



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

Landasan teori merupakan bab yang meliputi teori-teori sebagai acuan dalam melakukan penelitian. Teori yang tercantum tentunya berhubungan dengan topik yang diambil dan memiliki kebenaran apa adanya.

2.1. *State of the Art*

CodeIgniter pertama kali dirilis pada 28 Februari 2006 oleh Ellis Lab versi stabil 2.1.0. Kemudian diambil alih dan dikembangkan oleh British Columbia Institute of Technology (BCIT) yang merupakan Sekolah Tinggi Teknologi di Kanada pada bulan Oktober 2014 sampai sekarang menjadi versi 3.1.11 (Rofiah, 2018). Dengan logo api yang menyala, CodeIgniter dengan cepat “membakar” semangat para *web developer* untuk mengembangkan *web* dinamis dengan cepat dan mudah menggunakan *framework* PHP yang satu ini (Parlika, Mubarak, & Munir, 2017). *Framework* CodeIgniter membuat pengkodean dalam PHP menjadi lebih sederhana, cepat dan *user-friendly*. Selain itu, *framework* ini *open-source* dan mudah dikonfigurasi untuk menyesuaikan kebutuhan *developer* (Mandaviya, Raval, & Parekh, 2017)(Jahagirdar & Puranik, 2018). Lalu, pernyataan tersebut dibuktikan dengan hasil pengujian *load test* dan *stress test* didapatkan kesimpulan bahwa aplikasi *web* yang menggunakan *framework* CodeIgniter lebih baik dari sisi performasinya dibandingkan dengan aplikasi *web* yang menggunakan *framewrok* Laravel (Erinton, Negara, & Sanjoyo, 2017). Penggunaan CodeIgniter dengan

konsep MVC membuat penulisan kode program menjadi lebih terstruktur dan lebih aman karena user tidak berhubungan secara langsung dengan *database*. Selain itu, CodeIgniter menyediakan fungsi CRUD (*Create, Read, Update, Delete*) dan *library* yang mempermudah dalam pembuatan aplikasi tersebut (Somya, 2018).

2.2. Kurikulum

Pendidikan sebagai bagian dari kebutuhan manusia, memegang peranan yang sangat penting untuk menciptakan peradaban yang maju. Maju tidaknya suatu peradaban ditentukan oleh baik tidaknya mutu dari pendidikan yang ada pada waktu itu (Baharun, 2015). Dalam dunia pendidikan kurikulum bisa diartikan secara sempit maupun secara luas. Secara sempit kurikulum diartikan hanya sebagai sejumlah mata pelajaran yang harus ditempuh atau diselesaikan siswa di sekolah atau di perguruan tinggi. Secara lebih luas kurikulum diartikan tidak terbatas pada mata pelajaran saja, tetapi lebih luas daripada itu, kurikulum diartikan merupakan aktivitas apa saja yang dilakukan di sekolah dalam rangka mempengaruhi anak dalam belajar untuk mencapai suatu tujuan, termasuk didalamnya kegiatan belajar mengajar, mengatur strategi dalam proses belajar, cara mengevaluasi program pengembangan pengajaran (Wahyuni, 2015).

Sejak tahun 1945, kurikulum di Indonesia telah berulang kali diperbaharui dan disempurnakan. Penyempurnaan itu dilakukan berdasarkan perkembangan yang ada baik dari segi teknologi yang semakin canggih, perkembangan peserta didik, dan tuntutan standar yang ingin dicapai. Perubahan yang terjadi dalam kurikulum membawa kebaikan dalam setiap penyempurnaannya, hingga

perubahan kurikulum saat ini menjadi kurikulum 2013 (Kurniawan & Noviana, 2017).

Sebelum perubahan kurikulum sampai pada kurikulum 2013, terdapat kurikulum 2006 yang dikenal dengan sebutan Kurikulum Tingkat Satuan Pendidikan (KTSP). Berikut merupakan tabel perbandingan kurikulum 2006 dan kurikulum 2013:

Tabel 2.1. Tabel Perbandingan Kurikulum 2006 dan Kurikulum 2013
Sumber: (Setiadi, 2016)

No.	KTSP 2006	Kurikulum 2013
1.	Standar Kompetensi Lulusan diturunkan dari Standar Isi	Standar Kompetensi Lulusan diturunkan dari kebutuhan
2.	Standar Isi dirumuskan berdasarkan Tujuan Mata Pelajaran (Standar Kompetensi Lulusan Mata Pelajaran) yang dirinci menjadi Standar Kompetensi dan Kompetensi Dasar Mata Pelajaran	Standar Isi diturunkan dari Standar Kompetensi Lulusan melalui Kompetensi Inti yang bebas mata pelajaran
3.	Pemisahan antara mata pelajaran pembentuk sikap, pembentuk keterampilan, dan pembentuk pengetahuan	Semua mata pelajaran harus berkontribusi terhadap pembentukan sikap, keterampilan, dan pengetahuan,
4.	Kompetensi diturunkan dari mata pelajaran	Mata pelajaran diturunkan dari kompetensi yang ingin dicapai
5.	Mata pelajaran lepas satu dengan yang lain, seperti sekumpulan mata pelajaran terpisah	Semua mata pelajaran diikat oleh kompetensi inti (tiap kelas)

2.2.1. Kurikulum 2013

Kurikulum 2013 dikembangkan dari kurikulum 2006 (KTSP) yang dilandasi pemikiran tentang tantangan masa depan, persepsi masyarakat, perkembangan pengetahuan dan pedagogi, kompetensi masa depan, dan fenomena negatif yang mengemuka (Ranam & Amaliah, 2017).

Menurut (Mulyasa, 2014) kurikulum 2013 adalah kurikulum yang menekankan pada pendidikan karakter, terutama pada tingkat dasar yang akan menjadi fondasi pada tingkat berikutnya. Pelaksanaan penyusunan kurikulum 2013 adalah bagian dari melanjutkan pengembangan Kurikulum Berbasis Kompetensi (KBK) yang telah dirintis pada tahun 2004 dengan mencakup kompetensi sikap, pengetahuan, dan keterampilan secara terpadu, sebagaimana amanat UU 20 tahun 2003 tentang Sistem Pendidikan Nasional pada penjelasan pasal 35, di mana kompetensi lulusan merupakan kualifikasi kemampuan lulusan yang mencakup sikap, pengetahuan, dan keterampilan sesuai dengan standar nasional yang telah disepakati. Paparan ini merupakan bagian dari uji publik Kurikulum 2013, yang diharapkan dapat menjangkau pendapat dan masukan dari masyarakat (Surahman, 2018)

Kurikulum 2013 mempunyai empat kompetensi inti (KI) yang berisi tujuan dari proses pembelajaran. Rumusan kompetensi inti menggunakan notasi sebagai berikut (Menteri Pendidikan dan Kebudayaan Republik Indonesia, 2013) (Permendikbud No. 67 Tahun 2013 Tentang Kerangka Dasar Dan Struktur Kurikulum Sekolah Dasar/Madrasah Aliyah):

- a. Kompetensi Inti-1 (KI-1) untuk kompetensi inti sikap spiritual,
- b. Kompetensi Inti-2 (KI-2) untuk kompetensi inti sikap sosial,
- c. Kompetensi Inti-3 (KI-3) untuk kompetensi inti pengetahuan,
- d. Kompetensi Inti-4 (KI-4) untuk kompetensi inti keterampilan.

