

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Adapun sistem aplikasi administrasi penduduk berbasis *website* ini dibangun dengan menggunakan referensi yang sudah dipelajari sebelumnya.

1. Akbar, R., Nainggolan, E., & Khasanah, S. (2019). Sistem Informasi Pelayanan Warga RW 01 Kelurahan Rawa Buaya Berbasis Website. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 2(3), 99-104.
2. Angelina, Y., & Safitri, G. (2018). Faktor Penentu Kepuasan Masyarakat Pada Pelayanan Kartu Tanda Penduduk Elektronik (E-Ktp) Di Kecamatan Pracimantoro, Kabupaten Wonogiri. *Jurnal Bisnis Terapan*, 2(1), 53-60.
3. Haswan, F. (2018). Perancangan Sistem Informasi Pendataan Penduduk Kelurahan Sungai Jering Berbasis Web Dengan Object Oriented Programming. *Jurnal Teknologi Dan Open Source*, 1(2), 92-100.
4. Kurniawan, T. (2018). Pemodelan Use Case (Uml): Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(1), 77-86.

Kesimpulan yang dapat diambil dari penelitian di atas yaitu, perancangan dan pengimplementasian sistem aplikasi administrasi penduduk Baloi Indah RW 006 berbasis *web* merupakan keperluan dari kemajuan teknologi yang wajib dipenuhi. Karena dengan adanya aplikasi sistem administrasi penduduk berbasis *web* ini, pengolahan data penduduk bisa diselesaikan dengan efektif serta Ketua RW atau RT bisa mendapatkan akses informasi yang akurat dan cepat dari manapun dan kapanpun.

Dengan adanya layanan pengolahan data administrasi penduduk, ketua RT atau RW dapat mengurus secara langsung urusan administrasi data penduduk maupun membuat surat pengantar yang diperlukan oleh penduduk tanpa harus membongkar berkas data penduduk melainkan hanya dengan membuka situs *website* aplikasi administrasi penduduk ketua RT atau RW dapat mengerjakan tugasnya ditempat itu juga, baik di kantor lurah atau dirumah RT atau RW itu sendiri.

Ketua RW dan RT Balai Indah RW 006 dapat mengetahui sebuah informasi mengenai penduduk Balai Indah RW 006 tanpa harus mendatangi kantor lurah, melainkan informasi yang bisa didapatkan secara langsung dari aplikasi *website* tersebut dari pc, laptop ataupun smartphone.

2.2 Landasan Teori

2.2.1 Metodologi Scrum

Menurut penelitian dari Aulia M., Hilmi A., & Ridwan S. (2019) pada penelitiannya yang memiliki judul “*Scrum Method Implementation in a Software Development Project Management*” menyatakan scrum pertama kali dibuat oleh bapak Jeff Sutherland di tahun 1993, dimana target utama dibuatnya scrum adalah membuat scrum menjadi sebuah metodologi penelitian yang memakai prinsip yang terdapat dari metodologi *Agile* sebagai panutan. Scrum merupakan kerangka kerja yang memiliki sifat responsif tambahan pada pengembangan dalam proyek *software* dan pengelolaan ataupun pengembangan sistem aplikasi. Tujuannya yaitu "menerapkan strategi pengembangan produk holistik yang memiliki sifat fleksibel di mana tim pengembang dapat bekerja sama sebagai tim untuk mencapai tujuan bersama" berlawanan dari "pendekatan tradisional urutan". Scrum mempunyai proses yang lumayan kompleks dimana beberapa faktor yang dapat merubah hasil keluaran dari perancangan sebuah sistem (Aulia M., Hilmi A., & Ridwan S., 2019).

