

# **MAKALAH**

## **METODOLOGI AGILE DAN SCRUM DALAM PENGEMBANGAN SISTEM**

“Disusun dalam rangka pemenuhan tugas mata kuliah  
Pengantar Teknologi dan Informasi”



Disusun oleh :

**HARIRI LAIL**

**NIM 20353008**

**PROGRAM STUDI S1 AKUNTANSI**  
**STIE & STMIK JAYAKARTA**  
**TAHUN AKADEMIK 2020/2021**

## DAFTAR ISI

|   |    |
|---|----|
| HALAMAN JUDUL .....   | 1  |
| DAFTAR ISI.....   | 2  |
| BAB I PENDAHULUAN.....  | 3  |
| 1.1 Latar Belakang .....  | 3  |
| 1.2 Tujuan Penulisan.....   | 3  |
| BAB II PEMBAHASAN .....   | 4  |
| 2.1 Konsep Dasar Pengembangan Sistem.....                             | 4  |
| 2.1.1 Pengertian Pengembangan Sistem .....                            | 4  |
| 2.1.2 Tujuan Pengembangan Sistem .....                                | 4  |
| 2.1.3 Prinsip Pengembangan Sistem .....                               | 5  |
| 2.2 Metodologi <i>Agile</i> dalam Pengembangan Sistem.....            | 6  |
| 2.2.1 Tujuan dan Manfaat Metode Pengembangan <i>Agile</i> .....       | 7  |
| 2.2.2 Prinsip Metode Pengembangan <i>Agile</i> .....                  | 8  |
| 2.2.3 Kelebihan dan Kekurangan <i>Agile Development</i> .....         | 9  |
| 2.3. Metodologi Pengembangan <i>Scrum</i> .....                       | 10 |
| 2.3.1 Tahapan atau Aktivitas Pengembangan Metode <i>Scrum</i> .....   | 10 |
| 2.3.2 Pilar dalam Metode Pengembangan <i>Scrum</i> .....              | 13 |
| 2.3.3 Jenis Role dalam Metode Pengembangan <i>Scrum</i> .....         | 14 |
| 2.3.4 Kelebihan dan Kekurangan Metode Pengembangan <i>Scrum</i> ..... | 14 |
| 2.3.5 Contoh Kasus atau Karakteristik Proyek .....                    | 16 |
| BAB III PENUTUP .....   | 17 |
| 3.1 Kesimpulan .....  | 17 |
| DAFTAR PUSTAKA .....  | 18 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan sistem informasi saat ini telah mengalami pertumbuhan yang sangat pesat. Hal ini mengakibatkan timbulnya persaingan yang semakin ketat pada sektor bisnis dan antar organisasi. Oleh karena itu, banyak organisasi yang memulai memanfaatkan sistem informasi dan teknologinya dengan hanya memperhatikan kebutuhan sesaat dan penerapan sistem informasi yang belum terintegrasi dengan baik. Dengan adanya perencanaan strategis sistem informasi yang matang dan tepat, maka akan menghasilkan sebuah aliran informasi yang baik, tepat, dan sesuai dengan kebutuhan perusahaan. Sehingga akan mendukung setiap fungsi bisnis yang ada didalam organisasi perusahaan untuk melaksanakan kegiatannya dan menempatkan perusahaan dalam posisi yang unggul didalam persaingan bisnis yang ada.

Kaitannya dengan pengembangan/perkembangan perangkat lunak saat ini terdapat berbagai metode yang digunakan baik dilakukan secara individu maupun *team work*. Terdapat hubungan erat antara kualitas, biaya, dan waktu dalam pengembangan perangkat lunak. Secara garis besar terdapat enam langkah yang digunakan dalam metodologi pengembangan perangkat lunak diantaranya perencanaan, implementasi, *testing*, dokumentasi, *deployment*, dan *maintenance*.

*Agile Development Methods* adalah kumpulan metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek dimana memerlukan adaptasi yang cepat dari pengembang terhadap perubahan dalam bentuk apapun, dalam penggunaannya *Agile Development* memerlukan inovasi dan tanggung jawab yang baik antara tim pengembang dan klien agar kualitas dari perangkat lunak yang dihasilkan bagus dan seimbang.

### **1.2 Tujuan Penulisan**

Tujuan dari makalah ini adalah untuk memberikan penjelasan mengenai KOnsep Dasar Pengembangan Sistem dan Metodologi serta hal-hal yang berkaitan dengan *Agile Development Methods* serta *Scrum Methods*.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Konsep Dasar Pengembangan Sistem**

##### **2.1.1 Pengertian Pengembangan Sistem**

Pengembangan sistem informasi sering disebut proses pengembangan sistem (*System Development*). Terdapat beberapa definisi mengenai pengembangan sistem informasi diantaranya adalah :

- Aktivitas untuk menghasilkan sistem informasi berbasis computer untuk menyelesaikan permasalahan (*problem*) organisasi atau memanfaatkan kesempatan (*opportunities*) yang timbul.
- Kumpulan kegiatan para analis sistem, perancang, dan pemakai yang mengembangkan dan mengimplementasikan sistem informasi.
- Tahapan kegiatan yang dilakukan selama pembangunan sistem informasi
- Proses merencanakan, mengembangkan, dan mengimplementasikan sistem informasi dengan menggunakan metode, teknik, dan alat bantu pengembangan tertentu.

