

MODEL INTEGRASI DATA AKADEMIK PERGURUAN TINGGI DENGAN MEMANFAATKAN WEB SERVICE DAN PROXY OBJECT

Sopingi¹ dan Nurchim²

¹Sistem Informasi STMIK Duta Bangsa, ²Teknik Informatika STMIK Duta Bangsa

Jl. Bhayangkara No. 55 Surakarta 57154 Telp (0271) 719552

E-mail : sopingi@stmikdb.ac.id¹, nurchim@stmikdb.ac.id²

Abstrak

Aplikasi Feeder Pangkalan Data Pendidikan Tinggi (PDDIKTI) mengharuskan Operator memasukkan satu persatu data ke dalam aplikasi dengan sumber data yang sudah valid. Permasalahan yang dihadapi adalah sistem informasi akademik yang sudah ada di perguruan tinggi tidak memiliki struktur yang sama dengan Feeder PDDIKTI baik dari sisi basis data dan alur sistemnya. Oleh karena itu, operator diharuskan memverifikasi data yang sudah dimasukkan ke dalam Aplikasi Feeder PDDIKTI. Feeder PDDIKTI memiliki fitur web service yang dapat digunakan untuk memanipulasi data tanpa menggunakan interface Feeder PDDIKTI. Web service menjadi jembatan dalam pertukaran data antara basis data Feeder PDDIKTI dengan basis data akademik perguruan tinggi. Penulis melakukan penelitian untuk memodelkan sistem informasi akademik yang nantinya dapat digunakan untuk membangun sistem yang terintegrasi langsung dengan basis data Feeder PDDIKTI. Penulis memanfaatkan teknologi web service dan proxy object. Web service dapat menyelesaikan permasalahan interoperabilitas antar aplikasi yang berbeda platform baik perangkat lunak, system operasi maupun bahasa pemrogramannya. Sedangkan penggunaan proxy object bertujuan untuk memberikan akses kecepatan terhadap data yang tersedia di web service dan mencegah permintaan data yang sama secara berulang-ulang ke basis data. Hasil dari model yang penulis rancang mendukung dalam proses sinkronisasi pada data mahasiswa nasional, mahasiswa perguruan tinggi, konversi mahasiswa transfer atau pindah, mata kuliah, kurikulum, mata kuliah kurikulum, kelas perkuliahan, kartu rencana studi, aktifitas mengajar dosen, pengaturan range nilai, nilai mahasiswa, aktifitas kuliah mahasiswa dan mahasiswa lulus atau keluar.

Kata Kunci: Integrasi, Sinkronisasi, Feeder, Web Service, Proxy Object

I. PENDAHULUAN

Jumlah perguruan tinggi di Indonesia pada tahun 2016 mencapai 4.401 perguruan tinggi yang terdiri dari 1.105 Akademi, 240 Politeknik, 2.401 Sekolah Tinggi, 125 Institut dan 530 Universitas. Jumlah perguruan tinggi yang begitu banyak, dengan 24.057 program studi^[8], mengindikasikan diperlukannya sistem yang mampu menangani pelaporan akademik perguruan tinggi secara cepat, efektif dan efisien. Untuk memenuhi kebutuhan tersebut, Kementerian Riset, Teknologi dan Pendidikan membangun aplikasi Feeder Pangkalan Data Perguruan Tinggi (PDDIKTI) dan pada bulan Januari 2015. Selama periode bulan Februari sampai dengan Desember 2015 setiap operator perguruan tinggi disibukan dengan sistem Feeder PDDIKTI, yang memiliki tampilan dan prosedur yang jauh berbeda dari sistem sebelumnya yang menggunakan sistem operasi DOS. Pada sistem sebelumnya memungkinkan operator untuk membuka dan mengisi data langsung ke basis data yang menggunakan format DBF. Sedangkan Feeder PDDIKTI merubah total basis data menggunakan PostgreSQL dengan aplikasinya yang dienkripsi sehingga operator tidak dapat mengisikan data melalui basis data.

Aplikasi Feeder PDDIKTI mengharuskan Operator harus memasukkan satu per satu data yang berasal dari sumber yang sudah valid ke dalam aplikasi. Hal ini membutuhkan proses kerja tiga kali lipat dibandingkan aplikasi sebelumnya. Pertama, operator harus memvalidkan terlebih dahulu data akademik dari sistem yang sudah ada, baik manual maupun terkomputerisasi. Kedua, operator memasukkan data yang sudah valid tersebut ke Aplikasi Feeder PDDIKTI. Ketiga, operator memverifikasi data yang sudah dimasukkan kedalam Aplikasi Feeder PDDIKTI.

Adapun kelebihan Feeder PDDIKTI yaitu terdapatnya aplikasi web service yang dapat digunakan untuk memanipulasi data tanpa menggunakan interface Feeder PDDIKTI. Perguruan tinggi yang sudah memiliki sistem informasi akademik mempunyai struktur data yang tidak sesuai dengan struktur Web Service Feeder, diharuskan melakukan mapping terhadap struktur web service dan kemudian melakukan proses extraction, transform dan load database akademik ke dalam Web Service Feeder PDDIKTI dengan waktu yang relatif lama.

