

PENERAPAN TEKNOLOGI *QUICK RESPONSE CODE* DAN *APPLICATION PROGRAMMING INTERFACE* PADA PERANCANGAN APLIKASI PERPUSTAKAAN (STUDI KASUS : SMP NEGERI 25 SURAKARTA)

Annisa Nugraheni¹⁾, Maryam²⁾

^{1, 2)}Program Studi Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta
Jl. Ahmad Yani, Tromol Pos I, Pabelan, Kartasura, Surakarta 57102, Jawa Tengah
e-mail: l200180066@student.ums.ac.id¹⁾, mar290@ums.ac.id²⁾

ABSTRAK

SMP Negeri 25 Surakarta memiliki perpustakaan dengan banyak koleksi bahan pustaka tetapi untuk mengelola data masih menggunakan cara manual. Cara manual memiliki beberapa kekurangan antara lain lambat dalam proses pengolahan data, menimbulkan kesulitan ketika mencari data atau informasi, dan membutuhkan banyak ruang untuk menyimpan data. Oleh karena itu, penulis merancang aplikasi perpustakaan dengan menerapkan teknologi QR Code dan API di SMP Negeri 25 Surakarta. Aplikasi ini dikembangkan dengan metode Software Development Life Cycle (SDLC) model waterfall. Aplikasi perpustakaan ini dirancang berbasis website. Aplikasi ini dibuat menggunakan bahasa pemrograman PHP, framework codeigniter dan bootstrap, serta MySQL sebagai Database Management System (DBMS). Kemudian menambahkan ReST API server pada aplikasi tersebut agar data yang tersimpan dalam database dapat diakses untuk dikembangkan menjadi aplikasi android. Selanjutnya dibuat ReST API client dalam bentuk aplikasi android. ReST API client ini mengakses data milik ReST API server dan dikembangkan menjadi aplikasi baru. Hasil dari penelitian ini adalah aplikasi perpustakaan dengan menerapkan teknologi QR Code dan API di SMP Negeri 25 Surakarta.

Kata Kunci: API, codeigniter, perpustakaan, QR Code, waterfall

ABSTRACT

SMP Negeri 25 Surakarta has a library with a large collection of library materials, but to manage data still use manual methods. The manual method has several drawbacks, including being slow in processing data, causing difficulties when searching for data or information, and requiring a lot of space to store data. Therefore, the authors designed a library application by applying QR Code and API technology at SMP Negeri 25 Surakarta. This application was developed using the Waterfall Model Software Development Life Cycle (SDLC). This library application is designed based on the website. This application is made using the PHP programming language, codeigniter and bootstrap frameworks, and MySQL as a Database Management System (DBMS). Then add the ReST API server to the application so that the data stored in the database can be accessed to be developed into an android application. Next, a ReST API client is made in the form of an android application. This ReST API client accesses data belonging to the ReST API server and is developed into a new application. The result of this research is a library application by applying QR Code and API technology at SMP Negeri 25 Surakarta.

Keywords: API, codeigniter, library, QR Code, waterfall

I. PENDAHULUAN

SMP Negeri 25 Surakarta merupakan salah satu institusi pendidikan yang mempunyai perpustakaan dengan banyak koleksi bahan pustaka dan dikelola oleh petugas perpustakaan. Namun, untuk mengelola data di perpustakaan ini masih menggunakan cara manual. Cara manual mempunyai beberapa kekurangan, antara lain proses pengolahan data menjadi lambat, menimbulkan kesulitan ketika mencari informasi, serta membutuhkan banyak ruang untuk menyimpan data.

Semakin pesatnya perkembangan teknologi informasi saat ini menyebabkan sebagian besar bidang memanfaatkan teknologi informasi. Melalui teknologi informasi membuat pengelolaan data maupun informasi menjadi lebih mudah, cepat, dan meyakinkan [1]. Oleh karena itu, penulis memanfaatkan teknologi informasi untuk mengatasi masalah tersebut, yaitu dengan merancang aplikasi perpustakaan yang menerapkan teknologi *QR Code* dan API di SMP Negeri 25 Surakarta.

Sebelumnya beberapa penelitian sudah pernah dilakukan. Pada penelitian tersebut, aplikasi perpustakaan dirancang hanya untuk satu jenis aplikasi saja, seperti berbasis desktop [2], android [3], atau *website* [4]. Sedangkan saat ini teknologi sudah sangat maju, terbukti dengan terdapat banyak jenis aplikasi. Hal tersebut juga diiringi dengan berkembangnya teknologi yang memungkinkan aplikasi satu dengan lainnya saling terhubung. Teknologi

tersebut adalah *Application Programming Interface* (API).

Penelitian ini memanfaatkan teknologi API sehingga aplikasi yang dirancang dapat dikembangkan menjadi aplikasi lain. Aplikasi ini dirancang untuk berbasis *website* terlebih dahulu. Kemudian ditambahkan ReST API server pada aplikasi berbasis *website* tersebut. Selanjutnya dibuat ReST API *client* dalam bentuk aplikasi *mobile* dengan sistem operasi android. Penelitian ini menghasilkan dua jenis aplikasi, yaitu aplikasi perpustakaan berbasis *website* dan aplikasi daftar koleksi perpustakaan berbasis android.

Selain itu, pada penelitian terdahulu masih sedikit fitur yang dapat digunakan oleh anggota perpustakaan. Pada penelitian [5], anggota perpustakaan hanya dapat mencari buku dan mengisi buku tamu melalui sistem tersebut. Penelitian lain juga pernah dilakukan pada tahun 2017, aplikasi yang dirancang memberikan fitur kepada anggota perpustakaan berupa pencarian buku dan registrasi [6]. Oleh karena itu, pada penelitian ini dirancang aplikasi perpustakaan dengan penambahan fitur untuk anggota perpustakaan. Fitur baru tersebut adalah peminjaman dan perpanjangan peminjaman secara *online*.

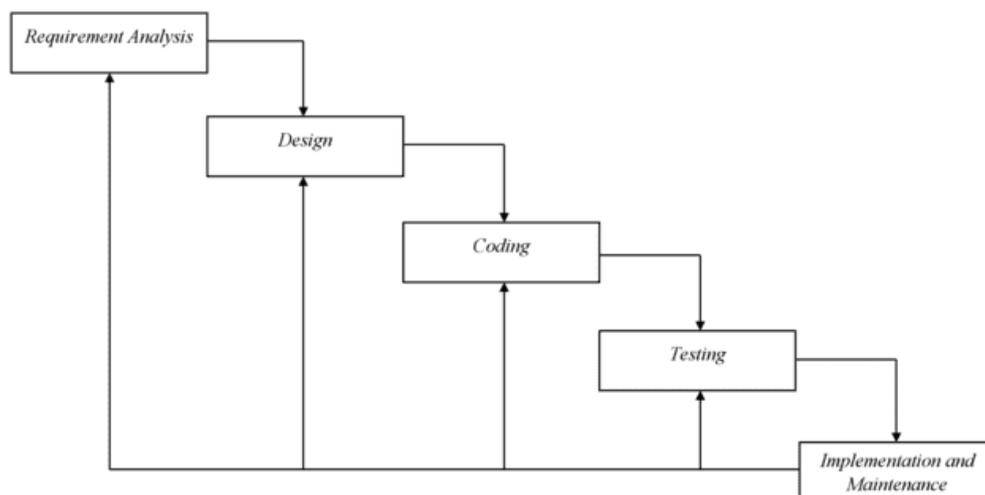
Perbedaan lain terletak pada aplikasi yang dilengkapi dengan fitur *barcode*. Pada penelitian [7], belum terdapat fitur *barcode* untuk peminjaman buku secara langsung. Maka dari itu, pada penelitian ini dirancang aplikasi dengan fitur tersebut. Pada penelitian [8] telah memanfaatkan teknologi *barcode* untuk transaksi di perpustakaan. Namun, jenis *barcode* yang digunakan adalah *barcode* 39 yang memiliki kekurangan dari segi biaya karena harus menggunakan *scanner* khusus.

