JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 8
No. 3
Desember
2022

Basis Data Awan Non-Relasional Firestore untuk Penyimpanan Data Pesan

Kristian Adi Nugraha[#1]

[#]*Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana*
Jl. Dr. Wahidin Sudirohusodo 5-25 Yogyakarta 55224
[1]adinugraha@ti.ukdw.ac.id

Abstrak— **Pandemi Covid-19 mengubah cara komunikasi masyarakat di berbagai bidang yang semula melalui tatap muka secara langsung, berubah menggunakan media daring. Meningkatnya kebutuhan akan proses komunikasi yang dilakukan secara daring harus didukung dengan infrastruktur yang tepat agar proses pengiriman informasi dapat dilakukan lancar dan tanpa adanya hambatan. Salah satu faktor pendukung infrastruktur komunikasi adalah basis data yang digunakan untuk menyimpan data pesan. Salah satu jenis basis data yang banyak digunakan saat ini adalah basis data non-relasional, karena lebih fleksibel dan dinamis dibandingkan dengan basis data relasional. Terdapat banyak jenis basis data non-relasional, salah satunya berbasis dokumen. Penelitian ini mencoba mengimplementasikan basis data non-relasional Firestore untuk menyimpan berbagai jenis bentuk data pesan. Hasil pengujian menunjukkan bahwa basis data Firestore cukup andal digunakan untuk menyimpan data pesan teks dengan rata-rata waktu penyimpanan adalah 0.4 detik untuk jumlah kata maksimal 100 kata. Sedangkan untuk pesan berbentuk media seperti gambar, rata-rata waktu penyimpanan adalah 4 detik untuk ukuran media maksimal 5 MB, di mana angka tersebut masih dalam batas toleransi respons (10 detik) namun penggunaannya harus disertai dengan adanya *feedback* pada antarmuka.**

Kata kunci— **NoSQL, pesan, basis data, non-relasional, dokumen**

I. PENDAHULUAN

Seiring dengan berkembangnya teknologi internet di kalangan masyarakat, kebutuhan akan segala jenis informasi saat ini harus dapat dilakukan melalui media daring, terlebih setelah adanya pandemi Covid-19. Berbagai macam bentuk media informasi mulai beralih secara bertahap dari wujud fisik atau konvensional menjadi berwujud daring seperti media cetak (koran, majalah), poster pengumuman, serta segala bentuk media komunikasi berbasis teks. Keunggulan utama dari penggunaan media daring sebagai sarana komunikasi adalah kecepatan dalam menyampaikan informasi yang dapat dilakukan dalam waktu singkat dengan biaya yang cukup terjangkau, sehingga keterbatasan jarak dan biaya tidak lagi menjadi kendala dalam melakukan pertukaran informasi [1].

Saat ini, hampir seluruh aktivitas komunikasi dilakukan melalui media daring, sebagian besar aktivitas dilakukan dengan menggunakan aplikasi pada *smartphone* dalam bentuk pesan singkat (*chat*), media sosial atau *e-mail*. Penggunaan media daring sebagai sarana komunikasi dua buah perangkat dapat dilakukan secara langsung tanpa perantara (*peer-to-peer*) [2]. Namun cara ini memiliki kekurangan apabila perangkat penerima informasi sedang tidak terhubung ke jaringan atau jaringan yang terhubung sedang bermasalah, akan mengakibatkan informasi tidak berhasil sampai ke tujuan. Solusi untuk menangani masalah tersebut adalah dengan menggunakan perangkat *server* sebagai perantara dengan tujuan untuk menampung sementara informasi yang dikirimkan oleh perangkat pengirim, kemudian diteruskan kepada perangkat penerima [3]. Apabila perangkat penerima sedang tidak terhubung ke jaringan internet atau memiliki masalah konektivitas, maka informasi tersebut akan tetap berada di dalam *server* dan akan dikirimkan ulang saat perangkat penerima telah terhubung ke jaringan internet. Seluruh aktivitas pertukaran informasi akan terekam oleh *server*, sehingga apabila informasi tersebut hilang dari perangkat pengirim atau penerima informasi, maka seluruh data dapat diunduh ulang dari *server*.

Penggunaan *server* sebagai perantara komunikasi antar perangkat secara daring tidak dapat terlepas dari komponen basis data sebagai media penyimpanan data. Basis data memiliki peran yang cukup vital karena seluruh data komunikasi yang dikirim dan diterima antar perangkat disimpan secara permanen di dalam basis data. Firestore merupakan salah satu basis data non-relasional terbaru dari layanan komputasi berbasis awan (*cloud computing*) Firebase, di mana Firebase sendiri merupakan sebuah *platform* layanan *cloud service* milik Google yang menyediakan berbagai macam layanan dan fitur-fitur terkait dengan komunikasi yang dilakukan secara daring [4]. Firestore dapat digunakan untuk menyimpan data dalam format non-relasional (NoSQL), yaitu mekanisme penyimpanan data tanpa menggunakan tabel dan/atau relasi [5]. Beberapa bentuk atau skema dari NoSQL antara lain adalah bentuk tree [6], dokumen [7], atau vector spasial [8]. Keuntungan utama dari format NoSQL dibandingkan dengan SQL adalah seluruh data yang disimpan tidak harus

memiliki atribut yang sama, sehingga dapat secara fleksibel menyesuaikan variasi data yang akan disimpan [9]. Berbeda dengan format SQL yang mewajibkan setiap data harus memiliki *field-field* sesuai dengan yang telah didefinisikan pada tabel, sehingga akan menyulitkan apabila data yang disimpan memiliki berbagai macam variasi karena seluruh data dianggap bersifat homogen.

