

# Web Service Implementation For Freelance Platform With Extreme Programming Method

## [Implementasi Web Service Untuk Platform Freelance Digital Dengan Metode Extreme Programming]

Rayhanantha Akbar Putra Prasetyo<sup>1)</sup>, Irwan Alnarus Kautsar<sup>\*,2)</sup>

<sup>1)</sup>Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

<sup>2)</sup>Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

\*Email Penulis Korespondensi: [irwan@umsida.ac.id](mailto:irwan@umsida.ac.id)

**Abstract.** *Freelancing has been trending for a few years lately, popular among most people nowadays. Technologies are growing further giving new opportunities for freelancing jobs, but instead of storing freelancer data, platforms are serving different data of freelancers. Web service was made to provide such functionality that could integrate freelancer data to many platforms while they develop their own main features. Extreme programming is used as a development method which having an easy workflow to maintain and sustain codes. A test result of the Black Box method achieved a result that web service could handle requests with 80% percentage of success with an average response time of 500ms to 900ms. The result of this web service states it successfully distributes freelancer data in JSON format within seconds, by implementing this web service on every freelance platform hope it could help in hiring and recruiting freelance workers.*

**Keywords** – *web service, application programming interface, digital freelance*

**Abstrak.** *Freelancing telah menjadi tren selama beberapa tahun belakangan ini, populer di kalangan kebanyakan orang saat ini. Teknologi yang semakin berkembang memberikan peluang baru untuk pekerjaan freelancing, tetapi alih-alih menyimpan data freelancer, platform melayani data freelancer yang berbeda. Layanan web dibuat untuk menyediakan fungsionalitas yang dapat mengintegrasikan data freelancer ke banyak platform sementara mereka mengembangkan fitur utama mereka sendiri. Extreme programming digunakan sebagai metode pengembangan yang memiliki alur kerja yang mudah untuk memelihara dan mempertahankan kode. Hasil pengujian dengan metode Black Box mendapatkan hasil bahwa web service dapat menangani request dengan persentase keberhasilan 80% dengan waktu respon rata-rata 500ms hingga 900ms. Hasil dari web service ini menyatakan berhasil mendistribusikan data freelancer dalam format JSON dalam hitungan detik, dengan mengimplementasikan web service ini pada setiap platform freelance diharapkan dapat membantu dalam proses perekrutan pekerja freelance.*

**Kata Kunci** – *layanan web, program aplikasi antarmuka, digital freelance*

## I. PENDAHULUAN

Beberapa tahun terakhir, berbagai bidang industri pekerjaan yang mencakup secara global dan domestik mengalami peningkatan dalam bentuk fleksibilitas waktu. Salah satunya adalah banyaknya bidang pekerjaan yang bersifat kontrak jangka pendek, kegiatan magang, outsourcing (alih daya), berjualan secara daring, pekerja lepas (freelance) yang diberi upah sesuai kebutuhan dan permintaan (on-demand), dan lain-lain [1]. Beberapa pekerjaan mulai beralih untuk dilakukan secara remote atau online (daring), menghadirkan tantangan dan faktor-faktor baru yang harus dipertimbangkan [2].

Freelance merupakan salah satu pekerjaan yang mengadaptasi bentuk kerja secara remote, bekerja secara daring dengan lingkungan yang fleksibel. Freelance dapat dilakukan dalam berbagai bidang pekerjaan dan terutama yang berkaitan dengan teknologi, perencanaan masing-masing seorang individu harus jelas untuk menyesuaikan dengan minat dan bakat agar tidak salah dalam memilih sebuah pekerjaan [3]. Beberapa platform pada website menyediakan tempat bagi para freelancer untuk mempromosikan dan melakukan kontrak kerja dengan klien atau penyewa jasa freelancer, tapi masing-masing platform tidak memiliki integrasi data pekerja sehingga masing-masing platform memiliki data pekerjanya masing-masing.

Dengan teknologi yang semakin canggih, penggunaan teknologi telah membantu dalam berbagai bidang, penggunaan web service akan diimplementasikan dalam mengintegrasikan data pada platform secara aman dan terdesentralisasi. Web service merupakan sistem yang difungsikan untuk melakukan interaksi antar komputer, sehingga masing-masing perangkat dapat terhubung, memudahkan pertukaran informasi serta data yang dibutuhkan [4]. Web service bisa dikatakan sebagai aplikasi web yang berinteraksi menggunakan standar protokol application-to-application dibalik antarmuka aplikasi [5]. Pengimplementasian web service memudahkan dalam pemeliharaan dan

perbaikan sistem dalam integrasi data ke berbagai aplikasi dan platform suatu sistem melalui internet, intranet, maupun ekstranet [6].