Tahapan dalam scrum dapat dibagi menjadi sebagai berikut:

1. *Product Backlog.*

Peneliti pada sebuah sistem aplikasi mengumpulkan dan mengatur segala keperluan untuk sistem aplikasi dan juga permintaan dari pemakai sistem, contohnya fungsi yang diperlukan oleh pemakai pada sistem. Sesudah targetnya ditentukan, segala keperluan dan permintaan pengguna akan dibagi menjadi beberapa poin kecil dan tiap poin tersebut memiliki tingkatan yang dikembangkan.

2. *Sprint Planning*.

Sprint planning adalah suatu langkah wajib yang dilakukan ketika *sprint* baru dimulai. Pada langkah yang ini peneliti akan menyusun bagian pekerjaan yang wajib diselesaikan pada 1 bagian *sprint*.

3. *Task Breakout*.

Dalam tahapan ini, peneliti akan memberikan penjelasan mengenai pekerjaan apa saja yang nantinya akan dikembangkan waktu *sprint*.

4. *Sprint*.

Sprint adalah suatu batas waktu atau *deadline* yang mempunyai tugas pekerjaan yang telah dibagi menjadi beberapa bagian, dimana *sprint* tersebut akan mengejar *deadline* untuk menghasilkan target yang bisa diambil dari *Product Backlog*.

Menurut Aulia M., Hilmi A., & Ridwan S., (2019) metodologi pengembangan scrum mempunyai beberapa keunggulan seperti:

1. Scrum dapat memberikan kepuasan terhadap pelanggan karena waktu penyelesaiannya yang optimal dan responsif dengan permintaan dari pengguna.
2. Meningkatkan kualitas pada aplikasi.
3. Menerima dan mengharapkan adanya perubahan.
4. Memberikan prediksi yang lebih tepat dan memerlukan waktu yang lebih sedikit dalam tahap pengembangan.
5. Dapat mengatur penjadwalan pada proyek dengan mudah.
6. Scrum dapat dipakai untuk mengubah, dan menyelesaikan aplikasi dengan cepat sesuai permintaan.
7. Banyak keuntungan untuk pelanggan maupun proyek manajer.
8. Scrum dapat beradaptasi dengan cepat sesuai permintaan dengan mudah.
9. Jadwal yang jelas *sprint* demi *sprint*.
10. Prediksi ruang lingkup dalam mengerjakan fitur utama dan kemudian fitur biasa.
11. Tidak pernah ada perubahan jadwal atau *sprint* secara mendadak.

12. Mencocokkan ruang lingkup apabila diperlukan demi mengejar tanggal selesai.

13. Pekerjaan jauh lebih mudah dengan pembagian tugas dan kerja sama.

14. Pekerjaan dapat menghasilkan juga menyelesaikan banyak tugas dengan cara sistematis.

2.2.2 Definisi E-Government

Berdasarkan dari peraturan bagi presiden pada nomor 3 tahun 2003 mengenai kebijakan dan juga strategi nasional dalam pengembangan *E-Government* sebagai upaya untuk meningkatkan kinerja pelayanan pemerintah terhadap masyarakat dan membuat pemerintahan yang baik (*good governance*), banyak peneliti dalam bidang teknologi informasi telah membuat studi dan penelitian pada bidang *E-Government*. Makna dari *E-government* adalah semua proses yang dilakukan oleh pemerintah terhadap pengambilan segala macam kebijakan serta dalam menyediakan pelayanan terhadap masyarakat dijalankan dengan melalui teknologi informasi yang bersifat intensif. *E-government* merupakan penggunaan teknologi informasi yang mampu meningkatkan relasi dari pemerintah juga pihak-pihak lainnya. Pemakaian teknologi ini akan menghasilkan sebuah relasi dalam bentuk baru seperti *government to citizen*, *government to business*, *government to government*, dan *internal efficiency & effectiveness* (Eka P., & Muhamad M., 2019).

