

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Ayat Al-Quran Yang Berhubungan Dengan Penelitian

Pegawai merupakan subjek penting dalam sebuah organisasi. Tanpa adanya pegawai yang baik, organisasi akan sulit berkembang karena kekuatan setiap organisasi terletak pada pegawai yang mengelola dan menanganinya. Dari permasalahan pada bab sebelumnya urusan kepegawaian tidak akan berjalan efisien karena waktu yang dibutuhkan relatif lebih lama dan hasil dari informasi yang didapatkan tidak tepat waktu, sehingga kebutuhannya tidak terpenuhi dengan baik karena informasi yang diberikan tidak akurat. Penataan yang dilakukan secara teratur, cepat dan akurat serta dapat disajikan dalam sebuah laporan tentunya mendukung kelancaran kegiatan operasional organisasi dan pengambilan keputusan yang tepat. Adapun ayat Al-Quran yang berkaitan tentang pengambilan keputusan yang tepat Allah berfirman yang berbunyi:

وَالْعَصْرِ ۝١ إِنَّ الْإِنْسَانَ لَفِي خُسْرٍ ۝٢ إِلَّا الَّذِينَ ءَامَنُوا وَعَمِلُوا الصَّالِحَاتِ
وَتَوَاصَوْا بِالْحَقِّ وَتَوَاصَوْا بِالصَّبْرِ ۝٣

Artinya : “1. Demi masa.

2. Sesungguhnya manusia itu benar-benar dalam kerugian,
3. kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran” (QS. Al-Ashr. 1-3:103).

Ayat di atas menjelaskan bahwa manusia memang benar-benar berada dalam kerugian apabila tidak memanfaatkan waktu yang telah diberikan oleh Allah secara optimal untuk mengerjakan perbuatan-perbuatan baik. Hanya individu-individu yang beriman dan kemudian mengamalkannya yang tidak termasuk orang yang merugi, serta mereka bermanfaat bagi orang banyak dengan melakukan aktivitas dakwah dalam banyak tingkatan.

2.2 Teori Yang Berhubungan dengan Penelitian

2.2.1 Sistem

Menurut Hanif Al Fatta (2007:3) sistem adalah sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antara objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai tujuan

Menurut Sutabri (2012:6) sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Bedasarkan kedua pengertian diatas, maka dapat disimpulkan bahwa pengertian sistem adalah sekumpulan elemen yang terintegasi dan bekerja bersama guna mencapai suatu tujuan tertentu.

2.2.2 Informasi

Menurut Davis (1995) dalam buku Hanif Al Fatta yang berjudul “Analisis dan Perancangan sistem Informasi, informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang.

Menurut Sutabri (2012:22) informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

Berdasarkan dari pernyataan diatas, maka dapat disimpulkan bahwa informasi adalah suatu data yang diolah menjadi nilai arti untuk digunakan dalam proses pengambilan keputusan.

2.2.3 Sistem Informasi

Menurut Hanif Al Fatta (2007:14) sistem informasi adalah sistem dengan komponen-komponen yang bekerja untuk mengolah data menjadi informasi.

Menurut Sutabri (2012:38) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Berdasarkan dari pernyataan diatas, maka dapat disimpulkan bahwa sistem informasi adalah suatu rangkaian komponen yang saling berkaitan untuk mengumpulkan, memproses serta menyimpan informasi yang mendukung fungsi operasi organisasi dalam pengambilan keputusan.

2.2.4 Data

Menurut Sutabri (2012:2) data adalah kenyataan yang menggambarkan suatu kejadian serta merupakan suatu bentuk yang masih mentah yang belum

dapat bercerita banyak sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi.

Menurut Sutarman (2012:3) data adalah fakta dari suatu pernyataan yang berasal dari kenyataan, dimana pernyataan tersebut merupakan hasil pengukuran atau pengamatan.

Berdasarkan pernyataan diatas, maka dapat disimpulkan bahwa data adalah suatu fakta yang menggambarkan suatu kejadian yang dapat diolah menjadi informasi.

2.2.5 Pengolahan Data

Menurut Sutabri (2012:6) pengolahan data merupakan bahan mentah untuk di olah yang hasilnya kemudian menjadi informasi. Dengan kata lain, data yang telah di peroleh harus di ukur dan di nilai baik dan buruk, berguna atau tidak dalam hubungannya dengan tujuan yang akan di capai.

Menurut Sutarman (2012:4) pengolahan data adalah proses perhitungan atau transformasi data input menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan.

Berdasarkan pernyataan diatas, maka dapat disimpulkan bahwa pengolahan data adalah data yang diolah menjadi bentuk yang lebih berarti yang berupa informasi.

2.2.6 Pegawai

Menurut Undang-Undang Ketenagakerjaan No.13 tahun 2003 mengemukakan bahwa pekerja adalah setiap orang yang bekerja dengan penerimaan upah atau imbalan dalam bentuk lain.

Menurut Undang-Undang Nomor 8 Tahun 1974 tentang pokok-pokok kepegawaian ada dua pengertian tentang pegawai negeri sipil, yaitu:

1. Pegawai Negeri Sipil adalah mereka yang telah memnuh syarat-syarat yang telah ditentukan dalam peraturan perundang-undangan yang berlaku, diangkat oleh pejabat yang berwenang dan disertai tugas dalam suatu jabatan negeri atau disertai tugas negara lainnya yang telah di tetapkan berdasarkan peraturan perundang-undangan dan dikaji menurut peraturan perundang-undangan yang berlaku.
2. Pegawai Negeri adalah unsur-unsur aparatur negara, abdi negara, dan andi masyarakat yang dengan penuh kesetiaan dan ketaatan kepada Pancasila, Undang-Undang Dasar Tahun 1945, negara, dan pemerintah menyelenggarakan tugas pemerintah dan pembangunan.

