

**BACKEND SISTEM MANAJEMEN RELAWAN BENCANA BERBASIS  
WEB MENGGUNAKAN *FRAMEWORK* CODEIGNITER 4**

**(Skripsi)**

**Oleh**

**M. FADHIL AZHARI  
NPM 1817051062**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

**BACKEND SISTEM MANAJEMEN RELAWAN BENCANA BERBASIS  
WEB MENGGUNAKAN *FRAMEWORK* CODEIGNITER 4**

**Oleh**

**M. FADHIL AZHARI**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA KOMPUTER**

**Pada**

**Jurusan Ilmu Komputer**

**Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2024**

## ABSTRAK

### BACKEND SISTEM MANAJEMEN RELAWAN BENCANA BERBASIS WEB MENGGUNAKAN FRAMEWORK CODEIGNITER 4

Oleh

**M. FADHIL AZHARI**

Manajemen relawan merupakan faktor penting dalam memperkuat misi organisasi kemanusiaan, terutama dalam konteks penanggulangan bencana. Badan Penanggulangan Bencana Daerah (BPBD) berupaya memperbaharui sistem yang ada untuk meningkatkan efisiensi respons terhadap bencana. Namun, pegawai di bidang kedaruratan masih mengelola data relawan secara konvensional dan terpisah, yang dapat menghambat efektivitas operasional. Penelitian ini bertujuan untuk mengimplementasikan sistem *backend* yang dapat mengelola data relawan secara terpadu dengan fitur modul admin, sehingga admin dapat mengatur data sesuai dengan bencana yang sedang terjadi. Sistem ini dikembangkan menggunakan metode *Rapid Application Development* (RAD) dengan *framework* CodeIgniter 4, menggunakan bahasa pemrograman PHP dan basis data MySQL. Selain itu, sistem ini telah diuji menggunakan metode *black-box testing* untuk memastikan fungsionalitasnya. Hasil dari penelitian ini diharapkan dapat meningkatkan manajemen relawan dan mendukung BPBD dalam mempercepat penanggulangan bencana.

Kata kunci: Manajemen Relawan, BPBD, Pegawai Bidang Kedaruratan, CodeIgniter 4, *Rapid Application Development* (RAD).

## **ABSTRACT**

### **BACKEND OF WEB-BASED DISASTER VOLUNTEER MANAGEMENT SYSTEM USING CODEIGNITER 4 FRAMEWORK**

**By**

**M. FADHIL AZHARI**

Volunteer management is a crucial factor in strengthening the mission of humanitarian organizations, especially in the context of disaster response. The Regional Disaster Management Agency (BPBD) is striving to renew its existing systems to enhance disaster response efficiency. However, staff in the emergency division still manage volunteer data conventionally and separately, which can hinder operational effectiveness. This research aims to implement a backend system that can manage volunteer data in an integrated manner with an admin module, allowing administrators to organize data according to the ongoing disaster. The system was developed using the Rapid Application Development (RAD) method with the CodeIgniter 4 framework, utilizing PHP programming language and MySQL database. Additionally, the system has been tested using black-box testing methods to ensure its functionality. The results of this study are expected to improve volunteer management and support BPBD in accelerating disaster response.

**Keywords:** Volunteer Management, BPBD, Emergency Staff, CodeIgniter 4, Rapid Application Development (RAD).

Judul Skripsi : **BACKEND SISTEM MANAJEMEN RELAWAN  
BENCANA BERBASIS WEB MENGGUNAKAN  
FRAMEWORK CODEIGNITER 4**

Nama Mahasiswa : M. Fadhil Azhari

NPM : 1817051062

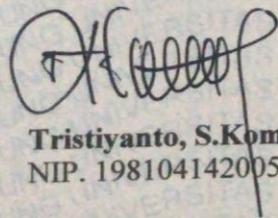
Program Studi : S1 Ilmu Komputer

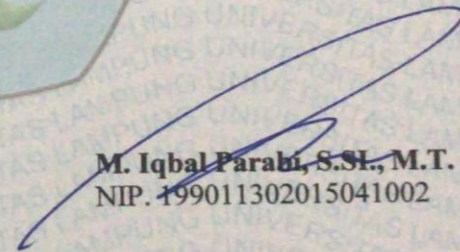
Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

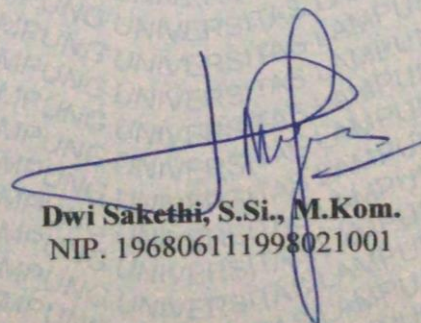


1. **Komisi Pembimbing**

  
**Tristiyanto, S.Kom., M.I.S., Ph.D.**  
NIP. 198104142005022002

  
**M. Iqbal Parahi, S.SI., M.T.**  
NIP. 199011302015041002

2. **Ketua Jurusan Ilmu Komputer**

  
**Dwi Sakethi, S.Si., M.Kom.**  
NIP. 196806111998021001

**MENGESAHKAN**

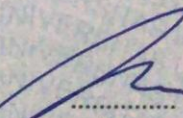
1. Tim Penguji

Ketua Penguji : **Tristiyanto, S.Kom., M.I.S., Ph.D.**



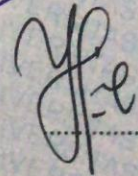
Penguji I : **M. Iqbal Parabi, S.SI., M.T.**

Sekretaris

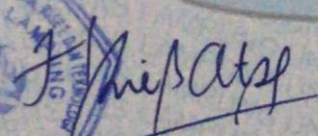


Penguji II : **Anie Rose Irawati, M.Cs.**

Anggota



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Dr. Eng. Heri Satria, S.Si., M.Si.**  
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: **08 Agustus 2024**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama: M. Fadhil Azhari

NPM: 1817051062

Dengan ini menyatakan bahwa skripsi saya yang berjudul "*BackEnd Sistem Manajemen Relawan Bencana Berbasis Web Menggunakan Framework CodeIgniter 4*" merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 2024



M. Fadhil Azhari

NPM. 1817051062

## RIWAYAT HIDUP



Lahir di Kembang Tanjung, pada tanggal 31 Mei 2000, sebagai anak kedua dari bapak Azhari Helmi (Alm) dan ibu Suharmi, S.Pkp. Menyelesaikan pendidikan dasar di SD Islam Ibnurusyd pada tahun 2012. Pendidikan menengah pertama di SMP IT Insan Rabbani Kotabumi pada tahun 2015, lalu melanjutkan ke jenjang Pendidikan menengah atas di SMAN 2 Kotabumi yang diselesaikan pada tahun 2018.

Pada tahun 2019, terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa melakukan beberapa kegiatan antara lain:

1. Menjadi Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2019/2020.
2. Menjadi anggota Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2018/2019.
3. Menjadi anggota Internal Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2019/2020.



## **MOTO**

“ Dari terbentur, terbentur, terbentur, kemudian terbentuk.”  
(Tan Malaka)

## **PERSEMBAHAN**

Puji dan syukur kehadirat Allah SWT karena atas berkat dan rahmat serta karuniaNya saya dapat menyelesaikan skripsi ini dengan baik.

Shalawat serta salam tak lupa saya sanjung agungkan kepada Nabi Muhammad SAW yang telah membawa umat manusia menuju zaman yang terang benderang ini dan syafaatnya senantiasa dinantikan di yaumul akhir kelak.

Kupersembahkan karya ini kepada:

### **Kedua Orang Tuaku Tersayang**

Yang tiada henti selalu memberi dukungan dan mengiringi langkah saya dengan doa. Saya ucapkan terima kasih banyak atas segala kasih sayang, didikan, dan perjuangan yang telah diberikan kepada saya hingga sekarang yang telah membawa langkah saya menuju seperti sekarang.

### **Seluruh Keluarga Besar Ilmu Komputer 2018**

**Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Lampung**

## SANWACANA

Puji syukur kehadiran Allah SWT atas berkat rahmat dan hidayah-Nya, saya dapat menyelesaikan skripsi dengan judul “Pengembangan Modul Dosen Sistem Manajemen Mutu Perkuliahan di Jurusan Ilmu Komputer”. Dalam pelaksanaannya, saya mendapat banyak bimbingan dan dukungan dari berbagai pihak, sehingga pada sanwacana ini saya ingin mengungkapkan ucapan terima kasih saya kepada:

1. Kedua orang tua tercinta yaitu Bapak Azhari Helmi (Alm) dan Ibu Suharmi serta kakak dan adik tersayang Rahma Khoirunnisa dan Aulia Faradilla Azhari yang selalu memberi dukungan, semangat, motivasi, kasih sayang dan doa tiada henti.
2. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.
3. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.
4. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Dosen Pembimbing 1 yang telah memberi bimbingan kepada saya dalam menyelesaikan skripsi dengan baik.
5. Bapak M. Iqbal Parabi, S.Si., M.T. selaku Dosen Pembimbing 2 yang telah memberi bimbingan kepada saya dalam menyelesaikan skripsi dengan baik..
6. Ibu Anie Rose Irawati, M.Cs. selaku Dosen Pembahas yang telah memberi banyak masukan dalam menyelesaikan skripsi ini.
7. Ibu Dewi Asiah Shofiana, S.Kom., M.Kom. selaku Dosen Pembimbing Akademik.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang membantu saya dalam perkuliahan.

