

Pembangunan Sistem Informasi Bagian Pelayanan dan Administrasi Desa pada Desa Campurejo

Muh. Thanthowi Lathif¹, Denny Sagita Rusdianto², Edy Santoso³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹lathif@student.ub.ac.id, ²denny.sagita@ub.ac.id, ³edy144@ub.ac.id

Abstrak

Balaidesa Campurejo merupakan pusat pelayanan bagi warga dan masyarakat Desa Campurejo, Kecamatan Sambil, Kabupaten Ponorogo, Jawa Timur. Pelayanan yang terdapat pada Balaidesa cukuplah beragam, seperti: pengajuan surat menyurat, pengurusan data penduduk, pengembangan desa, dan kebutuhan warga lainnya. Saat ini pelayanan tersebut dilakukan secara *manual* dengan cara memasukkan data pada *form* atau *file excel* yang tersedia. Sehingga proses pelayanan cukup memakan waktu yang lebih lama. Kemudian, terdapat permasalahan dalam penyimpanan data yang tidak tersentralisasi dan tidak tersimpan secara rapi. Sehingga ketika perangkat desa membutuhkan data tertentu, seperti: Laporan Data Penduduk, Rekap Penduduk Berdasarkan Dusun dan data lainnya akan membutuhkan waktu yang cukup lama, karena perlu mencari dan menghitung atau mengelompokkan secara *manual* kembali. Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah aplikasi. Aplikasi tersebut merupakan Sistem Informasi Desa (SIDESA) yang berbasis *website*. Aplikasi ini akan mempermudah perangkat desa dalam memasukkan data pada *form* yang tersedia dan menyimpan data hasil masukan pada *database* sehingga tersimpan dengan rapi dan mudah diakses untuk kedepannya. Aplikasi ini menggunakan beberapa bahasa pemrograman yaitu: *Mysql*, *PHP*, *CSS*, *JavaScript*, dll. Hasil pengujian aplikasi ini mendapatkan 100% valid untuk pengujian unit, integrasi, dan validasi. Sedangkan mendapatkan hasil 1 *major issues* dan 3 *minor issues* dalam pengujian *compatibility*

Kata kunci: *desa campurejo, slim, SIDESA*

Abstract

The Campurejo Village Hall is a service center for the residents and community of Campurejo Village, Sambil Subdistrict, Ponorogo Regency, East Java. The services provided at Balaidesa are quite diverse, such as: filing correspondence, managing population data, village development, and other community needs. Currently, the service is done manually by entering data in an available form or excel file. So that the service process takes quite a longer time. Then, there are problems in data storage that is not centralized and not stored neatly. So that when village officials need certain data, such as: Population Data Reports, Population Recaps Based on Dusun and other data, it will take quite a long time, because they need to find and calculate or group manually again. To overcome these problems, we need an application. The application is a website-based Village Information System (SIDESA). This application will make it easier for village officials to enter data on the available forms and store the data entered in the database so that it is stored neatly and easily accessible for the future. This application uses several programming languages, namely: Mysql, PHP, CSS, JavaScript, etc. The test results of this application get 100% valid for unit testing, integration, and validation. While getting the results of 1 major issues and 3 minor issues in compatibility testing.

Keywords: *desa campurejo, slim, SIDESA*

1. PENDAHULUAN

Desa Campurejo yang terdapat pada Kecamatan Sambit, Kabupaten Ponorogo, Jawa Timur memiliki banyak warga dan masyarakat.

Pusat pelayanan dilakukan pada Balaidesa Campurejo. Pelayanan yang diberikan mencakup pengurusan surat menyurat, namun terdapat permasalahan dalam melayani administrasi surat menyurat.

Permasalahan yang sering dialami disebabkan karena pengurusan surat menyurat masih menggunakan cara *konvensional* dengan mengetikkan pada *file excel*. Dalam peng-*input*-an data pada *file excel*, perangkat desa kesulitan dikarenakan *form* yang digunakan kurang rapi. Setelah melakukan proses peng-*input*-an data yang memakan waktu, data yang telah di-*input*-kan tidak tersimpan pada satu tempat yang sama, seperti pada *database*. Selain proses pengurusan surat menyurat yang masih dilakukan dengan cara *manual*. Perangkat desa juga mengalami kesulitan dalam melakukan rekap data. Dalam melakukan rekap data, seringkali terdapat beberapa *parameter* yang harus diseleksi, contohnya umur, status kawin, dusun, pekerjaan, dan lain sebagainya.

