

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

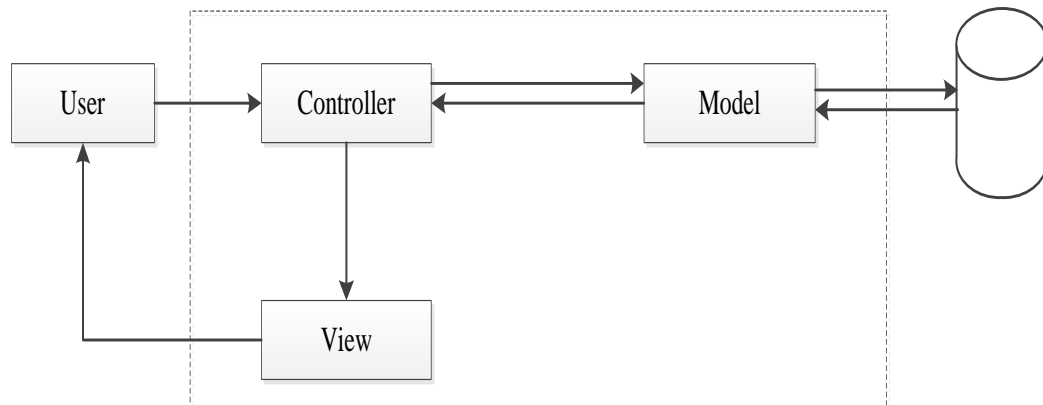
Pengertian menurut [7] Sistem informasi adalah cara-cara yang diorganisasi untuk mengumpulkan, memasukkan, dan mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

Definisi menurut [8] Sistem informasi adalah sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi mengenai saldo persediaan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegritasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

2.2 CodeIgniter

CodeIgniter menurut [9] merupakan *framework* untuk bahasa pemrograman PHP. *CodeIgniter* memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat mengembangkan dalam perangkat *web*, dekstop maupun *mobile*. *CodeIgniter* memilki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan.



Gambar 2.1 Arsitektur MVC

2.2.1 Web Based

Web Based menurut [10] adalah aplikasi yang dibuat berbasis web yang membutuhkan *web server* dan *browser* untuk menjalankannya. Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).

2.2.2 PHP

Menurut [11] PHP adalah *Perl Hypertext Preprocessor* bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis.

Menurut [12] berpendapat bahwa PHP adalah bahasa pemrograman *web server-side* yang bersifat *open source* atau mudah dikembangkan. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*.

2.2.3 MySql

Menurut [13] *MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan basis data.

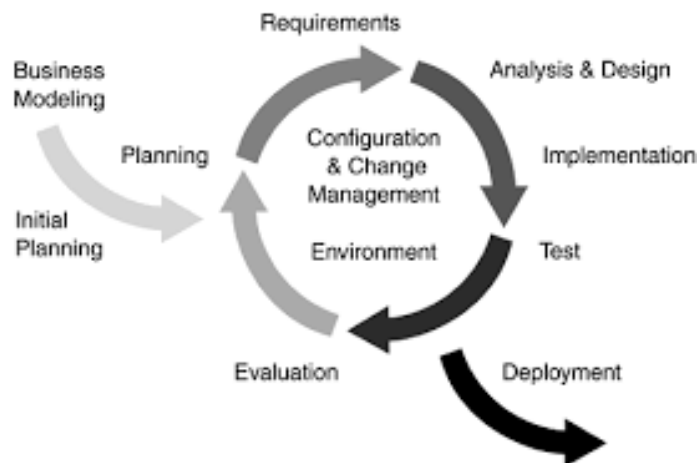
Menurut [14] mendefinisikan *mysql* MYSQL adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

MySQL merupakan bahasa standar yang paling banyak digunakan untuk mengakses *data base* relasional dan merupakan aplikasi yang dapat di gunakan secara bebas.

2.3 Metode RUP (*Rational Unified Process*)

Menurut [15] *Unified Process* merupakan sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan inkremental (bertahap dengan progres menaik). Iteratif bisa dilakukan di dalam setiap tahap, atau iteratif tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang inkremental (bertambah naik) dimana setiap iterasi akan memperbaiki iterasi berikutnya.