9. Teman-teman S1 Ilmu Komputer angkatan 2018 yang telah memberikan pengalaman yang sangat menyenangkan bersama-sama.
10. Krisna Benedicta selaku rekan skripsi yang saling bantu-membantu dalam proses pengerjaan skripsi ini.
11. Achmad Romadoni selaku pemberi bantuan dalam perkuliahan dan pengerjaan skripsi ini.

Bandar Lampung, 2024

**M. Fadhil Azhari**

NPM. 1817051062

## DAFTAR ISI

Halaman

<b>DAFTAR GAMBAR.....</b>	<b>iii</b>
<b>DAFTAR TABEL .....</b>	<b>vi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar belakang .....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah .....	4
1.4. Tujuan Penelitian .....	4
1.5. Manfaat Penelitian .....	5
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>6</b>
2.1. Penelitian Terdahulu.....	6
2.2. <i>Backend</i> .....	7
2.3. API.....	8
2.4. <i>JSON Web Token (JWT)</i> .....	9
2.5. Aplikasi Berbasis Web.....	10
2.5.1. PHP .....	11
2.5.2. XAMPP .....	11
2.5.3. MySQL .....	12
2.5.4. <i>Web Browser</i> .....	12
2.5.5. Postman .....	12
2.5.6. CodeIgniter .....	12
2.6. <i>Database</i> .....	13
2.7. Metode Pengembangan Sistem Model RAD .....	14
2.8. Pengujian Perangkat Lunak .....	15

<b>BAB III METODOLOGI PENELITIAN.....</b>	<b>16</b>
3.1. Waktu dan Tempat Penelitian .....	16
3.2. Alat Penelitian.....	16
3.3. Metode Pengumpulan Data.....	17
3.3.1. Data Primer .....	17
3.3.2. Data Sekunder .....	17
3.4. Tahapan Penelitian.....	17
3.4.1. Identifikasi Masalah .....	17
3.4.2. Studi Literatur.....	18
3.4.3. Penerapan metode RAD.....	18
3.4.4. Rencana Kebutuhan.....	18
3.4.5. Pengujian Sistem.....	21
3.4.6. Penulisan Laporan .....	24
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>26</b>
4.1. Hasil .....	26
4.2. Pembahasan .....	26
4.2.1. Penerapan Metode RAD .....	27
4.2.1.1. Perencanaan Kebutuhan.....	27
4.2.1.2. Desain Sistem .....	30
4.2.1.3. Proses Pengembangan dan Pengumpulan <i>feedback</i> .....	30
4.2.1.4. Desain Sistem Kedua .....	58
4.2.1.5. Proses Pengembangan dan Pengumpulan <i>feedback</i> Kedua .....	60
4.2.1.6. Implementasi.....	68
4.3. Pengujian Sistem.....	72
<b>BAB V SIMPULAN DAN SARAN .....</b>	<b>83</b>
5.1. Simpulan .....	83
5.2. Saran .....	83
<b>DAFTAR PUSTAKA .....</b>	<b>85</b>

## DAFTAR GAMBAR

Gambar	Halaman
1. Cara Kerja API .....	9
2. <i>Json Web Token Verification</i> .....	10
3. Tahapan Penelitian .....	17
4. <i>Use Case Diagram</i> Sistem Manajemen Relawan Bencana.....	19
5. ERD Sistem Manajemen Relawan Bencana .....	20
6. Hasil Perencanaan Kebutuhan. ....	28
7. <i>Use Case Diagram</i> Pertama.....	28
8. <i>Entity Relationship Diagram</i> Pertama. ....	29
9. Kode <i>EndPoint</i> Autentikasi. ....	32
10. Kembalian data <i>EndPoint</i> Autentikasi.....	33
11. Potongan Kode <i>EndPoint</i> Get Daftar Kegiatan. ....	34
12. Kembalian data <i>EndPoint</i> Get Daftar Kegiatan.....	34
13. Potongan kode <i>EndPoint</i> Tambah Daftar Kegiatan.....	35
14. Kembalian data <i>EndPoint</i> Tambah Daftar Kegiatan.....	35
15. Potongan Kode <i>EndPoint</i> Edit Daftar Kegiatan. ....	36
16. Kembalian data <i>EndPoint</i> Edit Daftar Kegiatan.....	36
17. Potongan Kode <i>EndPoint</i> Hapus Daftar Kegiatan.....	37
18. Kembalian data <i>EndPoint</i> Hapus Daftar Kegiatan. ....	37
19. Potongan Kode <i>EndPoint</i> Get Laporan Harian.....	38
20. Kembalian data <i>EndPoint</i> Get Laporan Harian. ....	38
21. Potongan kode <i>EndPoint</i> Tambah Laporan Harian. ....	39
22. Kembalian data <i>EndPoint</i> Tambah Laporan Harian.....	39
23. Potongan Kode <i>EndPoint</i> Edit Daftar Kegiatan. ....	40
24. Kembalian data <i>EndPoint</i> Edit Laporan Harian. ....	40
25. Potongan Kode <i>EndPoint</i> Hapus Daftar Kegiatan.....	41
26. Kembalian data <i>EndPoint</i> Hapus Laporan Harian.....	41
27. Potongan Kode <i>EndPoint</i> Get Target.....	42

28. Kembalian data <i>EndPoint</i> Get Target. ....	42
29. Potongan Kode <i>EndPoint</i> Tambah Target. ....	43
30. Kembalian data <i>EndPoint</i> Tambah Target. ....	43
31. Potongan Kode <i>EndPoint</i> Edit Target. ....	44
32. Kembalian data <i>EndPoint</i> Edit Target. ....	44
33. Potongan Kode <i>EndPoint</i> Hapus Target. ....	45
34. Kembalian data <i>EndPoint</i> Hapus Target. ....	45
35. Potongan Kode <i>EndPoint</i> Get Pemasukan. ....	46
36. Kembalian data <i>EndPoint</i> Get Pemasukan. ....	46
37. Potongan kode <i>EndPoint</i> Tambah Pemasukan. ....	47
38. Kembalian data <i>EndPoint</i> Tambah Pemasukan. ....	47
39. Potongan Kode <i>EndPoint</i> Edit Pemasukan. ....	48
40. Kembalian data <i>EndPoint</i> Edit Pemasukan. ....	48
41. Potongan Kode <i>EndPoint</i> Hapus Pemasukan. ....	49
42. <i>EndPoint</i> Edit Pemasukan. ....	49
43. Potongan Kode <i>EndPoint</i> Get Pengeluaran. ....	50
44. Kembalian data <i>EndPoint</i> Get Pengeluaran. ....	50
45. Potongan kode <i>EndPoint</i> Tambah Pengeluaran. ....	51
46. Kembalian data <i>EndPoint</i> Tambah Pengeluaran. ....	51
47. Potongan Kode <i>EndPoint</i> Edit Pengeluaran. ....	52
48. Kembalian data <i>EndPoint</i> Edit Pemasukan. ....	52
49. Potongan Kode <i>EndPoint</i> Hapus Pengeluaran. ....	53
50. Kembalian data <i>EndPoint</i> Hapus Pemasukan. ....	53
51. Potongan Kode <i>EndPoint</i> Get Relawan. ....	54
52. Kembalian data <i>EndPoint</i> Get Relawan. ....	54
53. Potongan Kode <i>EndPoint</i> Tambah Relawan. ....	55
54. Kembalian data <i>EndPoint</i> Tambah Relawan. ....	55
55. Potongan Kode <i>EndPoint</i> Edit Relawan. ....	56
56. Kembalian data <i>EndPoint</i> Edit Relawan. ....	56
57. Potongan Kode <i>EndPoint</i> Hapus Relawan. ....	57
58. Kembalian data <i>EndPoint</i> Hapus Relawan. ....	57
59. <i>Use Case Diagram</i> Kedua. ....	58



