

BAB II TIJAUAN PUSTAKA

1.1 Sistem Informasi

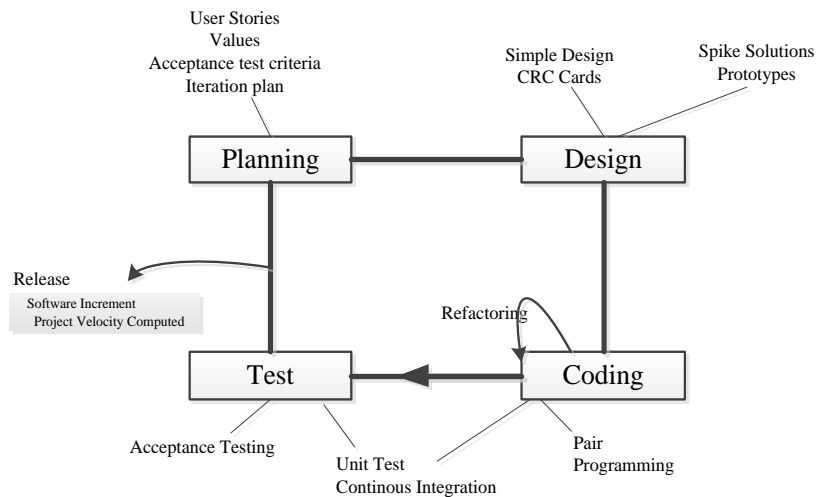
Sistem informasi merupakan sekumpulan elemen yang saling terhubung atau berkaitan untuk tujuan tertentu, proses pengolahan data mentah menjadi informasi secara tersistem dapat menghasilkan informasi yang lebih sederhana dan mudah digunakan oleh pengguna (Saputra, *et al.*, 2021).

1.2 Pondok Pesantren

Pendidikan Pesantren adalah pendidikan yang diselenggarakan oleh Pesantren dan berada di lingkungan Pesantren dengan mengembangkan kurikulum sesuai dengan kekhasan Pesantren dengan berbasis kitab kuning atau dirasah islamiah dengan pola pendidikan muallimin. Pondok Pesantren, Dayah, Surau, Meunasah, atau sebutan lain yang selanjutnya disebut Pesantren adalah lembaga yang berbasis masyarakat dan didirikan oleh perseorangan, yayasan, organisasi masyarakat Islam, dan/atau masyarakat yang menanamkan keimanan dan ketakwaan kepada Allah Swt., menyemaikan akhlak mulia serta memegang teguh ajaran Islam rahmatan lil'alamin yang tercermin dari sikap rendah hati, toleran, keseimbangan, moderat, dan nilai luhur bangsa Indonesia lainnya melalui pendidikan, dakwah Islam, keteladanan, dan pemberdayaan masyarakat dalam kerangka Negara Kesatuan Republik Indonesia (Indonesia, 2019).

1.3 Metode *Extreme Programming*

Extreme programming berdasarkan sejarah singkat bahwa pengembangan perangkat lunak banyak digunakan untuk pengembangan yang lebih cepat dengan meliputi tahapan *planning*, *design*, *coding* dan *testing* . Berikut merupakan konsep *Extreme programming* (Suryantara, 2017).



Gambar 1.1 *Extreme Programming*

1.3.1 Sejarah Singkat *Extreme Programming*

Extreme Programming merupakan salah satu metodologi rekayasa perangkat lunak yang banyak digunakan untuk pengembangan aplikasi oleh para developer, diperkenalkan oleh kent beck yang ditunjukan untuk menangani sebuah proyek penggajian yang dikenal sebagai C3 (*Chrysler Comprehensive Compensation*). Proyek tersebut dimulai pada maret 1996 yang terancam gagal karena rumitnya sistem mengalami kegagalan pada proses testing, kemudian pihak Chrysler menyewa Kent Beck sebagai konsultan di bidang *software engineering* yang kemudian disebut sebagai pencetus XP.

