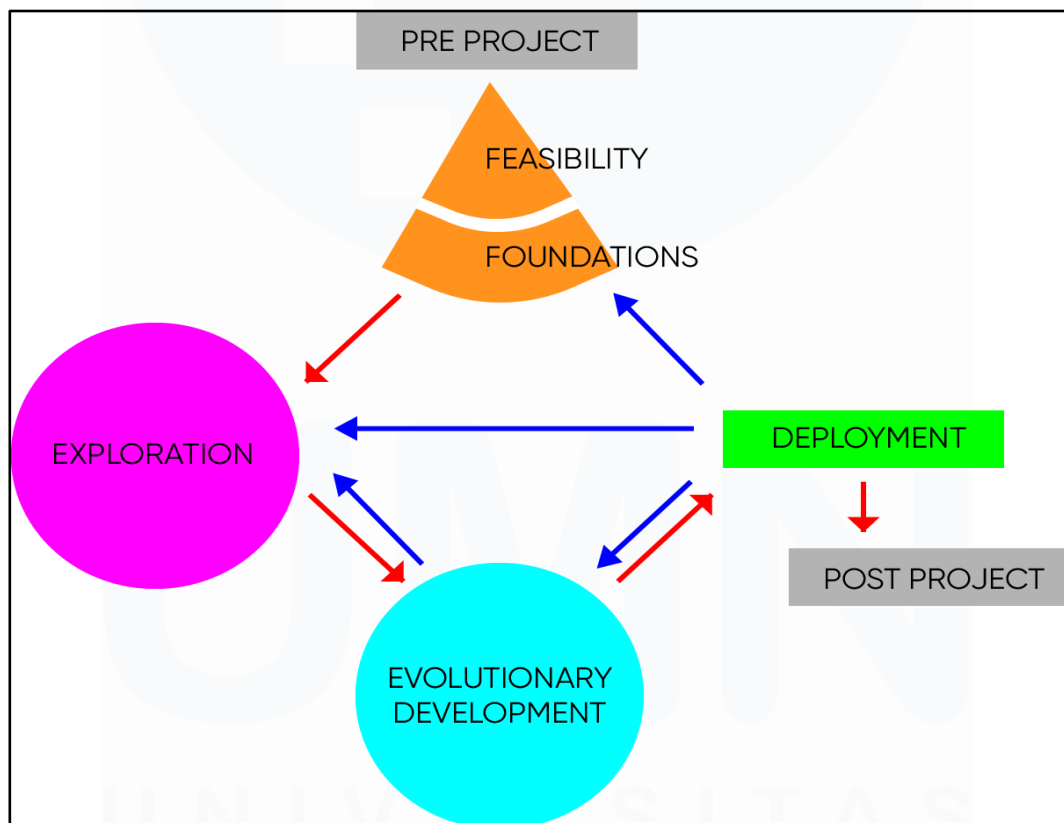


BAB III

METODE DAN HASIL PENELITIAN

3.1. Model Rancang Bangun

Pengembangan software dilaksanakan berdasarkan metodologi pengembangan software *Agile*. *Agile* adalah sekumpulan metode pengembangan software yang dilakukan secara bertahap dan berulang (iterasi) seperti pada Gambar 3.1. Metode *Agile* yang digunakan *Dynamic Systems Development Method (DSDM)* dimana metode ini berfokus pada keterlibatan seluruh anggota tim termasuk *stakeholder* [5].



Gambar 3.1 Diagram DSDM

DSDM yang digunakan memiliki 6 tahap diambil dari, yang terbagi menjadi 2 fase:

Fase Pra-Proyek (*Pre-Project*)

1. Studi Kelayakan (*Feasibility Study*): Pada tahap ini, dilakukan analisis untuk menentukan apakah proyek layak dilakukan atau tidak. Faktor-faktor yang dipertimbangkan adalah kelayakan bisnis, teknis, dan finansial.
2. Studi Bisnis (*Business Study*): Pada tahap ini, dilakukan analisis untuk memahami kebutuhan bisnis secara lebih mendalam. Hal ini dilakukan dengan melibatkan stakeholder, seperti pengguna akhir dan manajemen.

