

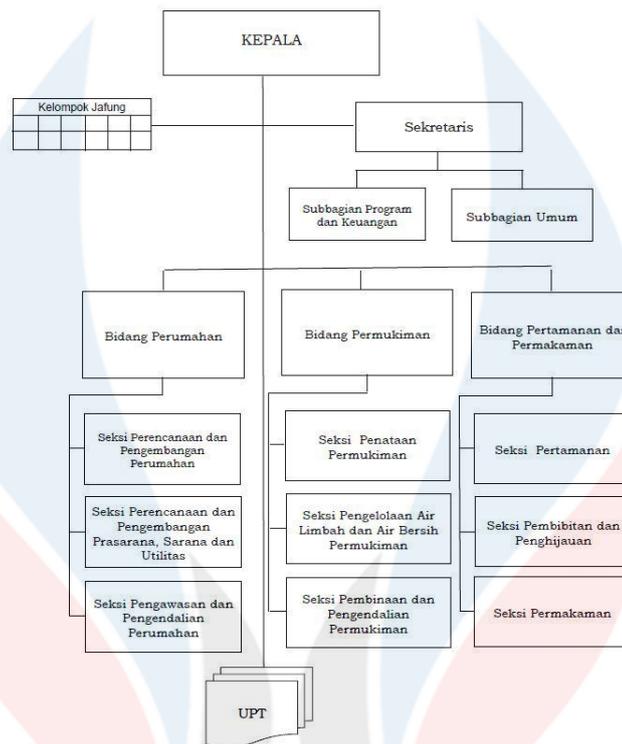
BAB 2

TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang berkaitan dengan penelitian ini. Tinjauan pustaka meliputi Dinas Perumahan dan Pemukiman, Sistem Informasi, *Personal Extreme Programming* (PXP), Laravel, React Native, MySQL, *Black Box Testing*, dan Penelitian Terdahulu.

2.1 Dinas Perumahan dan Pemukiman

Dinas Perumahan dan Pemukiman Kota Balikpapan adalah Organisasi Perangkat Daerah (OPD) yang mempunyai tugas menyelenggarakan urusan pemerintahan di bidang perumahan dan kawasan permukiman serta perencanaan, pengaturan, pengendalian, pengawasan dan pembinaan di bidang perumahan, permukiman, pertamanan dan permakaman (Balikpapan, 2016).



Gambar 2.1 Struktur Organisasi Dinas Perumahan dan Pemukiman

Gambar 2.1 menjelaskan mengenai struktur organisasi yang ada pada Dinas Perumahan dan Pemukiman Kota Balikpapan. Pada Dinas Perumahan dan Pemukiman Kota Balikpapan dipimpin oleh Kepala Dinas yang membawahi tiga bidang yaitu Bidang Perumahan, Bidang Pemukiman, dan Bidang Pertamanan dan Pemakaman. Pada Bidang Perumahan terdapat tiga seksi yaitu Seksi Perencanaan dan Pengembangan Perumahan, Seksi Perencanaan dan Pengembangan Prasarana, Sarana dan Utilitas, dan Seksi Pengawasan dan Pengendalian Perumahan. Pada Bidang Pemukiman, terdapat tiga seksi yaitu Seksi Penataan Permukiman, Seksi Pengelolaan Air Limbah dan Air Bersih Permukiman, dan Seksi Pembinaan dan Pengendalian Permukiman. Kemudian pada Bidang Pertamanan dan Pemakaman terdapat tiga seksi pula yaitu Seksi Pertamanan, Seksi Pembibitan dan Penghijauan, dan Seksi Pemakaman (Balikpapan, 2016).

Pada penelitian ini akan berfokus pada Bidang Pertamanan dan Pemakaman tepatnya pada Seksi Pembibitan dan Penghijauan selaku pelaksana pengelolaan data saldo tanaman. Yang bertanggung jawab pada proses pengelolaan data saldo tanaman adalah Kepala Seksi Pembibitan dan Penghijauan. Kemudian ada admin Dinas Perumahan dan Pemukiman yang bertugas untuk merekap dan mengkonfirmasi data saldo tanaman. Dan ada pengelola tempat pembibitan tanaman yang bertugas untuk mencatat pemasukan dan pengeluaran bibit tanaman di tempat pembibitan tanaman (Salmawati, 2020).

