

# **TUGAS AKHIR I**

## **JUDUL TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat mengerjakan dan menempuh ujian  
tugas akhir 2



Disusun Oleh:

Nama : Arnaz Adiputra  
NIM : A11.2014.08602

Program Studi : Teknik Informatika-S1

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS DIAN NUSWANTORO  
SEMARANG  
20...**

## **PERSETUJUAN SKRIPSI**

Nama : Arnaz Adiputra

NIM : A11.2014.08602

Program Studi : Teknik Informatika

Fakultas : Ilmu Komputer

Judul Tugas Akhir : .....

Tugas Akhir ini telah diperiksa dan disetujui,

Semarang, .....

<p>Menyetujui: Pembimbing</p> <p><b>Nama Dosen Pembimbing</b></p>	<p>Mengetahui: Dekan Fakultas Ilmu Komputer</p> <p><b>Dr. Abdul Syukur</b></p>
---	--



## **PENGESAHAN DEWAN PENGUJI**

Nama : Arnaz Adiputra

NIM : A11.2014.08602

Program Studi : Teknik Informatika

Fakultas : Ilmu Komputer

Judul Tugas Akhir : .....

Tugas akhir ini telah diujikan dan dipertahankan dihadapan Dewan Penguji pada Sidang tugas akhir tanggal ..... Menurut pandangan kami, tugas akhir ini memadai dari segi kualitas maupun kuantitas untuk tujuan penganugrahan gelar Sarjana Komputer (S.Kom.)

Semarang, .....

Dewan Penguji:

**Nama Dosen Penguji 1**

Anggota

**Nama Dosen Penguji 2**

Anggota

**Nama Ketua Dosen Penguji**

Ketua Penguji

## **HALAMAN RINGKASAN**

## DAFTAR ISI

PERSETUJUAN SKRIPSI.....	ii
PENGESAHAN DEWAN PENGUJI.....	iii
HALAMAN RINGKASAN.....	iv
DAFTAR ISI.....	v
DAFTAR TABEL.....	vi
DAFTAR GAMBAR.....	vii
DAFTAR LAMPIRAN.....	viii
ARTI LAMBANG, SINGKATAN, DAN ISTILAH.....	ix
BAB I PENDAHULUAN.....	1
1.1Latar Belakang.....	1
1.2Rumusan Masalah.....	3
1.3Batasan masalah.....	3
1.4Tujuan Penelitian.....	3
1.5Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	5
2.1Tinjauan Studi.....	5
2.2Tinjauan Pustaka.....	9
2.3Kerangka Pemikiran.....	14
DAFTAR PUSTAKA.....	16



## DAFTAR TABEL

Tabel 2.1: Tinjauan Pustaka.....	7
Tabel 2.2: Kerangka Pemikiran.....	14

## DAFTAR GAMBAR

Gambar 2:1: Client-Server Model.....	9
Gambar 2:2: Contoh syntax HTML.....	10
Gambar 2:3: Contoh syntax JSON.....	12

## **DAFTAR LAMPIRAN**

## **ARTI LAMBANG, SINGKATAN, DAN ISTILAH**

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Penerimaan mahasiswa baru Universitas Dian Nuswantoro merupakan gerbang awal bagi calon mahasiswa untuk dapat berkuliah di Universitas Dian Nuswantoro. Universitas Dian Nuswantoro membuka tiga jenis jalur pendaftaran salah satunya ialah jalur reguler. Jalur Reguler adalah jalur penerimaan mahasiswa baru untuk jenjang strata satu (S-1), Program Vokasi (D3), dan Diploma 4 melalui tes potensi akademik yang dilaksanakan secara mandiri atau kolektif. Pendaftaran Jalur Reguler dapat dilakukan secara langsung di sekretariat Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro atau online[1].