Pengembangan sistem merupakan penyusunan suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem Informasi secara teknis dapat didefinisikan sebagai sekumpulan komponen yang saling berhubungan, mengumpulkan atau mendapatkan, memproses, menyimpan, dan mendistribusikan informasi untuk menunjang pengambilan keputusan dan pengawasan dalam suatu organisasi. Selain menunjang proses pengambilan keputusan, koordinasi, dan pengawasan, sistem informasi juga dapat membantu manajer dan karyawan menganalisis permasalahan, menggambarkan hal-hal yang rumit, dan menciptakan produk baru.

##### **2.1.2 Tujuan Pengembangan Sistem**

1. *Performance* (kinerja), peningkatan kinerja bertujuan untuk meningkatkan jumlah transaksi dengan waktu yang secepat mungkin.
2. *Information* (informasi), peningkatan kualitas dari informasi tersebut sehingga akan menentukan kebijakan dari organisasi.
3. *Economy*, meningkatkan keuntungan dengan biaya yang minimum.
4. *Control* (pengendalian), digunakan untuk mengontrol atau mendeteksi adanya kesalahan pada suatu sistem.

5. *Efficiency* (efisiensi), pemanfaatan sumber daya semaksimal mungkin.
6. *Service*, peningkatan layanan oleh sebuah sistem.

### 2.1.3 Prinsip Pengembangan Sistem

Prinsip Pengembangan Sistem ada beberapa prinsip yang mempengaruhi pengembangan sistem informasi, yaitu:

1. Sistem yang digunakan adalah untuk manajemen sebuah sistem harus dapat mendukung segala kebutuhan yang diperlukan oleh manajemen.
2. Pemilik dan pengguna sistem harus terlibat dalam pengembangan keterlibatan pemilik pengguna sistem (*system owner* and *user*) adalah keharusan yang mutlak untuk keberhasilan pengembangan sistem.
3. Sistem yang dikembangkan adalah investasi modal yang besar modal yang digunakan untuk mengembangkan sistem informasi tidak sedikit, apalagi dengan digunakannya teknologi mutakhir. Seperti halnya dengan investasi modal lainnya yang dilakukan oleh perusahaan, maka setiap investasi modal harus mempertimbangkan 2 hal: semua alternatif yang ada harus diinvestigasi, investasi yang terbaik harus bernilai.
4. Tentukan tahapan pengembangan, penahapan akan membuat proses pengembangan yang menjadi aktivitas-aktivitas yang lebih kecil lebih mudah dikelola dan diselesaikan.
5. Sistem yang dikembangkan memerlukan orang yang terdidik orang yang terlibat dalam pengembangan maupun penggunaan sistem informasi harus merupakan orang yang terdidik tentang permasalahan-permasalahan yang ada dan terhadap solusi-solusi yang mungkin dilakukan.
6. Proses pengembangan sistem tidak harus urut langkah-langkah harus dilakukan secara bersama-sama
7. Jangan takut membatalkan proyek keraguan untuk terus melanjutkan proyek yang tidak layak lagi karena sudah terserapnya dana ke dalam proyek ini hanya akan membunag dana dengan sia-sia.
8. Dokumentasi, dokumentasi sangat berguna untuk pengembangan sistem berikutnya. Dokumentasi seharusnya dilakukan dari awal pengembangan sistem sampai proses tersebut selesai dilakukan.

## 2.2 Metodologi *Agile* dalam Pengembangan Sistem

***Agile Development*** erat kaitannya dengan metodologi dan proses pengembangan perangkat lunak, metodologi pengembangan perangkat lunak sendiri adalah sebuah metodologi yang digunakan untuk membuat struktur, rencana, dan kontrol pengerjaan suatu proyek sedangkan Proses pengembangan perangkat lunak adalah model-model dan metodologi yang digunakan untuk mengembangkan suatu perangkat lunak.

*Agility for a software Development organization, means the ability to adapt and react quick and effectively and appropriately to changes in its environment and to demands imposed by this environment* (Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J., 2002).

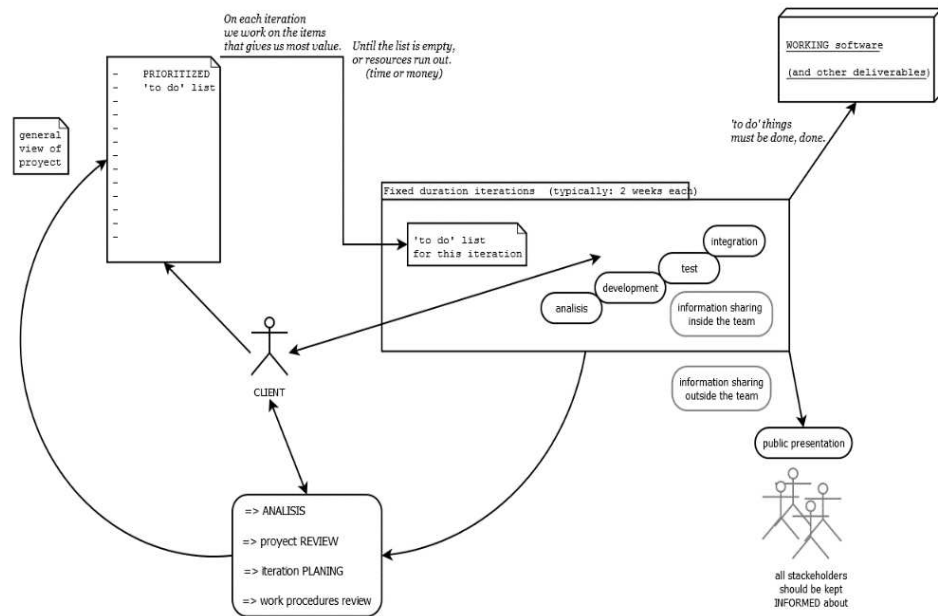
***Agile Development Methods*** secara singkat dapat diartikan sebagai sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun.