Jenis *barcode* yang digunakan pada penelitian ini adalah *barcode* yang memiliki kecepatan tinggi untuk menerjemahkan isi dan mampu menyimpan informasi baik secara horizontal maupun vertikal yang disebut *Quick Response Code* (*QR Code*) [9]. Kelebihan *QR Code* jika dibandingkan dengan jenis *barcode* lainnya adalah memiliki kapasitas yang lebih banyak, mampu menyimpan data dengan tipe yang beragam, dapat dibaca dari segala arah sehingga kemungkinan gagal dalam membaca sangat kecil, memiliki ketahanan hingga 30%, dan tidak harus menggunakan *scanner* khusus karena dapat dipindai menggunakan kamera yang terpasang pada laptop atau *smartphone* [10].

Penelitian ini merancang aplikasi perpustakaan yang menerapkan teknologi *QR Code* dan API di SMP Negeri 25 Surakarta serta memberikan manfaat bagi petugas maupun anggota perpustakaan. Melalui penelitian ini, diperoleh aplikasi perpustakaan yang dapat membantu, memudahkan, dan mempercepat dalam mengelola data, mencari informasi, serta tetap dapat menikmati fasilitas perpustakaan meskipun dilakukan secara *online*.

II. METODE PENELITIAN

Software Development Life Cycle (SDLC) merupakan alur yang dilakukan untuk mengembangkan *software* yang memiliki kualitas bagi pengguna [11]. Metode pengembangan aplikasi yang digunakan pada penelitian ini adalah SDLC dengan model *waterfall*. Model ini pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga dianggap sebagai model klasik tetapi paling sering digunakan dalam *software engineering* [12]. Pada model *waterfall*, setiap tahap menghasilkan produk, seperti sebuah aliran yang menuju ke tahap selanjutnya sehingga setiap tahapan harus dilakukan secara berurutan [13]. Model *waterfall* termasuk sederhana, memiliki biaya rendah, serta mempunyai tingkat *maintenance* yang rendah sehingga tidak harus melakukan pemeliharaan secara rutin [14]. Adapun tahapan model *waterfall* ditunjukkan pada Gambar 1.



Gambar 1. Tahapan model *waterfall*

A. Requirement Analysis

Requirement analysis merupakan tahapan pada metode *waterfall* yang digunakan untuk mengumpulkan data. Data tersebut digunakan untuk memenuhi fungsi yang dapat dijalankan oleh aplikasi [15]. *Requirement analysis* terdiri dari dua jenis, yaitu fungsional berisi fitur yang dapat dijalankan oleh aplikasi dan non-fungsional berisi perangkat yang mendukung untuk pengembangan aplikasi baik *software*, *hardware*, maupun *brainware*.

a. Analisis kebutuhan fungsional

- 1) Petugas perpustakaan
 - a) *Login*
 - b) Mengelola data diri, petugas, anggota, koleksi, transaksi, dan data permintaan
 - c) Mencetak laporan
 - d) Mengatur denda, batas waktu peminjaman, data pada laporan, seperti tahun pelajaran serta nama dan NIP kepala sekolah
 - e) Memverifikasi peminjaman, perpanjangan peminjaman, dan pendaftaran anggota perpustakaan
 - f) *Logout*
 - 2) Anggota perpustakaan
 - a) Mendaftar atau membuat akun sebagai anggota perpustakaan
 - b) *Login*
 - c) Mengelola profil atau data diri
 - d) Melihat dan mencari koleksi
 - e) Melakukan permintaan peminjaman dan perpanjangan peminjaman
 - f) *Logout*
- #### b. Analisis kebutuhan non-fungsional
- 1) Kebutuhan *software*
Web browser untuk mengakses aplikasi perpustakaan
 - 2) Kebutuhan *hardware*
 - a) Laptop atau PC atau *smartphone* untuk mengakses aplikasi perpustakaan
 - b) Web *camera* yang sudah terpasang pada laptop maupun *smartphone* atau perangkat keras eksternal untuk memindai *QR Code*
 - 3) Kebutuhan *brainware (user)*
 - a) Petugas perpustakaan yang berperan dalam mengelola data
 - b) Anggota perpustakaan yang berperan menerima data atau informasi

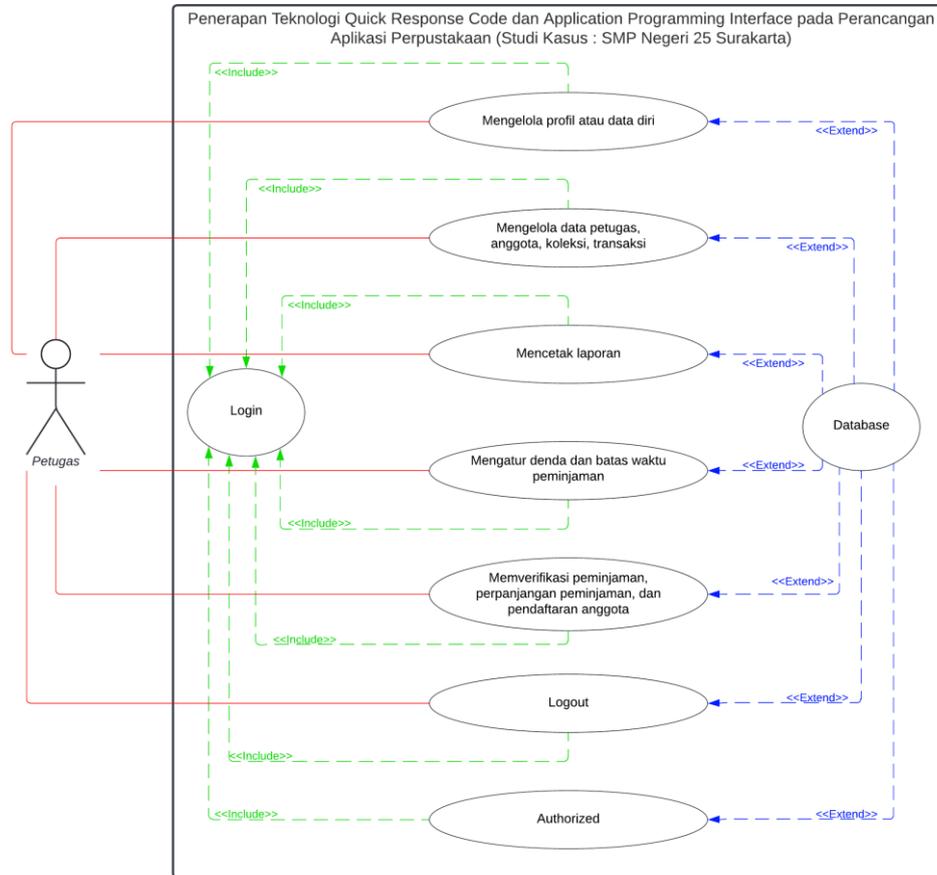
B. Design

Tahap *design* merupakan perancangan sistem dengan *Unified Modelling Language (UML)* yang terdiri dari *use case diagram*, *activity diagram*, dan *Entity Relationship Diagram (ERD)* [16].