2.2 Arsitektur Aplikasi

Secara umum, pengertian arsitektur adalah ilmu dalam merancang dan membangun keseluruhan lingkungan dimulai dari level makro seperti perencanaan kota, perancangan perkotaan, dan arsitektur lanskap hingga level mikro seperti desain bangunan, desain perabot, dan desain produk. Sedangkan aplikasi adalah serangkaian perintah yang dieksekusi sehingga dapat diproses oleh komputer dari *input* menjadi *output*. Dari pengertian arsitektur dan aplikasi dapat disimpulkan bahwa arsitektur aplikasi merupakan desain sebuah aplikasi dimana masing-masing komponen yang terdapat didalam aplikasi tersebut dapat membentuk suatu sistem dan saling berkomunikasi. Pada arsitektur aplikasi harus ditentukan pula kebutuhan apa saja yang diperlukan dalam memproses data (Betri, et al., 2017).

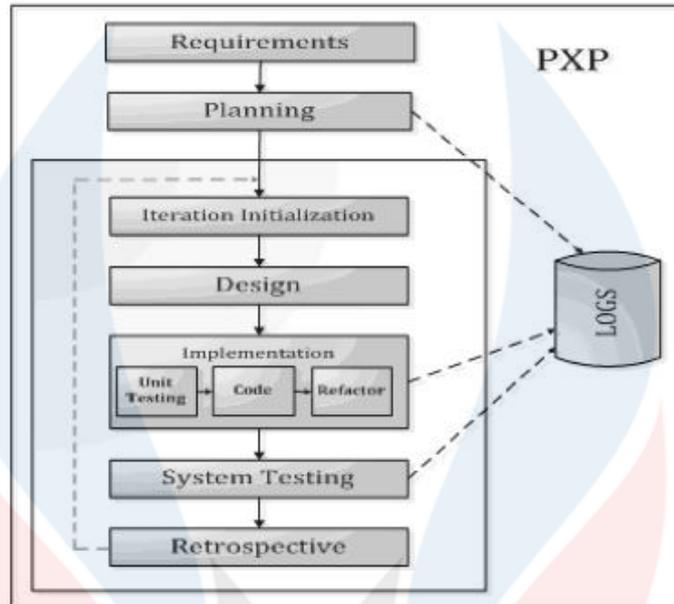
2.3 Sistem Informasi

Sistem informasi terdiri dari dua kata yaitu sistem dan informasi. Sistem adalah beberapa komponen yang disatukan

untuk mencapai satu tujuan, sementara informasi adalah data yang telah diolah sedemikian rupa sehingga berguna dalam pengambilan keputusan dan sebagai sumber informasi. Sistem informasi adalah sekumpulan komponen yang disatukan sehingga menjadi suatu komponen yang berfungsi untuk mengelola, mengolah, menyimpan, dan mendistribusikan informasi yang berguna untuk mendukung pengambilan keputusan atau mempermudah proses dalam suatu organisasi (Gunawan, et al., 2020). Sistem Informasi juga dapat didefinisikan sebagai suatu sistem yang memiliki tujuan untuk menyimpan, mengolah dan mengkomunikasikan informasi sehingga dapat membantu proses yang dilakukan pada suatu organisasi menjadi lebih mudah dan tidak membutuhkan sumber daya yang banyak (Puspitasari, 2015).

2.4 *Personal Extreme Programming*

Personal Extreme Programming (PXP) adalah salah satu metode yang digunakan dalam pengembangan perangkat lunak. Metode ini dirancang agar dapat diterapkan oleh pengembang secara individu. Metode ini dapat digunakan untuk menangani perubahan yang sering terjadi saat proses pengembangan perangkat lunak tanpa harus mengulang keseluruhan proses, melakukan pengembangan perangkat lunak dalam jangka waktu yang relatif singkat dan meminimalkan dokumentasi yang dihasilkan tanpa mengurangi prinsip dasar dari pengembangan perangkat lunak (Ulfi, et al., 2020).



Gambar 2.2 Fase Metode PXP

Seperti yang ditampilkan pada gambar 2.2, metode *Personal Extreme Programming* memiliki tujuh fase yaitu fase *requirements*, *planning*, *iteration initialization*, *design*, *implementation*, *system testing*, dan *retrospective*. Pada fase *requirements* akan dilakukan pengumpulan informasi yang akan digunakan pada sistem dengan melakukan wawancara kepada klien. Pada proses ini pula *user story* dibuat berdasarkan kebutuhan pengguna. *User story* adalah penjelasan singkat menggunakan bahasa sederhana yang ditulis berdasarkan sudut pandang pengguna terhadap suatu proses. Setelah *user story* selesai didefinisikan, kemudian masing-masing *user story* ditentukan *story point* nya, yang mana *story point* ini menentukan tingkat kesulitan, durasi yang dibutuhkan, dan prioritas pengerjaan. Kemudian pada fase *planning*, *user story* yang telah dibuat akan dikelompokkan menjadi beberapa kelompok yang akan dikerjakan pada setiap iterasi berdasarkan *story point* yang didapatkan pada fase *requirement*. Fase *iteration initialization* adalah tahap inialisasi dari setiap iterasi yang akan dilakukan. Pada tahap ini dibuat *use case diagram*

