



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**PENGEMBANGAN APLIKASI OPEN DATA KOTA BEKASI
TERINTEGRASI CKAN MENGGUNAKAN WEB FRAMEWORK
CODEIGNITER**

TUGAS AKHIR

INTAN MARIYAHSARI

0110220033

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**PENGEMBANGAN APLIKASI OPEN DATA KOTA BEKASI
TERINTEGRASI CKAN MENGGUNAKAN WEB FRAMEWORK
CODEIGNITER**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

STT NF
INTAN MARIYAHSARI
0110220033

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Intan Mariyhsari

NIM : 0110220033

Depok , 8 Agustus 2024

Tanda Tangan



Intan Mariyhsari

STT - NF

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Intan Mariyahsari

NIM : 0110220033

Program Studi : Teknik Informatika


Judul Skripsi : Pengembangan Aplikasi Open Data Kota Bekasi Terintegrasi CKAN Menggunakan Web Framework CodeIgniter


Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing

Penguji


(Henry Saptono, S.Si., M.Kom)


(Dr. Lukman Rosyidi, S.T., M.M., M.T)

Ditetapkan di : Depok

Tanggal : 27 Juli 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri. Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Tifani Nabarian, S.Kom., M.T.I., selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Bapak Henry Saptono, S.Si., M.Kom., selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Henry Saptono, S.Si., M.Kom., selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.
8. Manajer PT Ciptadra Softindo Bapak Rendy Renaldi Berahim beserta karyawan yang telah meluangkan waktunya untuk memberikan data yang diperlukan bagi penulisan ilmiah ini.
9. Teman-teman dekat yang telah meluangkan waktu untuk senantiasa memberikan bantuan dan dukungan kepada penulis.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 8 Agustus 2024

Intan Mariyachsari



STT - NF

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Intan Mariyahsari

NIM : 0110220033

Program Studi : Teknik Informatika

Jenis karya : Skripsi / Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

Pengembangan Aplikasi Open Data Kota Bekasi Terintegrasi CKAN Menggunakan
Web Framework CodeIgniter

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

STT - NF

Dibuat di : Depok

Pada tanggal : 8 Agustus 2024

Yang Menyatakan



(Intan Mariyahsari)

ABSTRAK

Nama : Intan Mariyahsari
NIM : 0110220033
Program Studi : Teknik Informatika
Judul : Pengembangan Aplikasi Open Data Kota Bekasi Terintegrasi
CKAN Menggunakan Web Framework CodeIgniter

Tugas akhir ini mengembangkan aplikasi *Open Data* Kota Bekasi yang terintegrasi dengan *CKAN* menggunakan *web framework CodeIgniter*. Aplikasi ini difokuskan pada pengembangan *Content Management System (CMS)* untuk mengelola, memverifikasi, dan mempublikasikan dataset secara *online*. Sebelumnya, pengelolaan *dataset* masih belum optimal dalam hal kemudahan penggunaannya. Implementasi keterbukaan data oleh pemerintah Kota Bekasi dilakukan melalui aplikasi ini, memungkinkan publikasi data secara *online* dalam format yang mudah diakses oleh masyarakat. Penelitian ini menggunakan metode kualitatif dengan melakukan analisis kebutuhan pengguna untuk merancang antarmuka yang intuitif dan *user-friendly*. Aplikasi dirancang untuk mengelola *dataset* secara efisien dan transparan, serta memastikan kualitas data yang dipublikasikan dengan menyediakan mekanisme pembaruan dan penghapusan *dataset* yang terintegrasi. Pengujian aplikasi dilakukan dengan *Black Box Testing* dan *User Acceptance Testing (UAT)* untuk memastikan fungsionalitas dan kemudahan penggunaan. Dengan *CKAN* sebagai platform utama, aplikasi ini diharapkan dapat meningkatkan transparansi dan keterbukaan data pemerintah Kota Bekasi, serta mendukung pelayanan publik yang lebih efektif dan akuntabel. Hasil penelitian ini memberikan kontribusi dalam pengembangan sistem informasi berbasis *Open Data*, meningkatkan pemanfaatan data terbuka, dan meningkatkan transparansi dalam layanan publik.

Kata kunci : *Open Data, CKAN, CodeIgniter, CMS, Transparansi Data*

ABSTRACT

Name : Intan Mariyahsari

NIM : 0110220033

Study Program : Informatic Engineering

Title : *Development of Bekasi City Open Data Application Integrated with CKAN Using CodeIgniter Web Framework*

This final project develops an Open Data application for Bekasi City integrated with CKAN using the CodeIgniter web framework. The application focuses on developing a Content Management System (CMS) to manage, verify, and publish datasets online. Previously, dataset management lacked ease of use. The implementation of data openness by the Bekasi City government through this application enables online data publication in accessible formats for the public. The research employs qualitative methods to analyze user needs for designing an intuitive and user-friendly interface. The application aims to efficiently and transparently manage datasets, ensuring the quality of published data by providing integrated mechanisms for dataset updates and deletions. Application testing includes Black Box Testing and User Acceptance Testing (UAT) to ensure functionality and ease of use. With CKAN as the primary platform, the application aims to enhance government data transparency and openness in Bekasi City, supporting more effective and accountable public services. The research contributes to the development of open data based information systems, enhancing the utilization of open data, and improving transparency in public services.

Key words : *Open Data, CKAN, CodeIgniter, CMS, Transparency of Data*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	vi
ABSTRAK	vii
<i>ABSTRACT</i>	<i>viii</i>
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan dan Manfaat Penelitian	2
1.4 Batasan Masalah	3
1.5 Sistematika Penulisan	3
BAB I PENDAHULUAN	3
BAB II KAJIAN LITERATUR	3
BAB III METODE PENELITIAN	4
BAB IV IMPLEMENTASI DAN EVALUASI	4
BAB V KESIMPULAN DAN SARAN	4
BAB II KAJIAN LITERATUR	5
2.1 Open Data	5
2.2 CKAN	5
2.3 Web Framework CodeIgniter	6
2.4 Application Programming Interface (API)	6

2.5	<i>Representational State Transfer (REST)</i>	7
2.6	Integrasi Sistem	7
2.7	PT Ciptadra Softindo.....	8
2.8	Pengujian Perangkat Lunak	8
2.8.1	<i>User Acceptance Test (UAT)</i>	9
2.8.2	<i>Black Box Testing</i>	9
2.9	Metodologi Pengembangan Perangkat Lunak	10
2.10	Penelitian Terkait.....	11
2.11	Posisi Penelitian	12
BAB III METODOLOGI PENELITIAN		15
3.1	Tahapan Penelitian	15
3.2	Rancangan Penelitian	16
3.2.1	Jenis Penelitian	16
3.2.2	Metode Analisis Data	17
3.2.3	Metode Pengumpulan Data	17
3.2.4	Metode Pengujian.....	18
3.2.5	Metode Implementasi dan Evaluasi.....	22
3.2.6	Lingkungan Pengembangan.....	24
BAB IV IMPLEMENTASI DAN EVALUASI.....		25
4.1.	Analisis dan Perancangan Sistem	25
4.1.1	Analisis Sistem Berjalan	25
4.1.2	<i>User Stories</i>	26
4.1.3	Identifikasi Aktor	27
4.1.4	Analisis Permasalahan dan Usulan Perbaikan.....	28
4.1.5	<i>Design Use Case Diagram</i>	29
4.1.6	<i>Design Activity Diagram</i>	31
4.1.5	<i>Class Diagram</i>	33
4.1.5	Arsitektur Sistem.....	35

4.1.6	Perancangan Antar Muka	36
4.1.7	Perancangan Pengujian	40
4.2.	Implementasi Sistem <i>Open Data</i> Kota Bekasi.....	45
4.2.1	Implementasi Integrasi <i>API CKAN</i> pada Aplikasi <i>Open Data</i> ..	45
4.2.2	Implementasi Antar Muka	46
4.3.	Pengujian Sistem	51
4.3.1	<i>Black Box Testing</i>	51
4.3.2	<i>User Acceptance Test (UAT)</i>	58
4.4.	Evaluasi Hasil	60
4.4.1	Evaluasi Hasil Uji.....	61
BAB V	62
KESIMPULAN DAN SARAN	62
5.1	Kesimpulan	62
5.2	Saran.....	62
DAFTAR PUSTAKA	64
LAMPIRAN	66

STT - NF

DAFTAR GAMBAR

Gambar 2. 1 Tahapan <i>Extreme Programming</i>	10
Gambar 3. 1 Tahapan Penelitian	15
Gambar 3. 2 Fase-fase model <i>waterfall</i>	22
Gambar 4. 1 Alur Pengelolaan <i>Dataset</i>	25
Gambar 4. 2 <i>Use Case Diagram</i>	30
Gambar 4. 3 <i>Diagram Activity Kelola Dataset</i>	31
Gambar 4. 4 <i>Diagram Activity Verifikasi Dataset</i>	32
Gambar 4. 5 <i>Class Diagram</i>	34
Gambar 4. 6 Arsitektur Sistem.....	36
Gambar 4. 7 <i>Mockup Halaman Home</i>	37
Gambar 4. 8 <i>Mockup Halaman List Dataset</i>	37
Gambar 4. 9 <i>Mockup Halaman Tambah Dataset</i>	38
Gambar 4. 10 <i>Mockup Halaman Edit Dataset</i>	38
Gambar 4. 11 <i>Mockup Halaman Detail Dataset</i>	39
Gambar 4. 12 <i>Mockup Halaman Verifikasi Dataset</i>	39
Gambar 4. 13 <i>Mockup Halaman Hapus Dataset</i>	40
Gambar 4. 14 <i>Mockup Halaman Tampil Dataset di Portal</i>	40
Gambar 4. 15 Implementasi <i>API CKAN</i> pada Aplikasi <i>Open Data</i>	45
Gambar 4. 16 Tampilan Halaman <i>Home</i>	46
Gambar 4. 17 Tampilan Halaman <i>List Dataset</i>	47
Gambar 4. 18 Tampilan Halaman <i>Tambah Dataset</i>	47
Gambar 4. 19 Tampilan Halaman <i>Edit Dataset</i>	48
Gambar 4. 20 Tampilan Halaman <i>Detail Dataset</i>	49
Gambar 4. 21 Tampilan Halaman <i>Verifikasi Dataset</i>	49
Gambar 4. 22 Tampilan Halaman <i>Hapus Dataset</i>	50
Gambar 4. 23 Halaman <i>Tampil Dataset</i> pada Portal	51
Lampiran 1 <i>Codingan Fungsi List Dataset</i>	66
Lampiran 2 <i>Codingan Fungsi Tambah Dataset</i>	66
Lampiran 3 <i>Codingan Fungsi Edit Dataset</i>	66
Lampiran 4 <i>Codingan Fungsi Detail Dataset</i>	66
Lampiran 5 <i>Codingan Fungsi Hapus Dataset</i>	67



STT - NF

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	11
Tabel 2. 2 Posisi Penelitian	12
Tabel 3. 1 Metode Pengujian <i>Black Box Testing</i>	18
Tabel 4. 2 Perancangan <i>Black Box Testing</i> – Login Pengguna	41
Tabel 4. 3 Perancangan <i>Black Box Testing</i> – Penambahan <i>Dataset</i>	41
Tabel 4. 4 Perancangan <i>Black Box Testing</i> – Edit <i>Dataset</i>	42
Tabel 4. 5 Perancangan <i>Black Box testing</i> – Verifikasi <i>Dataset</i>	43
Tabel 4. 6 Perancangan <i>Black Box Testing</i> – Menghapus <i>Dataset</i>	43
Tabel 4. 7 Perancangan <i>Black Box Testing</i> – Publikasi <i>Dataset</i>	44
Tabel 4. 8 <i>Black Box Test</i> – Login Pengguna	52
Tabel 4. 9 <i>Black Box Test</i> – Tambah <i>Dataset</i>	52
Tabel 4. 10 <i>Black Box Test</i> - Edit <i>Dataset</i>	53
Tabel 4. 11 <i>Black Box Test</i> - Verifikasi <i>Dataset</i>	54
Tabel 4. 12 <i>Black Box Test</i> - Hapus <i>Dataset</i>	55
Tabel 4. 13 <i>Black Box Test</i> - Publikasi <i>Dataset</i>	57
Tabel 4. 14 <i>User Acceptance Testing (UAT)</i> - Admin	58
Tabel 4. 15 <i>User Acceptance Test (UAT)</i> - OPD	59

STT - NF

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan pesat dalam teknologi informasi, didukung oleh infrastruktur internet, telah memasuki berbagai aspek kehidupan, terutama dalam konteks peningkatan pelayanan publik yang disediakan oleh pemerintah kepada masyarakat [1]. Perkembangan ini tidak hanya terjadi pada jenis perangkat lunak aplikasi yang mendukung, tetapi juga karena semakin meluasnya penggunaan teknologi informasi ini. Seiring dengan meratanya penggunaan teknologi informasi dikalangan masyarakat semakin memudahkan untuk mengakses informasi, dengan hal tersebut dapat dimanfaatkan oleh pemerintah Kota Bekasi untuk mempublikasikan sebuah informasi dan data secara terbuka.

Implementasi keterbukaan data oleh pemerintah Kota Bekasi mengenai sebuah informasi dan data dapat dilakukan dengan adanya Aplikasi *Open Data* Kota Bekasi, melalui Aplikasi *Open Data* Kota Bekasi mempublikasikan data secara *online* dalam format yang dapat diakses oleh masyarakat, keterlibatan pemerintah dalam menyediakan data terbuka dapat meningkatkan akuntabilitas, transparansi data, partisipasi masyarakat dalam mengawal pembangunan, efisiensi operasional dan inovasi pada Kota Bekasi.

Oleh karena itu, penelitian ini berfokus untuk merancang pengembangan sistem sebagai solusi yang efisien dalam mempublikasikan data dengan mengintegrasikannya dengan *CKAN* sebagai sumber data berbasis *web framework Codeigniter*. Solusi ini diharapkan akan memberikan kemudahan bagi para perangkat daerah Kota Bekasi untuk mempublikasikan informasi data mereka dan dapat dengan mudah diakses dan dimanfaatkan oleh masyarakat. Berdasarkan latar belakang yang telah disampaikan, hal ini mendorong penulis untuk melakukan penelitian yang berjudul

“Pengembangan Aplikasi Open Data Kota Bekasi Terintegrasi *CKAN* Menggunakan Web Framework Codeigniter”.

1.2 Rumusan Masalah

Dari latar belakang diatas, maka penulis mencoba menyajikan suatu Pengembangan Aplikasi *Open Data* Kota Bekasi Terintegrasi *CKAN* Menggunakan *Web Framework CodeIgniter*. Penguraian masalah dapat dirinci sebagai berikut yaitu:

1. Bagaimana rancangan aplikasi *Open Data* Kota Bekasi Terintegrasi *CKAN* Menggunakan *Web Framework Codeigniter* ?
2. Bagaimana efektifitas dari rancangan aplikasi *Open Data* Kota Bekasi Terintegrasi *CKAN* Menggunakan *Web Framework Codeigniter* ?

1.3 Tujuan dan Manfaat Penelitian

Dari latar belakang dan rumusan masalah diatas, maka tujuan dan manfaat dari pengembangan aplikasi *Open Data* Kota Bekasi terintegrasi *CKAN* dengan *web framework Codeigniter* adalah :

1.3.1 Tujuan Penelitian

1. Membuat rancangan dan mengembangkan aplikasi *Open Data* Kota Bekasi Terintegrasi *CKAN* Menggunakan *Web Framework Codeigniter*.
2. Mengetahui efektifitas dari aplikasi *Open Data* Kota Bekasi Terintegrasi *CKAN* Menggunakan *Web Framework Codeigniter*.

1.3.2 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut :

1. Bagi Penulis

Sebagai sarana untuk menerapkan konsep dan teori yang dipelajari dalam mata kuliah khususnya dalam perancangan dan pemrograman sistem.

2. Bagi Pemerintah Kota Bekasi

- a. Solusi yang diusulkan dapat membantu dalam pengolahan *dataset* dan mempercepat masyarakat dalam mengetahui informasi data yang telah dipublikasikan.

- b. Penggunaan *CKAN* akan memudahkan penyimpanan dan mendukung pembangunan aplikasi dan layanan berbasis data yang inovatif.

1.4 Batasan Masalah

Penelitian ini menekankan pada pengembangan frontend untuk pengolahan *dataset*. Batasan berikut dibuat untuk menentukan ruang lingkup penelitian :

1. Sistem yang akan dikembangkan akan berfokus pada pengolahan *dataset* mulai dari penginputan, pengeditan sampai memvalidasi data sebelum data akan dipublikasikan ke publik.
2. Sistem akan integrasi dengan *CKAN* akan menggunakan *API* yang telah ada tanpa membuat, modifikasi atau perubahan signifikan pada *API* tersebut.
3. Sistem akan berfokus pada data yang terpublikasikan dan tersedia melalui aplikasi *open data*.
4. Pengembangan sistem dilakukan di PT Ciptadra Softindo sebagai penyedia pelayanan sistem dalam melakukan pengembangan sistem.

1.5 Sistematika Penulisan

Dalam penyusunan tugas akhir ini, penulisan akan mengikuti urutan bab yang mencakup berbagai aspek penelitian. Berikut adalah struktur penulisan yang digunakan dalam laporan ini :

BAB I PENDAHULUAN

Bab Pendahuluan ini bertujuan untuk memberikan konteks yang mencakup latar belakang penelitian, penjelasan rumusan masalah, tujuan penelitian, manfaat hasil penelitian dan batasan ruang lingkup yang menjadi fokus utama penelitian.