Kelebihan *E-Government*:

- a. Bisa membentuk hubungan interaktif pada dunia usaha untuk menangani perubahan atau persaingan yang ada di perdagangan internasional.
- b. Mampu membentuk sebuah jaringan informasi juga transaksi pelayanan masyarakat berkualitas, dan ekonomis bagi masyarakat.
- c. Mampu membentuk saluran komunikasi dengan lembaga-lembaga negara dan juga menyediakan fasilitas layanan dialog publik.
- d. Bisa membentuk sistem manajemen yang proses kerjanya transparan dan efisien dalam memberikan layanan yang lebih baik kepada masyarakat.

- e. Pemberdayaan masyarakat dikerjakan melalui informasi yang mudah untuk diakses dari mana saja dan kapan saja.
- f. Dengan adanya *E-Government*, pemerintahan akan dapat berjalan dengan lebih lancar karena koordinasi dari pemerintah bisa dilakukan dengan pemakaian teknologi informasi.

Kekurangan *E-Government*:

- a. Tradisi saling berbagi belum ada.
- b. Pegawai negara banyak yang memiliki pola pikir lama atau gagap teknologi baru.
- c. Jumlah server dan *software* berlisensi yang terbatas disebabkan oleh harga yang mahal.
- d. Sedikitnya sumber daya manusia yang mahir pada bidang TI di Indonesia.
- e. Belum ada integrasi database atau sistem aplikasi yang disatukan dan dipakai secara menyeluruh.
- f. Hardware yang tidak memadai atau komputer lama masih banyak dipakai karena tidak cukup dana membeli yang baru bagi seluruh Indonesia.

2.2.3 Administrasi Desa

Administrasi desa merupakan segala proses dari kegiatan untuk pencatatan data penduduk desa sampai dengan informasi tentang pelaksanaan pemerintahan daerah pedesaan dalam arsip administrasi desa. Administrasi pada desa ditentukan dengan keputusan dari menteri pada negara tetapi cara pengerjaannya dan kerja operasionalnya ditentukan oleh bupati. Berdasarkan dengan undang-undang pada nomor 32 tahun 2004 mengenai pemerintahan dalam daerah pedesaan, dan peraturan pemerintah pada nomor 72 tahun 2005 mengenai desa dan peraturan pada nomor 79 tahun 2005 mengenai pelaksanaan dan pemantauan pelaksanaan pemerintahan daerah, serta peraturan kementerian pada negeri 32 tahun 2006. Upaya dalam meningkatkan kinerja manajemen dalam pemerintahan desa harus dilakukan suatu penataan pada administrasi sehingga menjadi lebih efektif dan

efisien manajemen tersebut, penataan administrasi adalah pencatatan berbagai macam data serta informasi atau administrasi untuk mendukung pelaksanaan pemerintahan daerah pedesaan, sehingga diperlukan adanya langkah untuk menyempurnakan pada pelaksanaan dalam administrasi desa (Gita K., & Fiqih S., 2018).

2.2.4 Sistem Informasi

Suatu sistem informasi dirancang dengan memiliki suatu karakteristik tertentu, yaitu komponen, lingkungan luar sistem, sistem data, keluaran, pengolahan, masukan, dan penghubung, serta tujuan ataupun sasaran informasi merupakan suatu data yang telah diinterpretasikan atau diklarifikasikan sebagai suatu hal yang bermanfaat mengenai proses pengambilan keputusan. Kemudian sistem informasi mengolah data yang belum diproses menjadi suatu informasi yang bisa berguna untuk yang memerlukannya, data merupakan bentuk mentah yang dalam penggunaannya diperlukan proses yang lebih lanjut lagi agar menghasilkan sebuah informasi. Sebuah informasi dapat disebut berkualitas apabila dilandasi atas tiga hal yakni (Agustina Simangunsong, 2018):

1. Akurat

Informasi wajib berbentuk fakta yang telah terbukti kebenarannya. Informasi akurat merupakan informasi yang benar dan dibutuhkan. Kalau informasi tidak akurat atau dibutuhkan, informasi tersebut tidak bisa didefinisikan sebagai informasi tetapi rumor.

