

PENERAPAN WEB SERVICE UNTUK SISTEM MANAJEMEN PERPUSTAKAAN PADA PERPUSTAKAAN DAERAH KABUPATEN KUDUS DENGAN METODE RESTFUL

Rizq Alwan Fauzan ^{a,1,*}, Mukhamad Nurkamid ^{b,2}, Rizkysari Meimaharani ^{c,3}

^a Universitas Muria Kudus, Jl. Lkr. Utara, Kayuapu Kulon, Gondangmanis, Kec. Bae, Kabupaten Kudus, Jawa Tengah

^b Universitas Muria Kudus, Jl. Lkr. Utara, Kayuapu Kulon, Gondangmanis, Kec. Bae, Kabupaten Kudus, Jawa Tengah

^c Universitas Muria Kudus, Jl. Lkr. Utara, Kayuapu Kulon, Gondangmanis, Kec. Bae, Kabupaten Kudus, Jawa Tengah

¹ fauzanalwan2001@gmail.com; ² muhammad.nurkamid@umk.ac.id; ³ rizky.sari@umk.ac.id

ARTICLE INFO

Keywords

Web Service
Application Programming
Interface (API)
Independent
Library
Management System
Waterfall Method

ABSTRACT

A web service or also known as a web Application Programming Interface (API) is a service that opens two or more independent systems that can communicate with each other, a Web service is created so that the library management system in the archives and library services of Kudus Regency can interact with other systems even though there are differences in the format, operating system, and programming language used between the client and server. The waterfall method is the method used by researchers in developing this system. This method is carried out with a systematic approach, the stages in this method are system requirements analysis, system design, program code generation, testing, and maintenance. Thus, this system is expected to be able to become an intermediary in exchanging data between one system and another, so that it can provide convenience for system managers to connect new systems with existing systems in the archives and library services of Kudus Regency without changing the existing system. been up and running before.

1. Pendahuluan

Web service dapat diartikan sebagai sistem perangkat lunak yang bertugas sebagai pendukung interaksi dari suatu sistem menuju sistem yang lainnya [1]. Dalam penelitian ini perpustakaan merupakan suatu pusat manajemen, sumber ilmu pengetahuan, penelitian, rekreasi, dan pelestarian khasanah ilmu pengetahuan [2]. Munculnya berbagai jenis aplikasi yang dapat membantu meningkatkan suatu pekerjaan baik yang berbasis desktop, web, ataupun aplikasi *mobile* pada *platform* android merupakan bukti berkembangnya suatu ilmu pengetahuan dan teknologi [3]. Aplikasi web adalah salah satu media yang dapat digunakan untuk mengakses internet [4].

Pengembangan *web service* merupakan salah satu dukungan dari suatu teknologi web pada komputasi terdistribusi. *Web service* merupakan suatu sistem untuk pertukaran suatu data antara sistem lain yang terlibat di dalamnya tanpa memperlumahkan perbedaan suatu platform yang digunakan pada masing-masing sistem [5]. Ada beberapa status, antara lain yaitu kode 404 yang artinya data yang diminta tidak ditemukan, kode 401 yang artinya permintaan yang tidak diotorisasi, kode 200 yang artinya semuanya berjalan (OK), dan kode 500 yang artinya ada kesalahan yang terdapat pada sistem yang tidak bisa dipulihkan oleh *server* [6].

Belum adanya sistem *web service* pada dinas kearsipan dan perpustakaan Kabupaten Kudus menjadikan permasalahan dalam mengomunikasikan antara sistem manajemen perpustakaan dengan sistem-sistem baru yang ada di dinas kearsipan dan perpustakaan Kabupaten Kudus. Seperti halnya ketika ada sistem baru yang bertugas untuk menginputkan beberapa buku baru dalam jumlah banyak

yang melibatkan beberapa orang dalam penginputannya agar mempercepat dalam proses input datanya atau sistem baru yang berbeda sistem operasi dan bahasa pemrogramannya, sehingga memerlukan akses ke dalam *database server* seperti pada sistem manajemen perpustakaan, maka dibuatlah suatu sistem *web service* sebagai perantara antara sistem dalam berkomunikasi dan bertukar data, sehingga dalam satu *database* dapat digunakan oleh beberapa sistem baru tanpa merubah sistem yang sudah berjalan sebelumnya. Dengan adanya *web service* di dinas kearsipan dan perpustakaan Kabupaten kudas akan mempermudah pengelola untuk menghubungkan sistem-sistem baru yang ada, meskipun ada beberapa perbedaan dalam format, sistem operasi, dan bahasa pemrogramannya. Sistem *web service* yang akan dibuat pada dinas kearsipan dan perpustakaan Kabupaten Kudus ini menggunakan metode *RESTful* dikarenakan metode ini lebih mudah dipelajari, lebih efisien dan cepat prosesnya, lebih simpel dalam pengembangannya, dan desain lebih dekat dengan web. API (Application Programming Interface) berbasis web merupakan sebuah tools yang membantu dalam penggunaan teknologi *REST* [7].

Web service akan diintegrasikan dengan sistem manajemen perpustakaan pada dinas kearsipan dan perpustakaan kabupaten kudas. dengan manajemen perpustakaan diharapkan bisa meningkatkan minat pengguna terhadap perpustakaan sebagai pemustakanya [8]. Bahasa pemrograman yang digunakan dalam perancangan sistem ini dengan menggunakan PHP, *MySQL* dan didukung dengan *framework codeigniter 4*. Seluruh *script* PHP dijalankan pada *server* tempat *script* itu dijalankan [9]. *MySQL* tidak seperti Apache, dikarenakan Apache merupakan software yang berkembang dari suatu komunitas umum, yang masing-masing penulisnya memiliki hak ciptanya sendiri [10]. *Codeigniter 4* memiliki keunggulan berjalan dengan cepat serta ringan dikarenakan resource-nya tidak begitu banyak dan didukung untuk penggunaan PHP versi 5 [11].

Tujuan dari penelitian ini adalah membuat dan mengimplementasikan *web service* dengan *RESTful* untuk sistem manajemen perpustakaan yang meliputi pendataan anggota, pendataan buku, peminjaman buku, pengembalian buku, denda keterlambatan, dan pelaporan pada dinas kearsipan dan perpustakaan Kabupaten Kudus, dengan menggunakan bahasa pemrograman PHP, basis data *MySQL*, dan *framework codeigniter 4*. Pada penelitian ini memiliki perbedaan dengan penelitian sebelumnya dalam fitur yang diberikan, di penelitian ini lebih menuju pada manajemen perpustakaan mulai dari proses input buku, proses peminjaman, proses pengembalian, dan pelaporan. Dipenelitian ini menggunakan *RESTful* dalam *Web Service* dengan alasan *RESTful* menggunakan protokol HTTP yang sudah tersedia secara luas, sehingga memungkinkan skalabilitas yang baik. Itu berarti dapat menangani beban yang tinggi dan mudah untuk ditingkatkan ke dalam sistem yang lebih besar jika dibutuhkan.