2.2.2. Sistem Penilaian Kurikulum 2013

Penilaian pendidikan merupakan proses pengumpulan dan pengolahan informasi untuk mengukur pencapaian hasil belajar peserta didik mencakup penilaian autentik, penilaian diri, penilaian berbasis portofolio, ulangan, ulangan harian, ulangan tengah semester, dan ulangan akhir semester (Wasito & Warso, 2014). Penilaian merupakan komponen penting dalam proses dan penyelenggaraan pendidikan. Upaya meningkatkan kualitas pendidikan dapat ditempuh melalui peningkatan kualitas pembelajaran dan kualitas sistem penilaiannya. Kualitas pembelajaran dapat dilihat dari hasil penilaiannya. Sistem penilaian yang baik akan mendorong guru untuk menentukan strategi mengajar yang baik dan memotivasi peserta didik untuk belajar dengan lebih baik (Aiman, 2016).

Sistem penilaian pada Kurikulum 2013 diatur dalam Permendikbud Nomor 66 Tahun 2013 di mana guru dan satuan pendidikan wajib mengevaluasi hasil belajar siswa secara komprehensif, menyeluruh, kompleks, dan valid (Mastur, 2017).

Salah satu alat ukur yang digunakan ialah penilaian autentik. Penilaian autentik adalah kegiatan menilai peserta didik yang menekankan pada apa yang seharusnya dinilai, baik proses maupun hasil dengan berbagai instrumen penilaian yang disesuaikan dengan tuntutan kompetensi yang ada di standar kompetensi (SK) atau kompetensi inti (KI) dan kompetensi dasar (KD) (Kunandar, 2013).

Perubahan lain dalam Kurikulum 2013 terdapat pada elemen standar isi. Kurikulum 2013 menghapus istilah Standar Kompetensi mata pelajaran, yang sebelumnya digunakan dalam KTSP, diganti dengan istilah Kompetensi Inti. Mata pelajaran tidak lagi disajikan secara terpisah, akan tetapi terintegrasi dalam bentuk tema (SD dan SMP). Berdasarkan hal itu Kurikulum 2013 dikatakan sebagai *integrated curriculum* ialah kurikulum yang meniadakan batas-batas antara berbagai mata pelajaran dan menyajikan bahan pelajaran dalam bentuk unit atau keseluruhan (Mansur & Mastur, 2018).

Rumus untuk mencari nilai akhir suatu mata pelajaran yang digunakan oleh Sekolah Dasar Santa Patricia adalah:

$$NA = \frac{(2 \times RPH) + PTS + PAS}{4},$$

Rumus 2.1. Rumus Nilai Akhir
Sumber: Sekolah Dasar Santa Patricia

dengan keterangan sebagai berikut:

NA	:	Nilai Akhir
RPH	:	Rata-rata Penilaian Harian
PTS	:	Penilaian Tengah Semester
PAS	:	Penilaian Akhir Semester

2.3. Framework CodeIgniter

Menurut (Sidik, 2012b), dengan menggunakan *framework*, kita tidak perlu membuat program dari awal, tetapi kita sudah diberikan *library* fungsi-fungsi yang sudah diorganisasikan untuk dapat membuat suatu program dengan cepat dan

menurut (Sukamto & Shalahuddin, 2011), *framework* merupakan kerangka kerja yang memudahkan *programmer* untuk membuat sebuah aplikasi sehingga *programmer* akan lebih mudah melakukan perubahan (*customize*) terhadap aplikasinya dan dapat memakainya kembali untuk aplikasi lain yang sejenis. Berdasarkan penjelasan di atas *framework* merupakan kerangka kerja yang memudahkan *programmer* untuk membuat aplikasi dengan *library* fungsi-fungsi yang sudah diorganisasikan untuk dapat membuat suatu program dengan cepat.

CodeIgniter adalah sebuah *framework* php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal (Sidik, 2012a).

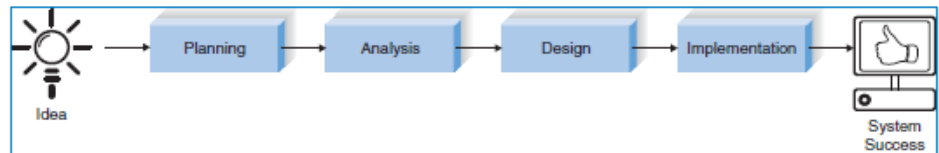
2.4. Metode Pengembangan Sistem

Proses membangun suatu sistem informasi selalu ditentukan oleh metodologi pengembangan yang berbeda. Sebuah metodologi pengembangan sistem informasi mengacu pada kerangka yang digunakan untuk merencanakan, mengelola, dan mengontrol proses pengembangan sistem informasi (Sommerville, 2015).

2.4.1. *Systems Development Life Cycle (SDLC)*

Menurut (Sugiyono, 2016), metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Penelitian menggunakan metode *Systems Development Life Cycle (SDLC)* yang merupakan proses menentukan bagaimana suatu

sistem informasi dapat mendukung kebutuhan bisnis, merancang sistem, membangunnya, dan mengirimkannya kepada pengguna (Dennis, et al., 2012). Metode SDLC mempunyai empat tahap dalam pembentukan sistem yaitu perencanaan, analisis, perancangan, dan implementasi.



