



Teknologi *Rest Api* untuk *Backup* Data pada Sistem Pengelolaan Data Peserta BLKK AI-Wasilah Berbasis Web

Sri Rahayu¹, Dede Kurniadi², Tina Maryana³

Jurnal Algoritma
Institut Teknologi Garut
Jl. Mayor Syamsu No. 1 Jayaraga Garut 44151 Indonesia
Email : jurnal@itg.ac.id

¹sriahayu@itg.ac.id

²dede.kurniadi@itg.ac.id

³1706086@itg.ac.id

Abstrak – *Backup* data dilakukan untuk tujuan menyelamatkan data dengan cara menyalin berkas dari tempat aslinya ke tempat penyimpanan kedua sehingga salinan ini dapat digunakan untuk memulihkan data yang asli jika terjadi kerusakan data ataupun kehilangan data. Pada penelitian ini *backup* data digunakan untuk mengurangi tingkat risiko kehilangan data dalam membantu admin mencari data yang dibutuhkan kembali pada saat data tersebut sudah tidak terdapat pada server utama. Pengembangan sistem ini menggunakan *Rational Unified Process* (RUP) dengan tahapan pertama *inception*, kedua *elaboration* dan tahapan ketiga *construction* dan menggunakan *Unified Modelling Language* (UML) sebagai pemodelannya serta metode pengujian teknik *black box testing*. Teknologi yang digunakan yaitu *REST API* dimana digunakan sebagai jembatan dalam proses sinkronisasi data antara kedua server. Penelitian ini menghasilkan sebuah sistem pengelolaan data yang dilengkapi fitur *backup* data dengan mengimplementasikan teknologi *REST API*. Pada sistem ini proses *backup* data dilakukan dengan cara admin melakukan sendiri proses *backup* data pada server kedua. Dengan demikian, ketika server utama mengalami masalah, data yang telah tersimpan masih ada cadangannya pada server kedua.

Kata Kunci – *Backup Data*; Pengelolaan Data; *Rational Unified Process*; *REST API*.

I. PENDAHULUAN

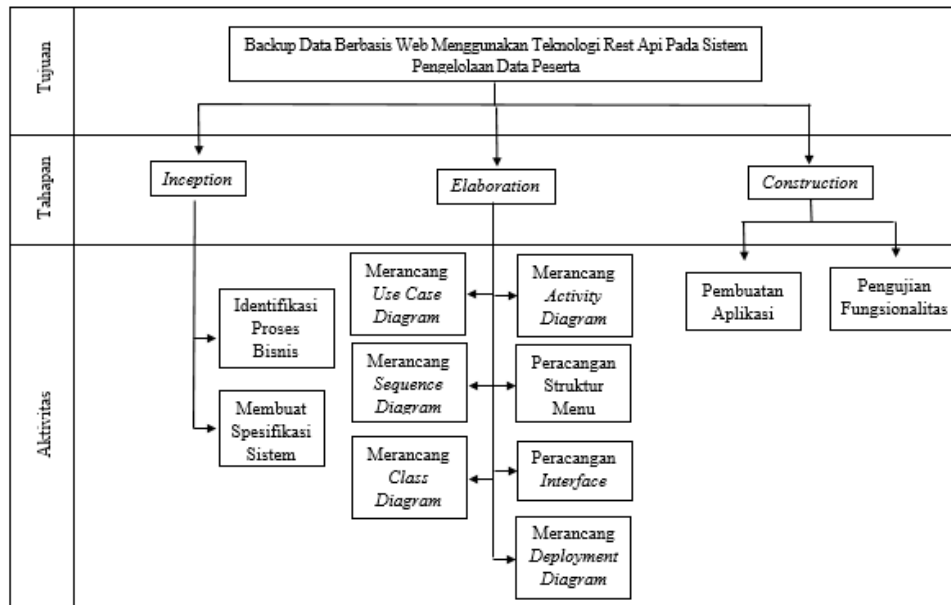
Data merupakan kejadian atau fakta yang berisi suatu jumlah, tindakan, kejadian, aktivitas atau transaksi yang tidak bermakna serta dirumuskan oleh sekelompok organisasi tertentu [1]. Pengelolaan data ialah proses memasukkan, mengelola dan mengeluarkan data menjadi sebuah informasi dimana informasi yang dihasilkan dapat berguna bagi suatu organisasi [2]. Manfaat Pengelolaan data antara lain bisa menjadi sebuah alat bukti apabila terjadi masalah, serta dapat digunakan sebagai alat pertanggung jawaban yang nantinya digunakan untuk pengambilan keputusan. Pengelolaan data yang tertib dan teratur dapat bermanfaat secara optimal, sedangkan jika pengelolaan data dilakukan tidak tertib akan menimbulkan masalah yang baru. Pengaruh dari penumpukan data dan pengelolaan data yang tidak menentu dapat mengakibatkan kinerja pelaksanaan tugas dan fungsi suatu organisasi menjadi terganggu. Jika terjadinya kesulitan dalam pencarian data maka hal tersebut dapat mengganggu proses pengelolaan data. Solusinya yaitu melakukan *backup* data yang bertujuan untuk penyelamatan data yang dilakukan dengan cara menyalin *file* yang berbeda dari tempat aslinya sehingga salinan ini dapat digunakan untuk memulihkan data yang asli jika terjadi kerusakan data ataupun kehilangan data [3].