BAB II KAJIAN LITERATUR

Pada bab ini akan membahas teori-teori yang mendukung pengembangan fitur, metode pengembangan perangkat lunak,

tinjauan pustaka terkait, serta penelitian ini akan dijelaskan dalam konteks kerangka konseptual yang lebih luas.

BAB III METODE PENELITIAN

Pada bab Metode Penelitian ini akan menguraikan mengenai desain penelitian yang dipilih, alat dan teknik yang digunakan, serta langkah-langkah yang diambil dalam merancang solusi pengolahan *dataset* pada aplikasi *open data*.

BAB IV IMPLEMENTASI DAN EVALUASI

Pada bab ini akan berfokus pada eksekusi proses pengolahan data yang telah diimplementasikan pada aplikasi *open data*, serta melakukan evaluasi terhadap hasil capaian yang berhasil dalam proses pengembangan.

BAB V KESIMPULAN DAN SARAN

Bab ini akan merangkum hasil-hasil dari penelitian, menguraikan kesimpulan yang didapat dan menyajikan rekomendasi yang sesuai untuk penelitian yang mungkin akan dilakukan di masa depan.

STT - NF

BAB II KAJIAN LITERATUR

2.1 *Open Data*

Open Data adalah konsep yang menandakan bahwa data dapat diakses oleh masyarakat secara bebas. Prinsip ini sejalan dengan prinsip-prinsip terbuka lainnya seperti sumber terbuka, pemerintah terbuka dan universitas terbuka. Dalam konteks pemerintahan, tujuan dari *Open Data* adalah untuk meningkatkan tingkat transparansi, akuntabilitas dan partisipasi masyarakat [2].

Selain itu, *Open Data* berfokus untuk memberikan akses terbuka terhadap informasi yang dihasilkan oleh berbagai entitas yang memungkinkan masyarakat, peneliti, pengembang dan organisasi untuk menjelajahi, menganalisis dan memanfaatkan informasi tersebut. *Open Data* berperan sebagai layanan publik dengan mempublikasikan data secara terbuka dan dapat diakses oleh semua pihak. Dengan demikian, *Open Data* tidak hanya menjadi sumber informasi, tetapi juga menjadi alat untuk memperkuat keterlibatan publik dan efektivitas layanan.

2.2 *CKAN*

CKAN adalah sebuah sistem manajemen data (*DMS*) yang bersifat terbuka, dirancang untuk mendukung agregasi dan penyediaan data melalui portal data. *CKAN* mempermudah proses publikasi, berbagi dan pemanfaatan data dengan mendukung ratusan portal data yang tersebar di seluruh dunia [3].

Selain itu, *CKAN* dirancang untuk memfasilitasi penyimpanan, manajemen dan distribusi *Open Data*. Fitur utama pada *CKAN* yaitu :

1. *Dataset*

Setiap *dataset* berisi metadata yang mencakup penjelasan, jenis data, perizinan serta informasi penting lainnya.

2. *Metadata*

Metadata pada *CKAN* adalah informasi yang mendeskripsikan atau menjelaskan *dataset*. Setiap metadata berisi tag, kategori,

pembuat, tanggal pembuatan, serta informasi tambahan yang membantu dalam memahami isi *dataset* yang tersedia.

3. Pencarian

CKAN memiliki fitur pencarian yang memungkinkan pengguna untuk memudahkan dalam menemukan *dataset* yang relevan berdasarkan kata kunci, kategori, tag atau parameter lainnya.

4. Visualisasi

CKAN mendukung untuk visualisasi data, memungkinkan untuk membuat grafik, tabel atau peta berdasarkan *dataset*. [4]

2.3 *Web Framework CodeIgniter*

Dalam proses pengembangan website penggunaan *framework* sangat membantu untuk mempermudah pekerjaan. Dalam hal ini, penulis memutuskan untuk menggunakan *framework CodeIgniter* dengan tujuan untuk mempercepat proses pengembangan dan menerapkan standarisasi pengembangan pada umumnya.

CodeIgniter merupakan sebuah *framework* PHP yang tangguh dengan ukuran yang sangat kecil, dirancang khusus untuk pengembang yang memerlukan alat bantu yang simpel untuk membangun aplikasi *web* yang kaya akan fitur [5]. *CodeIgniter* memiliki kerangka kerja yang hampir tidak memerlukan konfigurasi, kerangka kerja dengan tapak kecil dan tidak mengharuskan mematuhi aturan pengkodean yang ketat. [6]

2.4 *Application Programming Interface (API)*

Application Programming Interface (API) adalah serangkaian pedoman, protokol dan alat yang memfasilitasi interaksi antara berbagai aplikasi perangkat lunak. Dengan *API*, memungkinkan pengembang untuk menggunakan data, fungsi atau layanan yang diperlukan untuk membangun atau menyatukan aplikasi dengan sistem atau platform lain.

Tujuan penggunaan *API* adalah menyederhanakan pembangunan aplikasi dengan menyediakan akses ke data dan fitur tanpa memerlukan penambahan data secara manual. Aplikasi yang dikembangkan akan

memperoleh fitur dari aplikasi tujuan yang terintegrasi. Pengembangan aplikasi menjadi lebih efisien karena pengembang tidak perlu mengalokasikan waktu tambahan untuk membuat fitur yang serupa. Sebagai contoh, pemanfaatan integrasi *API* media sosial, peta, *gateway* pembayaran dan cuaca [7].

2.5 *Representational State Transfer (REST)*

Representational State Transfer (REST) adalah serangkaian prinsip arsitektur yang melakukan transmisi data melalui antarmuka yang terstandarisasi seperti *HTTP*. *REST API* beroperasi seperti aplikasi web konvensional, dimana klien mengirimkan permintaan ke *server* melalui protokol *HTTP* dan *server* merespons kembali kepada klien [8].

Prinsip utama dari *REST* adalah menggunakan metode *HTTP* (*GET*, *POST*, *PUT*, *DELETE*) untuk berkomunikasi dengan sumber daya (*resource*) yang diidentifikasi secara unik melalui *URL* (*Uniform Resource Locator*). Penerapan *REST* memungkinkan pengembang untuk membuat *API* (*Application Programming Interface*) yang terstruktur dan mudah dipahami, yang memungkinkan manipulasi sumber daya dengan efisien melalui protokol *HTTP*. Dengan ini, *REST* menjadi salah satu pendekatan yang populer dalam pengembangan layanan *web* dan integrasi sistem.

2.6 Integrasi Sistem

Integrasi sistem adalah Sistem yang saling terhubung, berkoordinasi, baik di antara subsistem maupun antar sistem, secara terintegrasi menggabungkan data dan sistem informasi yang sebelumnya terpisah [9]. Integrasi sistem adalah proses menggabungkan komponen atau modul perangkat lunak yang terpisah menjadi sebuah sistem yang lengkap dan berfungsi. Hal ini melibatkan pengujian komponen yang terpisah untuk memastikan bahwa mereka berinteraksi dengan benar dan memberikan hasil yang diharapkan ketika digabungkan menjadi satu sistem. Proses integrasi sistem biasanya dilakukan setelah pengembangan dan pengujian unit selesai, sebelum melanjutkan ke pengujian sistem secara menyeluruh. Tujuan dari integrasi sistem adalah untuk memastikan bahwa semua bagian

dari sistem bekerja dengan baik bersama-sama dan memenuhi persyaratan fungsionalitas yang ditetapkan.

2.7 PT Ciptadra Softindo

PT Ciptadra Softindo didirikan pada tahun 1999. Perusahaan ini diperkuat oleh tenaga kerja yang merupakan lulusan dari universitas terkemuka dan memiliki keahlian yang kompeten di bidangnya masing-masing. PT Ciptadra Softindo berfokus pada bidang konsultasi teknologi informasi (TI), menyediakan solusi TI untuk pengelolaan kota, pusat kontak dan asuransi inti. Perusahaan ini memiliki catatan portofolio yang mencakup ratusan proyek sistem informasi dan telah terbukti membantu meningkatkan produktivitas, pendapatan, efisiensi dan kepuasan pelanggan. Misi utamanya adalah menyederhanakan tugas *end user*, meningkatkan kepuasan, memberikan pengalaman kerja yang lebih menyenangkan dan menghasilkan produk dengan standar kualitas yang tinggi, sedangkan dalam visinya yaitu membuat perusahaan konsultan TI menjadi yang terkemuka di Asia Tenggara dengan standar kualitas internasional [10].

PT Ciptadra Softindo memiliki pemahaman yang mendalam tentang proses bisnis dalam berbagai industri. Selain itu, PT Ciptadra Softindo hadir untuk memberikan solusi dalam bidang IT untuk mendukung terlaksananya konsep *smart city* pada pemerintahan daerah untuk meningkatkan layanan publik lebih mudah, cepat dan efisien. Salah satu mitra yang bekerjasama dengan PT Ciptadra Softindo ialah Pemerintah Kota Bekasi pada Dinas Komunikasi, Informatika, Statistik dan Persandian Kota Bekasi.

2.8 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses yang dilakukan untuk mendeteksi dan mengidentifikasi kecacatan serta masalah pada suatu aplikasi. Tujuannya adalah untuk mengurangi dampak negatif yang mungkin timbul akibat kegagalan sistem tersebut [11]. Tujuan utamanya adalah untuk menjamin bahwa perangkat lunak memenuhi kualitas yang telah ditetapkan, keandalan dan perangkat lunak beroperasi sesuai dengan fungsi yang sudah ditetapkan.

Proses pengujian perangkat lunak adalah rangkaian langkah terorganisir yang dilakukan secara sistematis untuk menilai kualitas, fungsionalitas dan kinerja perangkat lunak. Proses ini mencakup tahapan mulai dari perencanaan, pembuatan skenario pengujian, eksekusi pengujian, hingga analisis hasil dan penyusunan laporan.

2.8.1 User Acceptance Test (UAT)

Hasil dari pengujian perangkat lunak *User Acceptance Testing (UAT)* meliputi evaluasi terhadap kelayakan penerimaan perangkat lunak oleh pengguna atau pihak berkepentingan. Ini dapat berupa penerimaan atau penolakan perangkat lunak berdasarkan kriteria penerimaan yang telah ditetapkan. Hasilnya didokumentasikan dalam laporan yang mencakup temuan, kesimpulan, dan rekomendasi, serta umpan balik dari pengguna yang dapat digunakan untuk perbaikan terakhir sebelum perilisan resmi. Hasil *UAT* juga menentukan kesiapan perangkat lunak untuk digunakan secara operasional dalam lingkungan produksi.

2.8.2 Black Box Testing

Black box testing adalah salah satu teknik pengujian perangkat lunak yang digunakan untuk menguji fungsionalitas perangkat lunak, baik pada tingkat unit-unit kecil maupun hasil integrasi keseluruhan. Dalam metode ini, pengujian dilakukan tanpa pengetahuan internal tentang struktur atau kode sumber perangkat lunak yang diuji, melainkan berfokus pada input dan output yang dihasilkan untuk memastikan bahwa perangkat lunak berjalan sesuai dengan yang diharapkan [13].

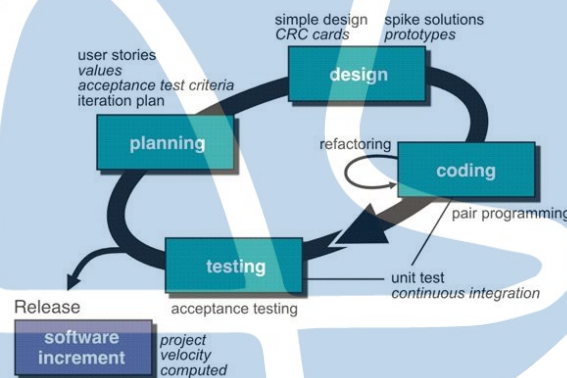
Pengujian *Black Box* fokus pada antarmuka, fungsionalitas, dan alur fungsi yang diperlukan oleh pengguna dalam aplikasi. Dalam pengujian ini, tidak ada pengujian yang dilakukan dengan menganalisis kode sumber program. Tujuan utama dari pengujian perangkat lunak *Black Box Testing* adalah untuk mengevaluasi fungsionalitas, kinerja, dan kecocokan antarmuka serta alur kerja aplikasi dari perspektif pengguna.

Dalam *Black Box Testing* untuk *API CKAN*, pengujian dilakukan

untuk menilai fungsionalitas dan respons *API* tanpa memperhatikan detail implementasi internalnya. Dengan ini, mencakup pengujian validasi input, respons sesuai dengan spesifikasi yang diharapkan, integrasi antarmuka, serta pengujian untuk memastikan keamanan dan keandalan *API*.

2.9 Metodologi Pengembangan Perangkat Lunak

Dalam penelitian ini, teknik pengembangan perangkat lunak yang diterapkan adalah metode *Extreme Programming (XP)*. *Extreme Programming (XP)* adalah sebuah metodologi yang diterapkan dalam pengembangan perangkat lunak dengan tujuan meningkatkan kualitas perangkat lunak dalam menghadapi perubahan dan memenuhi kebutuhan pelanggan [14]. Berikut ini adalah gambar dari struktur *Extreme Programming (XP)* dapat dilihat pada dibawah ini:



Gambar 2. 1 Tahapan *Extreme Programming*

Tahapan *Extreme Programming (XP)* melibatkan empat tahap utama dalam pengembangan perangkat lunak :

1. Perencanaan (*planning*): Tahap ini melibatkan pemahaman mendalam terhadap kebutuhan pengguna serta merencanakan proses pengembangan perangkat lunak.
2. Desain (*design*): Di tahap ini, dilakukan perancangan prototipe dan antarmuka yang akan digunakan dalam pengembangan perangkat lunak.

3. Pengkodean (*coding*): Proses penulisan kode program, pengujian kecil berulang dan integrasi kode ke dalam sistem terjadi pada tahap ini.
4. Pengujian (*testing*): Setelah kode program selesai dikembangkan, dilakukan pengujian menyeluruh untuk memastikan kualitas dan fungsionalitas perangkat lunak.

2.10 Penelitian Terkait

Dalam penelitian ini, kajian literatur yang relevan melibatkan penelusuran literatur sebelumnya yang telah membahas topik serupa dengan penelitian yang sedang dilakukan. Tujuan dari penelitian terkait adalah untuk menginvestigasi dan mengevaluasi kebutuhan dalam menyelesaikan masalah, serta memperoleh pemahaman tentang posisi penelitian yang telah dilakukan sebelumnya, informasi terkait penelitian ini dapat ditemukan dalam tabel berikut :

Tabel 2. 1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	Dewangga dkk., 2022	Program Jakarta Open Data Sebagai E-Government Dalam Pelayanan Informasi Publik Humas Pemrov DKI Jakarta	Implementasi <i>Open Data</i>	Pemprov DKI Jakarta	Meningkatkan transparansi data dan keterbukaan informasi publik[15].
2	Bustomi dkk., 2021	Inisiatif Open Data pada Usaha Mikro Kecil Menengah Kota Bandung	Implementasi <i>Open Data</i>	UMKM	Metode implementasi <i>Open Data</i> bagi umkm [16]
3	Mulyant	Pembanguna	Portal <i>Open</i>	Pemerint	Mengembangka

	o dkk., 2022	n Portal Open Data untuk Mendukung Open Government dan Smartcity	<i>Data</i>	ah Daerah Kota Gorontalo	n aplikasi portal <i>Open Data</i> untuk meningkatkan akses informasi dan efektivitas pemerintahan [17]
--	-----------------	---	-------------	-----------------------------------	--

Dalam Tabel Penelitian Terkait, dapat diketahui telah dilakukan beberapa penelitian terkait *Open Data*, termasuk studi tentang implementasi open data dalam administrasi pemerintahan daerah serta analisis efeknya terhadap tingkat transparansi dan partisipasi masyarakat.

2.11 Posisi Penelitian

Data dalam penelitian ini meliputi perbedaan dalam teknologi yang digunakan serta subjek penelitian yang berbeda, jika dibandingkan dengan penelitian sebelumnya :

Tabel 2. 2 Posisi Penelitian

No	Nama dan Tahun	<i>Web</i>	Integrasi <i>CKAN</i>	<i>CodeIgniter</i>	<i>Extreme Programming</i>
1.	Dewangga dkk, 2022 Program Jakarta Open Data Sebagai E-Government Dalam Pelayanan Informasi Publik Humas Pemprov DKI Jakarta	✓			
2.	Bustomi dkk., 2021	✓	✓		

	Inisiatif Open Data pada Usaha Mikro Kecil Menengah Kota Bandung				
3.	Putri dkk., 2021 Pengukuran Efektivitas dan Efisiensi Penerapan <i>Open Data System</i> Menggunakan Model <i>Delone and Mclean</i>	✓			
4.	Mariyahsari. I, 2024 Pengembangan Aplikasi Open Data Kota Bekasi Terintegrasi CKAN Menggunakan Web Framework CodeIgniter	✓	✓	✓	✓

Dari posisi penelitian yang disajikan, dapat disimpulkan bahwa implementasi *Open Data* di berbagai wilayah, seperti Jakarta, Bandung, dan

Bekasi, memiliki variasi dalam teknologi yang digunakan. Misalnya, Jakarta memiliki program *Open Data* yang dikembangkan sebagai bagian dari *E-Government* dengan dukungan situs *web*, tetapi tidak mengintegrasikan *CKAN* atau menggunakan *framework* seperti *CodeIgniter* atau metode pengembangan *ekstrim*. Di sisi lain, Bandung juga memiliki inisiatif *Open Data* dengan situs *web* dan integrasi *CKAN*, namun tidak menggunakan *CodeIgniter* atau metode pengembangan *ekstrim*. Sementara itu, Bekasi membangun portal *Open Data* dengan integrasi *CKAN* dan menggunakan *framework CodeIgniter* serta metode pengembangan *ekstrim*.