yang menggambarkan bagaimana keseluruhan fitur yang telah didefinisikan sebelumnya saling berhubungan. Selanjutnya adalah fase *design*, pada fase ini sistem akan mulai didesain sesuai dengan proses iterasi yang dikerjakan mulai dari desain basis data sampai dengan desain antarmuka pengguna. Pada tahap ini menghasilkan *entity relationship diagram*, *class diagram*, dan desain antarmuka aplikasi. Setelah itu ada fase *implementation* yang mana pada fase ini akan dimulai pengodean sistem dengan teknologi yang diperlukan. Kemudian berlanjut pada fase *system testing*, pada fase ini fungsionalitas dari sistem yang telah dibangun pada fase *mplementation* akan diuji. Setelah melakukan uji sistem, maka masuk kedalam fase *retrospective* yang mana dalam fase ini akan diambil kesimpulan berdasarkan hasil pengujian pada fase *system testing* apakah sistem yang telah dibuat telah berjalan dengan baik atau belum, jika belum maka iterasi akan dilakukan kembali, dan jika sudah sesuai maka akan dilanjutkan pada tugas berikutnya (Anjuliani, et al., 2015).

2.5 Use Case Diagram

Use case mengidentifikasi pelaku atau *actor* yang terlibat dalam interaksi serta jenis interaksinya. Kemudian ditambah dengan informasi tambahan yang menggambarkan interaksi dengan sistem. Informasi tambahan tersebut bisa berupa deskripsi tekstual atau berupa model grafis. *Use case* didokumentasikan dengan menggunakan diagram *use case*. Kumpulan *use case* mewakili semua kemungkinan interaksi yang akan dideskripsikan dalam kebutuhan sistem. *Actor* dalam proses, bisa manusia atau sistem lainnya diwakilkan dengan tokoh tongkat, *use case* diwakilkan dengan elips, dan garis menghubungkan *link* antara aktor dengan *use case* (Luthfi, 2017).

Tabel 2.1 menjelaskan mengenai notasi *use case* dan masing-masing fungsinya.

Tabel 2.1 Penggunaan Notasi Use Case Diagram

| Simbol | Deskripsi |
|---|--|
|  | <p>Fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case</p> |
|  | <p>Orang, proses, atau system lain yang berinteraksi dengan system informasi yang akan dibuat di luar system informasi yang akan akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p> |
|  | <p>Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.</p> |
|  | <p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p> |
|  | <p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.</p> |
|  | <p>Relasi use case tambahan ke sebuah</p> |

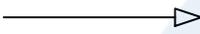
| Simbol | Deskripsi |
|--------|---|
| | use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya use case ini. |

2.6 Class Diagram

Class diagram digunakan ketika mengembangkan sistem model aplikasi berorientasi objek untuk menunjukkan kelas-kelas dalam sistem dan hubungan antar kelas tersebut. Sebuah objek kelas dapat dianggap sebagai definisi umum dari satu jenis objek sistem. Asosiasi adalah tautan antar kelas yang menunjukkan bahwa ada hubungan antara kelas-kelas ini. Akibatnya, setiap kelas mungkin harus memiliki pengetahuan tentang kelas yang terasosiasi dengannya (Luthfi, 2017). Tabel 2.2 menjelaskan notasi pada *class diagram* dan masing-masing fungsinya.

Tabel 2.2 Penggunaan Notasi Class Diagram

| Simbol | Deskripsi |
|---|--|
|  | Kelas pada struktur sistem |
|  | Sama dengan konsep interface dalam pemrograman berorientasi objek |
|  | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity |
|  | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity |

| Simbol | Deskripsi |
|---|--|
|  | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus) |
|  | Relasi antarkelas dengan makna kebergantungan antarkelas |
|  | Relasi antarkelas dengan makna semua-bagian (whole-part) |

2.7 Entity Relationship Diagram

Entity Relationship Diagram (ERD) merupakan sekumpulan cara atau alat-alat untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (*entity*) serta interaksi antar entitas-entitas tadi menggunakan beberapa notasi (Gunawan, et al., 2020). Tabel 2.3 menjelaskan notasi ERD dan masing-masing fungsinya.

