

## PERANCANGAN DATA BASE KASIR MENGUNAKAN MONGODB

Bagastian<sup>(1)</sup>, Khusnul Khotimah<sup>(2)</sup>, Meitry Ayu Handini<sup>(3)</sup>, Renny Fatimah Az Zahra<sup>(4)</sup>

*Sistem Informasi, Universitas Teknokrat Indonesia*

*Jalan. ZA. Pagar Alam, Labuhan Ratu, Kec. Kedaton, Kota Bandar Lampung, Lampung*

*Email: <https://www.teknokrat.ac.id><sup>1</sup>, [Bbastzaa@gmail.com](mailto:Bbastzaa@gmail.com)<sup>(2)</sup>, [husnul1503@gmail.com](mailto:husnul1503@gmail.com)<sup>(3)</sup>*

*, [meitry386@gmail.com](mailto:meitry386@gmail.com)<sup>4</sup> dan [rennyazzahra9@gmail.com](mailto:rennyazzahra9@gmail.com)<sup>(5)</sup>*

### Abstract

*The new technology of database has moved forward the relational. Now, the massive and unstructured data encourage experts to create a new type of database without using query. One of this technology is called NoSQL (Not Only SQL). One of the developing RDBMS that using this technique is MongoDB, which already supporting data storage technology that is no longer need for structured tables and rigid-typed of data. The schema was made flexible to handle the changes of data. Non relational data bases differ from the relational database model that has been used so far. The NoSQL database is very useful in continuously developing data, where the data is so complex that a relational database can no longer accommodate. MongoDB does not have tables, columns, and rows. In MongoDB there are only collections and documents. Documents contained in mongoDB can have different attributes than other documents even though they are in one collection.*

**Keyword:** *MongoDB, NoSQL, Technology, Database, Query*

### Abstrak

Perkembangan teknologi basis data baru tidak seperti basis data relasional pada umumnya. Kini, jumlah data yang sangat besar dan tidak terstruktur mendorong terciptanya basis data jenis baru yang mengarah pada pemanggilan data tanpa menggunakan query. Salah satu penerapannya adalah yang dikenal dengan basis data dengan NoSQL (Not Only SQL). Salah satu contoh RDBMS yang menerapkan ini adalah MongoDB yang menyediakan pendukung teknologi penyimpanan data yang tidak lagi terdapat hubungan antar tabel dan tidak menyimpan data dalam format tabel kaku. Selain itu, skema tabel dibuat fleksibel mungkin untuk menangani perubahan pada isi data. Basis data non relasional berbeda dengan model basis data relasional yang selama ini digunakan, Basis data NoSQL sangat berguna pada data-data yang terus menerus berkembang, dimana data tersebut sangat kompleks sehingga sebuah basis data relasional tidak lagi bisa mengakomodir. MongoDB tidak memiliki tabel, kolom, dan baris. Pada MongoDB yang ada hanyalah koleksi dan dokumen. Dokumen yang terdapat dalam mongoDB dapat memiliki atribut yang berbeda dengan dokumen lain walaupun berada dalam satu koleksi.

**Kata Kunci:** *MongoDB, NoSQL, Teknologi, Database, Query*

## 1. PENDAHULUAN

Di era milenial seperti saat ini banyak sekali orang-orang yang mekases internet sehingga perkembangan teknologi berkembang sangat pesat. Hasil survey yang telah dilakukan oleh Asosiasi Layanan Jasa Internet Indonesia yang menjadikan akses internet semakin mudah dapat memunculkan aplikasi baru yang berbasis web/aplikasi web (*Web Base Application/Web App*). Menurut Lofberg & Molin di jurnal mereka yang berjudul "*Web vs Standalone Application*" dalam Solahudin menyebutkan bahwa aplikasi web merupakan bentuk lanjutan dari pengembangan web page yang sebelumnya hanya ditampilkan informasi berupa statis [1]. Basis data non relasional berbeda dengan model basis data relasional yang selama ini digunakan, basis data non relasional menggunakan beberapa metode yang berbeda-beda.

