

Penerapan Konsep NoSQL Pada Perangkat Lunak Manajemen Properti

Jumaida¹, Alek Wijaya^{2*}

Universitas Bina Darma, Palembang, Indonesia

Email: ¹Jumaida33333@email.com, ^{2*}Alex_wj@binadarma.ac.id

Email Penulis Korespondensi: Alex_wj@binadarma.ac.id

Abstrak—Saat ini, teknologi sudah banyak digunakan untuk membantu dalam operasional bisnis. Berbagai bidang telah memanfaatkan, seperti di bidang properti. Pada umumnya pihak mengembang dalam mengelolah database biasanya menggunakan basis data sql yaitu query yang saling berelas. Tetapi seiring dengan kemajuan teknologi maka pengembang ingin menggunakan database NoSQL dalam mengelolah data untuk mengatasi terjadinya melonjaknya data karena pengelolaan database NoSQL sangat cocok untuk mengelola database dalam jumlah besar, dan memiliki beberapa model data. Database NoSQL dapat disimpan dengan menggunakan berbagai data yaitu ada 4 tipe data yaitu Graph-based, document-oriented, column-oriented dan key value. Hasil penelitian ini menunjukkan bahwa penerapan konsep NoSQL pada perangkat lunak manajemen properti dapat meningkatkan efisiensi dan kinerja pengelolaan data properti. Penggunaan basis data NoSQL memungkinkan fleksibilitas skema, skalabilitas horizontal, kinerja tinggi, dan skalabilitas global. Dengan demikian, perangkat lunak ini dapat memberikan kemudahan pengguna untuk lebih baik dan meningkatkan efisiensi operasional dalam manajemen properti.

Kata Kunci: Properti, NoSQL; Manajemen; Kinerja; Skalabilitas

Abstract—Currently, technology is widely used to assist in business operations. Various fields have utilized it, such as the property sector. Typically, developers managing databases use SQL databases, which involve related queries. However, with technological advancements, developers are interested in using NoSQL databases to manage data in order to cope with the increasing volume of data. NoSQL database management is well-suited for handling large databases and offers various data models, including graph-based, document-oriented, column-oriented, and key-value. The results of this research indicate that implementing NoSQL concepts in property management software can improve the efficiency and performance of property data management. The use of NoSQL databases allows for flexible schemas, horizontal scalability, high performance, and global scalability. Consequently, this software can provide users with convenience, enabling them to enhance operational efficiency in property management.

Keywords: Property; NoSQL; Management; Performance; Scalability

1. PENDAHULUAN

Dengan berkembangnya teknologi memberikan pengaruh besar bagi manajemen properti terutama dalam bidang pengelolaan data. Dalam mengembangkan properti developer membutuhkan penyimpanan yang besar untuk menyimpan data properti. Seiring bertambahnya data properti dan perubahan data yang mendadak membuat pihak developer kesulitan dalam mengelolah data Gambar, data Video, Data Pemasaran dan promosi properti yang sering bertambah dan tingkat hasil progress pembangunan yang harus di perbarui sesuai tingkat penyelesaian pembangunan. alasan penulis menggunakan basis data NoSQL karena NoSQL memiliki skalabilitas lebih tinggi dalam artian dapat berkembang sesuai dengan kebutuhan data yang ada [1].

Pengembang tidak menggunakan database SQL karena SQL mengharuskan pihak pengembang untuk menentukan skema dalam membangun struktur data sedangkan NoSQL tidak memiliki struktur skema (setidaknya skemanya dapat berubah-ubah) dan tidak terstruktur berelasi seperti MySQL. Selain itu database NoSQL dapat disimpan dalam berbagai cara baik itu graph-based, document-oriented, column-oriented, atau yang terorganisir dengan baik penyimpanan seperti Key Value. dalam Database NoSQL berkembang menyesuaikan kebutuhan data sehingga cocok diterapkan dalam pengelolaan database properti karena dalam pengelolaan database properti itu data yang di kelola dan disimpan itu sering berubah-ubah menyesuaikan tingkat kebutuhan pihak pengembang. penulis memilih tipe data dokumen oriented karena bisa menyimpan data dalam format dokumen [2].

Basis data NoSQL mengadopsi model relasional yang menggunakan tabel dengan skema tetap dan hubungan antar tabel untuk menyimpan dan mengelola data. Namun, ketika datanya semakin melonjak, terjadi masalah dalam hal skalabilitas dan kinerja, karena operasi yang kompleks dan pertumbuhan data yang cepat dapat menyebabkan keterbatasan performa. Perangkat lunak manajemen properti seringkali harus mengelola dan menyimpan jumlah data yang besar, termasuk informasi tentang properti, penyewa, kontrak, pembayaran, dan lain sebagainya. Dalam skenario ini, penggunaan basis data relasional tradisional mungkin tidak cukup efisien karena batasan-batasan dalam skalabilitas vertikal dan horizontal. Dengan memanfaatkan database NoSQL, sistem dapat dengan mudah ditingkatkan secara horizontal dengan menambahkan lebih banyak server dan memperoleh kinerja yang baik bahkan dengan volume data yang besar (Dianta et al., 2022).

Basis data relasional biasanya membutuhkan skema yang terstruktur sebelum data dapat disimpan. Namun, dalam manajemen properti, ada banyak variasi dalam jenis properti, informasi yang terkait, dan atribut yang relevan. Dengan NoSQL, skema fleksibel dapat diterapkan, yang memungkinkan penambahan, penghapusan, dan perubahan atribut pada properti tanpa mempengaruhi struktur data yang ada. Ini memberikan kebebasan yang lebih besar dalam mengelola informasi properti yang beragam [4].

Perangkat lunak manajemen properti adalah aplikasi atau sistem yang dirancang untuk membantu mengelola dan mengatur berbagai aspek yang terkait dengan properti. Perangkat lunak ini biasanya digunakan oleh agen properti, pengelola properti, atau pemilik properti untuk mengelola, mengawasi, dan mengoptimalkan operasi properti mereka. Beberapa fitur umum yang sering ada dalam perangkat lunak manajemen properti meliputi: Manajemen Penyewa, Pemasaran dan Pemajanan Properti, Manajemen Kontrak, Keuangan dan Pembayaran, serta Perawatan dan Perbaikan Properti [5].

Perangkat lunak manajemen properti dapat disesuaikan dengan kebutuhan dan ukuran properti yang dikelola, mulai dari skala kecil seperti apartemen atau rumah persewaan hingga skala besar seperti kompleks perumahan atau properti komersial. Tujuan utama perangkat lunak ini adalah untuk meningkatkan efisiensi, meningkatkan pengalaman penyewa, dan memudahkan tugas pengelolaan properti secara keseluruhan. Selain itu, perangkat lunak manajemen properti seringkali harus memberikan respons yang cepat kepada pengguna, baik dalam hal pencarian data, pembayaran, pengiriman notifikasi, dan lain sebagainya. Dalam kasus-kasus ini, NoSQL dapat memberikan performa yang tinggi karena arsitekturnya yang terdistribusi dan mendukung penyebaran data ke beberapa server. Dengan cara ini, permintaan dapat didistribusikan secara efisien dan waktu tanggap sistem dapat ditingkatkan [2].

Penerapan konsep NoSQL pada perangkat lunak manajemen properti juga memungkinkan pengolahan data yang kompleks. Beberapa basis data NoSQL, seperti graf atau dokumen, menyediakan model data yang cocok untuk mengelola informasi properti yang memiliki hubungan kompleks atau struktur yang tidak teratur. Misalnya, basis data graf dapat digunakan untuk menggambarkan hubungan antara properti, penyewa, dan kontrak sewa mereka [1].

