

PERANCANGAN SISTEM UJI SERTIFIKASI KOMPETENSI BERBASIS GRAPHQL

M Hafidz Amarul Ma'rufi¹, I Gede Susrama Mas Diyasa², Sugiarto³

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

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

Abstrak

Sertifikasi Kompetensi adalah proses pemberian sertifikat kompetensi yang dilakukan secara sistematis dan obyektif melalui ujian kompetensi yang mengacu pada skema sertifikasi yang telah dibuat oleh Lembaga Sertifikasi Profesi (LSP) dan disetujui oleh Badan Nasional Sertifikasi Profesi (BNSP). Proses yang terjadi pada sertifikasi kompetensi dimulai dengan pendaftaran sampai dengan pemberian sertifikat kompetensi kepada asesi. Pengolahan data uji sertifikasi kompetensi masih dilakukan secara manual mulai dari pendaftaran sampai dengan penyimpanan hasil uji yang belum tersistem dan rapi. Dengan penjelasan tersebut, maka diperlukan pergantian metode pengolahan data LSP dari manual menjadi metode pengolahan data LSP yang terkomputerisasi dan otomatis. Hasil dari penelitian ini yaitu Sistem Uji Sertifikasi Kompetensi berbasis API dengan gaya arsitektur *Graph Query Language* (GraphQL) yang dapat mempermudah pengembang antarmuka untuk mendefinisikan data yang dibutuhkan sesuai dengan kebutuhan sehingga akan mengefisienkan pertukaran data antara sistem dan antarmuka.

Kata kunci: Rancang, Bangun, *Application Programming Interface*, Sistem Informasi Uji Sertifikasi Kompetensi, GraphQL

1. PENDAHULUAN

Di era industri 4.0 ini, perkembangan sistem informasi menjadi hal yang sangat penting. Sistem informasi memiliki pengertian suatu susunan yang terdiri dari beberapa elemen seperti orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi dan sumber daya data yang melakukan pengolahan data. Sistem informasi banyak dikembangkan untuk menggantikan sistem dari konvensional karena proses penyimpanan informasi dapat lebih efisien jika menggunakan sistem informasi [1]. Seringkali pengembang lupa terhadap perkembangan teknologi dan permasalahan bidang tertentu dimasa datang yang berdampak pada umur sistem dengan teknologi yang digunakan. Untuk mengatasi permasalahan tersebut maka perlu dikembangkan aplikasi yang dapat terhubung dengan aplikasi lain antar platform menggunakan konsep *Application Programming Interface* (API). API adalah perangkat lunak yang mengizinkan dua aplikasi terhubung satu sama lain. API sangat penting dalam teknologi karena dapat membuat pemrograman jadi lebih mudah. [2]. Konsep terbaru dalam membangun API adalah *Graph Query Language* (GraphQL) yang dikembangkan oleh facebook. Dikutip dari *official web GraphQL*, pengertian GraphQL adalah bahasa permintaan untuk API dan runtime untuk memenuhi permintaan tersebut dengan data yang ada [3].

Lembaga Sertifikasi Profesi (LSP) adalah Lembaga pelaksana kegiatan sertifikasi kompetensi kerja yang mendapatkan lisensi dari BNSP [4]. Menurut Olivya, LSP adalah lembaga pelaksana tugas BNSP yaitu melaksanakan kegiatan sertifikasi kompetensi dan mengacu pada pedoman-pedoman yang ditetapkan BNSP. Sertifikasi Kompetensi adalah proses pemberian sertifikat kompetensi yang dilakukan secara sistematis dan obyektif melalui ujian kompetensi yang mengacu pada skema sertifikasi yang telah dibuat oleh LSP dan disetujui oleh BNSP. Proses sertifikasi kompetensi yang diselenggarakan oleh LSP terdiri dari registrasi calon peserta sampai dengan penerbitan sertifikat kompetensi. Pelaksanaan ujian kompetensi dilakukan dan dinilai oleh seorang asesor. Hasil evaluasi

dari ujian kompetensi kemudian menjadi dasar menentukan keputusan kelayakan peserta (asesi) untuk memperoleh sertifikat [5].