2. Tepat waktu

Waktu adalah sebuah kualitas dalam informasi yang tidak bisa ditinggalkan, jika informasi tidak tepat sesuai dengan waktunya, informasi tersebut tidak akan bisa diugunakan ataupun telah tidak lagi dibutuhkan.

3. Relevan

Informasi tidak ada hubungannya juga tidak dibutuhkan, apabila sebuah instansi atau aplikasi membutuhkan suatu informasi mengenai sebuah kendaraan tapi informasi yang muncul merupakan sebuah informasi tentang pakaian yang cantik tentu saja informasi tersebut tidak dapat

digunakan untuk perusahaan tersebut. Sehingga sebuah informasi wajib relevan sehingga bisa mempunyai fungsi dan arti.

Sistem informasi merupakan banyak komponen-komponen yang terkumpul dan saling ketergantungan satu dengan yang lainnya dan mampu bekerja sama untuk mengoleksi data, mengolah data, dan menyimpan informasi serta menghasilkan informasi yang relevan untuk mendukung langkah pengambilan sebuah keputusan, pengadilan atau suatu koordinasi (Agustina Simangunsong, 2018)

2.2.5 Web Service

Menurut penelitian dari Nofyat, Ibrahim, & Ambaria. (2018), definisi dari *Web Service* adalah sekumpulan halaman situs *website* yang telah dirilis pada internet dan mempunyai nama *domain* atau *url* yang bisa diakses secara global atau sedunia, internet juga dapat disebut dengan sebuah jaringan yang luas. Sama dengan jaringan *LAN* ataupun jaringan *WAN*, internet juga memakai *protocol* komunikasi yang sama yakni *TCP* atau *IP* (Nofyat, Ibrahim, & Ambaria, 2018).

2.2.6 HTML

HTML atau dipanggil dengan sebutan *Hypertext Markup Language* adalah bahasa pemrograman pada *web scripting* yang memiliki sifat *Client*, *Client* dapat menampilkan suatu informasi dengan memiliki bentuk sebuah teks, gambar atau multimedia dan juga dapat mengkoneksikan dari suatu tampilan *web page* ke *web page* lainnya dengan menggunakan fungsi *Hyperlink*. Untuk bisa memakai kode perintah bahasa pemrograman *HTML*, tidak diperlukan sebuah program khusus, pemakai bisa menggunakan *HTML* dengan *Notepad* atau *sublime* maupun teks *editor* lainnya yang mempunyai basis *GUI* seperti *Dreamweaver CS3*, dengan adanya program dengan basis *GUI* ini pemakai tidak perlu menginput kode *HTML* secara lengkap, terdapat beberapa fungsi umum yang sudah tersedia dalam *Icon Base* (Gita K., & Fiqih S., 2018).

2.2.7 PHP

PHP, yakni suatu bahasa pemrograman dengan bentuk *script server-side* yang sudah dirancang dengan maksud mengembangkan sebuah *website*. *PHP* bisa juga digunakan seperti bahasa pemrograman yang umum. pengembangan *PHP* dimulai sejak tahun 1995 oleh bapak Rasmus Lerdorf, sampai dengan saat ini *PHP* masih terus dikembangkan oleh *The PHP Group*. *PHP* juga disebut juga sebagai bahasa pemrograman dengan bentuk *server side* karena *PHP* bisa dimulai prosesnya dari komputer bagian *server*. *PHP* sangat berbeda apabila dibandingkan dengan suatu bahasa pemrograman lainnya yang memiliki bentuk *client-side*, contohnya bahasa pemrograman *JavaScript* yang dapat dipakai pada *web browser* (Eka P., & Muhamad M., 2019).