a. Use case diagram

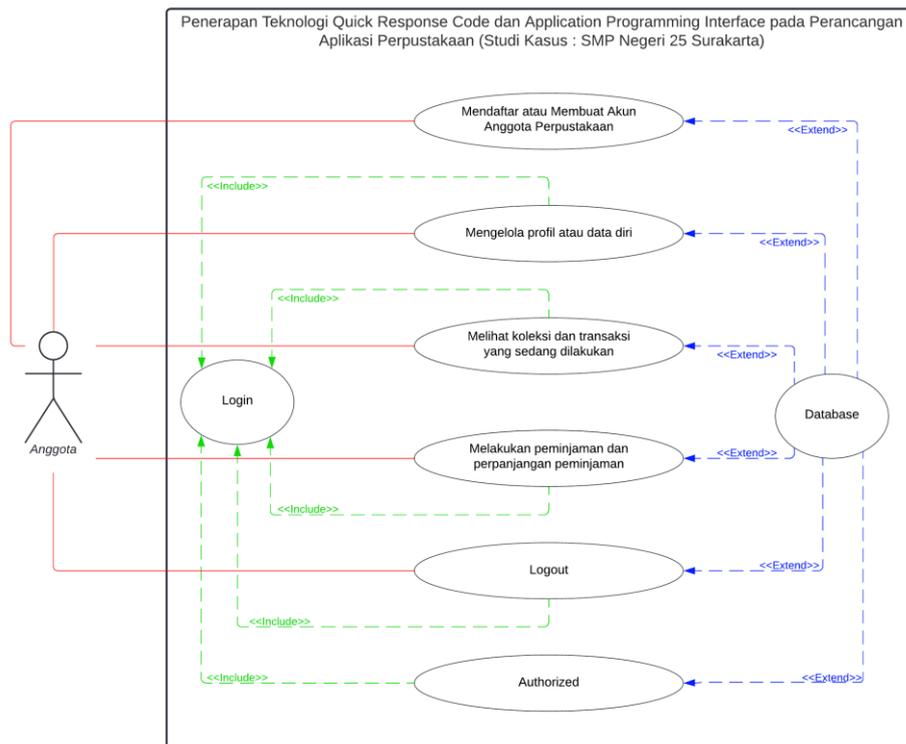
Use case diagram merupakan kumpulan skenario yang mungkin mengenai komunikasi antara sistem dan pengguna [17]. Gambar 2 merupakan *use case diagram* yang menunjukkan fungsi yang dapat diakses oleh *level user* petugas perpustakaan. Gambar 3 merupakan *use case diagram* yang menunjukkan fungsi yang dapat diakses oleh *level user* anggota perpustakaan.

Use Case Diagram
 Petugas Perpustakaan



Gambar 2. Use case diagram petugas perpustakaan

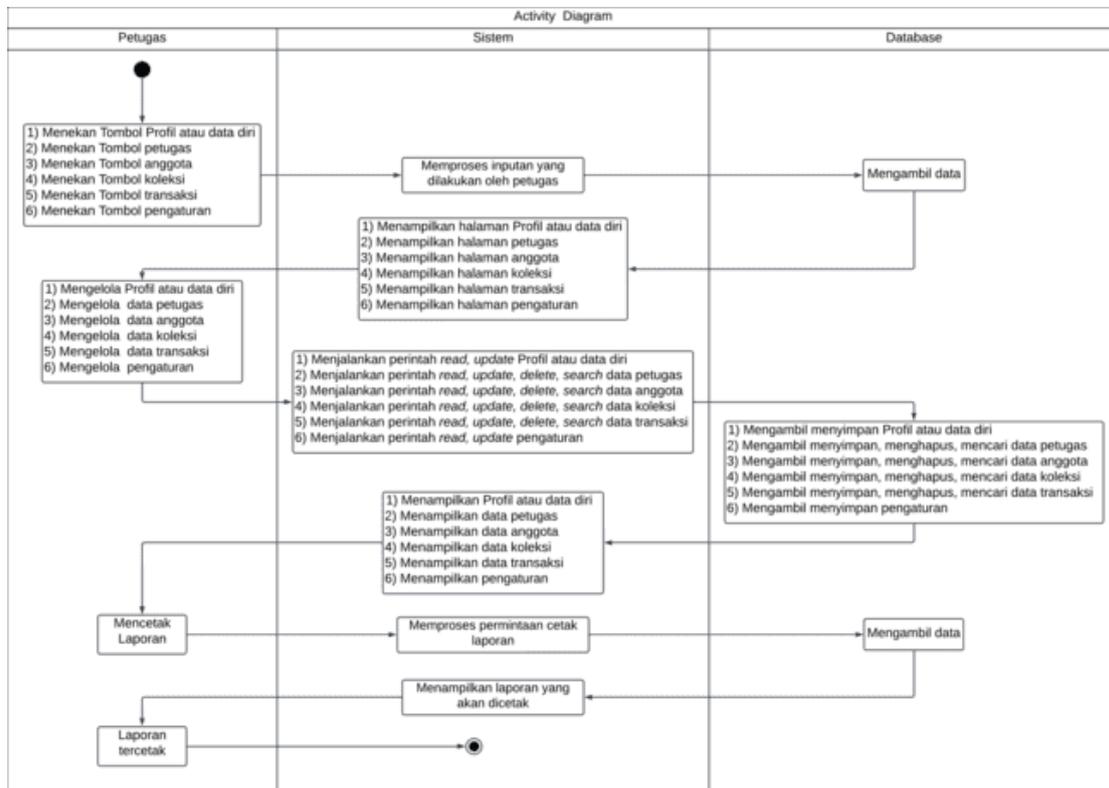
Use Case Diagram
 Anggota Perpustakaan



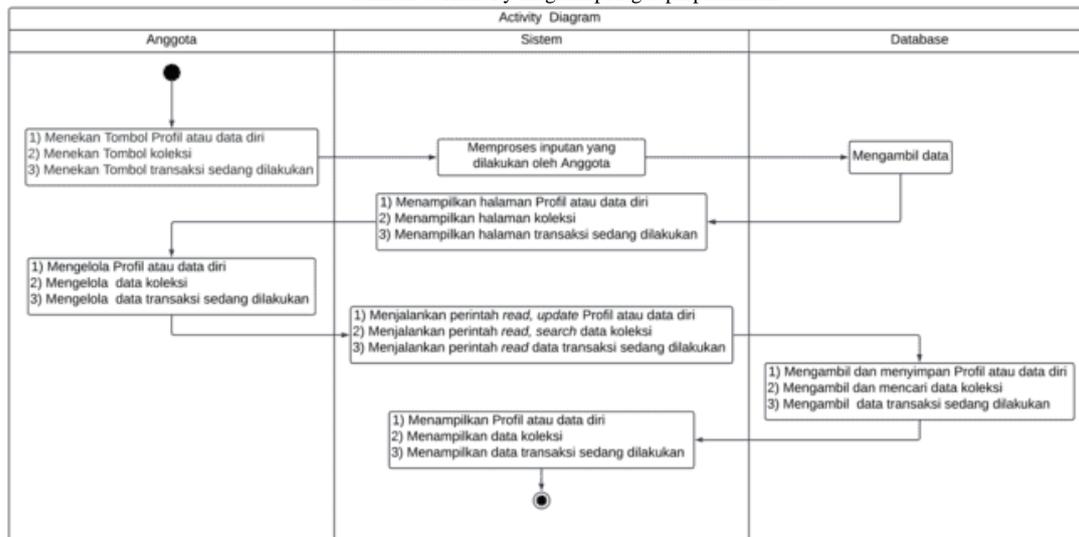
Gambar 3. Use case diagram anggota perpustakaan

b. *Activity diagram*

Activity diagram digunakan untuk menggambarkan perilaku sistem dan cara perilaku ini berinteraksi satu sama lain dengan mendefinisikan urutan tindakan diantara mereka [18]. Gambar 4 merupakan *activity diagram* yang menunjukkan alur interaksi antara petugas, sistem, dan *database*. Gambar 5 adalah *activity diagram* yang menggambarkan alur interaksi yang dilakukan oleh anggota, sistem, serta *database*.



