

**SISTEM INFORMASI DESA NEGARA TULANG BAWANG
KECAMATAN BUNGA MAYANG KABUPATEN LAMPUNG UTARA
BERBASIS *WEB***

(Skripsi)

Oleh

Rendi Adam



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2019**

**SISTEM INFORMASI DESA NEGARA TULANG BAWANG
KECAMATAN BUNGA MAYANG KABUPATEN LAMPUNG UTARA
BERBASIS WEB**

Rendi Adam

ABSTRAK

Pengelolaan data penduduk di Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara masih dilakukan dengan cara konvensional. Data-data tersebut disimpan dalam bentuk dokumen tercatat. Penelitian ini berupaya membangun sistem informasi desa agar pengelolaan data tersebut lebih mudah sehingga akses informasi lebih cepat dan akurat. Sistem ini dikembangkan dengan metode *waterfall*. Evaluasi sistem pada aspek tampilan oleh pengguna menunjukkan nilai rata-rata 78,38 % yang berarti sistem sudah baik. Hasil pengujian perangkat lunak menunjukkan bahwa fungsi-fungsi pada Sistem Informasi Desa Negara Tulang Bawang telah berjalan dengan baik sesuai requirement. Kesimpulannya adalah sistem ini layak digunakan untuk membantu kerja *staff* desa dalam pengelolaan data kependudukan.

Kata Kunci : Sistem Informasi, Desa, Waterfall

**SISTEM INFORMASI DESA NEGARA TULANG BAWANG
KECAMATAN BUNGA MAYANG KABUPATEN LAMPUNG UTARA
BERBASIS WEB**

Rendi Adam

ABSTRACT

Data management of population of Desa Tulang Bawang in Bunga Mayang district in Lampung Utara regency has been performed in old fashion. Data is written on the paper and stored in the filling cabinet. This research develop an information system to manage the data such that it can be easier and more accurate. This system was developed with the waterfall method. System evaluation on the aspect of display by the user shows an average value of 78.38% which means the system is good. The results of software testing indicate that the functions in the Country Information System of Tulang Bawang Village have been running well according to the requirements. The conclusion is that this system is feasible to be used to assist village staff in managing population data.

Keywords: Information System, Village, Waterfall.

**SISTEM INFORMASI DESA NEGARA TULANG BAWANG
KECAMATAN BUNGA MAYANG KABUPATEN LAMPUNG UTARA
BERBASIS *WEB***

Oleh

Rendi Adam

Skripsi

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

Pada

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



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2019**

Judul Skripsi : **SISTEM INFORMASI DESA NEGARA
TULANG BAWANG KECAMATAN
BUNGA MAYANG KABUPATEN
LAMPUNG UTARA BERBASIS *WEB***

Nama Mahasiswa : **Rendi Adam**

Nomor Pokok Mahasiswa : 1417051119

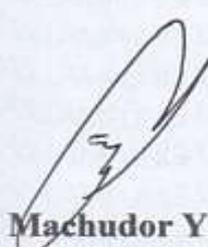
Program Studi : S1 Ilmu Komputer

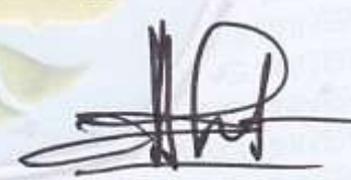
Fakultas : Matematika dan Ilmu Pengetahuan Alam



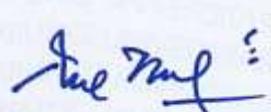
MENYETUJUI

1. Komisi Pembimbing


Ir. Machudor Yusman, M.Kom.
NIP. 19570330 198603 1 003


Dr. rer. nat. Akmal Junaidi, M.Sc.
NIP. 19710129 199702 1 001

2. Ketua Jurusan Ilmu Komputer


Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP. 19640616 198902 1 001

MENGESAHKAN

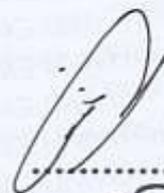
1. Tim Penguji

Ketua : Ir. Machudor Yusman, M.Kom.

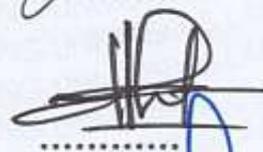
Sekretaris : Dr. rer. nat. Akmal Junaidi, M.Sc.

Penguji : Dwi Sakethi, M.Kom.

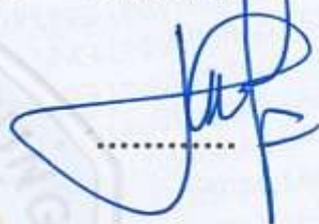
**Bukan
Pembimbing**



.....



.....



.....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Suratman, M.Sc.

NIP. 196406041990031002

Tanggal Lulus Ujian Skripsi: 14 juni 2019

PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul "**Sistem Informasi Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara Berbasis Web**" ini merupakan hasil karya sendiri dan bukan hasil karya orang lain. Semua hasil tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar akademik yang telah saya terima.

Bandar Lampung, 14 Juni 2019



Rendi Adam
1417051119

RIWAYAT HIDUP

Penulis dilahirkan di Kotabumi, Kabupaten Lampung Utara pada 5 September 1996. Merupakan anak kedua dari dua bersaudara, dari pasangan Bapak Rafdinal dan Ibu Mai Harti. Pendidikan Sekolah Dasar (SD) diselesaikan di SD PG Bunga Mayang pada tahun 2008, kemudian SMP Negeri 7 Kotabumi, Lampung Utara diselesaikan pada tahun 2011, kemudian melanjutkan sekolah di SMA Negeri 1 Kotabumi, Lampung Utara pada tahun 2014.

Pada tahun 2014 penulis terdaftar sebagai Mahasiswa S1 Jurusan Ilmu Komputer FMIPA Universitas Lampung. Selama menjadi mahasiswa, penulis aktif di beberapa organisasi kampus seperti diantaranya Himpunan Mahasiswa Ilmu Komputer (HIMAKOM), ROIS FMIPA Universitas Lampung, Badan Eksekutif Mahasiswa (BEM) FMIPA Universitas Lampung.

Pada Bulan Mei tahun 2017 penulis melakukan KP (Kerja Praktik) di UPT TIK PUSKOM Universitas Lampung, kemudian dilanjutkan Kuliah Kerja Nyata (KKN) pada bulan Juli 2017 di Desa Bandar Dalam, Kecamatan Sidomulyo, Kabupaten Lampung Selatan, Provinsi Lampung.

MOTTO

“Beradab Sebelum Berilmu, Beramal Dan Berkata”

(Rendi Adam)

PERSEMBAHAN

Aku persembahkan Skripsiku ini kepada kedua orang tua yang aku sayangi atas jasa-jasa mereka yang belum bisa aku balas. selalu menyemangatiku, memotivasiku untuk berjuang dalam menyelesaikan berbagai masalah ketika aku lemah dan tak berdaya. Begitu juga dengan kakak kandungku yang selalu bertukar ide-ide dengannya.

SANWACANA

Puji syukur penulis panjatkan kehadirat Allah Subhanahuwata'ala yang memberikan limpahan rahmat, karunia dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“SISTEM INFORMASI DESA NEGARA TULANG BAWANG KECAMATAN BUNGA MAYANG KABUPATEN LAMPUNG UTARA BERBASIS *WEB*”** adalah salah satu syarat untuk memperoleh gelar sarjana Ilmu Komputer di Universitas Lampung.

Penulis banyak mendapat bimbingan, nasehat, arahan, petunjuk dan saran dari berbagai pihak yang dapat memperluas pengetahuan penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik, maka dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc., selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
2. Bapak Didik Kurniawan, S.Si., MT., selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
3. Bapak Ir. Machudor Yusman, M.Kom., selaku Dosen Pembimbing I Jurusan Ilmu Komputer yang telah memberikan bimbingan dan motivasi dalam pembuatan skripsi ini.

4. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc., selaku Dosen Pembimbing II Jurusan Ilmu Komputer yang telah memberikan bimbingan, saran dan koreksi dalam pembuatan skripsi ini.
5. Bapak Dwi Sakethi, M.Kom., selaku Dosen Pembahas Jurusan Ilmu Komputer yang telah memberikan koreksi dan saran dalam pembuatan skripsi ini.
6. Kedua orang tua yang saya cintai dan sayangi. Saya memiliki ayah yang bernama Rafdinal dan Ibu yang bernama Mai Harti, dan kakak kandung saya bernama Rifaldi Andrian yang selalu memberikan semangat jiwa untuk menjadi pribadi yang tangguh dan memberikan materi serta doa dari kedua orang tua dan kakak saya. Terima kasih banyak atas kepercayaan selama masa perkuliahan dan saya belum bisa membalas jasa kedua orang tua saya dalam menyekolahkan saya di perguruan tinggi ini. Karena berkat pertolongan Allah melalui kedua orang tua, saya bisa seperti ini.
7. Teman-temanku angkatan Jurusan Ilmu Komputer 2014, Saiful Anwar, Rajes Ramli, Anis Purnomo, Anri Dicky, Ario Prabowo, Dicky Kurniawan AM, Amrullah Subekti, Dimas Aji Sukma, Mei Rusfandi, M Wildan Mubarak, Mahfud, Rahmat Purnama dan masih banyak lagi yang tidak bisa disebutkan satu persatu.
8. Teman-teman dan saudara-saudara tarbiyah dari Rois FMIPA Universitas Lampung, Kak Alaudin Al Ayubi, Kak Wahyudi, Kak Agum Aldino, Kak Suyitno, Kak Zubaidi, Kak Hammid Ammar Mansur Rijal, Kak Efrizal, Kak M. Aiman A, Kak Pranoto, Kak

Muzahid Ansori, Kak Rosyid Rosidi, Kak Herullah, Zarkoni, Badzlan Hasbi, Zhofar Murry Setiawan, Ganjar Andhilangi, M. Firza Ersa, Abdul Qodir, Anis Purnomo, Ahmad Aziz A.

9. Teman-teman Badan Eksekutif Mahasiswa (BEM) dan Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom).
10. Semua pihak yang turut membantu dalam penyusunan skripsi ini.

Terima Kasih, semoga skripsi ini dapat bermanfaat bagi banyak pihak.

Bandar Lampung, 14 Juni 2019

Penulis

Rendi Adam

DAFTAR ISI

Halaman

ABSTRAK	i
ABSTRACT	ii
PERSETUJUAN.....	iii
PENGESAHAN.....	iv
PERNYATAAN.....	v
RIWAYAT HIDUP.....	vi
MOTTO.....	vii
PERSEMBAHAN.....	viii
SANWACANA.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Pengertian Sistem	7
2.2 Pengembangan Sistem.....	7
2.3 Konsep Dasar Sistem.....	7
2.3.1 Pengertian Sistem.....	8
2.3.2 Karakteristik Sistem.....	8
2.3.3 Siklus Informasi	11
2.3.4 Kualitas informasi	12
2.3.5 Nilai Informasi	13
2.4 <i>E-Government</i>	14
2.5 <i>Government-to-Citizen (G2C)</i>	14
2.6 Desa	14
2.7 Pelayanan Pemerintahan Desa.....	15
2.8 Masalah Pelayanan Publik Pemerintahan Desa.....	16
2.9 <i>E-Governenment</i> dalam Pelayanan Publik	17

2.10	Pengertian Data	17
2.11	Pengolahan Data.....	18
2.12	<i>Model Waterfall</i>	19
2.13	Pemrograman <i>PHP</i>	23
	2.13.1 Sejarah <i>PHP</i>	23
	2.13.2 Kelebihan <i>PHP</i> Dari Bahasa Pemrograman Lain.....	24
2.14	<i>XAMPP</i>	25
	2.14.1 Bagian-Bagian Penting <i>XAMPP</i>	25
2.15	<i>MYSQL</i>	26
	2.15.1 Sejarah <i>MySQL</i>	28
2.16	<i>UML (Unified Modelling Language)</i>	29
	2.16.1 Konsep pemodelan menggunakan <i>UML</i>	30
	2.16.2 Notasi Dalam <i>UML</i>	31
2.17	Diagram Kelas (<i>Class Diagram</i>).....	37
2.18	Diagram <i>UseCase (UseCase Diagram)</i>	38