Berdasarkan latar belakang tersebut, tercipta gagasan untuk melakukan pengujian performa kecepatan penyimpanan dari basis data Firestore yang digunakan untuk menyimpan data-data bertipe pesan. Firestore telah banyak digunakan untuk menyimpan berbagai macam data, namun belum terdapat penelitian yang membahas mengenai penyimpanan data kompleks yang memiliki banyak atribut dengan tetap memperhatikan kecepatan menggunakan Firestore. Pesan-pesan tersebut nantinya akan diakses melalui aplikasi berbasis Android dengan menggunakan arsitektur berorientasi layanan (*service-oriented architecture*) untuk berbagai macam keperluan, seperti komunikasi personal antara dua pihak atau lebih, komunikasi satu arah (*broadcast*), atau komunikasi bentuk lainnya yang memiliki data berbentuk pesan di dalamnya. Luaran dari penelitian ini diharapkan dapat digunakan sebagai panduan mengenai apa saja yang perlu diperhatikan dalam membangun sebuah aplikasi yang memanfaatkan basis data non-relasional berbasis dokumen sebagai basis data penyimpanannya.

II. TINJAUAN PUSTAKA

Arsitektur berbasis layanan (*Service-oriented architecture*) merupakan salah satu jenis arsitektur perangkat lunak dengan skema *client-server*, di mana *server* bertugas untuk mengelola seluruh informasi yang ada, sedangkan *client* bertugas untuk menangani segala aktivitas yang berhubungan dengan pengguna. *Server* pada umumnya memiliki wujud dalam bentuk komputer yang terkoneksi pada jaringan, sedangkan untuk *client* dapat memiliki banyak bentuk. Salah satunya adalah *client* dalam bentuk kamera *CCTV* (*closed-circuit television*) yang bertugas untuk mengirimkan foto area parkir kepada *server* secara berkala, kemudian foto-foto tersebut akan diolah oleh *server* untuk menentukan ketersediaan lahan parkir [10, 11]. Selain itu, *client* juga dapat berbentuk komputer mikro (mikrokontroler) untuk implementasi teknologi *Internet of Things* (*IoT*) [12], salah satunya untuk keperluan penghitungan jenis kendaraan yang memasuki sebuah area secara otomatis [13]. Pada penelitian ini, *client* memiliki wujud berupa sebuah aplikasi berbasis perangkat *mobile*, khususnya untuk sistem operasi Android. Proses komunikasi oleh *client* terhadap *server* dilakukan dengan menggunakan perantara API (*Application Programming Interface*) yang terdapat pada *server*, yaitu sebuah antarmuka yang memungkinkan terjadinya komunikasi antara dua komputer (dalam hal ini *client* dan *server*) secara langsung [14].

Saat ini *service-oriented architecture* dapat dibangun dengan menggunakan layanan komputasi awan (*cloud computing*). Layanan *cloud computing* memudahkan para pengembang perangkat lunak untuk membangun sebuah aplikasi tanpa harus memikirkan sisi *server*, karena semuanya telah ditangani oleh *cloud computing* [15, 16, 17]. *Cloud computing* menawarkan fleksibilitas dalam penggunaan sumber daya *server*, karena pengguna layanan *cloud computing* pada umumnya hanya perlu membayar biaya berdasarkan banyak sedikitnya sumber daya yang digunakan dan dapat diubah sewaktu-waktu [18, 19, 20]. Salah satu layanan *cloud computing* yang tersedia adalah Firebase, di dalamnya terdapat beragam jenis fitur yang dapat digunakan, di antaranya dapat digunakan untuk melakukan penyimpanan basis data seperti Firestore [21]. Firestore merupakan layanan basis data non-relasional (NoSQL) berbasis dokumen yang terdapat pada Firebase. Basis data Firestore terdiri dari dua jenis komponen yaitu *collection* dan *document*. *Document* merupakan entitas dari setiap data, sedangkan *collection* merupakan kumpulan dari *document-document* yang sejenis. Pada basis data relasional, *collection* memiliki peran seperti *table*, sedangkan *document* merupakan *record-record* yang ada di dalam *table* tersebut. Kelebihan utama basis data non-relasional dibandingkan dengan basis data relasional adalah proses skalabilitas yang mudah dan fleksibilitas dalam penyimpanan data [22, 23, 24]. Hal tersebut tidak dapat diakomodasi oleh basis data relasional, karena basis data relasional mengharuskan data-data di dalamnya bersifat homogen, sehingga akan menyulitkan proses penyimpanan ketika data-data tersebut memiliki banyak varian dalam jumlah besar [25].