Dalam memajukan pembangunan desa, perangkat desa juga sering melakukan pengadaan barang. Terdapat masalah dalam melakukan penyusunan perencanaan pengadaan barang. Permasalahan tersebut disebabkan karena proses pengadaan dilakukan dengan mencetak pengajuan kemudian menemui sekertaris desa dan kepada desa, sedangkan seringkali sekertaris dan kepala desa melakukan dinas luar. Hal tersebut menyebabkan kendala ketika melakukan Laporan Pertanggung Jawaban (SPJ) dikarenakan surat pesanan terpisah-pisah dan tidak tersusun dengan rapi.

Setelah mengetahui beberapa permasalahan tersebut. [enulis mencoba untuk mengembangkan sistem bernama SIDESA (sistem informasi dan pelayanan desa). Dengan dikembangkannya sistem SIDESA, diharapkan permasalahan yang dialami oleh perangkat desa dapat terselesaikan dan lebih mudah dalam melakukan pelayanan pada warga dan masyarakat. Sistem ini akan berbasis *Website* sehingga dapat diakses secara *online* dan akan *terupdate* secara *realtime*. Data penduduk, data surat, akan tersimpan pada *database* sehingga tersusun dengan rapi dan lebih mudah dalam melakukan rekap data sesuai *parameter* yang ditentukan.

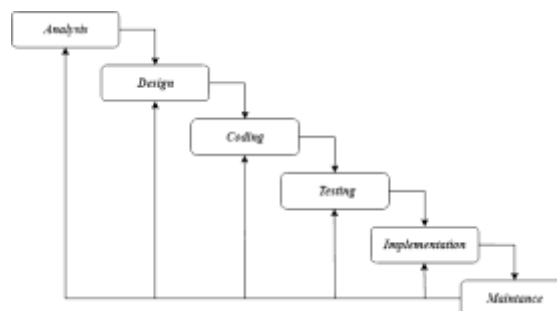
2. LANDASAN KEPUSTAKAAN

2.1 Metode Waterfall

Metode *Waterfall* adalah salah satu metode dalam melakukan pengembangan perangkat lunak yang sering digunakan. Metode ini menggunakan pendekatan *Software Development Life Cycle* (SDLC). Menurut

(Sukamto & Shalahuddin, 2018), Model SDLC air terjun (*waterfall*) sering juga disebut juga model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*).

Urutan pada metode *waterfall* bersifat serial. Tahapan pada metode *waterfall* antara lain: analisis, desain, coding, testing, implementasi, perawatan. Metode *Waterfall* ditampilk pada Gambar 1.



Gambar 1 Metode Waterfall

Sumber: (Adenowo & Adenowo, 2013)

2.2 Class Diagram

Class diagram digunakan dalam menerjemahkan model yang dibuat menjadi sebuah kode program. *Class diagram* akan menggambarkan hubungan antar kelas secara rinci. Hubungan atau yang biasa disebut relasi tersebut dapat dibagi menjadi tiga jenis, yaitu asosiasi, agregasi, dan generalisasi. Pada jenis Asosiasi merupakan makna umum antara dua kelas, Agregasi merupakan pernyataan suatu kelas merupakan atribut pada kelas lain, dan Generalisasi merupakan pernyataan antar kelas dalam hubungan *inheritance*.

2.3 Entity Relationship Diagram (ERD)

Dalam melakukan pembuatan suatu *database* dibutuhkan struktur atau *diagram* yang menghubungkan setiap entitas. *Diagram* tersebut disebut dengan *Entity Relationship Diagram* (ERD). Dengan dibuatnya *Entity Relationship Diagram* (ERD) maka akan mempermudah dalam pembuatan *database* dan menghindari terjadinya kesalahan dalam membuat *database*.

2.4 Use Case Diagram

Dalam suatu sistem, terdapat beberapa pengguna yang berinteraksi atau menggunakan sistem tersebut. Hubungan antara pengguna dan

sistem digambarkan dengan *Use Case Diagram*. Fungsi lain dari *Use Case Diagram* adalah untuk mengetahui seberapa banyak fitur yang dibutuhkan oleh sistem dalam memenuhi kebutuhan pengguna. Menurut (Kurniawan & Syarifuddin, 2020). *Use case diagram* merupakan diagram yang bekerja dengan mendeskripsikan tipikal interaksi antar *user* (pengguna) sebuah sistem dengan suatu sistem tersendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