Jalur reguler memiliki dua jenis prosedur, salah satunya pendaftaran online berbasis web. Untuk menggunakan pendaftaran online berbasis web calon mahasiswa diwajibkan untuk membaca tata tertib pendaftaran online yang sudah disediakan, mengunjungi website resmi Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro, mengisi formulir pendaftaran, mentranfer biaya pendaftaran di rekening yang sudah ditentukan, mengunggah bukti pembayaran, menunggu jadwal ujian, melakukan ujian dan keluar hasil ujian.

Jalur Reguler dengan prosedur pendaftaran online berbasis web merupakan bentuk perkembangan teknologi yang dimiliki Sekretariat Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro. Perkembangan teknologi di dunia juga diimbangi peningkatan pengguna internet di tiap tahunnya. Peningkatan tersebut dibuktikan oleh “The Statistics Portal”, ditahun terakhir kenaikan mencapai sembilan belas juta pengguna internet[2].

Data yang dipublikasi oleh “We Are Social” dan “Hootsuite” pada tahun 2017 yang penulis kutip dari Kadata, Indonesia merupakan negara terbesar nomer satu didunia dalam hal pertumbuhan penggunaan internet sebesar 51% dalam kurun waktu satu tahun[3]. Hal ini juga diimbangi dengan peningkatan pencarian di google dengan perangkat *mobile*, dari tahun 2013 sampai tahun 2014 kenaikan mencapai 1,8%, kenaikan ini akan bertambah tiap tahunnya[4].

Dengan meningkatnya penggunaan perangkat *mobile* untuk akses internet yang begitu pesat, maka hal tersebut menjadi penting untuk Sekretariat Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro untuk mulai mengembangkan aplikasi berbasis *mobile apps* agar dapat memudahkan calon mahasiswa untuk melakukan pendaftaran dengan jalur online dengan menggunakan perangkat *mobile*.

Pembangunan *mobile apps* memiliki beberapa komponen utama, salah satunya pembuatan *Restful Web Service*. *Restful Web Service* merupakan sebuah *service* yang digunakan untuk menghubungkan aplikasi *client* dengan server[5]. Untuk sekarang *restful* merupakan solusi terbaik untuk pembangunan sebuah API yang nantinya akan menjadi jembatan bagi aplikasi *client* dengan server, *syntax-nya* yang simple, *request* dan *response* yang mudah dibuat dan di parsing membuat *restful* menjadi populer dan digunakan dibanyak project[6]. Terlepas dari kelebihanannya *resful* juga memiliki beberapa kekurangan.

Kekurang *restful* adalah dalam project sekala besar memerlukan banyak *endpoint* dalam satu halaman. Hal ini ditinjau dari rata-rata untuk melakukan *HTTP request* membutuhkan hampir 1 detik, sedangkan satu halam dari aplikasi *client* membutuhkan lebih dari satu *endpoint*[7]. Disisi yang sama, kecepatan akses internet dengan perangkat *mobile* lebih lambat dari WiFi karena *latency* tertinggi yang dimiliki 3G mencapai 3500ms dan 4G mencapai 600ms[8] . Hal lain yang menjadi penting khususnya pengguna

aplikasi mobile ialah kecepatan akses internet di Indonesia masih jauh tertinggal dari negara lain. Dari survei yang dilakukan Akamai Technologies pada kuartal IV 2016, dari 15 negara Indonesia berada diperingkat 12 dengan rata rata kecepatan 6.7Mbps[9]. Dari data-data diatas terbukti *restful* kurang bagus untuk diimplementasikan di dalam sistem Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro.

Dari sekian kekurangan yang dimiliki *Restful* penulis menawarkan teknologi baru yaitu *GraphQL*. *GraphQL* merupakan sebuah *service* yang digunakan untuk menghubungkan aplikasi client ke server dengan konsep baru[10], dimana pengembang aplikasi client dapat meminta *response* dari *service GraphQL* tanpa atribut yang tidak dibutuhkan dalam *request*-nya. Dengan *GraphQL* pula pengembang bisa menghemat endpoint yang dibutuhkan. Teknologi ini dikembangkan oleh Facebook pada tahun 2015 dan sudah digunakan oleh beberapa perusahaan besar seperti Facebook, Github, Pinterest dan masih banyak lagi.