Salah satu contoh implementasi Firestore adalah adalah untuk pembangunan aplikasi berbasis Android yang dapat melakukan monitoring tingkat gula darah seseorang dari jarak jauh [26]. Mula-mula, seperangkat alat sensor berbasis infra merah digunakan untuk melakukan pengecekan gula darah dari pasien. Kemudian, nilai gula darah tersebut ditransmisikan ke aplikasi Android milik pasien untuk diteruskan ke basis data Firestore. Apabila posisi data telah tersimpan di *cloud*, maka dokter dapat melakukan monitoring terhadap pasien tersebut dengan cara mengunduh data gula darah dari *cloud* melalui aplikasi Android yang telah disediakan. Dengan demikian, dokter dapat terus memonitor banyak pasien sekaligus secara terus menerus tanpa perlu menunggu untuk bertatap muka secara langsung.

Penelitian lain mencoba membandingkan Firestore dengan basis data relasional SQL untuk pengelolaan data dengan struktur yang cukup kompleks [27]. Implementasi Firestore bertujuan untuk menggantikan basis data relasional yang dianggap terlalu membebani *server*, terlebih jika *server* tersebut dikelola secara mandiri. Berdasarkan hasil pengujian, secara keseluruhan Firestore menghasilkan efisiensi yang lebih baik dibandingkan dengan basis data relasional.

Contoh lain dari penerapan Firestore adalah pada pembangunan aplikasi *online quiz* [28]. Pada aplikasi tersebut, Firestore digunakan untuk menyimpan data *quiz* beserta jawaban yang dikirimkan oleh para peserta. Pada data *quiz*, jumlah atribut yang disimpan tidak banyak,

namun kecepatan pengiriman jawaban menjadi prioritas utama dalam penyimpanan data. Hal ini dikarenakan *quiz* memiliki waktu pengerjaan yang terbatas, sehingga data harus dapat dikirimkan secara cepat. Berdasarkan hasil pengujian, data *quiz* dapat disimpan dengan baik dan tepat waktu.

Berdasarkan penelitian-penelitian sebelumnya, maka dapat disimpulkan bahwa layanan *cloud computing*, khususnya basis data Firestore, dapat digunakan pada berbagai macam keperluan yang membutuhkan peran *server* sebagai pusat penyimpanan data dengan hasil yang cukup baik. Dengan demikian, akan digunakan salah satu layanan *cloud computing* Firebase, yaitu basis data non-relasional berbasis dokumen bernama Firestore untuk keperluan pengelolaan data pesan yang dapat diakses pada perangkat *mobile*, khususnya Android. Pada penelitian sebelumnya, Firestore digunakan untuk menyimpan data dengan mempertimbangkan salah satu aspek, yaitu kompleksitas data atau kecepatan penyimpanan. Sedangkan pada penelitian ini dilakukan pengujian dengan menggunakan dua aspek tersebut, yaitu relasi kompleksitas data terhadap kecepatan penyimpanan pesan.

III. METODE PENELITIAN

Penelitian yang dilakukan terbagi ke dalam beberapa tahap, yaitu pengumpulan data, perancangan sistem, pengujian sistem, serta analisa hasil pengujian.

A. Pengumpulan Data

Pengumpulan data dilakukan untuk mencari segala jenis variasi pesan berisi informasi yang terdapat pada berbagai aplikasi di *smartphone*. Data dikumpulkan melalui beberapa aplikasi yang memiliki unsur penyampaian informasi seperti aplikasi *chatting*, *email*, dan media sosial. Dari seluruh jenis informasi yang berhasil dikumpulkan, proses analisa dilakukan terhadap data pesan-pesan tersebut berdasarkan atributnya, kemudian dikategorikan menjadi tiga buah kategori yaitu:

1) Pesan standar: yaitu jenis pesan yang biasa dijumpai pada *e-mail*, di mana pesan ini dapat berbentuk surat, iklan, atau pengumuman dengan konten yang cukup panjang. Pesan ini dapat memiliki *formatting* seperti huruf besar/kecil, huruf tebal, cetak miring, warna huruf, dan sejenisnya. Umumnya proses *formatting* menggunakan HTML dan CSS agar dapat diproses pada platform *website* maupun *mobile*. Pesan jenis ini dapat memiliki konten yang panjang karena berisi banyak kalimat sekaligus dalam satu buah pesan, serta memiliki frekuensi untuk saling membalas pesan yang relatif rendah dengan jeda waktu yang lama.

2) Pesan catatan: yaitu jenis pesan yang digunakan sebagai pelengkap dari konten utama, di mana konten tersebut biasanya berupa non-teks dokumen seperti gambar, audio, atau video. Contohnya seperti pada aplikasi media sosial Instagram, di mana konten utama harus berupa gambar (foto), video, atau dokumen lainnya, sehingga pesan hanya akan digunakan sebagai pelengkap atau *caption* yang sifatnya opsional. Pesan catatan hampir serupa dengan pesan standar, namun perbedaan utama keduanya adalah pesan catatan memiliki konten berupa teks mentah (*raw*) tanpa adanya *formatting*, sehingga dapat langsung ditampilkan tanpa memerlukan proses tambahan.