Konsep NoSQL (Not Only SQL) merujuk pada pendekatan dalam pengelolaan data yang berbeda dengan sistem manajemen basis data relasional (RDBMS). Dalam konteks perangkat lunak manajemen properti, NoSQL dapat digunakan untuk mengelola data properti secara efisien dan fleksibel. Salah satu karakteristik utama NoSQL adalah kemampuannya dalam menangani model data yang fleksibel. Dalam manajemen properti, properti dapat memiliki atribut yang bervariasi, tergantung pada jenis properti (rumah, apartemen, tanah, dll.). NoSQL memungkinkan Anda untuk menyimpan data properti dengan struktur yang bervariasi tanpa memerlukan skema yang kaku seperti yang diperlukan dalam RDBMS [6].

Basis data dokumen adalah salah satu jenis NoSQL yang populer. Dalam konteks manajemen properti, setiap properti dapat direpresentasikan sebagai dokumen yang berisi semua informasi yang relevan seperti alamat, harga, deskripsi, foto, dan atribut properti lainnya. Basis data dokumen memungkinkan Anda untuk menyimpan, mengambil, dan mengubah dokumen-dokumen ini dengan cepat. NoSQL biasanya dirancang untuk mendukung skalabilitas horizontal yang mudah. Dalam konteks manajemen properti, ini berarti Anda dapat dengan mudah menambahkan lebih banyak server atau mesin untuk mengelola data properti yang berkembang dengan cepat. Ini penting karena industri properti seringkali melibatkan jumlah data yang besar dan terus berkembang [7].

NoSQL sering kali menawarkan kinerja yang lebih tinggi daripada RDBMS dalam hal waktu tanggap dan throughput. Dalam manajemen properti, performa yang tinggi penting untuk memastikan pengguna dapat mencari dan menampilkan properti dengan cepat. NoSQL dapat mengoptimalkan kinerja operasi seperti pencarian, penyimpanan, dan pengambilan data untuk mendukung pengalaman pengguna yang responsif. Dalam manajemen properti, mungkin ada kebutuhan untuk mendukung operasi di berbagai lokasi geografis. NoSQL dapat mendukung replikasi data di beberapa wilayah geografis, memungkinkan akses cepat ke data properti yang diperlukan di lokasi terdekat. Ini memungkinkan perusahaan manajemen properti untuk menyediakan layanan yang cepat dan responsif kepada pengguna di berbagai wilayah [8].

Penerapan konsep NoSQL dalam perangkat lunak manajemen properti dapat membantu meningkatkan fleksibilitas, skalabilitas, dan performa aplikasi, memungkinkan pengelolaan properti yang lebih efisien dan pengalaman pengguna yang lebih baik. Namun, sebelum mengadopsi solusi NoSQL, penting untuk mempertimbangkan kebutuhan bisnis dan karakteristik data properti yang akan dikelola untuk memastikan kesesuaian teknologi ini dengan kebutuhan spesifik pengguna [9].

Penggunaan konsep NoSQL dalam perangkat lunak manajemen properti dapat memberikan keuntungan dalam hal skalabilitas, fleksibilitas skema, performa tinggi, dan pengolahan data yang kompleks. Namun, penting untuk mempertimbangkan karakteristik dan persyaratan khusus dari proyek manajemen properti Anda sebelum memilih pendekatan basis data yang tepat. Penerapan konsep NoSQL dalam perangkat lunak manajemen properti dapat memanfaatkan skalabilitas, fleksibilitas skema, performa tinggi, dan pengolahan data yang kompleks untuk meningkatkan efisiensi, mengoptimalkan pengalaman pengguna, dan menyediakan wawasan yang lebih baik dalam pengelolaan properti secara keseluruhan [10].

Dengan demikian, penelitian ini bertujuan untuk eksplorasi penerapan konsep NoSQL dalam perangkat lunak manajemen properti dan mengevaluasi keuntungan serta kinerja yang dapat diberikan oleh pendekatan ini. Dengan demikian, diharapkan penggunaan basis data NoSQL dapat mengatasi kendala yang dihadapi dalam pengelolaan data properti dan memberikan hasil yang lebih baik dalam operasional dan pengalaman pengguna.

2. METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan metode penelitian kualitatif deskriptif. Penelitian kualitatif merupakan metode penelitian yang temuan-temuannya tidak diperoleh melalui prosedur statistik atau bentuk hitungan lainnya. Metode penelitian ini bertujuan untuk mengungkapkan gejala secara holistik-kontekstual melalui pengumpulan data alami dengan memanfaatkan diri peneliti sebagai instrumen kunci [11].

Metode penelitian ini akan memaparkan data-data yang sebenarnya terjadi dalam bentuk deskriptif. Penelitian deskriptif adalah metode penelitian yang menggambarkan fenomena-fenomena yang ada dan sedang berlangsung atau telah terjadi di masa lampau. Pada metode penelitian kualitatif, peneliti akan memfokuskan penelitian, memilih narasumber, mengumpulkan data, menafsirkan data, lalu membuat kesimpulan dari seluruh penelitian yang telah dilakukan [12].

3.2 Objek Penelitian

Tempat Penelitian dilakukan di PT. Berdikari Bangun Abadi proyek properti center point Jl. muara teladan pematang sungai simpit Kel. kayuara Kec. Sekayu Kab. Musi Banyuasin Provinsi Sumatera Selatan. Penelitian akan dilaksanakan selama empat bulan yakni dimulai dari bulan November 2022 sampai dengan bulan Februari 2023.

3.3 Teknik Pengumpulan Data

a. Observasi

Pada tahap ini, penulis melakukan pengamatan secara langsung atau observasi ke Instansi terkait, yaitu proyek properti center point, Kecamatan Sekayu, Kabupaten Musi Banyuasin, Provinsi Sumatera Selatan guna untuk mendapatkan data dan informasi yang dibutuhkan dalam perancangan penerapan konsep NoSQL pada perangkat lunak manajemen properti.

b. Wawancara

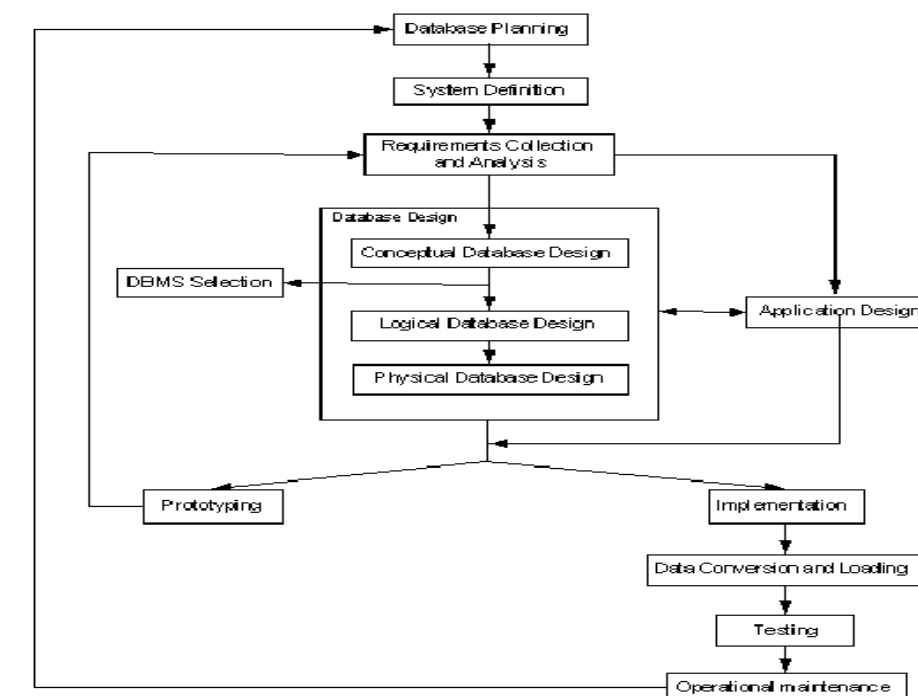
Untuk memperoleh informasi pelayanan dan data-data yang berkaitan dengan penelitian ini maka peneliti mengajukan pertanyaan dan diskusi dengan pihak perusahaan, terutama admin perusahaan serta jajarannya untuk mendapatkan informasi mengenai betuk dan gambaran data serta informasi untuk mempermudah penelitian.

