

## BAB 2 LANDASAN TEORI

### 2.1 Penelitian Terdahulu

Penelitian terdahulu memuat hasil karya yang dijadikan sebagai acuan oleh penulis dalam melakukan penelitian ini. Kegunaan penelitian terdahulu untuk mengumpulkan informasi yang berhubungan dengan materi sejenis yang berkaitan dengan tugas akhir penulis.

Rujukan pertama adalah penelitian yang dilakukan oleh (Sebri Hesinto, & Iis Sundari, 2019) mengenai Perancangan Aplikasi Administrasi Keuangan Sekolah Di Pondok Pesantren Modern Darussalam Menggunakan Metode *Waterfall*. Pesantren Modern Darussalam adalah institusi pendidikan agama berbasis pondok maupun non pondok yang mengatur segala hal yang berkaitan dengan penyelenggaraan Pendidikan formal maupun non formal dan mencakup administrasi keuangan didalamnya. Pada saat ini proses Pengelolaan administrasi keuangan sekolah di Pondok Pesantren Modern Darussalam dalam melakukan pembayaran keuangan sekolah masih menggunakan sistem yang belum terkomputerisasi dengan melakukan pencatatan pada buku, penyimpanan yang masih disimpan dalam lemari penyimpanan arsip, serta keamanan dari suatu data keuangan yang mengakibatkan laporan yang akan dibuat membutuhkan waktu sehingga proses kerja menjadi lambat serta sering terjadinya kesalahan dalam pembuatan laporan data keuangan dari dua sekolah yaitu Madrasah Tsanawiyah (MTS) dan Madrasah Aliyah (MA) Darussalam yang berada dalam kompleks Pondok Pesantren Modern Darussalam Kota Prabumulih. Penelitian ini dibuat dengan tujuan meningkatkan kinerja pengolahan data administrasi keuangan pesantren. Berdasarkan hasil pengujian dan perhitungan disimpulkan bahwa dengan adanya aplikasi administrasi keuangan sekolah ini dapat memberikan



kemudahan dan mendukung kegiatan administrasi keuangan sekolah.

Berikutnya adalah penelitian yang dilakukan oleh (Deni Prayoko Wijiyanto, Agung Riyantomo, & Nugroho Eko Budiyanto, 2018) mengenai rancang bangun sistem informasi administrasi keuangan berbasis web pada SMK NU Ungaran. SMK NU Ungaran merupakan salah satu SMK Rujukan di Jawa Tengah dari tahun 2016, SMK NU Ungaran ingin meningkatkan mutu pelayanan mereka, salah satunya pelayanan dalam administrasi keuangan. Suatu lembaga pendidikan memerlukan pengelolaan manajemen yang baik untuk melayani kebutuhan peserta didik. Untuk itu proses pendataan peserta didik baru yang benar serta administrasi keuangan peserta didik sangatlah berperan di lembaga pendidikan tersebut. Dalam hal ini, aplikasi berbasis web adalah pilihan yang tepat. Tujuan perancangan sistem informasi administrasi keuangan berbasis web pada SMK NU Ungaran adalah agar terbangunnya sistem informasi administrasi keuangan berbasis web pada SMK NU Ungaran yang nantinya dapat digunakan dalam membantu menyelesaikan permasalahan administrasi keuangan khususnya dalam hal penagihan yang sifatnya berkala juga penyediaan layanan informasi keuangan yang lebih baik kepada peserta didik.

Selanjutnya, penelitian yang dilakukan oleh (Imam Hanafi, dan Ema Utami, 2017) mengenai analisis dan perancangan sistem administrasi keuangan berbasis web pada pondok pesantren al-mujtama' pamekasan madura. Pondok pesantren adalah salah satu institusi pendidikan non formal yang bertujuan menghasilkan generasi-generasi berakhlak karimah, Kelancaran admisantrasi menjadi salah satu faktor pendukung kelancaran kegiatan belajar mengajar. Informasi yang cepat dan akurat dibutuhkan untuk pengambilan keputusan. Salah satu solusinya adalah pemanfaatan teknologi komputer yang dapat mengurangi human error, mempercepat proses serta dapat mempermudah pekerjaan manusia. Sistem Admisantrasi keuangan di Pondok pesantren Al-mujatama' saat



ini masih manual. Kwitansi dibuat dengan tulisan tangan sehingga mengakibatkan kwitansi tersebut tidak dapat dicetak ulang, selain membuat kwitansi, karyawan mencatat pembayaran di komputer lebih tepatnya di Microsoft Excel, hal ini tentu membuat karyawan bekerja dua kali dan sering terjadi antrian dalam proses pembayaran, Hal ini tentu saja sangat berpotensi terjadinya kesalahan manusia dan tidak efisiensi. Dengan demikian, dibutuhkan suatu sistem informasi akan mengurangi terjadinya kesalahan, meringankan pekerjaan dan memberikan informasi yang akurat, hal ini tentu menjadi sebuah solusi bagi Pondok Pesantren Al-mujtama'.