60. <i>Entity Relationship Diagram</i> Kedua .....	59
61. Potongan kode <i>EndPoint</i> Get Kecakapan. ....	60
62. Kembalian data <i>EndPoint</i> Get Kecakapan. ....	61
63. Potongan kode <i>EndPoint</i> Tambah Kecakapan. ....	61
64. Kembalian data <i>EndPoint</i> Tambah Kecakapan. ....	62
65. <i>EndPoint</i> Hapus Kecakapan. ....	62
66. Kembalian data <i>EndPoint</i> Edit Kecakapan. ....	63
67. Potongan Kode <i>EndPoint</i> Hapus Kecakapan. ....	63
68. Kembalian data <i>EndPoint</i> Hapus Kecakapan. ....	64
69. Potongan Kode <i>EndPoint</i> Get Koordinator Kecakapan. ....	65
70. Kembalian data <i>EndPoint</i> Get Koordinator Kecakapan. ....	65
71. Potongan Kode <i>EndPoint</i> Tambah Koordinator Kecakapan. ....	65
72. Kembalian data <i>EndPoint</i> Tambah Koordinator Kecakapan. ....	66
73. Potongan Kode <i>EndPoint</i> Edit Koordinator Kecakapan. ....	66
74. Kembalian data <i>EndPoint</i> Edit Koordinator Kecakapan. ....	67
75. Potongan Kode <i>EndPoint</i> Hapus Koordinator Kecakapan. ....	67
76. Kembalian data <i>EndPoint</i> Hapus Koordinator Kecakapan. ....	68
77. Potongan Kode <i>EndPoint</i> Get Daftar Tugas. ....	68
78. Kembalian data <i>EndPoint</i> Get Daftar Tugas. ....	69
79. Potongan Kode <i>EndPoint</i> Tambah Daftar Tugas. ....	69
80. Kembalian data <i>EndPoint</i> Tambah Daftar Tugas. ....	70
81. Potongan Kode <i>EndPoint</i> Edit Daftar Tugas. ....	70
82. Kembalian data <i>EndPoint</i> Edit Daftar Tugas. ....	71
83. Potongan Kode <i>EndPoint</i> Hapus Daftar Tugas. ....	71
84. Kembalian data <i>EndPoint</i> Hapus Daftar Tugas. ....	72

**DAFTAR TABEL**

Tabel	Halaman
1. Rencana Kebutuhan Website Sistem Manajemen Relawan Bencana.....	18
2. Skenario Normal pada pengujian fungsionalitas .....	21
3. Skenario Abnormal Pada Pengujian Fungsionalitas .....	24
4. <i>Limit Work</i> Proses Pengembangan Pertama.....	30
5. Pengujian Normal Pada Proses Autentikasi dan Fungsionalitas.....	73
6. Pengujian Abormal Pada Proses Autentikasi dan Fungsionalitas.....	80

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar belakang**

Indonesia adalah negara yang rawan akan bencana alam. Bencana alam meliputi faktor geologi diantaranya gempa bumi, tsunami dan letusan gunung api, bencana akibat hydrometeorologi diantaranya banjir, tanah longsor, kekeringan dan angin topan, bencana akibat faktor biologi diantaranya wabah penyakit manusia, penyakit tanaman atau ternak, hama tanaman), serta kegagalan teknologi (kecelakaan industri, kecelakaan transportasi, radiasi nuklir, pencemaran bahan kimia). Berbagai macam bencana alam yang datang silih berganti berdampak pada Indonesia karena terletak pada zona subduksi atau pertemuan tiga lempeng tektonik yang bertabrakan, yaitu lempeng Indo-Australia, Eurasia, dan Pasifik. Ketika salah satu lempeng ini bergerak, gempa bumi, letusan gunung berapi, dan tsunami di Indonesia. Selain itu, Indonesia juga terletak di jalur gempa bumi dan gunung berapi dianggap paling merusak oleh *US Geological Survey*. Garis ini dikenal sebagai cincin api pasifik membentang dari belahan bumi barat hingga Cile, lalu melintasi Jepang dan Asia Tenggara. Hal itu juga didukung oleh posisi Indonesia yang berada di tengah-tengah. Garis khatulistiwa dan iklim tropis dengan dua musim berpotensi menghadirkan keragaman bahaya seperti banjir, tanah longsor, angin topan, kekeringan, dan kebakaran hutan dan lahan (Hayatul, 2020).

Sesaat setelah kejadian bencana, yaitu pada fase tanggap darurat, tim dari BNPB/BPBD yang pertama yaitu Tim Reaksi Cepat (TRC). TRC BNPB/BPBD yang terdiri dari berbagai instansi teknis maupun non

teknis terkait akan turun ke daerah bencana. Tim ini bertugas melaksanakan kaji cepat terhadap dampak bencana yang terjadi, yang mencakup penilaian kebutuhan (*Need Assessment*), penilaian kerusakan dan kerugian (*Damage and Loses Assessment*), serta memberikan dukungan untuk BPBD Provinsi dan Kota dalam penanganan darurat bencana (Perka BNPB No. 9 Tahun 2008). *Need Assessment* merupakan serangkaian kegiatan untuk menentukan jumlah dan jenis bantuan yang diperlukan dalam upaya penyelamatan korban bencana meliputi SAR, bantuan medis, penyediaan pangan, penyiapan penampungan sementara, penyediaan air bersih dan sanitasi, sedangkan *Damage and Loses Assessment* merupakan serangkaian kegiatan untuk pengumpulan data primer dan sekunder tentang jenis, waktu, lokasi dan penyebab bencana serta kondisi mutakhir yaitu korban, kerusakan dan kerugian serta dampak bencana.

Penanggulangan bencana pada masa tanggap darurat harus dilakukan secara cepat, tepat, dan dibawah satu komando, yang dipimpin oleh Komandan Tanggap Darurat Bencana. Keadaan darurat dan kritis saat terjadi bencana mengharuskan komandan tanggap darurat untuk dapat mengambil keputusan secara cepat dan tepat. Saat kegiatan tanggap darurat bencana, data bencana dari TRC sangat berguna sebagai dasar pengambilan keputusan terkait operasi tanggap darurat. Selain data TRC tersebut, juga ada data lain yang dibutuhkan seperti data relawan, data donatur, data bantuan yang diberikan, jumlah persediaan dan lokasinya, dan jumlah transportasi yang tersedia. Banyaknya jenis dan jumlah data yang harus dianalisis dan proses rekap data yang manual dapat mengakibatkan pengambilan keputusan menjadi tidak maksimal seperti keterlambatan dan kesalahan dalam pengambilan keputusan. Contoh kesalahan pengambilan keputusan yaitu kesalahan dalam pendistribusian bantuan karena salah dalam menganalisis data.

Selain kesalahan dari pengambilan keputusan, masalah lainnya yang dapat muncul saat proses tanggap darurat yaitu pada saat pendataan dan

pengalokasian relawan. Dalam bidang sosial, relawan didefinisikan sebagai mereka yang bergerak di bidang kesejahteraan sosial, tetapi bukan berasal (lulusan) atau tidak mendapat pendidikan khusus dari sekolah pekerjaan sosial ataupun Ilmu Kesejahteraan Sosial (Arliani, P.N.A., dkk., 2015). Relawan yang diterjunkan ke lapangan harus memenuhi standar dan kualifikasi agar kebutuhan pengungsi dapat diatasi oleh relawan, seperti kebutuhan medis, logistik, dan air bersih. Pendataan dan pengalokasian relawan yang tidak merata dapat menyebabkan kerugian baik untuk pengungsi maupun untuk relawan, seperti kejadian relawan yang meninggal dunia pada gempa bumi di Lombok tahun 2018 yang disebabkan karena faktor kelelahan dan pembagian tugas yang tidak merata bagi relawan yang menyebabkan relawan tersebut memiliki tugas yang berat yang berujung kepada kematian (tribunnews.com, 2018).

Penelitian terkait sistem informasi bencana yang telah dilakukan diantaranya oleh (Elsa, R.G., 2017) yang berjudul “Perancangan Sistem Informasi Manajemen Kapasitas Logistik Penanggulangan Bencana Kota Padang”. Sistem informasi yang dirancang menyediakan informasi ketersediaan sumber daya, fasilitas, dan alat yang bisa digunakan untuk kegiatan logistik. (Mahdia, F., 2013) meneliti “Pemanfaatan *Google Maps API* untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis *Mobile Web* (Studi Kasus: BPBD Kota Yogyakarta)”. Penelitian ini membahas tentang perancangan kebutuhan logistik dan informasi posko yang dibuat dalam bentuk mobile web yang dilacak melalui GPS untuk menentukan letak posko, yang memudahkan donatur dan relawan dalam menyalurkan bantuan logistik dengan memanfaatkan layanan *Google Maps*. Terkait penyaluran bantuan logistik, (Mahdia, F., 2013) belum memperhitungkan kondisi sebenarnya dari suatu wilayah, seperti kondisi jalan yang rusak karena bencana yang menyebabkan terhambatnya transportasi logistik, dan sumber daya kendaraan pengangkut logistik yang tersedia karena *Google Maps* yang digunakan belum bisa menyediakan layanan terkait skenario tersebut.

Berdasarkan penelitian sebelumnya, belum ada penelitian mengenai sistem manajemen data relawan untuk mempercepat proses operasi tanggap darurat bencana perlu dilakukan. Dengan data dan informasi yang disediakan diharapkan dapat mempermudah pekerjaan karyawan BPBD dalam pengelolaan data relawan.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang maka dapat disimpulkan bahwa rumusan masalah yang akan diangkat pada penelitian ini yaitu bagaimana membangun sistem manajemen relawan berbasis *website*, untuk memudahkan pegawai BPBD Provinsi Lampung dalam memanajemen relawan terhadap bencana yang sedang terjadi pada suatu daerah.