c. Studi Pustaka

Peneliti melakukan pengumpulan data dengan cara mencari bahan referensi dari internet, jurnal, perpustakaan atau pun buku yang berkaitan dengan objek penelitian.

3.4 Metode Pengembangan Perangkat Lunak

Metode analisis dan perancangan yang dipakai menurut pendekatan Connolly dan Begg (2018) yang terdiri dari beberapa tahapan, yaitu [15]:



Gambar 1. Alur Perancangan Connolly dan Begg

Metode Srum memiliki beberapa tahapan yaitu:

a. *Database Planning*

Tahap perencanaan ini terkait dengan struktur *database* yang dirancang sesuai dengan kebutuhan sistem. Perancangan *database* juga mempertimbangkan relasi antar entitas, redundansi data, penentuan *primary key* dan *foreign key*. Selain itu juga, pada tahap ini dirancang fitur-fitur yang dibutuhkan seperti:

1. Mengelola dokumen *collection* (edit, tambah, hapus) pada Kategori
2. Mengelola dokumen *collection* (edit, tambah, hapus) pada Konsumen
3. Mengelola dokumen *collection* (edit, tambah, hapus) pada Gambar
4. Mengelola dokumen *collection* (edit, tambah, hapus) pada Post
5. Mengelola dokumen *collection* (edit, tambah, hapus) pada Siteplans
6. Mengelola dokumen *collection* (edit, tambah, hapus) pada Tentang
7. Mengelola dokumen *collection* (edit, tambah, hapus) pada User
8. Mengelola dokumen *collection* (edit, tambah, hapus) pada Video
9. Mengelola dokumen *collection* (edit, tambah, hapus) pada Visi Dan Misi

b. *System Definition*

Pada tahap ini, sistem yang dikembangkan didefinisikan secara jelas dan komprehensif. Batasan sistem dan identifikasi serta penjelasan rinci akan dilakukan. Selain itu, fungsi-fungsi yang ada pada sistem akan didefinisikan secara jelas. Secara ringkas, sistem dapat menampilkan informasi unit properti, informasi data konsumen, informasi gambar, menampilkan informasi pemasaran dan menampilkan tahap pembangunan properti.

c. *Requirements Collection and Analysis*

Di dalam tahap ini, dilakukan pengumpulan dan analisis kebutuhan terkait dengan sistem yang dikembangkan. Pada tahap ini akan diperdalam pemahaman terkait kebutuhan sistem yang diselaraskan dengan tujuan bisnis yang ingin dicapai. Selanjutnya, terdapat tiga bagian yang dibahas, yaitu: konsep, logika, dan fisik dari desain *database*.

Tujuan tahap ini memahami secara menyeluruh kebutuhan bisnis dan menghasilkan model data yang dapat mendukung pengorganisasian data.

Tabel 1. Dokumen *Collection Post*

No.	Field	Type	Keterangan
1.	<u>ID</u>	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	User_id	String	Identitas pengguna
3.	Category_id	String	Meyimpan post unik

Tabel 2. Dokumen *Collection Image*

No.	Field	Type	Keterangan
1.	<u>ID</u>	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	Judul	String	Judul Dari Gambar
3.	Caption	String	Menjelaskan Gambar
4.	Image	String	Gambar yang di simpan

Tabel 3. Dokumen *Collection Siteplan*

No.	Field	Type	Keterangan
1.	<u>ID</u>	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	Judul	String	Judul dari siteplan
3.	Desc	String	Deskripsi atau keterangan siteplan
4.	Type	String	Tipe rumah
5.	Harga	String	Harga dari unit rumah

Tabel 4. Dokumen *Collection Video*

No.	Field	Type	Keterangan
1.	<u>ID</u>	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	Judul	String	Judul dari Video
3.	Link	String	Alamat Link Video

Tabel 5. Dokumen *Collection Video*

No.	Field	Type	Keterangan
1.	<u>ID</u>	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	Judul	String	Judul dari Video
3.	Link	String	Alamat Link Video

Tabel 6. Dokumen *Collection* Kategori

No.	Field	Type	Keterangan
1.	_ID	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	Nama	String	Nama dari kategori yang akan dipilih ketika melakukan input

Tabel 7. Dokumen *Collection* Tentang

No.	Field	Type	Keterangan
1.	_ID	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	Judul	String	Judul dari tentang yang akan di upload
3.	Desc	String	Deskripsi atau keterangan tentang perusahaan
4.	Image	String	Gamabr upload untuk memperjelas tentang

