

Halaman Judul

LAPORAN PENELITIAN
**IMPLEMENTASI WEB SERVICE WAKTU SHALAT
BERBASIS TEKNOLOGI PAAS CLOUD COMPUTING**



oleh
W A G I T O, S.T., M.T.
NIDN : 0522126901
NPP : 961080

Mendapat Bantuan Biaya Penelitian dari Puslitbang dan PPM
Semester Ganjil 2021/2022

Universitas Teknologi Digital Indonesia
YOGYAKARTA
Tahun 2022

HALAMAN PENGESAHAN

Judul Penelitian : Implementasi Web Service Waktu Shalat Berbasis Teknologi PaaS Cloud Computing
Rumpun Ilmu : Informatika
Ketua Peneliti
a. Nama : Wagito, S.T., M.T.
b. NIDN : 0522126901
c. Jabatan Fungsional : Lektor Kepala
d. Jurusan/Prodi : Informatika
e. Nomor HP : 081215659383
f. Alamat email : wagito@utdi.ac.id
g. Anggota Peneliti Dosen : -
h. Anggota Peneliti Non Dosen : -

no.	Nama	NIM	Institusi
1	Rizki Irianto	185410098	Informatika UTDI Yogyakarta

i. Dana :
Yogyakarta, Agustus 2022

Mengetahui
Ketua Prodi


Dini Fakta Sari S.T., M.T.
NPP. 121172

Ketua Peneliti


Wagito, S.T., M.T.
NPP. 961080

Menyetujui
Kepala Puslitbang dan PPM
UTDI Yogyakarta


Ebbri Nova Lenti, S.Si, M.T.
NPP. 961079

Kata Pengantar

Puji syukur kepada Allah S.W.T. karena telah melimpahkan rahmat, hidayah dan taufik-Nya. Berkat pertolongan dan tuntunan-Nya serta dengan berbagai usaha akhirnya penelitian ini berhasil diselesaikan dengan baik.

Penelitian yang berjudul *Implementasi Web Service Waktu Shalat Berbasis Teknologi PaaS Cloud Computing* 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*. Teknologi yang dilibatkan penelitian ini terutama adalah layanan PaaS *Cloud Computing*, sekrip program PHP, teknologi PrayTime dan layanan data *web service*.

Hasil penelitian ini masih banyak kekurangannya, sehingga kritik dan saran yang membangun untuk lebih mengembangkan hasilnya sangat diharapkan. Semoga hasil penelitian ini bermanfaat bagi semua orang.

Penulis

Daftar Isi

Halaman Judul.....	
HALAMAN PENGESAHAN.....	ii
Kata Pengantar.....	iii
Daftar Isi.....	iv
Daftar Gambar.....	viii
ABSTRAK.....	x
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Target Luaran.....	5
BAB 2 TINJAUAN PUSTAKA.....	6
BAB 3 TEORI.....	8
3.1 Teknologi PrayTime.....	8
3.2 Teknologi <i>Web Service</i>	14
3.3 <i>Server</i> Web Nginx.....	16

3.4 PHP.....	18
3.5 Nginx.....	20
3.6 Cloud Computing.....	21
3.7 PaaS (<i>Platform As A Service</i>).....	22
3.8 PaaS Heroku.....	24
3.8.1 Registrasi Heroku.....	25
3.8.2 Heroku CLI (<i>Command Line Interpreter</i>).....	28
3.8.3 Membuat Aplikasi Heroku.....	29
3.8.4 <i>Deploy</i> Aplikasi Heroku.....	34
3.8.4.1 <i>Deploy</i> Menggunakan Heroku Git.....	35
3.8.4.2 Menghubungi Github.....	35
3.8.4.3 <i>Deploy</i> Aplikasi Docker.....	36
BAB 4 METODE PENELITIAN.....	38
4.1 Bahan Penelitian.....	38
4.2 Alat.....	38
4.3 Metodologi Penelitian.....	40
4.3.1 Rancangan Perangkat Keras.....	40
4.3.2 Rancangan Perangkat Lunak.....	41
4.3.3 Tahapan Penelitian.....	42
BAB 5 IMPLEMENTASI DAN PEMBAHASAN.....	43
5.1 Implementasi.....	43
5.1.1 Penyusunan Program Pada <i>Server</i> Lokal.....	43

5.1.1.1 Program PrayTime.php.....	43
5.1.1.2 Program index.php.....	45
5.1.1.3 Program ws-adzan.php.....	47
5.1.1.4 Program ws-adzan-tabel.php.....	48
5.1.1.5 Program ws-generate.php.....	49
5.1.2 Pembuatan Akun Layanan PaaS Heroku.....	50
5.1.3 Pembuatan Aplikasi PaaS Heroku.....	51
5.1.4 <i>Deploy</i> Aplikasi Pada Layanan PaaS Heroku.....	52
5.2 Pembahasan.....	54
5.2.1 Pengujian aplikasi Web Service.....	55
5.2.2 Pengujian Menggunakan Aplikasi Client Side.....	59
BAB 6 KESIMPULAN.....	61
6.1 Kesimpulan.....	61
6.1 Saran.....	61
Daftar Pustaka.....	62
LAMPIRAN.....	1
Program PrayTime.php.....	1
Program index.php.....	10
Program ws-adzan.php.....	14
Program ws-adzan-tabel.php.....	15
Program ws-generate.php.....	16
Program adzan-client.php.....	18

Curriculum Vitae.....	19
Personalia Penelitian.....	20
Biaya Penelitian.....	21
Jadwal Penelitian.....	22
Surat Keputusan.....	23
Review Naskah.....	26
Submit Jurnal.....	28

Daftar Gambar

Gambar 3.1 Bentuk Komunikasi <i>Web Service</i> (Jakob Jenkov, 2014).....	15
Gambar 3.2 Layanan Registrasi Heroku.....	26
Gambar 3.3 Konfirmasi Email.....	27
Gambar 3.4 Pengaturan <i>Password</i>	27
Gambar 3.5 Akun Heroku Berhasil Dibuat.....	28
Gambar 3.6 <i>Login</i> Melalui Web Heroku.....	30
Gambar 3.7 Membuat Aplikasi Melalui Web Heroku.....	30
Gambar 3.8 Dialog Membuat Aplikasi.....	31
Gambar 3.9 Dashboard Aplikasi PaaS Heroku.....	31
Gambar 3.10 Tautan URL <i>Login</i> PaaS Heroku.....	32
Gambar 3.11 Halaman <i>Login</i> CLI.....	32
Gambar 3.12 Halaman <i>Login</i> PaaS Heroku.....	33
Gambar 3.13 Pesan Berhasil <i>Login</i>	33
Gambar 3.14 Pesan Pada Shell Berhasil <i>Login</i> Heroku.....	33
Gambar 3.15 Aplikasi PaaS Heroku Berhasil Dibuat.....	34
Gambar 3.16 Halaman Kosong PaaS Heroku.....	34
Gambar 4.1 Disain Arsitektur Sistem.....	41
Gambar 5.1 Welcome Page.....	55
Gambar 5.2 Parameter Waktu Adzan.....	56

Gambar 5.3 Nilai Default Parameter Waktu Adzan.....	56
Gambar 5.4 Contoh Tautan Web Service.....	57
Gambar 5.5 Generate Tautan Web Service.....	58
Gambar 5.6 Tautan Hasil Generate.....	58
Gambar 5.7 Aplikasi Waktu Shalat.....	60

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 serta ditambah kriteria dari Kemenag Indonesia dan Ormas Muhammadiyah Indonesia. Penelitian berhasil menggunakan layanan *web service* untuk aplikasi waktu shalat berbasis *web*.

Kata kunci: *web service*, waktu shalat, *cloud*

BAB 1 PENDAHULUAN

1.1 Latar Belakang

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 *database*, *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.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian adalah bagaimana menyusun layanan *web service* yang berdasar pada layanan PaaS *cloud computing* untuk membarikan informasi waktu shalat.

1.3 Batasan Masalah

Batasan masalah yang ditetapkan pada penelitian ini adalah sebagai berikut.