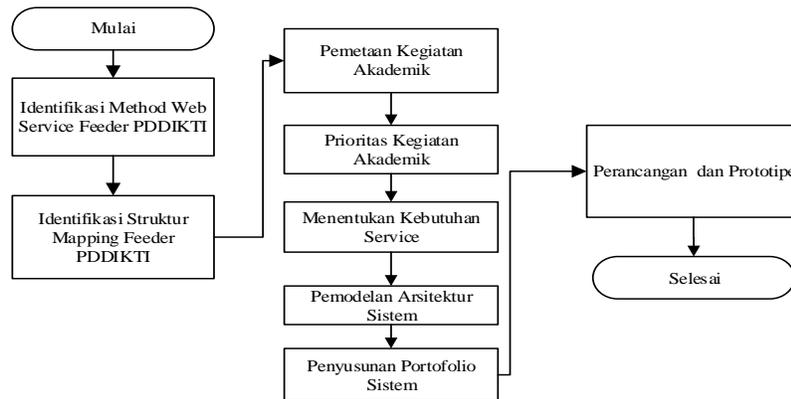
Setelah sistem *Feeder* PDDIKTI disosialisasikan, perguruan tinggi mulai melakukan pengembangan terhadap sistem informasi akademik baik melakukan perubahan atau membangun kembali sistem informasi akademik dari awal agar dapat menyesuaikan dengan *Feeder* PDDIKTI. Pengembangan yang dilakukan di perguruan tinggi masih dalam tahap pengembangan pelaporannya saja yang sering disebut sebagai proses *mapping*. Proses *mapping* membutuhkan waktu yang lama karena harus menata data terlebih dahulu sebelum dimasukkan ke dalam *Feeder*.

Berdasarkan permasalahan di atas maka penulis ingin menganalisa dan merancang model integrasi data akademik perguruan tinggi yang mampu melakukan *generate database* awal yang dapat digunakan untuk membangun sistem informasi akademik dan sinkronisasi data antara sistem informasi akademik dengan *web service Feeder* PDDIKTI. Sistem yang dirancang harus mampu terintegrasi dengan *web service Feeder* PDDIKTI maka sistem akan dirancang dengan memanfaatkan *web service*. *Web service* dapat menyelesaikan permasalahan interoperabilitas antar aplikasi yang berbeda platform baik perangkat lunak aplikasi, sistem operasi maupun bahasa pemrograman^[3]. Sedangkan isu utama *web service* adalah permasalahan kualitas layanan atau Qos (*Quality of service*). Matrik kualitas layanan pada *web service* terbagi dalam beberapa hal antara lain *accessibility*, *reliability*, *performance*^[5]. Untuk mengatasi permasalahan *performance*, berdasarkan penelitian sebelumnya menghasilkan kesimpulan bahwa metode pertukaran data pada *web service* memiliki pengaruh terhadap lamanya proses pertukaran data dan metode *Representational State Transfer* (REST) merupakan konfigurasi dengan nilai *latency* terbaik untuk diimplementasikan dalam proses integrasi data^[7]. Selain penggunaan REST penulis juga akan menerapkan *proxy object* untuk menjembatani *interface* dengan *web service*. *Proxy Object* akan menggantikan Objek asli dari *web service* yang diharapkan dapat memberikan kecepatan akses data dan efisiensi penggunaan sumber daya^[9].

Sebagai pendukung dalam penelitian ini berikut penelitian terdahulu yang pernah dilakukan: (1) Penelitian sebelumnya dengan judul “Perancangan Arsitektur *Enterprise* untuk perguruan tinggi Swasta menggunakan Togaf ADM” menghasilkan rancangan sistem informasi untuk setiap sub unit bisnis dengan tujuan utama meningkatkan kinerja di setiap sub organisasi agar menghasilkan informasi yang relevan, akurat, dan tepat waktu. Ruang lingkup penelitian ini dibatasi pada proses bisnis utama berdasarkan pemetaan menggunakan *value chain* dan hanya sampai pada fase *migration planning* pada TOGAF ADM. Penelitian ini menghasilkan *blueprint* arsitektur sistem informasi berupa kandidat-kandidat aplikasi untuk setiap sub organisasi yang disusun berdasarkan tingkat prioritas kebutuhan sehingga dalam implementasinya lebih terarah dan tidak mengganggu kinerja sistem informasi, baik yang sudah ada maupun sedang dibangun^[6]. (2) dalam penelitian berjudul “Implementasi *Service Oriented Architecture* dengan *web service* untuk Aplikasi Informasi Akademik” menggunakan metodologi berdasarkan konsep SOA yaitu *top-up*. Berdasarkan analisis SOA dihasilkan 4 *service* utama dan 5 *method*, yaitu mata kuliah *service*, dosen *service*, mahasiswa *service*, dan nilai *service*. *Service* dan *method* kemudian dirancang dan diimplementasikan menggunakan perancangan SOA dan teknologi *web service*, untuk menghasilkan suatu *interface* aplikasi yang siap diintegrasikan dengan aplikasi informasi akademik Fakultas Teknik UNSRAT. Setelah keempat *interface web service* ini berhasil diintegrasikan dengan aplikasi informasi akademik, diperoleh hasil bahwa sifat-sifat SOA berhasil diimplementasikan dalam penelitian ini, kecuali sifat *autonomy* dan *composability*. Kesimpulan dari penelitian ini adalah bahwa penggunaan SOA dan *web service* dalam perancangan sistem informasi suatu organisasi sangat tepat untuk menghasilkan suatu aplikasi yang memiliki tingkat penggunaan kembali yang tinggi, dan dapat digunakan antar *platform* dan antar sistem operasi^[4].