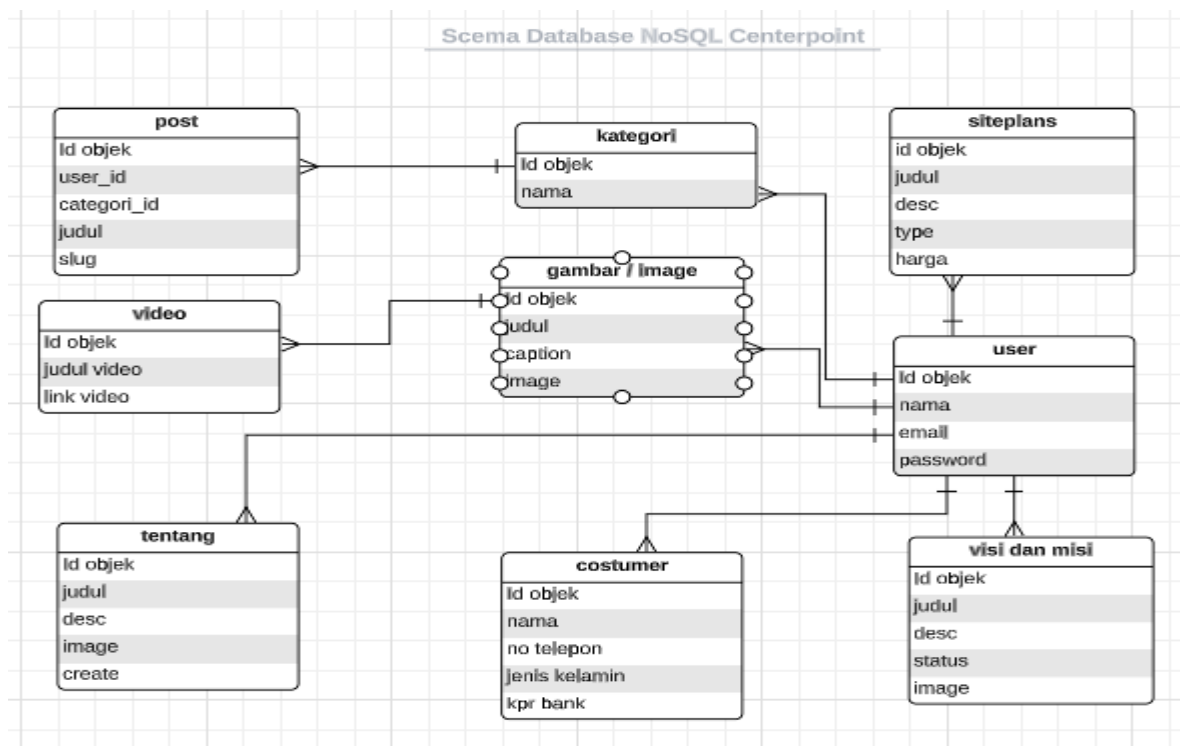
Tabel 8. Dokumen *Collection* User

No.	Field	Type	Keterangan
1.	_ID	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	Nama	String	Nama dari admin
3.	Email	String	Email dari admin untuk login
4.	Password	String	Kata sandi untuk login

Tabel 9. Dokumen *Collection* Visi Dan Misi

No.	Field	Type	Keterangan
1.	_ID	Object_id	Identifier Objek atau Primary Key dalam konteks basis data relasi
2.	Judul	String	Judul dari visi dan misi
3.	Desc	String	Deskripsi atau keterangan visi dan misi
4.	Status	String	Inisialisasi dari Visi dan Misi

Model data dirancang untuk memenuhi kebutuhan sistem. Secara lengkap atribut, dan hubungan antar tabel dapat dilihat. Tujuan dari perancangan database nosql ini yaitu untuk menyederhanakan pengembangan aplikasi dengan memungkinkan pengembang fokus pada logika bisnis, tanpa harus khawatir tentang pertimbangan yang rumit terkait struktur dan kinerja basis data.

**Gambar 1.** Skema Diagram NoSQL Center Point

d. *DBMS Selection (optional)*

Beberapa pertimbangan ditentukan pada tahap ini, seperti akses data, fitur dan fungsionalitas yang tinggi, dan ketersediaan dukungan terkait permasalahan yang mungkin terjadi. Dalam penelitian ini, digunakan MongoDB

sebagai *database*. MongoDB menggunakan model dokumen yang memungkinkan pengguna untuk menyimpan dan mengakses data dalam format dokumen JSON yang kaya dan bervariasi. Di dalam penelitian ini digunakan model data dokumen, dimana data disimpan dalam dokumen JSON yang fleksibel. MongoDB memiliki skema yang dinamis dan mendukung replikasi data dan partisi otomatis.

e. *Application Design*

Tujuan utama dari tahap ini yaitu menciptakan aplikasi yang efektif, efisien, dan mudah dioperasikan. Desain aplikasi juga mencakup penentuan struktur database yang sesuai, arsitektur aplikasi yang baik, serta alur kerja dan pengalaman pengguna yang optimal. Desain aplikasi juga mencakup penentuan struktur database yang sesuai, arsitektur aplikasi yang baik, serta alur kerja dan pengalaman pengguna yang optimal.

f. *Implementation*

Selama proses ini berjalan, pengujian yang komprehensif dilakukan untuk memverifikasi dan memvalidasi kualitas dan performa aplikasi. Pada tahap ini, dilakukan desain aplikasi melalui baris kode dan menerapkan serta mengkonfigurasi komponen yang diperlukan agar aplikasi dapat berfungsi dengan baik. Dokumentasi yang lengkap juga penting dalam proses ini untuk mendokumentasikan komponen, konfigurasi, dan petunjuk operasional yang diperlukan untuk memelihara dan mengelola aplikasi di masa depan.

g. *Data Conversion and Loading*

Pada tahap ini, data dari sumber yang berbeda seperti basis data yang ada, file teks, atau sistem eksternal lainnya, dikonversi ke format yang sesuai dengan struktur dan kebutuhan sistem baru. Proses ini dilakukan dengan teliti dan detail. Selain itu, memastikan bahwa data yang dikonversi dan dimuat ke dalam sistem baru akurat, konsisten, dan dapat diakses dengan benar oleh aplikasi.

h. *Testing*

Tujuan tahap ini untuk mengevaluasi kualitas, fungsionalitas, dan performa aplikasi perangkat lunak guna memastikan bahwa aplikasi tersebut memenuhi persyaratan yang telah ditentukan dan berfungsi sesuai yang diinginkan.

i. *Operational Maintenance*

Pada tahap ini melibatkan melakukan tugas rutin seperti memantau kesehatan sistem, mengelola sumber daya sistem, dan menyelesaikan masalah atau insiden yang timbul. Aspek penting lain dari pemeliharaan operasional adalah dukungan pengguna, yang melibatkan menangani pertanyaan pengguna, memberikan pelatihan, dan menyelesaikan masalah yang dilaporkan oleh pengguna.

3. HASIL DAN PEMBAHASAN

3.1 Penerapan Database NoSQL MongoDB

Penerapan konsep NoSQL pada perangkat lunak manajemen properti dapat memberikan sejumlah manfaat dan fleksibilitas. Berikut ini adalah beberapa contoh penerapan konsep NoSQL dalam perangkat lunak manajemen properti [13]:

- a. **Skema Fleksibel:** Basis data NoSQL, seperti MongoDB, tidak memerlukan skema yang telah ditentukan sebelumnya. Ini memungkinkan perangkat lunak manajemen properti untuk dengan mudah menyesuaikan dan menambahkan atribut atau entitas baru tanpa perlu mengubah skema atau struktur basis data. Misalnya, jika perlu menambahkan atribut khusus untuk properti, seperti fitur tambahan atau deskripsi khusus, hal ini dapat dilakukan tanpa mengganggu struktur yang ada.
- b. **Penyimpanan Dokumen:** Basis data NoSQL menggunakan model penyimpanan dokumen yang memungkinkan perangkat lunak manajemen properti menyimpan data properti sebagai dokumen terstruktur. Dokumen ini dapat berisi informasi tentang properti, seperti alamat, deskripsi, spesifikasi, gambar, fasilitas, dan atribut lainnya. Dengan menggunakan dokumen, informasi properti dapat disimpan dalam format yang lebih fleksibel dan mudah dikelola.
- c. **Skala Horizontal:** Basis data NoSQL dirancang untuk mendukung skala horizontal dengan mudah. Ini berarti perangkat lunak manajemen properti dapat dengan cepat dan mudah mengatasi pertumbuhan data dan beban kerja yang tinggi. Dengan menambahkan server tambahan ke kluster basis data, aplikasi dapat mengatasi volume data yang lebih besar dan meningkatkan kinerja secara horizontal.
- d. **Kueri Fleksibel:** Basis data NoSQL menyediakan fitur kueri yang fleksibel untuk mengambil dan memanipulasi data. Perangkat lunak manajemen properti dapat menggunakan fitur kueri NoSQL untuk menjalankan pertanyaan yang kompleks dan mendapatkan hasil yang diinginkan dengan efisien. Contohnya, perangkat lunak dapat melakukan pencarian berdasarkan kriteria tertentu, seperti wilayah, harga, tipe properti, atau atribut lain yang relevan.
- e. **Skala Vertikal:** NoSQL juga mendukung skala vertikal, yang memungkinkan perangkat lunak manajemen properti meningkatkan kinerja dengan meningkatkan sumber daya per server. Ini bisa dilakukan dengan menambahkan lebih banyak CPU, memori, atau penyimpanan ke server yang ada. Skala vertikal memungkinkan aplikasi untuk mengelola beban kerja yang lebih besar dengan lebih efisien.