BAB III METODE PENELITIAN 40

3.1	Tempat Dan Waktu Penelitian	40
3.2	Perangkat Penelitian	40
3.3	Analisa Kebutuhan	41
3.4	Tahapan Penelitian	41
	3.4.1 Tahap Perumusan Masalah	42
	3.4.2 Tahap Pengumpulan Data	42
3.5	UseCase Diagram	43
3.6	Class Diagram	44
3.7	<i>Activity Diagram</i>	45
3.8	<i>Sequence Diagram</i>	46
3.9	Rancangan Sistem	47

BAB IV HASIL DAN PEMBAHASAN 54

4.1	Hasil.....	54
4.2	Implementasi Sistem	55
4.3	Deskripsi Program.....	55
	4.3.1 Tampilan Halaman <i>Login</i>	55
	4.3.2 Tampilan Halaman Beranda Dan Menu Utama.....	56
	4.3.3 Tampilan Menu Profile Desa	57
	4.3.4 Tampilan Halaman Tentang.....	58
	4.3.5 Tampilan Menu Data Admin	59
	4.3.6 Tampilan Menu Data Penduduk	60
	4.3.7 Tampilan Menu Data Desa	61
	4.3.8 Tampilan Menu Tipologi Desa	62
	4.3.9 Tampilan Menu Tingkat Perkembangan Desa.....	63
	4.3.10 Tampilan Menu Info Wilayah.....	64
	4.3.11 Tampilan Menu Orbitasi	65
	4.3.12 Tampilan Menu Jumlah Tanah Bersertifikat	66
	4.3.13 Tampilan Menu Luas Tanah Kas Desa.....	67

4.3.14 Tampilan Menu Jumlah Penduduk	68
4.3.15 Tampilan Menu Mata Pencaharian	69
4.3.16 Tampilan Menu Tingkat Pendidikan Masyarakat.....	70
4.3.17 Tampilan Menu Jumlah Penduduk Miskin	71
4.3.18 Tampilan Menu UMR Kabupaten	72
4.3.19 Tampilan Menu Sarana Dan Prasarana.....	73
4.4 Pengujian Fungsional	74
4.4 Pengujian Kepuasan Pelanggan.....	82
4.5 Analisis Hasil Pengujian	88
BAB V SIMPULAN DAN SARAN.....	90
5.1 Simpulan.....	90
5.2 Saran	90
DAFTAR PUSTAKA	91
LAMPIRAN.....	93

DAFTAR TABEL

Tabel	Halaman
2.1 Simbol <i>Class Diagram</i>	37
2.2 Simbol <i>Usecase Diagram</i>	38
4.2 Tabel Skenario Pengujian <i>Security</i>	75
4.3 Tabel Skenario Pengujian <i>Database</i>	77
4.4 Tabel Skenario Pengujian <i>User Interface</i>	80
4.5 Tabel Daftar Responden.....	83
4.6 Hasil Penilaian Sistem Informasi Desa Negara Tulang Bawang.....	85
4.7 Tabel Interval Kategori Skala Likert.....	87

DAFTAR GAMBAR

Gambar	Halaman
2.1 Karakteristik Suatu Sistem.....	11
2.2 Siklus Informasi.....	12
2.3 Pilar Kualitas Informasi.....	13
2.4 Model <i>Waterfall</i>	20
2.5 <i>XAMPP Control Panel</i>	26
2.6 Notasi <i>Actor</i>	32
2.7 Notasi <i>Usecase</i>	32
2.8 Notasi <i>Class</i>	33
2.9 Notasi <i>Interface</i>	34
2.10 Notasi <i>Interaction</i>	34
2.11 Notasi <i>Note</i>	34
2.12 Notasi <i>Depedency</i>	35
2.13 Notasi <i>Association</i>	35
2.14 Notasi <i>Generalization</i>	36
2.15 Notasi <i>Realization</i>	36
3.1 Diagram Alir Penelitian.....	42
3.2 <i>Usecase Diagram</i>	44
3.3 <i>Class Diagram</i>	45
3.4 <i>Activity Diagram</i>	46
3.5 <i>Sequence Diagram</i>	47
3.6 Tampilan <i>Form Login</i>	48
3.7 Tampilan Beranda.....	48
3.8 Tampilan Profil Desa.....	49
3.9 Tampilan Data Penduduk.....	49

3.10	Tampilan Tambah Data.....	50
3.11	Tampilan Ubah Data	51
3.12	Tampilan Hapus Data.....	51
3.13	Tampilan Import Data	52
3.14	Tampilan Laporan Data penduduk + <i>Export Data to Excel</i>	52
3.15	Tampilan Tentang Desa	53
4.1	Tampilan Halaman <i>Login</i>	56
4.2	Tampilan Halaman Beranda Dan Menu Utama	57
4.3	Tampilan Menu Profil Desa	58
4.4	Tampilan Maps Pada Menu Tentang	59
4.5	Tampilan Menu Data Admin	60
4.6	Tampilan Menu Data Penduduk.....	61
4.7	Tampilan Menu Data Desa.....	62
4.8	Tampilan Menu Tipologi Desa	63
4.9	Tampilan Menu Tingkat Perkembangan Desa.....	64
4.10	Tampilan Menu Info Wilayah.....	65
4.11	Tampilan Menu Orbitrase	66
4.12	Tampilan Menu Jumlah Tanah Bersertifikat.....	67
4.13	Tampilan Menu Luas Tanah Kas Desa	68
4.14	Tampilan Menu Jumlah Penduduk.....	69
4.15	Tampilan Menu Mata Pencaharian	70
4.16	Tampilan Menu Tingkat Pendidikan Masyarakat	71
4.17	Tampilan Menu Jumlah Penduduk Miskin	72
4.18	Tampilan Menu UMR Kabupaten.....	73
4.19	Tampilan Menu Sarana Dan Prasarana	74

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kehadiran teknologi informasi bertujuan untuk mempermudah dalam pengerahan sumber daya, berbagi informasi dan mengkoordinasikan aktifitas masyarakat. Pemanfaatan teknologi informasi yang saat ini berkembang pesat semakin meluas seiring dengan meningkatnya kebutuhan akan ketersediaan informasi yang akurat dan cepat. Kemajuan teknologi informasi saat ini ialah pemanfaatan jaringan internet yang memungkinkan setiap orang dapat mengakses atau memperoleh data-data yang tersedia secara bersama-sama melalui jaringan yang saling terhubung. Dengan kemudahan dalam penggunaannya, penerapan teknologi berbasis sistem informasi menjadi salah satu pilihan alternatif untuk menghadapi permasalahan-permasalahan yang timbul akibat proses birokrasi pemerintahan yang semakin dinamis.

Informasi tentang desa merupakan hal yang penting untuk memberikan informasi kepada masyarakat. Bagi masyarakat, biasanya informasi mengenai desa didapatkan pada saat ada kegiatan desa. Namun walaupun telah mengikuti kegiatan tersebut, tetapi masih banyak masyarakat yang belum mengetahui secara detail mengenai desanya sendiri. Harapan dari Sistem informasi desa ini bertujuan untuk membantu pemerintah desa dalam mendokumentasikan data-data milik desa

berguna dalam pencarian data dan mempercepat pelayanan kepada penduduk. Penyebaran informasi desa dan potensi yang dimiliki oleh suatu desa dengan menggunakan teknologi yang sudah berkembang meningkatkan kualitas dan ketepatan data yang tersedia. Era teknologi dan globalisasi juga semakin mendorong timbulnya kebutuhan informasi yang cepat dan tepat. Hal itu dirasakan sangat vital bagi masyarakat saat ini di berbagai bidang.

Dengan dilakukannya pendataan penduduk dari tingkat pekon (kelurahan) yang memiliki manfaat untuk memudahkan kerja dari staff kelurahan maupun staff dari kecamatan Bunga Mayang.

Salah satu kegiatan yang dilakukan Kecamatan Bunga Mayang adalah pencatatan data penduduk yang dimulai dari tingkat kelurahan sampai dengan tingkat kecamatan dan diteruskan ke tingkat Kabupaten serta Provinsi. Dalam pendataan penduduk kecamatan Bunga Mayang dimulai berdasarkan peristiwa penting yaitu seperti pendataan penduduk berdasarkan kartu keluarga (KK), agama, jenis kelamin, pekerjaan, pendidikan, mendata jumlah lurah/desa, mendata jumlah sekolah.

Pendataan penduduk dimulai dari tingkat kelurahan. Staff kelurahan melakukan pendataan dengan cara mendatangi rumah rumah warga kelurahan secara langsung untuk mendapatkan data penduduk yang akurat. Dari pendataan tersebut dapat diambil statistik kependudukan seperti data kartu keluarga (KK), pendidikan, agama, jenis kelamin pekerjaan serta komposisi dari usia penduduk yang bertempat tinggal di desa tersebut.

Pengolahan data kependudukan dilakukan untuk laporan kependudukan Kecamatan Bunga Mayang serta pengendalian jumlah dan pertumbuhan penduduk serta pemerataan penyebaran penduduk. Selain itu, pengolahan data kependudukan dapat digunakan pencapaian kualitas sumber daya manusia dengan berbagai masalah seperti kemiskinan, kesehatan dan pengangguran.

Berdasarkan penjelasan dari beberapa kendala sistem pengolahan data kependudukan yang digunakan di Kecamatan Bunga Mayang maka perlu dikembangkan “Sistem Informasi Kecamatan Bunga Mayang Berbasis *Website*” dengan fungsi utama mengelola data kependudukan dengan lebih cepat dan menampilkan statistik data kependudukan di Kecamatan Bunga Mayang.

Kabupaten adalah pembagian wilayah administratif di Indonesia di bawah provinsi, yang dipimpin oleh seorang Bupati. Kabupaten maupun kota merupakan daerah otonom yang diberi wewenang mengatur dan mengurus urusan pemerintahannya sendiri. Sedangkan Kecamatan adalah pembagian wilayah administratif di Indonesia di bawah Kabupaten atau Kota. Kecamatan terdiri atas desa-desa atau kelurahan-kelurahan. Kecamatan atau sebutan lain adalah wilayah kerja camat sebagai perangkat daerah Kabupaten/Kota. (Pemerintah, 2005)

Kabupaten Lampung Utara yang mempunyai 23 Kecamatan, salah satunya adalah Kecamatan Bunga Mayang. Kecamatan Bunga Mayang terdiri atas 11 Desa yaitu Bunga Maju, Handuyang Ratu, Isorejo, Kota Napal, Mulyo Rejo I, Mulyo Rejo II, Negara Tulang Bawang, Negara Tulang Bawang Baru, Sukadana Ilir, Sukadana Udik, Tanah Abang. Desa adalah kesatuan masyarakat hukum yang memiliki batas-batas wilayah yang berwenang untuk mengatur dan mengurus

kepentingan masyarakat setempat, berdasarkan asal-usul dan adat istiadat setempat (Peraturan Pemerintah Nomor 57 Tahun 2005 tentang Desa).

Desa Negara Tulang Bawang mempunyai 7 Dusun diantaranya Dusun 1, Dusun 2, Dusun 3, Dusun 4, Dusun 5, Dusun 6 dan Dusun 7. Salah satu Visi dan Misi Desa Negara Tulang Bawang yaitu:

Visi:

Terwujudnya Desa Negara Tulang Bawang Sejahtera, Aman dan Makmur.

Misi:

1. Mewujudkan Masyarakat Desa Negara Tulang Bawang yang Mandiri uraian :
 - a. Meningkatkan kemandirian sumber daya manusia
 - b. Meningkatkan pangan, papan dan sandang masyarakat Desa Negara Tulang Bawang.
2. Mewujudkan Masyarakat Desa Negara Tulang Bawang yang Sejahtera
Uraian:
 - a. Pembangunan
 - 1) Meningkatkan pembangunan infrastruktur Desa
 - 2) Meningkatkan sumber daya alam yang ada
 - 3) Meningkatkan peran aktif BPD, LPMD, RT/RW, dan tokoh masyarakat dalam pembangunan Desa
 - 4) Meningkatkan peran serta masyarakat dalam berswadaya membangun Desa
 - b. Pemerintahan
 - 1) Menciptakan Sistem Pemerintahan yang Baik dan Demokratis.

c. Kemasyarakatan

- 1) Peningkatan dan pengembangan usaha kecil dan menengah.
- 2) Menjaga dan memelihara ketentraman, ketertiban, dan kerukunan warga.
- 3) Mewujudkan keluarga sehat sejahtera melalui peran aktif ibu-ibu PKK, Posyandu, dan organisasi lainnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, dapat dirumuskan masalah dalam pengembangan sistem yaitu :

1. Bagaimana mengembangkan Sistem Informasi Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara yang bisa membantu mempermudah *staff* Desa Negara Tulang Bawang dalam mengolah data kependudukan.
2. Bagaimana membuat Sistem Informasi Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara yang mampu menampilkan data penduduk berdasarkan informasi yang telah disebutkan.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Sistem Informasi Berbasis *Web*.
2. Sistem menampilkan data kependudukan serta pengelolaannya dengan menambah, menghapus dan mengubah data tersebut.
3. Sistem hanya mengelola data kependudukan saja.