Menurut [15] RUP juga merupakan pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang, fokus pada arsitektur, lebih diarahkan berdasarkan pengguna kasus. Proses pengulangan/iteratif pada RUP secara global dapat dilihat pada gambar berikut:



Gambar 2.2 RUP (*Rational Unified Process*)

2.4 Fase RUP (*Rational Unified Process*)

RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara teratif. Berikut ini penjelasan untuk setiap fase pada RUP.

1. *Inception* (Permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan dan mendefinisikan kebutuhan akan sistem yang akan dibuat, berikut adalah tahap yang dibutuhkan pada tahap ini:

- a. Memahami ruang lingkup dari proyek (termasuk biaya, waktu, kebutuhan, resiko dan lainnya).
- b. Membangun kasus bisnis yang dibutuhkan.

2. *Elaboration* (Perluasan/Perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih kepada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem.

3. *Construction* (konstruksi)

Tahap ini fokus pada pengembangan komponen fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *initial operational capability milestone* atau batas kemampuan operasional awal.

4. *Transition* (Transisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari batas kemampuan operasional wal. Aktifitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah terpenuhi oleh use.

2.5 Alat Pengembang Sistem (*Unified Modelling Language*)

Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram. Penyesuaian alat yang

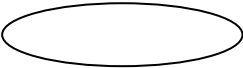


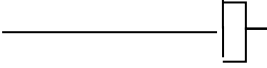
digunakan harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut [15] UML (*unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*).

2.5.1 Use Case Diagram

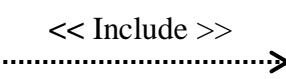

Menurut [15] *Use Case* adalah *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.1 berikut ini:

Tabel 2.1 Simbol *Use Case Diagram*

| No | Simbol | Deskripsi |
|----|---|---|
| 1. |  Usecase | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> . |
| 2. |  Aktor | Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda |
| 3. |  Asosiasi/association | merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor. |
| 4. |  Generalisasi | merupakan hubungan (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum |

Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)


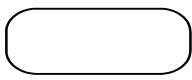
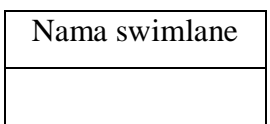
| No | Simbol | Deskripsi |
|----|---|---|
| 5. |  | Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan. |
| 6. |  | Ekstensi (<i>extend</i>) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu. |

Sumber : [15]

2.5.2 Activity Diagram

Menurut [15] *activity diagram* adalah *activity Diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada tabel 2.2 berikut ini :

Tabel 2.2 Simbol *Activity Diagram*

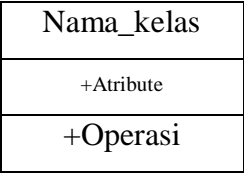


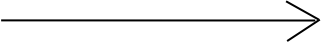
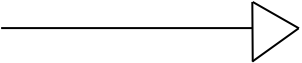
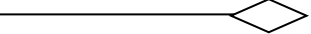
| No. | Simbol | Keterangan |
|-----|---|---|
| 1. |  | Status awal aktivitas sitem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2. |  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3. |  | Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4. |  | Penggabungan (<i>Join</i>) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5. |  | Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas. |
| 6. |  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

Sumber : [15]

2.5.3 Class Diagram

Menurut [15] *Class Diagram* adalah *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.3 berikut ini :

Tabel 2.3 Simbol *Class Diagram*

| No. | Simbol | Deskripsi |
|-----|--|--|
| 1. |  | Kelas pada struktur sistem. |
| 2. | <p>Antar Muka/Interface</p>  <p>Nama_Interface</p> | Sama dengan konsep interface dalam pemrograman berorientasi objek. |
| 3. | <p>Asosiasi / Asociation</p>  | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol |
| 4. | <p>Asosiasi Berarah / Directed Association</p>  | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol. |
| 5. | <p>Generalisasi</p>  | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
| 6. | <p>Agregasi / aggregation</p>  | Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>) |

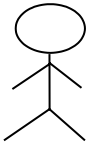

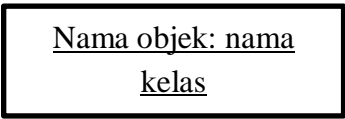

Sumber : [15].

2.5.4 Squence Diagram

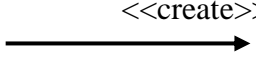

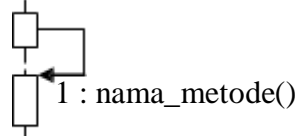

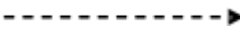
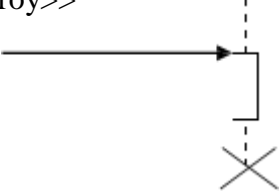
Diagram *squence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen

maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol yang ada pada *sequence diagram* pada Tabel 2.4.

