

## BAB 2

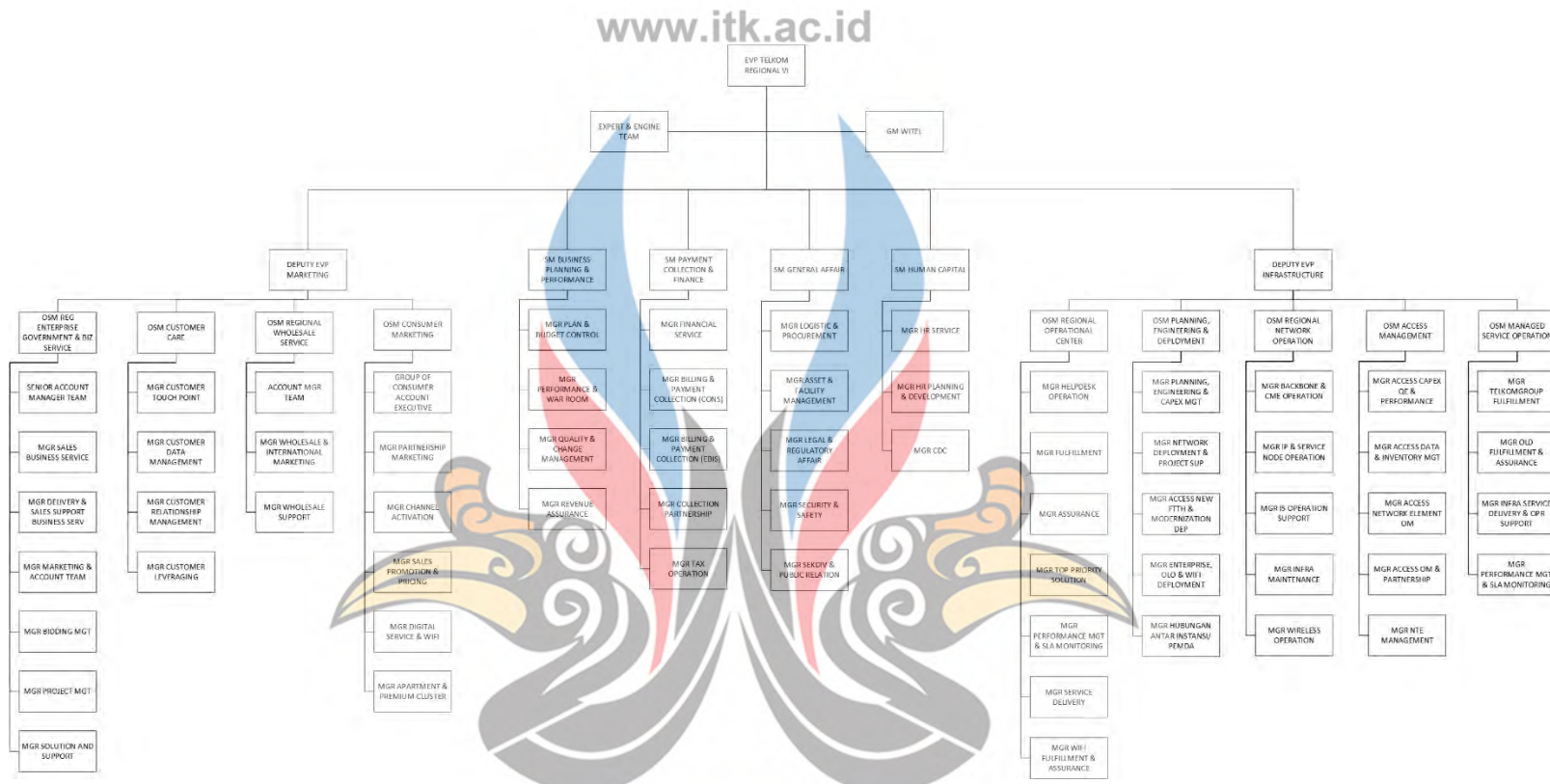
### TINJAUAN PUSTAKA

Bab ini akan membahas teori-teori yang digunakan sebagai dasar pengerjaan penelitian pengembangan Pengembangan *website HR Operational Dashboard*. Teori bersumber dari jurnal, buku, atau artikel. Teori yang digunakan diantaranya *Human Resource Information System, Software Development Life Cycle, Agile Software Development, Entity Diagram Relationship*, dan Penelitian Terdahulu.