II. METODE PENELITIAN

Pemodelan integrasi data akademik perguruan tinggi dengan memanfaatkan *web service* dan *object proxy* menggunakan pendekatan *Service Oriented Architecture* (SOA). SOA adalah sebuah framework yang mengintegrasikan proses bisnis dan mendukung infrastruktur teknologi informasi yang aman, berkomponen terstandarisasi yang dapat digunakan kembali dan disertakan dalam prioritas bisnis yang berubah-ubah^[1]. Tahap - tahap penelitian yang dilakukan seperti pada gambar 1 berikut ini.



Gambar 1. Alur Penelitian

1. Identifikasi *Method Web Service Feeder* PDDIKTI

Dalam tahap ini, dilakukan identifikasi *method* atau fungsi-fungsi yang dapat dipanggil untuk memanipulasi *database* Feeder PDDIKTI. Tahap ini dilakukan dengan menggunakan dokumentasi *web service* dari Feeder PDDIKTI

2. Identifikasi Struktur *Mapping Feeder* PDDIKTI

Tahap identifikasi struktur data, tipe dan panjang dari *field* yang digunakan untuk proses mengirim atau menerima data dari *Web Service* Feeder PDDIKTI. Selain itu juga mengidentifikasi struktur tabel *database* Feeder PDDIKTI yang dapat diakses melalui *web service*. Hasil dari tahap ini berupa tabel-tabel berserta struktur data yang dapat digunakan untuk sistem informasi akademik.

3. Pemetaan Kegiatan Akademik

Pada tahap ini dilakukan pemetaan terhadap kegiatan akademik yang ada di dalam Feeder PDDIKTI dan yang akan tercakup dalam sistem informasi akademik perguruan tinggi. Pada akhir tahap ini dihasilkan daftar kegiatan akademik yang akan diintegrasikan dengan Feeder PDDIKTI.

4. Prioritas Kegiatan Akademik

Pada tahap ini dilakukan penentuan urutan prioritas kegiatan akademik sehingga dihasilkan peta jalur kegiatan akademik yang harus berjalan sesuai urutannya.

5. Menentukan Kebutuhan Sistem

Untuk masing-masing kegiatan akademik yang telah memiliki urutan prioritas, kemudian dilakukan pengumpulan kebutuhan yang akan digunakan dalam sistem, sehingga diperoleh daftar kebutuhan yang harus terpenuhi.

6. Pemodelan Arsitektur Sistem

Pada tahap ini dilakukan pembuatan arsitektur berdasarkan daftar kegiatan akademik yang akan tercakup dalam sistem. Model arsitektur sistem dapat digunakan untuk menentukan kebutuhan perangkat keras, perangkat lunak serta infrastruktur dari sistem informasi akademik.

7. Penyusunan Portofolio Sistem

Portofolio sistem digunakan untuk menggambarkan keseluruhan berdasarkan tahap-tahap analisis sebelumnya.

8. Perancangan dan Prototipe

Tahap ini dilakukan perancangan diagram alir data berupa diagram konteks dan diagram alir data level 0 untuk sistem informasi akademik. Perancangan ini dapat digunakan untuk mengetahui alur data dari dan ke masing-masing entitas, proses dan penyimpanan data. Penulis juga membuat protipe berupa kode dan *query* untuk membuktikan

bahwa data siap untuk diintegrasikan. Pengujian yang dilakukan berupa percobaan pengambilan data dari *database* akademik kemudian dimasukkan ke *database Feeder PDDIKTI* menggunakan *web service*.

III. HASIL DAN PEMBAHASAN

Tahap awal yang dilakukan penulis adalah identifikasi *Method Web Service* dan struktur *Mapping Feeder PDDIKTI*. *Method* dari *web service Feeder PDDIKTI* didapat dengan mengakses URL *web service Feeder* yaitu: <http://alamatip:8082/ws/live.php?wsdl>. *alamatip* menunjukkan alamat IP dari komputer yang terpasang *Feeder PDDIKTI*. Langkah berikutnya penulis melakukan *data mapping*, struktur data *Feeder PDDIKTI* didapatkan dengan memanfaatkan *web service Feeder* dengan menggunakan *method ListTable* dan *GetDictionary*. Struktur tabel yang didapat dari *feeder* dikonversi menjadi *database MySql* yang nantinya dapat digunakan di sistem informasi akademik, sehingga dapat dilakukan migrasi data awal dari *feeder PDDIKTI* ke *database* sistem informasi akademik di perguruan tinggi. Untuk melakukan konversi diperlukan penyesuaian tipe data di masing-masing tabel antara *database postgresql* dengan *mysql*. Setelah dilakukan identifikasi *method* dan *mapping* tabel langkah berikutnya adalah melakukan analisis dan kebutuhan untuk memodelkan integrasi data akademik perguruan tinggi. Adapun tahapan analisis yang telah dilakukan adalah sebagai berikut:

A. Pemetaan Kegiatan Akademik

Kegiatan akademik yang ada di dalam aplikasi *Feeder* yang dapat digunakan untuk integrasi data akademik perguruan tinggi seperti pada tabel 1. Kegiatan akademik pada tabel 1 merupakan kegiatan akademik yang wajib dilaporkan ke *Feeder PDDIKTI* dan dapat ditambahkan menyesuaikan kebutuhan di perguruan tinggi.

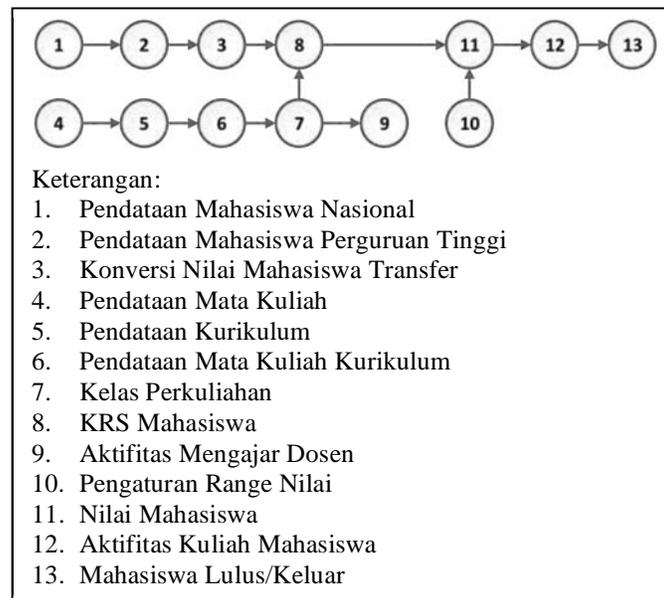
Tabel 1. Pemetaan Kegiatan Akademik

No	Kegiatan Akademik
1	Pendataan Mahasiswa Nasional Pendataan biodata mahasiswa untuk menjadi mahasiswa secara nasional. Data ini tidak diperbolehkan untuk ganda, sehingga perlu verifikasi apakah mahasiswa yang bersangkutan sudah terdaftar sebagai mahasiswa nasional atau belum.
2	Pendataan Mahasiswa Perguruan Tinggi Pendataan mahasiswa perguruan tinggi merupakan proses registrasi mahasiswa ke program studi. Mahasiswa nasional akan menjadi mahasiswa dari perguruan tinggi jika sudah dimasukkan data histori pendidikan atau program studinya.
3	Konversi Nilai Mahasiswa Transfer Pendataan konversi nilai bagi yang pindah atau melanjutkan pendidikan dari program studi atau perguruan tinggi lain.
4	Pendataan Mata Kuliah Pendataan mata kuliah beserta kode, sks, jenis dan kelompok mata kuliah pada perguruan tinggi
5	Pendataan Kurikulum Pendataan nama kurikulum untuk masing-masing program studi
6	Pendataan Mata Kuliah Kurikulum Pendataan mata kuliah yang ada di kurikulum yang sudah dibuat. Mata kuliah dapat mengambil dari mata kuliah yang sudah ada atau mata kuliah baru
7	Kelas Perkuliahan Kelas perkuliahan menyimpan jadwal perkuliahan yang di buka pada semester yang berjalan
8	KRS Mahasiswa Pendataan pengambilan kelas perkuliahan yang diambil oleh mahasiswa pada semester yang berjalan
9	Aktifitas Mengajar Dosen Pendataan dosen yang mengajar pada kelas perkuliahan pada semester berjalan
10	Pengaturan Range Nilai Range nilai digunakan untuk penentuan index dan skala nilai
11	Nilai Mahasiswa

No	Kegiatan Akademik
	Pendataan nilai kedalam KRS yang diambil mahasiswa
12	Aktifitas Kuliah Mahasiswa Pendataan status mahasiswa, index prestasi semester, SKS semester, index prestasi kumulatif dan total SKS yang sudah diambil.
13	Mahasiswa Lulus/Keluar Pendataan mahasiswa yang Lulus atau Keluar

B. Prioritas Kegiatan Akademik

Kegiatan akademik yang sudah dipetakan selanjutnya diurutkan prioritasnya, tujuan dari pengurutan prioritas adalah untuk membentuk jalur kegiatan akademik yang memiliki hubungan antar kegiatan akademik dan juga sebagai urutan proses sinkronisasi data yang akan dilakukan. Urutan kegiatan akademik gambar 2 sudah divalidasi selama 2 (dua) semester pada tahun akademik 2016/2017 ganjil dan 2016/2017 genap di 5 (lima) perguruan tinggi yaitu STMIK Duta Bangsa, AKPARTA Mandala Bhakti, Politeknik Indonusa, APIKES Citra Medika dan AKBID Citra Medika. Hasil validasi terhadap prioritas kegiatan akademik sebesar 94% kegiatan yang dilakukan di perguruan tinggi sesuai dengan urutan kegiatan akademik pada gambar 2.



Gambar 2. Prioritas Kegiatan Akademik

C. Menentukan Kebutuhan Sistem

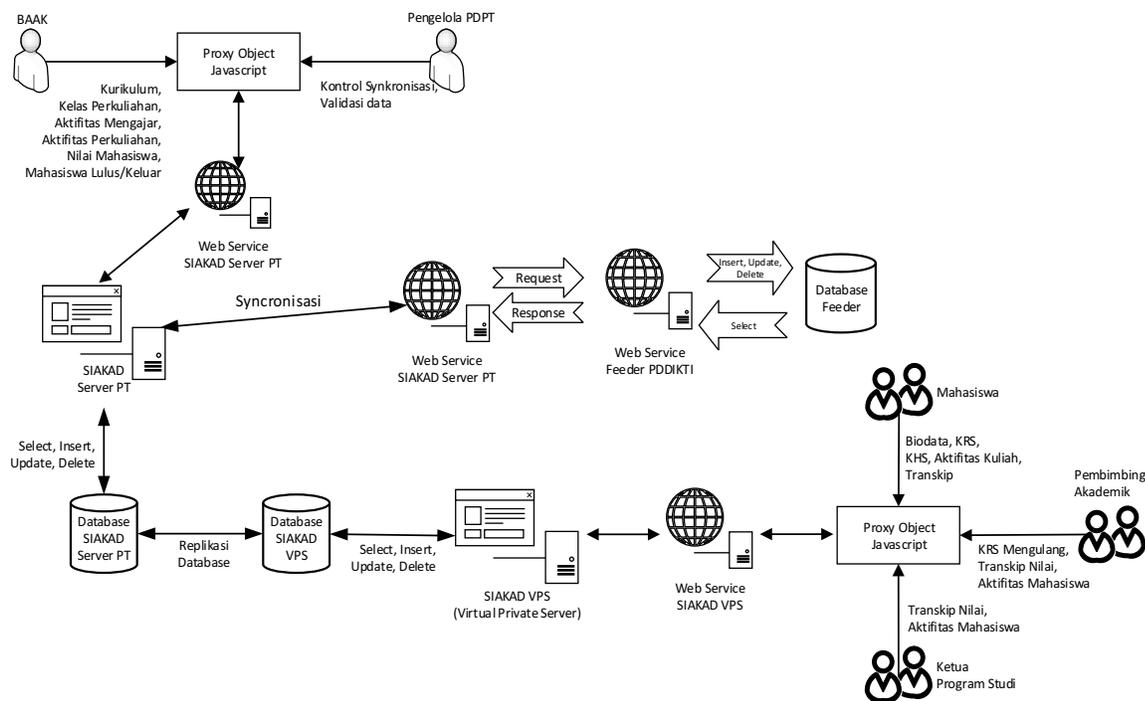
Berdasarkan pada hasil pemetaan dan prioritas kegiatan akademik berikut dapat ditentukan kebutuhan sistem informasi akademik untuk mendukung integrasi data akademik perguruan tinggi dengan Feeder PDDIKTI.

1. Melakukan proses *mapping* struktur data yang sesuai dengan struktur data aplikasi Feeder PDDIKTI.
2. Melakukan pengambilan data dari aplikasi Feeder PDDIKTI
3. Melakukan penyimpanan data ke *database* sistem informasi akademik.
4. Melakukan proses menyimpan, merubah dan menghapus data ke aplikasi Feeder PDDIKTI selama belum dilakukan sinkronisasi ke Forlap.
5. Melakukan pendataan mahasiswa nasional dan mahasiswa perguruan tinggi
6. Melakukan pendataan konversi nilai bagi mahasiswa pindahan atau transfer
7. Melakukan pendataan mata kuliah, kurikulum dan mata kuliah kurikulum
8. Melakukan pendataan kelas dan kelas perkuliahan
9. Melakukan pengisian KRS dan pencetakan KRS
10. Melakukan pendataan dosen pengampu pada kelas perkuliahan atau aktifitas mengajar dosen
11. Melakukan pembuatan absensi kuliah, kartu ujian dan absensi ujian

12. Melakukan pendataan range nilai
13. Melakukan pengisian nilai mahasiswa, cetak KHS dan transkrip nilai
14. Melakukan pendataan aktifitas perkuliahan mahasiswa
15. Melakukan pendataan mahasiswa lulus dan keluar
16. Melakukan laporan mahasiswa, dosen, mata kuliah, kurikulum, kelas perkuliahan, aktifitas mahasiswa, transkrip, aktifitas mengajar dosen, konversi nilai dan mahasiswa lulus/keluar

D. Pemodelan Arsitektur Sistem

Sebagai gambaran model integrasi data akademik perguruan tinggi dengan aplikasi Feeder PDDIKTI, berikut gambar 2 yang menggambarkan arsitektur sistem.



Gambar 2. Arsitektur Integrasi Data Akademik Perguruan Tinggi

Metode *web service* yang penulis gunakan adalah metode *Representative State Transfer (REST)*, dimana *REST server* menyediakan *resources* dan *REST client* mengakses *resources*. *Resource* dapat diakses melalui URIs (*Universal Resource Identifiers*) dan memberikan *response* berupa objek *JavaScript Object Notation (JSON)*.

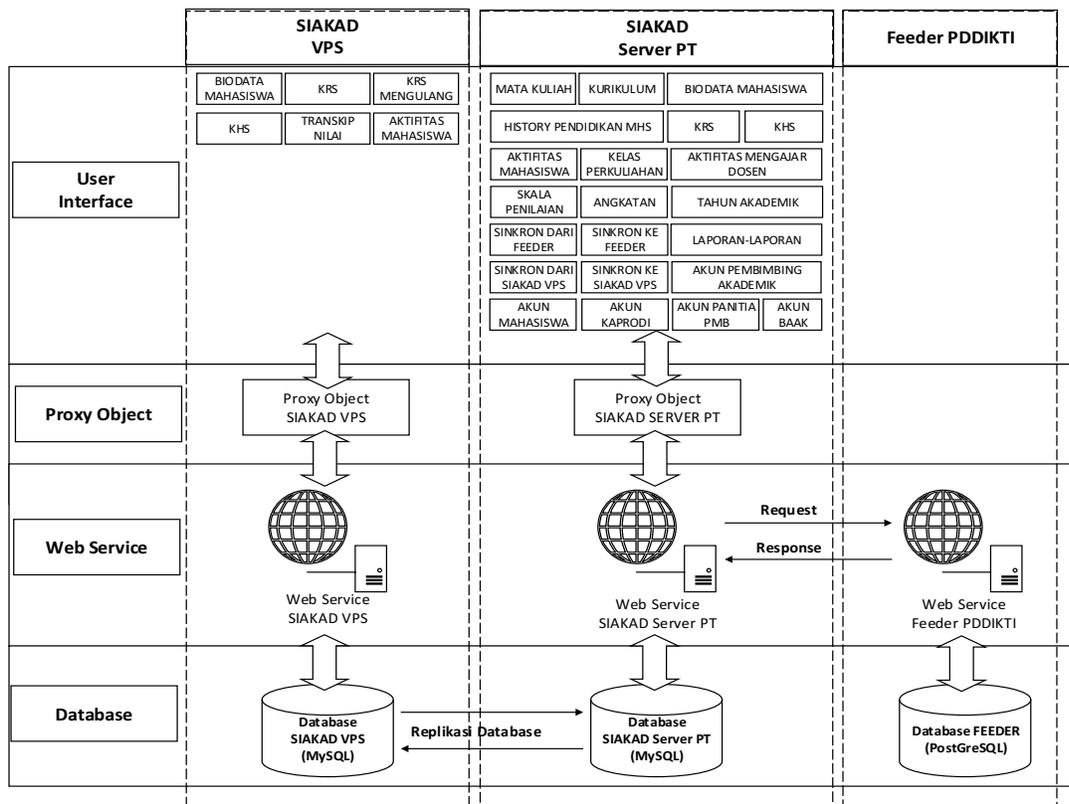
Interaksi antara interface dengan *web service* dijumpai oleh *proxy object* yang dibuat menggunakan bahasa pemrograman Javascript di masing-masing *interface*. Penggunaan *proxy object* akan menggantikan objek asli dari *web service* sehingga ketika *user* melakukan *request* terhadap suatu *method* maka akan diterima oleh *proxy object*, kemudian *proxy object* akan meneruskan *request* ke objek asli yang ada di *web service*. Jika *method* dengan *parameter* yang sama diakses secara berulang maka *proxy object* tidak akan melakukan *request* ke object asli, karena data *response* masih ada di *proxy object*.

Sebagai solusi untuk sistem informasi akademik yang memiliki *user* yang banyak maka penulis memisahkan *database* yang diakses oleh biro administrasi akademik (BAAK) dan pengelola PDPT dengan *database* yang diakses oleh mahasiswa, ketua program studi dan pembimbing akademik. Pemisahan ini diperlukan karena tidak setiap perguruan tinggi memiliki *server* sendiri yang memiliki akses *bandwidth* yang besar. *Server* perguruan tinggi yang

terkoneksi jaringan lokal di kampus akan memiliki kecepatan akses yang baik dari BAAK dan Pengelola PDPT. Begitu pula dengan penggunaan *Virtual Private Server* (VPS) dapat diakses dari jaringan internet dan memiliki kemampuan menangani trafik data yang besar sehingga kecepatan akses dapat terjamin. Untuk pertukaran data antara *server* perguruan tinggi dengan VPS penulis menerapkan teknik replikasi *database*. Replikasi *database* merupakan proses menyalin serta menjaga pertukaran data antar beberapa *database* [2]. Replikasi akan menjaga konsistensi data antara *database* SIAKAD di *server* perguruan tinggi dengan *database* SIAKAD di VPS, karena dengan diterapkannya replikasi akan terjadi sinkronisasi antara kedua *database* secara otomatis. Replikasi yang dapat diterapkan dalam perancangan ini berupa replikasi *master to master*, artinya setiap perubahan yang terjadi di *server* perguruan tinggi akan secara otomatis memicu perubahan di *server* VPS, begitu sebaliknya jika terjadi perubahan di *server* VPS maka akan memicu juga perubahan di *server* perguruan tinggi.

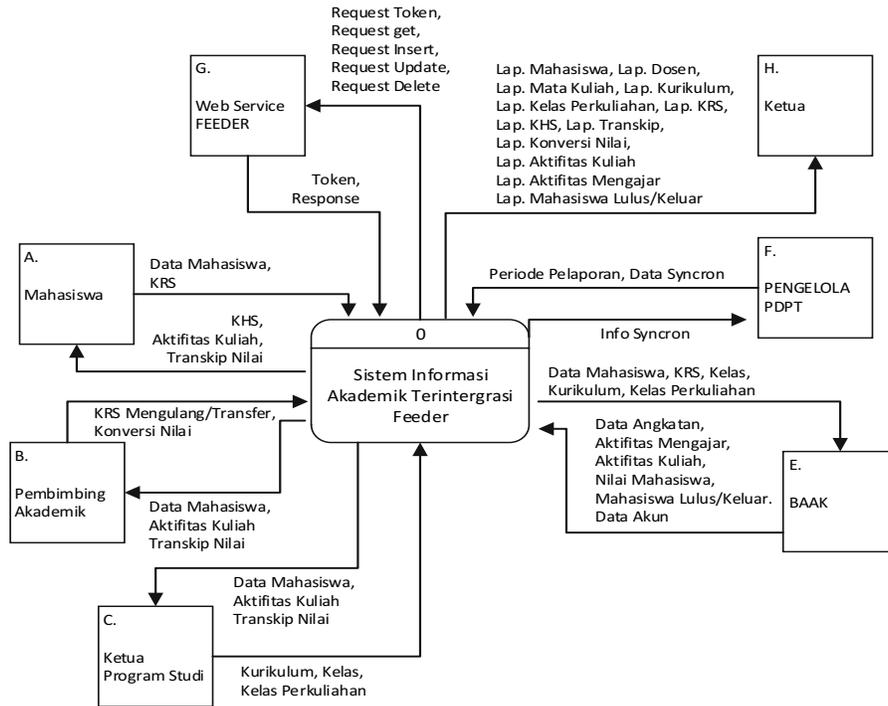
E. Penyusunan Portofolio Sistem

Portofolio sistem menggambarkan model integrasi data akademik perguruan tinggi secara keseluruhan