Tabel 2.4 Simbol-simbol *Sequence Diagram*

| No. | Simbol | Deskripsi |
|-----|---|---|
| 1 | Aktor  Atau <u>Nama aktor</u> Tanpa waktu aktif | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda di awal frase nama aktor |
| 2. | Garis hidup / <i>lifeline</i>  | Menyatakan kehidupan suatu objek |
| 3. | Objek  | Menyetakan objek yang berinteraksi peran |
| 4. | Waktu aktif  | Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya |

Tabel 2.4 Simbol-simbol *Sequence Diagram* (Lanjutan)

| No. | Simbol | Deskripsi |
|-----|--|---|
| 5. | Pesan tipe <i>create</i>  | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| 6. | Pesan tipe <i>call</i> 1 : nama_metode()   | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/ metode, maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. |
| 7. | Pesan tipe <i>send</i> 1 : masukan  | Meyatakan bahwa suatu objek mengirimkan data/masukan/informasi/ ke objek lainnnya, arah panah mengarah pada objek yang menerima kembalian. |
| 8. | Pesan tipe <i>return</i> 1: keluaran  | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| 9 | Pesan tipe <i>destroy</i> <<destroy>>  | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>Destroy</i> |

Sumber : [16]

2.6 Black BoxTesting

Black Box Testing merupakan pengujian yang dapat dilakukan dengan melakukan pengamatan, pada hasil eksekusi melalui beberapa data uji dan

memeriksa fungsional yang terdapat pada perangkat lunak. Jadi dapat kita dianalogikan seperti halnya kita melihat ke dalam kotak hitam, sehingga kita hanya bisa melihat tampilan luarnya saja tanpa kita tau apa yang ada didalam kotak hitam tersebut.

Sehingga sama seperti halnya dengan *Black Box Testing* yang hanya dapat mengevaluasi dari tampilan luarnya dan fungsionalitasnya. Tanpa harus mengetahui apa sesungguhnya yang terjadi dalam proses detilnya. Pada pengetahuan khusus dari struktur kode internal dan pengetahuan pada pemrograman dasar pada umumnya tidak diperlukan untuk *Black Box Testing*. Uji pada kasus yang dibangun disekitar spesifikasi dan persyaratan, yakni pada aplikasi yang seharusnya dilakukan.

2.7 Tinjauan Pustaka

Adapun dibawah ini adalah beberapa *literature* yang penulis gunakan dalam penelitian, dapat dilihat pada Table 2.5.

Tabel 2.5 Tinjauan Pustaka

| No. | Nama dan Tahun | Judul | Masalah | Metode | Hasil |
|-----|--------------------|--|---|--------------------------------------|---|
| 1 | Jimi Asmara (2019) | Rancang Bangun Sistem Informasi Desa Berbasis Website (Studi Kasus Desa Netpala) | Rendahnya pengetahuan dan pemanfaatan teknologi informasi bagi pemerintah desa setempat membuat pemanfaatan teknologi informasi dalam memberikan pelayanan kepada masyarakat kurang efektif | system development life cycle (SDLC) | Penelitian ini menghasilkan sebuah website informasi desa seperti menu utama yaitu, profile, lembaga desa, statistik dan potensi keunggulan desa. Dengan adanya website ini akses informasi akan semakin mudah,cepat dan akurat dan peningkatan pelayanan kepada masyarakat akan lebih baik lagi dengan harapan |

Tabel 2.5 Tinjauan Pustaka (Lanjutan)

