

**IMPLEMENTASI SERVICE ORIENTED ARCHITECTURE
(SOA) DENGAN RESTFUL API PADA SISTEM JARINGAN
ARSIP ISTIMEWA HANTARAN DATA ANTAR
LINI (SiJARI HanDAL)**



Disusun Oleh:

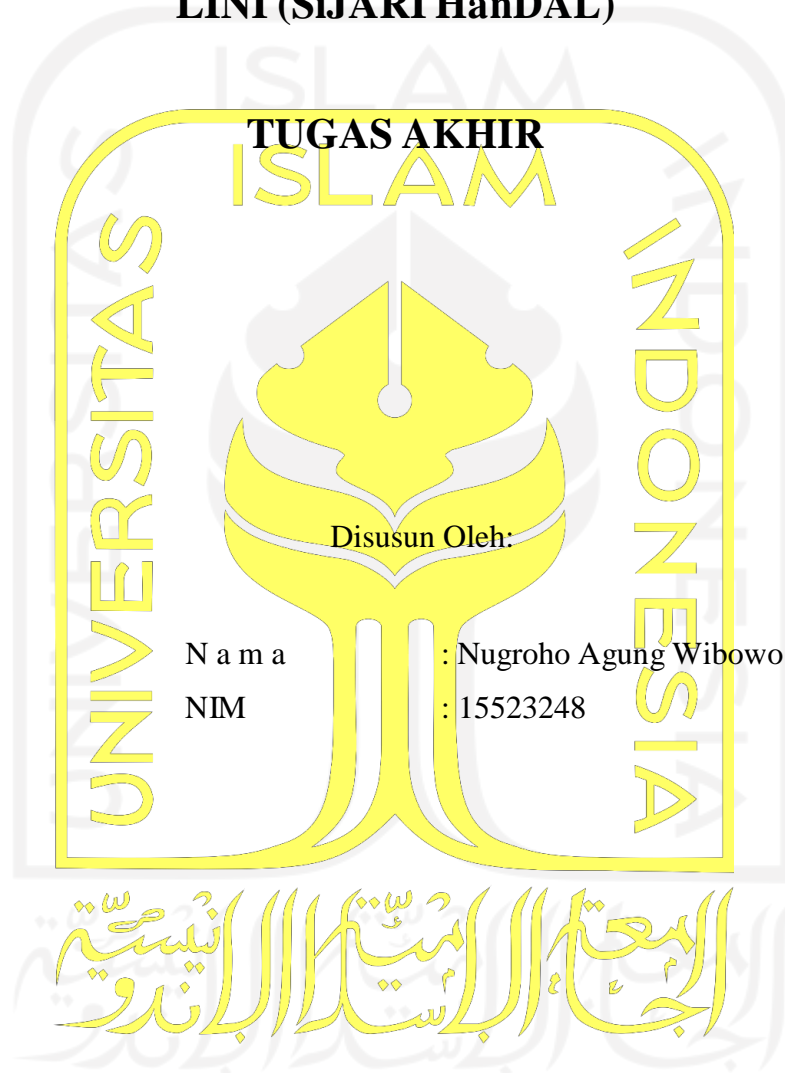
N a m a : Nugroho Agung Wibowo
NIM : 15523248

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM
INDONESIA**

2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**IMPLEMENTASI SERVICE ORIENTED ARCHITECTURE
(SOA) DENGAN RESTFUL API PADA SISTEM JARINGAN
ARSIP ISTIMEWA HANTARAN DATA ANTAR
LINI (SiJARI HanDAL)**



Yogyakarta, 22 Juli 2022

Pembimbing,

(Zainudin Zukhri, S.T, M.IT)

HALAMAN PENGESAHAN DOSEN PENGUJI

**IMPLEMENTASI SERVICE ORIENTED ARCHITECTURE
(SOA) DENGAN RESTFUL API PADA SISTEM JARINGAN
ARSIP ISTIMEWA HANTARAN DATA ANTAR**

LINI (SiJARI HanDAL)

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 22 Juli 2022

Tim Penguji

Zainudin Zukhri, S.T, M.IT

Anggota 1

Elyza Gustri Wahyuni, S.T., M.CS.

Anggota 2

Rian Adam Rajagede, S.KOM., M.CS.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nugroho Agung Wibowo

NIM : 15523248

Tugas akhir dengan judul:

**IMPLEMENTASI SERVICE ORIENTED ARCHITECTURE
(SOA) DENGAN RESTFUL API PADA SISTEM JARINGAN
ARSIP ISTIMEWA HANTARAN DATA ANTAR
LINI (SiJARI HanDAL)**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 Juli 2022



(Nugroho Agung Wibowo)

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil 'alamin, segala puji bagi Allah SWT atas segala nikmat, rahmat dan hidayah-Nya. Hanya karena nikmat dan karunia-Nya, penulis dapat menyelesaikan Tugas Akhir. Tugas Akhir ini penulis persembahkan kepada:

Kedua orang tua tercinta

Bapak Rahmat Santoso dan Ibu Ayeng Niawati

Yang telah membimbing dan mengajarkan ilmu sejak kecil, selalu mendoakan, memberikan nasehat dan semangat serta motivasi demi kelancaran semuanya. Semoga dengan prestasi kecil ini dapat membanggakan kedua orang tua tercinta. *Aamiin.*

Adik-adik tercinta,

Henny Sari Shania, Nurani Tri Hapsari, dan Nugroho Putra Raharjo,

Yang selama ini selalu menemani dan selalu memberikan semangat. Semoga tujuan kita membahagiakan kedua orang tua dapat terwujud. *Aamin.*

HALAMAN MOTO

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesudah kesulitan itu ada kemudahan”

(Q.S. Al Insyirah: 5-6)

“... janganlah kamu berputus asa dari rahmat Allah ...”

(Q.S. Az-Zumar: 53)

“Sesungguhnya Allah tidak akan mengubah suatu kaum, sebelum mereka mengubah keadaan mereka sendiri.”

(Q.S. Ar-Ra’d: 11)

“Hiduplah seakan-akan kamu mati esok hari dan belajarlh seolah kamu akan hidup selamanya.”

(M Gandhi)

الجمعة الإسلامية الأندلسية

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh.

Alhamdulillahirobbil'alamin, puji syukur atas kehadiran Allah swt yang telah melimpahkan rahmat dan hidayah-Nya kepada kita semua. Sholawat serta salam selalu tercurahkan kepada junjungan Nabi Muhammad saw. beserta keluarga dan sahabatnya sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul "Implementasi Service Oriented Architecture (SOA) dengan RESTful API Pada Sistem Jaringan Arsip Istimewa Hantaran Data Antar Lini (SiJARI HanDAL)" dengan baik dan lancar.

Laporan Tugas Akhir ini disusun sebagai salah satu syarat yang harus dipenuhi untuk menyelesaikan pendidikan pada jenjang Strata 1 (S1) Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penulis menyadari bahwa dalam penyusunan dan penulisan laporan ini tidak terlepas dari dukungan, bantuan serta bimbingan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Fathul Wahid, S.T., M.SC., Ph. D., selaku Rektor Universitas Islam Indonesia.
2. Bapak Prof., Dr., Ir. Hari Purnomo, M.T., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Hendrik, S.T., M.Eng., selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Zainudin Zukhri, S.T., M.IT., selaku Dosen Pembimbing Tugas Akhir.
6. Bapak dan Ibu Dosen Jurusan Informatika yang telah memberikan ilmunya kepada penulis.
7. Kedua orang tua tercinta Bapak Rahmat Santoso dan Ibu Ayeng Niawati yang senantiasa memberikan doa, dukungan, kasih sayang, serta nasihat yang tidak pernah putus kepada penulis.
8. Budhe, Pakdhe, Om, Tante, mas dan mbak yang selalu mendoakan dan membantu dalam banyak hal.
9. Bapak Burhan dan seluruh pihak Dinas Perpustakaan dan Arsip Daerah (DPAD) Daerah Istimewa Yogyakarta.
10. Dwi Ilham Prabowo, Dany Ryanto dan Sukron Nurtado yang selalu membantu selama masa perkuliahan dan selalu memberikan tumpangan di kosnya.

11. Widi Setyo, Agung Dwi Prabowo, Noer Ramadhan yang telah membantu dalam mengerjakan penulisan laporan tugas akhir ini.
12. Adrian Mulyawan, Rizky Ari Pratama beserta istri dan teman-teman “JKL Squad” yang telah banyak membantu dalam mengerjakan tugas akhir ini
13. Teman-teman Informatika angkatan 2015 “METAMORF” atas segala kerjasama dan semangatnya.
14. Semua pihak yang dicintai dan disayangi penulis, yang tidak dapat disebutkan satu persatu, terima kasih atas segala bantuan dan doanya.

Penulis menyadari bahwa dalam Tugas Akhir ini tidaklah sempurna dan tidak terlepas dari banyaknya kekurangan karena keterbatasan kemampuan dan pengalaman penulis. Oleh sebab itu, penulis mengharapkan kritik dan saran yang membangun sebagai bahan evaluasi agar di kemudian hari lebih baik lagi. Semoga Tugas Akhir ini bermanfaat bagi orang-orang yang menggunakannya. Aamiin.

Wassalamu'alaikum warahmatullahi wabarakatuh

Yogyakarta, 22 Juli 2022

(Nugroho Agung Wibowo)

SARI

Arsip sebagai sumber informasi merupakan salah satu bidang yang harus dipersiapkan sebaik-baiknya untuk dapat dilayanan kepada masyarakat sehingga diperlukan sistem pengelolaan yang tepat sesuai dengan perkembangan teknologi. Dinas Perpustakaan dan Arsip Daerah membutuhkan sistem yang mempermudah dan mempercepat pengguna dalam memperoleh data dari berbagai sumber secara terpadu. Berdasarkan hal tersebut, rumusan masalah dalam penelitian ini adalah bagaimana cara mengembangkan sistem yang memudahkan dalam pencarian arsip di Dinas Perpustakaan dan Arsip Daerah (DPAD) Daerah Istimewa Yogyakarta DIY dan bagaimana penerapan metode *prototyping* dalam pengembangan SiJARI HanDAL.

Tujuan dari penelitian ini adalah untuk menciptakan sistem layanan informasi kearsipan yang terpadu, efektif dan efisien. SiJARI HanDAL merupakan sebuah *website* yang menampilkan informasi arsip maupun buku berdasarkan kata kunci yang dimasukan, pengguna juga dapat memberikan *feedback* yang nantinya dapat menjadi masukan bagi Dinas untuk meningkatkan kualitas dalam melakukan pelayanan.

Metode penelitian yang digunakan adalah metode *prototyping*. Metode *prototyping* meliputi tahap Communication, Quick plan dan Modelling Quick design, Constraction of prototype, Deployment Delivery dan Feedback. SiJARI HanDAL dibuat dengan metode *RESTful API* dan untuk pengumpulan data digunakan dengan metode *scraping website* pada *web* SIKS dan *Jogjalib.com* milik DPAD DIY.

Hasil penelitian menunjukkan bahwa Penggunaan *SOA* dan *RESTful API* pada SiJARI HanDAL akan mempermudah pencarian data-data arsip dan buku milik Dinas Perpustakaan dan Arsip Daerah (DPAD) Daerah Istimewa Yogyakarta. Kelebihan penggunaan *SOA* dan *RESTful API* dapat mempermudah pertukaran data dari berbagai macam bahasa pemrograman. *REST API* tidak boros dalam penggunaan bandwidth dan hemat waktu dalam proses pertukaran data sehingga lebih efektif dan efisien. Penelitian ini juga menunjukkan bahwa metode *prototyping* dapat mempermudah pengembangan sistem karena dapat dilakukan perbaikan secara terus menerus sesuai dengan keinginan pengguna. Penelitian ini diharapkan dapat menjadi bahan pengembangan lebih lanjut untuk membuat sistem pencarian informasi mengenai arsip yang lebih baik.

Kata kunci: *Service Oriented Architecture (SOA)*, *RESTful API*, *prototyping*, *scraping*.

GLOSARIUM

<i>Prototyping</i>	Metode yang digunakan untuk mengembangkan sebuah perangkat lunak.
<i>Unified Modeling Language</i>	Bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.
<i>Laravel</i>	Kerangka kerja bahasa pemrograman PHP



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metode Penelitian	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Teknologi API.....	5
2.2 Service Oriented Architecture.....	5
2.3 Web Scraping.....	5
2.4 Entity Relationship Diagram (ERD).....	6
2.5 Unified Modeling Language (UML).....	6
2.6 Metode Pengujian	8
BAB III METODOLOGI PENELITIAN	9
3.1 Lokasi Penelitian.....	9
3.2 Metode Penelitian	9
3.3 Analisis.....	11
3.4 Unified Modelling Diagram.....	12
3.5 Perancangan Database.....	17

3.6	Perancangan Antarmuka	21
3.7	Perancangan RESTFul API.....	28
BAB IV HASIL DAN PEMBAHASAN.....		32
4.1	Hasil	32
4.2	Pengujian Sistem.....	39
BAB V PENUTUP		61
5.1	Kesimpulan	61
5.2	Saran.....	61
DAFTAR PUSTAKA.....		62
LAMPIRAN		64



DAFTAR TABEL

Tabel 2.1 Simbol pada <i>Use Case Diagram</i>	6
Tabel 2.2 Simbol pada <i>Activity Diagram</i>	7
Tabel 3.1 Tabel users	18
Tabel 3.2 Tabel items.....	18
Tabel 3.3 Tabel buku	19
Tabel 3.4 Tabel feedbackarsips.....	20
Tabel 3.5 Tabel feedbackbukus	20
Tabel 3.6 Tabel perpustakaan.....	20
Tabel 3.7 Tabel <i>Endpoint API service</i> Buku	29
Tabel 3.8 Tabel <i>Endpoint API service</i> Arsip	30
Tabel 3.9 Tabel <i>Endpoint API service</i> FeedbackArsip.....	30
Tabel 3.10 Tabel <i>Endpoint API service</i> FeedbackBuku.....	30
Tabel 3.11 Tabel <i>Endpoint API service</i> User	31
Tabel 4.1 Tabel Daftar Pengujian <i>Service</i> Arsip	47
Tabel 4.2 Tabel Daftar Pengujian <i>Service</i> Buku.....	53
Tabel 4.3 Tabel Daftar Pengujian <i>Service</i> Feedback.....	57
Tabel 4.4 Tabel Daftar Pengujian <i>Service</i> User.....	59

DAFTAR GAMBAR

Gambar 3.1 Lokasi Penelitian.....	9
Gambar 3.2 Metode Penelitian	10
Gambar 3.3 Proses <i>Scraping</i>	11
Gambar 3.4 <i>Use Case Diagram</i>	12
Gambar 3.5 <i>Activity Diagram</i> Mencari Arsip.....	13
Gambar 3.6 <i>Activity Diagram</i> Mencari Buku	14
Gambar 3.7 <i>Activity Diagram Feedback</i> Arsip.....	15
Gambar 3.8 <i>Activity Diagram Feedback</i> Buku.....	16
Gambar 3.9 <i>Entity Relationship Diagram</i> SiJARI HanDAL.....	17
Gambar 3.10 Rancangan Halaman <i>Login</i>	22
Gambar 3.11 Rancangan Halaman Register	23
Gambar 3.12 Rancangan Halaman Home.....	23
Gambar 3.13 Rancangan Halaman Hasil Pencarian	24
Gambar 3.14 Rancangan Halaman Detail Arsip.....	25
Gambar 3.15 Rancangan Halaman Detail Buku	26
Gambar 3.16 Rancangan Halaman Mitra.....	26
Gambar 3.17 Rancangan Halaman About	27
Gambar 3.18 Rancangan Halaman Pencarian Arsip Lanjutan.....	27
Gambar 3.19 Rancangan Halaman Pencarian Buku Lanjutan.....	28
Gambar 3.20 Arsitektur SiJARI HanDAL.....	29
Gambar 4.1 Halaman Login.....	32
Gambar 4.2 Halaman Register	33
Gambar 4.3 Halaman Home	34
Gambar 4.4 Halaman Hasil Pencarian	35
Gambar 4.5 Halaman Detail Arsip.....	36
Gambar 4.6 Halaman Detail Buku.....	36
Gambar 4.7 Halaman Pencarian Arsip Lanjutan	37
Gambar 4.8 Halaman Pencarian Buku Lanjutan.....	37
Gambar 4.9 Halaman Mitra	38
Gambar 4.10 Halaman About	39
Gambar 4.11 Membuat <i>File Unit Testing</i>	40
Gambar 4.12 Perintah Menjalankan <i>Unit Testing</i>	40

Gambar 4.13 <i>Source Code Unit Testing</i> Arsip	41
Gambar 4.14 <i>Source Code Unit Testing</i> Buku.....	41
Gambar 4.15 <i>Source Code Unit Testing Feedback</i>	42
Gambar 4.16 <i>Source Code Unit Testing User</i>	42
Gambar 4.17 Hasil <i>Unit Testing</i>	43
Gambar 4.18 <i>Testing Endpoint API list</i> arsip	44
Gambar 4.19 <i>Testing Endpoint API</i> cari arsip	45
Gambar 4.20 <i>Testing Endpoint API</i> search arsip.....	45
Gambar 4.21 <i>Testing Endpoint API</i> detail arsip	46
Gambar 4.22 <i>Testing Endpoint API</i> tahun arsip	46
Gambar 4.23 <i>Testing Endpoint API</i> totalcari arsip	47
Gambar 4.24 <i>Testing Endpoint API</i> author buku	48
Gambar 4.25 <i>Testing Endpoint API</i> cari buku	49
Gambar 4.26 <i>Testing Endpoint API</i> detail buku	49
Gambar 4.27 <i>Testing Endpoint API</i> koleksi buku	50
Gambar 4.28 <i>Testing Endpoint API</i> list buku	50
Gambar 4.29 <i>Testing Endpoint API</i> publisher buku.....	51
Gambar 4.30 <i>Testing Endpoint API</i> search buku	51
Gambar 4.31 <i>Testing Endpoint API</i> tahun buku	52
Gambar 4.32 <i>Testing Endpoint API</i> topik buku.....	52
Gambar 4.33 <i>Testing Endpoint API</i> totalcari buku	53
Gambar 4.34 <i>Testing Endpoint API</i> list feedback arsip	55
Gambar 4.35 <i>Testing Endpoint API</i> submit feedback arsip	55
Gambar 4.36 <i>Testing Endpoint API</i> list feedback buku.....	56
Gambar 4.37 <i>Testing Endpoint API</i> submit feedback buku	56
Gambar 4.38 <i>Testing Endpoint API</i> dologin user.....	56
Gambar 4.39 <i>Testing Endpoint API</i> email user	56
Gambar 4.40 <i>Testing Endpoint API</i> store email user	56
Gambar 4.41 <i>Test</i> tanpa API fitur pencarian.....	56
Gambar 4.21 <i>Test API</i> fitur pencarian	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Arsip sebagai sumber informasi merupakan salah satu bidang yang harus dipersiapkan sebaik-baiknya untuk dapat dilayanan kepada masyarakat. Arsip sebagai salah satu sumber informasi primer menjadi dokumen yang memiliki nilai autentisitas yang paling dapat diandalkan karena di dalam arsip memberikan data sesuai dengan saat arsip tersebut diciptakan. Oleh karena itu arsip yang memiliki nilai guna sekunder atau secara lazim disebut arsip statis harus dipelihara untuk dimanfaatkan oleh para pengguna. Di sisi lain, pelayanan publik menjadi isu yang strategis yang memiliki implikasi luas pada berbagai aspek kehidupan masyarakat. Pelayanan publik yang berkualitas merupakan salah satu indikator terjadinya perubahan penyelenggaraan pemerintahan yang berpihak pada peningkatan kesejahteraan masyarakat.

Kemunculan Internet di era teknologi yang sangat berkembang membuat lembaga arsip daerah mulai gencar melakukan pengembangan arsip digital. Internet memberikan sebuah jalan yang mudah untuk mendistribusikan informasi kepada masyarakat, tanpa harus melalui proses yang panjang. Melalui jaringan Internet, segala bentuk sumber informasi yang dikelola oleh Dinas Perpustakaan Arsip Daerah dapat diakses oleh masyarakat, tanpa dibatasi ruang dan waktu. Agar masyarakat mengetahui informasi dan layanan yang disediakan oleh Dinas Perpustakaan dan Arsip Daerah Daerah Istimewa Yogyakarta maka perlu ada media yang digunakan untuk mengenalkan layanan dan konten informasi, salah satunya adalah *website* yang menghubungkan beberapa arsip dan buku yang akan dicari secara cepat.

Hal yang menyebabkan proses pengelolaan dan pencarian arsip maupun buku mengalami kesulitan serta informasi yang dihasilkan kurang cepat dan tepat adalah *website* yang terlalu banyak dan kurang berkembang. Palsanya untuk mencari informasi arsip perlu membuka banyak *website* dan terkadang banyak *website* tidak dapat dibuka. Hal tersebut menyebabkan terbatasnya ruang akses bagi masyarakat untuk mendapatkan informasi yang lebih cepat dan lengkap mengenai arsip maupun sumber-sumber yang mendukung arsip tersebut seperti buku.

Untuk memenuhi kebutuhan pengguna yang membutuhkan data selengkap-lengkapnyanya, sebaiknya tidak bersumber dari satu dokumen saja tetapi dari banyak sumber yang saling melengkapi dengan cara mudah dan cepat. Oleh karena itu diperlukan suatu sistem yang memungkinkan pengguna untuk memperoleh data dari berbagai sumber secara terpadu. SiJARI

HanDAL merupakan konsep yang diharapkan memenuhi keinginan tersebut. SiJARI HanDAL merupakan sebuah *website* yang diharapkan dapat menampilkan informasi arsip maupun buku berdasarkan kata kunci yang dimasukan, pengguna juga dapat memberikan *feedback* yang nantinya dapat menjadi masukan bagi Dinas untuk meningkatkan kualitas dalam melakukan pelayanan.