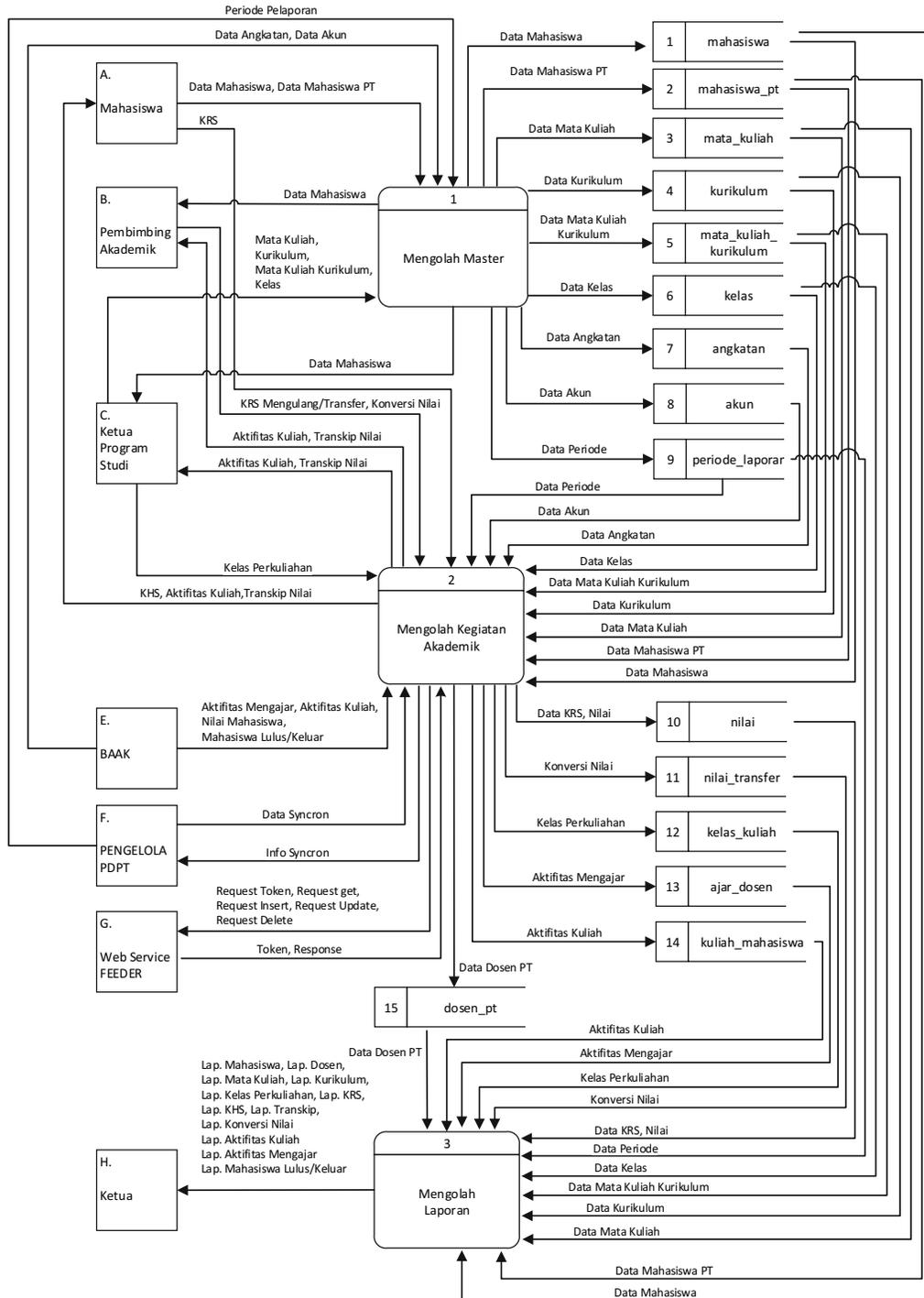


Gambar 3. Portofolio Sistem Informasi Akademik

Arus data pada model integrasi data perguruan tinggi digambarkan dalam diagram alir data berupa diagram konteks.



Gambar 4. Diagram Kontek Sistem Informasi Akademik



Gambar 5. DAD Level 0 Sistem Informasi Akademik

Pada DAD level 0 digambarkan tabel-tabel yang terlibat langsung pada proses transaksi dan sinkronisasi, untuk tabel-tabel yang lain berupa tabel referensi yang isinya sama dengan tabel yang ada di *feeder* PDDIKTI. Berdasarkan

perancangan dan prototipe yang penulis buat berikut contoh potongan kode dan *query* untuk menguji apakah data kartu rencana studi dan nilai dari *database* akademik dapat diintegrasikan ke *Feeder* PDDIKTI.