Balai Latihan Kerja (BLK) Komunitas Al-Wasilah adalah BLK Komunitas yang berada di kota Garut tepatnya di Jalan Raya Cipanas Baru, Kp. Pangkalan Kaler rt.02/rw.12, Kelurahan Pananjung, Kecamatan Tarogong Kaler, Kabupaten Garut, Provinsi Jawa Barat. BLK Komunitas Al-Wasilah berada dibawah naungan Yayasan Pesantren Luhur Al-Wasilah dan didirikan pada tahun 2018, serta mulai aktif menjalankan pelatihan pada tahun 2019. BLK Komunitas Al-Wasilah sampai saat ini (September 2020) telah melaksanakan 9 paket pelatihan dengan total jumlah peserta 144 orang yang terbagi kedalam 4 kejuruan yaitu *Basic Office*, *Computer Operator Assistant (COA)*, *Certified Computer Profesional Office Administrator (CCPOA)*, dan *Design Grafis*, dimana setiap paket/angkatan terdiri dari 16 orang peserta. Seiring dengan bertambahnya pelatihan setiap tahun maka akan berbanding lurus dengan bertambahnya data peserta yang akan dikelola atau di simpan. Pengelolaan data peserta di BLK Komunitas Al-Wasilah pada saat ini sudah menggunakan sebuah sistem yang membantu staf untuk mengelola data peserta. Sistem ini membuat pekerjaan staf BLK lebih cepat menyelesaikan proses pengelolaan data peserta. Namun, kendala yang dihadapi yaitu jika terjadi masalah pada sistem yang dapat menyebabkan proses pengelolaan data terhambat bahkan sampai terjadinya kehilangan data.

Pada penelitian sebelumnya, yang pertama sudah menggunakan Sistem *fail over* dengan DRBD Dan *Heartbeat* yaitu ketika terjadi masalah maka secara otomatis *server* utama dialihkan ke *server backup* untuk menggantikan *server* utama [4]. Penelitian kedua menghasilkan sebuah sistem informasi yang dapat diakses melalui web dan *android* yang digunakan untuk mempermudah mahasiswa dalam mendapatkan pelayanan dari USI [5]. Penelitian ketiga ini membantu dalam proses pengambilan data yang terdapat pada server dan nantinya data pada aplikasi *mobile* diperbaharui secara *realtime* dengan menggunakan *web service* yaitu teknologi *REST API* [6]. Penelitian keempat menghasilkan sebuah sistem informasi yang dimanfaatkan untuk pengelolaan keuangan di SMK Islam Atturmudziyyah Garut, sehingga proses pengelolaan data keuangannya lebih terorganisir. Selain itu sistem informasi ini juga dapat digunakan oleh siswa untuk melakukan pengecekan data keuangan masing-masing [7]. Penelitian kelima menghasilkan sebuah sistem yang membantu pencatatan data barang serta rekapan data transaksi di bengkel [8]. Berdasarkan penelitian-penelitian yang dirujuk, terlihat bahwa sistem *backup* dengan menggunakan teknologi *REST API* dibutuhkan untuk jembatan dalam proses sinkronisasi antara *server* utama dan *server* cadangan, dimana proses sinkronisasi ini dilakukan dengan cara *server* cadangan melakukan pengecekan untuk memastikan apakah data pada *server* ini telah sesuai dengan data yang ada pada *server* utama. Jika data belum sesuai maka data pada server utama diambil oleh *server* cadangan. Maka penulis mengambil judul Implementasi Sistem *Backup* Data Dengan Teknologi *REST API* Pada Sistem Pengelolaan Data Peserta BLK Komunitas Al-Wasilah Berbasis Web.

II. URAIAN PENELITIAN

Metodologi yang digunakan untuk Implementasi Teknologi *REST API* untuk melakukan *Backup* data pada sistem pengelolaan data peserta BLK Komunitas Al-Wasilah adalah *Rational Unified Process (RUP)* yang dibatasi sampai fase *construction*. Penggunaan metodologi *RUP* dalam penelitian ini karena *RUP* merupakan metodologi untuk pemrograman berorientasi objek dan setiap tahapan pengembangan sistem dilakukan secara berulang [9] serta pada penelitian sebelumnya yang dilakukan oleh [7] yang membahas tentang pengelolaan data juga menggunakan metodologi *RUP*. Proses perencanaan hingga pengujian sistem secara terorganisasi menjadi struktur hierarkis dituangkan dengan menggunakan *Work Breakdown Structure*. Dimana dengan *WBS*, setiap proses pekerjaan menjadi lebih detail, berdasarkan metode perancangan yang digunakannya. Berikut merupakan gambaran *WBS* yang mengacu kepada tujuan dan metode penelitian yang digunakan, sebagaimana tergambar pada Gambar 1.



Gambar 1: *Work Breakdown Structure* Implementasi Teknologi *REST API* Untuk Melakukan *Backup Data* Pada Sistem Pengelolaan Data Peserta BLK Komunitas AI-Wasilah

Berikut merupakan penjelasan untuk *WBS* pada gambar 1:

1. Tahap yang pertama adalah *Inception* yang terdiri dari dua aktivitas yaitu membuat spesifikasi sistem dan identifikasi proses bisnis. Dalam tahapan ini kebutuhan sistem dan proses bisnis untuk menambah fitur *backup* pada sistem pengelolaan data peserta BLK Komunitas AI-Wasilah diperoleh dari hasil studi literatur dan wawancara yang dilakukan pada proses identifikasi awal penelitian.
2. Tahap kedua yaitu *elaboration*, aktivitas dalam tahapan ini yaitu merancang *Unified Modelling Language* (UML) berupa *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram* dan *deployment diagram* yang dibuat dengan menggunakan *argoUML* yang merupakan *opensource UML modeling tool* yang digunakan untuk membuat model atau diagram untuk menggambarkan sebuah sistem. Menggunakan *java kelas foundation* yang memungkinkan *argoUML* dapat berjalan di berbagai *platform* [10]. Juga perancangan struktur menu dan *interface* yang dibuat dengan menggunakan *corel draw* yang merupakan aplikasi yang mengolah gambar vektor dapat berupa suatu garis ataupun bidang yang diolah berdasarkan pengaturan angka-angka vektor [11]. Hasilnya akan berupa *prototype* untuk sistem pengelolaan data peserta BLK Komunitas AI-Wasilah yang ditambah dengan fitur *backup*.
3. Tahap ketiga yaitu *construction*, dimana tahapan ini merupakan implementasi kode program yang sesuai dengan apa yang telah ditentukan pada tahapan sebelumnya yaitu proses penambahan fitur *backup* pada sistem pengelolaan data peserta BLK Komunitas AI-Wasilah, sehingga menghasilkan sistem yang siap di uji coba menggunakan *XAMPP* (*Apache, Mysql, PHP, dan Perl*) paket perangkat lunak yang berisi konfigurasi untuk *Web Server* untuk mempermudah dalam membangun program, *MySQL* aplikasi yang digunakan pengolahan *database* atau disebut juga *DBMS* (*Database Management Sistem*) dan juga bersifat jaringan [12] juga menggunakan *codeigniter* yang merupakan aplikasi *open source* yang berupa *framework* dengan model *MVC* (*Model, View, Controller*) untuk membangun *website* dinamis, yang berfungsi untuk memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah [13] yang dituangkan dalam sebuah *text editor* untuk kode program yang dapat berjalan di berbagai sistem operasi yaitu *sublime text* [14]. Pengujian pada tahapan ini menggunakan metode *blackbox testing*, yang berfokus pada fungsi perangkat lunak itu sendiri. Pengujian yang dilakukan oleh pengembang dan staff BLK Komunitas AI-Wasilah yang bertugas dalam proses pengelolaan data. Teknik ini menguji berjalannya fungsi dari perangkat lunak sudah berjalan dengan benar atau tidak, menguji kesalahan antarmuka yang ditampilkan perangkat lunak, menguji kesalahan struktur data dan akses basis data, menguji kesalahan performansi, dan kesalahan inisialisasi serta terminasi [15].

