

BAB 2

TINJAUAN PUSTAKA

Bab ini menjelaskan teori-teori yang bersumber dari buku, jurnal, ataupun artikel. Berfungsi sebagai dasar dalam melakukan pengerjaan penelitian dengan tujuan agar konsep dan teori penyelesaian permasalahan yang digunakan dapat dipahami.

2.1 Pondok Modern Asy-Syifa Balikpapan

Pondok Modern Asy-Syifa Balikpapan telah berdiri sejak tahun 1994 dan menjadi lembaga pendidikan berbasis pendidikan islami berasrama yang mandiri dengan status swasta penuh dan berbadan hukum dan berada di bawah naungan Yayasan Asy-Syifa Balikpapan. Kurikulum pendidikan yang digunakan oleh Pondok Modern Asy-Syifa Balikpapan mengkombinasikan 3 (tiga) kurikulum pendidikan, yaitu Kurikulum Pondok Modern Darussalam Gontor (PMDG), Kementerian Agama (Kemenag), serta Kementerian Pendidikan dan Kebudayaan (Kemendikbud). Metode pengajaran, situasi dan kehidupan para siswa atau santrinya diupayakan sama dengan Pondok Modern Gontor seperti hidup sederhana, disiplin, berwawasan luas, mempunyai kemampuan berbahasa arab dan inggris, kepramukaan, latihan kepemimpinan dalam organisasi, bela diri, membaca kitab berbahasa arab dan lain-lain, sehingga kemampuan pengetahuan dan sikap yang dimiliki para lulusannya berguna bagi perkembangan selanjutnya baik di lembaga pendidikan yang lebih tinggi (bagi yang melanjutkan pendidikannya) maupun di masyarakat kerja.

Pondok Modern Asy-Syifa Balikpapan memiliki visi terwujudnya suatu pendidikan yang memiliki dasar kokoh berlandaskan islam, mandiri, handal, terbuka, kreatif dan inovatif sehingga dapat menjadi Pondok Pesantren pilihan orang tua. Untuk menjalankan visi tersebut, Pondok Modern Asy-Syifa Balikpapan memiliki misi sebagai berikut:

1. Menyelenggarakan pendidikan 6 (enam) tahun.

2. Menerapkan kurikulum terpadu antara Pondok Modern Gontor, kurikulum Kemenag, serta kurikulum Kemendikbud.
3. Menyiapkan sarana dan prasarana memadai.
4. Menyiapkan tenaga edukatif yang selain mampu mentransfer ilmunya, juga dapat berperan sebagai fasilitator, motivator dan dinamisator.
5. Menyediakan tenaga administrator yang cakap dan memiliki etos kerja yang tinggi.
6. Menerapkan pendidikan teori dengan praktek dalam kehidupan sehari-hari.
7. Melaksanakan program yang jelas, terinci, sistematis dan dapat dipertanggungjawabkan.
8. Menciptakan kerjasama yang harmonis antara tenaga edukatif dan non-edukatif, santri, orang tua dan masyarakat.
9. Meningkatkan hubungan dengan instansi pemerintah dan swasta.
10. Meningkatkan kesejahteraan tenaga edukatif dan non-edukatif.

Pondok Modern Asy-Syifa Balikpapan pada saat ini dipimpin oleh Pimpinan Pondok Modern Asy-Syifa Balikpapan yang berhubungan satu sama lain dengan Ketua Yayasan Asy-Syifa Balikpapan, Ketua P4 (Pembina Personal Pendidikan Pondok) dan Ketua KMI (Kuliyatul Mu'allimin Al-Islamiyyah).

Pondok Modern Asy-Syifa mewajibkan santri tingkat MTs dan MA untuk tinggal di asrama serta mengikuti seluruh kegiatan yang ada di asrama. Terdapat organisasi santri KMI yang berada di bawah naungan Kepengasuhan Pondok Modern Asy-Syifa Balikpapan yang berfungsi untuk mengontrol jalannya kegiatan-kegiatan asrama yang bernama Organisasi Pelajar Pondok Modern (OPPM). Organisasi tersebut memiliki bagiannya masing-masing mulai dari bagian keamanan, ibadah, pengajaran, bahasa, koperasi pelajar yang mengontrol kegiatan sesuai dengan bidangnya masing-masing. OPPM memiliki masa jabatan selama 1 (satu) tahun dan akan dilakukan pergantian kepengurusan oleh tingkat selanjutnya setelah masa jabatan berakhir. Kegiatan yang menghasilkan prestasi internal diadakan oleh setiap seksi-seksi OPPM, seperti perlombaan pidato 3 (tiga) bahasa yang diadakan oleh bagian pengajaran, program *tahfidz* (hafalan Al-Qur'an) yang diadakan oleh bagian ibadah, kegiatan kepramukaan dan perlombaan kepramukaan yang diadakan oleh bagian koordinator kepramukaan. Kemudian untuk masalah

yang berhubungan dengan keamanan akan diawasi oleh bagian keamanan. Pelanggaran yang dilakukan oleh santri akan dicatat oleh bagian keamanan OPPM dan selanjutnya dilaporkan ke Staf Pengasuhan untuk diberlakukan sanksi berdasarkan tingkat pelanggarannya (ringan, sedang atau berat). Staf Pengasuhan akan mengeluarkan surat terkait pelanggaran yang dilakukan oleh santri sebelum dilakukan proses pemberlakuan sanksi dan dilakukan pencatatan pelanggaran pada buku pelanggaran. Di dalam tata tertib Pondok Modern Asy-Syifa Balikpapan disebutkan klasifikasi pelanggaran berat dengan sanksi berupa dikeluarkan dari Pondok Modern Asy-Syifa Balikpapan sebagai berikut:

1. Melanggar aturan dan norma syariat islam.
2. Melawan Pimpinan Pondok dan seluruh stafnya.
3. Menggunakan narkoba dan minuman yang memabukkan.
4. Bertato.
5. Mencuri (Barang, Uang, Pakaian, Makanan, dll).
6. Memberikan sanksi fisik yang mengakibatkan cedera (pemukulan).