Fase Siklus Hidup Proyek (*Project Lifecycle*)

3. Model Fungsional (*Functional Model*): Pada tahap ini, dibuat model fungsional yang menggambarkan sistem yang akan dikembangkan. Model ini dibuat dengan melibatkan pengguna dan stakeholder.
4. Desain dan Pembuatan (*Design and Build*): Pada tahap ini, dilakukan desain dan pembuatan sistem secara bertahap dalam iterasi yang singkat. Setiap iterasi menghasilkan versi sistem yang dapat diuji dan di umpan balik oleh pengguna.
5. Penerapan (*Implementation*): Pada tahap ini, sistem diimplementasikan ke lingkungan produksi. Hal ini dilakukan dengan pelatihan pengguna dan dukungan teknis.
6. Pasca-Penerapan (*Post-Implementation*): Pada tahap ini, dilakukan pemantauan dan evaluasi sistem untuk memastikan bahwa sistem berjalan dengan baik dan memenuhi kebutuhan pengguna.

3.1.1 Feasibility Study

Pada tahap Feasibility Study, tim melakukan penilaian kelayakan proyek dan identifikasi tujuan pengembangan sistem kependudukan bersama dengan pihak GMLS. Beberapa hasil identifikasi yang dihasilkan berupa :

- a. Sistem akan digunakan untuk membangun fondasi basis data dari penduduk di Desa Panggarangan.
- b. Basis data akan dibentuk dengan bantuan formulir online yang menampilkan seluruh data yang dibutuhkan terkait kebencanaan.

- c. Basis data yang dikumpulkan akan digunakan untuk pengembangan sistem-sistem kebencanaan pada masa yang akan datang.

3.1.2 Business Study

Berdasarkan hasil brainstorming dengan pihak GMLS selaku mitra, jenis data yang dibutuhkan untuk melakukan pendataan persebaran penduduk adalah data personal terkait kependudukan. Data kependudukan mencakup NIK, Nomor KK, Nama, Jenis Kelamin, Tanggal Lahir, Golongan Darah, Status Hidup/Meninggal, Agama, Status Perkawinan, Status Hubungan Keluarga, Pendidikan Terakhir, Jenis Pekerjaan, Status Ekonomi, Status Disabilitas, RT, RW, Status Zona Rawan Bencana, Jalan>Nama Kampung.

Data-data kependudukan tersebut dikumpulkan dengan melalui input data oleh admin atau relawan pada halaman *form website* GMLS. Pelaksanaan input data penduduk hanya dapat dilakukan oleh admin atau relawan GMLS agar data yang diinput adalah data valid.

Manajemen pengolahan data penduduk yang didapat terdapat pada tabel data di panel admin GMLS. Tabel data penduduk berisikan data input yang diperoleh dan dapat dilakukan edit serta delete terhadap data yang tidak relevan. Pengolahan data secara visual diolah dalam bentuk chart berdasarkan kategori masing-masing. Pengolahan data secara visual berfungsi untuk melakukan pemetaan aksi tanggap darurat bencana pasca bencana.

3.1.3 Design and Build Iteration

Pembuatan panel admin GMLS yang mencakup *form, dashboard*, dan tabel data pada website GMLS dibangun diatas library JavaScript, yaitu React dan menggunakan framework Next.js. Penggunaan Next.js sendiri didasari pada keberadaan website GMLS terlebih dahulu sehingga penambahan fitur akan mengikuti struktur yang sudah ada.

Penyimpanan data penduduk tersimpan dalam firebase realtime *database*. Penggunaan firebase didasari atas kemudahan penggunaan library dengan disediakannya *API* untuk menyimpan data di cloud server. Firebase realtime *database* melakukan sinkronisasi data dalam waktu milidetik setiap ada data yang berubah [6]. Secara langsung firebase akan melakukan sinkronisasi terhadap web GMLS.