1.4 Tujuan Penelitian

1. Mengembangkan sistem berbasis *Web*.
2. Membuat sistem yang memiliki kemampuan untuk mengelola data kependudukan.

1.5 Manfaat Penelitian

1. Membantu kerja *staff* desa dalam pengelolaan data kependudukan.
2. Sangat penting untuk mendukung perkembangan dan kemajuan di desa Negara Tulang Bawang.
3. Mempermudah pendataan dengan sistem menjadi lebih cepat dan akurat.

BAB II TINJAUAN PUSTAKA

2.1 Pengertian Sistem

Menurut Jerry Fitz Gerald, “sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”. (Jogiyanto, 2005:1)

2.2 Pengembangan Sistem

Pengembangan sistem (*systems development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada.

2.3 Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Berikut akan dijelaskan lebih detail tentang konsep-konsep dasar dari sistem, mulai dari pengertian dasar sistem menurut beberapa pakar, serta terdapat beberapa komponen yang merupakan bagian dari sifat atau karakteristik dari sebuah sistem.

2.3.1 Pengertian Sistem

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”. Pendekatan sistem yang merupakan jaringan kerja dari prosedur, yang telah dijabarkan diatas lebih menekankan pada urutan-urutan operasi di dalam sebuah sistem, yang berarti suatu sistem merupakan jaringan kerja dari berbagai operasi yang berurutan, berhubungan, berkumpul bersama dan bekerjasama demi penyelesaian suatu tujuan tertentu, sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem. Sebuah sistem dapat terdiri dari sistem-sistem bagian (subsistem) masing-masing dapat terdiri dari subsistem-subsistem yang lebih kecil lagi atau terdiri dari komponen-komponen yang saling berinteraksi dan saling berhubungan membentuk suatu kesatuan sehingga tujuan atau sasaran sistem dapat tercapai.

2.3.2 Karakteristik Sistem

Suatu sistem dapat mempunyai karakteristik atau sifat yang tertentu yaitu mempunyai komponen-komponen (*component*), batasan sistem (*boundary*), lingkaran luar system (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah sistem (*interface*), dan sasaran (*objective*). Karakteristik atau sifat - sifat dari sistem di bawah ini sebagai berikut.

a. Komponen

Suatu sistem terdiri dari sejumlah komponen yang saling bekerja sama atau sama lain membentuk suatu kesatuan. Setiap sistem tidak peduli betapapun

kecilnya selalu mengandung komponen - komponen atau subsistem- subsistem dimana setiap subsistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

b. Batasan Sistem

Batasan sistem merupakan daerah yang membatasi diantara suatu sistem dengan system lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang menunjukkan ruang lingkup sistem itu.

c. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga dapat bersifat merugikan sistem.

d. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya dengan melalui penghubung. Dengan adanya penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk suatu kesatuan.

e. Masukan Sistem

Masukan adalah suatu yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan, masukan signal. Masukan perawatan adalah

sesuatu yang dimasukan supaya sistem tersebut dapat beroperasi sedangkan masukan signal adalah sesuatu yang diproses untuk didapatkannya suatu keluaran.

f. Keluaran Sistem

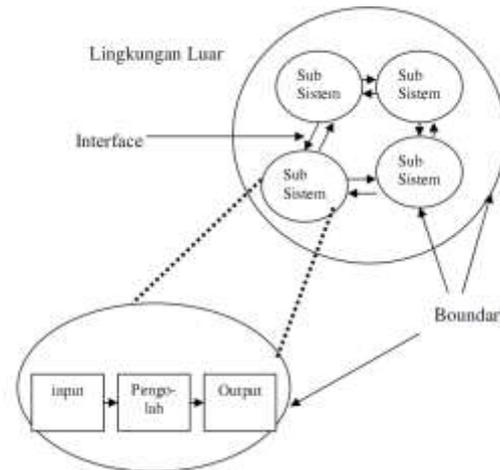
Keluaran sistem adalah hasil dari masukan yang diolah dan diklasifikasikan menjadi yang berguna. Keluaran dapat berupa masukan untuk subsistem lainnya.

g. Pengolahan Sistem

Sesuatu dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahannya. Pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran Sistem

Suatu sistem pasti memiliki tujuan/sasaran, apabila suatu sistem tidak mempunyai sasaran maka operasi sistem tidak akan adagunanya. Sasaran dari sistem dapat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan oleh sistem. Suatu sistem dikatakan berhasil bila sasaran yang diharapkan mengenai tujuannya. (Jogiyanto, 2005:3)
Karakteristik dari suatu sistem ditunjukkan pada Gambar 2.1.



Gambar 2.1 Karakteristik Suatu Sistem.

i. Keluaran Sistem

Keluaran sistem adalah hasil dari masukan yang diolah dan diklasifikasikan menjadi yang berguna. Keluaran dapat berupa masukan untuk subsistem lainnya.

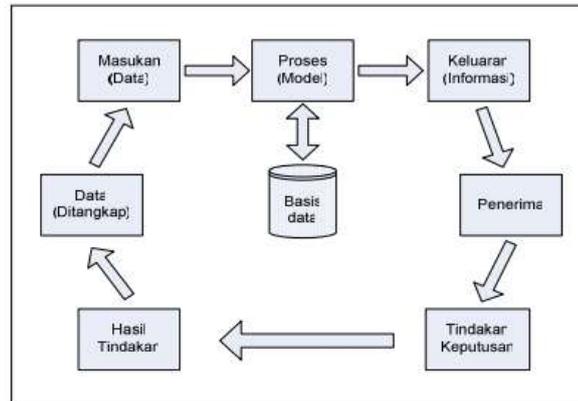
j. Pengolahan Sistem

Sesuatu dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahannya. Pengolah yang merubah masukan menjadi keluaran.

2.3.3 Siklus Informasi

Siklus Informasi dapat diartikan sebagai sebuah penggambaran pengolahan data menjadi informasi dan pemakaian informasi untuk pengambilan keputusan, hingga akhirnya dari tindakan hasil pengambilan keputusan tersebut dihasilkan data kembali. Data diolah melalui suatu model tertentu menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan

ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus informasi yang juga disebut sebagai siklus pengolahan data, ditunjukkan pada Gambar 2.2.



Gambar 2.2 Siklus Informasi.

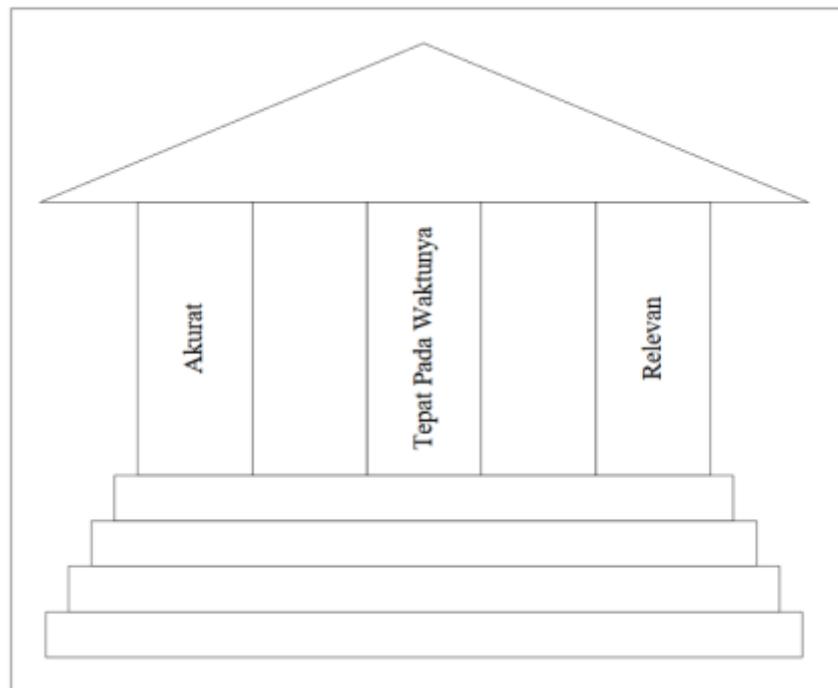
(Sumber: Abdul Kadir, Pengenalan Sistem Informasi, 2003)

2.3.4 Kualitas informasi

Kualitas dari suatu informasi (*quality of information*) tergantung dari tiga hal, yaitu:

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksudnya.
2. Tepat pada waktunya, berarti informasi yang datang ke penerima tidak boleh terlambat. Informasi yang sudah usang tidak ada nilainya lagi. Karena informasi merupakan landasan dalam pengambilan keputusan.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

Pilar kualitas informasi ditunjukkan pada Gambar 2.3.



Gambar 2.1 Pilar Kualitas Informasi.

(Sumber: Jogiyanto. HM, 2005, *Analiss dan Desain*, Andi Offset, Yogyakarta)

2.3.5 Nilai Informasi

Nilai dari informasi (*value of sistem*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan dalam suatu sistem dimungkinkan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah tertentu dengan biaya untuk memperolehnya, karena sebagian besar informasi dinikmati tidak hanya satu pihak di dalam perusahaan. Lebih lanjut sebagian informasi tidak dapat ditaksir keuntungannya dengan nilai uang, tetapi dapat ditaksir nilai efektifitasnya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness* atau *cost benefit*.

2.4 E-Government

E-government bukanlah sebuah perubahan secara fundamental yang berjangka pendek pada pemerintahan dan pemerintahan dan bukan pula sebagai awal dari permulaan era industriliasi. Artinya adalah bahwa *E-government* merupakan sebuah modernisasi pemanfaatan teknologi yang secara garis bukan sebuah perubahan yang sangat mendasar didalam sebuah tata pemerintahan yang dipastikan akan berjalan dalam jangka panjang dan bukan pula membuktikan bahwa ini merupakan awal dari sebuah proses pertumbuhan dan perubahan sosial. (Caldow, 2001)

2.5 Government-to-Citizen (G2C)

Government-to-Citizen (G2C) adalah penyampaian layanan publik dan informasi satu arah oleh pemerintah ke masyarakat, Memungkinkan pertukaran informasi dan komunikasi antara masyarakat dan pemerintah. Juga bertujuan untuk mendekatkan pemerintah dengan masyarakat melalui akses yang beragam agar masyarakat dapat dengan mudah menjangkau pemerintahnya untuk pemenuhan berbagai kebutuhan pelayanan. (Pascual, 2003)

2.6 Desa

Desa adalah suatu kesatuan masyarakat hukum berdasarkan adat dan hukum adat yang menetap dalam suatu wilayah tertentu batas-batasnya, memiliki ikatan lahir batin yang sangat kuat, baik karena keturunan maupun kesamaan kepentingan politik, ekonomi, sosial, dan keamanan, memiliki susunan pengurus yang dipilih bersama; memiliki kekayaan dalam jumlah tertentu dan berhak menyelenggarakan

rumah tangganya sendiri. Sedangkan menurut Undang-Undang RI No 6 Tahun 2014 Tentang desa, desa adalah kesatuan masyarakat hukum yang memiliki batas wilayah yang berwenang untuk mengatur dan mengurus urusan pemerintahan, kepentingan masyarakat setempat berdasarkan prakarsa masyarakat, hak asal usul, dan/atau hak tradisional yang diakui dan dihormati dalam sistem pemerintahan Negara Kesatuan Republik Indonesia. (Wasistiono dan Tahir 2006)