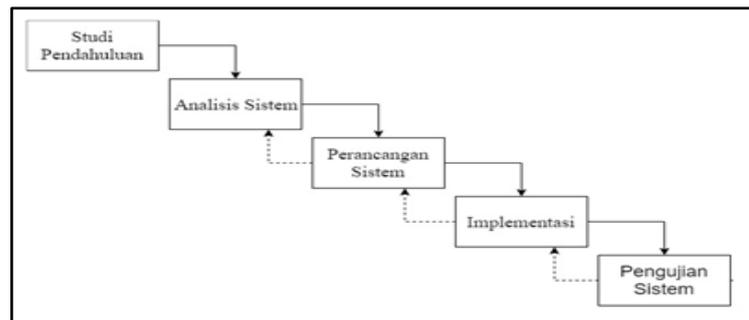
Dari proses yang dijelaskan, maka dalam mengelola data akan mengalami kesulitan untuk menyimpan maupun mencari data uji sertifikasi kompetensi jika dilakukan secara manual, begitu juga dengan proses pendaftaran jika dilakukan secara manual akan mengakibatkan kurang efektif dalam segi waktu. Untuk itu perlu adanya sistem yang dapat digunakan untuk mengelola data uji sertifikasi kompetensi dan dapat digunakan sebagai sistem untuk mendaftar uji sertifikasi secara daring.

Terdapat penelitian terdahulu yang berkaitan dengan sistem informasi administrasi lembaga sertifikasi profesi (LSP) [6]. Dalam penelitian tersebut, sistem informasi administrasi lembaga sertifikasi profesi berbasis web dengan menggunakan sistem *database* terpadu. Selanjutnya yaitu penelitian terdahulu mengenai Performance Analysis of GraphQL and RESTful in SIM LP2M of the Hasanuddin University [7]. Dalam penelitian tersebut disebutkan bahwa RESTful lebih unggul dalam waktu dan ukuran daripada GraphQL karena RESTful bersifat statis sedangkan GraphQL bersifat dinamis yang cocok untuk pengembangan sistem yang memiliki kebutuhan data yang dinamis.

Berdasarkan permasalahan diatas dan beberapa penelitian terdahulu maka dibuatlah sistem informasi yang akan menggunakan arsitektur GraphQL karena sistem yang akan dibangun memiliki data dinamis.

2. METODOLOGI

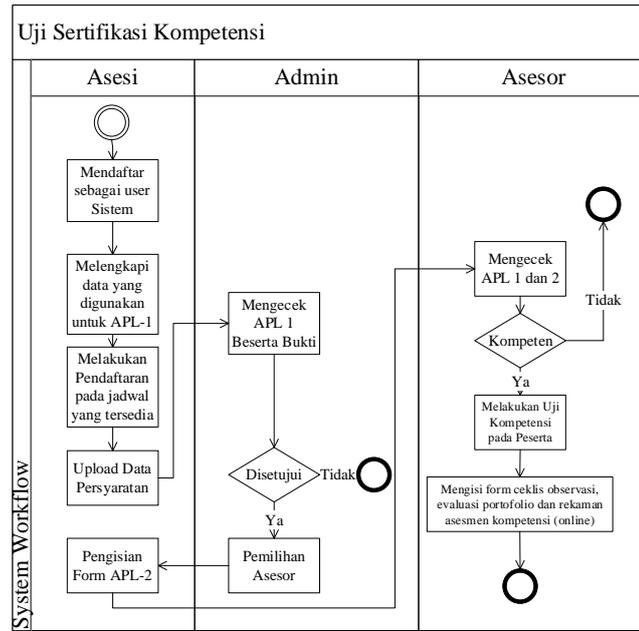
Metode yang digunakan dalam penelitian ini yaitu menggunakan metode SDLC (System Development Life Cycle) *Waterfall*. [8] SDLC merupakan siklus pengembangan sistem. Metode penelitian ini meliputi beberapa tahapan studi pendahuluan, analisis sistem, perancangan sistem, implementasi sistem dan pengujian sistem. Untuk melihat alur dari metode SDLC *Waterfall* dapat dilihat pada gambar 1.