Kemudian, penelitian yang dilakukan oleh (Ahmad Romdhoni, 2020) mengenai Sistem informasi Pembayaran Asrama Pondok Pesantren Darul Ulum Menggunakan Model Scrum. Permasalahan yang dihadapi oleh staf keuangan pondok pesantren Darul Ulum adalah kurangnya pemahaman teknologi informasi untuk kepentingan administrasi yang efisien dan terkomputerisasi sehingga mengakibatkan terjadinya masalah – masalah yang disebabkan oleh human error. Oleh karena itu, Sistem informasi Pembayaran Asrama Pondok Pesantren Darul Ulum Menggunakan Model Scrum ini diharapkan dapat menghasilkan output informasi berupa data santri, data pembayaran santri dan rekapan laporan pembayaran santri.

Terakhir penelitian yang dilakukan oleh (Nurhayyan Jahansyah, 2016) mengenai rancang bangun sistem informasi keuangan dengan metode incremental di asrama pondok tinggi pondok pesantren darul ulum jombang. Pondok Pesantren Darul 'Ulum (PPDU) Jombang adalah salah satu pondok pesantren yang berada di kota Jombang yang terletak di Rejoso, Peterongan, Jombang. Perkembangan zaman menuntut adanya sebuah sarana untuk mempercepat kinerja dan efisiensi dalam administrasi asrama. Untuk itu diperlukan adanya suatu sistem informasi keuangan yang dapat mencatat, memproses dan menghasilkan sebuah sistem informasi keuangan yang cepat dan aman. Sehingga dapat menyajikan



informasi tentang keuangan asrama mulai dari pemasukan, pengeluaran, pembayaran santri dan belanja rutin, hingga informasi tentang saldo yang dimiliki asrama. Luaran dari system ini menghasilkan output berupa informasi dari laporan keuangan dan laporan data transaksi.

Tabel 2.1 Penelitian Terdahulu

NO	Nama Peneliti dan Judul	Plat Form	Metode	Output
1.	Sebri Hesinto, & Iis Sundari (2019) Perancangan Aplikasi Administrasi Keuangan Sekolah Di Pondok Pesantren Modern Darussalam Menggunakan Metode Waterfall	Mobile	<ul style="list-style-type: none"> <li>• Metode dalam penelitian ini menggunakan metode <i>Waterfall</i></li> <li>• Menggunakan Aplikasi berbasis Mobile</li> </ul>	Aplikasi dapat menghasilkan output informasi berupa penyediaan layanan informasi keuangan yang lebih baik kepada peserta didik, terkait laporan keuangan, bukti pembayaran, dan laporan data santri.
2.	Deni Prayoko Wijiyanto, Agung Riyantomo, &	Web	<ul style="list-style-type: none"> <li>• Pembahasan studi kasus tentang pembaya</li> </ul>	Aplikasi dapat menghasilkan output berupa penagihan yang sifatnya berkala seperti

NO	Nama Peneliti dan Judul	Plat Form	Metode	Output
	Nugroho Eko Budiyanto (2018) Rancang Bangun Sistem Informasi Administrasi Keuangan Berbasis Web Pada SMK NU Ungaran		<p>ran SPP dan incidental setiap peserta didik.</p> <ul style="list-style-type: none"> <li>• Metode penelitian yang digunakan adalah pendekatan sistem SLC (System Life Cycle).</li> </ul>	SPP dan penyediaan layanan informasi keuangan yang lebih baik kepada peserta didik.
3.	Imam Hanafi, dan Ema Utami (2017) Analisis Dan Perancangan Sistem Administrasi Keuangan Berbasis Web Pada Pondok Pesantren	Web	<ul style="list-style-type: none"> <li>• Metode dalam penelitian menggunakan metode <i>PIECES</i></li> <li>• Pembahasan studi kasus tentang transaksi pembayaran dan</li> </ul>	Aplikasi dapat menghasilkan output berupa sebuah layanan terhadap santri dan wali santri dalam menyajikan Informasi Pembayaran dan Simpan pinjam yang lebih efisien.

NO	Nama Peneliti dan Judul	Plat Form	Metode	Output
	Al-Mujtama' Pamekasan Madura		simpan pinjam. • Aktor sistem yakni Admin dan Petugas saja	
4.	Ahmad Romdhoni (2020) Sistem informasi Pembayaran Asrama Pondok Pesantren Darul Ulum Menggunkaan Model Scrum.	Web	• Metode penelitian yang digunakan adalah metode <i>Scrum</i> • Pembahasan studi kasus tentang transaksi pembayaran asrama dan notifikasi melalui email terkait tagihan yang aktif.	Aplikasi dapat menghasilkan output berupa efektivitas pelayanan kepada wali santri dan transparansi terkait jumlah transaksi dan tagihan yang aktif, notifikasi melalui email.