2.7 Pelayanan Pemerintahan Desa

Pengaturan tata kelola desa yang terbaru mengikuti Undang-Undang no. 6 tahun 2014 tentang desa. Pengaturan desa dalam undang-undang ini bertujuan untuk memberikan pengakuan dan penghormatan atas desa yang sudah ada dengan keberagamannya sebelum dan sesudah terbentuknya Negara Kesatuan Republik Indonesia; memberikan kejelasan status dan kepastian hukum atas desa dalam sistem ketatanegaraan Republik Indonesia demi mewujudkan keadilan bagi seluruh rakyat Indonesia; melestarikan dan memajukan adat, tradisi, dan budaya masyarakat desa; mendorong prakarsa, gerakan, dan partisipasi masyarakat desa untuk pengembangan potensi dan Aset desa guna kesejahteraan bersama; membentuk Pemerintahan desa yang profesional, efisien dan efektif, terbuka, serta bertanggung jawab; meningkatkan pelayanan publik bagi warga masyarakat desa guna mempercepat perwujudan kesejahteraan umum; meningkatkan ketahanan sosial budaya masyarakat desa guna mewujudkan masyarakat desa yang mampu memelihara kesatuan sosial sebagai bagian dari ketahanan nasional; memajukan perekonomian masyarakat desa serta mengatasi kesenjangan pembangunan nasional; dan memperkuat masyarakat desa sebagai subjek pembangunan. Dalam

pembangunan nasional, desa memegang peranan yang sangat penting, sebab desa merupakan struktur pemerintahan terendah dari sistem pemerintahan Indonesia. Setiap jenis kebijakan pembangunan nasional pasti bermuara pada pembangunan desa sebab pembangunan Indonesia tidak akan ada artinya tanpa membangun desa, dan bisa dikatakan bahwa hari depan Indonesia terletak dan tergantung dari berhasilnya masyarakat membangun desa. Sehingga dengan semangat desentralisasi dalam otonomi daerah ini masyarakat haruslah dilibatkan atau diberdayakan dalam pembangunan desanya. Sebab disadari atau tidak bahwa pembangunan desa telah banyak dilakukan sejak dari dahulu hingga sekarang, tetapi secara umum hasilnya belum memuaskan terhadap peningkatan kesejahteraan masyarakat pedesaan. Sebagai miniatur negara Indonesia, desa menjadi arena politik paling dekat bagi relasi antara masyarakat dengan pemegang kekuasaan (perangkat desa). Di satu sisi, para perangkat desa menjadi bagian dari birokrasi negara yang mempunyai daftar tugas kenegaraan, yakni menjalankan birokratisasi di level desa, melaksanakan program-program pembangunan, memberikan pelayanan administratif kepada masyarakat. Tugas penting pemerintah desa adalah memberi pelayanan administratif (surat-menyurat) kepada warga. (Depdagri, 2007:27)

2.8 Masalah Pelayanan Publik Pemerintahan Desa

Secara umum telah diketahui masalah-masalah yang dihadapi di kelurahan, baik yang bersumber secara internal maupun yang eksternal, seperti semakin pesatnya kegiatan pembangunan yang hasil-hasilnya telah kita rasakan saat ini, namun demikian masih dapat ditemukan pula dampak yang dapat menimbulkan masalah

yang baru. Untuk mengatasi masalah tersebut, maka diperlukan kemampuan di bidang perencanaan pembangunan dan pemberian pelayanan yang baik dan berkualitas oleh para aparatur desa/kelurahan kepada masyarakat sehingga permasalahan yang kompleks dan rumit dapat diatasi. Permasalahan-permasalahan yang dihadapi Aparat Birokrasi desa dalam mengoptimalkan kegiatan Pemerintahan desa meliputi permasalahan internal yang berupa ketatalaksanaan, sumber daya manusia atau kompetensi Aparat Pemerintah desa, ketatalaksanaan, penggunaan teknologi administrasi yang masih kurang, dan manajemen birokrasi itu sendiri. Sedangkan masalah eksternal berupa dinamika masyarakat dan tumbuh kembangnya masalah yang dihadapi masyarakat. (Lulita, 2011)

2.9 *E-Government* dalam Pelayanan Publik

Di antara banyak harapan dari revolusi Informasi Komunikasi Teknologi (ICT) adalah potensi untuk memodernisasi organisasi pemerintah, memperkuat operasi mereka dan membuat mereka lebih responsif terhadap kebutuhan warganya. Banyak negara telah memperkenalkan program *E-Government* berbasis TIK dan menggunakannya untuk mengubah dimensi beberapa operasi pemerintahan, untuk menciptakan pemerintahan yang lebih mudah diakses, transparan, efektif, dan akuntabel. (Al-Khouri, 2011)

2.10 Pengertian Data

Data dapat didefinisikan sebagai deskripsi dari suatu dan kejadian yang kita hadapi. Data dapat berupa catatan-catatan dalam kertas, buku, atau tersimpan

sebagai *file* dalam *database*. Data sebagai bahan dalam suatu proses pengolahan data. Oleh karena itu, suatu data belum dapat berbicara banyak sebelum diolah lebih lanjut. (Al-Bahra Bin Ladjamudin, 2005, Hal:8)

Proses pengolahan data terbagi menjadi tiga tahapan, yang disebut dengan siklus pengolahan data (*Data Processing Cycle*) yaitu :

1. Pada Tahapan *Input*

Yaitu dilakukan proses pemasukan data ke dalam computer lewat media input (*Input Devices*).

2. Pada Tahapan *Processing*

Yaitu dilakukan proses pengolahan data yang sudah dimasukkan, yang dilakukan oleh alat pemroses (*Process Devices*) yang dapat berupa proses perhitungan, perbandingan, pengendalian, atau pencarian *distorage*.

3. Pada Tahapan *Output*

Yaitu dilakukan proses menghasilkan *output* dari hasil pengolahan data ke alat output (*Output Devices*) yaitu berupa informasi.

2.11 Pengolahan Data

Pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan. Operasi yang dilakukan dalam pengolahan data adalah sebagai berikut :

1. Data *input*

- a. *Recording*, transaksi data ke sebuah pengolahan data medium.
- b. *Coding*, transaksi data ke dalam bentuk lain.
- c. *Storing*, data atau informasi untuk pengambilan keputusan.

2. *Data transformation*
 - a. *Calculating*, operasi aritmatik terhadap data *field*.
 - b. *Summarizing*, proses akumulasi beberapa data, misalnya menjumlah jumlah
 - c. jam kerja setiap hari dalam seminggu menjadi nilai total jam kerja perminggu.

3. *Classifying*, data *group-group* tertentu :
 - a. *Categorizing*, data ke dalam *group* berdasar karakteristik tertentu.
 - b. *Sorting*, data ke dalam bentuk yang berurutan.
 - c. *Merging*, untuk dua atau lebih *set* data berdasar kriteria tertentu.
 - d. *Matching*, data berdasar keinginan pengguna terhadap *group* data.

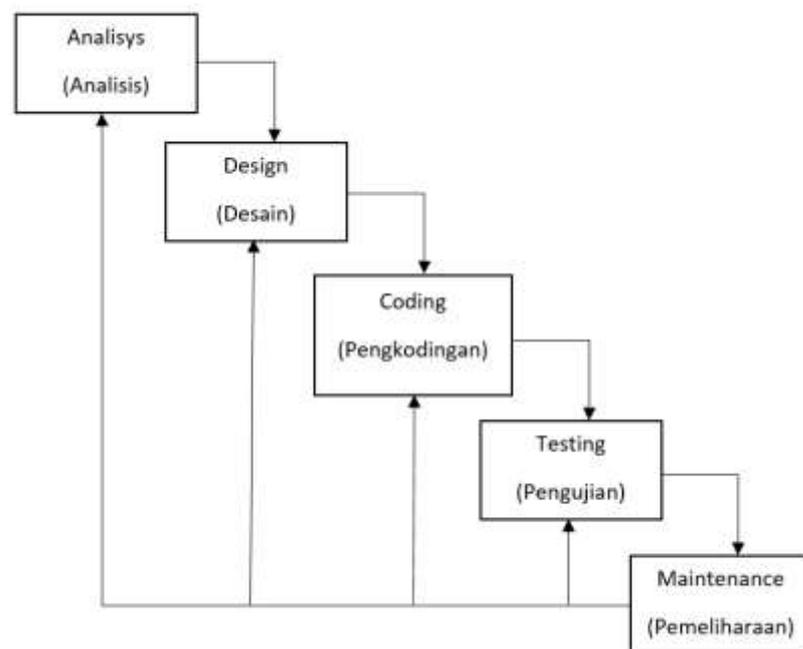
4. *Information output*.
 - a. *Displaying result*, menampilkan informasi yang dibutuhkan pemakai melalui monitor atau cetakan.
 - b. *Reproducing*, penyimpanan data yang digunakan untuk pemakai lain yang membutuhkan.
 - c. *Telecommunicating*, penyimpanan data secara elektronik melalui saluran komunikasi.

2.12 Model Waterfall

Model waterfall merupakan metode perancangan yang umumnya masih menggunakan urutan yang bertahap dan teratur. Pada metode *waterfall* jika tahap-tahap pada *waterfall* belum selesai dilakukan maka tahap selanjutnya tidak dapat

berjalan. *waterfall* dan *iterative*, kedua istilah tersebut sering disalahartikan, khususnya karena *iterative* sering dipandang sebagai proses yang kuno. Hasilnya adalah banyak proyek yang mengaku menggunakan *iterative* padahal sebenarnya menggunakan *waterfall*. Gaya *waterfall* adalah memecah sebuah proyek berdasarkan aktivitas. Untuk membuat perangkat lunak, harus melakukan kegiatan tertentu, yaitu analisis persyaratan, *design*, coding, dan testing. Dalam pengembangan *waterfall*, biasanya terdapat beberapa alih fungsi formal antar fase, tetapi sering kali dituntut untuk kembali fase sebelumnya.

Tahapan model *waterfall* ditampilkan pada Gambar 2.4.



Gambar 2.4 Model *Waterfall*.

Dalam pengembangan metode *waterfall* memiliki beberapa tahapan yang urut : requirement (analisis kebutuhan), design sistem (sistem desain), *coding* (pengkodean), *testing* (pengujian), *maintenance* (pemeliharaan).

1. *Requiemment* (analisis kebutuhan)

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bias melakukan sebuah penelitian dan wawancara. Seseorang sistem analisis perlu menggali informasi sebanyak-banyaknya dari user sehingga tercipta sebuah sistem computer yang bias melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini menghasilkan dokumen oleh user requiemment atau bias dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen ini menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman.

2. *Design System* (desain sistem)

Proses *design* menterjemahkan syarat kebutuhan kesebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini menghasilkan dokumen yang disebut *software requirement*. Dokumen ini digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

3. *Coding* (penulisan sinkode program)

Coding merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang menterjemahkan transaksi yang diminta oleh *user*. Tahapan ini merupakan tahapan secara nyata dalam

mengerjakan suatu sistem. Dalam artian penggunaan komputer bahkan dimaksimalkan dalam tahapan ini.

4. *Testing* (penerapan / pengujian program)

Tahapan ini bisa dikatakan selesai dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi yang digunakan oleh *user*.

5. *Maintenace* (pemeliharaan)

Perangkat lunak yang sudah disampaikan kepada pelanggan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau system operasi baru) atau karena pelanggan membutuhkan perkembangan fungsional.

Kelebihan *Waterfall model* :

1. Mudah diaplikasikan.
2. Memberikan *teample* tentang metode analisis, desain, pengkodean, pengujian, dan pemeliharaan.
3. Cocok digunakan untuk produk *software* yang sudah jelas kebutuhannya diawal. Sehingga minim kesalahannya.

Kekurangan *Waterfall model*:

1. *Waterfall model* bersifat kaku sehingga sulit untuk melakukan perubahan pada sistem perangkat lunak.
2. Terjadinya pembagian proyek menjadi tahap-tahap yang tidak fleksibel. Karena komitmen harus dilakukan pada tahap awal proses.
3. *Customer* harus sabar untuk menanti proyek selesai, karena dikerjakan tahap per tahap, menyelesaikan tahap awal baru bisa ketahap selanjutnya.

