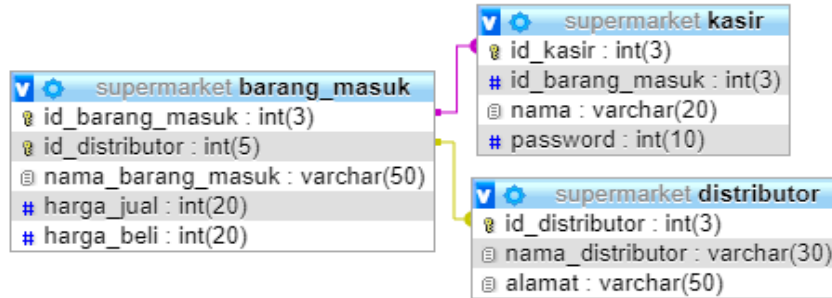
$$RC = 0.25. \Sigma R1:1 + 0.75. \Sigma R1:M \quad (5)$$

Dimana DC adalah *Database Complexity* (kompleksitas *database*). TC adalah *Table Complexity* (kompleksitas tabel). ColC adalah *Column Complexity* (kompleksitas kolom). KeyC adalah *Key Complexity* (kompleksitas key). RC adalah *Relation Complexity* (Kompleksitas relasi).

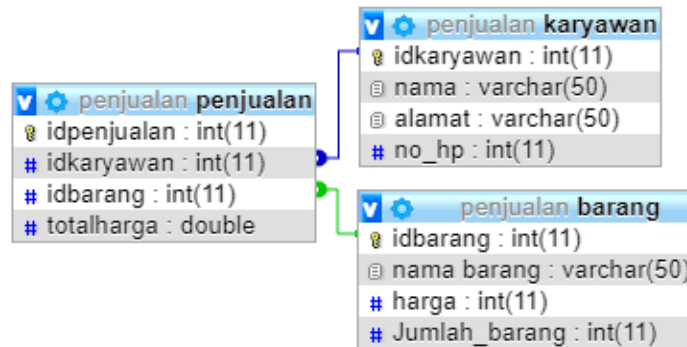
3.2. Studi Kasus *Database* Relasional



Gambar 2. Database Relasional P



Gambar 3. Database Relasional Q



Gambar 4. Database Relasional R

3.3. Hasil Pengukuran dan Pembahasan

Perbandingan hasil pengukuran disajikan pada tabel (5).

Tabel 5. Hasil pengukuran menggunakan formula kompleksitas

	P	Q	R	P;Q	P + Q
Tabel	3	3	3	6	6
Kolom	12	12	12	24	24
Relasi	2	2	2	4	4
PK	3	3	3	6	6
SK	0	0	0	0	0
FK	2	2	2	4	4
Index	0	0	0	0	0
Tipe Data	2.82	2.4	2.68	5.24	5.24
Lebar Data	266	217	238	483	483
Tipe Relasi	1.5	1.5	1.5	3	3
Calero dkk., 2001	24	24	24	38	38
Pavlic dkk., 2008	17	17	17	34	34
formula kompleksitas yang akan kami ajukan	39.04	31.97	35.04	71.01	71.01

Dari tabel (5) dapat dilihat hasil pengukuran dengan nilai yang sama untuk ketiga *database* relasional jika menggunakan dua formula kompleksitas yang sudah pernah diajukan sebelumnya. Sementara jika menggunakan formula yang kami usulkan diperoleh hasil yang berbeda-beda untuk ketiga RDB. Hal ini karena belum disertakannya bobot untuk setiap tipe data dan juga bobot untuk tipe relasi beserta perhitungan terhadap lebar data. Sehingga, jika tipe data atau lebar data pada sebuah kolom diubah. Maka, nilai kompleksitas akan ikut berubah juga. Dimana jika menggunakan dua formula kompleksitas lainnya, nilai kompleksitas yang diperoleh akan tetap sama.

Langkah terakhir adalah validasi dengan *Weyuker's Properties*. Hasil validasi dapat dilihat pada tabel (6). Berikut adalah pembahasan validasinya

Properti 1 : *Non-coarseness* : $(\exists P)(\exists Q);(|P| \neq |Q|)$. Metrik kompleksitas dapat membedakan dua *database* relasional yang berbeda karena nilai kompleksitas keduanya berbeda. Metrik kompleksitas yang kami usulkan dapat membedakan dua *database* relasional yang berbeda, hal ini dapat dilihat pada tabel (5) di mana setiap *database* relasional yang menjadi studi kasus memiliki metrik kompleksitas yang berbeda. Dengan demikian properti 1 telah dipenuhi oleh metrik kompleksitas kami.