3.1.4 Implementation

Pada tahap *Implementation*, dilakukan pengembangan fitur berupa *form page*, *admin page*, dan *dashboard page*.

Implementasi awal adalah dengan merancang *form page*. Formulir dirancang agar mudah digunakan dan dimengerti oleh penduduk desa, dengan tujuan meminimalisir kesalahan input data. Fitur validasi data juga diterapkan untuk memastikan keakuratan dan kelengkapan data yang dikumpulkan.

Dalam *admin page* meliputi kemampuan untuk mengubah data yang sudah tersedia, melihat data dalam bentuk tabel, dan menghapus data. Halaman admin didesain dengan sederhana dan responsif agar mudah digunakan dan intuitif.

Dashboard page berfungsi untuk memvisualisasi data - data kependudukan. Model diagram batang horizontal digunakan untuk menampilkan data secara visual dan informatif. Pengguna dapat memfilter data berdasarkan berbagai kriteria. Analisis data dan pengambilan keputusan difasilitasi melalui dashboard ini.

3.1.5 Post-Project

Seluruh fitur *database*, formulir, halaman admin, dan *dashboard* menjalani pengujian menyeluruh untuk memastikan fungsionalitas dan keamanan. Pengujian yang dilakukan antara lain:

- a. Pengujian Struktural

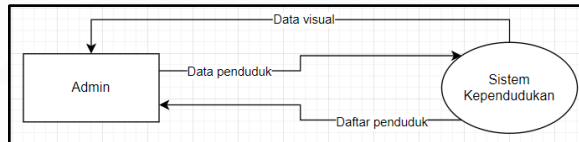
1. Skema: Pengujian ini dilakukan untuk membuat struktur *database* yang pasti dan sesuai dengan desain yang client harapkan.
 2. Tabel: Pengujian yang dilakukan pada bagian kolom, tipe data, serta indeks yang digunakan pada *database*.
 3. Prosedur Tersimpan: Tahap pengujian yang didasarkan pada prosedural sistem *database*
- b. Pengujian Fungsional
1. Integritas dan Konsistensi Data: Pengujian ini dilakukan untuk memeriksa bahwa data yang ada, merupakan data yang valid dan konsisten.
 2. Login dan Keamanan Pengguna: Pengujian yang dilakukan pada sektor akses pengguna dan hak akses.
- c. Pengujian Validasi Server Basis Data:
1. Pengujian yang dilakukan sebagai langkah akhir untuk memastikan server *database* bekerja dengan baik.

Penggunaan *vercel* untuk *hosting* juga tidak luput dari kelebihan *vercel* dalam jajarannya [7], antara lain:

- a. Performa *vercel* yang cepat: Penggunaan jaringan *CDN (Content Delivery Network)* sebagai jembatan untuk pengiriman konten statis dengan lebih cepat [8].
- b. *Preview Deployment*: *Vercel* memungkinkan penggunaanya untuk me-review situs web melalui preview.
- c. *Zero-Config Infrastructure*: Infrastruktur yang dihadirkan oleh *vercel* bersifat siap pakai, tanpa melakukan konfigurasi tambahan [9].

3.2 Hasil Perancangan *Data Flow Diagram*

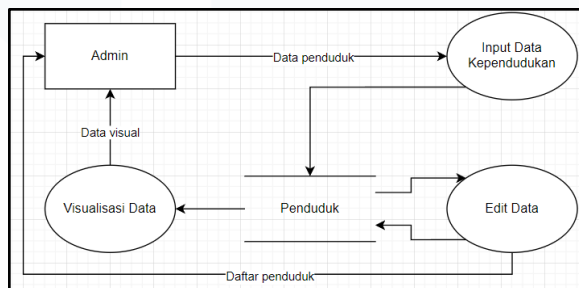
Gambar 3.2 menunjukkan DFD Level 0 gambaran aliran data antara admin, sistem aplikasi kependudukan, dan database.