NO	Nama Peneliti dan Judul	Plat Form	Metode	Output
5.	Nurhayyan Jahansyah , (2016) Rancang Bangun Sistem Informasi Keuangan Dengan Metode Incremental Di Asrama Pondok Tinggi Pondok Pesantren Darul Ulum Jombang	Web	<ul style="list-style-type: none"> <li>• Metode penelitian yang digunakan adalah metode <i>Incremental</i></li> <li>• Tempat studi kasus berada di Asrama Pondok Tinggi Pondok Pesantren Darul Ulum Jombang.</li> </ul>	Aplikasi dapat menghasilkan output berupa informasi tentang keuangan asrama mulai dari pemasukan, pengeluaran, pembayaran santri dan belanja rutin, hingga informasi tentang saldo yang dimiliki asrama.

Berdasarkan penelitian terdahulu di atas sistem ini belum sesuai dengan apa yang akan dibuat oleh penulis yaitu sistem informasi Administrasi Keuangan Pondok Pesantren Menggunakan Metode SCRUM, yang nantinya diharapkan pembina santri dapat mengelola administrasi keuangan dengan mudah dan pengasuh juga dapat memantau perputaran uang dengan mudah.

## 2.2 Kajian Pustaka

### 2.2.1 Sistem Informasi

Seperti telah diketahui bersama bahwa dewasa ini telah banyak para ahli mendefinisikan mengenai sistem. Beberapa pengertian sistem yang diambil dalam penelitian kali ini adalah pengertian yang diberikan oleh Sutabri. Sutabri memberikan pengertian sistem sebagai sekelompok unsur-unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Sutabri, 2012).

Sutabri mendefinisikan informasi sebagai data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

Sistem informasi merupakan kombinasi teratur dari orang-orang, perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi (Yakub, 2012).

Dari penjelasan yang diutarakan oleh Sutabri, peneliti menyimpulkan bahwa sistem informasi merupakan sekumpulan unsur yang telah dikaji secara teratur yang merupakan hubungan antara manusia dengan perangkat dan jaringan komputer yang digunakan untuk mencapai tujuan tertentu.

### 2.2.2 Administrasi Keuangan

Administrasi dalam arti sempit, yaitu berasal dari kata *Administratie* (bahasa Belanda) yang meliputi kegiatan: catat-mencatat, surat-menyurat, pembukaan ringan, ketik-mengetik, kegiatan menyusun keterangan-keterangan secara sistematis, dan pencatat-pencatatannya secara tertulis untuk didokumentasikan, agar mudah menemukannya bilamana akan dipergunakan lagi, baik secara terpisah-pisah maupun sebagai keseluruhan yang tidak terpisahkan, dan segala sesuatu yang





bersifat teknis ketatausahaan. Jadi pengertian administrasi sebagai ketatausahaan merupakan bagian kecil dari administrasi. (Affifudin, 2010)

Administrasi Keuangan adalah upaya pengelolaan mencakup semua aktivitas yang berhubungan erat dengan semua system keuangan untuk mencapai tujuan tiap perusahaan atau organisasi, Administrasi keuangan arti sempit adalah segala aktivitas yang berhubungan dengan pencatatan dan pemasukan serta pengeluaran untuk membiayai berbagai kegiatan organisasi, dalam bentuk berupa tata usaha atau tata pembukuan keuangan, Administrasi keuangan arti luas adalah kebijakan mengenai pengadaan serta penggunaan keuangan organisasi dalam mewujudkan kegiatan organisasi itu sendiri, dimana bentuknya berupa pengolahan keuangan meliputi perencanaan dan pengaturan serta pertanggung jawaban, dan pengawasan keuangan. Keuangan adalah semua hak milik organisasi, lembaga atau instansi yang dapat dinilai dengan uang, termasuk didalamnya barang yang dapat dinilai dengan uang dan dapat dijadikan milik organisasi. Sedangkan uang adalah alat pembayaran Negara. Manajemen keuangan merupakan salah satu substansi manajemen sekolah yang akan turut menentukan berjalannya kegiatan pendidikan di sekolah. Sebagaimana yang terjadi di substansi manajemen pendidikan pada umumnya, kegiatan manajemen keuangan dilakukan melalui proses perencanaan, pengorganisasian, pengarahan, pengkoordinasian, pengawasan atau pengendalian. (Maxmanroe, 2019)

### 2.2.3 Santri

Seperti yang telah kita ketahui selama ini, bahwa santri adalah orang yang mendalami ilmu keagamaan. Sementara itu, A. H. John menyebutkan bahwa istilah santri berasal dari Bahasa *Tamil* yang berarti guru mengaji (Suharto, 2011).