Gambar 2.1. Tahap Pembentukan Sistem Metode SDLC
Sumber: (Dennis, Wixom, & Roth, 2012)

Metode SDLC mempunyai tiga kelas metodologi yang terdiri dari beberapa model setiap kelasnya dengan perbandingan sebagai berikut:

Tabel 2.2. Tabel Perbandingan Metodologi SDLC
Sumber: (Dennis et al., 2012)

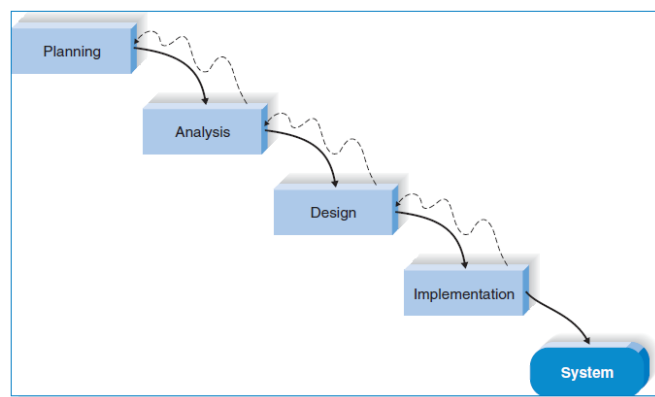
Ability to Develop Systems	Structured Methodologies		RAD Methodologies			Agile Methodologies	
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP	SCRUM
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Good	Good
That Are Complex	Good	Good	Good	Poor	Excellent	Good	Good
That Are Reliable	Good	Good	Good	Poor	Excellent	Excellent	Excellent
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Excellent	Excellent

A. Structured Methodologies

Metode terstruktur merupakan metode pengembangan sistem yang melibatkan pendekatan langkah-demi-langkah untuk proses pembangunan sistem, bergerak secara logis dari satu fase pengembangan ke fase berikutnya. Hasil dari setiap tahap pertama kali disampaikan kepada pelanggan untuk disetujui sebelum memulai tahap berikutnya (Dennis et al., 2012). Bagian dari metodologi ini adalah

1. *Waterfall Development*

Model *Waterfall* adalah model klasik dari rekayasa perangkat lunak dan juga disebut sebagai model siklus hidup sekuensial linier. Dalam model *Waterfall*, setiap fase harus diselesaikan secara keseluruhan sebelum fase berikutnya dapat dimulai. Pada akhir setiap fase, tinjauan dilakukan untuk menentukan apakah proyek berada di jalur yang benar dan apakah akan melanjutkan atau membuang proyek (Dora & Dubey, 2017). Berikut merupakan ilustrasi model *Waterfall*:

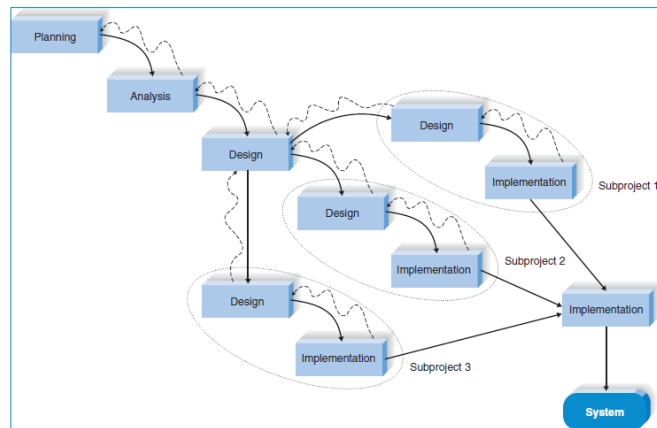


Gambar 2.2. Ilustrasi Model *Waterfall*
Sumber: (Dennis et al., 2012)

2. *Parallel Development*

Model *Parallel* berbeda dari model *Waterfall* karena model ini tidak menunggu penyelesaian tiap fase untuk memulai fase berikutnya. Seluruh proyek dipisahkan menjadi satu set proyek kecil, dan setiap

fase diimplementasikan secara terpisah untuk setiap proyek. Oleh karena itu, proses pengembangan terjadi bersamaan untuk semua sub proyek (Dennis et al., 2012). Berikut merupakan ilustrasi model *Parallel*:



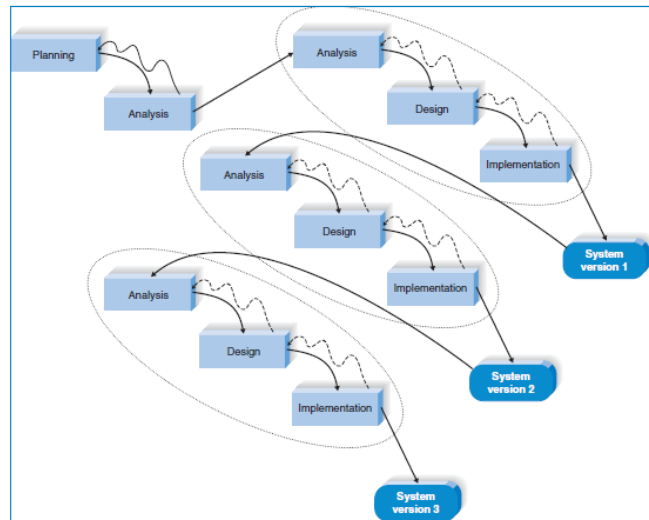
Gambar 2.3. Ilustrasi Model *Parallel*
Sumber: (Dennis et al., 2012)

B. *Rapid Application Development (RAD)*

RAD adalah sebuah proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan dalam waktu yang singkat. RAD menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana *working model* (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan pengguna dan selanjutnya disingkirkan (Aswati & Siagian, 2016). Bagian dari metodologi ini adalah:

1. *Phased Development / Iterative Development*

Melalui *Iterative Development*, seluruh proyek dipecah menjadi beberapa seri/versi yang akan dikembangkan secara berurutan. Yang paling penting dalam pendekatan ini adalah mengembangkan versi pertama dari sistem. Versi ini dikembangkan dengan cepat dengan menerapkan metode *mini waterfall*. Setelah diimplementasikan, pengguna dapat memberikan *feedback* untuk pengembangan sistem pada versi berikutnya. Dalam teknik ini, diperlukan pemahaman yang baik tentang ruang lingkup proyek, sehingga tim pengembangan dapat membangun sistem dengan kemampuan fungsional yang komprehensif hanya dalam waktu singkat (Delima, Santosa, & Purwadi, 2017). Berikut merupakan ilustrasi model *Iterative*:

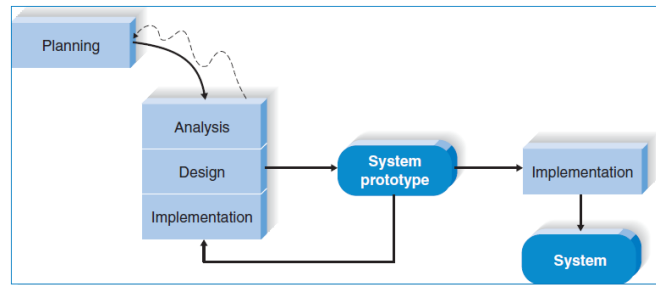


Gambar 2.4. Ilustrasi Model *Iterative*
Sumber: (Dennis et al., 2012)