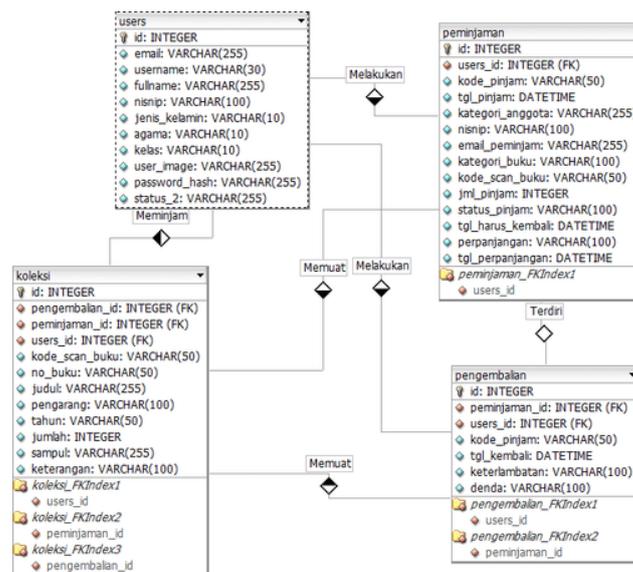
Gambar 4. Activity diagram petugas perpustakaan



Gambar 5. Activity diagram anggota perpustakaan

c. *Entity Relationship Diagram (ERD)*

ERD merupakan salah satu alat representasi visual data yang berfokus pada entitas sistem dan hubungan diantara entitas [19]. Gambar 6 menggambarkan entitas-entitas pada aplikasi ini dan hubungan diantara entitas tersebut.



Gambar 6. Entity Relationship Diagram

C. Coding

Aplikasi perpustakaan ini dirancang untuk versi *website* terlebih dahulu dengan menggunakan bahasa pemrograman PHP, *framework codeigniter* dan *bootstrap*, serta MySQL sebagai *Database Management System* (DBMS). Kemudian memanfaatkan API supaya data yang tersimpan pada *database* aplikasi versi *website* dapat digunakan untuk mengembangkan aplikasi lain. Pada penelitian ini, dirancang aplikasi daftar koleksi perpustakaan berbasis android menggunakan *ReST API client* untuk menguji bahwa API sudah berjalan.

Aplikasi berbasis *website* ini menggunakan *codeigniter* yang merupakan suatu kerangka kerja dengan fungsi memudahkan dalam pembuatan aplikasi *web* PHP sebagai *framework* untuk menangani kode program [20]. Aplikasi ini menggunakan *codeigniter* karena mempunyai jumlah *file* terkecil dan paling sederhana dibandingkan dengan *framework* PHP lainnya serta memiliki kinerja yang paling baik untuk data kompleks, operasi *Create Read Update Delete* (CRUD), dan unggah gambar [21].

Framework untuk menangani pembuatan tampilan pada aplikasi ini menggunakan *bootstrap* yang merupakan kerangka kerja *front end* dari *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS), dan *Java Script* yang paling populer untuk membuat aplikasi *website* dengan desain yang responsif serta mempermudah dan mempercepat dalam pengembangan web [22]. Suatu *website* dengan desain yang responsif memungkinkan untuk diakses oleh berbagai macam perangkat dengan ukuran layar yang berbeda, memudahkan dalam proses pemeliharaan, serta hanya memerlukan satu alamat domain saja [23].

D. Testing

Tahap *testing* dilakukan menggunakan pengujian *blackbox* dan *System Usability Scale* (SUS). Pengujian *blackbox* merupakan pengujian perangkat lunak untuk menentukan fungsionalitas aplikasi yang berfokus pada *output* yang diharapkan untuk setiap input [24]. SUS adalah pengujian dengan kuesioner standar yang dirancang untuk menilai kegunaan sistem yang dirasakan *user* [25]. Indikator dalam pengujian SUS, yaitu terdapat sepuluh pernyataan yang menjadi pedoman pengukuran pengujian. Setiap pernyataan memiliki bobot nilai dari 1 yang berarti Sangat Tidak Setuju (STS) sampai dengan 5 yang artinya Sangat Setuju (SS). Tabel 1 merupakan daftar pernyataan yang menjadi tolak ukur dalam pengujian SUS [26]. Adapun langkah penghitungan hasil pengujian SUS sebagai berikut : [16]

- Perhitungan skor memperhatikan pernyataan nomor ganjil dan genap
- Pernyataan nomor ganjil dihitung melalui jawaban dikurangi 1 sedangkan pernyataan nomor genap dihitung dari 5 dikurangi jawaban yang dipilih
- Melakukan penjumlahan skor pada masing-masing pernyataan untuk setiap responden kemudian hasilnya dikalikan dengan 2,5
- Menghitung rata-rata nilai
- Rata-rata nilai yang didapatkan diklasifikasikan sebagai berikut :
 - Usable* untuk rata-rata nilai diatas 70. Klasifikasi ini berarti sistem dapat diterima atau layak digunakan
 - Marginal* untuk rata-rata nilai antara 50 – 70. Klasifikasi ini berarti sistem masih perlu untuk ditingkatkan
 - Unacceptable* untuk nilai rata-rata dibawah 50. Klasifikasi ini berarti sistem tidak dapat diterima

TABEL I
PERNYATAAN PENGUJIAN SUS

No	Pernyataan	Bobot Nilai
1	Saya berpikir akan menggunakan aplikasi ini lagi	1 - 5
2	Saya merasa aplikasi ini rumit untuk digunakan	1 - 5
3	Saya merasa aplikasi ini mudah digunakan	1 - 5
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan aplikasi ini	1 - 5
5	Saya merasa fitur-fitur pada aplikasi ini berjalan dengan semestinya	1 - 5
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada aplikasi ini	1 - 5
7	Saya merasa kebanyakan orang akan memahami cara menggunakan aplikasi ini dengan cepat	1 - 5
8	Saya merasa aplikasi ini membingungkan	1 - 5
9	Saya merasa tidak ada hambatan dalam menggunakan aplikasi ini	1 - 5
10	Saya perlu belajar banyak hal terlebih dahulu sebelum saya dapat menggunakan aplikasi ini	1 - 5

E. Implementation and maintenance

Tahap terakhir, yaitu implementasi dan pemeliharaan (*maintenance*). Aplikasi ini diimplementasikan di SMP Negeri 25 Surakarta. Aplikasi versi *website* didaftarkan pada *web hosting* sehingga tidak perlu mengakses melalui *localhost*. Aplikasi versi android hanya dirancang untuk menguji fitur API sehingga diimplementasikan menggunakan emulator. Selain itu, dilakukan juga pelatihan kepada pengguna agar memahami tentang penggunaan aplikasi tersebut. Pemeliharaan aplikasi juga dilakukan untuk memperbaiki apabila terjadi kerusakan pada sistem.

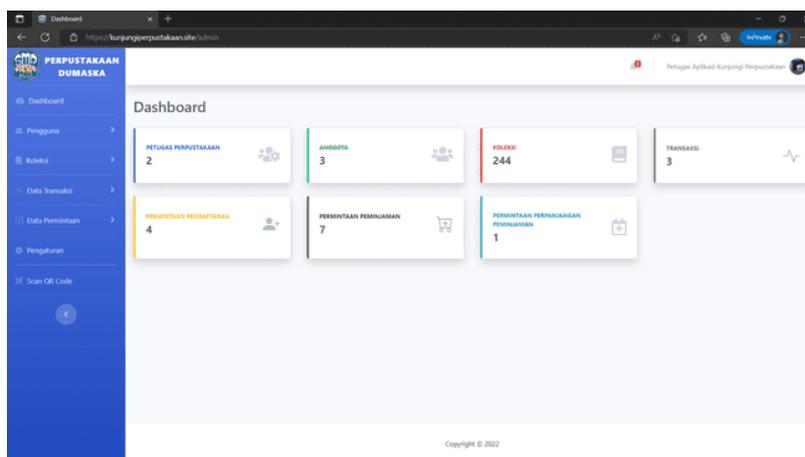
III. HASIL DAN PEMBAHASAN

A. Hasil

Penelitian ini menghasilkan aplikasi perpustakaan yang mendukung fitur *QR code*. Aplikasi ini juga menggunakan teknologi API sehingga data dari aplikasi versi *website* dapat digunakan untuk membuat aplikasi lain, dalam penelitian ini berupa aplikasi daftar koleksi perpustakaan berbasis android. Aplikasi ini dapat digunakan oleh petugas dan anggota perpustakaan. Sistem menampilkan menu yang dapat diakses sesuai dengan level *user*.