Selain dari klasifikasi pelanggaran berat, juga terdapat klasifikasi pelanggaran sedang dengan sanksi berupa peringatan, menyurati orang tua/wali, diumumkan di depan seluruh santri, pemanggilan orang tua dan skorsing sebagai berikut:

1. Menghina orang lain.
2. Memanggil teman dengan gelar/sebutan yang jelek.
3. Keluar pondok tanpa izin.
4. Bertindik.
5. Mewarnai rambut dan kuku.
6. Merokok.
7. Datang terlambat.
8. Anak lama masuk kamar anak baru begitu pula sebaliknya.
9. Berhubungan dengan orang luar.
10. Berkelahi.
11. Berkomunikasi dengan santriwati

Kemudian juga terdapat pelanggaran ringan dengan sanksi berupa diberi peringatan, tindakan langsung dan diberikan penugasan yang pelanggarannya

terbagi dalam beberapa jenis, berupa larangan secara umum, kamar, masjid atau mushola, ruang makan, kelas, kantin dan telepon sebagai berikut:

1. Mencuci/melaundry selain di tempat yang di tentukan Pondok.
2. Masuk kamar santri lain kecuali mendapat izin dari mudabbir.
3. Berbahasa daerah dan menggunakan bahasa rusak (plesetan).
4. Membawa alat elektronik (*handphone*, radio, tape recorder, laptop, speaker, flashdisk dan setrika).
5. Rambut menyentuh telinga mata, kerah baju, dan bermodel.
6. Makan dan minum sambil berdiri.
7. Memakai kacamata selain kacamata kir.
8. Menceoret coret dinding di sembarang tempat.
9. Memakai celana ketat dan norak (botol, Joger, dan lain-lain).
10. Meminjam peralatan dapur atau yang lainnya ke masyarakat.
11. Memakai jeans dan pakaian berkantong banyak atau bermodel aneh (benang tidak sama dengan warna kain/bahan serta pakai yang berbahan jeans).
12. Memakai sarung di bawah mata kaki.
13. Membiarkan pakaian di tempat jemuran lebih dari sehari.
14. Membawa senjata tajam (pisau, *cutter*, tang, dan obeng).
15. Melipat lengan baju dan celana serta menggulung sarung sampai lutut.
16. Memakai celana pendek.
17. Memakai topi di lingkungan asrama.
18. Berteriak-teriak dan bersiul.
19. Membawa buku bacaan yang tidak sesuai dengan alam pendidikan islami.
20. Bersepeda/motor di lingkungan asrama.
21. Membawa teman luar pondok ke dalam kamar.
22. Berbicara dengan teman luar di sekitar pondok.
23. Mengecat, mewarnai, dan memanjangkan rambut serta kuku.
24. Pinjam meminjam pakaian.
25. Memakai gelang, kalung, cincin, perhiasan dalam bentuk apapun.
26. Menjadi anggota organisasi selain organisasi pondok.
27. Berada di luar lingkungan pondok dan tempat-tempat terlarang.
28. Merusak peralatan dan sarana pondok.

29. Menempel logo, stiker, atau sejenisnya pada bangunan pondok termasuk lemari.
30. Membawa ATM kecuali mendapatkan izin dari pengasuhan.

2.2 Website Development

Aplikasi atau *software* menjadi bagian yang tidak dapat dipisahkan dalam suatu sistem komputer. Melihat dari lingkungan pengembangannya, terdapat pembagian dari aplikasi berupa aplikasi berbasis *desktop*, aplikasi berbasis *website* dan aplikasi berbasis *mobile*. Aplikasi berbasis *desktop* adalah aplikasi yang penggunaannya membutuhkan proses instalasi di komputer yang akan menggunakan aplikasi tersebut, contohnya seperti *microsoft office*, *mozilla firefox*, *google chrome* dan masih banyak lagi. Aplikasi berbasis *desktop* adalah aplikasi yang penggunaannya tidak memerlukan instalasi pada komputer yang akan menggunakannya, dimana penggunaannya dilakukan dengan cara membuka *browser* yang terhubung melalui jaringan ke server. Terakhir, aplikasi berbasis *mobile* adalah aplikasi yang hanya dapat berjalan pada perangkat bergerak (*mobile*) seperti *smartphone*, contohnya seperti LINE, *Whatsapp* dan masih banyak lagi (Solichin, 2005).