2.5 Slim

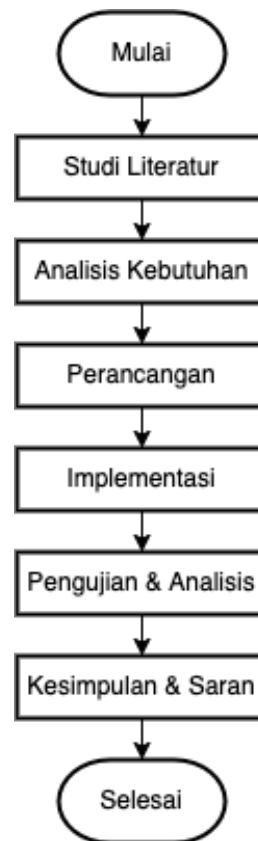
Slim merupakan salah satu *mikro framework* pada Bahasa pemrograman PHP. Menurut (Syahroni & Ubaidi, 2019), *Slim* biasanya digunakan untuk membuat webservice atau REST API. Konsep kerja pada *Slim* yang termasuk sederhana menjadikan alasan kenapa *slim* sering digunakan dalam melakukan proses HTTP *Request* dan *Response*. Seperti *framework* pada Bahasa PHP lainnya, *Slim* juga menganut struktur permodelan MVC atau *Model View Controller*.

2.6 AngularJs

Pada Bahasa pemrograman *JavaScript* terdapat banyak sekali *framework* yang tersedia, salah satunya adalah *AngularJs*. Dalam penerapannya, *framework AngularJS* mengadopsi permodelan *Model View Controller* (MVC). Pada tahun 2009, Misko Herverly dan Adam Abrons mengembangkan *AngularJs*

3. METODOLOGI

Pada penelitian ini terdapat beberapa tahapan yang dilakukan, tahapan tersebut dirangkum pada bagian Metodologi. Metodologi pada penelitian ini akan ditampilkan pada Gambar 2.



Gambar 2 Metodologi Penelitian

Pada tahapan Studi Literatur, peneliti melakukan pencarian penelitian terdahulu dan teori yang dapat digunakan sebagai referensi dalam melakukan penelitian ini. Pada penelitian ini penulis mencari studi literatur terkait:

1. Metode *Waterfall*
2. *Activity Diagram*
3. *Use Case Diagram*
4. *Class Diagram*
5. *Sequence Diagram*
6. *Entity Relationship Diagram*
7. *Slim*
8. *Angularjs*
9. *Hypertext Markup Language (HTML)*
10. *Hypertext Processing (PHP)*
11. *JavaScript*
12. *Cascading Style Sheet (CSS)*
13. *Mysql*
14. Pengujian *Whitebox*
15. Pengujian *Blackbox*

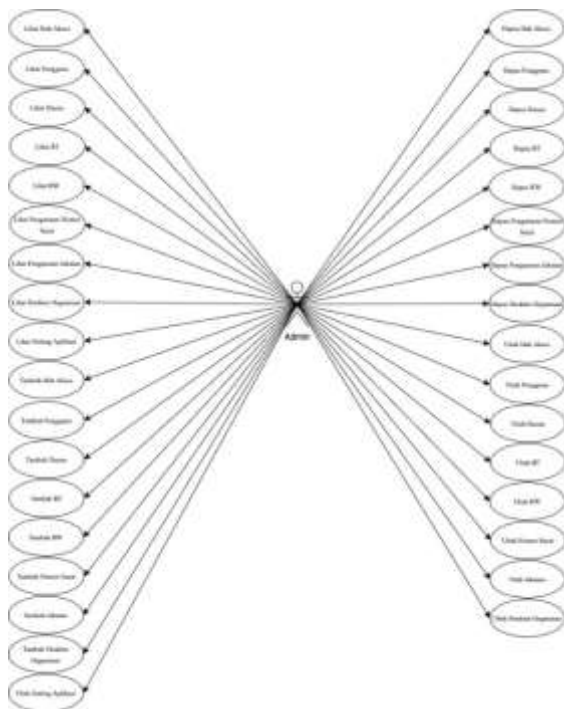
Pada tahapan Analisis Kebutuhan, peneliti melakukan analisis terhadap kebutuhan yang diperlukan dalam menyelesaikan penelitian ini. Pada tahapan Perancangan, peneliti melakukan perancangan desain dan struktur pada perangkat lunak yang akan dikembangkan. Pada tahapan Implementasi, peneliti melakukan

penerapan hasil dari tahapan-tahapan sebelumnya. Pada tahapan Pengujian dan Analisis, peneliti melakukan pengujian pada perangkat lunak yang telah dikembangkan. Pada tahapan Kesimpulan, peneliti akan menyimpulkan hasil dari pengujian dan analisis yang telah dilakukan, sehingga dapat ditarik sebuah kesimpulan dan saran agar dapat digunakan sebagai rujukan pada penelitian selanjutnya.