Penelitian ini menitik beratkan pada pengintegrasian beberapa modul yaitu modul arsip, modul perpustakaan menjadi sebuah sistem informasi yang terpadu. Pengintegrasian sistem ini menggunakan pendekatan *Service Oriented Architecture* (SOA). SOA dipilih dikarenakan SOA merupakan pendekatan terbaik dalam konteks integrasi sistem informasi (Warkim & Sensuse, 2017). SOA adalah suatu metode perancangan aplikasi yang membagi komponen- komponen atau layanan yang sudah ada tanpa harus merubah sistem dengan menambahkan modul-modul yang digunakan untuk mengintegrasikan sistem. Salah satu bentuk implementasi SOA adalah pertukaran data dapat terjadi pada platform yang berbeda dengan teknologi *web service* (Slameto, 2015).

1.2 Rumusan Masalah

Berdasarkan masalah yang terdapat pada latar belakang penelitian ini, maka dapat diketahui bahwa rumusan masalah dari penelitian ini adalah

- a. Bagaimana cara mengembangkan sistem yang memudahkan dalam pencarian arsip di Dinas Perpustakaan dan Arsip Daerah (DPAD) Daerah Istimewa Yogyakarta DIY?
- b. Bagaimana cara mengembangkan sistem dengan metode *SOA* dalam pengembangan SiJARI HanDAL?

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini diantaranya yaitu:

- a. SiJARI HanDAL terdiri dari modul Arsip dan modul Buku.
- c. Subjek penelitian dari sistem ini adalah Dinas Perpustakaan dan Arsip Daerah (DPAD).
- d. Sistem integrasi yang dibangun hanya berbasis *web*.
- e. Sistem dibangun dengan menggunakan *database* sintetis atau buatan.
- f. Pengumpulan data dilakukan secara *scraping web* pada *website* SIKS dan Jogjalib.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah menciptakan sistem informasi kearsipan yang terpadu, efektif dan efisien serta mengetahui penerapan metode *prototyping* dalam pengembangan SiJARI HanDAL.

1.5 Manfaat Penelitian

Adapun manfaat penelitian ini sebagai berikut:

- a. Mempermudah pengguna dalam mencari informasi tentang arsip dan sumber lain seperti buku.
- b. Diharapkan dapat meningkatkan efisiensi, efektivitas dan fleksibilitas dinas dalam melakukan pengembangan selanjutnya.
- c. Tidak perlu mengulangi pembuatan fitur yang sudah ada, tetapi dapat menggunakan layanan yang sudah dibuat.
- d. Bagi penulis sendiri diharapkan dapat menambah ilmu serta wawasan.

1.6 Metode Penelitian

Metode penelitian yang diterapkan pada pembuatan sistem SiJARI HanDAL ini adalah dengan metode *prototyping*. Adapun tahapan dari metode *prototyping* sebagai berikut:

- a. **Communication** : Pengembang & pengguna bersama-sama mendefinisikan dan mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat (diperoleh dari wawancara, diskusi, dan survei langsung).
- b. **Quick Plan & Modelling Quick Design** : Setelah kebutuhan, data, informasi, dan garis besar sistem telah diperoleh, selanjutnya pada tahap ini dilakukan proses perancangan sistem. Kemudian dilakukan pemodelan untuk merancang desain sistem.
- c. **Construction of Prototype** : Pada tahap ini dilakukan pembangunan sistem atau prototype.
- d. **Development, Delivery, Feedback** : Sistem yang telah selesai kemudian diserahkan kepada para pengguna untuk dievaluasi dan memberikan umpan balik yang akan digunakan untuk perbaikan dari sistem yang telah dibuat.

1.7 Sistematika Penulisan

Sistematika penulisan ditujukan untuk memudahkan dalam melakukan pembahasan tugas akhir ini. Secara garis besar, sistematika penulisan ini terbagi menjadi beberapa bab. Adapun uraian dari masing-masing bab tersebut sebagai berikut:

Bab I Pendahuluan, bab ini berisi uraian tentang latar belakang studi kasus yang diangkat, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan yang menggambarkan struktur penulisan laporan SiJARI HanDAL secara singkat.

Bab II Landasan Teori, bab ini berisi uraian teori yang melandasi pengembangan SiJARI HanDAL seperti penjelasan tentang *Restful API*, *Laravel*, *Service Oriented Architecture*, *Web Scraping*.

Bab III Metodologi Penelitian, bab ini berisi penjelasan metodologi yang digunakan dalam penelitian terhadap SiJARI HanDAL yang meliputi analisis kebutuhan, metode pengumpulan data, *unified modeling diagram*, perancangan *database*, dan perancangan antarmuka dari sistem.

Bab IV Hasil dan Pembahasan, bab ini berisi tentang uraian penjelasan terhadap SiJARI HanDAL yang telah dibangun secara keseluruhan dan pengujian untuk mengetahui kelebihan dan kekurangan dari sistem serta kevalidan sistem tersebut.

Bab V Kesimpulan dan Saran, bab ini berisi uraian kesimpulan yang merupakan rangkuman dari hasil penelitian SiJARI HanDAL.

BAB II LANDASAN TEORI

2.1 Teknologi API

2.1.1 Restful API

REST adalah sebuah layanan *web* yang menyediakan interoperabilitas suatu sistem di Internet yang memungkinkan sistem untuk mengakses dan memanipulasi data sesuai dengan layanan yang telah disediakan (Chen, Ji, Fan, & Zhan, 2017). *Restful API* adalah sebuah sistem terdistribusi menggunakan *web service* yang menyediakan kapabilitas di Internet dengan meng-generalisasi interface, dengan metode dokumentasi *API* agar pengembang dapat dengan mudah menggunakan fitur-fitur yang telah disediakan (Sutrisno, Assyidiq, & Santoso).

2.1.2 Laravel

Laravel adalah sebuah *framework web PHP* dengan konsep *model-view-controller (MVC)* yang dibuat oleh Taylor Otwell dan ditujukan untuk pengembangan aplikasi *web*. *Framework Laravel* juga sudah menyediakan beberapa fitur seperti, *authentication*, *routing*, dan *database migration*, dengan fitur-fitur tersebut pengembang dapat membangun sebuah aplikasi yang kompleks (Chen, Ji, Fan, & Zhan, 2017).

2.2 Service Oriented Architecture

Service Oriented Architecture (SOA) adalah sebuah pendekatan untuk memecahkan permasalahan yang besar menjadi sekumpulan kecil *service* untuk menyelesaikan masalah yang lebih spesifik (Hananta, 2013). SOA merupakan pendekatan terbaik dalam konteks integrasi sistem informasi (Warkim & Sensuse, 2017). SOA adalah suatu metode perancangan aplikasi yang membagi komponen-komponen atau layanan yang sudah ada tanpa harus merubah sistem dengan menambahkan modul-modul yang digunakan untuk mengintegrasikan sistem (Slameto, 2015). Salah satu bentuk implementasi SOA adalah pertukaran data dapat terjadi pada platform yang berbeda dengan teknologi *web service*.

2.3 Web Scraping

Web scraping adalah suatu teknik untuk mendapatkan informasi ataupun data pada suatu situs *web* secara otomatis (Satriajati, Panuntun, & Pramana, 2020). *Web scraping* dilakukan dengan tujuan untuk mencari informasi pada bagian tertentu pada suatu situs yang dituju sesuai

dengan kebutuhan (Satriajati, Panuntun, & Pramana, 2020). *Web scraping* merupakan suatu proses pengambilan data semi-terstruktur dari Internet, umumnya berupa halaman-halaman *web* dalam Bahasa markup seperti HTML (Josi, Abdillah, & Suryayusra, 2014). *Web scraping* akan diizinkan selama data yang tersedia tidak dilindungi hak cipta serta data yang diperoleh tidak digunakan untuk tujuan komersial (Flores, Permatasari, & Jasa, 2020).

2.4 Entity Relationship Diagram (ERD)


Entity Relation Diagram adalah gambaran data yang dimodelkan dalam sebuah diagram yang digunakan untuk mendokumentasikan data dengan cara menentukan apa saja yang terdapat pada setiap *entity* dan bagaimana hubungan antara *entity* satu dengan yang lainnya (Rahmayu, 2016). Sedangkan menurut Sukamto & Shalahuddin dalam (Tabrani, Suhardi, & Priyandaru, 2021), menyatakan bahwa *Entity Relationship Diagram* (ERD) merupakan sebuah diagram yang digunakan untuk permodelan basis data relational.

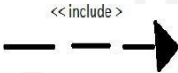
2.5 Unified Modeling Language (UML)

2.5.1 Use Case Diagram

Use case diagram merupakan diagram yang mendeskripsikan interaksi antara satu atau lebih *user* (pengguna) sebuah sistem dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Simatupang & Sianturi, 2019). Adapun simbol-simbol pada *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol pada *Use Case Diagram*



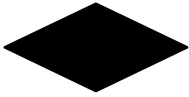

No	Simbol	Nama	Keterangan
1		Aktor	<i>User</i> (pengguna) atau sistem yang berinteraksi dengan sistem informasi yang akan dibuat.
2		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar

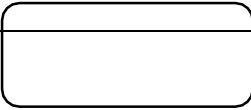
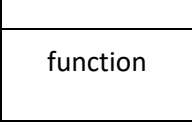


			unit atau actor.
3		Asosiasi	Penghubung antar aktor dengan <i>use case</i> .
4		<i>Include</i>	Menjelaskan bahwa <i>use case</i> memperluas perilaku dari <i>use case</i> yang diberikan.

2.5.2 Activity Diagram

Activity diagram adalah gambaran aktivitas dari sebuah sistem yang menjelaskan alur dari sebuah use case diagram yang telah dibuat. Adapun simbol-simbol *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol pada *Activity Diagram*

No	Simbol	Nama	Keterangan
1		Status Awal	Menunjukkan dimulainya suatu <i>workflow</i> pada sebuah <i>activity diagram</i> pada sebuah diagram.
2		Status Akhir	Menggambarkan akhir atau terminal dari sebuah <i>activity diagram</i>
3		Percabangan	Percabangan yang terdapat pilihan lebih dari 1
4		Penggabungan atau Join	Penggabungan yang mana lebih dari 1 aktivitas lalu digabungkan menjadi 1

5		Aktivitas	Aktivitas yang dilakukan oleh suatu sistem
6		Swimline	<i>Swimline</i> digunakan untuk memisahkan organisasi bisnis yang bertanggung jawab pada suatu sistem
			
7		Transition	<i>Transition</i> digunakan untuk menghubungkan ke aktivitas selanjutnya dari aktivitas sebelumnya

2.6 Metode Pengujian

2.6.1 Unit Testing

Unit testing adalah sebuah kode yang ditulis oleh pengembang yang digunakan untuk menguji bagian kecil dari suatu fungsionalitas kode (Kurniawan R. , 2016). Pada definisi lainnya, *unit testing* merupakan pengujian yang mengeksekusi setiap fitur untuk memastikan masing-masing fitur berfungsi sesuai dengan yang diharapkan. *Unit testing* termasuk dalam pengujian otomatis (W. Lewis, 2009).

Adapun keuntungan menggunakan *unit testing* adalah mencari error atau bug menjadi lebih mudah, *code* menjadi *reuseable*, waktu yang dibutuhkan untuk menakukan *debug* lebih singkat, *unit testing* juga dapat berfungsi sebagai dokumentasi proyek. Sedangkan kekurangan dalam menggunakan *unit testing* adalah tidak dapat menangkap setiap kesalahan dalam suatu program dan pada dasarnya *unit testing* berfokus pada unit kode sehingga tidak dapat menangkap kesalahan dalam tingkat sistem yang lebih luas (Kua, 2019).

BAB III METODOLOGI PENELITIAN

3.1 Lokasi Penelitian

Berdasarkan Peraturan Gubernur Daerah Istimewa Yogyakarta No. 70 Tahun 2018 tentang Kedudukan, Susunan Organisasi, Tugas, Fungsi, dan Tata Kerja Dinas Perpustakaan dan Arsip Daerah, (Pasal 4) Dinas mempunyai tugas membantu Gubernur melaksanakan urusan pemerintahan bidang perpustakaan dan urusan pemerintahan bidang kearsipan. Dinas Perpustakaan dan Arsip Daerah (DPAD) DIY berlokasi di Banguntapan, Bantul (Selatan Gedung Balai Layanan Perpustakaan/ Grahatama Pustaka Karangjambé Banguntapan, Bantul, Daerah Istimewa Yogyakarta). Adapun denah lokasi Dinas Perpustakaan dan Arsip Daerah (DPAD) Daerah Istimewa Yogyakarta (DIY) dapat dilihat pada Gambar 3.1.

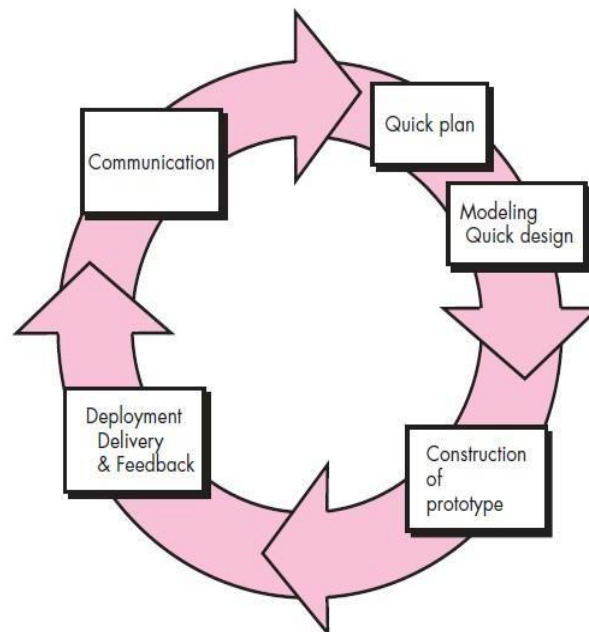


Gambar 3.1 Lokasi Penelitian

3.2 Metode Penelitian

3.2.1 Metode Prototyping

Metode penelitian yang diterapkan pada pembuatan sistem SiJARI HanDAL ini adalah dengan metode prototyping. Adapun tahapan dari metode prototyping dapat dilihat pada Gambar 3.2.



Gambar 3.2 Metode Penelitian

Sumber: (Susanto & Andriana, 2016)

- a. **Communication** : Pengembang & pengguna bersama-sama mendefinisikan dan mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat (diperoleh dari wawancara, diskusi, dan survei langsung).
- b. **Quick Plan & Modelling Quick Design** : Setelah kebutuhan, data, informasi, dan garis besar sistem telah diperoleh, selanjutnya pada tahap ini dilakukan proses perancangan sistem. Kemudian dilakukan pemodelan untuk merancang desain sistem.
- c. **Construction of Prototype** : Pada tahap ini dilakukan pembangunan sistem atau prototype.
- d. **Development, Delivery, Feedback** : Sistem yang telah selesai kemudian diserahkan kepada para pengguna untuk dievaluasi dan memberikan umpan balik yang akan digunakan untuk perbaikan dari sistem yang telah dibuat.

Metode penelitian model prototyping lebih cocok untuk sistem atau perangkat lunak yang dikembangkan bersifat *customize*, artinya sistem dikembangkan berdasarkan permintaan dan kebutuhan tertentu yang memiliki tujuan untuk mengimplementasikan sebuah metode tertentu pada suatu kasus (Susanto & Andriana, 2016).

Metode prototyping memiliki beberapa keunggulan dan kekurangan. Adapun keunggulan dari metode prototyping, antara lain sebagai berikut:

- a. Adanya komunikasi yang baik antara pengembang dan klien (user).
- b. Lebih hemat waktu dalam pengembangan sistem.
- c. Prototype dapat dengan mudah ditambah dan diubah.
- d. Klien berperan aktif dalam pengembangan sistem.

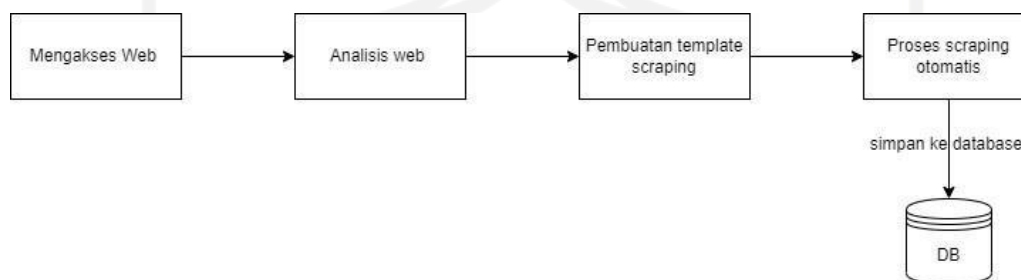
Sedangkan kekurangan dari metode prototyping adalah sebagai berikut:

- a. Klien terkadang tidak melihat atau menyadari bahwa sistem yang ada belum mencantumkan kualitas sistem secara keseluruhan dan belum memikirkan pemeliharaan untuk jangka panjang.
- b. Pengembang biasanya ingin cepat menyelesaikan proyek. Sehingga menggunakan algoritma yang tidak efisien dan sederhana untuk membuat prototype lebih cepat.
- c. Proses perancangan dan analisa terlalu singkat.

(Kurniawan T. , 2020).

3.2.2 Metode Pengumpulan Data

Berdasarkan batasan masalah yang ada, proses pengumpulan data dilakukan dengan melakukan *scraping web* pada *web* SIKS dan Jogjalib.com milik DPAD DIY. Proses *scraping* dilakukan dengan cara menganalisis *web* yang menjadi tujuan dari proses *scraping* kemudian melakukan penulisan script untuk melakukan *scraping*. Proses hasil dari melakukan *scraping* disimpan kedalam *database* untuk selanjutnya diolah kembali. Adapun gambar proses *scraping* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Proses *Scraping*

3.3 Analisis

3.3.1 Analisis Masalah

Dalam melakukan pencarian arsip maupun buku pengguna harus membuka banyak *website* yang berbeda-beda dan tak jarang pengguna kesulitan dalam membuka *website* tersebut sehingga memerlukan banyak waktu akibatnya informasi yang didapatkan kurang optimal.

Dalam perancangannya pun terdapat kendala seperti tidak mendapatkan izin untuk mengakses *database* milik Dinas Perpustakaan dan Arsip Daerah Daerah Istimewa Yogyakarta.

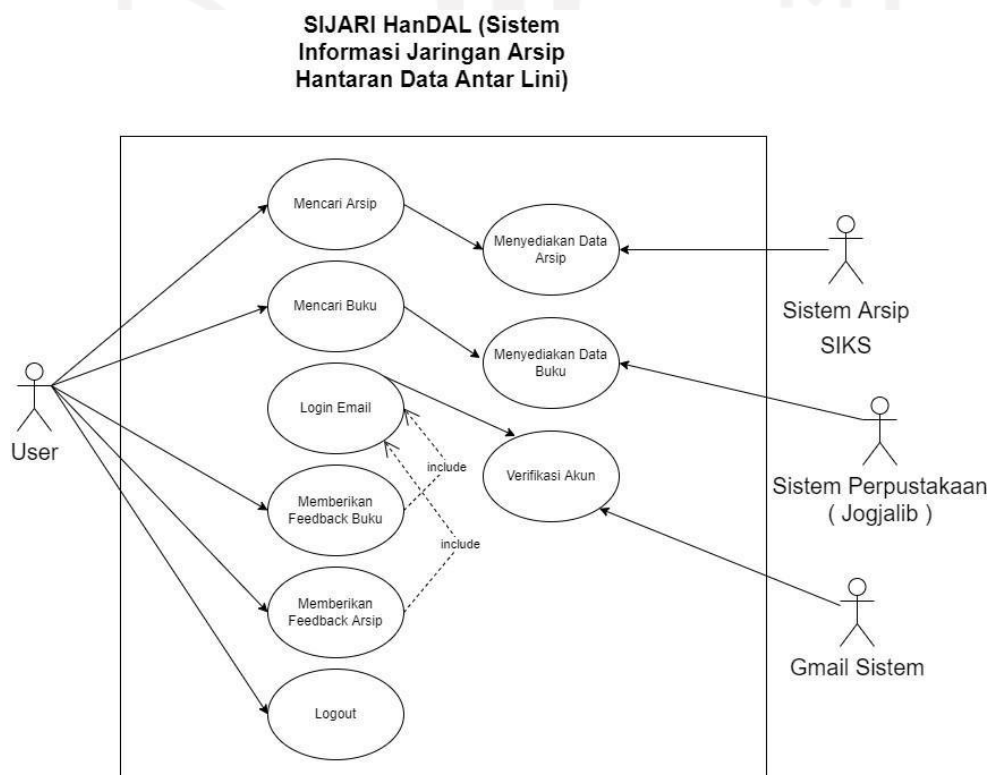
3.3.2 Usulan Penyelesaian

Berdasarkan penjabaran masalah diatas penulis mengusulkan untuk membuat SiJARI HanDAL dengan metode *Restful API* yang mana *datasenya* berupa *database* sintetis (*database* buatan) dan untuk pengumpulan datanya dilakukan dengan metode *Scraping Website* agar data yang didapat merupakan data real dari *website* yang sudah ada milik Dinas Perpustakaan dan Arsip Daerah Daerah Istimewa Yogyakarta.