1. *Server cloud* menggunakan layanan PaaS yang sudah tersedia di Internet, sehingga pada penelitian tidak dibahas cara menyusun *server PaaS cloud computing*.
2. Perhitungan waktu shalat didasarkan pada perangkat lunak praytimes yang dipublikasikan melalui situs <http://praytimes.org>

3. Layanan *web service* waktu shalat yang disediakan diwujudkan dalam bentuk format JSON.
4. Contoh pemanfaatan *web service* diwujudkan dalam bentuk program PHP yang dapat diakses menggunakan *browser* yang diinstall pada mesin *workstation*.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai pada penelitian yang dilakukan adalah sebagai berikut.

1. Menerapkan layanan *web service* pada layanan *cloud computing* berbasis PaaS.
2. Menyusun *web service* untuk memberi layanan waktu shalat menggunakan beberapa metode perhitungan dan kriteria.
3. Membuat contoh penggunaan layanan *web service* waktu shalat pada aplikasi berbasis *web*.

1.5 Manfaat Penelitian

Penelitian bermanfaat untuk menyusun *web service* bagi layanan waktu shalat menggunakan beberapa metode perhitungan dan kriteria. Penelitian yang dihasilkan bermanfaat bagi pengembang perangkat lunak yang memerlukan layanan *web service* waktu shalat.

1.6 Target Luaran

Hasil penelitian direncanakan dilakukan publikasi dan seminar pada kegiatan ilmiah. Publikasi yang diinginkan adalah pada jurnal ilmiah nasional yang sudah akreditasi atau minimal hasil penelitian bisa dipublikasikan pada jurnal ilmiah nasional.

BAB 2 TINJAUAN PUSTAKA

Beberapa tulisan dan penelitian berkaitan dengan penggunaan *cloud computing* pernah dipublikasikan melewati jurnal ataupun seminar. 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 Kejaksanaan Yang Profesional, Komunikatif dan Akuntabel* yang menyoal mengenai pemanfaatan *computing* dalam pelaksanaan reformasi birokrasi, peningkatan kualitas profesionalisme jaksa, dan mewujudkan kejaksanaan yang komunikatif serta akuntabel. Hasilnya menunjukkan bahwa *cloud computing* mampu mempercepat pelaksanaan pekerjaan , sehingga relevan diterapkan dalam membantu reformasi birokrasi di Kejaksanaan, dapat mendorong peningkatan profesionalisme dan dapat dijadikan sarana komunikasi sebagai tempat penyimpanan hasil kinerja institusi yang mudah diakses oleh publik (Rudi Pradisetia 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)

BAB 3 TEORI

3.1 Teknologi PrayTime

Uraian tentang teknologi PrayTime sepenuhnya diambil dari manualnya yang diterbitkan tahun 2011 melalui situs <http://praytimes.org/> (praytimes.org, 2011). PrayTime adalah sebuah proyek Islami dalam menyediakan perpustakaan sumber terbuka untuk menghitung waktu sholat Muslim. Versi pertama dari Waktu Sholat dirilis pada awal tahun 2007. Kode tersebut saat ini digunakan di berbagai situs *web* dan aplikasi Islam. Kode yang diterbitkan di situs *web* praytimes.org dilisensikan di bawah lisensi GNU LGPL. Kode-kode dapat digunakan secara bebas di situs *web* dan aplikasi apa pun, asalkan kredit diberikan pada karya asli dengan tautan kembali ke praytimes.org.

Tabel 1.1 Pengembang Pray Times

Language	Version	Developer/Translator
Python	ver 2	Saleem Shafi, Hamid Zarrabi-Zadeh
PHP	ver 1	Hamid Zarrabi-Zadeh
Java	ver 1	Hussain Ali Khan
C++	ver 1	Mohammad Ebrahim Mohammadi Panah
Objective C	ver 1	Hussain Ali Khan
C#	ver 1	Jandost Khoso

Pray Times awalnya ditulis dalam JavaScript, dan di-porting ke bahasa

pemrograman lain. Tabel 1.1 adalah daftar Pray Times dalam bahasa selain JavaScript beserta pengembangnya.

Versi terakhir kode-kode Pray Times dalam berbagai bahasa pemrograman tersebut dipublikasikan lewat situs <https://github.com/abodehq/Pray-Times>.

Penggunaan Umum Dalam JavaScript

Langkah pertama untuk menggunakan PrayTimes di halaman *web* atau widget adalah memasukkannya menggunakan baris seperti ini

```
<script type="text/javascript" src="PrayTimes.js"></script>
```

Setelah memasukkan PrayTimes.js, sebuah objek bernama PrayTimes dibuat dan siap digunakan. Waktu sholat bisa langsung didapatkan (menggunakan pengaturan default) dari objek ini. Misalnya, untuk mendapatkan jadwal sholat hari ini untuk lokasi dengan garis lintang 43, garis bujur -80, dan zona waktu -5, dapat memanggil fungsi

```
PrayTimes.getTimes(new Date(), [43, -80], -5);
```

Ada beberapa fungsi untuk menyesuaikan parameter perhitungan. Misalnya, dapat memanggil fungsi berikut (sebelum memanggil getTimes) untuk mengubah metode perhitungan menjadi ISNA

```
PrayTimes.setMethod('ISNA');
```

Cara Mendapatkan Waktu Sholat

Fungsi berikut digunakan untuk mengambil waktu sholat untuk tanggal dan lokasi tertentu:

```
getTimes(date, coordinates, timezone, dst, format)
```

Parameter input dijelaskan di bawah ini:

- **date**

Tanggal di mana waktu sholat dihitung. Tanggal hari ini dapat ditentukan menggunakan `new Date()`. Tanggal juga dapat dimasukkan sebagai rangkap tiga [year, month, day]. Misalnya, [2009, 2, 26] menentukan tanggal 26 Februari 2009.

- **coordinates**

Menentukan koordinat lokasi input sebagai triple [latitude, longitude, elevation]. Lintang adalah bilangan real antara -90 dan 90, bujur antara -180 dan 180, dan elevasi adalah bilangan positif, yang menentukan ketinggian dalam meter sehubungan dengan medan di sekitarnya. Parameter elevasi adalah opsional. Contoh koordinat yang valid adalah [-43.2, 80.6] dan [12.5, -25.8, 300].

- **timezone**

Perbedaan waktu Greenwich (GMT) dalam jam. Jika dihilangkan atau disetel ke 'otomatis', timezone diekstraksi dari sistem.

- **dst**

Waktu Musim Panas (*Daylight Saving Time*): 1 jika tanggal dalam waktu musim panas, 0 jika tidak. Jika dihilangkan atau disetel ke 'otomatis', dst diekstraksi dari sistem.

- **format**

Tampilan waktu dapat diatur sedemikian rupa sesuai format yang disediakan oleh perangkat lunak.

Tabel 1.2 Format Waktu

Format	Description	Example
24h	24-hour time format	16:45
12h	12-hour time format	4:45 pm
12hNS	12-hour format with no suffix	4:45
Float	Floating point number	16.75

- Return Value

Fungsi `getTimes` mengembalikan nilai berupa array asosiatif yang berisi sembilan waktu sholat. Setiap waktu dapat diakses melalui namanya.

Menetapkan Metode Perhitungan

Ada beberapa konvensi untuk menghitung waktu sholat. Konvensi default yang digunakan dalam `PrayTimes` adalah Liga Dunia Muslim. Metode penghitungan dapat diubah menggunakan fungsi berikut:

`setMethod (method)`

metode dapat berupa salah satu dari berikut ini:

- `MWL` : Muslim World League
- `ISNA` : Islamic Society of North America
- `Egypt` : Egyptian General Authority of Survey
- `Makkah` : Umm al-Qura University, Makkah
- `Karachi` : University of Islamic Sciences, Karachi

- Tehran : Institute of Geophysics, University of Tehran
- Jafari : Shia Ithna Ashari (Ja`fari)

Menyesuaikan Parameter

Parameter penghitungan waktu shalat dapat disesuaikan menggunakan fungsi berikut:

adjust (parameters)

parameters adalah berupa array asosiatif yang terdiri dari sejumlah parameter berikut.

Tabel 1.3 Parameter Penyesuaian Waktu

Parameter	Values	Description
imsak	degrees	twilight angle
	minutes	minutes before fajr
fajr	degrees	twilight angle
dhuhr	minutes	minutes after mid-day
asr	method	asr juristic method; see the table below
	factor	shadow length factor for realizing asr
maghrib	degrees	twilight angle
	minutes	minutes after sunset
isha	degrees	twilight angle
	minutes	minutes after maghrib
midnight	method	midnight method; see the table below
highLats	method	higher latitudes adjustment; see below

Metode Ashar

Penentuan waktu ashar dapat dilakukan menggunakan dua metode sesuai parameter sebagai berikut.