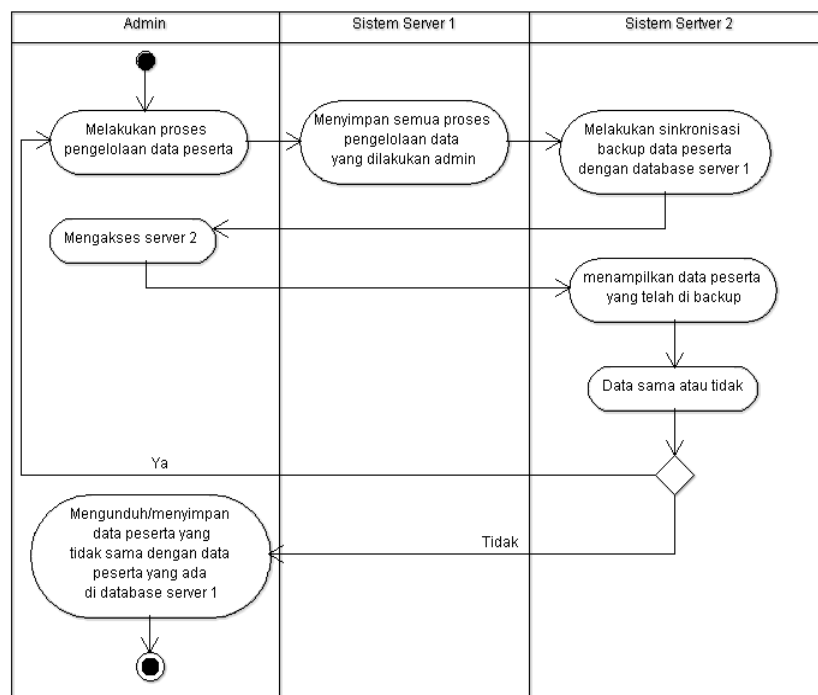
III. HASIL DAN DISKUSI

A. Inception

Aktivitas yang dilakukan dalam tahap ini yaitu melakukan identifikasi proses bisnis yang didapat dari proses wawancara dan studi literatur, serta membuat spesifikasi dari aplikasi yang akan dibuat.

1. Identifikasi Proses Bisnis

Tahapan ini merupakan tahap identifikasi untuk proses bisnis dimana tahapan ini akan menghasilkan sebuah proses bisnis sistem yang akan dibuat, pencarian data untuk membuat proses bisnis dilakukan dengan cara wawancara serta studi literatur. Pengguna yang terlibat dalam implementasi teknologi *REST API* untuk melakukan *backup* data peserta pada sistem pengelolaan data ini adalah admin. Berikut merupakan proses bisnis yang dihasilkan dalam tahapan ini:



Gambar 2: Proses Bisnis Implementasi Teknologi *REST API* Untuk Melakukan *Backup* Data Pada Sistem Pengelolaan Data Peserta BLK Komunitas AI-Wasilah

2. Membuat Spesifikasi Sistem

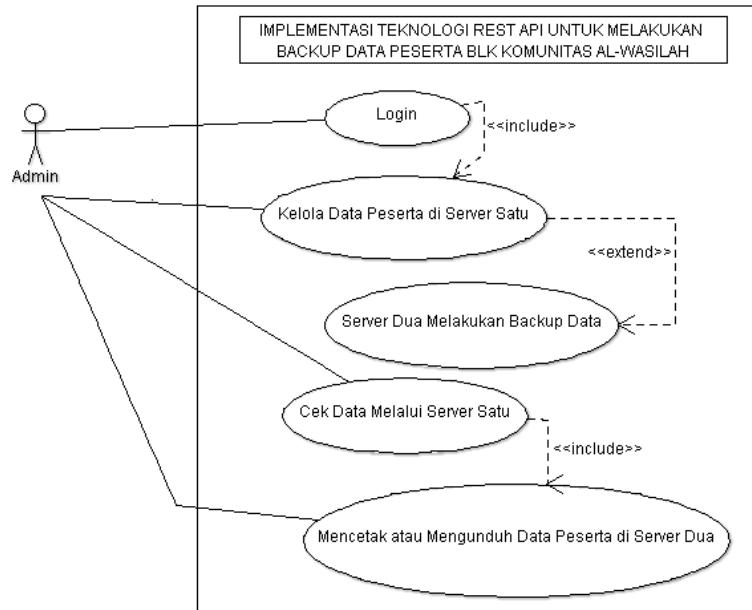
Tahapan ini menjelaskan mengenai kebutuhan apa saja yang diperlukan dalam mengembangkan sistem *backup* data peserta. Rincian persyaratan pada implementasi *REST API* untuk melakukan *backup* data peserta ini meliputi persyaratan tampilan, persyaratan sistem dan persyaratan pengembangan.