3.4 Unified Modelling Diagram

3.4.1 Usecase Diagram

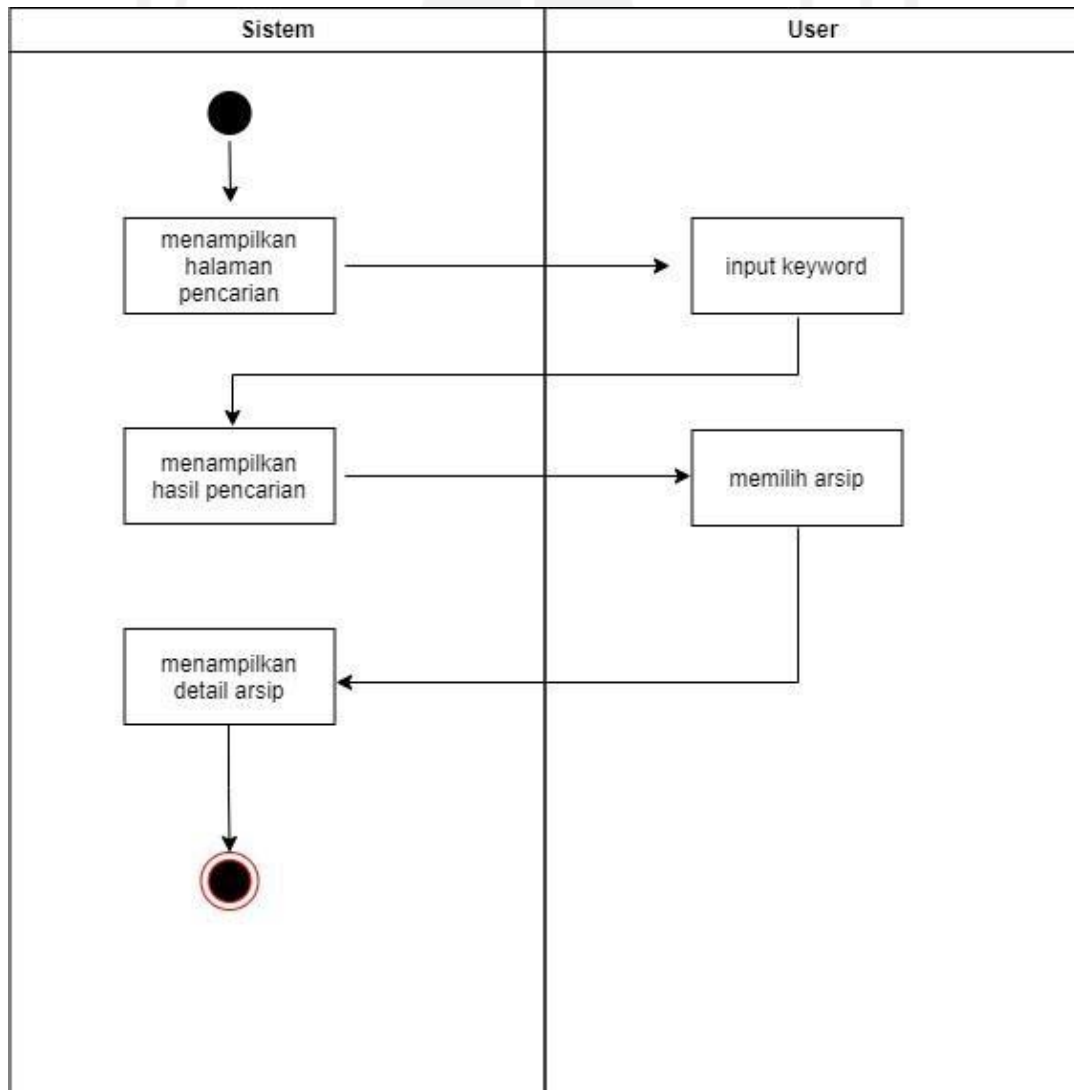
Usecase diagram adalah *diagram* yang digunakan untuk menggambarkan interaksi antara pengguna sistem dengan sistem. Adapun dalam *usecase diagram* ini memiliki satu aktor yaitu *user*. Aktor *user* diharuskan login ketika akan memberikan *feedback* untuk arsip maupun buku yang sedang dicari. *Usecase diagram* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Use Case Diagram

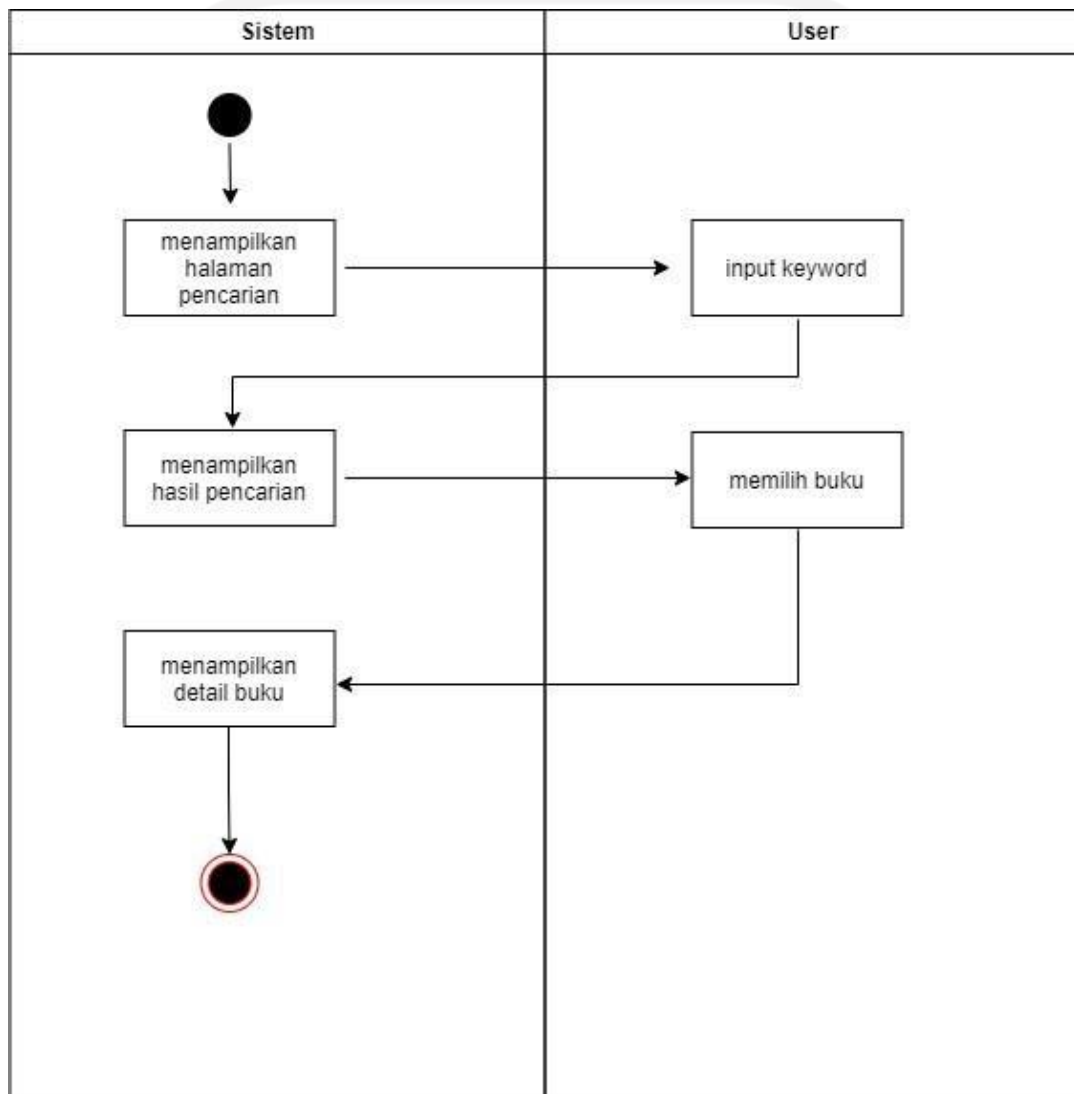
3.4.2 Activity Diagram

Activity diagram merupakan sebuah gambaran berbagai aliran aktifitas dalam sistem yang sedang dirancang, bagaimana aktifitas aliran berawal, bagaimana keputusan yang mungkin terjadi, dan bagaimana aktifitas itu berakhir. *Activity diagram* digunakan untuk menggambarkan aktifitas pada sistem secara global. Untuk mencari arsip pada sistem ini, user dapat memasukkan kata kunci dari arsip yang akan dicari, kemudian sistem akan menampilkan data-data dari kata kunci yang telah dimasukkan, setelah itu user dapat memilih arsip yang sedang dicari. Setelah memilih arsip, maka sistem akan menampilkan detail dari arsip yang dipilih. Adapun *activity diagram* dalam mencari sebuah arsip pada SiJARI HanDAL dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity Diagram Mencari Arsip

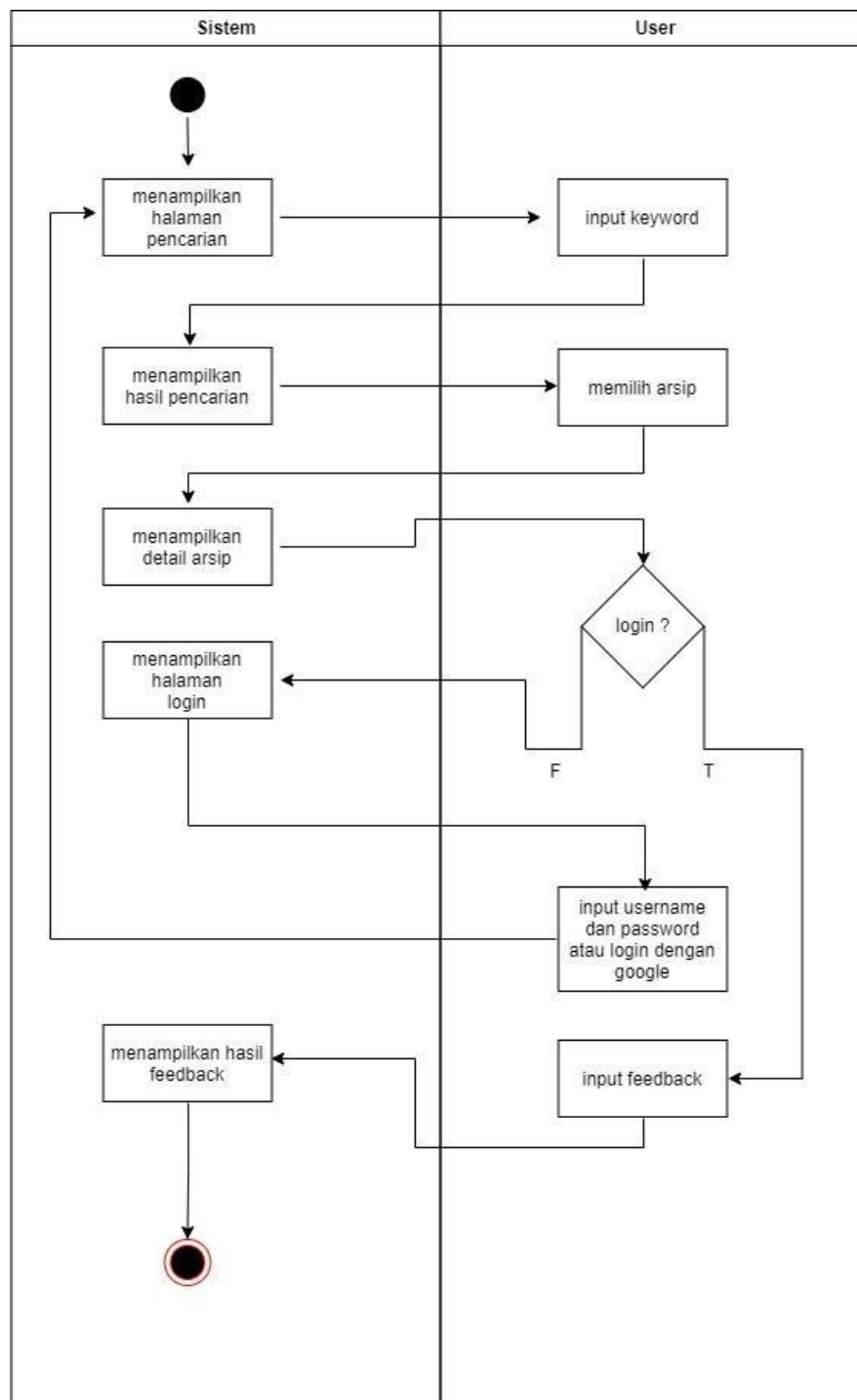
Adapun untuk untuk mencari buku pada sistem ini, user dapat memasukkan kata kunci dari buku yang akan dicari, kemudian sistem akan menampilkan data-data dari kata kunci yang telah dimasukkan, setelah itu user dapat memilih buku yang sedang dicari. Setelah memilih buku, maka sistem akan menampilkan detail dari buku yang dipilih. *Activity diagram* untuk mencari buku pada SiJARI HanDAL dapat dilihat pada Gambar 3.6.



Gambar 3.6 *Activity Diagram* Mencari Buku

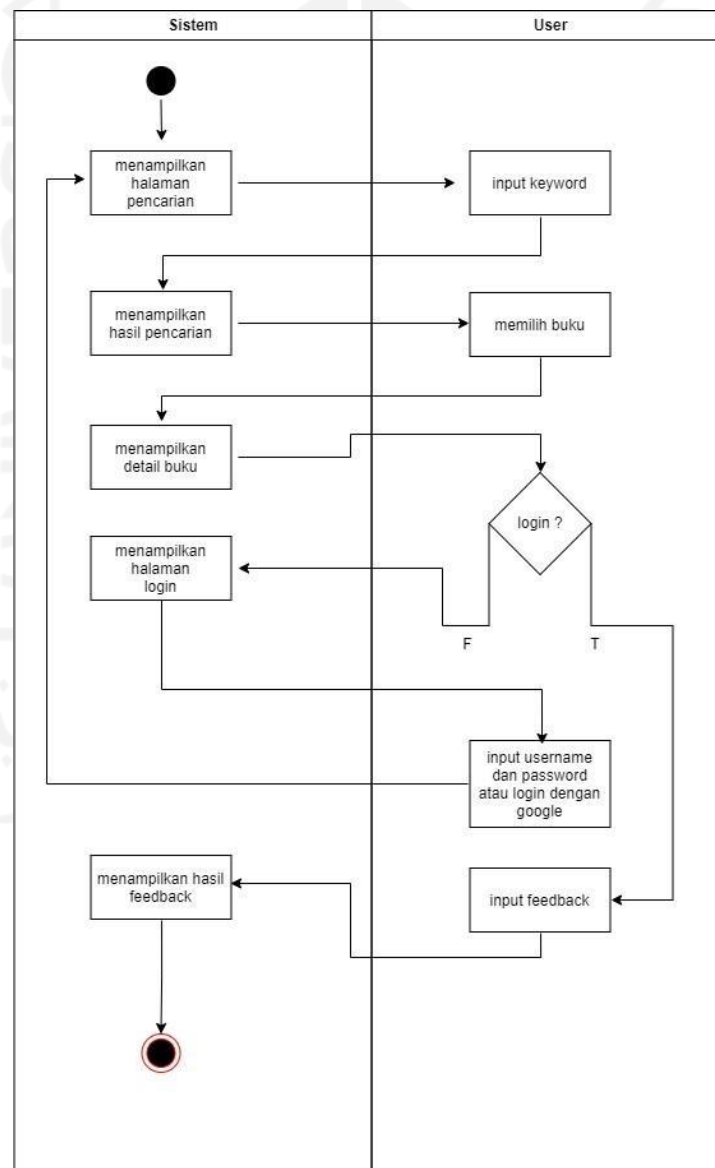
Adapun alur untuk memberikan *feedback* ataupun komentar pada arsip yang sedang dicari pada sistem ini, pertama user dapat memasukkan kata kunci dari arsip yang akan dicari, kemudian sistem akan menampilkan data dari kata kunci yang telah dimasukkan, setelah itu user dapat memilih arsip yang sedang dicari. Setelah memilih arsip, maka sistem akan menampilkan detail dari arsip yang dipilih. Kemudian user dapat memberikan *feedback* apabila

user telah melakukan *login*. Jika user belum melakukan *login*, maka sistem akan mengarahkan ke halaman login agar user melakukan login terlebih dahulu untuk dapat memberikan *feedback*. *Activity diagram* untuk memberikan *feedback* pada arsip yang sedang dicari pada SiJARI HanDAL dapat dilihat pada Gambar 3.7.



Gambar 3.7 Activity Diagram Feedback Arsip

Untuk dapat memberikan *feedback* atau komentar pada buku yang sedang dicari pada sistem ini, langkah pertama user dapat memasukkan kata kunci dari buku yang akan dicari, kemudian sistem akan menampilkan data-data dari kata kunci yang telah dimasukkan, setelah itu user dapat memilih buku yang sedang dicari. Setelah memilih buku, maka sistem akan menampilkan detail dari buku yang dipilih. Kemudian user dapat memberikan *feedback* apabila user telah melakukan login. Jika user belum melakukan login, maka sistem akan mengarahkan ke halaman login agar user melakukan login terlebih dahulu untuk dapat memberikan *feedback*. Adapun *activity diagram* untuk memberikan *feedback* pada buku yang sedang dicari pada SiJARI HanDAL dapat dilihat pada Gambar 3.8.



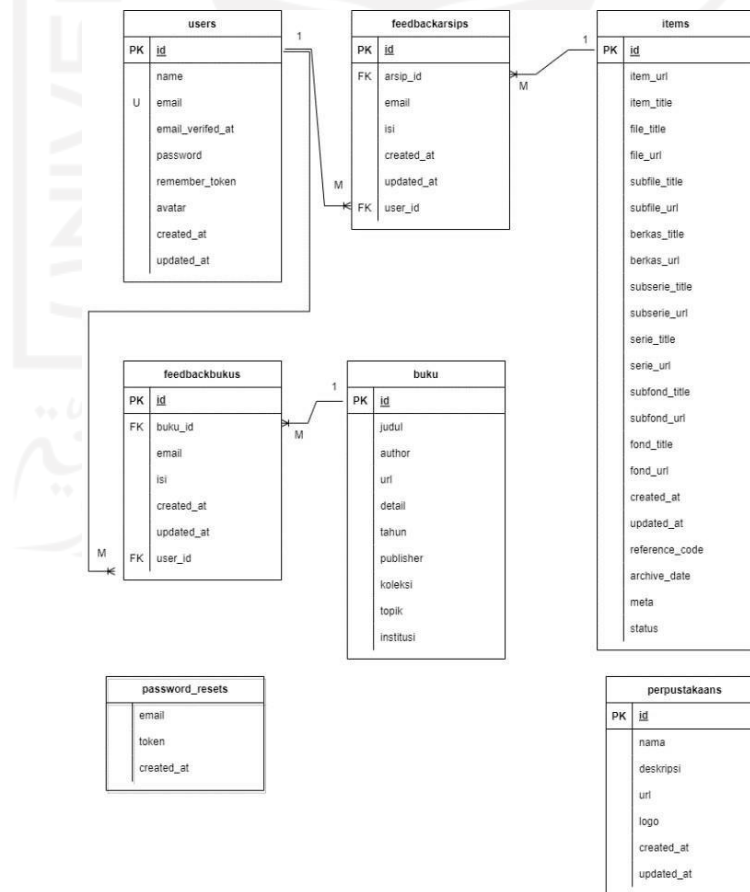
Gambar 3.8 Activity Diagram Feedback Buku

3.5 Perancangan Database

Berdasarkan batasan masalah yang ada yang disebabkan tidak mendapatkan akses ke *database* arsip dan perpustakaan, maka perlulah dibuatkan *database* imitasi untuk menampung data dari arsip dan buku yang akan ditampilkan di SiJARI HanDAL. SiJARI HanDAL memiliki *database* yang menggunakan beberapa tabel diantaranya yaitu tabel *users*, tabel *items*, tabel buku, tabel *feedbackarsips*, tabel *feedbackbukus*, tabel *perpustakaan*, dan tabel *password_resets*.

3.5.1 Entity Relationship Diagram

ERD (Entity Relationship Diagram) adalah sebuah diagram yang menjelaskan hubungan antar tabel dalam perancangan suatu basisdata. ERD bertujuan untuk menggambarkan hubungan antara beberapa basisdata. Pada penelitian ini, penulis merancang sebuah ERD yang menggambarkan rancangan basisdata pada sistem. ERD tersebut dapat dilihat pada Gambar 3.9.



Gambar 3.9 Entity Relationship Diagram SiJARI HanDAL

3.5.2 Struktur Tabel

Tabel 3.1 merupakan tabel users yang berisikan data dari user. Isi dari tabel users yaitu id, name, email, email_verified_at, password, remember_token, avatar, created_at dan updated_at. *User* disini merupakan pengguna dari SiJARI HanDAL. Tabel ini memiliki relasi 1:M (*One to Many*) terhadap tabel feedbackarsips maupun tabel feedbackbukus.

Tabel 3.1 Tabel users

Nama Kolom	Tipe Data	Panjang	Keterangan
Id	Unsigned bigint	20	Primary key
Name	Varchar	255	
Email	Varchar	255	Unique
email_verified_at	Timestamp		Nullable
Password	Varchar	255	
remember_token	Varchar	100	Nullable
Avatar	Varchar	255	Nullable
created_at	Timestamp		Nullable
updated_at	Timestamp		Nullable

Sedangkan Tabel 3.2 merupakan tabel items yang berisikan data dari arsip. Isi dari tabel ini yaitu id, item_url, item_title, file_title, file_url, subfile_title, subfile_url, berkas_title, berkas_url, subserie_title, subserie_url, serie_title, serie_url, subfond_title, subfond_url, fond_title, fond_url, created_at, updated_at, reference_code, archive_date, meta dan status. Tabel ini memiliki relasi 1:M (*One to Many*) terhadap tabel feedbackarsips.

