



UNIVERSITAS NUSANTARA PGRI KEDIRI
FAKULTAS TEKNIK DAN ILMU KOMPUTER

Program Studi : *Teknik Mesin, Teknik Elektronika, Teknik Industri,*
Teknik Informatika, Sistem Informasi

Alamat : Kampus II, Mojoroto Gang I No. 6 Kediri 64112
Website: www.ft.unpkediri.ac.id E-mail: ft@unpkediri.ac.id

SURAT KETERANGAN BEBAS PLAGIASI

Nomor: 0395/FTIK-UN PGRI Kd/C/VI/2024

Gugus Penjamin Mutu Fakultas Teknik dan Ilmu Komputer Universitas Nusantara PGRI Kediri menyatakan bahwa Skripsi/Tugas Akhir:

Nama : MUHAMMAD FIKRI PRATAMA
NPM : 2013030009
Judul : Rancang Bangun Sistem Integrasi Data Berbasis RESTful API
Program studi : Sistem Informasi
Fakultas : Fakultas Teknik Ilmu dan Ilmu Komputer

telah dideteksi tingkat plagiasinya dengan kriteria toleransi $\leq 30\%$ dan dinyatakan bebas dari plagiasi (Rincian hasil plagiasi terlampir)

Demikian surat ini dibuat untuk digunakan sebagaimana mestinya.



Mengetahui:
Dekan FTIK,

Dr. Sulistiono, M.Si.
NIDN: 0007076801

Kediri, 24 Juni 2024

Gugus Penjamin Mutu,

Dr. Risky Aswi Ramadhani, M.Kom.
NIDN: 0708049001

fikri cek plagiasi prodi.docx

by Sniftyska.Edu

Submission date: 24-Jun-2024 09:38PM (UTC+0900)

Submission ID: 2407852812

File name: fikri_cek_plagiasi_prodi.docx (2.4M)

Word count: 8420

Character count: 53810

PENDAHULUAN**1.1 Latar Belakang**

Dengan pesatnya kemajuan teknologi informasi, pentingnya data dalam konteks institusi pemerintah menjadi semakin tidak dapat dihindari. Data dan informasi merupakan elemen penting dalam menentukan keberhasilan suatu entitas di instansi pemerintah, terutama dalam era persaingan digital saat ini. Setiap kegiatan yang dilakukan oleh instansi pemerintah menghasilkan data yang menjadi bukti fisik pelaksanaan kegiatan.

Meskipun data tersebut tersedia dalam sistem instansi pemerintah, tantangan muncul ketika data tersebut perlu diproses. Proses pencarian menjadi sulit, dan banyak waktu yang terbuang karena, banyaknya data yang dibatasi perangkat teknologi. Pengelolaan data dan informasi merupakan aspek kunci dalam menentukan akuntabilitas dalam setiap kegiatan yang dilakukan.

Untuk mengatasi permasalahan tersebut, pemerintah mengeluarkan kebijakan melalui Peraturan Presiden No. 39 tahun 2019 tentang Satu Data In. Kebijakan ini bertujuan untuk mengatur tata kelola data yang dihasilkan oleh instansi pemerintah, baik pusat maupun daerah, untuk mendukung perencanaan, pelaksanaan, evaluasi, dan pengendalian pembangunan. Satu Data Indonesia menekankan pentingnya data yang akurat, mutakhir, terpadu, mudah diakses, dan dapat dibagikan antar sistem.

Salah satu cara integrasi data adalah menggunakan *RESTful API*. Konsep *RESTful API* menjadi pilihan dalam mewujudkan integrasi data yang efisien. Penerapannya memudahkan komunikasi data antar sistem, bahkan jika berbeda sistem operasi atau bahasa pemrograman. Diharapkan bahwa perancangan pusat informasi ini akan memberikan kemudahan dalam mengelola informasi yang diperlukan, memastikan ketersediaan data yang jelas, dan memenuhi kebutuhan data publik bagi masyarakat.

1.2 Identifikasi Masalah

Permasalahan yang terjadi antara lain :

1. Data belum terintegrasi pada sistem informasi masing-masing organisasi.
2. Pemborosan waktu dan sumber daya ketika ingin mencari data.

1.3 Rumusan Masalah

Bagaimana merancang sebuah sistem informasi yang mampu mengintegrasikan untuk mendapatkan data dari berbagai sistem informasi yang berbeda.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. Aplikasi satu data berbasis *web*.
2. Sistem integrasi dan komunikasi data menggunakan *RESTful API*.

1.5 Tujuan Penelitian

Tujuan penelitian ini adalah :

1. Mengintegrasikan data antar sistem informasi.

⁵¹ 1.6 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini adalah :

- ²⁸ 1. Manfaat bagi peneliti : Sebagai salah satu cara untuk mengaplikasikan ilmu yang didapatkan untuk membuktikan dalam sebuah penelitian. Selain itu, juga sebagai pengalaman diri untuk melakukan penelitian.
2. Manfaat bagi mitra : Mitra akan lebih cepat dalam mengambil keputusan karena sistem sudah terintegrasi sehingga kebutuhan data lebih efisien.

³ 1.7 Sistematika Penulisan

BAB I: Pendahuluan

Pada bagian ini, akan dibahas mengenai latar belakang, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

BAB II: Landasan Teori

Dalam bab ini, akan dibahas mengenai berbagai teori yang mendukung serta menjadi dasar dari penelitian yang sedang dilakukan

² BAB III: Metodologi Penelitian

Bagian ini akan menjelaskan tahapan pengembangan sistem informasi sesuai dengan metode pengembangan aplikasi yang telah dipilih untuk penelitian ini.

² BAB IV : Desain Sistem

Bab ini akan membahas tentang tinjauan lokasi, yang mencakup deskripsi lengkap lokasi penelitian, termasuk ⁸¹ visi dan misi, serta struktur organisasinya. Selain itu, akan dilakukan analisis proses bisnis dengan merancang arsitektur sistem

yang akan diterapkan dalam penelitian ini. Kemudian, akan dibahas pemodelan data dan proses untuk menggambarkan secara detail bagaimana sistem akan beroperasi.

Selanjutnya, akan dipaparkan desain *database* yang digunakan, termasuk nama tabel, tipe data, dan panjang data yang relevan. Terakhir, akan dijelaskan desain tampilan pengguna yang mencakup gambaran visual dari sistem yang akan dibuat, termasuk form input, tampilan output, laporan, dan pengalaman pengguna.

BAB V : Implementasi dan Pengujian

Bagian ini akan menjelaskan desain sistem yang telah direncanakan sebelumnya, serta hasil dari desain tersebut. Selanjutnya, akan dipaparkan tampilan dari sistem yang telah dirancang, serta laporan dari program yang mencakup implementasi dari desain user interface yang telah disusun sebelumnya, beserta beberapa potongan kode penting atau inti dari program beserta penjelasannya. Terakhir, akan dilakukan pengujian terhadap sistem untuk menilai apakah sudah layak digunakan oleh pengguna.

Bab VI : Penutup

Bab ini membahas mengenai kesimpulan dan saran yang berisi rangkuman dari hasil penelitian.

TINJAUAN PUSTAKA**2.1 Kajian Teori****2.1.1 Sistem Informasi**

27

Sistem informasi adalah sistem dalam suatu organisasi yang bertujuan untuk memenuhi kebutuhan pengolahan transaksi harian, mendukung operasi, kegiatan manajerial, dan strategi organisasi, serta menyediakan laporan-laporan yang diperlukan untuk pihak luar tertentu (Kasiana dkk., 2023).

10

2.1.2 Waterfall

Menurut Sholikhah, Sairan, dan Syamsiah (2017:47), model Waterfall merupakan model klasik dalam pengembangan perangkat lunak yang bersifat berurutan. Metode ini menggambarkan pendekatan yang sistematis dan berurutan dalam mengembangkan perangkat lunak. Tahapan-tahapan dalam metode Waterfall dimulai dari spesifikasi kebutuhan pengguna, kemudian dilanjutkan dengan tahapan perencanaan, pemodelan, konstruksi, pengujian sistem, penyerahan sistem kepada pengguna, dan dukungan pada perangkat lunak yang telah dihasilkan (Kurniawan dkk., 2020).

2.1.3 RESTful API

10

RESTful API adalah antarmuka yang memungkinkan dua sistem komputer bertukar informasi secara aman melalui internet. Sebagian besar aplikasi bisnis

perlu berkomunikasi dengan aplikasi internal maupun pihak ketiga untuk menyelesaikan berbagai tugas.

2.1.4 **JSON**

¹ *JSON* merupakan format pertukaran data yang menggunakan tipe data ringan berbasis pasangan *key-value*. *JSON*, singkatan dari *JavaScript Object Notation*, memiliki format teks yang mudah dibaca oleh manusia dan digunakan untuk merepresentasikan struktur data sederhana serta *array asosiatif* (disebut objek). Format *JSON* sering digunakan untuk mentransmisikan data terstruktur melalui koneksi jaringan dalam proses yang dikenal sebagai serialisasi (Roihan dkk., 2019).

2.1.5 **PHP**

¹³ *PHP*, yang merupakan singkatan dari *PHP: Hypertext Preprocessor*, adalah bahasa pemrograman yang digunakan untuk mengembangkan *website* dinamis. Berbeda dengan *HTML* yang hanya dapat menampilkan konten statis, *PHP* memiliki kemampuan untuk berinteraksi dengan *database*, *file*, dan *folder*, sehingga memungkinkan *PHP* untuk menampilkan konten yang dinamis dari sebuah *website* (Firliana & Rhohman, 2019).

2.1.6 **Black Box**

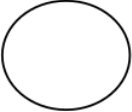

Metode *Blackbox testing* adalah pendekatan pengujian yang dilakukan tanpa harus mengetahui detail internal dari aplikasi, seperti *source code*. Dalam *Blackbox testing*, fokus utama adalah pada hasil yang dihasilkan dari *input* yang diberikan, tanpa memperhatikan bagaimana aplikasi mencapai hasil tersebut.

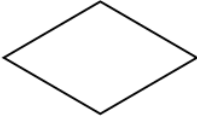

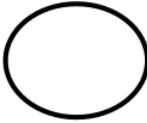
Proses *Blackbox testing* melibatkan pengujian aplikasi dengan berbagai input yang berbeda. Tujuan utama dari pengujian ini adalah untuk menentukan apakah aplikasi beroperasi sesuai dengan kebutuhan yang diinginkan oleh stakeholder. *Blackbox testing* juga dikenal sebagai pengujian berbasis fungsional atau spesifikasi, karena tidak melibatkan analisis atau pemeriksaan langsung terhadap *source code program* (Sasongko dkk., 2021).

2.1.7 Flowchart

Flowchart, yang juga dikenal sebagai bagan alir, adalah teknik analisis visual yang digunakan untuk menjelaskan berbagai aspek dari sistem informasi dengan jelas, singkat, dan logis. *Flowchart* digunakan untuk merekam cara proses bisnis dilakukan dan bagaimana dokumen mengalir melalui suatu organisasi. Ini merupakan representasi visual dari sistem, prosedur, dan pengendalian internal yang telah diterapkan oleh perusahaan. Menurut Indrajani (2011), *flowchart* adalah penggambaran grafis dari langkah-langkah dan urutan prosedur suatu program, yang seringkali memudahkan penyelesaian masalah yang memerlukan analisis dan evaluasi lebih lanjut (Tuasamu et al., 2023).

Tabel 2. 1 Notasi *Flowchart*

Simbol	Nama Simbol	Keterangan
	Start Event	Simbol yang digunakan untuk menunjukkan memulai kegiatan.
	Activity / taks	Kegiatan yang dilakukan dalam suatu area bisnis.

Simbol	Nama Simbol	Keterangan
	Gateway	Cabang yang menyatakan lebih dari satu syarat.
	Sequence flow	Menunjukkan aliran data dari satu simbol ke simbol lainnya.
	End Event	Simbol yang digunakan untuk menunjukkan mengakhiri kegiatan.

2.1.8 Desain Sistem

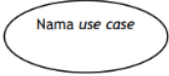
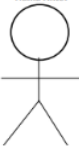


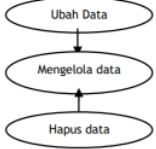


Menurut Mathiassen (2000, p197), Desain arsitektur sistem / *Architectural design* memiliki tujuan untuk menstrukturkan suatu sistem yang menggunakan komputerisasi. Desain ini sangat mempengaruhi dalam seberapa baik suatu sistem, seperti dalam kecepatan, keamanan, dan kemudahan dalam dimodifikasi. Desain arsitektur sistem ini menggambarkan rancangan mengenai keterkaitan antara software dan hardware serta alur komunikasi data yang digunakan untuk menjalankan sistem yang telah digunakan.

2.1.9 Use case

Diagram Kasus Penggunaan (*Use case Diagram*) adalah representasi visual dari perilaku sistem informasi yang sedang dibuat. Diagram ini memodelkan interaksi antara pengguna sistem dengan sistem itu sendiri melalui deskripsi naratif tentang bagaimana sistem tersebut digunakan (Setiawansyah dkk., 2022).

80

Tabel 2. 2 Notasi *Use case Diagram*


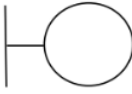
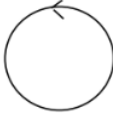
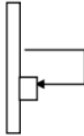


Simbol	Deskripsi
<p>Use Case</p> 	Merupakan unit pertukaran pesan melalui actor. (kata kerja)
<p>Actor</p> <p>Nama Aktor</p> 	Merupakan elemen proses yang terlibat dalam sebuah sistem actor belum tentu orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
<p>Asosiasi/association</p> 	Suatu proses Komunikasi antara actor & use case.
<p>Generalisasi/ generalization</p> 	<p>Hubungan secara umum & khusus dimana fungsi utama merupakan proses lebih dari fungsi lainnya, misalnya:</p> 
<p>Extend</p> <p>«extend»</p> 	Relasi use case tambahan ke sebuah use case dimana saat proses dapat berdiri sendiri
<p>Include</p> <p>«include»</p> 	Relasi antar 2 buah use case, dimana use case dasar akan ketergantungan terhadap yang lain

2.1.10 Sequence

Suatu diagram yang menggambarkan hubungan antara objek-objek serta durasi penggunaannya untuk menampilkan urutan pesan yang dikirim antara objek-objek tersebut, serta interaksi antara objek-objek yang terjadi pada titik tertentu dalam sistem, dikenal sebagai diagram urutan (*sequence diagram*). Berikut adalah tabel notasi sequence diagram (Endang Setyawati, t.t.).

41




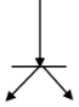

Tabel 2. 3 *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , gambaran sistem sebagai landasan dalam menyusun basis data.
	<i>Boundary Class</i> , menangani komunikasi antar lingkungan system
	<i>Control class</i> , bertanggung jawab terhadap kelas-kelas yang lain terhadap objek yang berisi logika
	<i>Recursive</i> , pesan buat dirinya
	<i>Activation</i> , mewakili proses durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , komponen yang digambarkan garis putus yang terhubung dengan objek

2.1.11 *Activity Diagram*

Activity diagram adalah model yang mengilustrasikan aktivitas dalam suatu sistem yang sedang beroperasi. Diagram ini berfungsi untuk menjelaskan aktivitas program tanpa menampilkan kode atau antarmuka pengguna. *Activity diagram* menggunakan berbagai simbol, di mana setiap simbol memiliki makna dan tujuan tertentu. Diagram ini terutama berfokus pada aktivitas dari sub-sistem tanpa perlu merinci detail internalnya (Viktoria, 2022).

Tabel 2. 4 Notasi Activity Diagram (Endang Setyawati, t.t.)

Gambar	Keterangan
	Mengawali proses pekerjaan
	Mengakhiri proses pekerjaan.
	Proses kegiatan berlangsung
	<i>Gambaran</i> menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) untuk menjelaskan dekomposisi.

2.1.12 ³ Class Diagram

Menggambarkan struktur statis class dalam sebuah sistem, *class diagram* mempresentasikan entitas yang ditangani oleh class sistem dan bagaimana class tersebut berhubungan dengan yang lain melalui berbagai cara, seperti *Associated* (sebuah class bergantung pada class lainnya), *Specialized* (sebuah class merupakan ³ *spesialisasi* dari class lainnya), atau *Package* (dikelompokkan bersama sebagai satu unit) (Viktoria, 2022).

2.1.13 ⁴³ Mysql

MySQL adalah sistem manajemen basis data yang memungkinkan ⁴⁷ pengelolaan kumpulan data secara cepat, dapat menampung jumlah informasi yang besar, dapat diakses oleh banyak klien secara simultan, dan mampu melakukan sinkronisasi data secara bersamaan (Nugroho dkk., 2022).

