

SISTEM INFORMASI DISPOSISI SURAT BERBASIS API MENGUNAKAN GRAPHQL

Rico Sandya Novenza¹, I Gede Susrama Mas Diyasa², Sugiarto³

E-mail: ¹ricosandyca.if@gmail.com, ²igsusrama.if@upnjatim.ac.id, ³sugiarto.if@upnjatim.ac.id

^{1,2,3}Program Studi Informatika, Fakultas Ilmu Komputer, UPN "Veteran" Jawa Timur

Abstrak

Dalam suatu instansi, surat merupakan suatu hal yang penting. Tidak hanya sebagai alat komunikasi, melainkan sebagai bukti atas sebuah kegiatan yang telah dilakukan maupun sebagai pemberi perintah. Pengelolaan surat harus dilakukan setepat-tepatnya sehingga informasi dapat disampaikan dengan cepat dan tepat ketika diperlukan. Namun, surat yang baru dibuat tidak dapat langsung dikirim ke penerima tujuan, tetapi harus melalui beberapa proses yang tidak singkat dan hal tersebut tentu akan memakan banyak waktu dan membuat proses pengiriman kurang efisien. Bahkan ada masanya pengiriman surat mengalami masalah yang membuat informasi yang disampaikan terganggu. Untuk menanggulangi masalah diatas, peneliti membangun sebuah sistem surat berbasis *GraphQL API*. Hal tersebut dilakukan untuk mengatasi masalah-masalah instansi terutama pada bagian penyuratan. Sistem yang dibangun berbasis *API*, agar nantinya *API* yang dibangun dapat dikonsumsi oleh berbagai *platform*. *GraphQL* digunakan dikarenakan lebih efektif dan efisien dibandingkan dengan arsitektur *API* serupa. Metode penelitian yang digunakan adalah metode *waterfall* yang sudah terbukti berhasil menangani penelitian dengan studi kasus serupa. Hasil dari penelitian ini adalah terbentuknya sebuah sistem surat masuk, keluar dan disposisi berbasis *GraphQL API* yang sudah lulus uji menggunakan metode *equivalence partitioning*. *API* yang sudah dibangun nantinya dapat langsung dikonsumsi oleh *client* agar aplikasi lebih interaktif bagi pengguna.

Kata kunci: *GraphQL, API, Surat Masuk, Surat Keluar, Disposisi.*

1. PENDAHULUAN

Teknologi informasi merupakan suatu bagian penting di dalam kehidupan masyarakat saat ini. Hal ini didasarkan pada perkembangan jaman yang semakin modern seperti contoh penggunaan sebuah sistem aplikasi untuk memudahkan pekerjaan ataupun aktivitas [1]. Dalam suatu instansi, surat merupakan alat komunikasi yang penting. Suatu instansi yang menerima surat masuk dan surat keluar tidak hanya sebagai alat komunikasi, melainkan sebagai bukti atas sebuah kegiatan yang telah dilakukan.

Pengelolaan surat harus dilakukan setepat-tepatnya sehingga informasi surat dapat disampaikan dengan cepat dan tepat ketika diperlukan [2]. Suatu surat yang baru dibuat tidak dapat langsung dikirim ke penerima tujuan, tetapi surat harus melalui persetujuan dan balasan dari beberapa tingkat [3]. Hal tersebut tentu akan memakan banyak waktu dan membuat proses pengiriman surat kurang efisien. Bahkan ada masanya surat mengalami pending pengiriman dikarenakan faktor jam kerja dan faktor lainnya yang membuat pengiriman surat terganggu. Kendala lainnya terkadang surat tidak sampai ke tujuan disposisinya dikarenakan hilang.