Penelitian lain yang telah membahas mengenai perancangan web service serta penggunaan fungsi web service pada aplikasi berbasis mobile adalah “*Development of Web Service and Telegram Bot for Location-Based Health Service Information System*” pada tahun 2022 yang dilakukan oleh Agoeng Dwi Djoelianto membahas tahapan perancangan web service dan bagaimana fungsi web service bekerja pada aplikasi. Metode perancangan menggunakan framework botogram yang mendukung penggunaan webhooks dan hasil perancangan merupakan sebuah aplikasi bot telegram yang mengolah data layanan kesehatan melalui web service, maka dengan itu aplikasi ini bisa diakses secara publik sehingga masyarakat dapat menggunakannya [7].

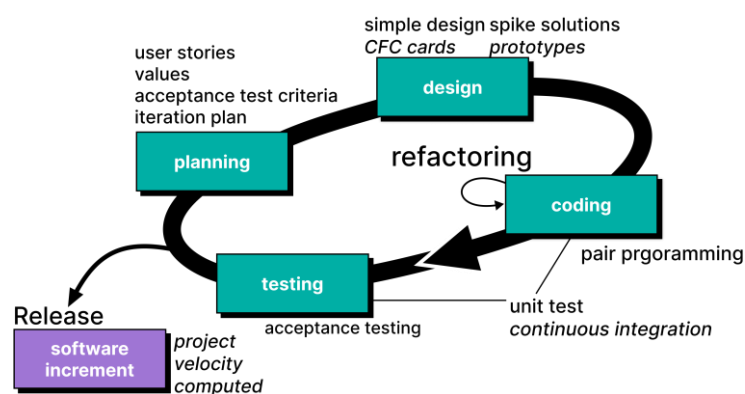
Penelitian lainnya adalah merancang web service untuk sistem informasi desa yang dilakukan oleh Wira Junardi dan rekan-rekannya terkait “*Sistem Informasi Desa Siaga Pangan Menghadapi Covid19 berbasis Web Service*” pada tahun 2020. Penelitian bertujuan untuk merancang sebuah sistem informasi untuk pasokan pangan pada sebuah desa. Metode yang digunakan dalam pengembangan sistem adalah Agile scrum agar sistem yang dikembangkan dapat diselesaikan secara cepat dan efektif. Pengujian pada sistem dilakukan menggunakan teknik Black Box dan hasil uji menyatakan fungsional sistem dapat bekerja sesuai harapan. Sistem hanya mencakup kontrol pada daerah pedesaan, tidak sampai meluas ke perkotaan atau provinsi [8].

Penelitian lain yang dilakukan oleh Ananda Dwi Prasetyo tentang “*Rancang Bangun Aplikasi Pelaporan Fasilitas Umum Berbasis Web Service Dalam Rangka Menuju Sidoarjo Smart City Dan Open Data*” membahas perancangan aplikasi untuk membantu melaporkan terkait kerusakan fasilitas yang ada pada daerah Sidoarjo. Penggunaan web service pada aplikasi digunakan untuk melakukan pertukaran informasi antara basis data dengan aplikasi dengan hasil yang menyatakan aplikasi dengan berbasis web service menunjukkan kelayakan dalam fungsionalitasnya sebagai aplikasi pelaporan fasilitas di daerah Sidoarjo [9].

Dengan demikian, web service merupakan sebuah aplikasi yang dikhususkan untuk melakukan pertukaran data antar sistem dan bersifat interoperabilitas. Web service sudah umum digunakan dalam berbagai sistem untuk menjaga kerahasiaan dan keamanan terhadap informasi data yang dipertukarkan antar sistem, membantu fleksibilitas sistem dalam mengolah dan mendistribusikan data [10]. Kehadiran web service membantu banyak dalam dunia pengembangan teknologi dengan tujuan mengamankan pertukaran informasi, beberapa web service dapat diakses secara publik namun beberapa hanya dapat diakses oleh pengguna yang memiliki otorisasi untuk menggunakan layanan web service. Integritas sistem yang saling terhubung melalui web service memberikan kemudahan kepada beberapa pengembang dalam memperoleh informasi yang dibutuhkan untuk sistem yang akan dikembangkan [11]. Pemeliharaan web service dikatakan cukup mudah dan tidak memberikan banyak beban yang harus dilakukan oleh pengembang lain jika sebelumnya telah menggunakan layanan web service tersebut, pengembang hanya cukup mendapatkan endpoint atau URL yang disediakan pada web service untuk melakukan interaksi dan permintaan data.