2. *Prototyping*

Dalam teknik pengembangan sistem *prototyping*, dilakukan fase analisis, desain, dan implementasi agar sistem dapat dikembangkan dengan cepat. Sistem yang merupakan versi pertama hanya berisi fitur minimal yang diperlukan. Selanjutnya, sistem akan diperkenalkan kepada pengguna. Semua komentar dan *feedback* dari pengguna akan dianalisis oleh pengembang dan digunakan sebagai dasar untuk melakukan analisis, desain, dan implementasi ulang untuk prototipe sistem selanjutnya. Siklus ini akan berlanjut hingga tim pengembang, pengguna, dan penyedia dana sepakat bahwa sistem telah memenuhi semua kebutuhan dan fungsi yang diperlukan oleh

organisasi (Delima et al., 2017). Berikut merupakan ilustrasi model *prototyping*:

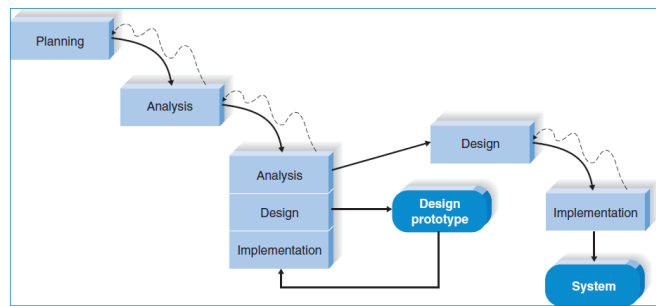


Gambar 2.5. Ilustrasi Model *Prototyping*
Sumber: (Dennis et al., 2012)

3. *Throwaway Prototyping*

Sementara dalam teknik *throwaway prototyping*, proses pengembangan aplikasi awal sama dengan teknik *prototyping*, tetapi setelah prototipe awal dikembangkan, lebih mungkin untuk mengeksplorasi kemungkinan desain alternatif daripada mengembangkan prototipe sistem baru untuk pengembangan aplikasi lebih lanjut. Sistem yang dikembangkan menggunakan teknik ini mungkin memerlukan beberapa desain prototipe selama tahap analisis dan desain. Setiap prototipe digunakan untuk meminimalkan risiko terkait sistem dengan membuat konfirmasi masalah penting yang sudah dipahami sebelum produk akhir dikembangkan. Setelah persetujuan diperoleh, desain ulang sistem akan

diikuti oleh pengembangan produk akhir. Sementara itu desain prototipe dari sistem sebelumnya akan dibuang atau tidak digunakan lagi (Delima et al., 2017). Berikut merupakan ilustrasi model *throwaway prototyping*:



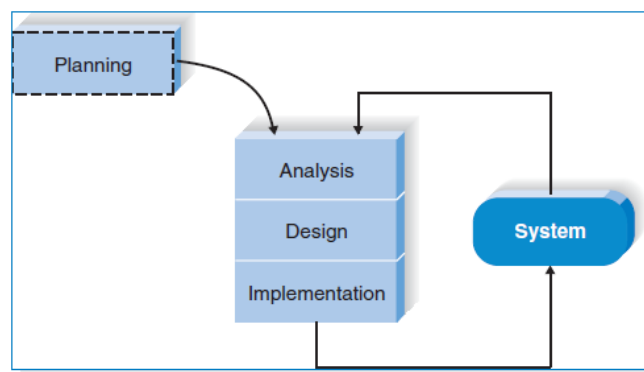
Gambar 2.6. Ilustrasi Model *Throwaway Prototyping*
Sumber: (Dennis et al., 2012)

C. *Agile Development*

Metodologi *Agile* sekarang menjadi model proses yang paling populer dan dapat digunakan dalam industri Pengembangan TI. Model *Agile* lebih fleksibel sifatnya daripada yang lain, model ini merekomendasikan minimal dokumentasi dan menghabiskan lebih banyak waktu untuk pengembangan. Hal ini memungkinkan keterlibatan pengguna akhir dan komunikasi antar tim kerja dan klien untuk memastikan kualitas (Musa & Tariq, 2017). Bagian dari metodologi ini adalah:

1. *Extreme Programming (XP)*

XP adalah metode yang paling sukses untuk mengembangkan perangkat lunak karena fokusnya pada kepuasan pelanggan. XP membutuhkan interaksi pelanggan maksimum untuk mengembangkan perangkat lunak. Ini membagi seluruh siklus hidup pengembangan perangkat lunak menjadi beberapa nomor siklus pengembangan singkat. Ini menyambut dan memasukkan perubahan atau persyaratan dari pelanggan pada setiap fase siklus hidup pengembangan (Sharma, Sarkar, & Gupta, 2012). Berikut merupakan ilustrasi model XP:



Gambar 2.7. Ilustrasi Model *Extreme Programming*

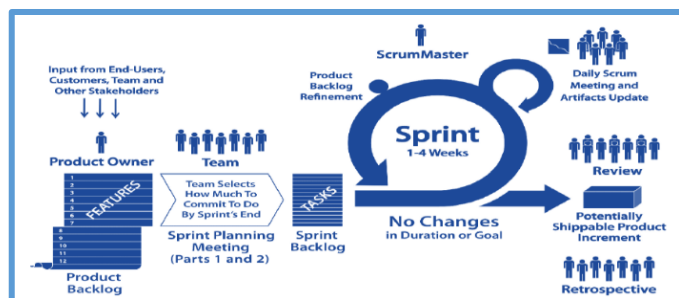
Sumber: (Dennis et al., 2012)

2. *Scrum*

Scrum adalah kerangka pengembangan perangkat lunak ringan yang mewujudkan atribut adaptif, evolusioner, dan kooperatif (Ashraf & Aftab, 2017).

Metode *agile* ini mencakup serangkaian praktik berulang bagi pengembang untuk bekerja sebagai sebuah tim, memberikan kontribusi keterampilan individu mereka untuk mengembangkan perangkat lunak yang berkualitas. Dalam *scrum*, perangkat lunak dikembangkan secara bertahap, menghasilkan versi yang berbeda, dan pada akhir setiap iterasi, produk akhir fungsional dikirim. Pelanggan dapat membuat perubahan atau melanjutkan pengembangan seperti yang direncanakan semula (Hernández, Izaguirre, Mendoza, & Escandón, 2016).