2.3 Teori Pendukung Lainnya

2.3.1 PHP (*HyperText Preprocessor*)

PHP singkatan dari PHP: *HyperText Preprocessor* yaitu bahasa pemrograman *web server side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server (server side HTML embedded scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Berarti halaman yang akan ditampilkan dibuat saat halaman

itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP di eksekusi pada *server* dimana *script* tersebut dijalankan (Anhar, 2010:3).

2.3.2 MySQL

MySQL dapat didefinisikan sebagai sebuah perangkat lunak sistem manajemen basis data atau DBMS SQL dari sekian banyak DBMS seperti: Oracle, MS SQL, Postrage SQL, dan lain-lain. MySQL merupakan DBMS yang *Multithread, Multi-User* yang bersifat gratis dibawah lisensi GNU *General Public Licence* (GPL) (Anhar, 2010:50).

2.3.3 XAMPP

XAMPP adalah perangkat lunak gratis yang bebas digunakan. Xampp berfungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari Apache HTTP Server, MySQL database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. XAMPP dikembangkan oleh perusahaan apache friends yang memiliki kelebihan bisa berperan sebagai server web apache untuk simulasi pengembang website (Murya, 2017: 1).

2.3.4 Sublime Text 3

Sublime text salah satu kode editor yang biasa digunakan oleh para programmer untuk membuat suatu program. Menurut Faridi (2015:3) menjelaskan bahwa “*Sublime Text 3* adalah editor berbasis *python*, sebuah teks editor yang elegan, kaya akan fitur, *cross platform*, mudah dan simple yang cukup terkenal di kalangan *developer* (pengembang), penulis dan desainer”.

2.3.5 Browser

Browser merupakan perangkat lunak untuk menjalankan program atau *script web*. Contoh *browser* adalah *Internet Explorer*, *Opera*, *Mozilla Firefox* dan lain-lain (Supardi, 2010:2).

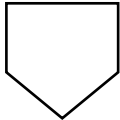
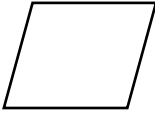

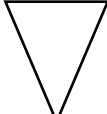
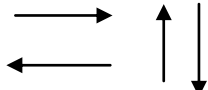
2.4 Teori yang Berhubungan dengan Analisis *Design*

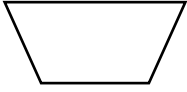

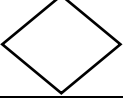
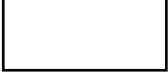

2.4.1 *Flowchart*

Menurut (Jogiyanto, 2005: 795), bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir sistem (*sistem flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menunjukkan urutan dari prosedur-prosedur dan menunjukkan apa yang dikerjakan sistem. Berikut Tabel

2.1 Simbol *Flowchart* :

Tabel 2.1 Simbol *Flowchart*

Simbol	Keterangan
	Penghubung Simbol untuk kelaur/masuk proses dalam lembar atau halaman lain
	<i>Input Output</i> Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya
	Dokumen Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak di kertas
	<i>On Line Storage</i> Simbol yang menyatakan bahwa data bahwa data di simbol ini akan disimpan
	Simbol Garis Alir Digunakan untuk menunjukkan arah selanjutnya yang akan di tuju dari simbol-simbol <i>flowchart</i>

	Manual Simbol yang menunjukkan pengolahan yang tidak menggunakan komputer
	Terminal Simbol yang menunjukkan untuk permulaan atau akhir suatu sistem
	Kondisi Simbol keputusan yang menunjukkan kondisi
	Proses Simbol yang menunjukkan pengolahan dilakukan oleh komputer
	Penghubung Simbol untuk keluar/masuk prosedur atau proses dalam lembar atau halaman yang masih sama

(Sumber : Jogiyanto, 2005:796)

Flowchart atau diagram alir adalah sebuah jenis diagram yang mewakili algoritme, alir kerja atau proses, yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis, dan urutannya dihubungkan dengan panah.

Pada penelitian ini flowchart hanya dipakai untuk menggambarkan sistem yang sedang berjalan dan menggambarkan tahapan penelitian.

2.4.2 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Sukamto, Rosa A dan M. Shalahuddin, 2018:133).

2.4.2.1 *Use Case Diagram*

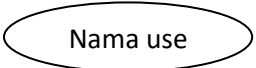
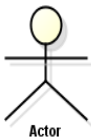

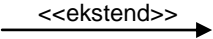
Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat,

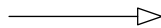
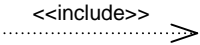
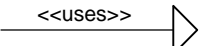
Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *Use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *Use case* yaitu pendefinisian apa yang disebut aktor dan *Use case*.

1. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tetapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (Sukamto, Rosa A dan M. Shalahuddin, 2018:155).

Tabel 2.2 Simbol *Usecase Diagram*

Simbol	Keterangan
<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerjadi awal frase nama <i>use case</i>
<p>Aktor / <i>Actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>
<p>Asosiasi / <i>association</i></p> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<p>Ekstensi/extend</p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; ditambahkan, misal arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.

<p>Generalisasi /<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p>
<p>Menggunakan /<i>include/uses</i></p>  	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i>.</p>




(Sumber : Sukamto, Rosa A dan M. Shalahuddin, 2018:156-158)



2.4.2.2 Activity Diagram

Diagram *activity* adalah menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan *interface* tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Tabel 2.3 Simbol Activity Diagram

Simbol	Deskripsi
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

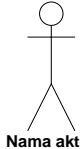
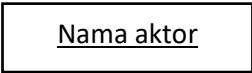

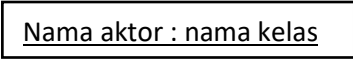
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir


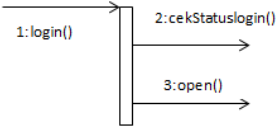


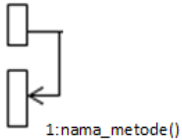

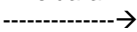

(Sumber : Sukamto, Rosa A dan M. Shalahuddin, 2018:162-163).