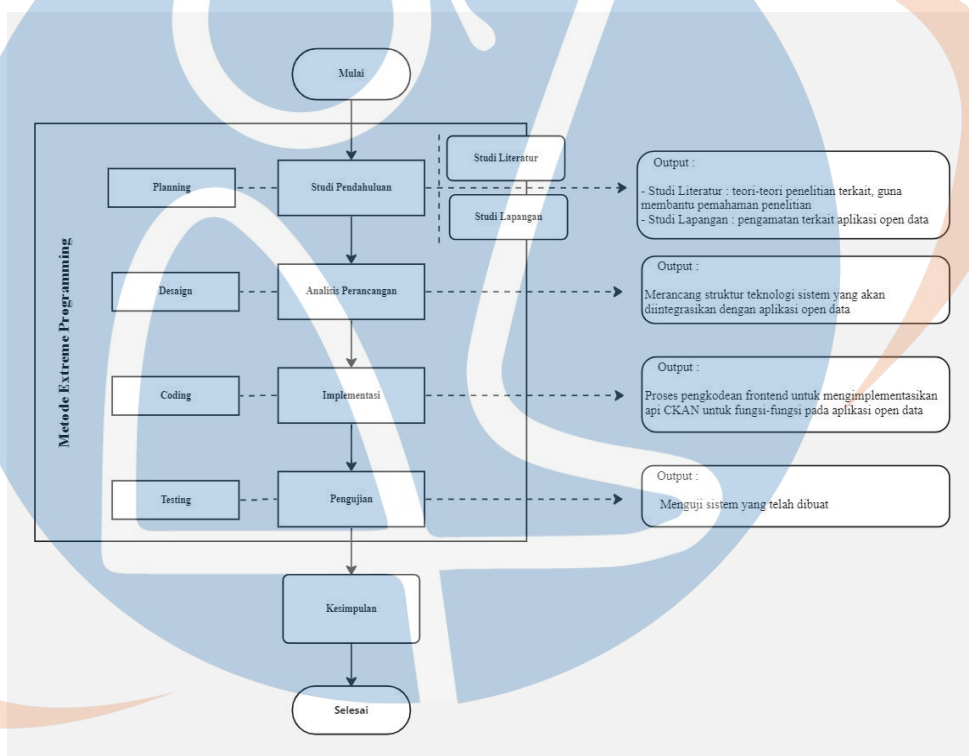
Dari perbandingan tersebut, dapat disimpulkan bahwa implementasi *Open Data* bervariasi tergantung pada wilayahnya, dengan beberapa wilayah lebih maju dalam penggunaan teknologi dan metodologi pengembangan. Penggunaan teknologi seperti *CKAN* dan pengembangan aplikasi dengan menggunakan *framework* seperti *CodeIgniter* dan metode *ekstrim programming* dapat memperkuat implementasi *Open Data* dan mendukung visi pemerintahan terbuka serta *Smart City*. Namun, penting juga untuk memperhatikan konteks dan kebutuhan lokal dalam mengembangkan solusi *Open Data* yang efektif.

STT - NF

BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Pada tahapan penelitian ini, berbagai langkah proses dijalankan sesuai dengan tujuan penelitian yang ditetapkan. Tahapan penelitian telah dijelaskan secara rinci oleh penulis untuk setiap proses.



Gambar 3. 1 Tahapan Penelitian

Pada Gambar 3.1 Diagram ini menunjukkan alur Metode *Extreme Programming* (XP) yang menggambarkan proses pengembangan sistem dengan Metode *Extreme Programming* (XP), dimulai dari **Perancangan**, dimana langkah-langkah proyek direncanakan . dilanjutkan dengan **Studi Pendahuluan** yang mencakup **Studi Literatur** untuk meneliti teori-teori terkait dan pengamatan yang relevan. Tahap berikutnya adalah **Desain**, merancang struktur teknologi sistem yang diintegrasikan dengan aplikasi *Open Data*. Kemudian , melalui **Analisis Perancangan**, desain tersebut

dianalisis. Tahap **Coding** melibatkan perkodean *frontend* untuk mengimplementasikan *API CKAN*, diikuti dengan Implementasi desain ke dalam sistem. Setelah itu, **Pengujian** dilakukan untuk memastikan sistem berfungsi dengan baik, dengan output berupa hasil pengujian. Akhirnya, kesimpulan diambil pada tahap **Kesimpulan**, dan proyek diakhiri pada tahap **Selesai**. Diagram ini menunjukkan bagaimana *Extream Programming* menggunakan *input* dari studi literatur dan lapangan untuk mengembangkan aplikasi *Open Data* secara efektif.

3.2 Rancangan Penelitian

Rancangan penelitian merinci rencana yang akan dilaksanakan dalam penelitian tersebut. Rancangan penelitian melibatkan strategi untuk menghadapi potensi hambatan, jenis penelitian yang dipilih dan metode pengumpulan data.

3.2.1 Jenis Penelitian

Jenis penelitian yang akan dilakukan dalam penelitian ini adalah penelitian pengembangan (*development research*). Penelitian pengembangan ini berfokus pada pengembangan solusi atau inovasi baru yang dapat mengatasi masalah atau meningkatkan kinerja suatu sistem, dalam situasi ini yaitu integrasi *CKAN* menggunakan *web framework CodeIgniter*. Penelitian ini melibatkan proses perancangan, pengembangan, dan pengujian suatu teknologi yang bertujuan untuk mengatasi berbagai tantangan yang terkait dengan penggunaan *open data*. Hasil dari penelitian ini meliputi :

1. Pengembangan aplikasi *Open Data* yang terintegrasi *CKAN* menggunakan *web framework CodeIgniter* telah berhasil diselesaikan.
2. Integrasi dengan *CKAN* memberikan keunggulan dalam manajemen dan publikasi data
3. Penggunaan *web framework CodeIgniter* mempercepat proses pengembangan aplikasi.

4. Hasil penelitian menunjukkan bahwa aplikasi tersebut mampu menyajikan data secara efisien dan terstruktur.
5. Kesimpulan dari penelitian ini menegaskan bahwa aplikasi tersebut dapat menjadi solusi yang efektif dalam pengolahan dan pemanfaatan data terbuka.

3.2.2 Metode Analisis Data

Metode analisis data yang digunakan dalam penelitian ini adalah metode kualitatif. Menurut Creswell dalam (Murdiyanto, 2020), penelitian kualitatif didefinisikan sebagai tahap investigasi terhadap fenomena sosial dan permasalahan manusia [18].

Metode ini dipilih karena penelitian ini menerapkan pengujian aplikasi sebagai cara untuk mengevaluasi keefektifan aplikasi melalui menguji aplikasi dan wawancara. Dengan metode ini, dapat diperoleh pemahaman yang lebih mendalam tentang efektivitas penggunaan aplikasi dan kecenderungan pengguna dalam memanfaatkan dan mengakses data terbuka yang diberikan oleh aplikasi tersebut.

3.2.3 Metode Pengumpulan Data

Pada penelitian ini, tahapan pengumpulan data dan informasi melalui beberapa tahapan seperti :

1. Wawancara

Wawancara adalah proses penting untuk mengumpulkan informasi data yang detail mengenai pengalaman dan pandangan individu tentang penggunaan aplikasi *Open Data*. Data yang terkumpul harus dianalisis dengan seksama untuk mengidentifikasi pola, tema, dan temuan kunci. Temuan-temuan kunci ini memberikan kontribusi bagi pemahaman yang lebih dalam tentang subjek penelitian.

2. Observasi

Aktivitas observasi dalam pengembangan aplikasi *Open Data* Kota Bekasi mencakup pemantauan langsung pada web *Open Data* Kota

Bekasi <https://danta.bekasikota.go.id/> dan proses terkait pengumpulan, pengelolaan, dan pemanfaatan data terbuka di Kota Bekasi, dengan tujuan untuk memahami praktik eksisting, kebutuhan pengguna, serta potensi perbaikan dan pengembangan dalam pengimplementasian aplikasi *Open Data* tersebut.

3. Studi Kepustakaan

Aktivitas studi kepustakaan dalam pengembangan aplikasi *Open Data* Kota Bekasi melibatkan penelusuran dan analisis terhadap sumber-sumber literatur, dokumentasi, dan riset terkait terkait untuk memperoleh pemahaman yang mendalam tentang kebutuhan, tantangan, dan praktik terbaik dalam pengembangan aplikasi *Open Data*, serta mempelajari konteks khusus dan karakteristik kota Bekasi untuk menghasilkan solusi yang sesuai dan efektif.

3.2.4 Metode Pengujian

Pada tahap pengujian, pengujian dilakukan menggunakan *User Acceptance Testing* (UAT). Pada tahap UAT adalah tahap pengujian dimana aplikasi atau sistem diuji oleh pengguna akhir untuk memastikan bahwa sudah sesuai dengan kebutuhan dan siap digunakan. Sebelum mencapai tahap UAT, aplikasi akan menjalani pengujian *black box testing* untuk memastikan fungsionalitasnya.

1. *Black Box Testing*

Pengujian *black box testing* dilakukan dengan menguji fungsionalitas aplikasi tanpa perlu mengetahui rincian kode didalamnya. Ini melibatkan pengujian fitur-fitur aplikasi untuk memastikan aplikasi berjalan sesuai dengan harapan dan kebutuhan. Dalam konteks ini, pengujian akan mencakup verifikasi integrasi dengan *CKAN*, pengelolaan *dataset*, dan fungsi lainnya terkait dengan aplikasi *Open Data* Kota Bekasi, dengan fokus pada input, output, dan fungsi lainnya.

Tabel 3. 1 Metode Pengujian *Black Box Testing*

No	Pengujian	<i>End Point</i>	Hasil Yang Diharapkan	Kesimpulan

1.	<i>Login</i> dengan <i>Username</i> dan <i>Password</i> yang benar.	http://localhost/op en-data-bekasi- 2021/account	Akan langsung masuk ke dalam aplikasi <i>CMS Open Data</i> .	Sesuai.
2.	<i>Login</i> dengan <i>Username</i> dan <i>Password</i> yang salah.	http://localhost/op en-data-bekasi- 2021/account	Akan muncul pesan " <i>Username dan Password Salah!</i> ".	Tidak sesuai.
3.	Menambah <i>dataset</i> dengan data yang benar dan menambah data yang wajib diisi (data visibilitas, organisasi dan file).	http://localhost/op en-data-bekasi- 2021/Home/ckan Custom	Akan muncul pesan " <i>Berhasil !, Data Berhasil Ditambah</i> ".	Sesuai.
4.	Menambah <i>dataset</i> dengan data yang salah dan	http://localhost/op en-data-bekasi- 2021/Home/ckan Custom	Akan muncul pesan " <i>Silahkan Lengkapi Data</i> ".	Tidak sesuai.

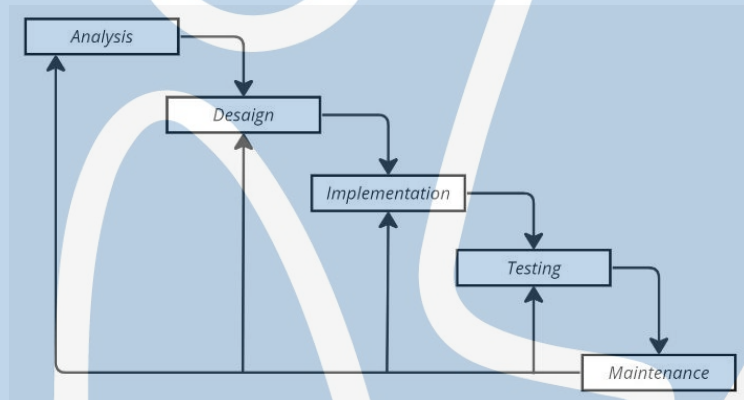
	tidak menambahkan data yang wajib diisi (data visibilitas, organisasi dan file).			
5.	Mengubah <i>dataset</i> dengan data yang ingin diubah dan data tidak boleh kosong (data visibilitas, organisasi dan file).	http://localhost/open-data-bekasi-2021/Home/edit_cakan/{id_dataset}	Akan muncul pesan “Berhasil!, <i>Data</i> berhasil diubah...”.	Sesuai.
6.	Mengubah <i>dataset</i> dengan data yang ingin diubah dan dengan data yang	http://localhost/open-data-bekasi-2021/Home/edit_cakan/{id_dataset}	Akan muncul pesan “Gagal!, <i>Data</i> tidak boleh kosong...”.	Tidak Sesuai.

	kosong (data visibilitas, organisasi dan file).			
7.	Memverifikasi <i>dataset</i> yang ingin dipublikasikan ke publik dengan visibilitas Publik.	http://localhost/open-data-bekasi-2021/Home/CkanCustom	Akan muncul modal untuk memverifikasi <i>dataset</i> jika klik Setuju akan muncul pesan “Berhasil!, Data Berhasil Di Verifikasi”.	Sesuai.
8.	Memverifikasi <i>dataset</i> yang ingin dipublikasikan ke publik dengan visibilitas Publik	http://localhost/open-data-bekasi-2021/Home/CkanCustom	Akan muncul modal untuk memverifikasi <i>dataset</i> jika klik Batal akan muncul pesan langsung membatalkan verifikasi.	Tidak Sesuai.
9.	Menghapus <i>dataset</i> yang ingin dihapus	http://localhost/open-data-bekasi-2021/Home/CkanCustom	Akan muncul pesan untuk menghapus <i>dataset</i> jika klik Ya, Hapus akan muncul pesan “Berhasil!, Data Berhasil Dihapus”.	Sesuai.
10.	Menampil	http://localhost/op	Akan menampilkan	Sesuai

kan <i>dataset</i> yang telah dipublikas ikan oleh admin	en-data-bekasi- 2021/user/dataset	list <i>dataset</i> yang telah dipublikasikan oleh admin melalui cms.	
---	--------------------------------------	---	--

3.2.5 Metode Implementasi dan Evaluasi

Metode implementasi dan evaluasi pada pengembangan aplikasi open data Kota Bekasi yang terintegrasi CKAN menggunakan web framework CodeIgniter dapat dijelaskan secara detail sebagai berikut:



Gambar 3. 2 Fase-fase model *waterfall*

Implementasi

Pada tahap metode implementasi menggunakan metode SDLC dengan model *waterfall*. Model *waterfall* adalah model klasik yang terstruktur secara sistematis dalam pengembangan perangkat lunak. Model *waterfall* adalah model satu arah yang dimulai dari tahap perencanaan hingga pemeliharaan, meliputi tahapan seperti perencanaan, desain sistem, implementasi, verifikasi, dan pemeliharaan [19]. Metode *waterfall* adalah metode yang umum digunakan oleh analis sistem. Proses pengembangan sistem menggunakan metode ini dilakukan secara teratur atau linear. Berarti bahwa jika satu langkah belum selesai, langkah berikutnya tidak dapat dimulai. Misalnya, jika langkah pertama belum selesai, langkah kedua tidak dapat dimulai, begitu seterusnya. Dengan demikian pengerjaan

dilakukan secara berurutan, dimulai dari langkah pertama, kemudian langkah kedua, langkah ketiga, dan seterusnya. Fase-fase dalam metode *waterfall* adalah sebagai berikut :

- Langkah-langkah dalam pengembangan aplikasi *Open Data* menggunakan metode *waterfall* adalah sebagai berikut : Tahap awal melibatkan persiapan lingkungan pengembangan yang memadai, termasuk konfigurasi *server* dan *database* yang dibutuhkan.
- Pengembang menggunakan *web framework CodeIgniter* untuk membangun aplikasi dengan menyesuaikan fitur-fitur yang diinginkan, seperti manajemen pengguna, antarmuka pengguna, dan integrasi dengan *CKAN*.
- Data terbuka dari Kota Bekasi diintegrasikan ke dalam aplikasi menggunakan protokol yang sesuai dengan *CKAN*.
- Fitur-fitur tambahan yang dibutuhkan, seperti visualisasi data, pencarian, dan filter, diimplementasikan sesuai kebutuhan.

Evaluasi

- Tahap evaluasi dimulai dengan pengujian fungsionalitas aplikasi menggunakan *black box testing* untuk memastikan bahwa semua fitur berjalan dengan baik.
- Melakukan wawancara untuk mengumpulkan informasi data yang detail mengenai pengalaman dan pandangan individu terhadap aplikasi *Open Data*.
- Metode analisis kualitatif digunakan untuk menganalisis data respons pengguna dan memperoleh pemahaman yang mendalam tentang kinerja aplikasi serta kepuasan dan kebutuhan pengguna.
- Evaluasi berkelanjutan dilakukan dengan memperhatikan umpan balik dari pengguna dan melakukan perbaikan atau peningkatan aplikasi sesuai kebutuhan.

Dengan menggunakan metode implementasi yang terstruktur dan evaluasi yang teliti, pengembangan aplikasi *open data* Kota Bekasi dapat mencapai tujuan yang diinginkan dan memberikan nilai tambah yang signifikan bagi pengguna dan pemangku kepentingan di Kota Bekasi.

3.2.6 Lingkungan Pengembangan

Lingkungan pengembangan mencakup semua yang diperlukan oleh seorang pengembang untuk menulis dan memperbaiki kode, serta untuk melakukan aktivitas pengembangan perangkat lunak secara keseluruhan.