3) Pesan singkat: yaitu jenis pesan yang biasa digunakan dalam percakapan melalui SMS atau aplikasi *chatting* seperti WhatsApp dan Telegram. Pesan jenis ini umumnya tidak memiliki *formatting* apapun, karena hanya berisi sedikit informasi yang langsung ke inti pesan. Umumnya pada satu buah pesan singkat hanya terdapat satu kalimat saja karena pesan jenis ini didesain untuk langsung dibalas dalam waktu singkat, sehingga frekuensi untuk saling membalas pesan relatif lebih tinggi dari pesan standar.

Dari ketiga kategori pesan yang telah disebutkan di atas, maka dibuat struktur dokumen penyimpanan yang dapat mengakomodasi ketiga jenis pesan tersebut seperti ditunjukkan pada Tabel 1.

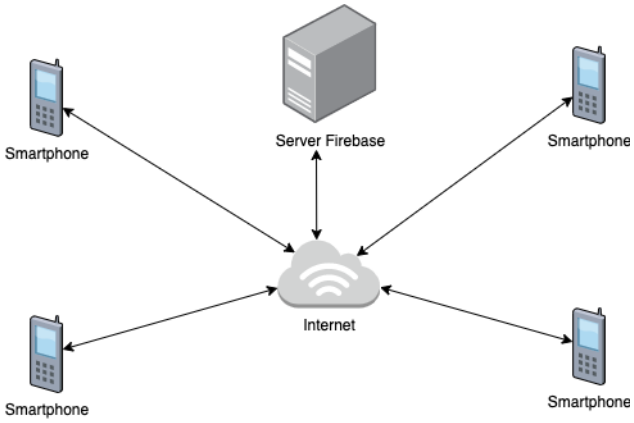
TABEL I
STRUKTUR DATA UNTUK DOKUMEN PESAN

attribute	jenis pesan		
	standar	catatan	singkat
id	w	w	w
jenis	w	w	w
pengirim	w	w	w
penerima	w	w	w
isi	w	o	w
media	o	w	o
formatting	w	o	o
status_pengiriman	w	w	w
tanggal_pengiriman	w	w	w
w = wajib, o = opsional			

Pada Tabel I, sebagian *attribute* yang bersifat wajib akan dibuat secara otomatis pada aplikasi, misalnya attribute id, id_pengirim, status_pengiriman, dan tanggal_pengiriman. Sedangkan *attribute* lainnya harus ditentukan secara mandiri oleh pengguna. Setelah menentukan bentuk struktur data, dihasilkan rancangan arsitektur perangkat lunak seperti yang ditunjukkan pada Gambar 1.

B. Perancangan Sistem

Pada rancangan arsitektur yang ditunjukkan oleh Gambar 1, seluruh perangkat dihubungkan melalui jaringan internet. Seluruh perangkat *smartphone* dapat mengirim maupun menerima data dari *server* berbasis Firebase yang memiliki komponen Firestore di dalamnya dengan menggunakan jaringan internet. Aplikasi pada *smartphone* Android yang digunakan untuk pengujian dibangun dengan menggunakan bahasa pemrograman Kotlin menggunakan Android SDK versi 30.



Gambar. 1 Arsitektur perangkat lunak

C. Pengujian Sistem

Tahap pengujian dilakukan dengan melakukan pengiriman pesan sebanyak 1000 pesan untuk masing-masing varian jenis pesan. Masing-masing pesan memiliki variasi perbedaan parameter di sisi banyaknya isi pesan (jumlah kata), adanya pemrosesan *formatting*, serta besarnya ukuran media yang dikirimkan seperti yang ditunjukkan pada Tabel II. Jumlah kata maksimal sebanyak 500 kata untuk pesan standar ditentukan berdasarkan jumlah rata-rata dokumen teks yang biasa digunakan di dalam *e-mail* atau deskripsi pada *website*. Sedangkan jumlah maksimal 100 kata untuk pesan catatan dan pesan singkat ditentukan berdasarkan rata-rata jumlah kata yang biasa digunakan masyarakat di dalam aplikasi media sosial maupun aplikasi pesan singkat. Untuk ukuran media maksimal 5000 KB (5 MB) ditentukan berdasarkan rata-rata ukuran foto atau gambar yang biasa digunakan masyarakat di dalam aplikasi media sosial setelah melalui proses kompresi. Seluruh variasi nilai dari parameter tersebut dibuat secara acak dan tersebar secara merata di dalam 1000 data uji untuk setiap varian jenis pesan.