2.4.2.3 Sequence Diagram

Diagram sekuen adalah menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *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 dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case* (Sukamto, Rosa A dan M. Shalahuddin, 2018:165).

Tabel 2.4 Simbol Sequence Diagram

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama aktor</p> <p>Atau</p>  <p>Nama aktor</p>	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 menggunakan kata benda di awal frase nama aktor
<p>Garis hidup/<i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
<p>Objek</p>  <p>Nama aktor : nama kelas</p>	Menyatakan objek yang berinteraksi pesan
Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya misalnya

	 <p>Maka cek Status Login () dan open () dilakukan di dalam metode login () aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i> << create >></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarahkan pada objek yang dibuat</p>
<p>Pesan tipe call 1: nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>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</p>
<p>Pesan tipe send 1: masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe return 1:keluaran</p> 	<p>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</p>
<p>Pesan tipe destroy << destroy >></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>.</p>

(Sumber : Sukamto, Rosa A dan M. Shalahuddin, 2018 hal 165-167)

2.4.2.4 Class Diagram

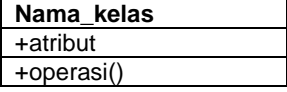

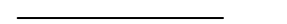
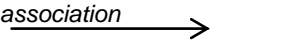
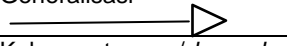
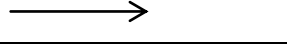
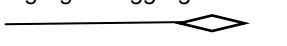
Diagram kelas atau *class* diagram adalah menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas diagram memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas

2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Diagram kelas dibuat agar programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Tabel 2.5 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi/association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/ <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

(Sumber : Sukamto, Rosa A dan M. Shalahuddin, 2018:146-147)

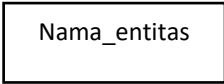
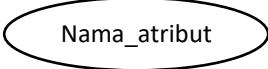
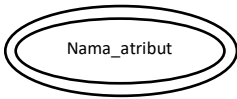

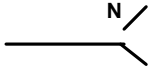
2.4.3 ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan *relationship* data. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS (*Object Oriented Database Management System*) maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen

(dikembangkan oleh Peter Chen), notasi Crow's, dan beberapa notasi lain. Namun paling banyak digunakan adalah notasi dari Chen (Sukamto, R A dan M. Shalahuddin. 2018:50).

Berikut pada Tabel 2.6 Simbol yang digunakan pada ERD dengan notasi Chen:

Tabel 2.6 Simbol ERD (*Entity Relationship Diagram*)

Simbol	Keterangan
Entitas (<i>Entity</i>) 	Orang, tempat atau benda memiliki nama tunggal dan ditulis dengan huruf besar berisi lebih dari 1 instance
Atribut 	Properti dari entitas harus digunakan oleh minimal 1 proses bisnis, dipecah dalam detail
Atribut multivalued 	<i>Field</i> atau data yang butuh isimpan alam suatu entitas yang dapat memiliki nilai lebih dai satu.
Relasi 	Menunjukkan hubungan antar 2 entitas, dideskripsikan dengan kata kerja.
Asosiasi / <i>association</i> 	Penghubung antar relasi an entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.

(Sumber: Sukamto, Rosa A dan M. Shalahuddin, 2018: 50-51)

2.5 Metode Pengembangan Sistem

Kata *Agile* berarti bersifat cepat, ringan, bebas bergerak, waspada. Kata ini digunakan sebagai kata yang menggambarkan konsep model proses yang berbeda dari konsep model-model proses yang sudah ada. *Agile* adalah hasil diskusi 17 pakar pengembangan perangkat lunak di tahun 2001 (Beck et al., 2001). Topik

utama diskusi tersebut adalah metode pengembangan sistem informasi yang sifatnya ringan.

Dalam *Agile Manifesto* terdapat 12 prinsip dasar, yaitu (Abrahamsson et al., 2002; Beck et al., 2001; Cockburn, 2002, 2007; Highsmith, 2002; Koch, 2005; Vanderjack, 2015) :

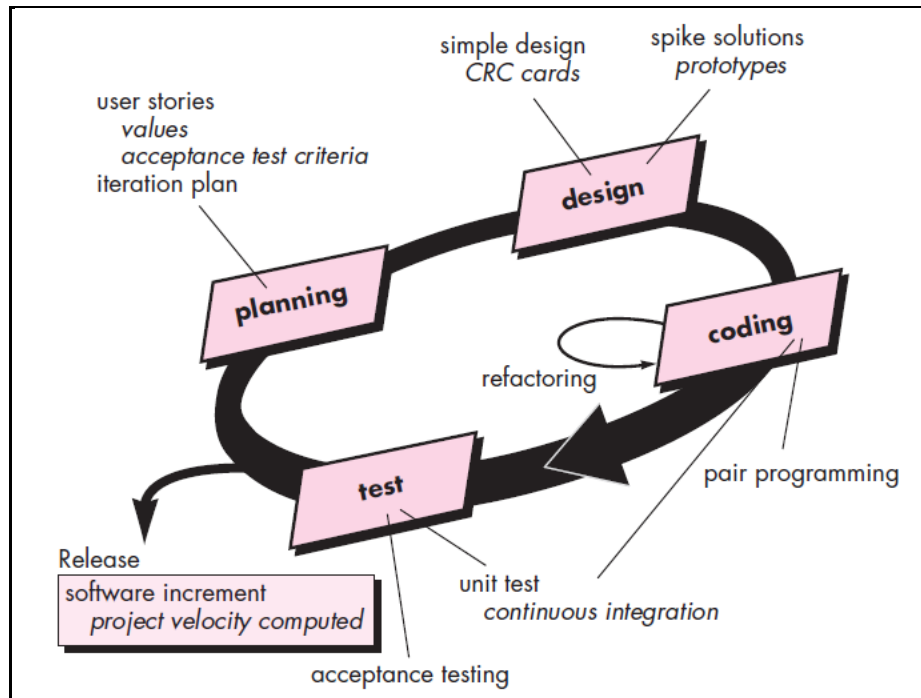
1. Memuaskan pelanggan atau pengguna dengan cara menghasilkan perangkat lunak yang bernilai tinggi secara terus-menerus dan sedini mungkin.
2. Menerima perubahan fungsional dan fitur (*requirements*) bahkan menjelang akhir proyek.
3. Perangkat lunak yang dapat digunakan dihasilkan sesering mungkin (dalam hitungan minggu dan bukan bulan).
4. Kerja sama yang erat antara pengguna dan pengembang dalam keseharian pengembangan sistem informasi.
5. Proyek dikerjakan oleh individu-individu dengan motivasi tinggi dan terpercaya.
6. Komunikasi terbaik adalah secara tatap muka (pada lokasi atau tempat yang sama).
7. Ukuran kemajuan proyek adalah tersedianya perangkat lunak yang sapat digunakan.
8. Pengembangan sistem informasi yang berkesinambungan dan terjaga momentumnya.
9. Prioritas yang terus-menerus akan keunggulan teknis dan rancangan yang baik.