*Agility* dalam pengembangan perangkat lunak diartikan sebagai kemampuan untuk beradaptasi dan bereaksi secara cepat dan efektif terhadap perubahan serta meminimalisir dampak terhadap perangkat lunak yang sedang dibangun. Kolaborasi antara tim pengembang dengan klien menjadi hal yang penting, klien dianggap sebagai anggota tim. Fleksibilitas di tawarkan kepada klien kaitannya dengan kesadaran dan pengetahuan pelanggan tentang apa yang dia inginkan.

*Agile Development* juga dapat didefinisikan dalam empat nilai atau sering disebut sebagai *Agile* manifesto, diantaranya :

1. Interaksi dan personal, dibutuhkan interaksi yang baik antara anggota atau personal pengembang karena tanpa adanya hal tersebut pengembangan perangkat lunak tidak akan berjalan dengan baik.
2. Perangkat lunak yang berjalan, hal ini berfungsi pada saat melakukan *Meeting* (demonstrasi) kepada klien. Perangkat lunak yang masih dalam proses akan lebih berguna dari sekedar dokumentasi
3. Kolaborasi dengan klien, ini merupakan ciri dari *Agile Development* dimana mengutamakan keterlibatan dan kolaborasi dari klien

4. Respon terhadap perubahan, sesuai dengan namanya *Agile Development* berfokus terhadap kecepatan respon tim ketika klien menginginkan perubahan saat proses pembuatan perangkat lunak



Gambar 1.1 Diagram *Agile Development Method*

### 2.2.1 Tujuan dan Manfaat Metode Pengembangan Agile

Berikut ini merupakan tujuan dari *Agile Development* yang terbagi menjadi tujuh, antara lain:

#### 1. *High – value & working App System*

Tujuan yang pertama, untuk menghasilkan sebuah perangkat lunak dengan nilai jual tinggi serta dapat menekan biaya pembuatan. Dan yang terpenting adalah dapat menghasilkan produk dengan kualitas yang baik.

#### 2. *Iterative, Incremental, Evolutionary*

*Agile* merupakan model pengembangan yang dilakukan secara iteratif, berulang – ulang, dan dapat mengalami perubahan apabila diperlukan. Dapat dikatakan bahwa, metode ini terbilang fleksibel dan dapat digunakan pada proyek pengembangan jangka pendek.

#### 3. *Cost Control & value – driven Development*

Proses pengembangan perangkat lunak dapat disesuaikan dengan kebutuhan dari pengguna (*user*). Kemudian, tim *developer* dapat mengontrol biaya dan waktu yang diperlukan untuk proses pengembangan *software* sesuai kebutuhan.

#### **4. *High – quality production***

Kualitas dari produk perangkat lunak tetap terjaga dengan baik meskipun biaya dan waktu yang diperlukan terbilang sedikit.

#### **5. *Flexible & Risk Management***

Fleksibel disini dapat diartikan dengan pertemuan dengan klien dapat dilakukan kapanpun sehingga fungsionalitas dari perangkat lunak dapat terjaga. Yang terpenting, dapat meminimalisir terjadinya kesalahan pada program maupun produk sebelum dilakukan proses deploy aplikasi.

#### **6. *Collaboration***

Proses kolaborasi disini dilakukan oleh setiap tim pengembang untuk mendiskusikan *feedback* yang diberikan oleh klien. Sehingga perlu adanya komunikasi dan koordinasi yang baik antar tim *developer*.

#### **7. *Self – organizing, Self – Managing Teams***

Tujuan terakhir dari metode *Agile* ini adalah pengembang diberikan akses untuk memajemen sendiri urusan *software Development*. Tugas dari seorang manajer untuk menjadi penghubung antara *developer* dan klien sehingga dapat mengurangi terjadi miss communication.

Manfaat dari penggunaan metode *Agile* sendiri dapat dirasakan tidak hanya dari sisi *developer* saja. Tetapi pada sisi *client*, *vendor*, serta manajer juga merasakan manfaat dari penggunaan *Agile*. *Client* dapat memberikan *feedback* kepada tim *developer* untuk menambahkan atau mengubah fitur dari aplikasi tersebut sebelum benar – benar dirilis. Kemudian, manajer dapat mengontrol kerja dari setiap tim dengan lebih baik. Dari segi *vendor* dapat mengurangi pemborosan dan dapat difokuskan pada peningkatan efisiensi dan pengembangan fitur. Dan dari sisi *developer* sendiri dapat meningkatkan produktivitas tiap departemen. Karena, setiap tim dapat melakukan pengerjaan tiap tugas tanpa perlu harus menunggu tim yang lain menyelesaikan tugas nya.

### **2.2.2 Prinsip Metode Pengembangan *Agile***

1. Prioritas utama proses *Agile* adalah memuaskan klien dengan menghasilkan perangkat lunak yang bernilai dengan cepat dan rutin.
2. Menyambut perubahan kebutuhan, walaupun terlambat dalam pengembangan perangkat lunak. Proses *Agile* memanfaatkan perubahan untuk keuntungan kompetitif klien.



