

Analisis dan Penerapan *Database* Mongoddb pada Aplikasi Manajemen Dokumen di PT. XYZ

Yulius^{1*}, Yeremia Alfa Susetyo²

^{1,2} Program Studi Teknik Informatika Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

article info

Article history:

Received 17 March 2023

Received in revised form

25 May 2023

Accepted 20 August 2023

Available online October 2023

DOI:

<https://doi.org/10.35870/jti.k.v7i4.1047>

Keywords:

Flask; Python; Document Management; MongoDB; NoSQL Database.

abstract

The development of NoSQL databases has become increasingly popular due to the need for diverse data storage systems that can accommodate varying attributes. MongoDB is one such NoSQL database that uses a document data model with JSON data types, allowing for storing data with diverse attributes. This study aims to analyze and implement MongoDB in a document management application at PT. XYZ uses the waterfall method. The discussion results include the application system architecture, the advantages of MongoDB such as its flexible schema and lack of downtime during schema changes, and its drawbacks such as limited web hosting support. The study also covers the implementation of MongoDB CRUD operations in the document management application and system testing using BlackBox Testing. Based on the research, it can be concluded that using MongoDB in the document management application at PT. XYZ provides an effective and efficient solution for managing documents.

abstrak

Perkembangan basis data NoSQL sudah cukup popular saat ini. Hal ini dikarenakan kebutuhan sistem penyimpanan data yang beragam sehingga diperlukan basis data yang dapat menyimpan data dengan atribut yang beragam. Salah satu basis data NoSQL adalah MongoDB. MongoDB menggunakan model data dokumen dengan tipe data BSON (Binary JSON) sehingga dapat melakukan penyimpanan data dengan atribut yang beragam. Penelitian ini bertujuan untuk menganalisis dan menerapkan database MongoDB pada aplikasi manajemen dokumen di PT. XYZ. Metode penelitian yang digunakan adalah metode waterfall. Hasil pembahasan mencakup arsitektur sistem aplikasi, kelebihan MongoDB yaitu skema yang fleksibel dan tidak melalui proses downtime setiap perubahan skema sedangkan kekurangan MongoDB yaitu belum memiliki banyak hosting web yang mendukung, implementasi CRUD MongoDB pada aplikasi manajemen dokumen, dan pengujian sistem menggunakan BlackBox Testing. Berdasarkan penelitian, disimpulkan penggunaan MongoDB dalam aplikasi manajemen dokumen di PT. XYZ memberikan solusi yang efektif dan efisien dalam mengatasi masalah pengelolaan dokumen.

Corresponding Author. Email: yiyus49@gmail.com ^{1}.

© E-ISSN: 2580-1643.

Copyright @ 2023. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISET)
(<http://creativecommons.org/licenses/by/4.0/>).

1. Latar Belakang

Perkembangan teknologi yang semakin canggih membuat penerapan teknologi diberbagai bidang sangat diperlukan. Penerapan teknologi dalam setiap bidang meningkatkan kemampuan dan efisiensi dalam melakukan setiap pekerjaan. Setiap perusahaan memiliki dokumen dengan jumlah yang besar. Kumpulan dokumen dalam jumlah besar cukup sulit untuk dikelola. Karenanya, dibutuhkan sebuah sistem untuk mengelola dokumen secara efektif dan efisien. Dengan menerapkan teknologi, maka proses komputasi dan pengolahan data semakin cepat dan efektif. Hal ini yang melatar belakangi penelitian ini [1].

Salah satu bidang yang memerlukan proses manajemen dokumen yaitu bidang industri ritel. PT.XYZ merupakan salah satu perusahaan yang bergerak di bidang industri ritel besar di Indonesia dan sudah berdiri selama lebih dari 20 tahun serta memiliki lebih dari 16.000 toko yang aktif setiap harinya dengan total transaksi sekitar 5 juta perhari. PT XYZ tidak hanya memiliki toko di Indonesia. PT.XYZ telah menerapkan proses penyimpanan berkas dalam bentuk digital yang dikelompokkan berdasarkan kategori. Saat ini proses penyimpanan data dalam perusahaan ini masih menggunakan teknologi basis data SQL yang memerlukan pembentukan kolom terlebih dahulu dalam pembuatan data baru. Hal ini kurang efektif apabila diperlukan perubahan atau pembuatan data baru yang memiliki atribut yang beragam. Oleh karena itu, diperlukan proses penyimpanan data yang dinamis, sehingga dapat menyimpan data dengan atribut yang beragam.