Penerapan konsep NoSQL dalam perangkat lunak manajemen properti memberikan kelebihan dalam hal fleksibilitas, skalabilitas, dan efisiensi dalam mengelola dan memanipulasi data properti. Namun, perlu diingat bahwa pemilihan basis data NoSQL dan implementasi yang tepat harus didasarkan pada kebutuhan bisnis dan karakteristik spesifik dari perangkat lunak manajemen properti yang Anda kembangkan.

Penerapan konsep NoSQL pada perangkat lunak manajemen properti melibatkan beberapa langkah dan keputusan desain. Berikut adalah langkah-langkah umum yang dapat Anda ikuti untuk menerapkan konsep NoSQL pada perangkat lunak manajemen properti [14].

- a. Analisis Kebutuhan: Identifikasi kebutuhan bisnis dan fungsionalitas yang diperlukan dalam perangkat lunak manajemen properti Anda. Ini meliputi pemahaman tentang jenis data properti yang akan disimpan, hubungan antar entitas seperti pengguna, agen, atau transaksi, serta operasi yang harus dijalankan pada data tersebut.
- b. Pemilihan Basis Data NoSQL: Berdasarkan analisis kebutuhan, pilih basis data NoSQL yang sesuai untuk perangkat lunak manajemen properti Anda. Contoh basis data NoSQL yang umum digunakan adalah MongoDB, CouchDB, atau Cassandra. Pertimbangkan fitur-fitur yang disediakan oleh masing-masing basis data, seperti fleksibilitas skema, dukungan kueri, skalabilitas, dan performa.
- c. Desain Skema Data: Rancang skema data yang sesuai untuk menyimpan informasi properti dan entitas terkait dalam basis data NoSQL. Dalam MongoDB, Anda dapat menggunakan dokumen BSON (Binary JSON) untuk menyimpan informasi properti dalam bentuk struktur data yang terstruktur. Tentukan atribut apa yang perlu disimpan untuk setiap entitas dan bagaimana hubungan antar entitas akan direpresentasikan.
- d. Pemodelan Dokumen: Dalam basis data NoSQL, perlu memahami bagaimana dokumen akan dimodelkan dan disimpan. Identifikasi entitas utama, seperti properti, pengguna, atau agen, dan desain dokumen yang mewakili entitas tersebut. Pastikan atribut dan relasi antar entitas terdokumentasi dengan baik dalam dokumen.
- e. Implementasi Fungsionalitas: Implementasikan logika bisnis dan fungsionalitas perangkat lunak manajemen properti menggunakan API atau driver yang disediakan oleh basis data NoSQL yang Anda pilih. Ini melibatkan penulisan kode untuk membuat, membaca, memperbarui, dan menghapus dokumen dalam basis data NoSQL.
- f. Skalabilitas: Pastikan perangkat lunak manajemen properti dapat mengatasi pertumbuhan data dan beban kerja yang tinggi. Pertimbangkan penggunaan replikasi atau klusterisasi untuk mendistribusikan beban kerja ke beberapa server dan meningkatkan performa aplikasi.
- g. Pengoptimalan Kueri: Jika aplikasi perlu melakukan kueri kompleks, pertimbangkan teknik pengoptimalan kueri yang disediakan oleh basis data NoSQL. Indeks yang sesuai dan strategi pengaturan kueri dapat membantu meningkatkan kinerja saat mengambil data properti yang relevan.
- h. Pengujian dan Pemeliharaan: Lakukan pengujian menyeluruh pada perangkat lunak manajemen properti untuk memastikan bahwa fungsionalitas dan performa sesuai dengan harapan. Selain itu, perhatikan pemeliharaan rutin seperti pemantauan kinerja, pemulihan bencana, dan pengamanan data untuk menjaga integritas dan keamanan sistem.

Penerapan konsep NoSQL pada perangkat lunak manajemen properti dapat memberikan fleksibilitas, skalabilitas, dan efisiensi dalam mengelola data properti. Namun, perlu mempertimbangkan kebutuhan bisnis dan karakteristik spesifik dari perangkat lunak manajemen properti Anda untuk memilih basis data NoSQL yang paling sesuai dan merancang skema data yang tepat.

Setelah melakukan perancangan database NoSQL peneliti melakukan implementasi ke mongo db untuk pembuatan database yang diberi nama database center Point yang memiliki 9 *collection* atau di dalam basis data relational itu disebut tabel. Penerapan database NoSQL MongoDB dalam perangkat lunak manajemen properti dapat memberikan beberapa manfaat dan mempengaruhi beberapa aspek dalam pengelolaan properti. Berikut adalah beberapa poin yang berkaitan dengan penerapan MongoDB [11]:

- a. Skalabilitas: MongoDB dirancang untuk mendukung skalabilitas horizontal dengan mudah. Dalam manajemen properti, di mana data properti, penyewa, kontrak, dan pembayaran dapat berkembang dengan cepat, MongoDB memungkinkan penambahan server dan partisi data yang efisien untuk menjaga kinerja sistem yang optimal. Hal ini memungkinkan sistem untuk terus berjalan dengan baik bahkan saat data mengalami pertumbuhan yang signifikan.
- b. Model Data Fleksibel: MongoDB adalah database dokumen NoSQL yang menggunakan model data fleksibel. Dalam manajemen properti, di mana struktur data properti dapat bervariasi dan atribut tambahan dapat ditambahkan atau diubah seiring waktu, MongoDB memungkinkan penyimpanan data yang tidak terstruktur atau semi-terstruktur. Ini memberikan fleksibilitas dalam mengelola informasi properti yang beragam tanpa harus mempertahankan skema yang kaku.
- c. Performa Tinggi: MongoDB menawarkan performa yang tinggi dan waktu tanggap yang cepat. Dengan arsitektur terdistribusi dan dukungan untuk penyebaran data di beberapa server, MongoDB dapat mendistribusikan beban kerja secara efisien dan memberikan respons yang cepat dalam operasi yang melibatkan pencarian, penambahan, penghapusan, dan pembaruan data properti.
- d. Kemampuan Querying yang Kuat: MongoDB menyediakan kemampuan querying yang kuat dengan dukungan indeks yang efisien. Dalam konteks manajemen properti, ini memungkinkan penggunaan query yang kompleks dan fleksibel untuk mencari properti berdasarkan kriteria tertentu, seperti lokasi, tipe properti, harga, atau fasilitas yang tersedia. Dengan adanya indeks, kueri dapat dilakukan dengan cepat dan efisien.
- e. Replikasi dan Kepetakan: MongoDB memiliki fitur replikasi yang memungkinkan replikasi data di beberapa server untuk tujuan keamanan dan ketersediaan. Ini memastikan bahwa data properti dapat dipulihkan jika terjadi kegagalan server. Selain itu, MongoDB juga mendukung shard, yang memungkinkan data untuk didistribusikan di beberapa server, mengoptimalkan penggunaan sumber daya dan meningkatkan kinerja sistem.