a. Halaman utama petugas perpustakaan

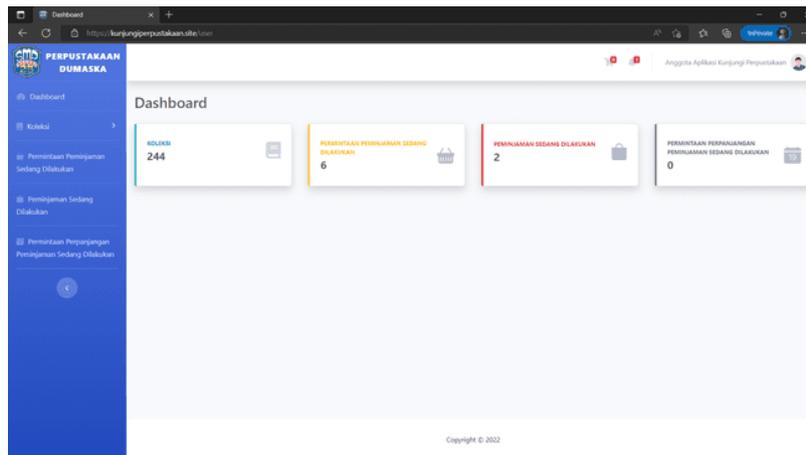
Pada halaman utama petugas perpustakaan tampil beberapa menu, yaitu *dashboard*, pengguna, koleksi, data transaksi, data permintaan, pengaturan, dan *scan QR code*. Gambar 7 merupakan tampilan halaman utama petugas perpustakaan.



Gambar 7. Halaman utama petugas perpustakaan

b. Halaman utama anggota perpustakaan

Pada halaman utama anggota perpustakaan tampil beberapa menu, seperti *dashboard*, koleksi, permintaan peminjaman sedang dilakukan, peminjaman sedang dilakukan, dan permintaan perpanjangan peminjaman sedang dilakukan. Gambar 8 adalah tampilan dari halaman utama anggota perpustakaan.

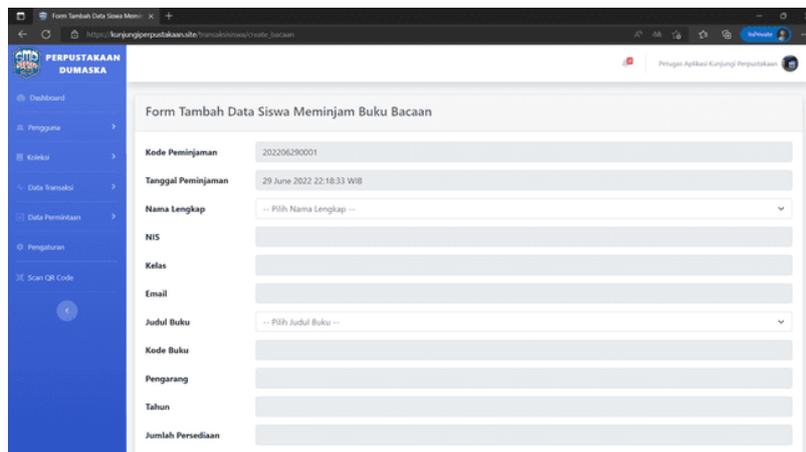


Gambar 8. Halaman utama anggota perpustakaan

c. Menu data transaksi

Menu data transaksi terbagi menjadi dua, yaitu transaksi yang dilakukan oleh siswa dan guru atau karyawan. Transaksi dapat berupa meminjam dan mengembalikan buku serta melakukan perpanjangan waktu peminjaman. Menu ini hanya dapat diakses oleh level *user* petugas perpustakaan. Melalui menu ini, petugas perpustakaan dapat melihat semua daftar transaksi yang sedang maupun telah selesai dilakukan. Petugas perpustakaan juga dapat mengelola data transaksi, berupa melakukan tambah data, cetak data dalam bentuk PDF, *excel*, maupun *print hardfile*. Fungsi cetak data dapat digunakan untuk mencetak semua data ataupun hanya data tertentu berdasarkan tanggal yang diinginkan. Petugas juga dapat melihat detail masing-masing data, mencari data, dan menghapus data.

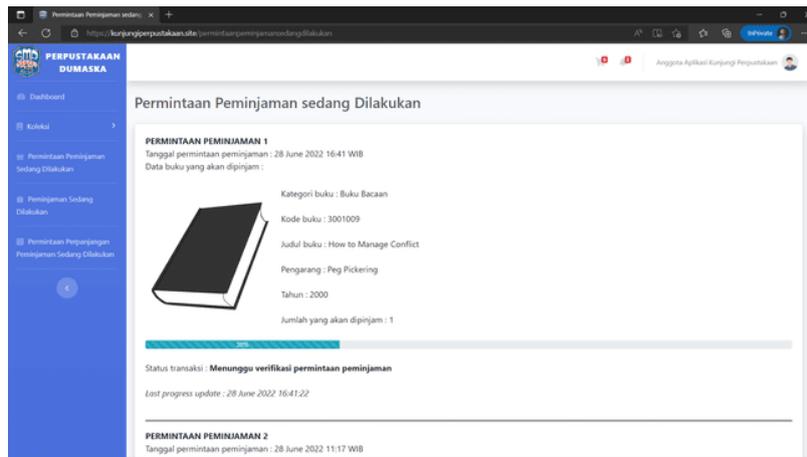
Salah satu data yang ditampilkan pada menu transaksi adalah status peminjaman. Status peminjaman akan berubah menjadi melewati batas waktu apabila buku belum dikembalikan dan waktu peminjaman telah selesai. Setiap petugas perpustakaan *login* atau membuka halaman *dashboard*, sistem bekerja mencari data peminjaman yang sudah melewati batas waktu dan buku belum dikembalikan. Kemudian mengubah status peminjaman menjadi melewati batas waktu serta mengirim peringatan kepada peminjam melalui pesan email dan notifikasi pada aplikasi ini. Gambar 9 merupakan halaman tambah pada menu data transaksi.



Gambar 9. Halaman tambah pada menu data transaksi

d. Halaman permintaan peminjaman sedang dilakukan

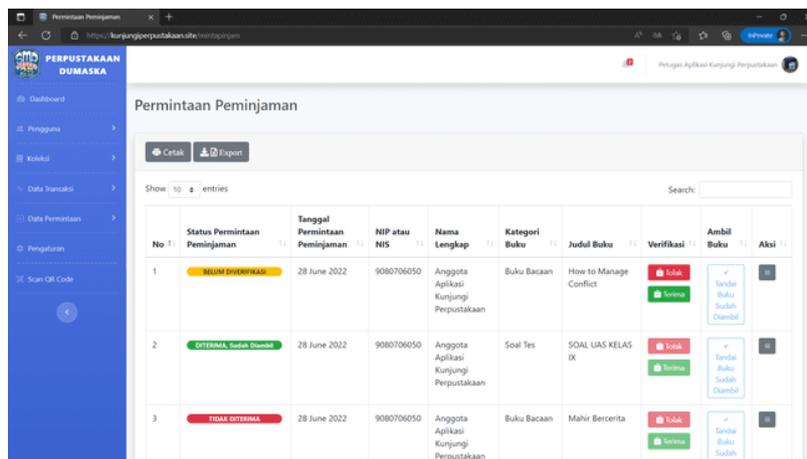
Halaman permintaan peminjaman sedang dilakukan hanya dapat diakses oleh level *user* anggota perpustakaan. Halaman ini berguna untuk menampilkan daftar permintaan peminjaman buku yang sedang dilakukan oleh anggota perpustakaan yang sedang *login*. Melalui halaman ini, anggota perpustakaan dapat melihat detail data permintaan peminjaman yang sedang ia lakukan. Pada halaman ini juga ditampilkan status permintaan peminjaman dan *progress bar*. Tampilan dari halaman permintaan peminjaman sedang dilakukan ditunjukkan oleh gambar 10.