4. Perubahan ditengah-tengah pekerjaan produk akan membuat bingung *team work* yang sedang membuat produk.
5. Adanya waktu menganggur bagi pengembang, karena harus menunggu anggota *team* proyek lainnya menuntaskan pekerjaanya.

2.13 Pemrograman PHP

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. *PHP* banyak dipakai untuk memrogram situs *web* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi *PHP* adalah forum (*phpBB*) dan *MediaWiki* (*software* di belakang Wikipedia). *PHP* juga dapat dilihat sebagai pilihan lain dari *ASP.NET/C#/VB.NET Microsoft*, *ColdFusion Macromedia*, *JSP/Java Sun Microsystems*, dan *CGI/Perl*. Contoh aplikasi lain yang lebih kompleks berupa *CMS* yang dibangun menggunakan *PHP* adalah *Mambo*, *Joomla!*, *Postnuke* dan *Xaraya*.

2.13.1 Sejarah PHP

Sejarah *PHP* Pada awalnya *PHP* merupakan kependekan dari Personal *Home Page* (Situs Personal). *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *FI* (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya *PHP/FI*. Dengan perilis kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan *PHP*. Pada November 1997, dirilis *PHP/FI 2.0*. Pada rilis ini *interpreter PHP* sudah

diimplementasikan dalam program *C*. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan *PHP/FI* secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *interpreter PHP* menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk *PHP* dan meresmikan rilis tersebut sebagai *PHP 3.0* dan singkatan *PHP* dirubah menjadi akronim berulang *PHP: Hypertext Preprocessing*. Pada pertengahan tahun 1999, Zend merilis *interpreter PHP* baru dan rilis tersebut dikenal dengan *PHP 4.0*. *PHP 4.0* adalah versi *PHP* yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi *web* kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis *PHP 5.0*. Dalam versi ini, inti dari *interpreter PHP* mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam *PHP* untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

2.13.2 Kelebihan *PHP* Dari Bahasa Pemrograman Lain

1. Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web server* yang mendukung *PHP* dapat ditemukan dimana - mana dari mulai *Apache, IIS, Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya makna dan *developer* yang siap membantu dalam pengembangan.

4. Dalam sisi pemahaman, *PHP* adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. *PHP* adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem. (Betha Sidiq, Ir. 2006)

2.14 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari *web* resminya.

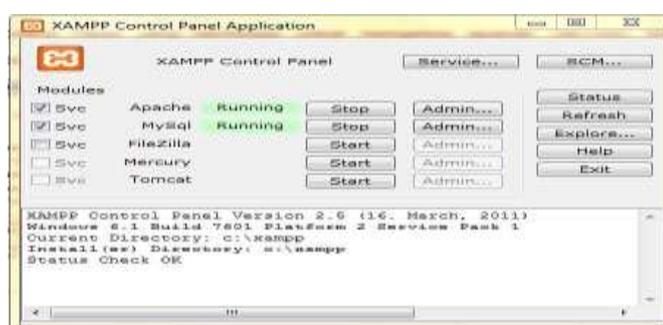
2.14.1 Bagian-Bagian Penting XAMPP

Mengenal bagian *XAMPP* yang biasa digunakan pada umumnya:

1. *htdocs* adalah folder tempat meletakkan berkas-berkas yang dijalankan, seperti berkas *PHP*, *HTML* dan skrip lain.

2. *Php MyAdmin* merupakan bagian untuk mengelola basis data *MySQL* yang ada dikomputer. Untuk membukanya, buka browser lalu ketikkan alamat <http://localhost/phpMyAdmin>, maka muncul halaman *phpMyAdmin*.
3. *Kontrol Panel* yang berfungsi untuk mengelola layanan (*service*) *XAMPP*. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

Aplikasi *XAMPP control panel* ditampilkan pada Gambar 2.5.



Gambar 2.5 XAMPP Control Panel.

2.15 MYSQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS* yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *PHP MyAdmin* adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen *MySQL*. Dengan menggunakan *phpmyadmin*, *user* dapat membuat *database*, membuat tabel, meng-*insert*, menghapus dan mengupdate data dengan *GUI* dan terasa lebih mudah, tanpa perlu mengetikkan perintah *SQL* secara manual. *PHP MyAdmin* dapat didownload secara *free* di <http://www.phpmyadmin.net>. Karena berbasis *web*, maka *phpmyadmin* dapat dijalankan di banyak sistem operasi, selama dapat menjalankan *webserver* dan *MySQL*. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis di

bawah lisensi *GNU General Public License (GPL)*, tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan *GPL*. Tidak sama dengan proyek-proyek seperti *Apache*, di mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia *MySQL AB*, di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan *MySQL AB* adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. *MySQL* adalah sebuah implementasi dari sistem manajemen basisdata relasional (*RDBMS*) yang didistribusikan secara gratis di bawah lisensi *GPL (General Public License)*. Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; *SQL (Structured Query Language)*. *SQL* adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem basisdata (*DBMS*) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah *SQL* yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, *MySQL* mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, *MySQL* dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata

kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis *web* (*wordpress*), *CMS*, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja *MySQL* pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional. (R.H Sianipar, 2013)

2.15.1 Sejarah *MYSQL*

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan *UNIREG* yang menggunakan koneksi *low-level ISAM database engine* dengan *indexing*. Pada saat itu Monty bekerja pada perusahaan bernama *TcX* di Swedia. *TcX* pada tahun 1994 mulai mengembangkan aplikasi berbasis *web*, dan berencana menggunakan *UNIREG* sebagai sistem *database*. Namun sayangnya, *UNIREG* dianggap tidak cocok untuk *database* yang dinamis seperti *web*. *TcX* kemudian mencoba mencari alternatif sistem database lainnya, salah satunya adalah *mSQL* (*miniSQL*). Namun *mSQL* versi 1 ini juga memiliki kekurangan, yaitu tidak mendukung *indexing*, sehingga performanya tidak terlalu bagus. Dengan tujuan memperbaiki performa *mSQL*, Monty mencoba menghubungi David Hughes (*programmer* yang mengembangkan *mSQL*) untuk menanyakan apakah ia tertarik mengembangkan sebuah konektor di *mSQL* yang dapat dihubungkan dengan *UNIREG ISAM*

sehingga mendukung *indexing*. Namun saat itu Hughes menolak, dengan alasan sedang mengembangkan teknologi *indexing* yang independen untuk *mSQL* versi 2.

Dikarenakan penolakan tersebut, David Hughes, *TcX* (dan juga Monty) akhirnya memutuskan untuk merancang dan mengembangkan sendiri konsep sistem database baru. Sistem ini merupakan gabungan dari *UNIREG* dan *mSQL* (yang source codenya dapat bebas digunakan). Sehingga pada May 1995, sebuah *RDBMS* baru, yang dinamakan *MySQL* dirilis. David Axmark dari *Detron HB*, rekanan *TcX* mengusulkan agar *MySQL* di 'jual' dengan model bisnis baru. Ia mengusulkan agar *MySQL* dikembangkan dan dirilis dengan gratis. Pendapatan perusahaan selanjutnya di dapat dari menjual jasa "*support*" untuk perusahaan yang ingin mengimplementasikan *MySQL*. Konsep bisnis ini sekarang dikenal dengan istilah *Open Source*. Pada tahun 1995 itu juga, *TcX* berubah nama menjadi *MySQL AB*, dengan Michael Widenius, David Axmark dan Allan Larsson sebagai pendirinya. Titel "*AB*" di belakang *MySQL*, adalah singkatan dari "*Aktiebolag*", istilah PT (Perseroan Terbatas) bagi perusahaan Swedia.

2.16 UML (Unified Modelling Language)

UML (Unified Modelling Language) adalah sebuah bahasa yang menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan *UML* dapat di buat model untuk semua jenis aplikasi *software*, dimana aplikasi tersebut dapat berjalan pada *hardware*, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Seperti bahasa-bahasa lainnya, *UML* mendefinisikan notasi dan *syntax* semantik.

Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan.

2.16.1 Konsep pemodelan menggunakan *UML*

Pemodelan menggunakan *Unified Modeling Language (UML)* merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan menggunakan *UML* merupakan pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. Terdapat berbagai diagram dalam merancang sistem menggunakan *UML*, diantaranya adalah:

1. *Usecase Diagram*

Usecase diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain *usecase diagram* secara grafis mendeskripsikan siapa yang menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu.

2. *Class Diagram*

Class diagram menggambarkan struktur *object* sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut.

3. *Sequence Diagram*

Sequence diagram secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *usecase* atau operasi. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan dalam sekuensi atau *timing* apa.

4. *Activity Diagram*

Activity diagram secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan *action* yang dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

2.16.2 Notasi Dalam UML

Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarakan berbagai diagram piranti lunak.

1. *Actor*

Actor menggambarkan segala pengguna *software* aplikasi (*user*). *Actor* memberikan suatu gambaran jelas tentang apa yang harus dikerjakan *software* aplikasi. Sebagai contoh sebuah *actor* dapat memberikan *input* ke dalam dan menerima informasi dari *software* aplikasi, perlu dicatat bahwa sebuah *actor* berinteraksi dengan *usecase*, tetapi tidak memiliki kontrol atas *usecase*. Sebuah *actor* mungkin seorang manusia, satu *device*, *hardware* atau sistem informasi lainnya. Notasi *actor* ditampilkan pada Gambar 2.6.



Gambar 2.6 Notasi Actor.

2. UseCase

Usecase menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, namun *usecase* hanya menjelaskan apa yang dilakukan oleh *actor* dan sistem bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut. Notasi *usecase* ditunjukkan pada Gambar 2.7.



Gambar 2.7 Notasi Usecase.

- a. **Use-case Konkret** adalah *usecase* yang dibuat langsung karena keperluan *actor*. *Actor* dapat melihat dan berinisiatif terhadapnya.
- b. **Use-case Abstrak** adalah *usecase* yang tidak pernah berdiri sendiri. *Usecase* abstrak senantiasa termasuk di dalam (*include*), diperluas dari (*extend*) atau memperumum (*generalize*) *usecase* lainnya.

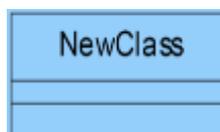
Untuk menggambarannya dalam *usecase* model biasanya digunakan *association relationship* yang memiliki *stereotype include*, *extend* atau *generalization relationship*. Hubungan *include* menggambarkan bahwa suatu *usecase* seluruhnya meliputi fungsionalitas dari *usecase* lainnya. Hubungan *extend* antar *usecase* berarti bahwa satu *usecase* merupakan tambahan

fungsionalitas dari *usecase* yang lain jika kondisi atau syarat tertentu terpenuhi.

3. *Class*

Class merupakan pembentuk utama dari sistem berorientasi objek, karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*.

Notasi *class* ditampilkan pada Gambar 2.8.



Gambar 2.8 Notasi *Class*.

Class mengabstraksikan elemen-elemen dari sistem yang sedang dibangun. *Class* bisa merepresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata. Notasi *class* berbentuk persegi panjang berisi 3 bagian persegi panjang paling atas untuk nama *class*, persegi panjang paling bawah untuk operasi, dan persegi panjang ditengah untuk atribut. Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas merepresentasikan informasi yang tersimpan di dalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh objek dan menggunakan kata kerja.

4. *Interface*

Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*. Implementasi operasi dalam *interface* dijabarkan oleh operasi di dalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang

mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan *prinsip enkapsulasi* dalam objek.

Notasi *interface* ditampilkan pada Gambar 2.9.



Gambar 2.9 Notasi *Interface*.

5. *Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar objek maupun hubungan antar objek. Biasanya *interaction* ini dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.

Notasi *interaction* ditampilkan pada Gambar 2.10.



Gambar 2.10 Notasi *Interaction*.

6. *Note*

Note digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa disertakan ke semua elemen notasi yang lain.

Notasi *note* ditampilkan pada Gambar 2.11.



Gambar 2.11 Notasi *Note*.

7. *Dependency*

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah.

Notasi *dependency* ditampilkan pada Gambar 2.12.



Gambar 2.12 Notasi *Dependency*.

Terdapat 2 *stereotype* dari *dependency*, yaitu *include* dan *extend*. *Include* menunjukkan bahwa suatu bagian dari elemen (yang ada di garis tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah). *Extend* menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan ke dalam elemen yang ada di garis dengan panah.