Website dapat diartikan sebagai sekumpulan halaman yang memiliki beberapa laman dan berisi informasi dalam bentuk data digital yang berupa teks, gambar, video, audio dan animasi lain melalui jalur koneksi internet. Untuk lebih jelasnya, *website* merupakan halaman yang ditampilkan oleh *browser* dan memiliki informasi (Abdulloh, 2016). Saat ini aplikasi berbasis *website* berkembang sangat pesat karena penggunaannya yang memiliki beberapa kelebihan dibandingkan dengan aplikasi berbasis *desktop*. Beberapa kelebihan yang dimiliki oleh aplikasi berbasis *website* adalah sebagai berikut (Solichin, 2005):

1. Penggunaan yang tidak memerlukan instalasi, sehingga adanya perubahan pada aplikasi (*update*) tidak perlu dilakukan proses *update* oleh *client* (pengguna).
2. Aplikasi dapat diakses dari mana saja melalui jaringan. Jika server aplikasi menggunakan LAN (jaringan intranet) maka aplikasi dapat diakses oleh seluruh komputer dalam jaringan tersebut, sementara jika server aplikasi

menggunakan internet (memiliki IP *Public* dan diletakkan di *web hosting*) maka aplikasi dapat diakses melalui internet.

3. Data aplikasi disimpan di sisi server, sehingga akses data dari sisi *client* dapat diatur tergantung dengan kebutuhan.
4. *Cross-platform* atau dapat diakses melalui komputer dengan berbagai OS seperti windows, linux ataupun mac yang memiliki *browser*.
5. Bagi *client*, penggunaan aplikasi tidak memerlukan spesifikasi yang tinggi karena hampir keseluruhan proses aplikasi dilakukan di server.
6. Lebih aman dari virus karena aplikasi berjalan di *browser*.

Aplikasi berbasis *website* juga memiliki kekurangan, terlebih dari ketergantungan aplikasi terhadap kondisi jaringan yang digunakan. Jaringan yang lambat akan mempengaruhi performa aplikasi, baik itu dari sisi kecepatan akses aplikasi maupun kecepatan proses di dalamnya. Kemudian, karena aplikasi dapat diakses dimanapun sehingga lebih rentan dari serangan pihak yang tidak bertanggung jawab. Sehingga perlindungan dan pengamanan aplikasi tersebut seutuhnya dilakukan oleh pengembang.

Pada pengembangan aplikasi berbasis web terdapat berbagai teknologi penting yang akan digunakan seperti *web server*, *web browser*, HTML, CSS, *Javascript*, PHP, *Database Server* dan IDE (*Editor*).

2.3 *Agile Software Development*

Agile methods mulai diperkenalkan pada dekade 90an untuk menangani kelemahan yang tidak mampu untuk menangani penambahan atau perubahan kebutuhan pada saat proses pengembangan *software*. *Agile Software Development* merupakan salah satu metode pengembangan *software* yang memiliki sifat adaptif dan responsif terhadap perubahan. Adaptif karena mudah menyesuaikan diri dengan keadaan, responsif karena cepat dalam merespon (cepat tanggap). Hasil yang dikembangkan sampai saat ini sudah cukup banyak, diantaranya adalah *Extreme Programming*, *Scrum Methodology*, *Crystal Family*, *Dynamic System Development Method*, *Adaptive Software Development* dan *Feature Driven Development* (Ilmawan & SN).

Pada tahun 2001 terdapat 17 (tujuh belas) praktisi yang mengungkapkan pandangan terhadap *Agile Software Development Methods* (ASDMs) yang di dalamnya terdapat bagian-bagian yang dianggap penting, seperti yang disebutkan pada Tabel 2.1, dimana ASDMs lebih memperhatikan individu dan interaksi ketimbang proses dan alat, software yang bekerja ketimbang dokumentasi yang komprehensif, kolaborasi *customer* ketimbang negosiasi kontrak dan menanggapi adanya perubahan ketimbang mengikuti rencana. Terdapat 12 (dua belas) prinsip yang menunjukkan karakteristik ASDMs seperti berikut (Cho, 2008):

1. Kepuasan pelanggan menjadi prioritas tertinggi melalui penyerahan awal yang berkelanjutan dari *software* yang berharga.
2. Menerima perubahan kebutuhan, meskipun perubahan tersebut terdapat di akhir pengembangan.
3. Mengirimkan *software* yang dikerjakan (yang berfungsi) dengan pilihan batas waktu yang paling singkat seperti beberapa minggu atau beberapa bulan.
4. Pebisnis dan pengembang harus bekerja bersama-sama setiap harinya selama pengerjaan proyek.
5. Proyek dibangun bersama dengan individu-individu yang memiliki motivasi dengan memberikan lingkungan dan dukungan sesuai dengan kebutuhan. Selain itu juga mempercayai mereka untuk menyelesaikan pekerjaannya.
6. Metode yang paling efisien dan efektif untuk menyampaikan informasi kepada tim pengembang adalah dengan cara *face-to-face* atau tatap muka.
7. *Software* yang berfungsi merupakan pengukur utama kemajuan.
8. Proses *agile* mempromosikan pengembangan yang berlanjut. Para sponsor, pengembang dan *user* harus mempertahankan kecepatan yang konstan tanpa batas.
9. Perhatian secara terus menerus terhadap teknis yang unggul dan desain yang baik akan meningkatkan kelincahan.
10. Kesederhanaan (mengurangi jumlah pekerjaan yang belum selesai) itu penting.
11. Arsitektur, kebutuhan dan desain yang terbaik muncul dan diatur oleh tim itu sendiri.

12. Tim menggambarkan caranya untuk menjadi lebih efektif, kemudian menyesuainya secara berkala.

Sementara menurut Ian Sommerville di dalam bukunya Software Engineering Edisi Kesembilan, metode agile memiliki 5 (lima) prinsip yang harus diterapkan dalam penggunaannya berupa *customer involvement*, *incremental delivery*, *people not process*, *embrace change* dan *main simplicity*. Penjelasan prinsip-prinsip tersebut dijelaskan pada Tabel 2.2.