3. Menghasilkan perangkat lunak yang bekerja secara rutin, dari jangka waktu beberapa minggu sampai beberapa bulan, dengan preferensi kepada jangka waktu yang lebih pendek.
4. Rekan bisnis dan pengembang perangkat lunak harus bekerja sama tiap hari sepanjang proyek.
5. Kembangkan proyek di sekitar individual yang termotivasi. Berikan mereka lingkungan dan dukungan yang mereka butuhkan, dan percayai mereka untuk menyelesaikan pekerjaan dengan baik.
6. Metode yang paling efisien dan efektif untuk menyampaikan informasi dari dan dalam tim pengembang perangkat lunak adalah dengan komunikasi secara langsung.
7. Perangkat lunak yang bekerja adalah ukuran utama kemajuan.
8. Proses *Agile* menggalakkan pengembangan berkelanjutan. Sponsor-sponsor, pengembang-pengembang, dan pengguna-pengguna dapat mempertahankan kecepatan tetap secara berkelanjutan.
9. Perhatian yang berkesinambungan terhadap keunggulan teknis dan rancangan yang baik meningkatkan *Agility*
10. Kesederhanaan (memaksimalkan sumber daya yang tersedia) adalah hal yang amat penting.
11. Arsitektur, kebutuhan, dan rancangan perangkat lunak terbaik muncul dari tim yang dapat mengorganisir diri sendiri.
12. Secara berkala, tim pengembang berefleksi tentang bagaimana untuk menjadi lebih efektif, kemudian menyesuaikan dan menyelaraskan kebiasaan bekerja mereka.

### 2.2.3 Kelebihan dan Kekurangan *Agile Development*

#### **Kelebihan *Agile Development***

Berikut ini merupakan beberapa kelebihan dari Metode Pengembangan *Agile*:

- Proses pengembangan perangkat lunak membutuhkan waktu yang relatif cepat dan tidak membutuhkan *resources* yang besar.
- Perubahan dapat ditangani dengan cepat sesuai dengan kebutuhan *client*.
- *Client* dapat memberikan *feedback* kepada tim pengembang dalam proses pembuatan program.
- Pelacakan requirement dengan melihat *Backlog* produk.
- Keterlibatan *user* secara aktif.
- Rilis yang lebih cepat dan berkala, fungsi dirilis setiap akhir iterasi.
- Testing dilakukan setiap saat.

### **Kekurangan *Agile Development***

Berikut ini merupakan beberapa kekurangan dari Metode Pengembangan *Agile*:

- *Agile* tidak cocok apabila dikerjakan oleh tim yang tidak memiliki komitmen untuk menyelesaikan proyek bersama – sama.
- Metode *Agile* ini kurang tepat apabila dikerjakan dengan jumlah skala tim yang besar (>20 orang).
- Tim pengembang harus selalu bersiap karena perubahan dapat terjadi sewaktu – waktu.
- Interaksi dengan *client* yang kadang terlalu berlebihan.
- *Agile* sulit diimplementasikan dalam proyek yang berskala besar.
- Membutuhkan manajemen tim yang terlatih.

## **2.3. Metodologi Pengembangan *Scrum***

Secara harfiah *Scrum* adalah kerangka kerja di mana orang dapat mengatasi masalah adaptif yang kompleks, sementara secara produktif dan kreatif memberikan produk dengan nilai setinggi mungkin. *Scrum* adalah iteratif dan pengembangan perangkat lunak kerangka kerja tambahan tangkas untuk proyek-proyek perangkat lunak dan mengelola produk atau pengembangan aplikasi. *Scrum* adalah sebuah *framework* yang mengimplementasikan proses *Agile Development*. Untuk menjelaskan bagaimana *Scrum* mengubah paradigma dalam proses bekerja walaupun sebenarnya metode ini biasanya digunakan secara tim, tapi tidak ada larangan untuk menggunakannya secara perorangan. Metode *Scrum* membagi proses *Development* menjadi beberapa *Sprint*.

### **2.3.1 Tahapan atau Aktivitas Pengembangan Metode *Scrum***

Aktivitas wajib dalam *Scrum* dibuat untuk menciptakan sebuah kesinambungan dan mengurangi adanya aktivitas lain yang tidak tercantum di dalam *Scrum*. Setiap tahapan atau aktivitas di dalam *Scrum* memiliki batasan waktu, yang artinya selalu memiliki durasi maksimum. Setiap aktivitas yang terdapat dalam *Scrum* berkesempatan untuk meninjau dan merubah sesuatu. Aktivitas dalam *Scrum* dirancang secara khusus untuk menciptakan transparansi dan peninjauan sampai ke tingkatan kritis. *Scrum* memiliki tiga kelompok fase, berikut merupakan kelompok dari fase *Scrum* ;

## **1. Pregame**

### **a. Perencanaan (*Planning*)**

Perencanaan dibagi berdasarkan jenis pengembangan produk. Apabila produk yang dikembangkan merupakan sebuah sistem yang baru, maka pada fase ini dilakukan konseptualisasi dan analisis. Sedangkan apabila produk yang dikembangkan merupakan sistem yang telah ada, maka pada fase ini hanya dilakukan analisis.

### **b. Arsitektur**

Merancang daftar *Backlog* yang akan diimplementasikan. Pada fase ini dilakukan modifikasi rancangan sistem dan rancangan pada level yang lebih tinggi seperti tampilan.