Tabel 2.3 Penggunaan Notasi Entity Relationship Diagram

| Simbol | Deskripsi |
|---|--|
|  | Notasi ini bernama entitas dan menggambarkan Individu yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain. |
|  | Notasi ini bernama atribut, merupakan properti yang dimiliki oleh suatu entitas, dimana dapat mendeskripsikan karakteristik dari entitas tersebut. |
|  | Notasi ini bernama relasi serta menunjukkan hubungan diantara sejumlah entitas yang berbeda. |

2.8 Laravel

Laravel adalah *framework open source* PHP berbasis web yang dikembangkan oleh Taylor Otwell dan Laravel menerapkan pola arsitektur *model-view-controller* (MVC). Tujuan dari dibuatnya Laravel adalah untuk meningkatkan kualitas pengembangan *website* tanpa perlu mengetahui terlalu dalam mengenai bahasa pemrograman PHP. Laravel mempunyai beberapa fitur unggulan yang dapat membuat pengembangan web menjadi lebih mudah. Fitur unggulan dari Laravel antara lain adalah mengenalkan cara yang berbeda dalam mengakses database relasional. Untuk menghubungkan Laravel dengan berbagai jenis basis data dalam satu aplikasi tidak perlu membuat perintah untuk setiap basis data, hanya diperlukan *syntax* yang disediakan dari Laravel yang bernama Eloquent maka *syntax* tersebut akan dikonversi sesuai dengan jenis basis data yang digunakan dalam aplikasi. Laravel juga memberikan kemudahan dalam membuat *Application Programming Interface* (API) sehingga Laravel dapat berperan menjadi *Back-end* aplikasi yang dapat dihubungkan dengan berbagai macam aplikasi *front-end* (Gunawan, et al., 2020). Pada penelitian ini akan menggunakan fitur API dari Laravel untuk menghubungkan antara basis data, *back end* dan aplikasi *mobile*.

2.9 React Native

React Native adalah *framework open source* yang dikembangkan oleh Facebook dengan tujuan untuk membuat pengembangan aplikasi *mobile* menjadi lebih mudah dan tidak perlu menguasai banyak bahasa pemrograman sesuai dengan deskripsi pada situs resminya yaitu "*Learn once, write anywhere*". React Native dibangun dengan bahasa Javascript dan dengan dukungan dari *framework* React.js, dengan menggunakan React

Native pengembang dapat membuat aplikasi android dan ios hanya dengan satu bahasa pemrograman saja yaitu Javascript. Penulisan kode React Native sangat dipengaruhi oleh pengembangan *website*. Pembuatan komponen tampilan pada React Native menggunakan JSX. JSX adalah *syntax* Javascript yang dibuat mirip dengan *syntax* HTML dengan beberapa perbedaan kecil. Dengan adanya JSX ini, membuat pengembang yang berasal dari pengembangan *website* dapat lebih mudah beradaptasi dalam membuat aplikasi mobile dengan *framework* React Native (Gunawan, et al., 2020).

2.10 Black Box Testing

Black Box Testing adalah salah satu cara dalam melakukan pengujian terhadap fungsi operasional dari sebuah perangkat lunak. Penguji yang ikut serta dalam pengujian ini adalah orang yang tidak ikut serta dalam pengembangan perangkat lunak, sehingga penguji tidak mengetahui detail struktur internal dari perangkat lunak yang sedang diuji. *Black Box Testing* berfokus kepada kebutuhan fungsional pada perangkat lunak dan dibandingkan spesifikasi kebutuhan perangkat lunak yang telah didefinisikan pada tahap awal (Utami, 2015).

2.11 Penelitian Terdahulu

Berdasarkan hasil penelitian terdahulu yang terdapat pada tabel 2.4 dapat disimpulkan bahwa dengan adanya penggunaan teknologi informasi dalam suatu proses terbukti dapat mengurangi waktu dan sumber daya yang dibutuhkan. Hal ini disebabkan karena ada proses yang dapat dilakukan secara otomatis oleh sistem sehingga proses yang dilakukan menjadi lebih efektif dan efisien (Nursaid, et al., 2020). Dengan adanya hasil tersebut dapat menjadi solusi dalam proses pendataan

saldo tanaman yang ada pada Dinas Perumahan dan Pemukiman Kota Balikpapan dimana proses tersebut masih dilakukan secara manual dan melalui proses yang panjang sehingga membutuhkan waktu yang lama, sumber daya yang lebih banyak dan rentan terhadap kesalahan dalam memasukkan data.