Properti 2 : *Granularity* : Metrik kompleksitas harus non-negatif. Pada tabel (5) menampilkan nilai metrik kompleksitas kami yang selalu positif, dan jika dilihat dari persamaan yang kami usulkan maka nilai metrik kompleksitasnya tidak akan pernah negatif. Dengan demikian properti 2 telah dipenuhi oleh metrik kompleksitas kami.

Properti 3 : *Non-uniqueness* : $(P \neq Q);(|P| = |Q|)$. Pada beberapa kasus dua *database* relasional yang berbeda dapat memiliki metrik kompleksitas yang sama. Dengan formula metrik kompleksitas yang kami usulkan akan sulit menemukan model *database* relasional yang berbeda dengan metrik yang sama. Hal ini disebabkan kami mengidentifikasi banyak desain *database* relasional yang bisa dibentuk, sehingga setiap *database* relasional akan memiliki metrik kompleksitas yang unik. Dengan demikian properti 3 tidak dipenuhi oleh metrik kompleksitas kami.

Properti 4 : *Design Implication* : $(\exists P)(\exists Q);(P \equiv Q \ \& \ |P| \neq |Q|)$. Metrik kompleksitas harus dapat membedakan dua *database* relasional yang memiliki jumlah tabel dan kolom yang sama tetapi dengan tipe data yang berbeda beda. Dari tabel (5), Metrik kompleksitas yang kami usulkan dapat membedakan dua *database* relasional yang memiliki jumlah tabel dan kolom yang sama tetapi dengan tipe data yang berbeda beda. Dengan demikian properti 4 telah dipenuhi oleh metrik kompleksitas kami.

Properti 5 : *Monotonicity* : $(\forall P)(\forall Q);(|P| \leq |P;Q| \ \& \ |Q| \leq |P;Q|)$. Metrik kompleksitas harus dapat menunjukkan adanya penggabungan beberapa elemen baru pada suatu *database* relasional. Metrik kompleksitas yang kami usulkan dapat menunjukkan adanya penggabungan elemen baru pada suatu *database* relasional. Berdasarkan tabel (5), jika *database* relasional P dan Q digabungkan akan menghasilkan metrik kompleksitas sebesar 71.01, Angka tersebut lebih besar daripada kompleksitas *database* relasional P sebesar 39.04 dan *database* relasional Q sebesar 31.97. Dengan demikian properti 5 telah dipenuhi oleh metrik kompleksitas kami.

Properti 6 : *Non-equivalence of Interaction*:

$$6a. (\exists P)(\exists Q)(\exists R); (|P| = |Q| \ \& \ |P;R| \neq |Q;R|)$$

$$6b. (\exists P)(\exists Q)(\exists R); (|P| = |Q| \ \& \ |R;P| \neq |R;Q|)$$

Metrik kompleksitas harus dapat membedakan dua *database* relasional yang identik. Ketika ada 3 *database* relasional P, Q, dan R dengan nilai metrik yang sama, kemudian *database* relasional P digabung dengan R atau *database* relasional Q digabung dengan R maka nilai metrik keduanya seharusnya tidak sama. Pada formula kami sangat sulit untuk menemukan dua *database* relasional yang mempunyai nilai metrik yang sama. Oleh karena itu, penyambungan model RDB P dan RDB R atau RDB Q dan RDB R pastilah menghasilkan nilai metrik yang tidak sama. Dengan demikian properti 6 telah dipenuhi oleh metrik kompleksitas kami.

Properti 7 : *Significance of Permutation* : $(\exists P)(\exists Q);(|P| \neq |Q|)$. Jika Q dibentuk oleh permutasi tipe data P nilai kompleksitasnya akan berubah. Pada metrik kompleksitas yang kami usulkan pertukaran tipe data dalam tabel yang tidak berpengaruh pada nilai kompleksitas karena pengukuran terhadap tipe data dilakukan dalam 1 tabel yang sama, hasilnya akan berbeda jika dilakukan pertukaran dengan tipe data yang ada pada tabel lain. Dengan demikian properti 7 tidak dipenuhi oleh metrik kompleksitas kami.