2. Metodologi Penelitian

Metode pengembangan sistem yang dilakukan pada penelitian ini yaitu dengan menggunakan metode *waterfall*. Karena sifatnya yang natural maka metode *waterfall* ini merupakan metode pengembangan perangkat lunak yang paling tua. Metode *waterfall* merupakan metode klasik yang sifatnya sistematis dan pembangunan sistemnya berurutan. Nama asli dari metode ini yaitu *linear Sequential model* atau sering juga disebut dengan *classic life cycle*. Bisa disebut sebagai *waterfall* dikarenakan tahap yang dilalui atau alur dari metode ini harus dilakukan secara bertahap, setiap tahap yang dilakukan harus diselesaikan terlebih dahulu tidak boleh loncat ketahap selanjutnya [12]. Metode *waterfall* ini merupakan metode milik Winston W. Royce. Ada beberapa tahapan dalam menggunakan metode *waterfall*, antara lain yaitu:

2.1. Requirement Analysis

Tahap *requirement analysis* merupakan langkah dalam menganalisa kebutuhan dalam sebuah sistem. Dalam tahap ini ada beberapa cara yang dilakukan untuk mengumpulkan data, yaitu observasi ke tempat penelitian, wawancara kepada pihak-pihak yang ada di dinas kearsipan dan perpustakaan

Kabupaten Kudus, dan melakukan studi literatur. Peneliti akan menggali informasi mengenai kebutuhan sistem melalui pihak yang ada di dinas kearsipan dan perpustakaan Kabupaten Kudus, sehingga peneliti dapat menciptakan sebuah sistem yang dapat menjalankan tugas-tugas yang diperintahkan oleh *user* dengan baik. Pada tahap ini akan menghasilkan sebuah data kebutuhan sistem yang berhubungan dengan kebutuhan yang diminta dari pihak dinas kearsipan dan perpustakaan Kabupaten Kudus sebagai acuan dalam pembuatan sistem.

2.2. *Design*

Tahap *design* merupakan proses selanjutnya setelah melakukan analisa kebutuhan sistem. Pada tahap ini peneliti akan mengimplementasikan kebutuhan sistem dalam bentuk *design*. Pembuatan *design* bertujuan untuk membantu memberikan suatu gambaran yang lengkap mengenai sistem yang akan dibuat. Tahap ini bisa membantu seorang pengembang dalam menyiapkan suatu kebutuhan perangkat keras dalam perancangan arsitektur sebuah sistem perangkat lunak yang nantinya dibuat keseluruhannya. *Design* juga berguna untuk mempresentasikan gambaran sistem yang akan dibuat kepada pihak dinas kearsipan dan perpustakaan Kabupaten Kudus.

2.3. *Implementation*

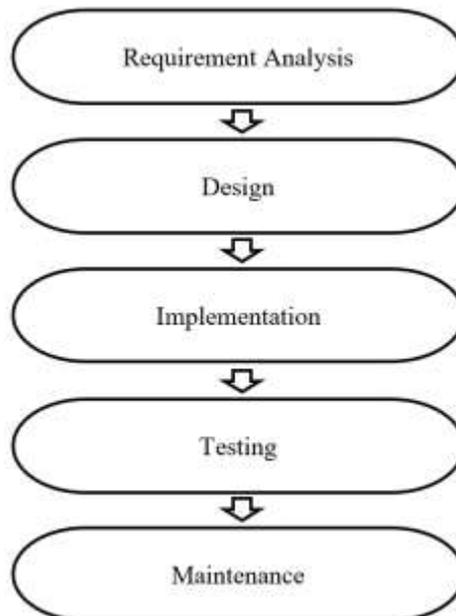
Tahap *implementation* merupakan tahap pembuatan program sesuai dengan kebutuhan sistem dan desain pada tahap sebelumnya. Pembuatan sistem ini dibagi menjadi beberapa modul kecil yang akan disatukan pada tahap selanjutnya. Di tahapan ini juga dilaksanakan pengujian dan pemeriksaan pada fungsionalitas modul yang telah dibuat, tujuannya yaitu untuk mengetahui kriteria sistem sudah memenuhi kebutuhan sistemnya atau belum, jika belum memenuhi kriteria maka dilakukanlah perbaikan atau penambahan beberapa fitur lagi, agar sesuai dengan kebutuhan pada sistem manajemen perpustakaan Kabupaten Kudus.

2.4. *Testing*

Tahap *testing* merupakan tahap pemeriksaan serta pengujian sistem secara menyeluruh agar sistem dapat diidentifikasi kemungkinan adanya kesalahan sistem dan kegagalan sistem. Sebelum melakukan tahap ini seluruh modul yang telah dikembangkan dan diuji tiap-tiap modulnya kemudian diintegrasikan dalam sebuah sistem secara menyeluruh, maka terbentuklah sistem manajemen perpustakaan Kabupaten Kudus yang siap dilakukan uji coba sistemnya.

2.5. *Maintenance*

Tahap terakhir yaitu tahap *maintenance*, tahap *maintenance* merupakan tahap pemeliharaan sistem yang sudah jadi secara berkala. Pada tahap ini pengembang dapat melakukan perbaikan kepada sistem yang masih terdapat kesalahan yang belum terdeteksi pada tahap-tahap yang sebelumnya. Dalam pemeliharaan sistem ada beberapa yang dapat dilakukan oleh pengembang, antara lain yaitu perbaikan kesalahan, perbaikan implementasi modul sistem, dan peningkatan sistem sesuai dengan kebutuhan dari sistem.



Gambar 1. Tahapan Metode *Waterfall*

3. Hasil dan Pembahasan

Hasil dari penelitian ini berupa suatu sistem *web service* yang memiliki fungsi sebagai perantara antara *client* dan *server*. Tidak adanya konektivitas antara sistem dan sulitnya mengembangkan suatu integrasi data merupakan masalah yang ada. Sehingga dalam prosesnya terjadi banyak kesulitan dalam menyinkronasikan suatu data dan banyaknya data yang duplikat. seiring perkembangannya teknologi maka, akan muncul aplikasi-aplikasi atau sistem baru yang akan dibuat. Maka dari itu, dibutuhkanlah *web service* berbasis *REST* agar aplikasi-aplikasi yang sudah berjalan sebelumnya masih dapat bisa diakses dan saling berkomunikasi tanpa merubah struktur programnya [13]. *REST* atau yang sering disebut dengan *RESTful* yaitu suatu arsitektur yang berada didalam *API* [14]. Dalam hal ini *web service* memiliki kedudukan sebagai *server*, sedangkan yang berkedudukan sebagai *client* yaitu sistem manajemen perpustakaan. Kedua sistem ini saling berdiri sendiri dan tidak dalam satu kesatuan.