10. Kesederhanaan adalah hal yang esensial. Kesederhanaan dicapai dengan cara meminimalkan aktivitas yang tidak bernilai tambah.
11. Team pengembang mengelola dirinya sendiri.
12. Secara rutin beradaptasi terhadap perubahan situasi selama proyek pengembangan sistem informasi berjalan.

Salah satu metode pengembangan perangkat lunak *agile* adalah *Extreme Programming*. *XP* dikatakan sesuai untuk pengembangan sistem informasi atau perangkat lunak berskala kecil sampai dengan menengah. Satu team *XP* idealnya terdiri atas tiga sampai sepuluh orang pemrograman (D. E. Avison & Fitzgerald, 2006).

Menurut Jeffries (2001) dalam buku Samiaji Sarosa yang berjudul “Metodologi Pengembangan Sistem Informasi” *XP* didefinisikan sub disiplin pengembangan perangkat lunak yang mengutamakan nilai-nilai kesederhanaan, komunikasi, umpan balik, dan keberanian.

Menurut Suryantara (2017:24) para pengembang perangkat lunak banyak menggunakan metode *extreme programming* untuk mengembangkan perangkat lunak dengan cepat. Tahapan pengembangan perangkat lunak dengan *XP* meliputi: *planning*/perencanaan, *design*/perancangan, *coding*/pengkodean, dan *testing*/pengujian.



(Sumber : Pressman, 2010:74)

Gambar 2.1 Kerangka Kerja *Extreme Programming*

2.5.1 *Planning* (Perencanaan)

Kegiatan perencanaan biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan untuk mengumpulkan kebutuhan-kebutuhan yang memungkinkan anggota tekni tim PX memahami konteks bisnis untuk perangkat lunak yang akan dikembangkan dan untuk merasakan perlunya keluaran-keluaran (*output*), fitur-fitur utama, dan fungsionalitas. Aktivitas-aktivitas mendengarkan pada dasarnya mengarah kepada pembuatan serangkaian “cerita” (juga disebut *user stories*).

2.5.2 *Design/Perancangan*

Perancangan XP dengan ketat mengikuti prinsip “tetap sederhana”. Sebuah hasil perancangan yang sederhana selalu lebih disukai daripada gambaran-gambaran yang lebih kompleks. XP mendorong penggunaan kartu CRC sebagai

mekanisme yang efektif untuk berpikir tentang perangkat lunak dalam konteks berorientasi objek. Kartu CRC (*class-responsibility-collaborator*) digunakan untuk mengidentifikasi dan mengatur kelas-kelas dalam konteks pemrograman berorientasi objek yang relevan dengan peningkatan perangkat lunak saat ini. CRC diperkenalkan oleh Kent Beck dan Ward Cunningham sekitar tahun 1989. Bentuk kartu CRC seperti Gambar 2.2.

Nama Kelas :		Nama Kelas : Dosen	
Kelas Induk :		Kelas Induk :	
Kelas Turunan :		Kelas Turunan :	
Tanggung Jawab		Kelas Terkait	
		NID	Jadwal
		Nama	Mahasiswa
		Jenis_Kelamin	Mata_Kuliah
		Status	Nilai
		Alamat	
		Simpan	
		Edit	
		Hapus	

Gambar 2.2 Contoh Kartu CRC

Deskripsi Tentang Kartu CRC:

- Nama kelas (Class Name): memberikan nama kelas.
- Kelas induk (Superclass): merupakan kelas induk (orang tua) dalam konsep pewarisan yang akan dibuat CRC-nya.
- Kelas turunan (Subclass): merupakan kelas anak dalam konsep pewarisan yang akan dibuat CRC-nya.
- Tanggung Jawab (Responsibilities): Atribut, operasi (methods) yang ada pada kelas yang dibuat CRC-nya.
- Kelas Terkait (Collaborators): kelas yang terkait dengan kelas yang dibuat CRC-nya tetapi bukan kelas induk (orang tua)/kelas anak (turunan).

Jika masalah perancangan yang sulit ditemui sebagai bagian dari perancangan suatu cerita, metode pengembangan cepat XP menyarankan

pembuatan langsung dari prototipe operasional dari bagian perancangan tersebut. Disebut sebagai solusi spike, prototipe perancangan diimplementasikan dan selanjutnya dievaluasi. Tujuannya adalah untuk mengurangi risiko-risiko yang akan timbul ketika pelaksanaan yang sesungguhnya dimulai dan untuk memvalidasi keaslian perkiraan untuk cerita yang berisi masalah-masalah perancangan.