Tabel 3.2 Tabel items

Nama Kolom	Tipe Data	Panjang	Keterangan
id	Unsigned bigint	20	Primary key
item_url	Varchar	255	
item_title	Varchar	255	
file_title	Varchar	255	
file_url	Varchar	255	
subfile_title	Varchar	255	
subfile_url	Varchar	255	

berkas_title	Varchar	255	
berkas_url	Varchar	255	
subserie_title	Varchar	255	
subserie_url	Varchar	255	
serie_title	Varchar	255	
serie_url	Varchar	255	
subfond_title	Varchar	255	
subfond_url	Varchar	255	
fond_title	Varchar	255	
fond_url	Varchar	255	
created_at	Timestamp		Nullable
updated_at	Timestamp		Nullable
reference_code	Varchar	255	
archive_date	Varchar	255	
Meta	Text		
Status	Varchar	255	

Tabel 3.3 merupakan tabel buku yang berisikan data dari buku. Isi dari tabel ini yaitu id, judul, author, url, detail, tahun, publisher, koleksi, topik, dan institusi. Tabel ini memiliki relasi 1:M (*One to Many*) terhadap tabel feedbackbukus.

Tabel 3.3 Tabel buku

Nama Kolom	Tipe Data	Panjang	Keterangan
Id	Unsigned bigint	20	Primary key
Judul	Varchar	255	
Author	Varchar	255	
url	Varchar	255	
Detail	Text		
Tahun	year	4	
Publisher	Varchar	255	
Koleksi	Varchar	255	
Topik	Varchar	255	
Institusi	Varchar	255	

Sedangkan Tabel 3.4 merupakan tabel feedbackarsips yang berisikan data tentang *feedback* dari banyak arsip. Isi dari tabel ini yaitu id, arsip_id, isi, created_at, updated_at dan user_id. Tabel ini memiliki relasi M:1 (*Many to One*) terhadap tabel users maupun tabel items.

Tabel 3.4 Tabel feedbackarsips

Nama Kolom	Tipe Data	Panjang	Keterangan
id	Unsigned bigint	20	Primary key
arsip_id	Varchar	255	Foreign key
isi	Text		
created_at	Timestamp		Nullable
updated_at	Timestamp		Nullable
user_id	Unsigned bigint	20	Foreign key

Tabel 3.5 merupakan tabel feedbackbuku yang berisikan data tentang *feedback* dari banyak buku. Isi dari tabel ini yaitu id, buku_id, isi, created_at, updated_at, dan user_id. Tabel ini memiliki relasi M:1 (*Many to One*) terhadap tabel users dan tabel buku. Tabel 3.6 merupakan tabel perpustakaan yang berisikan data dari institusi yang telah bekerjasama dengan Dinas Perpustakaan dan Arsip Daerah Daerah Istimewa Yogyakarta. Isi dari tabel ini yaitu id, nama, deskripsi, url, logo, created_at dan updated_at.

Tabel 3.5 Tabel feedbackbuku

Nama Kolom	Tipe Data	Panjang	Keterangan
id	Unsigned bigint	20	Primary key
buku_id	Varchar	255	Foreign key
isi	Text		
created_at	Timestamp		Nullable
updated_at	Timestamp		Nullable
user_id	Unsigned bigint	20	Foreign key

Tabel 3.6 Tabel perpustakaan

Nama Kolom	Tipe Data	Panjang	Keterangan
id	Unsigned bigint	20	Primary key
Nama	Varchar	255	

Deskripsi	Text		
url	Varchar	255	
Logo	Varchar	255	
created_at	Timestamp		Nullable
updated_at	Timestamp		Nullable

3.6 Perancangan Antarmuka

Rancangan antarmuka SiJARI HanDAL adalah rancangan awal yang berupa mockup yang digunakan untuk tahap awal dalam pembuatan desain antarmuka SiJARI HanDAL. Rancangan antarmuka ini terdiri dari beberapa tampilan sebagai berikut.

3.6.1 Rancangan Halaman Login

Rancangan halaman login yang akan dikembangkan dapat dilihat pada Gambar 3.10. Untuk menggunakan SiJARI HanDAL proses *login* tidak diharuskan namun proses login diperlukan untuk proses pemberian *feedback*. Pada halaman ini pengguna harus mengisi *username* dan *password*, namun apabila pengguna belum memiliki akun maka pengguna dapat melakukan registrasi terlebih dahulu ataupun pengguna dapat melakukan login dengan akun gmail.

3.6.2 Rancangan Halaman Register

Rancangan halaman *register* yang digunakan apabila pengguna hendak menggunakan SiJARI HanDAL dapat dilihat pada Gambar 3.11. Agar dapat mengakses fitur *feedback* maka pengguna diwajibkan memiliki akun terlebih dahulu.

3.6.3 Rancangan Halaman Home

Rancangan halaman home yang digunakan apabila pengguna hendak mencari arsip ataupun buku dapat dilihat pada Gambar 3.12. Halaman ini adalah halaman utama ketika membuka sistem.

3.6.4 Rancangan Halaman Hasil Pencarian

Rancangan halaman hasil pencarian yang digunakan pengguna untuk mencari arsip maupun buku yang sedang dicari dapat dilihat pada Gambar 3.13.

Gambar 3.10 Rancangan Halaman *Login*

3.6.5 Rancangan Halaman Detail Arsip

Rancangan halaman detail arsip yang digunakan pengguna untuk melihat arsip lebih detail dapat dilihat pada Gambar 3.14. Pada halaman ini juga terdapat kolom untuk memberikan *feedback* terhadap arsip yang sedang dicari.

3.6.6 Rancangan Halaman Detail Buku

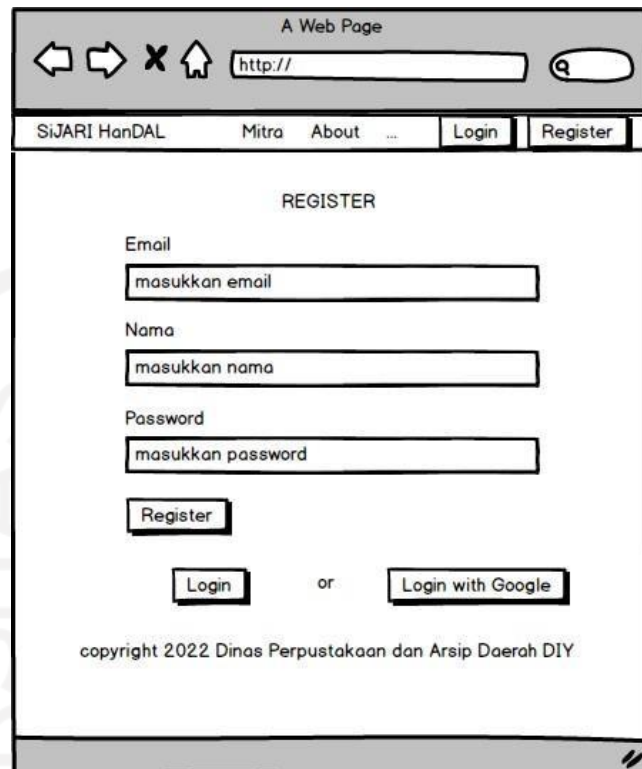
Rancangan halaman detail buku yang digunakan pengguna untuk melihat informasi buku lebih detail dapat dilihat pada Gambar 3.15. Pada halaman ini sama seperti halaman detail arsip, juga terdapat kolom untuk memberikan *feedback* terhadap buku yang sedang dicari.

3.6.7 Rancangan Halaman Mitra

Rancangan halaman mitra dapat dilihat pada Gambar 3.16. Pada halaman ini memberikan informasi institusi mana saja yang telah menjalani kerjasama dengan Dinas Perpustakaan dan Arsip Daerah Daerah Istimewa Yogyakarta.

3.6.8 Rancangan Halaman About

Rancangan halaman about yang nantinya berisikan informasi umum tentang SiJARI HanDAL dapat dilihat pada Gambar 3.17.



A Web Page

SiJARI HanDAL Mitra About ... Login Register

REGISTER

Email
masukkan email

Nama
masukkan nama

Password
masukkan password

Register

Login or Login with Google

copyright 2022 Dinas Perpustakaan dan Arsip Daerah DIY

Gambar 3.11 Rancangan Halaman Register



A Web Page

SiJARI HanDAL Mitra About ... Login Register

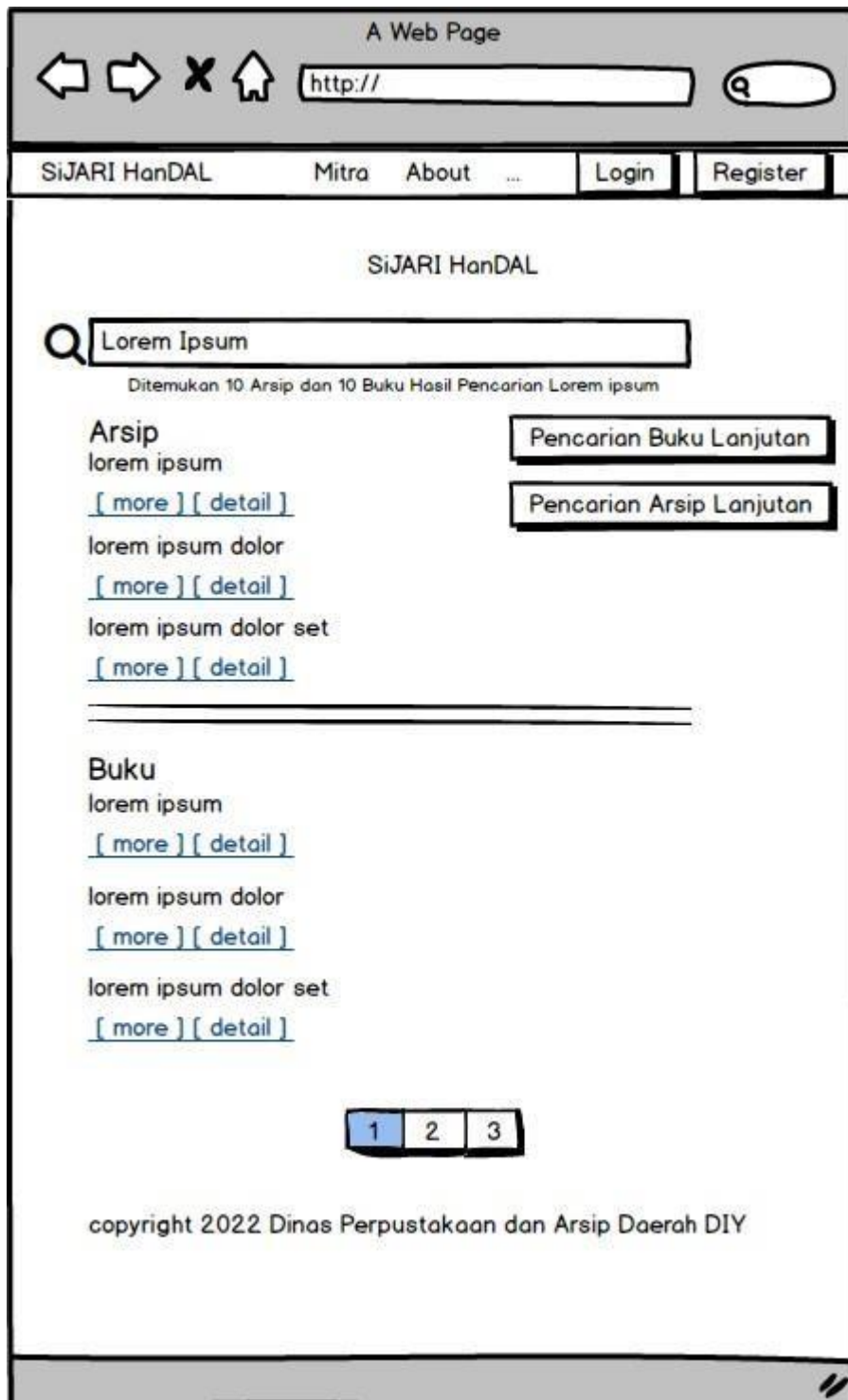
SiJARI HanDAL

Q Cari Arsip dan Buku

Pencarian Buku Lanjutan Pencarian Arsip Lanjutan

copyright 2022 Dinas Perpustakaan dan Arsip Daerah DIY

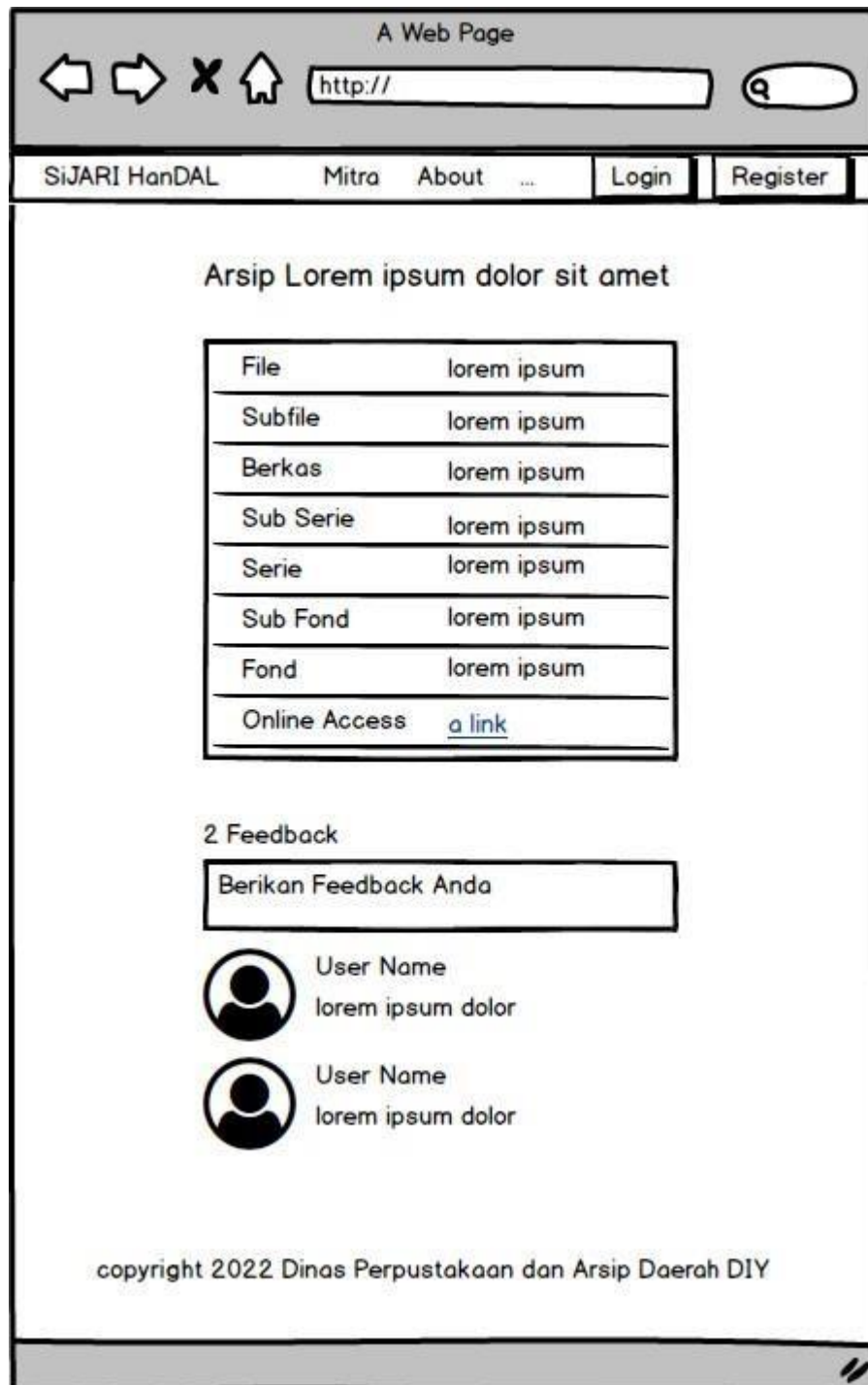
Gambar 3.12 Rancangan Halaman Home



Gambar 3.13 Rancangan Halaman Hasil Pencarian

3.6.9 Rancangan Halaman Pencarian Arsip Lanjutan

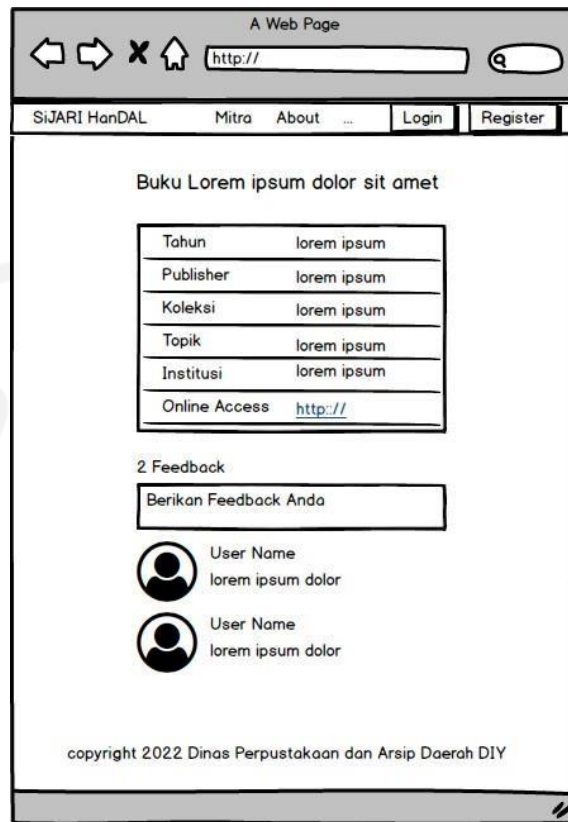
Rancangan halaman pencarian arsip lanjutan dapat dilihat pada Gambar 3.18. Pada halaman ini terdapat filter tahun untuk mencari arsip berdasarkan pada tahun yang dicari.



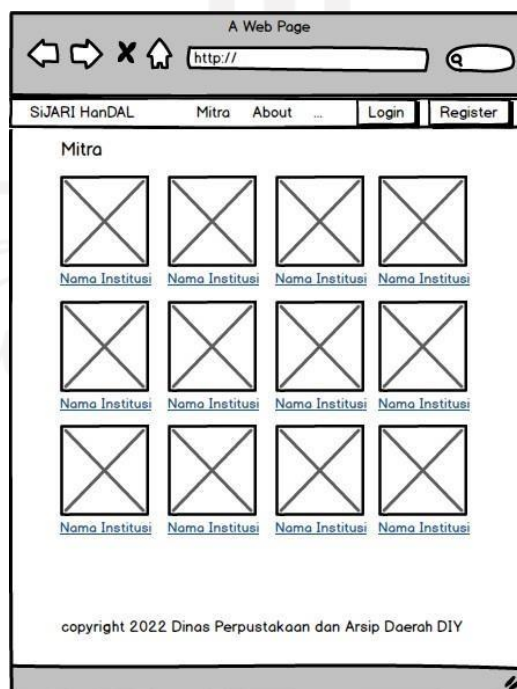
Gambar 3.14 Rancangan Halaman Detail Arsip

3.6.10 Rancangan Halaman Pencarian Buku Lanjutan

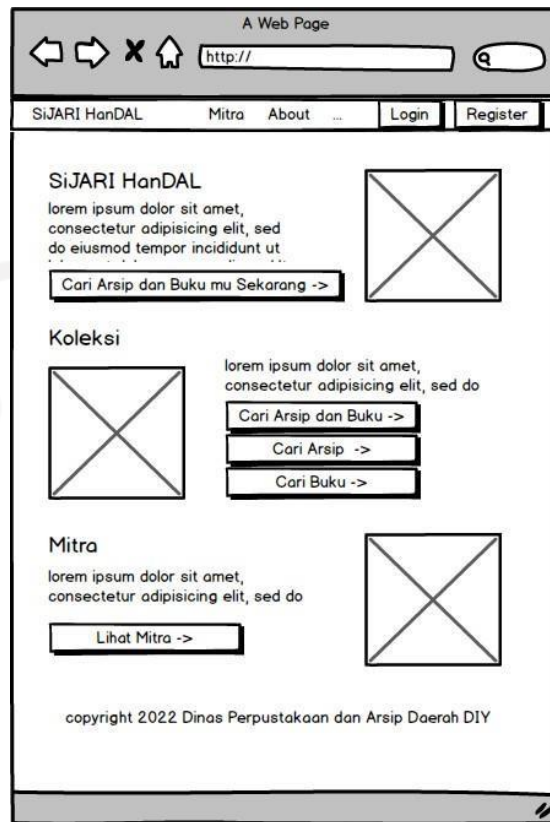
Rancangan halaman pencarian buku lanjutan dapat dilihat pada 3.19. Pada halaman ini terdapat filter penerbit, pengarang, dan tahun.