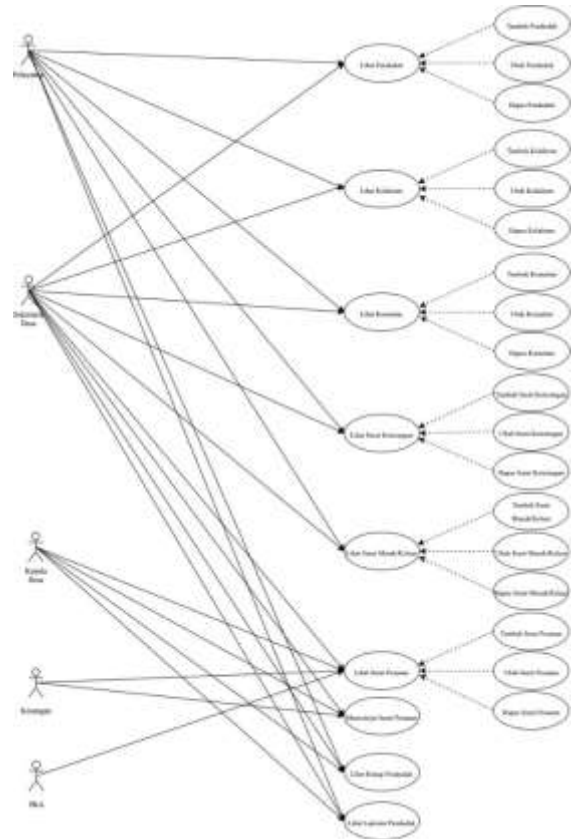
4. ANALISIS KEBUTUHAN

4.1 Use Case Diagram

Terdapat dua pembagian hubungan yang dapat digambarkan pada *use case diagram* menggunakan diagram UML.



Gambar 4.1 Use Case Diagram – Admin

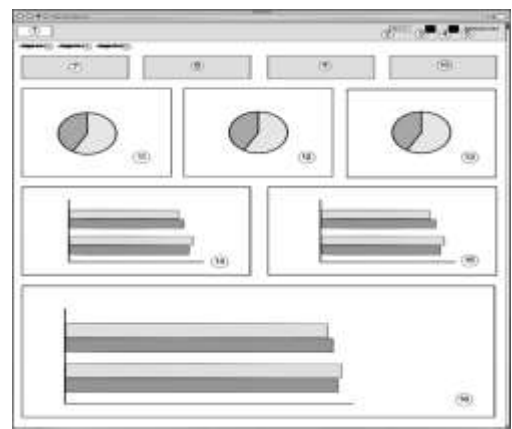


Gambar 4.2 Use Case Diagram

5. PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk membantu dalam mengimplementasikan tampilan antarmuka yang diinginkan. Perancangan antarmuka akan ditampilkan pada Gambar 3



Gambar 3. Perancangan Antarmuka

5.2 Implementasi Antarmuka

Implementasi antarmuka akan

menampilkan hasil dari perancangan antarmuka yang telah *didesain* sebelumnya. Hasil Implementasi Antarmuka akan ditampilkan pada Gambar 4.



Gambar 4. Implementasi Antarmuka

6. PENGUJIAN

6.1 Pengujian *Whitebox*

Pengujian *Whitebox* dilakukan pada pengujian unit dan pengujian integrasi dengan mengujikan pada method login, tambah pengguna, tambah surat keterangan, tambah surat pesanan, dan approve surat pesanan. Hasil dari pengujian Unit dan Integrasi didapatkan 100% valid.

6.2 Pengujian *Blackbox*

Pengujian *Blackbox* dilakukan pada pengujian validasi dengan mengujikan pada method login, tambah pengguna, tambah surat keterangan, tambah surat pesanan, dan approve surat pesanan. Hasil dari pengujian Validasi didapatkan 100% valid.

6.3 Pengujian *Compatibility*

Pengujian *Compatibility* dilakukan dengan memasukkan *url website* sistem pada *website Sortsite PowerMapper*. Hasil pengujian *Compatibility* mendapatkan satu *major issues*, dan tiga *minor issues*. Hasil pengujian *Compatibility* akan ditampilkan pada Gambar 5

Browser	IE	Edge	Firefox	Safari	Opera	Chrome
Version	11	100	99	15	85	100
Critical Issues	🟢	🟢	🟢	🟢	🟢	🟢
Major Issues	🟡	🟢	🟢	🟢	🟢	🟢
Minor Issues	🟡	🟢	🟡	🟡	🟢	🟢