8. *Association*

Association menggambarkan navigasi antar *class* (*navigation*), berapa banyak objek lain yang bisa berhubungan dengan satu objek (*multiplicity* antar *class*) dan apakah suatu *class* menjadi bagian dari *class* lainnya (*aggregation*).

Notasi *association* ditampilkan pada Gambar 2.13.

Gambar 2.13 Notasi *Association*.

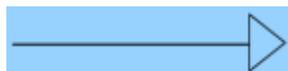
Navigation dilambangkan dengan penambahan tanda panah di akhir garis. *Bidirectional navigation* menunjukkan bahwa dengan mengetahui salah satu *class* bisa di dapatkan informasi dari *class* lainnya. Sementara *Unidirectional navigation* hanya dengan mengetahui *class* di ujung garis *association* tanpa

panah bisa di dapatkan informasi dari *class* di ujung dengan panah, tetapi tidak sebaliknya. *Aggregation* mengacu pada hubungan “*has-a*”, yaitu bahwa suatu *class* memiliki *class* lain, misalnya Rumah memiliki *class* Kamar.

9. Generalization

Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan *generalization*, *class* yang lebih spesifik (*subclass*) akan menurunkan atribut dan operasi dari *class* yang lebih umum (*superclass*) atau “*subclass is superclass*”. Dengan menggunakan notasi *generalization* ini, konsep *inheritance* dari prinsip hirarki dapat dimodelkan.

Notasi *generalization* ditampilkan pada Gambar 2.14.



Gambar 2.14 Notasi Generalization.

10. Realization

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah. Misalnya *class* merealisasikan *package*, *component* merealisasikan *class* atau *interface*. (Fowler Martin. 2005)

Notasi *realization* ditampilkan pada Gambar 2.15.



Gambar 2.15 Notasi Realization.

2.17 Diagram Kelas (*Class Diagram*)

Menurut Herlawati dan Widodo (2011), *class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok, yaitu nama (dan *stereotype*), atribut, dan metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- c. *Public*, dapat dipanggil oleh siapa saja

Simbol-simbol *Class Diagram* ditampilkan pada Tabel 2.1.

Tabel 2.1 Simbol Class Diagram.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboratio</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang

			terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.18 Diagram UseCase (UseCase Diagram)

Menurut Herlawati dan Widodo (2011), *usecase diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *usecase* merepresentasikan sebuah interaksi antara aktor dengan sistem. Seorang aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Simbol-simbol *Usecase Diagram* ditampilkan pada Tabel 2.2.

Tabel 2.2 Simbol Usecase Diagram.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>usecase</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>usecase</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>usecase</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Usecase</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

BAB III METODE PENELITIAN

3.1 Tempat Dan Waktu Penelitian

Penelitian dilakukan di Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara. Waktu penelitian dilakukan pada semester ganjil tahun ajaran 2017/2018.

3.2 Perangkat Penelitian

Hardware yang digunakan dalam sistem informasi kependudukan ini adalah *Lenovo V310-14ISK/Lenovo V310-15ISK* dengan spesifikasi sebagai berikut.

1. *AMD Radeon Graphics Processor (0x6660)*
2. *AMD Radeon (TM) R5 M430*
3. *4096 MB RAM Memory*
4. *1024 GB HDD*
5. *Display 14.0" HD (1366×768) Color, Antiglare, LED Back Light 250 NITS, 16:9 Aspect Ratio*

Perangkat lunak yang digunakan dalam rancangan ini antara lain :

1. Sistem operasi *Windows 10 Professional 64-bit*
2. *Visual Studio Code*

3. *XAMPP Server Version 3.2.2*
4. *Web browser Google Firefox dan Google Chrome*
5. *SQL-yog 64-bit*

3.3 Analisa Kebutuhan

Analisa kebutuhan sistem dilakukan dengan tujuan untuk menggambarkan secara garis besar sistem yang sedang berjalan di setiap alur, baik dari kendala sistem yang berjalan maupun menjabarkan kebutuhan *user*, mendefinisikan semua kebutuhan yang diberikan *user* pada aplikasi yang dibangun, dan menentukan dasar desain dari perangkat lunak. Aplikasi kebutuhan dari aplikasi yang dirancang dilihat dari beberapa sudut pandang yaitu dari *staff* yang membutuhkan informasi tentang pengolahan data, sehingga dapat dikemukakan beberapa analisa kebutuhan sebagai berikut:

1. Dibutuhkan sistem yang dapat memudahkan *staff* di Kecamatan dalam melakukan pengelolaan data kependudukan.
2. Dibutuhkan sebuah sistem yang dapat menyimpan data-data setiap memasukan data penduduk.
3. Dibutuhkan sistem yang dapat diakses kapan saja oleh *user*, dimana aplikasi tersebut memberikan informasi.

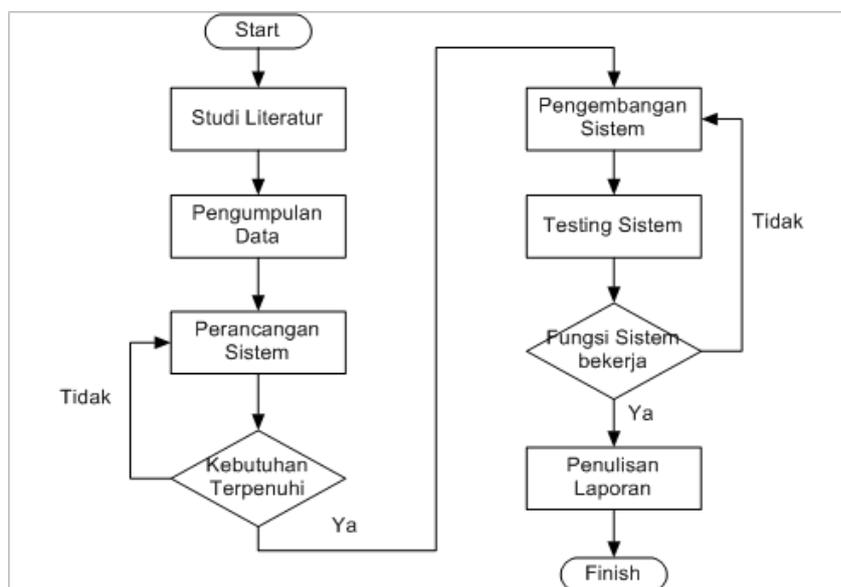
3.4 Tahapan Penelitian

Tahapan penelitian ini dilakukan dengan beberapa langkah yaitu studi literatur, perancangan sistem, pengembangan sistem, pengujian sistem, dan penulisan laporan.

3.4.1 Tahap Perumusan Masalah

Tahapan ini merupakan proses merumuskan dan membatasi masalah yang akan diteliti. Perumusan dan pembatasan masalah diperlukan agar dapat lebih mengarahkan peneliti dalam membuat sistem sehingga proyek yang dikerjakan tidak keluar dari batasan yang telah ditetapkan sebelumnya.

Diagram alir penelitian ditampilkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian.

3.4.2 Tahap Pengumpulan Data

Tahapan pengumpulan data dilakukan dengan menggunakan dua metode yaitu melalui studi pustaka dan wawancara.

a. Observasi

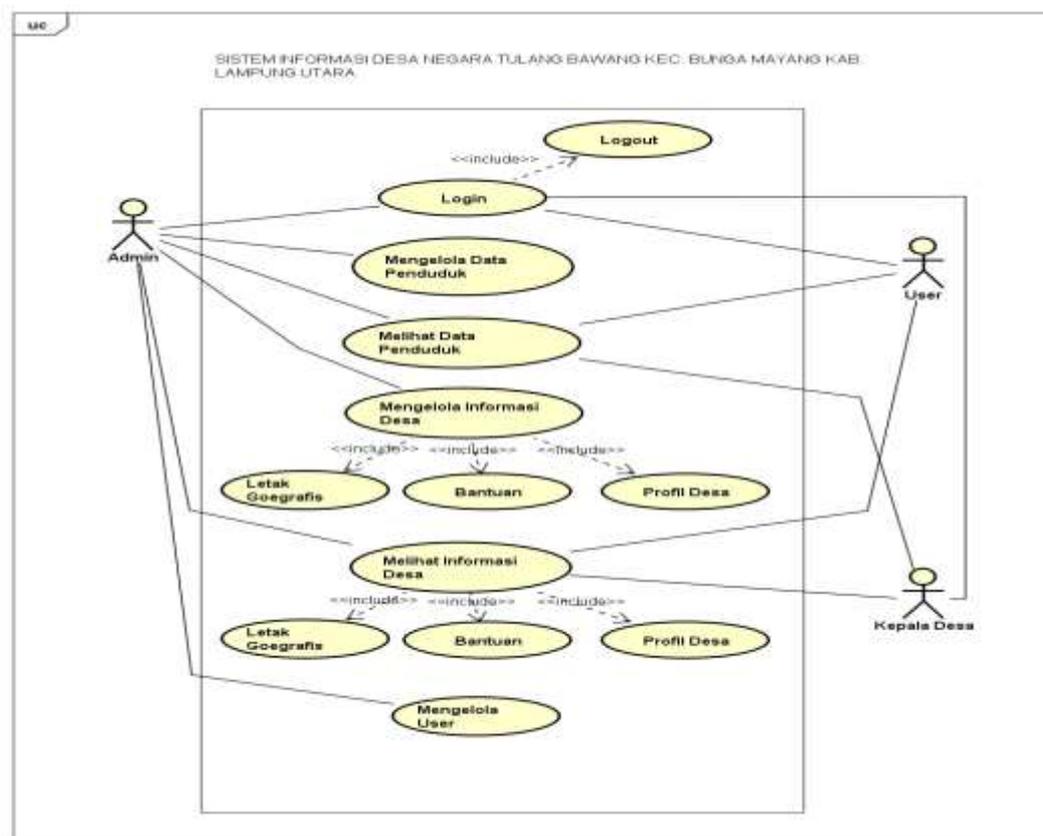
Pada tahap ini dilakukan proses pengumpulan data secara langsung kepada pihak yang terlibat dalam pengembangan aplikasi ini. Dalam metode observasi diperoleh struktur organisasi, dan database serta sistem yang dibutuhkan dalam penelitian ini.

b. Studi Pustaka

Pada tahap studi pustaka data dikumpulkan melalui berbagai literatur seperti pada buku, jurnal, ataupun dokumen yang berkaitan dengan penelitian ini. Hal ini bertujuan untuk mendapatkan data yang tidak ditemukan pada metode observasi. Selanjutnya data-data yang telah dikumpulkan disusun menjadi basis aturan yang digunakan dalam aplikasi.

3.5 Usecase Diagram

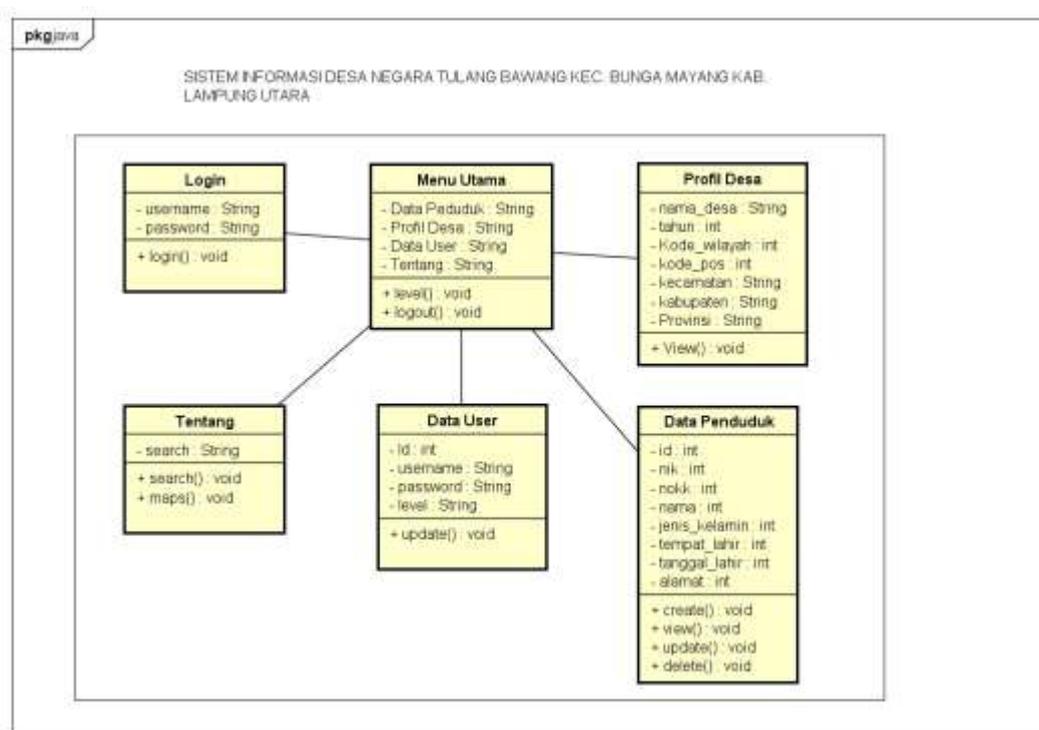
Usecase Diagram Sistem Informasi Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara Berbasis *Website*. *Usecase diagram* Sistem Informasi Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara Berbasis *Website* ditampilkan pada Gambar 3.2.