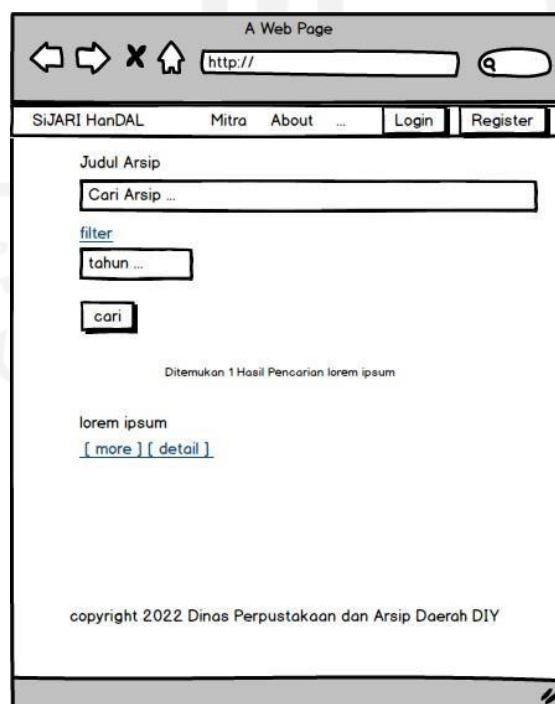
Gambar 3.15 Rancangan Halaman Detail Buku



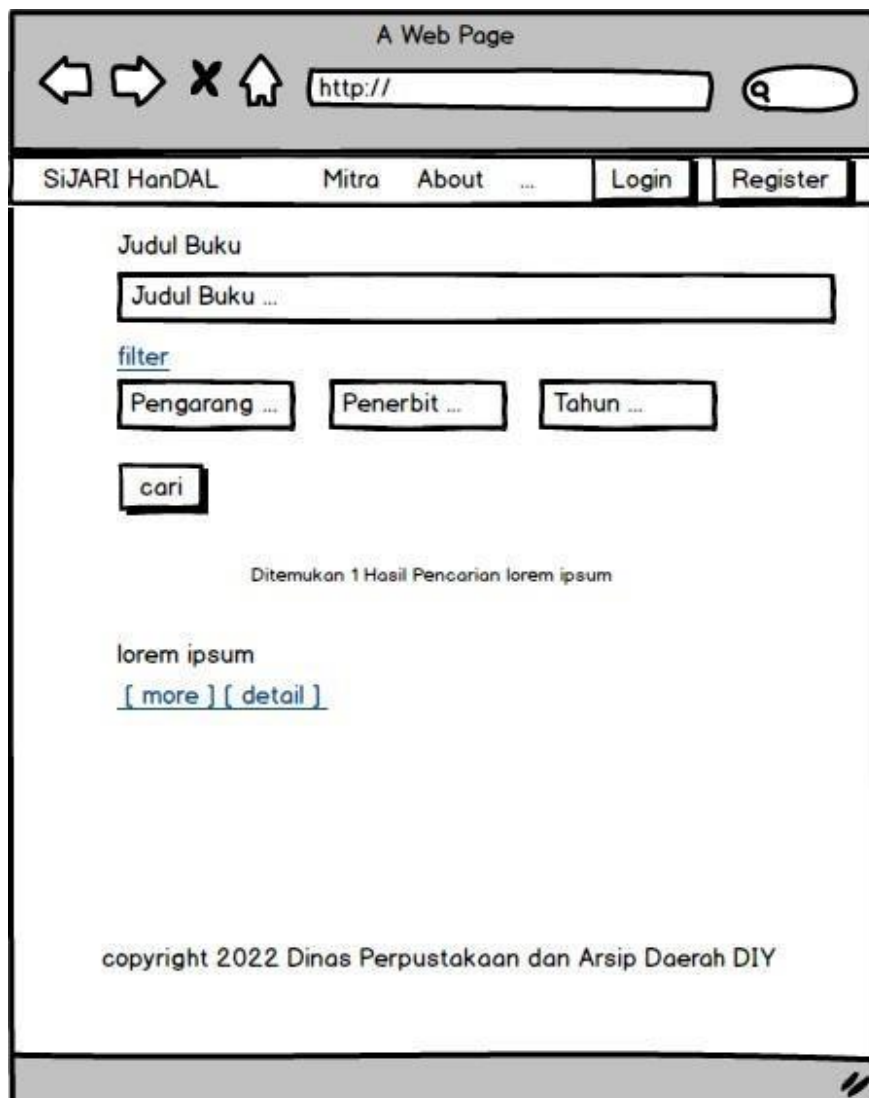
Gambar 3.16 Rancangan Halaman Mitra



Gambar 3.17 Rancangan Halaman About



Gambar 3.18 Rancangan Halaman Pencarian Arsip Lanjutan



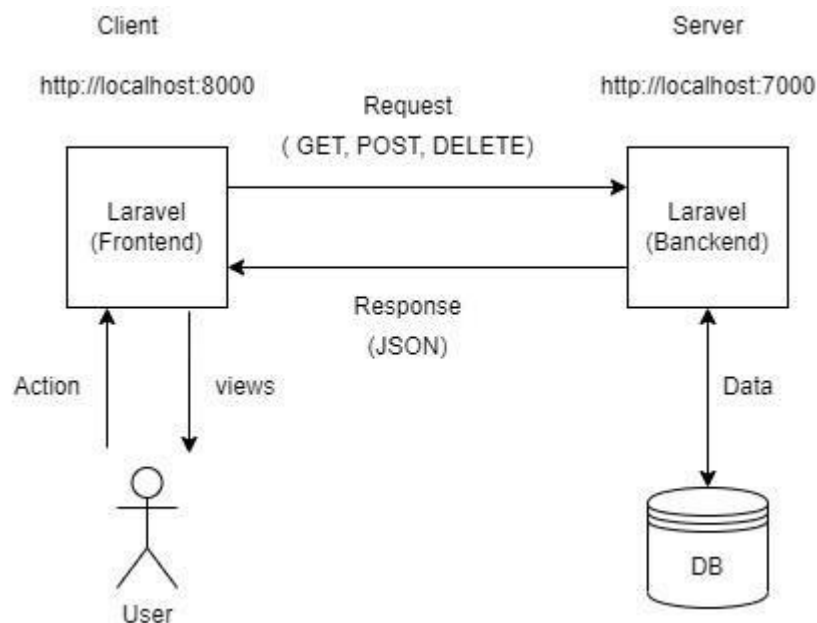
Gambar 3.19 Rancangan Halaman Pencarian Buku Lanjutan

3.7 Perancangan RESTful API

Perancangan *REST API* merupakan tahapan yang penting untuk dijadikan dasar dalam melakukan pengembangan sistem ini. Dalam perancangan *API* pada SiJARI HanDAL menggunakan arsitektur *REST* yang dibangun dengan *framework Laravel*. *REST API* mempresentasikan interaksi antara *client (Laravel)* dengan *server (Laravel)* pada SiJARI HanDAL seperti pada Gambar 3.20. Agar keduanya dapat saling berkomunikasi, maka diperlukan *API* sebagai penghubung antar aplikasi yang dibuat. Adapun beberapa *method API* yang biasanya digunakan seperti:

- GET* digunakan untuk membaca data dari *database*.
- POST* digunakan untuk membuat data baru dan memperbaharui data pada *database*.
- PUT* digunakan untuk mengupdate data yang sudah ada sebelumnya.

d. *DELETE* digunakan untuk menghapus data dari *database*.



Gambar 3.20 Arsitektur SiJARI HanDAL

Berdasarkan Gambar 3.20, *client* melakukan HTTP *Request* dengan *method* GET, POST, DELETE ke *server* yang menyediakan *resource* melalui URI atau yang sering disebut dengan *endpoint*. Kemudian REST *Server* memberi umpan balik berupa HTTP *response* sesuai dengan permintaan dari sisi *client*. Server akan mengembalikan *response* kepada *client* dalam format *JSON*. Keluaran yang dihasilkan berformat *JSON* tersebut akan diubah sedemikian rupa sehingga dapat dibaca pada sisi *client*.

Adapun rancangna penggunaan dari *API* yang akan dikembangkan sebagai berikut:

a. Rancangan *API* untuk *service* Buku

Tabel 3.7 Tabel *Endpoint API service* Buku

Endpoint	Method
/api/buku/list	GET
/api/buku/detail/{id}	GET
/api/buku/author/{author}	GET
/api/buku/tahun/{tahun}	GET
/api/buku/publisher/{publisher}	GET
/api/buku/koleksi/{koleksi}	GET

/api/buku/topik/{topik}	GET
/api/buku/cari	POST
/api/buku/totalcari	POST
/api/buku/search	POST

b. Rancangan API untuk *service* Arsip

Tabel 3.8 Tabel *Endpoint API service* Arsip

Endpoint	Method
/api/arsip/list	GET
/api/arsip/detail/{id}	GET
/api/arsip/tahun/{tahun}	GET
/api/arsip/cari	POST
/api/arsip/totalcari	POST
/api/arsip/search	POST

c. Rancangan API untuk *service Feedback*Arsip

Tabel 3.9 Tabel *Endpoint API service Feedback*Arsip

Endpoint	Method
/api/feedbackarsip/list/{id}	GET
/api/feedbackarsip/{id}/edit	GET
/api/feedbackarsip/submit/{id}	POST
/api/feedbackarsip/{id}/update	PUT
/api/feedbackarsip/{id}/delete	DELETE

d. Rancangan API untuk *service Feedback*Buku

Tabel 3.10 Tabel *Endpoint API service Feedback*Buku

Endpoint	Method
/api/feedbackbuku/list/{id}	GET
/api/feedbackbuku/{id}/edit	GET
/api/feedbackbuku/submit/{id}	POST

/api/feedbackbuku/{id}/update	PUT
/api/feedbackbuku/{id}/delete	DELETE

e. Rancangna *API* untuk *service User*

Tabel 3.11 Tabel *Endpoint API service User*

Endpoint	Method
/api/user/login	GET
/api/user/redirect	GET
/api/user/callback	GET
/api/user/dologin	POST
/api/user/email	POST
/api/user/store	POST
/api/user/{id}/update	POST

BAB IV

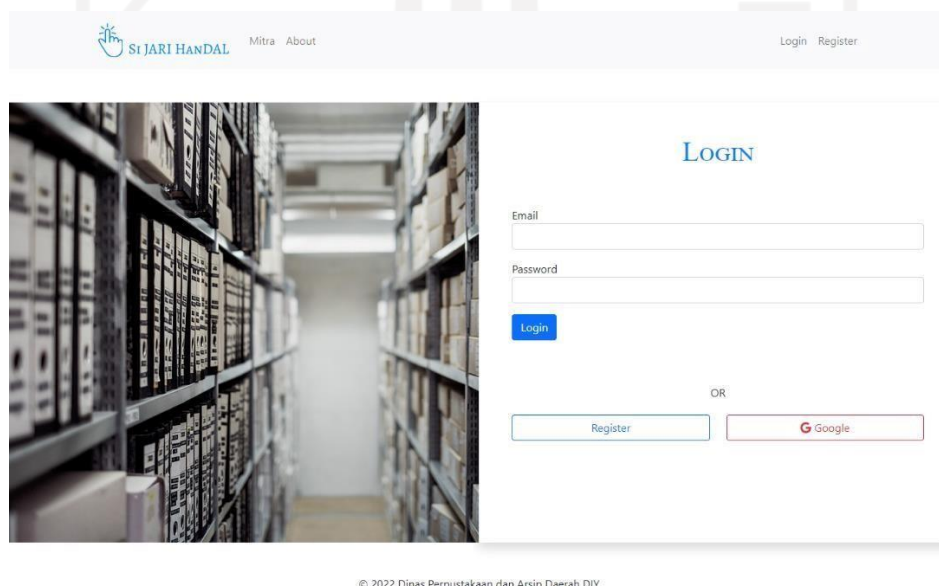
HASIL DAN PEMBAHASAN

4.1 Hasil

Berdasarkan pada perancangan yang telah dibuat pada bab sebelumnya, dibuatlah SiJARI HanDAL yang menggunakan Bahasa pemrograman *PHP* dengan bantuan *framework* Laravel. Pada bagian ini penulis akan melampirkan hasil *screenshot* dari sistem yang sudah dikembangkan. Bagian yang ditampilkan adalah *user interface* dari sistem. *User interface* adalah tampilan antarmuka sistem yang dapat dilihat dan diakses oleh pengguna website. *User interface* dari SiJARI HanDAL adalah sebagai berikut:

4.1.1 Login

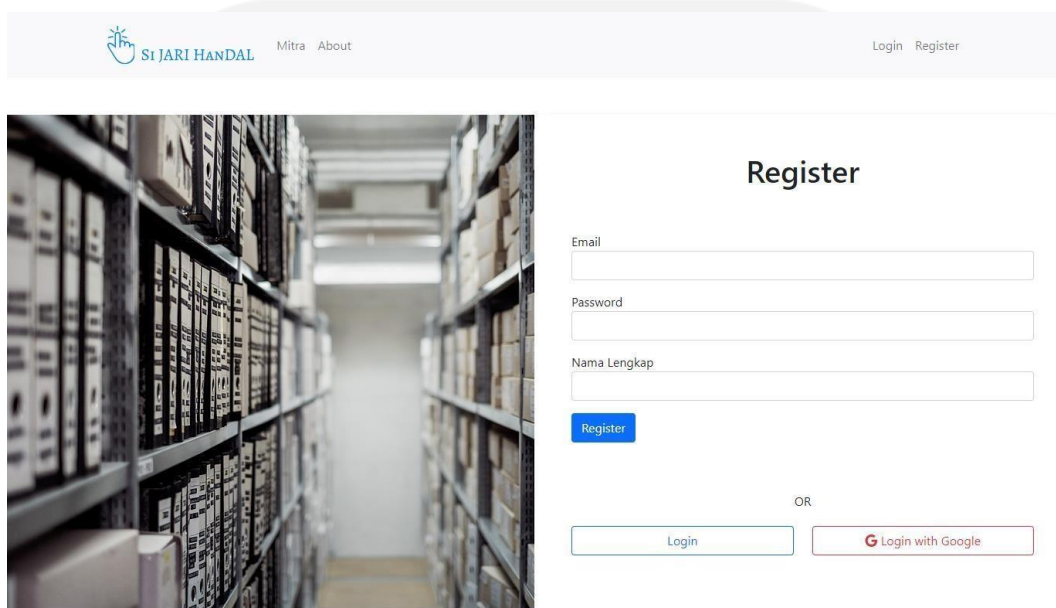
Pada halaman login pengguna dapat mengisi form *username* dan *password* berdasarkan *username* dan *password* yang telah dibuat pada form register, atau pengguna dapat melakukan login dengan menggunakan akun Google. Untuk mencari arsip maupun sumber lain seperti buku pengguna tidak diwajibkan untuk login. Pengguna diwajibkan login ketika akan memberikan *feedback* terhadap arsip maupun buku. Adapun halaman login yang telah dibuat dapat dilihat pada Gambar 4.1.



Gambar 4.1 Halaman Login.

4.1.2 Register

Pada halaman register pengguna dapat mengisi form *email*, *password* dan nama lengkap untuk mendapatkan akun aplikasi ini, atau pengguna dapat melakukan login dengan menggunakan akun Google. Adapun halaman register yang telah dibuat dapat dilihat pada Gambar 4.2.



The screenshot shows the 'Register' page of the SiJARI HanDAL website. The header includes the logo 'SI JARI HANDAL' and navigation links 'Mitra' and 'About'. On the right side of the header, there are links for 'Login' and 'Register'. The main content area is titled 'Register' and contains a registration form with the following fields: 'Email', 'Password', and 'Nama Lengkap'. Below these fields is a blue 'Register' button. Underneath the button, there is an 'OR' separator and two buttons: 'Login' and 'Login with Google'. The background of the page features a photograph of a library aisle with bookshelves.

© 2022 Dinas Perpustakaan dan Arsip Daerah DIY

Gambar 4.2 Halaman Register

4.1.3 Home

Halaman home adalah halaman awal ketika pengguna membuka website SiJARI HanDAL. Pada halaman home terdapat form untuk melakukan pencarian arsip maupun buku. Adapun halaman home yang telah dibuat dapat dilihat pada Gambar 4.3.

4.1.4 Hasil Pencarian

Halaman pencarian akan menampilkan data-data dari arsip maupun buku sesuai dengan kata kunci yang telah dimasukkan sebelumnya oleh pengguna. Adapun halaman hasil pencarian yang telah dibuat dapat dilihat pada Gambar 4.4.

4.1.5 Detail Arsip

Pada halaman detail arsip selain berisikan informasi yang lebih detail dari arsip yang sedang dicari juga terdapat form untuk memberi *feedback*. Pengguna diwajibkan untuk melakukan login terlebih dahulu agar dapat memberikan *feedback*. Adapun halaman detail arsip yang telah dibuat dapat dilihat pada Gambar 4.5.

4.1.6 Detail Buku

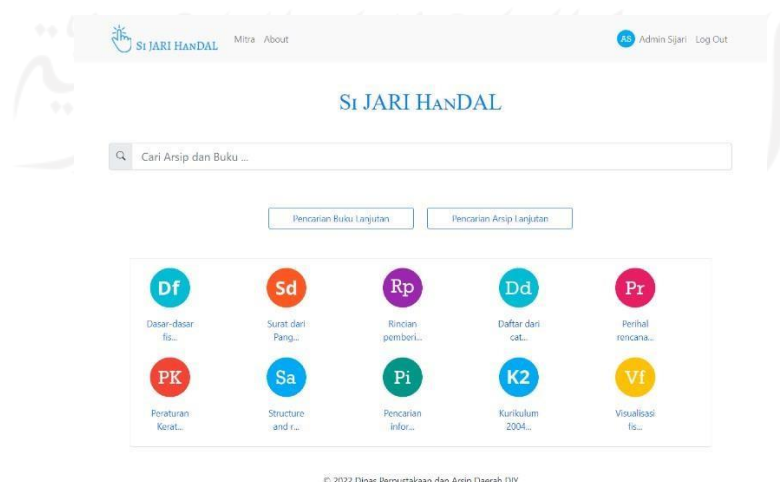
Pada halaman detail buku selain berisikan informasi detail tentang buku juga terdapat form untuk memberikan *feedback*. Pengguna diwajibkan login terlebih dahulu agar dapat memberikan *feedback*. Adapun halaman detail buku yang telah dibuat dapat dilihat pada Gambar 4.6.

4.1.7 Pencarian Arsip Lanjutan


Pada halaman pencarian arsip lanjutan terdapat *form* judul dan *form* tahun yang nantinya dapat diisi oleh pengguna sesuai dengan kebutuhan pencarian. Adapun halaman pencarian arsip lanjutan yang telah dibuat dapat dilihat pada Gambar 4.7.

4.1.8 Pencarian Buku Lanjutan

Pada halaman pencarian buku lanjutan terdapat 4 *form* yang berisikan *form* judul, *form* penulis, *form* penerbit serta *form* tahun. Adapun halaman pencarian buku lanjutan yang telah dibuat dapat dilihat pada Gambar 4.8.



Gambar 4.3 Halaman Home



Mitra About

[Login](#) [Register](#)

SI JARI HANDAL

Ditemukan **233 Arsip** dan **108 Buku** Hasil Pencarian **keuangan**

Pencarian Buku Lanjutan

Pencarian Arsip Lanjutan

Arsip

Buku

Surat Keputusan Gubernur Kepala Daerah DIY No. 173/Pem.D/U.P/B4 tanggal 14 Agustus 1980 tentang hukuman disiplin pegawai jabatan administratif bagi Ny. Endang Murniati, BA Staf Biro Keuangan Setwilda Prop. DIY. [arsip]
[\[more\]](#) [\[detail\]](#)

Peraturan Daerah Provinsi DIY Nomor 12 tahun 1996 tentang Kedudukan Keuangan Ketua, Wakil Ketua, dan anggota DPRD Provinsi DIY. [arsip]
[\[more\]](#) [\[detail\]](#)

Surat dari Badan Pengawas Keuangan dan Pembangunan Perwakilan Propinsi Jawa Timur nomor: s-06949/pw.13.8/1994 perihal petunjuk pelaksanaan pemeriksaan program Inpres Desa Tertinggal, kepada Inspektorat Wilayah kabupaten/Kotamadya se Jawa Timur [arsip]
[\[more\]](#) [\[detail\]](#)

Nota kesepakatan antara UNDP-IAI KASP-PPA Fakultas Ekonomi UGM dengan Pemkab Sleman No. 11/PK.KDH/A/2002 dan No. 012/PPA FE UGM/II/2002 tentang Implementasi Sistem Akuntabilitas Keuangan Daerah Pemerintah Kabupaten Sleman. [arsip]
[\[more\]](#) [\[detail\]](#)

Surat dari Gubernur Kepala Daerah DIY kepada Drs. Suhardjo, Irban Keuangan pada Inspektorat Wilayah Prop. DIY untuk monitoring dan pemeriksaan persiapan pemilu tahun 1982 pada pemerintah dati II Kabupaten Kulon Progo. [arsip]
[\[more\]](#) [\[detail\]](#)

Keputusan Camat/Ketua Panitia Pemungutan Suara Kecamatan Panjatan No : Ola/PPS/1981 tentang Pengangkatan Kepala Sub-Bagian Penyelenggara, Kepala Sub-Bagian Keuangan dan Pelaksana-pelaksana pada sekretariat PPS Kecamatan Panjatan. [arsip]
[\[more\]](#) [\[detail\]](#)

Keputusan Camat/Ketua panitia pemungutan suara Kecamatan Samigaluh No. 005/PPS/1981 tentang pemberhentian dengan hormat dari jabatan pelaksana Sekretariat PANWASLAKCAM, pelaksana Sekretariat panitia pemungutan suara dan Kepala Sub. Bagian Keuangan Sekreta [arsip]
[\[more\]](#) [\[detail\]](#)

Surat dari Badan Pengawas Pemilihan Umum Republik Indonesia ditujukan kepada Ketua Badan Pemeriksa Keuangan Republik Indonesia perihal Netralitas Ketua, Wakil Ketua, dan anggota Badan Pemeriksa Keuangan Dalam Pemilu Presiden dan Wakil Presiden Tahun 2009. [arsip]
[\[more\]](#) [\[detail\]](#)

Surat Departemen Keuangan No 5-1402/WPB.15/KP.0110/2009 tentang Konfirmasi Penyerapan Realisasi Anggaran Satuan Kerja sampai dengan Triwulan III Tahun 2009. [arsip]
[\[more\]](#) [\[detail\]](#)

Surat Badan Pengawas Pemilihan Umum Republik Indonesia No 890/Bawaslu/Set/VII/2009 tentang Permintaan Laporan Keuangan Semester I Tahun 2009 dan atau Serah Terima Barang Milik Negara. [arsip]
[\[more\]](#) [\[detail\]](#)

1

2

3

4

5

© 2022 Dinas Perpustakaan dan Arsip Daerah DIY

Gambar 4.4 Halaman Hasil Pencarian

SI JARI HANDAL Mitra About Admin Sijari Log Out

Surat dari Pangeran Haryo Pranala Mangala kepada Kawedanan Prajurit Nomor 53/72 perihal surat pengantar dari surat negara tanggal 10 Nl- Gatsu 2604/DS.2269/41 yang diserahkan lewat Parentah Hageng Keraton.