Perkembangan Big Data di Indonesia terus meningkat dan hampir menyentuh segala aspek kehidupan. Mulai dari perindustrian, kesehatan, E-commerce, media sosial, bahkan pemerintahan. Dengan munculnya istilah Big Data ini maka diperlukan model data yang memiliki skema yang fleksibel dimana basis data relasional yang model datanya berupa tabel dan memiliki keterbatasan dalam menangani jumlah data yang besar dan beragam. Untuk itu diperlukan kelas basis data dan model data yang baru, sehingga istilah "NoSQL" diciptakan. Model data NoSQL tidak berupa tabel yang tersusun atas baris dan kolom. Namun, basis data NoSQL bisa berupa dokumen, grafik atau nilai kunci [2]. Seiring perkembangan zaman sudah banyak basis data NoSQL yang populer, salah satunya adalah MongoDB. MongoDB adalah basis data NoSQL yang bersifat document based. MongoDB bersifat document based artinya MongoDB tidak memiliki tabel, kolom ataupun baris. MongoDB hanya memiliki koleksi dan dokumen. Data yang disimpan dalam basis data MongoDB berupa file JSON yang disebut dengan istilah BSON (Binary JSON). Sistem basis data MongoDB menggunakan key-value, artinya setiap dokumen dalam MongoDB dipastikan memiliki key [3].

Metode ini bergantung dari jenis basis data yang digunakan seperti Basis data NoSQL. Basis data NoSQL sangat berguna pada data-data yang terus menerus berkembang, dimana data tersebut sangat kompleks sehingga sebuah basis data relasional tidak lagi bisa mengakomodir. Salah satu bentuknya adalah ketika suatu data saling berhubungan satu sama lain maka akan muncul proses duplikasi data. Data tersebut saling memanggil ke beberapa permintaan, tambahan data baru, perubahan data dan lainlain dengan key yang sama karenafaktor hubungan antar data yang sama terjadi terus menerus, mendorong faktor redudansi data, data menjadi berlipat-lipat dan pada akhirnya

akan menyebabkan crash pada basis data berkonsep RDBMS.

Dalam basis data NoSQL meliputi hirarki, graf dan basis data berorientasi objek. Penggunaan basis data non relasional kembali marak seiring bertambahnya aplikasi berbasis web yang menuntut skalabilitas tinggi meskipun memiliki kelemahan pada redundansi dan konsistensi data, namun basis data non relasional dapat menyelesaikan permasalahan yang berkaitan dengan availability dan partition tolerance. Dari penjelasan diatas bahwa hal ini dianggap membuat pengelolaan data menggunakan MongoDB lebih baik dari pada SQL.

Munculah permasalahan seperti Banyaknya proses pengolahan data yang harus dilakukan dalam waktu yang singkat, Pada pengolahan data tersebut akan didapatkan pula data-data transaksi berupa data supplier, faktur, customer, barang dan jenis transaksi yang didalamnya masih banyak pula terdapat data yang harus diproses. Jadi dapat dirumuskan Bagaimana perbedaan antara SQL dengan Mongoddb untuk permasalahan diatas.

## 2. METODE PENELITIAN

Metode penelitian dilakukan dengan metode *Systematic Literature Review* yaitu dengan mengidentifikasi, mengevaluasi dan menginterpretasi penelitian-penelitian yang sudah ada sebelumnya seputar basis data MongoDB untuk menjawab pertanyaan-pertanyaan :

1. Karakteristik basis data MongoDB termasuk perbedaannya dengan basis data relasional
2. Kelebihan dan kekurangan basis data MongoDB
3. Implementasi Penggunaan MongoDB

## 3. HASIL DAN PEMBAHASAN

### 3.1 Karakteristik basis data MongoDB termasuk perbedaannya dengan basis data relasional

Didalam database relasional data disimpan dalam satu entitas yang dapat saling berhubungan. Perhubungan tersebut disebut relasi. didalam entitas terdiri dari beberapa atribut yang akan menjadi nama kolom didalam sebuah tabel yang akan dikaitkan dengan domain tertentu seperti integer atau string. Sedangkan MongoDB data disimpan dengan berorientasi sebuah dokumen yang memiliki format JSON atau BSON. Dokumen juga mendukung penggunaan seperti lists, pointers, embedded arrays atau nested document sehingga menyederhanakan akses data.