### **1.3. Batasan Masalah**

Dalam menyusun laporan ini, penyusun memberi batasan sebagai berikut :

1. Sistem ini digunakan oleh pegawai BPBD Provinsi Lampung.
2. Sistem ini digunakan untuk memanajemen kegiatan, target, kecakapan, laporan harian, pemasukan, pengeluaran, pendaftaran relawan pada suatu bencana.
3. Sistem yang dibuat berbasis *website* dan menggunakan *framework* CodeIgniter.
4. Penelitian ini sebatas pengembangan bagian *backend* pada sistem manajemen relawan.

### **1.4. Tujuan Penelitian**

Tujuan yang ingin dicapai dalam penelitian adalah untuk membangun dan melakukan implementasi *backend* pada sebuah sistem manajemen relawan berbasis web pada Instansi BPBD Provinsi Lampung.

### **1.5. Manfaat Penelitian**

Adapun manfaat penelitian ini sebagai berikut:

1. Terbantunya pegawai BPBD dalam memajemen relawan bencana.
2. Terbantunya masyarakat yang memiliki jiwa sosial untuk menjadi relawan.
3. Terbantunya tata kelola relawan dengan data yang lengkap dan akurat.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Penelitian Terdahulu**

Penelitian terdahulu bertujuan untuk membandingkan penelitian yang sudah ada dengan penelitian yang akan dilakukan. Selain itu, penelitian terdahulu akan dijadikan sebagai acuan dalam upaya tinjauan pustaka terkait dengan penelitian yang dilakukan. Penelitian tersebut antara lain: Penelitian terkait sistem informasi manajemen bencana yang telah dilakukan diantaranya oleh Kartadie, dkk. (2013) yang berjudul “Aplikasi Sistem Informasi Manajemen Penanggulangan Pasca Bencana”. Penelitian ini menggunakan metode Spiral hasil dari penelitian ini adalah sebuah *Software MID (Management Informations Disaster)*.

Siswanto (2017) meneliti “Sistem Informasi Manajemen Komando Tanggap Darurat Bencana Letusan Gunung Merapi”. Penelitian ini menggunakan metode FAST hasil dari penelitian ini adalah sebuah sistem informasi manajemen komando tanggap darurat bencana letusan Gunung Merapi.

Elsa (2017) meneliti “Perancangan Sistem Informasi Manajemen Kapasitas Logistik Penanggulangan Bencana Kota Padang”. Sistem informasi yang dirancang menyediakan informasi ketersediaan sumber daya, fasilitas, dan alat yang bisa digunakan untuk kegiatan logistik.

Pradhana P.F. (2020) meneliti “Sistem Informasi Manajemen Kebencanaan (SIMAK PRB)”. Sistem yang dirancang berfungsi



sebagai manajemen informasi kebencanaan dalam program pengurangan resiko bencana sekaligus sebagai aplikasi *mobile* untuk akses berita, artikel, dan informasi yang ada pada *website* BPBD Kabupaten Blitar.

Iskandar A., dkk. (2022) “Implementasi Sistem Pengelolaan Donasi, Kegiatan, dan Relawan Komunitas Sosial Turun Tangan Malang berbasis Web dengan Framework Codeigniter”. Sistem yang dibuat membantu komunitas Turun Tangan Malang dalam mencari donasi, mengelola kegiatan dan relawan.

## **2.2. Backend**

Pada pengembangan web, kebanyakan *framework* menggunakan model arsitektur MVC (*Model View Controller*). *Model* mengandung fungsi-fungsi yang berhubungan dengan *database*, seperti *create*, *read*, *update*, dan *delete* (Prabowo, D., 2015). *View* adalah *user interface* yang melakukan interaksi secara langsung dengan pengguna (Riana, D., 2018). *Controller* mengandung proses input dari pengguna, kemudian mengirim perintah kepada model dan *view* agar melakukan proses sesuai dengan masukan. Singkatnya, *controller* merupakan pemetaan aksi pengguna terhadap respons sistem (Hidayat, A., 2012).

Berdasarkan arsitektur MVC, *frontend developer* bertugas untuk membuat bagian *view*, yaitu membuat *user interface* yang berhubungan langsung dengan pengguna. Sedangkan *backend developer* bertugas untuk membuat *model* dan *controller*. *Backend* adalah tempat dimana proses pada suatu sistem informasi atau aplikasi berjalan, data dapat ditambahkan, diubah maupun dihapus. *Backend* biasanya mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna, seperti *server* dan *database*. *Backend* dibutuhkan dalam pengembangan sistem dan manajemen data pada sistem (Arhandi, P.P., 2016).

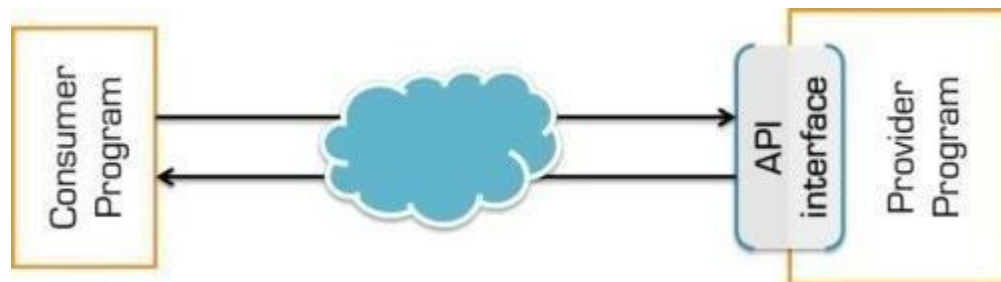
### 2.3. API

API merupakan antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan *developer* untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. (Cooksey, B., 2014) mendefinisikan API sebagai *tools* yang dapat membuat sistem saling terhubung. API terbagi menjadi 2 bagian, yaitu server sebagai penyedia dari API dan klien sebagai program yang memanipulasi data API. REST (*REpresentational State Transfer*) adalah jenis arsitektur sistem hypermedia terdistribusi. Dalam arsitektur *client-server* yang dikenal sebagai REST, server merespon permintaan yang dikirim oleh klien. REST menggunakan metode HTTP GET, POST, PUT, dan DELETE (Mumbaikar & Padiya, 2013). Menurut (Novianto, M.A., dkk., 2022) istilah yang biasa digunakan untuk menyatakan prinsip *uniform interface* pada REST adalah CRUD (*Create, Read, Update, Delete*). Berikut ini ulasan detail mengenai *method-method* tersebut :

- a. GET, operasi *read-only* yang digunakan untuk meminta informasi spesifik pada server dalam bentuk *query*. Karakteristik dari operasi GET adalah *idempotent* dan *safe*.
- b. PUT, operasi untuk meminta kepada *server* agar membuat sebuah *resource* baru.
- c. DELETE, operasi untuk menghapus *resource* tertentu.
- d. POST, operasi untuk membuat *resource* baru atau memodifikasi *resource* yang telah ada.

Perumpamaan yang bisa digunakan untuk menjelaskan API adalah seorang pelayan di restoran. Tugas pelayan tersebut adalah menghubungkan tamu restoran dengan juru masak. Jadi, tamu cukup memesan makanan sesuai daftar menu yang ada dan pelayan memberitahukannya ke juru masak.

Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan (Lawrence, A., 2020).



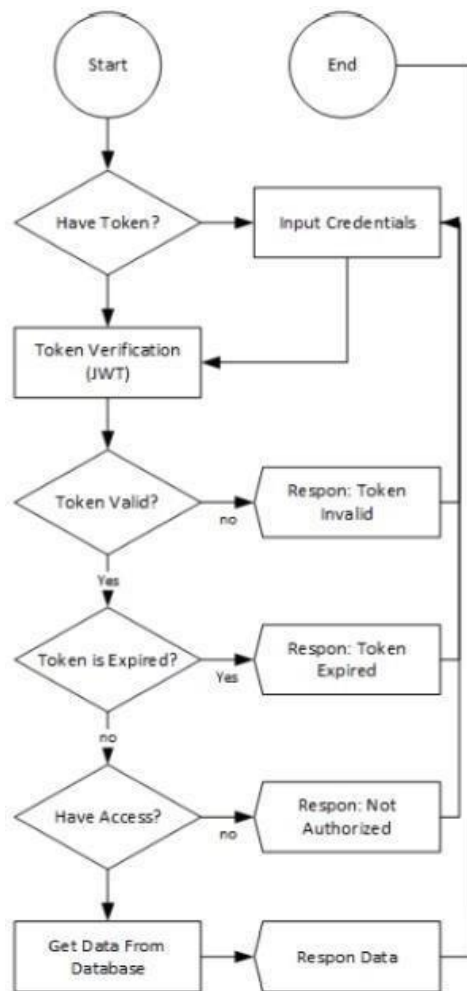
Gambar 2.1. Cara Kerja API

#### 2.4. *JSON Web Token (JWT)*

Pada proses transmisi data antar sistem, dibutuhkan keamanan tingkat tinggi yang mampu menjamin integritas data dalam hal keutuhan dan konsistensi data, khususnya sistem yang memiliki data sensitif. *Exposure* data sensitif melalui *Application Programming Interface (API)* merupakan salah satu kerentanan pada keamanan sistem berbasis web (Parabi, dkk., 2023).