Reference
Code
Archive
Date
File
Subfile
Berkas
Sub Serie
Serie
Sub Fond
Fond
Online Access <https://doad.jogjapost.go.id/sibs/archiv/site/surat-dari-pangeran-haryo-pranala-mangala-kepada-kawedanan-prajurit-nomor-5312-perihal-surat-pengantar-dari-surat-negara-daryad-10-nl-gatsu-2604ds226941-yang-diserahkan-lewat-parentah-hageng-keraton.34421>

1 Feedback

AS

Kirim

Nugroho Agung Wibowo 20 August 2022
detail masih belum lengkap harap segera dilengkapi

BALAS

AS

Kirim

AS Admin Sijari 20 August 2022
baik kak, akan coba kami lengkapi secepatnya

© 2022 Dinas Perpustakaan dan Arsip Daerah DIY

Gambar 4.5 Halaman Detail Arsip

SI JARI HANDAL Mitra About Admin Sijari Log Out

Dasar-dasar fisika komputasi menggunakan MATLAB (Edisi 1)

tahun

publisher

koleksi

topik

institusi

Online Access <http://search.jogjalib.com/record/uedlib-100786>

1 Feedback

AS

Kirim

AS Admin Sijari 19 August 2022
detail masih perlu di tambah

BALAS

AS

Kirim

AS Admin Sijari 20 August 2022
coba balas komen ini

© 2022 Dinas Perpustakaan dan Arsip Daerah DIY

Gambar 4.6 Halaman Detail Buku

The screenshot shows the SiJARI HandDAL search page. At the top, there is a navigation bar with the logo 'Si JARI HANDDAL' and links for 'Mitra' and 'About'. On the right side of the navigation bar, there are links for 'Login' and 'Register'. Below the navigation bar, there is a search input field with the text 'sejarah'. Underneath the search field, there is a 'Filter' section with a text input field containing '1970'. A blue 'Cari' button is located below the filter. The search results section shows 'Ditemukan 1 Hasil Pencarian sejarah' and a single result: 'Item: BPAD.OTDA.T21.XII.1.734 Sejarah Penjajahan Bangsa Asing di Indonesia.' with a '[more] [detail]' link. At the bottom, there is a copyright notice: '© 2022 Dinas Perpustakaan dan Arsip Daerah DIY'.

Gambar 4.7 Halaman Pencarian Arsip Lanjutan

The screenshot shows the SiJARI HandDAL search page. At the top, there is a navigation bar with the logo 'Si JARI HANDDAL' and links for 'Mitra' and 'About'. On the right side of the navigation bar, there are links for 'Login' and 'Register'. Below the navigation bar, there is a search input field with the text 'blogger'. Underneath the search field, there is a 'Filter' section with three text input fields containing 'wahyu', 'andi', and '2010'. A blue 'Cari' button is located below the filter. The search results section shows 'Ditemukan 1 Hasil Pencarian blogger' and a single result: 'Build your blogger XML template' with the author 'Penulis: Sya.ban, Wahyu' and a '[more] [detail]' link. At the bottom, there is a copyright notice: '© 2022 Dinas Perpustakaan dan Arsip Daerah DIY'.

Gambar 4.8 Halaman Pencarian Buku Lanjutan

4.1.9 About

Halaman about berisikan pada SiJARI HandDAL berisikan informasi umum tentang SiJARI HandDAL seperti jumlah arsip yang telah ada saat ini dan juga jumlah buku yang ada pada saat ini. Adapun halaman about yang telah dibuat dapat dilihat pada Gambar 4.10.

4.1.10 Mitra

Halaman mitra berisikan daftar institusi yang sudah melakukan kerjasama dengan Dinas Perpustakaan dan Arsip Daerah (DPAD) DIY (Daerah Istimewa Yogyakarta). Institusi yang sudah bekerjasama dengan DPAD berupa universitas, sekolah hingga kedinasan. Adapun halaman mitra yang telah dibuat dapat dilihat pada Gambar 4.9.


SI JARI HENDAL Mitra About Login Register

Mitra

 Dikabupaten Gunungkidul	 OPAD DIT	 DPK KABUPATEN SLEMAN	 PERPUSTAKAAN Hutan Progo
 Perustakaan Balai Akuatik DIY	 Perustakaan Balai Bahasa DIY	 Perustakaan Insipier Yogyakarta	 Perustakaan ISI Yogyakarta
 Perustakaan ITNY	 Perustakaan Kota Jogja	 Perustakaan LPP Yogyakarta	 Perustakaan MAN 3 Sleman
 Perustakaan Politeknik ATK Yogyakarta	 Perustakaan Politeknik Kemendik Yogyakarta	 Perustakaan SMAN 1 Yogyakarta	 Perustakaan SMP Muhammadiyah 2 Yogyakarta
 Perustakaan STIE YKPN Yogyakarta	 Perustakaan STIKES Bethesda Yolkum	 Perustakaan STIM YKPN	 Perustakaan STPMD APMD
 Perustakaan STPN	 Perustakaan UAD Yogyakarta	 Perustakaan UAIY	 Perustakaan UGM
 Perustakaan UIN Indonesia	 Perustakaan UIN Sunan Kalijaga	 Perustakaan UICDF	 Perustakaan UMB Yogyakarta
 Perustakaan UIR	 Perustakaan UMY	 Perustakaan UNISA Yogyakarta	 Perustakaan Universitas Janabada
 Perustakaan UMY	 Perustakaan UPN Yogyakarta	 Perustakaan USD	 Perustakaan UST Yogyakarta
 Perustakaan STIKES Surya Global	 Resource Centre UNIVERSITAS AMIKOM YOGYAKARTA	 STAIT Yogyakarta	 STMIK Ahmad Yani
 Universitas Jember Achmad Yani Yogyakarta			

© 2022 Dinas Perpustakaan dan Kearsifan Daerah DIY

Gambar 4.9 Halaman Mitra


Mitra About


[Login](#) [Register](#)

SI JARI HANDAL


Sebuah website satu pintu untuk pencarian semua koleksi arsip dan buku yang tersedia di Dinas Perpustakaan dan Arsip Daerah DIY

SIJARI HanDAL memudahkan pengguna dalam melakukan pencarian arsip beserta sumber-sumber yang mendukung seperti buku

Cari Arsip dan Buku Mu Sekarang →



Koleksi



Saat ini terdapat 11064 koleksi arsip dan 14250 koleksi buku

Cari Arsip dan Buku →


Cari Arsip →

Cari Buku →

Mitra

Kami bekerja sama dengan sejumlah Dinas dan Universitas yang ada di Yogyakarta

Lihat Mitra →



© 2022 Dinas Perpustakaan dan Arsip Daerah DIY

Gambar 4.10 Halaman About

4.2 Pengujian Sistem

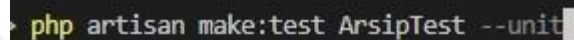
Pengujian sistem merupakan tahapan terakhir yang dilakukan dalam pengembangan sistem. Proses pengujian dilakukan untuk menguji apakah sistem SiJARI HanDAL sudah berjalan dengan baik dan sesuai dengan tujuan perancangan atau belum. Selain itu, pengujian juga digunakan untuk mendeteksi kesalahan yang mungkin terjadi dalam implementasi kode

program. Pada penelitian ini metode pengujian dilakukan dengan dua cara. Pertama pengujian dilakukan dengan metode *unit testing*. Kedua pengujian dilakukan dengan Postman.

4.2.1 Pengujian API dengan *Unit Testing*

Unit testing adalah teknik untuk menguji sebuah kode yang ditulis oleh pengembang yang digunakan untuk menguji bagian terkecil atau area spesifik dari suatu fungsional kode. *Unit testing* termasuk dalam pengujian otomatis sehingga pengembang perlu membuat kode program untuk menguji setiap fitur atau fungsionalitas sesuai dengan yang diinginkan.

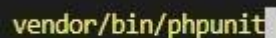
Pada kasus ini *unit testing* dilakukan pada *framework laravel* dengan cara membuat *file test*. Contoh membuat *file test* pada *laravel* dapat dilihat pada Gambar 4.11.



```
php artisan make:test ArsipTest --unit
```

Gambar 4.11 Membuat *File Unit Testing*

Adapun untuk menjalankan *unit testing* pada *laravel* dapat dilakukan dengan perintah dengan `php artisan test` ataupun dengan `phpunit` seperti pada Gambar 4.12.



```
vendor/bin/phpunit
```

Gambar 4.12 Perintah Menjalankan *Unit Testing*

Perbedaan antara perintah `phpunit` dengan `php artisan test` adalah pada `php artisan test` terdapat laporan pengujian *verbose* untuk memudahkan pengembangan dan *debugging*. Adapun *source code* dari *unit testing* pada fungsi *rest api* untuk modul arsip dapat dilihat pada Gambar 4.13. sedangkan *source code* untuk *unit testing* pada *rest api* untuk modul buku dapat dilihat pada Gambar 4.14. Untuk *source code unit testing api* untuk modul *feedback* dapat dilihat pada Gambar 4.15. Adapun *source code unit testing rest api* untuk modul user dapat dilihat pada Gambar 4.16.

```

1  <?php
2
3  namespace Tests\Unit;
4
5  use Illuminate\Foundation\Testing\RefreshDatabase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Tests\TestCase;
8  use App\Arsip;
9
10 class ArsipTest extends TestCase
11 {
12     public function test_lihat_semua_arsip()
13     {
14         $this->get('/api/arsip/list')
15             ->assertStatus(200);
16     }
17
18     public function test_cari_arsip()
19     {
20         $kw = ["keyword" => "ekonomi"];
21         $this->post('/api/arsip/cari', $kw)
22             ->assertStatus(200);
23     }
24
25     public function test_detail_arsip()
26     {
27         $id = 1;
28         $this->get('/api/arsip/detail/' . $id)
29             ->assertStatus(200);
30     }

```

Gambar 4.13 Source Code Unit Testing Arsip

```

1  <?php
2
3  namespace Tests\Unit;
4
5  use Illuminate\Foundation\Testing\RefreshDatabase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Tests\TestCase;
8
9  class BukuTest extends TestCase
10 {
11     public function test_lihat_semua_buku()
12     {
13         $this->get('/api/buku/list')
14             ->assertStatus(200);
15     }
16
17     public function test_cari_buku()
18     {
19         $param = ["keyword" => "ekonomi"];
20         $this->post('/api/buku/cari', $param)
21             ->assertStatus(200);
22     }
23
24     public function test_totalcari_buku()
25     {
26         $param = ["keyword" => "ekonomi"];
27         $this->post('/api/buku/totalcari', $param)
28             ->assertStatus(200);
29     }

```

Gambar 4.14 Source Code Unit Testing Buku


```

1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6  use Illuminate\Foundation\Testing\RefreshDatabase;
7  use Illuminate\Foundation\Testing\WithFaker;
8
9  class FeedbackTest extends TestCase
10 {
11     public function test_lihat_feedback_arsip()
12     {
13         $id_arsip = 2;
14         $this->get('/api/feedbackarsip/list/' . $id_arsip)
15             ->assertStatus(200);
16     }
17
18     public function test_submit_feedback_arsip()
19     {
20         $id_arsip = 3;
21         $param = [
22             "user_id" => 3,
23             "isi" => "coba unit testing",
24         ];
25         $this->post('/api/feedbackarsip/submit/' . $id_arsip, $param)
26             ->assertStatus(200);
27     }
28 }

```

Gambar 4.15 Source Code Unit Testing Feedback

```

1  <?php
2
3  namespace Tests\Unit;
4
5  use Tests\TestCase;
6
7  class UserTest extends TestCase
8  {
9
10     public function test_store_user()
11     {
12         $param = [
13             "email" => "unitesting@example.com",
14             "password" => "cobaunitesting",
15             "name" => "testing",
16         ];
17         $this->post('/api/user/store', $param)
18             ->assertStatus(200);
19     }
20
21     public function test_email_user()
22     {
23         $param = ["email" => "unitesting@example.com"];
24         $this->post('/api/user/email', $param)
25             ->assertStatus(200);
26     }
27 }

```

Gambar 4.16 Source Code Unit Testing User

Pada masing-masing *unit testing* setiap method yang ada selalui diawali dengan kata *test_* ini merupakan salah satu syarat pembuatan test, dengan begitu *phpunit* akan tau jika method tersebut adalah test dan menjalankannya. Pada setiap method diatas juga terdapat `assertStatus(...)` fungsi ini akan memeriksa respon status yang akan diterima sesuai dengan espektasi atau tidak. Adapun hasil dari *unit testing* dapat dilihat pada Gambar 4.17.

```
PASS Tests\Unit\ArsipTest
✓ lihat semua arsip
✓ cari arsip
✓ detail arsip
✓ totalcari arsip
✓ search arsip
✓ tahun arsip

PASS Tests\Unit\BukuTest
✓ lihat semua buku
✓ cari buku
✓ totalcari buku
✓ search buku
✓ detail buku

PASS Tests\Unit\ExampleTest
✓ example

PASS Tests\Unit\FeedbackTest
✓ lihat feedback arsip
✓ submit feedback arsip
✓ lihat feedback buku
✓ submit feedback buku

PASS Tests\Unit\UserTest
✓ store user
✓ email user

PASS Tests\Feature\ExampleTest
✓ example

Tests: 19 passed
Time: 2.71s
```

Gambar 4.17 Hasil *Unit Testing*

Gambar 4.17 hasil dari *unit testing* berhasil tanpa adanya *error*, yang artinya semua fungsi *Rest API* yang dibuat tiap modul berjalan sesuai dengan fungsinya. Dengan menggunakan *unit testing* memudahkan para pengembang dalam melakukan pengujian, sebab

setiap kali ada perubahan pada fungsi *Rest API* yang dibuat, tidak perlu melakukan pengujian menggunakan Rest Client, serta tidak perlu melakukan pengujian satu persatu per fungsi jika ada perubahan yang mempengaruhi fungsi-fungsi lainnya.

4.2.2 Pengujian API dengan Postman

Pengujian ini digunakan untuk menentukan kelayakan *REST API* yang telah dibuat agar dapat digunakan secara optimal pada sistem SiJARI HanDAL. Adapun cara menguji *API* dilakukan dengan Postman. Pengujian dilakukan dengan mengatur method sesuai dengan *endpoint* yang dimasukkan seperti pada Gambar 4.18.

Pengujian *API service* Arsip

The screenshot shows the Postman interface for a GET request to the endpoint `http://127.0.0.1:7000/api/arsip/list`. The response status is `200 OK`, with a time of `3.38 s` and a size of `8.02 MB`. The response body is a JSON array containing one item with the following details:

```

1 [{"id":1,"item_url":"http://dpad.jogjaprov.go.id/siks/archive/site/
susunan-dan-personalia-panitia-peringatan-hari-kesehatan-se-dunia-tahun-1992-di-kabupaten-dati-ii-kulon-progo-5117",
"item_title":"Susunan dan Personalia Panitia Peringatan Hari Kesehatan se-dunia tahun 1992 di Kabupaten Dati
II Kulon Progo.", "file_title":"","file_url":"","subfile_title":"","subfile_url":"","berkas_title":"","
"berkas_url":"","subserie_title":"","subserie_url":"","serie_title":"","serie_url":"","subfond_title":"","
"subfond_url":"","fond_title":"Fond: BPAD.OTDA.T21\\nSENARAI ARSIP KABUPATEN KULON PROGO (HASIL AKUISISI TAHUN
2003)\\n" "fond_url":"http://dpad.jogjaprov.go.id/siks/archive/site/

```

Gambar 4.18 *Testing Endpoint API list arsip*

Gambar 4.18 merupakan pengujian *API* dengan *endpoint* `/api/arsip/list`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status *code* 200. *Endpoint* ini digunakan untuk meminta *request* semua data arsip.

POST http://127.0.0.1:7000/api/arsip/cari?keyword=politik

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> keyword	politik			
Key	Value	Description		

Body Cookies Headers (10) Test Results Status: 200 OK Time: 4.47 s Size: 11.67 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 [{"id":12,"item_url":"http://dpad.jogjaprovo.go.id/siks/archive/site/peurat-perintah-permohonan-peserta-memerintahkan-kepada-sdr-maulana-nrp-26126-dkk-11-orang-untuk-mengikuti-santiaji-pemantapan-pembinaan-sosial-politik-daerah-51162","item_title":"Item: BPAD.OTDA.T21.XI.1.729\nSurat perintah permohonan peserta memerintahkan kepada Sdr. Maulana NRP. 26126 dkk. 11 orang untuk mengikuti Santiaji Pemantapan Pembinaan Sosial Politik Daerah.\n","file_title":"File: BPAD.OTDA.T21.XI\nPERTAHANAN DAN KEAMANAN\n","file_url":"http://dpad.jogjaprovo.go.id/siks/archive/site/ppertahanan-dan-keamanan-51159","subfile_title":"","subfile_url":"","berkas_title":"Berkas: BPAD.OTDA.T21.XI.1\nPPERTAHANAN DAN KEAMANAN\n","berkas_url":"http://dpad
```

Gambar 4.19 Testing Endpoint API cari arsip

Gambar 4.19 merupakan pengujian API dengan endpoint /api/arsip/cari. Setelah mengirim request, kemudian REST server memberi response bahwa request berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk mengambil data arsip berdasarkan parameter keyword yang dimasukkan. Pada endpoint ini data yang dikirimkan kembali berjumlah 10 data arsip.

POST http://127.0.0.1:7000/api/arsip/search?title=wisata&tahun=1970

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> title	wisata			
<input checked="" type="checkbox"/> tahun	1970			
Key	Value	Description		

Body Cookies Headers (10) Test Results Status: 200 OK Time: 380 ms Size: 1.47 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 [{"id":8,"item_url":"http://dpad.jogjaprovo.go.id/siks/archive/site/pdetail-plan-komplek-wisata-goa-kiskendop-51168","item_title":"Item: BPAD.OTDA.T21.XII.1.733\nplan Komplek Wisata Goa Kiskendo.\n","file_title":"File: BPAD.OTDA.T21.XII\nPARIWISATA\n","file_url":"http://dpad.jogjaprovo.go.id/siks/archive/site/ppariwisata-51165","subfile_title":"","subfile_url":"","berkas_title":"Berkas: BPAD.OTDA.T21.XII.1\nPARIWISATA\n","berkas_url":"http://dpad.jogjaprovo.go.id/siks/archive/site/pariwisata-51166","subserie_title":"","subserie_url":"","serie_title":"","serie_url":"","subfond_title":"","subfond_url":"","fond_title":"Fond: BPAD.OTDA.T21\nSENARAI ARSIP KABUPATEN KULON PROGO (HASIL AKUISISI TAHUN 2003)\n","fond_url":"http://dpad.jogjaprovo.go.id/siks/archive/site/psejarah-arsip-kabupaten-kulon-progo-hasil-akuisisi-tahun-2003brp-50737","created_at":null,"updated_at":null,"reference_code":"23400-24 BPAD.OTDA.T21.XII.1.733","archive_date":"January 1, 1970","meta":{"filter_1 REFERENCE CODE\n23400-24 BPAD.OTDA.T21.XII.1.733\nfilter_2 TITLE\nDetail plan Komplek Wisata Goa Kiskendo.\nfilter_3 LEVEL OF DESCRIPTION\nItem\nfilter_4 ARCHIVE DATE\nJanuary 1, 1970","status":""}]
```

Gambar 4.20 Testing Endpoint API search arsip

Gambar 4.20 merupakan pengujian API dengan *endpoint* /api/arsip/search. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. *Endpoint* ini digunakan untuk mengambil data arsip berdasarkan parameter *title* atau tahun yang dimasukkan.

GET http://127.0.0.1:7000/api/arsip/detail/10

Status: 200 OK Time: 624 ms Size: 114 KB

```

1 [{"id":1,"item_url":"http://dpad.jogjaprov.go.id/sike/archive/site/
susunan-dan-personalia-panitia-peringatan-hari-kesehatan-se-dunia-tahun-1992-di-kabupaten-dati-ii-kulon-progo-5117
7","item_title":"Susunan dan Personalia Panitia Peringatan Hari Kesehatan se-dunia tahun 1992 di Kabupaten Dati
II Kulon Progo.", "file_title":"","file_url":"","subfile_title":"","subfile_url":"","berkas_title":"","
"berkas_url":"","subserie_title":"","subserie_url":"","serie_title":"","serie_url":"","subfond_title":"","
"subfond_url":"","fond_title":"Fond: BPAD.OTDA.T21\\nSENARAI ARSIP KABUPATEN KULON PROGO (HASIL AKUISISI TAHUN
2003)\\n","fond_url":"http://dpad.jogjaprov.go.id/sike/archive/site/
peanarai-arsip-kabupaten-kulon-progo-hasil-akuisisi-tahun-2003brp-50737","created_at":null,"updated_at":null,
"reference_code":"","archive_date":"","meta":"","status":""}]

```

Gambar 4.21 Testing Endpoint API detail arsip

Gambar 4.21 merupakan pengujian API dengan *endpoint* /api/arsip/detail/:id. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. *Endpoint* ini digunakan untuk meminta *request* data arsip berdasarkan *id* yang dicari.