Nurcholish Madjid juga memiliki pendapat berbeda. Dalam pandangannya asal usul kata “Santri” dapat dilihat dari dua pendapat. Pertama, pendapat yang mengatakan bahwa “Santri” berasal dari kata “*sastr*”, sebuah kata dari bahasa *Sansekerta* yang artinya melek huruf. Pendapat ini menurut Nurcholish Madjid didasarkan atas kaum santri kelas *literary* bagi orang Jawa yang berusaha mendalami agama melalui kitab-kitab bertulisan dan berbahasa Arab. Kedua, pendapat yang mengatakan bahwa perkataan santri sesungguhnya berasal dari bahasa Jawa, dari kata “*cantrik*” berarti seseorang yang selalu mengikuti seorang guru ke mana guru ini pergi menetap (Yasmadi, 2015).

Dari beberapa pendapat di atas, dapat disimpulkan bahwa santri merupakan peserta didik yang menempuh dan bertempat tinggal dalam sebuah lembaga pendidikan Islam yang dididik oleh seorang Kyai sebagai pembersihan diri untuk menuju manusia yang religius.

#### 2.2.4 Pondok Pesantren

Menurut pendapat para ilmuwan, istilah pondok pesantren adalah merupakan dua istilah yang mengandung satu arti. Orang Jawa menyebutnya “pondok” atau “pesantren”. Sering pula menyebut sebagai pondok pesantren. Istilah pondok barangkali berasal dari pengertian asrama-asrama para santri yang disebut pondok atau tempat tinggal yang terbuat dari bambu atau barangkali berasal dari bahasa Arab “*funduq*” artinya asrama besar yang disediakan untuk persinggahan.

Jadi pesantren secara etimologi berasal dari kata *santri* yang mendapat awalan *pe-* dan akhiran *-an* sehingga menjadi *pe-santria-an* yang bermakna kata “*shastri*” yang artinya murid. Sedang C.C. Berg. berpendapat bahwa istilah *pesantren* berasal dari kata *shastri* yang dalam bahasa India berarti orang yang tahu buku-buku suci agama Hindu, atau seorang sarjana ahli kitab-



kitab suci agama Hindu. Kata *shastri* berasal dari kata *shastra* yang berarti buku-buku suci, buku-buku suci agama atau buku-buku tentang ilmu pengetahuan (Yasmadi, 2015).

M. Dawam Rahardjo memberikan pengertian pesantren sebagai sebuah lembaga pendidikan dan penyiaran agama Islam, itulah identitas pesantren pada awal perkembangannya. Sekarang setelah terjadi banyak perubahan di masyarakat, sebagai akibat pengaruhnya, definisi di atas tidak lagi memadai, walaupun pada intinya nanti pesantren tetap berada pada fungsinya yang asli, yang selalu dipelihara di tengah-tengah perubahan yang deras. Bahkan karena menyadari arus perubahan yang kerap kali tak terkendali itulah, pihak luar justru melihat keunikannya sebagai wilayah sosial yang mengandung kekuatan resistensi terhadap dampak modernisasi (Dhofier, 2015).

Dari penjelasan di atas peneliti menyimpulkan bahwa pondok pesantren adalah suatu lembaga pendidikan Islam yang dibimbing oleh seorang kiai atau pengasuh, diperuntukkan untuk membentuk manusia menjadi pribadi yang berlandaskan syariat Islam dan mandiri dengan segala macam aturan atau tata tertib yang ada.

### **2.2.5 Asrama Al-Furqon**

Asrama Al-Furqon merupakan salah satu asrama Pondok Pesantren Darul Ulum (PPDU). Asrama Al-Furqon merupakan asrama ke-27 yang berdiri sekitar tahun 2012 an. Didirikan oleh Putra ketujuh KH. A. Dimiyathi Romly, yaitu H. M. Musta'in Dzul Azmi Dimiyati, S.Sos. Asrama Al-Furqon ikut serta dalam mengembangkan dan mendidik sumber daya manusia, yang bertujuan untuk mengembangkan insan kamil yang dapat menguasai dan memiliki iman, Taqwa, IPTEK dan akhlakul karimah. Hal ini sesuai dengan moto asas pondok pesantren Darul Ulum: Berasas manfaat, berpikir cepat, bertindak tepat dan berzikir kuat. Asrama Al-Furqon dalam



pendidikan menerapkan pola terpadu yaitu kekuatan salaf, kekuatan formal dan IPTEK dalam upaya menciptakan kader bangsa yang mampu menghadapi tantangan dan kebutuhan masa depan bangsa. Memaksimalkan potensi sumber daya manusia yang profesional, mandiri dan berkepribadian Islam (PPDU, 2017).

### **2.2.6 Laravel**

Menurut Pribadi (2016), Laravel adalah sebuah framework PHP yang dirilis di bawah lisensi MIT: dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif: jelas dan menghemat waktu

Menurut (Wardana: 2014): Model View Controller (MVC) adalah sebuah pola pemrograman yang bertujuan memisahkan logika bisnis: logika data dan logika tampilan (interface). Laravel menggunakan konsep MVC: yang mana anda harus memisahkan kode database ke folder model: kode proses ke folder controller dan kode tampilan ke folder view.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data: controller: dan *user interface*.