*Json Web Token* adalah sebuah token berbentuk *string* JSON yang sangat padat (ukurannya), informasi mandiri yang gunanya sendiri untuk melakukan sistem autentikasi dan pertukaran informasi. Karena bentuknya kecil, token JWT dapat dikirim melalui URL, parameter HTTP POST atau di dalam *header* HTTP, dan juga karena ukurannya yang kecil maka dapat ditransmisikan dengan lebih cepat. Disebut informasi mandiri karena isi dari token yang dihasilkan memiliki informasi dari pengguna yang dibutuhkan, sehingga tidak perlu query ke basis data lebih dari satu kali (Rahmatulloh, A., 2018). JWT bekerja dimana token ini seperti *password*, jadi ketika *user* berhasil melakukan *login* maka *server* akan memberikan sebuah token. Nanti token tersebut akan disimpan oleh *user* pada *local storage* atau *cookies browser* dan bila *user* ingin mengakses halaman halaman tertentu maka harus menyertakan token tersebut. Untuk itu *user* akan mengirim balik token yang diberi sebelumnya sebagai bukti bila *user* ini, sudah melakukan *login*.

JWT adalah standar format untuk mengamankan informasi pribadi menjadi sebuah klaim yang akan di encode ke dalam bentuk JSON dan menjadi dari JSON Web Signature (JWS). Klaim akan dapat dilindungi dengan tanda tangan digital seperti *Message authentication code* (MAC) atau dienkripsi (Bradley, 2015).



Gambar 2.2. *Json Web Token Verification*

## 2.5. Aplikasi Berbasis Web

Aplikasi berbasis web adalah jenis aplikasi yang dapat dijalankan dengan *web browser*. Berbeda dengan sebuah situs web tradisional, aplikasi web menyediakan fungsionalitas interaktif dan memungkinkan pengguna untuk melakukan tugas tugas dan berinteraksi dengan konten yang dinamis.

Aplikasi web biasanya dirancang untuk meniru fungsionalitas dari aplikasi *desktop* atau *mobile* seraya memanfaatkan keunggulan dari platform web. Aplikasi berbasis web memiliki kapabilitas untuk menerima input pengguna, melakukan operasi-operasi kompleks, dan menyimpan maupun mengambil data dari server (Amanatidis, T., dan Chatzigeorgiou, A., 2016). Teknologi pemrograman seperti HTML, CSS, dan JavaScript serta bahasa pemrograman *server-side* seperti PHP, Python, dan Ruby digunakan untuk membangun aplikasi berbasis web. Selain itu, terdapat teknologi dan *framework* seperti Laravel, CodeIgniter, dan Ruby on Rails yang dapat digunakan untuk memudahkan pengembangan aplikasi berbasis web.

### **2.5.1. PHP**

PHP atau *Hypertext Preprocessor* merupakan bahasa pemrograman yang banyak digunakan dalam pengembangan *website* (*web development*). Bahasa ini umum dijalankan dalam komunikasi *side server* dan saat ini didukung oleh hampir semua sistem. PHP merupakan bahasa penulisan skrip yaitu bahasa yang mengotomatiskan *task execution* dalam *environment runtime* khusus. *Task* yang dimaksud mencakup instruksi *static page* dengan HTML dan CSS.

### **2.5.2. XAMPP**

XAMPP merupakan singkatan dari X (cross platform), A (Apache), M (MySQL/MariaDB), P (PHP), dan P (Perl). XAMPP adalah perangkat lunak berbasis *web server* yang bersifat *open source* yang didapatkan secara gratis dengan label GNU (General Public License) serta didukung di berbagai sistem operasi yang banyak digunakan web developer. XAMPP mampu menggantikan peran web hosting dengan cara menyimpan file *website* ke dalam hosting lokal dan dapat dipanggil melalui *browser*.

### 2.5.3. MySQL

MySQL adalah sebuah *database program server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user*, serta menggunakan perintah dasar SQL (*Structured Query Language*). *Database MySQL* merupakan suatu *database software* yang berbentuk RDBMS (*Relational Database Management System*) yang menggunakan bahasa permintaan SQL. MySQL merupakan *FreeSoftware* yang berada dibawah Lisensi GNU/GPL (*General Public License*).

### 2.5.4. Web Browser

*Web Browser* adalah aplikasi perangkat lunak yang digunakan untuk mengakses informasi tentang *World Wide Web* disebut *Web Browser*. Ketika pengguna meminta beberapa informasi, browser web mengambil data dari server web dan kemudian menampilkan halaman web di layar pengguna.

### 2.5.5. Postman

Postman merupakan sebuah *software* yang memuat fungsi lengkap pengembangan sistem dalam mengirimkan dan menerima respon *server*. Software ini mendukung pengembangan sistem REST API dengan mengklasifikasi *request* berdasarkan *request method*, URL dan parameter-parameter *request*.

### 2.5.6. CodeIgniter

*Framework* adalah sebuah *toolkit* yang di dalamnya terdapat berbagai *class-class* yang dapat digunakan untuk membuat web. Dengan menggunakan *framework* ini, memungkinkan untuk membuat web dengan mudah dan cepat. Sedangkan CodeIgniter merupakan sebuah *framework* yang dibuat dengan menggunakan bahasa PHP, yang dapat digunakan untuk pengembangan web secara cepat. CodeIgniter juga sudah menggunakan konsep MVC (*Models-View-Controller*) yang memisahkan

antara data dan presentasi sehingga memungkinkan pengembangan sebuah web dengan cepat dan memudahkan proses pengelolaan web tersebut. (Afuan, L., 2010).

## **2.6. Database**

Menurut (Kadir, A., 2003), basis data adalah suatu pengorganisasian, sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi.

*Database* terdiri dari data yang akan digunakan atau ditujukan untuk banyak pengguna, dimana setiap pengguna akan menggunakan data tersebut sesuai dengan tugas dan fungsinya, dan lain-lain pengguna juga dapat menggunakan data secara bersamaan. (Ladjamudin, 2005). 15 kelebihan *database* pada sistem pengolah *file* menurut Ladjamudin (2005) adalah sebagai berikut :

1. Mengurangi pemborosan data.
2. Penggunaan data lebih mudah.
3. Keamanan data lebih mudah.
4. Berbagi data selalu dapat dilakukan oleh setiap pengguna.

Sistem Manajemen Basis Data adalah perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membangun, dan mengelola *database* dan yang mengontrol akses ke *database*. DBMS berinteraksi dengan program aplikasi pengguna dan *database*. DBMS menyediakan fasilitas sebagai berikut :

- *Data definition language* (DDL), yang berguna untuk menentukan tipe data, struktur data dan kendala. Semua spesifikasi disimpan di basis data.
- Deskripsi katalog data dalam *database*, terdiri dari deskripsi data yang ada pada *database*.
- Mekanisme tampilan, yang berfungsi untuk menunjukkan pengguna data yang dibutuhkan dan hanya digunakan.

- *Data manipulation language* (DML), berguna untuk menyediakan fasilitas kueri data. Kontrol akses *database*, termasuk pengendalian:
  - a. Keamanan sistem, mencegah pengguna yang tidak sah dari mengakses basis data.
  - b. Integritas sistem.
  - c. Kontrol berbagi data.
  - d. Pencadangan dan pemulihan sistem.

## **2.7. Metode Pengembangan Sistem Model RAD**

Metode pengembangan sistem yang digunakan adalah *Rapid Application Development* (RAD). RAD adalah metodologi pengembangan sistem yang menggunakan desain aplikasi bersama, pembuatan prototipe, teknologi rekayasa perangkat lunak berbantu komputer, generator aplikasi, dan alat serupa untuk mempercepat proses desain. RAD mempromosikan pengembangan dan penyampaian program dan sistem yang akurat. Tahapan RAD:

### **a. Perencanaan Kebutuhan**

Perencanaan kebutuhan sistem merupakan tahapan untuk mengetahui apa saja yang menjadi kebutuhan sistem dengan cara mengidentifikasi kebutuhan informasi berdasarkan hasil pengumpulan data-data dari pengguna sistem tersebut.

### **b. Desain Sistem**

Tahap desain mencakup proses desain dan proses perbaikan yang dilakukan berulang agar tepat sesuai pada kebutuhan pengguna yang telah diidentifikasi pada tahap perencanaan kebutuhan.

### **c. Proses Pengembangan dan Pengumpulan *Feedback***

Tahap ini adalah tahapan dimana tahap desain sistem yang telah disepakati langsung dilakukan pengembangan dan terus menyesuaikan



*feedback* dari pengguna. Tahap ini akan selesai apabila seluruh kebutuhan pengguna telah sesuai dan dikembangkan.

#### **d. Implementasi**

Tahap implementasi merupakan tahap dimana seluruh sistem telah selesai dikembangkan dan kebutuhan pengguna sudah sesuai sehingga sistem siap untuk dilakukan proses pengujian.