Gambar 3.2 DFD Level 0

- a. Admin mengirimkan data penduduk ke Sistem Kependudukan.
- b. Sistem Kependudukan mengirimkan kembali daftar penduduk kepada admin.
- c. Sistem Kependudukan mengirimkan data visual kepada admin.

Gambar 3.3 menunjukkan DFD Level 1 yang berisi rincian proses di dalam sistem kependudukan.



Gambar 3.3 DFD Level 1

- a. Proses 1 : *Input* data kependudukan, proses mengumpulkan dan menambah data kependudukan ke dalam sistem.
- b. Proses 2 : *Edit* data, proses melakukan *CRUD* (*Create, Read, Update, Delete*) terhadap data penduduk.

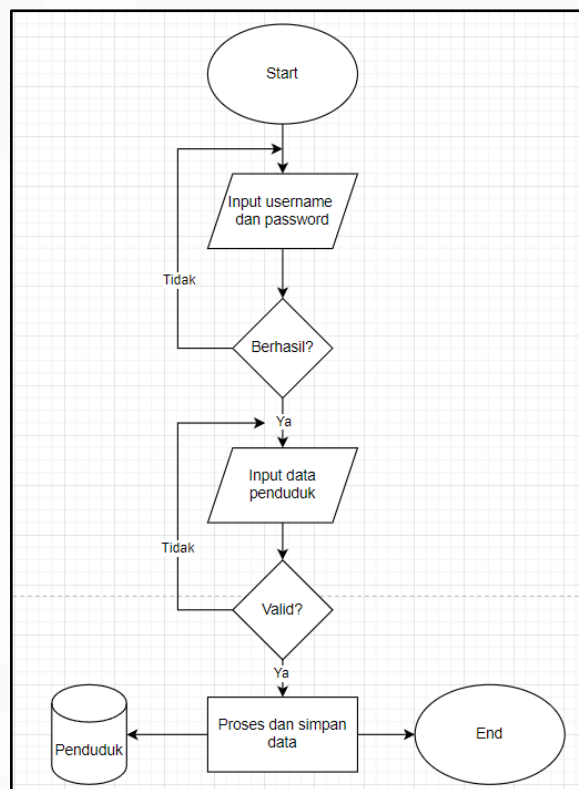
Proses 3 : Visualisasi data, proses menghasilkan bentuk visual dari data grafis yang dibuat untuk menyampaikan informasi dari hasil pengambilan data penduduk.

3.3 Hasil Perancangan Flowchart

Berdasarkan DFD yang sudah dibuat, maka Flowchart dibuat supaya menyesuaikan untuk setiap proses.

3.3.1 Flowchart Proses Pengisian Data

Gambar 3.4 menunjukkan Flowchart untuk proses pengisian data. Proses pengisian data dimulai dengan melakukan proses *login* terlebih dahulu pada panel admin GMLS. Admin akan melakukan input *username* dan *password*, jika berhasil akan ke tampilan *form* untuk melakukan input data penduduk, jika tidak maka akan kembali ke tampilan *login*. Selanjutnya, admin akan melakukan proses *input* data penduduk, apabila valid, data akan tersimpan di *database* Penduduk dan proses *input* data akan selesai. Ketika data tidak valid maka data tidak akan bisa di input ke dalam *database*.