Tabel 2.1 Pandangan Terhadap ASDMs

<i>More Valuable Items</i>	<i>Less Valuable Items</i>
<i>Individuals and interactions</i>	<i>Process and tools</i>
<i>Working software</i>	<i>Comprehensive documentation</i>
<i>Customer collaboration</i>	<i>Contract negotiation</i>
<i>Responding to change</i>	<i>Following a plan</i>

Tabel 2.2 Prinsip Penggunaan ADMs

<i>Principle</i>	<i>Description</i>
<i>Customer involvement</i>	Pelanggan harus ikut terlibat dalam proses pengembangan, dimana mereka berperan untuk menyediakan dan memprioritaskan kebutuhan baru, juga mengevaluasi serangkaian proses yang telah dilakukan.
<i>Incremental delivery</i>	Pengembangan <i>software</i> dilakukan secara bertahap dengan penentuan kebutuhan yang dilakukan oleh pelanggan dan akan dimasukkan pada setiap <i>increment</i> .
<i>People not process</i>	Keterampilan tim pengembang harus diakui dan dimanfaatkan. Anggota tim dibiarkan untuk bekerja dengan cara mereka sendiri tanpa harus bergantung dengan ketentuan.
<i>Embrace change</i>	Bersiap dengan adanya perubahan kebutuhan sistem sehingga dapat menyediakan desain sistem untuk menyesuaikan perubahan tersebut.

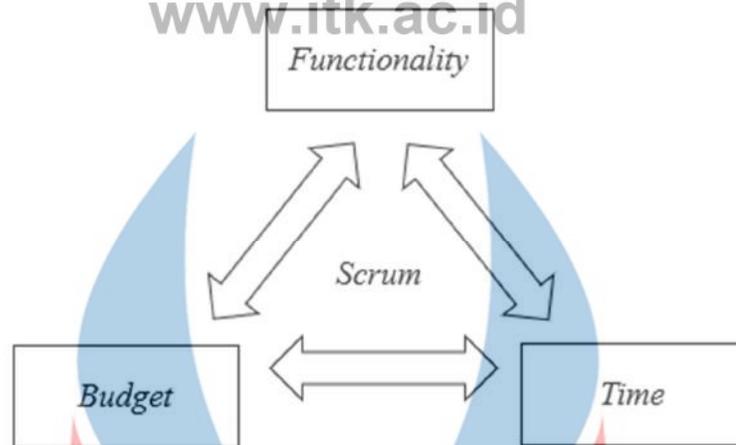
Maintain simplicity

Fokus pada sesuatu yang mudah untuk dikerjakan atau dimengerti, baik itu dalam hal *software* yang sedang dikembangkan maupun dalam proses pengembangan. Apabila memungkinkan, bekerja secara aktif sehingga kerumitan dari sistem dapat dihilangkan.

Dalam *agile*, keseluruhan proyek akan dikembangkan menjadi beberapa bagian tahap pengembangan yang singkat. Hal ini biasa disebut *iterasi* atau *sprint* yang setiap tahapannya dimulai setiap 2 minggu atau 4 minggu. Tujuan utama dari *agile* adalah untuk membentuk individu yang dapat bekerja secara mandiri dan dapat bekerja sama dalam pembuatan produk yang berkualitas (Firdaus, 2017).

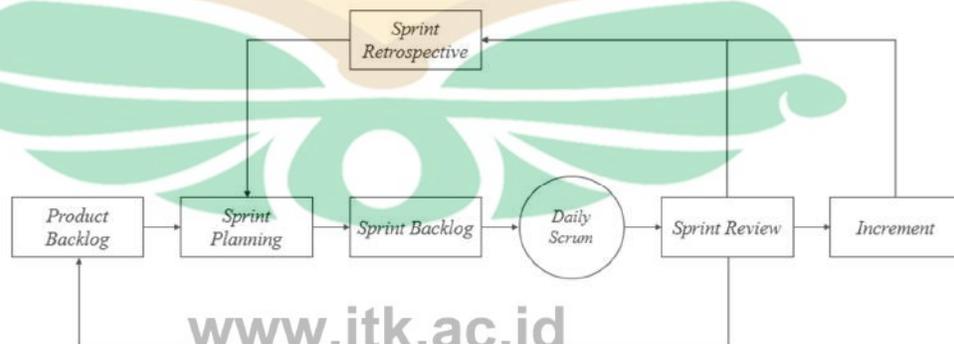
2.3.1 *Scrum*

Scrum merupakan salah satu model atau *framework* pada manajemen pengembangan proyek dari metodologi *agile*. *Scrum* dapat digunakan untuk melakukan pengembangan sistem secara keseluruhan, sebagian, maupun proyek internal atau pelanggan yang bertujuan untuk mengetahui permasalahan yang ada dan menyesuaikan diri sesuai dengan kondisi dari masalah tersebut. Pengembangan *software* yang dilakukan menggunakan *scrum* dilakukan secara singkat. Fokus utama terdapat pada pembinaan sumber daya manusia dengan langkah-langkah *framework* yang akan memudahkan tim untuk bekerjasama dengan individu yang ada dalam tim tersebut. Pada segitiga manajemen proyek *scrum* Gambar 2.1 diisi oleh fungsionalitas, biaya dan waktu. Dimana fungsionalitas lebih menekankan penyelesaian dari sebuah fungsi atau fitur yang di dalamnya sudah terdapat *quality*, *testing*, *documentation*, *review* dan sebagainya. Biaya berupa biaya dalam pelaksanaan proyek dan waktu sebagai batas waktu penyelesaian dari suatu proyek (Firdaus, 2017).