Sebagai langkah untuk menanggulangi beberapa masalah diatas, peneliti menganggap perlu mengembangkan sebuah sistem disposisi surat (SISURAT) dengan basis *API (Application Programming Interface)* dengan memanfaatkan teknologi *GraphQL*. Dengan memanfaatkan konsep *API*, sistem dapat saling terintegrasi dengan

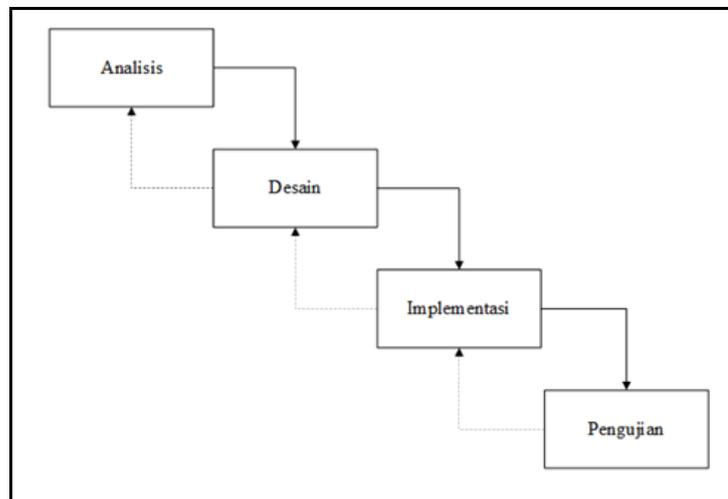
berbagai aplikasi *client* [4]. Hal ini juga sangat berguna jika pada waktu mendatang proses bisnis difasilitasi menuntut interaksi dengan sistem lain.

GraphQL dapat mengembalikan data sesuai dengan permintaan aplikasi *client* [4]. Sehingga data yang diterima oleh aplikasi *client* tidak ada yang percuma. Pada pengembangannya, peneliti menggunakan *TypeScript* sebagai bahasa pemrogramannya dengan memanfaatkan *library* dan *framework* untuk menunjang kebutuhan pembangunan aplikasi. Untuk membangun API berbasis *GraphQL*, peneliti menggunakan *Apollo Server*. *Library* ini dipilih karena dirasa cocok untuk digunakan dalam pengembangan sistem berbasis *GraphQL* [5].

Gagasan diatas menjadi alasan kuat untuk melakukan penelitian ini. Dengan harapan, penelitian ini mampu membangun sebuah sistem berbasis API yang dapat membantu proses penyuratan. Diharapkan pula peneliti ini agar menjadi sumber literatur dalam pemilihan teknologi pada API di penelitian selanjutnya.

2. METODOLOGI

Tahapan pengembangan sistem menggunakan pendekatan SDLC (*System Development Life Cycle*) dengan metode *waterfall* [6]. Metode ini digunakan karena setiap tahapan memiliki spesifikasinya sendiri, sehingga suatu sistem dapat mudah dikembangkan sesuai dengan yang diinginkan. Selain itu, metode *waterfall* juga telah terbukti berhasil dalam peneliti dengan studi kasus serupa [7][8].