1. *Query* ambil data dari *database* akademik.

```
$perintah = "SELECT wsia_kelas_kuliah.id_kls,wsia_mahasiswa_pt.id_reg_pd, nilai_angka,
nilai_huruf, nilai_indeks, FROM wsia_kelas_kuliah, wsia_mahasiswa_pt, mahasiswa,
wsia_nilai WHERE wsia_nilai.xid_kls = wsia_kelas_kuliah.xid_kls AND
wsia_nilai.xid_reg_pd = mahasiswa.no_pend AND trim(wsia_mahasiswa_pt.nipd) =
trim(mahasiswa.nim) AND wsia_kelas_kuliah.id_smt = '20161' LIMIT 1 OFFSET 0";

try {
    $db = koneksi();
    $qry = $db->query($perintah);
    $data = $qry->fetchAll(PDO::FETCH_OBJ);
    print_r(json_encode($data));
} catch (PDOException $salah) {
    exit(json_encode($salah->getMessage() ));
}
```

Hasil *output* dari potongan kode diatas adalah:

```
[{
  "id_kls": "441f06c6-eab6-4bdb-8f5b-d3566f0bedbb",
  "id_reg_pd": "92803ea1-3fcc-4561-babd-5e7e8c740fba",
  "nilai_angka": "83.1",
  "nilai_huruf": "A",
  "nilai_indeks": "4.00"
}]
```

2. Struktur data nilai dari *Feeder* PDDIKTI

```
$struktur=proxy()->GetDictionary(token(),"nilai");
$data=$struktur['result'];
print_r(json_encode($data));
```

Hasil *output* dari potongan kode diatas adalah:

```
{"id_kls":
  {"column_name": "id_kls",
  "pk": 1,
  "type": "uuid",
  "not_null": 1,
  "desc": "ID Kelas Kuliah. Web Service: GetRecordset: kelas_kuliah"},
"id_reg_pd":
  {"column_name": "id_reg_pd",
  "pk": 1,
  "type": "uuid",
  "not_null": 1,
  "desc": "Web Service: GetRecordset: mahasiswa_pt"},
"nilai_angka":
  {"column_name": "nilai_angka",
  "type": "numeric(4,1)",
  "desc": null},
```

```
"nilai_huruf":
  {"column_name": "nilai_huruf",
   "type": "character(3)",
   "desc": null},
"nilai_indeks":
  {"column_name": "nilai_indeks",
   "type": "numeric(4,2)",
   "desc": null}
}
```

3. Perintah memasukkan data ke *Feeder* PDDIKTI

```
$insert=proxy()->InsertRecordSet(token(),"nilai",json_encode($data));
$data=$insert['result'];
print_r(json_encode($data));
```

Hasil *output* dari potongan kode diatas adalah:

```
[{
  "error_code": "0",
  "error_desc": "",
  "id_reg_pd": "92803ea1-3fcc-4561-babd-5e7e8c740fba",
  "id_kls": "441f06c6-eab6-4bdb-8f5b-d3566f0bedbb "
}]
```

error_code bernilai 0 dan error_desc kosong menunjukkan bahwa proses memasukkan data ke *Feeder* PDDIKTI berhasil.

IV. KESIMPULAN

Tahap utama dalam model integrasi data akademik perguruan tinggi yaitu mengidentifikasi *Method Web Service* dan struktur tabel *database Feeder* PDDIKTI. Hasil identifikasi digunakan untuk membentuk struktur tabel pada sistem informasi akademik di perguruan tinggi, sehingga dapat dilakukan migrasi data awal dari *feeder* PDDIKTI ke *database* sistem informasi akademik di perguruan tinggi. Penggunaan *web service* dan *proxy object* dapat menjadi alternatif dalam pengembangan sistem informasi akademik yang terintegrasi dengan *feeder* PDDIKTI. Berdasarkan pengujian dengan menggunakan prototipe kode dan *query* yang penulis buat dapat disimpulkan bahwa model yang dirancang mendukung dalam proses sinkronisasi pada data mahasiswa nasional, mahasiswa perguruan tinggi, konversi mahasiswa transfer atau pindah, mata kuliah, kurikulum, mata kuliah kurikulum, kelas perkuliahan, kartu rencana studi, aktifitas mengajar dosen, pengaturan range nilai, nilai mahasiswa, aktifitas kuliah mahasiswa dan mahasiswa lulus atau keluar.

V. UCAPAN TERIMA KASIH

Terima kasih kepada (1) STMIK Duta Bangsa Surakarta, (2) AKBID Citra Medika Surakarta, (3) APIKES Citra Medika Surakarta, (4) Politeknik Indonusa Surakarta dan (5) AKPARTA Mandala Bhakti Surakarta.

VI. REFERENSI

- [1] Bieberstein, dkk. (2008). *Executing SOA: A Practical Guide for The Service Oriented Architecture*. IBM Press, Indiana
- [2] Connolly, dkk. (2005). *Database System : A Practical Approach to Design, Implementation and Management*. Edisi: 4. Pearson Education Limited. England
- [3] Juric, M.B., dkk. (2007). *SOA Approach to Integration*. Packt Publishing, Birmingham, B27 6PA, UK.
- [4] Kapojos, F, dkk. (2012). *Implementasi Service-Oriented Architecture dengan Web Service untuk Aplikasi Informasi Akademik*. Manado: Jurnal Teknik Elektro dan Komputer Vol 1, No 1 2012
- [5] Ladan Mohamad Ibrahim. Ph.D. (2011). *Web Services Metrics A Survey and A Classification*. International Conference on Network and Electronics Engineering IPCSIT vol.11 (2011): IACSIT Press, Singapore
- [6] Setiawan, Ridwan. (2015). *Perancangan Arsitektur Enterprise Untuk Perguruan Tinggi Swasta Menggunakan TogafADM*. Garut : Jurnal Algoritma ISSN : 2302-7339 Vol. 12 No. 1 2015

- [7] Yogiswara, dkk. (2014). *Kinerja Web Service pada Proses Integrasi Data*. Jurnal EECCIS Vol. 1, No. 1, Juni 2014
- [8] <http://forlap.dikti.go.id/> diakses 28 Desember 2016
- [9] <http://www.dofactory.com/javascript/proxy-design-pattern> diakses 15 Februari 2017