Saat ini, basis data NoSQL sudah banyak dipergunakan yaitu salah satunya adalah MongoDB. Berbeda dengan basis data SQL yang menggunakan kolom dan tabel, MongoDB bersifat *document-based* yang tidak menggunakan kolom dan tabel namun menggunakan koleksi dan dokumen yang berupa dokumen JSON (*JavaScript Object Notation*) yang sering disebut sebagai BSON (*Binary JSON*). Karena menggunakan JSON maka setiap data yang disimpan dalam basis data MongoDB memiliki *key and value* [3]. Salah satu keunggulan dari MongoDB dengan basis data *SQL* terdapat pada pengubahan skemanya. Pada basis data *SQL*, setiap pengubahan skema data harus

melalui proses *downtime* (basis data tidak dapat diakses), sedangkan pada basis data *No SQL*, pengubahan skema data dapat langsung dilakukan tanpa harus melalui *downtime* karena sifat *document-based* yang dimiliki MongoDB.

PT.XYZ membutuhkan sistem penyimpanan dokumen untuk menampung kategori dokumen yang memiliki atribut yang beragam. Dengan menggunakan basis data *No SQL* maka proses memasukkan dokumen baru dengan kategori / atribut yang beragam dapat dilakukan dengan mudah. Basis data *MongoDB* sangat cocok diterapkan pada Aplikasi Manajemen Dokumen di PT.XYZ. Berdasarkan beberapa permasalahan yang telah diuraikan di atas, maka rumusan masalah dalam penelitian ini adalah bagaimana menganalisis dan menerapkan *Database MongoDB* pada Aplikasi Manajemen Dokumen di PT. XYZ.

Penelitian dengan inti masalah manajemen dokumen pernah dilakukan oleh I Putu Susila Handika dan I Gede Totok Suryawan pada jurnal berjudul ‘Rancang Bangun *Document Management System* Untuk Mengelola Dokumen Standart Operational Procedure’ yang dipublikasikan pada tahun 2017. Pada penelitian ini menghasilkan aplikasi manajemen dokumen yang menggunakan basis data relasional dan menggunakan Bahasa pemrograman PHP. Penelitian ini berfokus pada pembangunan aplikasi manajemen dokumen pada perusahaan. *Output* dari penelitian ini adalah pembangunan aplikasi manajemen dokumen dengan menggunakan bahasa pemrograman PHP dan basis data relasional. Permasalahan yang terjadi adalah sistem pengiriman dokumen yang masih kurang efektif dan tradisional sehingga dibutuhkannya sistem yang dapat menyimpan dan mengirimkan dokumen sesuai dengan *Standart Operational Procedure* [4].

Selanjutnya penelitian dengan menggunakan teknologi MongoDB pernah dilakukan oleh Muhammad Alkaff, Iphan Fitrian Radam, dan Sugiantoro pada penelitian berjudul “Rancang Bangun Sistem Identifikasi Arus Lalu Lintas pada Simpang Tiga Menggunakan *Database NoSQL*” pada tahun 2020. Penelitian ini berfokus pada pembangunan aplikasi sistem identifikasi arus lalu lintas simpang tiga. Penelitian ini menghasilkan sistem identifikasi arus lalu lintas menggunakan *framework* PHP yaitu Laravel dan berfokus pada penggunaan *sort*

dan *filter* data. *Output* dari penelitian ini adalah aplikasi sistem identifikasi arus lalu lintas simpang tiga dengan menggunakan *framework* Laravel dan basis data *NoSQL* MongoDB. Penelitian ini membahas masalah pengolahan data simpang tiga yang masih menggunakan bantuan software pengolah angka *Microsoft Excel* dan terdapat proses manual pada *sort* dan *filter* serta tidak memiliki riwayat. Penggunaan basis data *NoSQL* karena data yang disajikan kompleks dan memiliki beberapa variable [5].

Penerapan MongoDB pernah dilakukan oleh Nunit Prihatoni Siregar, Kemas Rahmat dan Akbar Alfian pada penelitian berjudul “Analisis dan Implementasi Basis Data Terdistribusi *Horizontal* pada MongoDB untuk ClikKB BKKBN Regional Jawa Barat *Analysis*”. Yang membahas mengenai masalah data semi terstruktur yang memiliki atribut yang berbeda-beda. Penelitian ini menerapkan MongoDB pada aplikasi ClikKB BKKBN Regional Jawa Barat yang menggunakan *sharding* untuk mendistribusikan data ke dalam sekelompok node. Pada penelitian ini menerapkan basis data *NoSQL* yaitu *MongoDB* untuk pengembangan aplikasi, karena basis data *NoSQL* didesain khusus untuk memecahkan permasalahan *scalability* dan *reliability* pada sebuah aplikasi. Contoh kasusnya adalah data *Nonuniform*, yaitu data semi terstruktur atau sebuah data yang memiliki atribut yang beragam [6].