Menurut Naista (2017) mengemukakan bahwa framework adalah suatu struktur konseptual dasar digunakan untuk memecahkan atau menangani suatu masalah yang bersifat kompleks. Singkatnya, framework merupakan suatu kerangka kerja dari sebuah website yang akan dibangun. Dengan menggunakan kerangka tersebut, waktu yang perlukan dalam membangun sebuah website menjadi lebih singkat dan memudahkan dalam proses perbaikan.



Sedangkan definisi Laravel menurut Naista (2017) mengatakan bahwa Laravel merupakan salah satu framework berbasis PHP bersifat open source (terbuka), dan menggunakan konsep MVC (model – view – controller). Laravel berada di bawah lisensi MIT License dengan menggunakan Github sebagai tempat berbagi code menjalankannya.

### 2.2.7 Pemrograman Web

.Merupakan gabungan dari dua kata pemrograman dan web. Pemrograman berarti pemrograman dalam bahasa Inggris, yang berarti membuat proses, metode, dan program. Web merupakan kumpulan situs Internet pada jaringan komputer yang menyediakan teks, suara, grafik dan animasi. Halaman web adalah file teks murni dalam format sintaks HTML. (Rerung, 2018).

### 2.2.8 PHP

PHP merupakan bahasa pemrograman yang berupa struktur kode yang terdapat pada HTML, bekerja pada sisi server sehingga kode tersebut tidak nampak pada sisi client. PHP dibuat untuk bekerja menggunakan server database dan dirancang untuk membuat file HTML yang mengakses database begitu mudah atau umumnya dokumen yang dihasilkan adalah dokumen web dinamis, PHP adalah perangkat lunak bebas, yang berarti bebas digunakan tanpa harus membayar lisensi. (Mahendra, 2012). PHP memiliki keunggulan yakni memiliki perkembangan yang pesat, PHP sendiri adalah bahasa pemrograman *open source* sehingga pengguna dapat menggunakan secara bebas dan gratis.

### 2.2.9 MySQL

MySQL atau *My Structure Query Language* merupakan perangkat lunak untuk manajemen basis data.



Adapun perangkat lunak manajemen basis data yaitu Oracle, MS SQL, PostgreSQL dan lainnya (Prayitno & Safitri, 2015). Dapat disimpulkan bahwa MySQL adalah DBMS (*Database Management System*) yang menggunakan perintah SQL (*Structured Query Language*) yang banyak digunakan saat ini dalam pengembangan aplikasi berbasis web. MySQL terbagi menjadi dua lisensi, yang pertama adalah *Free Software* dimana siapapun dapat mengakses software tersebut. Dan yang kedua adalah *Shareware* dimana perangkat lunak berpemilik memiliki keterbatasan dalam penggunaannya. MySQL merupakan penghubung antara perangkat lunak dan server database.

### **2.2.10 Agile Development Method**

*Agile development method* merupakan sekumpulan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan iteratif, persyaratan dan solusinya didapat melalui kolaborasi antar tim yang terorganisir. Fokus *agile development* pada perkembangan yang cepat, perangkat lunak yang dirilis secara bertahap, dan pada masa proses perkembangannya melibatkan *client* secara langsung. Adapun tujuan dari *Agile development method* adalah sistem bernilai tinggi dan juga berfungsi, *Iterative, incremental, evolutionary*, produksi berkualitas tinggi, fleksibel dan manajemen risiko, dan kolaborasi (Mahendra & Eby Yanto, 2018).

Pengertian lain menurut Haryana (2019) *agile development method* merupakan metodologi pengembangan suatu perangkat lunak modern yang berdasarkan pada prinsip-prinsip pengembangan perangkat lunak jangka pendek, *agile development method* memerlukan penyesuaian yang cepat terhadap bentuk perubahan apapun dari pengembangnya. *Agile Manifesto* atau prinsip utama *agile development method* terdiri dari 4 (empat) prinsip:



- a. Interaksi dan personel lebih penting dibandingkan dengan proses dan alat Dalam *agile development method* proses dan alat tidak terlalu penting, dikarenakan proses dan alat yang baik tidak akan berfungsi dengan baik pula apabila tim nya tidak memiliki *skill* untuk menjalankannya. Sehingga pada *agile development method* lebih mementingkan tim dan interaksi antar tim yang baik.
- b. Keberadaan software dalam *agile development method* lebih penting jika dibandingkan dengan dokumentasi yang lengkap dari software, dikarenakan software yang berfungsi dengan baik dan dapat digunakan oleh *user* merupakan salah satu tujuan dari *agile development method*.
- c. Kolaborasi dengan client lebih penting daripada negosiasi kontrak Melibatkan client untuk memantau pembuatan software guna meminimalisir kesalahpahaman fitur yang diinginkan oleh client, sehingga software yang dikerjakan sesuai dengan permintaan client.
- d. Respon terhadap perubahan lebih penting dibandingkan dengan mengikuti rencana.