Dari beberapa hasil penelitian yang telah disajikan sebelumnya menunjukkan kurangnya penggunaan web service untuk integrasi data pada platform freelance. Integrasi data ditujukan untuk mempermudah [12] berbagai platform dalam pemeliharaan sehingga masing-masing platform bisa berfokus pada fitur lain yang akan menjadi keunggulan masing-masing platform. Oleh karena itu dibutuhkan penelitian lebih lanjut terkait pengembangan web service pada bidang pekerjaan freelance, dengan harapan penggunaan web service yang akan dikembangkan pada penelitian ini dapat membantu dalam mendistribusikan informasi ke berbagai platform freelance.

## II. METODE



Gambar 1. Alur Extreme Programming

. Extreme Programming (XP) merupakan metode pengembangan yang meringkas beberapa tahapan dalam pengembangan menjadi fleksibel, efisien dan adaptif. Metode ini cukup sederhana dan tidak rumit, serta mudah beradaptasi dengan adanya perubahan terhadap pengembangan dalam sistem yang bersifat ambigu dan tidak memiliki persyaratan kebutuhan yang jelas. Metode ini cocok untuk pengembangan secara individu atau dalam tim yang berskala kecil hingga medium [13]. Metode XP memiliki beberapa nilai yang mendasari bagaimana metode ini dilakukan, diantaranya [14]:

- a. Communication (Komunikasi)  
Masing-masing pengembang harus saling berkomunikasi dan menyampaikan apa yang harus disampaikan terkait proses pengembangan.
- b. Simplicity (Kesederhanaan)  
Seluruh aktifitas selama pengembangan harus dilakukan secara sederhana dan praktis dalam menyelesaikan sebuah masalah.
- c. Feedback (Umpan Balik)  
Umpan balik dari pengembang maupun pengguna dibutuhkan secara langsung untuk mengevaluasi kinerja sistem.
- d. Courage (Keberanian)  
Masing-masing pengembang harus berani dalam mencoba dan bereksperimen, dan yakin dalam melakukan pengembangan terhadap sistem untuk membentuk integritas dan keselarasan antar anggota pengembang.

Metode XP memiliki 4 tahapan utama dalam melakukan pengembangan aplikasi, berikut merupakan tahapan yang diperlukan dalam mengembangkan aplikasi menggunakan metode XP [15]:

1. Planning (Perencanaan)  
Tahapan ini dimulai dengan mengumpulkan prasyarat dan kebutuhan sistem untuk memberikan pemahaman serta gambaran yang jelas tentang fungsionalitas dan fitur utama pada sistem kepada pengguna. Dalam pengembangan web service, tahap dimulai dengan memberikan pemahaman tentang bagaimana sistem akan bekerja kemudian menganalisa kebutuhan agar sistem bisa digunakan oleh berbagai platform freelance.
2. Design (Perancangan)  
Tahap ini dilakukan pembuatan model alur distribusi data pada sistem setelah melakukan analisa kebutuhan pada tahap sebelumnya. Pemodelan menggunakan Data Flow Diagram (DFD) untuk memberikan kemudahan dalam memahami alur distribusi data ke platform.
3. Coding (Pengkodean)  
Pada tahap ini dilakukan implementasi dari rancangan model yang telah dibuat pada tahap sebelumnya menjadi sebuah kode program yang menghasilkan sistem prototipe. Pengembangan web service menggunakan bahasa pemrograman Typescript, dan untuk basis data yang digunakan dalam sistem adalah MongoDB.
4. Testing (Pengujian)  
Tahapan ini melakukan pengujian terhadap sistem yang telah dikembangkan. Pengujian ditujukan untuk mengetahui fungsionalitas dan fitur yang ada pada sistem bekerja sesuai perencanaan pada tahapan awal. Metode pengujian yang digunakan dalam menguji sistem adalah Black Box Testing, metode ini berfokus untuk menguji masing-masing fitur pada endpoint yang telah dibangun pada web service, untuk memastikan bahwa masing-masing endpoint menghasilkan keluaran yang sesuai dengan fungsinya.

### III. HASIL DAN PEMBAHASAN

#### 3.1 Planning

Sistem yang dikembangkan hanya menyediakan informasi dan beberapa data diri terkait para pekerja untuk digunakan pada platform *freelance*, informasi yang disediakan hanya mencakup informasi dasar yang dibutuhkan pekerja untuk membuat kontrak dengan klien dan melakukan interaksi serta komunikasi dalam bekerja. Sistem pada *web service* akan mengolah dan memberikan data pekerja yang dibutuhkan platform sesuai permintaan yang diinstruksikan melalui *endpoint* yang telah disediakan. Pengembangan sistem membutuhkan beberapa alat utama agar kinerja sistem sesuai dengan persyaratan yang telah direncanakan, alat yang digunakan bersifat gratis atau *open-source* dan telah mendukung beberapa *add-ons* diantaranya:

1. Visual Studio Code  
Kode editor yang sudah umum digunakan oleh berbagai pengembang dan telah menjadi *Integration Development Environment* (IDE) beberapa pengembang, yang artinya kode editor ini mendukung pengembangan dengan berbagai bahasa pemrograman karena sifatnya yang *open-source* dan komunitas yang aktif dalam mengembangkan alat bantu seperti *add-ons* dan lainnya [16].
2. Postman

Alat yang lingkupnya digunakan untuk melakukan serangkaian instruksi pada suatu URL melalui protokol HTTP atau HTTPS dan menampilkan respon dari instruksi yang dilakukan [17]. Postman umumnya digunakan untuk melakukan uji coba pada suatu *Application Programming Interface* (API), dan sistem yang akan dikembangkan ini merupakan salah satu dari jenis API yaitu *Web Service*.

### 3. NodeJS

NodeJS adalah lingkungan *runtime* berbasis *server* yang dibangun dari mesin Javascript V8 Chrome sehingga memungkinkan eksekusi kode Javascript secara *server-side*.

Pengembangan aplikasi web bisa dilakukan dengan cepat pada NodeJS karena didukung arsitektur pemrograman *event-driven* dan operasi *non-blocking I/O (asynchronous)* [18]. Penggunaan NodeJS sebagai *runtime server* diperuntukkan untuk memudahkan dalam mengembangkan aplikasi *web service* dengan menggunakan salah satu *library* yang ada pada Node.js yaitu Express. Express merupakan *framework* pengembangan yang menggunakan NodeJS pada bagian *back-end server-side* [19].

### 4. MongoDB

MongoDB merupakan basis data yang berbentuk *document-oriented model* (DOM), bersifat NoSQL (*No Structured Query Language*) yang artinya tidak memiliki relasi antar data. Basis data dengan bentuk DOM ini menggantikan konsep baris dan kolom dengan model yang lebih fleksibel yaitu dokumen, memberikan keunggulan untuk membentuk relasi yang kompleks dalam satu data atau dokumen [20]. MongoDB menjadi pilihan dalam pembuatan *web service* karena perubahan dalam dokumen yang bersifat fleksibel dan adaptif terhadap kebutuhan sistem.

### 5. MongoDB Compass

Alat yang membantu pengguna untuk berinteraksi dengan basis data MongoDB melalui antarmuka yang disediakan, sehingga interaksi dan pengolahan data pada basis data mudah dilakukan.

### 6. MongoDB Shell

Alat ini bekerja mirip dengan MongoDB Compass namun tidak ada antarmuka yang memberikan kemudahan dalam melakukan interaksi dan pengolahan data dalam basis data, seluruh aktifitas dilakukan melalui *Command Line Interface* (CLI) masing-masing sistem operasi dan hasil yang dari aktifitas melalui MongoDB Shell hanya berupa sebuah teks.

Alat-alat tersebut merupakan kebutuhan dasar dan wajib dalam mengembangkan *web service* ini agar sistem dapat berjalan sesuai dengan harapan, pada Tabel 1 diberikan spesifikasi lebih jelas tentang masing-masing alat yang telah disebutkan.

**Tabel 1.** Kebutuhan Alat Pengembangan

No	Alat / Software	Fungsi	Versi
1	Visual Studio Code	Kode editor	1.85.2
2	Postman	API Testing	10.22.0
3	NodeJS	Web Server	LTS 20.10.0
4	Express	Framework	4.18.2
5	MongoDB	NoSQL Database	7.0.2
6	MongoDB Compass	MongoDB GUI	1.41.0
7	MongoDB Shell	MongoDB CLI	2.1.1

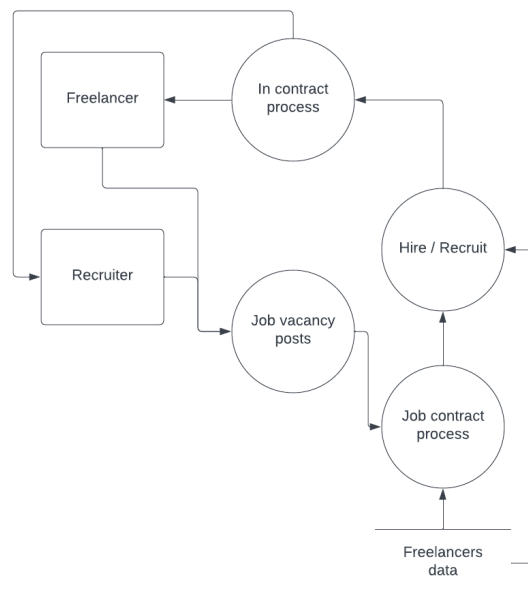
## 3.2 Design

Pemodelan pada sistem dilakukan menggunakan *Data Flow Diagram* (DFD) untuk mengetahui proses bagaimana data akan digunakan pada platform, dan mengetahui bagaimana proses platform dalam memasukkan data baru pada basis data melalui perantara *web service* yang dikembangkan. Alur interaksi yang dilakukan *web service* akan digambarkan dengan diagram Flowchart, hal ini untuk mengetahui bagaimana sistem dapat memvalidasi masukan dari platform sehingga data bisa tercatat dalam penyimpanan basis data. Penggambaran sederhana proses pertukaran data melalui *web service* ini digambarkan pada Gambar 2 yang merupakan DFD Level 0.