Berdasarkan dengan penelitian dan permasalahan yang ada, penerapan MongoDB dan Flask pada aplikasi manajemen dokumen masih sedikit ditemukan. Oleh karena itu penulis membangun satu aplikasi berbasis website menggunakan bahasa pemrograman Python dan menggunakan *framework* Flask di dalamnya. Penulis juga menggunakan basis data MongoDB yang merupakan salah satu *Schemaless Database*. *Schemaless Database* merupakan basis data yang tidak memerlukan pembentukan skema dan tabel di awal. Tujuan dari penelitian ini adalah untuk menganalisis dan menerapkan *Database MongoDB* pada Aplikasi Manajemen Dokumen di PT. XYZ.

Terdapat beberapa landasan teori yang digunakan pada penelitian ini yaitu Dokumen, *Python*, Flask, MongoDB, dan Google Cloud Storage. Dokumen terbagi menjadi dua yaitu sebagai kata kerja dan kata benda. Sebagai kata benda, dokumen merupakan setiap benda yang memuat rekaman informasi.

Sedangkan sebagai kata kerja, dokumen merupakan kegiatan merekam, mencatat dan membuat menjadi dokumen [7]. Dokumen digital merupakan data yang dapat disimpan dan ditransmisikan dalam bentuk kode biner. Dokumen tersebut dapat diakses, dibuat dan dihapus menggunakan komputer yang dapat membaca data biner serta mengolahnya sehingga data tersebut dapat digunakan dan dimanfaatkan [8].

Python merupakan bahasa pemrograman tingkat tinggi yang mendukung pemrograman berorientasi objek dinamis dan dapat digunakan untuk pengembangan berbagai program seperti *website*, *android*, kecerdasan buatan, dan program basis data. Python bersifat *open source* artinya terdapat kebebasan dalam pendistribusian dan penyalinannya [9][10].

Flask merupakan *framework* yang ditulis menggunakan bahasa pemrograman Python yang memiliki fleksibilitas dan skalabilitas yang tinggi [11]. Flask merupakan *framework* yang ringan dan mudah untuk dikembangkan [9]. Flask memungkinkan pengguna untuk membuat program lebih terstruktur.

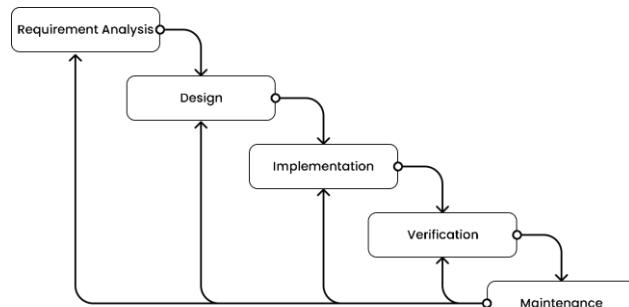
MongoDB merupakan salah satu basis data *NoSQL* yang dapat melakukan *query* pada berbagai bidang berbasis dokumen [12]. *NoSQL* merupakan kependekan dari “*Not Only SQL*” yang memiliki arti tidak hanya menggunakan sintaks SQL. MongoDB berbasis dokumen yang berarti tidak memiliki kolom, baris dan tabel namun menggunakan dokumen dan koleksi [2]. MongoDB menyimpan data dalam dokumen JSON atau disebut BSON (Binary JSON) [13]. BSON merupakan JSON yang memiliki tipe data seperti *float*, *integer*, *string*, *date*, *Boolean* dan sebagainya [14]. Karena berbentuk dokumen JSON, maka penyimpanan data dalam MongoDB menggunakan *key-value* sehingga setiap data pasti memiliki *key* [12].

Google Cloud Storage merupakan layanan penyimpanan data dari Google secara online dengan memanfaatkan koneksi internet untuk mengakses data. Berbeda dengan penyimpanan data pada perangkat keras, penyimpanan data secara online tidak membutuhkan perangkat tambahan dan hanya memerlukan koneksi internet saja [15]. *Google Cloud Storage* memiliki kelebihan yaitu diantaranya, *fleksibilitas* yang tinggi dan *efisiensi*. *Google Cloud Storage* merupakan salah satu layanan penyimpanan data dalam sistem komputasi

awan. Google cloud Storage merupakan salah satu penyimpanan berbasis object (*Object Storage*).