XP juga mendorong dilakukannya refaktorisasi, teknik konstruksi yang juga merupakan metode untuk mengoptimasi perancangan. Fowler (Fow00) menggambarkan refaktorisasi dengan cara sebagai berikut:

Refaktorisasi pada dasarnya adalah proses mengubah sistem perangkat lunak sedemikian rupa sehingga tidak mengubah perilaku eksternal kode, namun memperbaiki struktur internal yang ada di dalamnya. Ini merupakan cara yang disiplin untuk membersihkan kode (dan memodifikasi/menyederhanakan rancangan internal), yang pada gilirannya akan meminimalkan kemungkinan akan munculnya kesalahan-kesalahan program (*bug*).

2.5.3 Coding/Pengkodean

Setelah cerita dikembangkan dan karya rancangan awal dilakukan, tim perangkat lunak cepat (XP) tidak langsung beralih ke kode-kode program, tetapi lebih dahulu akan mengembangkan serangkaian unit pengujian yang akan menjalankan setiap cerita yang akan disertakan pada rilis yang ada.

Konsep kunci selama kegiatan pengkodean (dan salah satu yang paling dibahas tentang aspek XP) adalah pemrograman berpasangan (*pair*

programming). XP menyarankan bahwa dua orang pemrogram seharusnya bekerja sama pada satu komputer *workstation* untuk menuliskan kode-kode program untuk suatu cerita.

2.5.4 Testing/Pengujian

Tahap ini memfokuskan pada pengujian fitur-fitur yang ada pada aplikasi sehingga tidak ada kesalahan (*error*) dan aplikasi yang dibuat sesuai dengan keinginan klien (pelanggan). Pengujian pada aplikasi yang akan dibangun menggunakan *unit test* dan *acceptance testing*. Menurut Black (2002:5), *unit testing* merupakan pengujian yang mencakup pengujian sepotong atau sebagian kode. *Test case* untuk pengujian ini dapat dirancang secara struktural atau *behavioral*, tergantung pada standar dari organisasi atau pihak pengembang. *Unit testing* dapat berupa *white box testing* dalam arti bahwa *programmer* mengetahui struktur internal dari unit yang diuji dan berfokus pada bagaimana pengujian mempengaruhi operasi internal. Menurut Lewis (2009:134), setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem *software* memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *black box* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji.

Keuntungan dan Kerugian *Extreme Programming*

Setiap metode memiliki keuntungan dan kerugian, XP juga memiliki keuntungan dan kerugian. Adapun keuntungan dan kerugian tersebut ialah sebagai berikut:

Keuntungan:

1. Dalam hal ini XP menjalin komunikasi/interaksi yang baik dengan pelanggan (klien) pada pengembangan aplikasi.
2. Saling menghargai antar-developer dan meningkatkan komunikasi.
3. Dapat menjadi pembelajaran bagi orang lain (klien) dan dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat.
4. Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat.
5. Developer tidak bekerja secara berlebihan, dikarenakan XP memfokuskan pada aplikasi/program yang dibuat.
6. Dengan XP dapat membuat keputusan-keputusan yang bersifat ternilai.

Kerugian:

1. Perubahan selalu diterima sehingga developer harus selalu siap dengan perubahan-perubahan dari klien.
2. Di awal-awal pengembangan sistem tidak bisa membuat *code* (program) yang detail (prinsip simplicity).
3. XP hanya mengerjakan satu proyek dan dijalankan oleh satu tim. Tidak efektif mengerjakan proyek yang berbeda secara bersamaan (pararel).
4. XP tidak dapat dikerjakan bila developer saling terpisah jauh dengan klien, karena mengakibatkan terhambatnya komunikasi antara developer dengan klien.
5. XP belum terbukti dapat bekerja pada sistem yang mempunyai masalah skalabilitas. Skalabilitas adalah tingkat perencanaan kerja yang diperlukan berdasarkan ukuran proyek, kompleksitas proyek, dan ukuran tim.

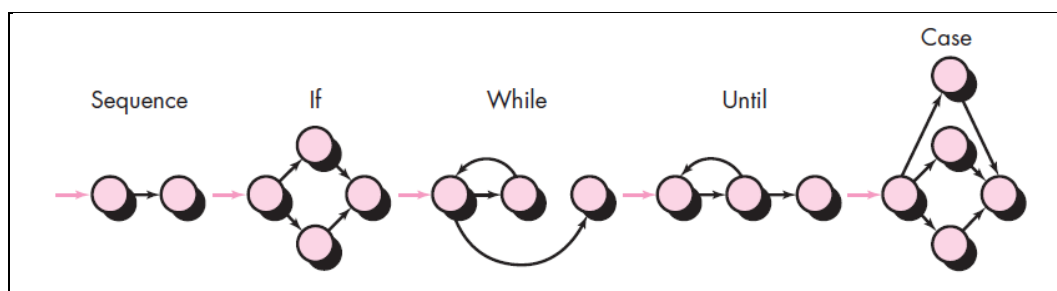
2.6 Pengujian Sistem

2.6.1 *White Box Testing*

Pressman (2012:588) menyatakan pengujian *whitebox testing*, terkadang disebut juga pengujian *glass-box testing*, merupakan sebuah filosofi perancangan *test case* yang menggunakan struktur kontrol yang dijelaskan sebagai bagian dari perancangan peringkat komponen untuk menghasilkan *test case*. Dengan menggunakan metode pengujian kotak putih, dapat memperoleh *test case* yang (1) menjamin bahwa semua jalur independen di dalam modul telah dieksekusi sedikitnya satu kali, (2) melaksanakan semua keputusan logis pada sisi benar dan salah, (3) melaksanakan semua *loop* pada batas mereka dan dalam batas-batas operasional mereka, dan (4) melakukan struktur data internal untuk memastikan kesahihannya.