2.2 Kajian Hasil Penelitian Terdahulu

Terdapat penelitian sebelumnya yang dilakukan oleh (Baharuddin dkk., 2022) ³⁸ dengan judul IMPLEMENTASI *WEB SERVICE* DENGAN METODE *REST API* UNTUK INTEGRASI DATA *COVID 19* DI SULAWESI SELATAN (2022). Sistem ini dibuat untuk memantau perkembangan jumlah kasus *COVID-19* di Sulawesi Selatan. Pengembangan sistem menggunakan teknologi *RESTful* untuk pertukaran data dalam format *JSON* melalui *URI (Uniform Resource Identifier)*. Dalam penelitian ini, penulis berhasil membuat sebuah *website* terpusat dengan memanfaatkan layanan *web* berbasis metode *REST API* untuk mengambil data dari berbagai sumber *database* yang berbeda, dengan tujuan mengintegrasikan atau menggabungkan data tersebut menjadi satu kesatuan.

Selanjutnya ada penelitian dari yang (Ghozaly, 2019) ⁵⁶ yang berjudul IMPLEMENTASI *REST API* PADA PUSAT INFORMASI MAHASISWA UNIVERSITAS TEKNOLOGI YOGYAKARTA (2019). ¹² Situs *web* pusat informasi mahasiswa berbasis *web* adalah platform yang menyajikan berbagai informasi terkait kegiatan kampus. *Website* ini terhubung dengan layanan *web* melalui implementasi *REST API*, yang memungkinkan integrasi dengan sistem lain menggunakan berbagai bahasa pemrograman.

Berikutnya ada penelitian dari (Salim dkk., 2021) ⁷ dengan judul INTEGRASI SISTEM INFORMASI PEMANTAUAN KUALITAS LINGKUNGAN AIR DAN UDARA MENGGUNAKAN *REST API* DAN *WEB SERVICE* (2021). ⁷ Pusat Teknologi Lingkungan mengoperasikan tiga sistem informasi pemantauan kualitas lingkungan, yakni Onlimo untuk memantau kualitas air sungai, Sipaku untuk

memantau udara ambien, dan Sipegaruk untuk memantau gas rumah kaca. Rancangan yang direkomendasikan adalah dengan menerapkan REST API pada setiap sistem informasi tersebut, serta membuat layanan web untuk mengambil data dari REST API tersebut.

Yang keempat ada penelitian dari Fauzan Prasetyo Eka Putra yang berjudul Pengembangan Sistem Informasi Laboratorium Terintegrasi Sistem Akademik Menggunakan Agile Scrum. Perancangan yang diajukan mencakup implementasi REST API yang sudah ada pada Sistem Informasi Akademik (SIKAD). Pendekatan ini memungkinkan pemrosesan data mahasiswa dan mata kuliah praktikum secara real-time, serta secara otomatis menyesuaikan perubahan data dari sistem akademik. Metode ini akan mempermudah pengelolaan data (Prasetyo Eka Putra et al., n.d.).

Penelitian terkait selanjutnya berjudul "Rancangan Pembuatan API Website Data Tanaman Obat Dan Langka Kabupaten Kediri." Perancangan situs informasi tentang tanaman obat dan langka di Kabupaten Kediri dilakukan menggunakan metode Waterfall, yang meliputi tahapan identifikasi, analisis, perancangan, implementasi, dan pengujian. Situs ini menggunakan Application Programming Interface (API) berbasis Node.js dan Express.js sebagai perantara antara klien dan server. Pengujian endpoint API dilakukan menggunakan alat Postman, dengan hasil yang ditampilkan dalam format JSON untuk penyimpanan dan transfer data. Dengan fokus pada kemudahan akses dan kecepatan, tujuan situs ini adalah memberikan informasi tentang tanaman obat dan langka di Kabupaten Kediri secara efisien kepada pengguna (Busro dkk., 2022).

Yang terakhir ada penelitian dari (Feryzal Fahlevi & Anugrah,³⁷ 2021) Implementasi Integrasi Sistem Informasi Manajemen Rumah Sakit Dengan Sistem Informasi Laboratorium Di Rumah Sakit PKU Muhammadiyah Sekapuk¹⁸ (2021). membangun sebuah *REST API* yang dapat mengintegrasikan dua yaitu sistem informasi sistem informasi manajemen rumah sakit (SIMRS) dengan sistem informasi laboratorium (LIS). Dari aplikasi LIS ke SIMRS dan pihak laboratorium¹⁸ tinggal mencetak hasil dan memberikannya kepada pasien.

METODOLOGI PENELITIAN

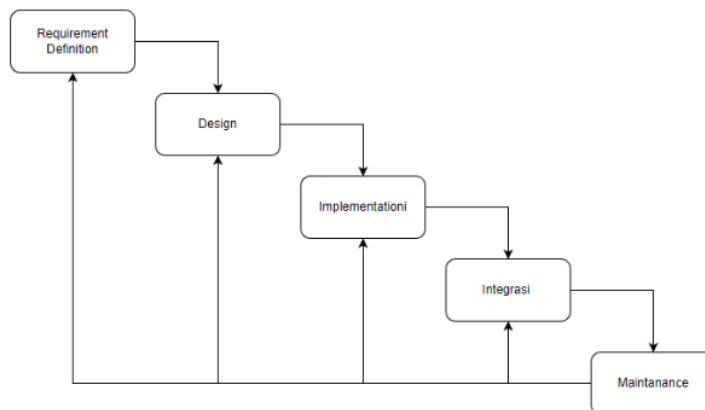
1.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan selama proses penelitian ini adalah sebagai berikut:

1. Wawancara untuk mendapatkan data yang diperlukan dalam sistem peneliti melakukan wawancara dengan Dinas komunikasi dan informatika Kabupaten Kediri.
2. Studi literatur untuk mendapatkan data-data yang bersifat teoritis maka dilakukan pengumpulan data dengan cara membaca dan mempelajari buku serta jurnal- jurnal yang membahas mengenai analisa perancangan sistem informasi.

1.2 Pengembangan Sistem

Dalam teknik penelitian ini menggunakan metode *waterfall* terdapat beberapa langkah pendekatan yaitu:



Gambar 3. 1 Pengembangan Sistem *Waterfall*

1. Requirement: Pengumpulan informasi tentang bagaimana satu data, proses bisnis yang terlibat, dan fitur apa yang harus dipenuhi oleh sistem informasi.
2. Design: perancangan sistem yang digunakan seperti pemodelan proses bisnis, perancangan antar muka pengguna, dan perancangan basis data yang efisien.
3. Implementation: pengembang melakukan *coding*, pengujian, dan integrasi komponen sistem untuk memastikan sistem berfungsi dengan baik.
4. Integrasi dan Penerapan : Ini melibatkan *instalasi* perangkat lunak yang diperlukan migrasi data, pelatihan pengguna, dan uji coba sistem sebelum digunakan secara penuh.
5. Maintenance : Pada tahap terakhir dalam Metode Waterfall, perangkat lunak yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan (Raffin dkk., 2022).

BAB IV

DESAIN SISTEM

4.1 Tinjauan Lokasi

Dinas Komunikasi dan Informatika memiliki tugas utama dalam melaksanakan kewenangan daerah di bidang pengelolaan Teknologi Informasi dan Komunikasi (TIK), serta menjalankan tugas pembantuan yang diberikan oleh pemerintah atau pemerintah Provinsi. Kegiatan dinas ini terfokus pada pengembangan dan pemeliharaan sistem informasi, jaringan komputer, produksi informasi, publikasi, dan pengembangan komunikasi publik.

Dalam lingkup Diskominfo Kabupaten Kediri, fokusnya terletak pada pengembangan teknologi informasi terutama terkait dengan situs *web* dan pengawasan kondisi jaringan dalam menerapkan teknologi informasi di setiap Satuan Kerja Perangkat Daerah (SKPD). Dinas ini memiliki tanggung jawab untuk memastikan optimalisasi pemanfaatan teknologi informasi di semua SKPD yang berada di wilayah Kabupaten Kediri. Saat ini, kantor Diskominfo Kabupaten Kediri berlokasi di Jalan Sekartaji No 02 Doko, Ngasem, Kabupaten Kediri.

4.1.1 Visi dan Misi

Dinas Komunikasi dan Informatika Kabupaten Kediri

Visi :

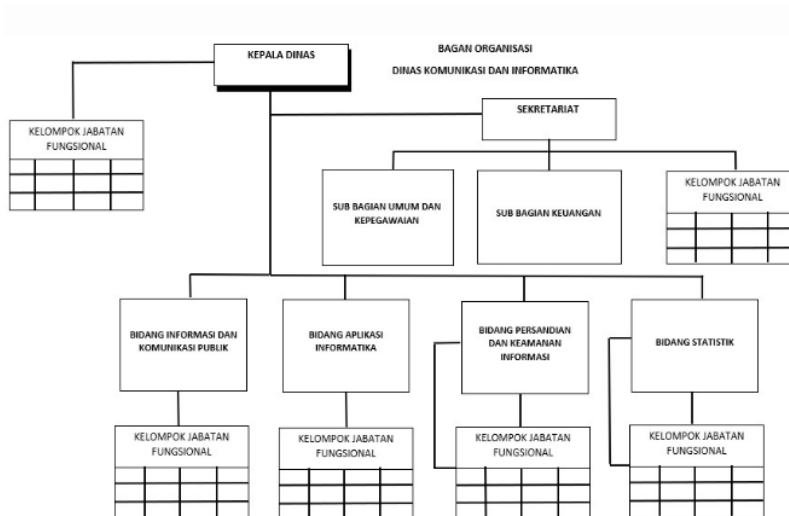
Menjadi Pusat Layanan Informasi dan TIK di Kabupaten Kediri

Misi :

1. Pengelolaan dan pelayanan informasi publik yang transparan, mudah diakses dan professional.
2. Peningkatan kinerja dan integritas aparatur pemerintah melalui pemenuhan saranadan prasarana TIK.
3. Mengembangkan infrastruktur TIK yang handal dan terpadu.
4. Memberdayakan potensi masyarakat melalui pemanfaatan teknologi informasi dankomunikasi.

69

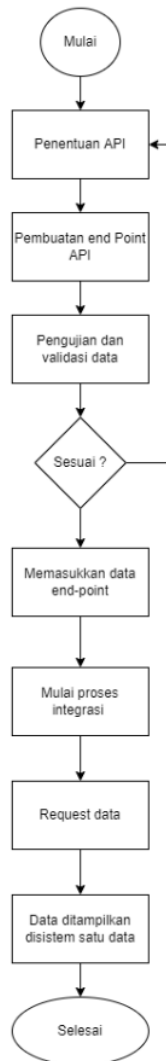
4.1.2 Struktur Organisasi



Gambar 4. 1 Bagan Organisasi

4.2 Alur Proses Sistem

Dari proses yang ingin diimplementasikan maka didapatkan alur flowchart yang seperti berikut ini :



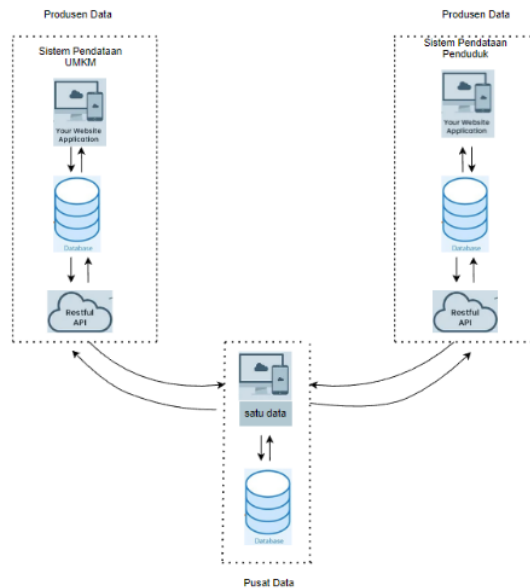
Gambar 4. 2 Alur Proses Sistem

Dari gambar 4.2 *flowchart* diatas, langkah-langkah untuk membangun sistem integrasi data dapat dijelaskan sebagai berikut:

1. Menentukan *API* yang akan diintegrasikan ke dalam sistem. Ini merupakan langkah awal untuk mengidentifikasi jenis data yang akan dimasukkan ke dalam sistem integrasi.
2. Membuat *endpoint API* untuk mengambil data dari sistem yang akan diintegrasikan. Ini merupakan langkah penting untuk menghubungkan sistem-sistem yang berbeda agar dapat saling berkomunikasi dan berbagi data.
3. Melakukan pengujian dan validasi data untuk memastikan bahwa data yang diambil sudah sesuai dengan kebutuhan. Langkah ini penting untuk menjamin keakuratan dan keberlangsungan data yang diintegrasikan.
4. Memasukkan data *endpoint* dalam *database* sistem satu data. Setelah proses Pengujian dan validasi selesai, data akan dimasukkan ke dalam *database* untuk dapat diakses dan dikelola secara efisien.
5. Memulai proses integrasi data ke dalam sistem satu data. Langkah ini melibatkan penggabungan data dari berbagai sumber ke dalam satu sistem yang terintegrasi.
6. Sistem satu data secara otomatis akan melakukan *request endpoint API* untuk menampilkan data yang tersedia dari hasil *request API*. Jika ada, data tersebut akan ditampilkan di sistem satu data .

4.3 Desain Arsitektur Sistem

Dari analisis sistem yang telah di dapatkan maka terdapat rancangan desain arsitektur seperti berikut :



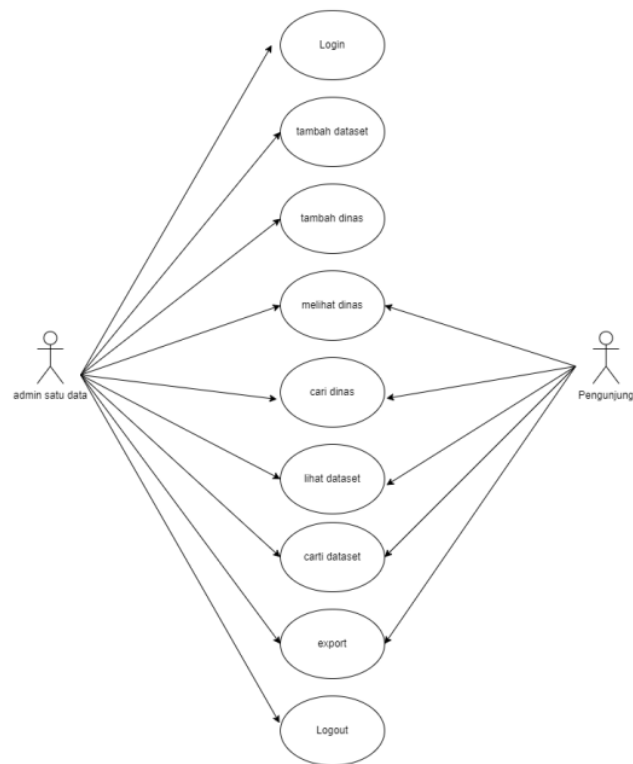
Gambar 4. 3 Arsitektur Sistem

Dari gambar 4.3 terdapat dua peran yaitu produsen data dan pusat data. Pusat data adalah unit yang melaksanakan kegiatan pengumpulan, dan pengelolaan data yang disampaikan oleh produsen data, serta menyebarluaskan Data. Produsen data adalah unit yang berfungsi untuk menghasilkan data. Sistem pendataan umkm dan sistem pendataan penduduk akan berperan sebagai produsen data sedangkan satu data akan berperan sebagai pusat data.

Proses Integrasi menggunakan metode *RESTfull API* untuk memfasilitasi komunikasi antara sistem satu data sebagai pusat data dengan sistem informasi yang berperan sebagai produsen data. *RESTful API* ini akan dijalankan oleh sistem satu data setiap kali data di request. *Endpoint API* akan berjalan melalui metode *HTTP GET*, dengan data yang dikirimkan dalam format *JSON* dan data akan di tampilkan dalam sistem satu data.

4.4 Pemodelan Data dan Proses

87 4.4.1 Use case Diagram (tugas masing masing aktor)



58

Gambar 4. 4 Use case Diagram

Pada Gambar 4.4 menyajikan *use case* sistem satu data antara Admin Satu Data dan Pengunjung yang terlibat dalam sistem ini. Penjelasan dari *use case* diagram pada Gambar 4.4 adalah sebagai berikut:

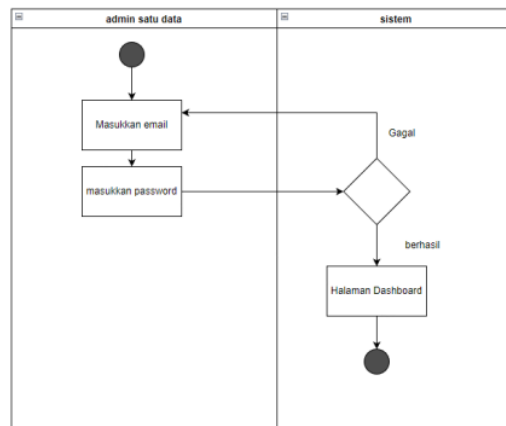
- a. Admin satu data : Merupakan Pengguna yang memiliki akses penuh di sistem satu data seperti *login*, tambah dinas, mencari dinas, menambah *dataset*, mencari *dataset*, mengekspor data dan *logout*.
- b. Pengunjung : Merupakan pengguna yang dapat melihat mencari dan mengekspor data

4.4.2 Activity diagram

Pada *activity diagram* ini memiliki 2 aktor yaitu admin satu data dan pengunjung.

