

SISTEM INFORMASI OUTBOUND BELANEGARA MENGGUNAKAN GRAPHQL API

¹Mohammad Iqbal Roziq, ²Chrystia Aji Putra, ³Agung Mustika Rizki, ⁴Sri Wibawani, ⁵Ariyono Setiawan, ⁶I Gede Susrama Mas Diyasa

^{1,2,3,6}Teknik Informatika, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur, ⁴Administrasi Negara, Fakultas Ilmu Sosial dan Ilmu Politik, UPN “Veteran” Jawa Timur, ⁵Manajemen Transportasi Udara, Politeknik Penerbangan Surabaya

Email: ¹m.iqbalroziq.if@gmail.com, ²ajiputra@upnjatim.ac.id, ³agung.mustika.if@upnjatim.ac.id,

⁴sriwibawani@upnjatim.ac.id, ⁵rmaryo4u@gmail.com, ⁶igsusrama.if@upnjatim.ac.id

Corresponding Author: igsusrama.if@upnjatim.ac.id

Abstrak. *Outbound belanegara merupakan merupakan kegiatan pendidikan karakter untuk membentuk kepribadian agar memiliki pondasi awal jiwa belanegara. Proses kegiatan ini dilakukan dengan melakukan praktikum lapangan yang melibatkan sekelompok mahasiswa dan beberapa orang pembina sebagai pemantau acara. Namun dengan adanya pandemi yang mewabah ke seluruh dunia, memaksa proses pendidikan yang berjalan luring berpindah ke metode daring. Dengan masalah yang sudah dijelaskan, maka dibutuhkan sebuah sistem yang dapat menggantikan kegiatan metode luring menjadi metode daring yang dapat melakukan pengawasan kegiatan kelompok maupun individual serta penilaian terhadap kegiatan tersebut. Sistem yang dibangun berbasis API sehingga nantinya dapat dimanfaatkan oleh berbagai macam platform. Perancangan sistem menggunakan metode waterfall yang sudah dibuktikan oleh peneliti lain dapat menangani studi kasus yang hampir sama. Hasil yang diperoleh dari penelitian ini yaitu sistem informasi outbound belanegara berbasis API menggunakan arsitektur GraphQL Query Language(GraphQL) yang nantinya dapat digunakan oleh pengembang antarmuka untuk membangun aplikasi yang optimal.*

Kata Kunci: *GraphQL, Outbound Belanegara, API.*

Penggunaan teknologi sistem informasi menjadi sangat berguna bagi perusahaan yang membutuhkan aliran informasi yang akurat, terpercaya, cepat, relevan dan detail. Penggunaan teknologi informasi saat ini sudah berkembang pesat di segala bidang perusahaan. Proses ini mengubah sistem yang semula masih manual menjadi terkomputerisasi [1]. Sistem informasi dalam suatu instansi dapat digunakan dalam hal apa saja, salah satunya adalah untuk mewujudkan misi dari instansi tersebut.

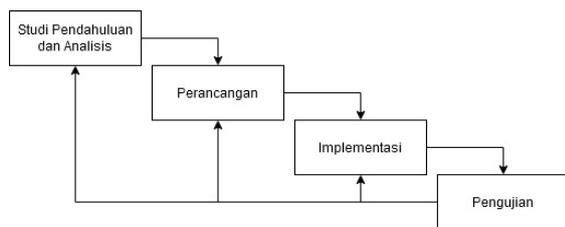
Belanegara adalah sikap dan perilaku warga negara yang dijiwai oleh kecintaan kepada Negara Kesatuan republik Indonesia (NKRI) yang berdasarkan Pancasila dan UUD 1945 dalam menjamin kelangsungan hidup bangsa dan negara [2]. Salah satu bentuk aktual secara umum adalah Pendidikan Belanegara berupa kegiatan intrakurikuler Mata Kuliah Pembentukan Kepribadian yaitu Pancasila, Pendidikan Kewarganegaraan, Bahasa Indonesia dan Widy Mwat Yasa serta kegiatan ekstrakurikuler seperti *Outbound* Belanegara. Bentuk pendidikan belanegara ini tentunya secara umum dilaksanakan tatap muka. Akan tetapi dunia dikejutkan dengan mewabahnya suatu penyakit yang di sebabkan oleh sebuah virus yang bernama corona atau dikenal dengan

covid-19 [3]. Penularan lewat kontak sosial antara manusia sulit di prediksi dan juga tidak bisa di hindarisehingga penyebarannya juga sangat pesat. Obat penawar juga belum bisa ditemukan menjadi penyebab banyaknya kematian.