Dengan masalah diatas dan fakta yang penulis paparkan, maka penulis tertarik untuk melakukan penelitian mengenai “Implementasi GraphQL pada Sistem Penerimaan mahasiswa Baru Universitas Dian Nuswantoro berbasis Web Service”. Untuk membangun *Backend API* untuk sistem Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro yang memiliki sedikit endpoint . Sehingga diharapkan dapat membantu kinerja dari Sekretariat Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro.

## 1.2 Rumusan Masalah

Berdasarkan apa yang sudah penulis jabarkan dilatar belakang, maka rumusan dalam masalah ini adalah bagaimana membangun Backend API untuk aplikasi mobile Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro dengan minimum *endpoint*.

### 1.3 Batasan Masalah

Agar penulisan penelitian ini sesuai dengan tujuan awal, maka perlu diberikannya batasan masalah sebagai berikut:

- a. Penulis menggunakan *GraphQL* yang digunakan sebagai metode *Web Service* dengan hasil berupa *JSON*.
- b. Aplikasi yang penulis kembangkan dalam penelitian ini berbasis *Backend API* untuk aplikasi mobile.
- c. *Backend API* meliputi pendaftaran, pendaftaran ulang, dan laporan statistik pada sistem Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro.

### 1.4 Tujuan Penelitian

Dari rumusan masalah yang di jelaskan sebelumnya, tujuan dari kegiatan penelitian ini adalah membangun *Backend API* untuk aplikasi mobile Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro dimana hemat penggunaan endpoint sehingga bisa mengurangi *latency* pada saat aplikasi mobile melakukan *request* ke *Backend API* yang penulis buat.

### 1.5 Manfaat Penelitian

#### 1.5.1 Bagi Penulis

- a. Penulis dapat mengimplentasikan pengetahuan yang didapat selama masa kuliah.
- b. Menambah wawasan penulis terkait implementasi *GraphQL* yang penulis dapatkan semasa kuliah.
- c. Memenuhi salah satu persyaratan kelulusan starta satu (S1), Program Studi Teknik Informatika, Fakutlas Ilmu Komputer, Universitas Dian Nuswantoro.



#### 1.5.2 Bagi Universitas Dian Nuswantoro

- d Menadapatkan alternatif solusi dalam sistem penerimaan mahasiswa baru berbasis *Web Service*.
- e Dapat dijadikan tambahan informasi dan rekomendasi topik penelitian kepada mahasiswa dengan minta yang sama.

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Studi

Pada tahun 2016 Facebook mengumumkan sebuah teknologi untuk melakukan request data ke Web API dengan konsep berbeda dibanding REST dengan nama GraphQL. Penelitian tentang “An Initial Analysis of Facebook’s GraphQL Language” yang diangkat oleh Olaf Hartig dan Jorge Pérez, merupakan sebuah penelitian awal dari GraphQL[10]. Konsep dari GraphQL sendiri ialah GraphQL tidak mendefinisikan model data secara langsung, melainkan secara implisit menganggap data model yang diimplementasikan berupa tampilan berbasis graph dari beberapa database yang mendasarinya. Kueri yang digunakan GraphQL menyerupai bentuk dari JSON(Javascript Object Notation). Penelitian ini menghasilkan sebuah teknologi baru untuk mengakses data pada Web API dengan konsep Graph.