Gambar 10. Halaman permintaan peminjaman sedang dilakukan

e. Menu verifikasi permintaan peminjaman

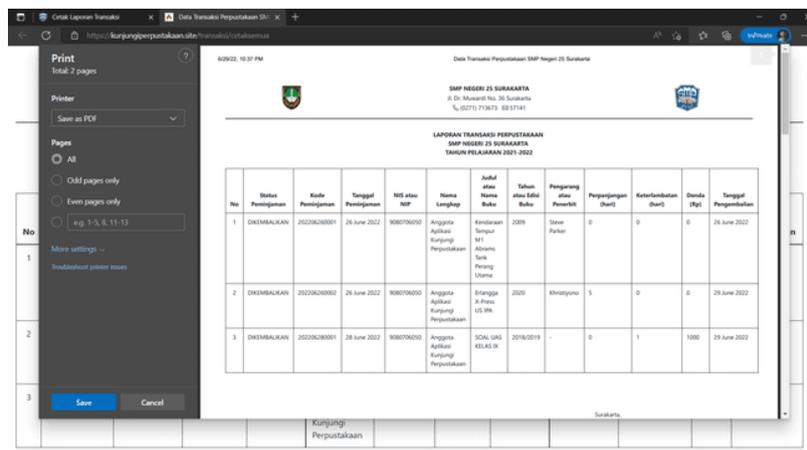
Menu verifikasi permintaan peminjaman hanya dapat diakses oleh *level user* petugas perpustakaan. Menu ini berfungsi untuk menerima atau menolak permintaan peminjaman buku yang dilakukan oleh anggota. Gambar 11 merupakan tampilan halaman permintaan peminjaman.



Gambar 11. Halaman permintaan peminjaman untuk verifikasi

f. Halaman cetak laporan

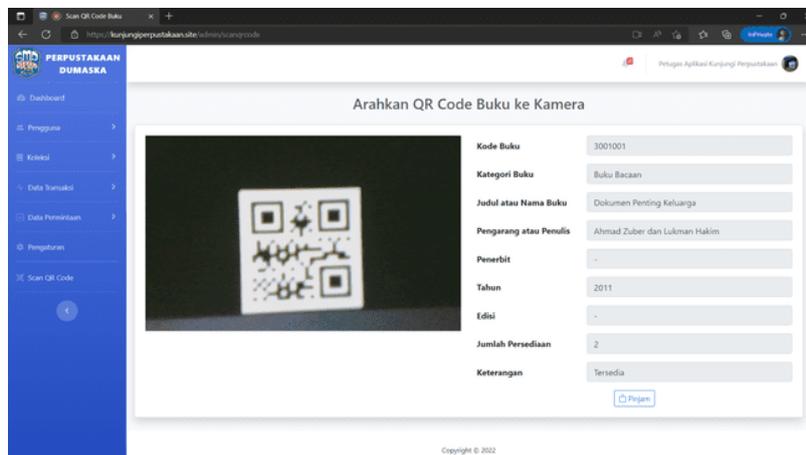
Halaman cetak laporan hanya dapat diakses oleh *level user* petugas perpustakaan. Halaman ini berfungsi untuk mencetak laporan transaksi yang dilakukan di perpustakaan tersebut, seperti peminjaman dan pengembalian buku serta permintaan perpanjangan waktu peminjaman. Hasil cetak laporan dapat berupa PDF, excel, maupun *printout*. Petugas dapat memilih untuk mencetak semua data atau hanya beberapa data berdasarkan tanggal yang diinginkan. Gambar 12 menunjukkan tampilan daftar peminjaman saat dicetak.



Gambar 12. Tampilan daftar peminjaman saat dicetak

g. Menu *QR code*

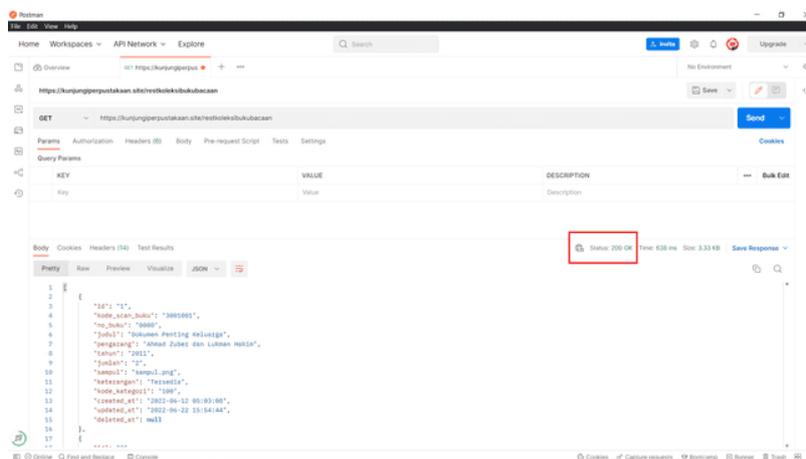
Menu *QR code* pada aplikasi ini terdiri dari dua jenis, yaitu *QR code* generator untuk mencetak dan *QR code* reader untuk memindai. Menu ini berfungsi untuk memudahkan dalam pencarian dan peminjaman buku. Gambar 13 merupakan tampilan *QR code* reader ketika melakukan pemindaian. Data hasil pemindaian muncul di sebelah *QR code* reader.



Gambar 13. Tampilan QR code reader ketika melakukan pemindaian

h. ReST API server

Aplikasi perpustakaan versi *website* yang sudah dikembangkan diberi fitur API. Pada penelitian ini, aplikasi tersebut memiliki kedudukan sebagai ReST API server yang berfungsi mengirim data dari *database* yang nantinya diakses oleh ReST API *client*. Penelitian ini menggunakan aplikasi Postman untuk menguji bahwa aplikasi perpustakaan versi *website* telah berhasil mengirim data. Gambar 14 merupakan hasil pengujian ReST API server menggunakan aplikasi Postman. Hasil tersebut menampilkan bahwa ReST API server telah berhasil mengirim data yang ditunjukkan oleh status kode 200 dan telah berhasil menangkap data yang dikirim.



Gambar 14. Hasil pengujian ReST API server menggunakan aplikasi Postman

i. ReST API *client*

ReST API *client* pada penelitian ini adalah aplikasi daftar koleksi perpustakaan berbasis android. Aplikasi ini berfungsi untuk menguji bahwa data yang telah dikirim oleh ReST API server dapat ditampilkan pada aplikasi yang berbeda. Gambar 15 merupakan tampilan daftar koleksi pada aplikasi android yang sudah dirancang dengan memanfaatkan teknologi API.



Gambar 15. Tampilan daftar koleksi pada aplikasi android

B. Pengujian Black Box

Pengujian *black box* dilakukan dengan tujuan memastikan aplikasi yang dikembangkan telah berfungsi dengan baik melalui uji coba dengan kondisi tertentu dan harapan yang ingin dicapai. Tabel 2 merupakan hasil pengujian *black box*.

TABEL II
HASIL PENGUJIAN *BLACK BOX*

No	Test Case	Test Case Description	Expected Result	Actual Result	Status
1.	Login	Menguji fungsi login dengan <i>username</i> , email, dan <i>password</i> benar	Berhasil login dan menampilkan halaman utama atau <i>dashboard</i>	Sesuai harapan	pass
		Menguji fungsi login dengan <i>username</i> , email, <i>password</i> salah	Gagal login dan tetap di halaman login	Sesuai harapan	pass
2.	Pendaftaran atau pembuatan akun anggota perpustakaan	Anggota perpustakaan melakukan permintaan pendaftaran	Data permintaan pendaftaran dapat terkirim ke email petugas dan tersimpan dalam <i>database</i>	Sesuai harapan	pass
		Verifikasi permintaan pendaftaran oleh petugas perpustakaan	Hasil verifikasi permintaan pendaftaran dapat tersimpan dalam <i>database</i> dan terkirim ke email pendaftar	Sesuai harapan	pass
3.	Peminjaman buku	Anggota perpustakaan melakukan permintaan peminjaman buku	Data permintaan peminjaman dapat tersimpan dalam <i>database</i> dan terkirim ke email petugas	Sesuai harapan	pass
		Petugas perpustakaan melakukan verifikasi permintaan peminjaman buku	Hasil verifikasi permintaan peminjaman dapat tersimpan dalam <i>database</i> dan terkirim ke email anggota	Sesuai harapan	pass
		Anggota perpustakaan melihat hasil verifikasi permintaan peminjaman buku	Dapat menampilkan detail data permintaan peminjaman termasuk hasil verifikasi	Sesuai harapan	pass
		Petugas perpustakaan memberi tanda pada permintaan peminjaman yang sudah diambil	Data pengambilan buku dapat tersimpan dalam <i>database</i> dan menambahkan data tersebut sebagai transaksi peminjaman baru	Sesuai harapan	pass

4.	Anggota perpustakaan melihat data peminjaman	Menguji fungsi peminjaman sedang dilakukan untuk melihat daftar peminjaman yang sedang dilakukan	Dapat menampilkan detail data peminjaman sedang dilakukan	Sesuai harapan	<i>pass</i>
	Perpanjangan waktu peminjaman buku				
	Anggota perpustakaan melakukan permintaan perpanjangan peminjaman	Menguji fungsi permintaan perpanjangan peminjaman melalui halaman peminjaman sedang dilakukan	Data permintaan perpanjangan peminjaman tersimpan dalam <i>database</i> dan terkirim ke email petugas	Sesuai harapan	<i>pass</i>
	Petugas perpustakaan melakukan verifikasi permintaan perpanjangan peminjaman	Menguji fungsi verifikasi permintaan perpanjangan waktu peminjaman buku	Hasil verifikasi permintaan perpanjangan peminjaman dapat tersimpan dalam <i>database</i> dan terkirim ke email anggota	Sesuai harapan	<i>pass</i>
	Anggota perpustakaan melihat data permintaan perpanjangan peminjaman	Menguji fungsi permintaan perpanjangan peminjaman sedang dilakukan untuk melihat daftar permintaan perpanjangan peminjaman yang sedang dilakukan	Dapat menampilkan detail data permintaan perpanjangan peminjaman sedang dilakukan	Sesuai harapan	<i>pass</i>
5.	Detail	Menguji fungsi detail pada setiap halaman untuk melihat setiap data yang tersimpan secara detail	Dapat menampilkan detail data sesuai dengan data yang dipilih	Sesuai harapan	<i>pass</i>
6.	Edit	Menguji fungsi edit pada setiap halaman untuk mengubah data yang ingin diperbarui oleh pengguna	Dapat menyimpan perubahan data ke dalam <i>database</i>	Sesuai harapan	<i>pass</i>
7.	Hapus	Menguji fungsi hapus pada setiap halaman untuk menghapus data yang dipilih oleh pengguna	Dapat menghapus data dari <i>database</i>	Sesuai harapan	<i>pass</i>
8.	Pencarian	Menguji fungsi cari pada setiap halaman untuk mencari data sesuai kata kunci yang dimasukkan	Dapat menampilkan data yang dicari	Sesuai harapan	<i>pass</i>
9.	Tambah	Menguji fungsi tambah pada setiap halaman untuk menambahkan data baru	Dapat menambahkan data baru ke dalam <i>database</i>	Sesuai harapan	<i>pass</i>
10.	Cetak				
	Cetak semua data	Menguji fungsi cetak semua data pada setiap halaman untuk mencetak semua data dalam bentuk PDF	Dapat mencetak semua data dalam bentuk PDF	Sesuai harapan	<i>pass</i>
	Cetak data berdasarkan tanggal	Menguji fungsi cetak berdasarkan tanggal pada halaman transaksi untuk mencetak data tertentu berdasarkan tanggal yang diinginkan dalam bentuk PDF	Dapat mencetak data sesuai tanggal yang diinginkan dalam bentuk PDF	Sesuai harapan	<i>pass</i>
11.	<i>Export excel</i>				
	<i>Export excel</i> semua data	Menguji fungsi <i>export excel</i> semua data pada setiap halaman untuk mencetak semua data dalam bentuk <i>excel</i>	Dapat mencetak semua data dalam bentuk <i>excel</i>	Sesuai harapan	<i>pass</i>
	<i>Export excel</i> berdasarkan tanggal	Menguji fungsi <i>export excel</i> berdasarkan tanggal pada halaman transaksi untuk mencetak data tertentu berdasarkan tanggal yang diinginkan dalam bentuk <i>excel</i>	Dapat mencetak data sesuai tanggal yang diinginkan dalam bentuk <i>excel</i>	Sesuai harapan	<i>pass</i>
12.	<i>Import excel</i>	Menguji fungsi <i>import excel</i> untuk memasukkan data dari <i>file excel</i> ke dalam <i>database</i>	Dapat menyimpan data dari <i>file excel</i> ke dalam <i>database</i>	Sesuai harapan	<i>pass</i>
13.	<i>QR code</i>				
	Cetak <i>QR code</i> semua data	Menguji fungsi cetak <i>QR code</i> semua data pada halaman koleksi untuk mencetak semua <i>QR code</i> dalam bentuk PDF	Dapat mencetak semua <i>QR code</i> dalam bentuk PDF	Sesuai harapan	<i>pass</i>

	Cetak <i>QR code</i> berdasarkan kode buku	Menguji fungsi cetak <i>QR code</i> berdasarkan kode buku pada halaman koleksi untuk mencetak <i>QR code</i> buku yang dipilih dalam bentuk PDF	Dapat mencetak <i>QR code</i> sesuai buku yang dipilih dalam bentuk PDF	Sesuai harapan	<i>pass</i>
	<i>Scan QR code</i>	Menguji fungsi <i>scan QR code</i> yang berfungsi untuk membaca data dalam <i>QR code</i>	Dapat membaca dan menampilkan data yang tersimpan dalam <i>QR code</i>	Sesuai harapan	<i>pass</i>
14.	<i>Logout</i>	Menguji fungsi <i>logout</i> untuk keluar dari sistem	Dapat keluar dari sistem dan kembali ke halaman <i>login</i>	Sesuai harapan	<i>pass</i>

C. Pengujian SUS

Hasil pengujian SUS didapatkan melalui kuesioner yang diisi oleh 34 responden. Kuesioner tersebut berisi sepuluh pernyataan dengan lima skala poin, yaitu 1 berarti Sangat Tidak Setuju (STS) sampai dengan 5 yang berarti Sangat Setuju (SS). Pada penelitian ini diperoleh nilai rata-rata pengujian SUS, yaitu 82,823 sehingga aplikasi ini termasuk dalam kategori *Usable*.