Dalam proses pengembangan suatu perangkat lunak, sudah pasti perubahan-perubahan dapat terjadi dari berbagai factor yaitu dari klient, developer maupun dari lingkungan (mengikuti perkembangan teknologi). Sehingga dalam *agile development method* tim developer diharuskan dapat beradaptasi dengan cepat terhadap perubahan-perubahan yang ada.

Agile memiliki kepekaan tinggi terhadap perubahan. Sehingga dalam metodologi Agile merespon semua perubahan yang ada daripada mengikuti rencana yang telah ada. (Kniberg & Bima, 2007) Metodologi Agile memiliki beberapa model yang dikenal cukup luas yaitu :



1. Agile Modelling
2. Extreme Programming (XP)
3. Scrum

Perbedaan pada setiap metode terletak pada praktek atau penerapan yang dilakukan dalam pengembangan perangkat lunak.

### 2.2.11 Metode Scrum

*Scrum* merupakan bagian dari *agile software development*, *Scrum* adalah suatu metode yang digunakan untuk mengembangkan suatu produk yang kompleks. Dalam *Scrum* para *developemya* dapat mengatasi masalah yang kompleks, dan juga diharuskan dapat menghasilkan produk yang bernilai tinggi. Karakteristik dari *framework Scrum* ialah ringan karena *Scrum* dapat mudah diimplementasikan pada berbagai permasalahan, sederhana dalam penerapan *framework*, *Scrum* stabil dan tidak dapat di modifikasi (Haryana, 2020).

Metode Kerangka Kerja Scrum adalah jenis baru dari metodologi Agile yang diharapkan dapat meningkatkan kecepatan, fleksibilitas dalam pengembangan perangkat lunak.(Schwaber & Sutherland, 2017) Scrum merupakan metode Agile yang cukup populer dimana produktifitas menjadi lebih tinggi. Prinsip kerja *Scrum* adalah bekerja secara iteratif dan bertahap hingga mencapai waktu yang telah ditentukan sehingga produk perangkat lunak yang dikembangkan dapat memenuhi kebutuhan yang diinginkan oleh konsumen. Metode Scrum dalam pengembangan perangkat lunak terbagi menjadi beberapa bagian *iteration* dimana setiap *iteration* disebut dengan *Sprint*.

Metode Scrum dimulai dengan mengumpulkan semua kebutuhan pengguna , namun tidak menutup kemungkinan bahwa semua kebutuhan harus berasal dari pengguna disaat awal-awal *Sprint*. Pengguna dapat mengubah kebutuhan kapan pun ketika sedang dalam pengembangan.





Mereka bisa menambahkan fitur baru, menghapus fitur baru , atau memperbaiki fitur baru.

### 2.2.12 Tahapan Scrum

Tahapan Scrum dibuat untuk menciptakan kesinambungan dan mengurangi tahapan lain yang tidak tercantum di dalam Scrum. Tidak adanya pelaksanaan salah satu tahapan ini akan mengurangi transparansi dan menghilangkan kesempatan untuk meninjau dan membuat perubahan.

#### 1. *Sprint*

Sebuah fase pengembangan perangkat lunak yang memiliki batasan waktu maksimal satu bulan dengan durasi yang konsisten sepanjang proses pengembangan produk. *Sprint* baru, langsung dimulai ketika *Sprint* sebelumnya selesai. *Sprint* memuat dan terdiri dari *Sprint Planning*, *Daily Scrum*, *Sprint Review* dan *Sprint Retrospective* (Dingsøyr dkk, 2012)

#### 2. *Sprint Planning*

Pekerjaan yang dilakukan di dalam *Sprint* direncanakan pada *Sprint Planning*. Perencanaan ini dibuat secara kolaboratif oleh seluruh anggota Tim Scrum. *Sprint Planning* dibatasi maksimum delapan jam untuk *Sprint* yang berdurasi satu bulan.

#### 3. *Daily Scrum*

*Daily Scrum* adalah kegiatan dengan batasan waktu maksimum selama 15 menit agar Tim Pengembang dapat mensinkronisasikan pekerjaan mereka dan membuat perencanaan untuk 24 jam kedepan. Hal ini dilakukan dengan meninjau pekerjaan semenjak tahapan *Daily Scrum* terakhir dan memperkirakan pekerjaan yang dapat dilakukan sebelum melakukan *Daily Scrum* berikutnya.