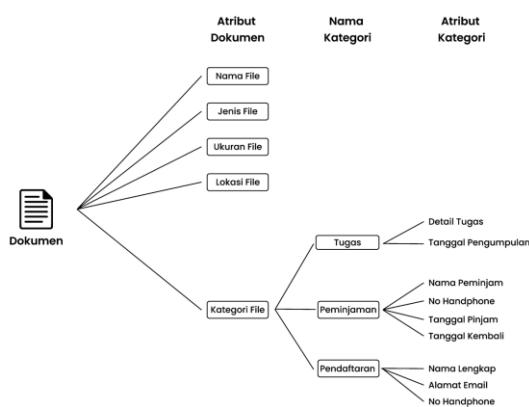
2. Metode Penelitian

Dalam penelitian Analisis dan Penerapan *Database MongoDB* Pada Aplikasi Manajemen Dokumen di PT. XYZ menggunakan metode penelitian *waterfall*, terdapat beberapa tahapan yang dilaksanakan dalam model *waterfall* untuk mencapai tujuan penelitian dan kesimpulan yang akurat. Proses penelitian ini penulis lakukan secara terurut seperti yang ada pada gambar 1.



Gambar 1. Metode *Waterfall*

Terdapat 5 tahapan dalam penelitian ini yaitu Analisis Kebutuhan, Desain, Implementasi, Verifikasi atau Pengujian, dan Pemeliharaan. Pada tahap pertama, dilakukan pengamatan untuk mengetahui kebutuhan sistem yang akan dibangun. Kemudian akan dilakukan analisa untuk mencari solusi yang tepat sehingga penerapan basis data *MongoDB* sesuai dengan kebutuhan.



Gambar 2. Visualisasi Aplikasi Manajemen Dokumen PT. XYZ

Pada penelitian kali ini data yang diperlukan untuk pembangunan aplikasi adalah: Data kategori

dokumen: yaitu merupakan data lengkap kategori dokumen yang ada di PT.XYZ seperti nama kategori dan tipe kategori. Data dokumen: yaitu merupakan data lengkap dokumen yang ada di PT.XYZ seperti nama dokumen, jenis dokumen, kategori dokumen, lokasi dokumen dan besaran dokumen. Data pengguna: yaitu data lengkap pengguna yang dapat mengakses aplikasi ini seperti nama, *username*, dan *password*.

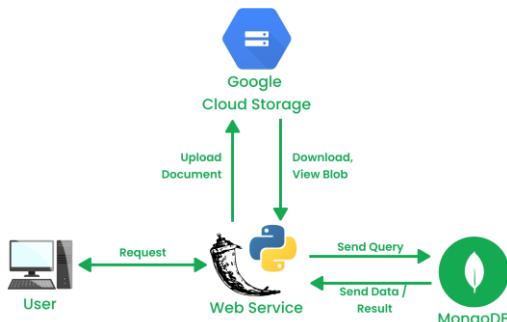
Tahapan kedua yaitu desain, Tahap ini adalah membuat desain dari aplikasi yang akan dibangun. Pembuatan desain aplikasi bertujuan untuk membantu menentukan persyaratan sistem dan mendefinisikan arsitektur sistem secara menyeluruh [16]. Selanjutnya pada tahap implementasi akan dilakukan penulisan kode guna membangun sistem berdasarkan hasil desain yang dilakukan di tahap sebelumnya. Pada penelitian ini, pembangunan aplikasi akan menggunakan bahasa pemrograman *python* serta menggunakan *framework flask*. Untuk basis data, akan menggunakan basis data *MongoDB* dan *Google Cloud Storage*. Setelah aplikasi berhasil melalui tahap implementasi, tahapan selanjutnya adalah proses verifikasi atau pengujian. Pada tahap ini akan dilakukan pengecekan oleh pihak perusahaan guna memastikan apakah penerapan basis data *MongoDB* pada aplikasi sudah sesuai dengan persyaratan dan kebutuhan sistem. Pengecekan ini juga bertujuan untuk memastikan semua modul dapat berjalan dengan benar. Tahap terakhir adalah melakukan perbaikan kesalahan yang ditemukan saat melakukan pengecekan di tahapan sebelumnya. Setelah dilakukan pengecekan dan ditemukan kesalahan, maka akan kembalikan ke pengembang agar dilakukan perbaikan pada kesalahan tersebut. Hal ini bertujuan untuk mendapatkan aplikasi yang terbaik.

3. Hasil dan Pembahasan

Arsitektur Sistem

Aplikasi Manajemen Dokumen ini memiliki peran penting dalam penyimpanan dokumen pada perusahaan. Dalam proses penyimpanan, aplikasi ini memiliki fungsi CRUD yaitu *Create*, *Read*, *Update* dan *Delete*. Fungsi *create* dapat ditemukan saat pengguna melakukan proses *upload* dokumen baru serta saat pengguna menambahkan nama kategori dokumen baru kedalam aplikasi. Fungsi *read* digunakan pada saat

menampilkan data dokumen. Fungsi *update* dapat ditemukan pada saat pengguna melakukan perubahan data pada dokumen seperti memasukan kategori dokumen. Fungsi *delete* digunakan pada saat pengguna menghapus dokumen ataupun kategori yang ada dalam aplikasi.