IV. KESIMPULAN

Berdasarkan pengujian *black box* yang telah dilakukan, aplikasi perpustakaan ini dapat berjalan sesuai dengan fungsi yang diharapkan. Aplikasi ini mampu mengelola data petugas, anggota, koleksi, transaksi, memindai *QR code*, dan mencetak laporan. Sedangkan berdasarkan pengujian SUS diperoleh rata-rata nilai, yaitu 82,823 sehingga aplikasi ini termasuk dalam kategori *Usable*. Aplikasi perpustakaan berbasis *website* sebagai ReST API server telah berhasil mengirim data dan aplikasi daftar koleksi perpustakaan berbasis android sebagai ReST API *client* telah berhasil menerima serta menampilkan data dari ReST API server.

DAFTAR PUSTAKA

- [1] D. W. Kurniawan and F. Y. Irsyadi, "Perancangan dan Pembuatan Aplikasi Manajemen Peminjaman Kendaraan Berbasis Web Dengan Framework Codeigniter," *Emit. J. Tek. Elektro*, vol. 21, no. 01, pp. 49–53, 2021.
- [2] R. Y. Panjaitan, H. V. Suparyo, and L. Sunarmintyastuti, "Perancangan Aplikasi Perpustakaan Berbasis Desktop di SMP Negeri 01 Citeureup," *J. Ris. dan Apl. Mhs. Inform.*, vol. 1, no. 01, pp. 17–23, 2020, doi: 10.30998/jrami.v1i01.151.
- [3] I. K. Wardani and Z. Rifa'i, "Rancang Bangun Aplikasi Perpustakaan Berbasis Android Menggunakan App Inventor," *J. Ilmu-Ilmu Inform. dan Manaj. STMIK*, vol. 15, no. 1, pp. 49–58, 2021.
- [4] D. W. Dari, A. O. Sari, and A. Astrilyana, "Rancang Bangun Aplikasi Perpustakaan Sekolah Berbasis Website," *J. Ilmu Pengetah. dan Teknol. Komput.*, vol. 4, no. 2, pp. 163–168, 2019, [Online]. Available: <http://ejournal.nusamandiri.ac.id/ejournal/index.php/jitk/article/view/929>.
- [5] S. Granita, S. Rohmaniyah, T. Gautama, and Y. Yulianti, "Pengembangan Sistem Informasi Perpustakaan Sekolah Berbasis Web," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 4, pp. 246–251, 2020, doi: 10.32493/jtsi.v3i4.7184.
- [6] B. D. Heriyanto, I. A. Kurniawan, and I. Taufik, "Sistem Aplikasi Perpustakaan Berbasis Web Menggunakan Framework Codeigniter pada SMAS Islamic Centre," *J. Ilm. Sisfotenika*, vol. 7, no. 2, pp. 187–196, 2017, doi: 10.30700/jst.v7i2.147.
- [7] R. Herawati, H. D. Wahyuningsih, and A. K. Prasetyo, "Aplikasi Perpustakaan STMIK AUB Surakarta Berbasis Android," *Go Infotech J. Ilm. STMIK AUB*, vol. 25, no. 2, pp. 97–111, 2019, doi: 10.36309/goi.v25i2.108.
- [8] Y. S. Nugroho and A. Basith, "Pemanfaatan Barcode untuk Transaksi di Perpustakaan Universitas Muhammadiyah Surakarta," *Pros. Semin. Nas. Sains dan Teknol. Fak. Tek.*, vol. 1, no. 1, pp. 80–86, 2010.
- [9] K. R. Agizki, R. Hayami, and H. Mukhtar, "Penerapan Quick Response (QR) Code Berbasis Web Di Puskesmas Payung Sekaki Pekanbaru," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 1, no. 1, pp. 1–7, 2020, doi: 10.37859/coscitech.v1i1.1795.
- [10] D. Q. P. A. Paramarta, A. Kusyanti, and M. Data, "Implementasi Algoritme Advance Encryption Standard (AES) pada Enkripsi dan Dekripsi QR-Code," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 12, pp. 6729–6736, 2018.
- [11] M. M. Gultom and M. Maryam, "Sistem Informasi Penjualan Material Bangunan Pada Toko Bangunan Berkah," *J. Tek. Inform.*, vol. 1, no. 2, pp. 79–86, 2020.
- [12] W. W. Royce, *Managing the Development of Large Software Systems: Concepts and Techniques*, no. March. 1987.
- [13] D. Bell, *Software Engineering for Students: A Programming Approach*, 4th ed. Edinburgh, England: Pearson Education, 2005.
- [14] M. V. Polishwala and D. A. kumar Shastri, "Comparative Analysis of Various Software Development Approaches," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 2, no. 3, pp. 88–91, 2021, doi: 10.48175/ijarsct-v2-i3-315.
- [15] W. N. Fathoni and M. Maryam, "Rancang Bangun Sistem Informasi Pelayanan Surat Keterangan Berbasis Web (Studi Kasus : Desa Dawungan Kecamatan Masaran Kabupaten Sragen)," *J. Pendidik. dan Teknol. Indones.*, vol. 1, no. 5, pp. 199–208, 2021, doi: 10.52436/1.jpti.38.
- [16] A. D. Rahmawati and A. Fatmawati, "Sistem Administrasi Desa Mendiro Kecamatan Ngrambe Kabupaten Ngawi Berbasis Web," *Emit. J. Tek. Elektro*, vol. 20, no. 02, pp. 134–140, 2020, doi: 10.23917/emit.v20i02.9893.
- [17] I. Nica and S. Ionescu, "Business process modeling. Using Unified Modeling Language to streamline the design of the TO-BE system within a company," *Theor. Appl. Econ.*, vol. XXVII, no. 4, pp. 89–106, 2020.
- [18] A. Jaffari and C. J. Yoo, "An Experimental Investigation into Data Flow Annotated- Activity Diagram-Based Testing," *J. Comput. Sci. Eng.*, vol. 13, no. 3, pp. 107–123, 2019, doi: 10.5626/JCSE.2019.13.3.107.
- [19] K. Lachová and P. Trebuña, "Modelling of Electronic Kanban System by Using of Entity Relationship Diagrams," *Acta Logist. - Int. Sci. J. about Logist.*, vol. 6, no. 3, pp. 63–66, 2019, doi: 10.22306/al.v6i3.115.
- [20] C. Vidal-Silva, C. Jiménez, E. Madariaga, and L. Urzúa, "Applying PHP Codeigniter for Easy Web Development," *Int. J. Sci. Technol. Res.*, vol. 9, no. 3, pp. 4209–4211, 2020.
- [21] A. Herdiansah, "System Development for Learning Process Monitoring in Private Lesson Institution Using Codeigniter Framework," *J. Informatics Sci.*, vol. 4, no. 1, pp. 10–16, 2021, doi: 10.31326/jisa.v4i1.861.
- [22] S. S. Gaikwad and P. Adkar, "A Review Paper on Bootstrap Framework," *IRE Journals (Iconic Res. Eng. Journals)*, vol. 2, no. 10, pp. 349–351, 2019.



- [23] M. R. Anwar, M. Hardini, and M. Anggraeni, "Review of Responsive Design Concept Based On Framework Materialize On The Website," *ADI J. Recent Innov.*, vol. 3, no. 1, pp. 59–65, 2021.
- [24] A. Verma, A. Khatana, and S. Chaudhary, "A Comparative Study of Black Box Testing and White Box Testing," *Int. J. Comput. Sci. Eng.*, vol. 5, no. 12, pp. 301–304, 2017, doi: 10.26438/ijcse/v5i12.301304.
- [25] J. R. Lewis and J. Sauro, "Item Benchmarks for the System Usability Scale," *J. Usability Stud.*, vol. 13, no. 3, pp. 158–167, 2018.
- [26] U. Ependi, T. B. Kurniawan, and F. Panjaitan, "System Usability Scale VS Heuristic Evaluation: A Review," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 10, no. 1, pp. 65–74, 2019, doi: 10.24176/simet.v10i1.2725.