- a. *Activity diagram* aktor admin satu data

1) *Activity diagram* login



Gambar 4. 5 Login Admin Satu Data

Pada Gambar 4.5 ditampilkan diagram aktivitas untuk proses *login* Admin Satu Data. Langkah-langkah yang dilakukan oleh admin untuk *login* adalah sebagai berikut: Admin memasukkan email dan password. Jika autentikasi berhasil, admin akan diarahkan ke halaman *dashboard*. Namun, jika autentikasi gagal, admin akan kembali ke halaman *login*.

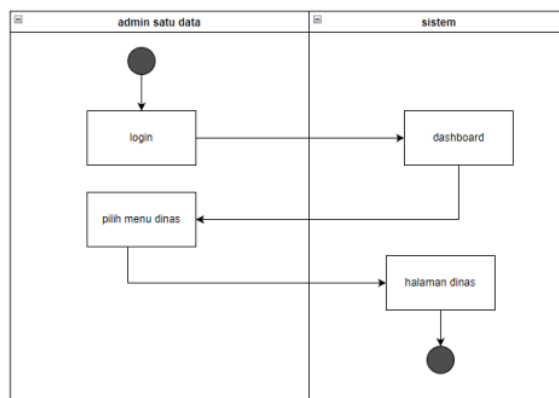
2) Tambah data dinas



Gambar 4. 6 Tambah Data Dinas

Pada Gambar 4.6 ditampilkan diagram aktivitas untuk proses penambahan data dinas di sistem satu data. Langkah pertama adalah *login* sebagai admin, lalu memilih menu dinas dan memilih opsi tambah data dinas. Admin kemudian mengisi nama, alamat, dan deskripsi dinas. Apabila proses berhasil, data akan disimpan. Namun, jika gagal, admin akan kembali ke halaman tambah data dinas.

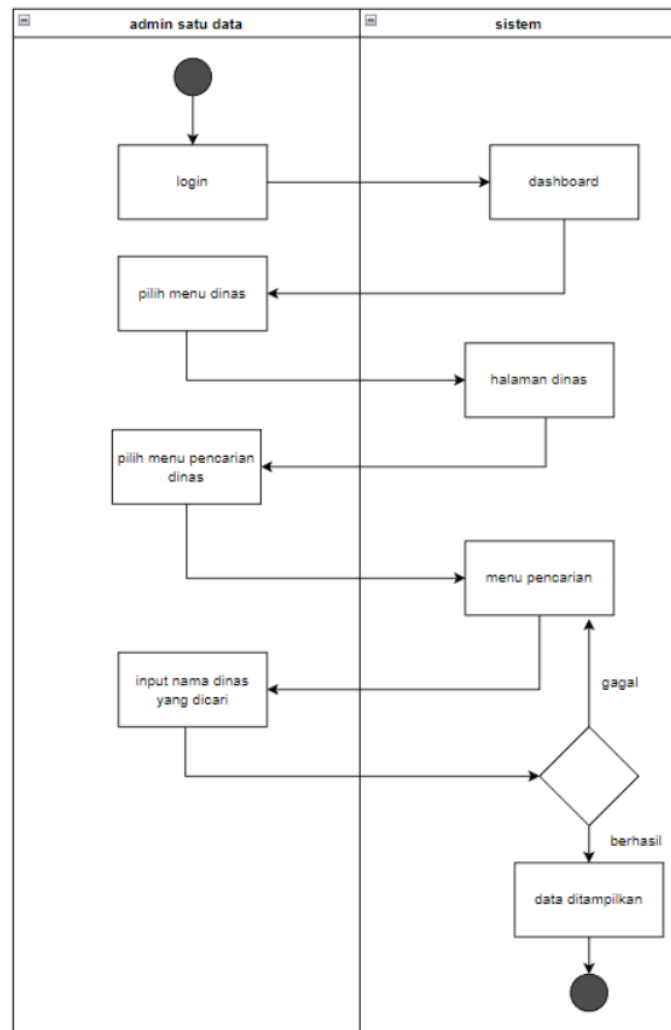
3) Melihat data dinas



5
Gambar 4. 7 Melihat Data Dinas

Pada Gambar 4.7 ditampilkan diagram aktivitas untuk proses melihat data dinas. Langkah-langkah yang dilakukan adalah sebagai berikut: admin *login*, kemudian memilih menu dinas, dan halaman dinas akan ditampilkan.

4) Mencari data dinas



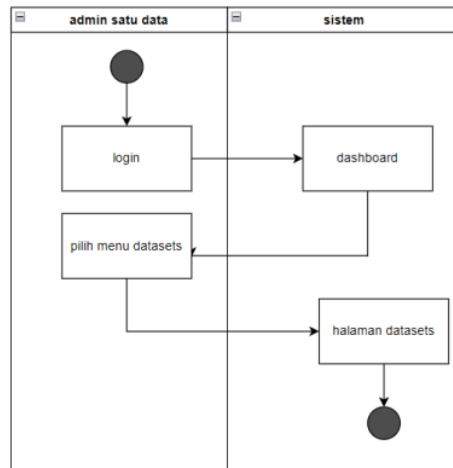
Gambar 4. 8 Mencari Data Dinas

Dalam Gambar 4.8, terdapat diagram aktivitas untuk pencarian data dinas. Langkah-langkahnya adalah sebagai berikut: admin melakukan *login*, memilih menu dinas, memilih opsi pencarian, menginput nama dinas yang ingin dicari. Jika data tersebut ada, maka akan ditampilkan; jika tidak, maka akan ada pesan yang menyatakan bahwa tidak ada data yang ditemukan.

5) Tambah *Dataset*

Di Gambar 4.9, terdapat diagram aktivitas untuk menambahkan *dataset* ke dalam sistem satu data. Langkah pertamanya adalah admin melakukan *login*, kemudian memilih menu *datasets* dan opsi tambah *dataset*. Admin kemudian mengisi nama *dataset*, dinas terkait, deskripsi *dataset*, *endpoint*, dan terakhir status *dataset*. Jika proses berhasil, data akan disimpan. Namun, jika gagal, admin akan kembali ke halaman tambah *dataset*.

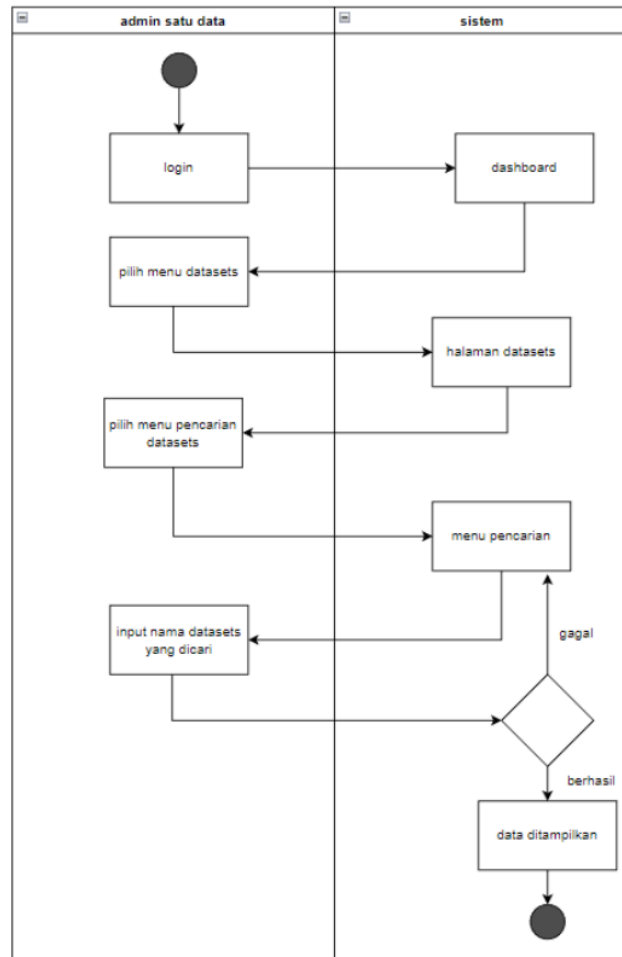
Gambar 4. 9 Menambah *Dataset*

6) Lihat *dataset*Gambar 4. 10 Lihat *Dataset*

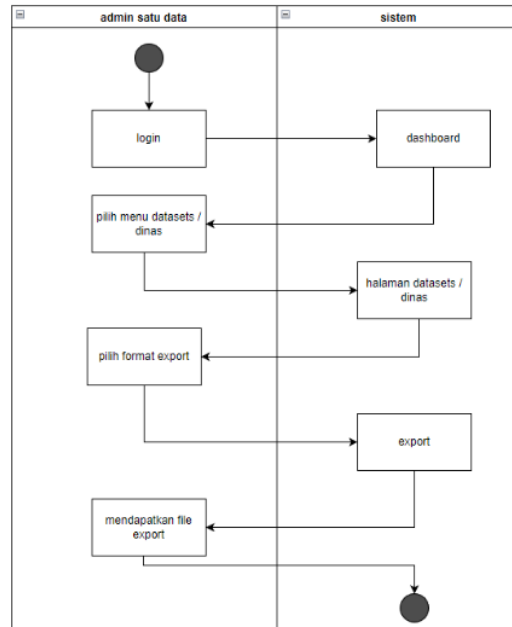
Dalam Gambar 4.10, terlihat diagram aktivitas untuk melihat data *dataset*. Langkah-langkahnya dimulai dengan admin melakukan *login*, lalu memilih opsi menu *dataset*, maka akan ditampilkan halaman *dataset*.

7) Pencarian *Dataset*

Di Gambar 4.11, terdapat diagram aktivitas untuk melakukan pencarian *dataset*. Langkah-langkahnya dimulai dengan admin melakukan *login*, memilih menu *datasets*, kemudian memilih opsi pencarian, dan menginput nama *dataset* yang ingin dicari. Jika *dataset* tersebut tersedia, maka akan ditampilkan; jika tidak, maka akan muncul pesan yang menyatakan bahwa tidak ada data yang ditemukan.

Gambar 4. 11 Pencarian *Dataset*

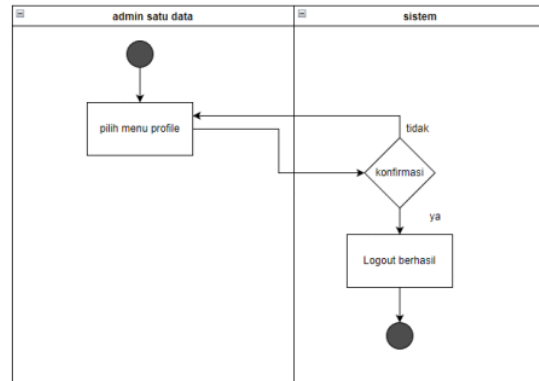
8) Export



Gambar 4. 12 Export Data

Di Gambar 4.12, terdapat diagram aktivitas untuk proses ekspor data. Langkah-langkahnya dimulai dengan admin melakukan *login*, memilih opsi menu *datasets* atau *dinas*, dan kemudian memilih opsi ekspor. Setelah memilih opsi ekspor, file akan otomatis diunduh.

9) Logout



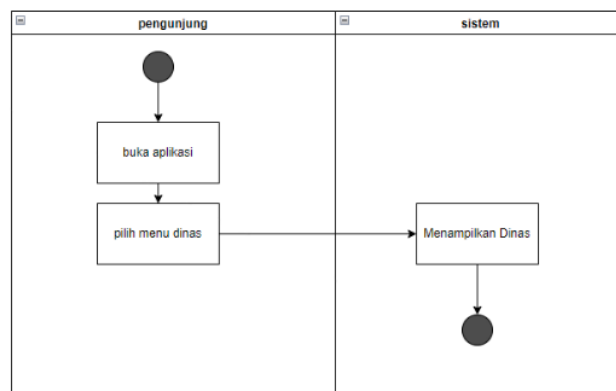
24

Gambar 4. 13 Logout

Di Gambar 4.13, terdapat diagram aktivitas untuk proses logout. Langkah-langkahnya dimulai dengan admin memilih menu profil dan mengkonfirmasi untuk logout.

b. activity diagram aktor pengunjung

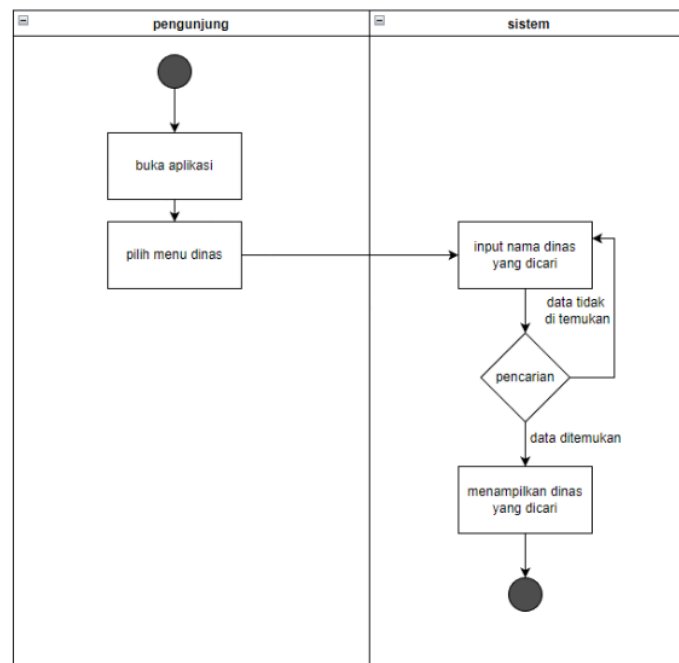
a. Melihat data dinas



Gambar 4. 14 Melihat Data Dinas

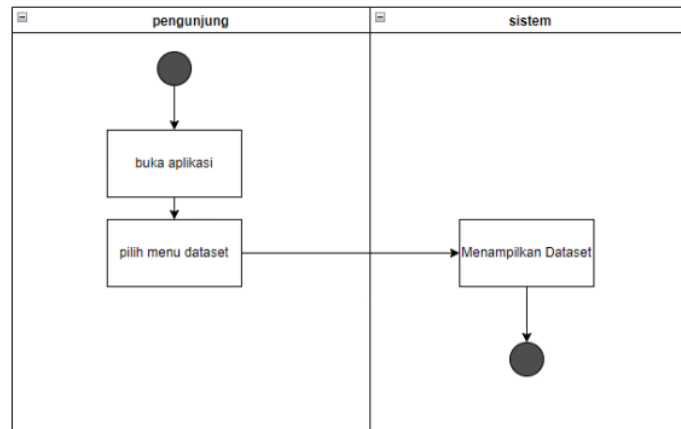
Pada Gambar 4.14, terdapat diagram aktivitas pengunjung untuk melihat data dinas. Langkah-langkah yang dilakukan adalah membuka situs *web* dan memilih menu dinas.

b. Mencari data dinas



5
Gambar 4. 15 Mencari Data Dinas

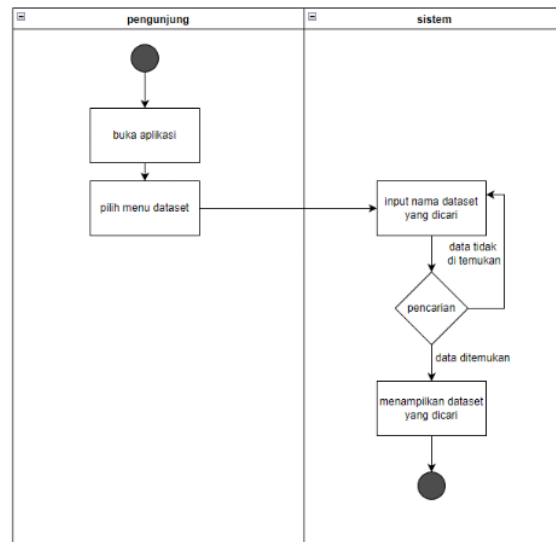
Pada Gambar 4.15, terdapat diagram aktivitas pengunjung untuk mencari data dinas. Langkah-langkahnya meliputi membuka situs *web*, memilih menu dinas, menginput nama dinas yang ingin dicari, dan kemudian menampilkan data dinas yang dicari. Jika data dinas tidak ditemukan, maka akan ditampilkan pesan bahwa data tidak tersedia.

c. Melihat *dataset*Gambar 4. 16 Melihat *dataset*

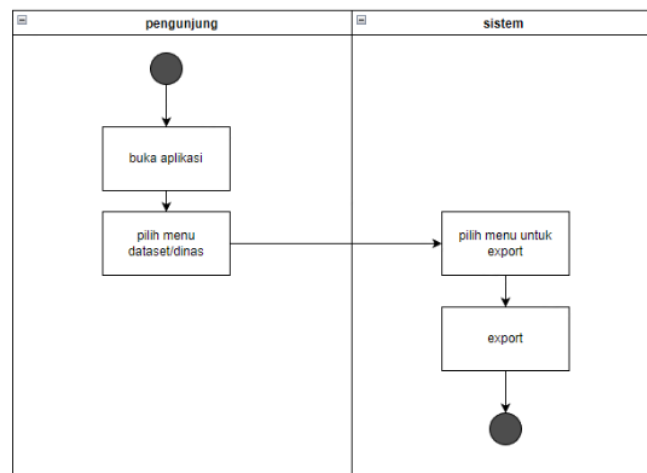
Pada Gambar 4.16, terdapat diagram aktivitas pengunjung untuk melihat *dataset*. Langkah-langkahnya dimulai dengan membuka situs *web* dan kemudian memilih menu *dataset*.

d. Pencarian *dataset*

Pada Gambar 4.17, terdapat diagram aktivitas untuk pencarian *dataset* oleh pengunjung. Langkah-langkahnya meliputi membuka situs *web*, memilih menu *dataset*, memasukkan nama *dataset* yang ingin dicari, dan kemudian menampilkan hasil pencarian. Jika *dataset* tidak ditemukan, akan muncul pesan yang menyatakan bahwa data tidak tersedia.

