**RANCANG BANGUN SISTEM MANAJEMEN PENDATAAN PENGADAAN BANTUAN UNTUK KELOMPOK TANI BERBASIS WEB MENGGUNAKAN METODE *PERSONAL EXTREME PROGRAMMING* (PXP)**

**(Studi Kasus : Dinas Pertanian Kabupaten Toba)**

**TUGAS AKHIR**

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1) di Program Studi Teknik Informatika, Jurusan Teknologi, Produksi dan Industri, Institut Teknologi Sumatera

Oleh:

**ACKYRA A. M. SIBARANI**

**118140160**

****

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI, PRODUKSI DAN INDUSTRI**

**INSTITUT TEKNOLOGI SUMATERA**

**LAMPUNG SELATAN**

**2022**

# LEMBAR PENGESAHAN

Tugas Akhir dengan judul “Tulis Judul Disini” adalah benar dibuat oleh saya sendiri dan belum pernah dibuat dan diserahkan sebelumnya, baik sebagian ataupun seluruhnya, baik oleh saya ataupun orang lain, baik di Institut Teknologi Sumatera maupun di institusi pendidikan lainnya.

|  |  |
| --- | --- |
| Lampung Selatan, DD-MM-YYYY  Penulis, | PHOTO BERWARNA |
| Nama Mahasiswa  NIM. XXXXXX |  |

Diperiksa dan disetujui oleh,

Pembimbing Tanda Tangan

1. Nama Pembimbing 1 + Gelar

NIP. XXXXXX ………………

2. Nama Pembimbing 2 + Gelar

NIP. XXXXXX ………………

Penguji Tanda Tangan

1. Nama Penguji 1 + Gelar

NIP. XXXXXXXXXXXX ………………

2. Nama Penguji 2+ Gelar

NIP. XXXXXXXXXXXX ………………

Disahkan oleh,

Koordinator Program Studi Teknik Informatika

Jurusan Teknologi, Produksi dan Industri

Institut Teknologi Sumatera

Nama Kaprodi + Gelar

NIP. XXXXXXXXXXXXXX

# HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir dengan judul “TULIS JUDUL DISINI” adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.**

**Nama : …………………………**

**NIM : ………………………….**

**Tanda Tangan : ………………………….**

**Tanggal : ………………………….**

# HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : ………………………..

NIM : ………………………..

Program Studi : Teknik Informatika

Jurusan : Jurusan Teknologi, Produksi dan Industri

Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

**TULIS JUDUL DISINI**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini, Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Lampung Selatan

Pada tanggal DD Bulan YYYY

Yang menyatakan,

Nama Mahasiswa

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penyusunan tugas akhir ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapan terima kasih kepada:

1. <isi dengan nama Rektor ITERA>
2. <isi dengan nama Kajur JTPI>
3. <isi dengan nama Kaprodi IF>
4. <isi dengan nama Sesprodi IF>
5. <isi dengan nama Koordinator TA>
6. <isi dengan nama Dosen Pembimbing>
7. Kedua Orang Tua, kakak dan adik yang selalu memberikan arahan selama belajar dan menyelesaikan tugas akhir ini.
8. <isi dengan nama orang lainnya>

Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi kita semua, amin. [Contoh]

# RINGKASAN

Judul TA

Nama Mahasiswa

Halaman Ringkasan berisi uraian singkat tentang latar belakang masalah, rumusan masalah, tujuan, metodologi penelitian, hasil dan analisis data, serta kesimpulan dan saran. Isi ringkasan tidak lebih dari 1500 kata (sekitar 3 halaman).

# ABSTRAK

Judul TA

Nama Mahasiswa

Halaman ABSTRAK berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INDONESIA tidak lebih dari 250 kata, dengan jarak antar baris satu spasi.

Pada akhir abstrak ditulis kata “Kata Kunci” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Kata kunci terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Kata kunci diketik pada jarak dua spasi dari baris akhir isi abstrak.

**Kata Kunci : Penambangan Data, Kecerdasan Buatan, Lampung Selatan**

# ABSTRACT

Judul TA (Bahasa Inggris)

Nama Mahasiswa

Halaman ABSTRACT berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INGGRIS tidak lebih dari 250 kata, dengan jarak antar baris satu spasi. Secara khusus, kata dan kalimat pada halaman ini tidak perlu ditulis dengan huruf miring meskipun menggunakan Bahasa Inggris, kecuali terdapat huruf asing lain yang ditulis dengan huruf miring (misalnya huruf Latin atau Greek, dll).

Pada akhir abstract ditulis kata “Keywords” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Keywords terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Keywords diketik pada jarak dua spasi dari baris akhir isi abstrak.

**Keywords : Data Mining, Artificial Intelligence, Lampung Selatan**

# DAFTAR ISI

[LEMBAR PENGESAHAN 2](#_Toc152684718)

[HALAMAN PERNYATAAN ORISINALITAS 3](#_Toc152684719)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS 4](#_Toc152684720)

[KATA PENGANTAR 5](#_Toc152684721)

[RINGKASAN 6](#_Toc152684722)

[ABSTRAK 7](#_Toc152684723)

[ABSTRACT 8](#_Toc152684724)

[DAFTAR ISI 9](#_Toc152684725)

[DAFTAR TABEL 12](#_Toc152684726)

[DAFTAR GAMBAR 13](#_Toc152684727)

[1 BAB I PENDAHULUAN 16](#_Toc152684728)

[1.1 Latar Belakang 16](#_Toc152684729)

[1.2 Rumusan Masalah 19](#_Toc152684730)

[1.3 Tujuan Penelitian 19](#_Toc152684731)

[1.4 Batasan Masalah 19](#_Toc152684732)

[1.5 Manfaat Penelitian 19](#_Toc152684733)

[1.6 Sistematika Penulisan 20](#_Toc152684734)

[2 BAB II TINJAUAN PUSTAKA 21](#_Toc152684735)

[2.1 Tinjauan Pustaka 21](#_Toc152684738)

[2.2 Dasar Teori 24](#_Toc152684739)

[2.2.1 Website 24](#_Toc152684740)

[2.2.2 Sistem Informasi 24](#_Toc152684741)

[2.2.3 Sistem Informasi Manajemen 25](#_Toc152684742)

[2.2.4 Agile Software Development Method 25](#_Toc152684743)

[2.2.5 Personal Extreme Programming 26](#_Toc152684744)

[2.2.6 Perancangan Sistem 29](#_Toc152684745)

[2.2.7 Database 30](#_Toc152684746)

[2.2.8 Entity Relationship Diagram (ERD) 31](#_Toc152684747)

[2.2.9 Framework 32](#_Toc152684748)

[2.2.10 Pengujian User Acceptance Test (UAT) 33](#_Toc152684749)

[3 BAB III METODE PENELITIAN 18](#_Toc152684750)

[3.1 Alur Penelitian 18](#_Toc152684751)

[3.2 Penjabaran Langkah Penelitian 19](#_Toc152684752)

[3.2.1 Studi Literatur 19](#_Toc152684753)

[3.2.2 Analisis Kebutuhan 19](#_Toc152684754)

[3.2.3 Perencanaan 20](#_Toc152684755)

[3.2.4 Inisialisasi Iterasi 20](#_Toc152684756)

[3.2.5 Perancangan 20](#_Toc152684757)

[3.2.6 Implementasi 21](#_Toc152684758)

[3.2.7 Pengujian Sistem 21](#_Toc152684759)

[3.2.8 Retrospektif 21](#_Toc152684760)

[3.2.9 Penulisan Laporan Akhir 21](#_Toc152684761)

[3.3 Alat dan Bahan Tugas Akhir 22](#_Toc152684762)

[3.3.1 Alat 22](#_Toc152684763)

[3.3.2 Bahan 22](#_Toc152684764)

[3.4 Metode Tugas Akhir 23](#_Toc152684765)

[3.4.1 Analisis Kebutuhan 23](#_Toc152684766)

[3.4.2 Tahap Perencanaan 25](#_Toc152684767)

[3.4.3 Tahap Inisiasi Iterasi 31](#_Toc152684768)

[3.4.4 Tahap Perancangan 33](#_Toc152684769)

[3.4.5 Tahap Implementasi 47](#_Toc152684770)

[3.4.6 Tahap Retrospektif 48](#_Toc152684771)

[3.4.7 Tahap pengujian sistem 48](#_Toc152684772)

[4 BAB IV HASIL DAN PEMBAHASAN 49](#_Toc152684773)

[4.1 Lingkungan Pembangunan Sistem 49](#_Toc152684774)

[4.2 Personal Extreme Programming 49](#_Toc152684775)

[4.2.1 Requirement 49](#_Toc152684776)

[4.2.2 Planning 49](#_Toc152684777)

[4.2.3 Iteration Initialization 50](#_Toc152684778)

[4.2.4 Design 50](#_Toc152684779)

[4.2.5 Iterasi ke Satu 50](#_Toc152684780)

[4.2.6 Iterasi ke Dua 60](#_Toc152684781)

[5 BAB V KESIMPULAN DAN SARAN 69](#_Toc152684782)

[a. Kesimpulan 69](#_Toc152684783)

[b. Saran 69](#_Toc152684784)

# DAFTAR TABEL

[Tabel 3.1 Pengguna sistem 23](#_Toc129678969)

[Tabel 3.2 User Stories 24](#_Toc129678970)

[Tabel 3.3 Kebutuhan Fungsional 24](#_Toc129678971)

[Tabel 3 4 Kebutuhan Non-fungsional 25](#_Toc129678972)

[Tabel 3.5 Estimasi Pengerjaan 26](#_Toc129678973)

[Tabel 3.6 Prioritas user Stories 29](#_Toc129678974)

[Tabel 3 7 Iterasi 32](#_Toc129678975)

# DAFTAR GAMBAR

[Gambar 3.1 Diagram alir penelitian 18](file:///C:\Users\User\Downloads\edit%20TA%20Ackyra.docx#_Toc129678952)

[Gambar 3.2 Struktural Dinas Pertanian Kabupaten Toba 20](#_Toc129678953)

[Gambar 3 3 Use case diagram iterasi 1 34](#_Toc129678954)

[Gambar 3.4 Use case diagram iterasi 2 35](#_Toc129678955)

[Gambar 3.5 Use case diagram iterasi 3 36](#_Toc129678956)

[Gambar 3.6 Use case diagram iterasi 4 37](#_Toc129678957)

[Gambar 3.7 Activity diagram manajemen user 38](#_Toc129678958)

[Gambar 3.8 Activity diagram manajemen kecamatan 39](#_Toc129678959)

[Gambar 3.9 Activity diagram manajemen bidang pertanian 40](#_Toc129678960)

[Gambar 3.10 Activity diagram manajemen kamus usulan 41](#_Toc129678961)

[Gambar 3.11 Activity diagram manajemen kelompok tani 42](#_Toc129678962)

[Gambar 3.12 Activity diagram pengajuan proposal 43](#_Toc129678963)

[Gambar 3.13 ER Diagram 44](#_Toc129678964)

[Gambar 3.14 Tampilan home 45](#_Toc129678965)

[Gambar 3.15 Tampilan beranda 46](#_Toc129678966)

[Gambar 3.16 Tampilan login 46](#_Toc129678967)

[Gambar 3.17 Tampilan manajemen user 47](#_Toc129678968)

**DAFTAR RUMUS**

**DAFTAR LAMPIRAN**

# BAB I PENDAHULUAN

## Latar Belakang

Perkembangan teknologi hingga pada saat ini banyak memberikan dampak positif yang dapat dirasakan oleh manusia. Perkembangan teknologi ini membantu manusia di banyak bidang yang memudahkan manusia sendiri [1]. Penerapan teknologi informasi saat ini dapat digunakan di banyak bidang. Salah satu penerapannya adalah perancangan dan pengembangan sebuah sistem informasi yang dapat digunakan oleh sebuah instansi pemerintah untuk berhubungan dengan masyarakat. Hubungan atau interaksi yang dapat dilakukan pemerintah dengan masyarakat adalah melalui pelayanan umum [2]. Pelayanan umum yang diberikan kepada masyarakat dapat dituangkan dalam bentuk pembuatan sistem informasi untuk mengelola dan mengoptimalkan penyaluran bantuan bibit kepada kelompok tani. Sistem informasi ini akan berperan penting bagi masyarakat karena dapat membantu penerimaan bantuan secara merata dan pengajuan bantuan dari masyarakat akan mudah disampaikan. Kelompok tani juga akan mendapatkan kesempatan yang sama dalam mendapatkan bantuan dari pemerintah setempat melalui sistem informasi ini. *E-Government* merupakan istilah yang populer dari lembaga pemerintahan. *E-Government* dibutuhkan di lembaga pemerintahan untuk mewujudkan pemerintahan yang efektif dan efisien serta memberikan layanan yang lebih baik [3].

Kabupaten Toba merupakan kabupaten yang terletak di Provinsi Sumatera Utara. Kabupaten Toba merupakan hasil pemekaran dari kabupaten Daerah tingkat II Tapanuli Utara dan diresmikan pada tanggal 9 Maret 1999 [4]. Kabupaten Toba dipimpin oleh seorang Bupati yang dibantu oleh jajaran-jajarannya. Bupati dan jajarannya berwenang memberikan pembinaan dan pelayanan kepada masyarakat Kabupaten Toba. Pelayanan yang diterima masyarakat Toba salah satunya melalui Dinas Pertanian Kabupaten Toba. Pelayanan yang diberikan Dinas Pertanian hingga saat ini adalah penyediaan benih di bidang tanaman pangan dan hortikultura, penyediaan bibit/benih ternak, penyediaan dan bimbingan penggunaan pupuk dan pestisida [5]. Salah satu cara untuk mewujudkan pelayanan yang maksimal dari Dinas Pertanian Kabupaten Toba ke masyarakat Toba adalah dengan membentuk kelompok tani. Kelompok tani ini merupakan sekelompok petani yang secara informal bersatu untuk mencapai tujuan berdasarkan kepentingan bersama dalam berusaha tani [6]. Pelayanan berupa program bantuan Dinas Pertanian akan disalurkan melalui kelompok tani. Pelayanan yang baik dapat diperoleh dengan adanya kerja sama yang baik antara masyarakat Kabupaten Toba dengan Pemerintah Kabupaten Toba.

Berdasarkan hasil diskusi dengan Kepala Sub Program Dinas Pertanian Kabupaten Toba, Ibu Nora R. Q. Hutabarat pada tanggal 04 Agustus 2022 dan hasil wawancara yang dituangkan dalam surat Permohonan Penelitian Tugas Akhir Nomor 520/555/DP/Sekret 2022 (dapat dilihat pada lampiran 1) menerangkan bahwa dalam melakukan pendataan pengadaan bantuan untuk kelompok tani masih dilakukan dengan pemberkasan manual. Alur pengajuan yang dilakukan dalam pengadaan bantuan tani adalah kelompok tani akan memberikan proposal pengajuan bantuan kepada badan penyuluh yang bertugas di Desa. Badan penyuluh dari setiap desa akan mengumpulkan semua berkas proposal ke Kecamatan. Data proposal yang sudah terkumpul di kecamatan akan diajukan oleh Koordinator badan penyuluh kepada pihak Dinas Pertanian. Kepala Bidang Dinas Pertanian akan melakukan pemeriksaan terhadap proposal yang diajukan tersebut lalu dilakukan penyusunan pengalokasian pengadaan bantuan tani yang bisa diberikan. Data pengalokasian yang sudah disusun akan diserahkan kepada Kepala Dinas untuk menunggu persetujuan Kepala Dinas dalam hal pelaksanaan pengadaan bantuan tani tersebut. Setelah mendapatkan persetujuan dari Kepala Dinas, maka Kepala Bidang akan melanjutkan proses pengadaan bantuan tani kepada proposal kelompok tani yang sudah dialokasikan. Semua proposal yang pernah diajukan akan disimpan oleh pihak Dinas Pertanian sehingga proposal yang belum dilaksanakan atau sudah dilaksanakan namun belum sepenuhnya selesai dapat diadakan kembali pada program pengadaan bantuan tani periode berikutnya.

Alur pengadaan bantuan tani saat ini memiliki masalah seperti data yang disimpan oleh pihak Dinas Pertanian tidak terintegrasi dengan baik dengan data yang dimiliki oleh Koordinator Badan Penyuluh sehingga menyebabkan adanya duplikasi data proposal pengadaan bantuan tani di program pengadaan bantuan tani yang baru. Duplikasi data proposal ini dapat mengakibatkan bantuan tani sulit dialokasikan oleh Kepala Bidang. Permasalahan lainnya adalah penyimpanan data yang dilakukan oleh pihak Dinas Pertanian rentan tidak terdata dengan baik karena masih bersifat konvensional sehingga rentan menyebabkan kehilangan data. Permasalahan tersebut diakibatkan oleh tidak terintegrasinya sistem pendataan yang diterapkan sehingga adanya pengadaan bantuan tani yang tidak merata bagi kelompok tani.

Permasalahan yang sudah dijabarkan menjadi kendala dalam penyaluran program bantuan tani sehingga Dinas Pertanian membutuhkan sebuah sistem pendataan yang dapat membantu Dinas Pertanian dalam mendata pengadaan bantuan tani untuk kelompok tani di Kabupaten Toba. Solusi yang ditawarkan adalah dengan membuat sebuah sistem informasi yang dapat mendata proposal bantuan tani dan mendata hasil pengadaan bantuan tani dengan baik dan terintegrasi bagi semua pihak yang terlibat, Sistem pendataan yang terintegrasi dengan baik membuat pengadaan bantuan kelompok tani bisa menghindari adanya tumpang tindih pengadaan bantuan tani, ketidakmerataan pengadaan bantuan tani, dan melancarkan pelaksanaan pengadaan bantuan tani.

Sistem informasi pendataan akan dikembangkan menggunakan metode *agile*. Sistem ini dibangun berbasis *website*. *Website* merupakan halaman situs yang dapat memberikan informasi berupa gambar maupun data yang akan memudahkan si pengguna [7]. Metode yang digunakan adalah metode *agile. Agile* merupakan metode pengembang yang memerlukan adaptasi yang cepat dari pengembang terhadap perubahan apa pun karena metode *agile* merupakan jenis pengembangan dalam jangka pendek [8]. Metode pengembangan *agile* juga merupakan metode yang memprioritaskan kepuasan dan keinginan kliensehingga dibutuhkan komunikasi yang baik antara pengembang dan klien [9].

Model *agile* yang digunakan peneliti dalam penelitian ini adalah *Personal Extreme Programming* (PXP) yang merupakan turunan dari metode *Extreme Programming* (XP). *Personal Extreme Programming* (PXP) merupakan model pengembangan yang dilakukan oleh pengembang tunggal [10]. Metode pengembangan ini bersifat lebih fleksibel dan lebih adaptif karena menyederhanakan setiap tahap dari proses pengembangan dan dapat dikembangkan oleh peneliti tanpa memiliki tim sehingga metode *Personal Extreme Programming* (PXP) merupakan metode yang tepat untuk digunakan dalam penelitian ini. *Personal Extreme Programming* (PXP) memiliki kerangka kerja *requirements, planning, iteration initialization, design, implementation, system testing,* dan *retrospective* [10]*.* Metode *agile* dengan model pengembangan *Personal Extreme Programming* (PXP) dinilai tepat dalam penelitian ini karena sesuai dengan kebutuhan pengembangan sistem manajemen pendataan pengadaan bantuan ini, yang membutuhkan metode pengembangan yang adaptif dan fleksibel.

Sistem manajemen informasi pengadaan bantuan tani yang akan dibangun membutuhkan pengujian guna untuk mengukur fungsionalitas. Pengujian fungsionalitas sistem yang dilakukan menggunakan *User Acceptance Test* untuk mengetahui apakah fungsi fitur yang dikembangkan sudah sesuai dengan kebutuhan yang telah ditentukan pada awal pengembangan. Pengujian UAT ini dilakukan oleh klien (Dinas Pertanian Kabupaten Toba). Pengujian *User Acceptance Test* merupakan pengujian yang dilakukan pada fase terakhir. Pengujian ini dilakukan untuk memastikan fitur-fitur yang dikembangkan sudah berjalan sesuai dengan yang diharapkan dan sesuai dengan kebutuhan klien. Pengujian *User Acceptance Test* dilakukan langsung oleh klien [11].

Sistem manajemen ini akan membantu pekerjaan Dinas Pertanian untuk memanajemen pendataan kelompok tani yang mengajukan bantuan, sehingga hasil yang diharapkan adalah bantuan tani dapat diberikan secara merata kepada seluruh kelompok tani yang ada. Sistem informasi ini juga akan membantu kelompok tani dalam mengajukan dan memantau kemajuan proses bantuan yang diajukan.

## Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan dan diuraikan, maka dapat rumusan masalah yang dapat diambil dalam penelitian ini adalah:

1. Bagaimana Rancang Bangun Sistem Manajemen Pendataan Bantuan untuk Kelompok Tani pada Dinas Pertanian Kabupaten Toba menggunakan metode *Personal Extreme Programming* (PXP)?
2. Bagaimana fungsionalitas system menggunakan *User Acceptance Test* yang dibangun agar dapat digunakan?

## Tujuan Penelitian

Tujuan dari dilakukannya penelitian ini adalah:

1. Merancang dan membangun sistem informasi pendataan pengadaan bantuan tani Kabupaten Toba menggunakan metode *Personal Extreme Programming* (PXP).
2. Melakukan pengujian terhadap sistem yang dirancang dan dibangun menggunakan *User Acceptance Test* untuk mengetahui fungsionalitas sistem.

## Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Pengadaan bantuan tani yang didata hanyalah yang terdaftar di kamus usulan pengadaan bantuan tani Dinas Pertanian Kabupaten Toba Tahun 2021-2026.
2. Pengguna utama sistem adalah pegawai dinas pertanian yaitu kepala dinas, kepala bidang, dan badan penyuluh.

## Manfaat Penelitian

Manfaat penelitian yang hendak dicapai dari penelitian yang dilakukan adalah sebagai berikut:

1. Bagi Dinas Pertanian Kabupaten Toba
   1. Membantu Dinas Pertanian dalam mendata kelompok tani yang ada.
   2. Membantu Dinas Pertanian dalam manajemen penyebaran bantuan tani kepada kelompok tani.
   3. Meningkatkan sarana penyaluran bantuan tani kepada kelompok tani di Kabupaten Toba.
2. Bagi Kelompok Tani
3. Membantu kelompok tani untuk mengetahui status pengajuan program bantuan yang diajukan.

## Sistematika Penulisan

Sistematika penulisan laporan Tugas akhir ini, maka gambaran singkat tentang penyusunan laporannya adalah sebagai berikut:

1.6.1 BAB I Pendahuluan

Bab I ini menguraikan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian serta sistematika penelitian.

1.6.2 BAB II Tinjauan Pustaka

Bab II ini membahas uraian tinjauan pustaka dari penelitian terkait yang digunakan sebagai acuan untuk merancang dan membangun sistem informasi hingga penyusunan laporan tugas akhir.

1.6.3 BAB III Metode Penelitian

Bab III ini membahas metode yang digunakan dalam perancangan dan pembangunan sistem yang akan dibuat serta menggambarkan bagaimana proses dilakukan dalam penelitian.

1.6.4 BAB IV Hasil Penelitian dan Pembahasan

Bab IV ini berisi hasil implementasi dan pengujian terhadap pengembangan sistem informasi dengan metodologi yang digunakan.

1.6.5 BAB V Kesimpulan dan Saran

Bab V ini berisi kesimpulan dan saran kepada pengembangan selanjutnya yang ingin melakukan pengembangan terhadap sistem informasi penyaluran bantuan kelompok tani ini.

# BAB II TINJAUAN PUSTAKA



## Tinjauan Pustaka

Penelitian yang akan dilakukan tidak terlepas dari penelitian-penelitian sebelumnya. Penelitian sebelumnya dijadikan penulis sebagai referensi dan bahan perbandingan serta kajian untuk penelitian yang akan dilakukan. Berikut penelitian terdahulu yang terkait dengan penelitian yang akan dilakukan oleh penulis:

1. Pada tahun 2021, Ida Siti Marfuah dan Yudie Irawan melakukan penelitian yang berjudul Sistem Informasi Penyaluran Bantuan Pupuk Bersubsidi pada Dinas Pertanian dan Pangan Kabupaten Rembang berbasis Web. Penelitian ini bertujuan membuat sistem yang dapat memanfaatkan komputer untuk mengelola pendistribusian pupuk bersubsidi dari pengawasan hingga penyaluran ke masyarakat. Penelitian ini menggunakan metode berorientasi objekdan menggunakan *Unified Modelling Language* (UML). Hasil dari penelitian ini adalah sistem informasi penyaluran bantuan pupuk bersubsidi pada Dinas Pertanian dan Pangan Kabupaten Rembang [12].
2. Pada tahun 2022, Titis Ulfa Mustikawati melakukan penelitian yang berjudul Sistem Informasi Pengelolaan Bantuan Desa Banjardowo. Penelitian ini bertujuan membuat sistem yang mampu mengelola data secara efektif, sehingga memberikan kemudahan kepada petugas dalam mengelola bantuan di Desa Banjardowo. Penelitian ini menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) dan *database* *MySql*. Pengujian yang dilakukan dalam penelitian ini adalah pengujian *blackbox-testing*. Hasil dari penelitian ini adalah pengelolaan data bantuan masyarakat di Desa Banjardowo menjadi lebih efektif [13].
3. Pada tahun 2020, Samsuriati melakukan penelitian yang berjudul Sistem Informasi Pengadaan dan Pengalokasian Pupuk pada Dinas Pertanian, Peternakan dan Perkebunan Kabupaten Majene Berbasis Web. Penelitian ini bertujuan untuk mengolah data - data yang berhubungan dengan pengadaan dan pengalokasian pupuk yang akan disebarkan ke penduduk setempat. Penelitian ini menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP), bahasa pemrograman HTML, dan *database* *MySql*. Hasil dari penelitian ini adalah menghasilkan Sistem Informasi Pengadaan dan Pengalokasian Pupuk pada Dinas Pertanian Majene berbasis *Web* [14]*.*
4. Pada tahun 2020, Indah Purnama Sari melakukan penelitian yang berjudul Sistem Informasi Permohonan Pengajuan Bantuan Bibit Perkebunan Pada Dinas Pertanian Dan Pangan Kabupaten Kudus. Penelitian ini bertujuan membantu proses pengajuan bantuan bibit perkebunan yang masih konvensional menjadi terkomputerisasi. Penelitian ini menggunakan metode *waterfall* dan perancangannya menggunakan UML (*Unified Modelling Languange*) Hasil dari penelitian ini adalah sistem informasi pengajuan bantuan bibit perkebunan di kabupaten kudus [15].

| No | Judul | Peneliti | Tahun | Metode | Hasil | Perbedaan dengan penelitian yang dilakukan |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Sistem Informasi Penyaluran Bantuan Pupuk Bersubsidi pada Dinas Pertanian dan Pangan Kabupaten Rembang berbasis *Web*. | Ida Siti Marfuah dan Yudi Irawan | 2021 | *Unified Modelling Language* (UML) | Sistem informasi yang dapat membantu penyaluran bantuan pupuk bersubsidi. Sistem informasi ini memanfaatkan komputer untuk mengelola pendistribusian pupuk bersubsidi dari pengawasan hingga penyaluran kemasyarakat. bersubsidi dari pengawasan hingga penyaluran ke masyarakat | Sistem informasi yang akan dibangun menggunakan metode *Personal Extreme Programming* (PXP) dan fitur yang akan dibuat berbeda dengan fitur yang ada dalam penelitian ini. |
| 2 | Sistem Informasi Pengelolaan Bantuan Desa Banjardowo | Titis Ulfa Mustikawati | 2022 | *Hypertext Preprocessor* (PHP), *MySql* | Pengelolaan data bantuan di Desa Banjardowo menjadi lebih efektif dimana petugas yang bekerja di bagian pengelolaan bantuan tidak mengguna-kan sistem manual lagi. | Sistem informasi yang akan dibangun menggunakan *Personal Extreme Programming* (PXP) dan memberikan pelayanan pengadaan bantuan tani di Kabupaten Toba. |
| 3 | Sistem Informasi Pengadaan dan Pengalokasian Pupuk pada Dinas Pertanian, Peternakan dan Perkebunan Kabupaten Majene Berbasis *Web*. | Samsuriati | 2020 | *Hypertext Preprocessor* (PHP), HTML, *MySql*. | Sistem Informasi pengadaan dan pengalokasian pupuk dan Dinas Pertanian Majene berbasis Web. Sistem informasi ini akan digunakan untuk mengelola data-data yang berhubungan dengan pengadaan dan pengalokasian pupuk. | Sistem informasi yang akan dibangun menggunakan metode *Personal Extreme Programming* (PXP) |
| 4 | Sistem Informasi Permoho-nan Pengajuan Bantuan Bibit Perkebunan Pada Dinas Pertanian Dan Pangan Kabupaten Kudus | Indah Purna-ma Sari | 2020 | *Unified Modelling Language* (UML) | Sistem informasi yang dihasilkan dari penelitian ini membantu proses pengajuan bantuan bibit perkebunan yang masih konvensional pada Dinas Pertanian dan Pangan Kabupaten Kudus. | Sistem informasi yang akan dibangun menggunakan metode *Personal Extreme Programming* (PXP) |

**Tabel 2.1 Ringkasan Penelitian Terdahulu**

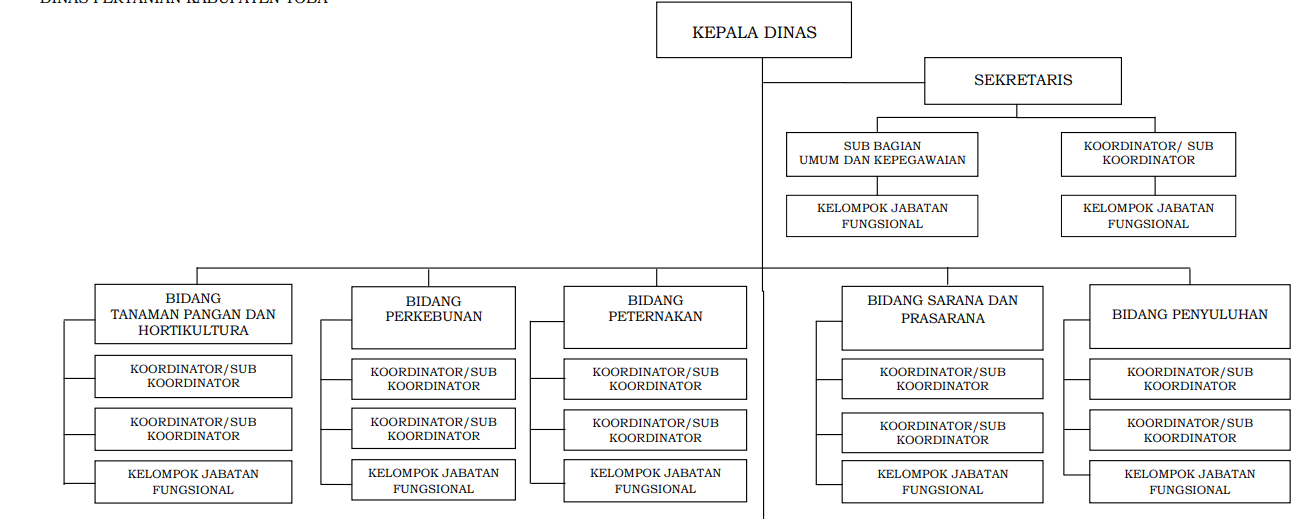
Berdasarkan penelitian sebelumnya, penulis mengajukan penelitian mengenai Pembuatan Sistem Pengadaan Bantuan Pertanian untuk Kelompok Tani di Kabupaten Toba. Perbedaan penelitian yang dilakukan peneliti dengan penelitian terkait adalah pada metode dan fitur yang akan dibangun peneliti. Metode yang digunakan peneliti adalah *Personal Extreme Programming* (PXP) dan fitur pembeda yang dibangun peneliti terletak pada fitur *monitoring* penyebaran bantuan. Fitur *monitoring* ini membantu pihak Dinas Pertanian dalam melakukan pengawasan bantuan.

Kelompok Tani

## Dasar Teori

### Dinas Pertanian

Dinas pertanian merupakan lembaga pemerintah yang dapat melakukan pemberdayaan terhadap hasil pertanian dan peternakan yang ada diwilayah masing – masing [16]. Dinas pertanian melakukan pemberdayaan dan didalam perannya pemerintah akan melakukan sosialisasi, pendampingan, pelatihan dan kunjungan lapangan, kemudian indikator ketiga pemerintah sebagai fasilitator dimana di dalamnya terdapat peran pemerintah dalam hal pengadaan bibit unggul, pengadaan pupuk, pestisida dan sarana produksi untuk pemberdayaan petani [16]. Dinas kabupaten Toba adalah lembaga pemerintah yang berada di Toba Samosir, salah satu yang menerapkan peran pemerintah dalam lebaga pertanian. Struktur perangkat Dinas Pertanian Kabupaten Toba dapat dilihat pada gambar 2. X berikut :



### Kelompok Tani

**Kelompok tani adalah organisasi yang beranggotakan petani yang bekerja sama untuk mencapai tujuan bersama. Kelompok tani berperan penting dalam kehidupan masyarakat tani karena dapat membantu menyelesaikan berbagai kegiatan dan permasalahan dalam berusaha tani.** [17]. Kelompok tani adalah wadah bagi para petani untuk berkolaborasi dalam menjalankan usaha pertanian. Kelompok tani dapat meningkatkan produktivitas pertanian, meningkatkan kemampuan organisasi dan kerja sama antar petani, serta memecahkan permasalahan pertanian. Kelompok tani memiliki peran penting dalam kehidupan masyarakat tani karena dapat membantu mereka dalam menjalankan usaha pertanian [17]. Kabupaten Toba melakukan pemberdayaan kepada kepada kelompok tani untuk mewujudkan kesejahteraan bagi para petani.

### Website

*Website* adalah media yang berisi kumpulan dari halaman-halaman situs yang terangkum dalam sebuah domain dan sub domain. Halaman *web* ditulis dapat diakses melalui HTTP dan ditulis dalam format HTML (*Hyper Text Markup Language*). Semua kumpulan dari informasi yang ada di *website* dapat membentuk sistem informasi yang besar [18]. Halaman *website* berisi data seperti gambar, suara, dan lainnya yang pengaksesannya menggunakan internet [7]. Berdasarkan uraian, penulis menyimpulkan bahwa *website* merupakan halaman situs yang berisi banyak informasi yang dapat diakses melalui internet. Informasi yang diperoleh dari halaman situs dapat berupa gambar, suara dan informasi dalam bentuk lainnya.

### Sistem Informasi

Sistem merupakan kumpulan maupun himpunan dari komponen, unsur ataupun variabel yang terorganisasi dan sifatnya terpadu karena berketergantungan dan saling berkaitan. Bagian-bagian atau komponen dari sistem ini bentuknya terpadu, tujuannya untuk mencapai sebuah tujuan. Informasi adalah data yang sudah diolah sehingga memiliki nilai tambah. Informasi diolah dan diklasifikasikan sedemikian rupa sehingga dapat digunakan untuk mengambil keputusan [19]. Sistem informasi merupakan komponen-komponen yang terkumpul adalah organisasi yang dapat difungsikan sebagai pengolahan untuk menghasilkan laporan yang dapat disajikan kepada pihak yang berwenang [20].

Berdasarkan uraian, penulis menyimpulkan bahwa sistem informasi adalah komponen yang terorganisasi yang dapat mengelola data sehingga menjadi sebuah informasi yang memiliki nilai tambah. Pengolahan informasi dapat dalam hal administrasi dan sebagainya.

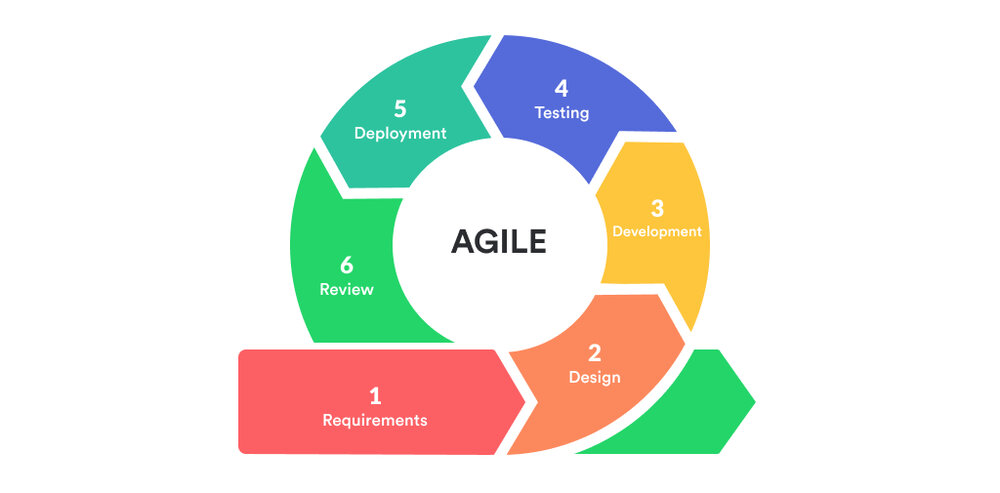
### Sistem Manajemen

Sistem informasi manajemen merupakan kumpulan sub sistem yang saling berhubungan, dan membentuk satu kesatuan, berinteraksi dan bekerja sama untuk melakukan suatu fungsi pengolahan data, menerima masukan berupa data kemudian melakukan pengolahan, dan ditahap akhirnya akan menghasilkan suatu keluaran berupa informasi yang berguna dan dapat dimanfaatkan guna mencapai sebuah tujuan [21].

Sistem informasi manajemen memuat sebuah sistem yang terintegrasi sehingga mampu memberikan informasi yang dapat menunjang jalannya sebuah operasi, manajemen ataupun pengambilan suatu keputusan.

### Agile Software Development Method

Metode *agile* merupakan metode yang bersifat *incremental* sehingga memberikan kemudahan dalam pengembangan tahapan kecil. Tahapan *incremental* ini berfokus untuk pengembangan perangkat lunak yang akan dilakukan cepat, bertahap, mengurangi *overhead* proses, menghasilkan kualitas tinggi karena melibatkan pengguna secara langsung.



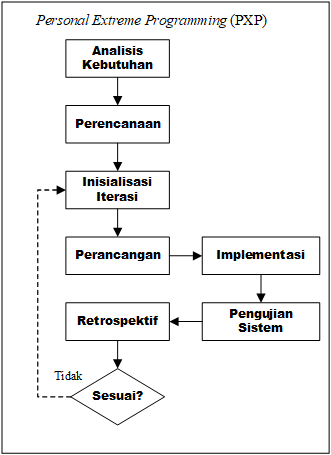
**Gambar 2.1Metode Agile**

Sumber : <http://www.binaracademy.com>

Metode *agile* memiliki beberapa jenis model di antaranya adalah *Extreme Programming, Adaptive Software Development, Dynamic Systems Development Method, Scrum,* dan *Agile Modeling* [22]*.*  Metode Agile memiliki perbedaan dengan metode lainnya dalam tahap pengembangannya. Metode lain, seperti *waterfall*, memiliki metode pengembangan yang kaku karena bersifat sekuensial atau terurut dimulai dari desain, pengodean, pengujian dan tahap pemeliharaan [23].

### Personal Extreme Programming

*Personal Extreme Programming* adalah turunan dari metode *Extreme Programming* (XP) yang juga merupakan metode pengembangan yang dirancang untuk *single developer* [24].

*Personal Extreme Programming* (PXP) merupakan proses pengembangan yang bersifat iteratif karena memberikan kebebasan kepada *programmer* dalam menangani perubahan yang terjadi pada saat pengembangan dan *Extreme Programming* (XP) dalam pengembangannya harus memiliki tim pengembang atau tidak bisa dilakukan secara personal.

**Gambar 2.2 Personal Extreme Programming**

Gambar 2. SEQ Gambar \\* ARABIC \s 1 1 Tahapan *Personal Extreme Programming* (PXP)

Sumber : Jurnal E-Bisnis, Sistem Informasi , Teknologi Informasi ESIT Vol. XVII No. 01 April 2022

Model ini didesain untuk dipergunakan *programmer* secara individu. PXP menuntut pengembang untuk bertanggung jawab untuk setiap tugas dan perubahan yang terjadi. PXP memiliki tahapan dalam proses pengembangannya [25]. Tahapan kerjanya dapat dilihat pada gambar 2.2 berikut. Rincian tahapan-tahapan pada PXP adalah sebagai berikut [26].

1. *Requirements*

*Requirements* merupakan tahapan pengembang mengumpulkan kebutuhan dengan wawancara dan diskusi dengan *client.* Kebutuhan-kebutuhan yang diperoleh dituliskan dalam bentuk *user stories*.

1. *Planning*

Pengembang menyusun dan membuat *task* yang akan dilaksanakan pada setiap iterasi berdasarkan *user stories* yang telah didapatkan. Pembagian *task* dilakukan berdasarkan prioritas dari *user stories* dan estimasi waktu pengerjaan.

1. *Iteration Initialization* adalah tahap awal yang dilaksanakan untuk memulai *task* yang akan dikerjakan. Tahap ini melakukan pemilihan tugas yang akan dijadikan fokus utama dari iterasi tersebut.
2. *Design*

Fase ini merupakan tahap untuk memodelkan modul sistem yang akan diimplementasikan selama proses iterasi. *Design* yang dibuat pengembang hanya berdasar dari kebutuhan *client* yang diperoleh pada tahap *requirement*.

1. *Implementation*

Tahap ini merupakan pengimplementasian setiap objek pada tahap *design* dalam kode program. Tahap ini memiliki tiga tahapan yaitu *Unit Testing*, *Code Generation*, dan *Code Refactoring*. *Unit testing* melakukan pengujian fungsionalitas *code* *program* di mana sebagian *code program* dituliskan oleh pengembang di awal tahap pengembangan lalu dilakukan pengujian. *Code generation* adalah *code program* setiap fitur yang telah lulus *unit testing* lalu dilanjutkan dengan melengkapi *code program* hingga selesai. Tahap terakhir adalah *refactoring* atau optimasi *code program* [26]*.*

1. *System Testing*

Tahap ini merupakan tahapan pengujian fungsionalitas semua fitur yang ada dalam sistem. Hasil pengujian disajikan sebagai *User Acceptance Test*. Pengujian dilakukan ketika pengembang telah mengirimkan *user acceptance test* kepada *user* untuk dilakukan validasi.

1. *Retrospective*

Tahapan ini merupakan tahapan terakhir. Tahap ini pengembang melakukan analisis terhadap waktu pengembangan, estimasi waktu pengerjaan, penyebab keterlambatan dan lain sebagainya untuk mencegah hal serupa terulang di iterasi selanjutnya.

### Perancangan Sistem

#### Use Case Diagram

*Use case diagram* merupakan sesuatu yang merepresentasikan kegiatan atau hal yang dapat dilakukan oleh aktor dalam menyelesaikan suatu pekerjaan. *Use case diagram* ini memodelkan kelakuan atau mendeskripsikan interaksi antar aktor dengan sistem informasi yang akan dibuat. Berikut merupakan simbol-simbol yang ada pada diagram *use case diagram* [27]:

Tabel 2.1 Simbol Use Case Diagram

| **Simbol** | **Nama** | **Keterangan** |
| --- | --- | --- |
| https://documents.lucid.app/documents/a86bdb19-9f68-48ad-8b36-4624b5714d2c/pages/.Q4MUjXso07N?a=138&x=1008&y=279&w=132&h=242&store=1&accept=image%2F*&auth=LCA%20dfe20608a50ded5bb24978b394c4c9dc9a019696dd7b1ea8c5faebec12e9c10c-ts%3D1684656518 | *Actor* | Menggambarkan manusia yang berinteraksi atau suatu hal yang menggunakan sistem |
|  | *Use Case* | Menjelaskan bagian utama dari kegunaan sistem |
|  | *Association Relationship* | Sebagai penghubung antara aktor dan *use case* yang berinteraksi. |
|  | *Directed Association Relationship* | Hubungan asosiasi yang diarahkan hanya kepada satu arah. |
|  | *Include Relationship* | Kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi |
|  | *Extend Relationship* | Menggambarkan perilaku yang opsional / hanya dijalankan pada kondisi tertentu pada *use case*. |

Sumber : Jurnal Yunahar Heriyanto

#### Activity Diagram

*Activity Diagram* merupakan aktivitas dari sebuah proses atau proses bisnis. *Activity diagram* ini menggambarkan aktivitas yang akan dilakukan oleh sistem bukan aktivitas yang dilakukan oleh aktor. Simbol-simbol yang diguanakan dalam *activity diagram* adalah sebagai berikut [27]:

Tabel 2.2 Simbol Activity Diagram

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | *Start* | *Start* untuk menyatakan awal dari suatu proses. |
|  | *Stop* | *Stop* untuk menyatakan akhir dari suatu proses. |
|  | *Decision* | *Decision* untuk menyatakan kondisi dari suatu proses. |
|  | *Action* | *Action* menyatakan aksi yang dilakukan dalam suatu arsitektur sistem. |
|  | *Swimlane* | *Swimlane* memisahkan organisai bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |
|  | *Join* | Penggabungan dimana yang lebih dari satu aktivitas lalu digabung menjadi satu. |

Sumber : Jurnal Yunahar Heriyanto

### Database

*Database* merupakan sekumpulan data yang teratur berdasarkan kriteria yang saling berhubungan. *Database* juga merupakan salah satu komponen yang penting karena berfungsi sebagai basis penyediaan informasi bagi pemakainya. Konsep *database* adalah kumpulan dari data-data berkumpul menjadi suatu berkas yang saling berhubungan dengan tata cara baru hingga terbentuk suatu informasi baru. Tujuan *database* adalah untuk menghindari kekacauan dalam hal pengolahan data yang jumlahnya besar [28]

#### MySQL

*MySQL* merupakan *database server* yang bersifat *open source*. Database ini banyak digemari karena memiliki API (*Application Programming Interface*) sehingga memungkinkan banyak bahasa pemrograman dapat mengaksesnya. *Database* terbagi menjadi dua yaitu *database* flat dan *database* relasional. *MySQL* merupakan *database* relasional. Dikatakan relasional karena memiliki struktur relasional yakni memiliki tabel-tabel untuk menyimpan data. *Database MySQL* mampu mengirim dan menerima data dengan cepat dan multiuser. *MySQL* memiliki dua bentuk yaitu *free software* dan *shareware*. Menggunakan *MySQL* yang *free software* membuat pengguna tidak harus membayar lisensi karena berada dibawah lisensi GNU/GPL (*General Public License)* [29]*.*

### Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) merupakan diagram yang berada dalam pembuatan *database*. Diagram ini memiliki notasi yang menghubungkan yang menghubungkan antara yang satu dengan yang lain. ERD berfungsi untuk memberikan gambaran bagaimana cara kerja *database* yang akan dibuat dalam pengembangan sistem [30].

ERD memiliki 3 elemen dasar, yaitu entitas, atribut, dan relasi [31]. Penjelasan tiga elemen dasar tersebut adalah sebagai berikut:

Tabel 2.3 Simbol ERD Diagram

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | Entitas | Menunjukkan suatu objek yang  dapat diidentifikasi dalam  lingkungan pemakai. |
|  | Relasi | Menunjukan adanya hubungan  diantaran sejumlah entitas yang berbeda. |
|  | Atribut | Berfungsi mendeskripsikan  karakter entitas (atribut yang  berfungsi sebagai kunci diberi  garis bawah). |
|  | Alur | Sebagai penghubung antara  relasi dengan entitas, relasi dan  entitas dengan atribut. |
|  | *Weak Entity* | Suatu entitas dimana  keberadaan dari entitas tersebut  tergantungdari entitas lain. |

### Framework

*Framework* merupakan kerangka kerja yang membantu programmer dalam pemgembangan aplikasi dan juga akan memudahkan programmer dalam melakukan perubahan dalam aplikasinya serta dapat digunakan kembali pada aplikasi lain yang sejenis [32].

*Framework* merupakan kerangka kerja yang memudahkan programmer untuk membuat aplikasi dengan *library* fungsi yang akan digunakan untuk memudahkan membuat aplikasi dengan cepat [33].

#### Laravel

*Laravel* adalah *framework opensource* yang diciptakan oleh Taylor Otwell. *Laravel* merupakan *framework bundle*, migrasi dan artisan CLI (*Command Line Interface*) yang menawarkan seperangkat alat dan arsitektur aplikasi yang menggabungkan banyak fitur terbaik dari kerangka kerja seperti *Codeigniter*, *Yii, ASP.NET MVC*, *Ruby on Rails*, *Sinatra* dan lain-lain. *Laravel* memiliki seperangkat sangat kaya *fitur* yang akan meningkatkan kecepatan pengembangan *web*. Pada *framework laravel* terdapat lima konsep arsitektur yang masing-masing mempunyai fungsi tersendiri di antaranya:

1. *Routes*

Berfungsi sebagai pemberi akses pada setiap *request* sesuai alur yang ditentukan. Di dalamnya memiliki empat instruksi standar di antaranya *Get, Put, Post, Delete*.

1. *Controller*

*Controller* merupakan bagian yang menjadi penghubung antara model dan *view*. *Controller* memiliki perintah-perintah yang berfungsi untuk memproses bagaimana data ditampilkan dari *model* ke *view* atau sebaliknya. Struktur *controller* pada penulisan kode program di *laravel* yakni *Index, Create, Store, Show, Edit, Update*, dan *Delete.*

1. *Model*

*Model* merupakan sekumpulan data yang memiliki fungsi-fungsi untuk mengelola suatu *table* pada sebuah *database*. Struktur pemodelan data pada laravel yakni memiliki fungsi yang terdiri dari *table, primary key* dan *fillable.* Di mana ketiga fungsi tersebut harus di *protected*. Pada bagian *table* harus diisi dengan nama *table* yang sesuai pada *database*, di bagian *primaryKey* harus diisi sesuai *primary key* pada *table* tersebut dan pada bagian *fillable* diisi dengan bagian-bagian yang mencakup dalam *table* tersebut.

1. *View*

*View* merupakan *file* yang berisi kode HTML (*HyperText Markup Language*) yang berfungsi untuk menampilkan suatu data ke dalam *browser*. *Format view* pada laravel harus menggunakan istilah *blade.*

1. *Migrations*

*Migrations* merupakan proses perancangan suatu *table*, dalam hal ini *migrations* berfungsi sebagai *blueprint database* atau dapat diistilahkan sebagai penyedia sistem kontrol untuk skema *database.*

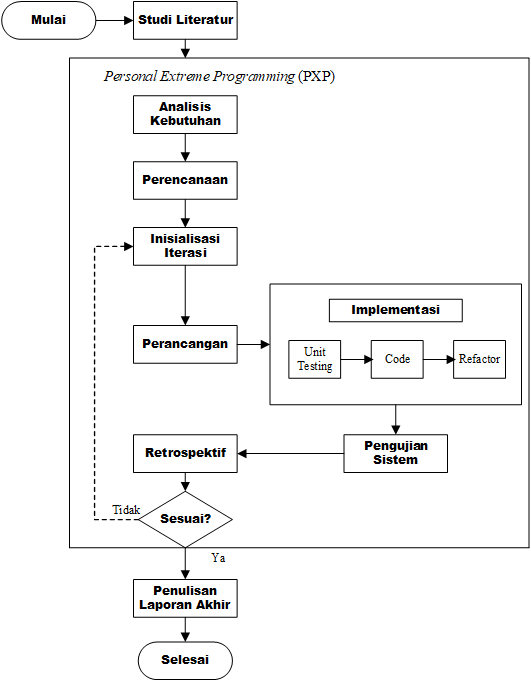
Keunggulan dari *framework* *laravel* yaitu *performance* lebih cepat, *reload* data lebih stabil, memiliki keamanan data, menggunakan fitur canggih seperti *blade* menggunakan konsep HMVC (*Hierarchical Model View Controller*), tersedianya *library-library* yang sudah siap untuk digunakan dan adanya fitur pengelolaan *migrations* untuk pembuatan skema *table* pada *database* [34].

### Pengujian User Acceptance Test (UAT)

Pengujian dalam pengembangan sistem manajemen pendataan pengadaan bantuan untuk kelompok tani ini menggunakan pengujian *User Acceptamce Test* (UAT). UAT adalah pengujian yang dilakukan oleh user atau klien dan dilakukan verifikasi apakah sistem telah berjalan sesuai dengan kebutuhan. Pengujian UAT ini menggunakan teknik pengujian *black box testing* untuk menguji sistem terhadap spesifikasinya. Pengujian ini merupakan fase terakhir dari proses pengujian sistem yang sedang dikembangkan. UAT berfungsi untuk memastikan fungsi fitur-fitur sudah sesuai atau berjalan sepertinya yang diharapkan dalam kebutuhan [11].

# BAB III METODE PENELITIAN

## Alur Penelitian

Alur penelitian merupakan tahapan pelaksanaan yang digunakan untuk membantu mempermudah jalannya penelitian. Alur penelitian ini dituangkan dalam bentuk *flowchart* atau diagram alir yang menggambarkan semua tahapan dari awal hingga akhir. Diagram alir dapat dilihat pada gambar 3.1 di bawah ini.

**Gambar 3.1Diagram alir penelitian**

Gambar 3. SEQ Gambar \\* ARABIC \s 1 1 Diagram Alir Penelitian

Tahapan pada diagram alir ini dimulai dari studi literatur lalu masuk ke tahapan *personal extreme programming* yaitu analisis kebutuhan, perencanaan, inisialisasi iterasi, perancangan, implementasi, pengujian sistem, retrospektif, penulisan laporan akhir dan selesai

## Penjabaran Langkah Penelitian

Penelitian ini dilakukan dengan langkah-langkah yang sudah digambarkan pada gambar 3.1. Untuk memperjelas isi dari setiap langkah penelitian maka berikut uraiannya.

### Studi Literatur

Perancangan sistem informasi desa ini memerlukan pemahaman teoritis terhadap bagian-bagian yang digunakan dalam merancang dan membangun sistem. Studi literatur dilakukan dengan belajar berbagai referensi baik dari jurnal, buku, dan situs-situs terpercaya. Pemahaman pengembang terhadap teoritis yang lebih baik diharapkan membuat pelaksanaan penelitian dilakukan dengan tepat dan hal-hal sulit dalam perancangan dapat diatasi. Jurnal yang dijadikan referensi dalam penelitian ini, beberapa di antaranya adalah jurnal Sistem Informasi Pengelolaan Bantuan Desa Banjardowo oleh Titis Ulfa Mustikawati dan jurnal Sistem Informasi Pengadaan dan Pengalokasian Pupuk pada Dinas Pertanian, Peternakan dan Perkebunan Kabupaten Majene Berbasis Web oleh Samsuriati.

### Analisis Kebutuhan

Analisis kebutuhan merupakan tahap pertama dalam metode PXP. Tahapan ini dijadikan pengembang untuk mengumpulkan kebutuhan-kebutuhan yang akan dituangkan ke dalam sistem. Pengumpulan kebutuhan ini dilakukan dengan wawancara dan diskusi bersama pihak Dinas Pertanian Toba. Kebutuhan-kebutuhan yang diperoleh dari hasil wawancara dituliskan dalam bentuk *user stories*. Struktur perangkat Dinas Pertanian Kabupaten Toba dapat dilihat pada gambar 3.2

### Perencanaan

Tahapan perencanaan ini, pengembang menyusun tugas-tugas yang akan dikerjakan dalam setiap iterasi. Penyusunan tugas dilakukan berdasarkan *user stories* yang telah diperoleh. Pengembang menyusun kebutuhan berdasarkan estimasi waktu pengerjaan dan prioritas. Penyusunan tugas-tugas yang dilakukan pengembang disebut dengan *practice planning game* [26]*.* Penjabaran terkait penjabaran iterasi ini dapat dilihat pada rumus di bawah ini.

**Rumus 3.1Diagram alir penelitian**

Total *Stories Point* = Jumlah keseluruhan dari estimasi waktu pengerjaan user stories.

*Velocity* = Waktu yang digunakan untuk mengerjakan setiap iterasi.

Total Iterasi = Jumlah iterasi yang terbentuk untuk semua *user stories*.

### Inisialisasi Iterasi

Inisiasi iterasi merupakan tahapan awal sebelum sebuah iterasi dimulai. Iterasi dimulai dengan pemilihan tugas yang menjadi fokus utama yang akan dikerjakan dari iterasi tersebut. Tugas yang dipilih diperoleh dari hasil perencanaan pada tahap sebelumnya yaitu tahapan perencanaan. Inisiasi iterasi ini akan menentukan nilai *velocity* untuk setiap iterasi.

### Perancangan

Tahapan perancangan, pengembang membuat model rancangan yang akan diimplementasikan selama proses iterasi. Desain yang dirancang hanya memenuhi kebutuhan pengguna yang diperoleh pada tahap analisis kebutuhan. Rancangan yang dibuat oleh pengembang merupakan *use case diagram* yaitu skema rancangan *prototype* aktor dan tugas yang dilakukan dari iterasi yang dijalankan [26].

### Implementasi

Implementasi merupakan tahapan mengeksekusi desain yang dibuat pada tahap perancangan ke dalam kode program sehingga dapat dipergunakan menjadi sistem pengadaan di Dinas Pertanian Toba. Tahapan implementasi memiliki tiga tahap yaitu *Unit Testing*, *Code Generation*, dan *Code Refactoring*.

### Pengujian Sistem

Pengujian sistem merupakan pengujian fungsionalitas yang dilakukan terhadap fitur-fitur yang telah diimplementasikan dari setiap iterasi. Pengujian dalam metode *personal extreme programming* (PXP) ini dilakukan oleh pengguna dan hasil pengujian disajikan dalam bentuk *User Acceptance Test.* Pengujian dilakukan ketika sistem yang dibangun sudah di *hosting* terlebih dahulu. Dokumen *User Acceptance Test* diberikan kepada pengguna untuk proses validasi. Pengguna akan memberikan verifikasi terkait fungsi dari sistem yang diuji telah sesuai atau tidak sesuai dengan yang diinginkan.

### Retrospektif

Retrospektif adalah tahapan terakhir dari proses iterasi. Pengembang melakukan analisis terhadap pengembangan sistem baik dari kesesuaian estimasi waktu pengerjaan, kendala yang menyebabkan keterlambatan, dan lain sebagainya. Analisis dilakukan untuk mencegah hal yang tersebut terulang kembali pada iterasi selanjutnya.

### Penulisan Laporan Akhir

Penulisan laporan akhir adalah tahapan menuangkan hasil penelitian ke dalam laporan. Laporan akhir ini akan dijadikan salah satu bukti dan syarat bahwa peneliti telah selesai melakukan penelitian terkait pengembangan sistem pengadaan di Dinas Pertanian dan telah didapatkan hasil berupa *website* dan sistem pengadaan bantuan yang dapat digunakan penduduk dan pegawai Dinas Pertanian Toba.

## Alat dan Bahan Tugas Akhir

Perancangan dan pembangunan sistem yang dilakukan dalam penelitian ini membutuhkan alat dan bahan yang digunakan peneliti untuk menunjang penelitian.

### Alat

Alat yang digunakan dalam penelitian sistem informasi pada Kantor Dinas Pertanian Kabupaten Toba adalah sebagai berikut:

1. *Software*
2. *Microsoft word*
3. *Microsoft visio*
4. *Sistem Operasi Windows 10*
5. *Visual studio code*
6. *Xampp*
7. *MySql*
8. *Hardware*
9. *Laptop* dengan prosesor intel *core* i3 dengan ram 4GB
10. *Printer*
11. *Flashdisk*
12. *Smartphone*

### Bahan

Bahan penelitian yang digunakan peneliti adalah hasil dari wawancara dan observasi yang dilakukan. Bahan-bahan penelitian yang digunakan adalah sebagai berikut:

1. Data penduduk yang merupakan anggota kelompok tani.
2. Data nama-nama ketua kelompok tani.
3. Data daftar bantuan yang akan dibagikan.
4. Proposal yang pernah diajukan.

## Metode Tugas Akhir

Metode penelitian yang dilakukan dalam pengembangan sistem informasi pendataan pengadaan bantuan di Kabupaten Toba ini adalah *Personal Extreme Programming* (PXP). Metode PXP memiliki tahapan analisis kebutuhan, perencanaan, inisiasi iterasi, perancangan, implementasi, pengujian sistem dan retrospektif. Metode ini dapat dilihat pada gambar 3.3 berikut.

Gambar 3. SEQ Gambar \\* ARABIC \s 1 2 Tahapan Metode PXP

Tahapan metode PXP dilakukan pengembang sesuai dengan setiap proses yang dilakukan dalam pengembangan sistem pengadaan Dinas Pertanian. Berikut adalah penjabaran tahapan metode PXP.

### Analisis Kebutuhan

Tahap analisis kebutuhan ini diperoleh dari *client* melalui wawancara dan diskusi bersama ibu Nora Hutabarat selaku Kepala Program Pertanian di Dinas Pertanian Kabupaten Toba. Hasil yang didapatkan oleh pengembang dituliskan dalam bentuk *user stories* dengan format “Sebagai <jenis pengguna>, < saya ingin melakukan tindakan sesuatu>, sehingga <mendapatkan manfaat dari tindakan tersebut>”. Deskripsi dari pengguna atau aktor dapat dilihat pada tabel 3.1 berikut.

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Kepala Dinas | Seseorang yang menetapkan rencana, merumuskan kebijakan, menetapkan penyelenggaraan program di Dinas Pertanian Toba. |
| 2 | Kepala Bidang | Seseorang yang menangani bidang masing-masing di Dinas Pertanian Toba. |
| 3 | Badan Penyuluh | Seseorang yang merangkum dan mengumpulkan informasi serta kebutuhan masyarakat. |

**Tabel 3.1 Pengguna sistem**

pada tabel 3.2 maka kebutuhan fungsional dan kebutuhan non fungsionalnya adalah sebagai berikut.

|  |  |  |
| --- | --- | --- |
| **No.** | **Fitur** | **Kebutuhan Fungsional** |
| 1 | Pengaturan User | * Kepala dinas dapat menambah user * Kepala dinas dapat melihat user * Kepala dinas dapat mengedit user * Kepala dinas dapat menghapus user |
| 2 | Manajemen Kecamatan | * Kepala dinas dapat menambah kecamatan * Kepala dinas dapat melihat kecamatan * Kepala dinas dapat mengedit kecamatan * Kepala dinas dapat menghapus kecamatan |
|  |  |  |
| **No.** | **Fitur** | **Kebutuhan Fungsional** |
| 3 | Manajemen Bidang Pertanian | * Kepala dinas dapat menambah bidang * Kepala dinas dapat melihat bidang * Kepala dinas dapat mengedit bidang * Kepala dinas dapat menghapus bidang |
| 4 | Manajemen Kamus Usulan | * Kepala bidang dapat menambah kamus usulan * Kepala bidang dapat melihat kamus usulan * Kepala bidang dapat mengedit kamus usulan * Kepala bidang dapat menghapus kamus usulan |
| 5 | Manajemen Kelompok Tani | * Badan Penyuluh dapat menambah kelompok tani * Badan Penyuluh dapat melihat kelompok tani * Badan Penyuluh dapat mengedit kelompok tani * Badan Penyuluh dapat menghapus kelompok tani |
| 5 | Mengajukan Proposal Kelompok Tani | * Badan Penyuluh dapat menambah proposal * Badan Penyuluh dapat melihat proposal * Badan Penyuluh dapat mengedit proposal * Badan Penyuluh dapat menghapus proposal |
| 6 | Manajemen Proposal Kelompok Tani | * Kepala bidang dapat melihat proposal kelompok tani sesuai bidang. * Kepala bidang dapat menyetujui/menolak proposal kelompok tani * Kepala bidang dapat mengubah status progres pengerjaan proposal * Kepala bidang dapat mengubah alokasi anggaran * Kepala bidang dapat membuat laporan per proposal |
| 7 | Manajemen Proposal Kelompok Tani | * Kepala dinas dapat membuat laporan rekapitulasi proposal * Kepala dinas dapat melihat proposal. |

**Tabel 3.2Kebutuhan Fungsional**

|  |  |
| --- | --- |
| **No.** | **Kebutuhan Non-fungsional** |
| 1 | Menggunakan OS Windows |
| 2 | Menggunakan RAM 16 |
| 3 | Menggunakan spesifikasi Laptop dengan prosesor AMD Ryzen 7 5800 |
| 4 | Menggunakan SSD 256 |

**Tabel 3.3Kebutuhan Non-fungsional**

Berdasarkan hasil kebutuhan untuk Rancang Bangun Sistem Manajemen Pendataan Bantuan untuk Kelompok Tani pada Dinas Pertanian Kabupaten Toba

ERD

Usecase

menggunakan metode *Personal Extreme Programming* (PXP) dituliskan *user stories* yang dapat dilihat pada gambar Tabel 3.1.

| *Kode User Stories* | *User Stories* |
| --- | --- |
| *Story-01* | Sebagai kepala dinas saya ingin dapat menambah user ke dalam *user*, mengedit data user, menghapus user sehingga apabila ada pergantian perangkat di Dinas Pertanian, pengaturan user dapat dilakukan. |
| *Story-02* | Sebagai kepala dinas saya ingin dapat menambah kecamatan sehingga apabila ada pemekaran dan penambahan kecamatan maka dapat ditambahkan ke dalam sistem. |
| *Story-03* | Sebagai kepala dinas saya ingin dapat menambah bidang sehingga mempermudah apabila ada perubahan struktural. |
| *Story-04* | Sebagai kepala dinas saya ingin membuat rekapitulasi dan melihat proposal yang diajukan sehingga proposal bisa saya pantau. |
| *Story-05* | Sebagai kepala bidang saya ingin dapat menambah, melihat dan mengedit kamus usulan sehingga mempermudah dalam mengelola kamus usulan. |
| *Story-06* | Sebagai kepala bidang saya ingin saya bisa melihat, menyetujui/menolak, mengubah status proposal yang diajukan sehingga mempercepat proses pengajuan proposal. |
| *Story-07* | Sebagai badan penyuluh saya ingin dapat menambah kelompok tani, melihat, mengedit, dan menghapus tani sehingga memudahkan saya dalam memanajemen kelompok tani. |
| *Story-08* | Sebagai badan penyuluh saya ingin dapat menambah proposal, melihat, mengedit dan menghapus proposal sehingga memudahkan saya dalam mengajukan proposal kelompok tani. |

**Tabel 3.4User Stories**

Berdasarkan *user story* yang sudah dijabarkan

### Tahap Perencanaan

Perencanaan terdiri dari membuat estimasi *user stories*, penentuan prioritas *user stories*, dan perencanaan iterasi. Perencanaan dibutuhkan untuk menentukan jalannya pengembangan sistem pengadaan bantuan kelompok tani di Dinas Pertanian Toba.

1. Estimasi *User Stories*

Estimasi *user stories* adalah perkiraan waktu yang digunakan untuk mengimplementasikan setiap *user stories*. Estimasi pengerjaan *user stories* dinamakan dengan *stories point*. Besar *stories point* berbanding lurus dengan waktu pengerjaan. Besar *stories point* adalah 1 maka waktu pengerjaannya adalah 2 hari [35]. Nilai estimasi dari *stories point* tergantung kepada pengembang yang menilai tingkat kesulitan di setiap *user stories* yang akan dikerjakan. Nilai *stories point* adalah dari 1/2, 1, 2, 3, 5, 8, 13, 20, 40, dan 80 [36]. Estimasi pengerjaan yang telah ditentukan pengembang berdasarkan tingkat kesulitan pada setiap *user stories* dapat dilihat pada tabel 3.5 adalah sebagai berikut.

| Kode *User Stories* | *User Stories* | *Stories Point* (Estimasi waktu pengerjaan *user stories*) | Estimasi Waktu (Hari) |
| --- | --- | --- | --- |
| *Story-01* | Sebagai kepala dinas saya ingin dapat menambah user ke dalam sistem, mengedit data user, menghapus user sehingga apabila ada pergantian perangkat di Dinas Pertanian, pengaturan user dapat dilakukan. | 3 | 6 |
| *Story-02* | Sebagai kepala dinas saya ingin dapat me - *manage* kecamatan dan desa yang ada di Kabupaten Toba sehingga apabila ada pemekaran dan penambahan daerah maka dapat ditambahkan ke dalam sistem. | 3 | 6 |
| *Story-03* | Sebagai kepala dinas saya ingin dapat menambah bidang sehingga mempermudah apabila ada perubahan struktural. | 3 | 6 |
| *Story-04* | Sebagai kepala dinas saya ingin membuat rekapitulasi dan melihat proposal yang diajukan sehingga proposal bisa saya pantau. | 3 | 6 |
| *Story-05* | Sebagai kepala bidang saya ingin dapat menambah, melihat dan mengedit kamus usulan sehingga mempermudah dalam mengelola kamus usulan. | 3 | 6 |
| *Story-06* | Sebagai kepala bidang saya ingin saya bisa melihat, menyetujui/menolak, mengubah status proposal yang diajukan sehingga mempercepat proses pengajuan proposal. | 3 | 6 |
| *Story-07* | Sebagai koordinator badan penyuluh saya ingin dapat menambah kelompok tani, melihat, mengedit, dan menghapus tani sehingga memudahkan saya dalam memanajemen kelompok tani. | 3 | 6 |
| *Story-08* | Sebagai badan penyuluh saya ingin dapat menambah proposal, melihat, mengedit dan menghapus proposal sehingga memudahkan saya dalam mengajukan proposal kelompok tani. | 3 | 6 |

**Tabel 3.5 Estimasi Pengerjaan**

1. Prioritas *User Stories*

Penentuan prioritas *user stories* ini, pengembang menggunakan aturan MoSCoW. MoSCoW merupakan akronim dari *Must have, Should Have, Could Have,* dan *Won’t Have.* *Must have* adalah bagian paling dasar dari sistem yang dibangun. *Should have* adalah bagian penting dari sistem yang dibangun namun ada jangka waktu tertentu yang digunakan untuk pengerjaannya. *Could have* adalah bagian sistem yang dapat dikeluarkan dari rencana apabila waktu yang dimiliki tidak ada lagi. *Won’t have* adalah prioritas yang tidak terlalu dibutuhkan untuk sistem yang akan dibangun [37].

Aturan MoSCoW mengelompokkan *user stories* pengembangan sistem pengadaan bantuan kelompok tani di Kabupaten Toba ada dalam 1 kategori yaitu:

1. *Must Have*
2. Kepala dinas dapat *login* dan menambah *user* ke dalam sistem, mengedit data *user*, menghapus *user. User stories* ini penting dan harus diimplementasikan karena dalam pemerintahan ada pergantian pegawai atau perangkat.
3. Kepala dinas dapat menambah kecamatan, mengedit, melihat data kecamatan bahkan menghapus data kecamatan. *User stories* ini penting karena di pemerintahan biasanya ada pemekaran kecamatan.
4. Kepala dinas dapat menambah bidang, melihat bidang, mengedit bidang, bahkan menghapus bidang. *User stories* ini penting karena bidang-bidang dalam pemerintahan biasanya ada penambahan sesuai dengan kebutuhan.
5. Kepala dinas dapat melihat proposal dan membuat rekapitulasi proposal. *User stories* ini penting dan harus diimplementasikan karena penting bagi seorang kepala dinas merekap laporan.
6. Kepala bidang dapat menambah kamus usulan, melihat kamus usulan, mengedit kamus usulan, dan menghapus kamus usulan. *User stories* ini penting dan harus diimplementasikan karena pengajuan proposal bantuan untuk kelompok tani harus berdasarkan kamus usulan.
7. Kepala bidang dapat melihat proposal kelompok tani, menyetujui/menolak proposal yang diajukan, mengubah status progres pengerjaan proposal, mengubah alokasi anggaran, dan membuat laporan per proposal. *User stories* ini sangat penting dan harus diimplementasikan karena kepala bidang sangat membutuhkan fitur ini.
8. Badan penyuluh dapat menambah kelompok tani, melihat kelompok tani, mengedit kelompok tani dan menghapus kelompok tani. *User stories* ini sangat penting dan harus diimplementasikan karena ini untuk mempermudah penyaluran bantuan.
9. Badan penyuluh dapat menambah proposal, melihat proposal, mengedit proposal, dan menghapus proposal. *User stories* ini sangat penting dan harus diimplementasikan karena proposal ini adalah yang dibutuhkan untuk pengajuan.

| Kode *User Stories* | *User Stories* | *Stories Point* (Estimasi waktu pengerjaan *user stories*) | Estimasi Waktu (Hari) | Estimasi Waktu (Hari) |
| --- | --- | --- | --- | --- |
| *Story-01* | Sebagai kepala dinas saya ingin dapat menambah user ke dalam user, mengedit data user, menghapus user sehingga apabila ada pergantian perangkat di Dinas Pertanian, pengaturan user dapat dilakukan. | 3 | 6 | *Must Have* |
| *Story-02* | Sebagai kepala dinas saya ingin dapat menambah kecamatan sehingga apabila ada pemekaran dan penambahan kecamatan maka dapat ditambahkan ke dalam sistem. | 3 | 6 | *Must Have* |
| *Story-03* | Sebagai kepala dinas saya ingin dapat menambah bidang sehingga mempermudah apabila ada perubahan struktural. | 3 | 6 | *Must Have* |
| *Story-04* | Sebagai kepala dinas saya ingin membuat rekapitulasi dan melihat proposal yang diajukan sehingga proposal bisa saya pantau. | 3 | 6 | *Must Have* |
| *Story-05* | Sebagai kepala bidang saya ingin dapat menambah, melihat dan mengedit kamus usulan sehingga mempermudah dalam mengelola kamus usulan. | 3 | 6 | *Must Have* |
| *Story-06* | Sebagai kepala bidang saya ingin saya bisa melihat, menyetujui/menolak, mengubah status proposal yang diajukan sehingga mempercepat proses pengajuan proposal. | 3 | 6 | *Must Have* |
| *Story-07* | Sebagai badan penyuluh saya ingin dapat menambah kelompok tani, melihat, mengedit, dan menghapus tani sehingga memudahkan saya dalam memanajemen kelompok tani. | 3 | 6 | *Must Have* |
| *Story-08* | Sebagai badan penyuluh saya ingin dapat menambah proposal, melihat, mengedit dan menghapus proposal sehingga memudahkan saya dalam mengajukan proposal kelompok tani. | 3 | 6 | *Must Have* |
| Total *stories point* | | 24 | 48 hari | |

**Tabel 3.6 Prioritas User Stories**

Tabel 3.6 menunjukkan *stories point* untuk penentuan estimasi dan *priority* yang menunjukkan urutan prioritas tiap *user stories*. Total *stories point* adalah 24 dengan estimasi 48 hari pengerjaan dari 8 *user stories*. Tahapan selanjutnya adalah membuat perencanaan iterasi menggunakan daftar *user stories* pada tabel 3.6

1. Perencanaan Iterasi

Perencanaan iterasi merupakan tahapan yang berisi *task* yang akan dikerjakan oleh pengembang dalam membangun sistem pengadaan bantuan kelompok tani di Dinas Pertanian Kabupaten Toba. Pengembang bersama dengan *client* memutuskan *user stories* yang mana yang akan dikerjakan terlebih dahulu. Metode penelitian PXP memiliki proses di dalamnya berupa perulangan atau iterasi. Iterasi ditentukan berdasarkan urutan prioritas *user stories* dan untuk menentukan jumlah iterasi maka dibutuhkan nilai *velocity*. Jumlah *stories point* dari semua *user stories* adalah 24 atau sama dengan 48 hari pengerjaan. Nilai *velocity* dari pengembang untuk setiap iterasi adalah 6 untuk memenuhi 48 hari waktu pengerjaan dari keseluruhan *user stories*.

Perhitungan untuk menentukan jumlah iterasi pada pengembangan sistem informasi Desa Way Huwi ini dapat dilihat pada rumus di bawah ini.

Jumlah iterasi yang diperoleh dari perhitungan pada rumus di atas adalah 4 iterasi. Pengimplementasian program untuk semua iterasi dapat diselesaikan dalam 48 hari di mana pengerjaan setiap iterasi adalah 12 hari (nilai *velocity* 6 sama dengan 12 hari pengerjaan) [35].

Penelitian pengembangan sistem informasi desa ini memiliki 4 iterasi. Pemilihan tugas untuk setiap iterasi akan ditentukan pada tahap inisiasi iterasi.

### Tahap Inisiasi Iterasi

Tahap inisiasi iterasi merupakan tahapan awal *iteration development*. Iterasi yang diimplementasikan adalah empat iterasi yang berisi tugas-tugas sebagai fokus utama untuk setiap iterasi. Pengembangan dilakukan secara berulang hingga semua iterasi terimplementasikan. Iterasi penelitian pada tahap inisiasi iterasi ini dapat dilihat pada tabel 3.7.

|  |  |  |  |
| --- | --- | --- | --- |
| Iterasi 1 | | | |
| *User Stories* | *Priority* | *Stories point* | Estimasi waktu (Hari) |
| Sebagai kepala dinas saya ingin dapat menambah user ke dalam sistem, mengedit data user, menghapus user sehingga apabila ada pergantian perangkat di Dinas Pertanian, pengaturan user dapat dilakukan. | *Must have* | 3 | 6 |
| Sebagai kepala dinas saya ingin dapat menambah kecamatan sehingga apabila ada pemekaran dan penambahan kecamatan maka dapat ditambahkan ke dalam sistem. | *Must have* | 3 | 6 |
| *velocity* | | 6 | 12 |
| Iterasi 2 | | | |
| *User Stories* | *Priority* | *Stories point* | Estimasi waktu (Hari) |
| Sebagai kepala dinas saya ingin dapat menambah bidang sehingga mempermudah apabila ada perubahan struktural. | *Must have* | 3 | 6 |
| Sebagai kepala dinas saya ingin membuat rekapitulasi dan melihat proposal yang diajukan sehingga proposal bisa saya pantau. | *Must have* | 3 | 6 |
| *velocity* | | 6 | 12 |
| Iterasi 3 | | | |
| *User Stories* | *Priority* | *Stories point* | Estimasi waktu (Hari) |
| Sebagai kepala bidang saya ingin dapat menambah, melihat dan mengedit kamus usulan sehingga mempermudah dalam mengelola kamus usulan. | *Must have* | 3 | 6 |
| Sebagai kepala bidang saya ingin saya bisa melihat, mengajukan proposal kepada Kepala Dinas, dan memasukkan data pengadaan bantuan kelompok Tani. | *Must have* | 3 | 6 |
| *Velocity* | | 6 | 12 |
| Iterasi 4 | | | |
| *User Stories* | *Priority* | *Stories point* | Estimasi waktu (Hari) |
| Sebagai badan penyuluh saya ingin dapat menambah data kelompok tani, melihat, mengedit, dan menghapus kelompok tani sehingga memudahkan saya dalam memanajemen kelompok tani. | *Must have* | 3 | 6 |
| Sebagai badan penyuluh saya ingin dapat menambah proposal, melihat mengedit dan menghapus proposal sehingga memudahkan saya dalam mengajukan proposal kelompok tani. | *Must have* | 3 | 6 |
| *Velocity* | | 6 | 12 |

**Tabel 3.7 Iterasi**

Tabel 3.7 menunjukkan ada 4 iterasi yang akan diimplementasikan dalam pengembangan sistem manajemen pengadaan bantuan kelompok tani kelompok tani di Dinas Pertanian Kabupaten Toba. *Client* dapat mengajukan tambahan kebutuhan dalam bentuk *user stories* selama proses pengembangan sistem. *Implementasi user stories* sudah dapat dilakukan sesuai dengan daftar iterasi. Tahapan selanjutnya adalah tahap perancangan.

### Tahap Perancangan

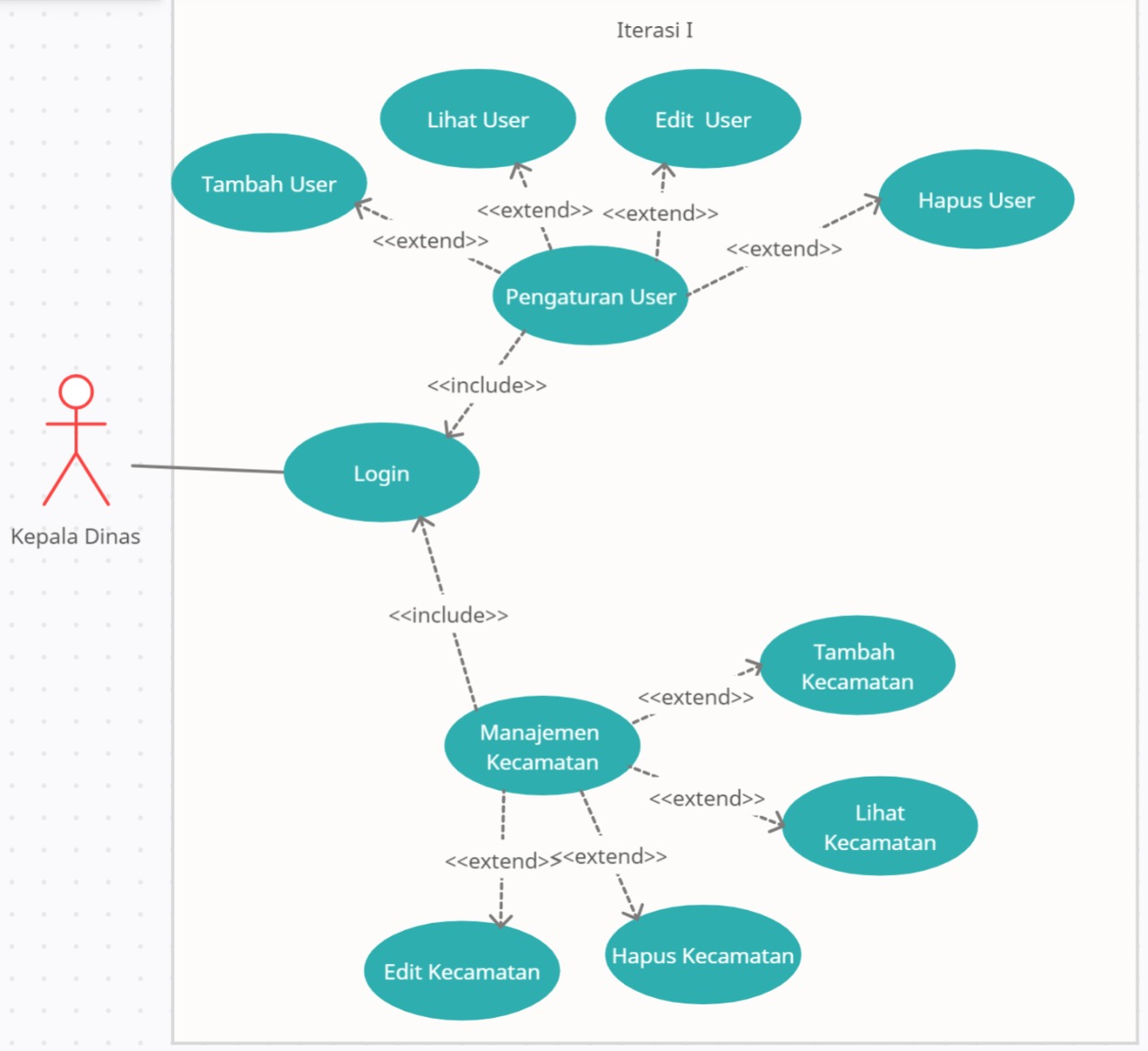
Pengembang membuat desain untuk semua *user stories* dalam membangun sistem informasi Desa Way Huwi dalam tahap ini. Desain dibuat sederhana untuk memodelkan sebuah iterasi yang sedang berlangsung. Pengembang memodelkan dengan membuat *use case diagram* untuk setiap iterasi

1. ***Use Case Diagram***

*Use Case Diagram* mendeskripsikan interaksi aktor pada sistem manajemen pengadaan bantuan kelompok tani. *Use case diagram* akan menjelaskan aktivitas yang dilakukan para aktor kepala dinas, kepala bidang, dan badan penyuluh. Setiap iterasi akan dijelaskan melalui *use case diagram* masing-masing.

1. *Use Case Diagram* Iterasi 1

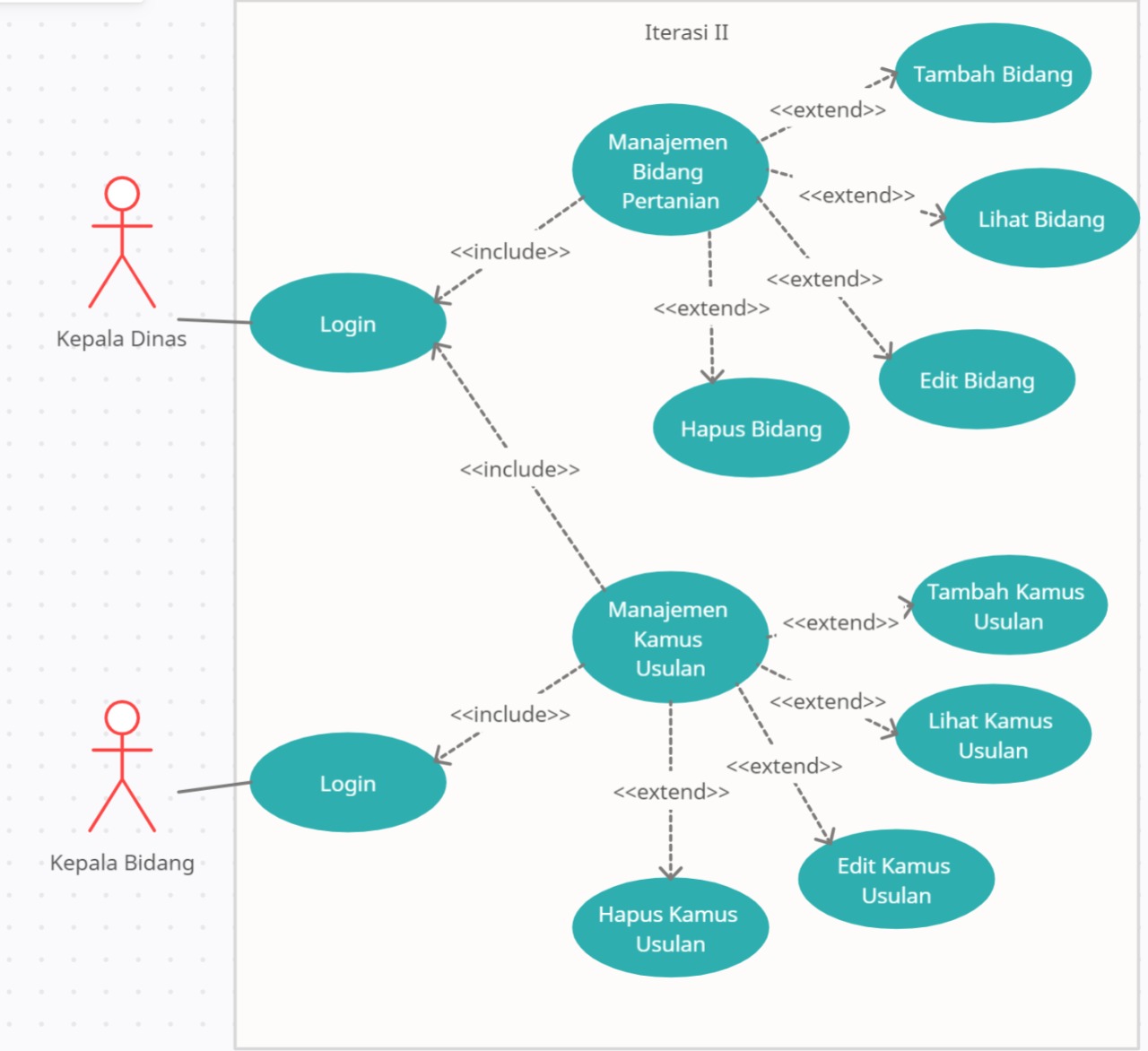
Iterasi pertama, pengembang akan memulai dari manajemen *user* dan manajemen kecamatan. Aktor yang ada pada iterasi pertama ini adalah kepala dinas. Kepala dinas ini mampu menambah *user*, melihat data *user*, mengedit data *user*, bahkan untuk menghapus *user*. Selain itu, kepala dinas ini juga mampu dalam memanajemen kecamatan. Fitur manajemen kecamatan, kepala dinas dapat menambah kecamatan, mengedit kecamatan, melihat data kecamatan dan menghapus data kecamatan. *Use case* iterasi 1 dapat dilihat pada gambar 3.3

****

**Gambar 3.3 Use Case Diagram Iterasi 1**

1. *Use Case Diagram* Iterasi 2

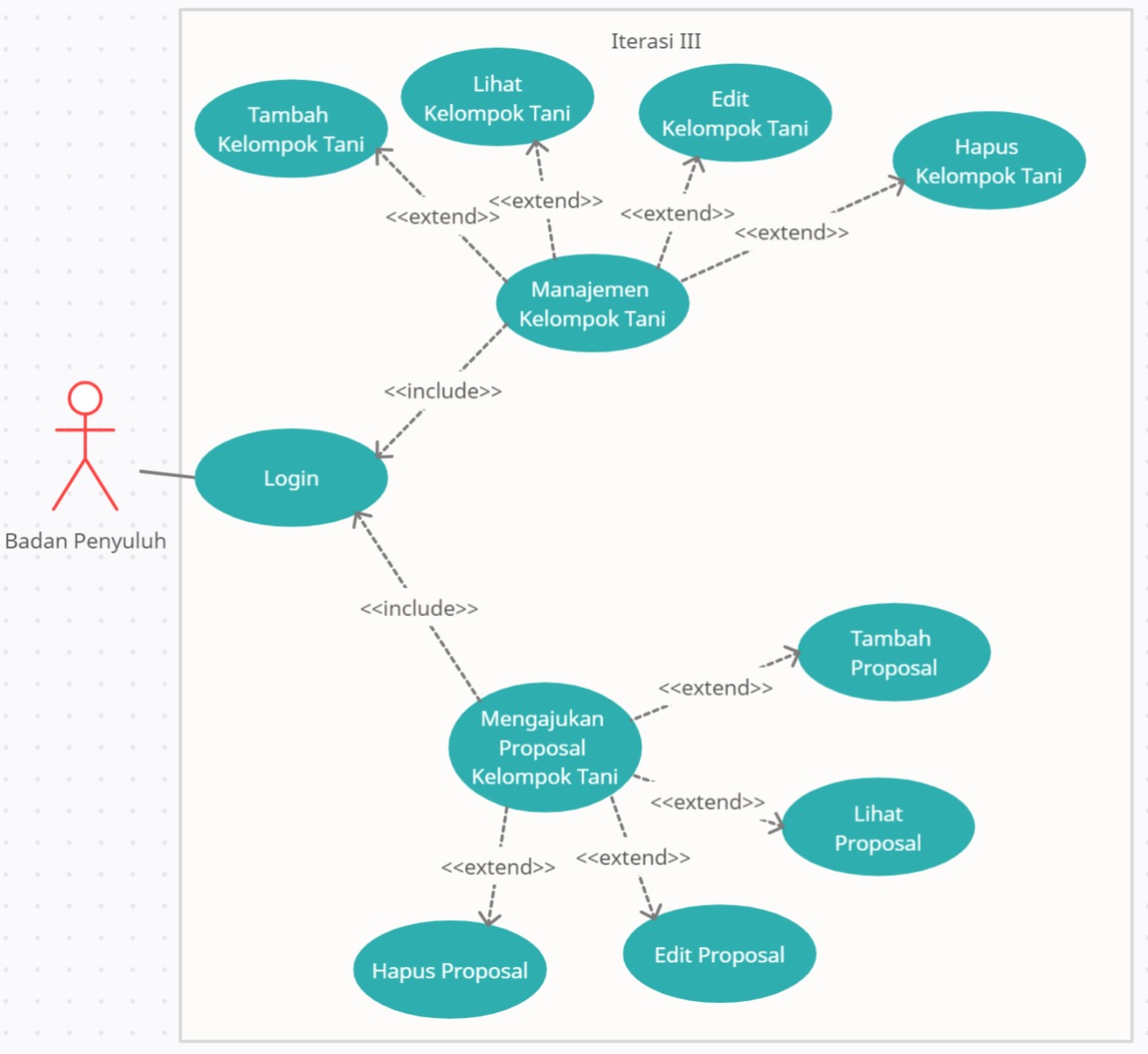
Iterasi kedua memiliki aktor kepala dinas dan kepala bidang. Kepala bidang mampu melakukan manajemen bidang pertanian. Kepala dinas mampu untuk menambah bidang, melihat bidang, mengedit bidang, dan menghapus data bidang. Aktor kedua adalah kepala bidang yang memanajemen kamus usulan. Kepala bidang ini mampu untuk menambah kamus usulan, mengedit kamus usulan, menghapus kamus usulan, dan mengedit kamus usulan. *Use case* diagram iterasi 2 dapat dilihat pada gambar 3.4.



**Gambar 3.4 Use Case Diagram Iterasi 2**

1. *Use Case Diagram* Iterasi 3

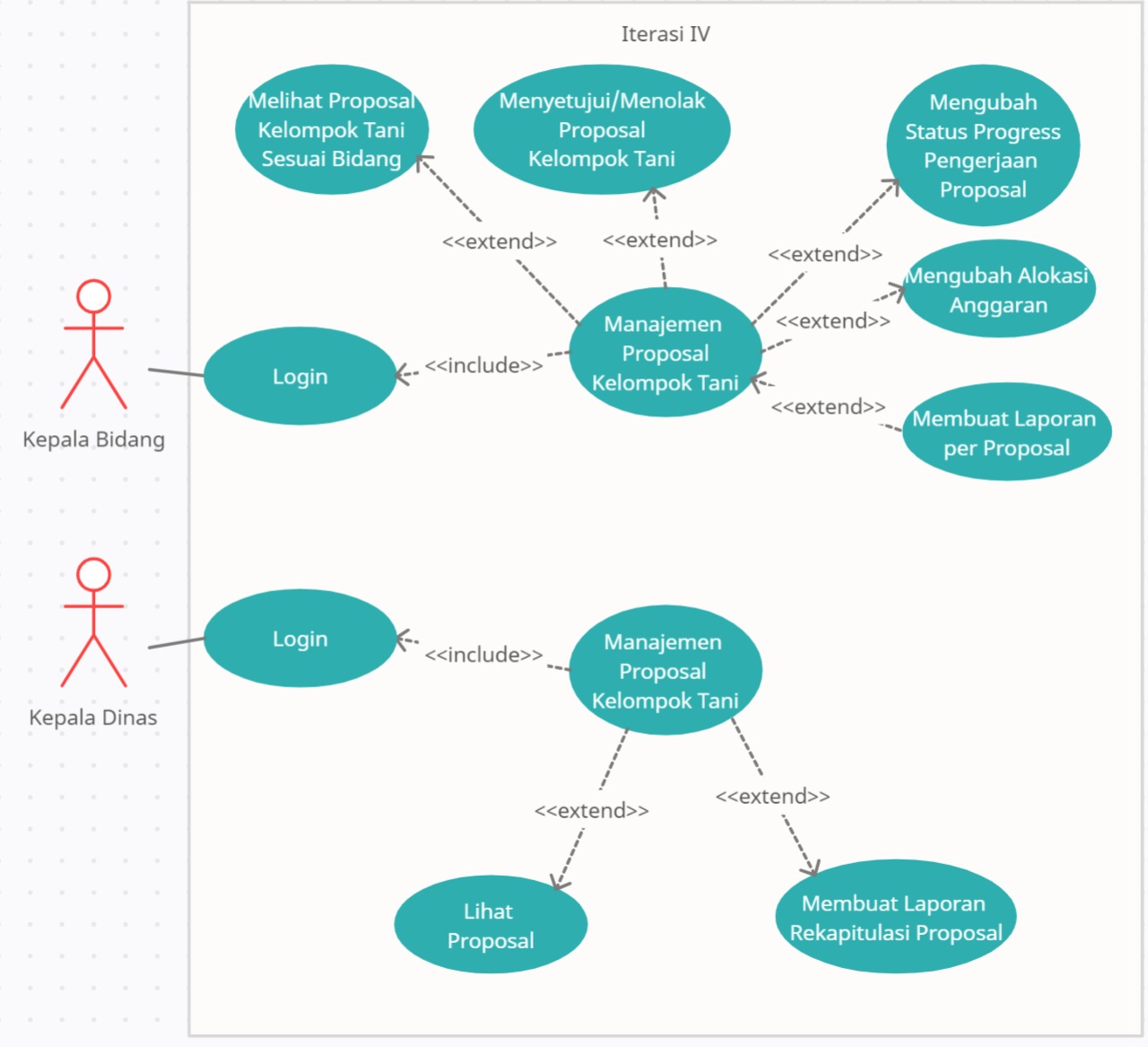
Iterasi ketiga memiliki aktor badan penyuluh. Badan penyuluh ini mampu untuk memanajemen kelompok tani dan mengajukan proposal kelompok tani. Badan penyuluh dalam memanajemen kelompok tani, mampu untuk menambah kelompok tani, mengedit kelompok tani, menghapus kelompok tani, dan melihat data kelompok tani. Badan penyuluh dan mengajukan proposal kelompok tani mampu menambah proposal, mengedit proposal, menghapus proposal, dan melihat proposal. *Use case diagram* iterasi 3 dapat dilihat pada gambar 3.5.



**Gambar 3.5 Use Case Diagram Iterasi 3**

1. *Use Case Diagram* Iterasi 4

Iterasi Keempat adalah iterasi terakhir dalam pengembangan sistem ini. Aktor dalam iterasi keempat ini adalah kepala bidang dan kepala dinas. Kepala bidang mampu melakukan manajemen proposal kelompok tani. Kepala bidang mampu untuk melihat proposal kelompok tani sesuai bidang, menyetujui/menolak proposal kelompok tani, mengubah status progres pengerjaan proposal, mengubah alokasi anggaran, dan membuat laporan untuk setiap proposal. Kepala dinas dalam iterasi ini mampu untuk membuat laporan rekapitulasi proposal dan melihat proposal. *Use case diagram* iterasi 4 dapat dilihat pada gambar 3.6.



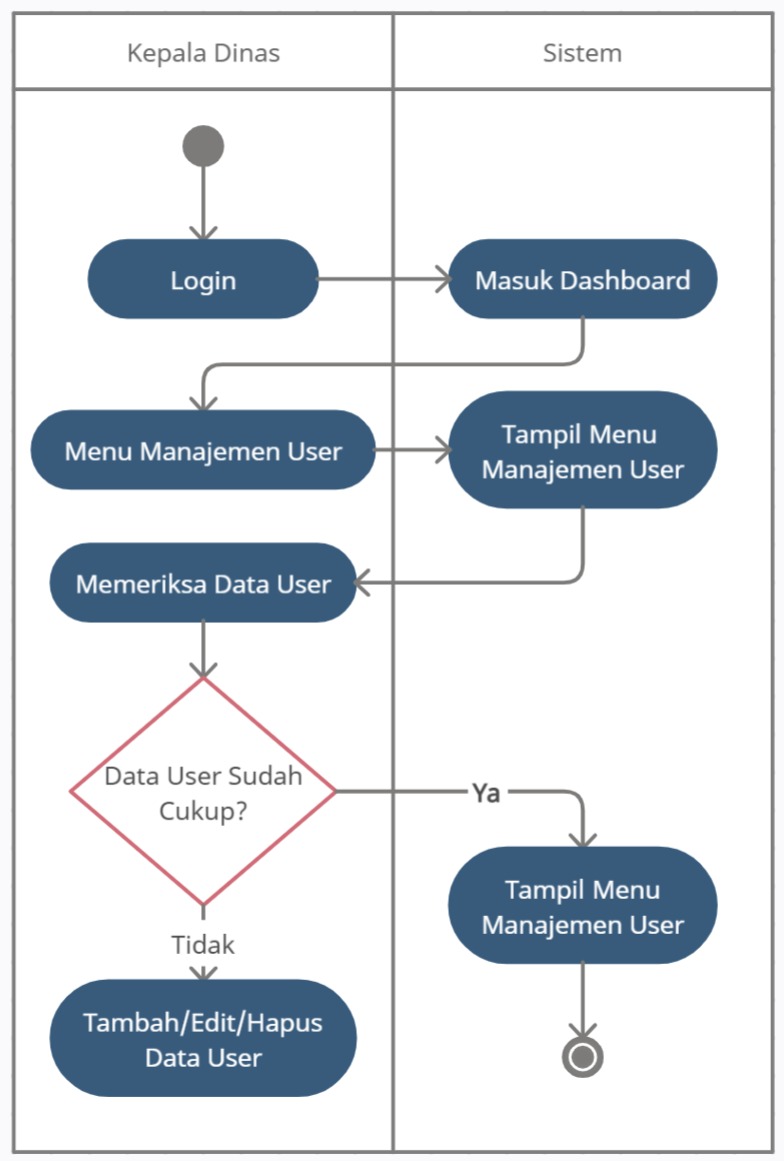
**Gambar 3.6 Use Case Diagram Iterasi 4**

1. ***Activity Diagram***

*Activity diagram* merupakan diagram pemodelan yang digunakan untuk menggambarkan proses yang berjalan dalam manajemen pengadaan bantuan kelompok tani di Kabupaten Toba. Pemodelan ini akan memperlihatkan urutan proses bisnis. *Activity diagram* dibuat berdasarkan *use case diagram* yang sudah dibuat dengan empat iterasi

1. Manajemen *User*

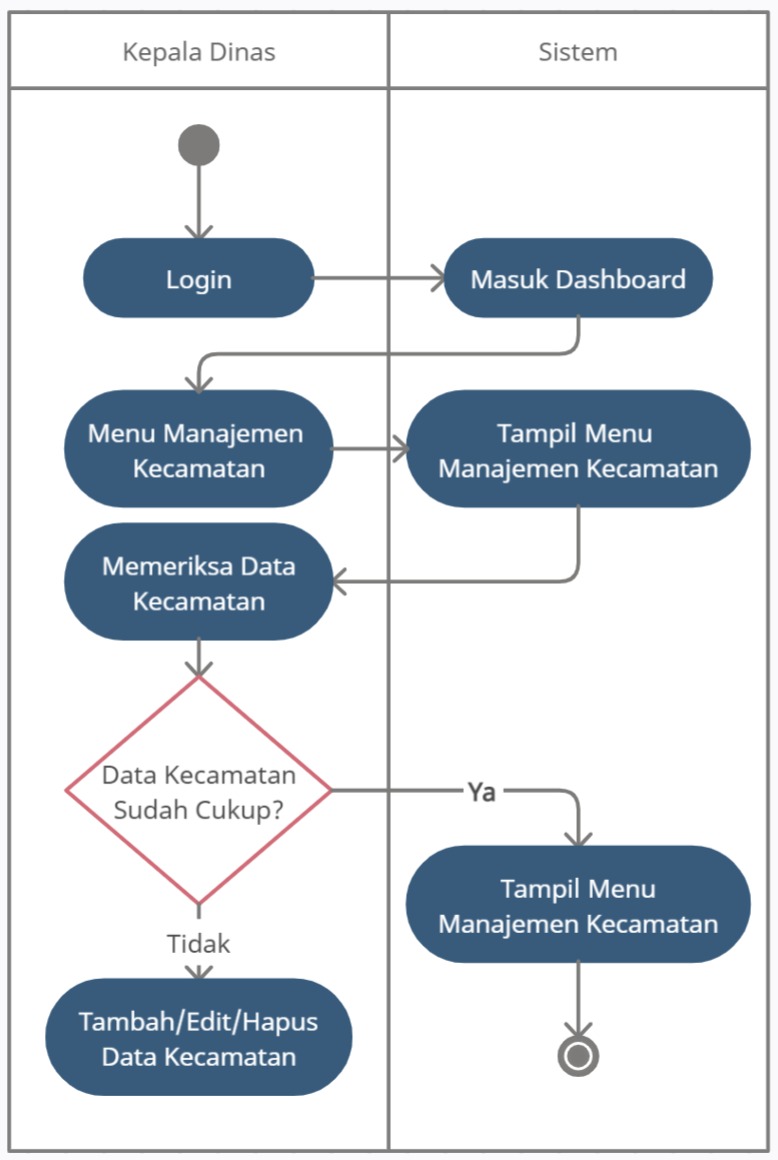
*Activity diagram* manajemen *user* sistem ini dimulai dari kepala dinas *login* ke sistem dan apabila *login* berhasil maka akan masuk ke halaman *dashboard*. Kepala dinas masuk ke *dashboard* dan dapat mengakses menu manajemen *user* lalu dapat memeriksa data *user*. Data *user* telah cukup maka akan ditampilkan disistem. Data *user* tidak cukup maka dapat dilakukan tambah, edit, atau hapus *user.*



**Gambar 3.7Activity diagram manajemen user**

1. Manajemen Kecamatan

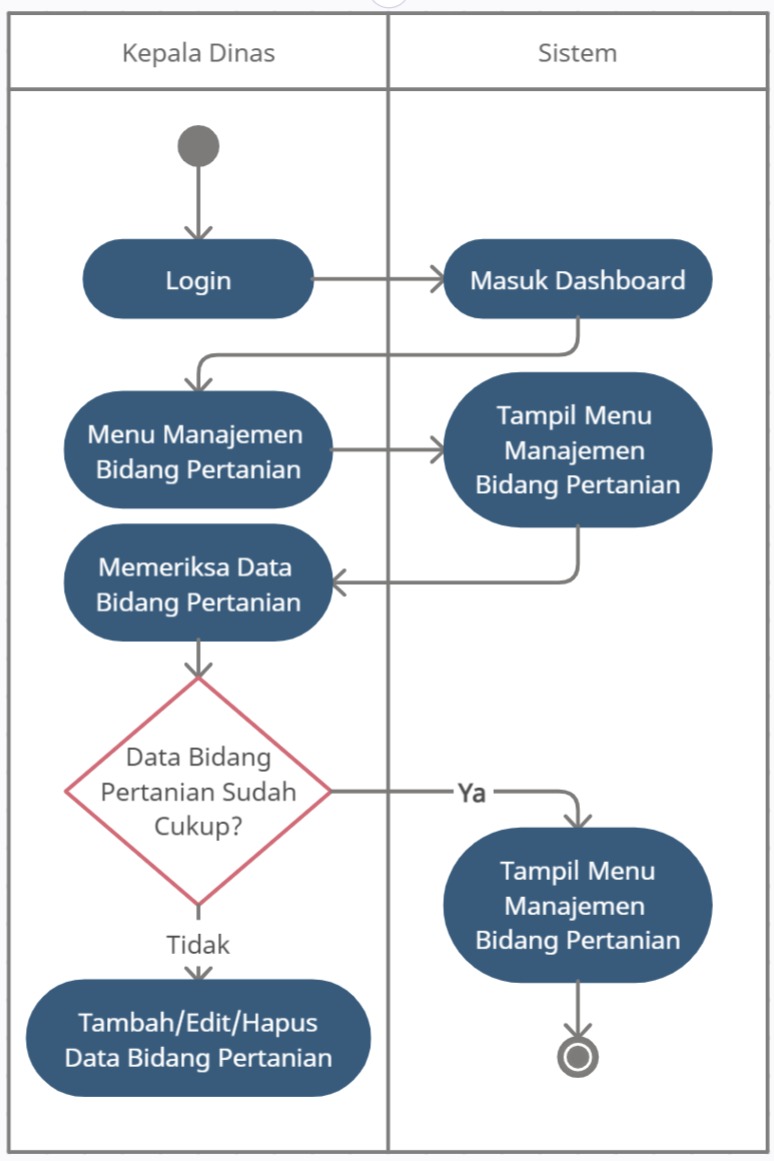
*Activity diagram* manajemen kecamatan dimulai dari *login* ke dalam sistem lalu masuk ke *dashboard* dan kepala dinas dapat menjangkau menu manajemen kecamatan. Setelah masuk, kepala dinas dapat memeriksa data di mana data telah cukup maka data kecamatan akan ditampilkan sistem. Data belum cukup maka kepala dinas dapat melakukan tambah data kecamatan, menghapus, dan mengedit data kecamatan.



**Gambar 3.8Activity diagram manajemen kecamatan**

1. Manajemen Bidang Pertanian

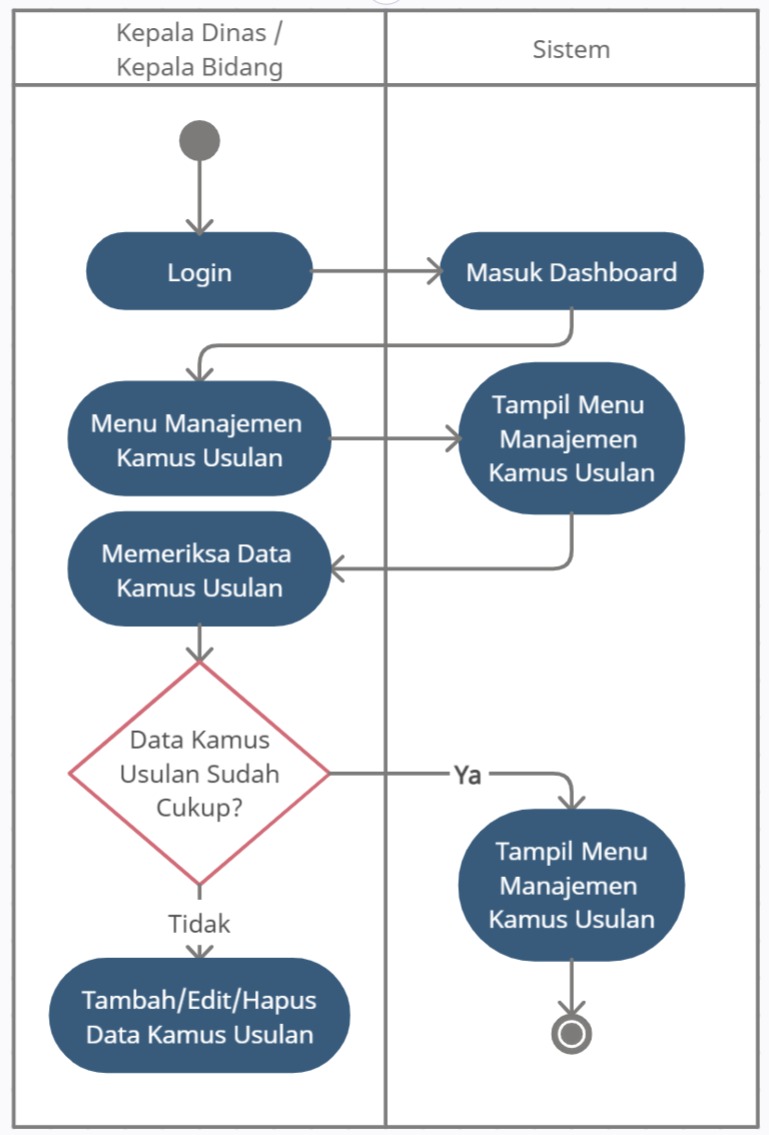
*Activity diagram* manajemen bidang pertanian dimulai dari *login* terlebih dahulu ke dalam sistem. Masuk ke *dashboard* dan kepala dinas dapat menjangkau menu manajemen bidang pertanian. Setelah masuk, kepala dinas dapat memeriksa data di mana data telah cukup maka data bidang pertanian akan ditampilkan di sistem. Data belum cukup maka kepala dinas dapat melakukan tambah data bidang pertanian, menghapus, dan mengedit data bidang pertanian.



**Gambar 3.9 Activity diagram manajemen bidang pertanian**

1. Manajemen kamus usulan

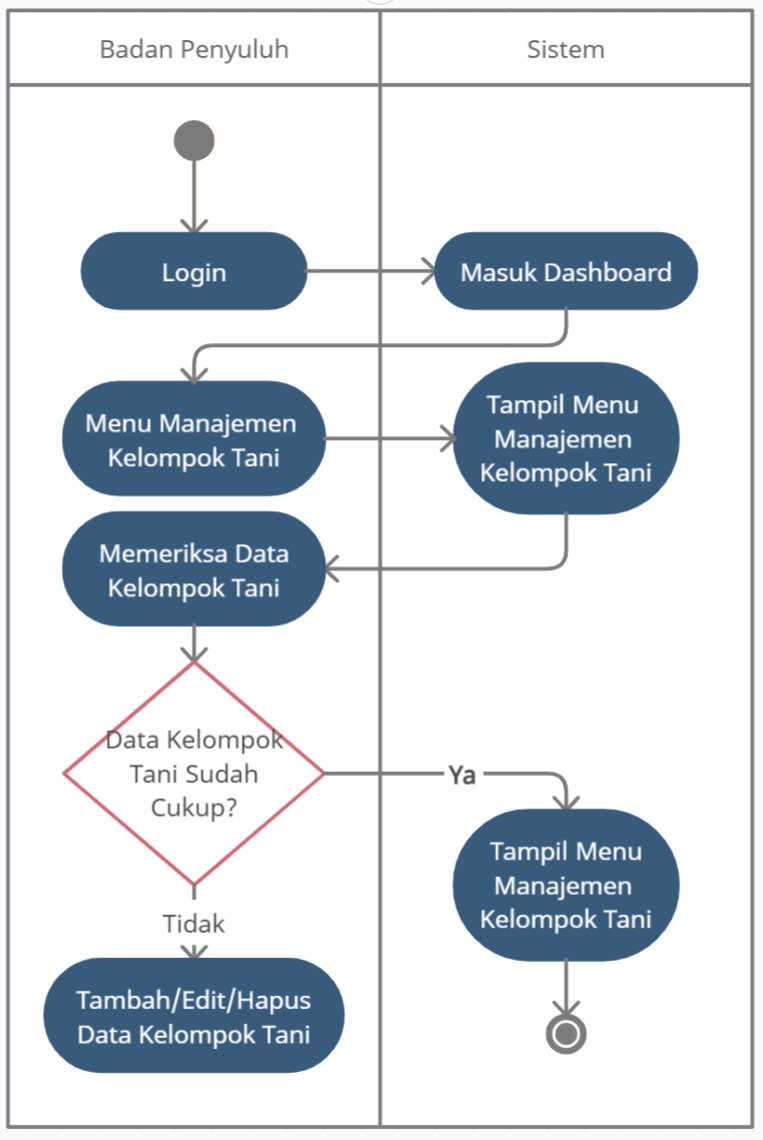
*Activity diagram* kamus usulan dimulai dari kepala dinas atau kepala bidang masuk ke sistem dan berhasil menuju *dashoard*. Masuk ke *dashboard* dan kepala dinas dapat menjangkau menu manajemen kamus usulan. Setelah masuk, kepala dinas dapat memeriksa data di mana data yang telah cukup di kamus usulan maka akan ditampilkan di sistem. Data belum cukup maka kepala dinas dapat melakukan tambah data kamus usulan, menghapus, dan mengedit data kamus usulan.



**Gambar 3.10 Activity diagram manajemen kamus usulan**

1. Manajemen Kelompok Tani

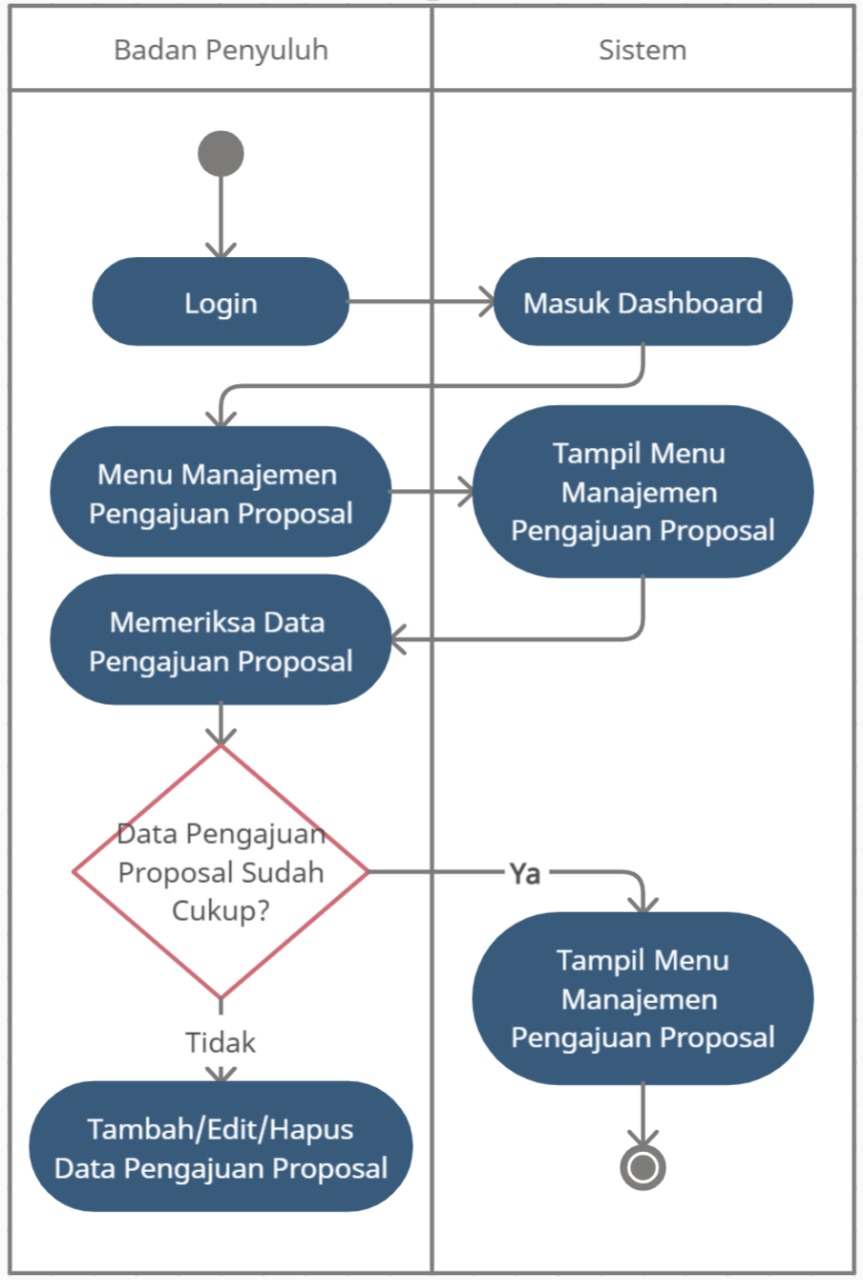
*Activity diagram* manajemen kelompok tani dimulai dari *login* ke sistem lalu masuk menuju *dashboard*. Badan penyuluh masuk dan dapat mengakses menu manajemen kelompok tani lalu dapat memeriksa data kelompok tani. Datakelompok tani telah cukup maka akan ditampilkan di sistem. Data kelompok tani tidak cukup maka dapat dilakukan tambah, edit, atau hapus data kelompok tani*.*



**Gambar 3.11 Activity diagram manajemen kelompok tani**

1. Pengajuan Proposal oleh Badan Penyuluh

*Activity diagram* pengajuan proposal oleh badan penyuluh dimulai dari *login* masuk ke *dashboard*. Badan penyuluh dapat mengakses menu pengajuan proposal tani lalu dapat memeriksa data kelompok tani. Datakelompok tani yang telah cukup akan ditampilkan di sistem. Data kelompok tani yang belum cukup maka dapat dilakukan tambah, edit, atau hapus proposal kelompok tani.*.*

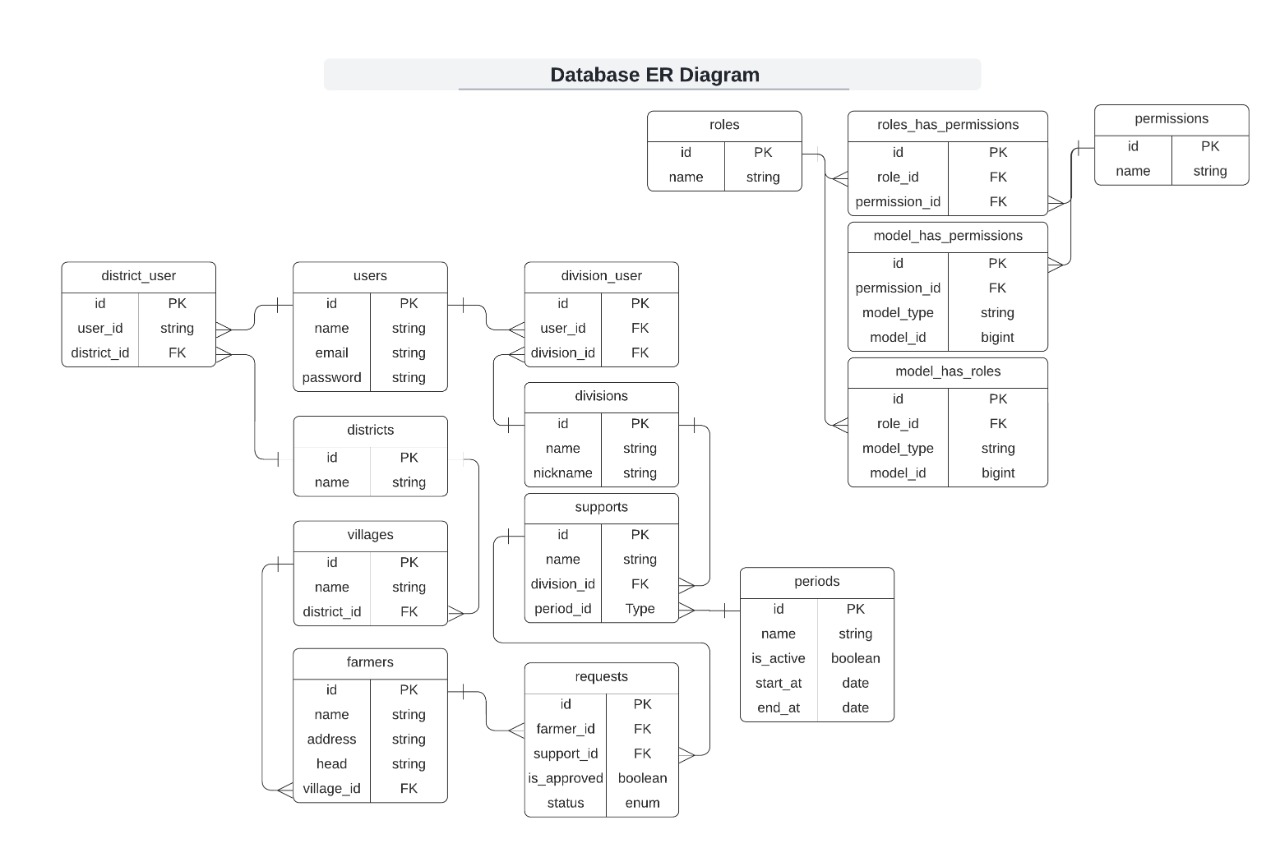


**Gambar 3.12 Activity diagram pengajuan proposal**

1. **Rancangan *Database***

ERD (*Entity Relation Diagram*) merupakan model yang digunakan untuk merancang basis data sehingga relasi data dapat digambarkan. Rancangan yang dibuat masih memungkinkan ada perubahan pada struktur diagram. Entitas yang digunakan pada *database* sistem ini adalah *roles, roles\_has\_permission, permission, model\_has\_permission, models\_has\_roles, district\_user, users, division\_users, districs, division, villages, supports, farmers, request, periods.*

*Roles* memiliki relasi *one to many* terhadap entitas *roles\_has\_permission* dan *model\_has\_roles.*  Entitas *permission* memiliki relasi *one to many* terhadap entitas *roles\_has\_permission* dan *models\_has\_roles.* Entitas *users* memiliki relasi *one to many* terhadap entitas *district\_user* dan *division\_user*. Entitas *district* memiliki relasi *one to many* terhadap entitas *villages* dan *district\_user.* Entitas *division* memiliki relasi *one to many* terhadap entitas *division\_user* dan *supports.* Entitas *periods* memili relasi *one to many* terhadap entitas s*upports*. Entitas *farmers* memiliki relasi *one to many* terhadap entitas *request*.

****

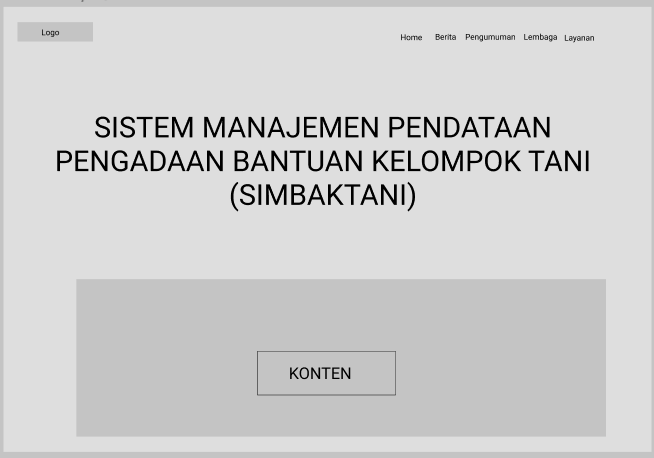
**Gambar 3.13 ER Diagram**

1. **Rancangan Interface**

Rancangan *interface* pada pembangunan sistem manajemen pengadaan bantuan ini sangat penting karena *interface* akan menghubungkan pengguna dengan sistem. Interaksi pengguna dengan sistem akan menciptakan *user experience* yang berkaitan erat dengan keberhasilan sistem. Rancangan *interface* *low-fidelity* pada pengembangan sistem pengadaan bantuan kelompok tani adalah sebagai berikut.

* + 1. Tampilan *Home*

Tampilan *home* merupakan halaman utama saat user mengakses sistem informasi pengadaan bantuan kelompok tani. Terdapat beberapa fitur dalam halaman ini yaitu logo, *home*, berita, pengumuman, dan Lembaga. Tampilan *home* dapat dilihat pada gambar 3.15.



**Gambar 3.14 Tampilan home**

* + 1. Tampilan *Dashboard*

Tampilan dashboard ini akan ditampilkan apabila *user* berhasil masuk ke dalam sistem. Fitur dalam sistem ini adalah manajemen *user*, manajemen kecamatan, manajemen bidang pertanian, manajemen kamus usulan, manajemen kelompok tani, manajemen proposal, manajemen proposal, dan manajemen laporan. Gambar 3.16 merupakan gambar tampilan *dashboard.*

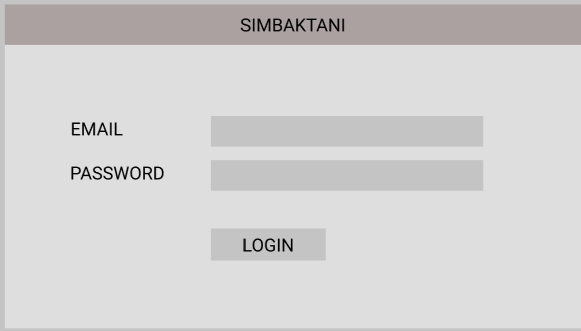


**Gambar 3.15 Tampilan beranda**

* + 1. Tampilan *login*

Tampilan *login* merupakan tampilan saat *user* hendak masuk ke sistem.

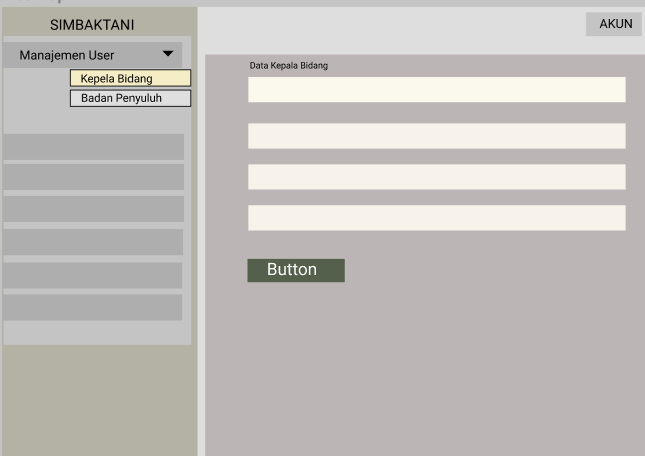
Tampilan *login* ini memiliki fitur *email* dan *password.*



**Gambar 3.16 Tampilan login**

* + 1. Tampilan manajemen *user*

Tampilan manajemen *user* ini memiliki fitur menambah data kepala bidang, menambah data badan penyuluh, dan melihat data kepala bidang dan badan penyuluh.



**Gambar 3.17Tampilan manajemen user**

### Tahap Implementasi

Implementasi merupakan proses menuliskan rancangan pada tahap perancangan kedalam code program. Implementasi dilakukan dengan pendekatan *Test Driven Development* (TDD). TDD memiliki tiga tahap diantaranya *unit testing, code generation,* dan *refactoring* yang akan diterapkan secara berulang pada masing-masing *user stories.* Tiga tahap TDD dibuat dalam bentuk implementasi kode dan dilakukan pengujian setelahnya. Bagian *refactoring* digunakan untuk optimasi kode sehingga dapat digunakan apabila diperlukan saja [38]. Tahap implementasi ini digunakan untuk menuangkan semua hasil perancangan dari setiap iterasi ke dalam kode sehingga sistem dapat digunakan oleh *client* dan diimplementasikan di Dinas Pertanian Kabupaten Toba.

### Tahap Retrospektif

Tahapan ini melakukan verifikasi terhadap semua *user stories* yang telah diimplementasikan dan dilakukan pengujian. Verifikasi dilakukan untuk perbandingan waktu estimasi dengan waktu realisasi sehingga dapat diketahui kendala-kendala penyebab *over* atau *under* estimasi pada pelaksanaan penelitian. Verifikasi ini bertujuan untuk mencegah perbedaan waktu estimasi pada penelitian selanjutnya.

### Tahap pengujian sistem

Tahapan ini merupakan pengujian hasil implementasi. Pengujian dilakukan oleh Kepala Dinas, kepala bidang, badan penyuluh Dinas pertanian Kabupaten Toba didampingi oleh pengembang. *Client* menguji fitur apakah sesuai dengan kebutuhan awal pada tahap analisis dan perencanaan.

#### Pengujian User Acceptance Test (UAT)

Pengujian *User Acceptance Test* merupakan pengujian yang berfungsi untuk memverifikasi bahwa fitur yang dibangun sudah sesuai dengan kebutuhan user. Pengujian ini dilakukan ditahap atau fase akhir setelah sistem selesai dalam pengembangan. UAT juga merupakan pengujian final *software* yang dikembangkan sebelum *software* digunakan [39]. Pengujian UAT ini juga dapat menghasilkan dokumen yang dijadikan sebagai bukti bahwa perangkat lunak yang dikembangkan telah diterima oleh user [40].

# BAB IV HASIL DAN PEMBAHASAN

## Lingkungan Pembangunan Sistem

Pada bagian ini akan dijabarkan proses iterasi yang dilakukan pada sub bab 3.4.3

## Hasil Analisa Penelitian

Analisis hasil penelitian dituangkan berdasarkan iterasi yang sudah dijabarkan pada bab 3. Penelitian ini memiliki 4 iterasi yang telah diimplementasikan sehingga menghasilkan sebuah sistem manajemen pendataan pengadaan bantuan untuk kelompok tani berbasis web menggunakan metode *Personal Extreme Programming* (PXP). Iterasi diperoleh dari *user stories* yang telah dikelompokkan dari prioritas *user stories*, estimasi waktu pengerjaan, dan perencanaan iterasi berdasarkan *stories point.*

### Iterasi ke Satu

1. **Inisiasi Iterasi**

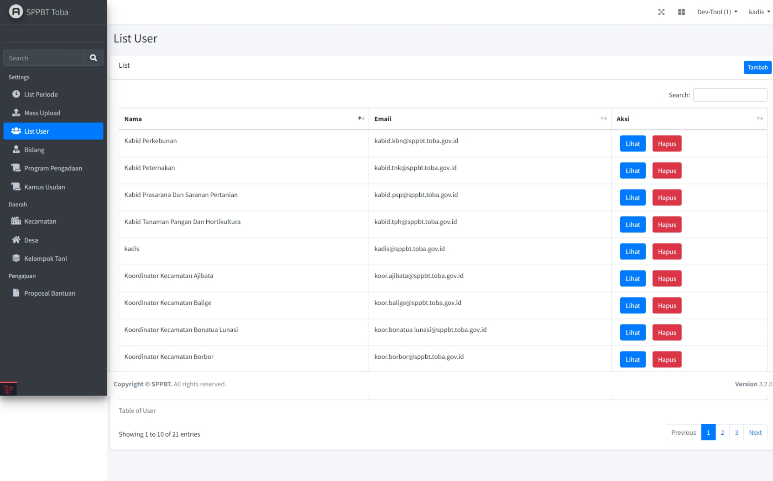
Iterasi pertama memiliki aktor yaitu Kepala Dinas. Iterasi pertama ini memiliki 3 *user stories* dengan estimasi waktu pengerjaan adalah 12 hari. Menu yang dihasilkan pada iterasi pertama ini adalah menu pengaturan dan artikel untuk admin, serta menu kependudukan untuk kasi pemerintahan. Tabel 4.2 adalah daftar dari inisiasi Iterasi 1.

Tabel 4.1 User Stories Iterasi 1

|  |  |  |  |
| --- | --- | --- | --- |
| Iterasi 1 | | | |
| *User Stories* | *Priority* | *Stories point* | Estimasi waktu (Hari) |
| Sebagai kepala dinas saya ingin dapat menambah user ke dalam sistem, mengedit data user, menghapus user sehingga apabila ada pergantian perangkat di Dinas Pertanian, pengaturan user dapat dilakukan. | *Must have* | 3 | 6 |
| Sebagai kepala dinas saya ingin dapat menambah kecamatan sehingga apabila ada pemekaran dan penambahan kecamatan maka dapat ditambahkan ke dalam sistem. | *Must have* | 3 | 6 |
| *velocity* | | 6 | 12 |

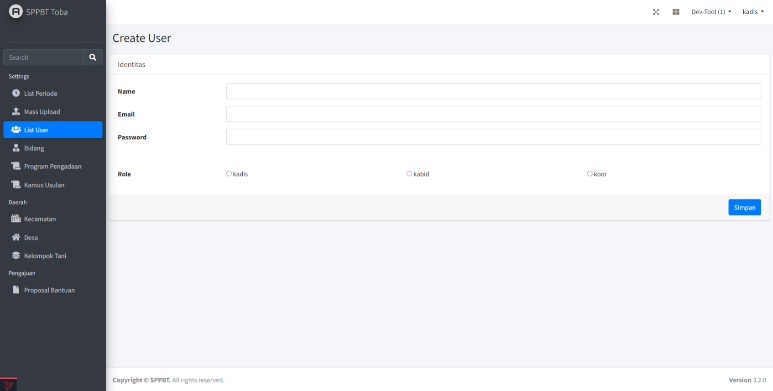
1. **Desain**

Tampilan sistem yang dibangun pada iterasi 1 ini adalah sebagai berikut.



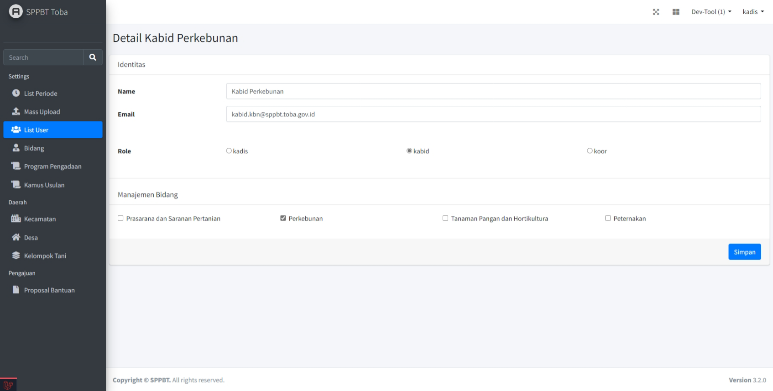
Gambar 4.1Tampilan List User

[Gambar 4. 1](#_bookmark118) merupakan tampilan yang menunjukkan tabel yang berisi seluruh user yang terdaftar di dalam sistem. Pada bagian ini kepala dinas dapat menghapus dan melihat detail pengguna secara langsung. Kepala dinas dapat menambah pengguna baru dengan menekan tombol tambah di pojok kanan atas. Tombol tambah akan mengarahkan pengguna ke halaman tambah pengguna.



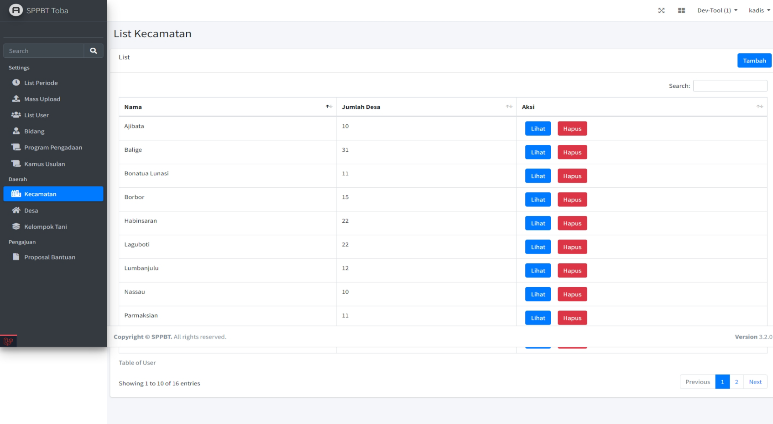
Gambar 4.2Tampilan Tambah Pengguna

[Gambar 4.](#_bookmark118) 2 merupakan tampilan yang menunjukkan *form* untuk menambah pengguna baru. Pengguna baru diharuskan memiliki atribut berupa nama, email, dan password. Ketika ingin menambahkan pengguna baru, pengguna baru tidak boleh tidak memiliki *role*, dan *role* tertentu memiliki beberapa bidang tanggung jawab. Jika *role* pengguna baru adalah kepala bidang, maka *form* akan menampilkan *form* baru yang berupa *checkbox* untuk menentukan divisi si pengguna baru. Jika *role* pengguna baru adalah badan penyuluh, maka *form* akan menampilkan *form* baru yang berupa *checkbox* untuk menentukan kecamatan si pengguna baru. Pengguna baru dapat dibuat dengan mengisi semua *form* yang ada dan menekan tombol simpan di kanan bawah.



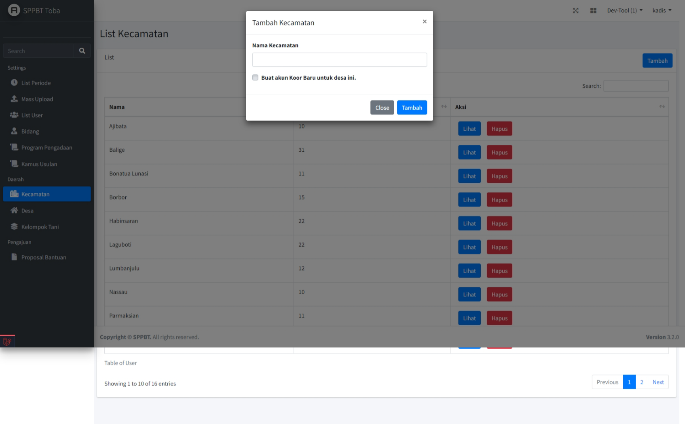
Gambar 4.3Tampilan Detail dan Ubah Pengguna

[Gambar 4.](#_bookmark118) 3 merupakan tampilan halaman detail pengguna yang dapat diakses dengan cara menekan tombol lihat pada tabel pengguna di halaman *list* pengguna. Pada halaman ini ditampilkan atribut pengguna beserta *role* dan divisi atau kecamatan yang terkait. Kepala dinas dapat memperbarui informasi pengguna melalui halaman ini dengan mengubah data yang sudah ada di dalam *form* pada halaman dan menekan tombol simpan yang ada di bagian kanan bawah.



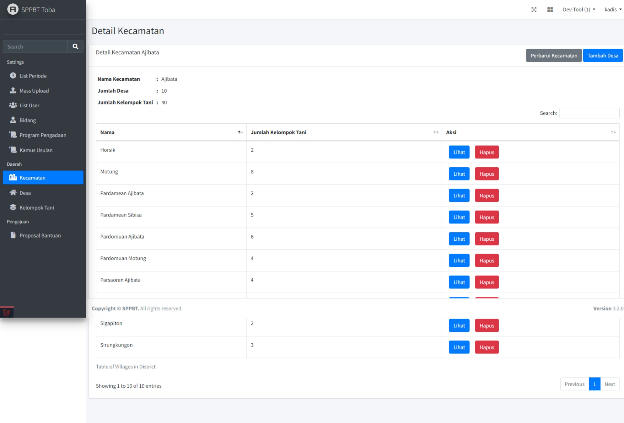
Gambar 4.4 Tampilan List Kecamatan

[Gambar 4.](#_bookmark118) 4 merupakan tampilan yang menunjukkan tabel yang berisi seluruh Kecamatan yang terdaftar di dalam sistem. Pada bagian ini kepala dinas dapat menghapus dan melihat detail kecamatan secara langsung. Kepala dinas dapat menambah pengguna baru dengan menekan tombol tambah di pojok kanan atas. Tombol tambah akan membuka sebuah *modal* yang berisi *form* untuk menambahkan data kecamatan baru.



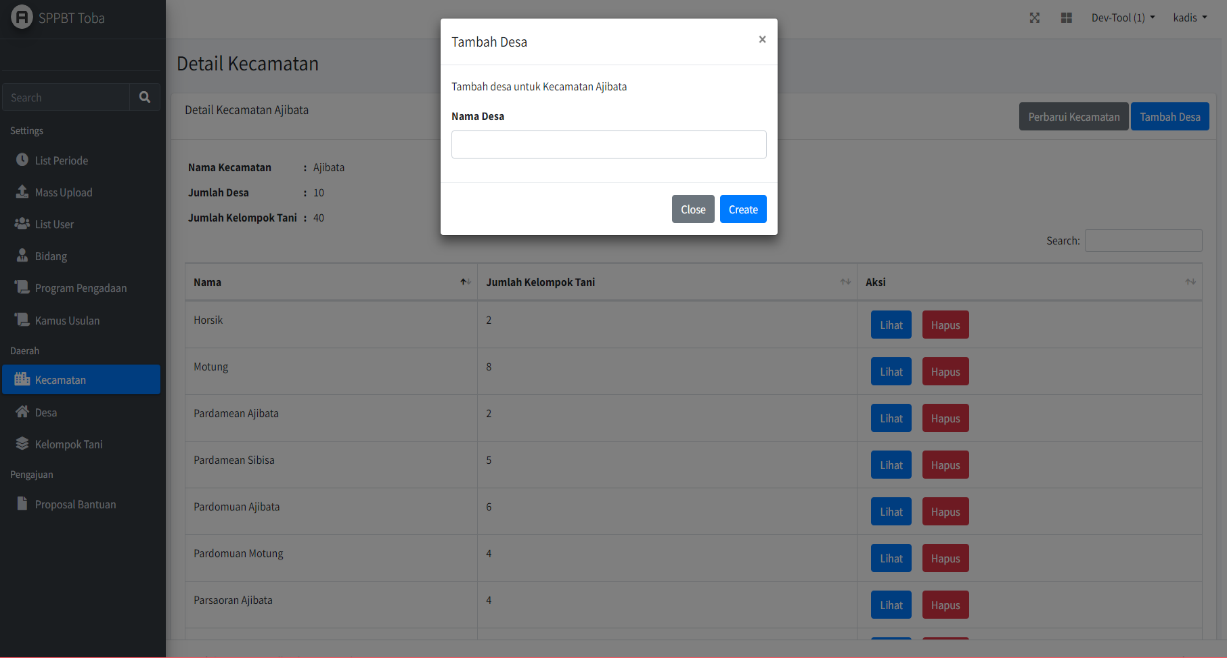
Gambar 4.5 Tampilan Tambah Kecamatan

[Gambar 4.](#_bookmark118) 5 merupakan tampilan yang menunjukkan munculnya *form* penambahan kecamatan jika menekan tombol tambah di pojok kanan atas pada halaman *list* kecamatan.



Gambar 4.6 Tampilan Detail Kecamatan

[Gambar 4.](#_bookmark118) 6 merupakan tampilan yang menunjukkan tampilan detail dari kecamatan yang dapat diakses dengan menekan tombol lihat di tabel *list* kecamatan di halaman *list* kecamatan. Isi dari halaman ini menunjukkan atribut kecamatan beserta *list* seluruh desa yang merupakan bagian dari kecamatan yang di lihat yang tersimpan di dalam sistem. Dalam halaman ini, pengguna juga dapat menambahkan data desa untuk kecamatan yang sedang diakses dengan menekan tombol yang ada di pojok kanan atas.



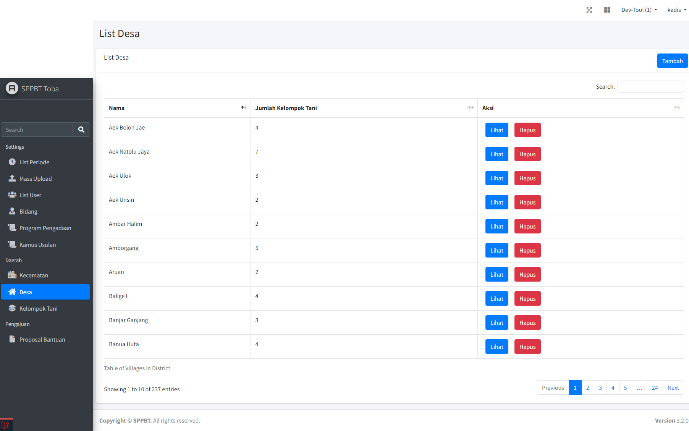
Gambar 4.7 Tampilan Tambah Desa untuk Kecamatan yang Sedang Diakses

[Gambar 4.](#_bookmark118) 7 merupakan tampilan yang menunjukkan munculnya *form* penambahan desa untuk kecamatan yang sedang diakses jika mengakses tombol tambah di pojok kanan atas halaman detail desa. *Form* ini bertujuan memberi aksesibilitas pada pengguna untuk menambahkan data desa baru sebagai bagian dari kecamatan yang sedang dilihat.



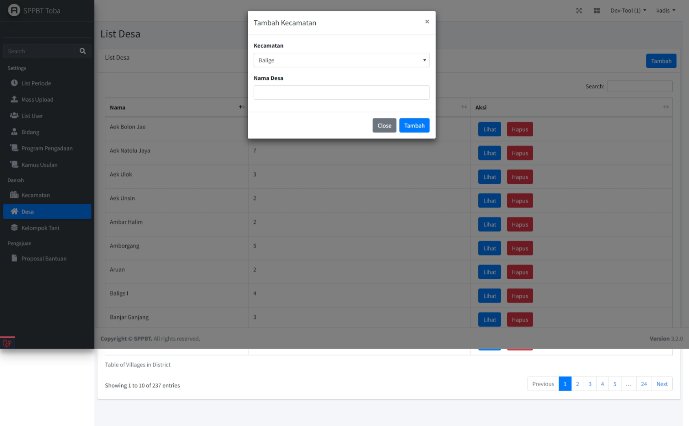
Gambar 4.8 Tampilan Detail Desa

[Gambar 4.](#_bookmark118) 8 merupakan tampilan halaman detail desa yang dapat diakses dengan menekan tombol detail di tabel *list* desa pada halaman detail kecamatan. Halaman ini berisi atribut yang dimiliki oleh desa.



Gambar 4.9 Tampilan List Desa

[Gambar 4.](#_bookmark118) 9 merupakan tampilan halaman *list* Desa. Halaman ini berisi *list* seluruh data desa yang tersimpan di dalam sistem. Pengguna dapat menambahkan data desa baru dengan menekan tombol tambah di kanan atas.



Gambar 4.10 Tampilan Tambah Desa

[Gambar 4.](#_bookmark118) 10 merupakan tampilan yang menunjukkan munculnya *form* penambahan desa. Form ini digunakan untuk menambahkan desa saat ada penambahan/pemekaran desa baru.

1. **Implementasi**

Implementasi dalam iterasi pertama meenerapkan Test Driven Drive (TDD) dalam prosesnya. TDD memiliki tiga tahapan yaitu unit testing, code generation, dan refactoring. Unit testing pada iterasi 1 adalah sebagai berikut.

* 1. Unit Tes

Pengujian fungsi yang sedang dikembangkan diperlukan sebagai dasar acuan sistem yang akan dibangun seperti yang dijelaskan pada Tabel 4.2

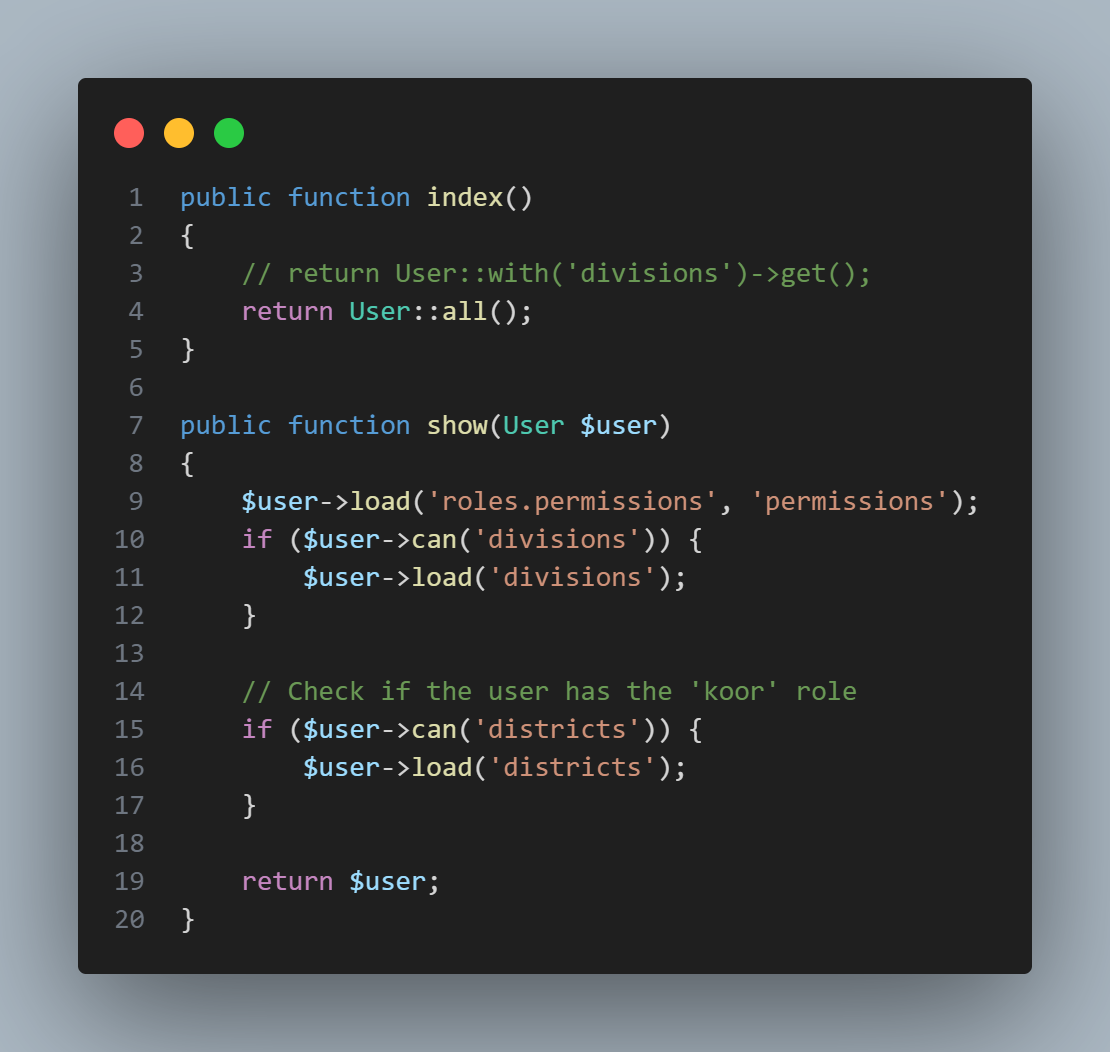
Tabel 4.2 Skenario Pengujian Iterasi 1

| *Kode User Story* | *Kode Skenario Pengujian* | *Skenario Pengujian* | *Aktor* | *Hasil yang diharapkan* |
| --- | --- | --- | --- | --- |
| US-01 | US-01-TS-01 | Kepala dinas menambahkan pengguna baru dengan *role* Kepala Dinas. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria tidak diperbolehkan | Kepala Dinas | Pengguna baru berhasil ditambahkan dan dapat melakukan autentikasi. |
| US-01 | US-01-TS-02 | Kepala dinas menambahkan pengguna baru dengan *role* Kepala Dinas. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria diperbolehkan | Kepala Dinas | Penambahan pengguna baru gagal. |
| US-01 | US-01-TS-03 | Kepala dinas menambahkan pengguna baru dengan *role* kepala bidang serta divisi yang menjadi tanggung jawab kepala bidang tersebut. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria tidak diperbolehkan | Kepala Dinas | Pengguna baru berhasil ditambahkan dan dapat melakukan autentikasi. |
| US-01 | US-01-TS-04 | Kepala dinas menambahkan pengguna baru dengan *role* kepala bidang serta divisi yang menjadi tanggung jawab kepala bidang tersebut. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria diperbolehkan | Kepala Dinas | Penambahan pengguna baru gagal. |
| US-01 | US-01-TS-05 | Kepala dinas menambahkan pengguna baru dengan *role* badan penyuluh serta wilayah kecamatan yang menjadi tanggung jawab badan penyuluh tersebut. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria diperbolehkan | Kepala Dinas | Pengguna baru berhasil ditambahkan dan dapat melakukan autentikasi. |
| US-01 | US-01-TS-06 | Kepala dinas menambahkan pengguna baru dengan *role* badan penyuluh serta wilayah kecamatan yang menjadi tanggung jawab badan penyuluh tersebut. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria tidak diperbolehkan. | Kepala Dinas | Penambahan pengguna baru gagal. |
| US-01 | US-01-TS-07 | Kepala dinas mengubah *role* pengguna. | Kepala Dinas | Pengguna dapat melakukan autentikasi dengan role yang sudah diubah. |
| US-01 | US-01-TS-08 | Kepala dinas mengubah bidang yang menjadi tanggung jawab pengguna dengan *role* kepala bidang. | Kepala Dinas | Pengguna dapat melakukan autentikasi dan memiliki perubahan bidang. |
| US-01 | US-01-TS-09 | Kepala dinas mengubah wilayah yang menjadi tanggung jawab pengguna dengan *role* badan penyuluh . | Kepala Dinas | Pengguna dapat melakukan autentikasi dan memiliki perubahan wilayah. |
| US-01 | US-01-TS-10 | Kepala dinas mengubah kredensial pengguna menggunakan format email dan kata sandi yang tidak sesuai dengan kriteria tidak diperbolehkan. | Kepala Dinas | Pengguna dapat melakukan autentikasi dengan kredensial yang baru dan tidak dapat melakukan autentikasi dengan kredensial yang lama. |
| US-01 | US-01-TS-11 | Kepala dinas mengubah kredensial pengguna menggunakan format email dan kata sandi yang tidak sesuai dengan kriteria diperbolehkan. | Kepala Dinas | Pengguna tidak mengalami perubahan kredensial dan tetap dapat melakukan autentikasi dengan kredensial yang lama. |
| US-01 | US-01-TS-12 | Kepala dinas menghapus pengguna. | Kepala Dinas | Pengguna terhapus dari sistem dan pengguna tidak lagi dapat melakukan autentikasi. |
| US-02 | US-02-TS-01 | Kepala dinas menambah data kecamatan baru. | Kepala Dinas | Data Kecamatan baru berhasil dibuat. |
| US-02 | US-02-TS-02 | Kepala dinas menambah data desa untuk kecamatan tertentu. | Kepala Dinas | Data desa untuk kecamatan berhasil dibuat |

Terdapat 14 unit tes yang dijalankan pada iterasi ini*.* Pengembang dapat melanjutkan untuk memulai tahapan selanjutnya dalam implementasi ini, yaitu *code generator*.

* 1. *Code Generator*

Code generator dalam iterasi 1 ini dapat dilihat gambar berikut :



Gambar 4.11 Fungsi Halaman List dan Detail Pengguna



Gambar 4.12 Fungsi Tambah Pengguna



Gambar 4.13 Fungsi Ubah Pengguna



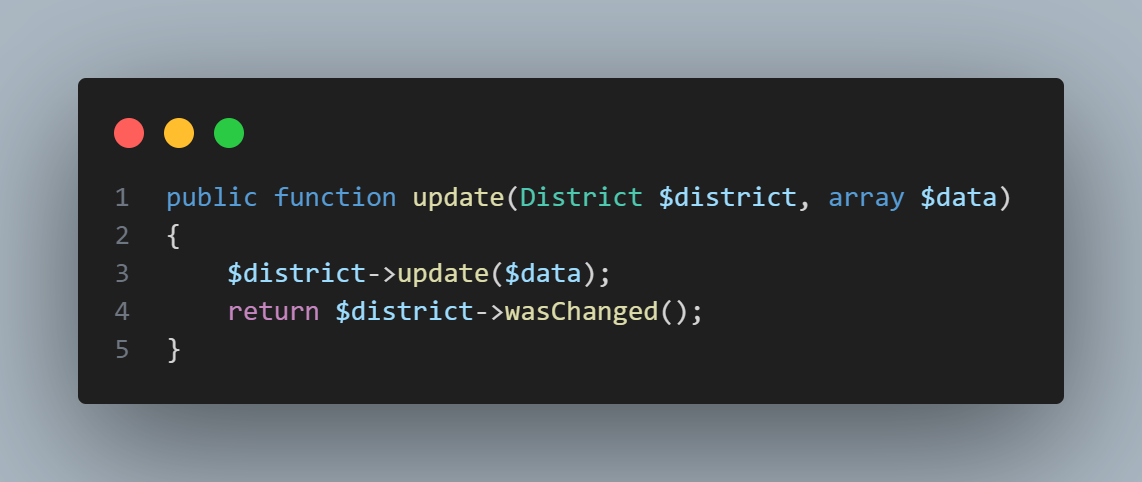
Gambar 4.14 Fungsi List Kecamatan



Gambar 4.15 Fungsi Tambah Kecamatan



Gambar 4.16 Fungsi Detail Kecamatan



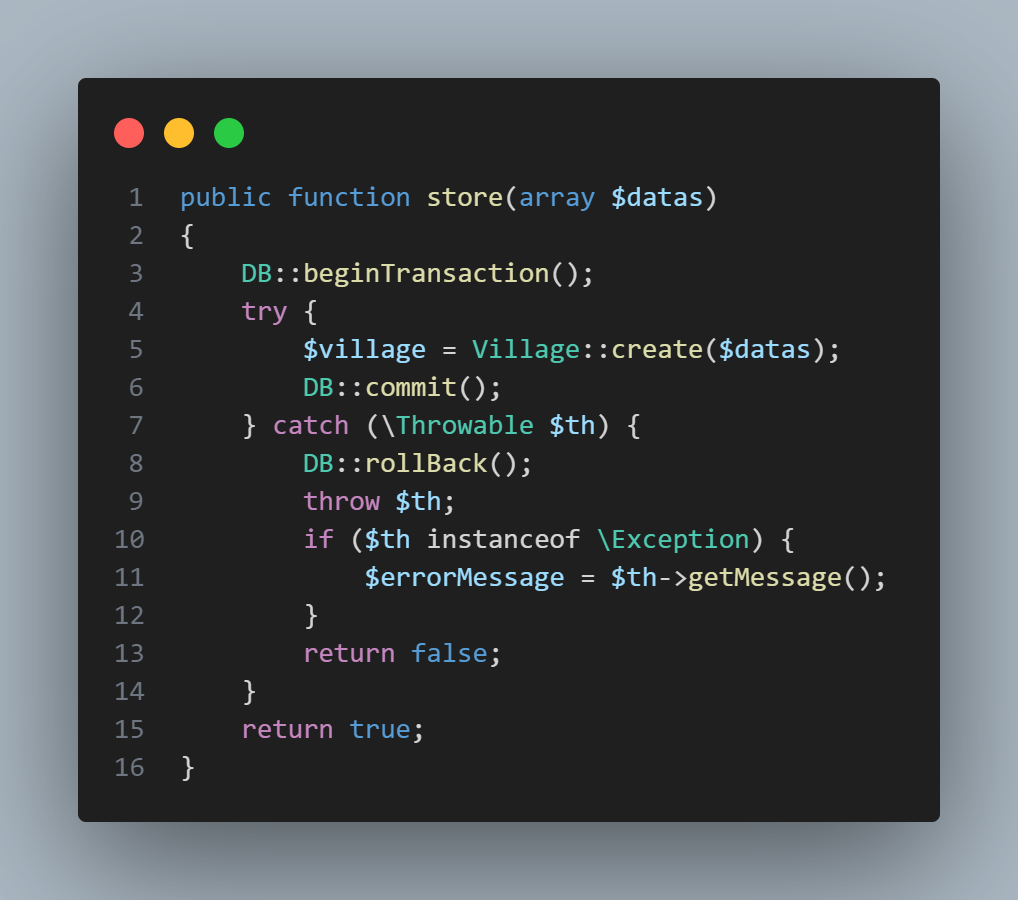
Gambar 4.17 Fungsi Ubah Kecamatan



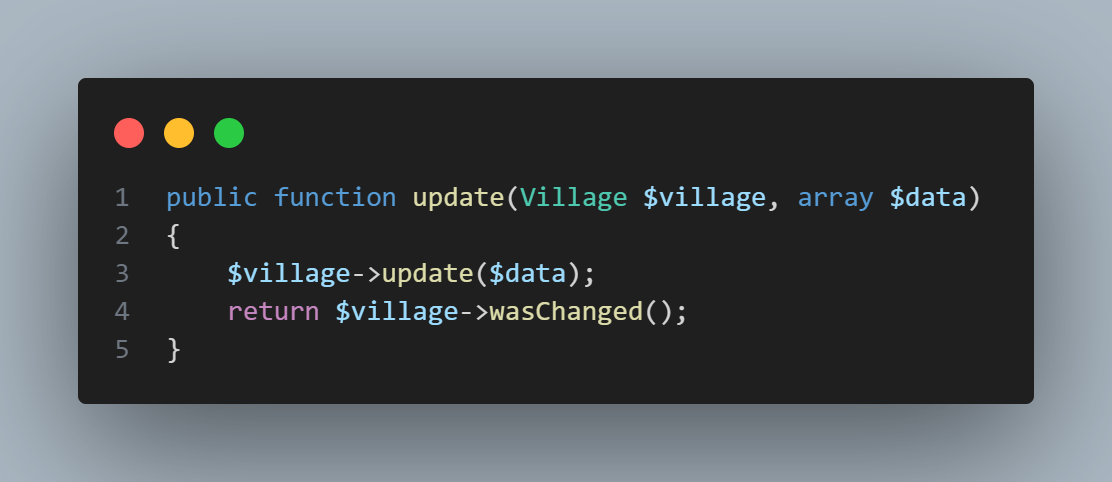
Gambar 4.18 Fungsi List Desa



Gambar 4.19 Fungsi Detail Desa



Gambar 4.20 Fungsi Tambah Desa



Gambar 4.21 Fungsi Ubah Desa

1. **Refactor**

*Refactor* merupakan tahap pengoptimasian kode dalam tahap implementasi program. Refaktor iterasi 1 dalam pembangunan sistem manajemen bantuan tani ini telah dilakukan pada saat pengimplementasian dalam kode program.

1. **Pengujian Sistem**

Tabel 4.3 merupakan tabel hasil pengujian dari implementasi dari iterasi 1. Pengujian dilakukan berdasarkan *user stories* yang sudah diperoleh sebelumnya. Pengujian ini menggunakan *User Acceptance Test* yang merupakan pengujian fungsionalitas sistem yang langsung diuji oleh *client*. Hasil pengujian dapat dilihat pada tabel 4.3.

Tabel 4.3 Hasil Pengujian Iterasi 1

| *Kode User Story* | *Kode Skenario Pengujian* | *Skenario Pengujian* | *Aktor* | *Hasil yang diharapkan* | *Status* |
| --- | --- | --- | --- | --- | --- |
| US-01 | US-01-TS-01 | Kepala dinas menambahkan pengguna baru dengan *role* Kepala Dinas. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria tidak diperbolehkan | Kepala Dinas | Pengguna baru berhasil ditambahkan dan dapat melakukan autentikasi. | Berhasil |
| US-01 | US-01-TS-02 | Kepala dinas menambahkan pengguna baru dengan *role* Kepala Dinas. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria diperbolehkan | Kepala Dinas | Penambahan pengguna baru gagal. | Berhasil |
| US-01 | US-01-TS-03 | Kepala dinas menambahkan pengguna baru dengan *role* kepala bidang serta divisi yang menjadi tanggung jawab kepala bidang tersebut. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria tidak diperbolehkan | Kepala Dinas | Pengguna baru berhasil ditambahkan dan dapat melakukan autentikasi. | Berhasil |
| US-01 | US-01-TS-04 | Kepala dinas menambahkan pengguna baru dengan *role* kepala bidang serta divisi yang menjadi tanggung jawab kepala bidang tersebut. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria diperbolehkan | Kepala Dinas | Penambahan pengguna baru gagal. | Berhasil |
| US-01 | US-01-TS-05 | Kepala dinas menambahkan pengguna baru dengan *role* badan penyuluh serta wilayah kecamatan yang menjadi tanggung jawab badan penyuluh tersebut. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria diperbolehkan | Kepala Dinas | Pengguna baru berhasil ditambahkan dan dapat melakukan autentikasi. | Berhasil |
| US-01 | US-01-TS-06 | Kepala dinas menambahkan pengguna baru dengan *role* badan penyuluh serta wilayah kecamatan yang menjadi tanggung jawab badan penyuluh tersebut. Penggunaan format email dan kata sandi yang tidak sesuai dengan kriteria tidak diperbolehkan. | Kepala Dinas | Penambahan pengguna baru gagal. | Berhasil |
| US-01 | US-01-TS-07 | Kepala dinas mengubah *role* pengguna. | Kepala Dinas | Pengguna dapat melakukan autentikasi dengan role yang sudah diubah. | Berhasil |
| US-01 | US-01-TS-08 | Kepala dinas mengubah bidang yang menjadi tanggung jawab pengguna dengan *role* kepala bidang. | Kepala Dinas | Pengguna dapat melakukan autentikasi dan memiliki perubahan bidang. | Berhasil |
| US-01 | US-01-TS-09 | Kepala dinas mengubah wilayah yang menjadi tanggung jawab pengguna dengan *role* badan penyuluh . | Kepala Dinas | Pengguna dapat melakukan autentikasi dan memiliki perubahan wilayah. | Berhasil |
| US-01 | US-01-TS-10 | Kepala dinas mengubah kredensial pengguna menggunakan format email dan kata sandi yang tidak sesuai dengan kriteria tidak diperbolehkan. | Kepala Dinas | Pengguna dapat melakukan autentikasi dengan kredensial yang baru dan tidak dapat melakukan autentikasi dengan kredensial yang lama. | Berhasil |
| US-01 | US-01-TS-11 | Kepala dinas mengubah kredensial pengguna menggunakan format email dan kata sandi yang tidak sesuai dengan kriteria diperbolehkan. | Kepala Dinas | Pengguna tidak mengalami perubahan kredensial dan tetap dapat melakukan autentikasi dengan kredensial yang lama. | Berhasil |
| US-01 | US-01-TS-12 | Kepala dinas menghapus pengguna. | Kepala Dinas | Pengguna terhapus dari sistem dan pengguna tidak lagi dapat melakukan autentikasi. | Berhasil |
| US-02 | US-02-TS-01 | Kepala dinas menambah data kecamatan baru. | Kepala Dinas | Data Kecamatan baru berhasil dibuat. | Berhasil |
| US-02 | US-02-TS-02 | Kepala dinas menambah data desa untuk kecamatan tertentu. | Kepala Dinas | Data desa untuk kecamatan berhasil dibuat | Berhasil |

Berdasarkan hasil tersebut maka tahap implementasi telah selesai dan siklus pengembangan dapat berlanjut ke tahapan retrospektif.

1. **Retrospektif**

Pada Penghujung iterasi dilakukan evaluasi terhadap pengembangan yang telah dijalankan seperti dijelaskan pada tabel 4.3

Tabel 4.4 Hasil Retrospektif Iterasi 1

|  |  |  |  |
| --- | --- | --- | --- |
| *Kode User Story* | *Estimasi (Hari)* | *Skala* | *Realisasi (Hari)* |
| US-01 | 6 | 3 | 6 |
| US-02 | 6 | 3 | 3 |

*User story* US-02 terjadi lebih cepat dari rencana hal ini dikarenakan adanya bantuan data kecamatan dan desa, serta arahan jelas dan rinci dari pihak dinas pertanian. Data lengkap kecamatan dan desa ini sangat membantu dalam menentukan struktur data penyimpanan dan dalam proses *seeding*. Arahan yang jelas dan rinci dari pihak dinas pertanian juga membantu pengembang dalam menggambarkan sistem yang diinginkan oleh klien. Dengan demikian, pelaksanaan iterasi ke satu dinyatakan selesai dengan total penggunaan waktu 9 hari.

### Iterasi ke Dua

1. **Inisiasi Iterasi**

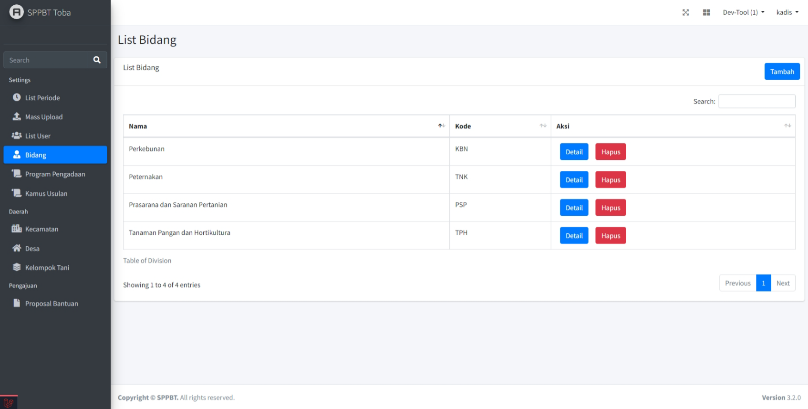
Pada tahap ini terdapat dua *user story* yang akan dijalankan. Kedua *user story* tersebut berkaitan dengan seluruh *user story* 3 (US-3) sampai *user story* 4(US-4). besaran *velocity* pada iterasi ini sebesar enam *point* yang berarti seluruh *user story* diharapkan selesai dalam kurun waktu 12 hari.

Tabel 4.5 User Stories Iterasi 2

|  |  |  |  |
| --- | --- | --- | --- |
| Iterasi 2 | | | |
| *User Stories* | *Priority* | *Stories point* | Estimasi waktu (Hari) |
| Sebagai kepala dinas saya ingin dapat menambah bidang sehingga mempermudah apabila ada perubahan struktural. | *Must have* | 3 | 6 |
| Sebagai kepala dinas saya ingin membuat rekapitulasi dan melihat proposal yang diajukan sehingga proposal bisa saya pantau. | *Must have* | 3 | 6 |
| *velocity* | | 6 | 12 |

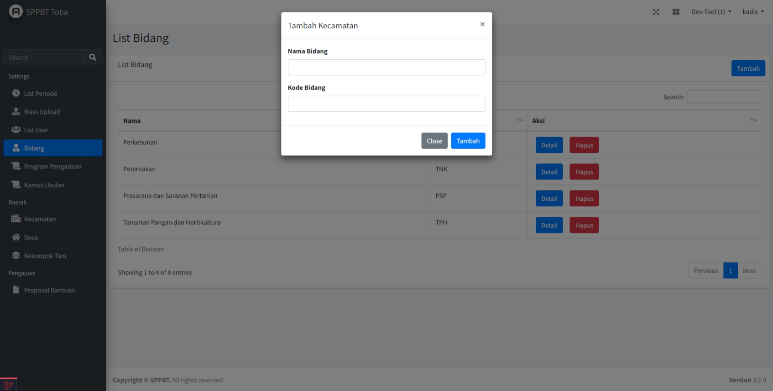
1. **Desain**

Tampilan sistem yang dibangun pada iterasi 1 ini adalah sebagai berikut.



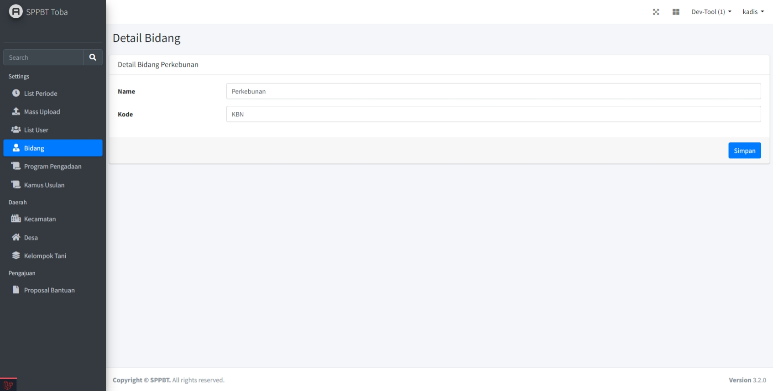
Gambar 4.22 Tampilan List Bidang

[Gambar 4.](#_bookmark118) 22 merupakan tampilan yang menunjukkan tabel yang berisi seluruh bidang yang tersimpan di dalam sistem. Pada bagian ini kepala dinas dapat menghapus dan melihat detail bidang secara langsung. Kepala dinas dapat menambah bidang baru dengan menekan tombol tambah di pojok kanan atas. Tombol tambah akan memunculkan *form* tambah bidang.



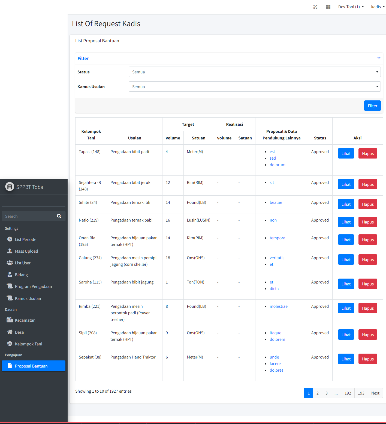
Gambar 4.23 Tampilan Tambah Bidang

[Gambar 4.](#_bookmark118) 23 merupakan tampilan yang menunjukkan *form* untuk menambah bidang baru. Bidang baru diharuskan memiliki atribut berupa nama dan kode bidang. Bidang baru dapat dibuat dengan mengisi semua *form* yang ada dan menekan tombol tambah di kanan bawah.



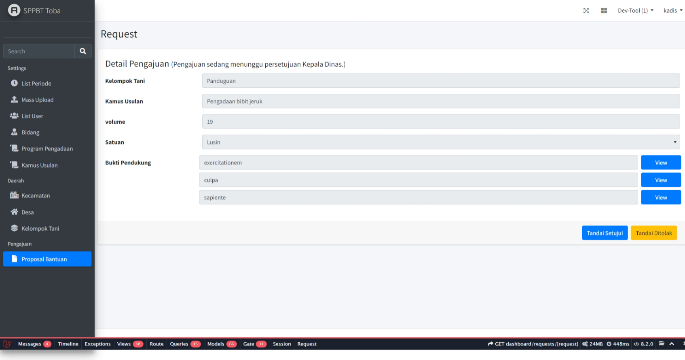
Gambar 4.24 Tampilan Detail Bidang

[Gambar 4.](#_bookmark118) 24 merupakan tampilan halaman detail bidang yang dapat diakses dengan cara menekan tombol lihat pada tabel bidang di halaman *list* bidang. Pada halaman ini pengguna terkait dapat memperbarui informasi bidang melalui halaman ini dengan mengubah data yang sudah ada di dalam *form* pada halaman ini dan menekan tombol simpan yang ada di bagian kanan bawah.



Gambar 4.25 Tampilan List Proposal Bagian Kepala Dinas

[Gambar 4.](#_bookmark118) 25 merupakan tampilan yang menunjukkan tabel yang berisi seluruh proposal bantuan yang masuk. Kepala dinas dapat melakukan *filtering* pada data proposal yang ada dengan menggunakan bantuan *filter* yang tersedia di halaman ini. Kepala dinas dapat menghapus data proposal pada halaman ini dengan menekan tombol hapus yang ada pada tabel *list* proposal.



Gambar 4.26 Tampilan Halaman Detail Proposal Bagian Kepala Dinas

Gambar 4.26 merupakan tampilan halaman detail proposal yang diajukan. Halaman ini dapat diakses dengan menekan tombol lihat yang ada pada tabel *list* proposal di halaman *list* proposal bagian kepala dinas. Kepala dinas hanya dapat mengubah atribut status proposal. Karena kepala dinas hanya dapat menyetujui atau menunda pelaksanaan bantuan dari apa yang ada di proposal. Kepala dinas dapat memutuskan untuk menyetujui dan menolak pelaksanaan bantuan dengan menekan tombol tandai setuju dan tombol tandai ditolak yang ada di bagian kanan bawah.

1. **Implementasi**

*User story* yang akan dijalankan pada tahap ini menyangkut kebutuhan fitur manajemen bidang dan manajemen proposal bagian kepala dinas. Fitur manajemen bidang bertujuan untuk menambah, mengedit dan menghapus bidang sedangkan manajemen proposal bagian kepala dinas bertujuan melihat, mengubah, dan merekapitulasi data proposal.

1. Unit Tes

Pengecekan fungsi yang sedang dikembangkan diperlukan sebagai dasar acuan sistem yang akan dibangun seperti yang dijelaskan pada [Tabel 4.](#_bookmark117) 3

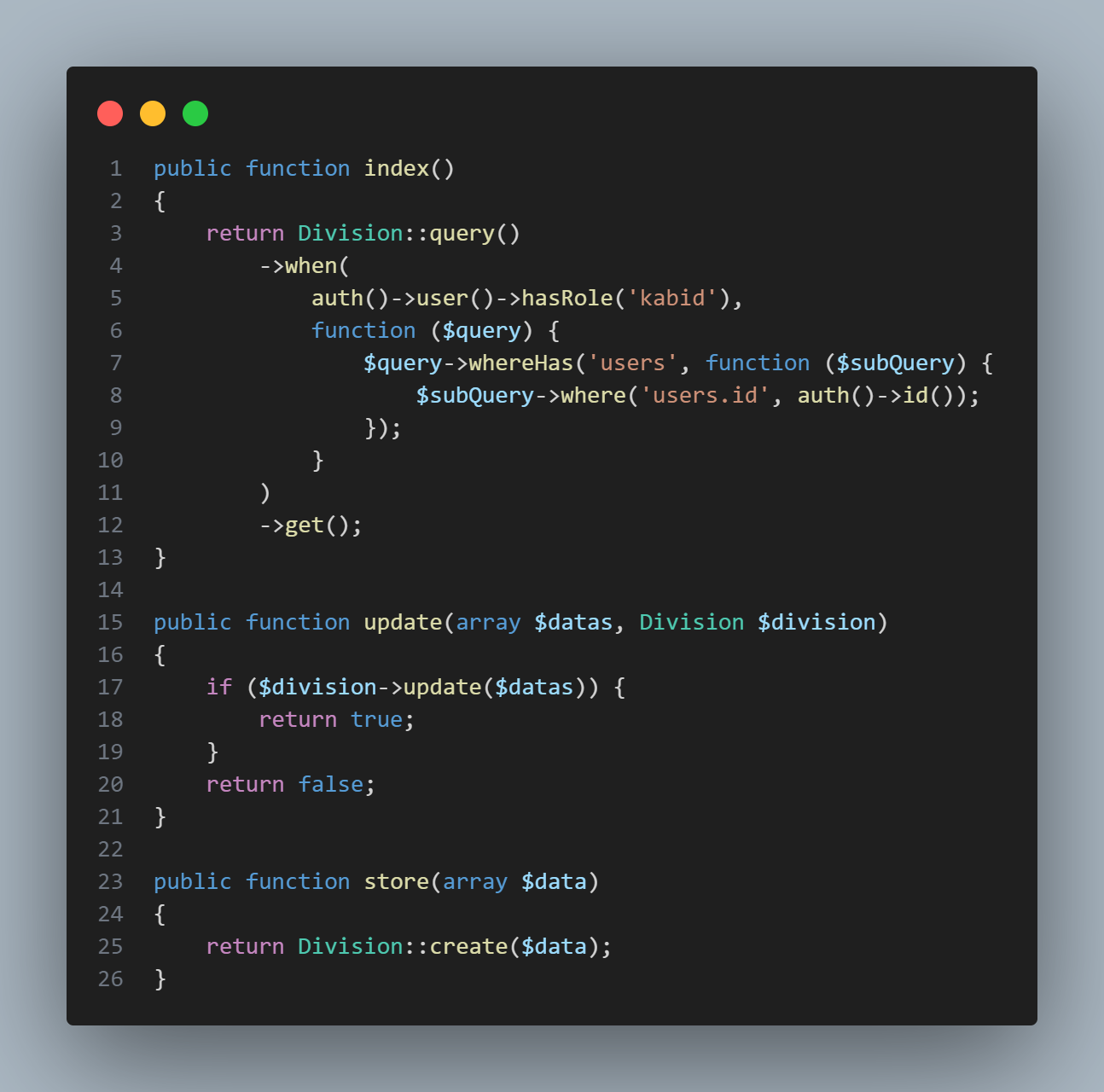
Tabel 4.6 Skenario Pengujian Iterasi 2

| *Kode User Story* | *Kode Skenario Pengujian* | *Skenario Pengujian* | *Aktor* | *Hasil yang diharapkan* |
| --- | --- | --- | --- | --- |
| US-03 | US-03-TS-01 | Kepala dinas menambahkan bidang baru. | Kepala Dinas | Data bidang baru berhasil ditambahkan. |
| US-03 | US-03-TS-02 | Kepala dinas mengubah atribut bidang. | Kepala Dinas | Data bidang berhasil diubah. |
| US-03 | US-03-TS-03 | Kepala dinas menghapus bidang. | Kepala Dinas | Data bidang yang diinginkan berhasil dihapus. |
| US-04 | US-04-TS-01 | Kepala dinas dapat mengakses semua proposal yang ada. | Kepala Dinas | Semua proposal atribut proposal dapat dilihat. |
| US-05 | US-04-TS-01 | Kepala dinas dapat mengubah status dari proposal yang ada. | Kepala Dinas | Semua proposal atribut proposal dapat dilihat. |

Terdapat 5 unit tes yang dijalankan pada iterasi ini*.* Pengembang dapat melanjutkan untuk memulai tahapan selanjutnya dalam implementasi ini, yaitu *code generator*.

1. Code Generator

Code generator dalam iterasi 1 ini dapat dilihat gambar berikut :



Gambar 4.27 Fungsi List, Ubah, dan Tambah Bidang



Gambar 4.28 Fungsi List Proposal Bagian Kepala Dinas



Gambar 4.29 Fungsi Detail Proposal Bagian Kepala Dinas



Gambar 4.30 Fungsi Perbarui Proposal

1. Refactory

Refactor merupakan tahap pengoptimasian kode dalam tahap implementasi program. Refaktor iterasi 2 dalam pembangunan sistem manajemen pengadaan bantuan tani ini telah dilakukan pada saat pengimplementasian dalam kode program.

1. **Pengujian Sistem**

Pengembang menyesuaikan kembali hasil dari pembuatan *code* dengan *unit testing* yang dibuat dan mendapati bahwa semua pengujian dinyatakan berhasil. Setelah melakukan debug, ternyata didapati bahwa penerapan fungsi.

Tabel 4.7 Pegujian Sistem

| *Kode User Story* | *Kode Skenario Pengujian* | *Skenario Pengujian* | *Aktor* | *Hasil yang diharapkan* | *Status* |
| --- | --- | --- | --- | --- | --- |
| US-03 | US-03-TS-01 | Kepala dinas menambahkan bidang baru. | Kepala Dinas | Data bidang baru berhasil ditambahkan. | Berhasil |
| US-03 | US-03-TS-02 | Kepala dinas mengubah atribut bidang. | Kepala Dinas | Data bidang berhasil diubah. | Berhasil |
| US-03 | US-03-TS-03 | Kepala dinas menghapus bidang. | Kepala Dinas | Data bidang yang diinginkan berhasil dihapus. | Berhasil |
| US-04 | US-04-TS-01 | Kepala dinas dapat mengakses semua proposal yang ada. | Kepala Dinas | Semua proposal atribut proposal dapat dilihat. | Berhasil |
| US-05 | US-04-TS-01 | Kepala dinas dapat mengubah status dari proposal yang ada. | Kepala Dinas | Semua proposal atribut proposal dapat dilihat. | Berhasil |

Berdasarkan hasil tersebut maka tahap implementasi telah selesai dan siklus pengembangan dapat berlanjut ke tahapan retrospektif.

1. Retrospektif

Pada Penghujung iterasi dilakukan evaluasi terhadap pengembangan yang telah dijalankan seperti dijelaskan pada tabel

Tabel 4.8 Retrospektif Iterasi 2

|  |  |  |  |
| --- | --- | --- | --- |
| *Kode User Story* | *Estimasi (Hari)* | *Skala* | *Realisasi (Hari)* |
| US-03 | 6 | 3 | 1 |
| US-04 | 6 | 3 | 8 |

*User story* US-03 terjadi lebih cepat dari rencana dikarenakan pada iterasi ke satu, struktur *database* dan beberapa inisiasi fungsi untuk fitur manajemen bidang telah selesai dikerjakan. Dengan begitu pengerjaan fitur manajemen bidang hanya tersisa tampilan dan fungsi aksi lainnya. *User Story* US-04 mengalami kendala di beberapa bagian, termasuk mendesain struktural database untuk mencocokan kembali alur antara sistem yang dibangun pengembang dengan yang diinginkan *client*. Dengan demikian, pelaksanaan iterasi ke dua dinyatakan selesai dengan total penggunaan waktu 9 hari.

### Iterasi ke Tiga

1. **Inisiasi Iterasi**

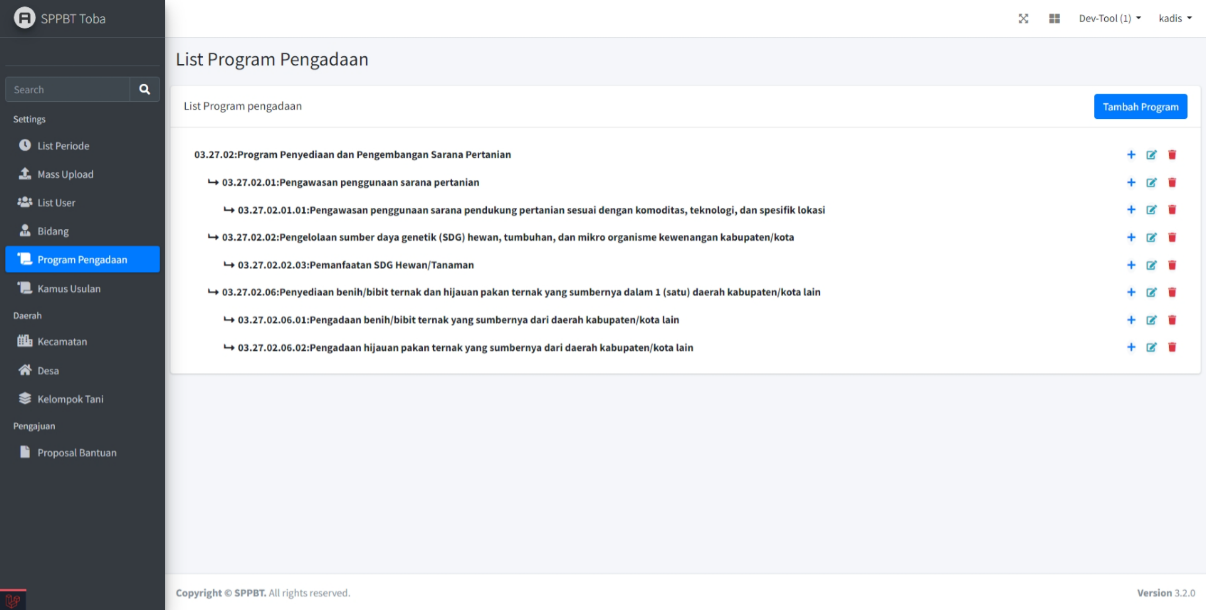
Pada tahap ini terdapat dua *user story* yang akan dijalankan. Kedua *user story* tersebut berkaitan dengan seluruh *user story* 5 (US-5) sampai *user story 6*(US-6). besaran *velocity* pada iterasi ini sebesar enam *point* yang berarti seluruh *user story* diharapkan selesai dalam kurun waktu 12 hari.

Tabel 4.9 User Stories Iterasi 3

|  |  |  |  |
| --- | --- | --- | --- |
| Iterasi 2 | | | |
| *User Stories* | *Priority* | *Stories point* | Estimasi waktu (Hari) |
| Sebagai kepala bidang saya ingin dapat menambah, melihat dan mengedit kamus usulan sehingga mempermudah dalam mengelola kamus usulan. | *Must have* | 3 | 6 |
| Sebagai kepala bidang saya ingin saya bisa melihat, mengajukan proposal kepada Kepala Dinas, dan memasukkan data pengadaan bantuan kelompok Tani. | *Must have* | 3 | 6 |
| *velocity* | | 6 | 12 |

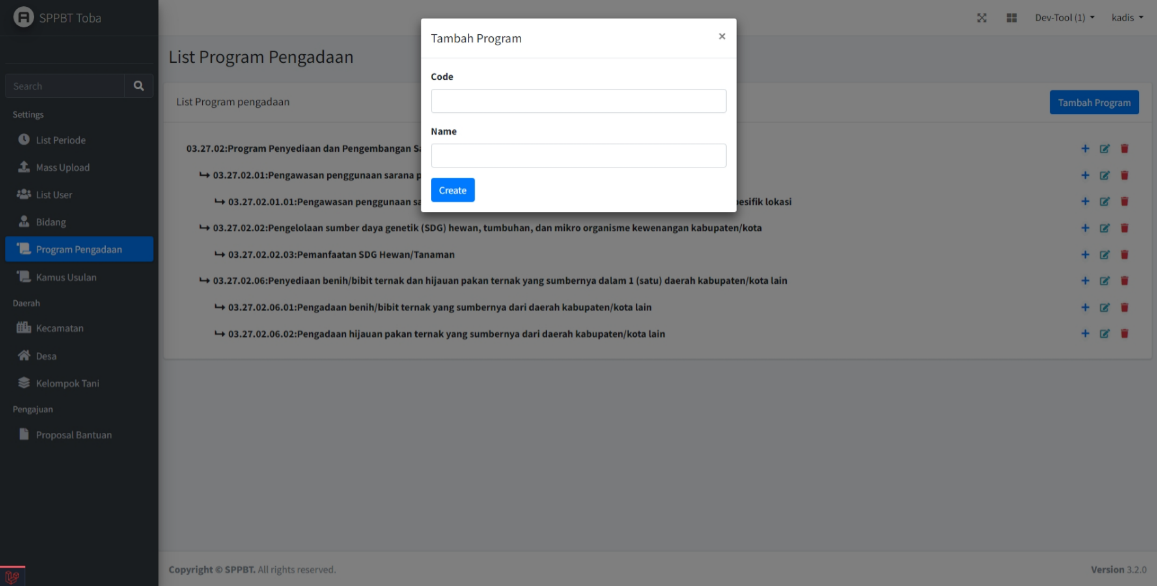
1. **Desain**

Tampilan sistem yang dibangun pada iterasi 1 ini adalah sebagai berikut.



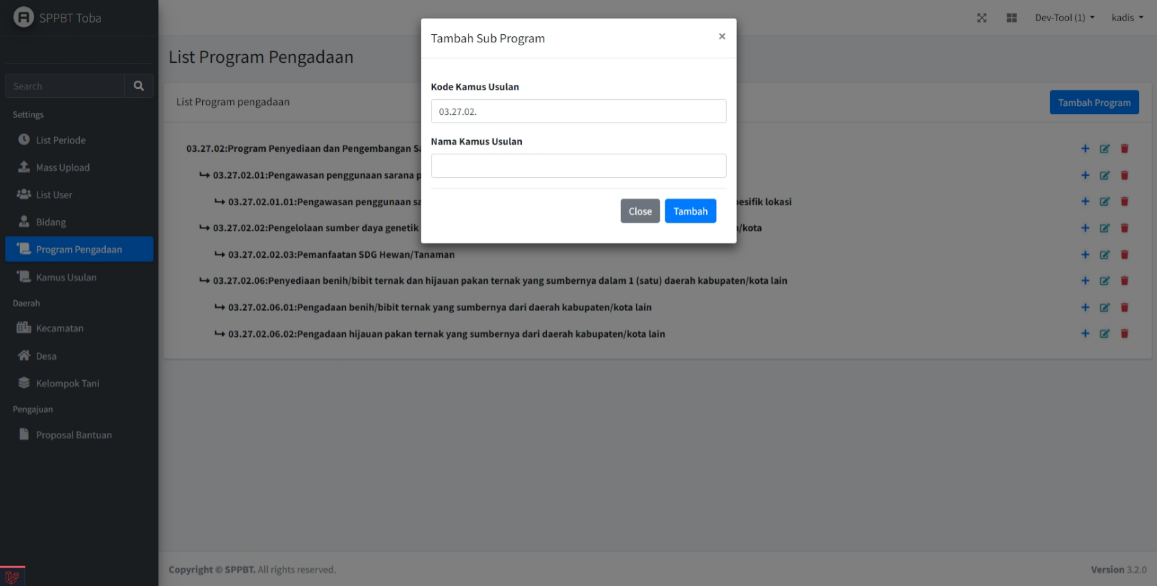
Gambar 4.31 Tampilan Program Pengadaan

[Gambar 4. 31](#_bookmark118) merupakan tampilan yang menunjukkan halaman utama program pengadaan. Pada halaman ini kepala bidang dapat melihat seluruh data program pengadaan yang tersimpan di dalam sistem. Kepala bidang dapat menambahkan program pengadaan baru dengan menekan tombol tambah program yang ada di bagian kanan atas. Kepala Bidang juga dapat menambah sub program, mengubah atribut program, dan menghapus program melalui halaman ini.



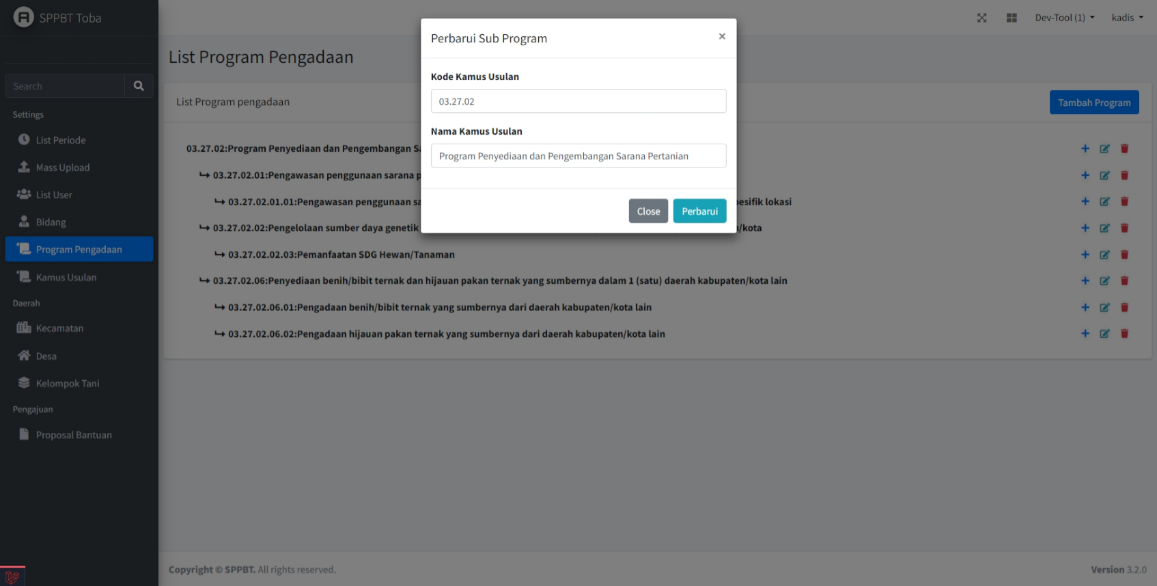
Gambar 4.32 Tampilan Form Tambah Program Pengadaan

Gambar 4.32 merupakan tampilan yang menunjukkan munculnya form tambah program pengadaan yang dapat diakses jika pengguna menekan tombol tambah program yang ada di halaman program pengadaan.



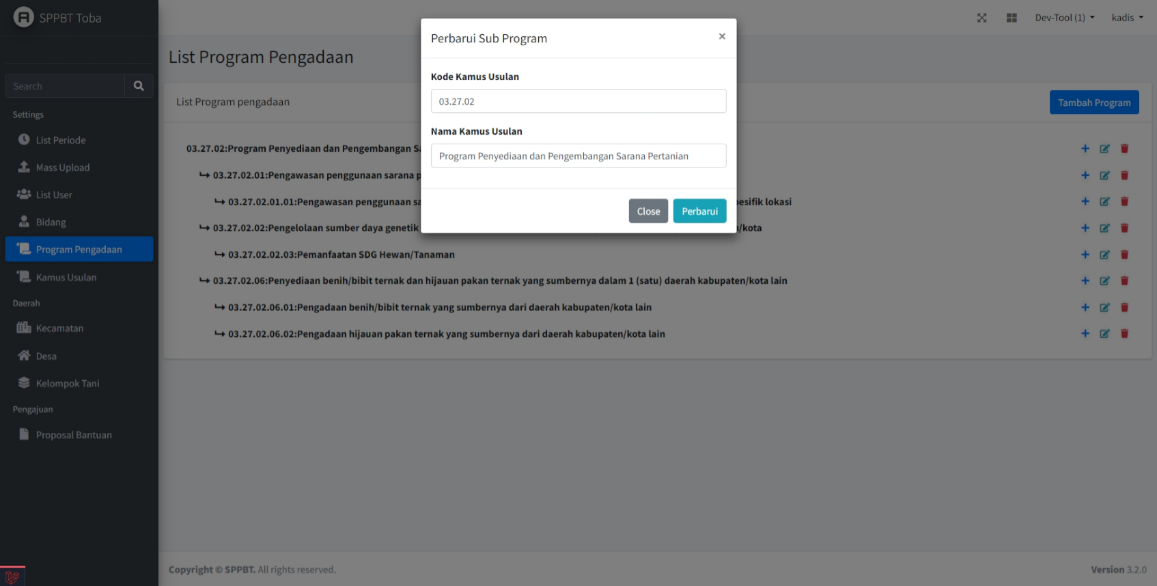
Gambar 4.33 Tampilan Form Tambah Sub Program Pengadaan

Gambar 4.32 merupakan tampilan yang menunjukkan munculnya form tambah sub program pengadaan pada program pengadaan yang sudah ada. Halaman ini dapat diakses jika pengguna menekan tombol dengan logo tambah yang ada di bagian sebelah kanan dari program pengadaan yang sudah ada sebelumnya.



Gambar 4.34 Tampilan Form Ubah Program Pengadaan

Gambar 4.32 merupakan tampilan yang menunjukkan munculnya form ubah sub program pengadaan pada program pengadaan yang sudah ada. Halaman ini dapat diakses jika pengguna menekan tombol dengan logo edit yang ada di bagian sebelah kanan dari program pengadaan yang sudah ada sebelumnya.

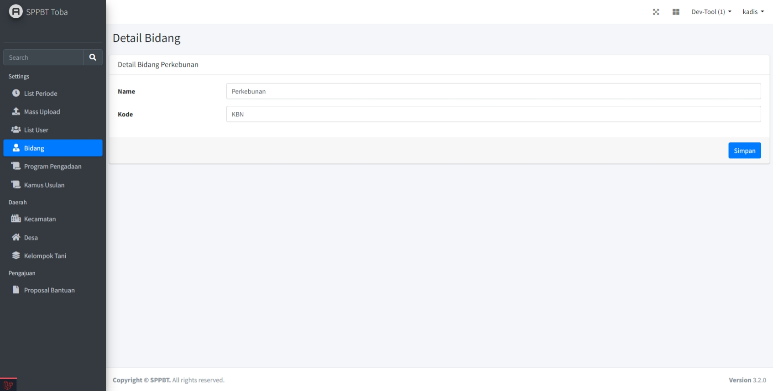


Gambar 4.35 Tampilan Form Ubah Program Pengadaan

Gambar 4.32 merupakan tampilan yang menunjukkan munculnya form ubah sub program pengadaan pada program pengadaan yang sudah ada. Halaman ini dapat diakses jika pengguna menekan tombol dengan logo edit yang ada di bagian sebelah kanan dari program pengadaan yang sudah ada sebelumnya.

[Gambar 4.](#_bookmark118) 12 merupakan tampilan yang menunjukkan *form* untuk menambah bidang baru. Bidang baru diharuskan memiliki atribut berupa nama dan kode bidang. Bidang baru dapat dibuat dengan mengisi semua *form* yang ada dan menekan tombol tambah di kanan bawah.

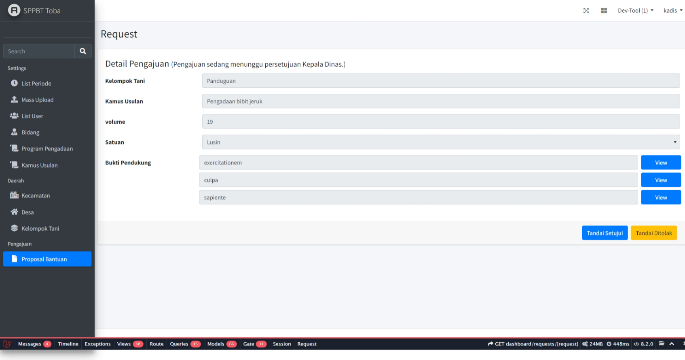
Gambar 4.13 Tampilan Detail dan Ubah Bidang



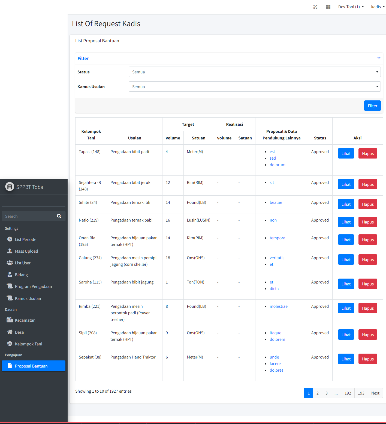
[Gambar 4.](#_bookmark118) 13 merupakan tampilan halaman detail bidang yang dapat diakses dengan cara menekan tombol lihat pada tabel bidang di halaman *list* bidang. Pada halaman ini pengguna terkait dapat memperbarui informasi bidang melalui halaman ini dengan mengubah data yang sudah ada di dalam *form* pada halaman ini dan menekan tombol simpan yang ada di bagian kanan bawah.

[Gambar 4.](#_bookmark118) 14 merupakan tampilan yang menunjukkan tabel yang berisi seluruh proposal bantuan yang masuk. Kepala dinas dapat melakukan *filtering* pada data proposal yang ada dengan menggunakan bantuan *filter* yang tersedia di halaman ini. Kepala dinas dapat menghapus data proposal pada halaman ini dengan menekan tombol hapus yang ada pada tabel *list* proposal.

Gambar 4.15 Tampilan Detail Proposal Bagian Kepala Dinas



Gambar 4.14 Tampilan List Proposal Bagian Kepala Dinas



Gambar 4.15 merupakan tampilan halaman detail proposal yang diajukan. Halaman ini dapat diakses dengan menekan tombol lihat yang ada pada tabel *list* proposal di halaman *list* proposal bagian kepala dinas. Kepala dinas hanya dapat mengubah atribut status proposal. Karena kepala dinas hanya dapat menyetujui atau menunda pelaksanaan bantuan dari apa yang ada di proposal. Kepala dinas dapat memutuskan untuk menyetujui dan menolak pelaksanaan bantuan dengan menekan tombol tandai setuju dan tombol tandai ditolak yang ada di bagian kanan bawah.

1. **Implementasi**

*User story* yang akan dijalankan pada tahap ini menyangkut kebutuhan fitur manajemen bidang dan manajemen proposal bagian kepala dinas. Fitur manajemen bidang bertujuan untuk menambah, mengedit dan menghapus bidang sedangkan manajemen proposal bagian kepala dinas bertujuan melihat, mengubah, dan merekapitulasi data proposal.

1. Unit Tes

Pengecekan fungsi yang sedang dikembangkan diperlukan sebagai dasar acuan sistem yang akan dibangun seperti yang dijelaskan pada [Tabel 4.](#_bookmark117) 3

Tabel 4.5 Skenario Pengujian Iterasi 2

| *Kode User Story* | *Kode Skenario Pengujian* | *Skenario Pengujian* | *Aktor* | *Hasil yang diharapkan* |
| --- | --- | --- | --- | --- |
| US-03 | US-03-TS-01 | Kepala dinas menambahkan bidang baru. | Kepala Dinas | Data bidang baru berhasil ditambahkan. |
| US-03 | US-03-TS-02 | Kepala dinas mengubah atribut bidang. | Kepala Dinas | Data bidang berhasil diubah. |
| US-03 | US-03-TS-03 | Kepala dinas menghapus bidang. | Kepala Dinas | Data bidang yang diinginkan berhasil dihapus. |
| US-04 | US-04-TS-01 | Kepala dinas dapat mengakses semua proposal yang ada. | Kepala Dinas | Semua proposal atribut proposal dapat dilihat. |
| US-05 | US-04-TS-01 | Kepala dinas dapat mengubah status dari proposal yang ada. | Kepala Dinas | Semua proposal atribut proposal dapat dilihat. |

Terdapat 5 unit tes yang dijalankan pada iterasi ini*.* Pengembang dapat melanjutkan untuk memulai tahapan selanjutnya dalam implementasi ini, yaitu *code generator*.

1. Code Generator

Code generator dalam iterasi 1 ini dapat dilihat gambar berikut :  
  
[MASUKKAN SS-an CODING]

1. Refactory

Refactor merupakan tahap pengoptimasian kode dalam tahap implementasi program. Refaktor iterasi 2 dalam pembangunan sistem manajemen pengadaan bantuan tani ini telah dilakukan pada saat pengimplementasian dalam kode program.

1. **Pengujian Sistem**

Pengembang menyesuaikan kembali hasil dari pembuatan *code* dengan *unit testing* yang dibuat dan mendapati bahwa semua pengujian dinyatakan berhasil. Setelah melakukan debug, ternyata didapati bahwa penerapan fungsi.

Tabel 4.6 Pengujian Sistem

| *Kode User Story* | *Kode Skenario Pengujian* | *Skenario Pengujian* | *Aktor* | *Hasil yang diharapkan* | *Status* |
| --- | --- | --- | --- | --- | --- |
| US-03 | US-03-TS-01 | Kepala dinas menambahkan bidang baru. | Kepala Dinas | Data bidang baru berhasil ditambahkan. | Berhasil |
| US-03 | US-03-TS-02 | Kepala dinas mengubah atribut bidang. | Kepala Dinas | Data bidang berhasil diubah. | Berhasil |
| US-03 | US-03-TS-03 | Kepala dinas menghapus bidang. | Kepala Dinas | Data bidang yang diinginkan berhasil dihapus. | Berhasil |
| US-04 | US-04-TS-01 | Kepala dinas dapat mengakses semua proposal yang ada. | Kepala Dinas | Semua proposal atribut proposal dapat dilihat. | Berhasil |
| US-05 | US-04-TS-01 | Kepala dinas dapat mengubah status dari proposal yang ada. | Kepala Dinas | Semua proposal atribut proposal dapat dilihat. | Berhasil |

Berdasarkan hasil tersebut maka tahap implementasi telah selesai dan siklus pengembangan dapat berlanjut ke tahapan retrospektif.

1. Retrospektif

Pada Penghujung iterasi dilakukan evaluasi terhadap pengembangan yang telah dijalankan seperti dijelaskan pada tabel

Tabel 4.7 Retrospektif Iterasi 2

|  |  |  |  |
| --- | --- | --- | --- |
| *Kode User Story* | *Estimasi (Hari)* | *Skala* | *Realisasi (Hari)* |
| US-03 | 6 | 3 | 1 |
| US-04 | 6 | 3 | 8 |

*User story* US-03 terjadi lebih cepat dari rencana dikarenakan pada iterasi ke satu, struktur *database* dan beberapa inisiasi fungsi untuk fitur manajemen bidang telah selesai dikerjakan. Dengan begitu pengerjaan fitur manajemen bidang hanya tersisa tampilan dan fungsi aksi lainnya. *User Story* US-04 mengalami kendala di beberapa bagian, termasuk mendesain struktural database untuk mencocokan kembali alur antara sistem yang dibangun pengembang dengan yang diinginkan *client*. Dengan demikian, pelaksanaan iterasi ke dua dinyatakan selesai dengan total penggunaan waktu 9 hari.

# BAB V KESIMPULAN DAN SARAN

## Kesimpulan

Berisi kesimpulan dari hasil dan pembahasan terkait penelitian yang dilakukan, dapat juga berupa temuan yang Anda dapatkan setelah melakukan penelitian atau analisis terhadap tugas akhir Anda. Berhubungan dengan poin pada rumusan masalah dan tujuan.

## Saran

Berisi saran mengenai aspek tugas akhir atau temuan yang dapat dikembangkan dan diperkaya di tugas akhir selanjutnya.

**DAFTAR PUSTAKA**

|  |  |
| --- | --- |
| [1] | M. Azizah, “Pengaruh Kemajuan Teknologi Terhadap Pola Komunikasi Mahasiswa Universitas Muhammadiyah Malang (UMM),” *Jurnal Sosiologi Nusantara,* vol. 6, no. 1, pp. 45-54, 2020. |
| [2] | Y. Rukayat, “Kualitas Pelayanan Publik Bidang Administrasi Kependudukan di Kecamatan Pasir Jambu,” *Jurnal Ilmiah Ilmu Administrasi,* vol. 11, no. 2, 2017. |
| [3] | E. Dermawan, “Implementasi E-Goverment Dalam Perancangan Sistem Informasi Pelayanan Pelanggan pada Kantor Urusan Agama Berbasis Web,” *Jurnal Cloud Information,* vol. 1, no. 1, pp. 15-22. |
| [4] | P. Toba, “Sejarah Daerah,” *DISKOMINFO TOBA,* 2021. |
| [5] | D. Toba, “Rencana Strategis (RENSTRA) 2021-2026,” 2021. |
| [6] | S. Nuryanti dan K. D. Swastika, “Peran Kelompok Tani Dalam Penerapan Teknologi Pertanian,” *Forum Penelitian Agro Ekonomi,* vol. 29, pp. 115-128, 2011. |
| [7] | A. Josi, “Penerapan Metode Prototiping dalam Pembangunan Website Desa (Studi Kasus Desa Sugihan Kecamatan Rambang),” *Jurnal Teknologi Informasi MURA,* vol. 9, no. 1, 2017. |
| [8] | K. S. Haryana, ““Penerapan Agile DEvelopment Methods dengan Framework Scrum pada Perancangan Perangkat Lunak Kehadiran Rapat Umum Berbasis QR-Code,” *Jurnal Computech & Bisnis,* vol. 13, no. 2, 2019. |
| [9] | R. Wiguna, “Penerapan Metode Agile dalam Pengembangan Sistem Infomasi Perpustakaan SMA N 1 Temon,” 2017. |
| [10] | Y. Dzhurov, I. Krasteva dan S. llieva, “Personal Extreme Programming - An Agile Process for Autonomous Developers,” *Researchgate,* 2009. |
| [11] | R. Supriatna, “Implementasi dan User Acceptance Test (UAT) Terhadap Aplikasi E-Learning pada Madrasah Aliyah Negero (MAN) 3 Kota Banda Aceh,” 2018. |
| [12] | I. S. Marfuah dan Y. Irawan, “Sistem Informasi Penyaluran Bantuan Pupuk Bersubsidi Pada Dinas Pertanian dan Pangan Kabupaten Rembang Berbasis Web,” *Jurnal Sistem Informasi dan Teknologi,* vol. 4, no. 1, 2021. |
| [13] | T. U. Mustikawati, “Sistem Informasi Pengelolaan Bantuan Desa Banjardowo,” *Skripsi,* 2022. |
| [14] | S. R. Tamin dan U. Khairat, “Sistem Informasi Pengadaan dan Pengalokasian Pupuk pada Dinas Pertanian, Peternakan, dan Perkebunan Kabupaten Majene Berbasis Web,” *Jurnal Peqguruang : Conference Series,* vol. 2, no. 1, 2020. |
| [15] | S. I. Pernama Sari, F. Nugraha dan A. P. Utomo, “Sistem Informasi Permohonan Pengajuan Bantuan Bibit Perkebunan pada Dinas Pertanian dan Pangan Kabupaten Kudus,” *Jurnal Sistem Informasi Dan Teknologi,* vol. 3, no. 2, 2020. |
| [16] | K. F. Rorimpandey, N. R. Pioh dan T. Tulung, “Peran Dinas Pertanian Dalam Pemberdayaan Kelompok Tani Di Desa Tonsewer Kecamatan Tompaso Barat Kabupaten Minahasa,” *E-journal Unsrat,* vol. 3, no. 3, 2023. |
| [17] | R. Z. H. Zahara dan H. , “FUNGSI KELOMPOK TANI PADA USAHA TANI PADI SAWAH DI GAMPONG UTEUN BUNTA KECAMATAN PEUSANGAN KABUPATEN BIREUEN,” *Jurnal AGRIFO,* vol. 6, no. 1, 2021. |
| [18] | T. Y dan A. M, “Analisis dan Lembaga Website Sebagai Sarana Informasi pada Lembaga Bahasa Kewirausahaan dan Komputer AKMI Baturaja,” *Jurnal Ilmiah Matrik,* vol. 19, no. 1, pp. 1-10, 2017. |
| [19] | D. M dan Q. J. Adrian, “Sistem Informasi Penjadwalan Dokter Berbasis Web dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah sakit Yukum Medical Centre),” *TEKNOINFO,* vol. 11, no. 2, pp. 30-37, 2017. |
| [20] | Y. Trimarsiah dan M. Arafat, “Analisis dan Perancangan Website Sebagai Sarana Informasi pada Lembaga Bahasa Kewirausahaan dan Komputer AKMI BATURAJA Menggunakan PHP dan MySQL,” *Jurnal Ilmiah Matrik,* vol. 19, no. 1, pp. 1-10, 2017. |
| [21] | U. P. N. ". V. J. Timur, “Teori Sistem Informasi Manajemen,” dalam *Sistem Infomasi Manajemen (SIM)*, 2013. |
| [22] | I. Mahendra dan D. T. Eby Yanto, “Sistem Informasi Pengajuan Kredit Berbasi Web Menggunakan Agile Development Methods pada Bank BRI Unit Kolonel Sugiono,” *Jurnal Teknologi dan Open Source,* vol. 1, no. 2, 2018. |
| [23] | W. A. Putra, I. Fitri dan D. Hidayatullah, “Implementasi Waterfall dan Agile dalam Perancangan E-commerce alat Musik Berbasis Website,” *Jurnal JTIK,* 2021. |
| [24] | S. A. Asri dan W. Setiawan, “Alternatif Penggunaan Model Pendekatan Agile pada Perancangan Sistem Informasi PKL Online,” *Jurnal Manajemen Teknologi dan Informatika,* vol. 5, no. 3, 2017. |
| [25] | Y. Dzhurov, I. Krasteva dan S. llieva, “Personal Ectreme Programming - An Agile Process for Autonomous Developers,” 2009. |
| [26] | U. M, M. G. I dan N. i, “Implementasi Metode Personal Extreme Programming dalam Pengembangan Sistem Manajemen Transaksi Perusahaan (Studi Kasus : CV. Todjoe sinar Group),” *Jurnal Repositor,* vol. 2, no. 3, pp. 261-268, 2020. |
| [27] | Y. Heriyanto, “Perancangan Sistem Informasi Rental Mobil Berbasis Web pada PT. APM RENT CAR,” *Jurnal Intra-Tech,* vol. 2, pp. 64-77, 2018. |
| [28] | S. W. Windiarti, “Sistem Manajemen Basis Data,” *Jurnal Manajemen Sistem Informasi,* Oktober 2020. |
| [29] | W. Komputer, Panduan MySQL Database Server, mediakita, 2010. |
| [30] | F. S. H. Lim dan T. Gantini, “Model Perancangan Aplikasi Konsultasi Pengobatan Herbal,” *Sarana Tugas Akhir Mahasiswa Teknologi Informasi,* vol. 3, no. 2, 2021. |
| [31] | M. Michael, Database Design, Application Development, and Administration, Chicago Business Press, 2011. |
| [32] | A. S. Rosa dan M. Salahuddin, “Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek),” *Bandung : modula,* 2011. |
| [33] | I. Solikin, “Perancangan Sistem Informasi Penjualan Berbasis Framework Model View Controller (MVC) pada PT. Thamrin Brother Cabang Oki,” *Jurnal Ilmiah Media Informatika,* vol. 4, no. 1, pp. 173-191, 2014. |
| [34] | I. G. Handika dan A. Purbasari, “Pemanfaatan Framework Laravel dalam Pembangunan Aplikasi E-Travel Berbasis Website,” *Konferensi Nasional Sistem Informasi,* pp. 1329-1334, 2018. |
| [35] | W. N. Cholifah, Y. Yulianingsih dan S. M. Sagita, “Pengujian Black Box Testing pada Aplikasi Action & Strategi Berbasis Android dengan Teknologi Phonegap,” *Satuan Tulisan Riset dan Inovasi Teknologi,* vol. 3, no. 2, pp. 206-210, 2018. |
| [36] | D. Rinaldi, “Pengembangan Aplikasi Permainan Edukasi untuk Mendukung Pembelajaran Siswa Sekolah Dasar,” *Jurnal Sistem dan Teknologi Informasi,* vol. 10, no. 4, pp. 1142-1148, 2022. |
| [37] | M. Azizah, “Pengaruh Kemajuan Teknologi Terhadap Pola Komunikasi Mahasiswa Universitas Muhammadiyah Malang (UMM),” *Jurnal Sosiologi Nusantara,* vol. 6, no. 1, pp. 45-54, 2020. |
| [38] | D. Rinaldi, “Pengembangan aplikasi Permainan Edukasi untuk Mendukung Pembelajaran Siswa Sekolah Dasar,” *Jurnal Sistem dan Teknologi Informasi,* vol. 10, no. 4, 2022. |
| [39] | M. A. Chamida, A. Susanto dan A. Latubessy, “Analisa User Acceptance Testing Terhadap Sistem Informasi Pengelolaan Bedah Rumah Di Dinas Perumahan Rakyat Dan Kawasan Permukiman Kabupaten Jepara,” *Indonesian Journal of Technology Informatics and Science (IJTIS),* vol. 3, no. 1, pp. 26-41, 2021. |
| [40] | R. Supriatna, “Implementasi dan User Acceptance Test (UAT) Terhadap Aplikasi E-Learning pada Madrasah Aliyah Negeri (MAN) 3 Kota Banda Aceh,” *Skripsi,* 2018. |
| [41] | I. Mahendra dan D. T. E. Yanto, “Sistem Informasi Pengajuan Kredit Berbasis Web Menggunakan Agile Development Methods pada Bank BRI Unit Kolonel Sugiono,” *Jurnal Teknologi dan Open Source,* vol. 1, no. 2, 2018. |

**LAMPIRAN**

1. Lampiran 1