2.6.1.1 Notasi Grafik Alir (*Flow Graph Notation*)

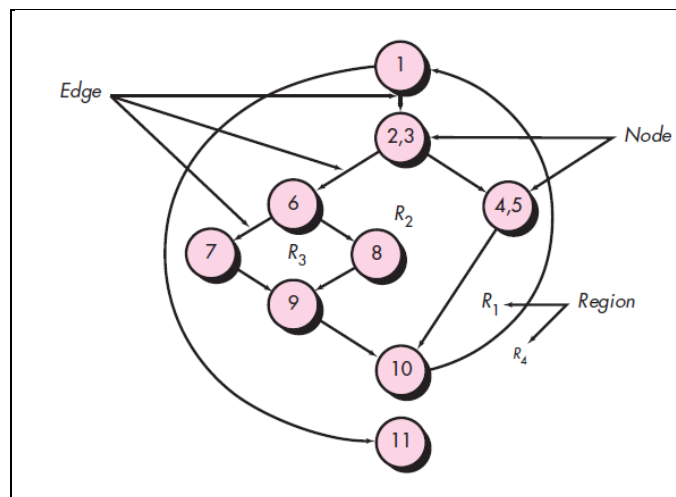
Notasi sederhana untuk representasi aliran kontrol yang disebut *grafik alir* (atau *grafik program*). Grafik alir (*flow graph*) menggambarkan arus kontrol logis dengan menggunakan notasi yang diilustrasikan pada Gambar 2.3. masing-masing bangunan srtuktur memiliki simbol grafik alir yang sesuai.



(Sumber : Pressman, 2010:486)

Gambar 2.3 Notasi Grafik Alir

Untuk menggambarkan penggunaan grafik alir, perhatikanlah representasi perancangan prosedural. Panah pada grafik alir, yang disebut *edge* atau *link*, merupakan aliran kendali dan analog. Sebuah *edge* harus berhenti di sebuah *node*, bahkan jika node tidak mewakili pertanyaan-pertanyaan prosedural (misalnya, lihat simbol-simbol grafik alir untuk konstruksi if-then-else). Area yang dibatasi oleh *edge* dan *node* disebut *region*. Ketika menghitung *region*, kita memasukkan pula area area di luar grafik sebagai *region*. Berikut ini adalah grafik alir:



(Sumber : Pressman, 2010:486)

Gambar 2.4 Grafik Alir

2.6.1.2 Jalur Program Independen (*Independent Program Paths*)

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pertanyaan-pertanyaan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak sepanjang setidaknya satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisikan. Misalnya, satu set jalur independen untuk grafik alir diilustrasikan pada Gambar 2.4 adalah:

Path 1 : 1 – 11

Path 2 : 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Path 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Path 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Perhatikan bahwa setiap jalur baru memperkenalkan *edge* baru.

Path 1 – 2 – 3 – 4 – 5 – 10 – 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

tidak dianggap sebagai jalur independen karena hanya merupakan kombinasi dari jalur yang sudah ditentukan dan tidak melintasi setiap *edge* baru.

Kompleksitas siklomatik dilandaskan pada teori Graph, dan menyediakan metrik perangkat lunak yang sangat berguna. Kompleksitas dihitung dalam salah satu dari tiga cara berikut:

1. Jumlah daerah-daerah grafik alir yang berhubungan dengan kompleksitas siklomatik.

2. Kompleksitas siklomatik $V(G)$ untuk grafik alir G didefinisikan sebagai

$$V(G) = E - N + 2$$

Dimana E adalah jumlah *edge* grafik alir dan N adalah jumlah *node* grafik alir

3. Kompleksitas siklomatik $V(G)$ untuk grafik aliran G juga didefinisikan sebagai

$$V(G) = P + 1$$

Dimana P adalah jumlah *node* predikat yang terdapat dalam grafik alir G

Dari Gambar diatas, kompleksitas siklomatik dapat dihitung menggunakan algoritma-algoritma sebagai berikut.

1. Grafik aliran memiliki empat *region*.
2. $V(G) = 11 \text{ tepi} - 9 \text{ node} + 2 = 4$
3. $V(G) = 3 \text{ node predikat} + 1 = 4$

Oleh karena itu, kompleksitas siklomatik dari grafik alir pada Gambar 2.4 adalah 4. Yang lebih penting, nilai $V(G)$ memberi batas atas untuk jumlah jalur independen yang membentuk basis set, dengan implikasi, sebuah batas atas pada jumlah pengujian yang harus dirancang dan dilakukan untuk memastikan cakupan dari semua pernyataan program.

2.6.2 Black Box Testing

Menurut Pressman (2012:597) *black box testing* atau pengujian kotak hitam juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian kotak hitam memungkinkan anda untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam bukan teknik alternative untuk kotak putih. Sebaliknya, ini merupakan pendekatan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari yang diungkap oleh metode kotak putih.

Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut : (1) fungsi yang salah atau hilang, (2) kesalahan Interface, (3) kesalahan dalam struktur data atau akses basis data eksternal, (4) kesalahan perilaku atau kinerja, dan (5) kesalahan inisialisasi dan penghentian. Dengan

menerapkan teknik kotak hitam, anda mendapatkan serangkaian *test case* yang memenuhi kriteria berikut : (1) *test case* yang mengurangi dengan jumlah yang lebih besar dari satu jumlah *test case* tambahan yang harus dirancang untuk mencapai pengujian yang wajar, dan (2) *test case* yang mengatakan sesuatu tentang ada atau tidaknya kelas kesalahan, daripada kesalahan yang terkait hanya dengan pengujian khusus yang telah dibuat (Pressman, 2012:597-598).