Gambar 4. 17 Pencarian *Dataset*

e. Export data



5

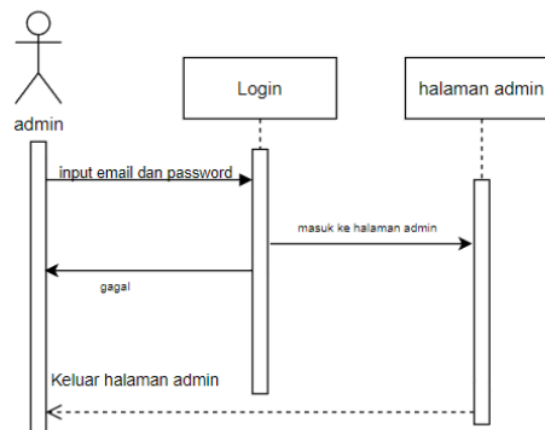
Gambar 4. 18 Export Data

Pada Gambar 4.18, terdapat diagram aktivitas pengunjung untuk mengekspor data. Langkah-langkahnya adalah membuka aplikasi, memilih menu *dataset* atau dinas, kemudian memilih opsi ekspor. Setelah itu, file akan diunduh secara otomatis.

82 4.4.3 Sequence Diagram

a. Sequence diagram admin satu data

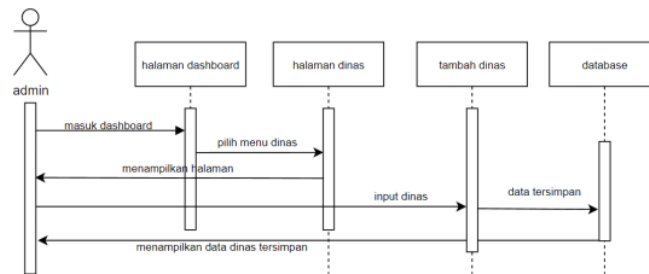
3 1) Login



Gambar 4.19 Login Admin

Pada gambar 4.19 adalah gambar sequence diagram *login* admin satu data menuju *dashboard* admin.

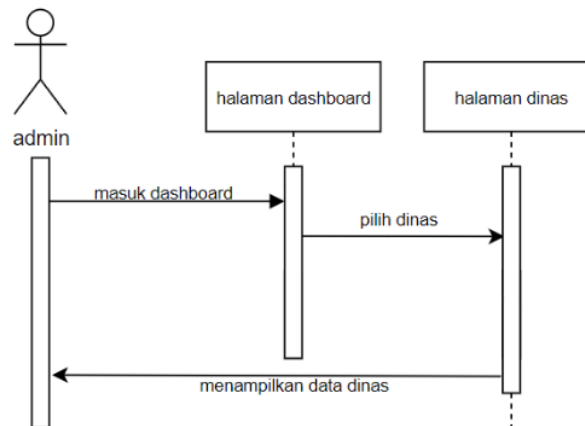
2) Tambah data dinas



5
Gambar 4. 20 Tambah Data Dinas

Pada Gambar 4.20, terdapat sequence diagram yang menggambarkan tahapan-tahapan penambahan data dinas yang dilakukan oleh admin satu data. Diagram ini menunjukkan langkah-langkah yang diambil oleh admin untuk memasukkan dan menyimpan data dinas baru ke dalam sistem.

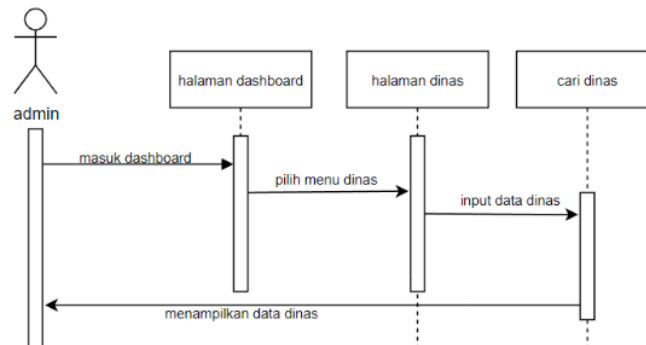
3) Lihat data dinas



Gambar 4. 21 Melihat Data Dinas

Pada gambar 4.21 adalah gambar diagram sequence diagram untuk admin melihat data dinas yang sudah terdaftar pada sistem data data.

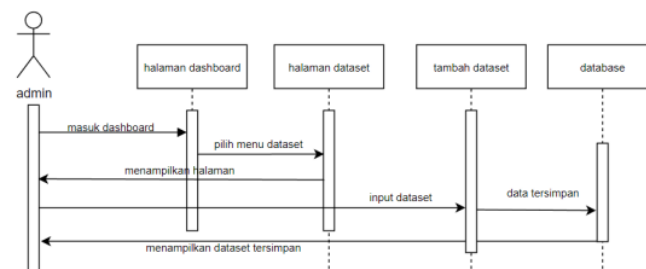
4) Pencarian data dinas



Gambar 4. 22 Pencarian Data Dinas

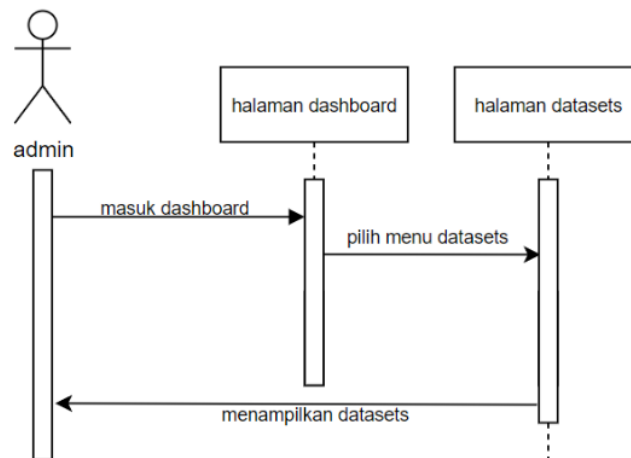
Pada Gambar 4.22, terdapat sequence diagram untuk pencarian data dinas oleh admin. Data dapat dicari berdasarkan seperti nama dinas. Diagram ini menunjukkan langkah-langkah yang dilakukan admin dalam memasukkan nama pencarian dan sistem menampilkan hasil yang sesuai.

5) Tambah *datasets*

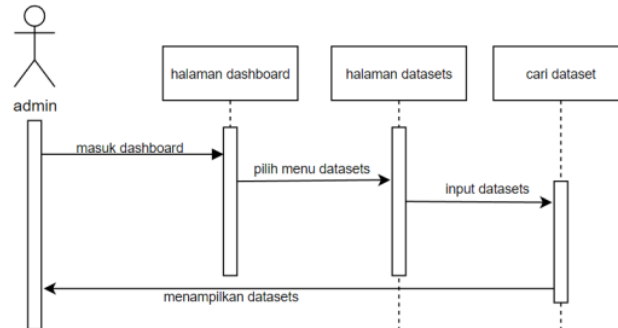


Gambar 4. 23 Tambah *Dataset*

Pada Gambar 4.23, terdapat diagram urutan (sequence diagram) yang menggambarkan proses penambahan *dataset* oleh admin. Diagram ini menunjukkan langkah-langkah yang diambil oleh admin untuk memilih opsi penambahan *dataset*, mengisi informasi yang diperlukan, dan bagaimana sistem merespons permintaan tersebut hingga *dataset* berhasil ditambahkan ke dalam sistem..

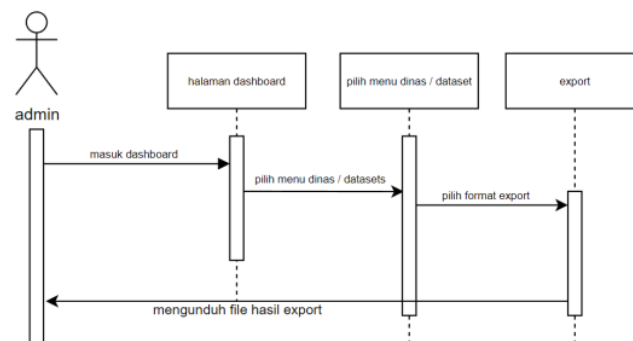
6) Lihat *datasets*Gambar 4. 24 Lihat *Datasets*

Gambar 4.24 adalah sequence diagram untuk admin dalam melihat *dataset*. Diagram ini menunjukkan proses menampilkan semua *dataset* yang telah ada dalam sistem satu data.

7) Pencarian *dataset*Gambar 4. 25 Pencarian *datasets*

Pada Gambar 4.25, terdapat diagram urutan (sequence diagram) untuk pencarian *dataset* oleh admin. *Dataset* dapat dicari berdasarkan berbagai kriteria seperti nama *dataset*. Diagram ini memperlihatkan langkah-langkah yang dilakukan oleh admin dalam memasukkan kriteria pencarian dan bagaimana sistem menampilkan hasil yang sesuai.

8) Export

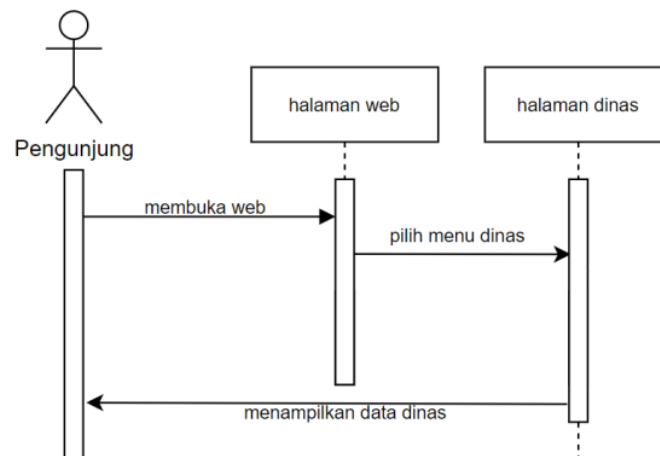


Gambar 4. 26 Export

Pada Gambar 4.26, terdapat diagram urutan (*sequence diagram*) yang menggambarkan proses *ekspor* data oleh admin. Diagram ini menunjukkan langkah-langkah yang dilakukan oleh admin untuk memilih data yang akan *diekspor* dan bagaimana sistem merespons permintaan tersebut hingga data berhasil diunduh.

b. Sequence diagram pengunjung

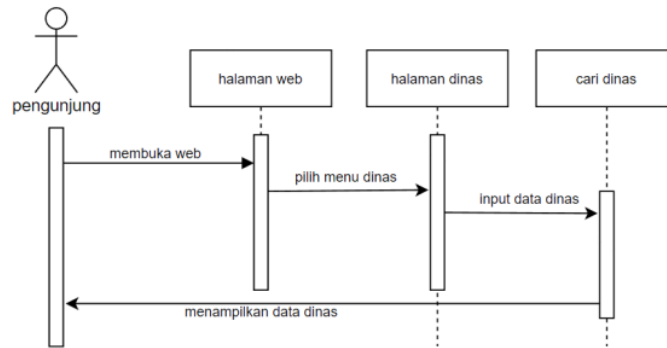
1) Melihat data dinas



5 Gambar 4. 27 Pengunjung melihat data dinas

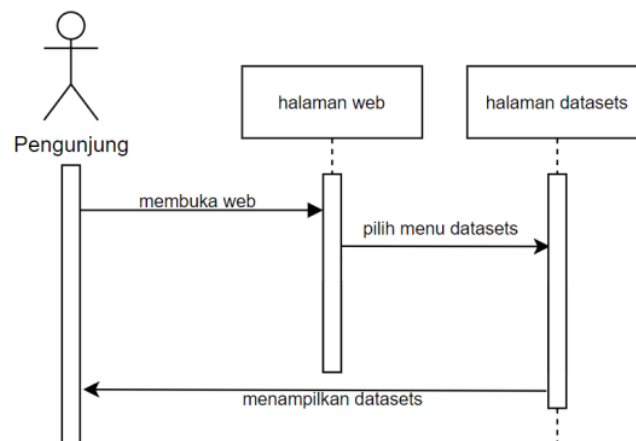
Pada Gambar 4.27, terdapat *sequence* diagram yang menggambarkan proses pengunjung melihat data dinas. Diagram ini menunjukkan langkah-langkah yang diambil oleh pengunjung untuk memilih menu dinas dan bagaimana sistem menampilkan data dinas yang tersedia kepada pengunjung.

2) Mencari data dinas



Gambar 4. 28 Pengunjung Mencari Dinas

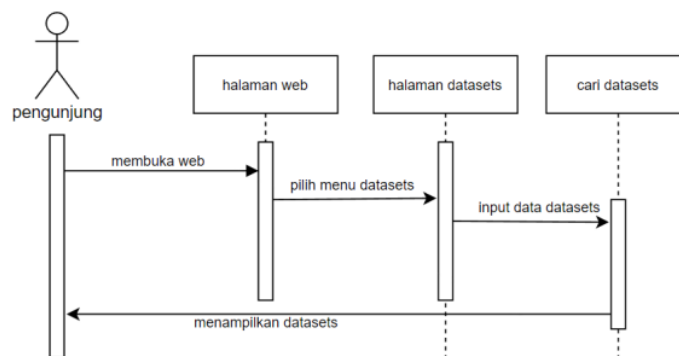
Pada Gambar 4.28, terdapat *sequence diagram* yang menggambarkan proses pencarian dinas oleh pengunjung. Diagram ini menunjukkan langkah-langkah yang diambil oleh pengunjung untuk memasukkan kriteria pencarian, seperti nama dinas, dan bagaimana sistem menampilkan hasil pencarian yang sesuai.

3) Melihat *dataset*

Gambar 4. 29 Pengunjung Melihat *Datasets*

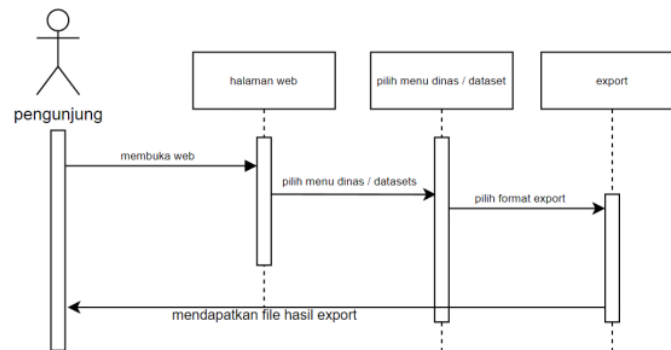
Pada Gambar 4.29, terdapat sequence diagram yang menggambarkan proses pengunjung melihat *dataset*. Diagram ini menunjukkan langkah-langkah yang dilakukan oleh pengunjung untuk memilih menu *dataset* dan bagaimana sistem menampilkan daftar *dataset* yang tersedia.