1. Lokasi Penelitian

Lokasi penelitian untuk pengembangan aplikasi *Open Data* Kota Bekasi Terintegrasi *CKAN* Menggunakan *Web Framework CodeIgniter* bisa mencakup tempat di mana tim pengembangan bekerja, pada kantor PT Ciptadra Softindo yang digunakan sebagai pusat pengembangan aplikasi.

2. Kebutuhan Alat

- ASUS Model Name : P243OU
- Windows 11 Pro 64-bit
- RAM 8GB
- Intel(R) Core(TM) i3-6006U CPU @2.000GHz (4CPUs)

3. Kebutuhan Bahan

- Text Editor : Visual Studio Code
- CKAN Version : CKAN 2.9
- PHP Version : 5.6
- CodeIgniter Version : 3.13
- Database : MySQL
- XAMPP Version : v3.3.0
- Platform API : Postman

STI - NF

BAB IV IMPLEMENTASI DAN EVALUASI

4.1. Analisis dan Perancangan Sistem

Dalam tahap analisis dan perancangan sistem, fokus utama adalah melakukan penelitian mendalam terhadap permasalahan pengelolaan *dataset* yang dilakukan secara langsung pada platform *CKAN*, terutama karena belum menggunakan *Content Management System* (CMS) dalam pengelolaan *dataset*. Sebagai analisis permasalahan, langkah pertama adalah melakukan analisis kebutuhan secara menyeluruh untuk memahami tantangan yang ada. Setelah itu, diusulkan solusi berupa pengembangan *frontend* yang berbasis *CodeIgniter*, dilengkapi dengan layanan pengelolaan *dataset* serta verifikasi *dataset* melalui CMS.

4.1.1 Analisis Sistem Berjalan

Dalam sistem yang berjalan saat ini, pengolahan *dataset* masih dilakukan secara langsung oleh admin melalui platform *CKAN*. Pada tahap ini, admin diwajibkan untuk mengisi *dataset* dan mengunggah *file dataset* secara manual ke *CKAN*. Langkah-langkah ini perlu dilakukan untuk mengelola *dataset*. Namun, proses pengelolaan yang dilakukan langsung di platform *CKAN* menimbulkan kendala karena antarmuka pengguna yang terbatas, sehingga mengurangi kualitas pengalaman pengguna dalam memahami sistem.



Gambar 4. 1 Alur Pengelolaan *Dataset*

Gambar 4.1 Alur Pengelolaan *Dataset* menunjukkan tahapan yang harus diikuti oleh seorang admin untuk mengelola *dataset* dalam *CMS Open Data* Kota Bekasi. Proses dimulai dengan admin masuk ke akun *CMS* menggunakan kredensial yang tepat. Setelah *login*, admin akan diarahkan ke halaman utama sistem, di mana mereka bisa menavigasi berbagai fitur dan opsi yang tersedia. Langkah selanjutnya adalah memilih opsi untuk

mengelola *dataset*, yang meliputi penambahan *dataset* baru, pengeditan *dataset* yang sudah ada, melihat detail *dataset*, atau menghapus *dataset* yang tidak lagi dibutuhkan. Setelah melakukan perubahan yang diperlukan pada *dataset*, admin harus menyimpan perubahan tersebut untuk memastikan bahwa semua data yang dimodifikasi atau ditambahkan tersimpan dengan benar dalam sistem. Dengan mengikuti tahapan ini, admin dapat memastikan bahwa data dalam sistem selalu diperbarui dan dikelola dengan baik.

4.1.2 User Stories

Setelah mewawancarai calon pengguna aplikasi *Open Data* Kota Bekasi, penulis menganalisis data yang dikumpulkan dan menyusun beberapa *user stories*. *User stories* tersebut antara lain sebagai berikut:

No	Sebagai	Saya Ingin	Sehingga
1.	Admin	Melihat semua list <i>dataset</i>	saya dapat mengelola <i>dataset</i> dengan mudah
		<i>Input dataset</i>	saya dapat menambahkan <i>dataset</i> dengan mudah
		Melakukan update <i>dataset</i>	Saya dapat <i>mengupdate dataset</i> dengan mudah
		Menghapus <i>dataset</i>	Saya dapat menghapus <i>dataset</i> dengan mudah
		Memverifikasi <i>dataset</i>	Saya dapat memverifikasi

			<i>dataset</i> dengan mudah
		Mempublikasikan <i>dataset</i>	Saya dapat mempublikasikan <i>dataset</i> ke portal <i>Open Data</i>
2.	OPD	Melihat semua list <i>dataset</i>	saya dapat mengelola <i>dataset</i> dengan mudah
		<i>Input dataset</i>	saya dapat menambahkan <i>dataset</i> dengan mudah
		Melakukan update <i>dataset</i>	Saya dapat <i>mengupdate dataset</i> dengan mudah
		Menghapus <i>dataset</i>	Saya dapat menghapus <i>dataset</i> dengan mudah
		Melihat list <i>dataset</i> yang sudah diverifikasi pada portal	Saya dapat melihat list <i>dataset</i> pada portal <i>Open Data</i>

4.1.3 Identifikasi Aktor

Berdasarkan user stories di atas, penulis mengidentifikasi beberapa aktor yang akan berinteraksi dengan aplikasi Open Data Kota Bekasi. Aktor-aktor tersebut adalah sebagai berikut:

No	Aktor	Deskripsi
1.	Admin	Fungsi admin pada aplikasi <i>Open Data</i> Kota Bekasi meliputi mengunggah, mengedit, dan menghapus <i>dataset</i> ; mengatur visibilitas data publik dan privat; memvalidasi dan memverifikasi data sebelum dipublikasikan; mengelola akun dan hak akses pengguna; serta memantau aktivitas aplikasi untuk memastikan aplikasi berjalan lancar dan data yang disajikan akurat serta dapat diakses oleh masyarakat dengan mudah.
2.	OPD	Fungsi OPD (Organisasi Perangkat Daerah) pada aplikasi <i>Open Data</i> yang memiliki akses untuk mengelola <i>dataset</i> saja meliputi tugas-tugas mengunggah, mengedit, dan menghapus <i>dataset</i> yang relevan dengan bidang mereka. OPD bertanggung jawab memastikan bahwa data yang mereka kelola akurat, lengkap, dan diperbarui secara berkala. Mereka juga harus mematuhi standar dan format data yang ditetapkan untuk memastikan konsistensi dan kualitas data dalam aplikasi.

4.1.4 Analisis Permasalahan dan Usulan Perbaikan

Pengelolaan dataset langsung dari platform CKAN sering menghadapi berbagai kendala, terutama dalam proses pengisian dataset. Selain itu, dataset yang sudah diunggah tidak melalui proses verifikasi terlebih dahulu. Untuk mengatasi kendala dalam pengelolaan dataset, diusulkan solusi berupa implementasi CMS yang dapat mempermudah pengelolaan dataset.

1. Analisis Permasalahan

Proses pengelolaan dataset yang dilakukan langsung pada platform CKAN menghadirkan tantangan, terutama dalam hal efisiensi kerja dan waktu.

Beberapa permasalahan yang telah diidentifikasi meliputi:

a. Kesulitan dalam mengelola *dataset*

Proses pengelolaan dataset termasuk pengisian, pengeditan, hapus langsung pada platform CKAN dirasa kurang efisien dari segi fungsi dan keefisiensi waktu.

b. Tidak adanya fitur verifikasi *dataset*

Tidak adanya fitur verifikasi *dataset* membuat proses validasi data menjadi kurang optimal dan rentan terhadap kesalahan.

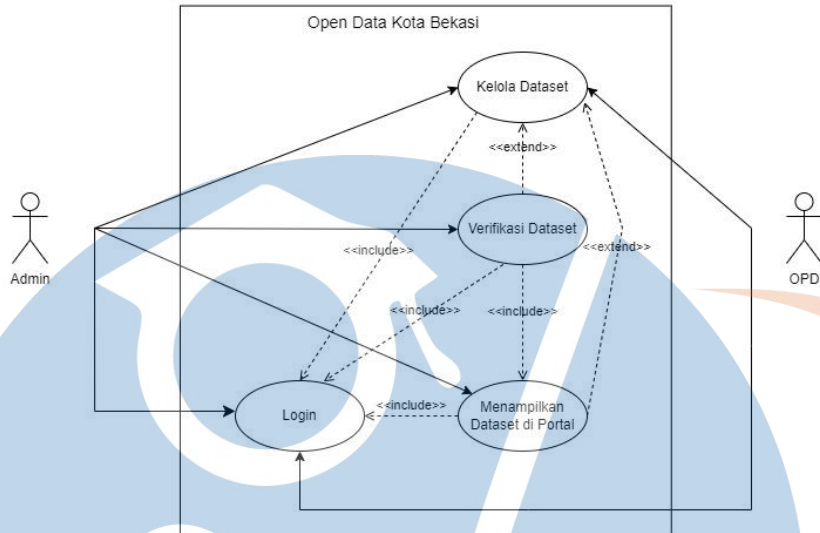
2. Usulan Perbaikan

Untuk mengatasi permasalahan tersebut, diusulkan penerapan sistem *Content Management System* (CMS) pada aplikasi *Open Data* Kota Bekasi. Implementasi CMS ini akan memberikan opsi solusi kepada pengguna dengan mempermudah pengelolaan *dataset*. Dengan menerapkan CMS, pengelolaan *dataset* akan menjadi lebih terstruktur dan sistematis. Selain itu, CMS ini akan dilengkapi dengan fitur verifikasi *dataset* yang efektif, memastikan bahwa data yang tersedia telah melalui proses validasi yang ketat dan dapat dipercaya.

4.1.5 Design Use Case Diagram

Diagram *Use Case* yang berfokus pada *frontend* aplikasi *Open Data* Kota Bekasi dengan pengelolaan *dataset* bertujuan untuk menggambarkan interaksi antara *admin* dan sistem dalam melakukan berbagai tugas pengelolaan *dataset*. Diagram ini memvisualisasikan bagaimana *admin* mengelola *dataset* melalui antarmuka pengguna aplikasi. Diagram *use case* ini sangat penting karena membantu memetakan dan memahami alur kerja dalam pengelolaan *dataset* melalui antarmuka pengguna, memastikan

bahwa semua langkah dan interaksi yang diperlukan diidentifikasi dengan jelas. Hal ini sangat berguna dalam perancangan, pengembangan, dan peningkatan sistem aplikasi *Open Data Kota Bekasi*.



Gambar 4. 2 Use Case Diagram

Pada gambar 4.2 *Use Case Diagram* tersebut menunjukkan interaksi antara dua aktor utama, yaitu Admin dan OPD (Organisasi Perangkat Daerah), dengan sistem "*Open Data Kota Bekasi*." Aktor Admin memiliki peran yang lebih luas dalam sistem, termasuk kemampuan untuk *login*, mengelola *dataset*, memverifikasi *dataset*, dan menampilkan *dataset* di portal. *Use case* "*Kelola Dataset*" bagi Admin mencakup berbagai aktivitas seperti menambah, mengubah, dan menghapus *dataset*.

Sementara itu, aktor OPD memiliki kemampuan yang lebih terbatas dalam sistem. OPD hanya dapat melakukan *login* dan mengelola *dataset*, yang berarti mereka dapat melakukan aktivitas dasar seperti menambahkan dan memperbarui *dataset*. Namun, OPD tidak memiliki akses untuk memverifikasi *dataset* atau menampilkan *dataset* di portal, yang merupakan tanggung jawab Admin.

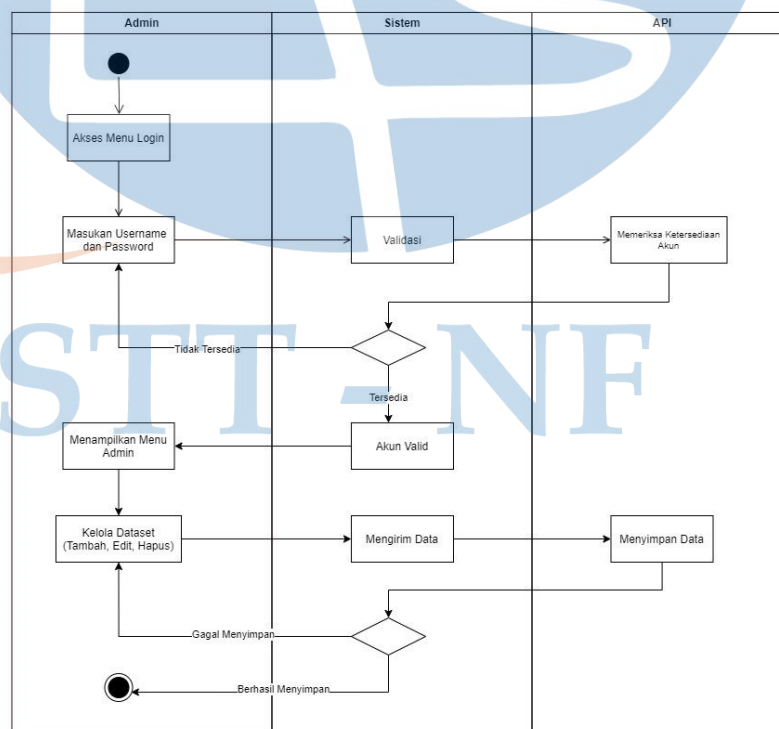
Dengan demikian, peran Admin adalah untuk memastikan integritas dan visibilitas data dengan memverifikasi dan menampilkan *dataset* di portal, sementara OPD bertugas untuk menyediakan dan memperbarui data yang mereka kelola. Diagram ini dengan jelas menunjukkan pembagian

tugas dan tanggung jawab antara Admin dan OPD dalam pengelolaan data di sistem "*Open Data Kota Bekasi*."

4.1.6 Design Activity Diagram

Activity Diagram adalah jenis diagram UML yang dipakai dalam merancang aplikasi, termasuk aplikasi *Open Data* untuk Kota Bekasi. Tujuannya adalah untuk menggambarkan secara visual alur kerja atau proses bisnis yang terdapat dalam aplikasi, mulai dari tahap awal hingga tahap akhir. Diagram ini berperan penting dalam membantu pengembang dan pemangku kepentingan lainnya memahami bagaimana aliran data berlangsung melalui sistem dan bagaimana berbagai komponen saling berinteraksi. Dalam kerangka aplikasi *Open Data* untuk Kota Bekasi, *Activity Diagram* dipakai untuk menunjukkan bagaimana *dataset* diproses. Penggunaan *Activity Diagram* memungkinkan pengembang memastikan bahwa tiap langkah dalam pengelolaan *dataset* berjalan dengan efisien dan tanpa kendala.

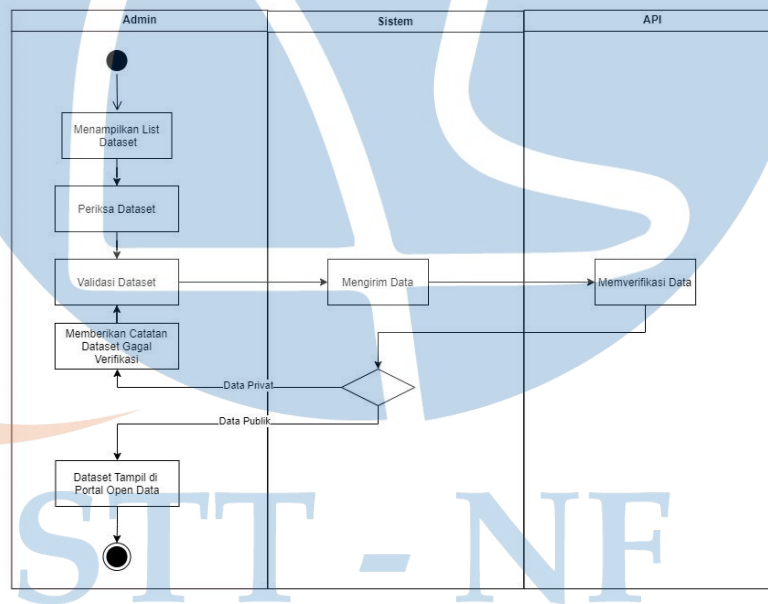
1. Diagram Activity Kelola Dataset



Gambar 4. 3 Diagram Activity Kelola Dataset

Pada Gambar 4.3 Diagram *Activity Kelola Dataset*, proses dimulai ketika *admin* mengakses menu *login*. *Admin* kemudian memasukkan *username* dan *password* yang akan divalidasi oleh sistem. Sistem mengirim permintaan ke *API* untuk memeriksa ketersediaan akun. Jika akun tidak tersedia, proses kembali ke *Admin* untuk memasukkan kembali *username* dan *password*. Namun, jika akun tersedia dan valid, sistem menampilkan menu *admin* kepada *Admin*. Setelah itu, *Admin* dapat mengelola *dataset* yang mencakup penambahan, pengeditan, dan penghapusan data. Sistem kemudian mengirimkan data yang telah dikelola oleh *Admin* ke *API* untuk disimpan. Jika penyimpanan data gagal, *Admin* diberitahu dan proses kembali ke langkah pengelolaan *dataset*. Jika penyimpanan data berhasil, proses selesai dan data berhasil disimpan.