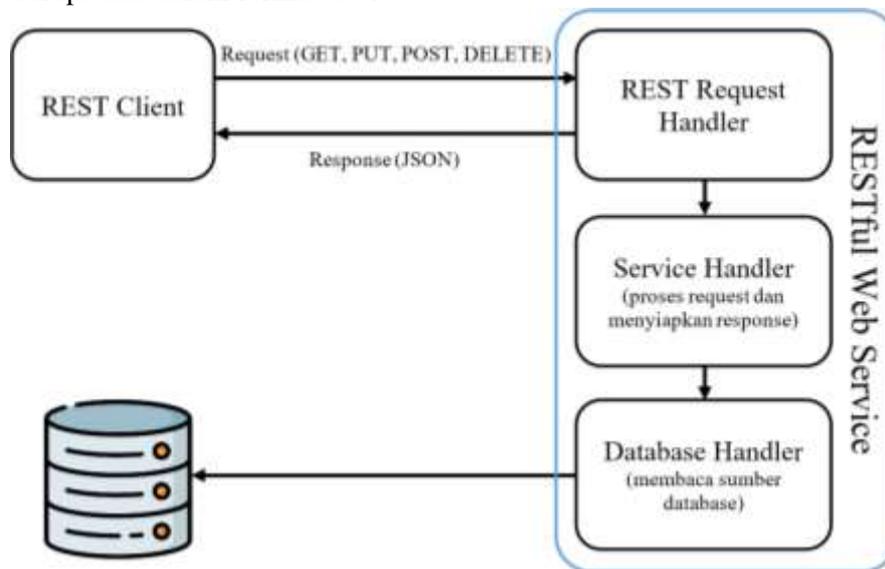
Pada perpustakaan daerah Kabupaten Kudus tidak hanya memiliki sistem manajemen perpustakaan saja, melainkan ada beberapa sistem lainnya yang memiliki fungsi dan tugasnya masing-masing. Adanya berbagai sistem pada perpustakaan daerah Kabupaten Kudus, maka diperlukanlah suatu integrasi antara satu sistem dengan sistem yang lainnya dengan tujuan untuk menghindari suatu duplikasi data pada *database* yang ada di setiap sistem yang berbeda serta meningkatkan *fleksibilitas* pada masing-masing sistemnya [15]. Hasil dari penelitian ini dibagi menjadi tiga tahap, yaitu tahap gambaran perancangan sistem, selanjutnya tahap perancangan sistem, dan yang terakhir tahap implementasi sistem.

3.1. Gambaran Perancangan Sistem

Dalam perancangan sistem ini *web service* bertugas untuk melakukan suatu pertukaran data dengan sistem atau aplikasi yang saling berdiri sendiri, dikarenakan beberapa sistem memiliki format dan bahasa pemrograman yang berbeda atau juga berjalan di *platform* yang berbeda. Pada perancangan *web service* ini arsitektur yang digunakan berbasis *REST*, juga sering kali disebut dengan *RESTful web service*. Untuk penerapannya *HTTP* merupakan metode dalam konsep arsitektur *REST*.

Di dalam struktur *REST*, komponen *REST server* merupakan penyedia sumber data sedangkan *REST client* merupakan pengakses data yang diberikan dari *REST server*, kemudian sumber data tersebut akan

ditampilkan kepada pengguna selanjutnya. Sumber data dari *REST server* ini direpresentasikan ke dalam bentuk teks berformat *JSON*. Pada sistem *web service* ini sebuah *client* akan mengirimkan suatu data atau sering disebut request dengan melalui *HTTP Request*, selanjutnya *server* akan merespon *request* yang dikirimkan oleh *client* dengan melalui *HTTP Response*, kemudian *response* akan dikirim kepada *client* berupa data dalam format *JSON*.



Gambar 2 Gambaran Perancangan Sistem

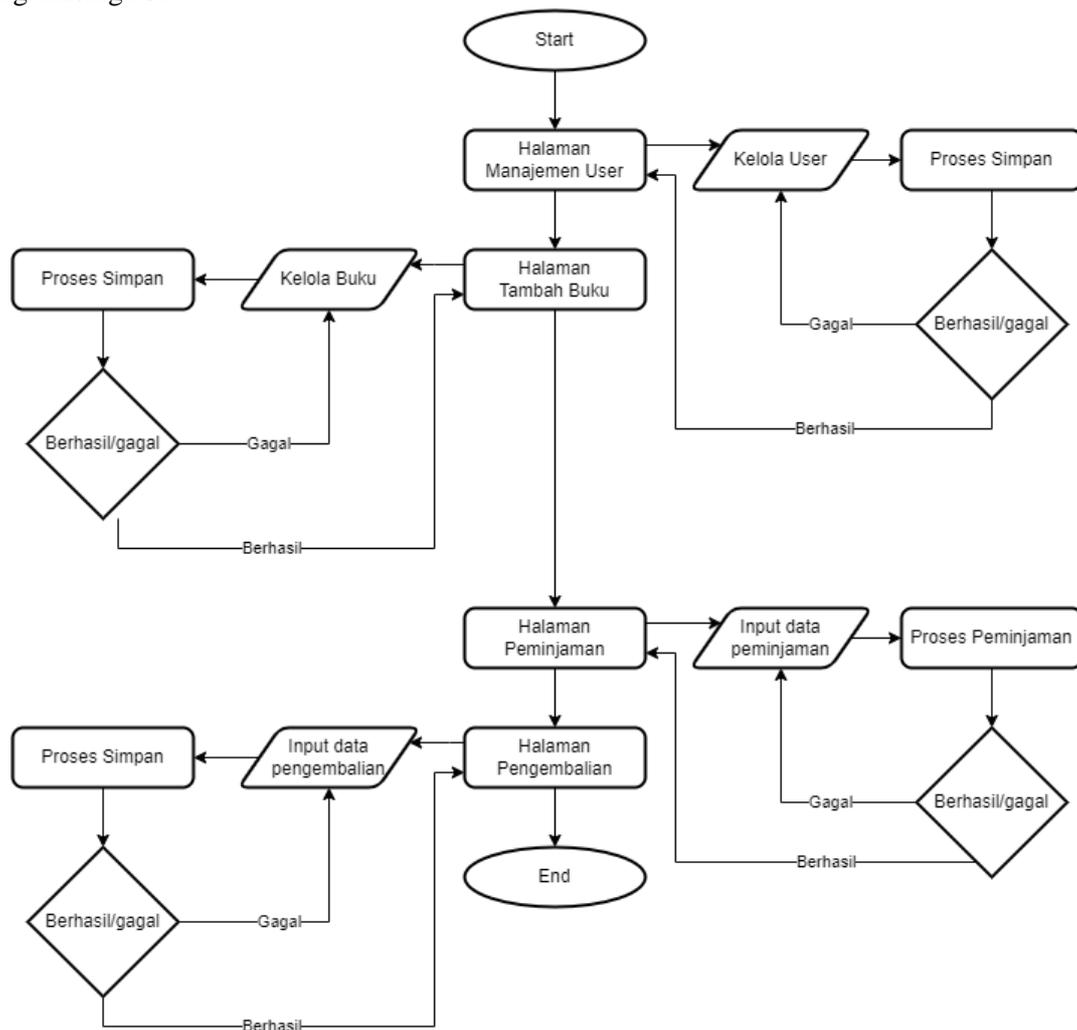
Web service pada sistem ini berfungsi untuk mengakses database yang ada pada sistem manajemen perpustakaan pada dinas kearsipan dan perpustakaan Kabupaten Kudus. Sistem manajemen perpustakaan berkedudukan sebagai *REST client* sedangkan *web service* berkedudukan sebagai *REST Server*. Setiap permintaan data yang diminta dari sistem manajemen perpustakaan akan dikirim kepada *web service*, kemudian *web service* akan memberikan *response* berupa data dan diteruskan kepada sistem manajemen perpustakaan.