4) Mencari *dataset*

Gambar 4. 30 Pengunjung Mencari *Datasets*

Pada Gambar 4.30, terdapat sequence diagram yang menggambarkan proses pencarian *dataset* oleh pengunjung. Diagram ini menunjukkan langkah-langkah yang dilakukan oleh pengunjung untuk memasukkan kriteria pencarian berdasarkan nama *dataset* dan bagaimana sistem menampilkan hasil yang relevan.

5) Export data



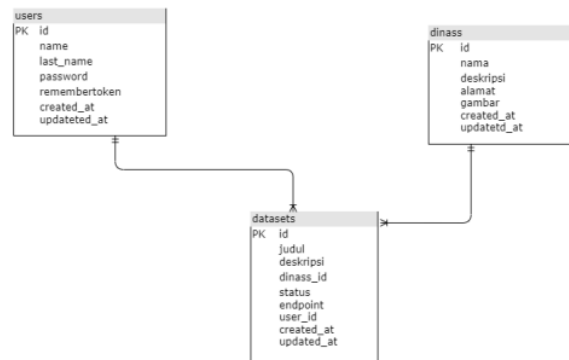
5

Gambar 4. 31 Pengunjung Mengunduh Data

Pada Gambar 4.31, terdapat sequence diagram yang menggambarkan proses ekspor data oleh pengunjung. Diagram ini menunjukkan langkah-langkah yang diambil oleh pengunjung untuk memilih data yang akan diekspor, memilih format ekspor yang diinginkan, dan bagaimana sistem merespons permintaan tersebut hingga data berhasil diunduh dalam format yang dipilih.

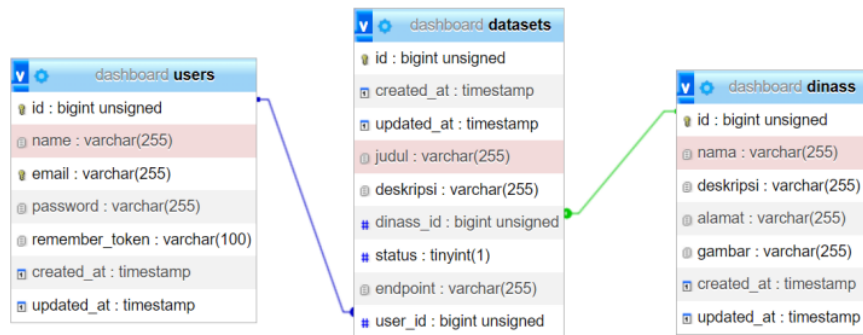
4.4.4 Class diagram

Pada gambar 4.32 adalah proses merancang struktur *database* yang akan digunakan untuk menyimpan, mengelola, dan mengakses data secara efisien. Desain ini melibatkan pembuatan skema yang mendefinisikan tabel, kolom, relasi antar tabel, dan aturan untuk menjaga integritas data. Tujuan utama dari desain *database* adalah memastikan bahwa data dapat diakses dengan cepat dan akurat, serta meminimalkan redundansi dan inkonsistensi.



Gambar 4. 32 Class Diagram

4.4.5 Desain Database



Gambar 4. 33 Desain Database

Pada gambar 4.33 menggambarkan hubungan antara pengguna, dinas, dan *dataset* dalam sistem. Terdapat tiga tabel utama, yaitu *users*, *dinass*, dan *datasets*. Desain database akan di bahas sebagai berikut:

1. tabel *users* berfungsi untuk menyimpan data users dengan 4 atribut berupa

Tabel 4. 1 Tabel Users

No.	Nama	tipe	keterangan
1	id	int	Primary key
2	name	Varchar (255)	
3	email	Varchar (255)	
4	password	Varchar (255)	
5	Created_at	timestamp	
6	Updated_at	timestamp	

2. Tabel dinas berfungsi untuk menyimpan data dinas. Dalam tabel dinas terdapat 7 field yaitu :

Tabel 4. 2 Dinas

No	nama	tipe	keterangan
1	id	int	Primary key
2	nama	varchar (255)	
3	Deskripsi	varchar (255)	
4	Alamat	varchar (255)	
5	Gambar	varchar (255)	
6	Created_at	timestamp	
7	Updated_at	timestamp	

3. Tabel *dataset* berfungsi menyimpan *dataset*. Dalam tabel *dataset* ini terdapat

9 field yaitu :

Tabel 4. 3 *Dataset*

No	nama	tipe	keterangan
1	id	int	Primary key
2	judul	varchar (255)	
3	Deskripsi	varchar (255)	
4	Dinass_id	int	Foreign Key
5	status	tinyint	
6	End_point	varchar 255	
7	User_id	int	Foreign Key
8	Created_at	timestamp	
9	Updated_at	timestamp	

Diagram ini membantu menggambarkan bagaimana data tentang pengguna, dinas, dan *dataset* saling berhubungan dan dikelola dalam sistem basis data, serta memfasilitasi operasi pengelolaan dan pengambilan data.

4.5 Desain Tampilan Sistem

A. Desain Tampilan *Login*

Gambar 4. 34 Login

Gambar 4.34 menunjukkan form login yang berfungsi untuk mengakses dashboard sistem satu data. Pengguna harus mengisi email dan kata sandi dengan benar pada form login untuk dapat masuk ke dalam dashboard.

B. Desain Landing page

Gambar 4. 35 Landing Page

Gambar 4.35 berfungsi sebagai landing page dari sistem satu data yang menyediakan antarmuka awal bagi pengguna untuk mengakses berbagai informasi dan dataset yang tersedia dalam sistem. Halaman ini memfasilitasi navigasi yang mudah dengan menyediakan tombol-tombol untuk berpindah ke halaman Beranda, Dataset, Organisasi, About, dan Login. Pengguna dapat mencari dataset tertentu melalui form pencarian

yang disediakan. Selain itu, bagian tengah halaman menampilkan *dataset* terbaru, memberikan akses cepat ke informasi terbaru.

C. Desain Halaman *Datasets*

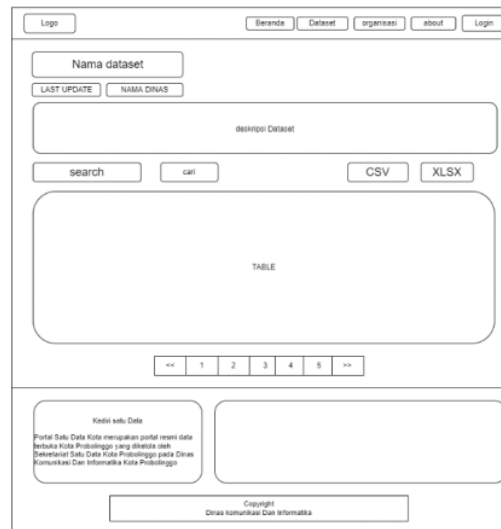
The wireframe illustrates the layout of the Datasets page. At the top, there is a navigation bar with a 'Logo' button and links for 'Beranda', 'Dataset', 'organisasi', 'about', and 'Login'. Below this is a search section titled 'Pencarian Dataset' with a descriptive sentence and a search input field with a 'Cari' button. The main content area is titled 'DATASET' and contains two identical dataset cards. Each card features a 'nama dataset' input field, 'last update' and 'Organisasi' buttons, a 'Deskripsi' text area, and a 'Preview' button. A pagination control with page numbers 1 through 5 is located below the cards. The footer includes a 'Kediri satu Data' section with a brief description of the system, a 'Copyright' notice for 'Dinas komunikasi Dan Informatika', and a large empty rectangular box.

Gambar 4. 36 *Dataset*

Gambar 4.36 menampilkan halaman *dataset* yang memuat semua *dataset* yang telah dimasukkan dan diintegrasikan oleh sistem satu data. Halaman ini berfungsi sebagai pusat informasi di mana pengguna dapat mengakses berbagai *dataset* yang tersedia dalam sistem. Setiap *dataset*

ditampilkan dengan informasi penting seperti nama, deskripsi, tanggal pembaruan terakhir, dan organisasi terkait.

D. Halaman Detail *Dataset*



Gambar 4. 37 Detail *Dataset*

Pada Gambar 4.37, terdapat gambaran detail tentang *dataset* yang memuat informasi terperinci dari setiap *dataset* dalam sistem satu data. Halaman ini menampilkan hasil integrasi *dataset*. Fitur-fitur yang tersedia di halaman ini mencakup fitur pencarian data serta kemampuan untuk mengekspor data dalam format CSV dan XLSX.

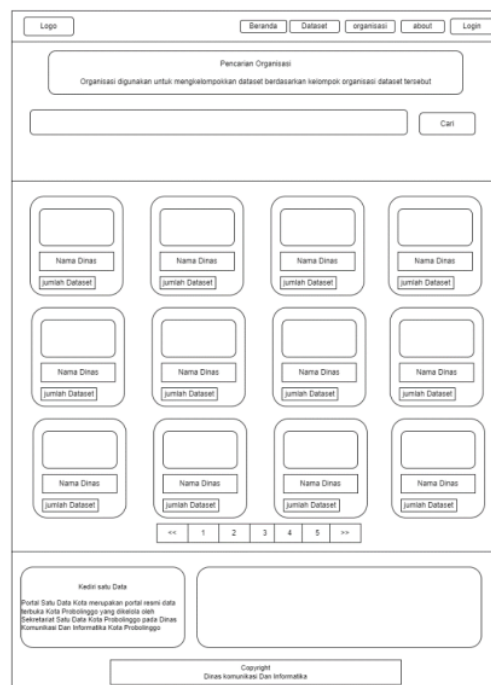
E. Halaman *About*

Pada gambar 4.38 adalah halaman about yang memberikan gambaran singkat tentang sejarah, nilai inti, dan tujuan dari *website* satu data tersebut, yang bertujuan untuk memberikan pemahaman yang lebih baik kepada pengunjung tentang identitas dan tujuan entitas yang bersangkutan.



Gambar 4. 38 About

F. Halaman Dinas



Gambar 4. 39 OPD

Pada gambar 4.39 adalah halaman dinas. Halaman ini berfungsi sebagai halaman yang menampilkan semua organisasi yang telah terintegrasi dengan sistem satu data. Halaman ini juga berfungsi sebagai halaman yang mengelompokkan *dataset* dari setiap organisasi.

G. Halaman Detail Dinas

Pada Gambar 4.40, ditampilkan halaman detail dinas. Halaman ini berfungsi untuk melihat informasi terperinci mengenai dinas, termasuk daftar *dataset* yang telah terintegrasi dalam sistem satu data. Pengguna dapat melihat detail seperti nama dinas, deskripsi, serta data yang dikelola oleh dinas tersebut.

Logo Beranda Dataset Organisasi about Login

Dataset Dinas namaDinas

Cari

jml dataset

nama dataset last update Organisasi Deskripsi Preview

nama dataset last update Organisasi Deskripsi Preview

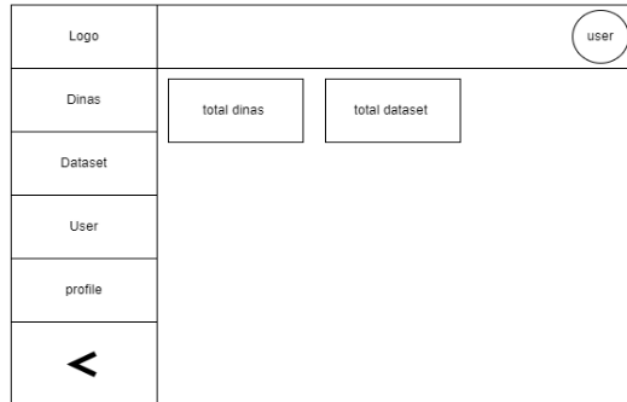
1 2 3 4 5

Kontak satu Data
Hubungi Satu Data Kita menggunakan portal resmi data
Berita Kita yang tersedia dan terintegrasi Satu
Data hadir pada Dinas Komunikasi Dan Informatika
Pondok

Copyright
Dinas Komunikasi Dan Informatika

Gambar 4. 40 Detail OPD

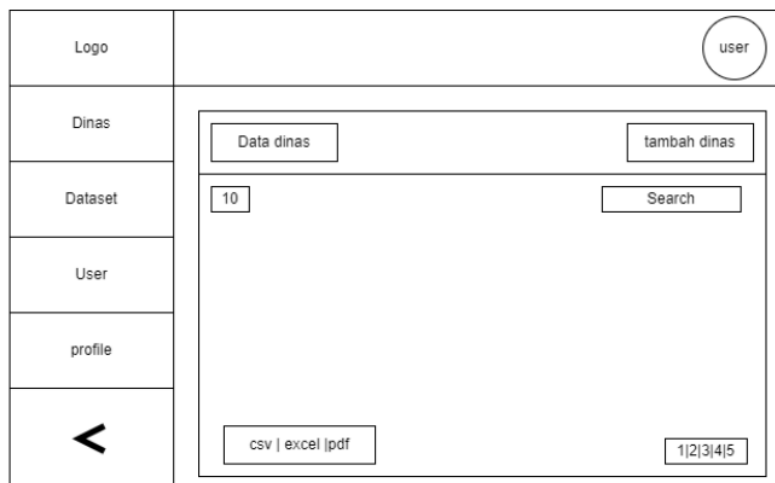
H. Dashboard Admin



⁵
Gambar 4. 41 Dashboard

Pada gambar 4.41 merupakan halaman *dashboard* admin untuk mengelola sistem satu data. Terdapat dua menu utama di dalam *dashboard* yaitu menu total dinas dan total *dataset*.

I. Menu Dinas



Gambar 4. 42 Menu dinas

Menu Dinas menampilkan jumlah dinas yang terdaftar dalam sistem, memungkinkan administrator untuk memantau, menambah, menghapus, atau memperbarui data dinas.

J. Input Dinas

Menu input dinas pada gambar 4.43 berguna untuk memasukkan data dinas baru ke dalam sistem, memungkinkan administrator untuk menambah informasi dinas yang belum terdaftar. Data yang dimasukkan kedalam form input dinas meliputi nama dinas ,alamat ,deskripsi dinas , dan gambar.

Logo	
Dinas	Tambah dinas
Dataset	nama dinas
User	alamat
profile	deskripsi
<	gambar
	Kembali Simpan

Gambar 4. 43 Input Dinas

K. Menu *Dataset*

Pada gambar 4.44 Menu *Dataset* menampilkan jumlah total *dataset* yang tersedia, dengan fitur untuk mengelola *dataset* termasuk penambahan, pengeditan, dan penghapusan *dataset*.

Logo	user
Dinas	dataset tambah dataset
Dataset	10 Search
User	
profile	
<	csv excel pdf 1 2 3 4 5

Gambar 4. 44 Menu *dataset*L. *Input Dataset*

Logo	user
Dinas	Tambah dataset
Dataset	nama dataser
User	nama dinas
profile	deskripsi
<	endpoint
	status <input type="radio"/> publish <input type="radio"/> draft Kembali Simpan

Gambar 4. 45 *Input Dataset*

Menu *Input Dataset* berfungsi untuk menambahkan *dataset* baru ke dalam sistem, memberikan kemampuan kepada administrator untuk memasukkan informasi *dataset* yang belum terdaftar. Data yang diinput ke

dalam form ini mencakup nama *dataset*, nama dinas, deskripsi, *endpoint*, serta opsi untuk mempublikasikan *dataset* atau menyimpannya sebagai draft.

BAB V

IMPELEMETASI DAN PENGUJIAN SISTEM

5.1 Arsitektur Sistem

5.1.1 Analisa kebutuhan perangkat keras (*Hardware*)

Perangkat Keras yang digunakan dalam pembuatan sistem satu data ini menggunakan spesifikasi computer sebagai berikut :

- a. Processor : AMD Ryzen 3 4300U (4C / 4T, 2.7 / 3.7GHz, 2MB L2 / 4MB L3)
- b. SSD : 512 GB
- c. Ram : 8 GB

5.1.2 Analisa Kebutuhan perangkat lunak (*Software*)

Perangkat lunak yang digunakan untuk pembuatan sistem satu data ini menggunakan beberapa perangkat lunak sebagai berikut :

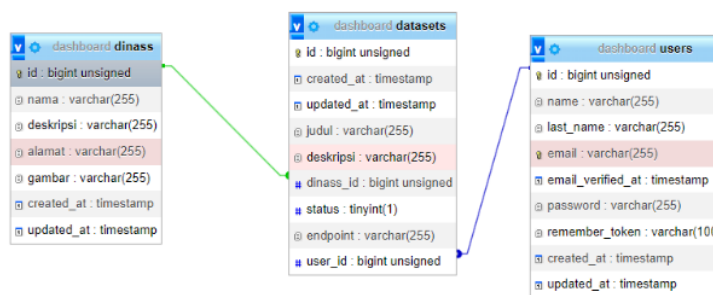
- 1. Sistem Operasi : Windows 11.
- 2. Hardware Pemrograman : Visual studio code dan laragon
- 3. Software Pendukung : Chrome, Postman.