Gambar 2. DFD Level 0

DFD Level 0 menjelaskan dasar bagaimana *web service* ini bekerja dalam platform. *Web service* mengelola dan merespon permintaan dari platform untuk kemudian dikembalikan lagi ke platform dengan berupa data dalam format *Javascript Object Notation (JSON)*. Alur penggunaan *web service* digambarkan pada Gambar 3 yang menjelaskan arah bagaimana data yang ada pada sistem akan digunakan pada platform.



Gambar 3. DFD Level 1

Data para pekerja atau *freelancer* digunakan ketika proses pencarian kerja atau perekrutan pekerja, dan masing-masing memiliki akun yang mana bisa menjadi *recruiter* atau *freelancer*, sehingga masing-masing pekerja bisa memberikan pekerjaan atau mencari kerja. *Web service* ini hanya menyediakan data para pekerja kepada platform yang akan menggunakan, sehingga integrasi data pekerja terhubung antar platform.

### 3.3 Coding

Pada tahap ini dilakukan pengembangan *web service* pada kode editor Visual Studio Code. Bahasa pemrograman yang digunakan dalam pengembangan adalah Typescript. Bahasa pemrograman Typescript merupakan bahasa baru yang dirancang dan didasarkan pada Javascript. Sintaks Typescript dan Javascript relatif sama, namun Typescript memiliki beberapa fitur tambahan yang tidak dimiliki Javascript, seperti *strong typing*, pemrograman *object-oriented* yang lebih lengkap, dan akses pada fitur-fitur ECMAScript terbaru [21]. Pengkodean menggunakan Typescript dilakukan untuk mengembangkan *server* dari NodeJS dengan *framework* Express untuk memberikan sistem *web service* yang optimal kepada platform.

### 3.4 Testing

Pengujian dilakukan menggunakan *software* Postman dengan metode uji *Black Box Testing*, dilakukan pada setiap *endpoint* untuk mengetahui apakah masing-masing *endpoint* bekerja sesuai harapan. *Web service* akan menyediakan beberapa *endpoint* yang sudah dikembangkan melalui tahap pengkodean, masing-masing *endpoint* memiliki fungsionalitas yang berbeda-beda.

Pengujian menggunakan Postman dilakukan dengan fitur *functional test* yang disediakan oleh postman, pengujian ini melakukan uji pada fungsi *web service* apakah bekerja sesuai dengan rancangan sistem. Pengujian dilakukan

dengan jumlah 100 iterasi dan 100 *dummy user data*, pengujian terbagi menjadi 3 sesi dan dilakukan secara urut mulai dari atas ke bawah pada Tabel 2, beberapa *endpoint* yang diujikan sesuai sesinya:

**Tabel 2.** Urutan Pengujian Sistem

ENDPOINT	METHOD	SESI
/auth/register	POST	1
/auth/login	POST	1
/users	GET	1
/user/:username	GET	1
/user/:username	PATCH	1
/auth/logout	GET	1
/users/public	GET	1
/auth/register	POST	2
/auth/login	POST	2
/transaction/new	POST	2
/transaction/id/:id	PATCH	2
/transaction/payment_id/:payment_id	PATCH	2
/transactions	GET	2
/auth/register	POST	3
/auth/login	POST	3
/user/:username	DELETE	3

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	freelance-api	100	9m 40s	3300	534 ms
<b>All Tests</b> Passed (2732) Failed (568) Skipped (0)					

**Gambar 4.** Hasil *Functional Test* Sesi 1

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	100	11m 46s	2900	867 ms
<b>All Tests</b> Passed (2357) Failed (543) Skipped (0)					

**Gambar 5.** Hasil *Functional Test* Sesi 2

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	100	4m 10s	1238	664 ms
<b>All Tests</b> Passed (1148) Failed (90) Skipped (0)					