#### 2.1 PT. Telkom Indonesia

PT Telkom Indonesia (Persero) Tbk (Telkom) adalah Badan Usaha Milik Negara (BUMN) yang berada pada bidang jasa layanan teknologi informasi dan komunikasi (TIK) dan jaringan telekomunikasi di Indonesia. Saat ini, Telkom berupaya untuk bertransformasi menjadi *digitaltelecommunication company*. Transformasi ini ditujukan agar Telkom menjadi lebih *lean (ramping)* dan *agile (lincah)* saat beradaptasi dengan perubahan dalam industri telekomunikasi. Sebagai salah satu BUMN, Telkom memiliki *core values AKHLAK* (Amanah, Kompeten, Harmonis, Loyal, Adaptif, dan Kolaboratif).

Telkom membagi Indonesia menjadi 7 regional dalam melakukan pelayanannya untuk mempermudah operasional. Telkom Regional VI dalam hal ini bertanggung jawab atas pelayanan Telkom di Pulau Kalimantan. Telkom Regional VI yang terletak di Kota Balikpapan, bertanggung jawab atas 6 Wilayah Usaha Telekomunikasi (Witel) dan lebih dari 50 Unit Bisnis. Gambar 2.1 menunjukkan struktur organisasi Telkom Regional VI.



**Gambar 2.1 Struktur Organisasi PT Telkom Regional VI**

Gambar 2.1 menunjukkan bahwa Telkom Regional VI dikepalai oleh seorang Executive Vice President (EVP), dibantu oleh Deputy, Expert Team, dan GM Witel, dalam membawahi 63 unit. Unit HR Planning and Development sebagai subjek penelitian berada dibawah Parent Unit Human Capital.

## 2.2 *Human Resources Information System (HRIS)*

HRIS atau Sistem Informasi Sumber Daya Manusia merupakan suatu sistem terintegrasi untuk mengumpulkan, menyimpan, dan menganalisa tentang sumber daya manusia didalam suatu organisasi. HRIS terdiri dari *database* dan perangkat komputer yang diperlukan untuk memanipulasi data, merekam, mengelola, menyajikan data sumber daya manusia (Jonni & Husain, 2016).

HRIS merupakan perpaduan antara bidang ilmu manajemen sumber daya manusia dan teknologi informasi. HRIS merupakan komponen yang saling terintegrasi untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi untuk mendukung pembuatan keputusan, koordinasi, kontrol, analisis, dan visualisasi dari kegiatan manajemen sumber daya manusia organisasi (Muhammad HF, 2018).

HRIS memungkinkan pengelolaan data sumber daya manusia yang efisien dan memberikan informasi yang lebih baik serta data yang akurat sehingga dapat membantu perusahaan dalam mengambil keputusan (Jonni & Husain, 2016).

## 2.3 *Software Development Life Cycle*

*Software Development Life Cycle* (SDLC) adalah kumpulan langkah yang diikuti untuk pengembangan, desain dan pemeliharaan yang sistematis untuk proyek perangkat lunak dan memastikan semua kebutuhan pengguna terpenuhi. Metodologi tadi dapat membantu memberikan produk sampai tepat waktu dan sesuai dengan keinginan *client*. Dikarenakan SDLC adalah metodologi siklus dan berulang, maka suatu perubahan akan dilakukan pada siklus berikutnya. Tahapan siklus SDLC terdiri dari analisis kebutuhan (*requirement analysis*), desain (*design*), implementasi (*implementation*), pengujian (*testing*), serta penyebaran dan pemeliharaan (*deployment and maintenance*) (Barjtya, Sharma, & Rani, 2017). Salah satu contoh SDLC yang umum digunakan adalah *Agile Development* (Budi, Siswa, & Abijono, 2016).