MongoDB merupakan sistem basis data yang menggunakan konsep keyvalue, artinya setiap dokumen dalam mongoddb pasti memiliki key. Hal ini berbeda dalam RDBMS yang bisa tidak

menggunakan primary key ketika membuat sebuah tabel. Sehingga walaupun kita membuat sebuah dokumen tanpa menggunakan primary key, tapi secara otomatis mongoddb memberinya sebuah key. Penggunaan konsep key-value sangat berperan penting, karena hal ini membuat mongoddb menjadi sistem basis data yang sangat cepat jika dibandingkan dengan non key-value seperti RDBMS

Dengan dipertimbangkannya ukuran data penggunaan memori dan efisiensi bandwidth ketika dilakukannya proses transmisi data maka dalam MongoDB maksimal penyimpanan data hanya sampai 16 MB. Limitasi penyimpanan dapat diatasi dengan GridFS yang memecah data menjadi beberapa data berukuran kecil. MongoDB memiliki berbagai macam fitur, seperti Map reduce based. Aggregation Framework, hampir mirip dengan Group By didalam MySQL, dalam MongoDB digunakan untuk agregasi, umumnya digunakan untuk memproses data dalam volume besar secara parallel.

### 3.2 Kelebihan dan Kekurangan MongoDB

#### 3.1.1 Kelebihan MongoDB

1. Performa yang ditawarkan MongoDB lebih cepat disebabkan oleh memcached dan format dokumennya yang berbentuk seperti JSON
2. Kita tidak perlu membuat struktur tabel, karena MongoDB akan otomatis membuatkan struktur tabelnya pada saat proses insert (fleksibel skema) .

#### 3.1.2 Kekurangan MongoDB

1. Belum banyak hosting yang support.
2. Fleksibilitas dalam query (sebagai contoh tidak adanya JOIN).
3. Selain itu, karena bervariasinya produk dan format penyimpanan, berpindah antar satu produk database ke produk noSQL lainnya perlu waktu untuk belajar.

Contohnya ketika anda pindah dari MongoDB ke Cassandra, maka anda harus belajar lagi dari awal, berbeda dengan database RDMS.

### 3.3 Implementasi Penggunaan MongoDB

#### 3.3.1 Tampilan dari tabel Detail\_transaksi

Tabel 1. Tabel Detail Transaksi

Key	Type	Deskripsi
-----	------	-----------

_id	Objectid	Collection Primary Key
total_bayar	int	Total bayar cust
tanggal	String	Tanggal transaksi
total_uang	int	Total uang

Tabel 1 menunjukkan dokumen detail\_transaksi yang menjelaskan deskripsi, tipe serta kunci setiap atribut yang ada pada detail transaksi.

Tabel 2. Tabel Barang

Key	Type	Deskripsi
-----	------	-----------

_id	String	Collection Primary Key
nama	String	Nama barang
harga_beli	Int	Harga beli di supplier
harga_jual	Int	Harga jual kepada cust
stok	String	Stok barang di gudang

Tabel 2 menunjukkan dokumen barang yang menjelaskan deskripsi, tipe serta kunci setiap atribut yang ada pada tabel barang.

Tabel 3. Tabel Suplier

Key	Type	Deskripsi
_id	String	Collection Primary Key
nama_supplier	String	Nama supplier
no_telp	Int	No telepon supplier
alamat	String	Alamat supplier

Tabel 3 menunjukkan dokumen Supplier yang menjelaskan deskripsi, tipe serta kunci setiap atribut yang ada pada tabel supplier.

Tabel 4. Tabel Faktur

Key	Type	Deskripsi
_id	String	Collection Primary Key
supplier_id	String	Collection Primary Key
barang_id	String	Collection Primary Key
nama_barang	String	Nama barang
qty	Int	Jumlah barang dari supplier
jumlah	int	

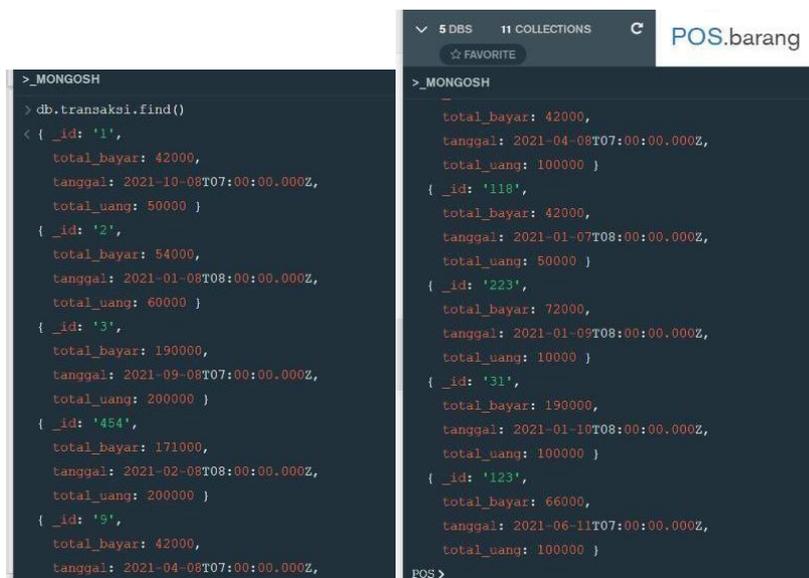
		Jumlah yang masuk digudang
--	--	----------------------------

