

IMPLEMENTASI WEB SERVICE WAKTU SHALAT BERBASIS TEKNOLOGI PAAS CLOUD COMPUTING

Wagito

Informatika, UTDI Yogyakarta
Jl. Raya Janti Karang Jambe No. 143 Yogyakarta
email: wagito@utdi.ac.id

Abstrak

Layanan web service menjadi pilihan untuk memudahkan penyebaran informasi melalui Internet. Layanan yang bisa diberikan bisa bermacam-macam sesuai kebutuhan, Salah satu layanan yang bisa diberikan adalah layanan waktu shalat. Layanan ini dapat dimanfaatkan oleh para pengembang perangkat lunak yang tanpa harus memikirkan bagaimana cara menghitung waktu shalat. Penelitian bertujuan untuk menerapkan layanan web service pada layanan cloud computing berbasis PaaS, menyusun web service untuk memberi layanan waktu shalat menggunakan beberapa metode perhitungan dan kriteria, serta membuat contoh penggunaan layanan web service waktu shalat pada aplikasi berbasis web. Layanan web service dapat diterapkan pada server PaaS cloud computing. Pada penelitian ini digunakan perangkat lunak praytimes untuk menghitung waktu shalat yang salah satunya ditulis dalam bentuk skrip PHP. Dengan demikian diperlukan layanan PaaS yang dapat menjalankan skrip PHP. Layanan web service ini, selanjutnya dipakai oleh pengembang perangkat lunak. Pada penelitian dikembangkan perangkat lunak sederhana yang memanfaatkan layanan web service waktu shalat. Penelitian berhasil menerapkan layanan web service waktu shalat yang diterapkan pada layanan PaaS cloud computing Heroku. Penelitian berhasil menyusun layanan web service waktu shalat berdasarkan beberapa metode dan kriteria dari PrayTime.org. Penelitian berhasil menggunakan layanan web service untuk aplikasi waktu shalat berbasis web.

Keywords: *web service, waktu shalat, cloud computing*

1. PENDAHULUAN

Web service merupakan aplikasi yang berisi sekumpulan basis data dan perangkat lunak atau bagian dari program perangkat lunak yang diakses melalui Internet menggunakan peranti perantara tertentu. Melalui *web service*, memungkinkan pengguna untuk mengatasi permasalahan berupa interoperabilitas dan mengintegrasikan sistem berbeda.

Pada umumnya, *web service* memiliki ciri khusus berupa URL layaknya *web*. Yang membuat berbeda adalah interaksi yang diberikan oleh *web service* itu sendiri. URL pada *web service* hanya mengandung sekumpulan informasi, perintah, dan konfigurasi (sintaksis yang berguna untuk membangun fungsi tertentu dari aplikasi).

Web service mampu menukar data tanpa memandang sumber *database*, bahasa yang digunakan, dan pada *platform* apa data tersebut dipakai. Kemampuan itulah yang memungkinkan *web service* menjadi jembatan penghubung untuk berbagai sistem.

Layanan yang diberikan oleh *web service* dapat berupa layanan yang bersifat privat maupun publik. Layanan privat hanya dipakai untuk melayani kebutuhan data pada lingkup yang privat. Sedangkan *web service* yang bersifat publik digunakan untuk layanan keperluan publik. Layanan publik memberikan data yang dipakai umum, salah satunya adalah waktu shalat.

Perhitungan waktu shalat memerlukan keahlian khusus terutama berkaitan dengan ilmu falak dan hukum-hukum agama Islam. Namun demikian sudah ada pengembang yang mengimplementasikan perhitungan waktu shalat, salah satunya yang dipublikasikan pada situs <http://praytimes.org>. Perangkat lunak yang dikembangkan bisa dimanfaatkan untuk kepentingan publik dengan menjadikannya layanan *web service*.

Pada masa sekarang juga berkembang apa yang disebut *cloud computing*. Dinamakan *cloud computing* karena informasi yang diakses dari jarak jauh di “awan” atau ruangan virtual. Perusahaan-perusahaan yang menyediakan layanan *cloud*, memungkinkan para

penggunanya menyimpan *file* dan aplikasi dari *server* jarak jauh. Pengguna juga bisa melakukan akses asalkan ada sambungan Internet. Ini berarti, pengguna tidak perlu berada di tempat tertentu untuk mendapatkan akses.