3.2. Perancangan Sistem

Perancangan sistem merupakan tahapan dimana peneliti membuat rancangan dasar sebuah sistem yang akan dibuat. Pada tahap ini pertama-tama peneliti membuat diagram alur sistem yang akan dibuat dengan menggunakan *flowchart*, kemudian memasuki tahapan selanjutnya, yaitu pembuatan rancangan *database* sebagai pusat penyimpanan data pada sistem, *flowchart* merupakan diagram perancangan sistem yang umum digunakan oleh beberapa perancang sistem, dengan adanya diagram ini diharapkan mampu mempermudah peneliti dalam perancangan sistem ini, serta mempermudah pembaca untuk memahami alur dari sistem ini.

Pada bagian menu manajemen *user* terdapat beberapa fitur kelola *user*, diantaranya yaitu tambah *user* yang berfungsi untuk mendaftarkan *user* baru, kemudian ada daftar *user* yang dikelompokkan sesuai hak akses *user*, pada fitur daftar *user*, terdapat fitur untuk mengelola *user*, mereset *password*, dan menghapus *user*. Pada bagian menu tambah buku terdapat beberapa fitur kelola buku, antara lain yaitu tambah buku baru yang berfungsi untuk mendaftarkan buku baru agar dapat disimpan pada basis data *server*, selanjutnya ada *import* buku yang berfungsi untuk menambahkan data buku dalam jumlah banyak yang datanya sudah disimpan pada file excel dengan format yang sudah disediakan. Pada bagian menu peminjaman terdapat beberapa fitur, diantaranya yaitu tambah peminjaman yang di dalamnya terdapat *form* untuk mengisikan data peminjaman baru, kemudian ada daftar peminjaman yang di

dalamnya menampilkan data peminjaman dari berbagai *user* dalam bentuk tabel, pada fitur ini memiliki fitur untuk melihat detail peminjaman masing-masing *user*, memperpanjang masa peminjaman masing-masing *user*, dan menghapus data peminjaman masing-masing *user*. Pada bagian menu pengembalian terdapat beberapa fitur, diantaranya yaitu pengembalian buku yang di dalamnya terdapat *form* untuk mengisikan data pengembalian buku, kemudian ada daftar pengembalian buku yang di dalamnya menampilkan data pengembalian buku dari berbagai *user* dalam bentuk tabel, pada fitur ini memiliki fitur untuk melihat detail pengembalian masing-masing *user* dan menghapus data pengembalian masing-masing *user*.



Gambar 3. Flowchart Sistem

Pada sistem manajemen perpustakaan yang terintegrasi dengan *web service* ini membutuhkan suatu *database* untuk menyimpan data-data yang ada pada sistem, database yang digunakan pada sistem ini yaitu dengan menggunakan database *MySQL*. Terdapat sembilan tabel, antara lain yaitu tabel migrations, tabel jwt, tabel user, tabel kategori, tabel rak, tabel buku, tabel transaksi, dan tabel denda. Pada database *server* ini juga memiliki relasi antar tabel yang saling menghubungkan antara tabel yang satu dengan tabel yang lainnya.

Tabel *migrations* merupakan tabel bawaan dari *framework codeigniter 4*, tabel ini otomatis dibuat saat proses migrasi dengan menggunakan terminal pada *framework codeigniter 4*. Tabel ini akan

menyimpan data migrasi setiap melakukan migrasi pada *database* dengan *codeigniter 4*. Tabel *jwt* merupakan tabel yang berfungsi untuk menyimpan data autentikasi pada *server* dengan *id_jwt* sebagai primarynya, tabel ini digunakan untuk mengamankan *server* agar tidak dapat diakses oleh sembarang orang. Tabel *user* merupakan tabel yang memiliki fungsi untuk menyimpan data diri *user* yang terdaftar dengan *id_user* sebagai primarynya. Tabel kategori merupakan tabel yang berfungsi untuk menyimpan data kategori buku dengan *id_kategori* sebagai primarynya. Tabel rak merupakan tabel yang berfungsi untuk menyimpan data rak buku dengan *id_rak* sebagai primarynya. Tabel buku merupakan tabel yang berfungsi untuk menyimpan data buku dengan *id_buku* sebagai primarynya. Tabel transaksi merupakan tabel yang berfungsi untuk menyimpan data transaksi berupa peminjaman buku, pengembalian buku, dan keterlambatan peminjaman dengan *id_transaksi* sebagai primarynya. Tabel denda merupakan tabel yang berfungsi untuk menyimpan data denda dengan *id_denda* sebagai primarynya.

Ada beberapa relasi antar tabel yang terdapat pada *database server*, relasi ini berfungsi untuk menghubungkan antara suatu tabel dengan tabel yang lainnya. Pada tabel buku terdapat relasi antar tabel kategori dengan *id_kategori* sebagai kunci tamunya dan tabel rak dengan *id_rak* sebagai kunci tamunya, pada tabel transaksi ini memiliki relasi antar tabel *user* dengan *id_user* sebagai kunci tamunya, dan tabel buku dengan *id_buku* sebagai kunci tamunya.