Tabel 4 menunjukkan dokumen faktur yang menjelaskan deskripsi, tipe serta kunci setiap atribut yang ada pada tabel faktur.

Tabel 5. Tabel Customer

Key	Type	Deskripsi
_id	String	Collection Primary Key
nama_pembeli	String	Nama pembeli
jenis_kelamin	String	Jenis kelamin
no_telp	Int	No telepon customer

Berikut tampilan pengimplementasian nya didalam MongoDB:



```

>_MONGOSH
> db.transaksi.find()
< { _id: '1',
  total_bayar: 42000,
  tanggal: 2021-10-08T07:00:00.000Z,
  total_uang: 50000 }
{ _id: '2',
  total_bayar: 54000,
  tanggal: 2021-01-08T08:00:00.000Z,
  total_uang: 60000 }
{ _id: '3',
  total_bayar: 190000,
  tanggal: 2021-09-08T07:00:00.000Z,
  total_uang: 200000 }
{ _id: '454',
  total_bayar: 171000,
  tanggal: 2021-02-08T08:00:00.000Z,
  total_uang: 200000 }
{ _id: '9',
  total_bayar: 42000,
  tanggal: 2021-04-08T07:00:00.000Z,
  total_uang: 100000 }
  
```

```

POS.barang
>_MONGOSH
total_bayar: 42000,
tanggal: 2021-04-08T07:00:00.000Z,
total_uang: 100000 }
{ _id: '118',
  total_bayar: 42000,
  tanggal: 2021-01-07T08:00:00.000Z,
  total_uang: 50000 }
{ _id: '223',
  total_bayar: 72000,
  tanggal: 2021-01-09T08:00:00.000Z,
  total_uang: 10000 }
{ _id: '31',
  total_bayar: 190000,
  tanggal: 2021-01-10T08:00:00.000Z,
  total_uang: 100000 }
{ _id: '123',
  total_bayar: 66000,
  tanggal: 2021-06-11T07:00:00.000Z,
  total_uang: 100000 }
POS >
  
```

Gambar 1. Menampilkan seluruh data dari collection transaksi

```
> db.transaksi.find().sort({tanggal:-1}) .limit(3)
< { _id: '1',
  total_bayar: 42000,
  tanggal: 2021-10-08T07:00:00.000Z,
  total_uang: 50000 }
  { _id: '3',
  total_bayar: 190000,
  tanggal: 2021-09-08T07:00:00.000Z,
  total_uang: 200000 }
  { _id: '123',
  total_bayar: 66000,
  tanggal: 2021-06-11T07:00:00.000Z,
  total_uang: 100000 }
POS>
```

Gambar 2. Menampilkan data tabel Transaksi secara descending berdasarkan tanggal

```
> _MONGOSH
> db.barang.find({"stok":{$in:[10,11,20]}})
< { _id: '51',
  nama: 'penghapus',
  harga_beli: 5000,
  harga_jual: 6000,
  stok: 11 }
  { _id: '21',
  nama: 'pensil',
  harga_beli: 12000,
  harga_jual: 18000,
  stok: 10 }
  { _id: '22',
  nama: 'stabilo merah',
  harga_beli: 5000,
  harga_jual: 9000,
  stok: 20 }
  { _id: '50',
  nama: 'penghapus',
  harga_beli: 5000,
  harga_jual: 6000,
```