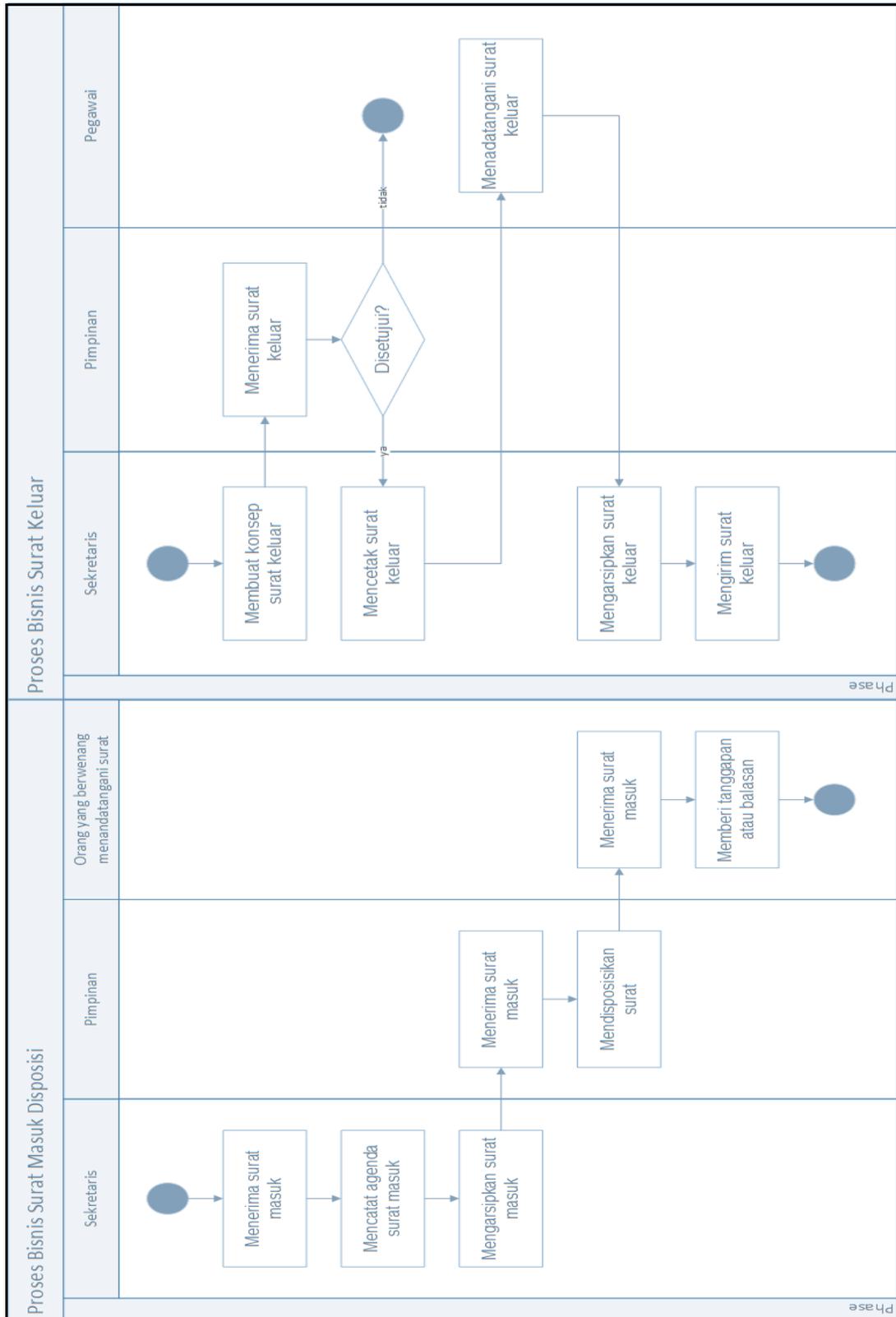
Gambar 2.1 Tahapan penelitian model *waterfall*.

Pada pengembangan sistem menggunakan model *waterfall*, terdapat empat tahapan. Diantaranya adalah analisis, desain, implementasi dan pengujian. Alur tahapan dapat kembali ke tahapan sebelumnya jika ternyata pada tahapan tersebut didapat data yang masih kurang. Sehingga sistem yang dibangun nanti benar-benar sesuai dengan keinginan.

2.1 Analisis Proses Bisnis

Tahapan pertama pada pengembangan sistem adalah analisis proses bisnis. Proses bisnis yang di analisis tentang proses surat menyurat pada suatu instansi. Pada tahapan ini dilakukan pengumpulan informasi terkait proses bisnis surat masuk dan surat keluar. Peneliti mengumpulkannya dari beberapa literatur sehingga terbentuk suatu *workflow* pada surat masuk dan surat keluar.