B. Elaboration

Hal yang dilakukan pada fase *elaboration* yaitu membuat rancangan atau arsitektur sistem. Penggambaran awalnya diimplementasikan menjadi diagram *use case*, diagram aktivitas, *sequence diagram*, *class diagram*, *deployment diagram* serta membuat struktur menu dan tampilan.

1. Perancangan Use Case Diagram

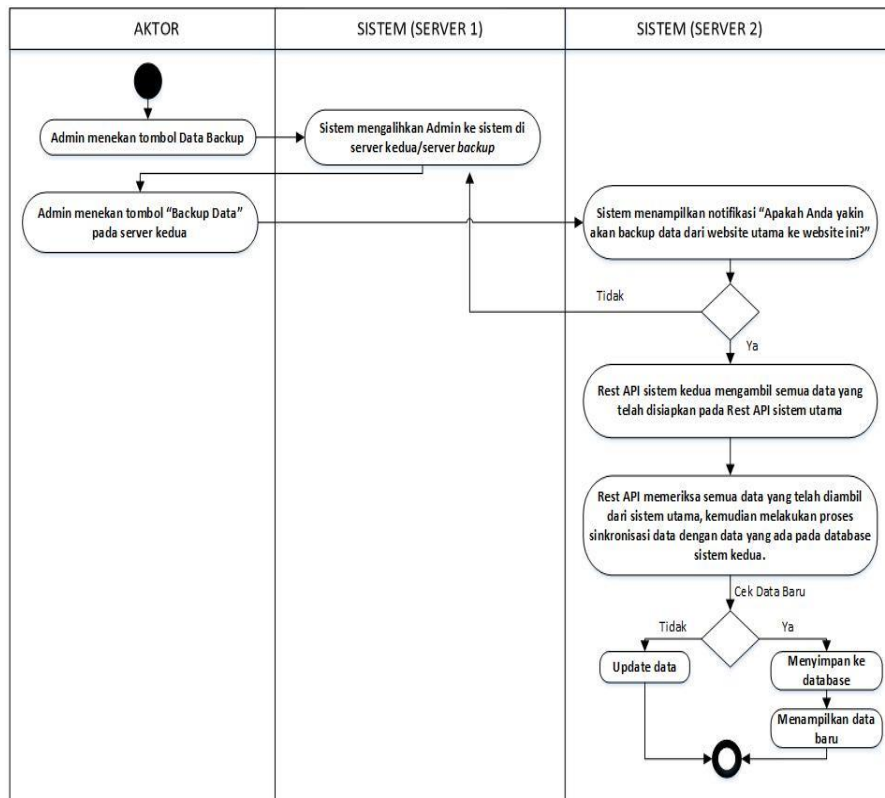
Perancangan *use case* sendiri digunakan untuk mengetahui fungsi-fungsi apa saja yang berada pada sistem serta siapa saja yang memiliki hak akses dari fungsi tersebut.



Gambar 3: Use Case Diagram Implementasi Teknologi REST API Untuk Melakukan Backup Data Pada Sistem Pengelolaan Data Peserta BLK Komunitas Al-Wasilah

2. Perancangan Activity Diagram

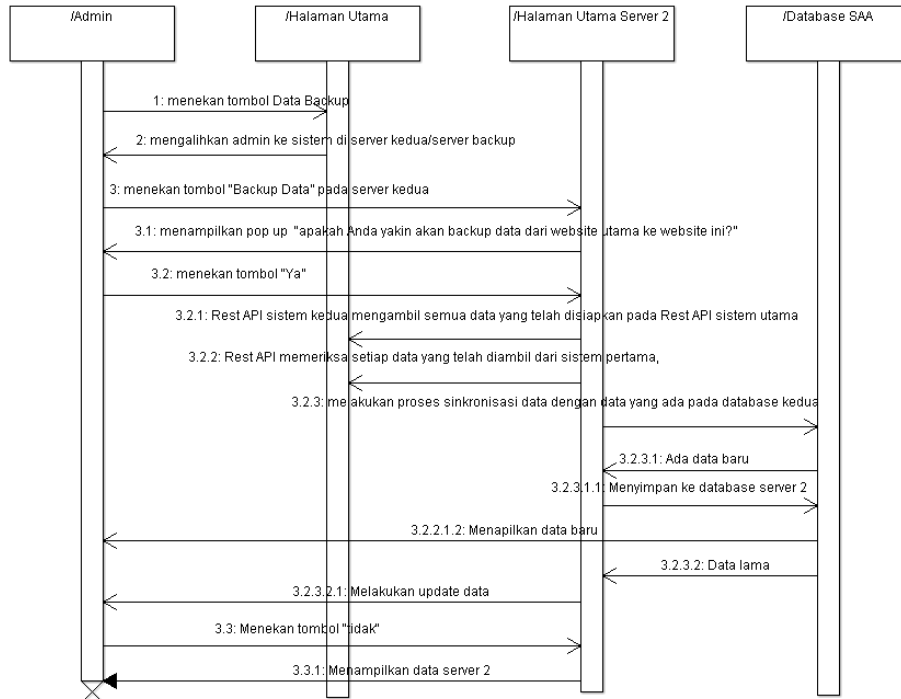
Selanjutnya dilakukan perancangan *activity diagram* yang digunakan sebagai pemodelan sebuah sistem dari *use case* yang telah dibuat. Berikut merupakan *activity diagram* untuk Sistem backup data peserta BLK Komunitas Al-Wasilah:



Gambar 4: Activity Diagram Backup Data

3. Perancangan Sequence Diagram

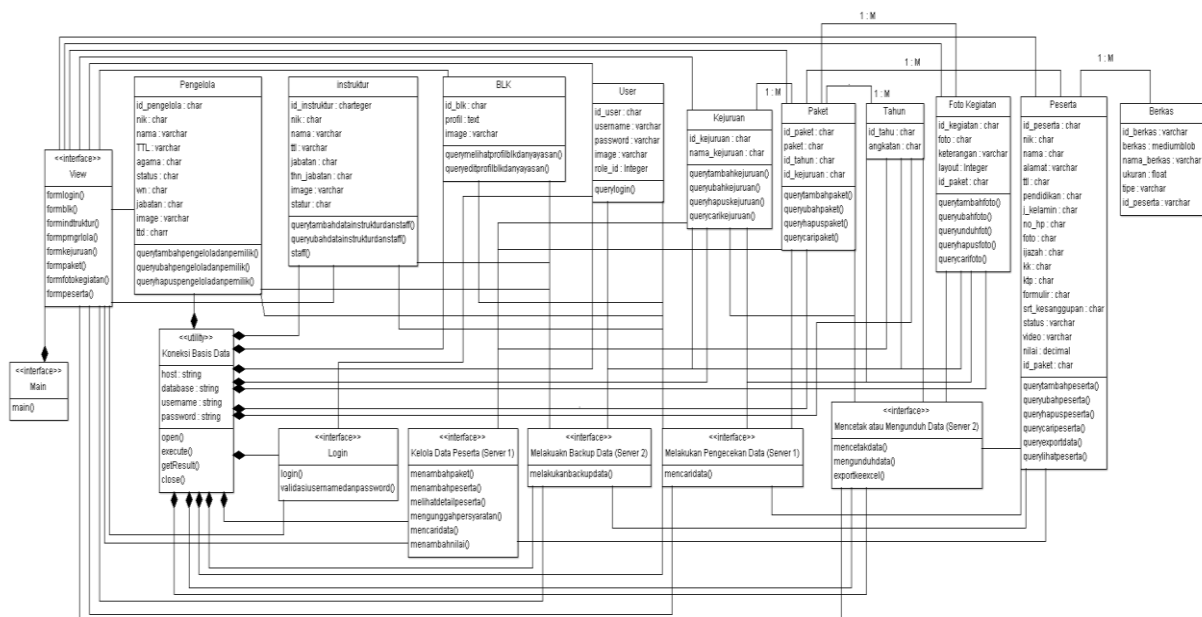
Sequence diagram digunakan untuk menggambarkan proses interaksi antara aktor dengan sistem. Sequence diagram yang dirancang untuk implementasi REST API pada sistem pengelolaan data peserta ini adalah sebagai berikut:



Gambar 5: Sequence Diagram Backup Data

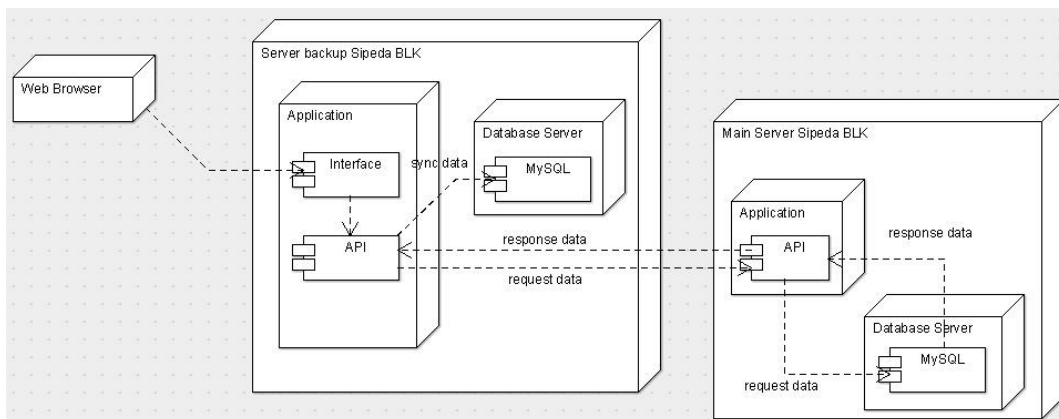
4. Perancangan Class Diagram

Class diagram sering disebut juga desain pemrograman yang merupakan sebuah hubungan rancangan pada database dengan interface. Berikut ini merupakan class diagram untuk implementasi REST API untuk melakukan backup data peserta pada Sistem pengelolaan data peserta :



Gambar 6: Class Diagram Backup Data Berbasis Web Menggunakan Teknologi REST API Pada Sistem Pengelolaan Data Peserta