Gambar 3. Arsitektur sistem

Gambar 3 merupakan sebuah arsitektur dari aplikasi manajemen dokumen dengan aktornya yaitu pengguna atau *user*. Proses dimulai dengan pengguna yang melakukan permintaan melalui web service yang dibangun menggunakan *framework* Flask dan bahasa pemrograman Python. Kemudian, Flask akan melakukan proses upload dokumen ke dalam Google Cloud Storage dan menerima hasil berupa objek. Objek tersebut kemudian diteruskan oleh *web service* ke *database* MongoDB untuk disimpan. Setelah disimpan, *database* MongoDB akan mengirimkan hasil kembali ke pengguna melalui *web service*.

Implementasi Kode Program

Penelitian yang dilakukan menghasilkan sebuah aplikasi yang menggunakan basis data MongoDB, *framework* Flask dan bahasa pemrograman Python.

Kode Program 1. Konfigurasi awal aplikasi pada file `__init__.py`

```

from flask import Flask
from google.cloud import storage
import globalvar, datetime, os
app = Flask(__name__)
app.secret_key = globalvar.SECRET_KEY
app.config["MONGO_URI"] =
globalvar.MONGO_URI
app.config["UPLOADED_PHOTOS_DEST"] =
globalvar.FILE_PATH
app.config['UPLOAD_FOLDER'] =
globalvar.FOLDER_PATH
app.config['PERMANENT_SESSION_LIFETIME'] =
datetime.timedelta(minutes=30)
bucket_name = globalvar.BUCKET_NAME
  
```

```

gv = globalvar.globalvar(app)
bcrypt = Bcrypt(app)
mongo = gv.mongo
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = globalvar.SERVICE_KEY
storage_client = storage.Client()
  
```

Kode program 1 merupakan pengaturan awal untuk aplikasi Flask, menentukan konfigurasi aplikasi secara keseluruhan, membuat objek Flask, menentukan kredensial akses Google Cloud Storage dan menginisialisasi variabel global aplikasi. Kode Ini juga memuat modul yang dibutuhkan seperti *globalvar*, *datetime*, dan *os*.

Kode Program 2. Import *framework* Flask

```

from flask import Flask
app = Flask(__name__)
if __name__ == '__main__':
    app.run(debug=True, port=80)
  
```

Kode program 2 merupakan pengaturan awal untuk aplikasi Flask dengan membuat objek Flask dan menjalankan aplikasi Flask dengan port 80.

Kode Program 3. Menampilkan Dokumen (Proses `select data`)

```

@app.route("/", methods=["GET", "POST"])
def index():
    if request.method != "POST":
        type = description = ""
        datas =
list(mongo.db.datas.find({'dataType' :
'file'}))
        category =
list(mongo.db.datas.find({'dataType' :
'category'}))
        return render_template("index.html",
datas=datas, category= category,
message=description, type=type)
  
```

Kode program 3 merupakan kode untuk mencari data pada MongoDB dengan perintah `mongo.db.datas.find({'dataType' : 'file'})` dengan parameter data yang dicari berupa *key* dan *value* lalu tipenya diubah menjadi *list* untuk ditampilkan ke halaman utama.

Kode Program 4. fungsi `insert_data_one` pada `globalvar.py` (Proses `insert data`)

```

def insert_data_one(self, collection, data,
title):
    type = keterangan = ""
    try:
        self.mongo.db[collection].insert_one(data)
    
```

```

    type , keterangan = "success", str(title)
+ " berhasil ditambahkan"
except:
    type , keterangan = "error", "Gagal
menambahkan data"
return type, keterangan

```

Kode Program 5. *fungsi insert_data_many* pada globalvar.py (Proses *insert data*)

```

def insert_data_many(self,collection, data,
title):
    type = keterangan = ""
    try:
self.mongo.db[collection].insert_many(data)
        type , keterangan = "success", str(title)
+ " berhasil ditambahkan"
    except:
        type , keterangan = "error", "Gagal
menambahkan data"
    return type, keterangan

```

Kode program 4 dan 5 merupakan implementasi dari fungsi untuk menambahkan data ke dalam database MongoDB. Terdapat dua fungsi yang berbeda, yaitu *insert_data_many* dengan perintah self.mongo.db[collection].**insert_many**(data) untuk menambahkan banyak data sekaligus dan *insert_data_one* dengan perintah self.mongo.db[collection].**insert_one**(data) untuk menambahkan satu data. Kedua fungsi akan mengembalikan tipe dan keterangan yang menunjukkan keberhasilan atau kegagalan penambahan data ke dalam database.