Berikut merupakan ilustrasi model *scrum*:



Gambar 2.8. Ilustrasi Model Scrum
Sumber: (Ashraf & Aftab, 2017)

2.5. *Unified Modelling Language (UML)*

Sistem yang dibangun akan menggunakan konsep *Object Oriented Programming (OOP)* untuk memudahkan pengembangan lebih lanjut. Karena menggunakan konsep OOP, maka perancangan pada pembuatan sistem ini menggunakan *Unified Modelling Language (UML)* untuk mempermudah

pengembang dalam membangun sistem (Sanjani, Hartati, & Sudarmaningtyas, 2014).

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Gata & Gata, 2013).

Menurut (Sukamto & Shalahuddin, 2014), UML terdiri dari 13 diagram yang dikelompokkan dalam 3 kategori. Berikut penjelasan singkat dari pembagian kategori tersebut:

- a. *Structure diagram* merupakan kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.
- b. *Behavior diagram* merupakan kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. *Behavior diagram* terdiri dari *use case diagram*, *activity diagram*, dan *state machine diagram*.
- c. *Interaction diagram* merupakan kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram*

terdiri dari *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction overview diagram*.

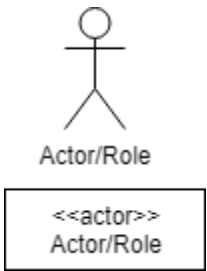
2.5.1. Diagram UML


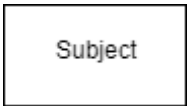
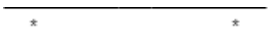
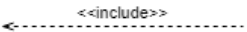
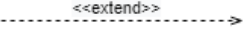

Diagram-diagram yang digunakan untuk membantu pembuatan sistem menurut (Dennis, Wixom, & Tegarden, 2010) adalah

A. Use Case Diagram

Use case diagram menggambarkan fungsionalitas suatu sistem dan merupakan model dialog antara aktor dan sistem. Berikut simbol-simbol yang ada pada *use case diagram*:

Tabel 2.3. Tabel Simbol-Simbol Use Case Diagram
Sumber: (Dennis et al., 2010)



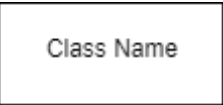




Simbol	Deskripsi
	<p><i>Actor:</i></p> <ul style="list-style-type: none"> • Seseorang atau sistem yang memperoleh manfaat dari dan bersifat eksternal terhadap subjek. • Digambarkan sebagai figur tongkat (default) atau jika aktor bukan manusia terlibat, sebagai persegi panjang dengan <<actor>> di dalamnya (alternatif). • Diberi label dengan perannya. • Dapat dikaitkan dengan aktor lain menggunakan asosiasi spesialisasi/superclass, dilambangkan dengan panah dengan panah berongga. • Ditempatkan di luar batas subjek.


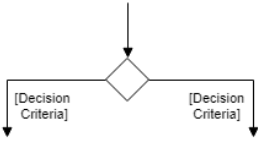
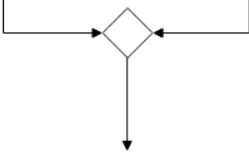
Simbol	Deskripsi
	<p><i>Use case:</i></p> <ul style="list-style-type: none"> • Mewakili bagian utama dari fungsionalitas sistem. • Dapat memperpanjang kasus penggunaan lain. • Dapat menyertakan use case lain. • Ditempatkan di dalam batas sistem. • Dilabeli dengan frase kata benda deskriptif.
	<p><i>Subject Boundary:</i></p> <ul style="list-style-type: none"> • Termasuk nama subjek di dalam atau di atas. • Mewakili ruang lingkup subjek, misalnya, sistem atau proses bisnis individu.
	<p><i>Association Relationship:</i></p> <ul style="list-style-type: none"> • Menautkan aktor dengan <i>use case</i> yang berinteraksi dengannya.
	<p><i>Include:</i></p> <ul style="list-style-type: none"> • Merupakan penyertaan fungsi dari satu <i>use case</i> di dalam yang lain. • Memiliki panah yang ditarik dari <i>use case</i> dasar ke <i>use case</i> yang dipakai.
	<p><i>Extend:</i></p> <ul style="list-style-type: none"> • Merupakan ekstensi dari <i>use case</i> untuk memasukkan perilaku opsional. • Memiliki panah yang ditarik dari <i>use case</i> ekstensi ke <i>use case</i> dasar.
	<ul style="list-style-type: none"> • <i>Generalisasi/generalization:</i> Merupakan <i>use case</i> penggunaan khusus ke yang lebih umum. • Memiliki panah yang ditarik dari <i>use case</i> khusus ke <i>use case</i> dasar.

B. *Activity Diagram*

Activity diagram menggambarkan *workflow* (aliran kerja) dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut simbol-simbol yang ada pada *activity diagram*:

Tabel 2.4. Tabel Simbol-Simbol *Activity Diagram*
Sumber: (Dennis et al., 2010)

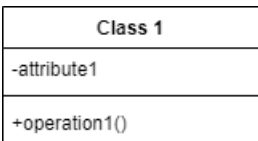
Simbol	Deskripsi
	<p><i>Action:</i></p> <ul style="list-style-type: none"> • Adalah perilaku yang sederhana, tidak dapat diurai. • Diberi label namanya.
	<p><i>Activity:</i></p> <ul style="list-style-type: none"> • Digunakan untuk mewakili serangkaian tindakan. • Diberi label namanya.
	<p><i>Object Node:</i></p> <ul style="list-style-type: none"> • Digunakan untuk mewakili suatu objek yang terhubung ke satu set aliran objek. • Diberi label berdasarkan nama kelasnya.
	<p><i>Control Flow:</i></p> <ul style="list-style-type: none"> • Memperlihatkan urutan eksekusi.
	<p><i>Object Flow:</i></p> <ul style="list-style-type: none"> • Memperlihatkan aliran suatu objek dari satu aktivitas (atau aksi) ke aktivitas lain (atau aksi).
	<p><i>Initial Node:</i></p> <ul style="list-style-type: none"> • Menggambarkan awal dari serangkaian tindakan atau kegiatan.
	<p><i>Final-activity Node:</i></p> <ul style="list-style-type: none"> • Digunakan untuk menghentikan semua aliran kontrol dan objek mengalir dalam suatu kegiatan (atau tindakan).