TABEL II
METODE PENGUJIAN PENYIMPANAN PESAN

attribute	jenis pesan		
	standar	catatan	singkat
isi	1 s/d 500 kata	1 s/d 100 kata	
formatting	ya	tidak	
media	100 s/d 5000 KB		-

Dari variasi parameter tersebut dilakukan pengukuran performa berdasarkan waktu yang dibutuhkan untuk memproses sebuah pesan sejak dikirimkan hingga berhasil diterima oleh penerima pesan. Terdapat tiga kategori batas waktu respon sistem terkait apa yang dirasakan oleh pengguna [29]. Untuk waktu respon di bawah 0.1 detik tidak dibutuhkan mekansime khusus seperti umpan balik (*feedback)* karena respon tersebut bersifat instan. Untuk waktu respon antara 0.1 sampai 1 detik, pengguna dapat merasakan adanya jeda (*delay*), namun tidak diperlukan adanya umpan balik karena rentang waktu tersebut masih dalam batas toleransi pengguna untuk tetap berada di depan layar. Sedangkan untuk waktu respon 1 sampai 10 detik diperlukan adanya *feedback* pada antarmuka agar pengguna tahu bahwa sistem sedang bekerja, jika tidak ada *feedback* apapun maka kemungkinan besar sistem tersebut akan ditinggalkan oleh pengguna untuk melakukan aktivitas lain.

Perangkat yang digunakan untuk pengujian adalah *smartphone* berbasis Android dengan prosesor Snapdragon 778G 5G dan memori 8GB. Sedangkan untuk jaringan, menggunakan koneksi internet *broadband* dengan kecepatan kurang lebih 70 Mbps untuk *upload* dan kurang lebih 80 Mbps untuk *download* (berdasarkan speedtest.net) pada saat pengujian dilakukan.

D. Analisa Hasil Pengujian

Analisa dilakukan terhadap seluruh data hasil pengujian yang terdiri dari tiga kategori, yaitu pesan standar, pesan catatan, dan pesan singkat. Setiap kategori terdiri dari 1000 data pesan, sehingga total terdapat 3000 data yang telah diuji. Proses analisa dilakukan dengan menghitung nilai rata-rata, standar deviasi, *confidence interval*, nilai minimum dan nilai maksimum untuk setiap kategori pesan. Nilai-nilai tersebut akan divisualisasikan dengan menggunakan diagram plot untuk melihat sebaran data hasil pengujian. Pada data dari kelompok yang memiliki nilai jumlah kata terkecil dan terbesar, dilakukan uji T-Test untuk mengetahui apakah terdapat perbedaan yang signifikan terhadap kedua kelompok tersebut.

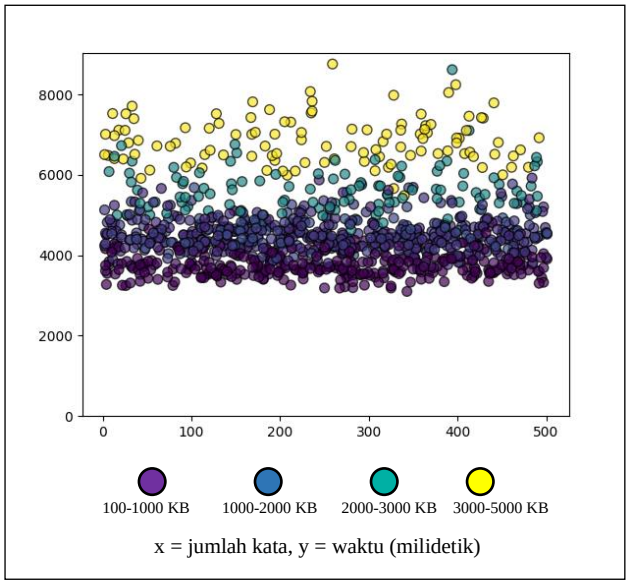
IV. HASIL DAN PEMBAHASAN

Hasil pengujian yang telah dilakukan berdasarkan skema pengujian pada Tabel II menunjukkan beberapa temuan berikut. Pada pengujian tahap pertama untuk jenis data pesan standar, di mana data tersebut terdiri dari 1 sampai 500 kata dan memiliki *file* lampiran berupa gambar dengan variasi ukuran antara 100 sampai 5000 *KiloBytes* (KB). Hasil pengujian untuk data pesan standar ditunjukkan pada Tabel 3.

TABEL III
HASIL PENGUJIAN PESAN STANDAR (MILIDETIK)

rata-rata	4644.012727
standar deviasi	1011.310069
confidence interval	59.76350773
min	3101
max	10844

Berdasarkan Tabel III, didapatkan rata-rata waktu yang dibutuhkan untuk mengirimkan data pesan ke *server* adalah 4644 milidetik atau 4.6 detik, dengan waktu tercepat adalah 3101 milidetik atau 3.1 detik dan waktu terlama adalah 10844 milidetik atau 10.8 detik. Diagram plot dari Tabel III ditunjukkan pada Gambar 2.