1.3.2 Kerangka Kerja *Extreme Programming*

Pengembangan yang dilakukan menggunakan XP dengan proses yang lebih cepat dengan tahapan seperti *planning*, *design*, *coding* dan *testing*.

1. *Planning*/Perencanaan

Tahap ini dimulain dengan pemahaman konteks bisnis dari aplikasi dengan mendefinisikan keluaran seperti fitur, fungsi, penentuan waktu dan biaya serta alur pengembangan.

2. *Design*/Perancangan

Tahap perencanaan secara sederhana dengan alat mendesain kartu CRC (*Class Responsibility Collaborator*) yang digunakan untuk pemetaan kelas-kelas yang akan digunakan pada diagram UML.

3. *Coding*/Pengkodean

Hal utama dalam pengembangan menggunakan XP yaitu *pair programming* (Proses pembuatan program melibatkan 2 atau lebih programmer).

4. *Testing/Pengujian*

Tahap ini fokus pada pengujian fitur pada aplikasi sehingga tidak ada kesalahan dan sesuai dengan proses bisnisnya.

1.3.3 Nilai Inti *Extreme Programming*

Nilai inti pada pengembangan sistem yaitu :

1. *Communication* (Komunikasi)

Komunikasi antar tim yang digunakan untuk saling berbagi pengetahuan dalam pengembangan.

2. *Simplicity* (Kesederhanaan)

Melakukan semua proses dengan sederhana dan mencoba mencari solusi yang paling sederhana.

3. *Feedback* (Masukan)

Masukan untuk mengetahui kemajuan dari proses dan kualitas perangkat lunak yang dibuat.

4. *Courage* (Kesalahan)

Kesuksesan pengembangan aplikasi harus memiliki keberanian, keyakinan dan integritas dalam pekerjaan.

5. *Respect* (Menghormati)

Menerapkan siklus pendek dan integrasi *continue*.

1.3.4 Aspek Dasar *Extreme Programming*

Aspek dasar pada penerapan metode *extreme programming* yaitu :

1. *The Planning Game*

Proses pendek dan cepat, mengutamakan aspek teknik, memisahkan unsur bisnis dengan unsur teknis dan pertemuan intensif antara klien dengan developer. Pada XP proses ini menggunakan terminologi “game” karena Beck menyarankan untuk menggunakan teknik *score card* dalam menentukan *requirements*.

2. *Small Releases*

Menyelesaikan bagian - bagian aplikasi dan melakukan presentasi kepada *client*, setelah mendapatkan persetujuan maka dilakukan penerapan keaplikasi.

3. *Metaphor*

Menggambarkan visi yang luas terhadap tujuan dari pengembangan perangkat lunak. Dengan Tujuan diharapkan komunikasi antara klien dengan developer akan berlangsung lebih baik.

4. *Simple Design*

Menghindari desain yang rumit dalam sebuah pengembangan perangkat lunak. Dengan desain yang simpel apabila terjadi perubahan dapat meminimalkan kesalahan.

5. *Refactoring*

Melakukan perubahan pada kode program dari perangkat lunak dengan tujuan meningkatkan kualitas dari struktur program tersebut tanpa mengubah cara program tersebut bekerja

6. *Testing*

Membuat test terhadap aplikasi berdasarkan model test yang telah ditentukan.

7. *Pair Programming*

Dua orang programmer saling bekerjasama di komputer yang sama untuk menyelesaikan sebuah unit .

8. *Collective Ownership*

Saling berbagi pengetahuan agar tidak saling ketergantungan pada programmer tertentu ataupun berbagai hambatan akibat perbedaan gaya menulis program dapat diperkecil.

9. *Coding Standard*

Dengan adanya coding standard yang telah disepakati terlebih dahulu maka pemahaman terhadap program akan menjadi mudah untuk semua programmer dalam tim.

10. *Continuous Integration*

Melakukan *build* sesering mungkin berbagai kesalahan pada program dapat dideteksi dan diperbaiki secepat mungkin.