Gambar 2.1 Segitiga Manajemen Proyek Scrum

Pendekatan *scrum* adalah metode *agile* yang *general*, akan tetapi lebih berfokus pada pengembangan yang dilakukan secara berulang, dimana penggunaan praktik atau cara pemrograman tidak diatur di dalam *scrum* (Sommerville, 2011). Scrum merupakan kerangka kerja sederhana yang efektif untuk kolaborasi tim pada *software project*. Terdapat tahapan *scrum framework* yang ditunjukkan pada Gambar 2.2, berupa penyusunan *product backlog* dengan isi berupa fitur-fitur yang akan dikembangkan pada aplikasi. Setelah itu dilakukan *sprint planning* dengan hasil berupa *sprint backlog* yang berisi rencana pengerjaan *sprint*. Kemudian setelah *sprint backlog* selesai disusun, dilakukan *sprint* dengan hasil berupa *product backlog* yang sudah selesai atau yang disebut dengan *increment* (Schwaber & Sutherland, 2017).



Gambar 2.2 *Scrum Framework* (Scrum 2020)

www.itk.ac.id

Penggunaan *scrum* telah banyak digunakan untuk mengembangkan *software*, *hardware*, *embedded software*, *networks of interaction function*, *autonomous vehicles* dan masih banyak lagi yang merupakan pengelolaan operasi organisasi dalam kehidupan sehari-hari.

2.3.2 *Scrum Team*

Scrum team adalah model yang dirancang untuk mengoptimalkan fleksibilitas, kreativitas, dan produktivitas dalam penyelesaian pekerjaan dengan mengatur dirinya sendiri dan lintas fungsional tanpa arahan orang lain yang berada di luar tim. *Scrum team* terdiri dari *product owner*, *development team*, dan *scrum master* (Schwaber & Sutherland, 2017). *Product owner* merupakan bagian dari *scrum team* yang dapat bekerja sama dengan anggota tim lain dan memiliki tanggung jawab untuk memaksimalkan hasil kerja tim dan nilai dari produk yang dikerjakan, kemudian memastikan pekerjaan yang dilakukan berhasil sesuai kebutuhan. *Scrum master* bukanlah orang yang menguasai *scrum*, akan tetapi merupakan sebutan akan suatu peran dalam *framework scrum* yang memiliki tanggung jawab untuk memastikan bahwa tim dapat menyelesaikan hambatan yang ada. Selain itu juga, *scrum master* memastikan komunikasi yang optimal antara *product owner* dengan tim serta memberikan fasilitas yang berhubungan dengan *scrum event* untuk membimbing *development team* agar menghasilkan produk yang memiliki nilai dari setiap kegiatan yang dilakukan. *Development team* adalah tim pengembang yang membuat dan menaikkan nilai produk yang diminta oleh *customer* (Firdaus, 2017).

2.3.3 *Scrum Event*

Scrum telah menentukan *event* yang bertujuan untuk mengurangi kebutuhan pertemuan atau rapat sehingga dapat berfokus pada pengembangan. Semua *event* adalah *time-boxed events*, dimana terdapat batas waktu maksimum dalam pengerjaan *event* tersebut. Setelah *sprint* dimulai, ditetapkan durasi waktu pengerjaan yang tidak dapat dipersingkat ataupun diperpanjang. Berikut ini merupakan penjelasan tiap *event* yang ada dalam *scrum*, meliputi *sprint*, *sprint*

www.itk.ac.id

planning, daily scrum, sprint review dan *sprint retrospective* (Schwaber & Sutherland, 2017): www.itk.ac.id

1. *Sprint*

Sprint adalah inti dari *scrum*, berupa pekerjaan yang dilakukan dengan waktu yang telah ditentukan dengan batas waktu tertentu sampai pekerjaan dapat dikatakan selesai, dapat digunakan dan memiliki potensi untuk dirilis. *Sprint* akan berlanjut ke *sprint* baru lainnya setelah *sprint* sebelumnya selesai. Selama proses *sprint* tidak terdapat perubahan yang akan membahayakan tujuan atau *sprint goal*, selain pada proses ini *scope* dapat dinegosiasikan kembali antara *product owner* dengan tim pengembang karena lebih banyak yang harus dipelajari.

2. *Sprint Planning*

Sprint planning adalah penentuan pekerjaan pada *sprint* yang akan dibuat dan dilakukan oleh tim *scrum*. Pada *sprint planning* akan dibahas hasil yang akan didapatkan pada setiap *sprint* dan bagaimana cara yang akan dilakukan untuk mencapai hasil tersebut.

3. *Daily Scrum*

Daily Scrum adalah *event* untuk tim pengembang yang biasanya dilakukan batas waktu 15 menit pada setiap hari selama dilakukannya *sprint*. Hal ini dilakukan untuk mengoptimalkan kerjasama dan kinerja tim dengan melakukan pemeriksaan pekerjaan sejak dilakukannya *daily scrum* terakhir dan memperkirakan pekerjaan *sprint* yang akan datang. Untuk mengurangi kesulitan, *daily scrum* diadakan pada waktu dan tempat yang sama setiap hari.

4. *Sprint Review*

Sprint review dilakukan di akhir *sprint* untuk memeriksa *increment* dan melakukan penyesuaian *product backlog* jika diperlukan. Tim *scrum* dan *stakeholder* membahas terkait apa saja hal yang dilakukan selama *sprint*. Kemudian setiap perubahan yang ada pada *product backlog* selama melakukan *sprint* akan dioptimalkan agar produk sesuai dengan keinginan. Di tahap ini juga dilakukan presentasi terkait *increment* untuk mendapatkan *feedback*.