## **2. Game**

Pada fase ini terdapat *Development Sprints*, yaitu pengembangan dari fungsionalitas yang baru, dengan memperhatikan variabel waktu, permintaan, kualitas, harga, dan kompetisi. Interaksi dari variabel tersebut akan mendefinisikan akhir dari fase ini. Terdapat banyak *Sprint* yang dilakukan pada fase ini. Berikut aktivitas-aktivitas yang dilakukan dalam metode *Sprint* :

### **a. Product Backlog**

Proyek yang menggunakan metode *Scrum* didasari oleh sebuah visi dari produk yang dibuat oleh *Product Owner*, dan dituliskan ke dalam *Product Backlog*. *Product Backlog* memuat daftar dari poin yang akan dibuat oleh tim yang didasari oleh prioritas kebutuhan yang dinilai berdasarkan kebutuhan customer. *Product Backlog* disusun disepanjang waktu pengerjaan proyek, dengan penambahan, pengurangan, dan perubahan prioritas yang berkesinambungan.

### **b. Sprint**

*Scrum* menstrukturisasi pengembangan produk dengan menggunakan satu runtunan pekerjaan yang disebut dengan *Sprint*, iterasi dari pekerjaan yang biasanya berdurasi 1-4 minggu. *Sprint* memiliki durasi dan diakhiri pada waktu yang spesifik tanpa memperhatikan status dari pekerjaan yang dilakukan, baik selesai maupun tidak. Pada durasi yang ditentukan tersebut, dilakukan pengembangan produk yang terdapat dalam proyek. Durasi dari *Sprint* tidak dapat diperpanjang.

### **c. Sprint Planning**

Pada awal dari setiap *Sprint*, dilakukan pertemuan *Sprint Planning*. Pada *Sprint Planning*, *Product Owner* dan tim *Scrum* melakukan *Review* terhadap *Product Backlog*, mendiskusikan hasil dan konteks dari poin yang akan dibuat, dan tim *Scrum* memilih poin-poin yang terdapat pada *Product Backlog* untuk dikerjakan dan dapat diselesaikan pada akhir dari *Sprint*, dimulai dari daftar teratas pada *Product Backlog*.

Setiap poin yang dipilih dari *Product Backlog* kemudian didesain dan didekomposisi dalam sebuah kumpulan pekerjaan individu. Daftar dari pekerjaan dituliskan ke dalam sebuah dokumen yang disebut dengan *Sprint Backlog*.

**d. *Daily Scrum Meeting***

*Daily Scrum* adalah kegiatan dengan batasan waktu maksimum selama 15 menit agar Tim Pengembang dapat mensinkronisasikan pekerjaan mereka dan membuat perencanaan untuk 24 jam ke depan. Hal ini dilakukan dengan meninjau pekerjaan semenjak *Daily Scrum* terakhir dan memperkirakan pekerjaan yang dapat dilakukan sebelum melakukan *Daily Scrum* berikutnya. Tim Pengembang menggunakan *Daily Scrum* untuk meninjau perkembangan menuju *Sprint Goal* dan meninjau tren perkembangan menuju selesainya pekerjaan yang ada di dalam *Sprint Backlog*. *Daily Scrum* mengoptimalkan kemungkinan Tim Pengembang akan mencapai *Sprint Goal*. *Daily Scrum* meningkatkan komunikasi, menghilangkan pertemuan-pertemuan lain, mengidentifikasi hambatan untuk dihilangkan, mendukung pembuatan keputusan secara cepat dan meningkatkan tingkat pengetahuan tim. Pertemuan ini adalah kunci dari proses peninjauan dan pengadaptasian.

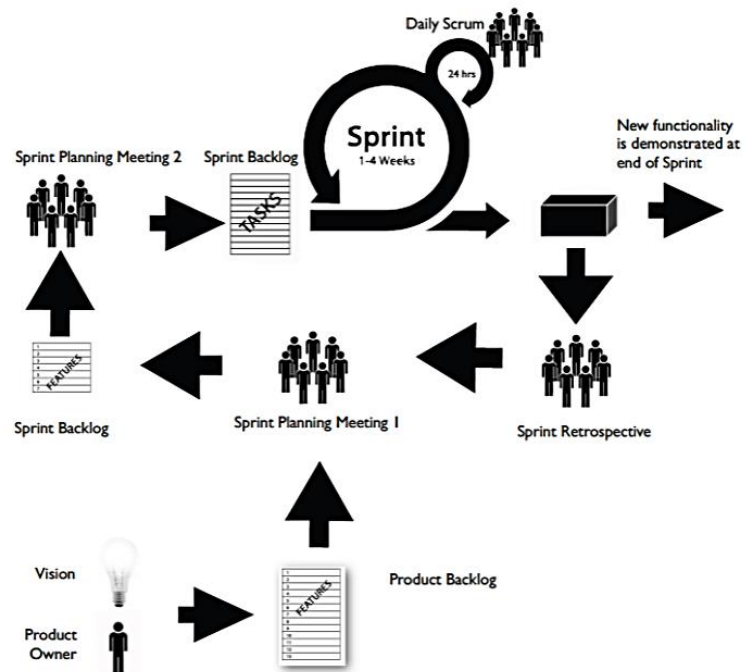
**e. *Sprint Review and Retrospective***

Setelah *Sprint* berakhir, terdapat satu aktivitas yang disebut dengan *Sprint Review*. *Sprint Review* melibatkan tim *Scrum* dan *stakeholder* untuk melakukan inspeksi pada pekerjaan yang telah selesai dilakukan selama *Sprint*. Inspeksi ini memuat diskusi mengenai pekerjaan yang sebelumnya dan rencana selanjutnya untuk pekerjaan tersebut. Pertemuan *Sprint Review* ini dihadiri oleh *Product Owner*, Tim Pengembang, *Scrum Master*, beserta *customer*, *stakeholders*, ahli, dan beberapa orang yang tertarik untuk bergabung. Melalui *Sprint Review*, tim *Scrum* selanjutnya mengadakan *Sprint Retrospective*. Pada *Sprint Retrospective*, tim *Scrum* berkesempatan untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan pada *Sprint* berikutnya.