## 2.4 Agile Software Development

*Agile software development* adalah salah satu model SDLC *incremental* yang mampu beradaptasi terhadap perubahan, menekankan kepuasan pelanggan dan kolaborasi tim (Anwer, Aftab, & Waheed, 2017). *Agile* bersifat lincah dan mampu beradaptasi, juga tidak kaku dan tidak memakan waktu lama terhadap jadwal pengembangan. Pelanggan dilibatkan selama pengembangan proyek, sehingga *agile* dapat meningkatkan kualitas produk disetiap iterasinya (Rastogi, 2015). Tabel 2.1 berisi beberapa prinsip dalam *Agile Software Development*.

Tabel 2.1 Prinsip *Agile Software Development*

Prinsip	Deskripsi
<i>Customer involvement</i>	Pelanggan harus terlibat erat selama proses pengembangan. Peran mereka adalah menyediakan dan memprioritaskan kebutuhan sistem baru dan mengevaluasi setiap iterasi sistem.
<i>Incremental delivery</i>	Perangkat lunak dikembangkan secara <i>inkremental</i> dengan persyaratan ditentukan oleh pelanggan pada setiap iterasi.
<i>People not process</i>	Keterampilan tim pengembangan harus diakui dan dieksploitasi. Anggota tim harus dibiarkan mengembangkan cara mereka untuk bekerja tanpa proses preskriptif.
<i>Embrace change</i>	Selalu siap dengan perubahan kebutuhan dan desain sistem untuk mengakomodasi perubahan.
<i>Maintain simplicity</i>	Fokus pada kesederhanaan baik perangkat lunak yang dikembangkan maupun proses pengembangan. Eliminasi kerumitan dalam sistem.

\*) (Sommerville, 2011)

Beberapa contoh model *agile software development* adalah *Scrum*, *Extreme Programming* (XP), *Feature Driven Development* (FDD), dan *Kanban*. (Anwer, Aftab, & Waheed, 2017).

Tabel 2.2 Perbandingan Scrum, XP, FDD dan Kanban

Karakteristik	Scrum	XP	FDD	Kanban
Pendekatan pengembangan	Iterasi dan inkremental	Iterasi dan inkremental	Iterasi dan inkremental	Inkremental
Ukuran tim	<10	<20	Banyak (Lebih dari 1 tim)	Tidak didefinisikan
Jangka waktu untuk satu iterasi	2-4 Minggu	1-6 Minggu	<2 Minggu	Tidak didefinisikan
Ukuran proyek	Semua tipe proyek	Proyek kecil	Proyek kompleks	Proyek kecil
Komunikasi tim	Informal	Informal	Berdasarkan dokumentasi	Informal tatap muka
Keterlibatan pelanggan	Melalui product owner selama berjalannya proyek	Selama berjalannya proyek	Pelanggan melalui laporan	Pelanggan melalui rilis produk secara bertahap

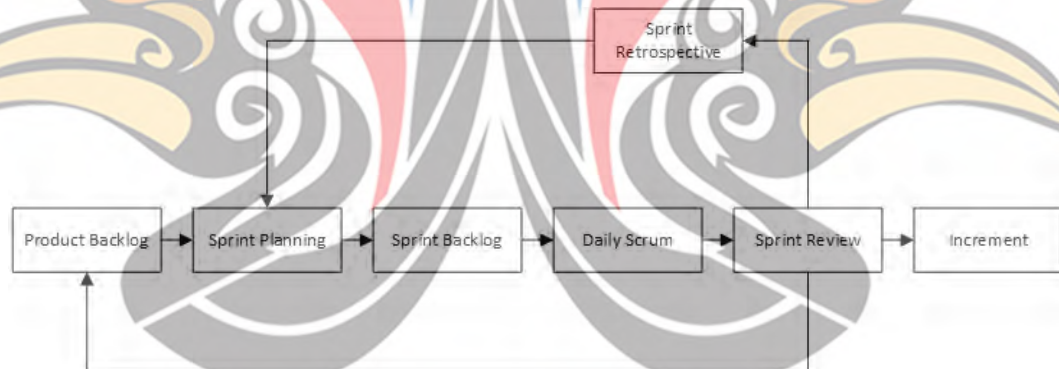
(Anand & Dinakaran, 2016)