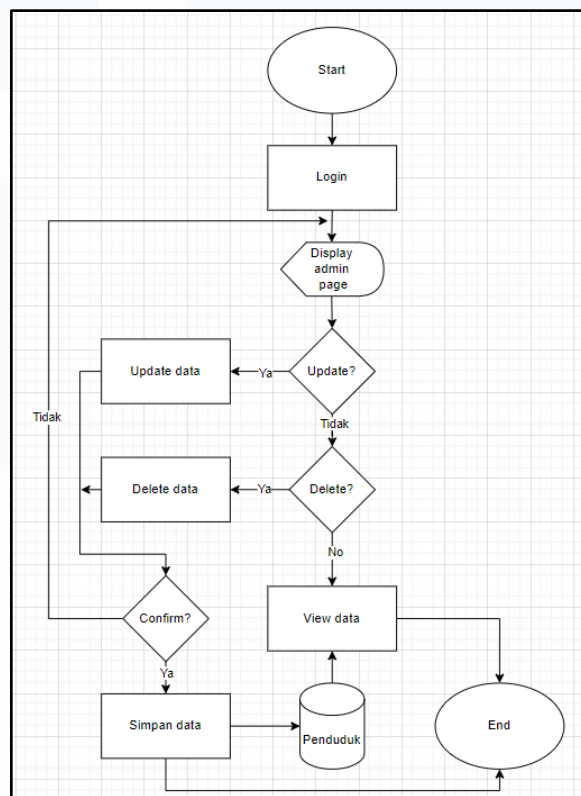


Gambar 3.4 Flowchart Proses Pengisian Data

3.3.2 Flowchart Proses Pengelolaan Data

Gambar 3.5 menunjukkan Flowchart untuk proses pengelolaan data. Proses pengelolaan data dimulai dengan proses *login* terlebih dahulu.

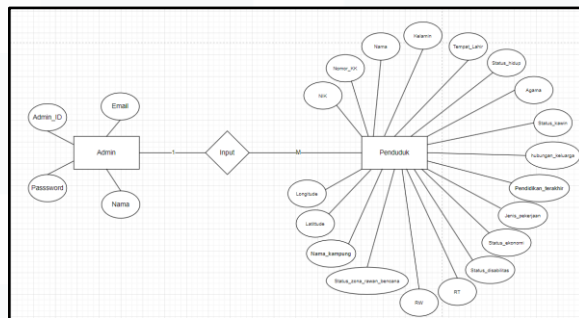
Setelah melakukan proses *login*, *user* akan beralih ke tampilan admin *page* yang berisikan daftar data penduduk dan menu untuk melakukan *update* dan *delete*. Jika *user* ingin melakukan *update* dan *delete* data, maka akan dilakukan validasi jika sudah sesuai data akan disimpan ke *database* penduduk dan proses *update* serta *delete* data akan selesai. *User* yang tidak melakukan *update* dan *delete* hanya akan melihat daftar data penduduk.



Gambar 3.5 *Flowchart* Proses Pengolahan Data

3.4 Hasil Perancangan Entity Relationship Diagram (ERD)

Gambar 3.6 Berikut menunjukkan entitas yang digunakan untuk mengelola *database* kependudukan di Desa Panggarangan.



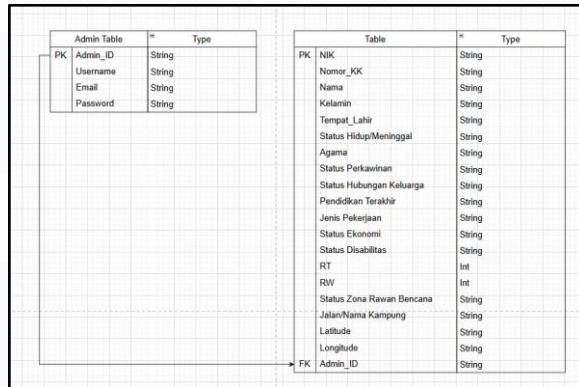
Gambar 3.6 Entity Relationship Diagram (ERD)

ERD yang digambarkan hanya terdiri dari dua buah entitas, *admin* dan penduduk dikarenakan *Firebase* adalah basis data non-relasional. Meskipun *Firebase* adalah *database* non-relasional, ERD ini tetap membantu dalam perancangan logis dari struktur data yang akan disimpan di dalam dokumen *Firebase*. Setiap entitas dalam diagram ini bisa diwakili sebagai koleksi dalam *Firebase*, dengan dokumen yang menyimpan data atribut terkait.