5. *Sprint Retrospective*

Sprint retrospective adalah sebuah kesempatan yang dilakukan tim *scrum* untuk memeriksa atau mengevaluasi diri sendiri dan membuat rencana perbaikan yang

akan dilakukan pada *sprint* selanjutnya. Pada dasarnya *sprint retrospective* dilakukan setelah *sprint review* berakhir dan sebelum *sprint planning* selanjutnya dimulai.

2.3.4 *Scrum Artifact*

Gambaran untuk mewakili pekerjaan atau nilai di dalam *scrum* adalah dengan menggunakan *scrum artifact*, dimana *scrum artifact* ini memiliki tujuan untuk menyediakan transparansi dan kesempatan untuk melakukan inspeksi dan juga adaptasi. Terdapat komponen-komponen yang ada di dalam *scrum artifact* dengan penjelasan sebagai berikut (Schwaber & Sutherland, 2017):

1. *Product Backlog*

Product backlog adalah daftar kebutuhan yang telah diketahui dan dibutuhkan dalam produk. *Product backlog* menjadi satu-satunya sumber kebutuhan untuk setiap perubahan yang akan dilakukan pada produk. Seluruh fitur, fungsi, kebutuhan, peningkatan dan perbaikan yang merupakan perubahan yang harus dilakukan pada produk tercantum dalam *product backlog*. Setiap *product backlog* akan diberikan skala prioritas yang akan digabungkan menjadi beberapa *sprint backlog*.

2. *Sprint Backlog*

Sprint backlog adalah kumpulan item pada *product backlog* yang dipilih untuk *sprint* ditambahkan dengan perencanaan untuk mengirimkan *increment* dan mewujudkan tujuan *sprint*. *Sprint backlog* berisi perkiraan yang dilakukan oleh tim pengembang terkait fungsionalitas yang akan ada di *increment* berikutnya dan pekerjaan yang diperlukan untuk menyelesaikan *increment* tersebut.

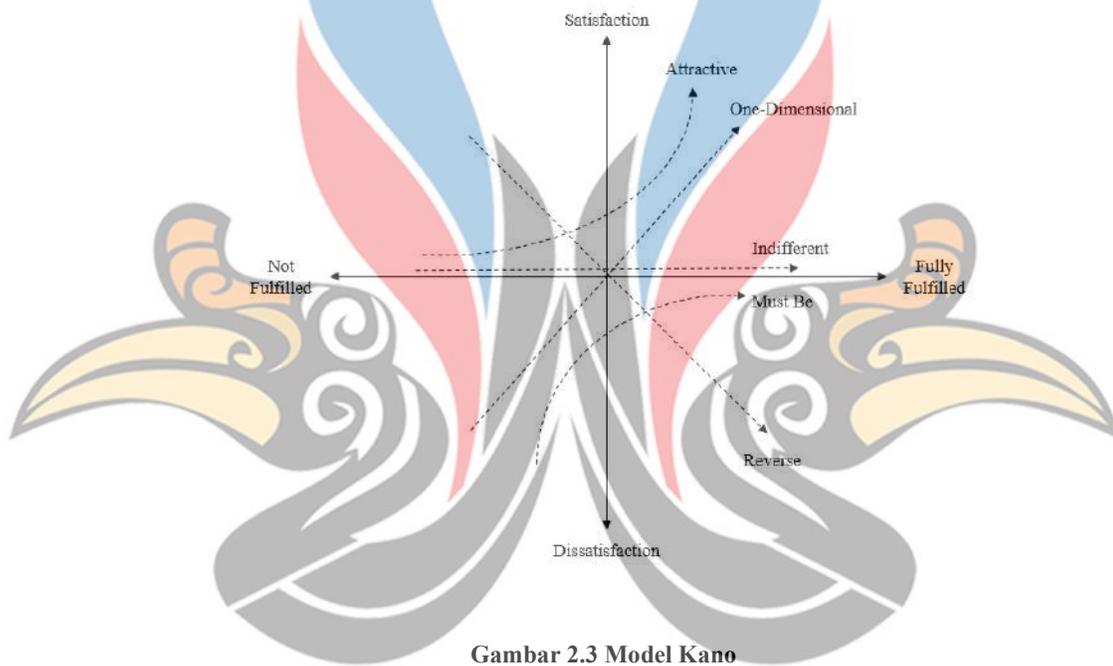
3. *Increment*

Increment adalah keseluruhan item *product backlog* yang diselesaikan selama *sprint* dan keseluruhan *sprint* sebelumnya. Di akhir *sprint*, *increment* harus selesai dan dalam kondisi yang bisa digunakan.

2.4 Metode Kano

Sebuah sistem yang baik memerlukan pertimbangan yang tidak hanya mengunggulkan segi teknologi, akan tetapi juga memerlukan pertimbangan

kepuasan dari sisi pengguna terhadap fungsionalitas sistem. Salah satu metode yang digunakan untuk mengukur kepuasan pengguna terhadap sistem yang dibuat adalah dengan menggunakan metode kano. Terdapat 2 (dua) aspek yang dikemukakan untuk tiap atribut kualitas, yaitu aspek objektif dan aspek subjektif. Aspek objektif melibatkan pemenuhan kualitas, sementara aspek subjektif melibatkan persepsi konsumen atau kepuasan. Dari kedua aspek tersebut dituangkan ke dalam suatu model yang dikenal sebagai metode kano. Gambar dari metode kano dapat dilihat pada Gambar (Rahmayuni, Humaira, & Defni, 2016).