### **2.8. Pengujian Perangkat Lunak**

Pengujian perangkat lunak dibutuhkan untuk melakukan konfirmasi dan validasi dari hasil pengembangan suatu perangkat lunak. Pengujian perangkat lunak dilakukan secara sistematis dengan berbagai strategi pengujian perangkat lunak untuk mencegah kesalahan yang dibuat oleh manusia dalam suatu sistem, atau dengan kata lain untuk mencegah perangkat lunak yang ada dari cacat (Huda, M., dkk., 2022).

Pengujian yang dilakukan pada penelitian ini adalah pengujian fungsionalitas. Pengujian fungsional merupakan metode pengujian perangkat lunak yang bertujuan untuk memverifikasi kesesuaian aplikasi perangkat lunak atau sistem yang sedang diuji terhadap persyaratan fungsional yang telah ditetapkan dalam desain perangkat lunak. Prosedur ini melibatkan pengujian aplikasi perangkat lunak atau sistem berdasarkan spesifikasi fungsional guna memastikan bahwa aplikasi tersebut beroperasi sesuai dengan harapan (Rafiq, M., dkk., 2020).

Pengujian fungsional dapat dilakukan dengan cara manual atau otomatis. Pengujian fungsional otomatis adalah varian pengujian perangkat lunak yang menggunakan alat perangkat lunak untuk mengotomatisasi pelaksanaan pengujian fungsional. Dalam pengujian fungsional otomatis, tindakan pengujian yang telah disiapkan sebelumnya dieksekusi oleh alat perangkat lunak. Tindakan tersebut direkam dan ditentukan sebelumnya, dan hasilnya dibandingkan dengan perilaku yang diharapkan. Keberhasilan atau kegagalan pengujian dilaporkan (Rafiq, M., dkk., 2020).

## **BAB III METODOLOGI PENELITIAN**

### **3.1. Waktu dan Tempat Penelitian**

Penelitian dilakukan pada semester genap 2023/2024 di Gedung Ilmu Komputer Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung dan dikediaman Lampung Selatan yaitu di Hajimena, Natar, Lampung Selatan.

### **3.2. Alat Penelitian**

Terdapat beberapa perangkat yang dibutuhkan dalam pengimplementasian sistem pendukung keputusan ini, yaitu sebagai berikut:

#### **3.2.1. Kebutuhan Perangkat Keras (*Hardware*)**

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:

- *Processor* : Intel core i3
- *Installed RAM* : 8.00 GB
- *System Type* : 64-bit operating system, x-64-based processor.

#### **3.2.2. Kebutuhan Perangkat Lunak (*Software*)**

Perangkat lunak yang digunakan dalam perancangan sistem informasi adalah sebagai berikut:

- Sistem Operasi : Windows 10 64 bit.
- Aplikasi:
  - o CodeIgniter 4 sebagai *framework*.

- o Visual Studio Code untuk *coding* program.
- o Star UML sebagai pemodelan sistem.
- o *Web Browser* untuk menjalankan sistem.
- o XAMPP sebagai *web server* lokal.

### 3.3. Metode Pengumpulan Data

#### 3.3.1. Data Primer

Data Primer didapatkan dari observasi dan wawancara langsung melalui fitur-fitur yang diperlukan dalam pengembangan *website*.

#### 3.3.2. Data Sekunder

Data sekunder diperoleh dari studi kepustakaan seperti buku, jurnal, artikel dan situs resmi. Situs resmi menyediakan informasi tentang penerapan *Framework Codeigniter 4*, Github, dan studi serupa.

### 3.4. Tahapan Penelitian

Pada penelitian ini terdapat lima tahapan, yaitu Identifikasi Masalah, Studi Literatur, Penerapan Metode RAD, Evaluasi Sistem, dan Pembuatan Laporan yang dapat dilihat pada Gambar 3.1.



Gambar 3.1. Tahapan Penelitian

#### 3.4.1. Identifikasi Masalah

Identifikasi masalah merupakan tahapan untuk mengidentifikasi masalah. Hasil dari tahap ini adalah rencana kebutuhan yang akan digunakan pada pengembangan sistem.

### 3.4.2. Studi Literatur

Studi literatur adalah teknik mengumpulkan data menggunakan dokumen-dokumen yang akan digunakan dalam analisis dan pengembangan sistem. Studi literatur berupa penelitian terdahulu dapat dilihat pada sub bab 2.1.

### 3.4.3. Penerapan metode RAD

Metode *Rapid Application Development* (RAD) memiliki beberapa tahapan yang terdiri dari perencanaan kebutuhan, desain sistem, proses pengembangan dan *feedback user*, dan implementasi. Tahapan-tahapan pada metode ini harus dilakukan secara berurutan, dalam artian apabila tahapan sebelumnya belum selesai, pengembangan tidak bisa dilanjutkan ke tahap selanjutnya.

### 3.4.4. Rencana Kebutuhan

Rencana Kebutuhan pada penelitian ini adalah sebagai berikut:

Tabel 3.1. Rencana Kebutuhan *Website* Sistem Manajemen Relawan Bencana

No.	Nama	Deskripsi
1.	Admin	<ol style="list-style-type: none"> <li>1. Dapat <i>login</i> ke sistem</li> <li>2. Dapat mengelola daftar kegiatan</li> <li>3. Dapat memvalidasi laporan harian</li> <li>4. Dapat mengelola target</li> </ol>
2.	Relawan	<ol style="list-style-type: none"> <li>1. Dapat <i>register</i> ke sistem</li> <li>2. Dapat <i>login</i> ke sistem</li> <li>3. Dapat mendaftar kegiatan</li> <li>4. Dapat mengakses target</li> <li>5. Dapat mengelola pemasukan</li> <li>6. Dapat mengelola pengeluaran</li> <li>7. Dapat menambahkan laporan harian</li> <li>8. Dapat mengubah laporan harian</li> </ol>

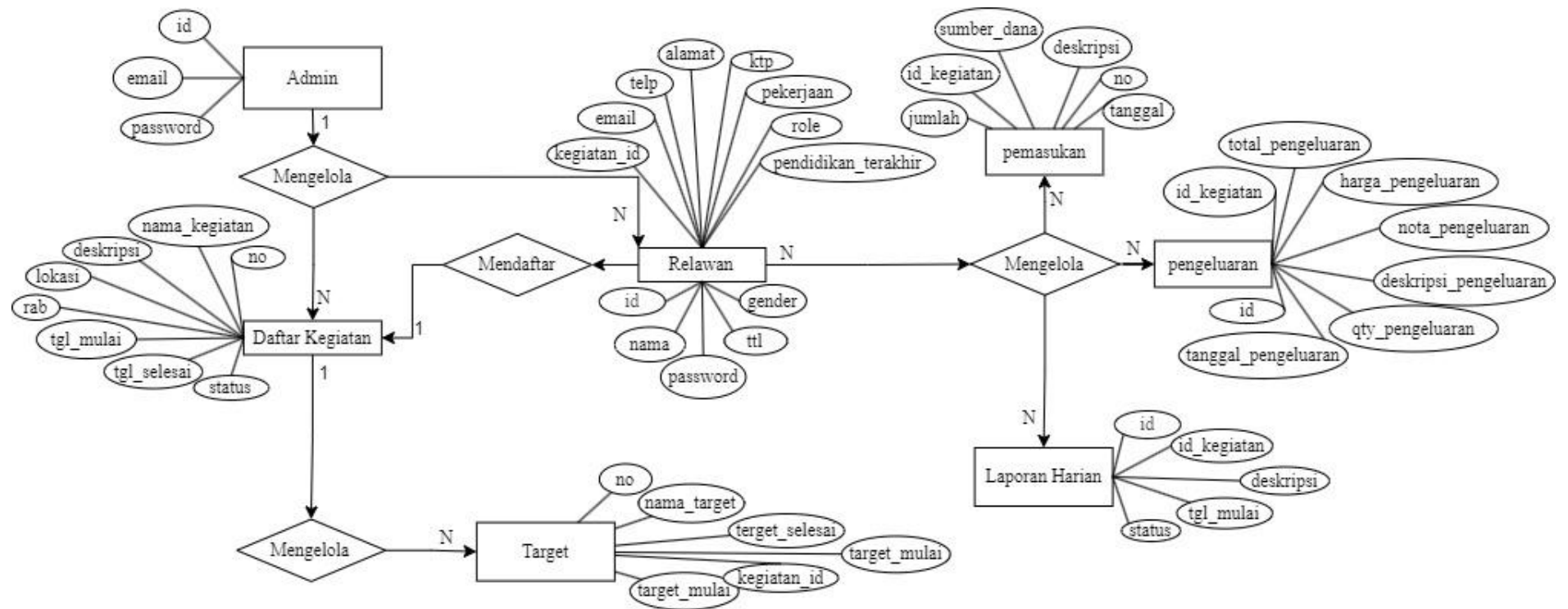
Pada Rencana Kebutuhan Tabel 3.1. diperoleh dari hasil analisis kebutuhan pengembangan *website* yang dilakukan *developer*. Desain

sistem kemudian diilustrasikan menggunakan *Use Case Diagram* dengan mengacu pada Rencana Kebutuhan yang dapat dilihat pada Gambar 3.2.