| No. | Nama dan Tahun | Judul | Masalah | Metode | Hasil |
|-----|---|---|--|--------------------------|--|
| 2 | Mohammad Ariyanto, Pressa Perdana S.S., Andi Rahmad Rahim, Sukaris, Nur Fauziyah (2021) | Sistem Informasi Desa Berbasis Web Di Desa Dahanrejo Kecamatan Kebomas Kabupaten Gresik | Desa Dahanrejo merupakan desa yang mayoritas adalah disektor persawahan dan perikanan namun hal tersebut belum sepenuhnya terekspose didunia maya. | Deskriptif dan kumulatif | Penelitian menghasilkan sistem yang mampu menyampaikan informasi desa meliputi profil, agenda, pengumuman, galeri, transparansi dana dan laporan tahunan. |
| 3 | Herpendi (2017) | Sistem Informasi Desa di Kecamatan Takisung | Informasi berkenaan dengan kepengurusan kependudukan didapatkan dengan datang langsung ke kantor desa atau ke kecamatan, begitu pula dengan segala bentuk formulir kelengkapan berkas kepengurusan kependudukan. Sistem yang sedang berjalan juga menjadikan lambannya proses penyampaian informasi oleh pihak desa atau kecamatan ke para penduduk. | Prototipe | Menghasilkan penelitian dengan penyampaian Informasi data-data kependudukan, dokumen dan formulir kepengurusan kependudukan, informasi kegiatan kecamatan dan desa serta profil desa bisa diakses oleh masyarakat desa secara langsung di laman Sistem Informasi Desa Kecamatan Takisung. Masyarakat akan dimudahkan karena informasi bisa diakses kapanpun dan dimanapun selama terhubung dengan jaringan internet. |

Tabel 2.5 Tinjauan Pustaka (Lanjutan)

| No. | Nama dan Tahun | Judul | Masalah | Metode | Hasil |
|-----|--|--|--|-------------|--|
| 4 | Andri Satria Silo , Frengky Aryanto (2019) | Sistem Informasi Desa Berbasis Web Model Government To Citizen | Dalam menyampaikan informasi kepada masyarakat, pemerintah desa Pambe masih menggunakan media lisan atau melalui pemberitahuan di rumah-rumah ibadah yang ada di desa Pambe maupun dengan mendatangi kantor Desa, sehingga proses penyampaian informasi kepada masyarakat kurang efektif | Diagram UML | dihasilkan sebuah aplikasi sistem informasi pengelolaan data desa Pambe yang efektif dan efisien. Berdasarkan penjelasan di atas, telah dibangun aplikasi berbasis web atau website desa Pambe dengan model Government to Citizen yang mengemas semua informasi desa. Masyarakat juga dapat melakukan pengurusan surat melalui website ini. Pengelolaan data keuangan bumdes yang terintegrasi dalam database, sehingga laporan keuangan dapat dipertanggungjawabkan secara transparan |

Tabel 2.5 Tinjauan Pustaka (Lanjutan)

| No. | Nama dan Tahun | Judul | Masalah | Metode | Hasil |
|-----|--|---|---|--|--|
| 5 | Neni Purwati, Halimah dan Agus Rahardi (2018) | Perancangan Website Program Studi Sistem Informasi Institut Informatika Dan Bisnis Darmajaya Bandar Lampung | Prodi Sistem Informasi (SI) merupakan salah satu prodi yang terdapat di IIB Darmajaya, tetapi belum memiliki fasilitas layanan yang memudahkan mahasiswa, alumni dan dosen untuk memperoleh informasi terkait prodiSI tersebut. | <i>RUP (Rational Unified Process)</i> | Semua informasi yang ada di website prodi (SI) diinputkan didalam database, sehingga masalah kesalahan penginputan informasi dapat diatasi |
| 6 | Nurjoko, Sushanty Saleh dan Sifaul Khoiri (2019) | Rancang Bangun Sistem Informasi Kependudukan Desa Bangun Rejo Berbasis E-Government | Sistem yang dilakukan masih sangat sederhana dan pencarian data akan menjadi tidak efisien dalam hal waktu dan tenaga dengan menggunakan sistem manual | Metode Structured Systems Analysis and Design (SSAD) | Sehingga dengan adanya sistem informasi kependudukan berbasis E government ini memudahkan warga dan petugas dalam mengolah data kependudukan |
| 7 | Bagus Prihadi, Ruki Rizal, Hendra Kurniawan dan Melda Agarina (2020) | Sistem Informasi Kerjasama Vendor Berbasis Web Pada Pt. Pelabuhan Indonesia Ii (Persero) Cabang Panjang | Proses pengolahan data kerjasama vendor yang masih ada dilakukan secara konvensional | <i>Rational Unified Process (RUP)</i> | Hasil penelitian ini adalah sistem dapat menampilkan akta perusahaan, izin perusahaan, kontrak kerjasama, dan notifikasi jatuh tempo kontrak kerjasama |