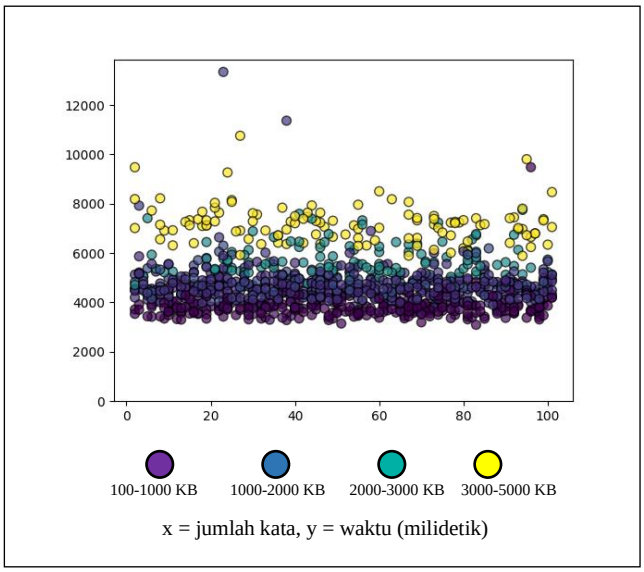
Gambar. 2 Hasil pengujian pesan standar

Pada diagram plot Gambar 2, dapat terlihat bahwa rata-rata waktu pengiriman data pesan standar berada pada rentang angka 3 sampai 8 detik. Pada diagram tersebut, dapat terlihat bahwa jumlah kata (1-500 kata) tidak berpengaruh terhadap lama waktu pengiriman data karena arah sebaran plot dengan warna serupa bergerak secara mendatar (horizontal). Hal tersebut dibuktikan dengan menggunakan uji T-Test untuk data dengan jumlah kata 1-100 dibandingkan dengan data yang memiliki jumlah kata 400-500 untuk ukuran media yang sama (100-1000 KB), menghasilkan nilai *p-value* sebesar 0.6393. Maka dapat disimpulkan bahwa kedua kelompok tidak memiliki perbedaan yang signifikan atau dengan kata lain dapat dianggap sama. Sementara itu, jumlah *file* memberikan dampak yang lebih besar terhadap lama waktu pengiriman data, hal ini ditandai dengan sebaran plot yang memiliki warna berbeda, berada pada area sumbu y yang berbeda satu sama lain. Pada diagram plot Gambar 2 dapat terlihat bahwa varian ukuran *file* terkecil (100-1000 KB) yang ditandai dengan plot berwarna ungu memiliki rata-rata waktu pengiriman data antara 3 sampai 5 detik, sementara itu varian ukuran *file* terbesar (3000-5000 KB) memiliki rata-rata waktu pengiriman data antara 5-8 detik.

TABEL IV
HASIL PENGUJIAN PESAN CATATAN (MILIDETIK)

rata-rata	4830.909754
standar deviasi	1159.232936
confidence interval	68.59863734
min	3095
max	13343

Berdasarkan Tabel IV, didapatkan rata-rata waktu yang dibutuhkan untuk mengirimkan data pesan dengan jenis catatan ke *server* adalah 4831 milidetik atau 4.8 detik, dengan waktu tercepat adalah 3095 milidetik atau 3.1 detik dan waktu terlama adalah 13343 milidetik atau 13.3 detik. Diagram plot dari Tabel IV ditunjukkan pada Gambar 3.



Gambar. 3 Hasil pengujian pesan catatan

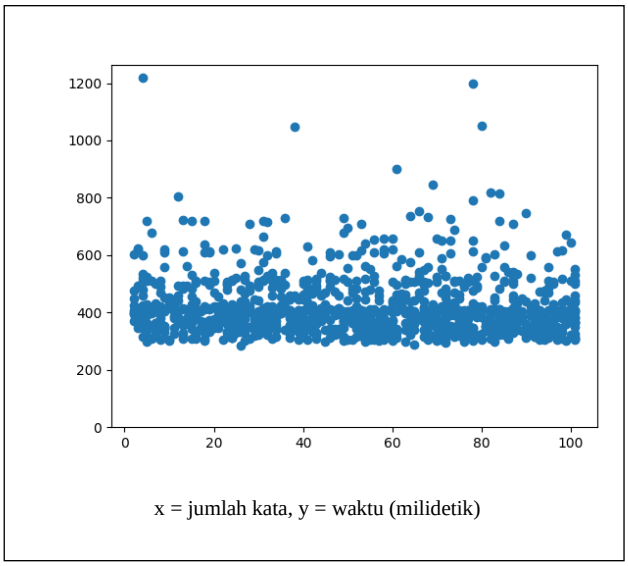
Pada diagram plot Gambar 3, dapat terlihat bahwa rata-rata waktu pengiriman data pesan catatan berada pada rentang angka 3 sampai 8 detik. Hasil ini hampir serupa dengan hasil pengujian untuk pesan standar yang ditunjukkan pada Gambar 2. Hal ini membuktikan bahwa jumlah kata (1-100 kata) pada pengujian pesan catatan tidak memberikan perbedaan yang signifikan jika dibandingkan dengan hasil pengujian pada pesan standar (1-500 kata), karena rata-rata waktu pengiriman data secara keseluruhan sama-sama berada pada angka 3 sampai 8 detik. Pengujian lain dilakukan dengan menggunakan uji T-Test untuk data dengan jumlah kata 1-20 dibandingkan dengan data yang memiliki jumlah kata 80-100 untuk ukuran media yang sama (100-1000 KB), menghasilkan nilai *p-value* sebesar 0.4628. Maka dapat disimpulkan bahwa kedua kelompok tidak memiliki perbedaan yang signifikan atau dengan kata lain dapat dianggap sama. Hasil pengujian tersebut juga membuktikan bahwa penambahan proses untuk *formatting* pada pesan standar juga tidak memberikan pengaruh yang signifikan terhadap waktu.