Kit Gustavsson dan Erik Stenlund pada penelitian “Efficient Data Communication Between a Web Client and a Cloud Environment” menjelaskan perbedaan arsitektur antara REST yang selama ini menjadi standar pembangunan sebuah web API dengan graphql yang baru baru ini muncul, penelitian ini tidak membahas mana yang lebih bagus dan mana yang lebih buruk, tetapi pembahasan lebih ke perbedaan teknik dari keduanya[11]. Penelitian ini memiliki dua tujuan utama, yang pertama untuk meneliti dan menunjukan perbedaan antara *REST* dan *GraphQL*. Kedua, berdasarkan dari hasil penelitian, harus terlebih dahulu membuat model keputusan untuk menentukan jenis teknik mana yang akan digunakan dan pengaruh saat menggunakan teknik tersebut dalam pengembangan *Web API*. Efek yang akan didapat ketika pihak pengembang menggunakan GraphQL ialah, pengembang harus bergantung dengan *dependencies* GraphQL

sedangkan *dependencies* tersebut merupakan *external dependencies*. Dan jika pengembang lebih memilih menggunakan REST API, pengembang perlu menentukan terlebih dahulu bahasa yang akan digunakan tetapi tidak perlu bergantung pada *external dependencies*. Dari sisi lain para pengembang juga harus mempertimbangkan segi performa, yang mana dari hasil penelitian ini menunjukan GraphQL dapat mengurangi beban dari server maupun *client*.

Penelitian “Implementing GraphQL as a Query Language for Deductive Databases in SWI-Prolog Using DCGs, Quasi Quotations, and Dicts” yang ditulis Mike Bryant membahas tentang apa GraphQL tersebut, perbandingan antara GraphQL dengan REST dan implementasi GraphQL pada SWI-Prolog yang memiliki Deductive Databases[12]. GraphQL merupakan sebuah *application layer* yang digunakan untuk query data dan manipulasi yang dikembangkan oleh Facebook. GraphQL ini juga dapat memproses data dari berbagai sumber database, contohnya menggabungkan data dari database relasional dengan database NoSql. Berbeda dengan REST, GraphQL hanya menyediakan sebuah *endpoint* yang fleksibel dan dapat dilakukan proses query pada *endpoint* tersebut. Penelitian ini telah menghasilkan SWI-Prolog Versi 7 atau yang disebut *GraphQL.pl* yang telah menggunakan server GraphQL.

EHRI atau The European Holocaust Research Infrastructure merupakan sebuah lembaga yang mendukung para peneliti Holocaust dalam bentuk infrastruktur digital. EHRI menyediakan akses data online mengenai informasi Holocaust tersebut melalui portal online, dan semua hal yang dapat membantu para peneliti Holocaust [13]. Penelitian yang diangkat oleh Mike Bryant berjudul “GraphQL for Archival Metadata: An Overview of the EHRI GraphQL API” membahas tentang penerapan GraphQL pada server EHRI[14]. Database yang digunakan EHRI merupakan Graph database Neo4j-based, hal ini memberikan banyak keuntungan ketika

mengimplementasikan GraphQL kedalam Graph database, beberapa keuntungannya ialah karena kedua teknologi tersebut sama sama memiliki konsep graph dan kueri dari GraphQL mewarisi dari kueri Neo4j-based. Tujuan dari penelitian ini ialah untuk mempermudah para peneliti dalam proses ekstrak data untuk tujuan penelitian.

*Tabel 2.1: Tinjauan Pustaka*

<b>No.</b>	<b>Nama</b>	<b>Tahun</b>	<b>Judul</b>	<b>Metode</b>	<b>Hasil</b>
1.	Olaf Hartig, Jorge Pérez	2016	An Initial Analysis of Facebook's GraphQL Language	Graph	Penelitian ini menghasilkan sebuah teknologi baru untuk mengakses data pada Web API menggunakan kueri berbentuk Graph.

2.	Kit Gustavsson, Erik Stenlundngo mpolan ,	2016	Efficient Data Communication Between a Webclient and a Cloud Environment	Represent ational State Transfer(R EST), GraphQL	Pada penelitian ini menghasilkan, jika pengembang lebih memilih menggunakan GraphQL, pengembang harus bergantung pada <i>external dependencies</i> dengan pertimbangan performa yang lebih cepat ketimbang REST API. Tetapi jika pengembang lebih memilih menggunakan REST API pengembang bisa tidak bergantung pada <i>external dependencies</i> .
----	---	------	---	---	--