#### 4. *Sprint Review*

*Sprint Review* diadakan di setiap akhir *Sprint* untuk meninjau Inkremen dan merubah *Product Backlog* bila diperlukan. Pada saat *Sprint Review*, Tim Scrum dan



stakeholder berkolaborasi untuk membahas apa yang telah dikerjakan dalam *Sprint* yang baru usai. Berdasarkan hasil tersebut dan semua perubahan *Product Backlog* pada saat *Sprint*, para hadirin berkolaborasi menentukan apa yang dapat dikerjakan di *Sprint* berikutnya, untuk mengoptimalkan nilai produk. Pertemuan ini bersifat informal, bukan merupakan *status meeting*, dan presentasi dari Inkremen diharapkan dapat mengumpulkan masukan dan menumbuhkan semangat kolaborasi.

#### 5. *Sprint Retrospective*

*Sprint Retrospective* adalah sebuah kesempatan bagi Tim Scrum untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya. *Sprint Retrospective* dilangsungkan setelah *Sprint Review* selesai dan sebelum *Sprint Planning* berikutnya. Ini adalah tahapan dengan batasan waktu maksimum selama tiga jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih pendek, batasan waktunya biasanya lebih singkat. *Scrum Master* memastikan bahwa tahapan ini dilaksanakan dan setiap hadirin memahami tujuannya. *Scrum Master* mengedukasi Tim Scrum untuk melaksanakannya dalam batasan waktu yang telah ditentukan. *Scrum Master* berpartisipasi sebagai rekan yang bertanggungjawab terhadap setiap proses yang terjadi ketika Scrum berlangsung.

### 2.2.13 Artefak Scrum

Artefak Scrum merepresentasikan pekerjaan atau nilai, bertujuan untuk memberikan ruang transparansi, dan kesempatan-kesempatan untuk meninjau ulang dan adaptasi akan perubahan yang telah akan terjadi. (Kenett dkk, 2016) Artefak yang didefinisikan oleh Scrum secara khusus dirancang untuk meningkatkan transparansi dari setiap informasi yang ada antara tim pengembang, *product owner* dan Scrum master, dengan begitu semua pihak dapat memiliki pemahaman yang sama terhadap artefak.



### 1. *Product Backlog*

*Product Backlog* adalah daftar terurut semua fitur yang telah diketahui saat ini yang harus ada di dalam produk. *Product Backlog* adalah satu-satunya daftar sumber kebutuhan untuk setiap perubahan yang terjadi didalam produk. Salah satu cara mendapatkan *Product Backlog* berasal dari *User Story* yang telah dibuat sebelumnya. *Product Owner* bertanggung-jawab terhadap *Product Backlog*, termasuk isinya, ketersediaannya, dan urutannya. *Product Backlog* sendiri merupakan sentral dari Scrum.

### 2. *Sprint Backlog*

*Sprint Backlog* adalah sekumpulan item *Product Backlog* yang telah dipilih untuk dikerjakan di *Sprint*, juga di dalamnya rencana untuk mengembangkan potongan tambahan produk dan merealisasikan *Sprint Goal*. *Sprint Backlog* adalah perkiraan mengenai fungsionalitas apa yang akan tersedia di Inkremen selanjutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitas tersebut menjadi potongan tambahan produk yang “Selesai”.

### 3. Inkremen

Inkremen adalah gabungan dari semua item *Product Backlog* yang diselesaikan pada *Sprint* berjalan dan nilai-nilai dari Inkremen *Sprint-Sprint* sebelumnya. Pada akhir *Sprint*, inkremen terbaru harus “Selesai”, yang artinya berada dalam kondisi yang berfungsi penuh dan memenuhi definisi “Selesai” yang dibuat oleh Tim Scrum. Terlepas apakah *Product Owner* akan merilis produknya, produk harus selalu berada dalam kondisi yang berfungsi penuh

### 4. Definisi Selesai

Ketika sebuah item *Product Backlog* atau Inkremen dikatakan “Selesai”, setiap pihak harus mengerti dengan apa yang dimaksud dengan “Selesai”. Walaupun definisi ini berbeda-beda antar tim Scrum, sesama anggota tim harus memiliki pemahaman yang sama mengenai pekerjaan yang harus mereka selesaikan guna memastikan adanya transparansi. Ini adalah definisi selesai untuk Tim Scrum dan ini



digunakan untuk memeriksa apakah pekerjaan untuk mengembangkan Inkremen dianggap selesai.

### 2.2.14 Tim Scrum

Tim Scrum tidak mencakup peran tradisional seperti programmer, desainer, tester atau analisa sistem. Tetapi para *stackholder* yang bekerja sama dalam proyek untuk menyelesaikan pekerjaan mereka secara kolektif dan berkomitmen untuk menyelesaikannya. Terdiri atas *Product Owner*, Tim Pengembang dan *Scrum Master*.

#### 1. *Scrum Master*

Fasilitator yang melayani tim tetapi bukanlah project manager. Memastikan fase metodologi Scrum dilakukan dengan baik, melindungi tim dan memastikan mereka tidak terlalu berkomitmen dengan yang dikerjakan karena perubahan dapat terjadi disetiap *Sprint*.