Dalam mengembangkan sistem yang sesuai untuk suatu proses diperlukan pemahaman yang baik terhadap proses tersebut, apa saja fitur yang diperlukan dalam sistem yang dapat menyelesaikan permasalahan yang ada. Target pengguna menjadi salah satu indikator penentu pada *platform* apa sistem tersebut akan dibangun (Wiguna, et al., 2018). Selain itu, perlu juga dilakukan perancangan tampilan sistem dan di validasi terhadap pengguna yang akan menggunakan sistem tersebut agar sistem yang nantinya dibuat dapat digunakan oleh pengguna dengan mudah dan nyaman (Hidayatullah, et al., 2020).

Metode pengembangan aplikasi juga perlu ditentukan agar dalam proses pembangunan sistem dapat berjalan dengan lancar. Dari penelitian terdahulu dapat disimpulkan bahwa metode yang paling tepat untuk pembangunan sistem pembibitan pohon pada Dinas Perumahan dan Pemukiman ini adalah metode *personal extreme programming* (PXP). Metode PXP memiliki kelebihan dapat beradaptasi terhadap perubahan dengan cepat dan tanpa harus mengulang proses pengembangan dari awal kembali. Metode PXP juga sangat kompatibel digunakan oleh pengembang yang hanya terdiri dari satu orang saja sehingga sangat cocok untuk digunakan pada pengembangan aplikasi SIMBION ini yang hanya dikembangkan oleh satu pengembang saja (Ulfi, et al., 2020).

Proses pengujian terhadap sistem yang telah selesai dikembangkan juga sangat penting untuk dilakukan. Tujuan dari proses pengujian ini adalah agar dapat diketahui apakah sistem

yang telah dibuat sudah memenuhi keinginan dari pengguna sistem dan juga dapat mengatasi masalah yang didefinisikan sebelumnya. Selain itu, perlu juga adanya proses *training* terhadap pengguna dan pembuatan dokumentasi agar ketika sistem digunakan pengguna dapat menggunakan sistem tersebut dengan baik dan jika terdapat kesulitan dapat merujuk pada dokumentasi yang dibuat untuk mengatasi kesulitan tersebut (Nawang, et al., 2017).