**Gambar 6.** Hasil *Functional Test* Sesi 3

Pengujian sesi 1 memberikan hasil bahwa dari 3300 pengujian, terdapat 2732 hasil uji yang menyatakan berhasil dan 568 pengujian yang dinyatakan gagal. Pengujian sesi 2 memberikan hasil dari 2900 pengujian, terdapat 2357 hasil uji yang berhasil dan 543 uji yang gagal. Pada pengujian sesi 3 memberikan hasil dari 1238 pengujian, 1148 hasil uji dinyatakan berhasil dan 90 uji dinyatakan gagal. Masing-masing hasil pengujian memiliki waktu respon dari *server* yang berbeda-beda, dapat dilihat pada (*Avg. Resp. Time*) rata-rata waktu respon yang merupakan seberapa lama *server* dalam memproses permintaan untuk memperoleh data yang diminta melalui *endpoint* masing-masing. Terdapat beberapa faktor yang memungkinkan jadi penyebab kegagalan, seperti masukan yang tidak sesuai dengan jenis masukan, password yang kurang rumit, nomor hp tidak sesuai standar Indonesia, kredensial pengguna tidak ditemukan dalam basis data, dan lainnya.

Hasil pengujian *endpoint* secara keseluruhan bisa dikatakan cukup berhasil, dilihat pada Tabel 3 diberikan informasi terkait masing-masing *endpoint* pada *web service* serta hasil pengujian masing-masing *endpoint*.

**Tabel 3.** Hasil Pengujian *Web Service*

ENDPOINT	METHOD	FUNGSI	HASIL
/users/public	GET	Meminta informasi ringkas semua pengguna	Berhasil mengambil informasi pengguna secara ringkas
/users	GET	Meminta informasi semua pengguna secara lengkap	Berhasil mengambil informasi pengguna secara lengkap
/transactions	GET	Meminta histori transaksi pengguna yang sedang <i>login</i>	Berhasil mendapatkan hasil transaksi pengguna yang sedang <i>login</i>
/user/:username	GET	Meminta informasi lengkap terkait pengguna yang <i>login</i> saat ini	Berhasil mengambil informasi terkait pengguna secara lengkap
/auth/logout	GET	Melakukan proses <i>logout</i> pengguna dari sistem, membatasi akses pengguna dalam sistem ketika tidak <i>login</i>	Berhasil melakukan <i>logout</i> pengguna sehingga pengguna butuh <i>login</i> lagi untuk mengakses sistem
/auth/register	POST	Mendaftarkan pengguna baru ke basis data	Berhasil mendaftarkan pengguna baru ke dalam sistem
/auth/login	POST	Melakukan <i>login</i> dengan masukan kredensial pengguna untuk mendapat akses ke dalam sistem	Berhasil melakukan <i>login</i> ke sistem sehingga pengguna memiliki akses dalam sistem

/transaction/new	POST	Membuat <i>log</i> transaksi baru	Berhasil membuat <i>log</i> transaksi baru
/user/:username	PATCH	Mengubah informasi terkait pengguna dengan informasi baru dengan username sebagai parameter	Berhasil mengubah informasi terkait pengguna seperti nama, alamat, dan no telpon
/transaction/id/:id	PATCH	Mengubah status transaksi dengan id transaksi sebagai parameter	Berhasil mengubah status <i>log</i> pada transaksi dengan id transaksi sebagai parameter
/transaction/payment_id/payment_id	PATCH	Mengubah status transaksi dengan payment_id pada transaksi sebagai parameter	Berhasil mengubah status <i>log</i> pada transaksi dengan payment_id transaksi sebagai parameter
/user/:username	DELETE	Menghapus informasi pengguna dari basis data	Berhasil menghapus informasi pengguna dari basis data

### 3.5 Implementasi Web Service

Pengimplementasian *web service* yang dikembangkan dapat dilakukan melalui pengembangan pada bagian *front-end* atau bisa digunakan sebagai penyedia data pekerja pada pengembangan di platform lain. Implementasi sederhana dapat dilakukan sebagai contoh menggunakan *endpoint* /users/public yang akan meminta informasi seluruh pengguna namun informasi pengguna yang ditampilkan hanya terbatas email dan username pengguna, dan informasi yang diminta hanya berjumlah 1 data per halaman dari seluruh informasi yang ada. Permintaan bisa disesuaikan dengan kebutuhan platform yang ingin menampilkan berapa jumlah data dalam satu permintaan.

```

{
  "status": "SUCCESS",
  "message": "Get all username and
email",
  "data": [
    {
      "identity": {
        "username": "rita91",
        "email":
"york_lowery@gmail.com"
      }
    },
    ],
  "totalPages": 85,
  "currentPage": 1,
  "sortBy": "",
  "sortOrder": "desc"
}

```

**Gambar 7.** Hasil *Endpoint* /users/public



Implementasi ini bisa dilakukan dengan menggunakan fungsi *fetch* yang dimiliki Javascript atau dengan *library* tambahan yaitu *Axios*. Data yang telah diminta dapat diproses lebih lanjut untuk ditampilkan pada halaman di website masing-masing platform atau di aplikasi mobile masing-masing platform.