5.2 Basis Data

Sistem Satu Data ini dikembangkan menggunakan *database* management system MySQL dan PHP versi 8.1 di aplikasi laragon . Sistem ini dirancang untuk mengelola data secara terpusat dan mencakup tiga tabel utama dalam *databasenya*:

tabel *users*, tabel *dataset*, dan tabel *dinas*. Tabel *users* digunakan untuk menyimpan informasi pengguna yang memiliki akses ke sistem, tabel *dataset* berfungsi untuk menyimpan data yang diunggah dan dikelola dalam sistem, sementara tabel *dinas* menyimpan data terkait instansi atau dinas yang terlibat.

5.2.1 Relasi Tabel



Gambar 5. 1 Relasi *database*

Pada gambar 5.1 adalah relasi database sistem satu data. Terdapat 3 tabel yang berperan di sistem satu data terdiri dari :

1. Tabel *users* : Menyimpan informasi mengenai pengguna sistem, seperti nama, alamat email, dan kata sandi.
2. Tabel *dinas* : Menyimpan informasi mengenai dinas atau organisasi yang terdaftar dalam sistem, seperti nama dinas, alamat, dan deskripsi
3. Tabel *dataset* : Menyimpan informasi mengenai *dataset* yang tersedia dalam sistem, seperti nama *dataset*, deskripsi, dan tanggal pembuatan.

Ketiga tabel tersebut saling berhubungan, yaitu tabel *dinas* dengan tabel *dataset*, serta tabel *users* dengan tabel *dataset*. Relasi antara tabel *users* dan tabel *dataset*

adalah **one-to-many**, yang berarti **satu dataset** hanya dimiliki oleh satu user, sedangkan satu user dapat memiliki banyak *dataset*. Relasi antara tabel dinas dan tabel *dataset* juga bersifat one-to-many, di mana satu dinas dapat memiliki banyak *dataset*, dan satu *dataset* hanya terkait dengan satu dinas.

5.2.2 Struktur Tabel

a. Tabel *Users*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	3 email	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	4 password	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	5 remember_token	varchar(100)	utf8mb4_0900_ai_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	6 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	7 updated_at	timestamp			Yes	NULL			Change Drop More

21

Gambar 5. 2 Tabel *Users*

Pada gambar 5.2 tersebut adalah gambar tabel user dalam *database* yang berfungsi untuk menyimpan informasi terkait pengguna dalam sistem. Tabel ini memiliki beberapa kolom penting, yaitu kolom id yang bertipe bigint dengan atribut auto-increment sebagai primary key, kolom name untuk menyimpan nama pengguna, kolom email untuk menyimpan alamat email pengguna, kolom password untuk menyimpan kata sandi pengguna, dan kolom remember_token untuk menyimpan token yang digunakan untuk mengingat sesi pengguna.

b. Tabel Dinas

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 nama	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	3 deskripsi	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	4 alamat	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	5 gambar	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	6 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	7 updated_at	timestamp			Yes	NULL			Change Drop More

Gambar 5. 3 Tabel Dinas

Pada gambar 5.3 adalah gambar tabel dinas dalam *database* ini berfungsi untuk menyimpan informasi terkait instansi atau dinas yang terlibat dalam sistem. Tabel ini memiliki beberapa kolom penting, yaitu kolom id yang bertipe bigint dengan atribut auto-increment sebagai primary key, kolom nama untuk menyimpan nama dinas, kolom deskripsi untuk memberikan deskripsi singkat mengenai dinas, kolom alamat yang menyimpan alamat lengkap dinas, dan kolom gambar untuk menyimpan tautan atau nama file gambar yang berkaitan dengan dinas tersebut. Selain itu, tabel ini juga memiliki dua kolom timestamp, `created_at` dan `updated_at`, yang berguna untuk mencatat waktu pembuatan dan pembaruan data pada tabel tersebut. Dengan struktur ini, tabel dinas mampu mengelola data dinas secara terorganisir dan efisien, mendukung fungsi-fungsi lain dalam sistem Satu Data.

c. Tabel *Dataset*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2 created_at	timestamp			Yes	NULL			
<input type="checkbox"/>	3 updated_at	timestamp			Yes	NULL			
<input type="checkbox"/>	4 judul	varchar(255)	utf8mb4_0900_ai_ci		No	None			
<input type="checkbox"/>	5 deskripsi	varchar(255)	utf8mb4_0900_ai_ci		No	None			
<input type="checkbox"/>	6 dinas_id	bigint		UNSIGNED	No	None			
<input type="checkbox"/>	7 status	tinyint(1)			No	None			
<input type="checkbox"/>	8 endpoint	varchar(255)	utf8mb4_0900_ai_ci		No	None			
<input type="checkbox"/>	9 user_id	bigint		UNSIGNED	No	None			

55

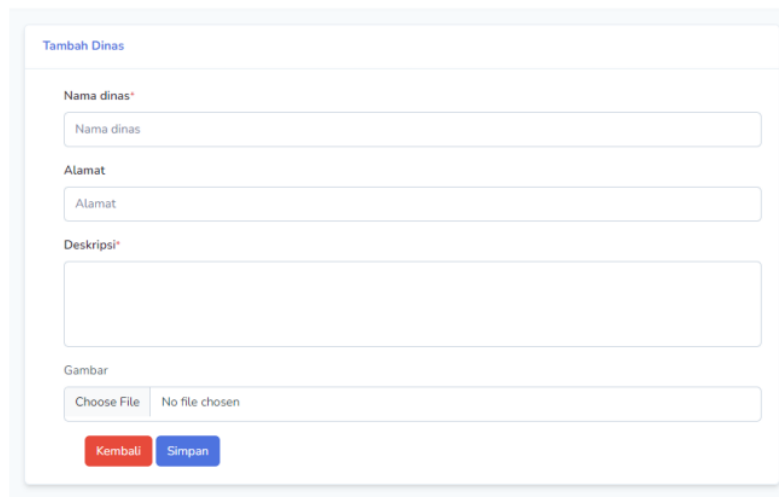
Gambar 5. 4 Tabel *dataset*

Pada gambar 5.4 tabel *dataset* dalam *database* ini berfungsi untuk menyimpan informasi mengenai berbagai kumpulan data yang dikelola oleh sistem. Tabel ini terdiri dari beberapa kolom utama, yaitu kolom *id* yang bertipe *bigint* dengan atribut *auto-increment* sebagai *primary key*, *created_at* dan *updated_at* yang bertipe *timestamp* untuk mencatat waktu pembuatan dan pembaruan data, serta kolom *judul* dan *deskripsi* yang masing-masing bertipe *varchar* untuk menyimpan *judul* dan *deskripsi* dari *dataset*. Selain itu, terdapat kolom *dinas_id* yang merupakan *foreign key* untuk mengaitkan *dataset* dengan tabel *dinas*, kolom *status* yang bertipe *tinyint* untuk menunjukkan status aktif atau tidaknya *dataset*, kolom *endpoint* yang menyimpan alamat URL terkait *dataset*, dan kolom *user_id* yang juga merupakan *foreign key* untuk mengaitkan *dataset* dengan pengguna dari tabel *users*. Dengan struktur ini, tabel *dataset* mampu mengorganisir data

secara komprehensif, memastikan keterkaitan antar tabel, dan mendukung fungsi-fungsi operasional dalam sistem Satu Data.

5.3 Tampilan input, output , laporan dan *script*

a. Input Dinas



The image shows a web form titled "Tambah Dinas". It contains the following fields and controls:

- Nama dinas***: A text input field with the placeholder "Nama dinas".
- Alamat**: A text input field with the placeholder "Alamat".
- Deskripsi***: A larger text area for entering a description.
- Gambar**: A file upload control with a "Choose File" button and the text "No file chosen".
- Buttons**: Two buttons at the bottom, "Kembali" (red) and "Simpan" (blue).

Gambar 5. 5 Input Dinas

Pada gambar 5.5 merupakan input dari data dinas yang digunakan untuk menambah informasi sebuah dinas. Form ini terdiri dari beberapa bidang input yaitu nama dinas berfungsi untuk nama dinas, Alamat digunakan untuk memasukkan Alamat dinas , dan deskripsi untuk memasukkan deskripsi tentang dinas tersebut , dan yang terakhir ada input file gambar yang berfungsi untuk menambah logo atau gambar dari dinas terkait.

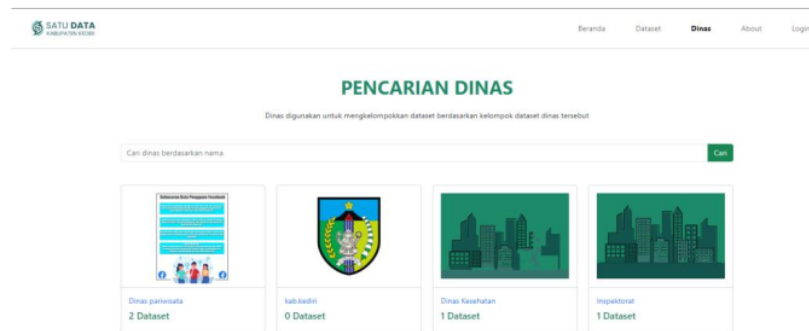
b. *Output data dinas di dashboard*

No	Nama	Deskripsi	Alamat	Action
1	Dinas pariwisata	Dinas Pariwisata merupakan unsur pelaksana Pemerintah Daerah di Bidang Pariwisata dan Ekonomi Kreatif yang dipimpin oleh seorang Kepala Dinas yang berada di bawah dan bertanggungjawab kepada Bupati melalui sekretaris Daerah.	Jl. bh. hantem ayayi	[Edit] [Delete]
2	kub kedri	pemit kedri adalah pemit yang mengursi bla bla bla bla	J. permda sejahtera	[Edit] [Delete]
3	Dinas Kesehatan	dinas kesehatan adalah	J. apa ada	[Edit] [Delete]
4	Inspektat	wang mengursi urusan intrusaadng	J. sano ssi	[Edit] [Delete]
5	Dinas Keasapan	dasgbbawdd	J. mana apa	[Edit] [Delete]
6	Dinas UMKM dan Usaha mikro	Dinas mempunyai tugas membantu Gubernur melaksanakan urusan pemerintahan Bidang Koperasi, Usaha Kecil dan Menengah yang menjadi kewenangan Daerah dan Tugas Pembantuan yang ditugaskan kepada Daerah.	J. arlangga no 45 kedri	[Edit] [Delete]

Gambar 5. 6 Data Dinas Di *Dashboard*

Gambar 5.6 menunjukkan tampilan output data dinas di halaman *dashboard*. Tabel ini berjudul "Data Dinas" dan berisi beberapa kolom: "No" untuk nomor urut, "Nama" untuk nama dinas, "Deskripsi" untuk deskripsi singkat mengenai dinas, "Alamat" untuk alamat dinas, dan "Action" yang menyediakan tombol untuk mengedit atau menghapus data dinas. Di bagian atas tabel, terdapat kontrol untuk memilih jumlah baris yang ditampilkan per halaman dan kotak pencarian untuk mencari data berdasarkan kata kunci tertentu. Di bagian kanan atas tabel, terdapat tombol "Tambah Dinas +" untuk menambahkan data dinas baru. Di bagian bawah tabel, terdapat tombol-tombol untuk mengekspor data dalam berbagai format seperti *Copy*, *Excel*, *PDF*, dan *Print*, serta kontrol navigasi untuk berpindah halaman. Tabel ini dirancang untuk memudahkan pengguna dalam mengelola dan mengakses informasi mengenai dinas-dinas yang terdaftar secara terstruktur dan efisien.

c. *Output Data Dinas di Landing page*



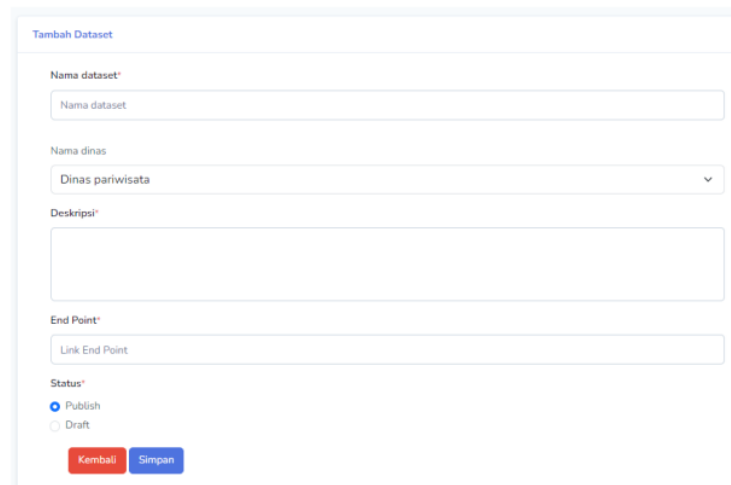
Gambar 5. 7 *Output Data Dinas di Landing Page*

Gambar 5.7 menunjukkan tampilan output data dinas di halaman *landing page* dari situs "Satu Data." Halaman ini berjudul "PENCARIAN DINAS" dan digunakan untuk mencari data dinas. Di bawah judul, terdapat kotak pencarian yang memungkinkan pengguna mencari dinas berdasarkan nama. Halaman ini juga memiliki deskripsi singkat yang menjelaskan bahwa dinas digunakan untuk mengelompokkan *dataset* berdasarkan kelompok dinas tersebut. Secara keseluruhan, gambar ini memperlihatkan bagaimana data dinas diorganisasikan dan ditampilkan di halaman *landing page*, memudahkan pengguna dalam mencari dan melihat *dataset* yang terkait dengan dinas-dinas tertentu.

d. *Halaman Input Dataset*

Pada gambar 5.8 terlihat sebuah formulir input *dataset* yang digunakan untuk menambahkan *dataset*. Formulir *dataset* ini terdiri dari beberapa input, yaitu "Nama *dataset*" yang digunakan untuk memberi nama dari *dataset*, "Nama dinas" yang berbentuk input select sehingga hanya bisa memilih dinas yang

telah dimasukkan sebelumnya dari input dinas. Selanjutnya, terdapat "Deskripsi" yang berfungsi untuk memberikan deskripsi dari *dataset* yang akan dimasukkan. Selain itu, ada input "End Point" yang berguna untuk menyimpan link *endpoint* yang nantinya akan diintegrasikan oleh sistem Satu Data. Terakhir, terdapat pilihan input button "Publish" atau "Draft" yang berarti jika dipilih "Publish" maka *dataset* akan dapat dilihat di halaman *landing page*, dan jika dipilih "Draft" maka *dataset* tidak akan ditampilkan di halaman *landing page*.



The image shows a web form titled "Tambah Dataset". It contains the following elements:

- A text input field labeled "Nama dataset*".
- A dropdown menu labeled "Nama dinas" with "Dinas pariwisata" selected.
- A large text area labeled "Deskripsi*".
- A text input field labeled "End Point*" with "Link End Point" as a placeholder.
- Two radio buttons under "Status*": "Publish" (selected) and "Draft".
- Two buttons at the bottom: "Kembali" (red) and "Simpan" (blue).

Gambar 5. 8 *Input Dataset*

e. Halaman *Output* Dinas

Gambar 5.9 menunjukkan tampilan output *dataset* pada halaman *dashboard* yang dirancang untuk memudahkan pengguna dalam mengelola data. Tabel pada *dashboard* ini berisi beberapa kolom penting yang mencakup informasi seperti nama *dataset*, deskripsi, dinas terkait, *endpoint* URL, penulis data, status

publikasi, dan aksi yang dapat diambil terhadap data tersebut. Terdapat juga tombol "Tambah *dataset*" di bagian atas tabel yang memungkinkan pengguna untuk menambahkan *dataset* baru. Fitur-fitur ini memberikan antarmuka yang intuitif untuk mengelola *dataset*, termasuk penambahan, pengeditan, dan penghapusan data, serta memantau status publikasinya.