- Standard : Shafii, Maliki, Jafari and Hanbali (shadow factor = 1)

- Hanafi : Hanafi school of thought (shadow factor = 2)

Metode Tengah Malam

Tengah malam dapat ditentukan menggunakan dua metode sesuai parameter sebagai berikut.

- Standard : The mean time from Sunset to Sunrise
- Jafari : The mean time from Maghrib to Fajr

Metode Garis Lintang Tinggi

Pada garis lintang tinggi, waktu shalat bisa disesuaikan menggunakan salah satu metode sesuai parameter berikut.

- None : No adjustments
- NightMiddle : The middle of the night method
- OneSeventh : The 1/7th of the night method
- AngleBased : The angle-based method (recommended)

Waktu Penyetelan

Waktu shalat dapat disetel lebih lanjut (untuk pencegahan) menggunakan fungsi berikut:

`tune (offsets)`

yang mana `offsets` adalah array asosiatif yang berisi offset waktu dalam menit untuk setiap waktu shalat.

Secara default, PrayTimes membulatkan menit ke nilai terdekat. Untuk membulatkan waktu tertentu ke atas, dapat disetel dengan +0,5 menit, dan untuk membulatkannya ke bawah, dapat disetel dengan -0,5 menit. Tuning adalah langkah terakhir setelah menghitung langkah, dan dengan demikian, tidak berpengaruh pada parameter perhitungan. Misalnya, jika Isha diatur menjadi 90 menit setelah matahari terbenam, menyetel matahari terbenam selama 5 menit tidak akan mendorong Isha maju.

3.2 Teknologi *Web Service*

Buku yang berbeda dan organisasi yang berbeda memberikan definisi yang berbeda untuk layanan *web*. Beberapa di antaranya tercantum di sini yang disarikan oleh Tutorial Point.

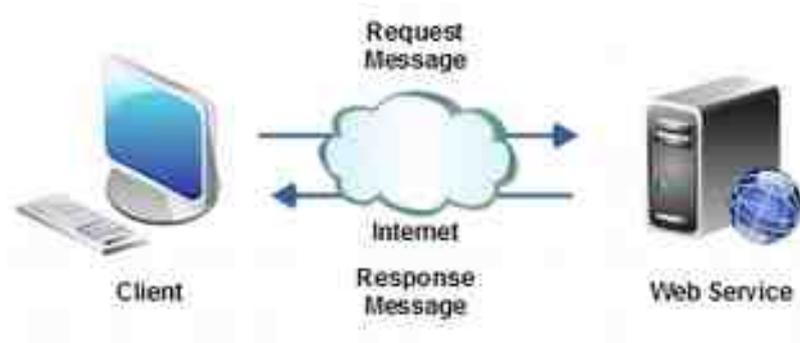
Layanan *web* adalah perangkat lunak apa pun yang tersedia melalui internet dan menggunakan sistem pesan XML standar. XML digunakan untuk mengkodekan semua komunikasi ke layanan *web*. Misalnya, klien memanggil layanan *web* dengan mengirimkan pesan XML, lalu menunggu respons XML yang sesuai. Karena semua komunikasi dalam XML, layanan *web* tidak terikat pada satu sistem operasi atau bahasa pemrograman mana pun. Java dapat berhubungan dengan Perl; Aplikasi Windows dapat berbicara dengan aplikasi Unix.

Layanan *web* adalah aplikasi mandiri, modular, terdistribusi, dinamis yang dapat dideskripsikan, diterbitkan, ditempatkan, atau dipanggil melalui jaringan

untuk membuat produk, proses, dan rantai pasokan. Aplikasi ini bisa lokal, terdistribusi, atau berbasis *web*. Layanan *web* dibangun di atas standar terbuka seperti TCP/IP, HTTP, Java, HTML, dan XML.

Layanan *web* adalah sistem pertukaran informasi berbasis XML yang menggunakan Internet untuk interaksi aplikasi-ke-aplikasi langsung. Sistem ini dapat mencakup program, objek, pesan, atau dokumen.

Layanan *web* adalah kumpulan protokol dan standar terbuka yang digunakan untuk pertukaran data antar aplikasi atau sistem. Aplikasi perangkat lunak yang ditulis dalam berbagai bahasa pemrograman dan berjalan di berbagai *platform* dapat menggunakan layanan *web* untuk bertukar data melalui jaringan komputer seperti Internet dengan cara yang mirip dengan komunikasi antar-proses pada satu komputer. Interoperabilitas ini (misalnya, antara Java dan Python, atau aplikasi Windows dan Linux) disebabkan oleh penggunaan standar terbuka.



Gambar 3.1 Bentuk Komunikasi *Web Service* (Jakob Jenkov, 2014)

Ketika klien dan layanan *web* berkomunikasi, mereka bertukar pesan. Pesan permintaan dikirim dari klien ke layanan *web*. Layanan *web* merespons dengan pesan respons. Ini seperti di HTTP biasa, di mana *browser web* mengirimkan permintaan HTTP ke *server web*, dan *server web* membalas dengan respons HTTP.

Layanan *web* adalah istilah umum untuk fungsi perangkat lunak yang di-host di lokasi yang dapat dialamatkan jaringan. Dalam pengertian umum ini dapat menyiratkan layanan berbasis *cloud*, layanan Utilitas atau bahkan aplikasi departemen. Istilah layanan *web* juga dapat digunakan dalam arti yang lebih spesifik, seperti layanan yang dihosting menggunakan SOAP yang dijelaskan menggunakan dokumen WSDL. Istilah yang lebih umum sering digunakan oleh komunitas JSON ketika menggambarkan layanan berbasis JSON. Layanan *web* JSON menggunakan istilah ini dalam pengertian umum. (IBM, 2022)

3.3 Server Web Nginx

Nginx [dibaca: engine x] adalah HTTP dan *reverse proxy server*, *server proxy email*, dan *server proxy TCP / UDP generik*, yang aslinya ditulis oleh Igor Sysoev. Untuk waktu yang lama, telah berjalan di banyak situs Rusia yang banyak dimuat termasuk Yandex, Mail.Ru, VK, dan Rambler. Menurut Netcraft, nginx melayani atau proxied 30,46% situs tersibuk di bulan Januari 2018. Berikut adalah beberapa kisah suksesnya: Dropbox, Netflix, Wordpress.com, FastMail.FM.

(Nginx Foundation, 2018)

3.2.1 Fitur Dasar Nginx

Nginx memiliki fitur dasar yang berkaitan dengan fungsi utama sebagai *server web* berikut

1. Melayani *file* statis dan indeks, autoindexing; buka *file* descriptor cache;
2. Accelerated reverse proxying dengan caching; load balancing dan fault tolerance;
3. Dukungan yang dipercepat dengan caching *server* FastCGI, uwsgi, SCGI, dan memcached; load balancing dan fault tolerance;
4. Arsitektur modular. Filter meliputi gzipping, rentang byte, respon chunked, XSLT, SSI, dan filter transformasi gambar. Beberapa inklusi SSI dalam satu halaman dapat diproses secara paralel jika ditangani oleh proxy atau FastCGI / uwsgi / SCGI;
5. Dukungan SSL dan TLS SNI;
6. Dukungan untuk HTTP / 2 dengan prioritas berbasis bobot dan ketergantungan.

Server web Nginx sudah mendukung beberapa sistem operasi. Di antara sistem operasi yang sudah diuji menggunakan Nginx adalah sebagai berikut.

1. FreeBSD 3 — 11 / i386; FreeBSD 5 — 11 / amd64;
2. Linux 2.2 — 4 / i386; Linux 2.6 — 4 / amd64; Linux 3 — 4 / armv6l, armv7l, aarch64, ppc64le;

3. Solaris 9 / i386, sun4u; Solaris 10 / i386, amd64, sun4v;
4. AIX 7.1 / powerpc;
5. HP-UX 11.31 / ia64;
6. macOS / ppc, i386;
7. Windows XP, Windows *Server* 2003.