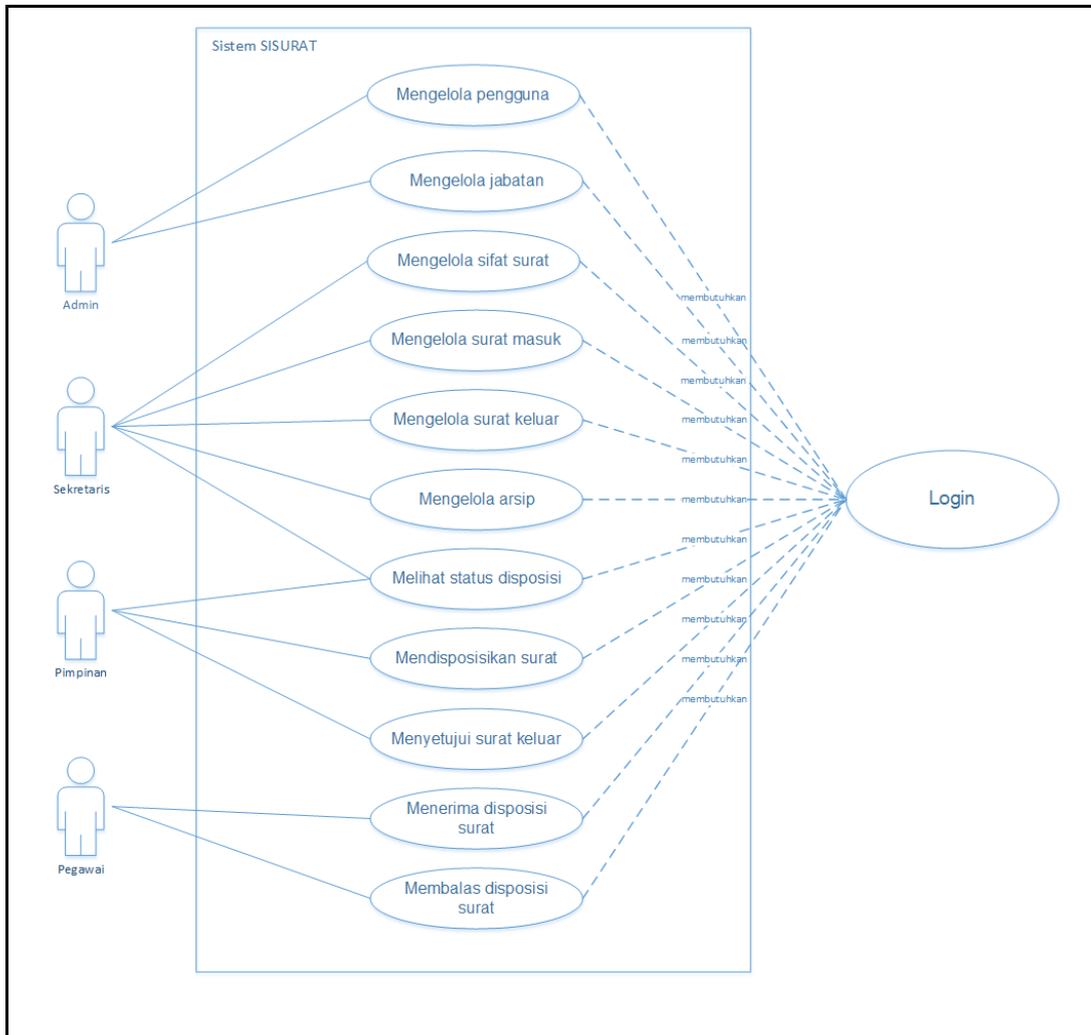
Sebagai hasil dari tahapan analisi ini, dibuatlah suatu *workflow* yang menjelaskan proses suatu surat masuk dan surat keluar



Gambar 2.2 Workflow Surat Masuk dan Surat Keluar.

2.2 Use Case Diagram

Analisa *use case diagram* dilakukan untuk menentukan aksi apa saja yang dapat dilakukan oleh pelaku dalam sistem. Berdasarkan analisa sebelumnya, terdapat tiga pelaku utama pada proses surat menyurat. Maka dari itu, peneliti membuat tiga pelaku dalam sistem ditambah satu pelaku admin sebagai kunci berjalannya sistem. Setiap pelaku memiliki hak dan perannya masing-masing. Suatu pelaku tidak dapat melakukan suatu aksi karena hal tersebut bukan haknya, sehingga hal tersebut perlu ditangani dalam sistem nantinya.