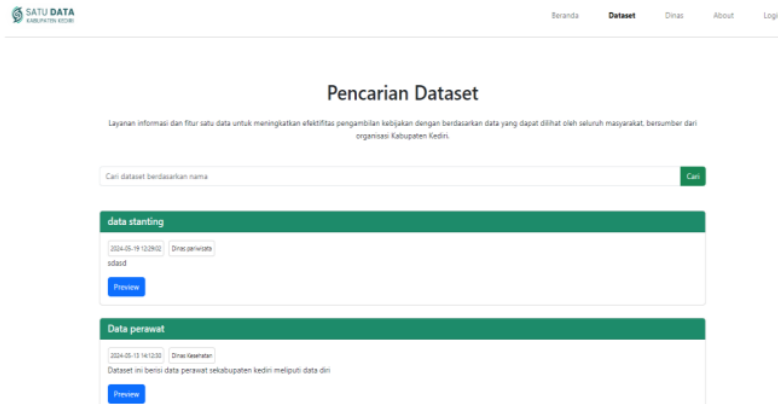
No	Nama Dataset	Deskripsi	Dinas	Endi-point	penulis	status	Action
1	Data tempat wisata	latata	Dinas pariwisata	http://55a-d5a-93f-ahmad@madhan.fy.diclipa/	Jahn	Draft	[Add] [Edit] [Delete]
2	data starting	sefasd	Dinas pariwisata	https://55a-d5a-93f-ahmad.fy.diclipa/1	Jahn	Publiah	[Add] [Edit] [Delete]
3	Data perawat	Dataset ini berisi data perawat sekabupaten kediri meliputi data diri	Dinas Kesehatan	http://127.0.0.1:8081/api/manajemen/kecamatan	Jahn	Publiah	[Add] [Edit] [Delete]
4	data diang	ini data untuk bla bla bla	Inspektorat	http://127.0.0.1:8081/api/manajemen/kecamatan	Jahn	Publiah	[Add] [Edit] [Delete]
5	Data pelatihan di Dinas koperasi	Data pelatihan yang telah diadakan Di kabupaten	Dinas UMKM dan Usaha mikro	http://127.0.0.1:8081/api/belatihan	Jahn	Publiah	[Add] [Edit] [Delete]
6	Data Kecamatan	Data kecamatan	Dinas UMKM dan Usaha mikro	http://127.0.0.1:8081/api/manajemen/kecamatan	Jahn	Publiah	[Add] [Edit] [Delete]

Gambar 5. 9 Output *Dataset* Di *Dashboard*

f. Halaman *Output* di *Landing page*

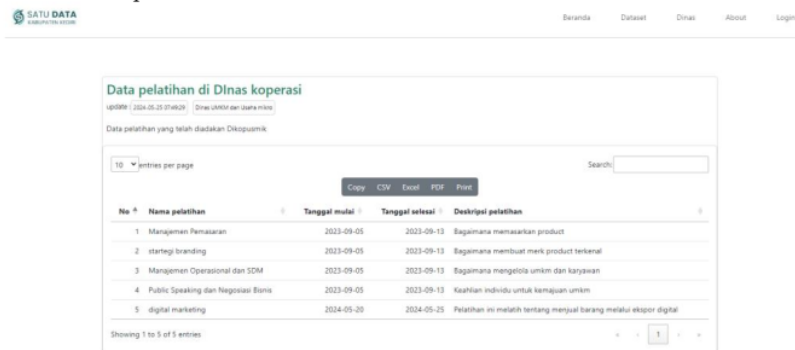
Gambar 5.10 menunjukkan halaman pencarian *dataset* dari sistem "Satu Data". Halaman ini dirancang untuk membantu pengguna dalam mencari dan mengakses *dataset* yang tersedia. Di bagian atas halaman, terdapat kolom pencarian yang memungkinkan pengguna untuk mencari *dataset* berdasarkan nama. Di bawah kolom pencarian, terdapat daftar *dataset* yang sesuai dengan kata kunci pencarian. Setiap entri *dataset* menampilkan informasi seperti nama *dataset*, tanggal dan waktu pembaruan, dinas yang bertanggung jawab, dan deskripsi singkat. Terdapat juga tombol "Preview" yang memungkinkan

pengguna untuk melihat pratinjau dari *dataset* tersebut. Fitur-fitur ini bertujuan untuk meningkatkan efektivitas pengambilan kebijakan berbasis data, yang dapat diakses oleh pengguna.



Gambar 5. 10 Output *Dataset* Di *Landing Page*

g. Halaman Laporan

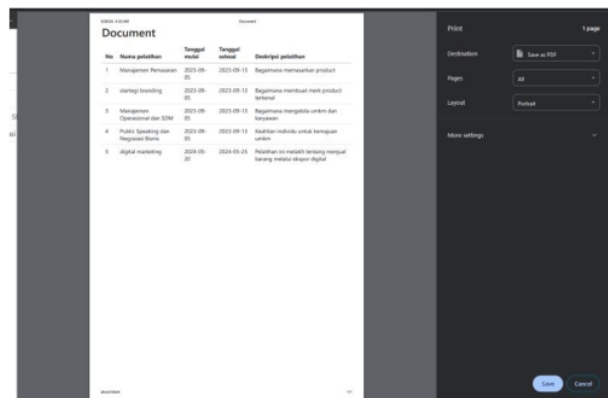


Gambar 5. 11 Output Detail *Dataset* di *landing Page*

Gambar 5.11 menampilkan fungsi halaman detail *dataset* setelah pengguna memilih opsi "Preview" dari halaman pencarian *dataset*. Ini adalah hasil dari *endpoint* yang dimasukkan dalam input *dataset*. Halaman ini

berfungsi untuk menyediakan informasi terperinci mengenai *dataset* yang dipilih. Di bagian atas halaman, terdapat informasi mengenai tanggal terakhir pembaruan *dataset* dan dinas yang bertanggung jawab atas *dataset* tersebut. Selain itu, halaman ini memiliki fitur-fitur untuk mempermudah pengelolaan dan penelusuran data, termasuk tabel yang berisi rincian *dataset*, opsi untuk menyalin data, mengunduh data dalam berbagai format (*CSV, Excel, PDF*), serta mencetak data. Terdapat juga kotak pencarian untuk memudahkan pencarian spesifik dalam tabel dan opsi untuk mengatur jumlah entri yang ditampilkan per halaman. Halaman ini dirancang untuk memberikan akses mudah dan terperinci kepada pengguna terhadap informasi *dataset* yang mereka butuhkan.

h. Contoh *Output* laporan dalam bentuk *print*



The image shows a print preview of a document titled "Document". The document contains a table with the following data:

No	Nama publikasi	Tanggal awal	Tanggal akhir	Deskripsi publikasi
1	Manajemen Perikanan	2023-08-01	2023-08-15	Regulasi manajemen perikanan
2	Strategi Inovasi	2023-08-01	2023-08-15	Regulasi inovasi hasil produk perikanan
3	Manajemen Operasional dan SDM	2023-08-01	2023-08-15	Regulasi pengelolaan sumber daya manusia
4	Public Spreading dan Regenerasi	2023-08-01	2023-08-15	Keputusan individu untuk lingkungan perikanan
5	Digital Marketing	2023-08-01	2023-08-15	Keputusan individu untuk lingkungan perikanan

The print sidebar on the right includes options for "Destination" (Save as PDF), "Pages" (All), "Layout" (Portrait), and "More settings". At the bottom of the sidebar are "Print" and "Cancel" buttons.

Gambar 5. 12 *Out Laporan*

Pada gambar 5.12 adalah tampilan output laporan yang berformat pdf. Output ini menampilkan berbagai informasi penting yang telah disusun dalam format yang rapi dan mudah dibaca.

i. *Migration User*

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password');
            $table->string('remember_token')->nullable();
            $table->timestamps();
        });
    }
}
```

Gambar 5. 14 *Migration Users*

Berikut adalah skrip migrasi untuk tabel "users" dalam *database*. Skrip ini bertujuan untuk membuat tabel "users" dengan struktur yang telah ditentukan melalui fungsi `Schema::create`. Adapun kolom-kolom yang dibuat meliputi:

1. `id` - kolom `primary key` yang berfungsi sebagai identifikasi unik untuk setiap entri di tabel "users".

2. name - kolom string yang menyimpan nama pengguna.
3. email - kolom string yang menyimpan alamat email pengguna.
4. password - kolom string yang menyimpan kata sandi pengguna.
5. remember_token - kolom string yang menyimpan token yang digunakan untuk mengingat sesi pengguna.
6. timestamps - dua kolom (created_at dan updated_at) yang secara otomatis mencatat waktu pembuatan dan pembaruan setiap entri.

Skrip migrasi ini memastikan bahwa tabel "users" memiliki struktur yang diperlukan untuk menyimpan informasi dasar mengenai pengguna.

j. Migration Dinas

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
return new class() extends Migration {
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('dinass', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->string('deskripsi');
            $table->string('alamat');
```

```
$table->string('gambar');  
$table->timestamps();  
});  
}  
};
```

Gambar 5. 15 *Script Migration Dinas*

Pada Gambar 5.15, ditampilkan skrip migrasi untuk tabel "dinass" dalam *database*. Skrip ini bertujuan untuk membuat tabel "dinass" dengan struktur yang telah ditentukan melalui fungsi `Schema::create`. Adapun kolom-kolom yang dibuat meliputi:

1. `id` - kolom primary key yang berfungsi sebagai identifikasi unik untuk setiap entri di tabel "dinass".
2. `nama` - kolom string yang menyimpan nama dinas.
3. `deskripsi` - kolom string yang menyimpan deskripsi dinas.
4. `alamat` - kolom string yang menyimpan alamat dinas.
5. `gambar` - kolom string yang menyimpan path atau URL gambar yang terkait dengan dinas.
6. `timestamps` - dua kolom (`created_at` dan `updated_at`) yang secara otomatis mencatat waktu pembuatan dan pembaruan setiap entri.

Skrip migrasi ini memastikan bahwa tabel "dinass" memiliki struktur yang diperlukan untuk menyimpan informasi dasar mengenai dinas, yang dapat

digunakan lebih lanjut dalam aplikasi. Dengan demikian, integritas data dan kemudahan dalam pengelolaan informasi dinas dapat terjamin.

k. *Model Dinas*

Pada gambar 5.16 ditampilkan model Dinas. Skrip ini berfungsi untuk mengelola tabel *database* dinass, termasuk menentukan field yang boleh dimasukkan atau yang tidak, serta mengelola relasi *database* dinass dengan tabel lain. Seperti yang ditunjukkan dalam gambar, terdapat relasi dengan tabel *dataset* yang bernilai *hasMany*, yang berarti satu dinas memiliki banyak *dataset*.

```
20
<?php
namespace App\Models;
use App\Models\dataset;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;
class dinas extends Model
{
    use HasFactory;
    protected $table = 'dinass';
    protected $guarded = ['id'];
    75
    public function datasets()
    {
        return $this->hasMany(dataset::class, 'dinass_id');
    }
}
```

Gambar 5. 16 *Script Model Dinas*

l. Controller Dinas

Pada gambar 5.17, terdapat skrip controller dinas yang berfungsi untuk mengatur input data dinas. Skrip ini mencakup berbagai tugas, seperti validasi input, menentukan form yang harus diisi, menampilkan pesan kesalahan jika terjadi kesalahan, memberikan notifikasi keberhasilan setelah data berhasil diinput, serta menghubungkan dan menyinkronkan data antara *database* dan tampilan (view). Dengan demikian, skrip ini memastikan bahwa semua data yang dimasukkan valid dan proses input berjalan lancar.

```
50 public function store(Request $request)
    {
        try {
            $validatedData = $request->validate([
                'nama' => 'required|string|max:30|unique:dinass',
                'alamat' => 'required|string|max:255',
                'deskripsi' => 'required|string',
                'gambar' =>
19 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
            ]);
            if ($request->hasFile('gambar')) {
                $file = $request->file('gambar');
                $gambarPath = $file->store('public/gambar_dinas');
                $validatedData['gambar'] = basename($gambarPath);
            }
        }
    }
}
```

```

        Dinas::create($validatedData);
        Session::flash('success', 'success');
        Session::flash('message', 'sukses menambahkan dinas');

        return redirect()->route('dinas.index');
    } catch (\Illuminate\Validation\ValidationException $e) {
        $errors = $e->validator->getMessageBag();
        return back()->withErrors($errors)->withInput();
    }
}

```

Gambar 5. 17 Controller Dinas

m. *Migration Dataset*

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('datasets', function (Blueprint $table) {
            $table->id();
            $table->timestamps();

```

```

$table->string('judul');
$table->string('deskripsi');
$table->unsignedBigInteger('dinass_id');
$table->foreign('dinass_id')->references('id')->on('dinass');
$table->boolean('status');









$table->string('endpoint');





$table->unsignedBigInteger('user_id');





$table->foreign('user_id')->references('id')->on('users');
});
}
};

```

Gambar 5.18 Migration Dataset

Pada Gambar 5.18, ditampilkan skrip migrasi untuk tabel "*datasets*" dalam *database* skrip ini bertujuan untuk membuat tabel "*datasets*" dengan struktur yang telah ditentukan melalui fungsi `Schema::create`. Adapun kolom-kolom yang dibuat meliputi:

1. `id` - kolom primary key yang berfungsi sebagai identifikasi unik untuk setiap entri.
2. `timestamps` - dua

 kolom (`created_at` dan `updated_at`) yang secara otomatis mencatat waktu pembuatan dan pembaruan setiap entri.
3. `judul` - kolom string yang menyimpan judul dari *dataset*.
4. `deskripsi` - kolom string yang menyimpan deskripsi dari *dataset*.
5. `dinass_id` - kolom unsigned big integer yang berfungsi sebagai foreign key, menghubungkan entri pada tabel "*datasets*" dengan tabel "*dinass*".

6. `status` - kolom boolean yang menyimpan status dari *dataset*.
7. `endpoint` - kolom string yang menyimpan informasi *endpoint* dari *dataset*.
8. `user_id` - kolom unsigned big integer yang juga berfungsi sebagai foreign key, menghubungkan entri pada tabel "*datasets*" dengan tabel "*users*".

Foreign key constraints pada kolom `dinas_id` dan `user_id` menggunakan metode `references` dan `on` untuk memastikan bahwa nilai yang dimasukkan ke dalam kolom-kolom ini harus sesuai dengan nilai `id` yang ada di tabel "*dinass*" dan "*users*". Dengan demikian, skrip migrasi ini tidak hanya mendefinisikan struktur tabel "*datasets*", tetapi juga menjaga integritas referensial antara tabel-tabel yang terhubung di dalam *database*.

n. Model *Dataset*

```
67
<?php
namespace App\Models;
use App\Models\User;
4
use App\Models\dinas;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Factories\HasFactory;
class dataset extends Model
{
    use HasFactory;
    protected $guarded = ['id'];
```

```

protected $table = 'datasets';
public function dinass(): BelongsTo
{
    return $this->belongsTo(dinas::class, 'dinass_id');
}
public function user(): BelongsTo
{
    return $this->belongsTo(User::class);
}
}

```

Gambar 5. 19 Model Dataset

o. Controller Dataset

```

48 public function store(Request $request)
{
    $validatedData = $request->validate([
        'judul' => 'required|max:255',
        'deskripsi' => 'required',
        'dinass_id' => 'required',
        'status' => 'required',
        'endpoint' => 'required',
    ]);

    $user_id = auth()->indeks();

    $validatedData['user_id'] = $user_id;

    dataset::create($validatedData);
    Session::flash('success', 'success');
    Session::flash('message', 'sukses menambahkan dataset');
}

```



```

return redirect()->route('dataset.index');
}

```

Gambar 5. 20 controller *dataset*

Pada Gambar 5.20 ditampilkan controller bernama *DatasetController* yang berfungsi untuk memvalidasi input *dataset* agar sesuai dengan yang diharapkan. Controller ini juga berguna untuk memberikan notifikasi error maupun sukses, serta menghubungkan data dengan view. Melalui fungsi-fungsi di dalamnya, *DatasetController* memastikan bahwa proses input *dataset* dilakukan dengan benar dan sesuai dengan aturan yang ditetapkan, serta memberikan umpan

p. *Contoh Endpoint API*

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Api\KecamatanController;
use App\Http\Controllers\Api\PelatihanController;

/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "api" middleware group. Make something great!
|
*/

Route::get("/manage-kecamatan", [KecamatanController::class, 'index']);
Route::get("/pelatihan", [PelatihanController::class, 'index']);

```

Gambar 5. 21 *Contoh End Point*

Gambar 5.21 merupakan contoh *endpoint* yang digunakan dalam sistem pendataan UMKM. *Endpoint* ini berfungsi sebagai titik akses untuk mengambil

data dari sistem tersebut. Penggunaan *endpoint* ini memungkinkan akses data dari luar sistem, memungkinkan integrasi dengan sistem lain untuk keperluan pengolahan data yang lebih luas.

q. *Controller End Point*

Gambar 5.22 menunjukkan contoh pengolahan data yang akan disampaikan melalui *endpoint* yang telah dijelaskan pada Gambar 5.20. Dalam konteks ini, penting untuk mengatur format data sedemikian rupa sehingga sesuai dengan permintaan dari sistem satu data.

```
use App\Http\Controllers\Controller;
use App\Models\Pelatihan;
use Illuminate\Http\Request;

2 references | 0 implementations
class PelatihanController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        $pelatihans = Pelatihan::all();

        if ($pelatihans->isEmpty()) {
            return response()->json(['data belum tersedia']);
        }

        $data = $pelatihans->map(function ($pelatihan) {
            return [
                'id' => $pelatihan->id,
                'Nama pelatihan' => $pelatihan->name_pelatihan,
                'Tanggal mulai' => $pelatihan->start_date,
                'Tanggal selesai' => $pelatihan->end_date,
                'Deskripsi pelatihan' => $pelatihan->description_pelatihan
            ];
        });

        return response()->json($data);
    }
}
```