Simbol	Deskripsi
	<p><i>Final-flow Node:</i></p> <ul style="list-style-type: none"> Digunakan untuk menghentikan aliran kontrol tertentu atau aliran objek.
	<p><i>Decision Node:</i></p> <ul style="list-style-type: none"> Digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. Diberi label dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.
	<p><i>Merge Node:</i></p> <ul style="list-style-type: none"> Digunakan untuk menyatukan kembali jalur keputusan yang berbeda yang dibuat menggunakan simpul keputusan.

C. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode/operasi. Berikut simbol-simbol yang ada pada *class diagram*:

Tabel 2.5. Tabel Simbol-Simbol *Class Diagram*
Sumber: (Dennis et al., 2010)


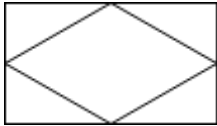
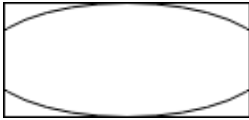
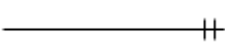

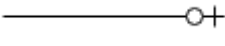
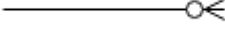
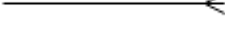
Simbol	Deskripsi
	<p><i>Class:</i></p> <ul style="list-style-type: none"> Merupakan jenis orang, tempat, atau hal yang perlu ditangkap dan disimpan oleh sistem. Memiliki nama yang diketik tebal dan berada di tengah-atas kompartemen.

Simbol	Deskripsi
	<p><i>Generalization:</i></p> <ul style="list-style-type: none"> • Merupakan semacam hubungan antara beberapa kelas.
	<p><i>Aggregation:</i></p> <ul style="list-style-type: none"> • Merupakan sebuah hubungan logis antara beberapa kelas atau kelas dan itu sendiri. • Merupakan bentuk khusus dari suatu asosiasi.
	<p><i>Composition:</i></p> <ul style="list-style-type: none"> • Merupakan sebuah hubungan fisik antara beberapa kelas atau kelas dan itu sendiri. • Merupakan bentuk khusus dari suatu asosiasi.

2.6. Entity Relationship Diagram (ERD)

Salah satu *tool* diagram yang digunakan untuk memodelkan abstraksi data adalah *Entity Relationship Diagram* (ERD). Menurut (Mulyani, 2016), *Entity Relationship Diagram* (ERD) adalah *tools* yang digunakan untuk melakukan pemodelan data secara abstrak dengan tujuan untuk mendeskripsikan atau menggambarkan struktur dari data yang digunakan. Adapun fungsi utama ERD yaitu sebagai alat untuk memodelkan hasil dari analisis data, sebagai alat untuk memodelkan data konseptual dan sebagai alat untuk memodelkan objek-objek dalam suatu sistem. Berikut simbol-simbol yang ada pada ERD versi Peter Chen (Sukamto & Shalahuddin, 2014):

Tabel 2.6. Tabel Simbol-Simbol ERD
Sumber: (Kendall & Kendall, 2014)

Simbol	Deskripsi
	<i>Entity:</i> Sebuah kelas orang, tempat, atau benda.
	<i>Associative entity:</i> Digunakan untuk menggabungkan dua entitas.
	<i>Attributive entity:</i> Digunakan untuk pengulangan grup.
	<i>To 1 relationship:</i> Tepat satu.
	<i>To many relationship:</i> Satu atau lebih.
	<i>To 0 or 1 relationship:</i> Hanya nol atau satu.
	<i>To 0 or more relationship:</i> Bisa nol, satu, atau lebih.
	<i>To more than 1 relationship:</i> Lebih dari satu.

2.7. Testing

Pengujian adalah satu set aktivitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktivitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik (Sukamto & Shalahuddin, 2014).

2.7.1. Black-box Testing

Black-box testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak (Jaya,

2018). Keuntungan penggunaan metode *black-box testing* (Ammann & Offutt, 2016) adalah

- a. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
- b. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
- c. *Programmer* dan tester keduanya saling bergantung satu sama lain.

Sedangkan kerugian dari metode *black-box testing* (Ammann & Offutt, 2016) adalah

- a. Uji kasus sulit dilakukan tanpa spesifikasi yang jelas.
- b. Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer*.
- c. Beberapa bagian *back-end* tidak teruji sama sekali.

2.7.2. *White-box Testing*

White-box Testing adalah salah satu cara untuk menguji suatu aplikasi dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Jika *output* yang dihasilkan dari pengujian tidak sesuai dengan modul, maka akan dilakukan kompilasi ulang dan di cek kembali kode-kode tersebut hingga sesuai dengan yang diharapkan (Mustaqbal, Firdaus, & Rahmadi, 2015).

Keuntungan penggunaan metode *white-box testing* (Nidhra & Dondeti, 2012) adalah

a. Kesalahan Logika

Menggunakan syntax ‘if’ dan syntax pengulangan. Langkah selanjutnya metode white box testing ini akan mencari dan mendeteksi segala kondisi yang di percaya tidak sesuai dan mencari kapan suatu proses perulangan di akhiri.

b. Ketidaksesuaian Asumsi

Menampilkan dan memonitor beberapa asumsi yang diyakini tidak sesuai dengan yang diharapkan atau yang akan diwujudkan, untuk selanjutnya akan dianalisa kembali dan kemudian diperbaiki.

c. Kesalahan Pengetikan

Mendeteksi dan mencaribahasa-bahasa pemograman yang di anggap bersifat *case sensitive*.

Sedangkan kerugian dari metode *white-box testing* (Nidhra & Dondeti, 2012) adalah pada perangkat lunak yang jenisnya besar, metode white box testing ini dianggap boros karena melibatkan banyak sumber daya untuk melakukannya.

2.7.3. Test Case

Test case adalah seperangkat nilai input, prakondisi eksekusi, hasil yang diharapkan, dan postkondisi eksekusi yang dikembangkan untuk

tujuan atau kondisi pengujian tertentu, seperti untuk menjalankan program tertentu atau untuk memverifikasi kepatuhan dengan persyaratan tertentu (Black, 2009). Dengan menggunakan *test case* penguji dapat menemukan *defect* atau *bug* pada aplikasi sebelum aplikasi digunakan pengguna. Kasus uji harus berisikan pengujian setiap menu sebuah aplikasi untuk mencegah adanya cacat. (Bentley, Bank, & NC, 2005)