2. Diagram *Activity Verifikasi Dataset*



Gambar 4. 4 Diagram *Activity Verifikasi Dataset*

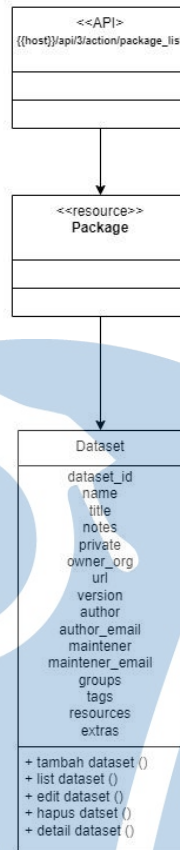
Pada Gambar 4.4 Diagram *Activity Verifikasi Dataset*, proses dimulai dengan sistem yang menampilkan daftar *dataset* yang ada. Setelah itu, *admin* memeriksa *dataset* dan melanjutkan dengan validasi untuk memastikan *dataset* memenuhi standar yang ditetapkan. Jika validasi berhasil, data dikirim ke sistem yang berinteraksi dengan *API* untuk

verifikasi data. *API* kemudian menerima dan memverifikasi data untuk memastikan akurasi dan kepercayaannya. Berdasarkan hasil verifikasi, sistem menentukan apakah data akan dipublikasikan sebagai data publik atau tetap bersifat *privat*. Jika data tidak lolos verifikasi, *admin* mencatat kegagalan verifikasi yang akan dikirim ke OPD dan proses validasi dilakukan kembali. Jika data lolos verifikasi dan dinyatakan sebagai data publik, *dataset* tersebut akan ditampilkan di portal *Open Data*, sehingga dapat diakses oleh publik.

4.1.5 *Class Diagram*

Diagram *class* untuk aplikasi *Open Data* Kota Bekasi yang terintegrasi dengan *API CKAN* berfungsi untuk menggambarkan struktur objek serta hubungan antar objek dalam sistem tersebut. Diagram ini memvisualisasikan kelas-kelas inti yang membentuk aplikasi, mencakup atribut dan metode yang ada pada setiap kelas. Selain itu, diagram class memperlihatkan relasi antar kelas, yang membantu dalam memahami bagaimana data dari *API CKAN* diakses, dikelola, dan digunakan dalam aplikasi.

STT - NF



Gambar 4. 5 Class Diagram

Pada Gambar 4.5 *Class Diagram*, Diagram diatas menggambarkan struktur dan alur interaksi *API* untuk aplikasi *Open Data* Kota Bekasi yang terintegrasi dengan *CKAN*. Diagram ini memperlihatkan beberapa *resource* utama yang berhubungan dengan pengelolaan *dataset* dalam *CKAN*. Di bagian atas diagram, terlihat *endpoint API* utama yaitu `{{host}}/api/3/action/package_list`, yang berfungsi untuk mengakses daftar paket data. Di bawah *endpoint* utama ini, terdapat *resource* `Package` yang menjadi induk dari berbagai operasi pengelolaan *dataset*. *Resource* ini memiliki beberapa turunan, yaitu `tambah dataset`, `list dataset`, `detail dataset`, `update dataset`, dan `hapus dataset`, masing-masing merepresentasikan operasi *CRUD* (*Create, Read, Update, Delete*) pada *dataset*.

- **Tambah Dataset:** *Resource* ini digunakan untuk menambahkan *dataset* baru dengan atribut seperti `dataset_id`, `name`, `title`, `notes`,

`private`, `owner_org`, `url`, `version`, `author`, `author_email`, `maintainer`, `maintainer_email`, `groups`, `tags`, `resources`, dan `extras`. Operasi ini menggunakan metode *POST* dengan *body* berisi data *dataset* yang akan ditambahkan.

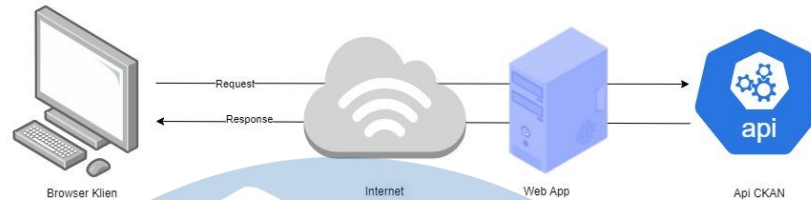
- **List Dataset:** *Resource* ini digunakan untuk mengambil daftar *dataset* yang tersedia menggunakan metode *GET* tanpa parameter tambahan.
- **Detail Dataset:** *Resource* ini digunakan untuk mengambil detail dari dataset tertentu berdasarkan `package_id`. Operasi ini menggunakan metode *GET* dengan `package_id` sebagai parameter.
- **Update Dataset:** *Resource* ini memungkinkan pembaruan informasi *dataset* yang ada dengan atribut yang sama seperti pada `tambah dataset`. Pembaruan dilakukan menggunakan metode *POST* dengan *body* berisi data yang diperbarui.
- **Hapus Dataset:** *Resource* ini digunakan untuk menghapus *dataset* berdasarkan `dataset_id`. Operasi ini juga menggunakan metode *POST* dengan *body* berisi `dataset_id` dari *dataset* yang akan dihapus.

Secara keseluruhan, diagram ini menjelaskan bagaimana setiap *resource* dan operasi berinteraksi dalam sistem, memungkinkan pengelolaan *dataset* secara efektif dalam aplikasi *Open Data* yang terintegrasi dengan *API CKAN*.

4.1.5 Arsitektur Sistem

Fungsi arsitektur sistem pada bagian CMS (*Content Management System*) aplikasi *Open Data* Kota Bekasi adalah untuk mengelola konten dan data yang akan ditampilkan dan diakses oleh pengguna. CMS ini memungkinkan administrator untuk melakukan berbagai operasi seperti penambahan, pengeditan, penghapusan, dan pengelolaan dataset. Melalui antarmuka yang *user-friendly*, administrator dapat mengorganisir *dataset*, menambahkan *metadata*, mengatur kategori, dan memastikan bahwa data yang tersedia selalu *up-to-date* dan relevan. Selain itu, CMS juga menyediakan fitur untuk memantau dan mengontrol akses pengguna, serta memastikan integritas dan keamanan data. Dengan adanya CMS,

pengelolaan data terbuka menjadi lebih efisien, terstruktur, dan mudah diakses oleh publik, mendukung transparansi dan akuntabilitas pemerintah Kota Bekasi.



Gambar 4. 6 Arsitektur Sistem

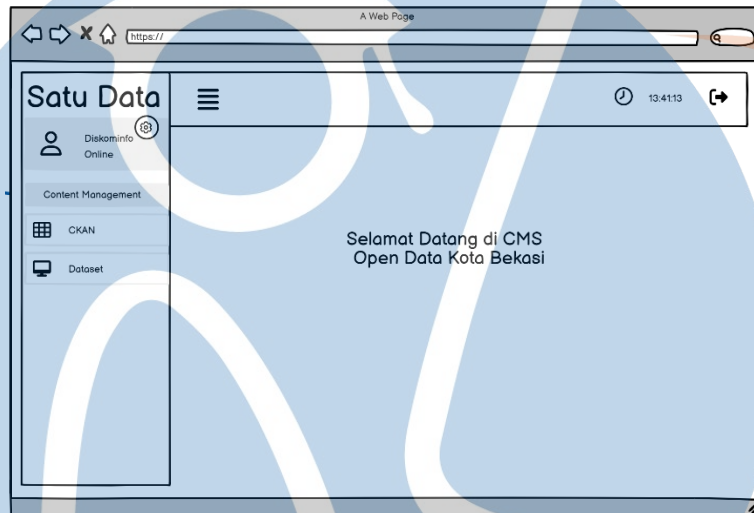
Pada Gambar 4.6 Arsitektur Sistem, menggambarkan alur kerja yang menunjukkan bagaimana sebuah aplikasi web berinteraksi dengan API CKAN melalui internet untuk melayani permintaan dari pengguna. Proses dimulai dari pengguna yang mengirimkan permintaan melalui browser klien. Permintaan ini dikirim melalui internet ke aplikasi web. Aplikasi web kemudian memproses permintaan tersebut dan, jika diperlukan, mengirimkan permintaan ke API CKAN untuk mendapatkan data atau menjalankan operasi tertentu. API CKAN memproses permintaan ini dan mengirimkan respons kembali ke aplikasi web. Aplikasi web kemudian memproses data yang diterima dari API CKAN, menyusun respons yang sesuai, dan mengirimkannya kembali ke browser klien melalui internet. Akhirnya, browser klien menerima respons tersebut dan menampilkannya kepada pengguna. Diagram ini menggambarkan hubungan dan aliran data antara klien, aplikasi web, dan API CKAN dalam sebuah arsitektur aplikasi berbasis web.

4.1.6 Perancangan Antar Muka

Fungsi perancangan antarmuka untuk aplikasi *Open Data* Kota Bekasi pada CMS adalah untuk menyediakan platform yang intuitif dan *user-friendly* bagi administrator dalam mengelola *dataset* kota secara efektif. Antarmuka ini memungkinkan pengguna untuk dengan mudah mengunggah, mengedit, dan menghapus data, serta mengatur kategori dan metadata yang relevan. Dengan desain yang jelas dan navigasi yang

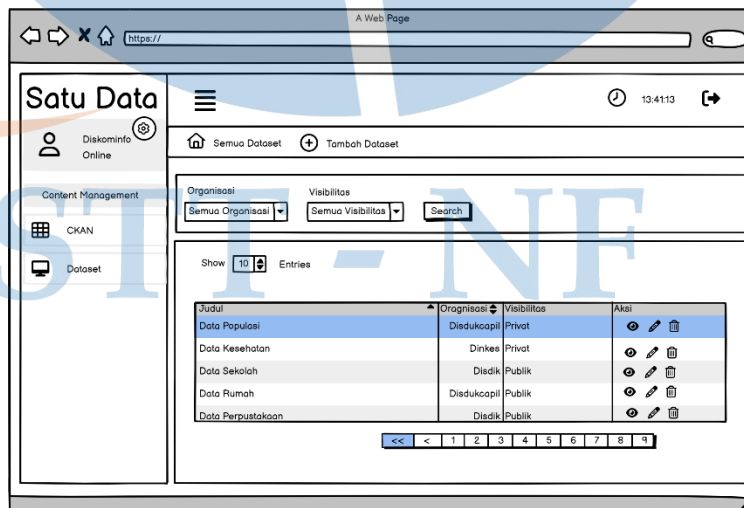
sederhana, antarmuka ini memastikan bahwa proses pengelolaan data dapat dilakukan dengan efisien tanpa memerlukan keahlian teknis yang mendalam. Selain itu, antarmuka ini juga mendukung fitur pencarian dan filter yang canggih, memungkinkan pengguna untuk dengan cepat menemukan dan mengakses data yang diperlukan, sehingga memudahkan penyebaran informasi dan transparansi data kepada publik. Berikut antarmuka pada aplikasi *CMS Open Data Kota Bekasi* :

1. *Mockup Halaman Home*



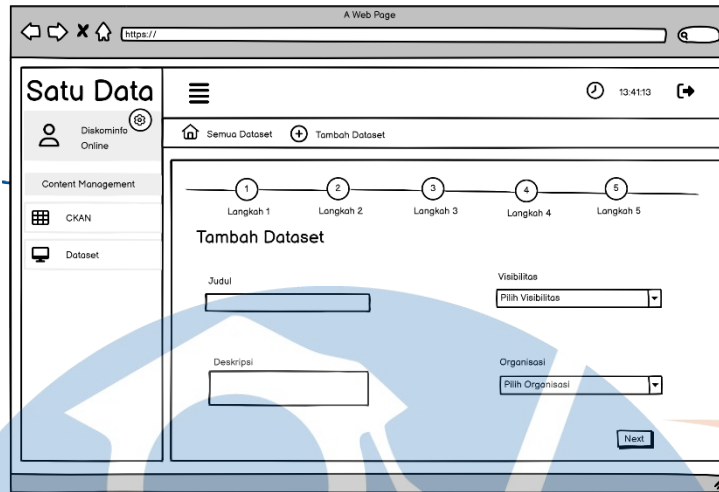
Gambar 4. 7 *Mockup Halaman Home*

2. *Mockup Halaman List Dataset*



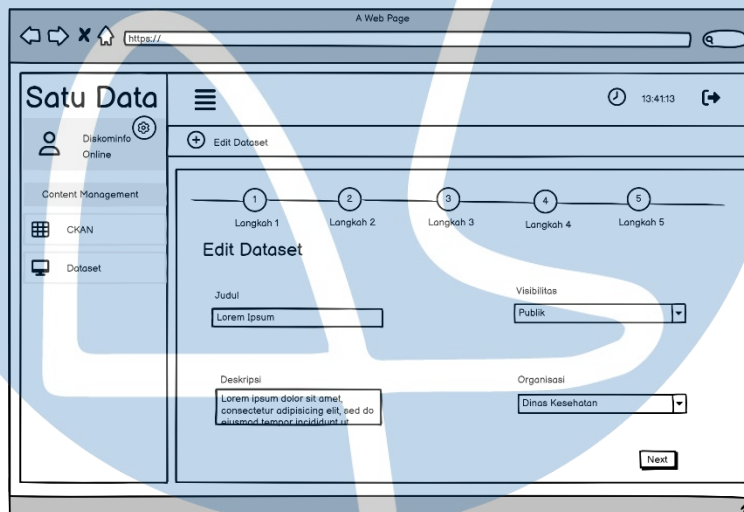
Gambar 4. 8 *Mockup Halaman List Dataset*

3. Mockup Halaman Tambah Dataset



Gambar 4. 9 Mockup Halaman Tambah Dataset

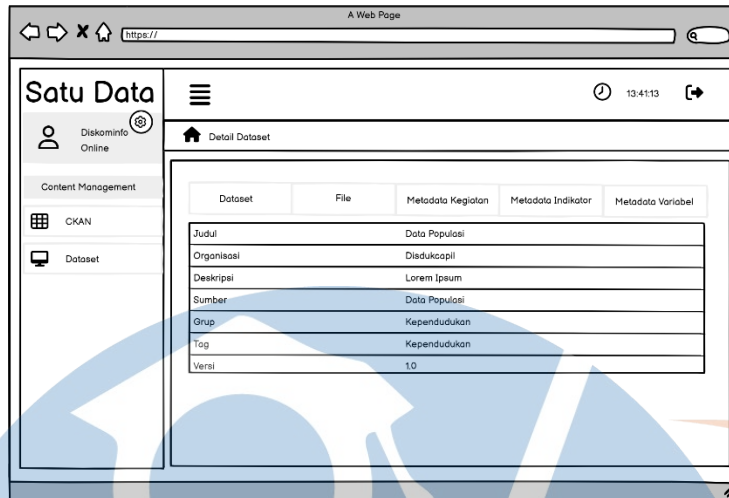
4. Mockup Halaman Edit Dataset



Gambar 4. 10 Mockup Halaman Edit Dataset

STT - NF

5. Mockup Halaman Detail Dataset



Gambar 4. 11 Mockup Halaman Detail Dataset

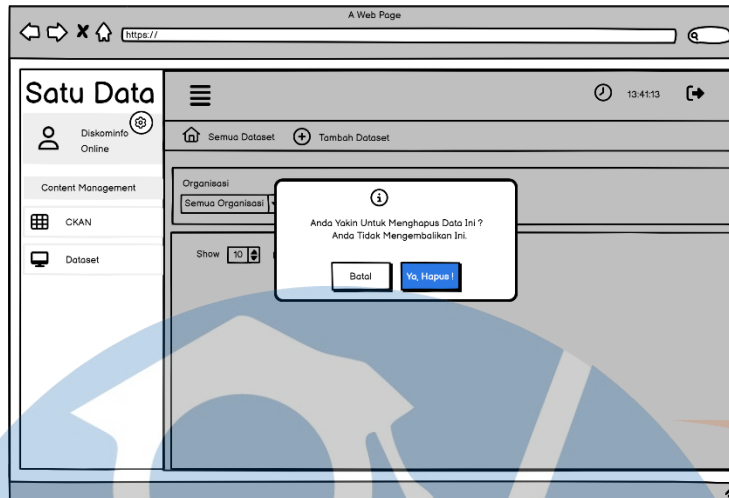
6. Mockup Halaman Verifikasi Dataset



Gambar 4. 12 Mockup Halaman Verifikasi Dataset

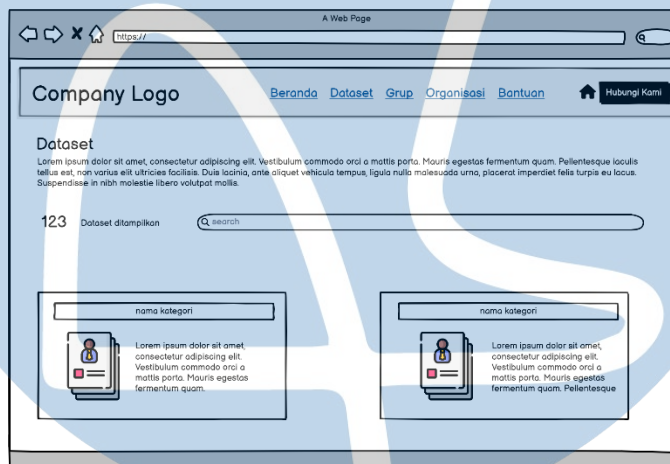
STT - NF

7. Mockup Halaman Hapus Dataset



Gambar 4. 13 Mockup Halaman Hapus Dataset

8. Mockup Halaman Tampil Dataset di Portal



Gambar 4. 14 Mockup Halaman Tampil Dataset di Portal

4.1.7 Perancangan Pengujian

Berdasarkan pembahasan sebelumnya, penelitian ini menggunakan metode pengujian *black box*. Perancangan pengujian *black box* ini menerapkan skala dalam *User Acceptance Testing (UAT)* untuk memastikan bahwa sistem dapat memenuhi kebutuhan pengguna.