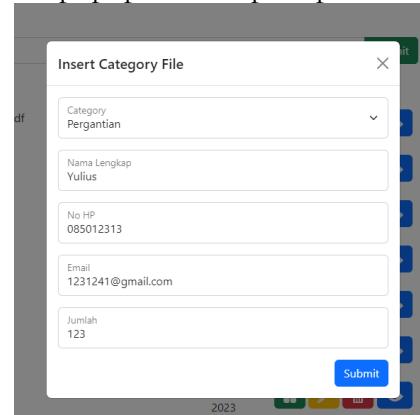
Implementasi UI

Name	Size	Last Modified	Action
1667317717_Yulius_CV_Academy2022.pdf	228.11 KB	01 November 2022	
1667805661_672019014.png	1.41 MB	07 November 2022	
1675086679_672019014_PKM-KC.docx	8.94 MB	30 January 2023	
1675086682_991004285.pdf	115.58 KB	30 January 2023	
1675086682_Drawing9.pdf	64.90 KB	30 January 2023	
1675086683_Essay.pdf	129.37 KB	30 January 2023	
1675086683_google_maps.pdf	1.35 MB	30 January 2023	
1675086684_Tugas_2.pdf	50.88 KB	30 January 2023	
1675086684_What_project_have_you_worked_about.docx	14.99 KB	30 January 2023	

Gambar 4. Tampilan Halaman Utama

Gambar 4 merupakan tampilan halaman utama setelah pengguna berhasil masuk. Dalam halaman ini terdapat daftar dokumen yang ada serta detail atribut dari masing-masing dokumen. Pengguna dapat menambahkan dokumen dengan memilih file dan menekan tombol *submit*. Pengguna juga dapat melakukan proses perubahan dan penambahan

atribut dokumen dengan cara menekan tombol berwarna hijau pada masing-masing dokumen dan akan muncul *popup menu* seperti pada Gambar 5.



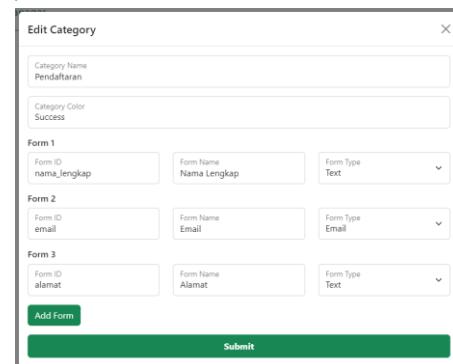
Gambar 5. Tampilan Popup Tambah Atribut File

Pada gambar 5 terdapat *Category* yang merupakan daftar nama kategori dokumen yang didapatkan dari *database*. Detail kategori dokumen terdapat pada halaman kategori dokumen seperti pada gambar 6.

Tambah Category Baru	Category Name	Category Color	List Form	Action
1 Perjalinan				
2 Pendaftaran				
3 Pemasangan				
4 Pergantian				

Gambar 6. Tampilan Halaman Kategori Dokumen

Pada halaman kategori dokumen, pengguna dapat menambah, mengubah dan menghapus kategori dokumen. Untuk melakukan proses perubahan, pengguna dapat menekan ikon pensil yang berada di kolom *action*. Dan akan muncul tampilan menu *update* kategori dokumen berupa *popup box* seperti pada gambar 7.



Gambar 7. Tampilan Menu *Update* Dokumen

Kelebihan dan Kekurangan MongoDB

Menurut Renaldi, et al (2020) mengatakan bahwa kelebihan basis data MongoDB dibanding dengan basis data lain yaitu MongoDB memiliki model data berbasis dokumen seperti file JSON yang disebut dengan BSON. Dengan model data berbasis dokumen, pengguna tidak perlu merancang struktur tabel seperti pada basis data SQL. Karena skema MongoDB lebih fleksibel, maka MongoDB memiliki penyimpanan data yang lebih besar dan *low-cost* [3]. Hal ini sesuai dengan konsep kategori dokumen yang ada di aplikasi manajemen dokumen milik PT.XYZ yang memiliki atribut yang beragam setiap kategori dokumennya.

Menurut Dipina, Shirin dan Surekha (2016) mengatakan keuntungan MongoDB adalah dapat dengan mudah melakukan perubahan skema tanpa

melalui proses downtime dan tanpa merubah aplikasi [18]. Sedangkan kekurangan MongoDB menurut Renaldi, et al (2022) adalah MongoDB belum banyak hosting yang mendukung. Selain itu, ketika pengguna berpindah dari basis data SQL ke NoSQL maupun NoSQL ke NoSQL lain, maka pengguna perlu mempelajari terlebih dahulu karena memiliki model data yang berbeda-beda [18].