Selain fitur dasar yang sudah disebutkan pada paragraf sebelumnya. *Server* nginx juga mempunyai fitur dasar yang berkaitan dengan akses *database* dan akses e-mail. Fitur-fitur ini berkaitan dengan kemampuan *server* Nginx sebagai *reverse proxy*, sehingga dapat digunakan sebagai pelindung pada jaringan privat.

3.4 PHP

PHP adalah singkatan dari "PHP: Hypertext Preprocessor". PHP adalah bahasa scripting open source yang banyak digunakan, dieksekusi di *server*, gratis untuk diunduh dan digunakan dan bahasa yang luar biasa dan populer. PHP cukup kuat untuk menjadi inti dari sistem blogging terbesar di *web* (WordPress), cukup dalam untuk menjalankan jejaring sosial besar dan cukup mudah untuk menjadi bahasa sisi *server* pertama bagi pemula.

File PHP dapat berisi teks, HTML, CSS, JavaScript, dan kode PHP. Kode PHP dieksekusi di *server*, dan hasilnya dikembalikan ke *browser* sebagai HTML biasa. *File* PHP memiliki ekstensi ".php".

PHP dapat menghasilkan konten halaman dinamis, dapat membuat,

membuka, membaca, menulis, menghapus, dan menutup *file* di *server*, dapat mengumpulkan data formulir, dapat mengirim dan menerima cookie, dapat menambah, menghapus, mengubah data di *database*, dapat digunakan untuk mengontrol akses pengguna, dapat mengenkripsi data (Refsnes Data, 2022)

PHP berjalan di berbagai *platform* (Windows, Linux, Unix, Mac OS X, dan sebagainya), kompatibel dengan hampir semua *server* yang digunakan saat ini (Apache, IIS, dan sebagainya), mendukung berbagai *database*, gratis diunduh dari sumber resmi PHP (www.php.net) dan mudah dipelajari dan berjalan secara efisien di sisi *server*.

Dengan keluarnya versi baru PHP 7, beberapa keuntungan bisa diperoleh antara lain jauh lebih cepat daripada rilis stabil populer sebelumnya (PHP 5.6), meningkatkan penanganan kesalahan, mendukung Deklarasi Tipe yang lebih ketat untuk argumen fungsi dan mendukung operator baru (seperti operator spaceship: $\lt;=>$).

Pada penelitian ini, banyak digunakan fungsi matematika berkaitan dengan ilmu ukur sudut. PHP mendukung banyak fungsi berkaitan dengan ilmu ukur sudut seperti: `pi()` untuk menentukan nilai π , `sin()` untuk menghitung nilai sinus suatu sudut, `cos()` untuk menghitung nilai cosinus suatu sudut, `asin()` untuk menghitung arcus sinus, `deg2rad()` untuk mengubah dari satuan derajat menjadi radian dan masih banyak lagi. Fungsi-fungsi tersebut sangat berguna dalam perhitungan waktu adzan.

3.5 Nginx

Nginx [dibaca: engine x] adalah HTTP dan *reverse proxy server*, *server proxy email*, dan *server proxy TCP / UDP generik*, yang aslinya ditulis oleh Igor Sysoev. Untuk waktu yang lama, telah berjalan di banyak situs Rusia yang banyak dimuat termasuk Yandex, Mail.Ru, VK, dan Rambler. Menurut Netcraft, nginx melayani atau proxied 30,46% situs tersibuk di bulan Januari 2018. Berikut adalah beberapa yang sukses: Dropbox, Netflix, Wordpress.com, FastMail.FM (Nginx Foundation, 2018).

Perangkat lunak Nginx memiliki fitur dasar sebagai *server web* sebagai berikut

1. Melayani *file* statis dan indeks, autoindexing; buka *file* descriptor cache;
2. Melayani *file* statis dan indeks, autoindexing; buka *file* descriptor cache;
3. Accelerated reverse proxying dengan caching; load balancing dan fault tolerance;
4. Dukungan yang dipercepat dengan caching *server* FastCGI, uwsgi, SCGI, dan memcached; load balancing dan fault tolerance;
5. Arsitektur modular. Filter meliputi gzipping, rentang byte, respon chunked, XSLT, SSI, dan filter transformasi gambar. Beberapa inklusi SSI dalam satu halaman dapat diproses secara paralel jika ditangani oleh *server proxy* atau FastCGI / uwsgi / SCGI;
6. Dukungan SSL dan TLS SNI;
7. Dukungan HTTP/2 dengan prioritas berbasis bobot dan ketergantungan.

3.6 *Cloud Computing*

Komputasi awan (*Cloud Computing*) adalah akses berdasarkan permintaan, melalui internet, ke sumber daya komputasi—aplikasi, *server* (*server* fisik dan *server* virtual), penyimpanan data, alat pengembangan, kemampuan jaringan, dan banyak lagi—dihosting di pusat data jarak jauh yang dikelola oleh penyedia layanan *cloud* (*cloud services provider* atau CSP). CSP menyediakan sumber daya ini dengan biaya berlangganan bulanan atau menagihnya sesuai penggunaan. (Sai Vennam, 2022)

Dibandingkan dengan TI lokal tradisional, dan bergantung pada layanan *cloud* yang Anda pilih, komputasi awan membantu melakukan hal berikut: (Sai Vennam, 2022)

1. Biaya TI yang lebih rendah: *Ccloud* memungkinkan meniadakan sebagian atau sebagian besar biaya dan upaya untuk membeli, menginstal, mengonfigurasi, dan mengelola infrastruktur lokal.
2. Tingkatkan kelincahan (*agility*) dan nilai waktu (*time-to-value*): Dengan *cloud*, organisasi dapat mulai menggunakan aplikasi perusahaan dalam hitungan menit, alih-alih menunggu berminggu-minggu atau berbulan-bulan agar TI merespons permintaan, membeli dan mengonfigurasi perangkat keras pendukung, dan menginstal perangkat lunak. *Ccloud* juga memungkinkan untuk memberdayakan pengguna tertentu—khususnya pengembang dan ilmuwan data—untuk membantu dalam hal perangkat

lunak dan mendukung infrastruktur.

3. Pengembangan (scale) lebih mudah dan hemat biaya: *Ccloud* memberikan elastisitas—daripada membeli kelebihan kapasitas yang tidak terpakai selama periode lambat, kapasitas dapat ditingkatkan dan diturunkan sebagai respons terhadap lonjakan dan penurunan lalu lintas. Aplikasi dapat disebar dengan memanfaatkan jaringan global penyedia *cloud*.

Istilah 'komputasi awan' juga mengacu pada teknologi yang membuat *cloud* bekerja. Ini mencakup beberapa bentuk infrastruktur TI virtual—*server*, perangkat lunak sistem operasi, jaringan, dan infrastruktur lain yang diabstraksikan, menggunakan perangkat lunak khusus, sehingga dapat digabungkan dan dibagi terlepas dari batasan perangkat keras fisik. Misalnya, *server* perangkat keras tunggal dapat dibagi menjadi beberapa *server* virtual.

3.7 PaaS (*Platform As A Service*)

Platform As A Service menawarkan lingkungan *runtime* untuk aplikasi. Ini juga menawarkan alat pengembangan dan penyebaran yang diperlukan untuk mengembangkan aplikasi. PaaS memiliki fitur alat *point-and-click* yang memungkinkan non-pengembang untuk membuat aplikasi *web*. (tutoriapoint, 2021)

App Engine dari Google dan Force.com adalah contoh vendor yang menawarkan PaaS. Pengembang dapat masuk ke situs *web* ini dan menggunakan

API bawaan untuk membuat aplikasi berbasis *web*.

Keuntungan menggunakan PaaS (Javatpoint, 2022)

1. Pengembangan yang Disederhanakan

PaaS memungkinkan pengembang untuk fokus pada pengembangan dan inovasi tanpa mengkhawatirkan manajemen infrastruktur.

2. Risiko lebih rendah

Tidak perlu investasi di muka dalam perangkat keras dan perangkat lunak. Pengembang hanya membutuhkan PC dan koneksi internet untuk mulai membangun aplikasi.

3. Fungsi bisnis bawaan

Beberapa vendor PaaS juga menyediakan fungsionalitas bisnis yang sudah ditentukan sehingga pengguna dapat menghindari membangun semuanya dari awal dan karenanya dapat langsung memulai proyek saja.