Dengan penerapan MongoDB, perangkat lunak manajemen properti dapat memanfaatkan keuntungan yang disediakan oleh NoSQL, seperti skalabilitas, fleksibilitas skema, performa tinggi, dan kemampuan querying yang kuat. MongoDB menjadi pilihan yang populer dalam industri pengelolaan properti karena kemampuan yang dimilikinya dalam

mengatasi tuntutan data yang besar dan kompleks dalam konteks manajemen properti. Dalam penerapan database NoSQL pada manajemen properti ini terdapat pengelola database dan pengunjung website:

Tabel 1. Pengguna

No.	Pengguna	Peran
1	Pengunjung/user	Lihat informasi
2	Admin	Melakukan kelola database

Dalam tabel di atas, terdapat daftar pengguna dan peran masing-masing pengguna dalam mengelola dan menjalankan *Website* manajemen Properti.

3.2 Tampilan Aplikasi

Penggunaan konsep NoSQL dalam pengembangan perangkat lunak manajemen properti memberikan beberapa manfaat yang signifikan. Berikut adalah beberapa manfaat NoSQL dalam konteks tersebut:

- Fleksibilitas Skema:** Basis data NoSQL memberikan fleksibilitas skema yang memungkinkan Anda untuk menyimpan data dengan struktur yang berbeda tanpa memerlukan skema yang terdefinisi secara ketat. Ini memudahkan penyesuaian dan perubahan pada struktur data properti, entitas terkait, atau atribut tambahan tanpa mengganggu operasional aplikasi.
- Skalabilitas Horizontal:** Basis data NoSQL dirancang untuk dapat mengatasi pertumbuhan data secara horizontal dengan mudah. Dengan memanfaatkan replikasi dan klusterisasi, Anda dapat mendistribusikan beban kerja ke beberapa server, meningkatkan kapasitas dan kinerja aplikasi manajemen properti ketika jumlah data dan pengguna meningkat.
- Performa Tinggi:** NoSQL menawarkan performa tinggi dalam pengolahan data properti karena desain mereka yang dioptimalkan untuk kueri cepat dan akses data paralel. Struktur penyimpanan data yang berbeda seperti dokumen atau grafik memungkinkan kinerja yang lebih baik dalam pengambilan, pembaruan, dan penghapusan data properti yang kompleks.
- Skala Baca-Tulis yang Seimbang:** Basis data NoSQL biasanya dirancang untuk memiliki skala baca-tulis yang seimbang, yang berarti mereka mampu menangani beban kerja operasi baca dan tulis dengan baik. Hal ini penting dalam perangkat lunak manajemen properti di mana terdapat operasi pencarian dan pembaruan data yang sering terjadi.
- Penyimpanan Data yang Beragam:** Dalam pengembangan perangkat lunak manajemen properti, terdapat beragam jenis data yang perlu disimpan seperti informasi properti, foto, video, dokumen, dan lain sebagainya. Basis data NoSQL dapat menyimpan data dalam berbagai format, termasuk dokumen, grafik, atau kolom, sehingga memungkinkan penyimpanan data yang lebih fleksibel dan efisien.
- Skalabilitas Keandalan:** NoSQL seringkali memiliki fitur keandalan yang kuat, seperti replikasi otomatis dan toleransi kesalahan. Hal ini memastikan bahwa data properti Anda tetap tersedia dan aman dalam situasi kegagalan server atau infrastruktur.
- Keterjangkauan:** NoSQL umumnya memiliki biaya yang lebih rendah dalam hal infrastruktur dan lisensi jika dibandingkan dengan basis data relasional tradisional. Ini dapat mengurangi biaya pengembangan dan operasional perangkat lunak manajemen properti.

Dengan memanfaatkan manfaat-manfaat ini, penggunaan NoSQL dalam pengembangan perangkat lunak manajemen properti dapat meningkatkan fleksibilitas, skalabilitas, performa, dan efisiensi dalam pengelolaan data properti dan operasi aplikasi secara keseluruhan.

3.21 Halaman Login

Admin melakukan kelola database dengan cara login ke form login admin. Berikut halaman *login* yang ditampilkan pada Gambar 2. Halaman login adalah halaman yang digunakan oleh pengguna untuk mengotentikasi diri dan mendapatkan akses ke dalam sistem atau aplikasi yang membutuhkan otorisasi. User diminta untuk menginputkan alamat email dan password. Fitur lupa kata sandi merupakan tautan atau tombol "Lupa Kata Sandi" biasanya ditempatkan di sekitar formulir login. Ketika pengguna mengalami kesulitan mengingat kata sandi mereka, mereka dapat mengklik tautan ini untuk mengatur ulang kata sandi atau mendapatkan bantuan dalam mengakses akun mereka. Hal ini umumnya mengarah ke halaman pemulihan kata sandi.

Tujuan utama dari halaman login adalah untuk memastikan bahwa pengguna yang mengakses sistem atau aplikasi adalah pengguna yang sah dan terotentikasi. Hal ini membantu melindungi data dan informasi sensitif serta menjaga keamanan sistem dari akses yang tidak sah.

Gambar 2. Halaman Login

3.2.2 Menu Halaman Postingan

Menu publikasi memiliki 2 halaman untuk admin melakukan publikasi postingan dan menentukan kategori dan kelola edit, hapus, lihat, tambah yang akan disimpan pada database setelah dikelola. Menu "Halaman Postingan" dalam MongoDB meliputi:

- Daftar Postingan:** Menu ini dapat berisi opsi untuk melihat daftar postingan yang ada dalam sistem. Ketika pengguna memilih opsi ini, aplikasi dapat mengambil data postingan dari MongoDB dan menampilkannya dalam bentuk daftar. Daftar ini dapat mencakup judul postingan, tanggal publikasi, penulis, atau atribut lain yang relevan.
- Tambah Postingan:** Menu ini memungkinkan pengguna untuk menambahkan postingan baru ke dalam sistem. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna memasukkan informasi tentang postingan, seperti judul, konten, kategori, atau atribut lain yang relevan. Setelah pengguna mengisi formulir dan mengirimkan data, aplikasi dapat menyimpan postingan baru ke dalam MongoDB.
- Edit Postingan:** Opsi ini memungkinkan pengguna untuk mengedit postingan yang sudah ada. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar postingan yang ada dan memungkinkan pengguna memilih postingan tertentu untuk diedit. Setelah pengguna memilih postingan, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna mengubah informasi postingan yang dipilih. Setelah pengguna mengirimkan perubahan, aplikasi dapat memperbarui data postingan yang ada di MongoDB.
- Hapus Postingan:** Menu ini memungkinkan pengguna untuk menghapus postingan yang sudah ada. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar postingan yang ada dan memungkinkan pengguna memilih postingan tertentu untuk dihapus. Setelah pengguna mengonfirmasi penghapusan, aplikasi dapat menghapus data postingan terkait dari MongoDB.

Selain itu, menu "Halaman Postingan" juga dapat mencakup opsi seperti pencarian postingan berdasarkan kriteria tertentu, pengurutan postingan, atau fitur komentar di bawah setiap postingan.

3.2.3 Menu kelola Siteplan

Pada menu Siteplan admin kelola database edit, tambah, hapus. Menu Tambah Siteplan merupakan menu yang memungkinkan pengguna untuk menambahkan siteplan baru ke dalam sistem. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna memasukkan informasi tentang siteplan, seperti nama, lokasi, deskripsi, atau atribut lain yang relevan. Setelah pengguna mengisi formulir dan mengirimkan data, aplikasi dapat menyimpan siteplan baru ke dalam MongoDB.