Gambar 1. Metodologi Penelitian

2.1 Analisis Sistem

Pada tahapan ini perlu dilakukan untuk menentukan masalah dan upaya yang dapat dilakukan untuk membangun sistem atau memperbaiki sistem yang telah ada sehingga permasalahan dapat diselesaikan dengan baik. Hasil dari analisis sistem dapat diperoleh dari hasil wawancara kepada LSP. Adapun hasil wawancara tersebut mengenai sistem yang akan dibangun digambarkan pada gambar 2.

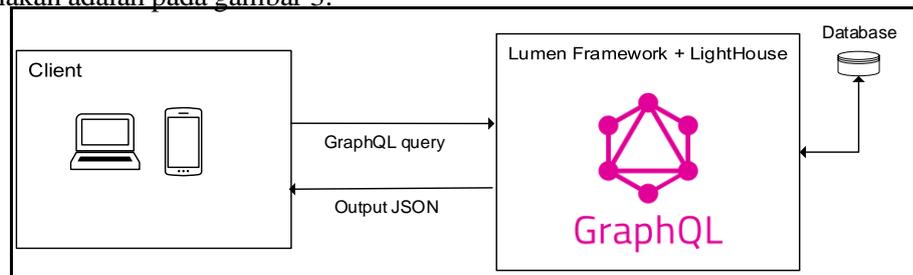


Gambar 2. System Workflow

Pada gambar 2 menggambarkan tentang alur uji sertifikasi kompetensi yang akan dibangun pada sistem uji sertifikasi kompetensi.

2.2 Perancangan Sistem

Pada tahapan ini merupakan tahap penguraian desain sistem seperti diagram *Unified Modeling Language* (UML) dan diagram basis data. UML merupakan bahasa pemodelan yang dapat menjabarkan secara rinci untuk analisa dan mencari apa yang diperlukan oleh sistem [9]. Dalam membangun sistem ini arsitektur sistem yang akan digunakan adalah pada gambar 3.



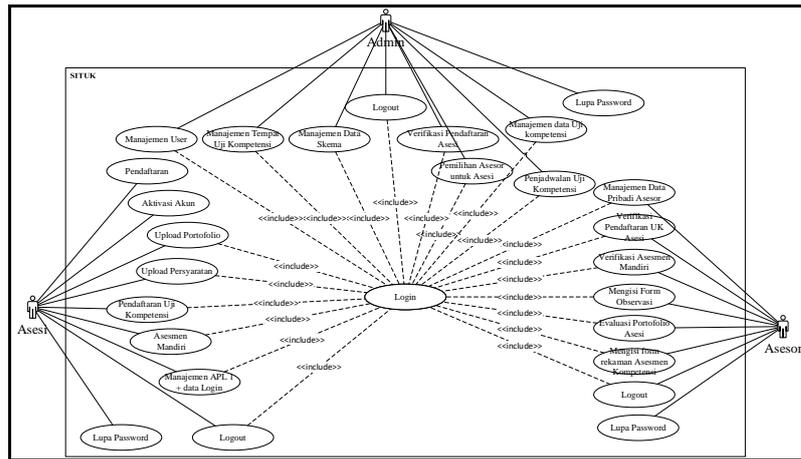
Gambar 3. Arsitektur Sistem

Pada gambar 3 menggambarkan bahwa sistem akan dibangun dengan *framework* Lumen dan *Lighthouse library* untuk dapat menggunakan GraphQL pada sistem ini. Alur pertukaran data antara *client* dan *server*. *Client* melakukan *request* kepada *server* dengan menuju end point GraphQL dan mengirim GraphQL *query* atau perintah GraphQL. Selanjutnya Lumen *framework* akan melakukan proses dan berhubungan dengan basis data dan mengirim respon data yang sesuai *request client*.

2.3 Perancangan Use Case Diagram

Use Case Diagram merupakan diagram yang menggambarkan tentang model fungsional setiap pengguna dengan menggunakan aktor dan use case. Aktor yang ada pada sistem informasi uji sertifikasi profesi meliputi admin, asesor dan asesi.