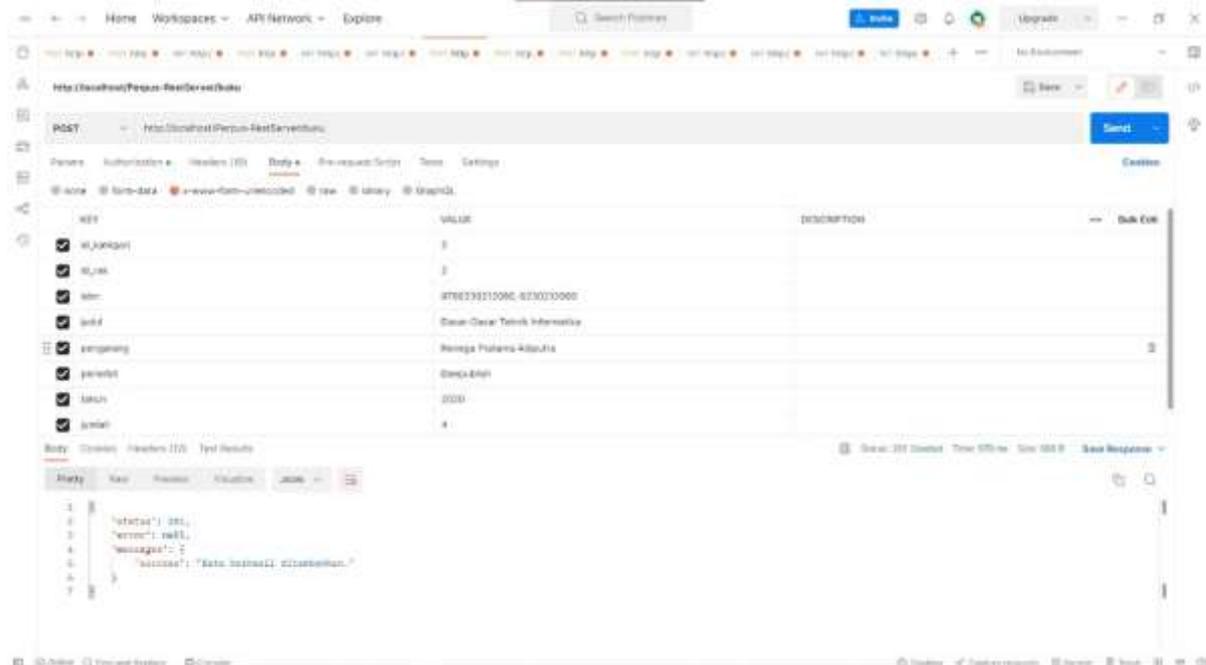
Gambar 4 Rancangan Database

Tahapan selanjutnya pada perancangan sistem ini merupakan tahapan uji coba *request* ke *server REST API*. *REST API* merupakan antarmuka yang digunakan oleh beberapa sistem dengan tujuan untuk saling bertukar informasi secara aman, *REST API* pada sistem ini diterapkan pada *web service* dan sistem manajemen perpustakaan, dalam pembuatannya peneliti menggunakan metode *RESTful*, ada beberapa layanan yang diterapkan pada *RESTful* ini untuk mengomunikasikan antara *web service* dengan sistem manajemen perpustakaan. Pada tahap ini akan dilakukan proses pengujian *web service* dengan *RESTful* sebelum diimplementasikan ke dalam sistem manajemen perpustakaan, alat yang dapat digunakan yaitu *postman*. Pada tahap ini akan dilakukan pengujian dengan *method POST, GET, PUT,*

dan *DELETE* pada masing-masing tabel yang ada pada *database server*. *POST* berfungsi untuk mengirimkan data, *GET* berfungsi untuk mengambil data, *PUT* berfungsi untuk mengubah data, dan yang terakhir *DELETE* berfungsi untuk menghapus data.

3.3. Implementasi Sistem

REST API pada sistem ini diterapkan pada web service dan sistem manajemen perpustakaan, dalam pembuatannya peneliti menggunakan metode RESTful, ada beberapa layanan yang diterapkan pada RESTful ini untuk mengomunikasikan antara web service dengan sistem manajemen perpustakaan. Pada tahap ini akan dilakukan proses pengujian web service dengan RESTful sebelum diimplementasikan ke dalam sistem manajemen perpustakaan, alat yang dapat digunakan yaitu postman. Sebagai contohnya untuk melakukan *request* pada tabel buku yang ada di *database* bisa mengakses pada *API* dengan memasukkan *URL* `http://localhost/Perpus-RestServer/buku` pada *postman*. Ada beberapa method yang digunakan untuk melakukan request, antara lain yaitu method *POST*, *GET*, *PUT*, dan *DELETE*. *POST* berfungsi untuk mengirimkan data, *GET* berfungsi untuk mengambil data, *PUT* berfungsi untuk mengubah data, dan yang terakhir *DELETE* berfungsi untuk menghapus data.