Dengan masalah yang telah dijelaskan, maka proses pendidikan belanegara tidak dimungkinkan untuk dilaksanakan dalam bentuk tatap muka, dengan begitu peneliti menganggap perlu dikembangkan sebuah sistem yang dapat menjadi alternatif dari kegiatan tatap muka dari salah satu kegiatan pendidikan belanegara yaitu *outbound*. Sistem yang dibuat memanfaatkan teknologi API (*Application Programming Interface*) menggunakan arsitektur GraphQL, dikembangkan oleh *facebook* yang merupakan bahasa permintaan untuk API dan runtime untuk memenuhi permintaan tersebut dengan data yang ada [4]. API sendiri merupakan konsep sistem yang dapat saling berintegrasi dengan berbagai aplikasi antarmuka [5], dengan kemungkinan integrasi dengan berbagai macam antarmuka maka pengembangan lebih lanjut dari sistem menjadi lebih terbuka.

I. Metodologi

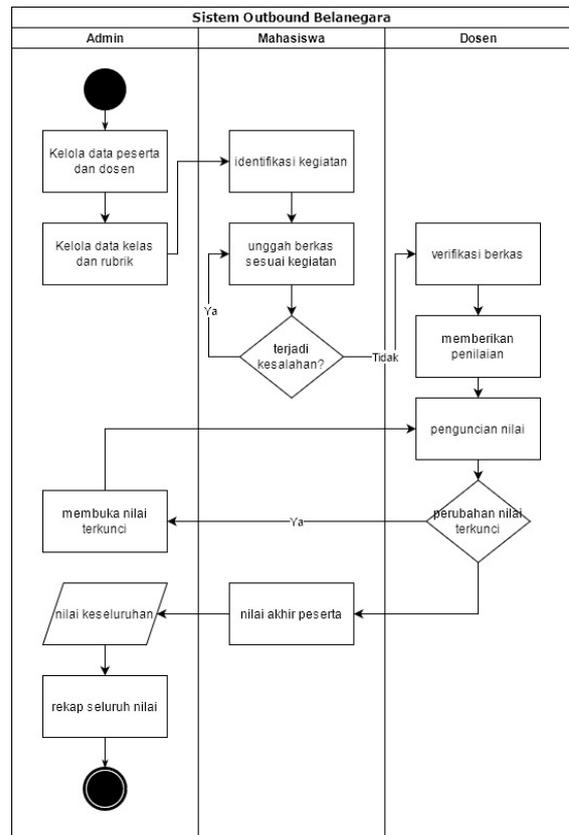
Metode pengembangan yang digunakan dalam sistem adalah SDLC (System Development Life Cycle) *waterfall* [6]. Merupakan siklus yang setiap tahapannya memiliki spesifikasi tersendiri, sehingga suatu sistem dapat dikembangkan dengan mudah sesuai keinginan. Pada pengembangan sistem menggunakan *waterfall* terdapat empat alur tahapan yaitu analisis dan studi kasus, perancang, implementasi, dan pengujian dengan kemungkinan dapat berbalik ke tahapan sebelumnya jika ada masalah yang signifikan. Alur metode *waterfall* dapat dilihat pada Gambar 1.