2.8. Penelitian Terdahulu

Tabel 2.7. Tabel Perbandingan Jurnal

Jurnal 1	
Nama Jurnal	Jurnal Sisfo Vol. 07 No.02, 2018, 149-164
Nama	Susilo Veri Yulianto dan Ardian Prima Atmaja
Judul	Rancang Bangun Sistem Informasi Kurikulum 2013 Tingkat Sekolah Dasar Berbasis Web dengan SDLC Waterfall
Metode	Pada penelitian ini, model pengembangan sistem informasi yang digunakan adalah waterfall. Waterfall adalah model pengembangan perangkat lunak yang paling banyak digunakan. Model pengembangan waterfall bersifat linear dari tahap perencanaan sampai tahap pemeliharaan. Tahapan berikutnya tidak akan dilakukan sebelum tahapan sebelumnya selesai, dan tidak bisa kembali ke tahap sebelumnya.
Hasil	Penelitian ini hanya meliputi tahapan analisis kebutuhan, perancangan sistem dan perangkat lunak, implementasi dan pengujian unit, dan integrasi dan pengujian sistem. Tahapan terakhir dari metode waterfall, yaitu tahap operasional dan perawatan, tidak termasuk dalam ruang lingkup penelitian ini. Hasil dari tahap analisis kebutuhan, perancangan sistem dan perangkat lunak dapat dilihat pada penjelasan di bagian metodologi. Untuk tahap implementasi dan pengujian unit dilakukan dengan memprogram (<i>coding</i>) terhadap sistem informasi yang dibangun berdasarkan analisis kebutuhan dan perancangan sistem dan perangkat lunak yang telah dilakukan sebelumnya. Pengujian unit dilakukan dengan metode <i>white box testing</i> untuk mengetahui dan memastikan kode program yang ditulis sudah benar.

	Tahap berikutnya adalah integrasi dan pengujian sistem. Sistem informasi yang telah dibangun dipindahkan ke web server dan diintegrasikan dengan basis data yang sudah dibuat dan diinstall sebelumnya pada suatu database server. Dalam penelitian ini, web server dan database server merupakan komputer yang sama. Sistem informasi tersebut dapat diakses melalui komputer client dengan menggunakan web browser. Pengujian sistem merupakan pengujian sistem informasi keseluruhan dengan melibatkan <i>user</i> yang sebenarnya. Pengujian ini menggunakan metode black box testing. Hasil pengujian dapat dilihat pada Tabel 4, 5, dan 6
Kesimpulan	Sistem informasi Kurikulum 2013 yang dikembangkan merupakan sistem informasi yang secara spesifik diperuntukan dalam membantu proses penilaian berdasarkan Kurikulum 2013 di tingkat Sekolah Dasar (SD). Dengan menggunakan sistem informasi ini, proses penilaian menjadi lebih mudah dan cepat. Dalam penggunaan sistem informasi ini, guru kelas perlu didampingi oleh operator sekolah yang berperan dalam mengelola data guru, siswa, orang tua atau wali, mata pelajaran, ekstrakurikuler, kelas, dan <i>user</i> (akun guru kelas).
Jurnal 2	
Nama Jurnal	Jurnal Ilmu Pengetahuan dan Teknologi Komputer Vol. 3 No. 1, Agustus 2017, 2527-4864
Nama	Nu'man Musyaffa
Judul	Rancang Bangun Sistem Informasi Akademik pada SMP Bina Nusa Wisata Tangerang Menggunakan Model Waterfall
Metode	Model yang digunakan dalam melakukan pengembangan software adalah model waterfall, yang meliputi beberapa tahap diantara lain: <ol style="list-style-type: none"> 1. Analisa Kebutuhan 2. Sistem dan Desain 3. Implementasi dan Pengujian Unit 4. Integrasi dan Pengujian Sistem 5. Pengoperasian dan Perawatan
Hasil	Setelah di desain sedemikian rupa, sistem informasi akademik Bina Nusa Wisata selanjutnya di implementasikan kedalam bentuk desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan seperti PHP dan MySQL. Program yang dibangun langsung diuji secara unit baik adanya kesesuaian serta kesalahan desain pada sistem informasi akademik Bina Nusa Wisata dan adanya kesalahan sistem (<i>bug</i>) dalam pengujian <i>script</i> program pada web. Selanjutnya dilakukan integrasi sistem informasi akademik Bina Nusa wisata dengan basis data yang telah disesuaikan

	<p>berdasarkan kebutuhan menggunakan perangkat lunak mysql, kemudian dilakukan pengujian sistem keseluruhan. seperti pengujian keamanan dalam sistem informasi ini mulai dari kerahasiaan siapa yang dapat mengakses data karyawan, siswa maupun guru. Selanjutnya adalah melakukan pengujian browser, Sistem Informasi Akademik dapat dijalankan pada beberapa browser baik menggunakan PC maupun Gadget sekalipun. Dan yang terakhir adalah pengujian kegunaan sistem informasi akademik diuji berdasarkan navigasi situs nya, terkait dengan menu, tombol atau tautan ke berbagai halaman dalam sistem informasi akademik Bina Nusa Wisata harus mudah dalam penggunaannya.</p> <p>Tahap terakhir adalah mengoperasikan sistem informasi akademik Bina Nusa Wisata dan melakukan secara berkala perawatan sistem informasi akademik seperti memperbaharui isi dari Sistem Informasi Akademik dan penyesuaian serta perubahan data, baik data karyawan, siswa, guru, data nilai, data mata pelajaran dan yang lainnya, melakukan monitor statistik sistem informasi akademik agar mendapatkan data- data yang bermanfaat untuk pengembangan sistem informasi akademik Bina Nusa Wisata, serta melakukan penambahan fitur – fitur menarik, seperti forum diskusi orang tua siswa dan guru, fasilitas chatting, dan fasilitas lainnya. Berikut adalah spesifikasi perangkat keras dan perangkat lunak untuk meng-hosting Sistem Informasi Akademik.</p>
Kesimpulan	<ol style="list-style-type: none"> 1. Aplikasi Sistem Informasi Akademik saat ini sangat dibutuhkan dalam rangka peningkatan penyampaian informasi yang cepat dan up to date. Hal ini dapat terlihat dari sistem yang sedang berjalan dimana untuk pembuatan data rapor. Wali kelas terlebih dahulu mengumpulkan nilai dari masing-masing guru mata pelajaran, dimana pengolahan data nilai dari guru mata pelajaran belum tentu tepat waktu sehingga dapat menghambat proses pembagian hasil nilai rapor. 2. Sistem Informasi Akademik ini dirancang untuk memenuhi kebutuhan informasi bagi siswa, karyawan ataupun masyarakat yang ingin mengetahui berbagai informasi tentang sekolah tanpa batasan ruang dan waktu. Dengan demikian penyampaian informasi dapat dilakukan tanpa harus datang langsung ke sekolah. 3. Adanya Sistem Informasi Akademik ini, karyawan (guru) dapat menginput nilai siswa tanpa harus datang ke sekolah dan menyerahkan nilai raport siswa tanpa harus datang ke sekolah dan bertatap muka dengan siswa yang bersangkutan