4. komunitas instan

Vendor PaaS sering menyediakan komunitas online di mana pengembang bisa mendapatkan ide untuk berbagi pengalaman dan mencari saran dari orang lain.

5. Skalabilitas

Aplikasi yang disebarakan dapat menskalakan dari satu hingga ribuan pengguna tanpa perubahan apa pun pada aplikasi.

Kekurangan lapisan komputasi awan PaaS (Javatpoint, 2022)

1. Penguncian vendor

Kita harus menulis aplikasi sesuai dengan *platform* yang disediakan oleh vendor PaaS, sehingga migrasi aplikasi ke vendor PaaS lain akan menjadi masalah.

2. Privasi Data

Data perusahaan, apakah itu penting atau tidak, akan bersifat pribadi, sehingga jika tidak terletak di dalam dinding perusahaan, dapat ada risiko dalam hal privasi data.

3. Integrasi dengan aplikasi sistem lainnya

Mungkin saja beberapa aplikasi bersifat lokal, dan beberapa ada di *cloud*. Jadi akan ada kemungkinan peningkatan kompleksitas ketika kita ingin menggunakan data yang di *cloud* dengan data lokal.

3.8 PaaS Heroku

Heroku adalah *Platform As A Service* (PaaS) berdasarkan sistem *container* terkelola untuk membangun, menjalankan, dan mengelola aplikasi modern di *cloud*. Fleksibilitas *platform* memungkinkan Anda untuk menggunakan bahasa atau kerangka kerja pilihan Anda, serta pola arsitektur populer seperti layanan mikro. Pengalaman pengembang yang dirancang dengan cermat mencakup alat, layanan, dan ekosistem pihak ketiga yang terintegrasi. Sebagai PaaS, Heroku sepenuhnya dikelola, artinya Heroku menangani *server*, perangkat keras, dan infrastruktur, sehingga pengguna dapat tetap fokus pada aplikasi. Ratusan ribu

pengembang menjalankan aplikasi di Heroku mulai dari prototipe sederhana hingga perusahaan global (Salesforce.com, 2022).

Heroku adalah *platform* sebagai layanan yang didasarkan pada sistem wadah terkelola, dengan layanan data terintegrasi dan ekosistem yang kuat, untuk menerapkan dan menjalankan aplikasi modern. Pengalaman pengembang Heroku adalah pendekatan aplikasi-sentris untuk pengiriman perangkat lunak, terintegrasi dengan alat dan alur kerja pengembang paling populer saat ini.

Aplikasi mulai memengaruhi dunia saat pelanggan mulai berinteraksi dengannya. Mengeluarkan aplikasi di alam liar, keluar ke Internet dengan cepat, dan mengulangi, cepat, adalah apa yang dapat membuat atau menghancurkan perusahaan.

Heroku berfokus tanpa henti pada aplikasi dan pengalaman pengembang di sekitar aplikasi. Heroku memungkinkan perusahaan dari semua ukuran merangkul nilai aplikasi, bukan gangguan perangkat keras, atau gangguan *server* - virtual atau lainnya.

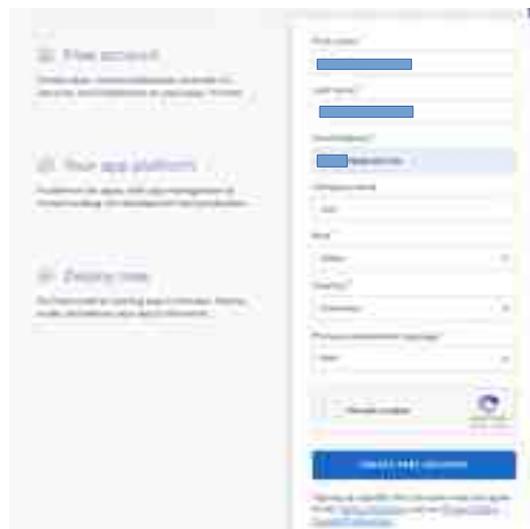
3.8.1 Registrasi Heroku

Heroku adalah layanan PaaS *Cloud Computing* yang dipakai secara bersama oleh banyak pengguna. Pengguna harus punya akun supaya dapat menggunakan layanan PaaS Heroku. Dengan akun tersebut, pengguna harus *login* untuk memperoleh layanan. Supaya mendapatkan akun, pengguna harus registrasi terlebih dahulu. Salah satu keuntungan menggunakan layanan PaaS Heroku

adalah adanya akun gratis yang bisa dipakai. Dengan akun tersebut, pengguna dapat mencoba membangun aplikasinya, namun dengan batasan-batasan tertentu. Namun, akun gratis hanya boleh digunakan untuk keperluan non komersial seperti pembuktian konsep, proyek MVP dan proyek personal (Salesforce, 2022).

Berikut ini adalah langkah-langkah untuk melakukan registrasi (membuat akun) pada layanan PaaS Heroku versi gratis.

1. Buka layanan registrasi Heroku melalui <https://signup.heroku.com/>, selanjutnya akan muncul beberapa isian sebagai berikut.

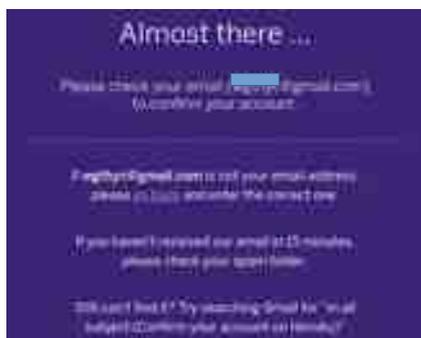


Gambar 3.2 Layanan Registrasi Heroku

Beberapa data yang diperlukan adalah first name, last name, email, company name, role, country dan primary development language. Selanjutnya klik tombol create free account.

2. Selanjutnya muncul dialog untuk konfirmasi kebenaran alamat email yang

dipakai untuk registrasi.



Gambar 3.3 Konfirmasi Email

Pengguna harus membuka alamat email yang dipakai untuk mendaftar layanan PaaS Heroku. Heroku akan mengirim email yang berisi tautan untuk aktivasi akun. Jika tautan di-klik, maka muncul dialog pengaturan *password*.



Gambar 3.4 Pengaturan *Password*

Ketentuan berkaitan dengan *password* pada Layanan PaaS Heroku adalah minimal delapan karakter dan mengandung kombinasi huruf angka simbol.

3. Apabila registrasi berhasil, maka muncul pesan selamat datang pada

layanan PaaS Heroku.



Gambar 3.5 Akun Heroku Berhasil Dibuat

Sampai pada tahap ini, berarti akun sudah dapat dipakai untuk masuk pada layanan PaaS Heroku.

3.8.2 Heroku CLI (*Command Line Interpreter*)

Heroku CLI dibangun dengan Open CLI Framework (oclif), yang dikembangkan di dalam Heroku/Salesforce. oclif tersedia sebagai kerangka kerja bagi pengembang mana pun untuk membangun CLI besar atau kecil. Kerangka kerja ini mencakup *generator* CLI, pembuatan dokumentasi otomatis, dan infrastruktur pengujian (Salesforce, 2022).

Kode untuk Heroku CLI juga open source. Itu tidak memerlukan Node.js atau dependensi lain untuk dijalankan. Kecuali Anda menginstal paket Debian/Ubuntu atau menggunakan npm install, CLI berisi biner Node.js sendiri yang tidak bertentangan dengan aplikasi lain.

Untuk menginstall Heroku CLI, dibutuhkan perangkat lunak Git. Untuk

menginstall Git pada Linux Ubuntu digunakan perintah berikut

```
# apt install git
```

Selanjutnya Heroku CLI dapat diinstall sesuai *platform* sistem operasi yang dipakai. Pada sistem operasi Linux, Heroku CLI dapat diinstall menggunakan *file* tar ball (tar.gz). *File* tar ball dapat didownload dari tautan <https://cli-assets.heroku.com/heroku-linux-x64.tar.gz>. Instalasi Heroku CLI juga dapat dilakukan menggunakan skrip instalasi. Pada sistem operasi Linux Distro Ubuntu dapat dilakukan dengan menjalankan 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. Untuk memastikan, apakah Heroku CLI sudah bekerja, bisa diperiksa dengan 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.