Berikut ini adalah skenario perancangan pengujian *black box testing* untuk pengembangan aplikasi *Open Data Kota Bekasi* yang disajikan dalam bentuk tabel:

Tabel 4. 1 Perancangan Black Box Testing – Login Pengguna

Skenario Uji		<i>Login Admin</i>	
End Point		http://localhost/open-data-bekasi-2021/account	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Login dengan <i>Username</i> dan <i>Password</i> yang benar.	Akan langsung masuk ke dalam aplikasi <i>CMS Open Data</i> .	Sesuai atau Tidak Sesuai
2.	<i>Login</i> dengan <i>Username</i> dan <i>Password</i> yang salah.	Akan muncul pesan “ <i>Username dan Password Salah!</i> ”.	Sesuai atau Tidak sesuai.

Dalam Tabel 4.1 Perancangan *Black Box Testing – Login Pengguna*, Perancangan pengujian *login* ke dalam sistem mencakup pengujian fitur *login* dengan berbagai skenario. Pengujian ini mencakup skenario di mana pengguna memasukkan *username* dan *password* yang benar untuk memastikan mereka dapat berhasil masuk ke dalam sistem. Selain itu, juga mencakup skenario di mana pengguna memasukkan *username* dan *password* yang salah untuk memastikan sistem memberikan pesan kesalahan yang sesuai.

Tabel 4. 2 Perancangan Black Box Testing – Penambahan Dataset

Skenario Uji		Penambahan <i>Dataset</i>	
End Point		http://localhost/open-data-bekasi-2021/Home/ckanCustom	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Menambah <i>dataset</i> dengan data yang benar dan menambah data yang wajib di isi (data visibilitas, organisasi dan <i>file</i>).	Akan muncul pesan “Berhasil <i>!, Data Berhasil Ditambah!</i> ”.	Sesuai atau Tidak Sesuai
2.	Menambah <i>dataset</i>	Akan muncul pesan “Silahkan	Sesuai

dengan data yang salah dan tidak menambah data yang wajib di isi (data visibilitas, organisasi dan <i>file</i>).	Lengkapi <i>Data</i> ".	atau Tidak sesuai.
---	-------------------------	--------------------

Dalam Tabel 4.2 Perancangan *Black Box Testing* – Penambahan *Dataset*, Perancangan pengujian tambah *dataset* ke dalam sistem mencakup pengujian fitur penambahan *dataset*, dimulai dari memastikan bahwa penambahan *dataset* dengan data yang benar berfungsi dengan baik hingga menguji fitur penambahan *dataset* ketika pengisian *dataset* tidak sesuai.

Tabel 4. 3 Perancangan *Black Box Testing* – Edit *Dataset*

Skenario Uji		Edit <i>Dataset</i>	
<i>End Point</i>		http://localhost/open-data-bekasi-2021/Home/edit_ikan/{id_dataset}	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Mengubah <i>dataset</i> dengan data yang ingin diubah dan data tidak boleh kosong (data visibilitas, organisasi dan <i>file</i>).	Akan muncul pesan “Berhasil!, <i>Data</i> berhasil diubah...”.	Sesuai atau Tidak Sesuai
2.	Mengubah <i>dataset</i> dengan data yang ingin diubah dan dengan data yang kosong (data visibilitas, organisasi dan <i>file</i>).	Akan muncul pesan “Gagal!, <i>Data</i> tidak boleh kosong...”.	Sesuai atau Tidak sesuai.

Dalam tabel 4.3 Perancangan *Black Box Testing* – Pengeditan *Dataset*, Perancangan pengujian edit *dataset* dalam sistem mencakup pengujian fitur

pengeditan *dataset*, mulai dari memastikan bahwa pengeditan *dataset* dengan data yang benar berfungsi dengan baik hingga menguji fitur pengeditan *dataset* ketika data yang diubah tidak sesuai.

Tabel 4. 4 Perancangan *Black Box testing* – Verifikasi *Dataset*

Skenario Uji		Memverifikasi <i>Dataset</i>	
End Point		http://localhost/open-data-bekasi-2021/Home/CkanCustom	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Memverifikasi <i>dataset</i> yang ingin dipublikasikan ke publik dengan visibilitas Publik.	Memverifikasi <i>dataset</i> yang ingin dipublikasikan ke publik dengan visibilitas Publik.	Sesuai atau Tidak Sesuai
2.	Memverifikasi <i>dataset</i> yang ingin dipublikasikan ke publik dengan visibilitas Publik	Akan muncul modal untuk memverifikasi <i>dataset</i> jika klik Batal akan muncul pesan langsung membatalkan verifikasi.	Sesuai atau Tidak sesuai.

Dalam tabel 4.4 Perancangan *Black Box Testing* – Memverifikasi *Dataset*, Perancangan pengujian verifikasi *dataset* dalam sistem mencakup pengujian fitur verifikasi *dataset*, mulai dari memastikan bahwa verifikasi *dataset* dengan data yang benar berfungsi dengan baik, hingga menguji fitur verifikasi ketika *dataset* yang diverifikasi tidak sesuai.

Tabel 4. 5 Perancangan *Black Box Testing* – Menghapus *Dataset*

Skenario Uji		Menghapus <i>Dataset</i>	
End Point		http://localhost/open-data-bekasi-2021/Home/CkanCustom	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Menghapus <i>dataset</i> yang ingin dihapus (klik “Ya, Hapus”)	Akan muncul pesan untuk menghapus <i>dataset</i> jika klik Ya, Hapus akan	Sesuai atau Tidak Sesuai

		muncul pesan “Berhasil!, Data Berhasil Dihapus”.	
2.	Menghapus <i>dataset</i> yang ingin dihapus (klik “Batal”)	Akan muncul pesan untuk menghapus <i>dataset</i> jika klik Batal maka akan kembali ke halaman list <i>dataset</i> yang telah dipublikasikan oleh <i>admin</i> melalui <i>cms</i> .	Sesuai atau Tidak sesuai.

Dalam tabel 4.5 Perancangan *Black Box Testing* – Menghapus *Dataset*, Perancangan pengujian hapus *dataset* dalam sistem mencakup pengujian fitur hapus *dataset*, mulai dari memastikan bahwa hapus *dataset* berfungsi dengan baik, hingga menguji fitur hapus ketika *dataset* yang dihapus tidak sesuai.

Tabel 4. 6 Perancangan *Black Box Testing* – Publikasi *Dataset*

Skenario Uji		Publikasi <i>Dataset</i>	
<i>End Point</i>		http://localhost/open-data-bekasi-2021/user/dataset	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Menampilkan <i>dataset</i> yang telah diverikasi oleh admin akan dipublikasikan ke publik.	Akan menampilkan list <i>dataset</i> yang telah dipublikasikan oleh admin melalui <i>cms</i> .	Sesuai atau Tidak sesuai.
2.	Menampilkan <i>dataset</i> yang belum diverikasi oleh admin maka tidak terpublikasikan ke publik.	Tidak akan menampilkan list <i>dataset</i> yang telah dipublikasikan oleh admin melalui <i>cms</i> .	Sesuai atau Tidak sesuai.

Dalam tabel 4.6 Perancangan *Black Box Testing* – Publikasi *Dataset*,

Perancangan pengujian publikasi *dataset* dalam sistem mencakup pengujian fitur menampilkan *dataset* yang terpublikasikan, mulai dari memastikan bahwa *dataset* yang dipublikasikan sudah diverifikasi, hingga menguji fitur menampilkan *dataset* yang terpublikasikan yang belum terverifikasi.

4.2. Implementasi Sistem Open Data Kota Bekasi

Implementasi Sistem *Open Data* Kota Bekasi bertujuan untuk memberikan akses yang terbuka dan transparan terhadap data yang dimiliki oleh pemerintah kota, sehingga dapat dimanfaatkan oleh masyarakat, peneliti, dan pengembang aplikasi. Sistem ini menggunakan platform *CKAN* (*Comprehensive Knowledge Archive Network*) untuk mengelola dan mendistribusikan data. Dengan integrasi *API CKAN*, pengguna dapat mengakses berbagai *dataset* yang tersedia secara *real-time* melalui aplikasi *web* yang *user-friendly*.

4.2.1 Implementasi Integrasi API CKAN pada Aplikasi Open Data

Implementasi *API CKAN* pada aplikasi *Open data* Kota Bekasi melibatkan penggunaan *API CKAN* untuk menghubungkan platform data *CKAN* dengan aplikasi *Open Data* Kota Bekasi, memungkinkan akses dan pengelolaan *dataset* publik secara efisien, termasuk pembaruan dan sinkronisasi data secara *real-time*.