2.2.8 MySQL

MySQL merupakan suatu proses implementasi dalam sistem manajemen data dengan basis data yang memiliki sifat relasional yang sudah di rilis untuk umum secara gratis. Semua pengguna bisa memakai *MySQL* secara bebas, akan tetapi dengan satu syarat aplikasi *software* yang di buat dengan *MySQL* tersebut tidak boleh dirancang sebagai sebuah produk yang sifatnya komersial atau bertujuan untuk menghasilkan profit. *MySQL* merupakan turunan dari suatu konsep utama yang telah ada basis datanya sebelumnya. *SQL* adalah inti daripada konsep pengoperasian untuk basis data, dan umumnya digunakan untuk penyeleksian dan penginputan data, membuat pengoperasian data yang telah dijalankan dapat berkerja secara mudah dan otomatis (Gita K., & Fiqih S., 2018).

2.3 Perancangan Sistem

2.3.1 UML



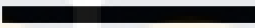



Menurut penelitian dari Tri A. K. (2018), *UML* merupakan suatu standar dari bahasa pemrograman yang telah umum dipakai pada dunia industri yang menentukan *requirement*, pembuatan analisis, desain, dan dengan menjelaskan arsitektur dari pemrograman yang berorientasi objek. *UML* adalah bahasa visual bagi pemodelan dan komunikasi tentang suatu sistem dengan memakai bermacam

diagram dan juga teks. Alat bantu yang dipakai untuk perancangan program aplikasi berorientasi object dengan basis UML yakni (Tri A. K., 2018):

1. Use Case Diagram

Menurut penelitian dari Tri A. K. (2018), *Use Case* adalah pemodelan mengenai keperluan sistem pada interaksi pemakai kepada sistem. *Use Case* dipakai juga pada validasi dan pengujian saat perancangan aplikasi. Tabel 2.2 berikut adalah tabel yang menguraikan simbol yang ada dalam *use case diagram* (Tri A. K., 2018).






Tabel 2.2 *Use Case Diagram*

No	Simbol	Keterangan
1		<i>User Case</i> ; aksi yang diharapkan atau bisa dilakukan dalam sebuah aplikasi.
2		<i>Actor</i> ; Satu peran koheren yang dipakai user untuk dapat berinteraksi dengan <i>use case</i> .
3		<i>Association</i> ; Penghubung aktor dengan <i>use case</i> yang menandakan keduanya mampu komunikasi dengan cara bertukaran pesan.
4		<i>Generalization</i> ; <i>use case</i> ini mampu mengambil turunan perilaku dan sifat parent <i>use case</i> .
5		<i>Extend</i> ; Simbol <i>Use case</i> yang eksplisit menggabungkan perilaku <i>use case</i> satu ke yang lainnya.
6		<i>Include</i> ; Simbol <i>Use case</i> yang eksplisit menggabungkan perilaku <i>use case</i> satu ke yang lainnya.

2. Activity Diagram

Menurut penelitian Anggia L. C., & Atik A.,(2018), *Activity Diagram* menjelaskan bermacam aliran dalam aktivitas dari sistem aplikasi yang sedang berkerja, bagaimanakah caranya setiap aliran mulai, *decision* yang dapat terjadi, juga bagaimanakah caranya aliran mereka berakhir. *Activity diagram* juga mampu menjelaskan proses aliran *parallel* yang dapat terjadi saat eksekusi suatu aksi. Tabel 2.3 berikut merupakan tabel yang akan menguraikan simbol-simbol yang terdapat dalam lambang *activity diagram*(Anggia L. C., & Atik A., 2018).






Tabel 2.3 Lambang *Activity Diagram*

No	Simbol	Keterangan
1		Simpul awal, simbol yang menandakan mulai nya proses
2		Aliran kontrol, menunjukkan arah <i>workflow</i> atau penghubung antara <i>activity</i> .
3		<i>Activity</i> , tugas yang berjalan pada keseluruhan proses.
4		<i>Decision</i> , cabang keputusan yang memiliki pilihan jalur alternatif yang ada.
5		Simpul akhir, symbol yang menandakan akhir dari suatu proses.