**3. *Postgame***

Pada fase ini terdapat *Closure*. *Closure* merupakan satu fase yang menyatakan bahwa manajemen dari tim merasa bahwa variabel dari waktu, kompetisi, permintaan, harga, dan kualitas telah sesuai untuk versi terbaru yang akan dirilis, dengan demikian proyek dinyatakan "*closed*" dan memasuki fase ini. Fase ini mempersiapkan produk yang telah dikembangkan untuk beberapa kelengkapan umum lainnya seperti integrasi, pengujian sistem, dokumentasi, persiapan materi pelatihan penggunaan sistem, dan persiapan materi pemasaran.

Berikut ditampilkan gambaran umum dari tahapan/aktivitas *Sprint* dalam model *Scrum*.



Gambar 2.1 tahapan/aktivitas *Sprint* dalam model *Scrum*.

### 2.3.2 Pilar dalam Metode Pengembangan *Scrum*

#### 1. Transparansi

Semua orang yang terlibat pelanggan, CEO, kontributor individual transparan dalam transaksi sehari-hari mereka dengan orang lain. Mereka semua percaya satu sama lain, dan mereka memiliki keberanian untuk saling mengikuti kabar baik serta kabar buruk. Semua orang berusaha dan secara kolektif berkolaborasi untuk tujuan organisasi bersama, dan tidak ada yang memiliki agenda tersembunyi.

#### 2. Inspeksi

Inspeksi dalam konteks ini bukan inspeksi oleh inspektur atau auditor tetapi inspeksi oleh semua orang di Tim *Scrum*. Inspeksi dapat dilakukan untuk produk, proses, aspek orang, praktik, dan peningkatan berkelanjutan.

#### 3. Adaptasi

Adaptasi dalam konteks ini adalah tentang perbaikan terus-menerus, kemampuan untuk beradaptasi berdasarkan hasil inspeksi. Setiap orang di organisasi harus mengajukan pertanyaan ini secara teratur, Apakah kita lebih baik daripada kemarin? Untuk organisasi berbasis laba, nilainya diwakili dalam bentuk laba.

### 2.3.3 Jenis Role dalam Metode Pengembangan *Scrum*

#### 1. *Product Owner*

Pengertian produk adalah tujuan dari proyek. *Product Owner* memastikan bahwa proyek berjalan sesuai yang diharapkan. *Product Owner* merupakan penjematan antara *client* dengan *team Development*. *Product Owner* akan menuliskan spesifikasi-spesifikasi sesuai cara pandang *client*, di lain pihak harus punya empati terhadap anggota *team*. *Product Owner* bertanggung jawab untuk memaksimalkan nilai produk dan hasil kerja Tim Pengembang

#### 2. *Team Member*

*Scrum Team Member* merupakan sekelompok individu yang bekerja sama untuk menghantarkan ataupun menaikkan nilai dari produk yang diminta oleh pelanggan. Prasyarat dari tim *Scrum* adalah Semua anggota berpartisipasi, Mampu berkomunikasi dengan baik dan *Meaningful Conversation*.

#### 3. *Scrum Master*

*Scrum Master* akan mencegah hal-hal yang mengalihkan focus *team*. *Scrum Master* akan membuat suasana kondusif supaya *team* dapat bekerja sama dalam mencapai goal. Selain itu, *Scrum Master* membantu memfasilitasi keputusan tim *Development* yang *self-organizing* dan menghapus hambatan-hambatan yang dihadapi. Di dalam organisasi, *Scrum Master* membantu organisasi untuk memahami *Scrum* dan membuat perubahan yang dapat mendukung proses *Scrum*.

### 2.3.4 Kelebihan dan Kekurangan Metode Pengembangan *Scrum*

#### Kelebihan *Scrum*

Berikut ini adalah beberapa kelebihan dari *Scrum* antara lain:

- *Scrum* dapat membantu perusahaan Anda dalam menghemat waktu dan biaya (dalam hal ini uang). Biaya *overhead* dari proses dan manajemen sangat minim sehingga dapat mengarahkan kita kepada hasil yang lebih cepat dan lebih murah.
- Dengan menggunakan metode *Scrum*, Anda dapat mentransformasikan bisnis yang sulit untuk diukur menjadi mudah untuk dikembangkan.
- Pada metode *Scrum*, pergerakan pengembangan *cutting edge* dapat dengan cepat dikodekan dan diuji menggunakan metode ini. Bagaikan kesalahan yang mudah untuk diperbaiki.
- Dengan menggunakan *Scrum*, Anda dapat mengontrol dan memonitoring aktivitas peningkatan dan penurunan beban pekerjaan yang bisa terjadi kapan saja.
- Seperti metodologi *Agile* pada umumnya, *Scrum* merupakan metode iterative yang membutuhkan *feedback* secara berkelanjutan dari *user* atau pengguna.

