



## IMPLEMENTASI *WEB SERVICE* UNTUK KOMUNIKASI DATA PADA APLIKASI PENJADWALAN UJIAN TUGAS AKHIR

Waode Nur Lailah<sup>1</sup>, LM.Fid Aksara<sup>\*2</sup>, Natalis Ransi<sup>\*3</sup>

<sup>1,\*2,\*3</sup>Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari  
e-mail : <sup>1</sup>nurla3392@gmail.com, <sup>\*2</sup>fid.aksara@uho.ac.id, <sup>\*3</sup>natalis.ransi@uho.ac.id

### Abstrak

Peranan *web service* diperlukan sebagai sebuah media untuk *sharing* data maupun fungsionalitas antar proses agar suatu aplikasi dapat terhubung dengan aplikasi lain tanpa tergantung dengan bahasa pemrograman maupun sistem operasi. Dengan menggunakan *web service* jadwal ujian Tugas Akhir tidak hanya diketahui oleh mahasiswa yang akan melakukan ujian sidang namun bisa diketahui juga oleh seluruh mahasiswa dimana jadwal ujian akan dikirimkan kepada penyedia *web service* kemudian mesin *web service* akan bekerja untuk menampilkan jadwal ujian pada beberapa *website* yang telah disediakan dan dengan menggunakan algoritma genetika jadwal ujian langsung dapat ditentukan secara otomatis. Disinilah peranan *web service* dalam melakukan proses pertukaran data. Dalam penelitian ini penyusunan dari *web services* yang akan digunakan adalah SOAP (*Simple Object Acces Protocol*) karena SOAP merupakan protokol yang sudah distandarisasi oleh W3C (*World Wide Web Consortium*), serta WSDL (*Web service Definition Language*) yang merupakan dokumentasi variabel yang digunakan dalam menerjemahkan sebuah bahasa di dalam *web services*.

Hasil yang didapatkan pada penelitian ini bahwa penerapan *web service* dapat berjalan dengan baik. Keberhasilan dapat dilihat dari hasil pengujian sistem dimana jadwal ujian Tugas akhir dapat tampil di beberapa *website* yang berbeda dan dapat mengubah data yang ada pada sistem yang berjalan tanpa masuk ke dalam sistem tersebut.

**Kata Kunci**—*Web Service*, SOAP, Algoritma Genetika

### Abstract

*The role of web service is needed as a medium for sharing data and inter-process functionality so that an application can connect with other applications without depending on the programming language or operating system. By using the web service Final Project exam schedule is not only known by students who will do the trial examination but can also be known by all students where the exam schedule will be sent to the web service provider then the web service engine will work to display the exam schedule on several websites that have been provided and by using genetic algorithms direct exam schedules can be determined automatically. This is where the role of web service is in carrying out the data exchange process. In this study, the technology that will be used is SOAP (Simple Object Access Protocol) because SOAP is a standardized protocol by W3C (World Wide Web Consortium), besides that it is also used Genetic Algorithm to automatically set exam schedules.*

*The results obtained in this study that the application of web services can run well. Success can be seen from the results of testing the system where the exam schedule The final task can appear on several different websites and can change the data that is on the system that runs without entering into the system.*

**Keywords** - *Web Service*, SOAP, Genetic Algorithm

## 1. PENDAHULUAN

Pemanfaatan teknologi dan informasi telah menjadi sebuah kebutuhan utama yang tidak bisa diabaikan begitu saja.

Dengan adanya teknologi informasi menyebabkan proses penyebaran dan petukaran informasi dapat dilakukan dengan cepat secara global tanpa adanya batasan waktu [1]. Tugas akhir adalah sarana untuk mengetahui kemampuan seorang mahasiswa apakah sudah menguasai ilmu yang diberikan dan layak untuk mengabdikan di masyarakat sesuai kompetensi yang telah diajarkan oleh kampus. Tugas akhir atau sering disingkat TA merupakan langkah awal untuk menghadapi dunia kerja yang akan dihadapi, dengan adanya Tugas Akhir mahasiswa dapat mempersiapkan diri untuk menyelesaikan proyek-proyek dimasa kerja nanti

Teknik Informatika merupakan salah satu jurusan yang ada di Fakultas Teknik, Universitas Halu Oleo. Seperti pada umumnya, untuk melakukan Tugas Akhir ada beberapa langkah yang harus di tempuh sebelum melakukan ujian Tugas Akhir diantaranya yaitu, menentukan judul yang telah disetujui oleh dosen pembimbing, mengajukan judul kepada Ketua jurusan kemudian penyusunan Tugas akhir dan penentuan jadwal ujian. Pada Jurusan Teknik Informatika jadwal ujian Tugas akhir hanya di ketahui oleh pihak jurusan, dosen pembimbing, dan penguji serta mahasiswa yang akan melaksanakan ujian. Padahal seharusnya hal tersebut bisa diketahui oleh seluruh mahasiswa Teknik Informatika guna untuk membantu mahasiswa mengisi kartu kontrol ujian yang telah disediakan oleh pihak jurusan sebelum melakukan ujian Tugas Akhir serta mengetahui bagaimana jalannya ujian Tugas Akhir.

Berdasarkan hal tersebut perlu digunakan teknologi *web service* yang dapat membantu memberikan solusi terhadap permasalahan yang ada. Peranan *web service* ini diperlukan sebagai sebuah media untuk *sharing* data maupun fungsionalitas antar proses agar suatu aplikasi dapat terhubung dengan aplikasi lain tanpa tergantung dengan bahasa pemrograman maupun sistem operasi. Dengan menggunakan *web service* jadwal ujian Tugas Akhir tidak hanya diketahui oleh mahasiswa yang akan melakukan ujian sidang

namun bisa diketahui juga oleh seluruh mahasiswa dimana jadwal ujian akan dikirimkan kepada penyedia *web service* kemudian mesin *web service* akan bekerja untuk menampilkan jadwal ujian pada beberapa *website* yang telah disediakan dan dengan menggunakan algoritma genetika jadwal ujian langsung bisa ditentukan secara otomatis. Disinilah peranan *web service* dalam melakukan proses pertukaran data. Dalam penelitian ini teknologi yang akan digunakan adalah SOAP (*Simple Object Acces Protocol*) karena SOAP merupakan protokol yang sudah distandarisasi oleh W3C (*World Wide Web Consortium*), selain itu juga digunakan Algoritma Genetika untuk mengatur jadwal ujian secara otomatis.

## 2. METODE PENELITIAN

### 2.1 Web Service

*Web service* adalah sebuah *software* yang dirancang untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Web service* secara teknis memiliki mekanisme interaksi antar sistem sebagai penunjang interoperabilitas, baik berupa agregasi (pengumpulan) maupun sindikasi (penyatuan). *Web service* memiliki layanan terbuka untuk kepentingan integrasi data dan kolaborasi informasi yang bisa diakses melalui internet oleh berbagai pihak menggunakan teknologi yang dimiliki oleh masing-masing pengguna. Sekalipun mirip dengan *Application Programming Interface* (API) berbasis web, *web service* lebih unggul karena dapat dipanggil dari jarak jauh melalui internet. Pemanggilan *web service* bisa menggunakan bahasa pemrograman apa saja dan dalam *platform* apa saja, sementara API hanya bisa digunakan dalam *platform* tertentu.

*Web service* dapat dipahami sebagai *Remote Procedure Call* (RPC) yang mampu memproses fungsi-fungsi yang didefinisikan pada sebuah aplikasi web dan mengekspos sebuah API atau *User Interface* (UI) melalui web [2]. Kelebihan *web service* adalah:

1. *Language independent*, dapat diakses dan dibangun oleh bahasa pemrograman apapun.
2. *Lintas platform*, tetap memungkinkan terjadinya pertukaran data, walaupun menggunakan perangkat-perangkat dengan sistem operasi yang berbeda.

3. Jembatan penghubung dengan *database*, *web service* dapat diibaratkan sebagai “jembatan” penghubung antara *database* dengan aplikasi tanpa memerlukan adanya *driver database*. Selain itu, juga tanpa perlu mengetahui *database* yang digunakan oleh *server* dan bagaimana bentuk struktur *database* tersebut apabila ingin mengaksesnya. Aplikasi hanya cukup mengetahui berbagai macam *method* dan fungsi yang telah disediakan oleh *web service*, sehingga dapat melakukan pemanfaatan terhadap fasilitasnya.
4. Penggunaan kembali komponen aplikasi, aplikasi-aplikasi yang berbeda dapat menggunakan sebuah fungsi yang sama.
5. Proses pertukaran data menjadi semakin lebih mudah dan cepat tanpa harus menyesuaikan aplikasi, *database*, dan *platform* yang digunakan.

## 2.2 SOAP (*Simple Object Acces Protocol*)

SOAP merupakan protokol komunikasi yang ringan berbasis *eXtensible Markup Language* (XML). Pesan SOAP berbentuk sekumpulan XML *Schema* yang mendefinisikan format untuk mentransmisikan pesan XML melalui jaringan, termasuk tipe data dan cara menstrukturkan pesan secara tepat sehingga dapat mudah dipahami oleh server atau *end-point* lainnya.

Pesan SOAP terdiri dari 3 bagian, yaitu :

1. *Envelope*, suatu kerangka yang mendefinisikan apa yang ada dalam pesan dan bagaimana pesan harus diproses serta menunjukkan resipien dari pesan tersebut
2. *Header*, berisi informasi yang berhubungan dengan keamanan dan *routing*. Keberadaan Header dalam SOAP bersifat *optional*
3. *Body*, berisi data di *mark-up* sebagai XML dan dimasukkan dalam format yang spesifik yang didefinisikan dalam XML *Schema*.

SOAP adalah sebuah XML *based markup language* untuk pergantian pesan diantara aplikasi-aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data *object* didalam *network*. SOAP mendefinisikan empat aspek didalam komunikasi yaitu *Message envelope*, *Encoding*, *RPC call convention*, dan bagaimana menyatukan sebuah *message* dan menggunakan *MIME-multipart*.

SOAP menspesifikan secara jelas bagaimana cara untuk *meng-encode header* HTTP dan *file XML* sehingga program pada suatu komputer dapat memanggil program pada komputer lain dan mengirimkan informasi, dan bagaimana program yang dipanggil memberikan tanggapan. SOAP adalah protokol ringan yang ditunjukkan untuk pertukaran informasi struktur pada lingkup desentralisasi, dan terdistribusi. SOAP menggunakan teknologi XML untuk mendefinisikan rangka kerja pemesanan terekstrensi di mana menyediakan konstruksi pesan yang dipertukarkan pada protokol berbeda. Rangka kerja dirancang bebas dari model pemrograman dan spesifikasi implementik semantik [1].

## 2.3 Algoritma Genetika

Algoritma Genetika merupakan salah satu metode untuk menentukan optimalisasi atas dasar Teori Darwin. Langkah prosedur algoritma ini diawali dengan menentukan suatu set solusi potensial dan melakukan perubahan dengan beberapa perulangan (iterasi) dengan algoritma genetika untuk menghasilkan solusi terbaik. Set solusi potensial ini ditetapkan diawal dan disebut dengan kromosom. Kromosom ini dibentuk secara random berupa susunan angka *binary* yang di *generate* dan dipilih. Keseluruhan set dari kromosom yang diobservasi mewakili suatu populasi [3]. Kromosom-kromosom berevolusi beberapa kali tahapan iterasi yang disebut dengan generasi. Generasi baru (*offsprings*) *digenerate* dengan teknik kawin silang (*crossover*) dan mutasi (*mutation*). *Crossover* meliputi pemisahan atau pemecahan (*splitting*) dua kromosom, kemudian mengkombinasikan setengah bagian dari masing-masing kromosom dengan pasangan-pasangan lainnya. Mutasi meliputi pertukaran (*flipping*) satu bit (bagian) dari kromosom dengan satu bagian lain dari kromosom lain yang menjadi pasangannya. Kromosom-kromosom tersebut kemudian berevolusi dengan suatu kriteria kesesuaian (*fitness*) yang telah ditetapkan, hasil yang terbaik akan dipilih sedangkan yang lainnya diabaikan. Proses ini dilakukan secara berulang-ulang sampai menemukan suatu kromosom yang memiliki kesesuaian terbaik (*best fitness*) untuk dijadikan sebagai solusi terbaik dari suatu masalah.

Parameter kontrol genetika diperlukan untuk mengendalikan operator-operator seleksi. Pemilihan parameter genetika menentukan penampilan kinerja algoritma genetika dalam memecahkan masalah [4]. Ada dua parameter dasar dari algoritma genetika, yaitu Probabilitas *Crossover* (Pc) dan Probabilitas Mutasi (Pm). Probabilitas *Crossover* (Pc) Probabilitas *Crossover* akan mengendalikan operator *crossover* dalam setiap generasi dalam populasi yang mengalami *crossover*. Semakin besar nilai probabilitas *crossover*, akan semakin cepat struktur individu baru terbentuk ke dalam populasi. Apabila nilai probabilitas *crossover* terlalu besar, maka individu yang merupakan kandidat solusi terbaik mungkin akan dapat hilang lebih cepat pada generasi selanjutnya. Nilai probabilitas *crossover* yang disarankan adalah berkisar antara 80% - 95%, sedangkan Probabilitas Mutasi (Pm) akan mengendalikan operator mutasi pada setiap generasi dengan peluang mutasi yang digunakan lebih kecil daripada peluang *crossover*. Pada seleksi alam murni, mutasi jarang sekali muncul, sehingga operator mutasi pada algoritma genetik tidak selalu terjadi. Nilai Probabilitas Mutasi yang disarankan kecil antara 0.5% - 1%.

#### 2.4 NuSOAP

NuSOAP adalah sebuah kumpulan *class-class* PHP yang memungkinkan *user* untuk mengirim dan menerima pesan SOAP melalui protokol HTTP. NuSOAP didistribusikan oleh NuSphere Corporation (<http://www.nusphere.com>) sebagai *open source toolkit* di bawah lisensi GNU/GPL.

NuSOAP merupakan *toolkit web service* berbasis komponen. NuSOAP memiliki sebuah *class* dasar yang menyediakan *method* seperti serialisasi variabel dan pemaketan SOAP-Envelope. Interaksi *web service* dilakukan dengan *class client* yang disebut dengan *class "soapclient"* dan *class server* yang disebut dengan *class "soap-server"*. *Class-class* ini mengizinkan *user* untuk melakukan proses pengiriman dan penerimaan pesan-pesan SOAP dengan bantuan beberapa *class-class* pendukung lainnya untuk melengkapi proses tersebut

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Implementasi

Implementasi rancangan antarmuka, yaitu:

##### 1. *Interface* Aplikasi

Pada *interface* Aplikasi diantaranya :

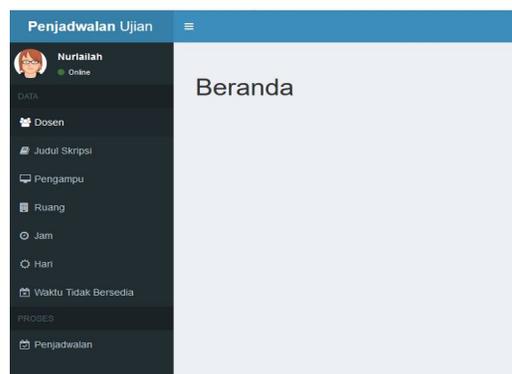
##### a) Halaman *Login*

Pada halaman ini merupakan form yang harus diisi oleh *user* untuk dapat mengakses menu dan fitur yang ada di dalam *web*. Pada *form* ini *user* diminta untuk memasukkan *username* dan *password* dapat dilihat pada Gambar 1.

Gambar 1 Halaman *Login*

##### b) Halaman Beranda

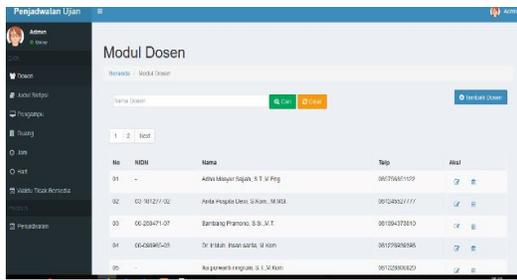
Halaman beranda merupakan halaman setelah step *login* berhasil. Pada halaman berisi beberapa menu seperti Dosen, Judul Skripsi, Pengampu, Ruang, Jam, Hari, Waktu tidak tersedia dan Tampilan beranda, dapat dilihat pada Gambar 2.



Gambar 2 Halaman Beranda

##### c) Halaman Dosen

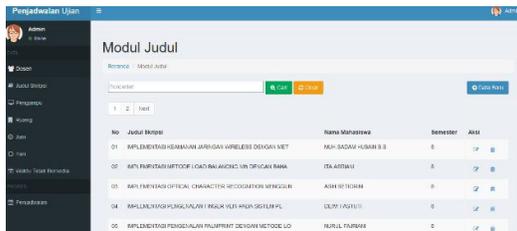
Halaman ini berfungsi untuk melihat, menambah, mengubah dan menghapus data Dosen, dapat dilihat pada Gambar 3.



Gambar 3 Halaman Dosen

d) Halaman Menu Judul

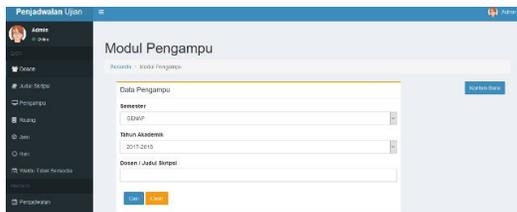
Halaman ini berfungsi untuk melihat, menambah, mengubah dan menghapus Judul, dapat dilihat pada Gambar 4.



Gambar 4 Halaman Menu Judul

e) Halaman Menu Pengampu

Halaman ini berfungsi untuk mencari data yang telah dimasukkan sebelumnya *user* yang lain, dapat dilihat pada Gambar 5.



Gambar 5 Halaman Pengampu

f) Halaman Edit Ruang

Pada halaman ini digunakan untuk melihat ruang ujian yang tersedia, dapat dilihat pada Gambar 6.



Gambar 6 Halaman Ruang

g) Halaman Waktu tidak bersedia

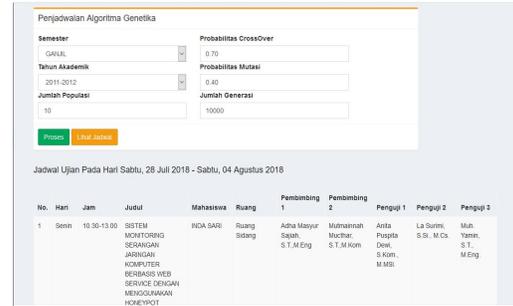
Pada halaman ini berisi nama dosen serta pilihan waktu dosen jika tidak bersedia, dapat dilihat pada Gambar 7.



Gambar 7 Halaman Admin

h) Halaman Penjadwalan

Pada halaman ini berisi jadwal ujian dan parameter dalam menentukan jadwal secara otomatis diantaranya ada Jumlah populasi, Probabilitas *Crossover*, Probabilitas Mutasi, serta jumlah generasi, serta tombol Lihat Jadwal yang berfungsi untuk pertukaran data serta menampilkan jadwal ujian pada *website* lain yang telah disediakan, dapat dilihat pada Gambar 8.



Gambar 8 Halaman Penjadwalan

3.2 Perhitungan Algoritma genetika

Misal terdapat 6 kombinasi sehingga sebuah kromosom dapat dibentuk pola sebagai berikut :

1. (D1,J2,M1)
2. (D2,J4,M3)
3. (D3,J1,M4)
4. (D4,J5,M2)
5. (D5,J3,M6)
6. (D6,J6,M5)

Pembuatan kromosom dilakukan secara acak untuk subgen jadwal yaitu :

(1,2,1), (2,4,3), (3,1,4), (4,5,2), (5,3,6), (6,6,5).

Yang harus diperhatikan dalam mengacak jadwal yang dipasangkan dengan dosen menguji tidak boleh digunakan dua kali karena satu jadwal berisi hari, jam, dan ruangan. Tidak mungkin hari, jam, dan ruangan yang sama digunakan dua ujian sekaligus.

Angka acak pertama R[1] adalah lebih besar dari C[1] dan kurang dari C[2] maka pilih Chromosom [1] sebagai *chromosome* pada populasi baru, dari bilangan acak yang telah dibangkitkan diatas maka populasi *chromosome* baru hasil proses seleksi

Jadwal

- Chromosom [1] = Chromosom [1]
- Chromosom [2] = Chromosom [1]
- Chromosom [3] = Chromosom [3]
- Chromosom [4] = Chromosom [4]
- Chromosom [5] = Chromosom [5]
- Chromosom [6] = Chromosom [1]

Jadi, berdasarkan hasil seleksi tersebut Chromosom [1] adalah (1,2,1) atau (D1.J2,M1)

### 3.3 Pengujian Sistem

Pengujian merupakan bagian tahap dari proses pengembangan perangkat lunak pengujian ini bertujuan untuk mengetahui kualitas pada perangkat lunak telah dibuat. Pengujian ini dilakukan untuk mengetahui apakah program yang dibuat telah sesuai seperti rencana dan rancangan sebelumnya. Pengujian dalam sistem ini, Penulis menggunakan metode pengujian *black box*.

#### a. Pengujian Black Box

Pada pengujian *Black Box* ini yang menjadi fokus adalah pada bagian fungsional aplikasi atau perangkat lunaknya. Pengujian ini bertujuan untuk menunjukkan fungsi aplikasi tentang mekanismenya. Sehingga diketahui data yang dimasukkan dapat berjalan sebagaimana mestinya dan informasi yang tersimpan dapat terjaga validannya.

#### b. Hasil Pengujian Black Box

Tabel 1 menunjukkan Hasil Pengujian *Black Box*.

Tabel 1 Hasil Pengujian *Blackbox*

No	Objek Pengujian	Hasil yang diharapkan	Hasil
1	Button Cari pada menu dosen	Mencari data dosen yang telah dimasukan ke dalam database	✓
2	Button Clear pada menu dosen	Menghapus hasil pencarian	✓
3	Button Tambah dosen	Menambahkan data dosen ke dalam database	✓
4	Button Simpan Data pada halaman Tambah Dosen	Menyimpan data yang telah ditambahkan ke dalam database	✓

5	Button Cari pada menu Judul	Mencari data Judul yang telah dimasukan ke dalam database	✓
6	Button Clear pada menu Judul	Menghapus hasil pencarian	✓
7	Button Tambah Judul	Menambahkan data judul ke dalam database	✓
8	Button Simpan Data pada halaman Tambah Judul	Menyimpan data yang telah ditambahkan ke dalam database	✓
9	Button Cari pada menu Pengampu	Mencari data dosen yang telah dimasukan ke dalam database	✓
10	Button Clear pada menu pengampu	Menghapus hasil pencarian	✓
11	Button Konten baru pada menu pengampu	Menyimpan data yang telah ditambahkan kedalam database	✓
12	Button proses pada menu penjadwalan	Memproses semua data yang telah ditambahkan dan menghasilkan jadwal ujian	✓
13	Button Lihat Jadwal	Menampilkan jadwal ke website lain (yang telah disediakan)	✓

### 3.3 Pengujian Web Services

Berikut ini adalah tampilan jadwal yang muncul pada *website* lain yang telah disediakan, dapat dilihat pada Gambar 9.

Jadwal Ujian Pada Tanggal Minggu, 19 Agustus 2018 - Minggu, 26 Agustus 2018									
No Hari	Jam	Judul	Mahasiswa	Ruang	Pembimbing 1	Pembimbing 2	Penguji 1	Penguji 2	Penguji 3
1	Senin 12.10-14.40	Modul Tambah Peserta Ujian Beradasasi asin Modul Jutub Tambah Data		Ruang Sidang	Dr. Iti Muh	Anita Puspa	Jumadi	Muh Yamin	Adha Mesyur
2	Selasa 10.30-13.00	PENERAPAN XML, WEB SERVICE, MENGGUNAKAN SIMPLE OBJECT ADDRESS PROTOCOL (SOAP) DALAM MENAMPILKAN JADWAL UJIAN TUGAS AKHIR PADA WEBSITE TEKNIK INFORMATIKA	WADOE	LAB	L. M. Fid	Natols Rans	Subardin	La Surmi	Nur Fajrah
3	Rabu 13.00-15.30	RANCANGAN BANGUNAN DAN IMPLEMENTASI SERVER VOIP MENGGUNAKAN ASTERISK DAN X.LITE PADA JURISAN TEKNIK INFORMATIKA UNIV. HALLI OLEO	TRY ARDANA	Ruang Sidang	Adha Mesyur	Bambang	La Surmi	Jumadi Nang Subardin	
4	Kamis 08.00-10.30	tesssssss		LAB	Adha Mesyur	Jayanti	Anita Puspa	Natols Rans	Lado

Gambar 9 Jadwal Ujian pada Beberapa *Website* yang Berbeda

Berikut adalah WSDL (*Web services definition language*) yang merupakan dokumentasi variabel dimana fungsinya untuk mendefinisikan *Web Services* dan menggambarkan bagaimana cara untuk mengakses *web services* tersebut, dapat dilihat pada Gambar 10.



Gambar 10 Informasi *Web Service* Jadwal Ujian Beserta *Method Readall* WSDL

Gambar 11 menunjukkan Pesan *Request* SOAP Jadwal.



Gambar 11 Pesan *Request* SOAP Jadwal

Gambar 12 menunjukkan Pesan *Response* SOAP Jadwal.



Gambar 12 Pesan *Response* SOAP Jadwal

#### 4. KESIMPULAN

Berdasarkan penelitian dan hasil pengujian yang dilakukan pada penelitian ini, maka dapat disimpulkan:

1. Penerapan XML *Web Services* SOAP dalam menampilkan jadwal ujian Tugas Akhir Teknik Informatika pada *website* lain dapat berjalan dengan baik.
2. Penggunaan Algoritma genetika sangat cocok untuk menangani masalah penjadwalan.
3. Semua menu yang ada pada sistem dapat berjalan sesuai fungsinya masing masing.

#### 5. SARAN

Saran yang dapat diberikan untuk pengembangan lebih lanjut terhadap penelitian ini adalah

1. Untuk menu tidak bersedia dapat disesuaikan dengan jadwal mengajar dosen yang telah ada.
2. Jadwal ujian bisa langsung dikirim ke email masing-masing dosen.

#### DAFTAR PUSTAKA

- [1] E. Sutanta and K. Mustofa, "Identifying The Needs of *Web service* to Data Synchronization Between Information Systems as E-Government Ecosystem at Bantul-Yogyakarta," *Tek. Inform. - STMIK Bandung*, Vol. 2, No. 3, pp. 20–26, 2012.
- [2] A. G. Y. Paramartha, G. K. Suryaningsih, and K. Y. E. Aryanto, "Implementasi *Web service* pada Sistem Pengideksan dan Pencarian Dokumen Tugas Akhir, Skripsi, dan Praktik Kerja Lapangan," *J. Sains dan Teknol.*, Vol. 5, No. 2, pp. 775–784, 2016.
- [3] E. Suhartono, "Optimasi Penjadwalan Mata Kuliah dengan Algoritma Genetika (Studi Kasus di AMIK JTC Semarang)," *Infokam*, Vol. 11, No. 5, pp. 132–146, 2015.
- [4] A. S. Riyadi, E. Retnandi, and A. Deddy, "Perancangan sistem informasi Berbasis *Website* Subsistem Guru di Sekolah Pesantren Persatuan Islam 99 Rancabango," *J. Algoritm.*, Vol. 9, No. 40, pp. 1–11, 2012.