Menu Edit Siteplan: Opsi ini memungkinkan pengguna untuk mengedit siteplan yang sudah ada. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar siteplan yang ada dan memungkinkan pengguna memilih siteplan tertentu untuk diedit. Setelah pengguna memilih siteplan, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna mengubah informasi siteplan yang dipilih. Setelah pengguna mengirimkan perubahan, aplikasi dapat memperbarui data siteplan yang ada di MongoDB.

Menu Hapus Siteplan: Menu ini memungkinkan pengguna untuk menghapus siteplan yang sudah ada. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar siteplan yang ada dan memungkinkan pengguna memilih siteplan tertentu untuk dihapus. Setelah pengguna mengonfirmasi penghapusan, aplikasi dapat menghapus data siteplan terkait dari MongoDB.

3.2.4 Menu kelola Galeri

Pada menu Galeri terdapat 2 halaman yaitu halaman gambar dan video admin kelola database edit, tambah, hapus disimpan pada database. Pada menu Tambah Gambar memungkinkan pengguna untuk menambahkan gambar baru ke dalam galeri. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna mengunggah gambar dan memberikan informasi tambahan seperti judul, deskripsi, atau atribut lain yang relevan. Setelah pengguna mengunggah gambar dan mengirimkan data, aplikasi dapat menyimpan gambar baru ke dalam MongoDB.

Menu Edit Gambar menjadi opsi yang memungkinkan pengguna untuk mengedit informasi gambar yang sudah ada dalam galeri. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar gambar yang ada dan memungkinkan pengguna memilih gambar tertentu untuk diedit. Setelah pengguna memilih gambar, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna mengubah informasi gambar seperti judul, deskripsi,

atau atribut lain yang relevan. Setelah pengguna mengirimkan perubahan, aplikasi dapat memperbarui data gambar yang ada di MongoDB.

Menu Hapus Gambar merupakan menu yang memungkinkan pengguna untuk menghapus gambar yang sudah ada dalam galeri. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar gambar yang ada dan memungkinkan pengguna memilih gambar tertentu untuk dihapus. Setelah pengguna mengonfirmasi penghapusan, aplikasi dapat menghapus data gambar terkait dari MongoDB. Selain itu, menu "Kelola Galeri" juga dapat mencakup opsi seperti pengaturan tampilan galeri, pengelompokan gambar berdasarkan kategori, atau fitur pencarian gambar.

3.2.5 Menu kelola Daftar Konsumen

Pada menu daftar konsumen admin kelola edit, hapus, tambah. Menu Daftar Konsumen berisi opsi untuk melihat daftar konsumen yang terdaftar dalam sistem. Ketika pengguna memilih opsi ini, aplikasi dapat mengambil data konsumen dari MongoDB dan menampilkannya dalam bentuk daftar. Daftar ini dapat mencakup informasi seperti nama, alamat, nomor telepon, atau atribut lain yang relevan.

Menu Tambah Konsumen merupakan menu yang memungkinkan pengguna untuk menambahkan konsumen baru ke dalam sistem. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna memasukkan informasi tentang konsumen, seperti nama, alamat, nomor telepon, atau atribut lain yang relevan. Setelah pengguna mengisi formulir dan mengirimkan data, aplikasi dapat menyimpan data konsumen baru ke dalam MongoDB.

Menu Edit Konsumen menjadi suatu opsi yang memungkinkan pengguna untuk mengedit informasi konsumen yang sudah terdaftar. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar konsumen yang ada dan memungkinkan pengguna memilih konsumen tertentu untuk diedit. Setelah pengguna memilih konsumen, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna mengubah informasi konsumen seperti nama, alamat, nomor telepon, atau atribut lain yang relevan. Setelah pengguna mengirimkan perubahan, aplikasi dapat memperbarui data konsumen yang ada di MongoDB.

Menu Hapus Konsumen: merupakan menu yang memungkinkan pengguna untuk menghapus konsumen yang sudah terdaftar dalam sistem. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar konsumen yang ada dan memungkinkan pengguna memilih konsumen tertentu untuk dihapus. Setelah pengguna mengonfirmasi penghapusan, aplikasi dapat menghapus data konsumen terkait dari MongoDB.

3.2.6 Menu kelola akun

Pada menu akun admin kelola edit, hapus, tambah. Menu Tambah Akun merupakan menu yang memungkinkan pengguna untuk membuat akun baru dalam sistem. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna memasukkan informasi untuk membuat akun baru, seperti nama pengguna, email, kata sandi, atau atribut lain yang relevan. Setelah pengguna mengisi formulir dan mengirimkan data, aplikasi dapat menyimpan data akun baru ke dalam MongoDB.

Menu Edit Akun merupakan opsi yang memungkinkan pengguna untuk mengedit informasi akun yang sudah terdaftar. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar akun yang ada dan memungkinkan pengguna memilih akun tertentu untuk diedit. Setelah pengguna memilih akun, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna mengubah informasi akun seperti nama pengguna, email, kata sandi, atau atribut lain yang relevan. Setelah pengguna mengirimkan perubahan, aplikasi dapat memperbarui data akun yang ada di MongoDB.

Menu Hapus Akun merupakan menu yang memungkinkan pengguna untuk menghapus akun yang sudah terdaftar dalam sistem. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan daftar akun yang ada dan memungkinkan pengguna memilih akun tertentu untuk dihapus. Setelah pengguna mengonfirmasi penghapusan, aplikasi dapat menghapus data akun terkait dari MongoDB.

3.3 Tampilan antar muka User / Pengunjung

3.3.1 Halaman lihat beranda

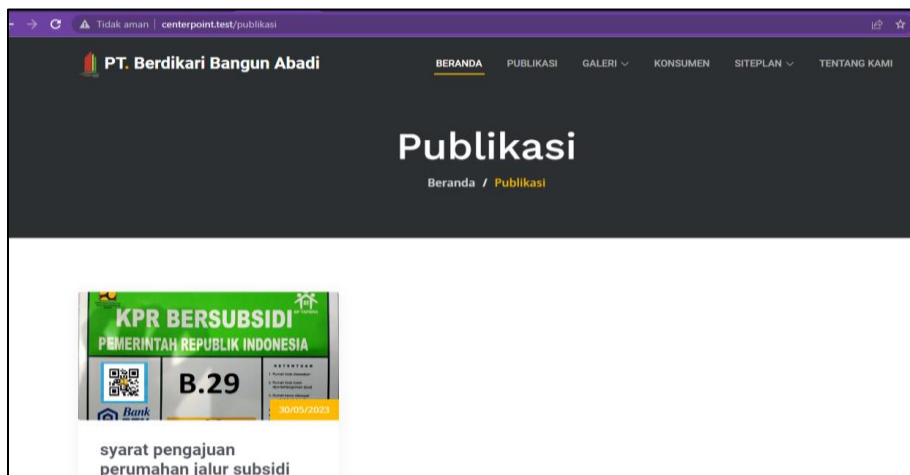
Halaman beranda adalah halaman utama dari aplikasi Anda yang menggunakan MongoDB sebagai basis data. Hal ini dapat dirancang untuk memberikan pengguna informasi umum, tampilan ringkasan, atau navigasi ke fitur-fitur utama dalam aplikasi. Pada halaman beranda pengunjung dapat melihat menu yang ditampilkan pada halaman website manajemen Properti yang terdiri dari beranda, publikasi, galeri, konsumen, siteplan, dan tentang kami. Berikut halaman beranda pengunjung yang ditampilkan pada Gambar 3.



Gambar 3. Halaman Beranda

3.3.2 Halaman lihat publikasi

Pada halaman lihat publikasi akan menampilkan informasi publikasi yang di kelola oleh admin. Berikut halaman publikasi pengunjung yang ditampilkan pada Gambar 4.