Pengujian Sistem

Pengujian sistem yang dilakukan pada aplikasi manajemen dokumen yaitu *black box testing*. Pengujian ini dilakukan karena menguji kelancaran program tanpa menguji desain dan kode program yang digunakan [17]. Adapun hasil dari pengujian sistem ini sebagai berikut:

Tabel 1. Hasil Pengujian Sistem

Skenario Uji	Hasil yang diharapkan	Hasil Uji	Hasil
Melakukan proses <i>insert</i> dokumen (<i>insert</i>)	Sistem akan melakukan proses upload data ke dalam Google Cloud Storage dan memasukkan atribut data ke MongoDB lalu menampilkan pesan sukses	Sistem melakukan proses upload data dan memasukkan atribut data ke MongoDB menampilkan notifikasi	Valid
Memasukkan tipe dokumen yang tidak diijinkan (<i>insert</i>)	Sistem akan melakukan pengecekan tipe file lalu tipe file yang tidak diijinkan tidak akan di proses dan menampilkan pesan gagal	Sistem melakukan pengecekan tipe file dan file tidak di proses lalu menampilkan pesan notifikasi gagal	Valid
Menekan salah satu file pada daftar file yang memiliki kategori (<i>select</i>)	Sistem akan melakukan pengecekan kategori dokumen, lalu menampilkan kategori dokumen pada halaman utama.	Sistem melakukan pengecekan dan kemudian menampilkan kategori dokumen berserta dengan detail kategori pada halaman utama.	Valid
Manambahkan kategori ke dalam dokumen (<i>update</i>)	Sistem akan melakukan proses update data dokumen dengan menambahkan atribut kategori beserta detailnya kemudian menampilkan pesan notifikasi berhasil	Sistem melakukan proses update data dengan menambahkan kategori ke dalam dokumen lalu menampilkan pesan berhasil	Valid
Menekan tombol delete pada detail kategori dokumen (<i>delete</i>)	Menampilkan <i>popup</i> konfirmasi, dan menekan “Ya” lalu menghapus kategori tersebut dari	Menampilkan <i>popup</i> konfirmasi, dan menekan “Ya” lalu menghapus kategori tersebut dari data dokumen kemudian	Valid

	dokumen dan menampilkan pesan berhasil	menampilkan pesan notifikasi berhasil
Melakukan proses delete dokumen (<i>delete</i>)	Melakukan proses hapus file pada Google Cloud Storage dan MongoDB kemudian menampilkan pesan berhasil	Melakukan proses hapus file pada Google Cloud Storage dan MongoDB kemudian menampilkan pesan berhasil Valid

Berdasarkan hasil pengujian sistem yang dilakukan menggunakan *black box testing* pada tabel 1, dapat disimpulkan bahwa semua sistem aplikasi dapat berfungsi dengan baik. Hal ini dapat dilihat melalui hasil pengujian yang memenuhi semua hasil yang diharapkan.

4. Kesimpulan dan Saran

Berdasarkan penelitian dan pengujian yang dilakukan, dapat disimpulkan bahwa penerapan database MongoDB dalam aplikasi manajemen dokumen di PT. XYZ memberikan solusi yang efektif dan efisien dalam mengatasi masalah organisasi dan pengelolaan dokumen. MongoDB juga memiliki beberapa kelebihan yaitu memiliki model data berupa dokumen sehingga dapat dengan mudah merubah skema datanya. Perubahan skema juga tidak melalui proses *downtime*. MongoDB juga memiliki kekurangan yaitu masih belum banyak hosting web yang mendukung dan juga pengguna harus mempelajari basis data MongoDB karena memiliki model data yang berbeda. Hasil pengujian sistem menggunakan *black box testing* juga memenuhi semua hasil yang diharapkan. Hal ini menunjukkan bahwa MongoDB sangat cocok untuk digunakan dalam aplikasi manajemen dokumen dan dapat meningkatkan produktivitas dan efisiensi kerja di PT. XYZ.

Saran yang dapat diberikan penulis adalah untuk terus meningkatkan kualitas aplikasi manajemen dokumen dengan melakukan perbaikan dan pengembangan fitur-fitur yang diperlukan. Serta diperlukan analisa lebih lanjut mengenai keamanan dan pemantauan data pada aplikasi manajemen dokumen di PT.XYZ.