- Dengan adanya *short Sprint* dan *constant feedback*, *Scrum* dapat dengan mudah mengatasi setiap perubahan yang terjadi.
- Dengan adanya *daily Scrum Meeting*, memungkinkan *Scrum* untuk mengukur produktivitas individu, hal ini mengarah pada peningkatan produktivitas dari setiap anggota tim.
- Dengan *Scrum*, setiap ada masalah yang timbul dapat diidentifikasi dengan baik pada pertemuan harian dan oleh karena itu setiap masalah dapat di selesaikan dengan cepat.
- Dengan menggunakan metode *Scrum*, Anda dapat dengan mudah untuk mengirim produk berkualitas sesuai dengan waktunya.
- *Scrum* dapat bekerja dengan berbagai teknologi dan bahasa pemrograman. Namun secara khusus berguna untuk pengembangan proyek dengan teknologi web 2.0 ataupun media proyek baru lainnya.

### **Kekurangan *Scrum***

Dimana ada kelebihan pasti selalu ada kekurangan, berikut ini adalah beberapa kekurangan dari *Scrum* antara lain:

- *Scrum* bisa menjadi salah satu penyebab utama terjadinya *scope creep*, kecuali ada tanggal akhir tertentu. Stakeholder proyek atau manajemen akan terus menuntut fungsi dan fitur baru untuk disampaikan.
- Setiap tugas harus didefinisikan dengan baik, karena hal ini dapat mempengaruhi perkiraan biaya dan waktu pengerjaan proyek. Jika tidak didefinisikan dengan baik maka semua hal tersebut tidak akan akurat. Dalam kasus seperti ini, biasanya tugas dapat tersebar di beberapa *Sprint*.
- Jika anggota tim Anda tidak berkomitmen dengan baik, maka proyek Anda tidak akan selesai atau bahkan bisa gagal.
- *Scrum* dapat bekerja dengan baik jika seorang *Scrum Master* dapat mempercayai tim yang mereka kelola. Jika *Scrum Master* terlalu mengontrol secara ketat, hal ini dapat menyebabkan tim menjadi tertekan dan stress, sehingga mengakibatkan demoralisasi dan kegagalan dari proyek tersebut.
- Jika sering terjadi pergantian anggota tim saat pengembangan proyek berlangsung, hal ini dapat menyebabkan efek yang kurang baik bagi perkembangan proyek tersebut, proyek akan semakin lama selesai dari waktunya.

### 2.3.5 Contoh Kasus atau Karakteristik Proyek

Dalam sejarah *Scrum*, sudah banyak proyek yang telah memakai metode *Scrum* dalam pengembangan produknya. Berikut beberapa contoh kasus penggunaan *Scrum* dalam pengembangan produknya, antara lain :

#### 1. *Google AdWords*

Merupakan proyek pengembangan perangkat lunak yang paling menguntungkan dalam sejarah. Proyek ini merupakan salah satu implementasi proyek yang paling menarik dalam *Scrum*. Aplikasi yang dibangun menjadi penyumbang terbesar pertumbuhan pendapatan *Google* bahkan menjadi pemegang pasar yang jauh di atas Intel dan tepat di atas *Chevron*, salah satu perusahaan minyak yang paling menguntungkan di dunia. Proyek *AdWords* menggunakan *Scrum* dalam pengembangannya, dengan memiliki tim yang tersebar secara terdistribusi di lima lokasi dan menyatukan secara virtual semua produk *Google* setiap dilakukan perilisan. Dalam perkembangannya, manajer dari proyek *Google* dibutuhkan untuk mengisi beberapa struktur tim. Hadirnya manajer ini pada tim *Scrum* bertujuan untuk membantu memberikan arahan untuk menyelesaikan prioritas pekerjaan yang tertinggi yang dirasa sulit diimplementasikan oleh tim. Dengan hal ini, tim tidak lagi membutuhkan *Scrum Master*, dikarenakan tim sudah dapat berjalan dengan sendirinya.

#### 2. *SirsiDynix*

Proyek *SirsiDynix* merupakan suatu proyek yang mengimplementasikan sistem perpustakaan warisan dengan lebih dari 12.500 daerah pengimplementasian. Tim yang sangat besar menjadi pemeran dalam pengerjaan proyek ini. Di dalam pengerjaannya yang sudah berlangsung selama bertahun-tahun terdapat perubahan lingkungan bisnis yang menangani permintaan-permintaan baru di tengah pengerjaan proyek. Dan yang paling menyulitkan, industri pengembangan perangkat lunak perpustakaan sedang dalam fase konsolidasi. *Dynix* telah memulai proyek pada tahun 2002 dan kemudian dikonsolidasikan dengan *Sirsi* pada tahun 2005 dan membentuk *SirsiDynix*. Menjadi suatu keuntungan, *Dynix* memulai proyek dengan menggunakan proses *Agile* yang memungkinkan adaptasi terhadap perubahan permintaan selama proyek berlangsung. Dalam pengembangannya, proyek ini juga didukung oleh *StarSoft* yang memiliki sejarah penggunaan *XP* yang memberikan hasil yang baik dalam pengembangan perangkat lunak berbasis sistem. Kombinasi dari resiko yang tinggi, skala yang besar, perubahan permintaan, konsolidasi dan akuisisi faktor bisnis, dan pengalaman *SirsiDynix* dengan *Scrum* yang dikombinasikan dengan *StarSoft* yang menggunakan *XP* mengantarkan proyek ini untuk menggunakan implementasi *Scrum* yang terintegrasi.