Tabel 2.7 Contoh Pengujian *Black Box Testing*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Menu Keluar	Ketika memilih tombol keluar	Keluar dari aplikasi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

Jadi dapat disimpulkan bahwa *black box testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak atau proses menjalankan aplikasi untuk mengetahui apakah ada *error* atau ada fungsi yang tidak berjalan sesuai yang diharapkan. Sehingga untuk pengujian pada sistem informasi pengolahan data pegawai ini peneliti lebih memilih menggunakan *black box testing* untuk menguji semua fungsi yang terdapat didalam sistem.

2.7 Penelitian Sebelumnya

Tabel 2.8 Penelitian Sebelumnya

No	Penulis	Judul	Tahun	Isi
1	Artiyan Prasetya, Adam Hendra Brata, Mahardeka Tri Ananta	Pengembangan Aplikasi Pemesanan Lapangan Futsal Di Kota Malang Berbasis Android Menggunakan Metode Pengembangan <i>Extreme Programming</i> (Studi Kasus Champion Tidar, Zona SM Futsal, dan Viva Futsal)	2018	Banyak penyedia lapangan yang tidak memberikan informasi atau kemudahan dalam memesan lapangan tersebut. Menurut hasil survey Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) pada tahun 2016, 47.6% atau 63,1 juta masyarakat Indonesia menggunakan internet melalui perangkat mobile mereka. Melihat kondisi di lapangan, untuk membantu memberikan informasi dan membantu memberi kemudahan dalam memesan lapangan, aplikasi pemesanan lapangan diimplementasikan pada perangkat mobile, dengan sistem operasi Android. Aplikasi ini dikembangkan menggunakan metode pengembangan Extreme Programming yang merupakan metode paling sederhana dalam proses perancangan dan pengkodean sehingga dapat menghasilkan waktu pengembangan yang relatif lebih cepat. Selain itu, metode ini cocok digunakan untuk menghadapi kebutuhan dari pengguna yang tidak jelas dan mudah berubah.
2	Ressa Priskila	Perancangansistem Informasi Persediaan Barang Pada Perusahaan Karya Cipta Buana Sentosa Berbasis Web Dengan Metode Extreme Programing	2018	Sebagai sebuah perusahaan pendistribusi hasil laut, pengelolaan persediaanbarang adalah bagian yang sangat penting dalam bisnis proses perusahaan. Kesulitan mencari informasi mengenai stok barang dan pembuatan laporan barang masuk dan keluar menjadi masalah yang dihadapi perusahaan. Untuk mengatasi masalah tersebut dibutuhkan sistem informasi yang dapat memenuhi kebutuhan informasi perusahaan. Penelitian ini bertujuan untuk merancang sistem informasi persediaan barang berbasis web pada perusahaan Karya Cipta Buana Sentosa menggunakan metode Extreme Programming (XP). Sistem informasi ini terdiri dari 3 pengguna yaitu admin, staf gudang dan manajer. Dengan adanya sistem informasi ini, pengelolaan persediaan barang menjadi lebih efektif dan efisien, pencarian informasi persediaan/stok dan laporan juga lebih akurat dan cepat.

3	Bayu Ade Candra, Kurnia Muludi, Anie Rose Irawati	Rancang Bangun Sistem Informasi Manajemen Terpadu (SIMANTEP) Online PT. PLN (Persero) Sektor Pembangkitan Tarahan Lampung Dengan Metode Extreme Programming	2012	Penelitian ini menciptakan Sistem Informasi Manajemen Terpadu atau SIMANTEP Online yang merupakan sistem informasi terintegrasi yang memberikan layanan online pengelolaan dokumen di PT. PLN (Persero) Pembangkit Listrik Sektor Tarahan Lampung. Sistem informasi dibuat menggunakan Extreme Programming (XP), salah satu Pengembangan Agile Metode. Hasil dari penelitian ini adalah sistem informasi online yang tidak hanya dapat diakses di Tarahan Area kantor PLTU, tetapi juga dapat diakses oleh internet. SIMANTEP Online juga memberikan kemudahan untuk pengguna dalam hal manajemen dokumen dan kontrol, mulai dari pengajuan, validasi, dan distribusi dokumen dengan layanan online.
4	Muhdar Abdurahman	Sistem Informasi Data Pegawai Berbasis Web Pada Kementerian Kelautan Dan Perikanan Kota Ternate	2018	Kementerian Kelautan dan Perikanan Nusantara Ternate merupakan salah satu instansi pemerintah yang ada pada kota ternate pada bidang Kelautan dan Perikanan, dalam mengelola dan media menyimpan informasi yang berhubungan dengan data, profil dan kegiatan pegawai masih dilakukan secara manual. Pendataan secara manual dianggap tidak efektif serta membutuhkan waktu, di saat melakukan pengecekan dan pencarian data-data pegawai tersebut harus membuka satu persatu Arsip Pegawai. Penelitian ini dilakukan pada Kementerian Kelautan dan Perikanan Nusantara Ternate, Sistem yang dirancang berbasis Web sebagai media untuk informasi kepada masyarakat, Bahasa Pemrograman yang digunakan adalah HTML, PHP, CSS dan MySQL sebagai database, diharapkan mempermudah Admin dalam mengelola data pegawai secara efisien dan media informasi kepada masyarakat pada Kelautan dan Perikanan
5	Sri Andriati Asri, Widyadi Setiawan	Alternatif Penggunaan Model Pendekatan Agile Pada Perancangan Sistem Informasi Pkl Online	2015	Metode pengembangan perangkat lunak agile merupakan metode yang telah banyak digunakan. Keistimewaan metode ini adalah lebih berorientasi pada program, penggunaan dokumentasi sedikit dan adaptif. Penelitian ini bertujuan untuk menemukan model pendekatan agile yang cocok untuk pengembangan sistem informasi yang akan dibuat. Berdasarkan hasil penelitian, model pendekatan agile Personal Extreme Programming (PXP) cocok digunakan untuk mengembangkan sistem informasi tersebut.
6	Raisya Rahmi, Rika Perdana Sari, Rahmat Suhatman	Pendekatan Metodologi Extreme Programming Pada Aplikasi E-Commerce (Studi kasus Sistem Informasi Penjualan Alat-alat Telekomunikasi)	2016	Extreme Programming atau XP merupakan salah satu metode Agile yang berbeda dari metode Agile yang lain. XP tidak memiliki dokumentasi formal selama proses pembangunannya. Tahapan-tahapan yang harus dilalui dengan XP yaitu, tahapan eksplorasi,