3.8.3 Membuat Aplikasi Heroku

Membuat aplikasi pada PaaS Heroku dapat dilakukan dengan dua cara yaitu melalui *web* dan melalui CLI Heroku.

Membuat Aplikasi PaaS Heroku Lewat *Web*

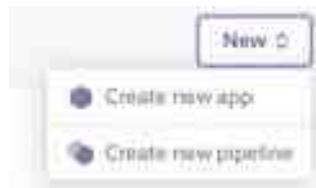
Langkah-langkah membuat aplikasi melalui *web* adalah sebagai berikut.

1. Buka halaman *login* <https://id.heroku.com/login> kemudian *Login* ke Heroku



Gambar 3.6 *Login* Melalui Web Heroku

2. Jika berhasil, maka masuk pada halaman dashboard. Kemudian pilih menu New->Create new app untuk membuat aplikasi baru



Gambar 3.7 Membuat Aplikasi Melalui Web Heroku

3. Selanjutnya akan muncul dialog untuk membuat aplikasi. Ditentukan nama aplikasi yang akan dibuat dan negara.



Gambar 3.8 Dialog Membuat Aplikasi

4. Selanjutnya pilih Create app. Jika pembuatan aplikasi berhasil, maka selanjutnya akan masuk pada dashboard aplikasi



Gambar 3.9 Dashboard Aplikasi PaaS Heroku

Sampai tahap ini, maka aplikasi PaaS Heroku berhasil dibuat. Aplikasi yang berhasil dibuat masih dalam keadaan kosong. Aplikasi yang sudah dibuat dapat diakses melalui URL [https://\[nama-aplikasi\].herokuapp.com/](https://[nama-aplikasi].herokuapp.com/)

Membuat Aplikasi PaaS Heroku Menggunakan CLI

Langkah-langkah membuat aplikasi melalui CLI adalah sebagai berikut.

1. *Login* PaaS Heroku melalui CLI, ketikkan perintah berikut pada *shell*

```
$ heroku login
```

Jika komputer ada fasilitas *browser* GUI, maka halaman *login* CLI *web* akan terbuka. Jika tidak ada fasilitas *browser* GUI, maka bisa dibuka halaman URL yang mana tautannya ditampilkan pada *shell*.

```
~/heroku/bin-adaah-262287$ heroku login
Warning: heroku update available from 7.10.0 to 7.10.1
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/ed8ea2a5-3ee7-42e8
-8a39-4429a-7a5d419540af1z7m8AWTAA
YCA_M0HEF_D
```

Gambar 3.10 Tautan URL *Login* PaaS Heroku

- Halaman yang tampil adalah sebagai berikut



Gambar 3.11 Halaman *Login* CLI

Pada halaman *Login* Heroku CLI diklik tombol *Login*

- Selanjutnya akan terbuka halaman untuk memasukkan *user* (berupa email) dan *password*



Gambar 3.12 Halaman *Login* PaaS Heroku

4. Apabila berhasil *Login*, maka muncul pesan berikut pada *web*



Gambar 3.13 Pesan Berhasil *Login*

5. Pada *shell* juga muncul pesan berikut

```

root@kali:~/heroku/wa-silran-2022# heroku login
Warning: heroku update available from 7.14.0 to 7.15.0
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/wd8ea2a5-2ee7-42e8
-8a39-44[REDACTED]ix0DAuMjQ1lJlsmS41N646AFix7qeSAMIAX
VGA, Manu...
Logging in... done
Logged in as [REDACTED]@gmail.com

```

Gambar 3.14 Pesan Pada Shell Berhasil *Login* Heroku

6. Selanjutnya untuk membuat aplikasi PaaS Heroku cukup diberi perintah

create pada Heroku sebagai berikut (Misalnya nama aplikasinya ws-adzan-202207).

```

~/heroku/ws-adzan-202207$ heroku create ws-adzan-202207
Warning: heroku update available from 7.31.1 to 7.39.1.
Creating ws-adzan-202207... done
https://ws-adzan-202207.herokuapp.com/ | https://git.heroku.com/ws-adzan-202207.git

```

Gambar 3.15 Aplikasi PaaS Heroku Berhasil Dibuat

Dengan demikian aplikasi bernama ws-adzan-202207 berhasil dibuat. *browser* dibuka URL <https://ws-adzan-202207.herokuapp.com/>. Karena materi halaman belum di-*deploy* (disebarkan), tampilan yang muncul adalah sebagai berikut.



Gambar 3.16 Halaman Kosong PaaS Heroku

3.8.4 *Deploy* Aplikasi Heroku

Untuk melakukan *deploy* halaman pada PaaS Heroku dapat dilakukan dengan tiga macam cara yaitu menggunakan Heroku Git, menghubungi Github dan *deploy* aplikasi Docker. Penjelasan masing-masing cara diuraikan pada paragraf-paragraf berikut.

3.8.4.1 *Deploy* Menggunakan Heroku Git

Langkah untuk menggunakan git pada *shell* atau tool GUI untuk *deploy* aplikasi

Install the Heroku CLI adalah sebagai berikut

1. Download dan install CLI Heroku.
2. Jika sudah tersedia, bisa langsung *login* ke PaaS Heroku melalui *shell*.

```
$ heroku login
```

3. Selanjutnya dilakukan cloning repository

Misalnya menggunakan git untuk cloning source code ws-adzan-202207 pada mesin lokal

```
$ heroku git:clone -a ws-adzan-202207
$ cd ws-adzan-202207
```

4. Melakukan *deploy* perubahan

Semua perubahan atau penambahan *file* source code dilakukan pada tahap ini. Selanjutnya apabila sudah siap, maka *deploy* dapat dilakukan dengan beberapa langkah berikut.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

3.8.4.2 Menghubungi Github

Cara yang ini dilakukan dengan menghubungkan aplikasi *web* Heroku ke GitHub untuk mengaktifkan perbedaan dan *deploy* kode. Langkah-langkah yang dilakukan pada cara ini adalah sebagai berikut.

1. Tampilkan perbedaan kode pada GitHub

Menghubungkan aplikasi ke repositori GitHub untuk melihat perbedaan

komit di log aktivitas.

2. *Deploy* perubahan menggunakan GitHub

Menghubungkan ke repositori memungkinkan untuk menerapkan cabang ke aplikasi.

3. *Deploys* otomatis dari GitHub

Pilih cabang untuk disebarkan secara otomatis setiap kali push (dorong).

4. Buat aplikasi review pada pipeline

Pipeline yang terhubung ke GitHub dapat mengaktifkan aplikasi review, dan membuat aplikasi untuk permintaan pull (tarik) baru.

3.8.4.3 *Deploy* Aplikasi Docker

Heroku Container Registry memungkinkan untuk menerapkan aplikasi berbasis Docker ke Heroku. Langkah-langkah *deploy* aplikasi docker adalah sebagai berikut.

1. Instal CLI Heroku

Download dan install CLI Heroku. Jika belum dilakukan, maka masuk ke akun Heroku dan bisa diikuti petunjuk untuk membuat kunci publik SSH baru.

```
$ heroku login
```

2. *Login* ke Container Registry

Untuk keperluan pada cara ini, harus menyiapkan Docker secara lokal.

Output akan terlihat saat menjalankan perintah ini.

```
$ docker ps
```

Selanjutnya, dapat dilakukan sign pada Container Registry.

```
$ heroku container:login
```

3. Push aplikasi berbasis Docker

Build Dockerfile in the directori kerja dan push image Docker.

```
$ heroku container:push web
```

4. *Deploy* perubahan

Lepaskan image baru yang dipush untuk *deploy* aplikasi.

```
$ heroku container:release web
```

Demikian uraian tiga macam cara untuk melakukan *deploy* aplikasi ke layanan PaaS Heroku yang disarikan dari petunjuk pada dashboard Heroku (Salesforce, 2022).

BAB 4 METODE PENELITIAN

4.1 Bahan Penelitian

Bahan materi yang digunakan dalam penelitian berupa buku teks maupun tulisan lain yang berhubungan dengan layanan *server cloud computing* terutama layanan *Platform As A Services*. Layanan *cloud computing* untuk perancangan menggunakan layanan *cloud computing* lokal yang ada di institusi. Setelah sistem sudah berjalan dengan betul, dicoba untuk dilakukan migrasi ke layanan *cloud computing* publik.