Gambar 3.2 Usecase Diagram.

3.6 Class Diagram

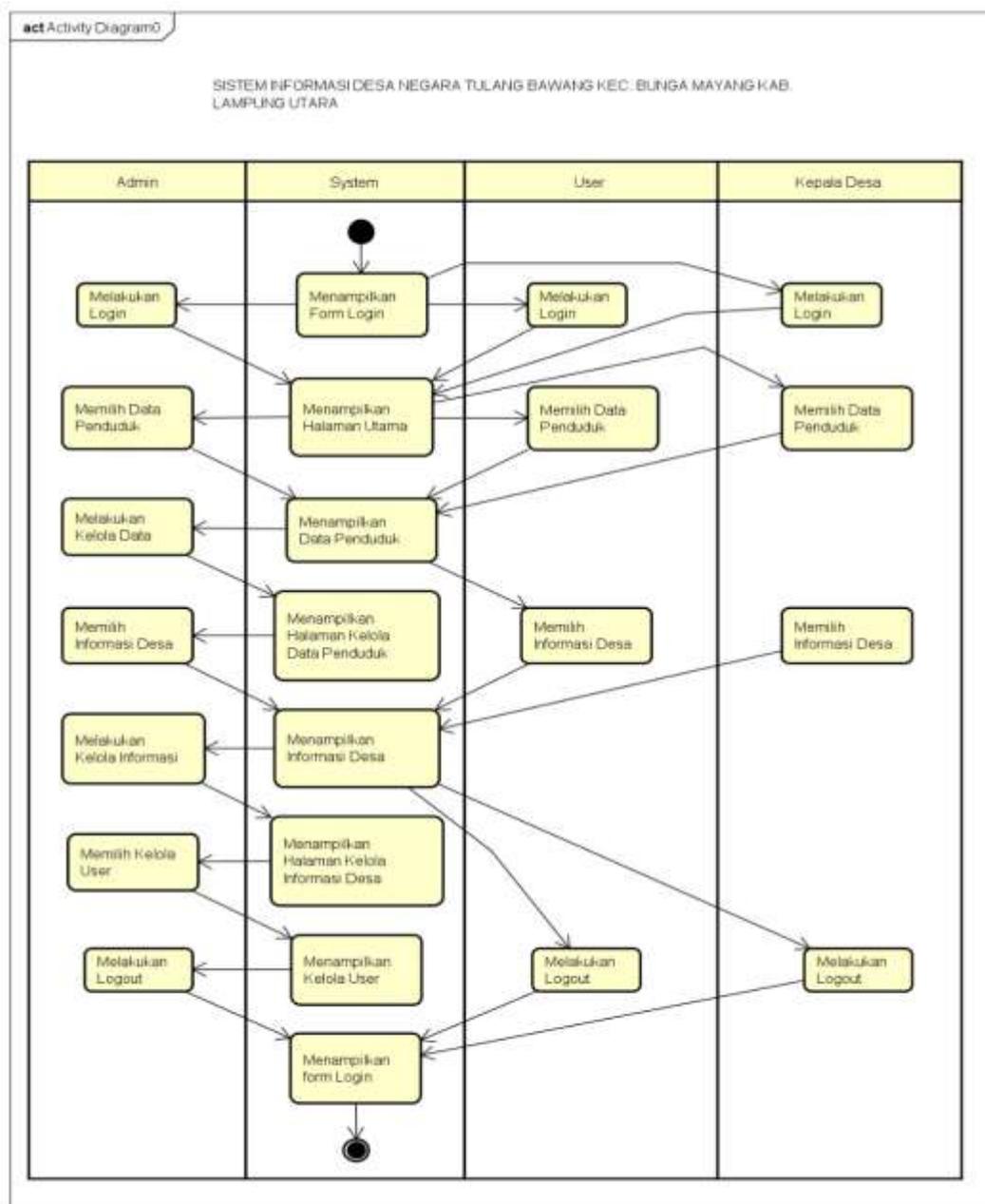
Langkah selanjutnya adalah membuat *class diagram* berdasarkan *usecase diagram* tersebut. *class diagram* ini harus berisikan objek-objek yang terdapat di dalam sistem. *class diagram* adalah *class* yang dapat direpresentasikan ke dalam sebuah *interface* atau sebaliknya merupakan implementasi dari sebuah *interface*. Gambar diatas merupakan *class diagram* yang menggambarkan beberapa *class* yang digunakan dalam sistem ini beserta atribut-atribut, proses-proses dan relasi-relasi yang terdapat di dalamnya. *Class diagram* Sistem Informasi Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara Berbasis Website ditampilkan pada Gambar 3.3.



Gambar 3.3 Class Diagram.

3.7 Activity Diagram

Activity diagram merupakan alur kerja (*workflow*) atau kegiatan dari sebuah sistem atau menu yang ada pada perangkat lunak. *Activity Diagram* juga digunakan untuk mendefinisikan urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan serta rancang menu yang ditampilkan pada perangkat lunak. *Activity diagram* dari Sistem Informasi Desa Negara Tulang Bawang Kecamatan Bunga Mayang Kabupaten Lampung Utara Berbasis Website ditampilkan pada Gambar 3.4.



Gambar 3.4 Activity Diagram.

3.8 Sequence Diagram

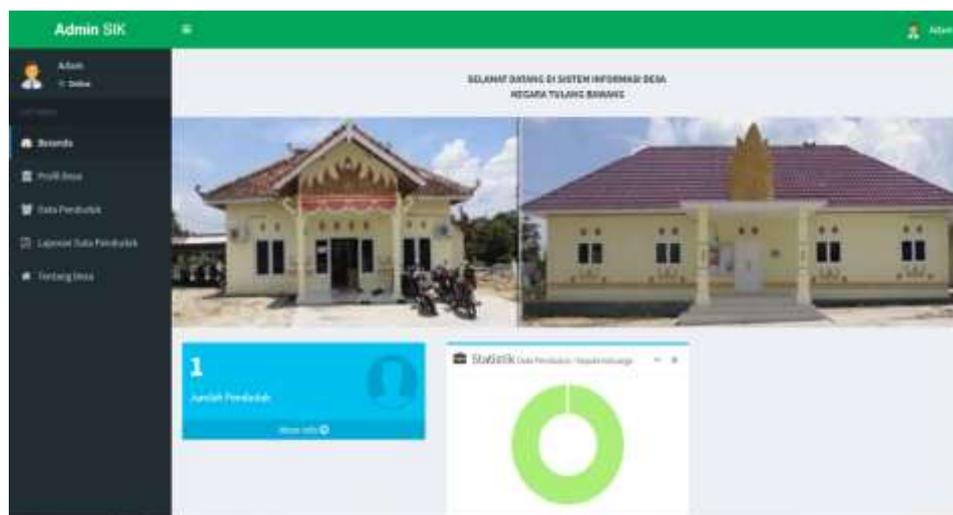
Sequence Diagram adalah salah satu dari diagram – diagram yang ada pada *UML*, *sequence diagram* ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah objek. *Sequence diagram* dari Sistem Informasi Desa Negara



Gambar 3.6 Tampilan *Form Login*.

2. Beranda

Tampilan beranda sistem informasi Desa Negara Tulang Bawang yang dikelola oleh admin petugas desa. Ditampilkan pada Gambar 3.7.



Gambar 3.7 Tampilan Beranda.

3. Profil Desa

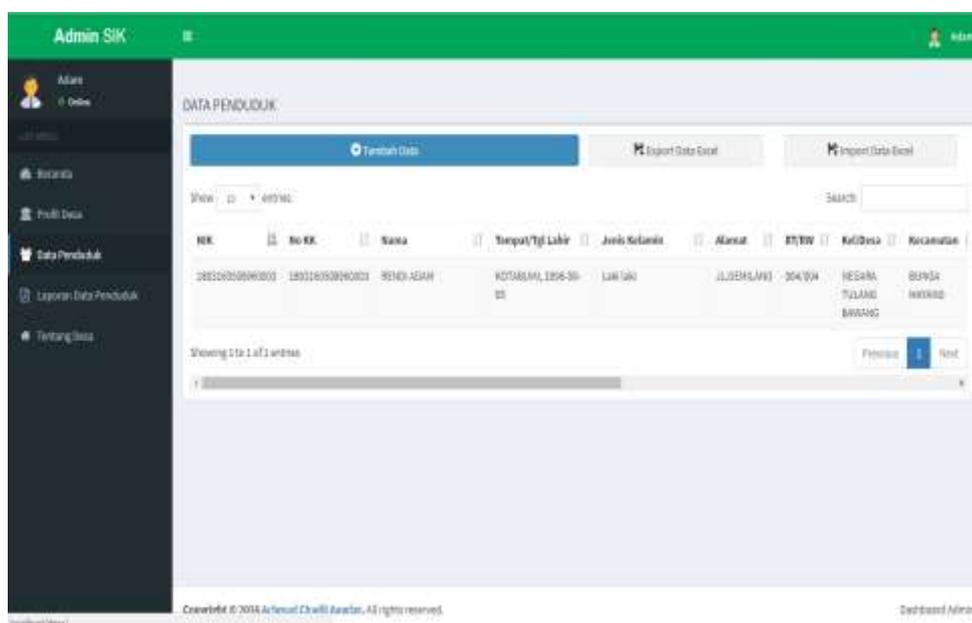
Profil Desa Negara Tulang Bawang berupa struktur pemerintahan serta visi dan misi yang ada di Desa Negara Tulang Bawang. Profil desa ditampilkan pada Gambar 3.8.



Gambar 3.8 Profil Desa.

4. Data Penduduk

Data penduduk yang telah dimasukkan oleh admin petugas desa ditampilkan pada Gambar 3.9.



	Kel/Desa	Kecamatan	Agama	Status Perkawinan	Pekerjaan	Kewarganegaraan	No Telp	Aksi
1	NEGARA TULANG BARANG	BUNGA MAHANG	Islam	BELUM KAWIN	PELAJAR	WNI	081233072122	Update Delete

Gambar 3.9 Tampilan Data Penduduk.

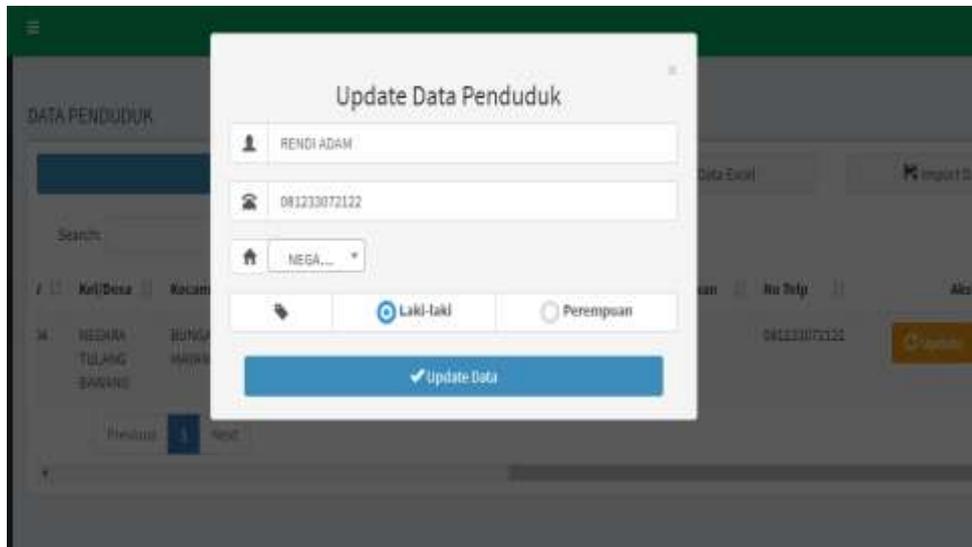
5. Tambah data

Didalam data penduduk terdapat tambah data untuk menambahkan data warga yang ingin didata berupa nama, NIK, no. KK, tempat lahir, tanggal lahir, jenis kelamin, alamat, RT/RW, nama desa, agama, status dan pekerjaan ditampilkan pada Gambar 3.10.