Gambar 4. Halaman publikasi

Halaman publikasi adalah halaman di aplikasi Anda yang digunakan untuk mengelola publikasi atau konten yang disimpan dalam basis data MongoDB. Hal ini dapat melibatkan fitur-fitur berikut:

- Daftar Publikasi Pengguna: Menu ini dapat berisi opsi untuk melihat daftar publikasi yang dibuat oleh pengguna tertentu. Ketika pengguna memilih opsi ini, aplikasi dapat mengambil data publikasi yang sesuai dari MongoDB dan menampilkannya dalam bentuk daftar. Daftar ini dapat mencakup judul publikasi, tanggal publikasi, atau atribut lain yang relevan. Pengguna dapat mengklik pada publikasi untuk melihat rincian lebih lanjut.
- Tampilan Publikasi: Opsi ini memungkinkan pengguna untuk melihat rincian lengkap dari publikasi tertentu yang mereka pilih. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan halaman dengan informasi seperti judul publikasi, konten, tanggal publikasi, atau atribut lain yang relevan. Pengguna dapat membaca publikasi secara lengkap dan melihat semua detail yang terkait.
- Edit Publikasi: Jika pengguna yang memiliki hak akses yang sesuai, opsi ini memungkinkan pengguna untuk mengedit publikasi yang mereka buat. Ketika pengguna memilih opsi ini, aplikasi dapat menampilkan formulir atau halaman yang memungkinkan pengguna mengubah informasi publikasi seperti judul, konten, tanggal publikasi, atau atribut lain yang relevan. Setelah pengguna mengirimkan perubahan, aplikasi dapat memperbarui data publikasi yang ada di MongoDB.
- Hapus Publikasi: Jika pengguna yang memiliki hak akses yang sesuai, menu ini memungkinkan pengguna untuk menghapus publikasi yang mereka buat. Ketika pengguna memilih opsi ini, aplikasi dapat mengonfirmasi penghapusan dan jika dikonfirmasi, menghapus data publikasi terkait dari MongoDB.

4. KESIMPULAN

Penerapan konsep NoSQL pada perangkat lunak manajemen properti. Peneliti menyimpulkan. Untuk mengadopsi NoSQL, perlu melakukan pemodelan data yang baik untuk mengoptimalkan kinerja dan kegunaan sistem manajemen properti agar dapat dilakukan Pemilihan jenis database NoSQL harus Berdasarkan pada kebutuhan spesifik sistem dan karakteristik data yang dapat diolah. Dalam penelitian ini Memerlukan evaluasi dan pemantauan secara berkala terhadap performa sistem manajemen properti yang menggunakan

konsep NoSQL untuk memastikan kinerja sistem dapat berjalan sesuai kebutuhan pengguna. Berdasarkan hasil implementasi MongoDB dapat diimplementasikan terhadap sistem manajemen properti dengan penyimpanan tipe struktur data dokumen, *Design* basis data dengan menggunakan pemodelan NoSQL MongoDB bersifat dinamis, pemodelan data dapat langsung disesuaikan berdasarkan kebutuhan. Berdasarkan hasil penelitian perbandingan NoSQL dan MySQL salah satunya terletak pada struktur data dan penyimpanan. data Penelitian ini dapat menghasilkan sebuah sistem yang layak digunakan untuk mengelola data seluruh manajemen properti. Database NoSQL dirancang dengan tujuan khusus untuk model data spesifik dan memiliki skema fleksibel untuk membuat aplikasi modern. Database NoSQL banyak dikenal secara luas karena memudahkan pengembang dalam faktor fungsionalitas, dan kinerja dalam berbagai skala.

REFERENCES

- [1] L. R. Maghfiroh and I. Santoso, "Nosql Model Data Warehouse Metadata Survei Dinamis Studi Kasus : Survei Rumah Tangga," pp. 55–74, 2019.
- [2] M. R. Alifi, T. Semiawan, D. C. U. Lieharyani, and H. Hayati, "Pemodelan Data Relasional pada NoSQL Berorientasi Dokumen," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 11, no. 3, pp. 183–191, 2022.
- [3] I. A. Diantaa, A. A. Aqhamb, and D. Setiawan, "Penerapan big data untuk mengatur sistem analisis data," *J. Ilm. Tek. Mesin, Elektro Dan Komput.*, vol. 2, no. 1, pp. 2809–0799, 2022.
- [4] Y. Tunardi and R. Layona, "Nosql Technology In Android Based Mobile Chat Application Using Mongodb," *ComTech Comput. Math. Eng. Appl.*, vol. 5, no. 2, p. 553, 2018, doi: 10.21512/comtech.v5i2.2180.
- [5] T. Oktavia, "Perancangan Model Basis Data Sistem Operasional Berbasiskan Customer Relationship Management," *ComTech Comput. Math. Eng. Appl.*, vol. 4, no. 2, p. 1179, 2023, doi: 10.21512/comtech.v4i2.2590.
- [6] R. Gunawan, "Pengukuran Query Respon Time pada NoSQL Database Berbasis Document Stored," *J. Siliwangi*, vol. 4, no. 2, pp. 100–103, 2018.
- [7] W. N. Suliyan, "Studi Literatur Basis Data SQL dan NoSQL," *Kilat*, vol. 8, no. 1, pp. 48–51, 2019, doi: 10.33322/kilat.v8i1.460.
- [8] A. Fadli, M. I. Zulfa, A. W. Widhi Nugraha, A. Taryana, and M. S. Aliim, "Analisis Perbandingan Unjuk Kerja Database SQL dan Database NoSQL Untuk Mendukung Era Big Data," *J. Nas. Tek. Elektro*, vol. 9, no. 3, 2020, doi: 10.25077/jnte.v9n3.774.2020.
- [9] D. Kriestanto and A. B. Arnado, "Implementasi Website Pencarian Kos dengan NoSql," *JIKO (Jurnal Inform. dan Komputer)*, vol. 2, no. 2, p. 103, 2017, doi: 10.26798/jiko.2017.v2i2.66.
- [10] H. Maryanto and N. Santoso, "Pengembangan Aplikasi Manajemen Logistic dan Purchasing pada Perusahaan Jasa Konstruksi berbasis Mobile (Studi Kasus: AMF-HAQ Engineering And Consultant)," vol. 6, no. 3, pp. 1121–1128, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>.
- [11] G. A. Bracchi and F. P. Paolini, "A relational data base management system," *Proc. ACM Annu. Conf. ACM 1972*, pp. 1080–1089, 2019, doi: 10.1145/800194.805904.
- [12] D. Kurniawan and Kurniati, "Penerapan Mongodb Pada Sistem Informasi Management Academic SDN 13 Banyuasin," *Semin. Has. Penelit. Vokasi*, vol. 3, no. 2, pp. 81–88, 2022.
- [13] T. L. Sinaga, N. Charibaldi, and N. H. Cahyana, "Perbandingan Waktu Respon Aplikasi Database NoSQL Elasticsearch dan MongoDB pada Pengujian Operasi CRUD," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 8, no. 1, pp. 22–35, 2023, doi: 10.14421/jiska.2023.8.1.22-35.
- [14] J. T. Santoso, "Komputasi Awan (Cloud Computing)," *Computing*, pp. 212–217, 2016.
- [15] Connolly T, Begg C. (2018). Database System: A Practical Approach to Design, Implementation, and Management. Fourth Edition, Addison Wesley, England.