5. Perancangan *Deployment Diagram*

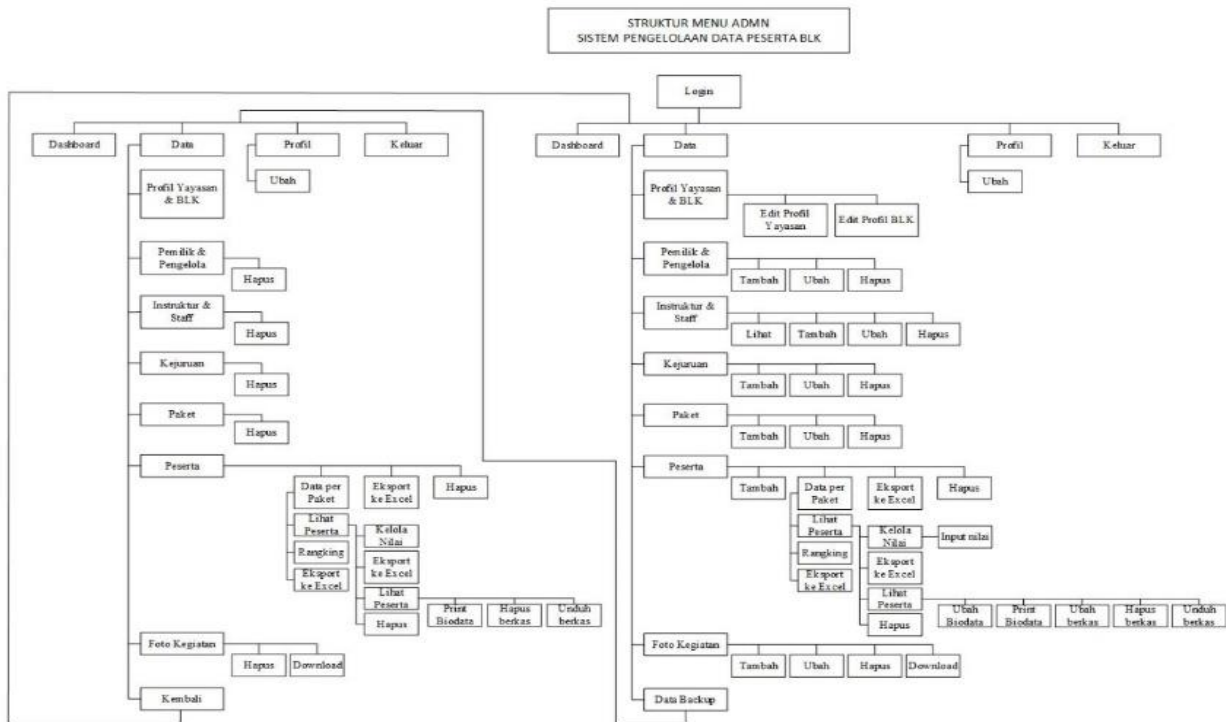


Gambar 7: *Deployment Diagram* Backup Data Berbasis Web Menggunakan Teknologi *REST API* Pada Sistem Pengelolaan Data Peserta

Perancangan *deployment diagram* bertujuan untuk menggambarkan proses yang terjadi pada saat *backup* data dari *server* utama ke *server backup*, pada sistem pengelolaan data peserta BLK Komunitas Al-Wasilah. Ketika Admin melakukan *backup* data yang dapat diakses pada *server* kedua (*server backup*), maka *REST API* pada *server* kedua akan menghubungi *REST API* *server* utama untuk mengambil data yang tersimpan di *database server* utama. Selanjutnya *REST API* *server* utama akan mengambil semua data pada *database server* utama, sehingga semua data yang telah disiapkan dapat di akses oleh *REST API* *server* kedua dalam bentuk JSON. Data yang telah diterima dari *server* utama di ubah dari *file* JSON menjadi *file* HTML, selanjutnya melakukan proses sinkronisasi data yang diterima dengan data pada *database server backup* dengan kondisi jika terdapat data yang baru pada *server* utama maka data ditambahkan ke *server backup*, jika ada data yang diubah pada *server* utama maka data yang ada di *server* kedua juga di *update* sesuai data pada *server* utama, dan jika data pada *server* utama dihapus maka data di *server* kedua tidak akan terhapus. Setelah proses sinkronisasi selesai maka data yang telah tersimpan di *database server backup* ditampilkan melalui *browser* berupa HTML.

6. Perancangan Tampilan Struktur Menu

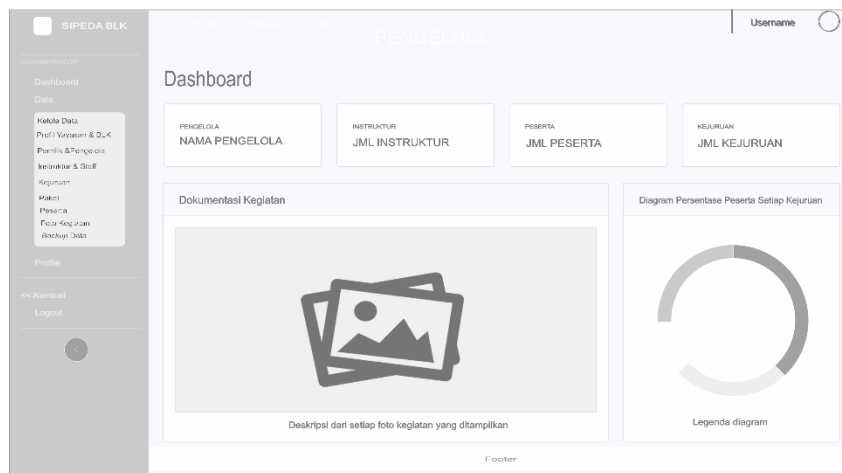
Tahapan perancangan struktur menu ini digunakan untuk perancangan gambaran rincian Sistem. Berikut merupakan rincian struktur menu untuk implementasi *REST API* untuk *backup* data pada Sistem pengelolaan data peserta :



Gambar 8: Struktur Menu *Backup* Data Berbasis Web Menggunakan *Teknologi REST API* Pada Sistem Pengelolaan Data Peserta

7. Perancangan *Interface*

Perancangan *interface* dilakukan untuk gambaran atau tampilan sementara. Adapun tampilan *interface* yang telah dirancang adalah sebagai berikut:



Gambar 9: *Interface Dashboard Admin*

C. Construction

Pada tahapan ini merupakan tahap implementasi dari perancangan yang sebelumnya telah dibuat ke dalam bahasa pemrograman.

1. Implementasi Program

PHP merupakan bahasa pemrograman yang digunakan dalam pembuatan sistem pengelolaan data yang dilengkapi fitur *backup* menggunakan teknologi *REST API* ini serta *codeigniter* yang digunakan sebagai

framework dalam pembuatan sistem. Berikut merupakan implementasi Rest API pada kode program di server utama dan server kedua.

```

1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  use chriskacerguis\RestServer\RestController;
6
7  class Api extends RestController {
8
9      function __construct($config = 'rest') {
10         parent::__construct($config);
11         $this->load->database();
12     }
13
14     //Menampilkan data
15     function index_get() {
16         $id = $this->get('id');
17         $table = $this->get('table');
18         $id_name = $this->get('id_name');
19         // print_r($id); die();
20         $tables = $this->db->list_tables();
21         $results = [];
22
23         foreach ($tables as $table_name)
24         {
25             if ($id == '' && $id_name) {
26                 $results[$table_name] = $this->db->get($table_name)->result();
27             } else {
28                 if ($table_name == $table && $id_name) {
29                     $this->db->where($id_name, $id);
30                 }
31                 $results[$table_name] = $this->db->get($table_name)->result();
32             }
33         }
34
35         if($table) {
36             $this->response($results[$table], 200);
37         } else {
38             $this->response($results, 200);
39         }
40     }
41 }
42 ?>
    
```