3.5 Struktur Data Tabel

Struktur data tabel yang digunakan melibatkan dua buah entitas, *admin* dan penduduk, seperti yang terlihat pada Gambar 3.7. Dengan peran *admin* sebagai pengguna aplikasi dan penduduk sebagai data yang tersimpan dalam basis data.



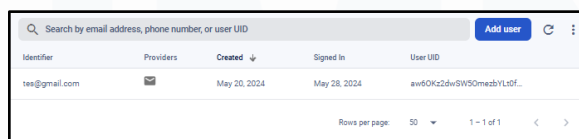
Gambar 3.7 Struktur Tabel Data

3.5.1

Pembahasan Hasil Perancangan Struktur

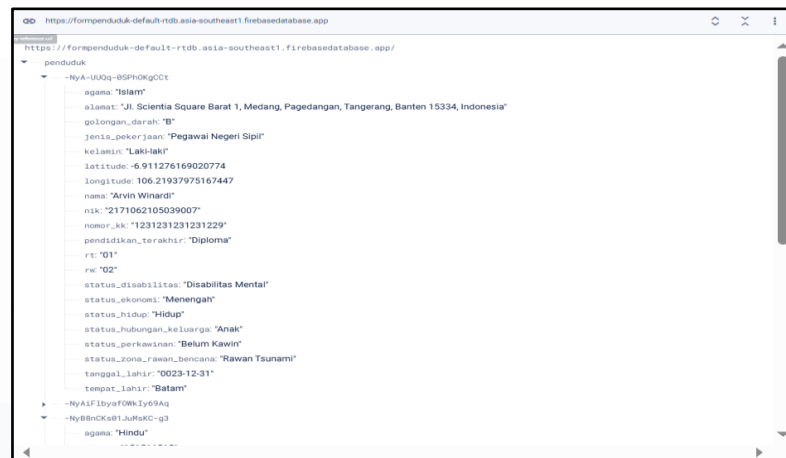
Data Tabel

Gambar 3.8 menunjukkan Tabel untuk Tabel Admin. Data *admin* terbentuk dari fitur autentikasi pengguna yang disediakan *Firebase*. Data yang terbentuk terdiri dari *username*, *email*, *user_id*, dan *password* yang terenkripsi secara otomatis.



Gambar 3.8 Implementasi Data Tabel Admin

Gambar 3.9 menunjukkan tampilan *firebase* yang menjabarkan setiap input user dan dikelompokkan ke dalam struktur data yang sudah dibuat agar ditampilkan dengan lebih jelas.



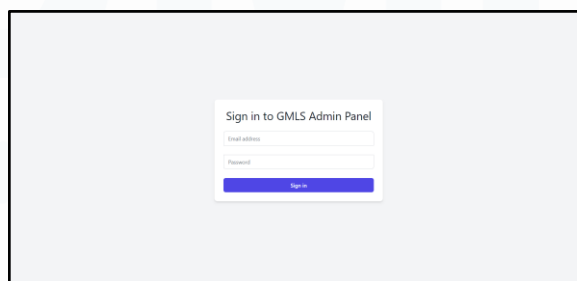
Gambar 3.9 Implementasi Data Tabel Penduduk

3.6 Pembahasan Tampilan Aplikasi

Setelah melalui tahapan perancangan dan pengembangan, aplikasi kependudukan Desa Panggarangan dalam rangka mendukung program mitigasi bencana oleh Gugus Mitigasi Lebak Selatan telah selesai dibangun. Bagian ini akan membahas hasil dari pengembangan aplikasi tersebut serta analisis dari implementasi berbagai fitur yang telah dirancang.