Cloud computing bisa bersifat publik atau privat. *Cloud* publik menyediakan layanannya secara publik di Internet. Sementara di lain sisi, *cloud* privat hanya menyediakan layanannya ke orang-orang tertentu. Juga ada opsi *hybrid*, yang mengombinasikan baik *cloud* publik maupun *cloud* privat.

Salah satu jenis layanan *cloud computing* adalah PaaS yang menyediakan *platform* untuk kepentingan pengguna. Pada layanan PaaS *cloud computing* disediakan *platform* yang dapat dimanfaatkan pengguna untuk membuat aplikasi. Pengguna dapat membuat aplikasi sendiri dengan banyak fitur yang sudah tersedia seperti keamanan *platform*, OS, sistem, *web server*, dan *framework* aplikasi. Pengguna dapat lebih fokus pada pengembangan aplikasi.

Penelitian ini mencoba untuk membuat layanan *web service* tentang waktu shalat yang didasarkan pada layanan PaaS *cloud computing*. Diharapkan hasilnya dapat dimanfaatkan oleh pengembang perangkat lunak lain yang memerlukan layanan waktu shalat secara cepat.

Pada penelitian ini ditetapkan batasan masalah berkaitan dengan cukup luasnya lingkup yang bisa dicakup penelitian. Batasan masalah yang ditetapkan pada penelitian adalah bahwa *server cloud* menggunakan layanan PaaS yang sudah tersedia di Internet, sehingga pada penelitian tidak dibahas cara menyusun *server PaaS cloud computing*, perhitungan waktu shalat didasarkan pada perangkat lunak praytimes yang dipublikasikan melalui situs <http://praytimes.org>, layanan *web service* waktu shalat yang disediakan diwujudkan dalam bentuk format JSON dan contoh pemanfaatan *web service* diwujudkan dalam bentuk program PHP yang dapat diakses menggunakan *browser* yang diinstall pada workstation.

Penelitian bertujuan untuk menerapkan layanan *web service* pada layanan *cloud computing* berbasis PaaS, menyusun *web service* untuk memberi layanan waktu shalat menggunakan beberapa metode perhitungan dan kriteria dan membuat contoh penggunaan layanan *web service* waktu shalat pada aplikasi berbasis *web*.

Sebelumnya pernah dilakukan riset yang berkaitan dengan penelitian yang dilakukan.

Penelitian *Platform As A Service* (PaaS) Sebagai Layanan Sistem Operasi *Cloud Computing* membahas layanan PaaS, karakteristik, keuntungan, kekurangan, dan beberapa provider yang menyediakan layanan PaaS (Yunie Cancer, 2016).

Penelitian Pemanfaatan Teknologi *Cloud Computing* Dalam Reformasi Birokrasi Guna Mewujudkan Kejaksaaan Yang Profesional, Komunikatif dan Akuntabel yang menyoal mengenai pemanfaatan *cloud computing* dalam pelaksanaan reformasi birokrasi, peningkatan kualitas profesionalisme jaksa, dan mewujudkan kejaksaaan yang komunikatif serta akuntabel. Hasilnya menunjukkan bahwa *cloud computing* mampu mempercepat pelaksanaan pekerjaan, sehingga relevan diterapkan dalam membantu reformasi birokrasi di Kejaksaaan, dapat mendorong peningkatan profesionalisme dan dapat dijadikan sarana komunikasi sebagai tempat penyimpanan hasil kinerja institusi yang mudah diakses oleh publik (Rudi Pradisietia Sudirdja, 2020).

Penelitian Aplikasi *Cloud Computing* Untuk Mendukung *Collaborative Research* Pada Pembimbingan Tugas Akhir Di Jurusan Teknik Informatika FTI UII yang memberikan alternatif model bagi penerapan *collaborative research* melalui layanan teknologi *cloud computing* yang diterapkan di lingkungan Jurusan Teknik Informatika FTI UII. Melalui model tersebut diharapkan aktivitas penelitian pada pembimbingan tugas akhir dapat lebih berkualitas karena adanya aktivitas kolaborasi dan *sharing* antara dosen dan mahasiswa bimbingannya (Yudi Prayudi, 2011).