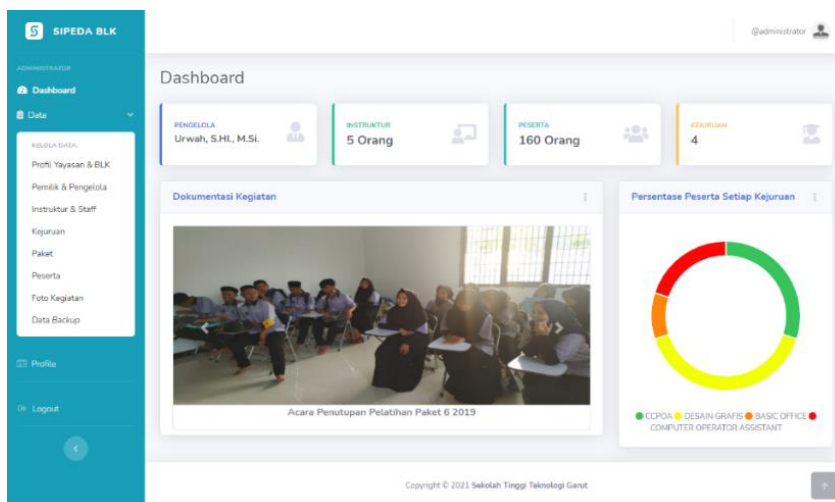
Gambar 10: Kode Program penggunaan API Server 1

```

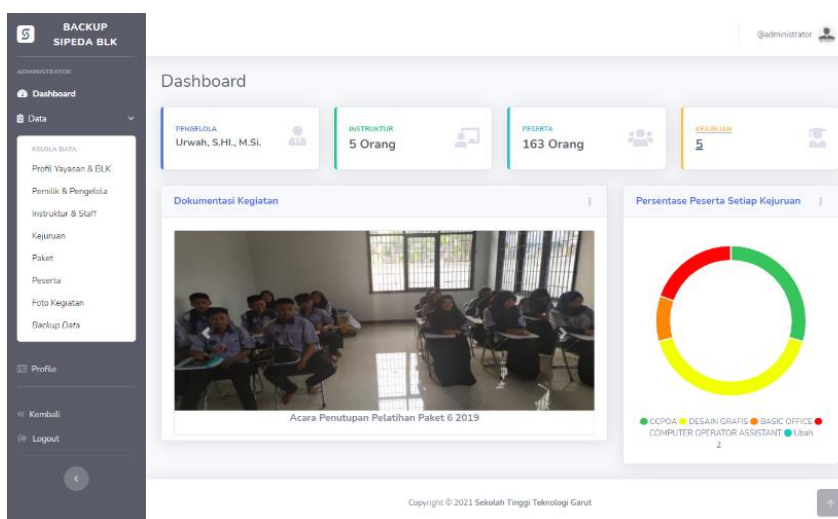
5  use chriskacerguis\RestServer\RestController;
6
7  class Api extends RestController {
8
9      function __construct($config = 'rest') {
10         parent::__construct($config);
11         $this->load->database();
12     }
13
14     function backup_get() {
15         $this->load->helper('url');
16
17         $url = "http://localhost/blk/api";
18         $data['data_server1'] = file_get_contents($url);
19
20         $dataServer1 = json_decode($data['data_server1'], TRUE);
21
22         $this->load->model('ApiModel');
23
24         $idList = [
25             'berkas' => 'id_berkas',
26             'bik' => 'id_bik',
27             'foto_kegiatan' => 'id_kegiatan',
28             'instruktur' => 'id_instruktur',
29             'kejuruan' => 'id_kejuruan',
30             'paket' => 'id_paket',
31             'pengelola' => 'id_pengelola',
32             'peserta' => 'id_peserta',
33             'tahun' => 'id_tahun',
34             'user' => 'id_user',
35         ];
36
37         foreach ($dataServer1 as $tables => $values) {
38             $nameIdTable = $idList[$tables];
39             if ($nameIdTable != ''){
40                 foreach ($values as $value) {
41                     $row_count = $this->ApiModel->getDataById($tables, $value[$nameIdTable], $nameIdTable);
42                     if ($row_count == 0) {
43                         $results = $this->ApiModel->action($value, $tables, $nameIdTable, 'add');
44                     } else {
45                         $results = $this->ApiModel->action($value, $tables, $nameIdTable, 'update');
46                     }
47                 }
48             }
49         }
50
51         $this->session->set_flashdata('message', '<div class="alert alert-success" role="alert">Data telah berhasil di <i>backup</i>/</div>');
52         redirect('Admin');
53     }
54 }
    
```

Gambar 11: Kode Program penggunaan API Server 2

Berikut implementasi rancangan *interface* yang telah dibuat:



Gambar 12: Implementasi *Dashboard Admin Server 1*



Gambar 13: Implementasi *Dashboard Admin Server 2*

2. Pengujian

Blackbox testing merupakan metode yang digunakan dalam menguji fungsi sistem. Pengujian pada penelitian ini dilakukan untuk memastikan apakah *backup* data yang dilakukan sudah berjalan dengan lancar sebagaimana mestinya dengan melihat berapa lama waktu yang diperlukan untuk *backup* tersebut dan memperhatikan hasil data yang telah di *backup* [16]. Hasil pengujian nya dituangkan dalam tabel dibawah ini :

Tabel 1: Hasil Pengujian *Backup* data peserta

No	Jenis File	Ukuran File	Waktu Backup	Kualitas Hasil Backup
1	Teks	1 data peserta (26 kata)	6 detik	Sesuai
		10 data peserta (260 kata)	7 detik	Sesuai
		16 data peserta (410 kata)	8 detik	Sesuai

No	Jenis File	Ukuran File	Waktu Backup	Kualitas Hasil Backup
2	<i>File extension</i>	1 peserta 4 <i>image</i> (1,81 MB)	7 detik	Sesuai
	<i>image (dot jpg, jpeg, & png)</i>	10 peserta 38 <i>image</i> (9,60 MB)	13 detik	Sesuai
		16 peserta 59 <i>image</i> (14,9 MB)	18 detik	Sesuai
		128 peserta 404 <i>image</i> (105 MB)	118 detik	Sesuai

Setelah melakukan pengujian *backup* data peserta dengan menggunakan sampel beberapa data peserta yang dibagi menjadi tiga kali pengujian dengan data yang berbeda berupa teks, serta empat kali pengujian dengan tambahan file berkas berupa *image*. Hasil pengujian menunjukkan bahwa semakin banyak data yang di *backup* semakin bertambah pula waktu yang dibutuhkan untuk melakukan *backup* data, serta jumlah kata dan ukuran gambar di *server* utama sesuai dengan jumlah kata dan ukuran gambar hasil *backup* di *server backup*.