Tabel 2.4 Penelitian Terdahulu

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|-----------|--|-------------|---|---|---|--|
| 1 | Aplikasi ISC (Informatics Student Center) Menggunakan Metode Personal Extreme Programming Berbasis Android | 2015 | Pengaksesan informasi yang kurang praktis pada sistem <i>E-Learning</i> dikarenakan berbasis <i>website</i> yang mana tampilan <i>website</i> lebih nyaman digunakan menggunakan PC ataupun laptop sementara banyak mahasiswa yang mengakses sistem menggunakan <i>smartphone</i> . Sehingga diperlukan sistem berbasis <i>mobile</i> untuk mengatasi | Penelitian ini menggunakan metode <i>Personal Extreme Programming</i> (PXP) | Pembuatan aplikasi menggunakan metode <i>Personal Extreme Programming</i> terbukti memudahkan pengembang dalam mengembangkan aplikasi pada sisi melakukan perubahan sistem saat proses pengembangan aplikasi. | Dalam mengembangkan aplikasi dengan satu pengembang dan pengerjaan sistem yang cepat metode <i>Personal Extreme Programming</i> sangat cocok untuk dijadikan metode dalam melakukan pengembangan aplikasi. |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|--|------|---|--|---|--|
| | | | permasalahan tersebut. | | | |
| 2 | Rancang Bangun Sistem Informasi Manajemen Aset Berbasis Web Untuk Optimalisasi Penelusuran Aset Di Teknik Industri UNDIP | 2016 | Pengelolaan terhadap asset baru seperti gedung, komputer, meja dan lain-lain pada Program Studi Teknik Industri Universitas Diponegoro masih belum baik, sehingga diperlukan sistem yang dapat mengelola data aset menjadi lebih efektif. | Penelitian ini menggunakan metode <i>waterfall</i> | Pengembangan sistem informasi manajemen asset berbasis web terbukti memudahkan pengguna dalam mengelola aset pada Teknik Industri UNDIP | Pembuatan aplikasi berbasis <i>website</i> sangat baik digunakan saat pengguna dari sistem tersebut menggunakan perangkat PC atau laptop dalam mengakses sistem. |
| 3 | Rancang Bangun Sistem Informasi Pengolahan Data | 2017 | Pengelolaan data persediaan barang pada perusahaan yang | Penelitian ini menggunakan metode | Sistem informasi yang telah dibuat terbukti dapat | Pengujian pada aplikasi yang telah dibuat sangat penting |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|--|------|---|--|--|--|
| | Persediaan Berbasis Desktop Dengan Metode Waterfall | | masih dilakukan secara manual menyebabkan sulitnya mengecek persediaan barang, mencari data, dan memasukkan data. Sehingga diperlukan sistem yang dapat melakukan pengelolaan data persediaan barang secara otomatis. | <i>waterfall</i> | menyelesaikan permasalahan pengolahan data persediaan dengan dibuktikan pengujian fitur aplikasi sudah sesuai dengan perancangan | sebagai salah satu indikator apakah aplikasi telah sesuai kebutuhan pengguna atau belum. |
| 4 | Rancang Bangun Sistem Informasi Manajemen Administrasi Surat Menggunakan | 2017 | Pengolahan administrasi surat pada Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri | Penelitian ini menggunakan metode <i>waterfall</i> | Berdasarkan hasil pengujian pada aspek <i>efficiency</i> yang mendapat nilai 92.75% | Proses mendefinisikan kebutuhan pengguna dalam aplikasi harus dilakukan dengan seksama agar aplikasi |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|-----------|---|-------------|--|--|---|---|
| | Framework Laravel Pada Jurusan Teknik Elektro | | Semarang masih dilakukan secara konvensional yang membuat proses pembuatan surat menjadi lebih panjang dan lama. Oleh sebab itu diperlukan sistem yang dapat membuat proses pembuatan surat menjadi lebih singkat. | | membuktikan bahwa pengembangan aplikasi dapat membuat proses menjadi lebih efisien. | yang dihasilkan dapat benar-benar membantu pengguna. |
| 5 | Sistem Informasi Pengolahan Data Tanaman Hias Dan Pohon Pelindung Di Taman Kota | 2017 | Saat ini proses penentuan jenis tanaman hias dan pohon pelindung pada Dinas Kebersihan dan | Penelitian ini menggunakan metode <i>System Development Life Cycle</i> | Dengan dibuatnya sistem informasi pengolahan data tanaman hias dan pohon pelindung | Pembuatan sistem informasi pengelolaan data akan sangat membantu dalam proses pengelolaan |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|-----------|--|-------------|--|--|--|--|
| | Pariaman | | Pertamanan Kota Pariaman masih dilakukan secara manual sehingga menyebabkan tanaman mati karena tidak sesuai penentuan jenisnya. | (SDLC) | terbukti dapat meminimalisir kesalahan dalam memasukkan data, memudahkan dalam pembuatan laporan karena dilakukan secara otomatis oleh sistem. | data saldo tanaman pad Dinas Perumahan dan Pemukiman Kota Balikpapan sehingga sumber daya dan waktu yang diperlukan dapat berkurang. |
| 6 | Perancangan Sistem Informasi Manajemen Data Korban Bencana Berbasis Mobile Android | 2017 | Saat ini sistem administrasi dan informasi korban bencana masih dikelola secara internal dan hanya bisa diakses pada pusat informasi | Penelitian ini menggunakan metode <i>waterfall</i> | Sistem informasi manajemen data korban bencana yang dibuat terbukti dapat mengelola data secara otomatis | Pengembangan aplikasi berbasis android dibutuhkan ketika pengguna aplikasi membutuhkan aplikasi yang praktis dibawa kemana saja. |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|---|------|---|---|--|---|