11. *40-hours Week*

Beck berpendapat bekerja 8 jam sehari dan 5 hari seminggu adalah maksimal untuk tiap programmer.

12. *On-Site Customer*

XP menganjurkan bahwa ada anggota dari klien yang terlibat pada proses pengembangan perangkat lunak. Apabila ada kesalahan dalam pengembangan diharapkan klien dapat segera memberikan masukan untuk koreksinya.

1.3.5 Tujuan *Extreme Programming*

Tujuan metode *extreme programming* untuk menghasilkan perangkat lunak yang berkualitas tinggi dan lebih produktif dan mengurangi biaya selama ada perubahan dalam pengembangan perangkat lunak menggunakan siklus pengembangan perangkat lunak singkat.

1.3.6 Daur Hidup Metodologi *Extreme Programming*

Metode XP dapat diterapkan bila:

1. Adanya perubahan yang sangat cepat
2. Memiliki resiko yang tinggi pada pembuatan aplikasi
3. Dalam tim pengembangan aplikasi dengan sedikit programmer
4. Mampu mengotomatisasikan uji sistem
5. Keterlibatan peran serta klien secara langsung
6. Harus ada komunikasi yang baik

1.3.7 Keuntungan dan Kerugian *Extreme Programming*

Keuntungan pada penerapan metode XP yaitu:

1. Dalam hal XP menjalin komunikasi yang baik dengan klien pada pengembangan aplikasi
2. Saling menghargai antar developer dan meningkatkan komunikasi
3. Dapat menjadi pembelajaran bagi orang lain
4. Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat
5. Dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat
6. Developer tidak berkerja secara berlebihan
7. Dengan XP dapat membuat keputusan yang bersifat teknikal




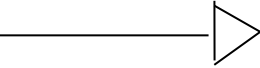
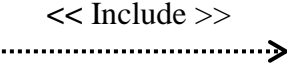
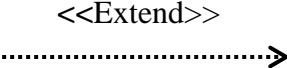
1.4 *Unified Modelling Language (UML)*

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 (Rosa and Shalahuddin, 2019), *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 *Unified Modelling Language*.

1.4.1 Use Case Diagram

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 (Rosa dan Salahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.1.



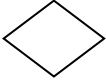

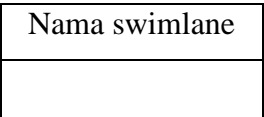

Tabel 1.1 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Use case</i> : 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: seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi (<i>association</i>): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

1.4.2 Activity Diagram

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 (Rosa dan Salahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.2.

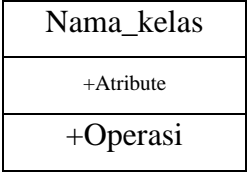
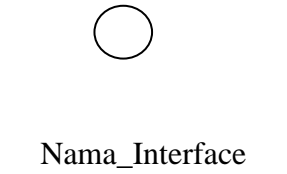
Tabel 1.2 Simbol *Activity Diagram*

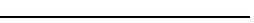
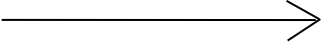
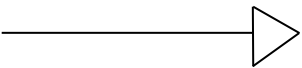
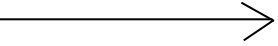
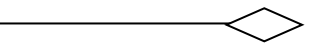
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari 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.

1.4.3 Class Diagram

Class diagram mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa dan Salahuddin, 2019). Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.3.

Tabel 1.3 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.		Sama dengan konsep interface dalam pemrograman berorientasi objek.

No.	Simbol	Deskripsi
3.	Asosiasi / Association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>symbol</i>
4.	Asosiasi Berarah / <i>Digunakan Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>symbol</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Ketergantungan / dependency 	Relasi antar kelas dengan makna ketergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

1.5 Web Based

Web Based adalah aplikasi yang dibuat berbasis *web* yang membutuhkan *web server* dan *browser* untuk menjalankannya (Urbieta *et al.*, 2019).