## VII. SIMPULAN

Berdasarkan hasil implementasi dan uji coba yang dilakukan, didapatkan persentase hasil uji yang berhasil sebesar 83.86% dan yang gagal sebesar 16.14% dari total 7438 uji dengan rata-rata waktu respon server berkisar antara 500ms hingga 900ms, hasil uji dapat dinyatakan bahwa web service yang dikembangkan cukup layak untuk digunakan dalam integrasi data para freelancer di berbagai platform, maka masing-masing platform memiliki data pekerja yang sama sehingga masing-masing platform bisa berfokus pada pengembangan fitur lainnya. Metode Extreme Programming (XP) dapat digunakan dalam proses pengembangan sistem yang membutuhkan adaptasi dan perubahan dalam struktur sistem. Penelitian ini memiliki potensi untuk perkembangan selanjutnya dengan saran sebagai berikut:

1. Memberikan fitur enkripsi tingkat lanjut pada pengolahan data seperti password dan JWT session.
2. Memberikan fitur verifikasi pada email pengguna dalam web service, sehingga platform tidak perlu mengembangkan program untuk verifikasi pekerja.
3. Mengubah struktur kode sehingga respon dari web service melalui server dapat dipersingkat hingga 200ms.

Agar penggunaan web service ini dapat dipahami dan berkerja sesuai harapan pengguna, dibutuhkan dokumentasi terkait petunjuk teknis penggunaan sistem. Untuk mengetahui apakah sistem web service ini layak digunakan sebagai layanan yang membantu dalam industri dan platform digital freelance, dibutuhkan evaluasi lebih lanjut dengan metode yang tersedia.

## UCAPAN TERIMA KASIH

Saya ingin mengucapkan terima kasih atas kehadiran Allah SWT yang telah melancarkan segala urusan saya dalam masa perkuliahan hingga titik ini, tidak lupa juga saya ucapkan terima kasih kepada keluarga, teman-teman, dan saudara yang telah menyemangati dan berkontribusi besar maupun kecil dalam perancangan program dan penulisan karya ilmiah ini, tidak lupa juga ucapan terima kasih kepada Universitas Muhammadiyah Sidoarjo yang telah memberikan fasilitas dalam menunjang penelitian ini.

## REFERENSI

- [1] A. Y. Firdasanti, A. D. Khailany, N. A. Dzulkirom, T. M. P. Sitompul, and A. Savirani, "Mahasiswa dan Gig Economy: Kerentanan Pekerja Sambilan (Freelance) di Kalangan Tenaga Kerja Terdidik," *J. PolGov*, vol. 3, no. 1, pp. 195–234, Dec. 2021, doi: 10.22146/polgov.v3i1.2866.
- [2] F. Flood, "Leadership in the Remote, Freelance, and Virtual Workforce Era," in *Global Encyclopedia of Public Administration, Public Policy, and Governance*, A. Farazmand, Ed., Cham: Springer International Publishing, 2019, pp. 1–5. doi: 10.1007/978-3-319-31816-5\_3825-1.
- [3] A. Habibah and O. M. Siregar, "Pengaruh Karakteristik Individu Dan Pekerjaan Terhadap Kepuasan Kerja Freelance Generasi Millennial," *J. Ekon. Akunt. Dan Manaj. Indones.*, vol. 2, no. 01, Art. no. 01, Apr. 2023.
- [4] P. B. Ramadhanu and A. T. Priandika, "RANCANG BANGUN WEB SERVICE API APLIKASI SENTRALISASI PRODUK UMKM PADA UPTD PLUT KUMKM PROVINSI LAMPUNG," *J. Teknol. Dan Sist. Inf.*, vol. 2, no. 1, Art. no. 1, Mar. 2021, doi: 10.33365/jtsi.v2i1.609.
- [5] G. Barbaglia, S. Murzilli, and S. Cudini, "Definition of REST web services with JSON schema," *Softw. Pract. Exp.*, vol. 47, no. 6, pp. 907–920, Jun. 2017, doi: 10.1002/spe.2466.
- [6] R. Somya and T. M. E. Nathanael, "PENGEMBANGAN SISTEM INFORMASI PELATIHAN BERBASIS WEB MENGGUNAKAN TEKNOLOGI WEB SERVICE DAN FRAMEWORK LARAVEL," *J. Techno Nusa Mandiri*, vol. 16, no. 1, Art. no. 1, Mar. 2019, doi: 10.33480/techno.v16i1.164.
- [7] A. D. Djoelianto, I. A. Kautsar, and M. A. Rosid, "Development of Web Service and Telegram Bot for Location-Based Health Service Information System:," *Procedia Eng. Life Sci.*, vol. 2, no. 2, Art. no. 2, Jun. 2022, doi: 10.21070/pels.v2i2.1280.
- [8] A. P. Septiani, W. Junardi, A. Amaliah, A. Bachtiar, J. I. Mahendra, and Muhammad Irfan Muttaqin, "Sistem Informasi Desa Siaga Pangan Menghadapi Covid19 berbasis Web Service," *J. Sist. Cerdas*, vol. 3, no. 3, Art. no. 3, Dec. 2020, doi: 10.37396/jsc.v3i3.63.