TABEL V
HASIL PENGUJIAN PESAN SINGKAT (MILIDETIK)

rata-rata	420.3427273
standar deviasi	101.7190685
confidence interval	6.011102354
min	283
max	1218

Berdasarkan Tabel V, didapatkan rata-rata waktu yang dibutuhkan untuk mengirimkan data pesan ke *server* adalah 420 milidetik atau 0.4 detik, dengan waktu tercepat adalah

283 milidetik atau 0.3 detik dan waktu terlama adalah 1218 milidetik atau 1.2 detik. Diagram plot dari Tabel V ditunjukkan pada Gambar 4.



Gambar. 4 Hasil pengujian pesan singkat

Pada diagram plot Gambar 4, dapat terlihat bahwa rata-rata waktu pengiriman data pesan singkat berada pada rentang 0.3 sampai 0.7 detik. Perbedaan utama data pesan singkat dibandingkan dengan data pesan standar dan catatan adalah tidak adanya lampiran *file* apapun di dalamnya, sehingga isi pesan hanya terdiri dari teks saja. Diagram plot Gambar 4 menunjukkan bahwa jumlah kata tidak berpengaruh terhadap lama waktu pengiriman data dari *client* ke *server*. Sebagai perbandingan, nilai rata-rata untuk pengiriman data dengan jumlah 1-20 kata memiliki nilai 427.61 milidetik (0.43 detik), sedangkan pengiriman data dengan jumlah 80-100 kata memiliki nilai rata-rata 410.93 milidetik (0.41 detik). Hasil pengujian nilai T-Test menghasilkan *p-value* 0.051 untuk nilai alpha 0.05, yang memiliki arti bahwa kedua hasil tersebut dapat dianggap serupa.

V. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa performa penyimpanan data pesan dari *client* ke *server* pada basis data non-relasional berbasis dokumen (dalam penelitian ini menggunakan Firestore) tidak dipengaruhi oleh besar kecilnya data berbentuk teks dengan rata-rata waktu pengiriman 0.4 detik (1-500 kata). Proses penyimpanan data lebih dipengaruhi oleh besar kecilnya data dalam bentuk *file* dengan rata-rata waktu pengiriman data sekitar 4 detik (ukuran 100-5000 KB). Dengan demikian, dapat disimpulkan bahwa basis data non-relasional berbasis dokumen cukup andal digunakan untuk membangun aplikasi perpesanan dengan isi berupa teks. Untuk isi lain berupa media, basis data non-relasional berbasis dokumen masih dapat digunakan karena masih berada di dalam batas waktu toleransi (10 detik), namun penggunaannya harus disertai dengan adanya *feedback* pada antarmuka.

REFERENSI

[1] N. Schillewaert and P. Meulemeester, "Comparing Response Distributions of Offline and Online," *International Journal of Market Research*, vol. 47, no. 2, pp. 163-178, 2005.

[2] L. Zhang, Z. Wang, T. Mei and D. D. Feng, "A Scalable Approach for Content-Based Image Retrieval in Peer-to-Peer Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 858 - 872, 2016.

[3] A. Barker, J. B. Weissman and J. I. v. Hemert, "Reducing Data Transfer in Service-Oriented Architectures: The Circulate Approach," *IEEE Transactions on Services Computing*, vol. 5, no. 3, 2012.

[4] S. Sarkar, S. Gayen and S. Bilgaiyan, "Android Based Home Security Systems Using Internet of Things(IoT) and Firebase," in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2018.

[5] H. Varshney, A. S. Allahloh and M. Sarfraz, "IoT Based eHealth Management System Using Arduino and Google Cloud Firestore," in *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, Aligarh, India, 2019.

[6] W. Y. Mok, "A Logical Database Design Methodology for MongoDB NoSQL Databases," in *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, Singapore, 2021.

[7] M. R. Alifi, T. Semiawan, D. C. Lieharyani and H. Hayati, "Relational Data Modeling on the Document-Based NoSQL," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 11, no. 3, pp. 183-191, 2022.

[8] D. Zhang, Y. Wang, Z. Liu and S. Dai, "Improving NoSQL Storage Schema Based on Z-Curve for Spatial Vector Data," *IEEE Access* , vol. 7, pp. 78817 - 78829, 2019.