Gambar 3.10 Tampilan Tambah Data.

6. Ubah data

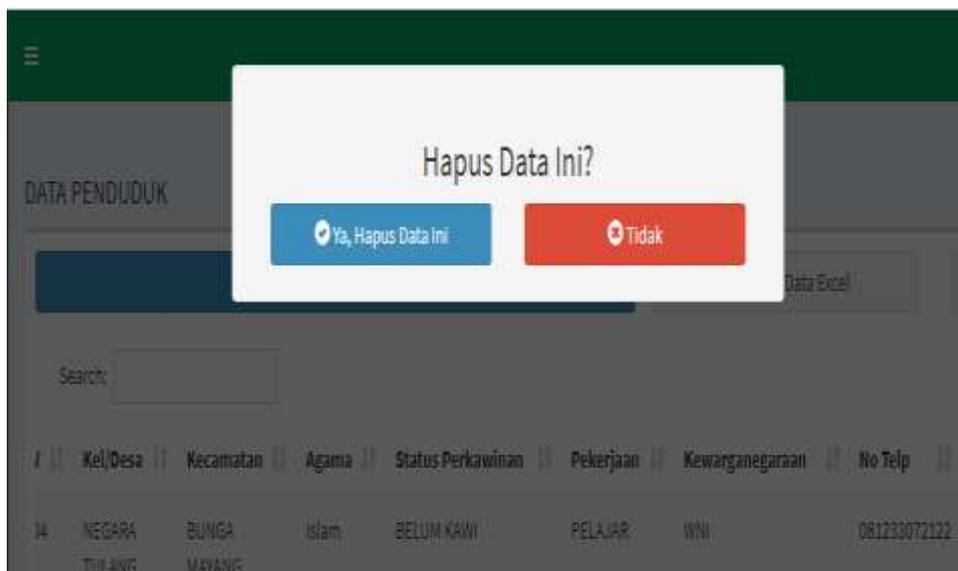
Update data penduduk untuk mengubah data tersebut. Ditampilkan pada Gambar 3.11.



Gambar 3.11 Tampilan Ubah Data.

7. Hapus data

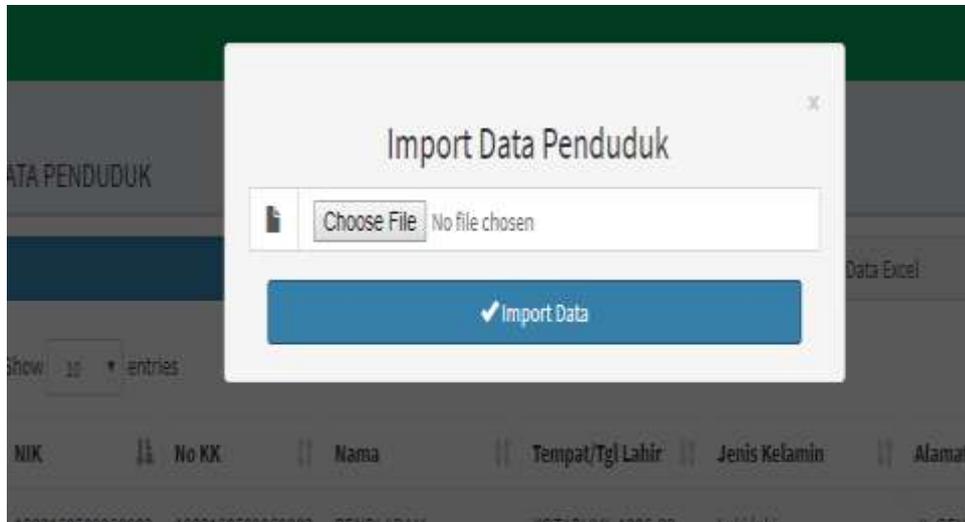
Menghapus data ditampilkan pada Gambar 3.12.



Gambar 3.12 Tampilan Hapus Data.

8. Import data

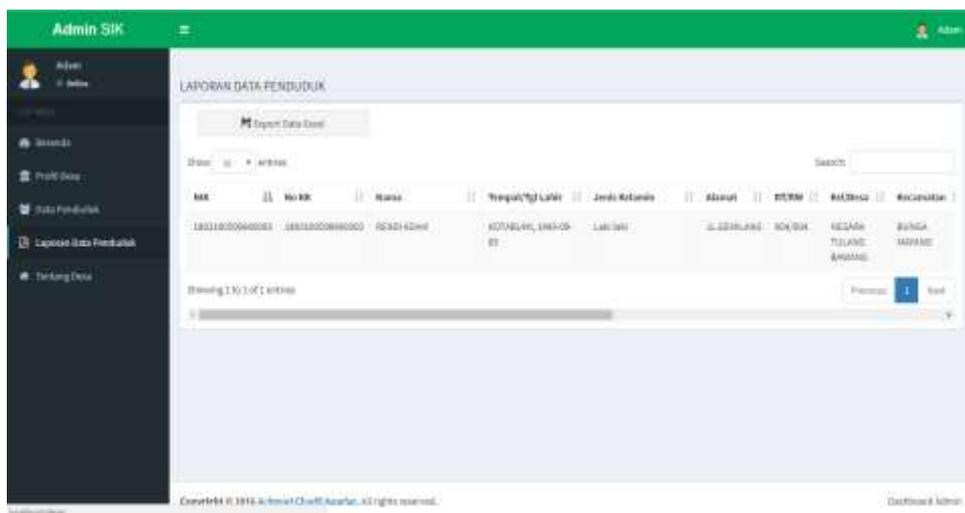
Mengimport data penduduk ditampilkan pada Gambar 3.13.



Gambar 3.13 Tampilan *Import Data*.

9. Laporan Data Penduduk + export data to excel

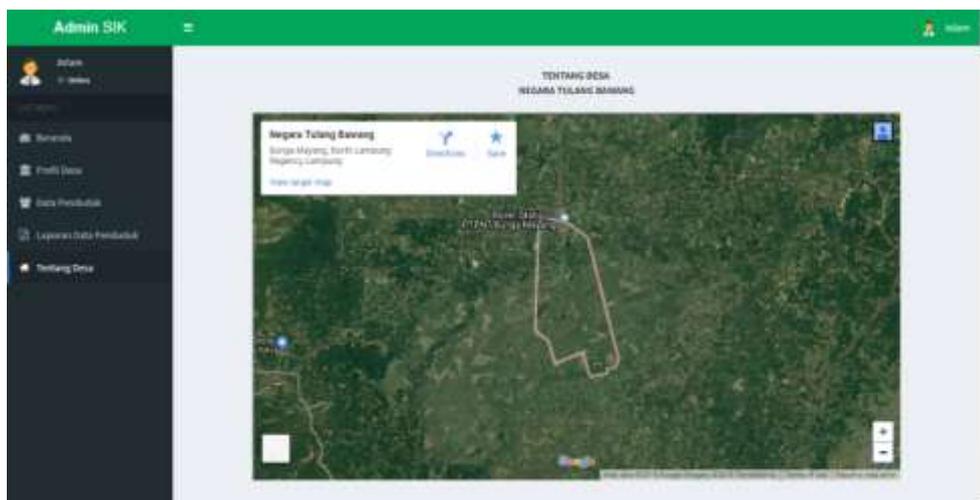
Laporan data penduduk terdapat *export data excel*. Ditampilkan pada Gambar 3.14.



Gambar 3.14 Tampilan Laporan Data Penduduk + *Export Data To Excel*.

10. Tentang Desa

Di dalam menu sistem informasi ini terdapat pilihan “tentang desa” yang didalamnya adalah sebuah *maps* letak keberadaan Desa Negara Tulang Bawang. Ditampilkan pada Gambar 3.15.



Gambar 3.15 Tampilan Tentang Desa.

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian yang dilakukan dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem yang dirancang khusus untuk memfasilitasi perangkat desa.
2. Hasil pengujian menunjukkan bahwa fungsi-fungsi yang terdapat pada sistem berjalan dengan baik dan sesuai yang diharapkan.
3. Sistem dapat berjalan dengan baik untuk meningkatkan dan membantu kerja *staff* desa dalam pengelolaan data kependudukan, mempermudah pendataan dengan sistem menjadi lebih cepat dan akurat.

5.2 Saran

Berberapa saran yang diberikan setelah dilakukan penelitian ini untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Sistem ini diharapkan dapat dikembangkan lebih lanjut dengan memberi fungsi tambahan berupa fungsi untuk mencetak *report* pada data penduduk.
2. Aplikasi dapat dikembangkan pada penelitian selanjutnya menggunakan sistem android.

DAFTAR PUSTAKA

- Al-Bahra bin Ladjamudin. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta : Graha Ilmu. Elcom
- Al Fatta, Hanif. (2007). *Analisis dan Perancangan Sistem Informasi*. Yogyakarta : Penerbit Andi.
- Al-Khouri, A.M. (2011) *An Innovative Approach For E-Government Transformation*. diunduh dari <http://arxiv.org/ftp/arxiv/papers/1105/1105.6358.pdf>. diakses Jumat, 01 Februari 2013.
- Betha, Sidiq, Ir. 2006. *Pemrograman Web Dengan PHP*. Bandung : Informatika
- Caldow, Janet. 2001. *Intitute for Electronic Government*. UK : IBM Corporation.
- Departemen Pendidikan Nasional, 2005. *Peraturan Pemerintah Nomor 19 Tahun 2005, tentang Standar Nasional Pendidikan*, Jakarta: Depdiknas.
- Jogiyanto, H.M., 2005, *Analisa dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*, Andi, Yogyakarta.
- Kadir, Abdul. (2003). *Pengenalan Sistem Informasi*. Yogyakarta : Penerbit Andi.
- Lulita, M.C. (2011). *Kinerja birokrasi desa dalam meningkatkan pelayanan umum di Desa Cukurgondang Kecamatan Grati Kabupaten Pasuruan*. Diunduh dari <http://library.um.ac.id/free-contents/index.php/pub/detail/kinerja-birokrasi-desa-dalam-meningkatkan-pelayanan-umum-di-desa-cukurgondang-kecamatan-grati-kabupaten-pasuruan-mega-clara-lulita-48942.html>.
- Patricia, Pascual J. 2003. *E-Government: e-Primers for The Informations Economy Society and Policy*. UNDP Asia Pasific.
- R H Sianipar, S.T., M.T., M.Eng, Ph.D, *Teori Dan Implementasi JAVA*, 1st ed. Bandung, Indonesia: Informatika Bandung, 2013.
- Sadu Wasistiono & Irwan Tahir, 2006: *Prospek Pengembangan Desa*, Bandung: CV. Fokusmedia.

- Sianipar, R.H. (2013). *Membangun Web dengan PHP dan MySQL* : Bandung : Informatika.
- Undang-Undang Republik Indonesia. 2005. UU Nomor 57 Tahun 2005 *Tentang Desa*.
- Wasistiono S, Tahir I. 2006. *Prospek Pengembangan Desa*. Fokus Media: Bandung.
- Depdagri (2007). *Naskah Akademik Tentang Rancangan Undang-Undang Tentang Desa*. Direktorat Pemerintahan Desa Dan Kelurahan, Direktorat Jenderal Pemberdayaan Masyarakat Dan Desa, Departemen Dalam Negeri, Jakarta.