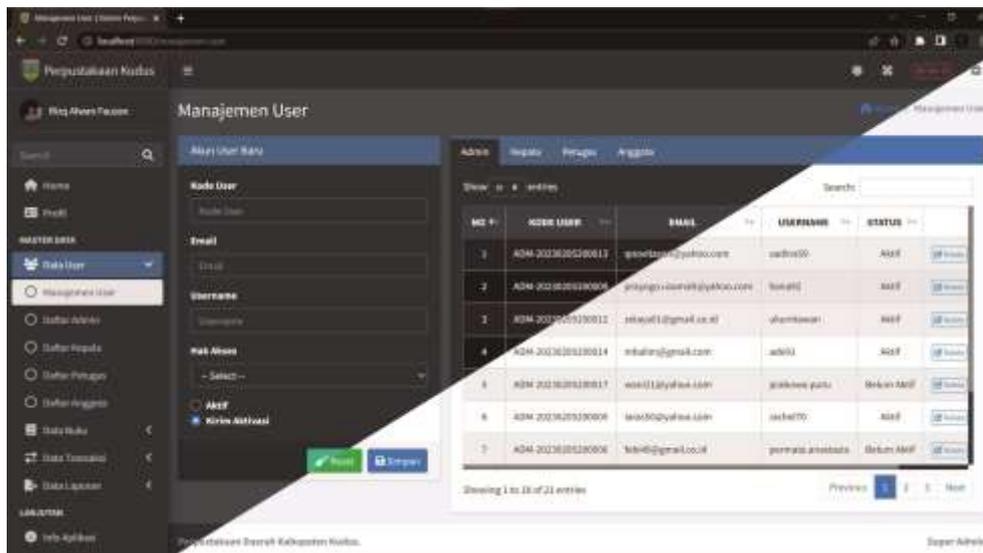
Gambar 5. Tampilan *Postman*

Seperti yang sudah dijelaskan sebelumnya, sistem manajemen perpustakaan ini terintegrasi dengan *web service*, dimana *database* yang berfungsi untuk menyimpan data terpisah dari sistem manajemen perpustakaan, melainkan berada pada *server web service*. Untuk melakukan pertukaran suatu data antara *web service* dan sistem manajemen perpustakaan dapat dilakukan melalui *HTTP request*, *HTTP request* adalah mekanisme pengiriman suatu data pada protokol *HTTP* yang bertugas untuk bertukar data antara *client* dan *server*. Berikut merupakan tampilan *dashboard* dari sistem manajemen perpustakaan yang sudah diintegrasikan dengan *web service*. Semua data diambil dari *server web service* yang kemudian ditampilkan pada sistem manajemen perpustakaan yang berkedudukan sebagai *client*.



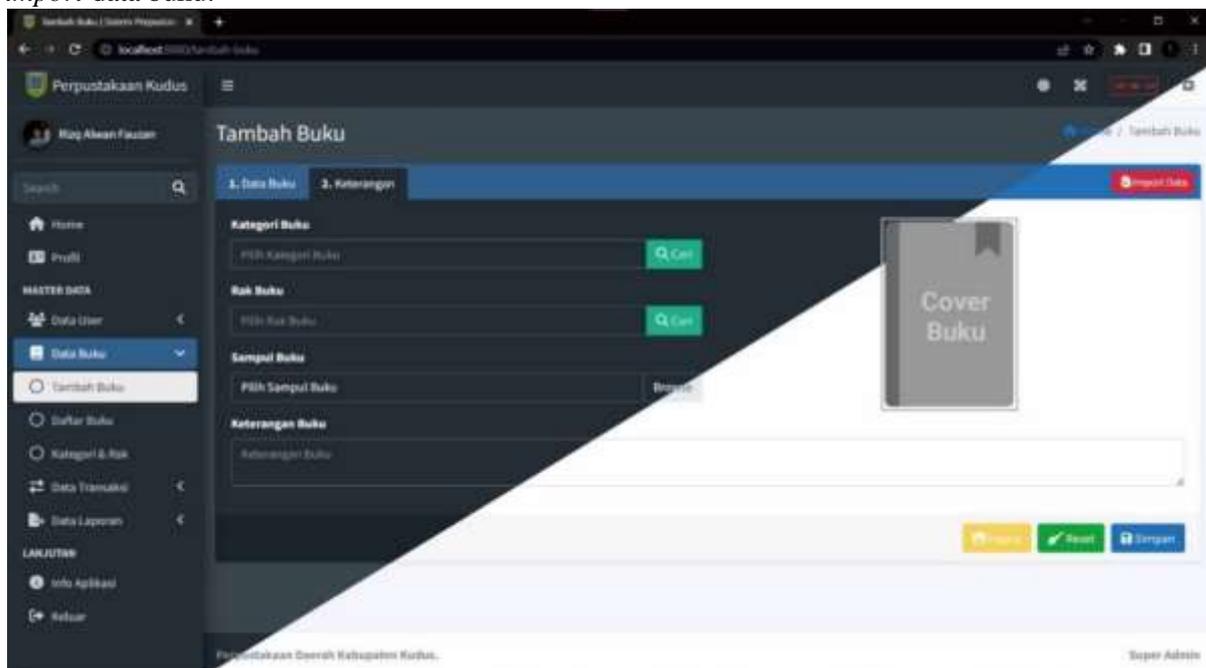
Gambar 6. Tampilan Dashboard

Menu manajemen *user* merupakan menu yang berfungsi untuk mengelola akun *user* yang ada pada sistem manajemen perpustakaan, pada menu manajemen *user* ini terdapat beberapa fitur, diantaranya yaitu tambah akun *user* baru, daftar *user*, kelola *user*, *reset password user*, dan hapus *user*. Fitur yang pertama dari menu manajemen *user* yaitu fitur tambah akun *user* baru, fitur ini berupa *content* yang berisi *form* untuk mendaftarkan *user* baru. Fitur yang kedua dari menu manajemen *user* yaitu fitur daftar *user*. Daftar *user* ini dibagi menjadi empat, yaitu *user* admin, *user* kepala, *user* petugas, dan *user* anggota, fitur ini berupa *content* yang masing-masing berisi tabel yang di dalamnya merupakan daftar dari *user* dengan hak akses tertentu yang sudah terdaftar pada sistem manajemen perpustakaan. Fitur yang ketiga dari menu manajemen *user* yaitu fitur kelola *user*, setelah menekan tombol kelola pada tabel daftar *user*, akan muncul *pop up* berupa *form* untuk mengelola data akun dari suatu *user*. Fitur yang keempat dari menu manajemen *user* yaitu fitur *reset password user*, setelah menekan tombol *reset* pada tabel daftar *user*, akan muncul *pop up* berupa konfirmasi untuk mereset *password* dari suatu *user*. Fitur yang terakhir dari menu manajemen *user* yaitu fitur hapus *user*, setelah menekan tombol hapus pada tabel daftar *user*, akan muncul *pop up* berupa konfirmasi untuk menghapus suatu *user* yang terdaftar.