Gambar 1. Metodologi Penelitian

Analisis Sistem

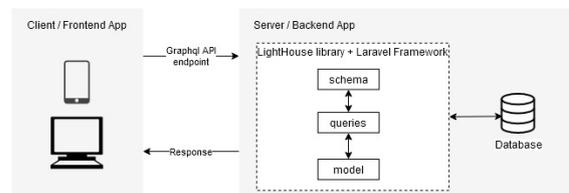
Tahapan ini diperlukan untuk mendapatkan informasi kebutuhan sistem yang diperlukan sebelum perancangan dilakukan. Analisis yang dilakukan berkaitan tentang proses kegiatan *outbound* belanegara yang sebelumnya telah dilakukan secara luring menjadi secara daring. Proses analisis diperoleh dari hasil wawancara dengan pihak Lembaga Pengembangan Pembelajaran dan Penjaminan Mutu (LP3M). Hasil analisis yang dikumpulkan diubah menjadi suatu workflow sistem yang menjelaskan alur tahapan sistem yang dapat dilihat pada Gambar 2.



Gambar 2. Workflow Sistem *Outbound* Belanegara

Arsitektur Sistem

Sistem yang menggunakan konsep API dimana terdapat dua tipe yaitu APICall dan APIClient [7]. Kedua tipe tersebut menerapkan suatu proses yang biasa disebut *request* dan *response* antara *client* dan *server* dalam pertukaran data.



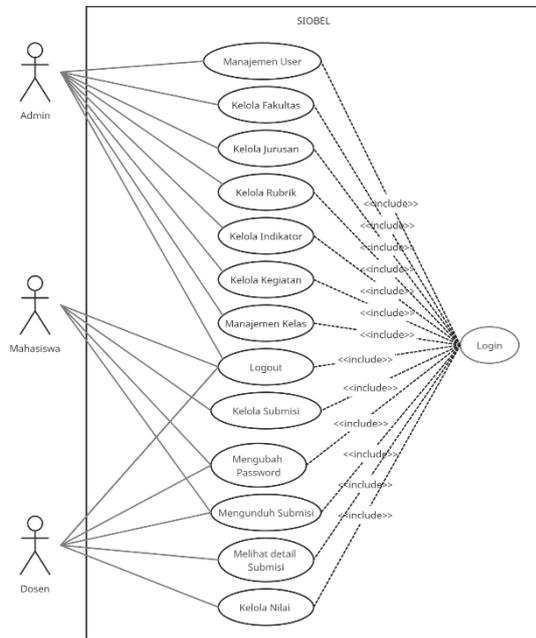
Gambar 3. Arsitektur Sistem

Pada gambar 3 menjelaskan proses pertukaran data dari *client* ke *server* dan sebaliknya. Response yang dikirim dari *client* diterima oleh *server* dan diproses berdasarkan *schema*, *queries*, dan *model* pada *graphql* untuk mendapatkan data yang sesuai dari *database* dan dikembalikan lagi ke *client* sebagai *response*.

Use Case Diagram

Perancangan *use case diagram* dirancang untuk menggambarkan aksi apa saja yang bisa dilakukan oleh pengguna didalam sistem menggunakan model fungsional dengan aktor

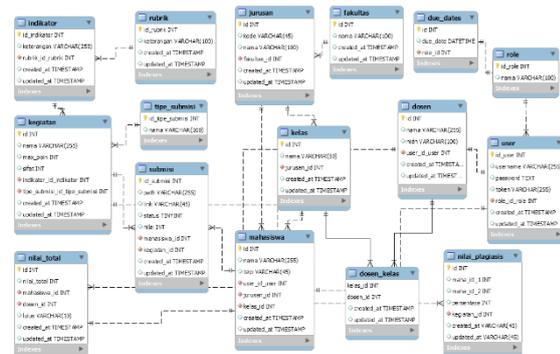
dan *use case* [8]. Pengguna atau aktor yang ada pada sistem *outbound* belanegara meliputi admin, dosen, dan mahasiswa. Setiap aktor memiliki hak dan perannya masing-masing dan tidak dapat melakukan aksi yang bukan haknya. Peran masing-masing aktor dapat dilihat pada gambar 4.



Gambar 4. Use Case Sistem

Basisdata

Setiap sistem daring memerlukan suatu tempat penyimpanan data atau yang disebut basisdata. Basis data diartikan sebagai markas atau gedung, sedangkan data merupakan representasi barang didalamnya [9]. Struktur basisdata yang digunakan pada sistem *outbound* belanegara bertipe SQL (Structured Query Language) dengan DBMS (Database Management System) diterapkan pada MySQL.