Tabel 2.2 menjelaskan perbandingan antara *scrum*, XP, FDD, dan *kanban*. FDD merupakan metode pengembangan yang penggunaannya cocok untuk proyek yang kompleks dengan tim yang sangat besar. Sedangkan untuk proyek kecil, lebih baik menggunakan *scrum*, XP atau *kanban*. *Scrum* dan XP memecah kebutuhan sistem menjadi beberapa iterasi, sedangkan pada *kanban* proyek dilaksanakan secara kontinyu dan produk dirilis berkala. Penggunaan *kanban* lebih tepat untuk tim yang mempunyai banyak pekerjaan yang tidak terencana selama proyek berlangsung. *Scrum* dan XP berjalan berdasarkan iterasi dan inkremental dengan fokus yang berbeda. *Scrum* berfokus kepada manajemen proyek sedangkan XP berfokus pada teknik pengembangan. *Scrum* memberikan kebebasan pengembang dalam memilih *product backlog* yang akan dikerjakan sedangkan XP mengikuti urutan yang sudah ditentukan oleh pelanggan. Selama iterasi berjalan, pada XP pelanggan bisa mengubah *item backlog* yang dikerjakan selama *item* itu belum

dikerjakan. Sedangkan pada *scrum*, perubahan dikerjakan setelah *sprint* selesai. Karena sifatnya tersebut, XP lebih cocok untuk pengembang *advanced* yang siap akan perubahan perencanaan. Maka dari itu penelitian ini memilih *scrum* sebagai model pengembangan perangkat lunak.

#### 2.4.1 Scrum

*Scrum* merupakan salah satu model pengembangan *agile* untuk pengelolaan pengembangan perangkat lunak yang menerapkan paradigma *object-oriented* (OO) (Fadlullah & Lesmidayarti, 2017). *Scrum* merupakan pengembangan perangkat lunak *agile* yang paling banyak digunakan. Ide dibalik *scrum* adalah untuk menangani kekurangan pada pengembangan perangkat lunak tradisional. Dalam *scrum*, setiap rilis produk direncanakan sesuai dengan kebutuhan pelanggan, tekanan waktu, kualitas produk dan sumber daya yang tersedia (Anwer, Aftab, & Waheed, 2017).



Gambar 2.2 Metodologi Scrum

Gambar 2.1 merupakan alur dari model *scrum*. *Product backlog*, *sprint backlog*, *increment* merupakan artefak pada *scrum*. *Event* dalam model *scrum* diantaranya *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective*. Dalam model *scrum*, terdapat sejumlah individu yang berkerja sama yang dinamakan *scrum team* dan dibagi menjadi 3 kategori diantaranya *developers*, *product owner*, dan *scrum master* (Schwaber & Sutherland, 2020).



### 2.4.1.1 Scrum Team

*Scrum team* adalah tim kecil yang terdiri dari sejumlah individu yang terdiri dari *developers*, seorang *scrum master*, dan seorang *product owner*. *Scrum team* harus cukup gesit dan mampu menyelesaikan pekerjaan didalam *sprint* yang biasanya berjumlah 10 orang atau kurang. (Schwaber & Sutherland, 2020).