Properti 8 : *No change on renaming*. Metrik kompleksitas tidak akan berubah jika terjadi perubahan nama pada elemen *database* relasional. Metrik kompleksitas yang kami usulkan, mendasarkan perhitungan pada struktur *database* relasional sehingga perubahan pada nama tidak mempengaruhi metrik kompleksitas. Dengan demikian properti 8 telah dipenuhi oleh metrik kompleksitas kami.

Properti 9 : *Interaction Complexity* : $(\exists P)(\exists Q);(|P| + |Q| < |P;Q|)$. Metrik kompleksitas gabungan beberapa *database* relasional harus lebih besar daripada sebelum digabung. Dari tabel (5) diperoleh bahwa

nilai kompleksitas gabungan P ditambah Q sama dengan nilai kompleksitas saat *database* relasional P dan Q digabungkan. Dengan demikian properti 9 tidak dipenuhi oleh metrik kompleksitas kami.

Tabel 6. Hasil validasi menggunakan *Weyuker's Properties*

Properti	Terpenuhi
1 <i>Non-Coarsenes</i>	✓
2 <i>Granularity</i>	✓
3 <i>Non Uniqueness</i>	-
4 <i>Design Implication</i>	✓
5 <i>Monotonicity</i>	✓
6 <i>Non-equivalence of Interaction</i>	✓
7 <i>Significance of Permutation</i>	-
8 <i>No Change on Renaming</i>	✓
9 <i>Interaction Complexity</i>	-

Setelah dilakukan Validasi Metrik menggunakan *Weyuker's Properties*, diperoleh 6 dari 9 properti yang terpenuhi.

4. KESIMPULAN

Paper ini telah mengidentifikasi bahwa metrik untuk sebuah *database* relasional adalah: jumlah tabel, jumlah kolom, Tipe data, Lebar data, *Primary Key*, *Secondary Key*, *Foreign Key*, *index*, Relasi dan jenis relasi. Formula kompleksitas pada paper ini dapat direpresentasikan menggunakan model *tree*. bentuk formula kompleksitas dapat dilihat pada persamaan (1). Setelah dilakukan pengujian dengan membandingkan hasil pengukuran yang diperoleh menggunakan formula kompleksitas dari peneliti lain, ditemukan bahwa formula kompleksitas yang kami ajukan lebih komprehensif dan lebih sensitif terhadap perubahan kecil. Hasil validasi dengan *Weyuker's Properties* diperoleh dari 9 properti terpenuhi.

5. UCAPAN TERIMAKASIH

Ucapan terima kasih kami sampaikan kepada kawan-kawan di jurusan Teknik Informatika UIN Maulana Malik Ibrahim yang mendukung kelancaran berlangsungnya penelitian ini.

DAFTAR PUSTAKA

- [1] Ir. Harianto Kristanto, *Konsep & Perancangan Database*. Penerbit Andi, 1993.
- [2] R. Van Solingen, V. Basili, G. Caldiera, and H. D. Rombach, "Goal question metric (gqm) approach," *Encycl. Softw. Eng.*, 2002.
- [3] N. Fenton and J. Bieman, *Software metrics: a rigorous and practical approach*. CRC press, 2014.
- [4] T. J. McCabe, "A Complexity Measure," *IEEE Trans. Softw. Eng.*, vol. SE-2, no. 4, pp. 308–320, 1976.
- [5] H. M. Sneed and O. Foshag, "Measuring Legacy Database Structures," in *Software Measurement*, Springer, 1999, pp. 143–158.
- [6] C. Calero, M. Piattini, and M. Genero, "Metrics for controlling database complexity," in *Developing Quality Complex Database Systems: Practices, Techniques and Technologies*, IGI Global, 2001, pp. 48–68.
- [7] M. Pavlić, M. Kaluža, and N. Vrček, "Database Complexity Measuring Method," in *Proceedings of the 19th Central European Conference on Information and Intelligent Systems*, 2008, p. 1.
- [8] E. J. Weyuker, "Evaluating Software Complexity Measures," *IEEE Trans. Softw. Eng.*, vol. 14, no. 9, pp. 1357–1365, 1988.