| | | | yang menyebabkan data menjadi sulit untuk diakses oleh masyarakat. | | serta dapat diakses oleh masyarakat dari mana saja karena berbasis aplikasi android. | |
| 7 | Pengembangan Sistem Informasi Keanekaragaman Hayati Data Pohon Dan Endemik Pada Hutan Hujan Tropis Kalimantan | 2017 | Dengan banyaknya jenis flora dan fauna yang ada di hutan hujan tropis Kalimantan diperlukan pendataan untuk setiap jenis flora dan fauna, tetapi pengelolaan data keanekaragaman hayati di Kalimantan | Penelitian ini menggunakan metode pengembangan perangkat lunak pendekatan model proses <i>incremental</i> | Pengembangan sistem informasi keanekaragaman hayati data pohon dan endemik pada hutan hujan tropis Kalimantan terbukti dapat memudahkan dalam proses penentuan jenis | Proses pencarian dan pengelolaan data yang baik menjadi komponen utama dalam membuat aplikasi yang melibatkan data yang banyak. |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|--|------|--|--|--|--|
| | | | masih belum dikelola secara baik dan belum tersimpan pada satu basis data yang sama, sehingga diperlukan sistem yang dapat mengelola data keanekaragaman hayati sehingga data tersebut dapat disimpan dalam satu basis data yang sama. | | dan pencarian data pohon. | |
| 8 | Penggunaan Framework Laravel Dalam Rancang Bangun Modul Back-End Artikel | 2017 | Para pelaku bisnis kesulitan dalam memasarkan produk serta mencari sumber edukasi terkait bisnis. | Penelitian ini menggunakan metode <i>Prototyping</i> | Penggunaan Framework Laravel sebagai <i>Back-end</i> dari sebuah aplikasi berbasis | <i>Framework</i> Laravel cocok untuk digunakan sebagai <i>Back-end</i> yang kemudian dapat |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|---|------|--|--|--|---|
| | Website Bisnisbisnis.ID | | Masyarakat juga mengalami kesulitan dalam mendapatkan informasi bisnis yang diperlukan. Sehingga diperlukan sistem yang dapat mengatasi masalah tersebut. | | <i>website</i> sangat membantu pengembang dengan adanya fitur pembuatan API pada Laravel. | dihubungkan kepada aplikasi lain berbasis <i>mobile</i> maupun <i>website</i> . |
| 9 | Perbandingan Phonegap Dan React Native Sebagai Framework Pengembangan Aplikasi Mobile | 2018 | Terdapat kesulitan saat memilih <i>framework</i> untuk pengembangan aplikasi <i>mobile</i> , sehingga dibutuhkan perbandingan antara dua <i>framework</i> untuk pengembangan | Peneliti melakukan pengujian dengan berbagai perangkat menggunakan dua <i>framework</i> yang berbeda | <i>Framework</i> React Native menghasilkan tampilan yang lebih alami dan mirip seperti aplikasi yang dibangun dengan bahasa <i>native</i> Java | React native cocok untuk pengembangan aplikasi berbasis <i>mobile</i> dikarenakan dapat membuat aplikasi Android dan IOS hanya dengan satu bahasa yaitu Javascript. |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|--|------|--|---|---|---|
| | | | aplikasi <i>mobile hybrid</i> yaitu Phonegap dan React Native. | yaitu Phonegap dan React Native | untuk Android dan bahasa Swift pada IOS. React Native juga menggunakan sumber daya memori dan CPU yang lebih rendah dari Phonegap | |
| 10 | Rancang Bangun Aplikasi Point Of Sales Distro Management System Menggunakan Framework React Native | 2018 | Para pelaku bisnis distro masih menjalankan bisnisnya tanpa menggunakan perkembangan teknologi dan masih secara manual, sehingga mengalami kesulitan dalam | Penelitian ini menggunakan metode <i>Classic Life Cycle</i> | Pengembangan aplikasi Distro Management System terbukti dapat memudahkan pengguna dalam melakukan transaksi dan | Penentuan <i>platform</i> aplikasi yang akan dibuat ditentukan dari kebutuhan pengguna yang menjadi target. |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|---|------|---|---|---|---|
| | | | mengelola transaksi dan manajemen produk. Oleh sebab itu diperlukan sistem yang dapat membantu proses pada distro menjadi otomatis. | | memanajemen produk. | |
| 11 | Rancang Bangun Sistem Informasi Manajemen Data Mahasiswa Dan Dosen Terintegrasi | 2018 | Terdapat pengelolaan data mahasiswa dan dosen yang masih belum akurat dan belum terintegrasi sehingga dibutuhkan sistem yang dapat mengintegrasikan data mahasiswa dan dosen. | Penelitian ini menggunakan metode <i>System Development Life Cycle</i> (SDLC) | Dengan pembangunan sistem informasi manajemen data mahasiswa dan dosen terbukti dapat menyajikan data yang lebih akurat dan terstruktur serta | Penyimpanan data pada basis data yang sama sangat diperlukan dalam mengintegrasikan beberapa aplikasi agar data yang disimpan tidak terpisah. |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|---|------|---|--|--|---|
| | | | | | data disimpan dalam satu penyimpanan yang sama sehingga memudahkan dalam proses integrasi | |
| 12 | Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus: Toko Uda Fajri) | 2020 | Proses pengelolaan persediaan barang yang masih manual menyebabkan sulitnya mencari informasi persediaan barang, kesalahan dalam memasukkan data, dan kesulitan dalam mengecek data | Penelitian ini menggunakan metode <i>Prototyping</i> | Pengembangan sistem informasi pengelolaan barang yang dibuat terbukti memudahkan pengguna dalam proses pengelolaan persediaan barang | Pembuatan sistem yang dapat membuat proses manual menjadi otomatis terbukti efektif dalam membantu proses menjadi lebih cepat dan tanpa membutuhkan sumber daya yang besar. |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|--|------|--|--|---|--|
| | | | persediaan barang. Sehingga diperlukan sistem yang dapat melakukan proses pengelolaan data secara otomatis. | | dibuktikan dengan hasil reaksi pengguna dalam mengoperasikan sistem sebesar 96% atau sangat mudah | |
| 13 | Aplikasi Monitoring Perusahaan Penanaman Modal (Studi Kasus DPMPTSP Kota Denpasar) | 2020 | Proses pemantauan atau monitoring terhadap perusahaan penanaman modal di Kota Denpasar masih dilakukan dengan cara mendatangi langsung perusahaan tersebut dan proses pencatatan data masih secara | Penelitian ini menggunakan metode <i>Prototyping</i> | Penggunaan metode <i>Prototyping</i> dalam merancang sistem informasi terbukti memudahkan pengembang dalam merencanakan fitur-fitur dan | Perencanaan dari sisi struktur basis data, fitur-fitur aplikasi, dan desain tampilan aplikasi perlu dilakukan secara baik dan benar agar proses implementasi dapat berjalan lebih mudah. |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|--|------|--|---|---|--|
| | | | manual yang menyebabkan waktu dan sumber daya yang diperlukan dalam proses monitoring menjadi lebih besar, sehingga diperlukan sistem yang dapat mengurangi sumber daya dan waktu yang diperlukan dalam proses monitoring. | | desain tampilan yang akan diimplementasikan pada sistem. | |
| 14 | Implementasi Metode Personal Extreme Programming Dalam | 2020 | Pengelolaan proses bisnis pada CV. Todjoe Sinar Group yang masih dilakukan secara manual | Penelitian ini menggunakan metode <i>Personal Extreme</i> | Metode <i>Personal Extreme Programming</i> (XP) terbukti memudahkan | Metode <i>Personal Extreme Programming</i> (XP) cocok untuk digunakan oleh pengembang yang |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|--|------|--|-----------------------------------|--|---|
| | Pengembangan Sistem Manajemen Transaksi Perusahaan | | menyebabkan beberapa permasalahan. Oleh sebab itu diperlukan sistem yang dapat membuat proses bisnis tersebut dapat dilakukan secara otomatis. | Programming (PXP) | pengembang dalam melakukan perubahan sistem saat proses pengembangan tanpa harus melakukan proses pengembangan dari awal kembali dan pengembang dapat melakukan proses pengembangan secara individu. | bekerja secara individu dan tidak membutuhkan dokumentasi yang banyak serta membutuhkan adaptasi dalam perubahan sistem saat proses pengembangan. |
| 15 | Aplikasi Manajemen Surat Masuk dan Surat | 2020 | Penanganan dan pengelolaan surat yang belum baik dan | Penelitian ini menggunakan metode | Aplikasi manajemen surat masuk dan keluar | Aplikasi yang dikembangkan perlu memperhatikan aspek |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|----|--|------|---|-----------------------------------|---|--|
| | Keluar Berbasis Mobile Di Perum BULOG Subdrive | | masih dilakukan secara manual pada Perum BULOG Subdrive menyebabkan beberapa masalah, oleh karena itu diperlukan sistem yang dapat mengelola surat masuk dan keluar untuk menyelesaikan permasalahan tersebut | waterfall | terbukti dapat menyelesaikan permasalahan yang terjadi dikarenakan proses yang masih manual. Aplikasi dapat diakses dari mana saja yang membuat pengguna lebih mudah dalam mengoperasikan aplikasi. | kemudahan pengguna dalam mengoperasikan aplikasi. |
| 16 | Pengembangan Sistem Informasi Pembibitan Pohon | | Proses pencatatan dan monitoring bibit tanaman yang harus | Penelitian ini menggunakan metode | Dengan adanya sistem ini diharapkan dapat | Perlu memperhatikan pada <i>platform</i> apa sistem akan dibuat, |

| No | Title | Year | Problem Statement | Method | Key Findings | Remarks |
|-----------|--|-------------|--|---|--|---|
| | (SIMBION) Pada Dinas Perumahan dan Pemukiman Kota Balikpapan | | melalui proses yang panjang serta dikerjakan oleh banyak orang menyebabkan sering terjadinya kesalahan dalam memasukkan data bibit tanaman. Selain itu, terdapat kesulitan dalam merekap data dikarenakan proses perekapan data masih dilakukan secara manual. | <i>Personal Extreme Programming (PXP)</i> | membantu pihak Dinas Perumahan dan Pemukiman Kota Balikpapan dalam mengelola data bibit tanaman sehingga menjadi lebih tertata rapi dan kesalahan dalam memasukkan data dapat diminimalisir. | sehingga pengguna dapat menggunakan sistem dengan nyaman. |