3.6.1 Tampilan *Login* untuk *Admin*

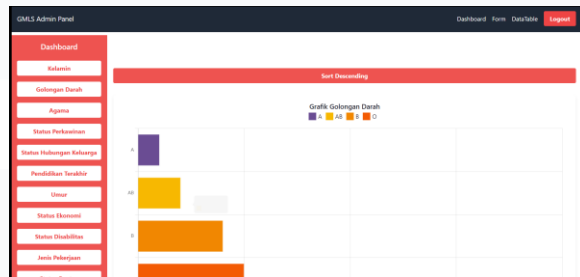
Tampilan *login* untuk admin merupakan halaman pertama yang diakses oleh pengguna yang memiliki hak akses khusus. Tampilan Gambar 3.10 merupakan halaman dirancang untuk memberikan keamanan dengan meminta pengguna memasukkan *username* dan *password* yang *valid* sebelum dapat mengakses fitur-fitur admin lainnya.



Gambar 3.10 Admin page

3.6.2 Tampilan Grafik pada *Dashboard* GMLS

Menu *dashboard* pada Gambar 3.11 menampilkan persebaran data penduduk Desa Panggarangan. *Dashboard* GMLS menampilkan beberapa fitur data yang meliputi persebaran data kelamin, golongan darah, agama, status perkawinan, status hubungan keluarga pendidikan terakhir, umur, status ekonomi, status disabilitas, jenis pekerjaan, status rentan.



Gambar 3.11 Halaman Dashboard

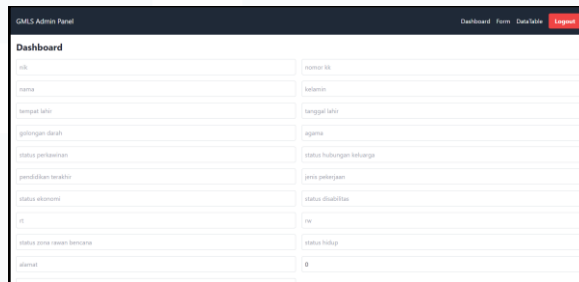
3.6.3 Tampilan *Form Input* Data Penduduk

Tampilan *form* input pada Gambar 3.12 di panel admin GMLS merupakan halaman yang digunakan untuk melakukan input data penduduk Desa Panggarangan yang dilakukan oleh *admin* GMLS.

Gambar 3.12 Form data penduduk

3.6.4 Tampilan *Form Editing Data Penduduk*

Tampilan pada Gambar 3.13 memungkinkan admin untuk mengedit data penduduk yang telah diinput. Halaman editing ini dirancang agar mudah digunakan, dengan tabel yang menampilkan semua data penduduk dan tombol edit yang memungkinkan perubahan data secara langsung.



Gambar 3.13 *Form editing data table*

3.6.5 Tampilan Tabel Data

Halaman *data table* pada Gambar 3.14 merupakan kumpulan informasi data penduduk yang didapat dari *input* dari halaman *form*.



nik	nomor kk	nama	kelamin	tempat lahir	tanggal lahir	golongan darah	agama	status pernikahan	status hubungan keluarga	pendidikan terakhir	jenis pekerjaan
18041818181818	1111111111111111	siapa	Laki-laki	Sarang	2024-01-08	B	Kristen	Belum Kawin	Kepala Keluarga	Dokter	Netelan
99999999999999	99999999999999	Dayak	Laki-laki	Cilegon	2024-02-14	B	Islam	Kawin	Kepala Keluarga	SD	Netelan
1234567891234567	1234567891234567	Eduan	Laki-laki	Jakarta	2023-08-10	O	Katolik	Belum Kawin	Anak	SMA/SMK	Mahasiswa
2171062705039007	1231231231231239	Avin Winardi	Laki-laki	Batam	1923-12-31	B	Islam	Belum Kawin	Anak	Diploma	Pegawai Negeri Sipil

Gambar 3.14 *Data table*

Gambar 3.15 merupakan tampilan fitur edit data pada halaman *data table*. Kesalahan input data penghapusan data penduduk dilakukan pada halaman ini.