Sistem informasi dengan berbasis *website* yang dibangun dapat mempermudah pencarian dokumen oleh pihak yang membutuhkan tanpa proses yang terlalu lama (Halimah *et al.*, 2019). Alasan pentingnya suatu *website* memiliki faktor utama dalam pembangunan sebuah *website* yaitu *User Interface*, tanpa adanya analisis rancangan desain yang tepat tentunya pemanfaatan *website* tidak dapat optimal (Agarina, *et al.*, 2019).

Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya:

1. Tidak membutuhkan *hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.
2. Server yang dibutuhkan cukup diinstallkan *tools* pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).

4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

1.5.1 CodeIgniter

CodeIgniter adalah *Framework* untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. *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* (Raharjo, 2018).

Codeigniter adalah sebuah aplikasi gratis yang berupa kerangka kerja untuk membangun website menggunakan bahasa pemrograman PHP (Heru, 2018).

1.5.2 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman untuk membuat website atau situs dinamis dan mengenai rangkaian bahasa pemrograman antara *client side scripting* dan *server side scripting* (Heru, 2018).

PHP merupakan bahasa pemrograman yang digunakan secara luas untuk penanganan, pembuatan dan pengembangan sebuah situs web dan biasanya bersamaan dengan HTML (Oetomo and Maharginono, 2020).

Sehingga PHP merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa HTML dan *script* lainnya.

1.6 Pengujian *Black Box Testing*

Black box testing yaitu pengujian perangkat lunak dari segi pendefinisian fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Rosa dan Salahuddin, 2019).

Pengujian yang dilakukan dengan membuat kasus yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji dilakukan harus dibuat dengan benar dan salah, seperti proses *login* “Jika user memasukan *username* dan *password* yang benar maka dapat *login* ?”.

1.7 Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk mendapatkan bahan perbandingan dan acuan. Selain itu, untuk menghindari anggapan kesamaan dengan penelitian ini. Maka dalam kajian pustaka ini peneliti mencantumkan hasil-hasil penelitian terdahulu sebagai berikut:

Tabel 1.4 Penelitian Terdahulu

Nama	Judul	Variabel	Metode Analisis	Hasil Analisis
Muhammad Fajri, Waskita Cahya, Filda Angellia (2021)	Rancang Bangun Sistem Informasi Pendaftaran Siswa Santri Baru	Pendaftaran santri	studi pustaka, dan wawancara	Sistem pendaftaran ini diharapkan dapat memudahkan admin dalam meng-input, memproses, dan mengelola data pendaftaran santri baru. Mempermudah bagi calon siswa untuk melakukan pendaftaran. Membuat pengelolaan data jadi lebih efektif dan efisien.
Tri Wahyudi (2017)	Rancang Bangun Sistem Informasi Pondok Pesantren (Studi Kasus Darul Abror Watumas)	Informasi Pondok Pesantren	Waterfall	Website yang peneliti bangun ini diharapkan dapat menjadi media informasi bagi masyarakat melalui fasilitas yang disediakan dan menjadi media komunikasi antara pihak Pondok Pesantren Darul Abror Watumas Purwokerto dengan masyarakat, orang tua, santri dan alumni melalui fasilitas buku tamu pada menu utama website
Meltriana Sapitri (2021)	Rancang Bangun Aplikasi Pengolahan Data Kegiatan Santri Di Pondok Pesantren	Pengolahan Data Kegiatan	Waterfall	Dalam penelitian ini, perangkat lunak dikembangkan untuk membantu pelayanan informasi meliputi pengelolaan data berita informasi dan penyedia informasi.

	Kh. Ahmad Dahlan			
Micki Ronaldo dan Donaya Pasha (2021)	Sistem informasi Pengelolaan datasantri Pondok Pesantren An-Nahl Berbasis website	Pengolahan data santri	Objek Orientasi	Menghasilkan Data Pondok Pesantren An-Nahl yang memiliki beberapa fitur seperti menampilkan informasi profil, data santri, pengumuman hingga setoran hafalan santri/wati. Mengenai proses penyampaian informasi secara konvensional