[9] F. Davardoost, A. B. Sangar and K. Majidzadeh, "Extracting OLAP Cubes From Document-Oriented NoSQL Database Based on Parallel Similarity Algorithms," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 2, pp. 111 - 118, 2020.

[10] K. A. Nugraha, "Deteksi Area Parkir Mobil Berbasis Marker Menggunakan Moment Invariants dan K-NN," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 5, no. 1, pp. 112-121, 2019.

[11] K. A. Nugraha, "Metode Background Substraction untuk Pencarian Tempat Parkir Menggunakan Kamera Pengawas," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 1, pp. 92-101, 2020.

[12] K. MATSUI, "A Proposal of Web API Design for IoT Data Utilization Based on Smart Communities," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, Nara, Japan, 2018.

[13] K. A. Nugraha and L. K. P. Saputra, "Pemanfaatan Raspberry Pi untuk Sistem Penghitung Mobil Otomatis pada Kampus UKDW," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 3, no. 3, pp. 539-549, 2017.

[14] C. E. Swandi, K. A. Nugraha, D. Sebastian and Restyandito, "Middleware Development to Connect Telegram Messenger and Instant Messenger for the Elderly," in *The 5th International Conference on Information Technology and Digital Applications (ICITDA 2020)*, Yogyakarta, 2020.

[15] B. Lin, F. Zhu, J. Zhang, J. Chen, X. Chen, N. N. Xiong and J. L. Mauri, "A Time-Driven Data Placement Strategy for a Scientific Workflow Combining Edge Computing and Cloud Computing," *IEEE Transactions on Industrial Informatics* , vol. 15, no. 7, pp. 4254 - 4265, 2019.

[16] S. Wang, A. Zhou, F. Yang and R. N. Chang, "Towards Network-Aware Service Composition in the Cloud," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1122 - 1134, 2020.

[17] E. Hossny, S. Khattab, F. A. Omara and H. A. Hassan, "STAGER: Semantic-Based Framework for Generating Adapters of Service-Based Generic-API for Portable Cloud Applications," *IEEE Transactions on Services Computing* , vol. 14, no. 3, pp. 903 - 914, 2021.

[18] C. Wu, A. N. Toosi, R. Buyya and K. Ramamohanarao, "Hedonic Pricing of Cloud Computing Services," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 182 - 196, 2021.

[19] M. T. Islam, S. Karunasekera and R. Buyya, "Performance and Cost-Efficient Spark Job Scheduling Based on Deep Reinforcement Learning in Cloud Computing Environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 7, pp. 1695 - 1710, 2022.

[20] C. Yang, Y. Liu, X. Tao and F. Zhao, "Publicly Verifiable and Efficient Fine-Grained Data Deletion Scheme in Cloud Computing," *IEEE Access*, vol. 8, pp. 99393 - 99403, 2020.

[21] W.-J. Li, C. Yen, Y.-S. Lin, S.-C. Tung and S. Huang, "JustIoT Internet of Things based on the Firebase real-time database," in *2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)*, Hsinchu, Taiwan, 2018.

[22] S. Ramzan, I. S. Bajwa, B. Ramzan and W. Anwar, "Intelligent Data Engineering for Migration to NoSQL Based Secure Environments," *IEEE Access*, vol. 7, pp. 69042 - 69057, 2019.

[23] K. Goel and A. H. M. T. Hofstede, "Privacy-Breaching Patterns in NoSQL Databases," *IEEE Access*, vol. 9, pp. 35229 - 35239, 2021.

[24] A. Krechowicz, S. Deniziak and G. Łukawski, "Highly Scalable Distributed Architecture for NoSQL Datastore Supporting Strong Consistency," *IEEE Access*, vol. 9, pp. 69027 - 69043, 2021.

[25] S. Chouliaras and S. Sotiriadis, "Real-Time Anomaly Detection of NoSQL Systems Based on Resource Usage Monitoring," *IEEE Transactions on Industrial Informatics* , vol. 16, no. 9, pp. 6042 - 6049, 2019.

[26] B. E. Manurung, H. R. Munggaran, G. F. Ramadhan and A. P. Koesoema, "Non-Invasive Blood Glucose Monitoring using Near-Infrared Spectroscopy based on Internet of Things using Machine Learning," in *2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129)*, Depok, West Java, Indonesia, 2019.

[27] I. M. Sukarsa, I. K. A. M. Antara, P. W. Buana, I. P. A. Bayupati, N. W. Wisswani and D. W. Puteri, "Data storage model in low-cost mobile applications," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 28, no. 2, pp. 1128-1138, 2022.

[28] Y. Sukmana and Y. Rosmansyah, "The Use of Cloud Firestore For Handling Real-time Data Updates: An Empirical Study of Gamified Online Quiz," in *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT)*, Sanya, China, 2021.

[29] J. Nielsen, "Response Times: The 3 Important Limits," Nielsen Norman Group, 1 Januari 1993. [Online]. Available: https://www.nngroup.com/articles/response-times-3-important-limits/. [Accessed 1 Oktober 2022].