3.	Falco Nogatz, Dietmar Seipel	2017	Implementing GraphQL as a Query Language for Deductive Databases in SWI-Prolog Using DCGs, Quasi Quotations, and Dicts	GraphQL	Pada penelitian ini menghasilkan SWI-Prolog versi 7 atau disebut GraphQL.pl yang sudah menggunakan atau mengimplementas i GraphQL pada sistem tersebut.
----	---------------------------------------	------	--	---------	--

4.	Mike Bryant	2017	GraphQL for Archival Metadata: An Overview of the EHRI GraphQL API	GraphQL	Pada penelitian ini menghasilkan response data yang lebih mudah diekstrak bagi pengguna(dalam hal ini peneliti) dan juga API yang dihasilkan dari implementasi GraphQL menjadi lebih cepat dikarenakan konsep kueri API-nya memiliki konsep yang sama dengan sist database yang dimiliki EHRI.
----	-------------	------	--	---------	--



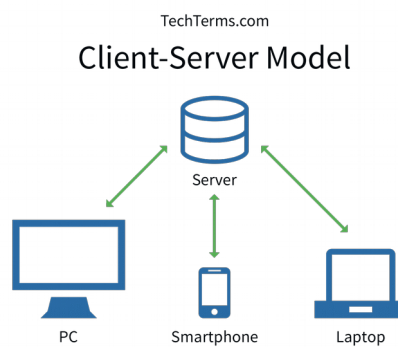
## 2.2 Tinjauan Pustaka

### 2.2.1 Interoperabilitas

Merupakan sebuah karakter dari sistem yang dapat bekerja dengan sistem lain, diakses tanpa ada batasan dari sistem tersebut[15].

### 2.2.2 Client-Server Model

Merupakan sebuah aplikasi yang terdistribusi dimana memiliki server yang bekerja sebagai penyedia layanan atau *resource* dan Client yang menggunakan layanan tersebut[16]. Secara umum relasi yang dimiliki server dengan *client* ialah one-to-many, jadi sebuah server bisa diakses banyak *client* dalam waktu yang bersamaan. Berikut gambaran dari Client-Server Model



Gambar 2:1: Client-Server Model

### 2.2.3 World Wide Web

Adalah sebutan untuk semua bagian internet yang dapat diakses menggunakan software web browser. Sebuah World Wide Web atau yang sering disebut web biasanya terdiri dari beberapa halaman web(*web page*)[17].