Gambar 5. 22 Contoh *End Point*

r. *Controller Untuk Menangkap Data Dari API*

Pada Gambar 5.22 ditampilkan sebuah controller untuk melihat detail data API yang diambil dari *endpoint* yang tersimpan dalam *database* satu data. Data akan diambil menggunakan HTTP GET. Jika data tersedia, informasi yang akan

ditampilkan mencakup nama *dataset*, pembaruan terakhir, nama dinas, deskripsi *dataset*, dan *endpoint*. Informasi ini akan ditampilkan dalam bentuk tabel. Namun, jika data tidak tersedia, maka tidak ada data yang akan ditampilkan.

```

public function detail($id)
{
    try {
        $dataset = Dataset::find($id);
        // Ensure dataset is found
        if (!$dataset) {
            return redirect()->back()->with('error', 'Dataset tidak ditemukan');
        }
        $endpoint = $dataset->endpoint;
        // Validasi URL
        if (filter_var($endpoint, FILTER_VALIDATE_URL) === FALSE) {
            echo "<script>alert('Data belum ada')</script>";
            echo "<script>window.location = '/'</script>";
        }
        $response = Http::get($endpoint);
        if (!$response->successful()) {
            return redirect()->back()->with('error', 'Dataset tidak ditemukan');
        }
        if ($response->successful()) {
            $endpointTable = $response->json();
            if (!is_array($endpointTable) || empty($endpointTable) ||
                !is_array($endpointTable[0])) {
                throw new \Exception('Invalid data format');
            }
        }
        return view('detail-dataset', [
            'dataset' => $dataset,
            'endpointTable' => $endpointTable,
            'success' => true,
            'access_at' => Date::now()
        ]);
    }
}

```

```

    );
    //code...
} catch (\Throwable $th) {
    return view('detail-dataset', [
        'dataset' => $dataset,
        'endpointTable' => 'data tidak ditemukan',
        'success' => false,
        'access_at' => Date::now()
    ]);
    dd($th);
    throw $th;
}
}

```

Gambar 5. 23 Controller Untuk Menangkap Data Dari API

s. View Untuk Menampilkan Data

```

<div class="col-md-4 p-2">
    <h3 style="color:rgb(28, 133, 101) ">{{ $dataset->judul }}</h3>
    <div class="tanggal-nama">
        <div>
            <small>update :</small>
        </div>
        <div class="tanggal">{{ $access_at }}</div>
        <div class="nama-dinas">
            <a style="text-decoration: none;color:black"
                href="{{ url('/front-dinas/' . $dataset->dinass->id) }}">{{ $dataset-
                >dinass->nama }}</a>
        </div>
    </div>
    </div>
    <div class="col-md-4 p-2">
        {{ $dataset->deskripsi }}
    </div>
    @if ($endpointTable && $success)
        <div class="card mt-3">
            <div class="card-body">

```

```

<table class="table table-responsive" id="myTable">
  <thead>
    <tr>
      @foreach (array_keys($endpointTable[0]) as $key)
        <th>{{ $key }}</th>
      @endforeach
    </tr>
  </thead>
  <tbody>
    @foreach ($endpointTable as $row)
      <tr>
        @foreach ($row as $cell)
          <td>{{ $cell }}</td>
        @endforeach
      </tr>
    @endforeach
  </tbody>
</table>
</div>
@else
  <div style="text-align: center; ">
    <h3> Data belum tersedia</h3>
  </div>
@endif
</div>

```

Gambar 5. 24 View Detail Data

Pada Gambar 5.24, ditampilkan sebuah script untuk menampilkan detail *dataset* yang berisi data yang diambil dari controller. Data yang ditampilkan dari api harus dalam format array dengan indeks nol. Script ini akan mengatur tampilan data yang meliputi berbagai informasi penting seperti nama *dataset*, pembaruan terakhir, nama dinas, deskripsi *dataset*, dan *endpoint*. Informasi dari *endpoint* yang di eksekusi tersebut akan disajikan dalam bentuk tabel agar

lebih mudah dibaca dan dianalisis. Jika data tidak tersedia, maka pesan yang sesuai akan ditampilkan untuk menunjukkan bahwa data tidak ditemukan.

5.4 Pengujian Sistem

Tabel 5. 1 Blackbox Testing

No	Proses		Diuji oleh	Tgl	Hasil	
					Sesuai	Tidak
1	Modul yang di uji	: <i>Login</i> aplikasi	Ali mun'im, S.Kom., M.Si.	21/06/2024		
	Prosedur pengujian	: - Akses Aplikasi melalui <i>website</i> -Input email dan Password				
	input	: Email: test@example.com password :password			✓	
	Out put(jika berhasil)	: Sistem menampilkan <i>dashboard</i> aplikasi			✓	
	Out put (jika gagal)	: Sistem error dan tidak bisa <i>login</i>			✓	

	Hasil yang didapat	: Menampilkan <i>dashboard</i> aplikasi			✓	
2	Modul yang diuji	: Tambah dinas	Ali mun'im, S.Kom., M.Si.	21/06/ 2024		
	Prosedur Pengujian	1. <i>Login</i> sebagai admin. 2. Pilih menu tambah data dinas				
	input	Nama dinas : dinas koperasi dan usaha mikro Alamat : jl airlangga no 45 deskripsi : dinas Dinas mempunyai tugas membantu melaksanakan urusan pemerintahan Bidang Koperasi, Usaha Kecil dan Menengah yang menjadi kewenangan Daerah dan Tugas			✓	

		Pembantuan yang ditugaskan kepada Daerah Foto dinas : foto.jpg				
	Out put (jika berhasil)	: Data dinas berhasil input dari <i>dataset</i>			✓	
	Out put jika gagal	: -Apabila data dinas dengan nama yang sama maka tidak bisa disimpan - foto yang lebih dari 2 mb tidak bisa disimpan -jika ada form yang tidak di isi akan tidak disimpan			✓	
	Hasil yang didapat	: Data disimpan kedalam databse datu data di tabel dinas			✓	
3	Modul yang di uji	: Tambah <i>dataset</i>	Ali mun'im, S.Kom., M.Si.	21/06/ 2024		

Prosedur pengujian	:	1. <i>login</i> sebagai admin 2.pilih menu <i>dataset</i> 3.pilih menu tambah <i>dataset</i>				
input	:	- Nama <i>dataset</i> : Data pelatihan -Nama dinas :dinas koperasi dan usaha mikro -Deskripsi : data pelatihan yang pernah diadakan dinas koperasi dan umkm -End point : -Status : publish			✓	
Out put(jika berhasil)	:	Jika berhasil data akan masuk kedalam <i>database</i> satu data di table <i>dataset</i>			✓	
Out put(jika gagal)	:	Jika gagal akan ada notifikasi error seperti :			✓	

		- data dengan nama yang sudah ada . -form harus diisi semua				
	Hasil yang didapat	: <i>Dataset</i> berhasil disimpan ke databse satu data di tabel <i>dataset</i>			✓	
4.	Modul yang diuji	: Searching <i>dataset</i>	Ali mun'im, S.Kom., M.Si.	21/06/ 2024		
	Prosedur pengujian	: - Buka aplikasi - pilih search				
	Input	: Data pelatihan			✓	
	Out put(jika berhasil)	: Menemukan data pelatihan			✓	
	Out put (jika gagal)	: Data tidak ditemukan			✓	
	Hasil yang di dapat	: Menemukan data pelatihan			✓	
5	Modul yang di uji	: Preview <i>dataset</i> (menguji integrasi)	Ali mun'im, S.Kom., M.Si.	21/06/ 2024		

Prosedur pengujian	:	Buka aplikasi Pilih salah satu <i>dataset</i> dan klik preview				
Out put(jika berhasil)		Detail <i>dataset</i> akan ditampilkan beserta table (hasil integrasi berupa data)			✓	
Out put (jika gagal)		Detail <i>dataset</i> akan ditampilkan tetapi dengan notif “Data belum tersedia”			✓	
Hasil yang di dapat		Data hasil integrasi bisa dilihat dari detail <i>dataset</i> yang di preview berupa data yang di tampilkan di table			✓	

BAB VI

PENUTUP

6.1 Kesimpulan

²⁵ Berdasarkan uraian penelitian dan pembahasan yang dilakukan pada bab-bab sebelumnya maka dapat dibuat kesimpulan sebagai berikut :

1. Penelitian yang telah dilakukan menghasilkan sebuah sistem informasi satu data.
2. Penelitian ini berhasil merancang dan mengimplementasikan sistem integrasi data berbasis *RESTful API* yang efektif dan efisien. Sistem ini memungkinkan pengambilan data dari berbagai sumber yang berbeda, mengintegrasikannya, dan menyajikannya dalam format yang mudah diakses oleh pengguna.
3. Implementasi sistem integrasi data berbasis *RESTful API* ini memberikan kemudahan bagi pengguna dalam mengakses berbagai *dataset* yang sudah terintegrasi. Hal ini meningkatkan efisiensi kerja dan mempercepat informasi karena dapat diakses dalam satu platform terpadu.

⁴⁶ 6.2 Saran

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, berikut adalah beberapa saran yang dapat diberikan:

1. Peningkatan Interaktivitas UI/UX

Visualisasi Data: Tambahkan elemen visual seperti diagram batang, diagram garis, dan grafik lainnya untuk membuat antarmuka lebih interaktif dan informatif. Gunakan library seperti *Chart.js* atau *Google Charts*.

fikri cek plagiasi prodi.docx

ORIGINALITY REPORT

23%

SIMILARITY INDEX

21%

INTERNET SOURCES

6%

PUBLICATIONS

13%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Brawijaya Student Paper	1%
2	repository.unpkediri.ac.id Internet Source	1%
3	docplayer.info Internet Source	1%
4	Submitted to University of Northampton Student Paper	1%
5	repository.usd.ac.id Internet Source	1%
6	kc.umn.ac.id Internet Source	1%
7	garuda.kemdikbud.go.id Internet Source	1%
8	ismetek.itbu.ac.id Internet Source	1%
9	Submitted to Cyryx College, Maldives Student Paper	1%

10	Submitted to Universitas Pamulang Student Paper	<1 %
11	codingtricks.io Internet Source	<1 %
12	eprints.uty.ac.id Internet Source	<1 %
13	Submitted to Universitas Muhammadiyah Surakarta Student Paper	<1 %
14	ecampus.sttind.ac.id Internet Source	<1 %
15	socs.binus.ac.id Internet Source	<1 %
16	Ifanda Reza Damasta, Mochamad Alfian Rosid, Ade Eviyanti, Uce Indahyanti. "Implementasi Online Marketplace pada Industri Rumahan di Desa Bluru Permai Sidoarjo", Innovative Technologica: Methodical Research Journal, 2024 Publication	<1 %
17	Submitted to Institut Pemerintahan Dalam Negeri Student Paper	<1 %
18	ejournal-binainsani.ac.id Internet Source	<1 %

19	Submitted to University of Wales Swansea Student Paper	<1 %
20	gitea.geologie.ac.at Internet Source	<1 %
21	Submitted to Universitas Muria Kudus Student Paper	<1 %
22	digilib.uns.ac.id Internet Source	<1 %
23	jurnal.umpar.ac.id Internet Source	<1 %
24	repository.dinamika.ac.id Internet Source	<1 %
25	proceeding.unpkediri.ac.id Internet Source	<1 %
26	simki.unpkediri.ac.id Internet Source	<1 %
27	Submitted to Keimyung University Student Paper	<1 %
28	repository.unibos.ac.id Internet Source	<1 %
29	gowakab.go.id Internet Source	<1 %
30	thesis.binus.ac.id Internet Source	<1 %

31	www.laravel.web.id Internet Source	<1 %
32	peraturan.bpk.go.id Internet Source	<1 %
33	www.karir-kerja.com Internet Source	<1 %
34	www.snip2code.com Internet Source	<1 %
35	Submitted to LL DIKTI IX Turnitin Consortium Part II Student Paper	<1 %
36	journal.fkpt.org Internet Source	<1 %
37	repository.stialan.ac.id Internet Source	<1 %
38	publikasi.polije.ac.id Internet Source	<1 %
39	repo.darmajaya.ac.id Internet Source	<1 %
40	Submitted to Fakultas Ekonomi Universitas Indonesia Student Paper	<1 %
41	Submitted to Sharda University Student Paper	<1 %

42	Submitted to Universitas Islam Majapahit Student Paper	<1 %
43	www.scribd.com Internet Source	<1 %
44	123dok.com Internet Source	<1 %
45	bhinneka.com Internet Source	<1 %
46	repository.unja.ac.id Internet Source	<1 %
47	Rizki Wahyu Nugroho, Teguh Andriyanto, Rini Indriati. "Sistem Informasi Izin Online Berbasis Web Menggunakan Framework Codeigniter", Generation Journal, 2022 Publication	<1 %
48	Submitted to Universitas Pendidikan Ganesha Student Paper	<1 %
49	Submitted to Universitas Putera Batam Student Paper	<1 %
50	Submitted to University of Ulster Student Paper	<1 %
51	id.123dok.com Internet Source	<1 %
52	journal.untar.ac.id Internet Source	<1 %

53	smart.stmikplk.ac.id Internet Source	<1 %
54	vdokumen.com Internet Source	<1 %
55	Submitted to UIN Sultan Syarif Kasim Riau Student Paper	<1 %
56	ejurnal.dipanegara.ac.id Internet Source	<1 %
57	Submitted to Universitas Bengkulu Student Paper	<1 %
58	bintara-predator.blogspot.com Internet Source	<1 %
59	gitlab.mitic.gov.py Internet Source	<1 %
60	jidt.org Internet Source	<1 %
61	overcoder.net Internet Source	<1 %
62	Abdul Hafiz Syaputra, Ucuk Darussalam, Winarsih Winarsih. "Rancang Bangun Sistem Pengelolaan Laundry menggunakan Metode Waterfall", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2021 Publication	<1 %

63	eprints.pancabudi.ac.id Internet Source	<1 %
64	es.scribd.com Internet Source	<1 %
65	jurnal.darmajaya.ac.id Internet Source	<1 %
66	kisah-25-nabi-dan-rasul-367.softonic.kr Internet Source	<1 %
67	Submitted to University of Greenwich Student Paper	<1 %
68	doobedoo-myadvanture.blogspot.com Internet Source	<1 %
69	pdfcoffee.com Internet Source	<1 %
70	www.coursehero.com Internet Source	<1 %
71	www.riauonline.co.id Internet Source	<1 %
72	Ade Wardani, Miftahurrahma Rosyda. "Implementasi Aplikasi Laporan Jalan untuk Pemetaan dan Pelaporan Jalan Rusak di Wilayah Kota Tarakan", Jurnal SAINTEKOM, 2024 Publication	<1 %

73 Prasetya Elektra Zulfikar, Sampe Hotlan Sitorus, Uray Ristian. "SISTEM PRESENSI MENGGUNAKAN VERIFIKASI PALM PRINT DENGAN METODE PRINCIPAL COMPONENT ANALYSIS DAN EUCLIDEAN DISTANCE", Coding Jurnal Komputer dan Aplikasi, 2021
Publication <1 %

74 amikmahaputra.ac.id
Internet Source <1 %

75 elar.khmnu.edu.ua
Internet Source <1 %

76 id.scribd.com
Internet Source <1 %

77 jimfeb.ub.ac.id
Internet Source <1 %

78 jurnal.untagsmg.ac.id
Internet Source <1 %

79 ngoplosbareng.blogspot.com
Internet Source <1 %

80 ojs.trigunadharma.ac.id
Internet Source <1 %

81 pt.scribd.com
Internet Source <1 %

82 repository.uin-suska.ac.id
Internet Source <1 %

83	spot.zcu.cz Internet Source	<1 %
84	vdocuments.site Internet Source	<1 %
85	www.jogloabang.com Internet Source	<1 %
86	Brian Amy Prastyo. "Pengaturan Tata Kelola Big Data Untuk Sinkronisasi Penerapan Satu Data Indonesia Dan Sistem Pemerintahan Berbasis Elektronik", Soedirman Law Review, 2024 Publication	<1 %
87	repository.ub.ac.id Internet Source	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On

fikri cek plagiasi prodi.docx

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79

PAGE 80

PAGE 81

PAGE 82

PAGE 83

PAGE 84

PAGE 85

PAGE 86

PAGE 87

PAGE 88

PAGE 89

PAGE 90

PAGE 91

PAGE 92