Sebagai layanan *web service* digunakan perangkat lunak PrayTimes dari situs <http://praytimes.org>. Perangkat lunak ini berupa skrip PHP yang dapat menampilkan waktu shalat menurut beberapa metode dan kriteria. Skrip ini juga bisa dimodifikasi supaya menghasilkan layanan *web service* yang diinginkan. Modifikasi dilakukan agar layanan *web service* menjadi lebih mudah untuk diterapkan pada perangkat lunak yang memerlukan.

4.2 Alat

Alat yang digunakan dalam penelitian ini berupa perangkat keras dan perangkat lunak. Perangkat lunak yang digunakan dalam penelitian berupa sistem

operasi dan perangkat lunak pendukung lain sebagai berikut.

1. Sistem Operasi Mageia 6
2. *Server web* Nginx
3. Sekrip Bash
4. PHP
5. HTML
6. PrayTimes
7. *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*. Berikut ini disajikan spesifikasi penting peralatan *server* dan PC desktop yang digunakan.

1. *Server* publik *cloud computing Platform As A Service* Heroku
2. *Server* privat *cloud computing*
3. PC Desktop
 - a) Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
 - b) RAM 4 GB.
 - c) Tipe sistem 64 bit.
 - d) Gigabit Ethernet
 - e) Simpanan Hardisk

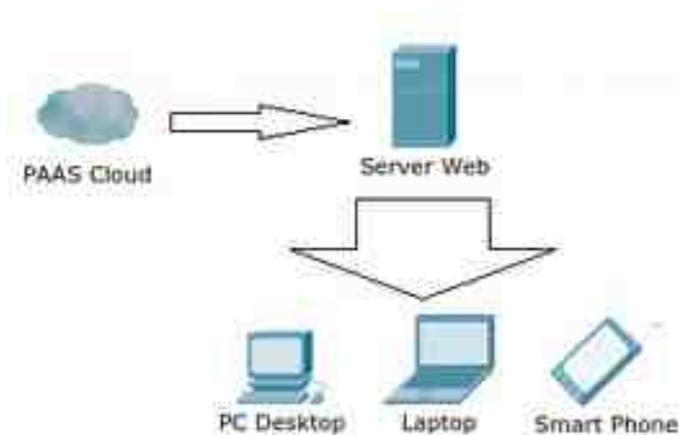
4.3 Metodologi Penelitian

Penelitian dilakukan melalui beberapa tahapan mulai dari rancangan perangkat keras, rancangan perangkat lunak sampai implementasi sistem.

4.3.1 Rancangan Perangkat Keras

Pada penelitian dilibatkan beberapa perangkat keras mulai dari *server* PaaS *Ccloud*, *server web* dan *workstation*. Workstation dapat berupa PC desktop, Laptop atau smart phone. Perangkat keras *server* PaaS *Ccloud* menggunakan layanan yang sudah tersedia di Internet. *Server web* bisa menggunakan *server* institusi atau *server* yang tersedia di Internet juga. Workstation bisa menggunakan komputer lokal milik pribadi.

Disain serta susunan perangkat keras untuk penelitian ditunjukkan pada Gambar 4.1. Perangkat-perangkat tersebut saling dihubungkan menggunakan sistem jaringan Internet.



Gambar 4.1 Disain Arsitektur Sistem

4.3.2 Rancangan Perangkat Lunak

Perangkat lunak yang dipakai dalam penelitian ini diimplementasikan pada masing-masing mesin perangkat keras. *Server PaaS cloud* menggunakan yang sudah tersedia di Internet. Yang diperlukan pada *server PaaS* adalah *platform PHP*. Supaya bisa melayani *web service* waktu shalat, pada *server PaaS cloud* diimplementasikan perangkat lunak *praytimes*. Layanan *web service* yang dihasilkan berupa format JSON.

Server web dipakai untuk mengimplementasikan *web service* dari *server PaaS cloud* dalam bentuk tampilan *web*. Pada *server web* diperlukan implementasi perangkat lunak Nginx, PHP dan HTML. *Server web* ini yang bertugas mengimplementasikan *web service* waktu shalat dari *server PaaS*. Selanjutnya *server web* akan melayani permintaan dari *workstation*.

Workstation berfungsi untuk mengakses layanan *server web*, yang dipakai bisa berupa PC desktop, Laptop ataupun smart phone. Pada *workstation* hanya diperlukan *browser*. Salah satu perangkat lunak *browser* yang dapat dipakai adalah Mozilla Firefox. *Browser-browser* yang juga dimungkinkan untuk dipakai seperti Google Chrome atau *browser* lain yang berbasis pada *smart phone* (Android, iPhone).

4.3.3 Tahapan Penelitian

Penelitian dilakukan dengan tahapan-tahapan mulai penentuan objek

sampai dengan pengambilan kesimpulan sebagai berikut.

Persiapan:

1. Pengumpulan bahan dan materi
2. Mempelajari bahan dan materi
3. persiapan perangkat keras (pada PaaS *Cloud*, *server web*, PC Desktop)
4. persiapan perangkat lunak (pada PaaS *Ccloud*, *server Web*, PC Desktop)

Pelaksanaan:

1. penyusunan program pada *server* lokal (pada *server Web*)
2. pembuatan akun pada layanan PaaS Heroku (pada PaaS *Ccloud*)
3. pembuatan aplikasi PaaS Heroku (pada PaaS *Ccloud*)
4. *deploy* aplikasi pada layanan PaaS Heroku (pada PaaS *Ccloud*)
5. pengujian aplikasi *web service* (pada PC Desktop)
6. pengujian menggunakan aplikasi *Client Side* (pada PC Desktop)
7. analisis hasil

Penyelesaian:

1. Penyusunan hasil penelitian
2. Pembuatan laporan

BAB 5 IMPLEMENTASI DAN PEMBAHASAN

5.1 Implementasi

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*.

5.1.1 Penyusunan Program Pada *Server* Lokal

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.

5.1.1.1 Program PrayTime.php

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 Ccloud Computing* adzan. Sekrip lengkap program PrayTime.php ditampilkan pada Lampiran.

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 waktu adzan dalam layanan web service
.....
?>
```

Langkah pertama menggunakan kode program PrayTime2022.php adalah menyertakan kode tersebut pada skrip 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).

5.1.1.2 Program index.php

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*. Kode program lengkap halaman indeks disertakan pada Lampiran.

Macam *web service* yang disediakan

1. `ws-adzan`: menentukan waktu adzan pada lokasi
2. `ws-adzan-tabel`: membuat tabel abadi waktu adzan pada tahun tertentu

Parameter Yang Bisa Diatur

1. `me`: method yang dipakai
opsi:

[0] Ithna Ashari

[1] University of Islamic Sciences, Karachi

[2] Islamic Society of North America (ISNA)

[3] Muslim World League (MWL)

[4] Umm al-Qura, Makkah

[5] Egyptian General Authority of Survey

[6] Institute of Geophysics, University of Tehran

[7] Kemenag

[8] Muhammadiyah

2. la: latitude letak tempat

3. lo: longitude letak tempat

4. tz: timeZone daerah waktu

5. aj: metode juristic untuk menentukan waktu Ashar

opsi:

[0] Shafii (default)

[1] Hanafi

6. dm: menit setelah mid-day untuk waktu dhuhur (0 default)

7. hl: menyesuaikan metode untuk garis lintang yang lebih tinggi

opsi:

[0] No adjustment (default)

[1] middle of night

[2] 1/7th of night

[3] angle/60th of night

8. ye: menentukan tahun perhitungan, terutama untuk membuat tabel abadi

Nilai Default Parameter

Jika parameter tidak diatur nilainya maka digunakan nilai default untuk menentukan waktu adzan di kota Yogyakarta menggunakan metode Muhammadiyah

1. metode=8 (Muhammadiyah)
2. latitude=-7.7956
3. longitude=110.3695
4. time zone=7
5. juristik ashar=0 (shafii)
6. waktu dhuhur=0 (setelah mid day)
7. untuk garis lintang tinggi=0(tidak ada penyesuaian)
8. tahun=sekarang

Cara Memggunakan Web Service

Data waktu adzan dapat diperoleh dengan dua cara:

1. menggunakan contoh-contoh tautan
2. menggunakan fungsi GENERATE

5.1.1.3 Program ws-adzan.php

`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. Keseluruhan kode program `ws-adzan` disertakan pada Lampiran. *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.

5.1.1.4 Program `ws-adzan-tabel.php`

`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 fajar, 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;isyah
20220101;04:06;05:25;11:42;15:09;17:59;17:59;19:29
.....
```

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. Keseluruhan kode program `ws-adzan-tabel` disertakan pada Lampiran.