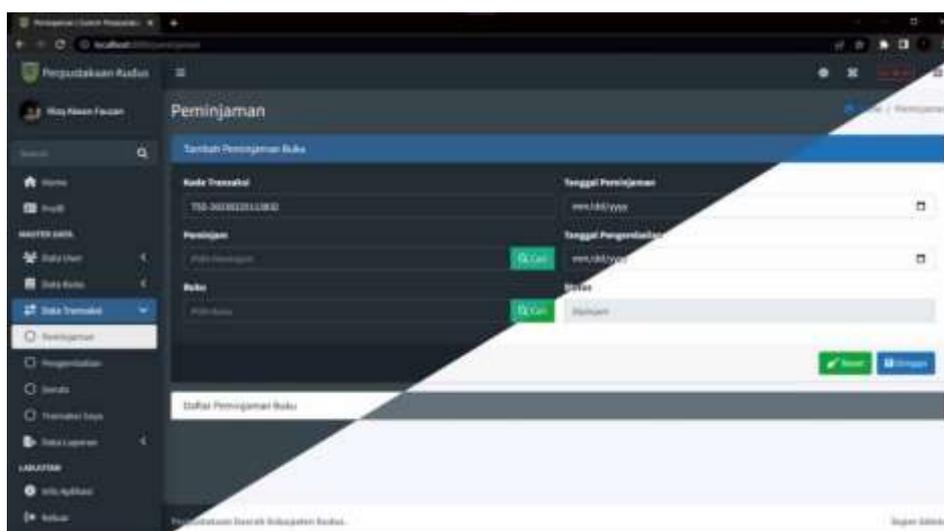
Gambar 7. Tampilan Menu Manajemen User

Menu tambah buku merupakan menu yang berfungsi untuk menambahkan data buku baru pada sistem manajemen perpustakaan, pada menu tambah buku ini terdapat dua fitur, yaitu tambah buku satu per satu dan *import* buku. Fitur yang pertama dari menu tambah buku yaitu fitur tambah buku satu per satu, fitur ini berupa *content* yang berisi *form* untuk menginputkan data dari suatu buku, ada dua langkah dalam menambahkan buku langkah pertama yaitu menginputkan data buku, langkah kedua yaitu menginputkan keterangan buku. Fitur yang kedua dari menu tambah buku yaitu fitur *import* buku, setelah menekan tombol *import* buku akan muncul sebuah *pop up* yang isinya berupa *form* untuk memilih file buku yang akan diimport, file ini harus memiliki format excel agar dapat melakukan proses *import* data buku.



Gambar 8. Tampilan Menu Tambah Buku

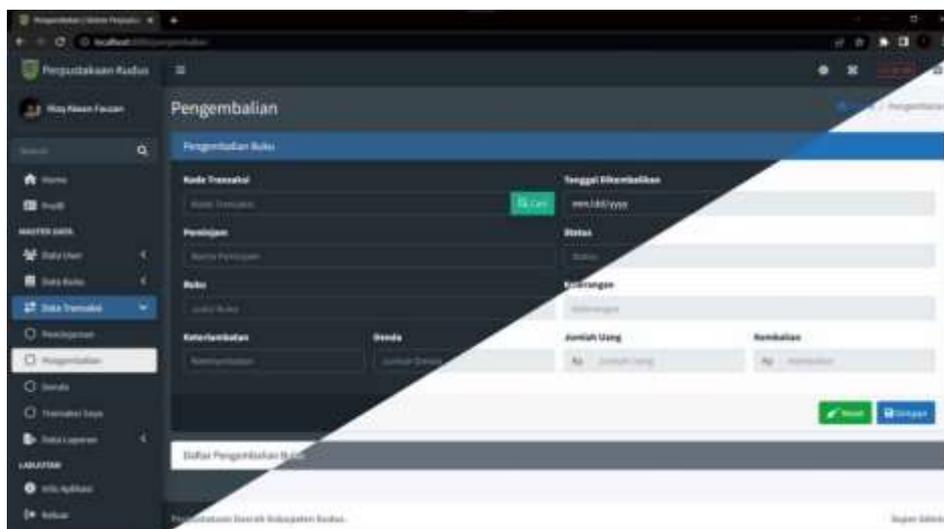
Menu peminjaman merupakan menu yang berfungsi untuk melakukan proses transaksi berupa peminjaman buku, serta memberikan informasi mengenai data peminjaman yang dilakukan oleh anggota yang terdaftar pada sistem manajemen perpustakaan, pada menu peminjaman ini terdapat beberapa fitur, diantaranya yaitu tambah peminjaman buku, informasi data peminjaman buku, detail peminjaman buku, perpanjang peminjaman buku, dan hapus peminjaman buku. Fitur yang pertama dari menu peminjaman yaitu fitur tambah peminjaman buku, fitur ini berupa *content* yang berisi *form* untuk menambahkan peminjaman buku kepada *user* anggota yang terdaftar pada sistem manajemen perpustakaan. Fitur yang kedua dari menu peminjaman yaitu fitur daftar peminjaman buku, fitur ini berupa *content* yang berisi tabel daftar peminjaman buku, di dalamnya berisi mengenai peminjaman-peminjaman buku yang dilakukan oleh suatu *user* anggota. Fitur yang ketiga dari menu peminjaman yaitu fitur detail peminjaman, setelah menekan tombol detail pada tabel daftar peminjaman buku, akan muncul *pop up* berupa informasi mengenai data suatu *user* anggota dan buku yang ada pada proses peminjaman. Fitur yang keempat dari menu peminjaman yaitu fitur perpanjang peminjaman, setelah menekan tombol perpanjang pada tabel daftar peminjaman buku, akan muncul *pop up* berupa *form* untuk memperpanjang suatu peminjaman buku yang dilakukan oleh suatu *user*. Fitur yang terakhir dari menu peminjaman yaitu fitur hapus peminjaman, setelah menekan tombol hapus pada tabel daftar peminjaman buku, akan muncul *pop up* berupa konfirmasi untuk menghapus suatu data peminjaman yang ada pada sistem manajemen perpustakaan.



Gambar 9. Tampilan Menu Peminjaman

Menu pengembalian merupakan menu yang berfungsi untuk melakukan proses transaksi berupa pengembalian buku dan memberikan informasi mengenai data pengembalian yang dilakukan oleh *user* anggota yang terdaftar pada sistem manajemen perpustakaan, pada menu pengembalian ini terdapat beberapa fitur, diantaranya yaitu pengembalian buku, informasi data pengembalian buku, detail pengembalian buku, dan hapus pengembalian buku. Fitur yang pertama dari menu pengembalian yaitu fitur pengembalian buku, fitur ini berupa *content* yang berisi *form* untuk mendata pengembalian buku yang telah dipinjam oleh *user* anggota yang terdaftar pada sistem manajemen perpustakaan. Fitur yang kedua dari menu pengembalian yaitu fitur daftar pengembalian buku, fitur ini berupa *content* yang berisi tabel daftar pengembalian buku, di dalamnya berisi mengenai data pengembalian buku yang dilakukan oleh suatu *user* anggota. Fitur yang ketiga dari menu pengembalian yaitu fitur detail pengembalian, setelah menekan tombol detail pada tabel daftar pengembalian buku, akan muncul *pop up* berupa

informasi mengenai data suatu *user* anggota dan buku yang ada pada proses pengembalian. Fitur yang terakhir dari menu pengembalian yaitu fitur hapus data pengembalian, setelah menekan tombol hapus pada tabel daftar pengembalian buku, akan muncul *pop up* berupa konfirmasi untuk menghapus suatu data pengembalian yang ada pada sistem manajemen perpustakaan.