Gambar 3.2. *Use Case Diagram* Sistem Manajemen Relawan Bencana

*Use Case Diagram* pada Gambar 3.2 akan menjadi acuan dalam pengembangan sistem pada penelitian ini. *Entity Relationship Diagram* (ERD) pada Gambar 3.3 merupakan hasil dari *Use Case Diagram*.



Gambar 3.3.ERD Sistem Manajemen Relawan Bencana

### 3.4.5. Pengujian Sistem

Pengujian yang dilakukan pada penelitian ini adalah prosedur pengujian *black-box*. Pengujian *black-box* ini bertujuan untuk menentukan ketepatan fungsionalitas sistem dengan yang diinginkan. Pengujian ini hanya menguji tahap *alpha* yang akan dilakukan oleh tim pengembang untuk memastikan sistem berhasil berjalan sesuai skenario. Berikut ini skenario pengujian yang dilakukan:

Tabel 3.2. Skenario Normal pada pengujian fungsionalitas

No.	Kelas Uji	Daftar Pengujian	Hasil yang diharapkan
1.	<i>EndPoint Login</i>	Fungsi <i>login</i>	Masuk ke dalam sistem
2.	<i>EndPoint Create token</i>	<i>Create token</i>	Token berhasil dibuat
3.	<i>EndPoint Validasi token</i>	Validasi token	Token berhasil divalidasi
4.	<i>EndPoint Get Daftar Kegiatan</i>	Mengirim seluruh data Daftar Kegiatan	Berhasil mengirim seluruh data Daftar Kegiatan
5.	<i>EndPoint Tambah Daftar Kegiatan</i>	Menambah data kegiatan yang di- <i>input</i> -kan ke dalam basis data	Berhasil Menambah data yang di- <i>input</i> -kan ke dalam basis data

<b>No.</b>	<b>Kelas Uji</b>	<b>Daftar Pengujian</b>	<b>Hasil yang diharapkan</b>
6.	<i>EndPoint</i> Edit Daftar Kegiatan	Mengubah data kegiatan yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Mengubah data kegiatan yang telah di- <i>input</i> -kan ke dalam basis data
7.	<i>EndPoint</i> Hapus Daftar Kegiatan	Menghapus data kegiatan yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Menghapus data kegiatan yang telah di- <i>input</i> -kan ke dalam basis data
8.	<i>EndPoint Get</i> Daftar Laporan Harian	Mengirim seluruh data Daftar Laporan Harian	Berhasil Mengirim seluruh data Daftar Laporan Harian
9.	<i>EndPoint</i> Tambah Laporan Harian	Menambah data Laporan Harian yang di- <i>input</i> -kan ke dalam basis data	Berhasil Menambah data Laporan Harian yang di- <i>input</i> -kan ke dalam basis data
10.	<i>EndPoint</i> Edit Laporan Harian	Mengubah data Laporan Harian yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Mengubah data Laporan Harian yang telah di- <i>input</i> -kan ke dalam basis data
11.	<i>EndPoint</i> Hapus Laporan Harian	Menghapus data Laporan Harian yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Menghapus data Laporan Harian yang telah di- <i>input</i> -kan ke dalam basis data
12.	<i>EndPoint</i> Daftar Target	Mengirim seluruh data Daftar Target	Berhasil Mengirim seluruh data Daftar Target
13.	<i>EndPoint</i> Tambah Target	Menambah data Target yang di- <i>input</i> -kan ke dalam basis data	Berhasil Menambah data Target yang di- <i>input</i> -kan ke dalam basis data



No.	Kelas Uji	Daftar Pengujian	Hasil yang diharapkan
14.	<i>EndPoint</i> Edit Target	Mengubah data Target yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Mengubah data Target yang telah di- <i>input</i> -kan ke dalam basis data
15.	<i>EndPoint</i> Hapus Target	Menghapus data Target yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Menghapus data Target yang telah di- <i>input</i> -kan ke dalam basis data
16.	<i>EndPoint Get</i> Pemasukan	Mengirim seluruh data Pemasukan	Berhasil Mengirim seluruh data Pemasukan
17.	<i>EndPoint</i> Tambah Pemasukan	Menambah data Pemasukan yang di- <i>input</i> -kan ke dalam basis data	Berhasil Menambah data Pemasukan yang di- <i>input</i> - kan ke dalam basis data
18.	<i>EndPoint Edit</i> Pemasukan	Mengubah data Pemasukan yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Mengubah data Pemasukan yang telah di- <i>input</i> -kan ke dalam basis data
19.	<i>EndPoint</i> Hapus Pemasukan	Menghapus data Pemasukan yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Menghapus data Pemasukan yang telah di- <i>input</i> -kan ke dalam basis data
20.	<i>EndPoint Get</i> Pengeluaran	Mengirim seluruh data Pengeluaran	Berhasil Mengirim seluruh data Pengeluaran
21.	<i>EndPoint</i> Tambah Pengeluaran	Menambah data Pengeluaran yang di- <i>input</i> -kan ke dalam basis data	Berhasil Menambah data Pengeluaran yang di- <i>input</i> - kan ke dalam basis data
22.	<i>EndPoint Edit</i> Pengeluaran	Mengubah data Pengeluaran yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Mengubah data Pengeluaran yang telah di- <i>input</i> -kan ke dalam basis data
23.	<i>EndPoint</i> Hapus Pengeluaran	Menghapus data Pengeluaran yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Menghapus data Pengeluaran yang telah di- <i>input</i> -kan ke

No.	Kelas Uji	Daftar Pengujian	Hasil yang diharapkan
			dalam basis data
24.	<i>EndPoint Get</i> Pendaftaran Relawan	Mengirim seluruh data Pendaftaran Relawan	Berhasil Mengirim seluruh data Pendaftaran Relawan
25.	<i>EndPoint Tambah</i> Pendaftaran Relawan	Menambah data Pendaftaran Relawan yang di- <i>input</i> -kan ke dalam basis data	Berhasil Menambah data Pendaftaran Relawan yang di- <i>input</i> - kan ke dalam basis data
26.	<i>EndPoint Edit</i> Pendaftaran Relawan	Mengubah data Pendaftaran Relawan yang telah di- <i>input</i> - kan ke dalam basis data	Berhasil Mengubah data Pendaftaran Relawan yang telah di- <i>input</i> -kan ke dalam basis data
27.	<i>EndPoint Hapus Pendaftaran</i> Relawan	Menghapus data Relawan yang telah di- <i>input</i> -kan ke dalam basis data	Berhasil Menghapus data Relawan yang telah di- <i>input</i> -kan ke dalam basis data

Tabel 3.3. Skenario Abnormal Pada Pengujian Fungsionalitas

No.	Kelas Uji	Daftar Pengujian	Hasil yang diharapkan
1.	<i>EndPoint Login</i>	<i>Input email</i> dan <i>password</i> salah	Tidak berhasil <i>login</i>
2.	<i>EndPoint Login</i>	<i>Input email</i> dan <i>password</i> kosong	Tidak berhasil <i>login</i>
3.	<i>EndPoint Validasi token</i>	<i>Input token</i> salah	Token gagal divalidasi
4.	<i>EndPoint Tambah Kegiatan</i>	<i>Input data kegiatan</i> tidak sesuai format	Gagal menambahkan kegiatan
5.	<i>EndPoint Edit Kegiatan</i>	Mengubah data kegiatan tidak sesuai format	Gagal mengubah kegiatan
6.	<i>EndPoint Tambah Laporan</i> Harian	Menambah data Laporan Harian tidak sesuai format	Gagal Menambah data Laporan Harian
7.	<i>EndPoint Edit Laporan</i> Harian	Mengubah data Laporan Harian tidak sesuai format	Gagal Mengubah data Laporan Harian
8.	<i>EndPoint Tambah Target</i>	Menambah data Target tidak sesuai format	Gagal Menambah data Target
9.	<i>EndPoint Edit Target</i>	Mengubah data Target tidak sesuai format	Gagal Mengubah data Target

No.	Kelas Uji	Daftar Pengujian	Hasil yang diharapkan
10.	<i>EndPoint</i> Tambah Pemasukan	Menambah data Pemasukan tidak sesuai format	Gagal Menambah data Pemasukan yang di- <i>input</i> -kan ke dalam basis data
11.	<i>EndPoint Edit</i> Pemasukan	Mengubah data Pemasukan tidak sesuai format	Gagal Mengubah data Pemasukan
12.	<i>EndPoint</i> Tambah Pengeluaran	Menambah data Pengeluaran tidak sesuai format	Gagal Menambah data Pengeluaran
13.	<i>EndPoint Edit</i> Pengeluaran	Mengubah data Pengeluaran tidak sesuai format	Gagal Mengubah data Pengeluaran
14.	<i>EndPoint</i> Tambah Pendaftaran Relawan	Menambah data Pendaftaran Relawan tidak sesuai format	Gagal Menambah data Pendaftaran Relawan
15.	<i>EndPoint Edit</i> Pendaftaran Relawan	Mengubah data Pendaftaran Relawan tidak sesuai format	Gagal Mengubah data Pendaftaran Relawan

### 3.4.6. Penulisan Laporan

Pada akhir penelitian dan telah melakukan pengembangan perangkat lunak. Tahap selanjutnya adalah penulisan laporan yang bertujuan untuk bahan referensi penelitian yang akan datang maupun sebagai dokumentasi terhadap penelitian ini.