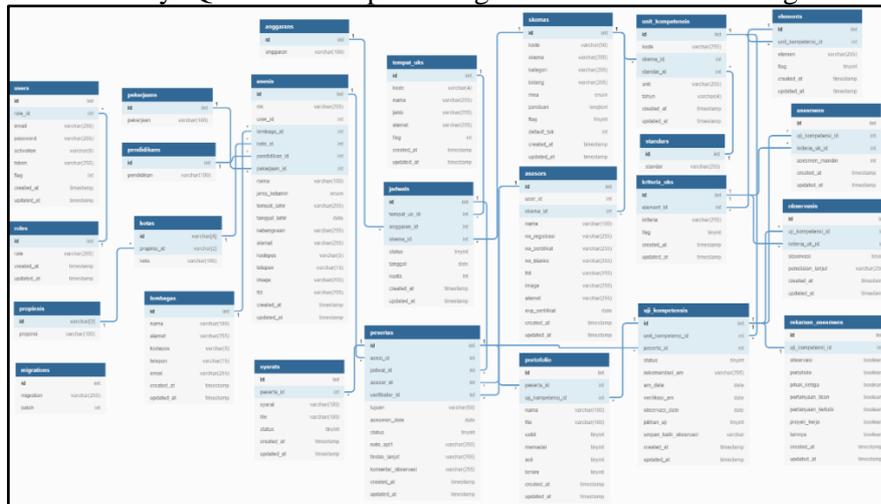
Gambar 4 menunjukkan peran masing-masing aktor pada sistem informasi uji sertifikasi kompetensi. pada *usecase* pendaftaran hanya bisa dilakukan oleh asesi. Untuk menambah data asesor harus dilakukan oleh aktor admin dengan *usecase* manajemen user.



Gambar 4. Use Case Diagram

2.4 Perancangan Basis Data

Perancangan basis data dilakukan untuk menggambarkan basis data yang akan diimplementasikan ke sistem. *Database Management System (DBMS)* yang akan digunakan adalah MySQL. Hasil dari perancangan basis data adalah sebagai berikut.



Gambar 5. Rancangan Tabel Basis Data

Gambar 5 menunjukkan rancangan basis data yang memiliki jumlah tabel sebanyak 25 tabel. Pada beberapa entitas memiliki kolom *id*, *created_at* dan *updated_at* karena menggunakan migration yang ada pada *Lumen framework*.

3. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas perancangan sistem lebih rinci serta implementasi dari sistem yang dirancang.

3.1 Implementasi GraphQL Schema

Dalam *GraphQL Schema* dibedakan menjadi 3 bagian yaitu *Type* objek, *mutation* dan *query*. *Type* objek seperti halnya class dalam pemrograman yaitu mendefinisikan objek kelas, mutaiton digunakan untuk *request* yang melibatkan perubahan dalam basis data dan query digunakan untuk *request* untuk mengambil data dari basis data.

Type objek yang terdapat pada sistem ini berjumlah 24 type objek. *Type* objek tersebut yaitu Asesi, Asesor, User, Role, Kota, Propinsi, Pekerjaan, Pendidikan, Lembaga, Anggaran, Peserta, Syarat, Jadwal, TempatUk, Skema, Portofolio, UjiKompetensi, UnitKompetensi, Standar, Element, Asesmen, Observasi dan RekamanAsesmen contoh implementasi *type* Objek ada pada gambar 6(a).

```

type Skema {
  id : ID!
  kode : String!
  skema : String!
  kategori : String!
  bidang : String!
  mea : String!
  panduan : String!
  flag : Int!
  default_tuk : Int!
  created_at : String!
  updated_at : String!

  unitKompetensi : [UnitKompetensi] @hasMany
  jadwal : [Jadwal] @hasMany
  asesor : [Asesor] @hasMany
  tempatUk : TempatUk @belongsTo
}

extend type Mutation @middleware(checks: [CheckToken]) {
  createSkema(
    kode : String!
    skema : String!
    kategori : String!
    bidang : String!
    mea : String!
    panduan : String!
    default_tuk : ID!
  ) : Skema @middleware(checks: [CheckToken,
  CheckAdmin]) @field(resolver:
  "App\Http\GraphQL\Mutations\SkemaMutations@create"
  method: "post")
}
    
```