#### 2.2.4 HTML

Merupakan teknologi dasar untuk membangun sebuah halaman web(*web page*). HTML digunakan untuk mendefinisikan konten dari halaman web tersebut, seperti link, paragraf, gambar, heading, dan lain sebagainya[18]. Berikut merupakan contoh syntax dari HTML.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title></title>
5    </head>
6    <body>
7
8    </body>
9  </html>
10
```

Gambar 2:2: Contoh syntax HTML

#### 2.2.5 HyperText Transfer Protocol

Merupakan sebuah protokol *application layer* untuk mengirim atau menerima sebuah dokumen seperti HTML dan lain lain. HTTP digunakan untuk menyambungkan antara web browser dan web server. HTTP juga digunakan sebagai penghubung antara *client-server model*, dimana *client* meminta tanggapan(*response*) dengan menggunakan permintaan(*request*)[19].

### 2.2.6 Web Service

Merupakan salah satu bentuk Client-Server model yang termasuk ke dalam Interoperabilitas dengan melakukan komunikasi melalui World Wide Web(WWW) dan HyperText Transfer Protocol (HTTP). Web Service menyediakan sebuah layanan yang dapat diakses oleh semua platform dan kerangka kerja[20].

### 2.2.7 REST API

Merupakan sekumpulan fungsi yang mana developer dapat melakukan kegiatan request dan response[21]. Ada enam aturan dimana sebuah sistem dikatakan REST API, berikut aturan aturan tersebut[11].

- f Client-Server : Secara arsitektur REST memisahkan pemrosesan sistem menjadi dua komponen. Server merupakan komponen yang menyediakan layanan dan menanggapi permintaan untuk service tersebut. Client merupakan komponen yang terhubung ke server untuk melakukan permintaan ke server.
- g Stateless : Server tidak melihat status sesi dari Client. Setiap Request yang dikirim melalui Client harus berisi seluruh informasi yang dibutuhkan agar server dapat mengerti apa yang harus dikirim ke Client.
- h Cacheable : Response yang dikirim oleh server harus cacheable. Hal ini bertujuan untuk menghindari request yang tidak diperlukan.
- i Uniform Interface : Dengan perbedaaan komponen dari sistem REST untuk melakukan komunikasi dari kedua komponen memerlukan standar yang sama(Uniform

Interface). Hal ini juga mengurangi efisiensi dalam mengirim informasi, karena informasi yang merupakan bentuk standar sedangkan dari pihak aplikasi client memiliki kebutuhan yang berbeda.

- j Layered System : Sistem ini berada di layer yang berbeda. Satu layer hanya bisa berinteraksi dengan layer terdekatnya. Tetapi dari komponen komponen sistem tidak perlu mengerti satu sama lain, asalkan keduanya bekerja dengan baik maka komunikasi data juga akan bekerja.

#### 2.2.8 API

Ialah sekumpulan fungsi dan protokol yang digunakan untuk membangun aplikasi perangkat lunak. API berperan sebagai penerima tanggapan dari aplikasi *client* dan mengirim pesan ke aplikasi *client* tersebut[22].

#### 2.2.9 GraphQL

Merupakan sebuah bahasa query API untuk mengakses data yang ada. GraphQL menyediakan deskripsi data yang lengkap dan mudah dimengerti oleh API, dapat mengakses data persis seperti apa yang diinginkan *client*[23].

#### 2.2.10 JSON

Merupakan format pertukaran data yang ringan, mudah untuk dibaca dan di tulis oleh manusia, mudah diurai menjadi bahasa mesin. JSON berasal dari subnet bahasa pemrograman JavaScript Standard ECMA-262 3dr Edition-Desember 1999. JSON merupakan format text yang independen namun sangat familiar bagi kebanyakan bahasa pemrograman lain seperti C, C++,

C#, Java, JavaScript, Perl, Python, PHP dan masih banyak lagi. Hal ini menjadikan JSON sebagai bahasa pertukaran data yang ideal[24]. Berikut merupakan contoh dari syntax JSON