- [9] A. D. Prasetyo, I. A. Kautsar, and N. L. Azizah, "Rancang Bangun Aplikasi Pelaporan Fasilitas Umum Berbasis Web Service Dalam Rangka Menuju Sidoarjo Smart City Dan Open Data," *JIPJ J. Ilm. Penelit. Dan Pembelajaran Inform.*, vol. 7, no. 4, Art. no. 4, Nov. 2022, doi: 10.29100/jipi.v7i4.3259.
- [10] B. Baharuddin, H. Wakkang, and B. Irianto, "IMPLEMENTASI WEB SERVICE DENGAN METODE REST API UNTUK INTEGRASI DATA COVID 19 DI SULAWESI SELATAN," *J. Sintaks Log.*, vol. 2, no. 1, pp. 236–241, Feb. 2022, doi: 10.31850/jsilog.v2i1.1035.
- [11] R. F. Lesmana and J. A. Razaq, "SISTEM PENELITIAN DAN PENGABDIAN MASYARAKAT DENGAN INTEGRASI DATA AKADEMIK MENGGUNAKAN REST API," *J. Manaj. Inform. Dan Sist. Inf.*, vol. 6, no. 1, pp. 1–13, Jan. 2023, doi: 10.36595/misi.v6i1.677.
- [12] I. Setiawan, R. Aditia, and M. R. R. Hidayat, "Penerapan Integrasi Data Pegawai Negeri Sipil di Pemerintah Kota Padang," *J. Teknol. Dan Komun. Pemerintah.*, vol. 5, no. 1, Art. no. 1, Jun. 2023, doi: 10.33701/jtkp.v5i1.3416.
- [13] A. Akhtar, B. Bakhtawar, and S. Akhtar, "EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS," *Int. J. Technol. Innov. Manag. IJTIM*, vol. 2, no. 2, Art. no. 2, Oct. 2022, doi: 10.54489/ijtim.v2i2.77.
- [14] N. A. Septiani and F. Y. Habibie, "Penggunaan Metode Extreme Programming Pada Perancangan Sistem Informasi Pelayanan Publik," *J. Sist. Komput. Dan Inform. JSON*, vol. 3, no. 3, p. 341, Mar. 2022, doi: 10.30865/json.v3i3.3931.
- [15] A. Supriyatna, "METODE EXTREME PROGRAMMING PADA PEMBANGUNAN WEB APLIKASI SELEKSI PESERTA PELATIHAN KERJA," *J. Tek. Inform.*, vol. 11, no. 1, pp. 1–18, May 2018, doi: 10.15408/jti.v11i1.6628.
- [16] A. Del Sole, *Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux*. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-6901-5.
- [17] D. Westerveld, *API Testing and Development with Postman: A practical guide to creating, testing, and managing APIs for automated software testing*. Packt Publishing Ltd, 2021.
- [18] M. F. Nugroho, A. Primajaya, and M. Jajuli, "RANCANG BANGUN REST API APLIKASI MANAJEMEN TOKO MENGGUNAKAN NODEJS PADA CANTIKA PAINT," vol. 7, no. 6, 2023.
- [19] Frensia Tanaga Anaclaudia, Dian Pramana, and I Made Arya Budhi Saputra, "Reactjs and Expressjs Implementation In PMK ITB STIKOM Bali Activity Management," *Aptisi Trans. Technopreneurship ATT*, vol. 5, no. 3, pp. 1–11, Mar. 2023, doi: 10.34306/att.v5i3.313.
- [20] S. Bradshaw, E. Brazil, and K. Chodorow, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. O'Reilly Media, Inc., 2019.
- [21] P. Japikse, K. Grossnicklaus, and B. Dewey, *Building Web Applications with .NET Core 2.1 and JavaScript: Leveraging Modern JavaScript Frameworks*. Berkeley, CA: Apress, 2020. doi: 10.1007/978-1-4842-5352-6.

**Conflict of Interest Statement:**

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.