Gambar 2.3 Model Kano

Kategori sebuah produk dalam metode kano dibedakan menjadi *must-be* atau *basic needs* atau *threshold*, *one dimensional* atau *performance needs* atau *linear*, *attractive* atau *excitement needs* atau *delighters*, serta *reverse*. Pembahasan kategori-kategori tersebut adalah sebagai berikut:

1. *Must-be*, pelanggan merasa tidak puas jika kinerja atribut yang bersangkutan rendah.
2. *Linear*, tingkat kepuasan pelanggan akan tinggi dikarenakan kinerja atribut yang tinggi.

3. *Delighters*, meningkatnya kinerja atribut akan meningkatkan tingkat kepuasan pelanggan, tetapi penurunan kinerja atribut tidak akan menurunkan tingkat kepuasan.
4. *Reverse*, tingkat kepuasan pelanggan berbanding terbalik dengan hasil kinerja atribut.

2.5 *Framework Laravel*

Salah satu *framework* PHP terbaik dikembangkan oleh Taylor Otwell yang dimulai pada April 2011 adalah *framework laravel*. Awal mula pembuatan proyek ini dikarenakan tidak ditemukannya *framework* yang *up-to-date* dengan versi PHP. Karena keterbatasan sumber daya, mengembangkan *framework* yang sudah ada sebelumnya juga bukan suatu ide yang bagus. Akhirnya dengan melihat keterbatasan tersebut dibuatlah *framework* dengan nama *Laravel* (Yudhanto & Prasetyo, 2018).

Framework Laravel lebih menekankan pada kesederhanaan dan fleksibilitas desainnya, sehingga pengguna *framework* ini semakin meningkat dari tahun ke tahun. Hal ini juga disertai dengan kelebihan-kelebihan yang dimiliki oleh *framework Laravel*, yaitu *expressive*, *simple* dan *accessible* dengan penjelasan berikut (Aminudin, 2015):

1. *Expressive*, seorang *programmer* diharapkan akan langsung tau kegunaan dari suatu sintaks yang diperlihatkan oleh sintaks *Laravel* meskipun belum pernah mempelajari sintaks tersebut atau bahkan menggunakannya.
2. *Simple*, dengan adanya *Eloquent ORM* yang melakukan *looping* terhadap variabel akan membuat *Laravel* menjadi begitu simpel dan mudah diakses.
3. *Accessible*, dokumentasi yang dibuat selengkap mungkin oleh *code developer Laravel* setiap kali dilakukan perlisian versi terbarunya.

Konvensi struktur direktori yang digunakan oleh *Laravel* yaitu menggunakan pola MVC (*Model-View-Controller*) dengan skema direktori yang bernama *Model*, *View* dan *Controller*. Pola tersebut dikumpulkan dalam sebuah direktori utama dengan nama *src* atau *app* seperti pada *Code Igniter*. Pada skema tersebut, direktori *Model* digunakan untuk menyimpan *class* PHP yang berhubungan dengan model *database*. Kemudian direktori *View* digunakan untuk menyimpan file-file yang

berhubungan dengan tampilan aplikasi (*interface*). Dan direktori *Controller* digunakan untuk menyimpan *class* PHP yang berhubungan dengan *application logic* (Yudhanto & Prasetyo, 2018).

Dalam *Laravel*, terdapat keterbaruan alat untuk melakukan interaksi dengan *database* yang disebut dengan *migration*. Dengan menggunakan *migration*, pengembang dapat melakukan modifikasi sebuah *database* pada sebuah platform secara independen dengan mudah, karena implementasi skema *database* direpresentasikan dalam sebuah *class*. *Migration* dapat berjalan pada beberapa *database* yang telah didukung oleh *Laravel*, seperti MySQL, SQLITE dan MSSQL. Untuk implementasi *active record* menggunakan standar modern OOP yang disebut dengan *Eloquent*. Dalam *Laravel* juga terdapat *artisan*, sebuah *command line interface* yang memungkinkan pengembang untuk berinteraksi dengan aplikasi untuk melakukan sebuah aksi seperti *migration*, *testing*, atau bahkan membuat *controller* dan *model* (Luthfi, 2017).

2.6 Penelitian Terdahulu

Pada penelitian terdahulu terdapat hasil dari penelitian yang pernah dilakukan oleh peneliti sebelumnya, hasil penelitian tersebut dapat digunakan sebagai referensi dan acuan pengembangan sistem berdasarkan dengan metode dan sistem yang dibangun. Terdapat rangkuman hasil dari penelitian terdahulu pada Tabel 2.3 yang menjelaskan tahun publikasi, nama peneliti, judul penelitian, permasalahan, saran penyelesaian, metode dan hasil yang diperoleh dari penelitian tersebut.