Gambar 5. Basisdata Sistem

Pada gambar 5 menunjukkan keseluruhan tabel yang digunakan pada

sistem. Elemen repetitif seperti id merupakan kolom wajib sebagai kunci pada setiap tabel, sedangkan elemen *created_at* dan *updated_at* merupakan kolom yang dibuat secara otomatis oleh *framework* pada sistem.

II. Hasil dan Pembahasan

Setelah melalui semua proses pengembangan sistem, sehingga terbentuklah suatu API berbasis GraphQL.

Skema GraphQL

Skema pada GraphQL terdiri dari beberapa bagian yaitu objek, *mutation*, dan *query*. Objek mendefinisikan entitas yang ada pada basisdata untuk mendapatkan data yang sesuai. Contoh objek dapat dilihat pada gambar 6.

```
type Submisi {
  id: ID!
  path: String
  link: String
  status: Int
  nilai: Int
  plagiasi: Int
  created_at: String
  updated_at: String

  mahasiswa: Mahasiswa @belongsTo
  kegiatan: Kegiatan @belongsTo
}
```

Gambar 6. GraphQL Objek

Mutation mendefinisikan fungsi yang mengatur proses keluar masuk data sesuai dengan alur sistem yang telah dirancang juga mengatur hak aksi yang dimiliki oleh aktor. Contoh *mutation* dapat dilihat pada gambar 7.

```
extend type Mutation @middleware(checks: ["App\\Http\\Middleware\\ValidateToken", "App\\Http\\Middleware\\ValidateDosen"]){
  updateNilai(
    submisi_id: Int!
    kelas_id: Int!
    dosen_id: Int!
    nilai: Int!
  ): Boolean @field(resolver: "NilaiMutator@updateNilai" method: "post")
  kunciNilai(
    mahasiswa_id: Int!
    kelas_id: Int!
    dosen_id: Int!
  ): NilaiTotal @field(resolver: "NilaiMutator@kunciNilai" method: "post")
  kunciKelas(
    kelas_id: Int!
    dosen_id: Int!
  ): Boolean @field(resolver: "NilaiMutator@kunciKelas")
}
```

Gambar 7. GraphQL Mutation

Query mendefinisikan request yang akan dilakukan oleh sisi *client*. *Query* didefinisikan berdasar objek yang ada dengan pemberian opsi

pengambilan data penuh atau sebagian dan limitasi data response. Contoh *mutation* dapat dilihat pada gambar 8.

```

type Query {
  # batch result
  users(limit: Int!): [User] @all
  roles: [Role] @all
  dosens(limit: Int!): [Dosen] @all
  mahasiswa(limit: Int!): [Mahasiswa] @all
  submisis(limit: Int!): [Submisi] @all
}
    
```

Gambar 8. GraphQL Query

Implementasi API

Hasil dari skema GraphQL yang dirancang adalah sebuah skema API, *mutation* dan *query* yang telah dibuat dapat dilakukan uji coba dengan beberapa implementasinya sebagai berikut.

Gambar 9. Mutation Login

Pada gambar 9 merupakan API yang digunakan untuk melakukan otentikasi pengguna didalam sistem. Pada *mutation* ini informasi yang dikirimkan merupakan data akun pengguna dengan kembalian data berupa data relasi yang ada pada akun pengguna tersebut.

Gambar 10. Query User

Pada gambar 10 merupakan API yang digunakan untuk mendapatkan data pengguna. Fungsi API ini hanya dapat diakses oleh pengguna dengan jabatan admin pada sistem dengan alasan untuk keamanan data pengguna lain. Pada *query* ini informasi yang dikirimkan berupa id unik dari pengguna dengan data kembalian berupa data detail dari user tersebut beserta segala relasi yang ada. Fungsi lain yang memiliki hubungan penting API ini yaitu CRUD (*create, read, update, delete*) pada *user*

dengan nama *createUser, updateUser, dan deleteUser*.