IV. KESIMPULAN

Sesuai dengan hasil penelitian dan pembahasan penelitian disimpulkan bahwa Implementasi REST API membantu dalam proses *backup* data baik itu tulisan ataupun *image* antara *server* utama dan *server backup* telah berhasil, dimana *REST API* pada *server* kedua mengakses data yang telah disiapkan oleh *server* utama lalu melakukan sinkronisasi data. Ketika terdapat data baru di *server* utama, maka data akan ditambahkan ke *server backup* dan jika terdapat data yang diperbaharui di *server* utama, maka data yang telah ada sebelumnya di *server backup* akan di *update*. Diharapkan untuk pengembangan kedepannya dalam melakukan proses *backup* data dapat dilakukan secara otomatis dengan waktu yang telah ditentukan, serta dilengkapi dengan fitur berpindah *server* secara otomatis sehingga jika terjadi *down* pada *server* utama sistem masih dapat digunakan pada *server* kedua.

DAFTAR PUSTAKA

- [1] M. Abdurahman, "Sistem Informasi Data Pegawai Berbasis Web Pada Kementerian Kelautan Dan Perikanan Kota Ternate," *J. Ilm. Ilk. - Ilmu Komput. Inform.*, vol. 1, no. 2, pp. 70–78, 2018, doi: 10.47324/ilkominfo.v1i2.10.
- [2] Nawassyarif, M. Julkarnain, and K. Rizki Ananda, "Sistem Informasi Pengolahan Data Ternak Unit Pelaksana Teknis Produksi Dan Kesehatan Hewan Berbasis Web," *J. Inform. Teknol. dan Sains*, vol. 2, no. 1, pp. 32–39, 2020, doi: 10.51401/jinteks.v2i1.556.
- [3] E. Haryadi, A. Abdussomad, and R. Robi, "Implementasi Sistem Backup Data Perusahaan Sebagai Bagian dari Disaster Recovery Plan," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 29, no. 2, pp. 6–11, 2019, doi: 10.37277/stch.v29i2.331.
- [4] M. F. S. Afif Tito, M. F. Afif, and T. Suryono, "Implementasi Disaster recovery plan dengan Sistem Fail Over Menggunakan DRBD dan Heartbeat pada Data Center FKIP UNS," *Indones. J. Netw. Secur.*, vol. 2, no. 2, pp. 64–69, 2013.
- [5] S. P. Indriyani and D. Kurniadi, "Rancang Bangun Sistem Informasi Pelayanan Unit Sistem Informasi," *J. Algoritm.*, vol. 6, no. October, pp. 68–75, 2017.
- [6] A. Pamuji, "Naskah publikasi rancang bangun web service menggunakan representational state transfer untuk pengolahan data barang," 2020.
- [7] S. Rahayu and P. A. Rahayu, "Perancangan Sistem Informasi Pengelolaan Keuangan Berbasis Web di Sekolah Menengah Kejuruan Islam Atturmudziyyah Garut," *J. Algoritm.*, vol. 14, no. 2, pp. 538–545, 2017, doi: 10.33364/algoritma/v.14-2.538.
- [8] A. Permana and A. Mulyani, "Perancangan Aplikasi Pengelolaan Data Penjualan Sparepart Kendaraan Bermotor Berbasis Web," *J. Algoritm.*, vol. 17, no. 1, pp. 8–14, 2020, doi: 10.33364/algoritma/v.17-1.8.
- [9] R. A. Sukamto and M. Salahuddin, *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*.

- Bandung: Informatika Badung, 2018.
- [10] H. Hasugian, “Rancang Bangun Sistem Informasi Management Vendor Untuk Mendukung Electronic Procurement Rekayasa Online Pada Pt . Rekayasa Industri,” *Semin. Nas. Inform. 2012*, vol. 2012, no. semnasIF, pp. 97–102, 2012.
 - [11] R. Agustina, “Pelatihan Desain Grafis Untuk Perangkat Desa Dalam Rangka Peningkatan Sdm Di Desa Ngawonggo Kecamatan Tajinan Kab. Malang,” *J. Pengabd. Masy. Univ. Merdeka Malang*, vol. 2, no. 1, pp. 37–42, 2017, doi: 10.26905/abdimas.v2i1.1289.
 - [12] A. Josi, “Penerapan Metode Prototyping Dalam Membangun Website Desa (Studi Kasus Desa Sugihan Kecamatan Rambang),” *Jti*, vol. 9, no. 1, pp. 50–57, 2017.
 - [13] U. Asidhiqi and A. Hartanto, “PEMBUATAN DAN PERANCANGAN SISTEM E-LETTER BERBASIS WEB DENGAN CODEIGNITER DAN BOOTSTRAP Studi Kasus : Kantor Kecamatan Klego,” *Data Manaj. dan Teknol. Inf.*, vol. 14, no. 4, p. 54, 2013.
 - [14] E. Subowo and M. Saputra, “Sistem Informasi Peternakan Ayam Broiler Di Kabupaten Pekalongan Berbasis Web Dan Android,” *Surya Inform.*, vol. 6, no. 1, pp. 53–65, 2019.
 - [15] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, “PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN),” vol. I, no. 3, pp. 31–36, 2015.
 - [16] J. F. Andry, “Pengembangan Aplikasi Backup Dan Restore Secara Automatisasi Menggunakan Sdlc Untuk Mencegah Bencana,” *J. Muara Sains, Teknol. Kedokt. dan Ilmu Kesehat.*, vol. 1, no. 1, pp. 29–38, 2017, doi: 10.24912/jmstkik.v1i1.389.