Tabel 2.3 Penelitian terdahulu

No	Tahun	Peneliti	Penelitian yang Dilakukan
1	2016	Novia Nurdiana P	Permasalahan: Perlunya implementasi IT di dunia pendidikan sebagai penunjang perkembangan siswa di lingkungan sekolah dan transparansi perilaku pihak sekolah dan orang tua Saran Penyelesaian: Adaptasi peranan IT di dunia pendidikan dengan perancangan sistem yang membantu pihak guru dan orang tua mengetahui perilaku siswa di sekolah

No	Tahun	Peneliti	Penelitian yang Dilakukan
			<p>Metode: <i>Extreme Programming</i></p> <p>Hasil: Sistem Monitoring Pelanggaran Siswa</p>
2	2017	Afrian Firdaus, Dwi Rosa Indah dan Idris	<p>Permasalahan: Pengkoordinasian dalam melakukan monitoring dan evaluasi bidikmisi membutuhkan waktu yang lama. Selain itu terdapat kekurangan akurasi data dan pengarsipan yang kurang baik</p> <p>Saran Penyelesaian: Pengembangan sistem monitoring dan evaluasi berbasis web yang dapat mengatasi kendala dan kekurangan yang terjadi</p> <p>Metode: <i>Scrum</i></p> <p>Hasil: Sistem Informasi Monitoring Mahasiswa Bidikmisi</p>
3	2018	Duwi Paryanto	<p>Permasalahan: Penggunaan sistem yang masih manual dalam manajemen pelanggaran siswa sehingga sering terdapat kesalahan pada informasi. Selain itu informasi tidak akurat dan <i>up-to-date</i>. Masalah lainnya yaitu pencarian data yang membutuhkan waktu yang sangat lama.</p> <p>Saran Penyelesaian: Perancangan sebuah sistem monitoring untuk mencatat pelanggaran siswa.</p> <p>Metode: <i>Object Oriented Analysis and Design</i></p> <p>Hasil: Sistem Monitoring Pelanggaran Siswa</p>
4	2018	Muhammad Amzulian Rinaldy dan Lusi Fajarita	<p>Permasalahan: Proses pemberian poin pelanggaran dan bimbingan konseling yang tidak terdokumentasi dan masih dilakukan secara manual sehingga penyediaan informasi terkait poin pelanggaran dan bimbingan konseling menjadi terlambat</p>

No	Tahun	Peneliti	Penelitian yang Dilakukan
			<p>Saran Penyelesaian: Analisa dan perancangan sistem informasi terkait poin pelanggaran siswa dan bimbingan konseling</p> <p>Metode: <i>Object Oriented Analysis and Design</i></p> <p>Hasil: Sistem Informasi Poin Pelanggaran dan Bimbingan Konseling</p>
5	2018	Sri Indah Darmawanti dan Handrie Noprisson	<p>Permasalahan: Belum memiliki sistem informasi berbasis web, yang mengakibatkan proses penyampaian informasi tidak berjalan dengan maksimal dan kurangnya komunikasi luar akademik antar guru dan murid</p> <p>Saran Penyelesaian: Perancangan dalam hal sistem informasi akademik</p> <p>Metode: <i>Waterfall</i></p> <p>Hasil: Aplikasi Monitoring Pelanggaran Siswa (MONPELSIS)</p>
6	2018	Suaidah dan Irvan Sidni	<p>Permasalahan: Kurangnya pemanfaatan teknologi sehingga sekolah mengalami kemunduran sarana dan prasarana, sehingga harus dilakukan perubahan agar dapat memanfaatkannya dengan baik</p> <p>Saran Penyelesaian: Perancangan sistem monitoring yang mempermudah orang tua mengawasi prestasi siswa</p> <p>Metode: <i>Waterfall</i></p> <p>Hasil: Sistem Informasi Monitoring Prestasi Akademik dan Aktivitas Siswa</p>
7	2019	Abdul Rasyid Sabirin, La Raufun dan Amrol	<p>Permasalahan: Belum ada aplikasi khusus untuk mengelola dan memonitoring data kehadiran siswa serta pengolahan data terkait pelanggaran siswa di sekolah</p>

No	Tahun	Peneliti	Penelitian yang Dilakukan
			<p>Saran Penyelesaian: Merancang dan membuat sistem informasi kehadiran dan pelanggaran siswa</p> <p>Metode: <i>Waterfall</i></p> <p>Hasil: Sistem Informasi Kehadiran dan Pelanggaran Siswa</p>
8	2019	Abdul Rokhman	<p>Permasalahan: Tuntutan adanya pelayanan yang prima untuk orang tua santri</p> <p>Saran Penyelesaian: Pengembangan sistem informasi monitoring santri</p> <p>Metode: <i>Action Research</i></p> <p>Hasil: Sistem Informasi Monitoring Santri</p>
9	2019	Aniria Dawolo, Zulafwan dan Haris Tri Saputra	<p>Permasalahan: Banyaknya faktor-faktor yang menyebabkan tidak terpantaunya pelanggaran yang dilakukan siswa</p> <p>Saran Penyelesaian: Membangun sebuah aplikasi yang berfungsi untuk memantau pelanggaran yang dilakukan oleh siswa</p> <p>Metode: <i>Waterfall</i></p> <p>Hasil: Sistem Monitoring Pelanggaran Siswa</p>
10	2019	Tofik Hidayat dan Sekreningsih Nita	<p>Permasalahan: Terdapat kendala dalam pengolahan data prestasi siswa, dimana pengolahan tersebut masih menggunakan cara yang manual dengan cara mencatat prestasi siswa dalam buku</p> <p>Saran Penyelesaian: Perancangan aplikasi pengolahan data prestasi siswa yang akurat dan terstruktur</p> <p>Metode: <i>Waterfall</i></p>

No	Tahun	Peneliti	Penelitian yang Dilakukan
			Hasil: Aplikasi Informasi Monitoring Prestasi Siswa

Tabel 2.3 merupakan penelitian terdahulu yang digunakan sebagai acuan dan referensi pengembangan sistem berdasarkan pada metode dan sistem yang dibangun, dan digunakan untuk membantu dalam melakukan penelitian.