Gambar 11. Query Submisi

Pada gambar 11 merupakan API yang digunakan untuk mengambil semua atau sebagian data submisi pada sistem. API ini dapat diakses oleh pengguna dengan jabatan admin dan dosen juga pengguna dengan jabatan mahasiswa yang bersangkutan dengan submisi yang ada. Pada *query* ini data yang dikirimkan berupa atribut apa saja dari submisi yang diinginkan beserta batasan kembalian yang sesuai. Terdapat juga fungsi lain yang memiliki hubungan dengan API ini yaitu CRUD pada submisi dengan nama *createSubmisi, updateSubmisi, deleteSubmisi, dan fungsi-fungsi lain yang memiliki relasi tidak langsung dengan entitas submisi*.

Gambar 12. Query Nilai

Pada gambar 12 merupakan API yang digunakan untuk mengambil semua atau sebagian data nilai pada sistem. API ini dapat diakses oleh admin secara keseluruhan dan dapat diakses oleh dosen juga mahasiswa yang memiliki relasi dengan nilai yang bersangkutan. Pada *query* ini data yang dikirimkan berupa atribut nilai yang diinginkan dalam data kembalian yang akan diterima. Terdapat juga fungsi lain yang berhubungan dengan API ini yaitu *createNilai, dan deleteNilai*.

III. Kesimpulan

Berdasarkan penelitian sistem *outbound* belanegara yang dilakukan maka dapat disimpulkan bahwa perancangan dan pembuatan sistem *outbound* belanegara dapat menjadi alternatif bagi kegiatan yang *outbound* belanegara yang pada tahun-tahun sebelumnya dilaksanakan secara tatap muka. Dengan teknologi GraphQL yang merupakan salah satu bentuk API menjadikan sistem dapat melibatkan antarmuka dari berbagai model platform yang ada, juga dengan bentuk data yang dikembalikan lebih fleksibel dari tipe API lain membuat tidak ada data kembalian yang percuma.

IV. Daftar Pustaka

- [1] Sari, A. O., & Nuari, E. (2017). Rancang Bangun Sistem Informasi Persediaan Barang Berbasis Web Dengan Metode Fast (Framework For The Applications). *Jurnal PILAR Nusa Mandiri*, 261-266.
- [2] UPN “VETERAN” JAWA TIMUR. (den 12 July 2021). Hämtat från <https://www.upnjatim.ac.id/kampus-belanegara/>
- [3] Komalasari, K., Arafat, Y., & Mulyadi, M. (2020). Principal’s Management Competencies in Improving the Quality of Education. *Journal of Social Work and Science Education*, 181-193.
- [4] GraphQL. (den 10 July 2021). *A query language for your API*. Hämtat från <https://graphql.org/>
- [5] Lind, J., & Soames, K. (2019). *Detecting Cycles in GraphQL Schemas*. Sweden: Linköping University | Department of Computer and Information Science.
- [6] Maesaroh, S., Rohmayani, D., Ramlan, & Arsul. (2019). Rancang Bangun Sistem Informasi Kepegawaian (SIMPEG) Dengan SDLC Metode Waterfall Studi Kasus Di Kantor BKPLD Kabupaten Tasikmalaya. *Jurnal TEDC*, 197-202.
- [7] Susrama, I. G., Setiawan, A., & Kholis, M. (2019). Rancang Bangun Aplikasi “W - Mass (Weight Monitor Assistant)” Berbasis Android Studio Dengan Bahasa Native Java. *Jurnal Penelitian Politeknik Penerbangan Surabaya*, 1-19.
- [8] Ma’rufi, M. H., Diyasa, I. G., & Sugiarto. (2020). Perancangan Sistem Uji Sertifikasi Kompetensi Berbasis GraphQL. *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, 651-658.
- [9] Syahdindo, R., Amin, M. N., Floribunda, S. C., & Diyasa, I. G. (2019). Sistem Informasi Enterprise Resource Planning (ERP) Untuk Menunjang Pembayaran SPP (Studi Kasus : SMA Wachid Hasyim 4 Waru). *Teknika : Engineering and Sains Journa*, 25-30.