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $this->ckanApiUrl . '/api/3/action/package_search');
curl_setopt($ch, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Cookie: ',
    'Authorization: '
));
$output = curl_exec($ch);
```

Gambar 4. 15 Implementasi *API CKAN* pada *Aplikasi Open Data*

Gambar 4.15 Implementasi *API CKAN* pada *Aplikasi Open Data*, alur kerja dari gambar di atas dimulai dengan inisialisasi *cURL*, diikuti dengan pengaturan *URL* tujuan, a bgen pengguna, dan opsi pengembalian transfer. Kemudian, ditambahkan *header* *HTTP* yang diperlukan, termasuk *cookie* dan *token* otorisasi. Setelah semua pengaturan selesai, permintaan dieksekusi menggunakan ``curl_exec()``, dan hasilnya disimpan dalam variabel ``$output``.

4.2.2 Implementasi Antar Muka

Implementasi antarmuka pada CMS aplikasi *Open Data* Kota Bekasi melibatkan pembuatan tampilan yang *user-friendly* untuk mengelola dan mengakses *dataset* publik secara efisien. Antarmuka ini dirancang untuk memudahkan pengguna dalam menelusuri, mencari, dan mengunduh data, serta memungkinkan administrator untuk menambah, memperbarui, dan menghapus *dataset* dengan mudah. Integrasi dengan *CKAN* menggunakan *API* memungkinkan sinkronisasi data secara otomatis, memastikan informasi yang ditampilkan selalu *up-to-date* dan akurat. Dengan demikian, aplikasi *Open Data* ini memberikan akses yang transparan dan mudah kepada masyarakat terhadap data publik Kota Bekasi.

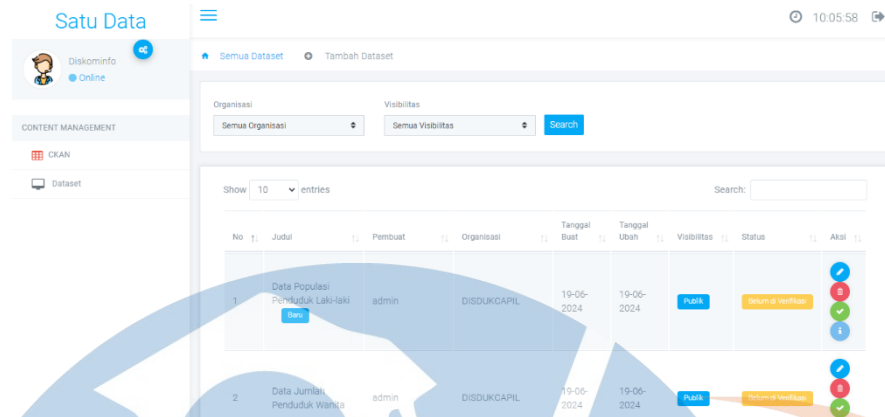
1. Tampilan Halaman *Home*



Gambar 4. 16 Tampilan Halaman *Home*

Gambar 4.16 Tampilan Halaman *Home* adalah tampilan halaman utama dari CMS aplikasi *Open Data* Kota Bekasi, yang menampilkan menu *CKAN* dan *Dataset*. Menu *CKAN* berfungsi untuk ke halaman admin *CKAN*. Menu *Dataset* berfungsi untuk ke halaman *Dataset*.

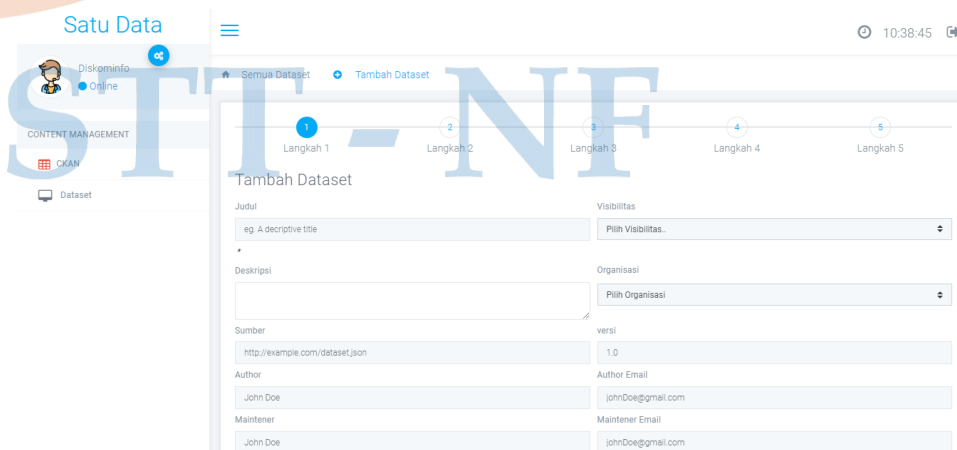
2. Tampilan Halaman List *Dataset*



Gambar 4. 17 Tampilan Halaman List *Dataset*

Gambar 4.17 Tampilan Halaman List *Dataset*, Pada halaman ini pengguna dapat melihat dan mengatur berbagai *dataset* yang ada. Fungsinya mencakup pencarian *dataset* berdasarkan organisasi dan visibilitas, pengelolaan *dataset* seperti menambah, mengedit, dan menghapus data, serta memantau status verifikasi dan visibilitas *dataset*. Halaman ini juga memungkinkan administrator untuk melakukan tindakan verifikasi dan publikasi *dataset*, memastikan bahwa data yang tersedia telah melalui proses validasi yang diperlukan sebelum dapat diakses oleh publik. Dengan demikian, halaman daftar *dataset* memberikan kontrol penuh atas pengelolaan dan penyajian data dalam sistem.

3. Tampilan Halaman Tambah *Dataset*

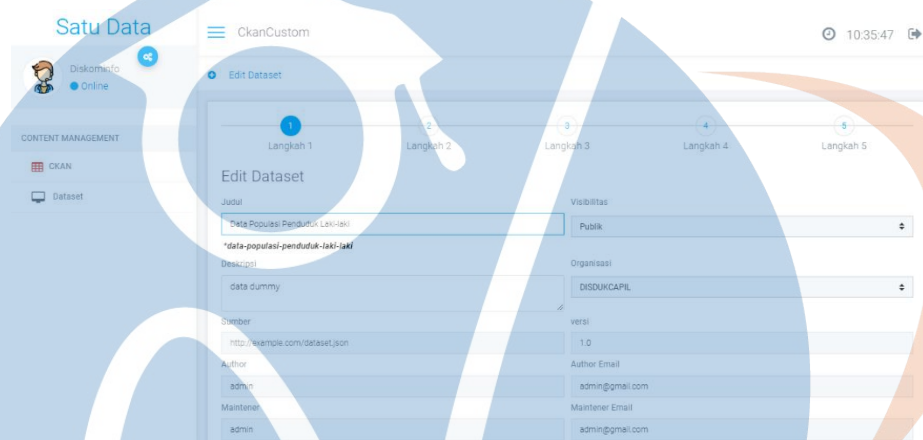


Gambar 4. 18 Tampilan Halaman Tambah *Dataset*

Gambar 4.18 Tampilan Halaman Tambah *Dataset*, melalui halaman

ini, Admin dan OPD dapat memasukkan berbagai informasi terkait *dataset*, seperti judul, deskripsi, kategori, dan sumber data. Pengguna juga dapat mengunggah file data yang relevan dan menentukan visibilitas *dataset*, apakah publik atau privat. Fitur ini memastikan bahwa data yang penting dan terbaru dapat ditambahkan dengan mudah ke dalam sistem, mendukung transparansi dan ketersediaan informasi bagi publik.

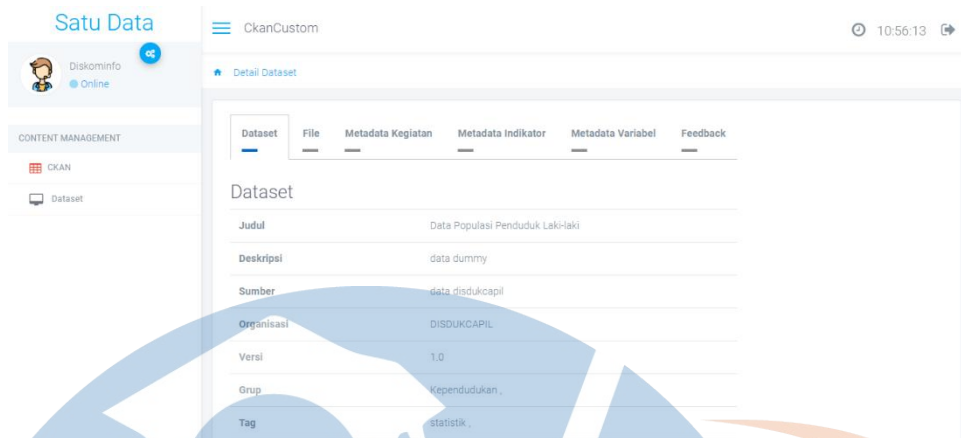
4. Tampilan Halaman Edit *Dataset*



Gambar 4. 19 Tampilan Halaman Edit *Dataset*

Gambar 4.19 Tampilan Halaman Edit *Dataset*, pada halaman ini memungkinkan pengguna yang berwenang, seperti Admin atau OPD, mengubah informasi yang ada dalam *dataset*. Fitur ini memungkinkan pengguna untuk memperbarui *dataset* yang sudah ada, memperbaiki kesalahan, menambahkan informasi baru, atau menghapus data yang sudah tidak relevan. Dengan adanya halaman ini, kualitas dan akurasi data yang disajikan di portal dapat terjaga, memastikan bahwa pengguna data lainnya selalu mengakses informasi yang terkini dan valid. Proses pengeditan ini juga mencakup pengaturan visibilitas *dataset*, sehingga pengguna dapat mengontrol siapa saja yang dapat melihat dataset tersebut.

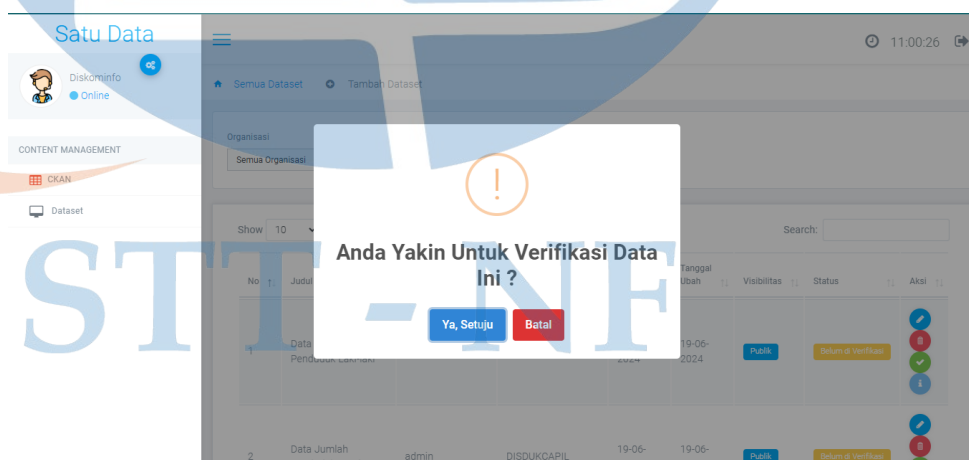
5. Tampilan Halaman Detail Dataset



Gambar 4. 20 Tampilan Halaman Detail Dataset

Gambar 4.20 Tampilan Halaman Detail *Dataset*, pada halaman pengguna dapat melihat berbagai rincian terkait *dataset*, seperti judul, deskripsi, pembuat, organisasi asal, tanggal pembuatan dan pembaruan, status visibilitas (publik atau privat), dan status verifikasi. Fungsionalitas ini membantu memastikan bahwa pengguna dapat memahami konteks dan kualitas data, serta memungkinkan administrasi yang lebih baik melalui tindakan pengelolaan yang diperlukan.

6. Tampilan Halaman Verifikasi *Dataset*

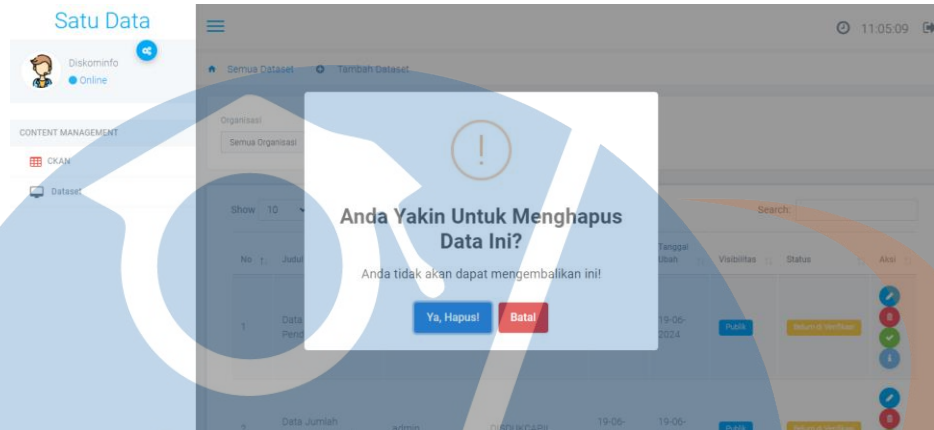


Gambar 4. 21 Tampilan Halaman Verifikasi Dataset

Gambar 4.21 Tampilan Halaman Verifikasi *Dataset* diatas adalah halaman untuk memverifikasi *dataset*, halaman tersebut menunjukkan proses verifikasi *dataset* dalam sistem manajemen *dataset*. Ketika

pengguna mencoba untuk memverifikasi *dataset*, sistem menampilkan kotak dialog konfirmasi untuk memastikan bahwa pengguna yakin dengan tindakan yang akan diambil. Tombol "Ya, Setuju" akan melanjutkan proses verifikasi, sementara tombol "Batal" akan membatalkannya.

7. Halaman Hapus *Dataset*

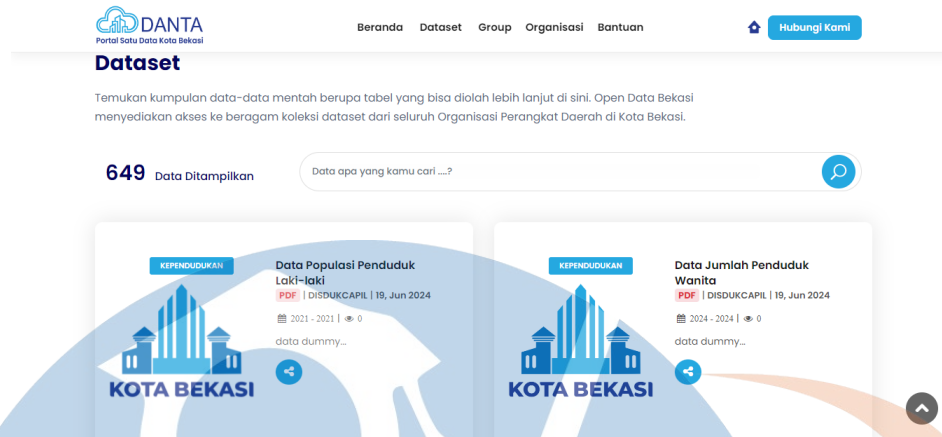


Gambar 4. 22 Tampilan Halaman Hapus *Dataset*

Gambar 4.22 Tampilan Halaman Hapus *Dataser*, pada halaman ini menampilkan kotak dialog konfirmasi penghapusan data. Pada kotak dialog tersebut terdapat pesan "Anda Yakin Untuk Menghapus Data Ini?" dengan peringatan tambahan "Anda tidak akan dapat mengembalikan ini!" Terdapat dua tombol aksi di bagian bawah kotak dialog: "Ya, Hapus!" (berwarna biru) untuk mengonfirmasi penghapusan data, dan "Batal" (berwarna merah) untuk membatalkan tindakan tersebut. Fungsi utama dari kotak dialog ini adalah untuk memastikan bahwa pengguna benar-benar yakin sebelum melakukan tindakan penghapusan *dataset* yang tidak dapat dibatalkan.

STT - NF

8. Halaman Tampil *Dataset* pada Portal



Gambar 4. 23 Halaman Tampil *Dataset* pada Portal

Gambar 4.23 Halaman Tampil *Dataset* pada Portal, gambar diatas adalah halaman portal dari aplikasi *Open Data* Kota Bekasi, pada halaman ini menampilkan list *dataset* yang sudah diverifikasi terlebih dahulu oleh admin dan dapat dikonsumsi oleh publik.

4.3. Pengujian Sistem

Pengujian sistem adalah proses evaluasi menyeluruh yang bertujuan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditetapkan serta memenuhi kebutuhan pengguna. Proses ini mencakup berbagai jenis pengujian yang dilakukan pada tingkat sistem secara keseluruhan, setelah semua komponen individu (unit) dan subsistem (integrasi) telah melewati tahap pengujian masing-masing.

4.3.1 *Black Box Testing*

Dalam penelitian ini, pengujian sistem dilakukan menggunakan metode *black box*. Pengujian ini bertujuan untuk menguji kinerja sistem berdasarkan spesifikasi yang telah dirancang dan membandingkannya dengan hasil yang diperoleh, dengan fokus utama pada fungsionalitas fitur. Skenario pengujian dan uji coba sistem diambil dari perancangan pengujian *black box* sebelumnya, dan hasil pengujian disajikan menggunakan metode *black box testing*.

Tabel 4. 7 Black Box Test – Login Pengguna

Skenario Uji		<i>Login</i> Pengguna	
End Point		http://localhost/open-data-bekasi-2021/account	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	<i>Login</i> dengan <i>Username</i> dan <i>Password</i> yang benar.	Akan langsung masuk ke dalam aplikasi <i>CMS Open Data</i> .	Sesuai
2.	<i>Login</i> dengan <i>Username</i> dan <i>Password</i> yang salah.	Akan muncul pesan “ <i>Username dan Password Salah!</i> ”.	Sesuai

Pada Tabel 4.7 *Black Box Test – Login* Pengguna, pada tabel di atas menunjukkan penerapan pengujian sistem menggunakan metode *black box testing* untuk fitur *login* admin pada aplikasi *CMS Open Data* Bekasi. *Black box testing* dilakukan tanpa melihat ke dalam kode sumber, fokusnya pada pengujian fungsionalitas berdasarkan *input* dan *output* yang diharapkan. Dua skenario diuji: pertama, login dengan *username* dan *password* yang benar di mana hasil yang diharapkan adalah berhasil masuk ke aplikasi; kedua, *login* dengan *username* dan *password* yang salah di mana hasil yang diharapkan adalah munculnya pesan kesalahan. Hasil uji menunjukkan bahwa sistem berfungsi sesuai dengan yang diharapkan dalam kedua skenario tersebut.

Tabel 4. 8 Black Box Test – Tambah Dataset

Skenario Uji		Penambahan <i>Dataset</i>	
End Point		http://localhost/open-data-bekasi-2021/Home/ckanCustom	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Menambah <i>dataset</i> dengan data yang benar dan menambah	Akan muncul pesan “Berhasil <i>!, Data</i> Berhasil Ditambah”.	Sesuai

	data yang wajib di isi (data visibilitas, organisasi dan <i>file</i>).		
2.	Menambah <i>dataset</i> dengan data yang salah dan tidak menambah data yang wajib di isi (data visibilitas, organisasi dan <i>file</i>).	Akan muncul pesan “Silahkan Lengkapi <i>Data</i> ”.	Sesuai

Pada Tabel 4.8 *Black Box Test – Tambah Dataset*, di atas adalah tabel skenario uji untuk fitur penambahan *dataset* pada aplikasi *CMS Open Data* Kota Bekasi menggunakan metode *black box testing*. Tabel ini merinci dua skenario pengujian. Skenario pertama adalah menambahkan *dataset* dengan data yang benar dan lengkap, termasuk data visibilitas, organisasi, dan *file*, dengan hasil yang diharapkan berupa pesan "Berhasil! Data Berhasil Ditambah". Hasil uji menunjukkan bahwa pesan tersebut sesuai dengan harapan. Skenario kedua adalah menambahkan *dataset* dengan data yang salah atau tidak lengkap, di mana hasil yang diharapkan adalah munculnya pesan "Silahkan Lengkapi Data". Hasil uji juga menunjukkan bahwa pesan tersebut sesuai dengan harapan. Pengujian ini fokus pada memvalidasi output yang dihasilkan dari input yang diberikan tanpa memeriksa struktur internal kode aplikasi.

Tabel 4. 9 *Black Box Test - Edit Dataset*

Skenario Uji		Edit <i>Dataset</i>	
End Point		http://localhost/open-data-bekasi-2021/Home/edit_ckan/{id_dataset}	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Mengubah <i>dataset</i> dengan data yang ingin	Akan muncul pesan “Berhasil!, <i>Data</i> berhasil	Sesuai

	diubah dan data tidak boleh kosong (data visibilitas, organisasi dan <i>file</i>).	diubah...".	
2.	Mengubah <i>dataset</i> dengan data yang ingin diubah dan dengan data yang kosong (data visibilitas, organisasi dan <i>file</i>).	Akan muncul pesan "Gagal!, <i>Data</i> tidak boleh kosong...".	Sesuai

Pada Tabel 4.9 *Black Box Test – Edit Dataset*, tabel di atas menunjukkan pengujian sistem menggunakan metode *black box testing* untuk memastikan fungsionalitas perubahan *dataset*. Pengujian pertama dilakukan dengan mengubah *dataset* yang valid, yaitu data yang ingin diubah tidak kosong dan termasuk visibilitas, organisasi, serta *file*. Hasil yang diharapkan dari pengujian ini adalah munculnya pesan "Berhasil!, *Data* berhasil diubah...". Pengujian ini dinyatakan sesuai jika hasil uji menunjukkan hasil yang diharapkan.

Pengujian kedua dilakukan dengan mengubah *dataset* yang tidak valid, yaitu data yang ingin diubah kosong atau tidak mencakup visibilitas, organisasi, dan *file*. Hasil yang diharapkan adalah munculnya pesan "Gagal!, *Data* tidak boleh kosong...". Pengujian ini juga dinyatakan sesuai jika hasil uji menunjukkan hasil yang diharapkan. Dalam kedua pengujian ini, pendekatan *black box testing* digunakan untuk memeriksa *output* yang dihasilkan tanpa memperhatikan struktur internal kode atau logika program.

Tabel 4. 10 *Black Box Test - Verifikasi Dataset*

Skenario Uji		Memverifikasi <i>Dataset</i>	
End Point		http://localhost/open-data-bekasi-2021/Home/CkanCustom	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji

1.	Memverifikasi <i>dataset</i> yang ingin dipublikasikan ke publik dengan visibilitas Publik.	Memverifikasi <i>dataset</i> yang ingin dipublikasikan ke publik dengan visibilitas Publik.	Sesuai
2.	Memverifikasi <i>dataset</i> yang ingin dipublikasikan ke publik dengan visibilitas Publik	Akan muncul modal untuk memverifikasi <i>dataset</i> jika klik Batal akan muncul pesan langsung membatalkan verifikasi.	Sesuai

Pada Tabel 4.10 Black Box Test – Verifikasi Dataset, pada tabel di atas menampilkan pengujian sistem menggunakan metode *black box testing* untuk memverifikasi proses publikasi *dataset* dengan visibilitas publik. Pada pengujian pertama, dilakukan verifikasi terhadap *dataset* yang ingin dipublikasikan dengan visibilitas publik. Hasil yang diharapkan adalah *dataset* tersebut berhasil diverifikasi dan dipublikasikan dengan visibilitas publik. Pengujian ini dinyatakan sesuai jika hasil uji menunjukkan *output* yang diharapkan.

Pengujian kedua dilakukan untuk memverifikasi *dataset* yang juga ingin dipublikasikan dengan visibilitas publik. Dalam pengujian ini, jika pengguna mengklik tombol "Batal" saat modal verifikasi muncul, sistem harus menampilkan pesan bahwa verifikasi telah dibatalkan. Hasil yang diharapkan adalah munculnya pesan yang menegaskan pembatalan verifikasi. Pengujian ini juga dinyatakan sesuai jika hasil uji menunjukkan hasil yang diharapkan. Kedua pengujian ini menggunakan metode *black box testing* untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi tanpa memperhatikan struktur internal atau logika program.

Tabel 4. 11 Black Box Test - Hapus Dataset

Skenario Uji	Menghapus <i>Dataset</i>
End Point	http://localhost/open-data-bekasi-

		2021/Home/CkanCustom	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Menghapus <i>dataset</i> yang ingin dihapus (klik “Ya, Hapus”)	Akan muncul pesan untuk menghapus <i>dataset</i> jika klik Ya, Hapus akan muncul pesan “Berhasil!, Data Berhasil Dihapus”.	Sesuai
2.	Menghapus <i>dataset</i> yang ingin dihapus (klik “Batal”)	Akan muncul pesan untuk menghapus <i>dataset</i> jika klik Batal maka akan kembali ke halaman list <i>dataset</i> yang telah dipublikasikan oleh <i>admin</i> melalui <i>cms</i> .	Sesuai

Pada Tabel 4.11 *Black Box Test – Hapus Dataset*, tabel di atas menampilkan pengujian sistem menggunakan metode *black box testing* untuk memverifikasi fungsi penghapusan *dataset*.

Pengujian pertama dilakukan dengan menghapus *dataset* yang diinginkan, di mana pengguna mengklik "Ya, Hapus". Hasil yang diharapkan adalah munculnya pesan konfirmasi bahwa *dataset* berhasil dihapus dengan pesan "Berhasil!, Data Berhasil Dihapus". Pengujian ini dinyatakan sesuai jika hasil uji menunjukkan pesan yang diharapkan.

Pengujian kedua dilakukan juga untuk menghapus *dataset*, namun kali ini pengguna mengklik "Batal". Hasil yang diharapkan adalah sistem akan kembali ke halaman daftar *dataset* yang telah dipublikasikan oleh admin melalui CMS, tanpa menghapus *dataset* tersebut. Pesan yang menegaskan bahwa penghapusan dibatalkan akan muncul. Pengujian ini juga dinyatakan sesuai jika hasil uji menunjukkan hasil yang diharapkan.

Tabel 4. 12 Black Box Test - Publikasi Dataset

Skenario Uji		Publikasi <i>Dataset</i>	
End Point		http://localhost/open-data-bekasi-2021/user/dataset	
No	Pengujian	Hasil yang Diharapkan	Hasil Uji
1.	Menampilkan <i>dataset</i> yang telah diverifikasi oleh admin akan dipublikasikan ke publik.	Akan menampilkan list <i>dataset</i> yang telah dipublikasikan oleh admin melalui <i>cms</i> .	Sesuai.
2.	Menampilkan <i>dataset</i> yang belum diverifikasi oleh admin maka tidak terpublikasikan ke publik.	Tidak akan menampilkan list <i>dataset</i> yang telah dipublikasikan oleh admin melalui <i>cms</i> .	Sesuai.

Tabel 4.12 *Black Box Test* – Publikasi *Dataset*, tabel diatas berfokus pada validasi fungsionalitas. Dalam pengujian pertama, sistem diuji untuk memastikan bahwa *dataset* yang telah diverifikasi oleh admin ditampilkan kepada publik melalui CMS. Hasil yang diharapkan adalah munculnya daftar *dataset* yang telah dipublikasikan, dan hasil uji menunjukkan bahwa fungsi ini berjalan sesuai dengan harapan. Pada pengujian kedua, sistem diuji untuk memastikan bahwa dataset yang belum diverifikasi oleh admin tidak akan ditampilkan kepada publik. Hasil yang diharapkan adalah tidak adanya daftar dataset yang belum diverifikasi yang muncul di publik, dan hasil uji menunjukkan bahwa fungsi ini juga berjalan sesuai dengan harapan. Kedua pengujian ini menunjukkan bahwa sistem berhasil membedakan antara *dataset* yang diverifikasi dan yang belum diverifikasi dalam hal publikasi.

4.3.2 User Acceptance Test (UAT)

User Acceptance Testing (UAT) pada aplikasi *Open Data* Kota Bekasi melalui *Content Management System (CMS)* bertujuan untuk memastikan bahwa semua fitur penting berfungsi sesuai dengan kebutuhan pengguna. *UAT* ini melibatkan pengujian fungsi pengelolaan *dataset*, verifikasi *dataset*, dan publikasi *dataset*. Dalam pengujian fungsi kelola *dataset*, pengguna memastikan bahwa mereka dapat menambah, mengedit, dan menghapus *dataset* dengan mudah. Pada verifikasi *dataset*, admin menguji apakah mereka dapat memverifikasi *dataset* dengan benar, dan hanya *dataset* yang telah diverifikasi yang bisa dipublikasikan. Terakhir, pada fungsi publikasi *dataset*, pengujian memastikan bahwa *dataset* yang diverifikasi tampil di portal publik dengan benar, sementara *dataset* yang belum diverifikasi tidak ditampilkan. *UAT* ini memastikan bahwa *CMS* memenuhi semua kebutuhan operasional dan fungsional sebelum diluncurkan secara resmi.

1. Admin

UAT ini dilakukan oleh 2 orang admin. Terdapat empat hasil yang diharapkan pada *user* admin.

Tabel 4. 13 *User Acceptance Testing (UAT) - Admin*

No	Aktifitas	Hasil yang Diharapkan	Status (Sesuai/Tidak Sesuai)
1.	<i>Login</i>	Admin dapat <i>login</i> menggunakan <i>email</i> dan <i>password</i> yang benar	Sesuai
		Admin tidak dapat <i>login</i> menggunakan <i>email</i> dan <i>password</i> yang salah	Sesuai
2.	<i>Kelola Dataset</i>	Admin dapat melihat list <i>dataset</i>	Sesuai
		Admin dapat menambah data <i>dataset</i>	Sesuai
		Admin dapat mengedit <i>dataset</i>	Sesuai
		Admin dapat menghapus	Sesuai

		<i>dataset</i>	
		Admin dapat melihat detail <i>dataset</i>	Sesuai
3.	Verifikasi <i>Dataset</i>	Admin dapat melihat status <i>dataset</i>	Sesuai
		Admin dapat memverifikasi <i>dataset</i> jika <i>dataset</i> dengan data benar	Sesuai
		Admin dapat tidak memverifikasi <i>dataset</i> jika <i>dataset</i> dengan data salah	Sesuai
		Admin dapat memberi catatan jika gagal dalam memverifikasi	Sesuai
4.	Publikasi <i>Dataset</i>	Admin dapat mempublikasikan <i>dataset</i> setelah di verifikasi	Sesuai
		Admin dapat melihat <i>dataset</i> yang sudah di publikasikan	Sesuai

Dari hasil pengujian *UAT* yang dilakukan terhadap admin, ditemukan bahwa 4 dari 4 fungsionalitas yang diuji berhasil. Ini menunjukkan bahwa semua fungsionalitas aplikasi pada sisi admin beroperasi dengan sukses, mencapai tingkat keberhasilan 100%.

2. OPD

Tabel 4. 14 User Acceptance Test (*UAT*) - *OPD*

No	Aktifitas	Hasil yang Diharapkan	Status (Sesuai/Tidak Sesuai)
1.	<i>Login</i>	<i>OPD</i> dapat <i>login</i> menggunakan <i>email</i> dan	Sesuai

		<i>password</i> yang benar	
		OPD tidak dapat <i>login</i> menggunakan <i>email</i> dan <i>password</i> yang salah	Sesuai
2.	Kelola <i>Dataset</i>	OPD dapat melihat list <i>dataset</i>	Sesuai
		OPD dapat menambah data <i>dataset</i>	Sesuai
		OPD dapat mengedit <i>dataset</i>	Sesuai
		OPD dapat menghapus <i>dataset</i>	Sesuai
		OPD dapat melihat detail <i>dataset</i>	Sesuai
3.	Verifikasi <i>Dataset</i>	OPD dapat melihat catatan jika gagal dalam memverifikasi	Sesuai
4.	Publikasi <i>Dataset</i>	OPD dapat melihat <i>dataset</i> yang sudah di publikasikan	Sesuai

Dari hasil pengujian UAT yang dilakukan terhadap opd, ditemukan bahwa 4 dari 4 fungsionalitas yang diuji berhasil. Ini menunjukkan bahwa semua fungsionalitas aplikasi pada sisi opd beroperasi dengan sukses, mencapai tingkat keberhasilan 100%.

4.4. Evaluasi Hasil

Pada tahap evaluasi ini, penilaian dilakukan terhadap hasil pengujian sistem untuk mengukur sejauh mana sistem yang telah dikembangkan sesuai dengan spesifikasi dan kebutuhan yang telah ditentukan. Proses evaluasi mencakup analisis hasil uji fungsionalitas dan kinerja, serta identifikasi area yang membutuhkan perbaikan atau pengembangan lebih lanjut.

Hasil pengujian menunjukkan bahwa seluruh fungsionalitas sistem yang diuji berfungsi sesuai harapan. Tidak ada masalah yang ditemukan selama pengujian, menandakan bahwa sistem memiliki keandalan dan kualitas yang baik. Namun, sistem masih perlu diuji lebih lanjut untuk

menilai aspek keamanan dan kinerja guna memastikan sistem beroperasi secara optimal.

4.4.1 Evaluasi Hasil Uji

Setelah melakukan pengujian menggunakan metode *Black Box Testing* dan *User Acceptance Test (UAT)*, diperoleh hasil bahwa dalam *Black Box Testing*, *developer* melaksanakan 12 pengujian. Dari semua pengujian tersebut, seluruhnya berhasil tanpa ada satu pun yang gagal. Dengan hasil ini, dapat disimpulkan bahwa sistem telah berhasil menyelesaikan semua skenario pengujian dengan benar dan memenuhi harapan, sehingga dianggap berhasil dalam pengujian melalui metode *black box testing*.

Dalam pengujian *Acceptance Test (UAT)*, hasilnya menunjukkan bahwa dari 22 uji fungsional yang dilakukan, semuanya berhasil. Hasil tersebut menunjukkan bahwa setiap fungsi dalam aplikasi berjalan sesuai dengan spesifikasi yang ditetapkan dan memenuhi kebutuhan pengguna.

Berdasarkan kedua hasil pengujian tersebut, dapat disimpulkan bahwa sistem berhasil melewati uji fungsional menggunakan dua metode, yaitu *black box testing* dan *user acceptance test (UAT)*.

STT - NF

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian tugas akhir yang berjudul “Pengembangan Aplikasi Open Data Kota Bekasi Terintegrasi CKAN Menggunakan Web Framework CodeIgniter” dapat ditarik kesimpulannya sebagai berikut :

1. Rancangan aplikasi *Open Data* Kota Bekasi yang terintegrasi dengan *CKAN* menggunakan *web framework CodeIgniter* telah berhasil disusun. Rancangan ini mencakup berbagai aspek teknis seperti struktur *API*, antarmuka pengguna, dan integrasi dengan platform *CKAN* untuk mengelola *open data*. Dengan menggunakan *CodeIgniter*, pengembangan aplikasi menjadi lebih terstruktur dan efisien, memanfaatkan fitur-fitur bawaan *framework* tersebut untuk mendukung pengelolaan data yang lebih baik.
2. Efektivitas dari rancangan aplikasi ini telah dievaluasi melalui pengujian fungsional dan uji coba pengguna (*User Acceptance Test*). Hasil pengujian menunjukkan bahwa aplikasi mampu memenuhi spesifikasi fungsional yang ditetapkan dan kebutuhan pengguna, seperti kemudahan akses dan pengelolaan data. Integrasi dengan *CKAN* memungkinkan penyimpanan dan distribusi data yang lebih efisien, sehingga mempermudah pengguna dalam mengakses informasi yang dibutuhkan. Selain itu, penggunaan *web framework CodeIgniter* terbukti memberikan kinerja yang stabil dan responsif, yang mendukung efektivitas aplikasi dalam operasional sehari-hari.

5.2 Saran

Berdasarkan hasil penelitian mengenai pengembangan *frontend* untuk integrasi *API CKAN* pada aplikasi *Open Data* menggunakan *web framework CodeIgniter*, terdapat beberapa saran yang dapat dipertimbangkan sebagai panduan untuk penyempurnaan penelitian di masa mendatang. Penulis memberikan beberapa rekomendasi berikut untuk pengembangan lebih lanjut:

1. Pengembangan Lebih Lanjut:

Aplikasi Open Data Kota Bekasi yang terintegrasi dengan CKAN telah menunjukkan efektivitas yang baik. Namun, pengembangan lebih lanjut diperlukan untuk menambahkan fitur-fitur tambahan yang dapat meningkatkan user experience.

2. Peningkatan Keamanan

Untuk memastikan keamanan data yang dikelola, disarankan untuk memperkuat langkah-langkah keamanan, seperti implementasi enkripsi data dan perlindungan terhadap serangan siber. Ini penting untuk menjaga integritas dan kerahasiaan data.

3. Melakukan penelitian terkait dengan penggunaan *API CKAN* dalam pengembangan aplikasi *Open Data*.



STT - NF

DAFTAR PUSTAKA

- [1] A. Nurkholis, E. R. Susanto, and S. Wijaya, “Penerapan Extreme Programming dalam Pengembangan Sistem Informasi Manajemen Pelayanan Publik,” *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 5, no. 1, pp. 124–134, 2021.
- [2] I. S. dan P. K. B. Dinas Komunikasi, “Informasi Dashboard Open Data.” Accessed: Mar. 26, 2024. [Online]. Available: <https://danta.bekasikota.go.id/>
- [3] CKAN, “The world’s leading open source data management system.” Accessed: Mar. 26, 2024. [Online]. Available: <https://ckan.org/>
- [4] CKAN, “What is CKAN?” Accessed: Mar. 26, 2024. [Online]. Available: <https://docs.ckan.org/en/2.10/user-guide.html#what-is-ckan>
- [5] CodeIgniter, “CodeIgniter.” Accessed: Mar. 26, 2024. [Online]. Available: <https://www.codeigniter.com/>
- [6] CodeIgniter, “Welcome to CodeIgniter 4.” Accessed: Mar. 26, 2024. [Online]. Available: https://www.codeigniter.com/user_guide/intro/index.html
- [7] Monaria Aritonang, “Application Programming Interface.” Accessed: Mar. 27, 2024. [Online]. Available: https://www.academia.edu/112528658/APPLICATION_PROGRAMMING_INTERFACE
- [8] I. Kurniawan and F. Rozi, “REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android,” 2020. [Online]. Available: <http://jurnal-itsi.org>
- [9] Darma Jufri and Sagala Gaffar Hafiz, “Pengaruh Kualitas Sistem Informasi Akuntansi Terhadap Kualitas Informasi Akuntansi (Studi Emptiris di Indonesia),” *Jurnal Ilmiah MEA (Manajemen, Ekonomi, dan Akuntansi)*, vol. 4, no. 1, pp. 227–237, 2020.
- [10] Ciptadra Softindo, “About US.” Accessed: Mar. 26, 2024. [Online]. Available: <https://ciptadrasoft.com/Home/About>
- [11] B. A. Priyaungga, D. B. Aji, M. Syahroni, N. T. S. Aji, and A. Saifudin, “Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 3, no. 3, p. 150, Aug. 2020, doi: 10.32493/jtsi.v3i3.5343.
- [12] E. Suprpto, “User Acceptance Testing (UAT) Refreshment PBX Outlet Site BNI Kanwil Padang,” *Jurnal Civronlit Unbari*, vol. 6, no. 2, p. 54, Oct. 2021, doi: 10.33087/civronlit.v6i2.85.
- [13] M. Syarif and E. B. Pratama, “Analisis Metode Pengujian Perangkat Lunak Black

- Box Testing Dan Pemodelan Diagram UML Pada Aplikasi Veterinary Services Yang Dikembangkan Dengan Model Waterfall,” *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 5, no. 2, 2021.
- [14] L. Ariyanti, M. Najib, D. Satria, and D. Alita, “Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan,” *Jurnal Teknologi dan Sistem Informasi (JTSI)*, vol. 1, no. 1, pp. 90–96, 2020.
- [15] F. Dewangga, Y. Santoso, R. Rezeky, M. L. Toruan, D. A. Widarini, and E. K. Wiyati, “Program Jakarta Open Data Sebagai E-Government Dalam Pelayanan Informasi Publik Humas Pemprov DKI Jakarta,” *Jurnal Cyber PR*, vol. 2, no. 1, p. 94, 2022.
- [16] T. Bustomi, E. Marlovia, U. Pasundan Bandung, K. Bandung, and J. Barat, “Kebijakan: Jurnal Ilmu Administrasi Inisiatif Open Data Pada Usaha Mikro Kecil Menengah Kota Bandung,” *Jurnal Ilmu Administrasi*, vol. 12, 2021, doi: <https://doi.org/10.23969/kebijakan.v12i2.3756>.
- [17] A. Mulyanto, Moh. H. Koniyo, and R. M. T. Yassin, “Pembangunan Portal Open Data untuk Mendukung Open Government dan Smart City (Studi Kasus: Pemerintah Daerah Kota Gorontalo),” *Jurnal Teknik*, vol. 20, no. 1, pp. 31–43, Jul. 2022, doi: 10.37031/jt.v20i1.239.
- [18] M. W. Magister, A. Pendidikan, U. Kristen, and S. Wacana, “Pendekatan Penelitian Pendidikan: Metode Penelitian Kualitatif, Metode Penelitian Kuantitatif dan Metode Penelitian Kombinasi (Mixed Method),” *Jurnal Pendidikan Tambusai*, vol. 7, pp. 2896–2910, Apr. 2023, doi: <https://doi.org/10.31004/jptam.v7i1.6187>.
- [19] S. H. Bariah, M. Irsad, and S. Putra, “Penerapan Metode Waterfall Pada Perancangan Sistem Informasi Pengolahan Data Nilai Siswa,” *Jurnal Petik*, vol. 6, no. 1, 2020.

LAMPIRAN

1. Potongan Kode Program Integrasi API CKAN

A. List Dataset

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $this->ckanApiUrl . '/dataset_list');
curl_setopt($ch, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Cookie: ',
    'Authorization: '
));
$output = curl_exec($ch);
```