Gambar 6. (a) Implementasi GraphQL Type; (b) Implementasi GraphQL Mutation

Mutation yang terdapat pada sistem informasi uji sertifikasi kompetensi berjumlah 52 mutation yang meliputi peran yang terdapat usecase para aktor yang terdapat pada sistem informasi uji kompetensi. Untuk dapat menggunakan mutation pada query GraphQL harus diawali dengan perintah mutation. Mutation membutuhkan parameter yang akan dikirim ke sistem untuk diproses dan menerima respons sesuai dengan attribute. Contoh dari implementasi GraphQL *mutation* ada pada gambar 6(b).

```

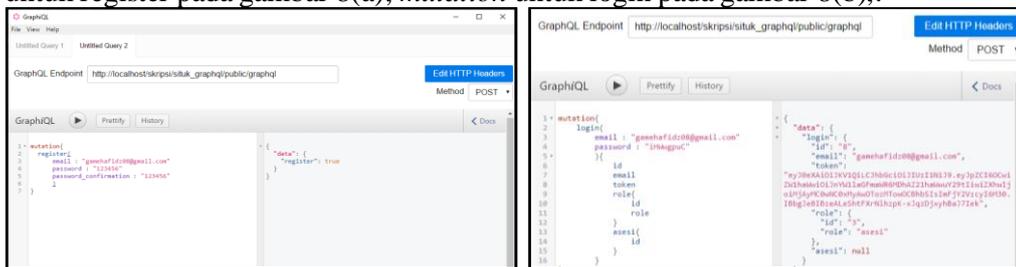
extend type Query @middleware(checks:
[CheckToken]) {
  skemas : [Skema] @all
  skema(id: ID! @eq) : Skema @find
}
    
```

Gambar 7. Implementasi GraphQL Query

Pada gambar 7 adalah contoh penerapan GraphQL *Query*. *Query* yang terdapat pada sistem informasi uji sertifikasi berjumlah 44 query. Penulisan query didasarkan pada *type* objek yang diminta. Seperti halnya Skemas digunakan untuk mengambil data *type* objek Skema dengan jumlah seluruh data Skema. Sedangkan query Skema yang memiliki parameter id digunakan untuk mengambil *type* objek Skema dengan id yang dipilih.

3.2 Hasil dan Uji Coba

Dalam melakukan uji coba hasil dari implementasi GraphQL ini akan dilakukan uji coba terhadap beberapa *mutation* dan *query* yang telah dibuat. Tahap awal untuk menggunakan sistem ini adalah otentikasi. Hasil implementasi otentikasi meliputi *mutation* untuk register pada gambar 8(a), *mutation* untuk login pada gambar 8(b).



Gambar 8. (a) Mutation register; (b) Mutation login

Sedangkan *mutation* untuk proses otentikasi lainnya yaitu *mutation* untuk aktivasi akun user yang telah melakukan registrasi pada gambar 9(a), *mutation* untuk mengganti password pada gambar 9(b).

Gambar 12. (a) Mutation asesmenMandiri; (b) Mutation observasi;

Untuk hasil dari *mutation* untuk mengisi form evaluasi portofolio adalah pada gambar 13(a) dan hasil dari *mutation* untuk mengisi form rekaman asesmen uji kompetensi adalah pada gambar 13(b).

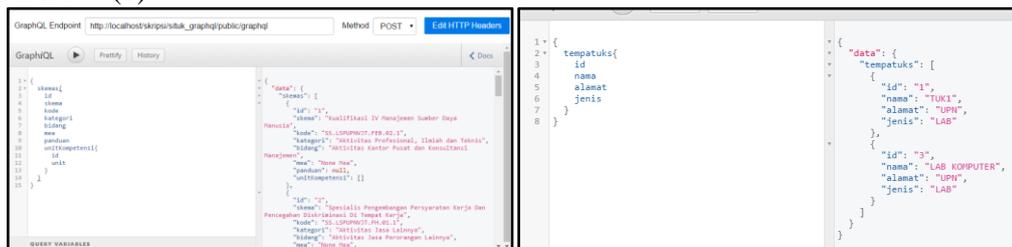


(a)

(b)

Gambar 13. (a) Mutation evaluasiPortofolio; (b) Mutation rekamanAsesmenKompetensi

Untuk hasil uji coba GraphQL *Query*. Contoh yang akan digunakan adalah mengambil data skema pada gambar 14(a) dan mengambil data tempat uji kompetensi pada gambar 14(b).