Gambar 3. Menampilkan data tabel Transaksi secara ascending berdasarkan tanggal

```
>_MONGOSH
-
  nama: 'pensil',
  harga_beli: 12000,
  harga_jual: 18000,
  stok: 10 }
{ _id: '22',
  nama: 'stabilo merah',
  harga_beli: 5000,
  harga_jual: 9000,
  stok: 20 }
{ _id: '50',
  nama: 'penghapus',
  harga_beli: 5000,
  harga_jual: 6000,
  stok: 11 }
{ _id: '588',
  nama: 'penghapus',
  harga_beli: 5000,
  harga_jual: 6000,
  stok: 11 }
POS >
```

Gambar 4. Menampilkan jumlah stok yang ingin di cari

```
> db.transaksi.aggregate([{$group:{_id:"$id",total_bayar:{$last:"$total_uang"}},{$sort:{total_bayar:-1}}])
< { _id: '3', total_bayar: 200000 }
  { _id: '454', total_bayar: 200000 }
  { _id: '9', total_bayar: 100000 }
  { _id: '31', total_bayar: 100000 }
  { _id: '123', total_bayar: 100000 }
  { _id: '2', total_bayar: 60000 }
  { _id: '1', total_bayar: 50000 }
  { _id: '118', total_bayar: 50000 }
  { _id: '223', total_bayar: 10000 }
POS >
```

Gambar 5. Menampilkan seluruh data dalam tabel supplier

```
>_MONGOSH
> db.supplier.find()
< { _id: '12341',
  nama_supplier: 'ira',
  no_telp: '087688997766',
  alamat: 'sukarame' }
{ _id: '12342',
  nama_supplier: 'ayu',
  no_telp: '089766445533',
  alamat: 'tanjung karang' }
{ _id: '12343',
  nama_supplier: 'dini',
  no_telp: '08997788654',
  alamat: 'sukabumi' }
{ _id: '12344',
  nama_supplier: 'renny',
  no_telp: '08995432123',
  alamat: 'teluk betung' }
{ _id: '12345',
  nama_supplier: 'fatimah',
  no_telp: '08223456543',
  alamat: 'sukabumi' }
{ _id: '12346',
  nama_supplier: 'zahra',
  no_telp: '087655443322',
  alamat: 'sukabumi' }
{ _id: '12347',
  nama_supplier: 'bagas',
  no_telp: '0895352714865',
  alamat: 'pahoman' }
{ _id: '12348',
  nama_supplier: 'meimei',
  no_telp: '0895352765432',
  alamat: 'wayhalim' }
{ _id: '12349',
  nama_supplier: 'mei',
  no_telp: '0895570714861',
  alamat: 'panjang' }
{ _id: '12310',
  nama_supplier: 'rey',
  no_telp: '089788996677',
  alamat: 'katibung' }
```

Gambar 6. Mengurutkan bayar berdasarkan total\_uang terbesar sampai terkecil (*descending*)

```
> db.transaksi.aggregate([{$group: {_id: "$_id", total_bayar: {$last: "$total_uang"}}, {$sort: {total_bayar: 1}}])
< { _id: '223', total_bayar: 10000 }
  { _id: '1', total_bayar: 50000 }
  { _id: '118', total_bayar: 50000 }
  { _id: '2', total_bayar: 60000 }
  { _id: '9', total_bayar: 100000 }
  { _id: '31', total_bayar: 100000 }
  { _id: '123', total_bayar: 100000 }
  { _id: '3', total_bayar: 200000 }
  { _id: '454', total_bayar: 200000 }
POS>
```

Gambar 7. Mengurutkan bayar berdasarkan total\_uang terkecil ke terbesar (*ascending*)

#### 4. KESIMPULAN

Berdasarkan hasil dan pembahasan dan studi kepustakaan diketahui bahwa MongoDB merupakan basis data NoSQL berbasis dokumen (document based). MongoDB menyimpan datanya dalam bentuk file seperti JSON yang disebut BSON (Binary JSON). MongoDB memiliki fitur-fitur seperti lists, pointers, embedded arrays atau nested document sehingga menyederhanakan akses data, Map reduce based Aggregation Framework untuk agregasi, Schema Less Database untuk struktur data yang fleksibel, Replication and fail-over support untuk mendukung replikasi data. Dari data yang didapatkan, bisa dikatakan bahwa MongoDB memiliki performansi yang baik dibandingkan dengan MySQL.