## **BAB V**

### **SIMPULAN DAN SARAN**

#### **5.1. Simpulan**

Penelitian yang telah dilakukan memiliki hasil yang dapat disimpulkan sebagai berikut:

1. Telah Dibangunnya *Backend* Sistem Manajemen Relawan Bencana berbasis *website* menggunakan *framework* CodeIgniter 4.
2. *User* pada sistem ini terbagi menjadi 4 *role* yaitu *admin*, relawan, ketua tim, dan koordinator kecakapan. Pada *role admin* dapat mengelola kecakapan, daftar kegiatan, dan *user*. Untuk *role* relawan dapat memilih kegiatan serta mengelola pemasukan dan pengeluaran. Sedangkan ketua tim dan koor kecakapan memiliki akses yang sama dengan relawan, namun untuk koor kecakapan dapat mengisi laporan harian yang nantinya akan divalidasi oleh ketua tim. Untuk *role* ketua tim dapat mengelola target dan memvalidasi laporan mingguan.
3. Pengembangan sistem ini menggunakan metode RAD.
4. Sistem Manajemen Relawan Bencana ini diuji menggunakan pengujian dengan metode *black-box testing* dan mendapatkan hasil sesuai dengan skenario yang diharapkan.

#### **5.2. Saran**

Berdasarkan rangkaian proses yang telah dilakukan, didapati beberapa saran untuk penelitian ini dimasa mendatang. Beberapa saran tersebut adalah sebagai berikut:

1. Mengembangkan fitur untuk manajemen distribusi relawan agar pembagian tugas relawan lebih merata.
2. Menambahkan fitur notifikasi bencana untuk memperingatkan relawawn akan terjadinya bencana sehingga relawan dapat bertindak menanggulangi bencana.
3. Menambahkan fitur notifikasi ketika relawan telah divalidasi oleh admin agar relawan mengetahui status validasinya.
4. Menambahkan fitur presensi relawan untuk meningkatkan akurasi presensi relawan.

## DAFTAR PUSTAKA

- Aufan, L. (2010). Pemanfaatan Framework Codeigniter dalam Pengembangan Sistem Informasi Pendataan Laporan Kerja Praktek Mahasiswa Program Studi Teknik Informatika Unsoed. *JUITA: Jurnal Informatika*, 1(2).
- Amanatidis, T., & Chatzigeorgiou, A. (2016). Studying the evolution of PHP web applications. *Information and Software Technology*, 72, 48–67. <https://doi.org/10.1016/j.infsof.2015.11.009>
- Arhandi, P.P. (2016). Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan dengan Menggunakan Metode *BackEnd* dan Front End. *Jurnal Teknologi Informasi: Teori, Konsep, Dan Implementasi*, 7(1).
- Arliani, P. N. A., Sulastri, S., & Taftazani, B. M. (2015). Penerapan pengetahuan dan keterampilan pekerja sosial oleh relawan dalam pendampingan kepada anak penderita kanker. *Prosiding Penelitian dan Pengabdian kepada Masyarakat*, 2(1)
- Bradley, J. (2015). JSON Web Token(JWT), 1–30.
- CodeIgniter User Guide, [Online]. Available: <https://codeigniter.com/userguide3/index.html>.
- Cooksey, B. (2014). An Introduction to APIs. *Zapier, Inc.*
- Prabowo, D. (2015). Website E-Commerce Menggunakan Model View Controller (MVC) Dengan Framework Codeigniter Studi Kasus: Toko Miniatur. *Data Manajemen dan Teknologi Informasi (DASI)*, 16(1), 23.
- Elsa, R.G. (2017). *Perancangan Sistem Informasi Manajemen Kapasitas Logistik Penanggulangan Bencana Kota Padang* (Doctoral dissertation, Universitas Andalas).
- Harley Hahn's internet & web yellow pages (1st ed.). (1999). McGraw-Hill Osborne Media.
- Hayatul, (2020). Upaya Pengurangan Risiko Bencana Melalui Pelibatan Penyandang Disabilitas Indonesia. *Jurnal Manajemen Bencana*, 10..

- Hidayat, A., & Surarso, B. (2012). Penerapan Arsitektur Model View Controller (MVC) Dalam Rancang Bangun Sistem Kuis Online Adaptif. *Seminar Nasional Teknologi Informasi dan Komunikasi* (Vol. 2012, pp. 57-64).
- Huda, M., Yaqin, M. A., Kurniawan, R. F., Wahyu, M., & Choiri, F. (2022). Survei Strategi Pengujian Software Menggunakan Metode Systematic Literature Review. *ILKOMNIKA: Journal of Computer Science and Applied Informatics E*, 4(1), 116–133. <http://journal.unublitar.ac.id/ilkomnika>
- Iskandar, A., Kharisma, A. P., & Sutrisno, S. (2022). Implementasi Sistem Pengelolaan Donasi, Kegiatan, dan Relawan Komunitas Sosial Turun Tangan Malang berbasis Web dengan Framework Codeigniter. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(10), 5083- 5092.
- Kadir, A. (2003). Pengenalan Sistem Informasi. Yogyakarta : *Andi*.
- Kartadie, R., Asharudin, F., Suryanto, T., Hascarya, P. W., Mando, L. F., Rajab, A., dan Syam, A. (2013). Aplikasi Sistem informasi Manajemen Penanggulangan Pasca Bencana. *Semnasteknomedia Online*, 1(1), 17- 137.
- Ladjamudin, A. (2005). Analisis dan desain sistem informasi. *Yogyakarta: Graha Ilmu*, 1, 1-6.
- Lawrence, A. (2020). API: Pengertian, Fungsi, dan Cara Kerjanya. *NIAGAHOSTER Blog: <https://www.niagahoster.co.id/blog/api-adalah>*.
- Mahdia, F. 2013. Pemanfaatan Google Maps Api untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web (Studi Kasus: Badan Penanggulangan Bencana Daerah Kota Yogyakarta). *Yogyakarta: Universitas Ahmad Dahlan*.
- Mumbaikar, S., & Padiya, P. (2013). Web Services Based On SOAP and REST Principles. *International Journal of Scientific and Research Publications*, 1-4.
- Niqmah. 2018. Relawan PMI Meninggal Dunia saat Bertugas di Lombok, Ini Kronologi dan Tugas Sehari-hari Almarhum. <http://www.tribunnews.com/nasional/2018/08/25/>.
- Novianto, M. A., & Munir, S. (2022). Analisis dan Implementasi Restful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi. *Jurnal Informatika Terpadu*, 8(1), 47-61.

- Pandit, P., & Tahiliani, S. (2015). AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria. *International Journal of Computer Applications*, 120(10), 16–21. <https://doi.org/10.5120/21262-3533>
- Parabi, M. I., Kurniawan, D., & Prabowo, R. (2023). Peningkatan Keamanan Transmisi Data REST API Menggunakan Enkripsi SHA-512. *Jurnal Komputasi*, 11(1), 75-83.
- Postdot Technology. 2017. *Postman is the most complete API Development Environment*. [Online]. San Francisco. Tersedia: <https://www.getpostman.com/postman> [22 Agustus 2023].
- Pradhana, F. P. (2020). Pengembangan Aplikasi Android Sistem Informasi Manajemen Kebencanaan Pengurangan Resiko Bencana (SIMAK PRB) Meningkatkan Kesiapsiagaan Menghadapi Bencana di Kabupaten Blitar. *REVITALISASI: Jurnal Ilmu Manajemen*, 9(1), 1-11.
- Riana, D., Sanjaya, R., dan Kalsoem, O. (2018). Sistem Informasi Manajemen Laboratorium Patologi Anatomi Menggunakan Model MVC Berbasis Laravel Framework. *Konferensi Nasional Sistem Informasi (KNSI) 2018*.
- Rafiq, M., Ashraf, R., & Abid, H. (2020). Automated VS. Manual Testing: A Scenario Based Approach Towards Application Development. *Gyancity Journal of Electronics and Computer Science*, 5(1), 47–55. <https://doi.org/10.21058/gjecs.2020.51006>
- Siswanto, L. (2017). Sistem Informasi Manajemen Komando Tanggap Darurat Bencana Letusan Gunung Merapi. *Respati*, 7(19).
- Rahmatulloh, A., Sulastrri, H., dan Nugroho, R. 2018. Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512". *J. Nas. Tek. Elektro dan Teknologi. Inf.*, vol. 7, no. 2.
- What is the web? - Definition from Techopedia. (2012, September 21). *Techopedia.com*. <https://www.techopedia.com/definition/5613/web>