3. Sequence Diagram

Menurut penelitian dari Anggia L. C., & Atik A., (2018), *Sequence diagram* dipakai dalam penguraian *scenario* dengan bentuk gambar secara sequence. *Diagram* yang satu ini menggambarkan contoh sebuah objek dan pesan yang akan ditaruh di berbagai macam objek yang ada dalam *use case diagram*. Tabel 2.4 berikut merupakan tabel yang menguraikan simbol - simbol yang ada dalam *sequence diagram*(Anggia L. C., & Atik A., 2018).




Tabel 2.4 Lambang *Sequence Diagram*



No	Simbol	Keterangan
1		<i>Actor Lifeline</i> , aktor yang berinteraksi secara langsung pada sistem
2		<i>Object Lifeline</i> , objek yang berhubungan secara langsung dan mampu bertukar an pesan.
3		<i>Activation</i> , aksi dalam pengiriman atau penerimaan pesan.
4		<i>Message</i> , pesan yang dikirimkan untuk <i>activation</i> ke arah yang ditunjukkan
5		<i>Message Return</i> , pesan yang dikembalikan dari <i>activation</i> ke arah yang ditunjukkan.

4. Flowchart

Menurut penelitian dari Hidayat (2018), makna dari *Flowchart* adalah berbagai macam simbol yang akan menggambarkan serangkaian proses alur program, yang dimulai dari awal berjalanya program sampai dengan akhir alur dari program tersebut. *Flowchart* dapat juga dipakai dalam penggambaran dari urutan langkah suatu pekerjaan dalam sebuah algoritma. Adapun pengertian dari simbol-simbol *flowchart* dapat dilihat dari Tabel 2.5 berikut. (Hidayat, 2018).

Tabel 2.5 Lambang *Flowchart*







No	Simbol	Keterangan
1		<i>Terminator</i> , berfungsi untuk memulai juga mengakhiri sebuah proses.
2		<i>Process</i> , berfungsi untuk menunjukkan sebuah pengolahan data yang dijalankan oleh sistem.
3		<i>Data</i> , berfungsi untuk menginput data ataupun mengeluarkan hasil dari sebuah proses.

4		<i>Connector</i> , adalah simbol untuk masuk juga keluarnya sebuah prosedur dalam halaman yang sama.
5		<i>Off-page Connector</i> , adalah simbol untuk masuk juga keluarnya sebuah prosedur pada halaman yang berbeda.

2.3.2 Entity Relationship Diagram (ERD)

Dari penelitian dari Anggia L. C., & Atik A.,(2018), dijelaskan bahwa *ERD* adalah sebuah model jaringan yang memakai susunan dari data yang telah disimpan dari sistem acak. *ERD* dipakai untuk memperlihatkan data objek beserta dengan berbagai hubungan yang terdapat di objek tersebut dengan memakai simbol *entity* dan juga simbol *relationship* yang dibuat pertama kali oleh Chen di tahun 1976. Tabel 2.6 berikut merupakan tabel yang menguraikan simbol- simbol yang ada dari *ERD* (Anggia L. C., & Atik A., 2018).

Tabel 2.6 *Entity Relationship Diagram*

No	Simbol	Keterangan
1		<i>Entitas</i> ; Entitas merupakan suatu kesatuan objek lain, dimana tiap entitas dibatasi dengan atribut.
2		<i>Atribut</i> ; Atribut adalah sifat atau ciri dari sebuah entitas yang emberikan penjelasan secara detail.
3		<i>Relationship</i> ; berfungsi untuk menyatakan himpunan relasi.
4		<i>Document</i> ; <i>Document</i> sebagai penanda dalam suatu dokumen ataupun laporan.
5		<i>Manual Input</i> ; sebuah proses yang dimana pemakai menginputkan informasi dengan cara <i>manual</i> .
6		<i>Database</i> ; Simbol yang menyatakan basis data.