Gambar 10. Tampilan Menu Pengembalian

4. Kesimpulan

Hasil dari penelitian ini adalah sebuah sistem *web service* dengan menggunakan metode *RESTful* yang berfungsi untuk mengomunikasikan antara *client* dan *server*, sehingga *client* bisa mengambil data atau mengirimkan data kepada *server* yang berfungsi sebagai tempat menyimpan data. sistem *web service* pada penelitian ini diterapkan di dinas kearsipan dan perpustakaan Kabupaten Kudus yang memiliki fungsi untuk mengintegrasikan sistem manajemen perpustakaan agar dapat berinteraksi dengan *database* yang ada di *server*, dalam hal ini *web service* memiliki kedudukan sebagai *server*, sedangkan sistem manajemen perpustakaan berkedudukan sebagai *client*. *Web service* dan sistem manajemen perpustakaan merupakan sistem yang saling berdiri sendiri, dengan kata lain kedua sistem ini tidak berada dalam satu kesatuan. Metode yang digunakan peneliti untuk mengembangkan sistem aplikasi ini adalah metode *waterfall*, dimana metode ini merupakan suatu pendekatan yang sistematis dengan beberapa tahapan tertentu. Tahapan tersebut yakni analisis kebutuhan sistem, desain sistem, pembuatan kode program, pengujian, dan pemeliharaan.

Sistem *web service* dengan metode *RESTful*, secara garis besar merupakan media interaksi antara *server* dengan *client*. Sistem ini mampu memberikan kemudahan bagi pengelola sistem untuk menghubungkan sistem yang satu dengan yang lainnya tanpa melakukan perubahan dari sistem yang sebelumnya sudah berjalan. Sistem ini juga dapat diintegrasikan dengan sistem lain yang memiliki perbedaan dalam format, sistem operasi, dan bahasa pemrogramannya, dikarenakan sistem ini berdiri sendiri atau tidak dalam satu kesatuan dengan sistem lainnya dan untuk pertukaran datanya dilakukan melalui protokol HTTP. Oleh karena itu sistem *web service* cocok untuk diterapkan pada suatu instansi yang memiliki sistem atau aplikasi lebih dari satu sehingga dalam penyimpanan datanya menjadi lebih terstruktur pada satu *database server*.

Daftar Pustaka

- [1] Norkhimah, "Penerapan Webservice Pada Sistem Informasi Perpustakaan," *J. Teknol. Inf.*, vol. 13, no. 3, pp. 17–24, 2018, doi: 10.35842/jtir.v13i3.248.
- [2] Ade Yansyah, S. H. Al-Ikhsan, and I. Yanuarsyah, "Rancang Bangun Web Service Informasi Perpustakaan Di Kota Bogor," *Univ. Ibn Khaldun Bogor*, 2018.
- [3] L. Ramadhani, R. Amalia, and F. Puspita, "Implementasi Firebase Realtime Database Pada Aplikasi Integrated Perpustakaan Smk Prestasi Prima," *Semin. Nas. Ris. dan Teknol. (Semnas Ristek) 2021*, pp. 283–288, 2021.
- [4] E. R. Subhiyakto, Y. P. Astuti, and N. K. Ningrum, "Pengembangan Aplikasi Katalog Perpustakaan Terintegrasi Menggunakan Metode RESTful pada Perpustakaan Kota Semarang dan Perpustakaan Daerah Jawa Tengah," *J. Inform. J. Pengemb. IT*, vol. 03, no. 02, pp. 161–166, 2018.
- [5] Herfandi, M. Julkarnain, and M. Hanif, "Desain dan Implementasi Restful Web Services Untuk Integrasi Data dan Aplikasi," *J. Inform. Teknol. dan Sains*, vol. 4, no. 1, pp. 36–41, 2022, doi: 10.51401/jinteks.v4i1.1529.
- [6] R. Rizal and A. Rahmatulloh, "Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *J. Ilm. Inform.*, vol. 7, no. 01, p. 54, 2019, doi: 10.33884/jif.v7i01.1004.
- [7] I. K. Mohidin, "Penerapan Teknologi Rest Api Pada Aplikasi Perpustakaan Digital Politeknik Gorontalo," *J. Technopreneur*, vol. 10, no. 1, pp. 34–39, 2022, doi: 10.30869/jtech.v10i1.922.
- [8] F. Luthfiyah, "Manajemen Perpustakaan Dalam Meningkatkan Layanan Perpustakaan," *J. el-Idare*, vol. 1, no. 2, pp. 189–202, 2020, [Online]. Available: <http://jurnal.radenfatah.ac.id/index.php/El-idare/article/view/676/608>
- [9] Y. Sugiyani and M. Risdiyani, "Model Pembelajaran Elektronik (E-Learning) Untuk Pkbn Melati Cibeber Di Kota Cilegon," *J. Prosisko*, vol. 4, no. 1, pp. 37–42, 2017.
- [10] A. Rusman and A. N. Desy, "Sistem Informasi Perpustakaan Pada SMK Muhammadiyah 09 Jakarta Selatan," *Simp. Nas. Ilmu Pengetah. dan Teknol.*, pp. 163–167, 2017, [Online]. Available: <http://www.elib.unikom.ac.id>
- [11] S. P. Nabila and H. Amnur, "Rancang Bangun Sistem Informasi Desa Menggunakan Framework Codeigniter 4 Bagian Kelahiran dan Kematian pada Desa Cageur," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 2, no. 2, pp. 56–62, 2021, doi: 10.30630/jitsi.2.2.34.
- [12] M. Mailasari, "Sistem Informasi Perpustakaan Menggunakan Metode Waterfall," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 8, no. 2, pp. 207–214, 2019, doi: 10.32736/sisfokom.v8i2.657.
- [13] A. Firdaus, S. Widodo, A. Sutrisman, S. G. Fadhilah Nasution, and R. Mardiana, "Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Web Service Pada Jurusan Teknik Komputer Polstri," *J. Inform.*, vol. 5, no. 2, pp. 81–87, 2019.
- [14] Febyana Wijaya, A. Jacobus, and A. Sambul, "Implementasi Web Service Pada Sistem Informasi Perpustakaan Perguruan Tinggi," *J. Tek. Inform.*, vol. 16, no. 4, 2021.
- [15] Z. Sa'diyah, K. E. Aziz, M. H. Dardiri, and M. A. Yaqin, "Pengembangan Web Service Sistem Informasi Sekolah," *Jurasik (Jurnal Ris. Sist. Inf. dan Tek. Inform.)*, vol. 5, no. 1, p. 154, 2020, doi: 10.30645/jurasik.v5i1.180.