Lampiran 1 Codingan Fungsi List Dataset

B. Tambah Dataset

```
$url = $this->ckanApiUrl . '/dataset_create';

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, 1);
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($data));
$result = curl_exec($curl);
curl_close($curl);

$resultDecoded = json_decode($result);
```

Lampiran 2 Codingan Fungsi Tambah Dataset

C. Edit Dataset

```
$url = $this->ckanApiUrl . '/dataset_update/' . $id;

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($data));
curl_setopt($curl, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');
curl_setopt($curl, CURLOPT_HTTPHEADER, array(
    'Authorization: ',
    'Content-Type:application/json'
));
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($curl);
```

Lampiran 3 Codingan Fungsi Edit Dataset

D. Detail Dataset

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $this->ckanApiUrl . '/dataset_show/' . $id);
curl_setopt($ch, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Authorization: '
));
```

Lampiran 4 Codingan Fungsi Detail Dataset

E. Hapus Dataset

```
$url = $this->ckanApiUrl . '/dataset_delete/' . $id;

$data = [
    "id" => $id
];

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($data));
curl_setopt($curl, CURLOPT_HTTPHEADER, array(
    'Authorization: ',
    'Content-Type: application/json'
));
```

Lampiran 5 Codingan Fungsi Hapus Dataset

F. Verifikasi Dataset

```
$url = $this->ckanApiUrl . '/dataset_update/' . $id;

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, 1);
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($data));
curl_setopt($curl, CURLOPT_HTTPHEADER, array(
    'Authorization: ',
    'Content-Type: application/json'
));
```

Lampiran 6 Codingan Fungsi Verifikasi Dataset

STT - NF