Penelitian Perancangan dan Implementasi Sistem Kuesioner Untuk Survei Berbasis *Cloud Computing* dengan studi kasus sistem pelayanan dan akademik di lingkungan kampus STIE IBBI Medan dapat memudahkan dalam melakukan survei dengan sistem kuesioner *cloud computing*, Menerapkan konsep serta teori perancangan dan implementasi sistem kuesioner berbasis *cloud computing* dengan studi kasus sistem pelayanan dan akademik di lingkungan kampus STIE IBBI Medan agar mengetahui dengan cepat tingkat pelayanan mahasiswa (Fajrillah, 2018).

Penelitian Implementasi Dan Optimalisasi *Cloud Computing* Dalam Internet Of Things (IoT) menyimpulkan bahwa perpaduan teknologi *cloud computing* dan Internet of Things yang efisien ini dapat memberikan

persepsi cerdas tentang penggunaan sumber daya sesuai kebutuhan. Dalam dunia pendidikan, untuk membentuk pribadi yang mempunyai karakter terdidik diperlukan cara yang tepat dan smart. Implementasi smart campus dengan teknologi Internet of Things bisa menyelesaikan masalah dalam proses pembelajaran dan mempunyai kehandalan yang tinggi jika terintegrasi dengan *cloud computing* (Ninik Sri Lestari, 2018)

2. METODE PENELITIAN

Perangkat lunak yang digunakan dalam penelitian berupa sistem operasi dan perangkat lunak pendukung lain yang meliputi sistem operasi Mageia 6, *server web* Nginx, sekrip Bash, sekrip PHP, sekrip HTML, sekrip PrayTimes dan *browser* Mozilla Firefox. Perangkat keras yang digunakan dalam penelitian meliputi *server* publik yang punya layanan PaaS *cloud computing*, *server* privat *cloud computing* dan PC *desktop* untuk membangun layanan *web service*. Spesifikasi penting PC *desktop* yang digunakan meliputi *server* publik *cloud computing Platform As A Service* Heroku, *server* privat *cloud computing* dan PC Desktop dengan spesifikasi Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz, RAM 4 GB, tipe sistem 64 bit, Gigabit Ethernet dan simpanan SSD.

Penyusunan Program Web Service

Untuk memperoleh informasi yang diperlukan untuk penelitian, diperlukan langkah-langkah mulai dari penyiapan konfigurasi *server* privat *cloud computing*, deploy layanan *web service* pada *server* publik PaaS *cloud computing* Heroku sampai memanfaatkan layanan *web service* menggunakan program berbasis *web*.

Beberapa program diperlukan untuk menyusun *web service* waktu adzan antara lain PrayTime, index, ws-adzan, ws adzan tabel dan ws-generate. Program PrayTime diambil dari situsnya di Internet, sedangkan empat program lainnya disusun untuk membentuk layanan *web service* waktu adzan.

Program PrayTime.php merupakan program yang disusun oleh Hamid Zarrabi-Zadeh yang dapat didownload dari situs resminya <http://praytimes.org/>. Pada penelitian ini, program tersebut digunakan sebagai program utama untuk penyusunan *web service*. Isi sekrip program tidak diubah, digunakan sesuai sekrip

aslinya. Nama *file* diganti menjadi PrayTime2022.php dan diletakkan pada direktori kerja proyek *web service* PaaS *Cloud Computing* adzan.

Untuk menggunakan sekrip program PrayTime.php digunakan sintaksis sebagai berikut.

```
<?php
//menyertakan kode program PrayTime2022.php
include('PrayTime2022.php');
//mengeset parameter
.....
//mengambil tanggal
.....
//inisialisasi objek
$prayTime = new PrayTime($method);
$prayTime->setAsrMethod($asrJuristic);
$prayTime->setDhuhrMinutes($dhuhrMinutes);
$prayTime->setHighLatsMethod($adjustHighLats);
//menghitung waktu adzan
$times=$prayTime->getPrayerTimes($date,$latitude,$
longitude,$timeZone);
$day = date('Ymd', $date);
//tampilkan layanan web service waktu adzan
.....
?>
```

Langkah pertama menggunakan sekrip PrayTime2022.php adalah menyertakan kode tersebut pada sekrip PHP. Selanjutnya dilakukan pengaturan parameter yang diperlukan pada perhitungan waktu adzan meliputi metode yang digunakan, metode penentuan waktu Ashar, metode penentuan waktu Dzuhur, pengaturan untuk daerah yang punya besaran *latitude* tempat yang tinggi, tanggal waktu penghitungan, *latitude* dan *longitude* tempat serta zona waktu.