#### 1. Product Owner

*Product owner* adalah seseorang yang mewakili pemangku kepentingan dengan tugas memaksimalkan nilai produk yang dihasilkan *scrum team*. Tugas *product owner* meliputi:

- Menyampaikan *product goal*.
- Merancang dan menyampaikan *product backlog*.
- Memastikan *product backlog* transparan, terlihat dan mudah dipahami

*Product owner* berjumlah satu orang yang mewakili kebutuhan para pemangku kepentingan yang tertulis pada *product backlog*. Maka pemangku kepentingan harus meyakinkan *product owner* jika ingin ada perubahan pada *product backlog*.

#### 2. Scrum Master

*Scrum master* merupakan seseorang yang memimpin *scrum team*. *Scrum master* bertanggung jawab atas efektivitas *scrum team*. *Scrum master* membantu *developer* dengan beberapa cara meliputi:

- Melatih anggota tim dalam manajemen diri.
- Membantu *developer* untuk menciptakan produk yang berkualitas.
- Menghilangkan hambatan kemajuan *developer*.
- Memastikan bahwa semua *scrum event* berlangsung dan bersifat positif, produktif, dan tepat waktu.

*Scrum master* juga membantu *product owner* dengan beberapa cara meliputi:

- Membantu menemukan teknik untuk menentukan definisi *product goal* yang efektif dan manajemen *product backlog*.
- Membantu *scrum team* memahami kebutuhan *product backlog*.
- Membantu menetapkan perencanaan produk empiris.

#### 3. Developers

[www.itk.ac.id](http://www.itk.ac.id)

*Developers* adalah orang yang bertugas untuk mengembangkan item pada *product backlog* menjadi *increment* yang siap digunakan disetiap *sprint*. Keterampilan yang dibutuhkan pengembang beragam dan tidak hanya terbatas pada *programmer*. *Developers* bertanggung jawab atas:

- Membuat *sprint backlog*.
- Menyesuaikan rencana setiap hari.
- Menanamkan kualitas dengan mengikuti definisi “*Done*”.
- Saling bertanggung jawab sebagai profesional.

Definisi “*done*” yang dimaksud merupakan patokan apakah *product backlog* yang telah dikerjakan sudah dapat dirilis atau belum. Tentunya *item* yang dikerjakan *developer* juga harus memenuhi lama waktu yang disepakati dalam satu *sprint*, jika *sprint* selesai dan *item* belum memenuhi definisi *done* maka *item* akan dilanjutkan pada *sprint* berikutnya.

#### 2.4.1.2 *Scrum Event*

*Event* didalam *scrum* dibuat untuk meminimalisir kebutuhan rapat yang tidak ditentukan didalam *scrum*. *Event* didalam *scrum* memiliki durasi, sehingga setiap *event* memiliki waktu maksimal. Setelah *sprint* dimulai, maka durasinya bersifat tetap dan tidak dapat diperpendek maupun diperpanjang. *Event* dalam *scrum* meliputi *sprint*, *sprint planning*, *daily scrum*, *sprint review*, dan *sprint retrospective* (Schwaber & Sutherland, 2020).

##### 1. *Sprint*

*Sprint* merupakan jantung dari *scrum*, dimana ide diubah menjadi nilai. *Sprint* memiliki durasi 1 bulan atau kurang. *Sprint* baru dimulai segera setelah *sprint* sebelumnya berakhir. Waktu *sprint* bersifat tetap, sehingga jika sudah ditentukan ketika *sprint* pertama sudah ditentukan lamanya, maka *sprint* berikutnya harus mengikuti lama waktu dari *sprint* sebelumnya.

##### 2. *Sprint Planning*

*Sprint planning* merencanakan apa yang dilakukan didalam *sprint* dan disusun oleh seluruh *scrum team*. *Sprint planning* memiliki waktu maksimal 8 jam untuk *sprint* dengan jangka waktu 1 bulan.