#### 2. *Product Owner*

Pihak yang bertanggung jawab terhadap suksesnya pengembangan produk dan hasil kerja tim pengembang, biasanya adalah representasi dari konsumen. *Product Owner* satu-satunya orang yang bertanggung jawab mengelola *Product Backlog*.

#### 3. Tim Pengembang

Sekelompok orang yang bertanggung jawab untuk menghasilkan product perangkat lunak. Dibentuk dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaannya secara mandiri. Sinergi yang ada di dalam tim akan meningkatkan efisiensi dan efektifitas dari Tim Pengembang secara keseluruhan.

### 2.2.15 Pilar Scrum

#### 1. Transparansi

Semua variabel yang perlu diketahui dibuat transparansi agar semua orang yang peduli dengan pengembangan



perangkat lunak dapat melihatnya. Transparansi mengharuskan aspek-aspek tersebut didefinisikan dengan standar yang sama, sehingga semua orang yang terlibat memiliki pemahaman yang sama mengenai apa yang sedang ditinjau.

### 2. Inspeksi

Semua variabel yang dibuat transparan tersebut ditinjau setiap waktunya, peninjauan seandainya tidak dilakukan terlampaui sering karena dapat menyebabkan terhambatnya pekerjaan. Paling bermanfaat jika dilakukan secara rutin oleh peninjau yang kompeten pada saat pekerjaan berjalan.

### 3. Adaptasi

Bila variabel yang telah ditinjau ternyata akan menyebabkan hasil akhir yang tidak sesuai dengan harapan, maka perencanaan awal harus diubah.

## 2.2.16 *Unified Modeling Language (UML)*


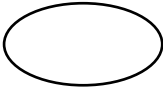



*Unified Modeling Language (UML)* adalah keluarga notasi grafis yang di dukung oleh mitra model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO) (Fowler, 2015).

### A. *Use Case Diagram*

*Use Case Diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). Sehingga pembuatan *use case diagram* lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* menggambarkan sebuah interaksi Antara aktor dengan sistem. (Prabowo, 2011).



Tabel 2.2 Simbol Usecase Diagram


Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	<i>Use Case</i> : Perangkat tertinggi dan fungsionalitas yang dimiliki sistem.
	<i>Association</i> : relasi antar aktor dan <i>Use Case</i>
	<i>Generalisasi</i> : untuk memperlihatkan struktur pewaris yang terjadi.
	<i>Depedency</i> : menggambarkan hubungan dua elemen atau lebih






## B. Class Diagram

*Class* adalah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi) (Prabowo, 2011). Kelas memiliki tiga area pokok:

1. Nama (*Class Name*)
2. Atribut
3. Metode (*Operation*)

Tabel 2.3 Simbol Class Diagram




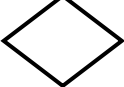

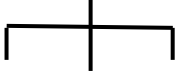

Simbol	Keterangan
	<i>Generalization</i> : hubungan di mana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data

	dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Class</i> , himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i> , Deskripsi dari urutan aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Realization</i> , Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i> , hubungan di mana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i> , menghubungkan Antara objek satu dengan lainnya.

### C. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktivitas yang lainnya, atau dari aktifitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case* (Prabowo, 2011). Notasi yang digunakan dalam *activity diagram* adalah sebagai berikut.



Tabel 2.4 Simbol Activity Diagram

Simbol	Keterangan
	Titik Awal.
	Titik Akhir.
	<i>Activity</i> .
	Pilihan untuk pengambilan keputusan.
	<i>Fork</i> : untuk menunjukkan kegiatan yang dilakukan secara paralel.
	<i>Rake</i> : menunjukkan adanya dekomposisi.
	Aliran akhir ( <i>Flow Final</i> )

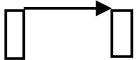


#### D. Sequence Diagram

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Guna untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem (Prabowo, 2011).

Tabel 2.5 Simbol Sequence Diagram

Simbol	Keterangan
	Aktor, menggambarkan pengguna sistem.
	<i>Lifeline</i> , objek antarmuka yang saling berinteraksi



	<i>Message</i> , spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktivitas yang terjadi.
	<i>Message</i> , spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktivitas yang terjadi.
	<i>Self Message</i> , menggambarkan aktivitas pesan pada aktor itu sendiri.

### 2.2.17 Pengujian *Black Box*

*Black Box testing* berfokus pada spesifikasi fungsional dari perangkat lunak, Tester dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. *Blackbox* testing bukanlah solusi alternatif dari *Whitebox*. Testing tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *WhiteBox* Testing (Nidhra & Dondeti, 2012).

*BlacxBox Testing* cenderung untuk menemukan hal hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*)
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*)
5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan pertanyaan berikut:

- 1) Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?

- 2) *Input* seperti apa yang dapat menjadi bahan kasus uji yang baik?
- 3) Apakah sistem sensitif pada *input* tertentu?
- 4) Bagaimana sekumpulan data dapat diisolasi?
- 5) Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?
- 6) Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