				perencanaan, iterasi pengembangan sistem, dan tahap produksi akhir. Pendekatan XP digunakan untuk membangun sebuah Sistem Informasi Penjualan Alat-alat Telekomunikasi berbasis online atau yang lebih dikenal dengan istilah E-Commerce, dan dengan prinsip Business to Customer (B2C). Sistem ini diperuntukan bagi penjual yang ingin mengembangkan kegiatan jual beli barang dagangan kepada customer secara online. Berdasarkan hasil penelitian diperoleh hasil bahwa dokumentasi perancangan sistem hanya dilakukan pada tahap awal pembangunan yaitu tahap eksplorasi dan tahap perencanaan. Kemudian apabila terjadi perubahan pada saat pembuatan aplikasi, perancangan sistem tidak akan diubah, hanya sistem yang sedang dibangun saja yang berubah
7	Nia Oktaviani	Sistem Informasi Pegawai Berbasis Web Dengan Metode Waterfall Pada SMA AISYIYAH 1 Palembang	2017	Sistem Informasi Pegawai pada SMA AISYIYAH 1 Palembang adalah sebuah sistem yang mengelola data guru dan karyawan di SMA AISYIYAH 1 Palembang. Sistem ini akan mengolah serta memberikan informasi terhadap data pegawai yang ada dengan cepat, mempermudah dalam melakukan perubahan data dan dokumentasi terhadap data-data yang baru seperti data pegawai, data pembagian tugas kerja, dan data absensi, dimana untuk sebelumnya pada Subbag Tata Usaha SMA AISYIYAH 1 Palembang belum memiliki suatu sistem terkomputerisasi yang dapat mengolah data-data tersebut secara cepat dan efisien. Adapun metode pengembangan sistem yang digunakan adalah Waterfall model.
8	Dahlan Abdullah	Perancangan Sistem Informasi Pengolahan Data Pegawai Berbasis Web di SETDAKAB Aceh Utara	2014	Pada Sekretariat Kabupaten Aceh Utara memiliki lebih kurang 240 pegawai negeri. Selama ini dalam proses pengolahan data pegawai masih menggunakan Microsoft office excel, untuk memperoleh atau mengedit data pegawai harus menjumpai bagian yang mengelola data tersebut, sehingga terpikir oleh saya untuk merancang sebuah sistem yang dapat mempermudah kinerja para pegawai. Perancangan dilakukan dengan menggunakan DFD dan ERD untuk memodelkan data dan proses. Dengan adanya sistem ini, diharapkan akan dapat memudahkan dalam pengolahan data pegawai.
9	Muhammad Faizal, Sanda Listya Putri	Sistem Informasi Pengolahan Data Pegawai Berbasis Web (Studi Kasus di PT Perkebunan Nusantara VIII Tambaksari)	2017	Penggunaan teknologi komputer khususnya dalam hal pengolahan data pada saat ini sangat diperlukan. Baik untuk kepentingan perorangan maupun suatu instansi termasuk pada Sistem Informasi Pengolahan data Pegawai di PT Perkebunan Nusantara VIII

				(Tambaksari). Proses pengolahan data pegawai ini memerlukan suatu pencatatan, penyimpanan, serta pelaporan tentang data-data pegawai. Dari permasalahan diatas maka dibuatlah Sistem Informasi Pengolahan data Pegawai di PT Perkebunan Nusantara VIII (Tambaksari) dengan menggunakan bahasa pemrograman PHP dan database MySQL.
10	Bayu Rianto	Perancangan Aplikasi Pengolahan Data Berkas Pajak Pada Kpp Pratama Pekanbaru Berbasis Desktop	2018	Sistem Informasi pengarsipan berkas dan pencarian berkas pada KPP Pratama Pekanbaru tampa saat ini masih konvensional dalam menyelenggarakan fungsi pengarsipan data dan pencarian data pegawai belum melakukannya dengan tepat sasaran dan tepat waktu, dimana dalam proses pengarsipan data pegawai harus mencari tempat yang masih sedikit berkasnya untuk menyimpan atau pengarsipan data. Begitu juga dengan hal pencarian data yang ingin diminta oleh pimpinan pegawai di kpp pratama harus membongkar kembali berkas yang menumpuk satu persatu sehingga memakan waktu yang lama. Penelitian ini memiliki tujuan utama agar terbangunnya sistem informasi yang dapat digunakan dalam sistem pengarsipan dan pencarian berkas, diharapkan dengan adanya sistem ini memaksimalkan pekerjaan pegawai agar dalam pe

Penelitian sebelumnya menjelaskan tentang perbandingan dari beberapa penelitian mengenai Sistem Informasi Pengolahan Data Pegawai dan penerapan metode *Extreme Programming* yang sudah dilaksanakan. Adapun perbedaan yang dimiliki dari judul Sistem Informasi Pengolahan Data Pegawai berbasis web adalah metode pengembangan sistem yang digunakan yaitu *Extreme Programming (XP)*, bahasa pemrograman yang digunakan yaitu PHP (*HyperText Preprocessor*), dan sistem yang dihasilkan yaitu data pegawai, pengajuan cuti, pengajuan surat izin dan surat peringatan.