## BAB III

### PENUTUP

#### 3.1 Kesimpulan

*Agile Development Methods* adalah sekelompok metodologi pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun. Kolaborasi antara tim pengembang dengan klien menjadi hal yang penting dalam *Agile Development*, klien dianggap sebagai anggota tim. Fleksibilitas di tawarkan kepada klien kaitannya dengan kesadaran dan pengetahuan pelanggan tentang apa yang dia inginkan.

Terdapat dua belas prinsip pada *Agile Development* yang menjadi suatu dasar bagi tim agar sukses menerapkan *Agile Development Methods* dalam pengembangan atau pembuatan perangkat lunak. Komunikasi dan koordinasi antar tim dapat ditingkatkan dengan menggunakan metode ini, selain itu kepuasan klien dan kualitas perangkat lunak yang dihasilkan dapat terjaga.

Terdapat dua puluh metode yang termasuk dalam *Agile* metodologi. Metode-metode tersebut sering digunakan oleh praktisi pengembang perangkat lunak dalam pembuatan atau dalam pengembangan. Salah satunya yaitu metode *Scrum* yang merupakan salah satu model dari metodologi *Agile* pada manajemen pengembangan proyek. *Scrum* bisa digunakan untuk pengembangan *system* secara keseluruhan, pengembangan *system* sebagian maupun proyek internal/pelanggan

Tujuan utama *Scrum* adalah untuk inspect & adapt yang berarti bahwa melihat permasalahan yang ada, dan melakukan adaptasi terhadap masalah tersebut. Pengembangan perangkat lunak menggunakan *Scrum* menekankan untuk mengambil setiap langkah pada pengembangan perangkat lunak secara singkat.

Metode *Scrum* dapat digunakan sebagai kerangka kerja untuk mengembangkan interoperabilitas sistem dengan waktu yang efisien. Pada contoh beberapa studi kasus pengembangan dengan menggunakan metode *Scrum* dapat menghemat waktu hingga 30 sampai 60% dibandingkan dengan menggunakan metode Waterfall.

## DAFTAR PUSTAKA

- Intan Andini. (2014, December 21). [MAKALAH] Pengembangan Sistem Informasi. Retrieved January 9, 2021, from Intan Andini website:  
<https://intanandini209.wordpress.com/2014/12/21/makalah-pengembangan-sistem-informasi/>
- Metode *Agile*: Pengertian, Tujuan, Jenis, Manfaat, dan Prinsip. (2020, August 8). Retrieved January 9, 2021, from Sekawan Media | *Software House & System Integrator Indonesia* website:  
<https://www.sekawanmedia.co.id/metode-Agile-Development/>
- Williams, L., & Tomayko, J. (2015). *Agile Software Development. Computer Science Education*, 12(3), 167–168. Retrieved from  
[https://www.academia.edu/23009791/Agile\\_software\\_Development](https://www.academia.edu/23009791/Agile_software_Development)
- Oktavianti, G. (2019, April 22). PENGEMBANGAN SISTEM INFORMASI. Retrieved January 9, 2021, from ResearchGate website:  
[https://www.researchgate.net/publication/332553098\\_PENGEMBANGAN\\_SISTEM\\_INFORMASI](https://www.researchgate.net/publication/332553098_PENGEMBANGAN_SISTEM_INFORMASI)
- Unknown. (2016, October 21). METODOLOGI PENGEMBANGAN SISTEM: Alat dan Teknik dalam pengembangan sistem, Analisis sistem & Pemrograman sistem, Pengetahuan & Keahlian yang dibutuhkan analisis sistem, *Team* pengembangan sistem. Retrieved January 9, 2021, from Blogspot.com website: <http://sweett-dispositionn.blogspot.com/2016/10/metodologi-pengembangan-sistem-alat-dan.html>
- Syafnidawaty. (2020, April 5). Metode *Agile* – UNIVERSITAS RAHARJA. Retrieved January 9, 2021, from Raharja.ac.id website: <https://raharja.ac.id/2020/04/05/metode-Agile/>
- Metode *Agile*: Pengertian, Tujuan, Jenis, Manfaat, dan Prinsip. (2020, August 8). Retrieved January 9, 2021, from Sekawan Media | *Software House & System Integrator Indonesia* website:  
<https://www.sekawanmedia.co.id/metode-Agile-Development/>
- Metode Pengembangan Perangkat Lunak, *Scrum* - PDF Free Download. (2014). Retrieved January 10, 2021, from Docplayer.info website: <https://docplayer.info/56660434-Metode-pengembangan-perangkat-lunak-Scrum.html>
- Metode *Scrum*. (2015, October 14). Retrieved January 10, 2021, from Ranjani Ryan website:  
<https://ranjaniryan.wordpress.com/2015/10/14/metode-Scrum/>
- rachmat. (2019, July 9). Pengertian *SCRUM* dalam *Software Development*. Retrieved January 10, 2021, from rachmat.ID - Inspires You website: <https://rachmat.id/pengertian-Scrum>
- rachmat. (2019, July 26). Kelebihan dan Kekurangan pada *SCRUM*. Retrieved January 10, 2021, from rachmat.ID - Inspires You website: <https://rachmat.id/kelebihan-dan-kekurangan-Scrum>
- Firdaus, M. (2017). *Seminar Nasional Teknologi Informasi dan Multimedia 2017 IMPLEMENTASI KERANGKA KERJA SCRUM PADA MANAJEMEN PENGEMBANGAN SISTEM INFORMASI*. Retrieved from  
<https://ojs.amikom.ac.id/index.php/semnasteknomedia/article/viewFile/1635/1626>