GET http://127.0.0.1:7000/api/arsip/tahun/1970

Status: 200 OK Time: 702 ms Size: 5.97 KB

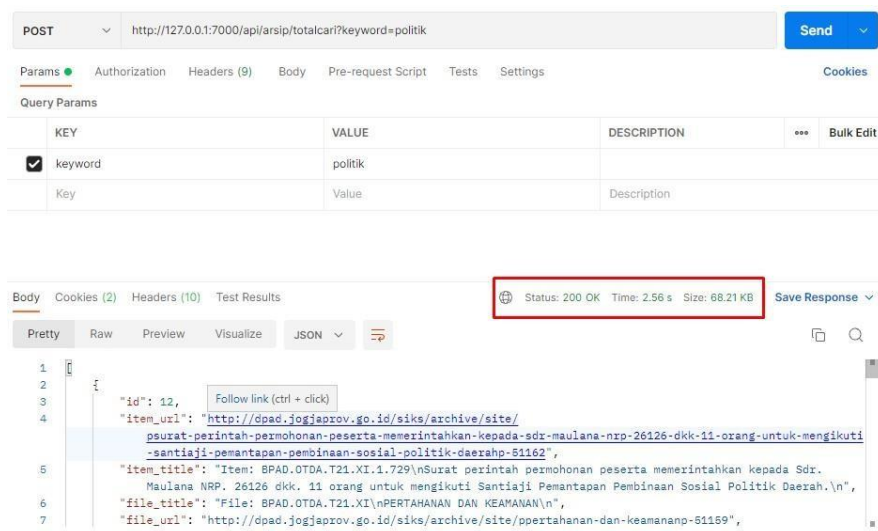
```

1 [{"id":7,"item_url":"http://dpad.jogjaprov.go.id/sike/archive/site/
pejarah-penjaahan-bangsa-asing-di-indonesiap-51169","item_title":"Item: BPAD.OTDA.T21.XII.1.734\\nSejarah
Penjajahan Bangsa Asing di Indonesia.\\n","file_title":"File: BPAD.OTDA.T21.XII\\nPARIWISATA\\n","file_url":"http://
dpad.jogjaprov.go.id/sike/archive/site/ppariwisataap-51169","subfile_title":"","subfile_url":"","
"berkas_title":"Berkas: BPAD.OTDA.T21.XII.1\\nPARIWISATA\\n","berkas_url":"http://dpad.jogjaprov.go.id/sike/archive/
site/pariwisata-51169","subserie_title":"","subserie_url":"","serie_title":"","serie_url":"","subfond_title":"","
"subfond_url":"","fond_title":"Fond: BPAD.OTDA.T21\\nSENARAI ARSIP KABUPATEN KULON PROGO (HASIL AKUISISI TAHUN
2003)\\n","fond_url":"http://dpad.jogjaprov.go.id/sike/archive/site/
peanarai-arsip-kabupaten-kulon-progo-hasil-akuisisi-tahun-2003brp-50737","created_at":null,"updated_at":null,
"reference_code":"23400-24 BPAD.OTDA.T21.XII.1.734","archive_date":"January 1, 1970","meta":{"filter_1 REFERENCE
CODE\\n23400-24 BPAD.OTDA.T21.XII.1.734\\nfilter_2 TITLE\\nSejarah Penjajahan Bangsa Asing di Indonesia.\\nfilter_3
LEVEL OF DESCRIPTION\\nItem\\nfilter_4 ARCHIVE DATE\\nJanuary 1, 1970","status":""}},{ "id":8,"item_url":"http://dpad.

```

Gambar 4.22 Testing Endpoint API tahun arsip

Gambar 4.22 merupakan pengujian *API* dengan *endpoint* `/api/arsip/tahun/:tahun`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. *Endpoint* ini digunakan untuk meminta *request* data arsip berdasarkan tahun yang dicari.



Gambar 4.23 *Testing Endpoint API* totalcari arsip

Gambar 4.23 merupakan pengujian *API* dengan *endpoint* `/api/arsip/totalcari`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. *Endpoint* ini digunakan untuk mengirim *request* data arsip berdasarkan parameter *keyword* yang dimasukkan. Pada *endpoint* ini semua data arsip berdasarkan *keyword* yang dimasukkan yang dikirimkan kembali.

Berdasarkan pada pengujian yang telah dilakukan, penggunaan *API* relative efisien karena tidak boros dalam penggunaan *bandwidth* dan tidak memakan waktu yang lama saat pertukaran data. Untuk lebih jelasnya dapat dilihat Tabel 4.1 yang menunjukkan seluruh daftar *API* untuk *service* arsip yang diuji beserta hasilnya.

Tabel 4.1 Tabel Daftar Pengujian *Service* Arsip

Endpoint	Fitur	Hasil	Waktu (ms)	Bandwidth (kilobyte)
<code>/api/arsip/list</code>	Menampilkan seluruh data arsip	sukses	3380	8020

/api/arsip/ cari	Mengambil 10 data arsip berdasarkan parameter <i>keyword</i> “politik”	sukses	4470	11,67
/api/arsip/ search	Mengambil data arsip berdasarkan parameter <i>title</i> “wisata” atau parameter tahun “1970”	sukses	380	1,47
/api/arsip/ detail/:id	Menampilkan detail arsip berdasarkan <i>id</i>	sukses	624	1,14
/api/arsip/ tahun/:tahun	Menampilkan arsip berdasarkan tahun	sukses	702	5,97
/api/arsip/ totalcari	Mengambil data arsip berdasarkan <i>keyword</i> “politik”	sukses	2560	68,21

Pengujian API service buku

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7000/api/buku/author/andrea
- Status:** 200 OK
- Time:** 473 ms
- Size:** 3.54 KB
- Response Body (JSON):**

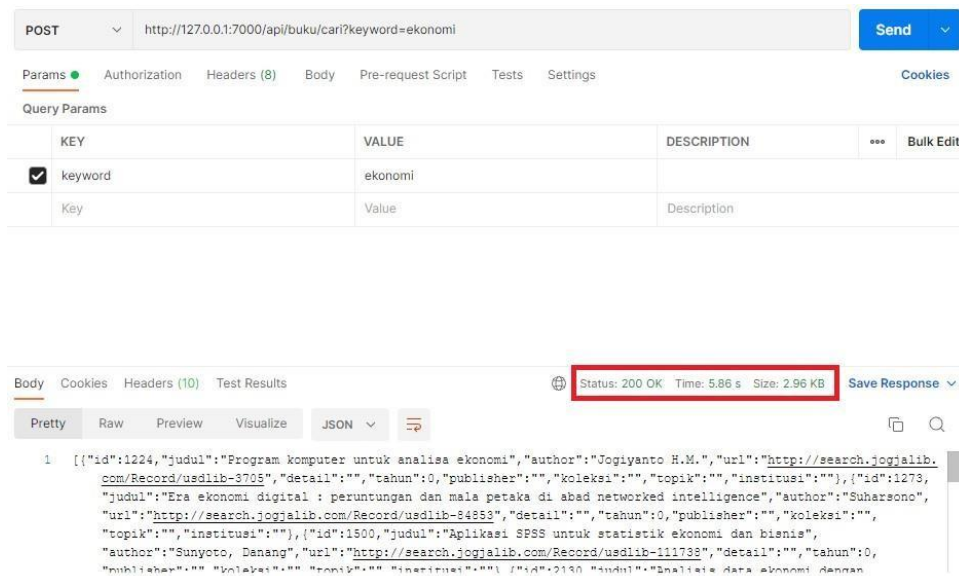
```
[{"id":2108,"judul":"Tip & trik manipulasi foto dan teks dengan photoshop","author":"Avellino, Andreas","url":"http://search.jogjalib.com/Record/usdlib-61989","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":2260,"judul":"Panduan desain grafis lengkap CorelDraw 12, Photoshop cs, Fagemaker 7","author":"Avellino, Andreas","url":"http://search.jogjalib.com/Record/usdlib-62620","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":2345,"judul":"Panduan praktis menggunakan microsoft office word 2003","author":"Agus B., Andreas","url":"http://search.jogjalib.com/Record/usdlib-62336","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":3254,"judul":"Rancangan dan pengembangan modul elektronik materi turunan fungsi dengan program Geogebra","author":"Proklamanto, Andreas Ricky","url":"http://search.jogjalib.com/Record/usdlib-118914","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":3358,"judul":"Peranan program video sebagai sarana untuk membangun komunikasi iman di kalangan kaum muda Katolik Lingkungan Bener Paroki Kometiran Yogyakarta","author":"Iswadi, Andreas","url":"http://search.jogjalib.com/Record/usdlib-74944","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":3386,"judul":"Using Games in Teaching Reading in the First Grade of SMP St."}

```

Gambar 4.24 Testing Endpoint API author buku

Gambar 4.24 merupakan pengujian API dengan endpoint /api/buku/author/:author. Setelah mengirim request, kemudian REST server memberi response bahwa request berhasil yang ditandai dengan status code 200. Endpoint ini

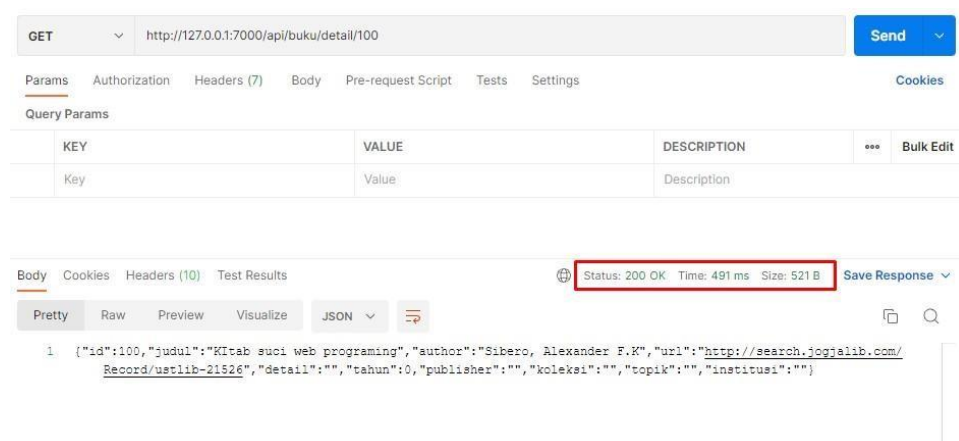
digunakan untuk meminta *request* semua data buku berdasarkan nama *author* yang dimasukkan.



The screenshot shows a REST client interface for a POST request to `http://127.0.0.1:7000/api/buku/cari?keyword=ekonomi`. The request body is empty. The response status is `200 OK`, with a time of `5.86 s` and a size of `2.96 KB`. The response body is a JSON array of 10 book records, each containing fields like `id`, `judul`, `author`, `url`, `detail`, `tahun`, `publisher`, `koleksi`, `topik`, and `institusi`.

Gambar 4.25 Testing Endpoint API cari buku

Gambar 4.25 merupakan pengujian API dengan *endpoint* `/api/buku/cari`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. *Endpoint* ini digunakan untuk mengirim *request* semua data buku berdasarkan *keyword* yang dimasukkan. Pada *endpoint* ini data yang dikirimkan kembali berjumlah 10 data buku.



The screenshot shows a REST client interface for a GET request to `http://127.0.0.1:7000/api/buku/detail/100`. The response status is `200 OK`, with a time of `491 ms` and a size of `521 B`. The response body is a single JSON object representing a book record with fields like `id`, `judul`, `author`, `url`, `detail`, `tahun`, `publisher`, `koleksi`, `topik`, and `institusi`.

Gambar 4.26 Testing Endpoint API detail buku

Gambar 4.26 merupakan pengujian API dengan endpoint `/api/buku/detail/:id`. Setelah mengirim request, kemudian REST server memberi response bahwa request berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk meminta request data buku berdasarkan id dari buku.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7000/api/buku/koleksi/yogya
- Status:** 200 OK, Time: 361 ms, Size: 14.81 KB
- Response Body (XML):**

```

1 [{"id":22,"judul":"Build your blogger XML template","author":"Sya,ban, Wahyu","url":"http://search.jogjalib.com/Record/unylib-0000046319","detail":{"id":"unylib-0000046319","oai_dc":{"id":"unylib-00000463192016-08-08 00:00:00Perpustakaan UAJYBuild your blogger XML templateSya,ban, WahyuWEB PROGRAMMINGAndi2010TEXT006.76 Sya b<br>Perpustakaan UAJY<br>Perpustakaan Yogyakarta<br>WEB PROGRAMMING<br>WEB PROGRAMMING<br>Sya,ban, Wahyu<br>Build your blogger XML template<br>"}

```

Gambar 4.27 Testing Endpoint API koleksi buku

Gambar 4.27 merupakan pengujian API dengan endpoint `/api/buku/koleksi/:koleksi`. Setelah mengirim request, kemudian REST server memberi response bahwa request berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk meminta request data buku berdasarkan koleksi yang dicari.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7000/api/buku/list
- Status:** 200 OK, Time: 35.79 s, Size: 3.59 MB
- Response Body (XML):**

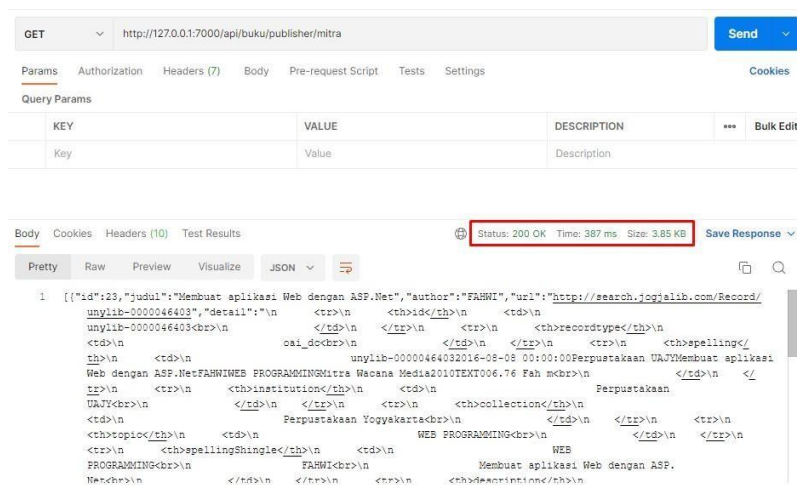
```

1 [{"id":22,"judul":"Build your blogger XML template", "author": "Sya,ban, Wahyu", "url": "http://search.jogjalib.com/Record/unylib-0000046319", "detail": {"id": "unylib-0000046319", "oai_dc": {"id": "unylib-00000463192016-08-08 00:00:00Perpustakaan UAJYBuild your blogger XML templateSya,ban, WahyuWEB PROGRAMMINGAndi2010TEXT006.76 Sya b<br>Perpustakaan UAJY<br>Perpustakaan Yogyakarta<br>WEB PROGRAMMING<br>WEB PROGRAMMING<br>Sya,ban, Wahyu<br>Build your blogger XML template<br>"}

```

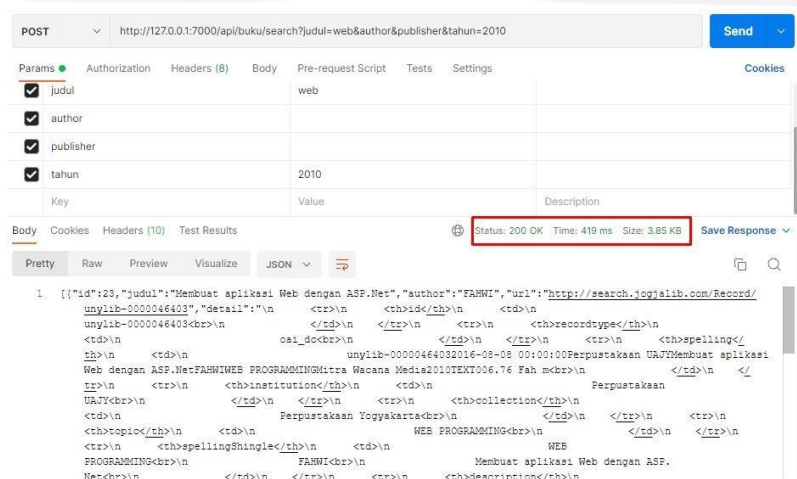
Gambar 4.28 Testing Endpoint API list buku

Gambar 4.28 merupakan pengujian API dengan endpoint `/api/buku/list`. Setelah mengirim request, kemudian REST server memberi response bahwa request berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk meminta request semua data buku.



Gambar 4.29 Testing Endpoint API publisher buku

Gambar 4.29 merupakan pengujian API dengan endpoint `/api/buku/publisher/:publisher`. Setelah mengirim request, kemudian REST server memberi response bahwa request berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk meminta request data buku berdasarkan publisher dari buku yang dicari.



Gambar 4.30 Testing Endpoint API search buku

Gambar 4.30 merupakan pengujian API dengan endpoint `/api/buku/search`. Setelah mengirim request, kemudian REST server memberi response bahwa request berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk mengirim request data buku berdasarkan judul, author, publisher atau tahun dari buku yang ingin dicari.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7000/api/buku/tahun/2010
- Status:** 200 OK, Time: 393 ms, Size: 736 KB
- Response Body (XML):**

```

1 [{"id":22,"judul":"Build your blogger XML template","author":"Sya,ban, Wahyu","url":"http://search.jogjalib.com/Record/unylib-0000046318","detail":":\n <tr>\n <th>id</th>\n <td>\n unylib-0000046318<br>\n </td>\n </tr>\n <th>recordtype</th>\n <td>\n oai_dc<br>\n </td>\n </tr>\n <th>spelling</th>\n <td>\n unylib-00000463182016-08-08 00:00:00Perpustakaan UAJYBuild your blogger XML templateSya,ban, WahyuWEB PROGRAMMINGAndi2010TEXT006.76 Sya b<br>\n </td>\n </tr>\n <th>institution</th>\n <td>\n Perpustakaan UAJY<br>\n </td>\n </tr>\n <th>collection</th>\n <td>\n Perpustakaan Yogyakarta<br>\n </td>\n </tr>\n <th>topic</th>\n <td>\n WEB PROGRAMMING<br>\n </td>\n </tr>\n <th>spellingShingle</th>\n <td>\n WEB PROGRAMMING<br>\n </td>\n </tr>\n <th>description</th>\n <td>\n Sya,ban, Wahyu<br>\n Build your blogger XML template<br>\n </td>\n </tr>\n </table>

```

Gambar 4.31 Testing Endpoint API tahun buku

Gambar 4.31 merupakan pengujian API dengan endpoint `/api/buku/tahun/:tahun`. Setelah mengirim request, kemudian REST server memberi response bahwa request berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk meminta request data buku berdasarkan tahun dari buku yang sedang dicari.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:7000/api/buku/topik/web
- Status:** 200 OK, Time: 296 ms, Size: 14.81 KB
- Response Body (XML):**

```

1 [{"id":22,"judul":"Build your blogger XML template","author":"Sya,ban, Wahyu","url":"http://search.jogjalib.com/Record/unylib-0000046318","detail":":\n <tr>\n <th>id</th>\n <td>\n unylib-0000046318<br>\n </td>\n </tr>\n <th>recordtype</th>\n <td>\n oai_dc<br>\n </td>\n </tr>\n <th>spelling</th>\n <td>\n unylib-00000463182016-08-08 00:00:00Perpustakaan UAJYBuild your blogger XML templateSya,ban, WahyuWEB PROGRAMMINGAndi2010TEXT006.76 Sya b<br>\n </td>\n </tr>\n <th>institution</th>\n <td>\n Perpustakaan UAJY<br>\n </td>\n </tr>\n <th>collection</th>\n <td>\n Perpustakaan Yogyakarta<br>\n </td>\n </tr>\n <th>topic</th>\n <td>\n WEB PROGRAMMING<br>\n </td>\n </tr>\n <th>spellingShingle</th>\n <td>\n WEB PROGRAMMING<br>\n </td>\n </tr>\n <th>description</th>\n <td>\n Sya,ban, Wahyu<br>\n Build your blogger XML template<br>\n </td>\n </tr>\n </table>

```

Gambar 4.32 Testing Endpoint API topik buku

Gambar 4.32 merupakan pengujian *API* dengan *endpoint* `/api/buku/topik/:topik`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. *Endpoint* ini digunakan untuk meminta *request* data buku berdasarkan topik dari buku yang akan dicari.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** `http://127.0.0.1:7000/api/buku/totalcari?keyword=ekonomi`
- Params:** Authorization, Headers (8), Body, Pre-request Script, Tests, Settings, Cookies
- Query Params:**

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> keyword	ekonomi			
Key	Value	Description		
- Body:** Cookies, Headers (10), Test Results
- Status:** 200 OK, Time: 1782 ms, Size: 316.02 KB
- Response Body (JSON):**

```
1 [{"id":1224,"judul":"Program komputer untuk analisa ekonomi","author":"Jogiyanto H.M.,"url":"http://search.jogjalib.com/Record/usdlib-3765","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":1279,"judul":"Era ekonomi digital : peruntungan dan mala petaka di abad networked intelligence","author":"Suharsono","url":"http://search.jogjalib.com/Record/usdlib-84853","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":1500,"judul":"Aplikasi SPSS untuk statistik ekonomi dan bisnis","author":"Sunnyoto, Danang","url":"http://search.jogjalib.com/Record/usdlib-111733","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":2130,"judul":"Analisis data ekonomi dengan menggunakan SPSS","author":"Sunnyoto, Danang","url":"http://search.jogjalib.com/Record/usdlib-117387","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":2305,"judul":"Structural equation model SEM dalam riset ekonomi : menggunakan LISREL [Ed.1]","author":"Sarwono, Jonathan","url":"http://search.jogjalib.com/Record/usdlib-90886","detail":"","tahun":0,"publisher":"","koleksi":"","topik":"","institusi":""}, {"id":2538,"judul":"Multivariate analysis Structural equation modelling [SEM] lisrel dan amos : aplikasi di manajemen,"}
```

Gambar 4.33 *Testing Endpoint API* totalcari buku

Gambar 4.33 merupakan pengujian *API* dengan *endpoint* `/api/buku/totalcari`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. *Endpoint* ini digunakan untuk mengirim *request* data buku berdasarkan parameter *keyword* yang dimasukkan. Pada *endpoint* ini semua data buku berdasarkan *keyword* yang dimasukkan.