5. Daftar Pustaka

- [1] Amirillah, C.D.R., Andriyana, S. and Benrahman, B., 2020. Perancangan Aplikasi Document Management System Berbasis Web Universitas Nasional dengan Metode Waterfall. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 5(1), pp.45-52. DOI: <http://dx.doi.org/10.30998/string.v5i1.6353>.
- [2] Farozi, M.F.M., 2019. Desain Basis Data Non Relasional Nosql MongodB Pada Website Sistem Informasi Akademik. *SISKOMTI*, 1(1), pp.24-39.
- [3] Renaldi, R., Santoso, B.C. and Natasya, Y., 2020. Tinjauan Pustaka Sistematis terhadap Basis Data MongoDB. *Jurnal Inovasi Informatika*, 5(2), pp.132-142.
- [4] Handika, I.P.S. and Suryawan, I.G.T., 2017. Rancang Bangun Document Management System Untuk Mengelola Dokumen Standart Operational Procedure. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)* p-ISSN, 2355, p.7699.
- [5] Alkaff, M., Radam, I.F. and Sugiantoro, S., 2020. Rancang Bangun Sistem Identifikasi Arus Lalu Lintas pada Simpang Tiga Menggunakan Database NoSQL. *Jurnal Teknik Informatika Dan Sistem Informasi*, 6(2), pp. 194–205. DOI: <https://doi.org/10.28932/jutisi.v6i2.2567>.
- [6] Siregar, N.P., Wiharja, K.R.S. and Gozali, A.A., 2015. Analisis Dan Implementasi Basis Data Terdistribusi Horizontal Pada MongodB Untuk Clikkb Bkkbn Regional Jawa Barat. *eProceedings of Engineering*, 2(2), pp. 6672–6682.

- [7] Suryana, T., 2012. Pengelolaan dokumen sebagai sarana komunikasi internal unikom. *J. Komput. Dan Inform*, 1(1), pp.1-9.
- [8] Muhidin, S.A., Winata, H. and Santoso, B., 2016. Pengelolaan arsip digital. *JPBM (Jurnal Pendidikan Bisnis Dan Manajemen)*, 2(3), pp.178-183.
- [9] Novindri, G.F. and Saian, P.O.N., 2022. Implementasi Flask Pada Sistem Penentuan Minimal Order Untuk Tiap Item Barang di Distribution Center Pada PT XYZ Berbasis Website. *Jurnal Mnemonic*, 5(2), pp.81-85. DOI: <https://doi.org/10.36040/mnemonic.v5i2.4670>.
- [10] Clinton, R.M. and Sengkey, R., 2019. Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini-Komputer Raspberry Pi. *Jurnal Teknik Elektro dan Komputer*, 8(3), pp.181-192. DOI: <https://doi.org/10.35793/jtek.v8i3.26655>.
- [11] HWicaksono, H.A. and Setiyawati, N., 2022. Pembangunan Python Script Generator pada Pengembangan Aplikasi Berbasis Web. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, 5(1), pp.157-166. DOI: <https://doi.org/10.37792/jukanti.v5i1.472>.
- [12] Supriyanto, S. and Rohmat, R., 2023. Plant Monitoring Using a Web-View-Based Android Application as a Realization of the Implementation of the Smart Agriculture Concept. *Jurnal Rekayasa Elektrika*, 19(2), pp. 31–37.
- [13] Ferdiansyah, N., Rahayu, D.A. and Permala, R., 2019. Komparasi Kemampuan PostgresSQL, MariaDB Dan MongoDB Dalam Memproses Data AIS Satelit Lapan. *Prosiding SIPTEKGAN XXIII Tahun 2019*, pp.231-238.
- [14] Silalahi, M., 2018. Perbandingan performansi database mongodb dan mysql dalam aplikasi file multimedia berbasis web. *Computer Based Information System Journal*, 6(1), pp.63-63. DOI: <https://doi.org/10.33884/cbis.v6i1.574>.
- [15] Yannisto, Y. and Wibowo, T., 2020, November. PERANCANGAN DAN IMPLEMENTASI STORAGE BERBASIS CLOUD PADA PERUSAHAAN PT INDONESIA WEDA BAY INDUSTRIAL PARK. In *Conference on Business, Social Sciences and Innovation Technology*. 1(1), pp. 362-369.
- [16] Wahid, A.A., 2020. Analisis metode waterfall untuk pengembangan sistem informasi. *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, pp.1-5.
- [17] Ningrum, F.C., Suherman, D., Aryanti, S., Prasetya, H.A. and Saifudin, A., 2019. Pengujian black box pada aplikasi sistem seleksi sales terbaik menggunakan teknik equivalence partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), pp.125-130.
- [18] Dipina Damodaran, B., Salim, S. and Vargese, S.M., 2016. MongoDB vs MySQL: a comparative study of performance in supermarket management system. *International Journal of Computational Science and Information Technology (IJCSITY)*, 4(2), pp.31-38. DOI: <https://doi.org/10.5121/ijcsity.2016.4204>.