5.1.1.5 Program `ws-generate.php`

`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). Keseluruhan kode program `ws-generate.php` disertakan pada Lampiran.

5.1.2 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: xxxxxx (disembunyikan)
- last name: .
- Email: xxxxxx@gmail.com (disembunyikan)
- 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.

5.1.3 Pembuatan Aplikasi PaaS Heroku

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.

1. Login ke PaaS Heroku

```
user01@localhost:~$ 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/a417f3e1-85b0-4f7d-
887e-76d4d3ec8bac?
requ [REDACTED] U1bgYA-
FNdG [REDACTED] mm1M6V6qe1XEGI
Logging in... done
Logged in as [REDACTED]@gmail.com
```

2. Membuat Aplikasi

```
user01@localhost:~/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,

5.1.4 Deploy Aplikasi Pada Layanan PaaS Heroku

Setelah aplikasi ws-adzan berhasil dibuat, langkah selanjutnya adalah

deploy semua sekrip 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.

```
user01@localhost:~/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.

1. Masuk pada direktori lokal aplikasi ws-adzan

```
user01@localhost:~/heroku$ cd ws-adzan
```

2. Menambahkan daftar semua *file* yang ada pada direktori lokal ws-adzan

```
user01@localhost:~/heroku/ws-adzan$ git add .
```

3. Membuat daftar semua *file* yang akan di-*deploy* pada layanan PaaS

Heroku yang belum ada pada *server*.

```
user01@localhost:~/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-table.php
create mode 100644 ws-adzan.php
create mode 100644 ws-generate.php
```

4. *Deploy* semua *file* yang belum ada pada *server* Heroku.

```
user01@localhost:~/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*.

5.2 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.

5.2.1 Pengujian aplikasi Web Service

Tampilan awal layanan *web service* waktu shalat berisi informasi tentang perangkat lunak *ws-adzan* yang ditampilkan pada Gambar 5.1. Tampilan awal *web service* waktu shalat berisi beberapa tab yaitu Home, Parameter, Default Parameter dan Cara menggunakan.



Gambar 5.1 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*).



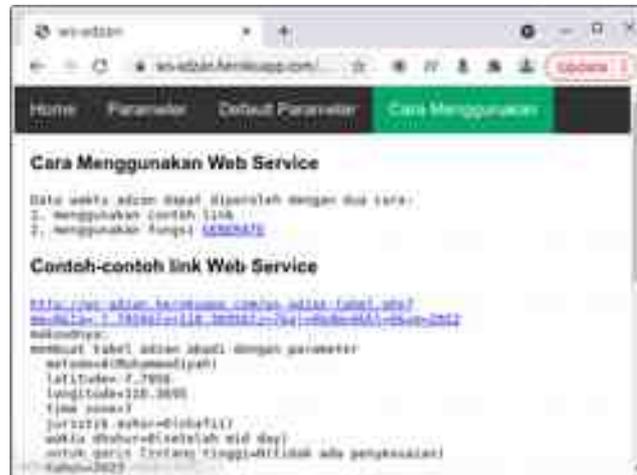
Gambar 5.2 Parameter Waktu Adzan

Pada tab Parameter (Gambar 5.2) 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. Tap Default Parameter ditampilkan pada Gambar 5.3.



Gambar 5.3 Nilai Default Parameter Waktu Adzan

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 5.4 Contoh Tautan Web Service

Tab selanjutnya pada layanan *web service* waktu shalat adalah cara menggunakan. Pada tab ini ditampilkan informasi cara menggunakan *web service* yaitu 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 pada Gambar 5.4.

Gambar 5.5 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 5.6.



Gambar 5.6 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.

5.2.2 Pengujian Menggunakan Aplikasi Client Side

Pada bagian ini 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* Keroku.

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()`. Kode php lengkap program `adzan-client`

ditampilkan pada Lampiran.



The screenshot shows a web browser window with the title 'Adzan'. The address bar contains 'localhost:3000'. The main content of the page is titled 'Waktu Shalat Yogyakarta' and includes the date 'tanggal: 17-07-2022'. Below this is a table with prayer times for various times of the day.

Waktu	Waktu
Fajar	04:39
Tertit	05:53
Dzuhur	11:45
Ashar	15:06
Maghrib	17:36
Isha	18:50

Gambar 5.7 Aplikasi Waktu Shalat

Hasil tampilan program adzan-client ditampilkan pada Gambar 5.7. Contoh waktu shalat yang ditampilkan adalah untuk kota Yogyakarta pada tanggal 17 Juli 2022.

BAB 6 KESIMPULAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari hasil pembahasan dan percobaan dalam penelitian ini adalah sebagai berikut.

1. Layanan *web service* waktu shalat berhasil diterapkan pada layanan PaaS *cloud computing* Heroku.
2. Layanan *web service* waktu shalat berhasil disusun berdasarkan beberapa metode dan kriteria dari PrayTime.org serta ditambah kriteria dari Kemenag Indonesia dan Ormas Muhammadiyah Indonesia.
3. *Web service* berhasil dibuat dan digunakan oleh aplikasi waktu shalat berbasis *web*.

6.1 Saran

Saran yang diajukan untuk pengembangan dan penelitian lebih lanjut dari penelitian ini adalah sebagai berikut.

1. Layanan *web service* waktu shalat dicobakan pada sistem jam shalat digital yang menerapkan pewaktu lokal untuk update waktu shalat.
2. Layanan *web service* waktu shalat dicobakan pada sistem embedded jam shalat digital menggunakan perangkat keras Raspbery atau Orange Pi.

Daftar Pustaka

- Achour, M., dkk, 2017, *PHP Manual documentation*, <http://php.net/manual/en/>
- Fajrillah, 2018, *Perancangan dan Implementasi Sistem Kuesioner Untuk Survei Berbasis Ccloud Computing*, jurnal OSF Preprints, October 31
- 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>
- Jakob Jenkov, 2014, *Web Service Message Formats*, <https://jenkov.com/tutorials/web-services/message-formats.html>
- Javatpoint, 2022, *Ccloud Computing Tutorial*, <https://www.javatpoint.com/platform-as-a-service>
- Nginx Foundation, 2018, *nginx*, <https://nginx.org/en/>
- Ninik Sri Lestari, 2018, *Implementasi Dan Optimalisasi Ccloud Computing Dalam Internet Of Thinks (IoT)*, Jurnal Isu Teknologi, Vol 13 No 2
- PrayTimes.org, 2011, *Pray Times Manual*, <http://praytimes.org/manual>
- Refsnes Data, 2022, *PHP Tutorial*, <https://www.w3schools.com/php/>
- Rudi Pradisetia Sudirdja, 2020, *Pemanfaatan Teknologi Ccloud Computing Dalam Reformasi Birokrasi Guna Mewujudkan Kejaksanaan Yang Profesional, Komunikatif dan Akuntabel*, Jurnal Hukum & Pembangunan, Vol 50, No 4
- Sai Vennam, 2022, *Cloud Computing*, <https://www.ibm.com/cloud/learn/cloud-computing>
- Salesforce, 2022, *Guided Price Estimator*, <https://www.heroku.com/pricing>
- Salesforce, 2022, *Heroku Dashboard*, <https://dashboard.heroku.com>
- Salesforce, 2022, *PaaS Heroku*, <https://www.heroku.com/what>
- Salesforce, 2022, *The Heroku CLI*, <https://devcenter.heroku.com/articles/heroku-cli#cli-architecture>
- Tutorialpoint, 2021, *Cloud Computing Tutorial*, https://www.tutorialspoint.com/cloud_computing/cloud_computing_platform_as_a_service.htm
- Tutorials Point, 2022, *Web Services Tutorial*, <https://www.tutorialspoint.com/webservices>
- 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)
- Yunie Cancer, 2016, *Platform As A Service (PaaS) Sebagai Layanan Sistem Operasi Cloud Computing*, Jurnal TIMES, Vol 5 No 1