	dikarenakan semua informasi sudah ada di dalam sistem informasi akademik yang bersifat online.
Jurnal 3	
Nama Jurnal	Jurnal Sistem Informasi Vol. 01 No.02, Oktober 2017, 2579-5341
Nama	Haida Dafitri dan Marina Elsera
Judul	Rancang Bangun Sistem Informasi Akademik Berbasis Web (Studi Kasus: SMA Swasta Harapan I Medan)
Metode	Dalam rancang bangun sistem informasi akademik ini penulis menggunakan model scrum, model ini digunakan dianggap lebih baik untuk mengelola proyek secara praktikal. Dikarenakan model ini dapat menuntun tim peneliti untuk melakukan hal-hal yang perlu dan menyarankan hal-hal yang dianggap tidak perlu dalam menginspeksi proses dan melakukan adaptasi terus menerus untuk mengerjakan proses tahap demi tahap sesuai dengan kebutuhan pihak sekolah.
Hasil	<p>Perancangan arsitektur framework dari sistem perangkat lunak yang dibangun. Proses deskripsi arsitektur mengadopsi spesifikasi sistem, model analisis dan interaksi subsistem yang telah didefinisikan pada tahap analisis sistem sebelumnya. Pada Gambar dijelaskan bahwa admin memiliki wewenang untuk mengelola data akademik sekolah, akses database. Sistem informasi akademik menyediakan berbagai menu yang dapat digunakan oleh siswa, guru yang ingin memberikan, menerima dan melihat informasi akademik melalui web. Misal Guru dapat menginput hasil nilai ujian harian, nilai mid dan nilai uas mata pelajaran yang telah diajarkan dikelas tersebut, wali kelas dapat menerima nilai yang telah diinput oleh berbagai guru sehingga pada akhirnya wali kelas dapat mencetak hasil raport siswa kelas yang dikelolanya. Setelah itu siswa dapat melihat nilai ujian harian, mid dan uas yang telah diinput oleh masing-masing guru yang mengajar mata pelajaran sesuai dengan roster jadwal belajar.</p> <p>Dalam merancang suatu sistem, perlu diketahui dan diidentifikasi terlebih dahulu kebutuhan-kebutuhan dari sistem yang akan dibuat yang telah disesuaikan dengan kebutuhan dari sisi <i>user</i>, fungsionalitas sistem yang akan dirancang serta dukungan lingkungan yang dibutuhkan. Beberapa proses yang digunakan untuk membangun sistem informasi akademi tersebut dapat dilakukan dengan cara:</p> <ol style="list-style-type: none"> 1. Mencari dan mengumpulkan data siswa/siswi di SMA Swasta Harapan I Medan 2. Mencari dan mengumpulkan data guru-guru di SMA Swasta Harapan I Medan

	<p>3. Mencari dan mengumpulkan data orang tua/wali murid siswa SMA Swasta Harapan I Medan</p> <p>Langkah selanjutnya adalah menyiapkan kebutuhan sistem. Kebutuhan sistem yang diperlukan meliputi dua kebutuhan yaitu kebutuhan perangkat keras dan perangkat lunak yang digunakan.</p>
Kesimpulan	<ol style="list-style-type: none"> 1. Sistem Informasi akademik yang dibangun dengan model scrum terdiri dari beberapa menu yang mempunyai fungsi dan pendukung dalam penyampaian informasi akademik yang dapat digunakan oleh guru/wali kelas, siswa dan orangtua/ wali siswa dalam menginput dan mengelola nilai bagi guru dan wali kelas dapat mensinkronkan nilai siswa sehingga hasil akhir dapat mencetak raport, sedangkan siswa, orangtua/wali murid dapat melihat nilai ujian harian, mid, uas, prestasi siswa lainnya maupun absensi kehadiran siswa di sekolah SMA Swasta Harapan I Medan. 2. Dengan adanya sistem informasi akademik yang dibangun dapat membantu pihak sekolah dalam menyampaikan sebuah informasi akademik kepada orang tua/wali murid, siswa dan guru secara efektif dan efisien baik dari segi dana operasional dan waktu. 3. Pemanfaatan model scrum yang digunakan dalam rancang bangun sistem informasi akademik tersebut mendapatkan hasil yang baik dan sesuai dengan kebutuhan yang diinginkan oleh pihak sekolah dalam merubah sistem informasi akademik konvensional menjadi komputerisasi berbasis web. Sistem yang dikembangkan bersifat dinamis sehingga dapat digunakan selamanya, sehingga disaat masuk pada ajaran baru dapat digunakan seperti biasanya, hanya saja tugas admin untuk memanipulasi data siswa dan guru. 4. Siswa dan orang tua/siswa dapat mengetahui prestasi, nilai dan informasi akademik melalui web dan melalui E-SMS

Berdasarkan tabel perbandingan jurnal 2.7, sistem informasi dibuat berdasarkan beberapa penelitian terhadap sekolah yang membutuhkan sistem menggunakan model *Waterfall* oleh (Yulianto & Atmaja, 2018). Model *Waterfall* yang dipakai tidak mencakup dua tahapan terakhir yaitu tahap operasional dan perawatan. Adapun model *Waterfall* yang mencakup semua tahapan digunakan untuk membuat sistem informasi SMK Bina Nusa Wisata (Musyaffa, 2017). Selain

itu, sistem informasi SMA Swasta Harapan I Medan dibuat menggunakan model *Scrum* (Dafitri & Elsera, 2017).

Adapun hal yang diadopsi dari ketiga jurnal tersebut yaitu struktur *interface website*, struktur *database*, dan struktur perancangan sistem (Dafitri & Elsera, 2017; Musyaffa, 2017; Yulianto & Atmaja, 2018). Pembaharuan yang membedakan penelitian dari ketiga jurnal tersebut adalah model metode penelitian yang dipakai yaitu model *Extreme Programming (XP)* dan memakai *Framework CodeIgniter*.

Sistem penilaian akademik Sekolah Dasar Santa Patricia akan dibangun mengikuti kebutuhan sekolah, oleh karena itu sistem cocok menggunakan model XP yang fleksibel dan mengakomodasi perubahan yang dibuat selama iterasi. Selain itu, menggunakan *Framework CodeIgniter* yang menggunakan metode MVC (*Model, View, Controller*) juga memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal.