

Pemodelan Data Relasional pada NoSQL Berorientasi Dokumen

Muhammad Riza Alifi¹, Transmissia Semiawan^{2*}, Djoko C.U. Lieharyani³, Hashri Hayati⁴

Intisari—Teknologi manajemen data yang terus berkembang telah mendorong popularitas *not only structured query language* (NoSQL) berorientasi dokumen menjadi salah satu model data yang paling banyak digunakan saat ini. Selain popularitas tersebut, terdapat keunggulan menarik, yaitu kemampuannya menawarkan fleksibilitas penyimpanan data dalam berbagai bentuk dan ukuran, baik untuk data terstruktur maupun tidak terstruktur. Namun, fleksibilitas model data tersebut menjadi tantangan tersendiri karena dapat berdampak pada pembentukan skema yang lebih kompleks jika tidak mempertimbangkan pola desain yang sesuai dengan kebutuhan. Studi ini bertujuan untuk melakukan pemodelan data relasional pada NoSQL berorientasi dokumen pada tingkat konseptual, logis, dan fisik. Rancangan tingkat konseptual dibentuk berdasarkan proses, aturan, dan kebutuhan bisnis. Sementara itu, rancangan tingkat logis dan fisik dibentuk berdasarkan pola desain *extended reference* dan *computed* yang ditentukan dari beban kerja operasi. Rancangan model data relasional pada NoSQL berorientasi dokumen berhasil dibentuk menggunakan *entity relationship diagram* (ERD), dengan notasi Chen untuk tingkat konseptual dan CRD baik untuk tingkat logis maupun fisik. Fokus rancangan tingkat konseptual terletak pada representasi entitas, atribut, dan relasi. Berbeda dengan tingkat konseptual yang cenderung abstrak, fokus rancangan tingkat logis terletak pada representasi skema *collection* (*embedded* dan *reference*), termasuk pola desain yang dipengaruhi oleh pembentukan relasi. Selanjutnya, fokus rancangan tingkat fisik adalah merepresentasikan skema ke dalam bentuk yang lebih konkret. Rancangan tingkat fisik hampir sama dengan tingkat logis. Perbedaannya hanya terletak pada penambahan detail untuk tipe data dan struktur data. Evaluasi rancangan model data dilakukan menggunakan daftar tilik pemeriksaan untuk setiap tingkatan. Studi ini berkontribusi untuk merancang model data dengan keunggulan kinerja membaca yang lebih baik (*read-intensive*) karena tidak perlu melakukan operasi gabungan (*join*) antar-*collection* yang memiliki relasi dan tidak perlu melakukan pengulangan proses komputasi untuk atribut turunan.

Kata Kunci—NoSQL Berorientasi Dokumen, Pemodelan Data, *Extended Reference*, *Computed*.

I. PENDAHULUAN

Perkembangan manajemen data dalam suatu basis data (*database*) saat ini harus mampu menjawab tantangan *big data*.

^{1,2,3,4} Jurusan Teknik Komputer & Informatika, Politeknik Negeri Bandung, Jl. Gegerkalong Hilir, Ds. Ciwaruga, Kabupaten Bandung, Kode Pos 40559, Jawa Barat, INDONESIA (telp: 022-2013789; faks: 022-2013889; email:

¹muhammad.riza@polban.ac.id, ²t.semiawan@polban.ac.id,

³djoko.cahyo@polban.ac.id, ⁴hasri.hayati@polban.ac.id)

*Corresponding Author

[Diterima: 28 Desember 2021, Revisi: 8 Juli 2022]

Tantangan “5V” dikenal mendominasi praktik dan penelitian *big data*, yaitu *volume*, *velocity*, *variety*, *veracity*, dan *value* [1]. Ada beberapa isu yang terkait dengan tantangan manajemen data, di antaranya adalah pertimbangan lokasi penyimpanan data yang terpusat (*centralized*) atau terdistribusi (*distributed*); penggunaan metode pengaksesan data yang berdasarkan perolehan informasi (*information retrieval*) atau perolehan data (*data retrieval*); dan penentuan operasi pencarian data menggunakan pencocokan pola (*pattern matching*) atau pencocokan *string* yang tepat (*exact string matching*).

Pemodelan data (*data modeling*) merupakan suatu landasan untuk menjawab tantangan tersebut. Pada proses perancangan dan pengembangan sistem data, pemodelan data merupakan bagian yang penting. Pemodelan data menyediakan teknik untuk menggambarkan kebutuhan informasi di dunia nyata dengan cara yang dapat dipahami pengguna serta membantu perancang mengimplementasikan kebutuhan informasi tersebut ke sistem basis data [2]. Pemodelan data dapat memberikan gambaran dalam bentuk model konseptual, model logis, dan model fisik [1]–[5]. Model konseptual basis data *structured query language* (SQL) digunakan sebagai representasi dari konsep sistem nyata yang diperoleh berdasarkan kebutuhan klien [6]. Tahap berikutnya adalah mentransfer model konseptual ke tingkat logis dengan cara mengonversi set entitas, atribut, dan *relationship* ke skema relasional yang menjelaskan tabel dan strukturnya serta relasi antartabel. Model logis kemudian dikonversi ke tingkat fisik, yaitu pembuatan skema relasional sesuai manajemen basis data yang dipilih [7].

Berbagai jenis model data telah berkembang, mulai dari model data relasional sampai dengan model data graf [3], [4]. Hal ini sejalan dengan perkembangan penanganan basis data relasional yang terstruktur sampai dengan data nonrelasional dengan skala besar yang termasuk dalam proses *big data* [8]. Saat ini, pengelolaan data nonrelasional dengan NoSQL (*not only SQL*) telah menggantikan pengelolaan basis data dengan SQL yang telah mendominasi komputasi pengelolaan basis data relasional selama beberapa dekade [8], [9].

Secara umum, NoSQL memiliki pendekatan yang berbeda dengan SQL, terutama dalam penyimpanan dan pengaksesan data [9]. Hal ini dapat dilakukan dengan empat kategori pemodelan data yang berorientasi pada nilai-kunci (*key-value-based*), dokumen (*document-based*), kolom (*column-based*), dan graf (*graph database*) [3], [8], [10]–[13]. Dengan pemodelan ini, NoSQL tidak hanya dapat memudahkan pemrosesan analitik, tetapi juga dapat digunakan untuk menangani pemrosesan transaksi yang ditangani oleh model data relasional [3], [10], [14].

Sejalan dengan evolusi pengelolaan basis data dari SQL ke NoSQL, teknologi perangkat (*tools*) pemodelan data pun ikut berkembang. Beberapa studi terkait pengembangan *tools*

pemodelan data yang sudah banyak digunakan antara lain *entity relationship diagram* (ERD), *collection relationship diagram* (CRD), *document data model* (DDM), *UML-Class diagram*, dan *NoSQL Abstract Model* (NoAM) [3], [14], [15].

ERD merupakan salah satu *tools* yang dapat digunakan untuk memodelkan data relasional yang merepresentasikan struktur data dan informasi pada tingkat konseptual, logis, dan fisik, yang dapat diimplementasikan pada basis data [3]. Selain itu, terdapat beberapa *tools* pemodelan data relasional, yaitu CRD, yang dapat mempresentasikan struktur data pada tingkat logis dan fisik; DDM pada tingkat logis; *UML-Class diagram* pada tingkat konseptual; dan NoAM, yang cenderung dapat diimplementasikan pada tingkat logis maupun fisik.

Referensi [14] menggunakan CRD untuk pemodelan data NoSQL berorientasi dokumen. CRD dapat digunakan untuk representasi tingkat logis maupun fisik yang dekat dengan implementasi aplikasi. Pada CRD tingkat logis, komponen yang ditampilkan adalah entitas, relasi *embedded* dan *reference*, dan *cardinality* pada relasi. Sementara itu, CRD tingkat fisik terdiri atas seluruh komponen yang ada pada tingkat logis ditambah penerapan pola desain (*design pattern*), struktur, dan tipe data. Pola desain yang digunakan pada studi ini yaitu *schema versioning*, *subset*, *computed*, *bucket*, dan *extended reference*. Selain menjelaskan penggunaan diagram tingkat logis dan fisik, studi ini juga secara sekilas menunjukkan penggunaan ERD sederhana (tanpa disertai atribut) sebagai representasi diagram tingkat konseptual.

Referensi [14] menyatakan bahwa model data NoSQL diperlukan untuk meminimalkan kesalahan saat implementasi basis data, terutama dalam hal penjaminan *atomicity* operasi penting dan kinerja yang lebih baik dengan menyesuaikan prioritas operasi. Referensi [15] mengusulkan NoAM sebagai model data abstrak yang bersifat universal untuk basis data NoSQL. NoAM mengeksplorasi elemen-elemen yang sama pada setiap basis data NoSQL dan berupaya untuk menyeimbangkan setiap perbedaan dan variasi yang ditemukan. NoAM terdiri atas tiga komponen, yaitu 1) basis data adalah sekumpulan dari *collection*; 2) *collection* adalah sekumpulan dari *block* yang memiliki pengenalan unik; dan 3) *block* adalah sekumpulan *entry* yang tidak kosong. Setiap *entry* memiliki sepasang *entry key* dan *entry value*. Hal yang unik di sini adalah studi ini menyatakan model abstrak, tetapi sudah merepresentasikan struktur data larik (*array*) dan objek (*object*) yang umumnya ditemukan pada model data fisik.

Referensi [3] membangun model data NoSQL berorientasi dokumen pada tingkat logis dalam bentuk DDM. DDM tersebut dihasilkan melalui proses transformasi dari CDM pada tingkat konseptual yang direpresentasikan oleh *UML-Class diagram* yang telah dimodifikasi. Pemetaan setiap elemen yang ada pada DDM dan CDM adalah 1) *class* dengan *collection*; 2) *attribute* dengan *column*; dan 3) *association* dengan *embedded* atau *reference*. Studi ini hanya membahas model data tingkat konseptual dan logis, belum membahas pemodelan data tingkat fisik.

Berdasarkan beberapa studi tersebut, terlihat bahwa pemodelan data dapat dilakukan dengan berbagai *tools* pemodelan, seperti ERD, CRD, DDM, *UML-Class diagram*, dan NoAM. Perkembangan *tools* pemodelan tidak terlepas dari

proses terhadap basis data yang digunakan, mulai dari proses pembentukan, penyimpanan, sampai pengaksesan basis data tersebut. Hal ini juga memperhatikan karakteristik data yang ada. Suatu model data yang dibentuk dengan *tools* apa pun harus dapat merepresentasikan atau mengabstraksikan rancangan karakteristik dan perilaku basis data yang akan dipakai oleh suatu aplikasi tertentu dalam menghasilkan informasi yang dibutuhkan oleh pengguna aplikasi tersebut. Model data tersebut harus dapat dilihat dari sudut pandang konseptual (yang menunjukkan fungsi aplikasi) serta sudut pandang logis dan fisik (yang menunjukkan proses teknis) dari penggunaan basis data tersebut.

Dengan struktur data yang semakin kompleks, pemodelan data NoSQL menjadi tantangan yang perlu menjadi pertimbangan. Khususnya pada NoSQL yang berorientasi dokumen, belum ada *tools* yang dapat dengan tepat merepresentasikan atau memodelkan struktur data dan informasi, baik secara konseptual, logis, maupun fisik. Model data NoSQL tersebut harus mampu menangani fleksibilitas bentuk dan ukuran data yang terstruktur, semiterstruktur, dan *polymorphic* [10].

Studi ini mencoba untuk melakukan pemodelan relasional pada struktur data NoSQL berorientasi dokumen. Pemodelan dilakukan pada tingkat konseptual, logis, dan fisik, yang diterapkan pada kasus Aplikasi Pengajuan Tugas Akhir di Jurusan Teknik Komputer dan Informatika (JTK), Politeknik Negeri Bandung (POLBAN). Diagram model data yang dirancang berlandaskan hasil sintesis beberapa studi [3], [14], [15], untuk dapat diimplementasikan pada sistem manajemen basis data MongoDB. Rancangan yang dihasilkan pada studi ini akan dikontribusikan pada perkuliahan di Program Studi D4 Teknik Informatika, JTK, khususnya sebagai bahan studi pengembangan perangkat lunak berbasis web.

II. METODOLOGI

Studi ini mengadopsi metodologi fleksibel studi sebelumnya [14], [16], yang telah dimodifikasi sebagai dasar proses perancangan model data untuk seluruh tingkat abstraksi (konseptual, logis, dan fisik). Dalam membentuk model data tingkat logis dan fisik, beban kerja (*workload*) diperlukan sebagai bahan pertimbangan untuk menentukan relasi (*embedded* dan *reference*), struktur data, dan pola desain. Studi dijalankan dengan beberapa tahap kegiatan, yaitu studi pustaka, perencanaan pemodelan data, pelaksanaan atau pembentukan model data, dan evaluasi model data.

A. Studi Pustaka

Tahap ini mempelajari studi literatur yang terkait dengan topik pemodelan data NoSQL berorientasi dokumen pada setiap tingkat abstraksi (konseptual, logis, dan fisik), termasuk mengidentifikasi masalah berdasarkan hasil kajian teoretis.

B. Perencanaan

Tahap ini meliputi kegiatan persiapan sebelum pelaksanaan pemodelan dijalankan, yang terdiri atas hal-hal sebagai berikut.

1) *Penentuan Proses Bisnis (Business Processes)*, *Aturan Bisnis (Business Rules)*, dan *Kebutuhan Bisnis (Business*

TABEL I
PROSES BISNIS

Kode	Proses Bisnis	Deskripsi
PB-01	Pendaftaran KoTA	Kelompok Tugas Akhir (KoTA) mendaftarkan diri dengan ketentuan memiliki anggota maksimal tiga orang untuk D3 dan dua orang untuk D4. Setiap KoTA yang mendaftar diharuskan mengajukan topik tugas akhir (TA) dan judul TA, calon pembimbing, serta data anggota. Pendaftaran KoTA dilakukan oleh perwakilan kelompok kepada koordinator TA. Mahasiswa hanya boleh terdaftar pada satu KoTA. Topik TA dipilih dari daftar topik yang disediakan dan dapat berjumlah lebih dari satu.
PB-02	Penetapan Kode KoTA	Proses penetapan Kode untuk setiap KoTA dengan ketentuan struktur tertentu yang menunjukkan kelas-prodi mahasiswa dan nomor urut Kode berdasarkan NIM terkecil anggota KoTA. Kode KoTA untuk satu tahun ajaran bersifat unik.
PB-03	Penetapan Pembimbing	Pembimbing 1 dan Pembimbing 2 untuk setiap kelompok ditetapkan dengan mempertimbangkan topik TA dan topik yang diminati dosen. Setiap dosen dapat memilih lebih dari satu topik dari daftar topik yang diberikan.
PB-04	Perhitungan Beban Pembimbing	Beban pembimbing adalah jumlah KoTA yang dibimbing oleh seorang dosen.

Needs): Kasus yang digunakan pada studi ini yaitu kasus riil pengembangan Aplikasi Pengajuan Tugas Akhir di Politeknik Negeri Bandung. Model konseptual merepresentasikan abstraksi pemahaman dalam skenario pemanfaatannya [17]. Untuk dapat lebih memahami abstraksi atau domain permasalahan pada kasus yang diberikan, dibutuhkan suatu skenario atau deskripsi dari domain permasalahan yang dapat ditinjau dari proses bisnis, aturan bisnis, dan kebutuhan bisnis yang ditentukan lebih dahulu. Rancangan pemodelan data untuk aplikasi yang dikembangkan mencakup empat proses bisnis (Tabel I), enam aturan bisnis (Tabel II), dan dua puluh kebutuhan bisnis (Tabel III).

Beberapa asumsi untuk membatasi ruang lingkup studi kasus antara lain:

- data pembimbing dan topik yang sesuai dengan keahlian dosen telah disediakan;
- data mahasiswa telah disediakan, tetapi belum ditentukan setiap mahasiswa sebagai anggota Kelompok Tugas Akhir (KoTA); dan
- sebagian validasi data, *integrity constraint*, dan *atomic transaction* antar-*collection* dilakukan pada sisi aplikasi.

Pemodelan konseptual adalah aktivitas penggambaran secara formal beberapa aspek dunia fisik dan sosial di sekitar untuk tujuan pemahaman dan komunikasi. Model konseptual

TABEL II
ATURAN BISNIS

Kode	Deskripsi
BR-01	Setiap Kelompok Tugas Akhir (KoTA) harus memiliki dua pembimbing.
BR-02	Jumlah maksimum anggota KoTA untuk D3 sebanyak tiga orang dan untuk D4 sebanyak dua orang.
BR-03	Panjang maksimum nomor induk mahasiswa (NIM) adalah sembilan digit.
BR-04	Panjang maksimum pengajuan judul TA adalah lima belas kata.
BR-05	Jumlah maksimum pengajuan pembimbing adalah lima orang dan jumlah minimum adalah tiga orang.
BR-06	Setiap KoTA akan diberi Kode KoTA dengan struktur "KoTA-" [lima digit] + <kode_prodi_kelas> [satu digit] + <no_urut> [dua digit]. Penentuan <kode_prodi_kelas> sebagai berikut: D3 Kelas A: 1; D3 Kelas B: 2; D4 Kelas A: 3; D4 Kelas B: 4.

berkonsentrasi pada penggambaran entitas, tipe data, relasi, operasi pengguna, dan *constraints* [17]. Penentuan entitas pada model konseptual sangat mempertimbangkan deskripsi atau penjelasan pada proses bisnis, aturan bisnis, dan kebutuhan bisnis dari kasus tersebut sebagai gambaran kasus riil yang dimodelkan. Selain itu, skenario juga dibutuhkan untuk mengukur keberhasilan penggambaran atau pengabstraksian kasus di atas dalam model logis dan model fisik.

2) *Penentuan Beban Kerja*: Beban kerja memiliki peran penting, khususnya dalam pembentukan model data tingkat logis dan fisik. Beban kerja dalam studi ini merupakan pengukuran intensitas operasi, termasuk permintaan informasi (*query*) yang sesuai dengan kebutuhan bisnis dari aplikasi. Pengukuran dilakukan berdasarkan perkiraan intensitas pelaksanaan, standar waktu pemrosesan, dan kualitas kerja setiap operasi. Dengan pengukuran ini, rancangan dapat terlihat lebih optimal dan tepat guna karena tiap operasi yang dilaksanakan dapat diukur waktu operasinya sesuai dengan pola desain dan model struktur data yang diberikan. Untuk itu, diperlukan penentuan operasi yang terlibat dalam aplikasi (Tabel IV) serta penentuan kuantitas dan kualitas operasi (Tabel V).

Operasi perlu ditentukan lebih dahulu untuk melihat potensi waktu operasi yang tinggi berdasarkan struktur data yang telah ditentukan. Dengan melihat potensi waktu yang tinggi, pola desain diharapkan dapat diterapkan dengan lebih efisien. Tabel IV menunjukkan daftar operasi yang harus tersedia pada aplikasi berdasarkan proses, aturan, dan kebutuhan bisnis. Daftar operasi tersebut akan dieksekusi oleh mesin basis data.

Kuantifikasi operasi bertujuan untuk mengukur nilai perkiraan dari rutinitas operasi yang dilakukan dan penetapan standar waktu pemrosesan operasi. Sementara itu, kualifikasi operasi merupakan deskripsi yang menunjukkan karakteristik atau tingkat urgensi operasi secara kualitatif. Kuantifikasi dan kualifikasi operasi ditentukan berdasarkan eksperimen pengujian aplikasi yang telah dilakukan sebelumnya. Tabel V menampilkan hasil pengukuran nilai kuantifikasi dan penentuan kualifikasi dengan asumsi bahwa setiap operasi dilakukan pada saat periode pengajuan tugas akhir.

TABEL III
KEBUTUHAN BISNIS

Kode	Deskripsi
REQ-F-01	Aplikasi dapat menangani penambahan data KoTA untuk Prodi D3 dan D4 (NIM, Nama, Jurusan, Prodi, Kelas, Topik, Judul, dan Pembimbing).
REQ-F-01-01	Aplikasi dapat menyimpan semua KoTA yang dimasukkan oleh pengguna pada basis data.
REQ-F-01-02	Aplikasi dapat melakukan pemeriksaan duplikasi data yang sudah dimasukkan sebelumnya.
REQ-F-02	Aplikasi dapat membangkitkan (<i>generate</i>) secara otomatis Kode KoTA.
REQ-F-02-01	Aplikasi membangkitkan secara otomatis Kode KoTA setelah semua KoTA pada Prodi D3/D4 selesai mengirimkan data.
REQ-F-02-02	Aplikasi dapat mengambil satu NIM anggota terkecil dari setiap anggota KoTA pada setiap kelas, kemudian melakukan penyortiran data NIM tersebut dari semua KoTA pada setiap kelas.
REQ-F-02-03	Aplikasi dapat membangkitkan Kode KoTA sesuai dengan [BR-06].
REQ-F-03	Aplikasi dapat menampilkan seluruh data KoTA yang sudah diberi Kode.
REQ-F-04	Aplikasi dapat mencocokkan (<i>matching</i>) data antara topik dosen dengan topik mahasiswa.
REQ-F-05	Aplikasi dapat membantu dalam menetapkan data pembimbing untuk setiap KoTA.
REQ-F-05-01	Aplikasi dapat menampilkan daftar data dosen yang memiliki topik sama dengan KoTA.
REQ-F-05-02	Aplikasi dapat menampilkan daftar data dosen untuk dipilih sebagai pembimbing.
REQ-F-05-03	Aplikasi dapat menghilangkan data dosen terpilih pada Pembimbing 1 dari daftar data dosen untuk dipilih sebagai Pembimbing 2.
REQ-F-05-04	Aplikasi dapat memunculkan rekap sederhana berisikan daftar KoTA dengan pembimbing yang sudah ditentukan.
REQ-F-06	Aplikasi dapat menghitung beban setiap dosen yang telah ditetapkan sebagai pembimbing.
REQ-F-06-01	Aplikasi dapat menghitung Beban Pembimbing setelah semua pembimbing ditetapkan (<i>assigned</i>) pada semua KoTA.
REQ-F-06-02	Aplikasi dapat menghitung beban Pembimbing 1 untuk setiap dosen.
REQ-F-06-03	Aplikasi dapat menghitung beban Pembimbing 2 untuk setiap dosen.
REQ-F-06-04	Aplikasi dapat menghitung beban total bimbingan untuk setiap dosen.
REQ-F-07	Aplikasi dapat menampilkan daftar data beban bimbingan dari setiap dosen.

3) *Penentuan Pola Desain*: Pada NoSQL berorientasi dokumen, umumnya pemodelan relasi antar-*collection* yang terdiri atas 1) 1-ke-1 dengan *embedded*; 2) 1-ke-*N* dengan *embedded* atau *reference*; dan 3) *M*-ke-*N* dengan *embedded* atau *reference*. Setiap struktur *embedded collection* dan *reference collection* yang merepresentasikan relasi antar-*collection* memiliki keunggulan dan kelemahan. *Embedded* mengorganisasi data dengan struktur denormalisasi (*redundant*), menyediakan kemudahan untuk memperoleh dan memanipulasi data terkait, hanya dengan operasi tunggal.

TABEL IV
DAFTAR OPERASI

Kode	Operasi	Jenis	Deskripsi
OP-01	Memuat data pada formulir pengajuan KoTA.	Baca	Memuat seluruh data topik, Calon Pembimbing, dan mahasiswa.
OP-02	Menyimpan data pada formulir pengajuan KoTA.	Tulis	Menyimpan data topik, judul, Calon Pembimbing, NIM setiap mahasiswa.
OP-03	Memuat data pada formulir penetapan pembimbing.	Baca	Memuat hanya satu data KoTA yang telah mengajukan, meliputi: NIM, nama, topik, judul, dan pembimbing.
OP-04	Menyimpan data pada formulir penetapan pembimbing.	Tulis	Menyimpan data topik, judul, calon pembimbing, NIM berdasarkan KoTA yang dipilih.
OP-05	Memuat data KoTA yang disortir untuk menetapkan kode KoTA.	Baca	Memuat data KoTA yang telah disortir berdasarkan: 1. NIM terkecil pada KoTA, 2. NIM terkecil pada seluruh KoTA yang dikelompokkan berdasarkan prodi dan kelas.
OP-06	Menyimpan hasil penetapan kode KoTA.	Tulis	Menyimpan data hasil penetapan kode KoTA disertai dengan aturan: 1. berdasarkan hasil sortir, 2. berdasarkan <kode_prodi_kelas>.
OP-07	Memuat data kelompok yang telah ditetapkan pembimbing.	Baca	Memuat seluruh data KoTA yang telah mengajukan, meliputi: Kode KoTA, NIM, Nama, Topik, Judul, Pembimbing 1 dan Pembimbing 2.
OP-08	Memuat data beban dosen yang telah ditetapkan sebagai pembimbing.	Baca	Memuat seluruh data beban dosen yang telah ditetapkan sebagai pembimbing, meliputi: Kode Dosen, Nama Dosen, Beban Pembimbing 1, Beban Pembimbing 2, Total Beban Pembimbing.

Sebaliknya, *reference* dengan struktur normalisasi (*non-redundant*), sama seperti model data relasional pada umumnya, harus melakukan operasi gabungan (*join*) untuk setiap *collection* yang terkait. Tentunya penggunaan operasi gabungan tersebut akan berdampak terhadap menurunnya kinerja operasi membaca.

Pada studi ini terdapat dua jenis relasi, yaitu 1-ke-*N* dan *M*-ke-*N*. Namun, pada NoSQL berorientasi dokumen, keduanya dapat menghasilkan struktur *collection* yang sama karena adanya dukungan struktur data larik. Selain itu, terdapat tiga operasi yang memiliki beban kerja tinggi. Operasi ini digunakan sebagai bahan pertimbangan dalam menentukan pola desain.

Penerapan pola desain dapat memberikan keleluasaan struktur *collection* dengan menyesuaikan kebutuhan aplikasi, sehingga tidak harus kaku menggunakan *embedded* dan

TABEL V
KUANTIFIKASI DAN KUALIFIKASI OPERASI

Kode	Beban Kerja	
	Kuantifikasi	Kualifikasi
OP-01	20-35 baca/hari < 3 detik	Tidak ada data lampau.
OP-02	20-35 tulis/hari < 3 detik	Beban kerja rendah.
OP-03	20-35 baca/hari < 3 detik	Tidak ada data lampau.
OP-04	20-35 tulis/hari < 3 detik	Beban kerja rendah.
OP-05	3-10 baca/aksi < 5 detik	<ul style="list-style-type: none"> • Tidak ada data lampau. • Berpotensi beban kerja tinggi dengan melibatkan dua data, yaitu Mahasiswa dan KoTA.
OP-06	3-10 tulis/aksi < 5 detik	<ul style="list-style-type: none"> • Penulisan secara berkala. • Berpotensi beban kerja sedang, yang hanya melibatkan data KoTA saja.
OP-07	20-35 baca/hari < 5 detik	<ul style="list-style-type: none"> • Tidak ada data lampau. • Berpotensi beban kerja tinggi dengan melibatkan empat data, yaitu Mahasiswa, KoTA, Dosen, dan Topik
OP-08	20-35 baca/hari < 5 detik	<ul style="list-style-type: none"> • Tidak ada data lampau. • Berpotensi beban kerja tinggi dengan melibatkan dua data, yaitu KoTA dan Dosen.

reference. Terdapat dua pola desain yang dapat diterapkan, yaitu *extended reference* dan *computed*.

Extended reference memungkinkan penggunaan sekaligus struktur *embedded collection* dan *reference collection* [14]. Penggunaan pola desain ini dipertimbangkan berdasarkan hasil analisis beban kerja pada Tabel IV dan Tabel V, yang berupaya menghindari penggunaan operasi gabungan antar-collection dengan memanfaatkan *embedded*, khususnya untuk operasi yang memiliki potensi beban tinggi. Pada kasus ini, operasi yang memiliki potensi beban tinggi adalah pada data KoTA dan Dosen. Selain itu, pola ini pun tetap mempertahankan *subcollection* melakukan *reference* terhadap *collection* yang dirujuk, sehingga jika hanya dokumen tertentu yang diperlukan, tidak perlu dilakukan pengaksesan seluruh dokumen pada *subcollection*, cukup dengan langsung mengakses dokumen *collection* yang dirujuk oleh *subcollection*.

Berikut ini merupakan ilustrasi perbandingan struktur data *reference* dan *extended reference* menggunakan format JavaScript Object Notation (JSON).

Reference

A1 = {a0, a1, a2, a3, a4};
 B1 = {b0, b1, b2, a0};
 B2 = {b3, b4, b5, a0};
 B3 = {b6, b7, b8, a0};

Pada struktur data *reference* di atas, A1 merupakan dokumen pada *collection A* yang memiliki *ID* dengan nilai a0. Kemudian, B1, B2, dan B3 merupakan dokumen pada *collection B* yang melakukan *reference* terhadap dokumen A1 berdasarkan *ID* yang dimiliki dokumen A1. Proses untuk memperoleh

dokumen pada *collection A* dan *B* dapat diilustrasikan menggunakan aljabar relasional, dengan melibatkan operator gabungan seperti *natural join* (\bowtie): $A \bowtie B$.

Extended Reference

A1 = {a0, a1, a2, a3, a4,
 [
 {b0, b1, b2},
 {b3, b4, b5},
 {b6, b7, b8}
]
 };
 B1 = {b0, b1, b2, a0};
 B2 = {b3, b4, b5, a0};
 B3 = {b6, b7, b8, a0};

Pada struktur data *extended reference* di atas, A1 merupakan dokumen yang melakukan *embedded*, sehingga di dalamnya terdapat subdokumen B1, B2, dan B3. Dokumen B1, B2, dan B3 yang melakukan *reference* ke A1 dapat mempertahankan penggunaan *ID* untuk A1, yaitu a0, atau dapat juga dihilangkan. Proses untuk memperoleh dokumen pada *collection A* dan *B* yang menerapkan pola desain *extended reference* tidak perlu melibatkan operasi gabungan: A.

Computed bertujuan untuk mengurangi intensitas pengulangan komputasi; dalam kasus ini, *computed* secara logis dilakukan untuk menghitung Beban Pembimbing pada Dosen berdasarkan peran pada Dosen yang ada di dalam KoTA. Maka, pada saat *query* untuk memuat data Beban Pembimbing, hanya diperlukan proses membaca saja, tidak perlu dilakukan komputasi untuk menghitung ulang.

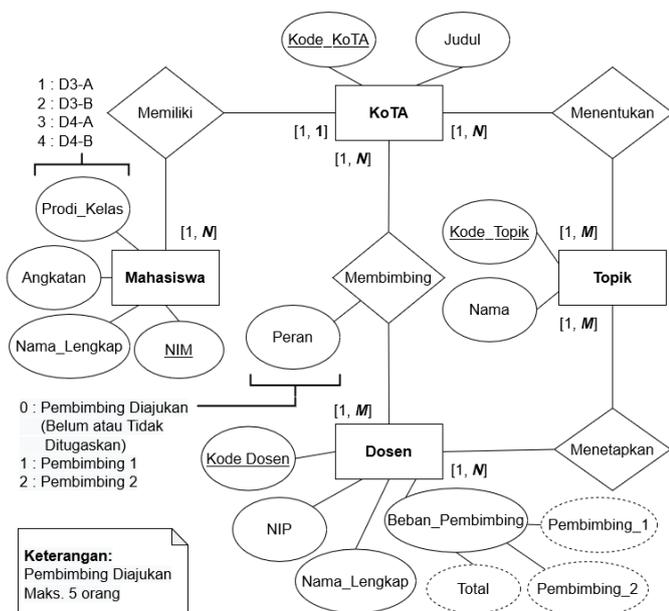
C. Pelaksanaan Pemodelan Data

Fleksibilitas model data berorientasi dokumen dapat menyebabkan skema menjadi lebih kompleks. Oleh karena itu, dalam melakukan pemodelan data perlu, dipertimbangkan proses bisnis, aturan bisnis, kebutuhan bisnis, dan beban kerja pada operasi. Pemodelan data yang dilakukan dijelaskan sebagai berikut.

1) *Perancangan Model Data Konseptual*: Model data konseptual harus memberikan gambaran besar (*big picture*) aplikasi atau sistem dengan representasi yang utuh, termasuk cara mengorganisasi dan keterkaitan dengan aturan dan kebutuhan bisnis. Tahap ini bertujuan untuk mengidentifikasi objek konseptual dan atributnya serta menentukan keterkaitan (relasi) antarobjek yang secara keseluruhan mencirikan perilaku aplikasi yang dirancang. Identifikasi dilakukan dengan mengacu pada proses bisnis, aturan bisnis, serta kebutuhan bisnis yang telah ditentukan sebelumnya [18]. *Tools* pemodelan yang digunakan pada model data konseptual adalah ERD dengan notasi Chen.

2) *Perancangan Model Data Logis*: Model data logis bersifat tidak terlalu abstrak dan harus mampu menggambarkan dengan cukup rinci implementasi teknis dari model konseptual yang terbentuk. *Tools* pemodelan yang digunakan pada model data logis yaitu CRD logis.

3) *Perancangan Model Data Fisik*: Model data fisik harus digambarkan dengan konkret, sesuai dengan target mesin basis



Gbr. 1 Model data konseptual Aplikasi Pengajuan Tugas Akhir.

TABEL VI
DAFTAR ENTITAS, ATRIBUT, DAN RELASI

No.	Entitas	Atribut	Relasi (Referensi ke Entitas)
1	Mahasiswa	<ul style="list-style-type: none"> NIM Nama_Lengkap Angkatan Prodi_Kelas 	--
2	KoTA	<ul style="list-style-type: none"> Kode_KoTA Judul 	<ul style="list-style-type: none"> Mahasiswa (KoTA.Kode_KoTA): 1-ke-N
3	Dosen	<ul style="list-style-type: none"> Kode_Dosen NIP Nama_Lengkap Beban_Pembimbing <ul style="list-style-type: none"> Pembimbing_1 Pembimbing_2 Total 	<ul style="list-style-type: none"> KoTA (Dosen.Kode_Dosen): M-ke-N Atribut Deskriptif: Peran
4	Topik	<ul style="list-style-type: none"> Kode_Topik Nama 	<ul style="list-style-type: none"> KoTA (Topik.Kode_Topik): M-ke-N Dosen (Topik.Kode_Topik): M-ke-N

data yang akan digunakan, yang pada studi ini adalah MongoDB. Tools pemodelan yang digunakan pada model data fisik yaitu CRD fisik.

D. Evaluasi

Evaluasi atau pemeriksaan rancangan model data dilakukan secara sekuensial, mulai dari fase yang paling abstrak sampai konkret. Fase awal atau sebelumnya harus dipenuhi terlebih dahulu untuk dapat melanjutkan ke fase berikutnya. Setiap rancangan model data memiliki daftar tilik pemeriksaan yang

TABEL VII
PENERAPAN POLA DESAIN

No.	Pola Desain	Relasi/Atribut Terkait	Kode Operasi	Keterangan
1	Extended reference	KoTA-Mahasiswa (1-ke-N)	5, 7	Mahasiswa reference dan embedded ke KoTA.
		Dosen-KoTA (M-ke-N)	7	KoTA reference ke Dosen dan Dosen embedded ke KoTA.
		Topik-KoTA (M-ke-N)	7	KoTA reference ke Topik dan Topik embedded ke KoTA.
		Topik-Dosen (M-ke-N)	7	Dosen reference ke Topik dan Topik embedded ke Dosen.
2	Computed	Dosen.Pembimbing_1 Hitung ((KoTA.Peran = 1))	8	Menghitung beban Pembimbing 1.
		Dosen.Pembimbing_2 Hitung ((KoTA.Peran = 2))	8	Menghitung beban Pembimbing 2.
		Dosen.Total (Dosen.Pembimbing_1 + Dosen.Pembimbing_2)	8	Menghitung total beban pembimbing.

berbeda, menyesuaikan referensi gambar dan tabel yang digunakan.

1) *Pemeriksaan Rancangan Model Data Konseptual:* Pemeriksaan ini dilakukan dengan memetakan kesesuaian antara model data konseptual dengan proses bisnis, aturan bisnis, dan kebutuhan bisnis yang ditentukan berdasarkan objek konseptual yang terbentuk.

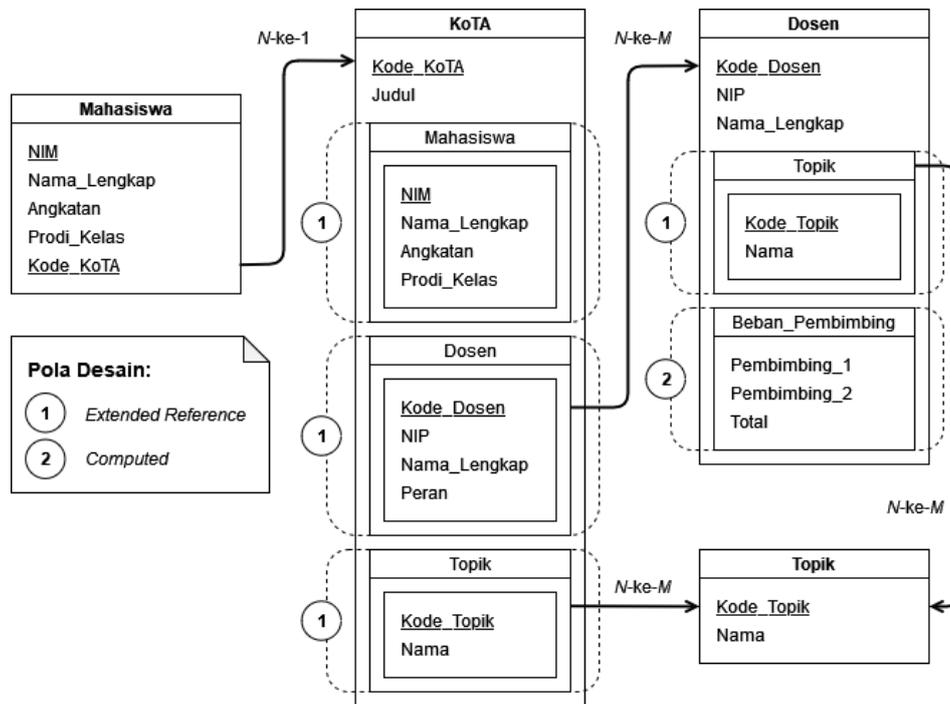
2) *Pemeriksaan Rancangan Model Data Logis:* Pemeriksaan dilakukan dengan memetakan kesesuaian antara model data konseptual dan pola desain yang ditentukan berdasarkan beban kerja.

3) *Pemeriksaan Rancangan Model Data Fisik:* Pemeriksaan dilakukan dengan memetakan kesesuaian antara model data logis terhadap collection, field, tipe data, dan struktur data yang didefinisikan pada model data fisik. Ruang lingkup evaluasi pada model data fisik tidak termasuk pengujian langsung pada mesin basis data.

III. HASIL DAN EVALUASI

A. Hasil

1) *Pemodelan Data Konseptual:* Berdasarkan analisis deskripsi proses bisnis (Tabel I), aturan bisnis (Tabel II), dan kebutuhan bisnis (Tabel III), teridentifikasi beberapa objek konseptual yang diperlukan untuk membangun Aplikasi Pengajuan Tugas Akhir. Objek-objek tersebut adalah empat entitas dengan dua belas atribut dan empat relasi (Tabel VI).



Gbr. 2 Model data logis Aplikasi Pengajuan Tugas Akhir.

Relasi menunjukkan keterkaitan logis antarentitas, disertai dengan atribut dan nilai keterkaitan (*cardinality*). Simbol *cardinality* satu-ke-banyak ditandai dengan 1-ke-N, sedangkan banyak-ke-banyak ditandai dengan M-ke-N.

Selain itu, terdapat satu objek deskriptif sebagai atribut Peran pada relasi antara entitas Dosen dan KoTA. Atribut deskriptif tersebut bertujuan untuk menunjukkan bahwa terdapat tiga jenis status penetapan pembimbing, yaitu 1) Dosen yang diajukan oleh KoTA, tetapi belum atau tidak ditetapkan sebagai pembimbing; 2) Dosen yang sudah ditetapkan menjadi Pembimbing 1; dan 3) Dosen yang sudah ditetapkan menjadi Pembimbing 2.

Dengan *tools* pemodelan ERD, entitas, atribut, dan relasi yang teridentifikasi dapat menggambarkan dengan baik perilaku aplikasi dari sudut pandang pemodelan data. Pada Gbr. 1, entitas Dosen dipertimbangkan untuk memiliki atribut komposit (*composite attribute*), yaitu Beban Pembimbing, yang berupa atribut turunan (*derived attribute*) untuk Pembimbing_1, Pembimbing_2, dan Total. Selain itu, terdapat atribut deskripsi (*descriptive attribute*) Peran pada relasi Membimbing antara Dosen-KoTA untuk menetapkan dosen sebagai Pembimbing 1, Pembimbing 2, atau belum ditetapkan.

2) *Pemodelan Data Logis*: Dalam pemodelan ini, semua entitas pada model data konseptual diimplementasikan sebagai *collection*. Dengan menggunakan *tools* CRD, pola desain *extended reference* dan *computed* diterapkan pada pemodelan data logis ini. Tabel VII menunjukkan penerapan pola desain di atas dikaitkan dengan operasi berdasarkan beban kerja.

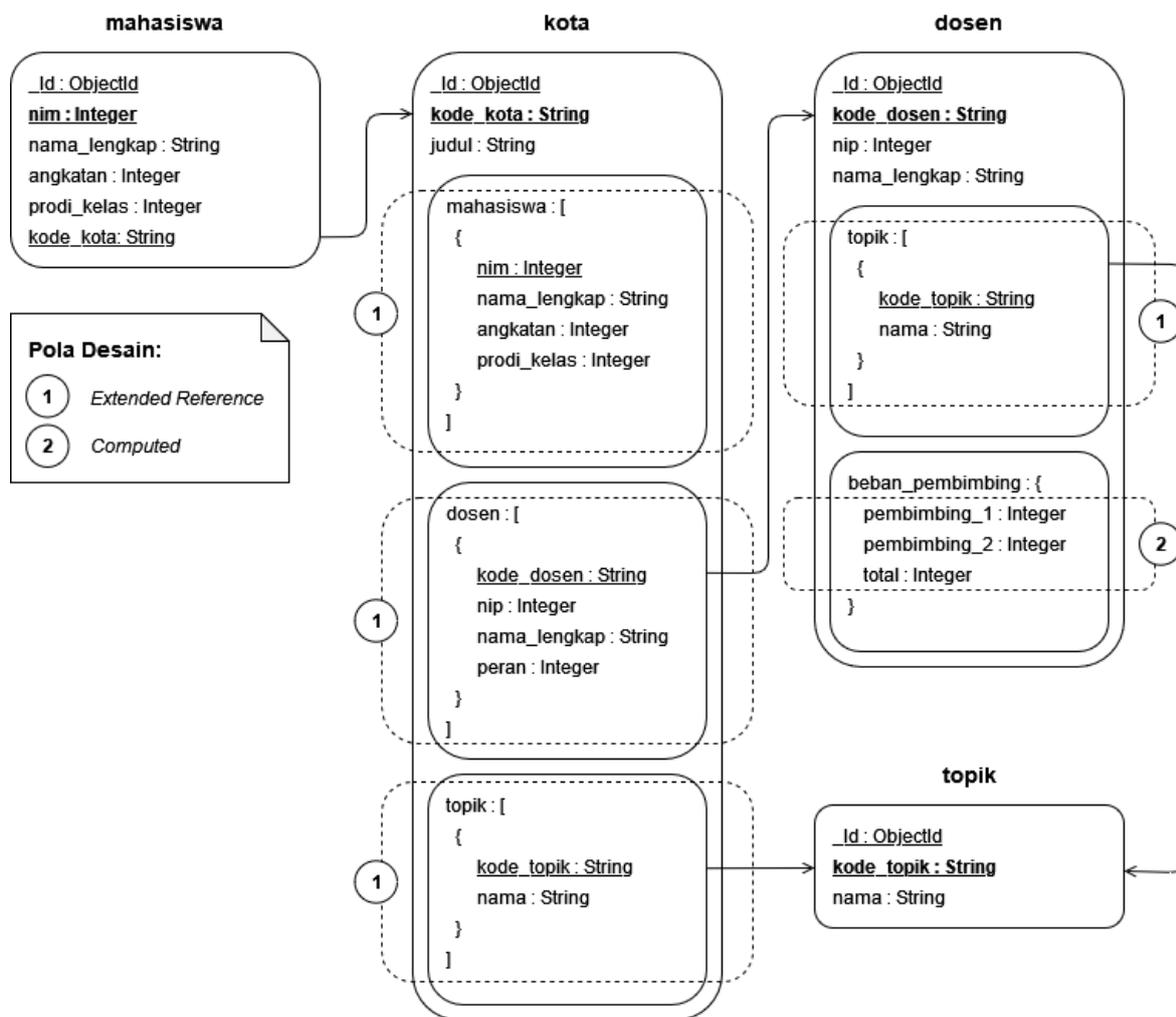
Gbr. 2 menunjukkan setiap relasi yang terbentuk antar-*collection*, baik 1-ke-N maupun M-ke-N menggunakan pola *extended reference*. Terdapat perbedaan implementasi *extended reference* untuk kedua jenis relasi tersebut. Dalam

relasi KoTA-Mahasiswa (1-ke-N), atau dapat dibaca terbalik Mahasiswa-KoTA (N-ke-1), pada *collection* Mahasiswa (*collection* anak) terdapat *field* Kode_KoTA sebagai *reference* terhadap Kode_KoTA pada *collection* KoTA (*collection* induk). Sementara itu, pada relasi M-ke-N untuk Dosen-KoTA, Topik-KoTA, dan Topik-Dosen tidak dialokasikan *field* khusus sebagai *reference*. Selain itu, terdapat pola *computed* pada seluruh atribut yang dimiliki oleh *subcollection* Beban Pembimbing pada *collection* Dosen. Dengan penerapan pola *extended reference*, terlihat bahwa terjadi operasi baca yang sangat intensif (*read-intensive*) pada operasi relasi (operasi 5 dan 7 pada Tabel VII). Hal ini juga terjadi pada penerapan pola *computed* pada proses perhitungan beban pembimbing (operasi 8 pada Tabel VII). Ini berarti dengan adanya *read-intensive*, waktu pemrosesan dapat lebih efisien karena dalam operasi relasi tidak diperlukan proses penggabungan (*join*) dan tidak terjadi pengulangan dalam operasi perhitungan beban pembimbing.

3) *Pemodelan Data Fisik*: Pemodelan data fisik, yang dibangun dengan *tools* CRD fisik (Gbr. 3), hanya memperjelas model data logis berdasarkan penggunaan tipe dan struktur data. Hal ini dimaksudkan untuk memudahkan proses implementasi pada mesin basis data yang dituju. Tipe data yang digunakan yaitu *ObjectId*, *string*, dan *integer*, sedangkan struktur data yang digunakan adalah objek, untuk merepresentasikan *subcollection*, ditandai dengan kurung kurawal “{ }”; dan larik, untuk menampung atribut komposit, dengan tanda berupa kurung siku “[]”.

B. Evaluasi

1) *Pemeriksaan Rancangan Model Data Konseptual*: Hasil pemeriksaan rancangan model data konseptual menunjukkan



Gbr. 3 Model data fisik Aplikasi Pengajuan Tugas Akhir.

bahwa entitas disertai dengan atribut dan relasi yang terbentuk pada rancangan model data konseptual dapat merepresentasikan objek nyata yang sesuai dengan proses bisnis, aturan bisnis, dan kebutuhan bisnis. Seperti yang terlihat pada Gbr. 1, proses bisnis PB01 (Pendaftaran KoTA) teridentifikasi berdasarkan entitas KoTA; Mahasiswa; Dosen; dan Topik; serta relasi KoTA-Mahasiswa; KoTA-Dosen; KoTA-Topik; dan Dosen-Topik. Entitas KoTA berelasi dengan entitas Mahasiswa untuk mengakomodasi pengelolaan data KoTA. Proses pendaftaran KoTA dapat dilakukan sesuai dengan kebutuhan, dengan adanya relasi terhadap entitas Dosen untuk menentukan pembimbing 1 dan 2, dan relasi terhadap entitas Topik untuk menentukan kesesuaian topik antara KoTA dan pembimbing. Proses bisnis PB-02 (Penetapan Kode KoTA) terlihat berdasarkan relasi KoTA-Mahasiswa. Proses penetapan Kode KoTA dapat dilakukan dengan adanya atribut Kode_KoTA pada entitas KoTA. Proses bisnis PB-03 (Penetapan Pembimbing) terlihat berdasarkan relasi KoTA-Dosen. Proses penetapan pembimbing 1 dan 2 memerlukan adanya relasi antara entitas KoTA dan Dosen. Penetapan pembimbing tersebut mempertimbangkan kesesuaian topik antara KoTA dan pembimbing, sehingga memerlukan keterlibatan entitas Topik, baik antara KoTA dan Topik,

maupun Dosen dan Topik. Proses bisnis PB-04 (Perhitungan Beban Pembimbing) terlihat berdasarkan relasi Dosen-KoTA. Perhitungan beban pembimbing dapat diakomodasi dengan adanya atribut komposit Beban_Pembimbing yang terdiri atas atribut turunan Pembimbing_1, Pembimbing_2, dan Total. Terdapat juga atribut deskriptif Peran pada relasi KoTA-Dosen sebagai sumber data untuk menghitung beban pembimbing.

2) *Pemeriksaan Rancangan Model Data Logis*: Rancangan model data logis sudah sesuai dengan model data konseptual dan pola desain (Tabel VII) yang diturunkan dari hasil analisis beban kerja dengan melakukan perkiraan kuantifikasi dan kualifikasi setiap operasi (Tabel V). Pembentukan *collection* Mahasiswa, KoTA, Dosen, dan Topik disertai dengan *field* sesuai dengan entitas dan atribut pada model data konseptual. *Collection* Mahasiswa menerapkan *reference* dan *embedded* ke KoTA. Penerapan ini sesuai dengan OP-05 dan OP-07. *Collection* KoTA menerapkan *reference* ke Dosen dan Dosen *embedded* ke KoTA untuk memenuhi terbentuknya relasi *M-ke-N* yang sesuai dengan OP-07. Penerapan seluruh *field* dari *collection* Dosen sesuai dengan entitas dan atribut pada model data konseptual. *Collection* KoTA menerapkan *reference* ke Topik dan Topik *embedded* ke KoTA berdasarkan OP-07.

Collection Dosen menerapkan *reference* ke Topik dan Topik *embedded* ke Dosen berdasarkan OP-07.

3) *Pemeriksaan Rancangan Model Data Fisik*: Rancangan model data fisik yang telah diperiksa menunjukkan konsistensi terhadap rancangan model data logis yang mendefinisikan setiap *collection*, *field*, tipe data, dan struktur data. Pada model data fisik, penerapan struktur data larik objek menjadi bagian penting dalam membentuk relasi *M-ke-N* antara KoTA-Mahasiswa, Dosen-KoTA, Topik-KoTA, dan Topik-Dosen. Selain itu, struktur data objek dapat digunakan untuk mengakomodasi atribut turunan.

Dari evaluasi di atas, terlihat bahwa *tools* perancangan model data relasional ERD dengan notasi Chen dapat digunakan untuk model data tidak terstruktur pada NoSQL berorientasi dokumen di tingkat konseptual (*repurposing*). Demikian pula CRD, dapat digunakan untuk merepresentasikan tingkat logis maupun di tingkat fisik.

IV. KESIMPULAN

Hasil studi menunjukkan bahwa model data relasional dapat diterapkan pada basis data NoSQL berorientasi dokumen dengan memanfaatkan *tools* pemodelan ERD disertai notasi Chen pada model data konseptual, CRD logis pada model data logis, dan CRD fisik pada model data fisik. Studi ini berkontribusi pada perancangan model data dengan keunggulan kinerja membaca yang lebih baik (*read-intensive*) karena tidak memerlukan operasi gabungan (*join*) antar *collection* yang memiliki relasi (*extended reference*) dan tidak perlu melakukan pengulangan proses komputasi untuk atribut turunan (*computed*). Selain itu, terdapat temuan menarik pada rancangan model data logis dan fisik, yaitu pembentukan relasi *1-ke-N* dan *M-ke-N* dapat menghasilkan struktur *collection* yang sama, tidak seperti model data relasional pada data terstruktur yang membentuk struktur tabel yang berbeda. Hal ini terjadi karena adanya dukungan struktur data larik objek pada *collection*.

KONFLIK KEPENTINGAN

Penulis menyatakan bahwa representasi maupun interpretasi studi ini terbebas dari berbagai situasi atau kondisi yang terlibat dengan konflik kepentingan pribadi atau organisasi tertentu.

KONTRIBUSI PENULIS

Konseptualisasi, Muhammad Riza Alifi dan Transmissia Semiawan; Studi Kasus, Djoko Cahyo Utomo L.; Metodologi, Muhammad Riza Alifi dan Djoko Cahyo Utomo L.; Validasi, Transmissia Semiawan dan Hashri Hayati; Peninjauan dan penyuntingan, Muhammad Riza Alifi, Transmissia Semiawan, dan Hashri Hayati; Pengawasan, Transmissia Semiawan.

UCAPAN TERIMA KASIH

Studi ini terlaksana dengan dukungan hibah kompetitif untuk meningkatkan kapasitas program studi di Politeknik Negeri Bandung, dengan nomor hibah B/75.5/PL1.R7/PG.00.03/2021.

REFERENSI

- [1] V.C. Storey dan I. Y. Song, "Big Data Technologies and Management: What Conceptual Modeling Can Do," *Data and Knowl. Eng.*, Vol. 108, hal. 50–67, Mar. 2017.
- [2] H.E. Samra, A.S. Li, B. Soh, dan M.A. AlZain, "Review of Contemporary Database Design and Implication for Big Data," *Int. J. Smart Educ., Urban Soc.*, Vol. 12, No. 4, hal. 1–11, Okt. 2021.
- [3] K. Shin, C. Hwang, dan H. Jung, "NoSQL Database Design Using UML Conceptual Data Model Based on Peter Chen's Framework," *Int. J. Appl. Eng. Res.*, Vol. 12, No. 5, hal. 632–636, Des. 2017.
- [4] A. Silberschatz, H.F. Korth, dan S. Sudarshan, *Database System Concepts*, 7th ed., New York, AS: McGraw-Hill Book Company, 2020.
- [5] IBM Cloud Education (2020) "Data Modeling," [Online], <https://www.ibm.com/cloud/learn/data-modeling>, tanggal akses: 28-Sep-2021.
- [6] S. Bjeladinovic, "A Fresh Approach for Hybrid SQL/NoSQL Database Design Based on Data Structuredness," *Enterp. Inf. Syst.*, Vol. 12, No. 8–9, hal. 1202–1220, Okt. 2018.
- [7] G. Hampel, "Preparing the Conceptual Model of a Database," *QUAESTUS Multidiscip. Res. J.*, Vol. 19, hal. 290–304, Jun. 2021.
- [8] J.R. Lourenço, dkk., "Choosing the Right NoSQL Database for the Job: A Quality Attribute Evaluation," *J. Big Data*, Vol. 2, hal. 1–26, Agu. 2015.
- [9] A. Schram dan K.M. Anderson, "MySQL to NoSQL: Data Modeling Challenges in Supporting Scalability," *SPLASH '12: Proc. 3rd Annu. Conf. Syst., Program., Appl.: Softw. Humanit.*, 2012, hal. 191–202.
- [10] L. Schaefer (2021), "What Is NoSQL?" [Online], <https://www.mongodb.com/nosql-explained>, tanggal akses: 6-Sep-2021.
- [11] J. Pokorný, "Functional Querying in Graph Databases," *Vietnam J. Comput. Sci.*, Vol. 5, hal. 95–105, Mei 2018.
- [12] A. Meier dan M. Kaufmann, "NoSQL Databases," dalam *SQL & NoSQL Databases*, Wiesbaden, Jerman: Springer Fachmedien Wiesbaden, 2019, hal. 201–218.
- [13] A. Haseeb dan G. Pattun, "A Review on NoSQL: Applications and Challenges," *Int. J. Adv. Res. Comput. Sci.*, Vol. 8, No. 1, hal. 203–207, 2017.
- [14] Y. Genkina (2020) "Data Modeling with MongoDB," [Online], https://webassets.mongodb.com/_com_assets/cms/mongodb_data_modeling_with_mongodb-44y55ekiu3.pdf, tanggal akses: 6-Sep-2021.
- [15] P. Atzeni, F. Bugiotti, L. Cabibbo, dan R. Torlone, "Data Modeling in the NoSQL World," *Comput. Stand., Interfaces*, Vol. 67, hal. 1–14, Jan. 2020.
- [16] J. LaBreck dan D. Coupal (2020) "Advanced Schema Design Patterns," [Online], https://webassets.mongodb.com/_com_assets/cms/mongodb_advanced_schema_design_patterns-pndclmx967.pdf, tanggal akses: 6-Sep-2021.
- [17] B. Thalheim, "Conceptual Model Notions – A Matter of Controversy: Conceptual Modelling and its Lacunas," *Enterp. Model., Inf. Syst. Archit. (EMISAJ)*, Vol. 13, hal. 9–27, Feb. 2018.
- [18] T. Semiawan, "User Interface Design Analysis Pertaining to Computational Thinking Framework," *Proc. 8th Int. Conf. Inform., Environ., Energy, and Appl.*, 2019, hal. 238–242.