```
{
  "success": false,
  "message": "Permission denied",
  "decoded": {
    "userId": 3,
    "iat": 1514417528,
    "exp": 1514503928
  }
}
```

*Gambar 2:3: Contoh syntax JSON*

#### 2.2.11 Deductive Databases

Merupakan sebuah sistem database yang dapat mengambil sebuah kesimpulan berdasarkan aturan atau fakta yang ada. Deductive database ialah sebuah basis data relasional yang mendukung pemodelan data yang lebih kompleks[25].

#### 2.2.12 Database Relational

Merupakan sebuah sistem database yang mengatur datanya menggunakan tabel yang saling dihubungkan dengan relasi, dalam sebuah tabel terdapat baris dan kolom seperti tabel semestinya. Dari setiap baris data terdapat atribut kunci untuk menjadi pembeda dari data lain. Bentuk database ini bertujuan salah satunya untuk menghindari data yang redundan[26]. Data yang terdapat pada database ini biasa disebut record

#### 2.2.13 Database NoSql

Sistem database dimana untuk mengatur datanya tidak menggunakan tabel yang saling dihubungkan dengan relasi, sistem database ini mempermasalahkan data yang redundan dengan alasan untuk mempercepat performa dari database tersebut[27].

#### 2.2.14 SWI-Prolog

Prolog Memiliki tujuan utama ialah untuk menggabungkan antara logika pemrograman dengan kecerdasan buatan dan perhitungan ilmiah. Sedangkan SWI-Prolog merupakan sebuah perpustakaan yang menyediakan sekumpulan fungsi, logika pemrograman dalam prolog sendiri.

### 2.3 Kerangka Pemikiran

Berikut merupakan kerangka pemikiran dari penelitian ini,

*Tabel 2.2: Kerangka Pemikiran*

Problem
<ul style="list-style-type: none"> <li>• Belum adanya Backend API dari sistem Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro</li> </ul>
Approach
<ul style="list-style-type: none"> <li>• Pembuatan Backend API menggunakan GraphQL untuk sistem Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro</li> </ul>
Development
<ul style="list-style-type: none"> <li>• Sever Side: Node.js dengan menggunakan framwork express.js</li> </ul>
Implementation
<ul style="list-style-type: none"> <li>• Server Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro menyediakan informasi yang diperlukan oleh pengguna aplikasi dan juga berperan sebagai server utama untuk menyimpan data penerimaan mahasiswa baru</li> </ul>
Evaluation and Validation
<ul style="list-style-type: none"> <li>• Mengecek seluruh fitur yang ada pada server GraphQL</li> </ul>
Result
<ul style="list-style-type: none"> <li>• Backend API Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro berbasis GraphQL Web Service</li> </ul>

*a. Problem*

Sebelum melakukan penelitian ini, peneliti sudah melakukan kajian studi terlebih dahulu dengan topik terkait. Dari sinilah penulis menemukan permasalahan untuk melandasi penelitian ini. Permasalahan yang ditemukan ialah, belum adanya sistem Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro berbasis aplikasi mobile.

*b. Approach*

Dari ditemukannya permasalahan diatas, penulis mencoba untuk menemukan solusi untuk memecahkan permasalahan Backend API yang memiliki sedikit endpoint dan mengirimkan response data berupa JSON yang persis seperti aplikasi mobile butuhkan.

*c. Development*

Metode untuk memecahkan masalah sudah dibahas sebelumnya, selanjutnya akan Backend API dari sistem Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro akan diimplementasikan menggunakan bahasa pemrograman Node.js dengan *framework* Express.

*d. Implementation*

Setelah tahap *development* selesai maka menghasilkan sebuah aplikasi berupa Backend API Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro yang siap digunakan untuk aplikasi mobile dan juga data dari penerimaan mahasiswa baru akan disimpan di server Backend API tersebut.



*e. Evaluation and Validation*

Tahap selanjutnya merupakan tahap pengecekan semua *response* yang ada pada Backend API Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro.

*f. Result*

Setelah seluruh tahap terselesaikan maka hasil dari penelitian ini berupa Backend API Penerimaan Mahasiswa Baru Universitas Dian Nuswantoro yang nantinya diharapkan dapat menyelesaikan masalah yang sudah diurai diatas.

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Instrumen Penelitian**

Dalam melakukan sebuah penelitian tentu saja diperlukan berbagai macam perangkat yang digunakan, yaitu:

##### **3.1.1**

## DAFTAR PUSTAKA

- [1]..“Penerimaan Mahasiswa Baru 2018 Universitas Dian Nuswantoro.” [Online]. Available: <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>. [Accessed: 11-Dec-2017].
- [2].....“Number of internet users worldwide from 2005 to 2017 (in millions),” *Statista*. [Online]. Available: <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>. [Accessed: 11-Dec-2017].
- [3]. “Pertumbuhan Pengguna Internet, Indonesia Nomor 1 di Dunia,” Senin, Mei-2017. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2017/05/22/pertumbuhan-pengguna-internet-indonesia-nomor-1-di-dunia>. [Accessed: 11-Dec-2017].
- [4].....Greg Sterling, “Report: Mobile Search Queries 29 Percent Of Total But Growth Modest.” [Online]. Available: <https://searchengineland.com/report-mobile-search-queries-29-percent-of-total-but-growth-modest-217501>. [Accessed: 11-Dec-2017].
- [5].....“What Are RESTful Web Services?” [Online]. Available: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>. [Accessed: 11-Dec-2017].
- [6]David, “Rails 1.2: REST admiration, HTTP lovefest, and UTF-8 celebrations.” [Online]. Available: <http://weblog.rubyonrails.org/2007/1/19/rails-1-2-rest-admiration-http-lovefest-and-utf-8-celebrations/>. [Accessed: 11-Dec-2017].
- [7].....Simon E Spero, “Analysis of HTTP Performance problems.” [Online]. Available: <https://www.w3.org/Protocols/HTTP-NG/http-prob.html>.
- [8].....“Optimizing for Mobile Networks.” [Online]. Available: <https://hpbn.co/optimizing-for-mobile-networks/#anticipate-network-latency-overhead>. [Accessed: 11-Dec-2017].
- [9] “Di Asia-Pasifik, Kecepatan Internet Indonesia Jauh Tertinggal,” 27-Mar-2017. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2017/03/27/di>

asia-pasifik-kecepatan-internet-indonesia-jauh-tertinggal. [Accessed: 11-Dec-2017].

[10].....O. Hartig and J. Pérez, “An Initial Analysis of Facebook’s GraphQL Language,” in *AMW 2017 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017.*, 2017, vol. 1912.

[11].....E. Stenlund and K. Gustavsson, “Efficient data communication between a webclient and a cloud environment,” 2016.

[12]...F. Nogatz and D. Seipel, “Implementing GraphQL as a Query Language for Deductive Databases in SWI-Prolog Using DCGs, Quasi Quotations, and Dicts,” *ArXiv Prepr. ArXiv170100626*, 2017.

[13]. “EHRI Mission Statement.” [Online]. Available: <https://ehri-project.eu/about-ehri>. [Accessed: 27-Dec-2017].

[14]...Mike Bryant, “GraphQL for Archival Metadata: An Overview of the EHRI GraphQL API,” presented at the Big Data, Boston, 2017.

[15].....“Interoperability.” [Online]. Available: <https://en.wikipedia.org/wiki/Interoperability>. [Accessed: 27-Dec-2017].

[16].....“Client–server model.” [Online]. Available: [https://techterms.com/definition/client-server\\_model](https://techterms.com/definition/client-server_model). [Accessed: 26-Dec-2017].

[17].....“Apa itu World Wide Web ?” [Online]. Available: [http://faculty.petra.ac.id/dwikris/docs/desgrafisweb/www/4-apaitu\\_www.html](http://faculty.petra.ac.id/dwikris/docs/desgrafisweb/www/4-apaitu_www.html). [Accessed: 26-Dec-2017].

[18].....“HTML.” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Learn/HTML>. [Accessed: 26-Dec-2017].

[19].....“HTTP.” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP>. [Accessed: 26-Dec-2017].

[20].....“What Are Web Services?” [Online]. Available: <https://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html>. [Accessed: 26-Dec-2017].

- [21].....“Do you know what a REST API is?” [Online]. Available: <https://www.sitepoint.com/developers-rest-api/>. [Accessed: 26-Dec-2017].
- [22]..“Pengertian API (Application Programming Interface).” [Online]. Available: <http://developer.erabelajar.com/api-application-programming-interface/>. [Accessed: 26-Dec-2017].
- [23].....“GraphQL.” [Online]. Available: <http://graphql.org/>. [Accessed: 26-Dec-2017].
- [24]...“Introducing JSON.” [Online]. Available: <https://www.json.org/>. [Accessed: 26-Dec-2017].
- [25].....“Deductive Databases.” [Online]. Available: <http://www3.cs.stonybrook.edu/~warren/xsbbook/node12.html>. [Accessed: 26-Dec-2017].
- [26].....“Relational database.” [Online]. Available: [https://en.wikipedia.org/wiki/Relational\\_database](https://en.wikipedia.org/wiki/Relational_database). [Accessed: 26-Dec-2017].
- [27].....“NoSQL.” [Online]. Available: <https://en.wikipedia.org/wiki/NoSQL>. [Accessed: 26-Dec-2017].