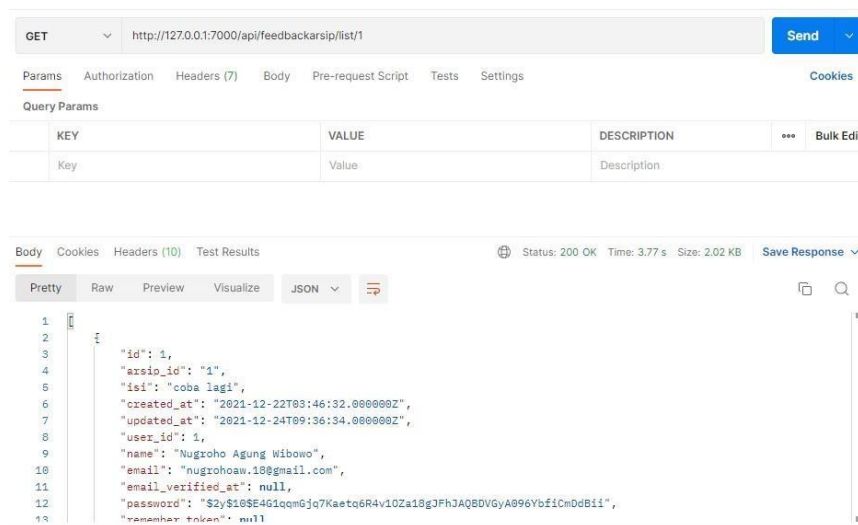
Berdasarkan pada pengujian yang telah dilakukan, penggunaan *API* relative efisien karena tidak boros dalam penggunaan *bandwidth* dan tidak memakan waktu yang lama saat pertukaran data. Untuk lebih jelasnya dapat dilihat Tabel 4.2 yang menunjukkan seluruh daftar *API* untuk *service* buku yang diuji beserta hasilnya.

Tabel 4.2 Tabel Daftar Pengujian *Service* Buku

Endpoint	Fitur	Hasil	Waktu (ms)	Bandwidth (kilobyte)

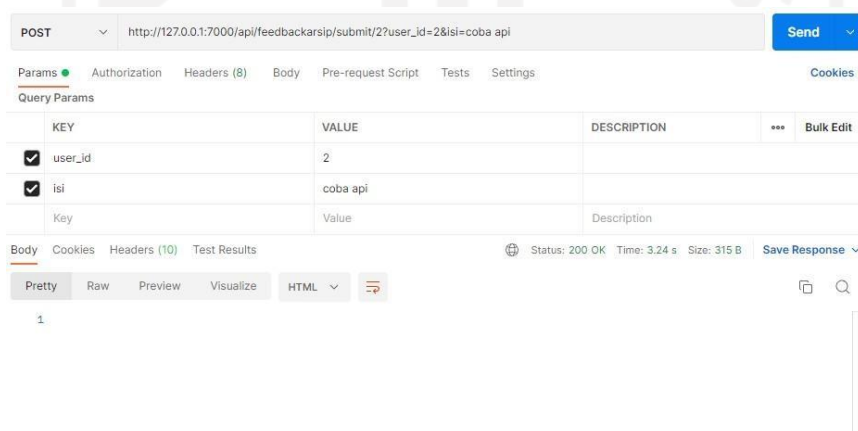
/api/buku/list	Menampilkan seluruh data buku	sukses	35790	3580
/api/buku/cari	Mengambil 10 data buku berdasarkan parameter <i>keyword</i> ""	sukses	5860	2,96
/api/buku/search	Mengambil data buku berdasarkan parameter <i>title</i> "" atau parameter <i>tahun</i> ""	sukses	419	3,85
/api/buku/detail/:id	Menampilkan detail buku berdasarkan <i>id</i>	sukses	491	0,521
/api/buku/tahun/:tahun	Menampilkan buku berdasarkan tahun	sukses	393	7,36
/api/buku/totalcari	Mengambil data buku berdasarkan <i>keyword</i> ""	sukses	1782	316,02
/api/buku/author/:author	Menampilkan data buku berdasarkan <i>author</i>	sukses	473	3,54
/api/buku/publisher/:publisher	Menampilkan data buku berdasarkan <i>publisher</i>	sukses	387	3,85
/api/buku/koleksi/:koleksi	Menampilkan data buku berdasarkan koleksi	sukses	361	14,81
/api/buku/topik/:topik	Menampilkan data buku berdasarkan topik	sukses	296	14,81

Pengujian API service feedback



Gambar 4.34 Testing Endpoint API list feedback arsip

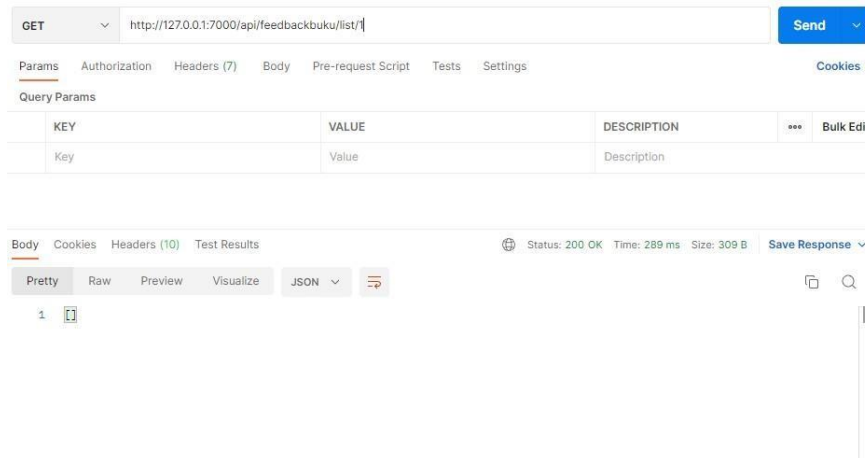
Gambar 4.34 merupakan pengujian API dengan endpoint `/api/feedbackarsip/list/:id`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk meminta *request* data *feedback* arsip berdasarkan *id* dari arsip yang dicari.



Gambar 4.35 Testing Endpoint API submit feedback arsip

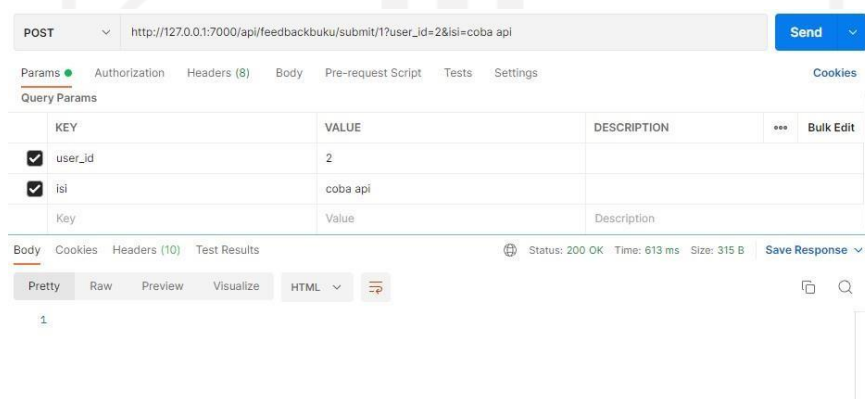
Gambar 4.35 merupakan pengujian API dengan endpoint `/api/feedbackarsip/submit/:id`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. Endpoint ini

digunakan untuk mengirim *request* data *feedback* arsip ke *database* berdasarkan parameter *user_id* dan isi yang dimasukkan.



Gambar 4.36 Testing Endpoint API list feedback buku

Gambar 4.36 merupakan pengujian API dengan endpoint `/api/feedbackbuku/list/:id`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk meminta *request* data *feedback* buku berdasarkan *id* dari buku yang dicari.



Gambar 4.37 Testing Endpoint API submit feedback buku

Gambar 4.37 merupakan pengujian API dengan endpoint `/api/feedbackbuku/submit/:id`. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status code 200. Endpoint ini digunakan untuk mengirim *request* data *feedback* buku ke *database* berdasarkan parameter *user_id* dan isi yang dimasukkan.

Tabel 4.3 Tabel Daftar Pengujian *Service Feedback*

Endpoint	Fitur	Hasil	Waktu (ms)	Bandwidth (kilobyte)
/api/feedbackarsip/list/:id	Menampilkan data <i>feedback</i> arsip berdasarkan id arsip	sukses	3770	2.02
/api/feedbackarsip/submit/:id	Mengirim data <i>feedback</i> arsip ke <i>database</i> untuk disimpan, dengan parameter yang telah dimasukkan	sukses	3240	0,315
/api/feedbackbuku/list/:id	Menampilkan data <i>feedback</i> buku berdasarkan id buku	sukses	289	0,309
/api/feedbackbuku/submit/:id	Mengirim data <i>feedback</i> buku ke <i>database</i> untuk disimpan, dengan parameter yang telah dimasukkan	sukses	613	0,315

Pengujian *API service user*

Pengujian *API* dengan *endpoint* /api/user/dologin dapat dilihat pada Gambar 4.38. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status *code* 200. *Endpoint* ini digunakan untuk mengirim *request* data *user* ke *database* ketika *user* sedang *login*.

Pengujian *API* dengan *endpoint* /api/user/email/:email dapat dilihat pada Gambar 4.39. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status *code* 200. *Endpoint* ini digunakan untuk mengirim *request* data *user* ke *database* untuk mengecek *user* dengan *email* yang sesuai dengan parameter ada atau tidak.

Pengujian *API* dengan *endpoint* /api/user/store dapat dilihat pada Gambar 4.40. Setelah mengirim *request*, kemudian *REST server* memberi *response* bahwa *request* berhasil yang ditandai dengan status *code* 200. *Endpoint* ini digunakan untuk mengirim *request* data *user* ke *database* untuk mengecek *user* dengan *email* yang sesuai dengan data yang dimasukkan ada atau tidak. Jika data tidak ada maka *database* akan menyimpan data tersebut.

POST ▼ http://127.0.0.1:7000/api/user/dologin?email=postman@example.com&password=cobapostman Send ▼

Params ● Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> email	postman@example.com			
<input checked="" type="checkbox"/> password	cobapostman			
Key	Value	Description		

Body Cookies Headers (10) Test Results ⊕ Status: 200 OK Time: 484 ms Size: 309 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 📄 🔍

1 📄

Gambar 4.38 Testing Endpoint API dologin user

POST ▼ http://127.0.0.1:7000/api/user/email?email=postman@example.com Send ▼

Params ● Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> email	postman@example.com			
Key	Value	Description		

Body Cookies Headers (10) Test Results ⊕ Status: 200 OK Time: 370 ms Size: 487 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 📄 🔍

```

1  {
2    "id": 7,
3    "name": "postman",
4    "email": "postman@example.com",
5    "email_verified_at": null,
6    "avatar": null,
7    "created_at": "2022-05-31T06:23:21.000000Z",
8    "updated_at": "2022-05-31T06:23:21.000000Z"
9  }

```

Gambar 4.39 Testing Endpoint API email user

POST ▼ http://127.0.0.1:7000/api/user/store?email=postman@example.com&password=cobapostman&name=postman Send ▼

Params ● Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> email	postman@example.com			
<input checked="" type="checkbox"/> password	cobapostman			
<input checked="" type="checkbox"/> name	postman			
Key	Value	Description		

Body Cookies Headers (10) Test Results ⊕ Status: 200 OK Time: 1146 ms Size: 315 B Save Response ▼

Pretty Raw Preview Visualize HTML ▼ 📄 🔍

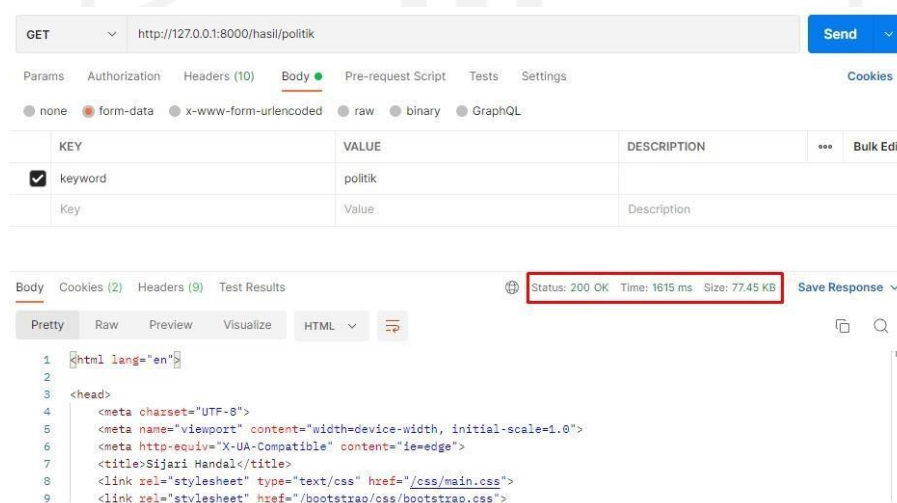
1 📄

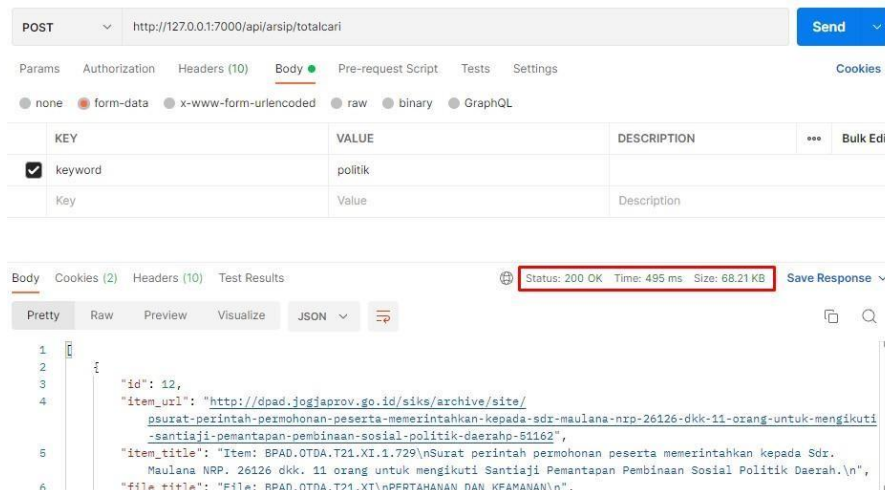
Gambar 4.40 Testing Endpoint API store email user

Tabel 4.4 Tabel Daftar Pengujian *Service User*

Endpoint	Fitur	Hasil	Waktu (ms)	Bandwidth (kilobyte)
/api/user/dologin	Mengirim data user berdasarkan email user yang dimasukkan apakah user tersebut sedang login atau tidak	sukses	484	0,309
/api/user/email/:email	Mengirim data user ke <i>database</i> untuk dicek apakah user tersebut ada atau tidak	sukses	370	0,487
/api/user/store	Mengirim data user untuk kemudian disimpan ke <i>database</i>	sukses	1146	0,315

Perbandingan menggunakan API dengan tidak menggunakan API

Gambar 4.41 *Test tanpa API* pada fitur pencarian



Gambar 4.42 Test API fitur pencarian

Pada Gambar 4.41 dapat dilihat waktu yang dihasilkan untuk mencari *keyword* “politik” adalah 1615 ms dengan *bandwidth* sebesar 77,45 KB. Jika dibandingkan dengan Gambar 4.42 yang mana pencarian dengan *keyword* “politik” menghasilkan waktu yang lebih singkat sebesar 495 ms dengan *bandwidth* sebesar 68,21 KB. Jadi dapat disimpulkan apabila tidak dibuatkan *api* maka waktu yang dibutuhkan *relative* lebih lama serta *bandwith* yang dihasilkan juga lebih besar.

Berdasarkan hasil dan pembahasan diatas ada kelebihan dan kekurangan. Kelebihan dari menggunakan *REST API* adalah *API* dapat bekerja secara efisien karena tidak boros dalam penggunaan *bandwidth* dan waktu yang dibutuhkan relatif singkat dalam proses pertukaran data. Selain itu, kelebihan pengujian dalam menggunakan *unit testing* adalah waktu yang dibutuhkan dalam melakukan pengujian relatif singkat karena hanya melakukan pengujian pada beberapa fungsi saja.

Adapun kekurangan dari SiJARI HanDAL adalah masih banyak data atau informasi yang tidak ditampilkan pada sistem yang telah dibuat yang disebabkan oleh tidak sempurnanya dalam melakukan proses *scraping* sehingga data yang disimpan ke *database* banyak yang kosong. Adapun kendala dalam melakukan proses *scraping* adalah koneksi jaringan yang terputus secara tiba-tiba dan *website* yang dituju mengalami *down*.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan dari hasil dan pembahasan yang dilakukan dapat disimpulkan bahwa:

- a. Penggunaan *SOA* dan *RESTful API* pada SiJARI HanDAL akan mempermudah pencarian data-data arsip dan buku milik Dinas Perpustakaan dan Arsip Daerah (DPAD) Daerah Istimewa Yogyakarta. Kelebihan penggunaan *SOA* dan *RESTful API* dapat mempermudah pertukaran data dari berbagai macam bahasa pemrograman. *REST API* tidak boros dalam penggunaan *bandwidth* dan hemat waktu dalam proses pertukaran data sehingga lebih efektif dan efisien.
- b. Pengembangan sistem SiJARI HanDAL dengan metode *SOA* dapat dilakukan dengan menggunakan konsep *Restful API*.

5.2 Saran

Berdasarkan implementasi dan pengujian terhadap sistem SiJARI HanDAL, menurut penulis sistem tersebut masih memiliki kekurangan, sehingga perlu dikembangkan lebih lanjut. Sebaiknya SiJARI HanDAL menggunakan *database* original agar informasi dari arsip maupun buku yang ditampilkan berupa informasi yang *realtime* dan *up to date*.

DAFTAR PUSTAKA

- Chen, X., Ji, Z., Fan, Y., & Zhan, Y. (2017). Restful API Architecture Based on Laravel Framework. *Journal of Physics: Conference Series*, 1.
- Flores, V. A., Permatasari, P. A., & Jasa, L. (2020). Penerapan Web Scraping Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan Keyword. *Majalah Ilmiah Teknologi Elektro*, vol.19.
- Hananta, J. S. (2013). Pendekatan Service Oriented Architecture (SOA) pada Pelaksanaan E-Government di Kementerian Hukum dan HAM RI. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 255.
- Josi, A., Abdillah, L. A., & Suryayusra. (2014). Penerapan Teknik Web Scraping pada Mesin Pencari Artikel Ilmiah. *Computer Science*, 159-162.
- Kurniawan, R. (t.thn.). Pembangunan Class Library Pada Domain Recipes Apps Berbasis Windows Phone Apps. *KOMPUTA*, 2.
- Rahmayu, M. (2016). Rancang Bangun Sistem Informasi Pada Rumah Sakit Dengan Layanan Intranet Menggunakan Metode Waterfall. *Jurnal Evolusi*, 34-35.
- Satriajati, S., Panuntun, S. B., & Pramana, S. (2020). Implementasi WEB Scraping Dalam Pengumpulan Berita Kriminal pada Masa Pandemi COVID-19. *Seminar Nasional Official Statistic 2020: Statistic in the New Normal: A Challenge of Big Data and Official Statistic*, 301-303.
- Simatupang, J., & Sianturi, S. (2019). PERANCANGAN SISTEM INFORMASI PEMESANAN TIKET BUS PADA PO. HANDOYO BERBASIS ONLINE. *Intra-Tech*, 16-18.
- Slameto, A. A. (2015). INTEGRASI SISTEM INFORMASI LABORATORIUM DENGAN MENGGUNAKAN PENDEKATAN SERVICE ORIENTED ARCHITECTURE (SOA). *Jurnal Ilmiah DASi*, 18 -26.
- Susanto, R., & Andriana, A. D. (2016). Perbandingan Model Waterfall dan Prototyping Untuk Pengembangan Sistem Informasi. *Majalah Ilmiah UNIKOM*, 2-4.

- Sutrisno, Assyidiq, M., & Santoso, S. (t.thn.). Perancangan Sistem Pemasangan Iklan Online pada Aplikasi E-Commerce (E-Gemanusa) Menggunakan Metode Restful API dan Framework Laravel. 122.
- Tabrani, M., Suhardi, & Priyandaru, H. (2021). SISTEM INFORMASI MANAJEMEN BERBASIS WEBSITE PADA UNL STUDIO. *JURNAL ILMIAH M-PROGRESS*, 14-15.
- Warkim, & Sensuse, D. I. (2017). Model Integrasi Sistem dengan Pendekatan Metode Service Oriented Architecture dan Model View Controller pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia. *Jurnal Teknik Informatika dan Sistem Informasi*, Volume 3.



LAMPIRAN