##### 3. *Daily Scrum*



*Daily scrum* adalah *event* dengan batas waktu 15 menit bagi *developer* untuk merencanakan apa yang akan dilakukan dalam 24 jam kedepan. *Daily scrum* meningkatkan performa dan kolaborasi dari pekerjaan yang dilakukan dengan menginspeksi hasil yang telah dikerjakan dan melakukan perencanaan terhadap pekerjaan selanjutnya.

4. *Sprint Review*

*Sprint review* dilaksanakan setelah satu *sprint* selesai. *Sprint review* bertujuan memeriksa hasil *sprint* yang baru selesai dan menentukan adaptasi yang akan dilakukan nantinya. *Scrum team* melaporkan hasil *sprint* kepada *product owner* dan merencanakan hal-hal yang dapat membantu mengoptimalkan nilai produk. Waktu pelaksanaan maksimal *sprint review* adalah 4 jam dalam waktu 1 bulan *sprint*.

5. *Sprint Retrospective*

*Sprint retrospective* bertujuan untuk merencanakan bagaimana kualitas dan efektivitas ditingkatkan untuk *sprint* berikutnya. Pada *event* ini dibahas hal-hal yang sudah terjadi dalam *sprint* yang sudah terjadi, dan *improvement* apa yang didapatkan. Waktu pelaksanaan maksimal *sprint retrospective* adalah 3 jam dalam waktu 1 bulan *sprint*.

### 2.4.1.3 *Scrum Artifact*

*Scrum artifact* merepresentasikan nilai-nilai yang dirancang untuk memaksimalkan transparansi informasi agar seluruh peserta *scrum* memiliki pemahaman yang sama (Schwaber & Sutherland, 2020).

1. *Product Backlog*

*Product backlog* adalah daftar yang berisi berbagai hal yang diperlukan pada pengembangan produk dan merupakan satu satunya sumber persyaratan perubahan yang dibutuhkan untuk perubahan yang akan diberlakukan pada produk. *Product backlog* berisi semua fitur, fungsi, kebutuhan, peningkatan, dan perbaikan yang merupakan perbaikan yang harus dirilis pada produk yang akan dirilis. Masing masing item pada *product backlog* memiliki deskripsi, urutan, dan nilai.

2. *Sprint Backlog*

[www.itk.ac.id](http://www.itk.ac.id)

*Sprint backlog* merupakan item dari *product backlog* yang akan diselesaikan pada *sprint*. *Sprint backlog* terdiri dari *sprint goal* (mengapa), beberapa item dari *product backlog* (apa), serta rencana yang dapat dilakukan untuk menyampaikan *increment* (bagaimana).



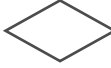

3. *Increment*

*Increment* adalah kumpulan hasil dari item *product backlog* yang telah diselesaikan. Pada akhir *sprint*, *increment* harus dalam kondisi bisa digunakan dan memenuhi kriteria “*done*” dari *scrum team*. Produk harus bisa digunakan terlepas dari *product owner* akan merilisnya atau tidak.

**2.5 Entity Diagram Relationship**

*Entity relationship diagram* (ERD) merupakan suatu konsep untuk menguraikan struktur basis data. Tujuan ERD adalah untuk menggambarkan hubungan dalam perancangan basisdata serta memodelkan struktur data. Tabel 2.3 menunjukkan notasi yang digunakan pada ERD (Lestari, 2016).

**Tabel 2.3 Notasi ERD**

<b>Simbol</b>	<b>Nama</b>	<b>Deskripsi</b>
	<i>Entity</i>	Menunjukkan entitas apa saja dalam sistem
	<i>Attribute</i>	Menunjukkan atribut pada sistem
	<i>Relation</i>	Menunjukkan relasi antar entitas
	<i>Link</i>	Menunjukkan hubungan

**2.6 Penelitian Terdahulu**

Rangkuman hasil penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang telah dilakukan dapat dilihat pada Tabel 2.4.