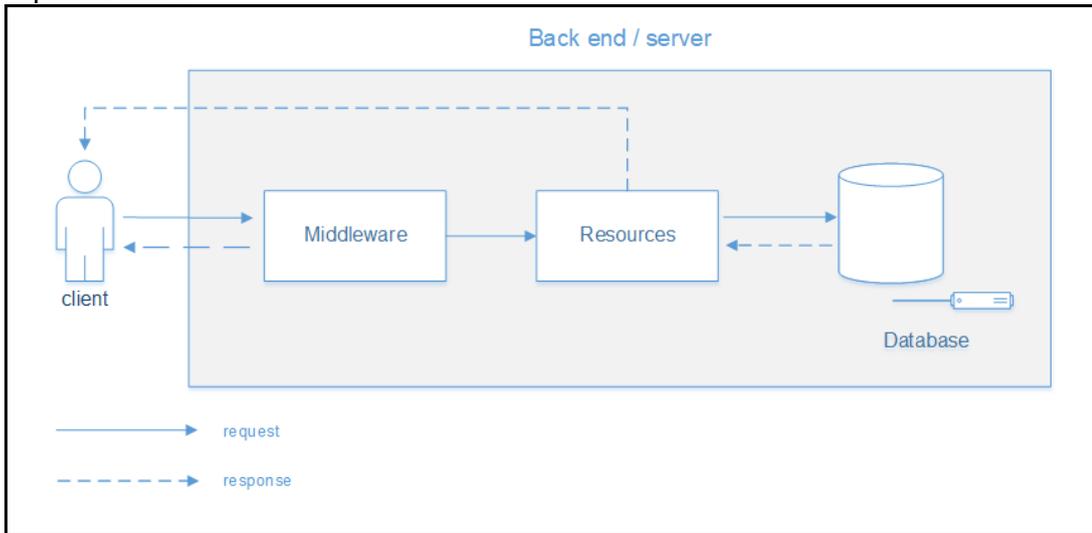


Gambar 2.3 Analisa Use Case Diagram.

2.3 Desain Arsitektur Server

Sistem yang dibangun adalah berbasis API dengan menerapkan konsep *request-response* antara *client* dan *server* dalam bertukar informasi. Terdapat dua tipe sumber data pada sistem, yaitu sumber data yang tidak membutuhkan otorisasi dengan kata lain sumber data ini dapat diakses oleh pengguna diluar pelaku utama sistem, dan sumber data yang membutuhkan otorisasi dengan kata lain hanya pengguna yang terotorisasi lah yang dapat mengakses sumber data tersebut [9].

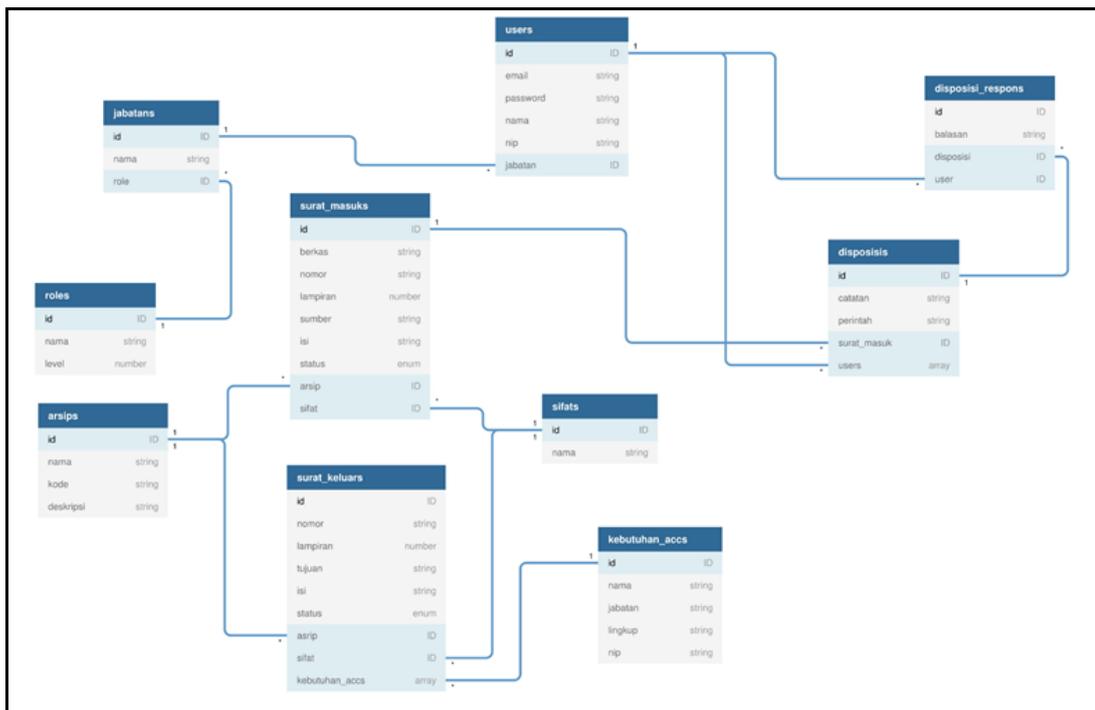
Untuk menangani hal tersebut, maka diterapkanlah *middleware* pada setiap API untuk menyaring data otorisasi yang dikirim melalui *request header* yang dilakukan oleh aplikasi *client*.