Gambar 5. Hasil Pengujian *Compatibility*

7. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang didapatkan pada penelitian akan mengacu pada bab pendahuluan pada bagian rumusan masalah. Kesimpulan pada penelitian ini antara lain:

1. Hasil analisis kebutuhan dibagial menjadi beberapa bagian yaitu: Gambaran Umum yang akan menjelaskan tentang sistem secara umum. Identifikasi Aktor yang akan menjelaskan pengguna yang menggunakan sistem. Identifikasi Kebutuhan Fungsional dan Kebutuhan Non-fungsional yang akan menjelaskan tentang fitur yang terdapat pada sistem. *Use Case Diagram* yang akan menjelaskan terkait hubungan aktor dengan sistem. Dan *Use Case Scenario* yang akan menjelaskan alur masing-masing kebutuhan fungsional sistem.
2. Hasil perancangan pengembangan sidsa mencakup Perancangan Sistem yang terbagi menjadi beberapa bagian, antara lain: Perancangan Arsitektur, Perancangan *Sequence Diagram*, Perancangan *Class Diagram*, Perancangan Algoritma, Perancangan Basis Data, dan Perancangan Antarmuka.
3. Hasil implementasi pengembangan sidsa akan terbagi menjadi beberapa bagian, antara lain: Spesifikasi Sistem, Implementasi Basis Data, Implementasi Kode Program, Implementasi Antarmuka.
4. Pada Pengujian Kebutuhan Fungsional yang telah dilakukan dengan Pengujian Unit dengan menggunakan 5 fitur sistem didapatkan nilai 100% valid, Pengujian Integrasi dengan menggunakan 2 fitur sistem didapatkan 100% valid, Pengujian Validasi dengan menggunakan 5 fitur sistem didapatkan 100% valid. Pada Pengujian Kebutuhan Non-Fungsional dengan Pengujian Validasi *Compatibility* didapatkan satu *Major Issues* pada browser *Internet Explorer* Versi 11, dan tiga

Minor Issues pada browser Internet Explorer versi 11, Firefox versi 99, Safari versi 15

8. SARAN

Terdapat beberapa saran yang dapat digunakan pada penelitian selanjutnya, yaitu:

1. Melakukan penambahan fitur notifikasi menggunakan whatsapp *API* apabila pengajuan surat telah selesai diproses dan dapat diambil pada kantor balaidesa. Sehingga penduduk mendapatkan informasi tersebut dari *whatsapp* dan bisa mengambil surat pada kantor balaidesa.
2. Membuat aplikasi berbasis *mobile* dengan tujuan penduduk bisa membuat pengajuan surat langsung dari ponsel. Selain itu penduduk juga bisa melihat proses pembuatan surat sudah sampai tahap apa. Apabila surat selesai dibuat dan penduduk sudah mendapatkan notifikasi, maka penduduk bisa *download* surat digital ataupun bisa mengambil di kantor balaidesa.
3. Membuat aplikasi pelayanan otomatis menggunakan *Monitor touchscreen* dan *e-ktp scanner* serta *printer*. Aplikasi ini dipergunakan untuk membantu pekerjaan bagian pelayanan. Jadi penduduk yang datang ke kantor balaidesa, bisa langsung *tap* *e-ktp* pada alat, setelah muncul biodata yang bersangkutan, penduduk bisa memilih surat apa yang mau dibuat, serta mengisikan keperluannya. Setelah mendapatkan persetujuan dari bagian pelayanan atau perangkat desa, maka surat langsung otomatis diprint pada alat tersebut.

9. DAFTAR PUSTAKA

- Sukamto, R. A., & Shalahuddin, M. (2018). *REKAYASA Perangkat Lunak: Terstruktur dan Berorientasi Objek*. BANDUNG: Informatika.
- Kurniawan, T. B., & Syarifuddin. (2020).

PERANCANGAN SISTEM APLIKASI PEMESANAN MAKANAN DAN MINUMAN PADA CAFETERIA NO CAFFE DI TANJUNG BALAI KARIMUN MENGGUNAKAN BAHASA PEMROGRAMAN PHP DAN MYSQL. *Jurnal TIKAR*, 2.

- Syahroni, A. W., & Ubaidi. (2019). Implementasi Darurat Keamanan Dan Kesehatan Berbasis Mobile Di Desa Waru Barat Pamekasan. *InfoTekJar :Jurnal Nasional Informatika dan Teknologi Jaringan*.
- Adenowo, A. A., & Adenowo, B. A. (2013). Software Engineering Methodologies: A Review of the Waterfall Model and Object- Oriented Approach. *International Journal of Scientific & Engineering Research*.