Web service yang ditampilkan bisa berupa waktu sesaat dalam satu hari (tanggal tertentu) maupun tabel waktu selama satu tahun. Waktu sesaat tanggal tertentu dipakai untuk aplikasi yang selalu online setiap saat, sehingga waktu bisa selalu diupdate setiap ganti tanggal hari. Sedangkan tabel waktu digunakan untuk aplikasi yang tidak selalu online, sehingga perlu sekali mendownload tabel waktu adzan untuk dipakai selamanya. Untuk tabel waktu setahun, dianjurkan menggunakan dua macam tabel waktu tahun yaitu untuk tabel waktu adzan tahun biasa (bulan Pebruari berumur 28 hari) dan tabel waktu adzan tahun kabisat (bulan Pebruari berumur 29 hari).

Program index.php merupakan tampilan halaman selamat datang (*welcome page*) *web service* adzan. Halaman ini berisi macam layanan yang disediakan, parameter yang bisa diatur untuk mendapatkan layanan, nilai *default* parameter dan cara menggunakan *web service*.

Macam *web service* yang disediakan meliputi ws-adzan (menentukan waktu adzan

pada lokasi) dan ws-adzan-tabel (membuat tabel abadi waktu adzan pada tahun tertentu). Parameter yang bisa diatur meliputi method (Ithna Ashari, University of Islamic Sciences Karachi, Islamic Society of North America ISNA, Muslim World League MWL, Umm al-Qura, Makkah, Egyptian General Authority of Survey, Institute of Geophysics, University of Tehran, Kemenag dan Muhammadiyah), *latitude* letak tempat, *longitude* letak tempat, *time zone* daerah waktu, metode juristic untuk menentukan waktu Ashar, menit setelah mid-day untuk waktu dhuhur, menyesuaikan metode untuk garis lintang yang lebih tinggi dan tahun perhitungan.

Program `ws-adzan.php` merupakan program *web service* yang akan menghasilkan informasi tentang waktu adzan pada tanggal tertentu. Waktu adzan yang dihasilkan berupa waktu fajar, matahari terbit, dzuhur, ashar, matahari terbenam, maghrib dan isya. Tujuh macam waktu tersebut dipisahkan oleh tanda semi kolom (;) dan ditampilkan dalam bentuk string. Formatnya kira-kira sebagai berikut

ff:ff;rr:rr;dd:dd;aa:aa;ss:ss;mm:mm;ii:ii

yang mana f untuk fajar, r untuk terbit, d untuk dzuhur, a untuk ashar, s untuk terbenam, m untuk maghrib dan i untuk isya. Waktu ditampilkan dalam bentuk format 24 jam. Sebagai contoh adalah 17:40 yang mana angka 17 menunjukkan jam 17 sedangkan angka 40 menunjukkan lebih 40 menit.

Karena *web service* ditampilkan dalam bentuk string yang dipisahkan tanda semi kolom, pengguna (pengembang) perangkat lunak adzan bisa melakukan parsing (pemisahan) untuk menentukan waktu shalat. *Web service* ws-adzan dipakai untuk program yang selalu online untuk mengambil informasi waktu shalat setiap saat (atau setiap pergantian hari). Kalau program berbasis *web*, maka informasi waktu adzan dapat diambil saat halaman dibuka.

Program `ws-adzan-tabel.php` merupakan program *web service* yang akan menghasilkan informasi tentang waktu adzan dalam bentuk tabel pada tahun tertentu. Waktu adzan yang dihasilkan berupa waktu fajr, matahari terbit, dzuhur, ashar, matahari terbenam, maghrib dan isya. Tabel yang dihasilkan berformat csv dengan tanda pemisah semi kolom. Tiap baris informasi dipisahkan dengan tanda ganti baris. Formatnya kira-kira sebagai berikut

tanggal; subuh; terbit; dzuhur; ashar; terbenam; maghrib; isya
20220101;04:06:05;25;11:42;15:09;17:59;17:59;19:29
..... dan seterusnya

yang mana kolom pertama sampai delapan

secara urut berisi informasi tanggal, subuh, terbit, dzuhur, ashar, terbenam, maghrib dan isya. Waktu ditampilkan dalam bentuk format 24 jam.

Informasi *web service* yang ditampilkan program `ws-adzan-tabel.php` dipakai untuk pengembang aplikasi yang tidak selalu online. Informasi ini bisa dipakai sebagai tabel abadi aplikasi pewaktu adzan. Namun perlu disertakan dua tabel yaitu untuk tabel adzan tahun biasa dan tabel adzan untuk tahun kabisat. Tabel ini bisa diambil sekali dan dipakai seterusnya.

Program `ws-generate.php` adalah program yang digunakan untuk memudahkan pengguna untuk menyusun tautan *web service* waktu adzan berdasarkan semua parameter yang berpengaruh. Masukan program ini adalah semua parameter yang berpengaruh yaitu metode (kriteria perhitungan), koordinat lokasi tempat (*latitude*, *longitude*), *time zone*, juristik ashar, penentuan waktu dhuhur, koreksi untuk garis lintang tinggi dan tahun.

Hasil keluaran program `ws-generate.php` adalah dua macam tautan *web service* yaitu tautan untuk *web service* waktu tanggal tertentu dan tautan untuk *web service* tabel waktu adzan. Tautan *web service* waktu adzan pada tanggal tertentu menghasilkan waktu adzan tanggal tertentu. Tautan *web service* tabel waktu adzan menghasilkan tabel waktu adzan selama waktu satu tahun yang diwujudkan dalam bentuk *file* * csv.

Untuk memperbagus tampilan *web*, pada `ws-generate.php` disertakan fitur Bootstrap. Fitur Bootstrap juga membuat tampilan menjadi responsif sesuai *browser* yang dipakai oleh pengguna (*desktop* maupun *mobile*).

Pembuatan Akun Layanan PaaS Heroku

Akun PaaS Heroku dari melalui tautan <https://signup.heroku.com/>. Pada penelitian ini digunakan akun gratis. Data yang dimasukkan adalah name: (xxxxx), last name: ., Email: (xxxxx), company name: UTDI, role: Other, country: Indonesia, primary development language: PHP.

Selanjutnya akan dikirimkan email konfirmasi dari Heroku untuk memastikan email yang dipakai betul atau tidak. Email tersebut juga berisi tautan untuk aktivasi akun Heroku sekaligus fitur untuk membuat kata sandi.

Sampai tahap ini, akun Heroku berhasil dibuat serta diaktifkan. Selanjutnya akun ini bisa dipakai untuk menggunakan layanan PaaS

Heroku, salah satunya adalah membuat aplikasi.

Membuat aplikasi pada PaaS Heroku dapat dilakukan dengan dua cara yaitu melalui *web* dan melalui CLI Heroku. Pada penelitian ini digunakan cara melalui CLI. Oleh karena itu langkah pertama adalah menginstall Heroku CLI. Untuk menginstall Heroku CLI, dibutuhkan perangkat lunak Git. Pada penelitian digunakan sistem operasi Linux distro Ubuntu. Untuk menginstall Git pada Linux Ubuntu digunakan perintah berikut

```
# apt install git
```

Heroku CLI diinstall menggunakan menggunakan skrip instalasi yang dijalankan secara langsung lewat jaringan Internet menggunakan bantuan perangkat lunak curl. ULI Heroku digunakan perintah berikut dari shell.

```
# curl https://cli-assets.heroku.com/install-ubuntu.sh | sh
```

File-file yang diperlukan untuk menjalankan Heroku CLI akan didownload dan langsung diinstall pada sistem. Heroku CLI dipastikan sudah bekerja, dengan menjalankan perintah berikut pada shell

```
# heroku -v
heroku/7.59.2 linux-x64 node-v12.21.0
```

Dengan demikian, Heroku CLI sudah dapat dipakai untuk memanfaatkan fasilitas layanan PaaS Heroku.

Aplikasi *web service* waktu shalat diberi nama ws-adzan. Perintah Heroku untuk membuat aplikasi ws-adzan adalah sebagai berikut.

```
$ heroku login
warning: heroku update available from 7.59.2 to 7.60.2.
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/(xxxxx)?requestor=(xxxxx)
Logging in... done
Logged in as wghyc@gmail.com
```

```
~/heroku$ heroku create ws-adzan
warning: heroku update available from 7.59.2 to 7.60.2.
```

```
Creating ● ws-adzan... done
https://ws-adzan.herokuapp.com/ |
https://git.heroku.com/ws-adzan.git
```

Aplikasi ws-adzan berhasil dibuat dan siap untuk digunakan/dilengkapi sesuai program yang telah disusun. Program yang telah disusun harus di-deploy ke layanan PaaS Heroku pada aplikasi ws-adzan,

Setelah aplikasi ws-adzan berhasil dibuat, langkah selanjutnya adalah deploy semua skrip program yang diperlukan untuk menjalankan aplikasi *web service* waktu shalat terutama yaitu PrayTime2022.php, ws-adzan.php, ws-adzan-

tabel.php dan ws-generate.php. Sebelum melakukan deploy, aplikasi ws-adzan yang sudah dibuat pada heroku harus di-clone terlebih dahulu pada direktori kerja. Direktori kerja yang digunakan adalah heroku dibawah direktori home pengguna.

```
~/heroku$ heroku git:clone -a ws-adzan
warning: heroku update available from 7.59.2 to 7.60.2.
Cloning into 'ws-adzan'...
warning: You appear to have cloned an empty repository.
```

Perintah ini akan menyebabkan aplikasi ws-adzan pada *server* Heroku akan di-clone ke direktori lokal ws-adzan. Heroku telah menyiapkan semua keperluan *file* yang diperlukan untuk jalannya aplikasi ws-adzan pada layanan PaaS Heroku.

Setelah semua *file* aplikasi sudah berhasil di-clone, selanjutnya semua *file* yang diperlukan untuk jalannya aplikasi *web service* waktu shalat dipindahkan pada direktori ws-adzan di bawah direktori heroku. Modifikasi *file* yang diperlukan, semua dilakukan pada direktori ws-adzan tersebut.

Proses deploy dilakukan dengan cara sinkronisasi menggunakan perangkat lunak git. Urutan perintah untuk melakukan proses sinkronisasi adalah sebagai berikut.

```
~/heroku$ cd ws-adzan
~/heroku/ws-adzan$ git add .
~/heroku/ws-adzan$ git commit -am "ws-adzan update"
[master (root-commit) 314e4cb] ws-adzan update
12 files changed, 3669 insertions(+)
create mode 100644 PrayTime2022.php
create mode 100644 css/font-awesome.min.css
create mode 100644 fonts/fontawesome-webfont.eot
create mode 100644 fonts/fontawesome-webfont.svg
create mode 100644 fonts/fontawesome-webfont.ttf
create mode 100644 fonts/fontawesome-webfont.woff
create mode 100644 fonts/fontawesome-webfont.woff2
create mode 100644 index.php
create mode 100644 version.php
create mode 100644 ws-adzan-tabel.php
create mode 100644 ws-adzan.php
create mode 100644 ws-generate.php
```

```
~/heroku/ws-adzan$ git push heroku master
Counting objects: 16, done.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (16/16), 415.75 KiB | 5.54 MiB/s, done.
```

```
remote: Verifying deploy... done.
To https://git.heroku.com/ws-adzan.git
* [new branch] master -> master
```

Beberapa bagian dari respon Heroku tidak ditampilkan, untuk menghemat tempat, namun tidak mengurangi makna penulisan.

Sampai tahap ini, semua *file* yang diperlukan untuk menjalankan *web service* layanan waktu shalat sudah berhasil di-deploy. Selanjutnya dapat dilakukan pengujian hasil pemuatan *web*

service melalui browser.

3. HASIL DAN PEMBAHASAN

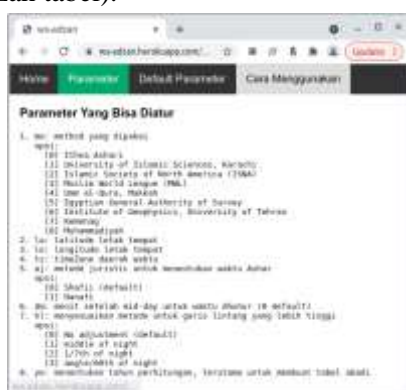
Pada pembahasan ini terutama akan ditampilkan keberhasilan hasil pengujian *web service* waktu shalat dan bagaimana cara menggunakan *web service* pada suatu aplikasi *web*. Aplikasi yang dikembangkan menggunakan *web service* tidak terbatas pada aplikasi *web* bisa aplikasi dekstop, aplikasi *mobile* atau pun aplikasi *embedded system*. Yang penting aplikasi tersebut dapat melakukan akses data pada *web service* waktu adzan.

Tampilan awal layanan *web service* berisi informasi tentang perangkat lunak *ws-adzan* ditampilkan pada Gambar 1. Tampilan awal berisi beberapa tab yaitu Home, Parameter, Default Parameter dan Cara menggunakan.



Gambar 1. Welcome Page

Informasi yang ditampilkan pada tab Home meliputi versi perangkat lunak, program dasar perhitungan waktu shalat dan informasi dua layanan *web service* waktu shalat (*ws-adzan* dan *ws-adzan tabel*).



Gambar 2. Parameter Waktu Adzan

Pada tab Parameter (Gambar 2) ditampilkan semua parameter yang perlu diatur untuk mendapat layanan waktu shalat beserta penjelasannya. Karena cukup banyak parameter yang perlu diatur, untuk memudahkan beberapa parameter dibuat *default*. Nilai *default* parameter ditampilkan pada tab Default Parameter. Tab Default Parameter ditampilkan pada Gambar 3.



Gambar 3. Nilai Default Parameter

Untuk menentukan waktu shalat, beberapa parameter biasanya diberi nilai *default* yang artinya banyak digunakan secara umum. Umumnya yang tidak *default* adalah metode yang digunakan, lokasi geografis tempat dan zona waktu.



Gambar 4. Contoh Tautan Web Service

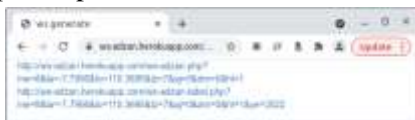
Tab selanjutnya pada layanan *web service* waktu shalat adalah cara menggunakan. Pada tab ini ditampilkan cara memanfaatkan *web service* menggunakan contoh-contoh tautan yang sudah disediakan atau dengan membuat tautan sendiri. Jika pengguna menginginkan untuk membuat tautan sendiri, maka dapat dilakukan menggunakan bantuan *ws-generate*. Program *ws-generate* akan menghasilkan tautan *web service* berdasarkan parameter yang diatur oleh pengguna secara interaktif. Tampilan program *ws-generate* ditunjukkan Gambar 4.



Gambar 5. Generate Tautan Web Service

Jika semua parameter sudah diatur dan tombol generate di-klik, maka akan menampilkan dua macam tautan yaitu tautan untuk menampilkan waktu shalat pada tanggal tertentu dan tautan untuk download tabel waktu

shalat selama satu tahun. Contoh hasil tautan *web service* hasil dari program *ws-generate* ditampilkan pada Gambar 6.



Gambar 6. Tautan Hasil Generate

Dari dua macam tautan yang dihasilkan, pengguna bisa memilih, yang mana yang diperlukan. Tautan mana yang dipilih, sangat tergantung pada aplikasi yang akan dikembangkan oleh pengguna.

Selanjutnya akan ditunjukkan bagaimana menggunakan *web service* waktu shalat pada aplikasi *web*. Aplikasi *web* bisa ditempatkan pada *server* lokal atau *server* pada Internet. Contoh program *adzan-client.php* ditempatkan pada *server* lokal, namun mengambil data dari layanan *web service* waktu shalat PaaS *cloud computing* Heroku.

Untuk mengambil data dari *web service*, paling mudah menggunakan sintaksis berikut.

```
$waktu=file_get_contents('http://ws-  
adzan.herokuapp.com/ws-adzan.php?me=8&la=-  
7.7956&lo=110.3695&tz=7');  
$awaktu = explode(";", $waktu);  
$tahun=substr($awaktu[0],0,4);  
$bulan=substr($awaktu[0],4,2);  
$tanggal=substr($awaktu[0],6,2);
```

Langkah pertama adalah mengambil data pada layanan *web service* *ws-adzan* menggunakan fungsi *file_get_contents()*. Hasilnya disimpan dalam bentuk string pada variabel *\$waktu*. Langkah berikutnya mengubah string menjadi variabel larik *\$awaktu*. Waktu shalat dari fajar, terbit, dzuhur, ashar, terbenam, maghrib dan isya disimpan pada variabel larik *\$awaktu* dari indeks 1 sampai 7. Sedangkan indeks satu menyimpan format tahun-bulan-tanggal dalam satu string. Untuk memperoleh tahun, bulan dan tanggal digunakan operasi string parsing menggunakan fungsi *substr()*.



Gambar 7. Aplikasi Waktu Shalat

Hasil tampilan program *adzan-client* ditampilkan pada Gambar 7. Contoh waktu shalat yang ditampilkan adalah untuk kota

Yogyakarta pada tanggal 17 Juli 2022.

4. KESIMPULAN

Kesimpulan yang dapat diambil dari hasil pembahasan dan percobaan dalam penelitian ini adalah bahwa layanan *web service* waktu shalat berhasil diterapkan pada layanan PaaS *cloud computing* Heroku, layanan *web service* waktu shalat berhasil disusun berdasarkan beberapa metode dan kriteria dari PrayTime.org dan *web service* berhasil dibuat dan digunakan oleh aplikasi waktu shalat berbasis *web*.

5. REFERENSI

- [1] Achour, M., dkk, 2017, *PHP Manual documentation*, <http://php.net/manual/en/>
- [2] Fajrillah, 2018, *Perancangan dan Implementasi Sistem Kuesioner Untuk Survei Berbasis Cloud Computing*, jurnal OSF Preprints, October 31
- [3] IBM, 2022, *CICS Transaction Server for z/OS (Concepts of JSON web services)*, <https://www.ibm.com/docs/en/cics-ts/5.3?topic=services-concepts-json-web>
- [4] Jakob Jenkov, 2014, *Web Service Message Formats*, <https://jenkov.com/tutorials/web-services/message-formats.html>
- [5] Javatpoint, 2022, *Cloud Computing Tutorial*, <https://www.javatpoint.com/platform-as-a-service>
- [6] Nginx Foundation, 2018, *nginx*, <https://nginx.org/en/>
- [7] Ninik Sri Lestari, 2018, *Implementasi Dan Optimalisasi Cloud Computing Dalam Internet Of Things (IoT)*, Jurnal Isu Teknologi, Vol 13 No 2
- [8] PrayTimes.org, 2011, *Pray Times Manual*, <http://praytimes.org/manual>
- [9] Refsnes Data, 2022, *PHP Tutorial*, <https://www.w3schools.com/php/>
- [10] Rudi Pradisetia Sudirdja, 2020, *Pemanfaatan Teknologi Cloud Computing Dalam Reformasi Birokrasi Guna Mewujudkan Kejaksaaan Yang Profesional, Komunikatif dan Akuntabel*, Jurnal Hukum & Pembangunan, Vol 50, No 4
- [11] Sai Vennam, 2022, *Cloud Computing*, <https://www.ibm.com/cloud/learn/cloud-computing>
- [12] Salesforce, 2022, *Guided Price Estimator*, <https://www.heroku.com/pricing>
- [13] Salesforce, 2022, *Heroku Dashboard*, <https://dashboard.heroku.com>
- [14] Salesforce, 2022, *PaaS Heroku*, <https://www.heroku.com/what>

- [15] Salesforce, 2022, *The Heroku CLI*, <https://devcenter.heroku.com/articles/heroku-cli#cli-architecture>
- [16] Tutorialpoint, 2021, *Cloud Computing Tutorial*, https://www.tutorialspoint.com/cloud_computing/cloud_computing_platform_as_a_service.htm
- [17] Tutorials Point, 2022, *Web Services Tutorial*, <https://www.tutorialspoint.com/webservices>
- [18] Yudi Prayudi, 2011, *Aplikasi Cloud Computing Untuk Mendukung Collaborative Research Pada Pembimbingan Tugas Akhir Di Jurusan Teknik Informatika FTI UII*, Seminar Nasional Aplikasi Teknologi Informasi (SNATI)
- [19] Yunie Cancer, 2016, *Platform As A Service (PaaS) Sebagai Layanan Sistem Operasi Cloud Computing*, Jurnal TIMES, Vol 5 No 1