Dari perusahaan-perusahaan yang memilih MongoDB sebagai basis datanya, bisa disimpulkan alasan kenapa mereka memilih MongoDB :

- MongoDB memiliki fleksibilitas yang tinggi karena memiliki fitur Schema Less Database
- Memiliki perfoma yang bagus
- High availability
- Memiliki banyak fitur meningkatkan skalabilitas dan fleksibilitas
- Mudah dalam penggunaan, penyetelan dan di maintain.
- Mampu mengolah data dalam jumlah yang sangat besar dan data yang tidak tetap.

#### UCAPAN TERIMAKASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini. peneliti ingin mengucapkan terima kasih kepada semua teman-teman seperjuangan, terutama yang memberikan semangat dan dorongan kepada peneliti untuk menyelesaikan penelitian ini secepatnya. Penelitian ini dapat diselesaikan dengan maksimal berkat kerjasama dan bantuan dari berbagai pihak,

oleh karena itu peneliti mengucapkan terima kasih kepada semua pihak yang telah berkontribusi secara maksimal dalam menyelesaikan penelitian ini. Peneliti sebagai manusia biasa menyadari sepenuhnya bahwa masih banyak kekurangan dalam penelitian ini, baik dari segi tata bahasa, struktur kalimat maupun isi. Oleh karena itu, dengan segala kerendahan hati, penulis sebagai peneliti menerima segala kritik dan saran yang membangun dari para pembaca.

## REFERENCES

- M. A. I. K. D. N. Solahudin, "Rancang Bangun Sistem Informasi Staycation Berbasis Web dengan Implementasi Teknologi Mern Stack," *Manajemen Informatika*, vol. 12, no. 1, p. 12, 2021.
- Fathansyah, *Basis Data Revisi Ketiga*, Bandung: Informatika, 2018.
- F. Putra. E.K & Rahmayeni, "IMPLEMENTASI DATABASE MONGODB UNTUK SISTEM INFORMASI BIMBINGAN KONSELING BERBASIS WEB," *Teknoif*, pp. 67-73, 2016.
- Al fatta, Hanif, 2007, Analisis dan perancangan system informasi untuk keunggulan perusahaan dan organisasi kelas dunia, Andi offset – STMIKAMIKOM Yogyakarta, Yogyakarta
- Nugroho, Bunafit 2004, Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL, Gaya Media, Yogyakarta
- Bradshaw, S., Brazil, E., & Kristina, C. (2019). MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. Dalam S. Bradshaw, E. Brazil, & C. Kristina, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage* (hal. 514). Sebastopol: O'Reilly Media, Inc.
- Deari, R., Zenuni, X., Ajdari, J., Ismaili, F., & Raufi, B. (2018). ANALYSIS AND COMPARISON OF DOCUMENT-BASED DATABASES WITH SQL RELATIONAL DATABASES: MONGODB VS MYSQL. *Proceedings of the International Conference on Information Technologies*, 1-10.
- Farozi, M. (2019). DESAIN BASIS DATA NON RELASIONAL NOSQLMONGODB PADA WEBSITE SISTEM INFORMASI AKADEMIK. *Jurnal Sistem Informasi Komputer dan Teknologi Informasi (SISKOMTI)*, 24-39.
- Fathansyah. (2018). *BASIS DATA Revisi Ketiga*. Bandung: INFORMATIKA.
- Junaidi, A. (2016). Studi Perbandingan Performansi Antara MongoDB. *ANNUAL RESEARCH SEMINAR 2016*, 460- 465.
- Kurniawan, A. K., Pramukantoro, E. S., & Trisnawan, P. H. (2019). Perbandingan Kinerja Cassandra dan MongoDB Sebagai Backend IoT Data Storage. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 364-371.
- Putra, E. K., & Rahmayeni, F. (2016). IMPLEMENTASI DATABASE MONGODB UNTUK SISTEM INFORMASI BIMBINGAN KONSELING BERBASIS WEB. *TEKNOIF*, 67-73.
- Silalahi, M., & Wahyudi, D. (2018). PERBANDINGAN PERFORMANSI DATABASE MONGODB DAN MYSQL DALAM APLIKASI FILE MULTIMEDIA BERBASIS WEB. *Computer Based Information System Journal*, 65-77.
- Winaya, I. G., & Ashari, A. (2016). Transformasi Skema Basis Data Relasional Menjadi Model Data Berorientasi Dokumen pada MongoDB. *IJCCS*, 47-58.