Tabel 2.4 Penelitian terdahulu

No	Nama dan Tahun Publikasi	Hasil
1	Febrianto dkk, 2020	<p><b>Judul :</b> Pengembangan Sistem Pengelolaan dan Pemantauan Proyek dengan Metode Agile Pola Scrum</p> <p><b>Kesimpulan &amp; Saran :</b> Pengembangan sistem menggunakan <i>Agile Development Method</i> dengan pola pendekatan <i>Scrum</i> mempermudah untuk pengembang lebih cepat mendapatkan ulasan dari fitur-fitur yang dikerjakan, dan sistem dapat langsung dipakai oleh Visi Prima Group karena proses deployment yang dilakukan di setiap akhir sprint</p> <p><b>Metode :</b> <i>Scrum</i></p> <p><b>Hasil :</b> Sistem Pengelolaan dan Pemantauan Proyek</p>
2	Saepurrahman dkk, 2019	<p><b>Judul :</b> Sistem Pencatatan Dan Pendataan Manajemen SDM Dengan Model <i>Scrum</i></p> <p><b>Kesimpulan &amp; Saran :</b> Sistem memudahkan proses pendataan pegawai sehingga menghasilkan informasi yang lebih efektif dan efisien</p> <p><b>Metode :</b> <i>Scrum</i></p> <p><b>Hasil :</b> Sistem Pencatatan dan Pendataan Manajemen SDM</p>
3	Gizwanda, 2018	<p><b>Judul :</b> Pengembangan Sistem Informasi Penelitian dan Pengabdian Masyarakat (Studi Kasus: Institut Teknologi Kalimantan)</p> <p><b>Kesimpulan &amp; Saran :</b> Sistem informasi yang mampu memenuhi kebutuhan pihak LPMM ITK</p> <p><b>Metode :</b> <i>Scrum</i></p> <p><b>Hasil :</b> Sistem Informasi Penelitian dan Pengabdian Masyarakat</p>
4	Wardhani dkk, 2018	<p><b>Judul :</b> Sistem Informasi Manajemen Sumber Daya Manusia Berbasis Web (Studi Kasus: Pt. Klik Teknologi Indonesia)</p> <p><b>Kesimpulan &amp; Saran :</b> Bahwa sistem informasi manajemen SDM yang dikembangkan membuat pengelolaan SDM di perusahaan menjadi lebih mudah, cepat dan akurat</p> <p><b>Metode :</b> <i>Waterfall</i></p> <p><b>Hasil :</b> Sistem Informasi Manajemen Sumber Daya Manusia</p>



No	Nama dan Tahun Publikasi	Hasil
5	Firdaus dkk, 2016	<p><b>Judul :</b> Penerapan <i>Scrum Agile Development</i> Dalam Pengembangan Sistem Informasi <i>Monitoring</i> Mahasiswa Bidikmisi Berbasis Web (Studi Kasus Di Universitas Sriwijaya)</p> <p><b>Kesimpulan &amp; Saran :</b> Pengembangan dengan model Scrum sangat membantu dalam menghasilkan produk aplikasi sistem <i>monitoring</i> akademik dan non akademik mahasiswa Bidik Misi.</p> <p><b>Metode :</b> <i>Scrum</i></p> <p><b>Hasil :</b> Sistem <i>Monitoring</i> Akademik dan non Akademik Mahasiswa Bidik Misi</p>

Tabel 2.4 menunjukkan daftar penelitian terdahulu yang menjadi referensi untuk membantu penelitian yang akan dikerjakan. Saepurrahman dan Wardhani mengembangkan sistem manajemen Sumber Daya Manusia sehingga menjadi referensi pengembangan fitur fitur yang akan dikerjakan. Febrianto, Gizwanda dan Firdaus menjadi referensi penggunaan model *scrum*. Beragam metode digunakan pada pengembangan tersebut, diantaranya adalah *waterfall*, *scrum*. Berdasarkan Tabel 2.2, maka penulis menggunakan model *scrum*, salah satu dari pengembangan *agile* dikarenakan sifatnya yang mudah beradaptasi dan cepat terhadap perubahan.