(a)

(b)

Gambar 14. (a) Query skemas untuk mengambil semua data skema; (b) Query tempatuks untuk mengambil semua data tempat uji kompetensi

Dari beberapa contoh uji coba yang dilakukan cukup untuk mewakili ujicoba dari 54 *mutation* dan 44 *query* yang telah dibangun. Hasil dari seluruh uji coba yang dilakukan dapat diampil keputusan bahwa seluruh *mutation* dan *query* yang ada telah sesuai dengan yang diinginkan dan dapat digunakan untuk membangun sistem *front-end* lainnya.

4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang dilakukan, maka dapat ditarik kesimpulan bahwa dengan perancangan sistem informasi uji sertifikasi kompetensi dapat mempermudah dalam pengelolaan data yang berkaitan dengan uji sertifikasi kompetensi dan dapat membantu proses uji sertifikasi kompetensi dari awal proses pendaftaran hingga akhir uji sertifikasi kompetensi. Hak akses yang ada pada sistem meliputi asesi, asesor dan admin. Sistem ini cocok untuk dibangun dengan menggunakan API dengan GraphQL. Sehingga sistem ini dapat dibangun *front-end* yang bersifat *multiplatform*. Berdasarkan hasil implementasi dan perancangan yang telah dilakukan, terdapat beberapa saran yang perlu diperhatikan dalam mengembangkan sistem ini supaya sistem informasi uji sertifikasi dapat dimanfaatkan oleh pengguna, maka sistem ini harus dibangun aplikasi *front-end* yang dapat berupa aplikasi web dan aplikasi mobile. Untuk itu dibutuhkan pengembang selanjutnya yang dapat membangun aplikasi tersebut.

5. DAFTAR RUJUKAN

[1] Firman, A., Wowor, H. F., & Najoran, X. (2016). Sistem Informasi Perpustakaan Online Berbasis Web . *E-journal Teknik Elektro dan Komputer* , 29-36.

[2] Apa itu API (2019, March 26). Retrieved June 01, 2020, from <https://netmonk.id/apa-itu-api/>

- [3] GraphQL: A query language for APIs. (n.d.). Retrieved June 01, 2020, from <https://graphql.org/>
- [4] Peraturan Badan Nasional Sertifikasi Profesi, 2 (Tentang Pembentukan Lembaga Sertifikasi Profesi Agustus 2014).
- [5] Olivya, M. (2017). Perancangan Aplikasi Evaluasi Ujian Sertifikasi Kompetensi pada Lembaga Sertifikasi Profesi. *Prosiding Seminar Hasil Penelitian (SNP2M)* (pp. 123-128). Makasar: Politeknik Negeri Ujung Pandang.
- [6] Perdana Putra, F. H. (2017). Sistem Informasi Administrasi Lembaga Sertifikasi Profesi STIKI Malang untuk Pengelolaan Sertifikasi TIK. *Journal of Information and Technology (J-INTECH)*, 71-77.
- [7] Hartina, D. A., Lawi, A., & Panggabean, B. L. (2018). Performance Analysis of GraphQL and RESTful in SIM LP2M of the Hasanuddin University. The 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT) 2018, (pp. 237-240).
- [8] Firmansyah, Y., & Udi. (2018). Penerapan Metode SDLC Waterfall Dalam Pembuatan Sistem Informasi Akademik Berbasis Web Studi Kasus Pondok Pesantren Al-Habi Sholeh Kabupaten Kubu Raya, Kalimantan Barat. *Jurnal Teknologi & Manajemen Informatika*
- [9] Suendri, "Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem," *ALGORITMA: Jurnal Ilmu Komputer dan Informatika Vol.3*, pp. 1-9, 2018.