Gambar 2.4 Arsitekur server.

2.4 Desain Database

Pada sistem yang dibangun akan digunakan sebuah basis data yang berfungsi untuk menyimpan semua data sistem. Desain basis data ini mengikuti aturan struktur pada basis data bertipe *NoSQL* dengan harapan desain ini dapat diterapkan pada *MongoDB* sebagai DBMS (*Database Management System*) pada sistem yang akan dibangun.



Gambar 2.5 Desain Database.


```

1 query {
2   suratMasuks {
3     berkas
4     nomor
5     isi
6     sifat { nama }
7   }
8 }

```

```

1 {
2   "Authorization": "Bearer eyJhbGciOiJIUzI1NiI
3 }

```

```

{
  "data": {
    "suratMasuks": [
      {
        "berkas": "2020-01-22T14:15:58.991Z.pdf",
        "nomor": "SRM001",
        "isi": "Undangan rapat pengelolaan sumber daya dan teknologi informasi",
        "sifat": {
          "nama": "Biasa"
        }
      }
    ]
  }
}

```

TRACING QUERY PLAN

Gambar 3.2 Query “suratMasuks”.

Query “suratMasuks” digunakan untuk mengambil semua data surat masuk pada database. API ini hanya dapat diakses oleh Sekretaris. Terdapat juga operasi yang lain seperti CRUD (*create, read, update, delete*) pada query “createSuratMasuk”, “updateSuratMasuk” dan “deleteSuratMasuk”.

```

1 query {
2   suratKeluars {
3     tujuan
4     nomor
5     kebutuhan_accs {
6       nama
7       nip
8       jabatan
9     }
10  }
11 }

```

```

1 {
2   "Authorization": "Bearer eyJhbGciOiJIUzI1NiI
3 }

```

```

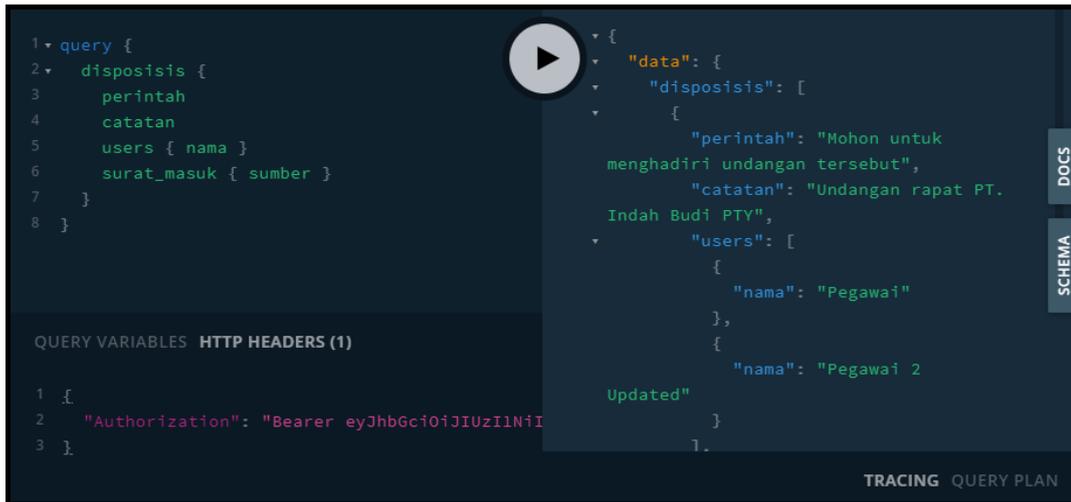
{
  "data": {
    "suratKeluars": [
      {
        "tujuan": "Rektor Kampus UPN",
        "nomor": "SRK001",
        "kebutuhan_accs": [
          {
            "nama": "Rico Sandyca Novenza",
            "nip": "1634010052",
            "jabatan": "Kepala Divisi Teknologi"
          }
        ]
      },
      {
        "tujuan": "Ketua Jurusan Teknik Informatika"
      }
    ]
  }
}

```

TRACING QUERY PLAN

Gambar 3.3 Query “suratKeluars”.

Query “suratKeluars” digunakan untuk mengambil semua data surat keluar pada database. API ini hanya dapat diakses oleh Sekretaris. Sama halnya dengan query “suratMasuks”, query “suratKeluars” juga memiliki operasi CRUD pada “createSuratKeluar”, “updateSuratKeluar” dan “deleteSuratKeluar” serta operasi lain penunjang kebutuhan penyuratan pada surat keluar.



Gambar 3.4 Query “disposisi”.

Query “disposisi” merupakan query yang dapat mengambil seluruh data disposisi pada semua surat masuk. Query ini hanya dapat diakses oleh Sekretaris dan Pimpinan. Pada data disposisi ini pula terdapat semua balasan dari Pegawai yang menerima disposisi. Selain itu, juga terdapat operasi CRUD yang sejenis dengan fitur ini seperti “createDisposisi”, “updateDisposisi” dan “deleteDisposisi” dimana operasi-operasi ini hanya dapat dilakukan oleh Pimpinan.

3.2 Hasil Pengujian

Tabel 3.1 Hasil Pengujian Fungsionalitas API.

No	Kelas uji	Jabatan	API	Hasil yang diharapkan	Hasil
1	Mengelola pengguna	Admin	users, user, createUser, updateUser, deleteUser	Response sukses dan bekerja dengan baik	√
		Selain Admin		Semua request ditolak karena tertotorisasi sebagai pengguna yang tidak memiliki hak (<i>Forbidden</i>)	√
2	Mengelola surat masuk	Sekretaris	suratMasuks, suratMasuk, createSuratMasuk, updateSuratMasuk, deleteSuratMasuk	Response sukses dan bekerja dengan baik	√
		Selain Sekretaris		Semua request ditolak karena tertotorisasi sebagai pengguna yang tidak memiliki hak (<i>Forbidden</i>)	√

3	Mengelola surat keluar	Sekretaris	suratKeluar, suratKeluar, createSuratKeluar, updateSuratKeluar, deleteSuratKeluar	Response sukses dan bekerja dengan baik	√
		Selain sekretaris		Semua <i>request</i> ditolak karena tertotorisasi sebagai pengguna yang tidak memiliki hak (<i>Forbidden</i>)	√
4	Mengelola disposisi	Pimpinan	disposisi, disposisi, createDisposisi, updateDisposisi, deleteDisposisi	Response sukses dan bekerja dengan baik	√
		Selain pimpinan		Semua <i>request</i> ditolak karena tertotorisasi sebagai pengguna yang tidak memiliki hak (<i>Forbidden</i>)	√
5	Membalas disposisi	Pegawai	createBalasan-Disposisi	Response sukses dan bekerja dengan baik	√
		Selain pegawai		Semua <i>request</i> ditolak karena tertotorisasi sebagai pengguna yang tidak memiliki hak (<i>Forbidden</i>)	√

Tabel 3.2 Hasil Pengujian Keamanan Token Otentikasi.

No	Kelas Uji	Skenario Uji	Hasil yang diharapkan	Hasil
1	Kadaluwarsa token	Menggunakan <i>token</i> sebelum satu jam setelah melakukan otentikasi atau <i>login</i> pada sistem	<i>Token</i> dapat digunakan dan berfungsi dengan baik	√
		Menggunakan <i>token</i> diatas satu jam setelah melakukan otentikasi atau <i>login</i> pada sistem	<i>Token</i> tidak dapat digunakan, dan semua permintaan ditolak	√
		Melakukan <i>login</i> ulang saat <i>token</i> lama kadaluwarsa dan menggunakan <i>token</i> yang baru	<i>Token</i> dapat digunakan dan berfungsi dengan baik	√

		Menggunakan <i>token</i> baru diatas satu jam setelah melakukan otentikasi atau <i>login</i> kedua pada sistem	<i>Token</i> tidak dapat digunakan, dan semua permintaan ditolak	√
--	--	--	--	---

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan penelitian yang dilakukan dari proses perancangan, implementasi dan pengujian sistem, beberapa kesimpulan yang dapat diambil adalah:

1. Teknologi GraphQL memiliki konsep yang hampir sama dengan teknologi REST yang melibatkan server dan client dalam bertukar informasi yang dikembalikan dalam format JSON. Data yang dikembalikan oleh GraphQL lebih fleksibel sesuai dengan permintaan client sehingga tidak ada data yang percuma.
2. Sebagian besar resources pada server SISURAT membutuhkan otorisasi pengguna untuk mengaksesnya, sehingga semua data dan proses penyuratan lebih aman.
3. Semua API yang dibangun pada SISURAT telah lolos pengujian dengan menggunakan metode uji equivalence partitioning yang merupakan sinyal baik bagi SISURAT.

4.2 Saran

Terkait hasil akhir penelitan, peneliti memiliki beberapa saran untuk pengembangan sistem dan saran terkait penelitian selanjutnya dengan studi kasus serupa, diantaranya adalah:

1. Mengikutsertakan pihak yang ahli dalam di bidang terkait tentang surat pada suatu instansi untuk memperbaiki alur dan proses bisnis SISURAT agar lebih baik dan lebih sesuai dengan target.
2. Membangun aplikasi client atau front end yang dapat berupa website ataupun mobile yang mengkonsumsi API SISURAT agar aplikasi lebih interaktif bagi pengguna.
3. Menganalisa lebih lanjut terkait performa dari segi kecepatan akses data dari teknologi GraphQL dan REST dalam hal pembangunan API untuk menentukan teknologi yang lebih sesuai dengan kebutuhan sistem.

5. DAFTAR RUJUKAN

- [1] D. Rahmawati, N. Kumaladewi and Y. Sugiarti, "Sistem Informasi Disposisi Surat Berbasis Android," *Applied Information Systems and Management (AISM)*, vol. 1, no. 1, pp. 45-50, 2018.
- [2] J. A. Prawono and A. R. Pamungkas, "Sistem Informasi Pengelolaan Surat Masuk dan Surat Keluar di STMIK AUB Surakarta," *INFORMATIKA*, vol. 2, no. 1, pp. 26-33, 2015.
- [3] I. N. Diana, M. M. Anwar, R. S. N and I. G. Susrama M.D, "Analisis Perancangan Sistem Informasi Pengarsipan Dan Disposisi Surat Berbasis Chronological Filling

- System," *SEMINAR SANTIKA*, pp. 27-32, 2019.
- [4] J. Lind and K. Soames, *Detecting Cycles in GraphQL Schemas*, 16th ed., Swedish: Linköping University | Department of Computer and Information Science, 2019.
- [5] Meteor Development Group Inc., "Apollo GraphQL," 2019. [Online]. Available: <https://www.apollographql.com/>. [Accessed 18 Agustus 2019].
- [6] I. G. Susrama Mas Diyasa, A. Setiawan and M. Kholis, "Rancang Bangun Aplikasi "W-Mass (Weight Monitor Assistant)" Berbasis Android Studio dengan Bahasa," *Jurnal Penelitian*, vol. 4, no. 2, pp. 1-19, 2019.
- [7] K. G. Nuryani, R. Agustiansyah and N. Ambarsari, "Membangun Sistem Informasi Pengelolaan Surat Menggunakan Metode Waterfall Studi Kasus Direktorat Perencanaan dan Pengembangan Pendanaan Pembangunan," *e-Proceeding of Engineering*, vol. 2, no. 1, pp. 1098-1102, 2015.
- [8] Y. B. Anggara, S. and J. Lemantara, "Rancang Bangun Aplikasi Pengelolaan Surat Masuk dan Surat Keluar pada Pengadilan Tinggi Surabaya," *JSIKA*, vol. 5, no. 7, pp. 1-7, 2016.
- [9] I. G. Susrama Mas Diyasa, F. Nadhira and T. Ardianto, "Pemilihan Kegiatan Organisasi Mahasiswa Menggunakan Algoritma Probabilitas Bayes," *JANAPATI*, vol. 9, no. 1, pp. 45-54, 2020.