status dengan energi	pendidikan terakhir	jenis pekerjaan	status ekonomi	status disabilitas	rt	rw	status area rawan bencana	status hidup	alamat	latitude	longitude	Actions
ok	Diploma	Pejabat Negeri Sipil	Mesengih	Disabilitas Mental	01	02	Rawan Tsunami	Hidup	R. Satrio Satrio Barat 1, Madang, Papadangan, Pangemang, Bantem, 15334, Indonesia	-89132419820374	106,219279167467	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
o	SD	Pelajar / Mahasiswa	Kaya	Disabilitas Fisik	10	1	Tidak Rawan Bencana	Hidup	www	-4.1898762	106,6883312	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
o	SD	Karyawan BUMN	Mesengih	Disabilitas Fisik	30	30	Tidak Rawan Bencana	Hidup	151511515	-89133855	106,2191969	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
ok	Sarjana	Pelajar / Mahasiswa	Mesengih	Disabilitas intelektual	30	30	Tidak Rawan Bencana	Hidup	ooo	-4.2226432	106,5844736	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
data lengkap	SMP	Pelajar / Mahasiswa	Kisikin	Tidak Ada	1	1	Rawan Gempa	Meninggal	ooooo	-4.2226432	106,5844736	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
o	Diploma	Tertara Nasional Indonesia	Mesengih	Disabilitas Sensorik	1	1	Rawan Longsor	Hidup	www	-4.1895768	106,688793	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Gambar 3.15 Data table

3.7 Testing

Metode testing yang digunakan pada aplikasi kependudukan Desa Panggarangan adalah *greybox testing*. *Greybox Testing* adalah metode dimana *user* memiliki akses pengetahuan terbatas terhadap komponen yang diuji. Tahap uji aplikasi dilakukan secara berkala atau *regression test* pada komponen yang sudah siap uji.

Tabel 3.1 Hasil Testing

Greybox Testing				
No.	Forms	Step	Expected Result	Output
1.	Login	<ul style="list-style-type: none"> - Akses link https://www.gmls.org/login - Masukkan <i>username</i> dan <i>password</i> - Klik <i>Login</i> 	<i>Admin</i> berhasil ke menu panel <i>admin</i> GMLS	Berhasil <i>login</i> ke panel <i>admin</i> GMLS
2.	Input Data	<ul style="list-style-type: none"> - Akses link https://www.gmls.org/login - Login - Akses menu 	<i>Admin</i> berhasil melakukan <i>input data</i> penduduk	Berhasil melakukan <i>input data</i> penduduk

		<p>“Form”</p> <ul style="list-style-type: none"> - Masukkan data penduduk sesuai <i>field form</i> yang diminta - Klik <i>Submit</i> 		
3.	Show Data	<ul style="list-style-type: none"> - Akses link https://www.gmls.org/login - Login - Akses menu “<i>DataTable</i>” 	<i>Admin</i> dapat melihat data penduduk yang sudah di- <i>input</i>	Berhasil melihat data penduduk yang sudah di- <i>input</i>
4.	Edit Data	<ul style="list-style-type: none"> - Akses link https://www.gmls.org/login - Login - Akses menu “<i>DataTable</i>” - Klik “<i>Edit</i>” pada data <i>user</i> - Isi <i>field form</i> pada data yang ingin diubah - Klik “<i>Update Record</i>” 	<i>Admin</i> dapat mengubah data penduduk	Muncul konfirmasi dan berhasil melakukan <i>edit data</i>
5.	Delete Data	<ul style="list-style-type: none"> - Akses link https://www.gmls.org/login - Login - Akses menu “<i>DataTable</i>” 	<i>Admin</i> dapat menghapus data penduduk	Muncul konfirmasi dan berhasil melakukan <i>delete data</i>

		<ul style="list-style-type: none"> - Klik "Delete" pada data <i>user</i> 		
6.	Dashboard	<ul style="list-style-type: none"> - Akses link https://www.gmls.org/login - Login - Akses menu "Dashboard" 	<i>Admin</i> dapat melihat visualisasi data penduduk yang terkumpul	Berhasil melihat visualisasi data penduduk dalam bentuk <i>chart</i>

