

BAB 2

LANDASAN TEORI

Beberapa telaah literatur yang akan dibahas dalam perancangan, pembangunan, dan pengembangan sistem perjalanan dinas karyawan pada sistem informasi HR berbasis ERP (Odoo) adalah sebagai berikut:

2.1. *Human Resource Information System (HRIS)*

Human Resource Information System (HRIS) adalah sistem yang memungkinkan untuk memperoleh, menyimpan, memproses, menganalisis, mengambil, memulihkan, dan menyebarkan informasi tentang sumber daya manusia dalam organisasi dengan melibatkan perangkat lunak, staff, data, dan proses-proses tertentu [9]. Manfaat utama dari implementasi HRIS dalam departemen SDM yaitu mengotomatisasi tugas dan fungsi yang berkaitan dengan SDM sehingga bisa meningkatkan efisiensi operasi. Implementasi HRIS secara berkelanjutan juga memungkinkan perusahaan untuk mengurangi pengeluaran dengan mengurangi jumlah karyawan yang tugasnya diotomatisasi oleh HRIS dan mengurangi penggunaan kertas, serta memfasilitasi komunikasi antara manajemen puncak dengan karyawan, dan mampu menghasilkan data dan informasi yang relevan sebagai dasar untuk pembuatan keputusan yang strategis dalam organisasi.

2.2. *Enterprise Resource Planning (ERP)*

Enterprise Resource Planning (ERP) adalah sistem perangkat lunak yang digerakkan oleh sistem manajemen bisnis yang menyediakan solusi terintegrasi dari aspek-aspek operasi / proses bisnis dalam bentuk modular untuk kebutuhan pemrosesan informasi dalam sebuah perusahaan, serta secara efisien dan efektif mengelola sumber daya yang ada, termasuk di dalamnya sumber daya manusia [10]. Dalam implementasinya, ERP saat ini banyak memanfaatkan perkembangan teknologi digital untuk lebih meningkatkan mobilitas, efektifitas komunikasi, dan efisiensi, salah satunya adalah dengan penggunaan ERP berbasis web. Pemanfaatan perkembangan teknologi digital ini diharapkan dapat meningkatkan keunggulan strategis maupun kompetitif dari perusahaan yang menggunakan ERP [11]. Namun dalam prakteknya masih terdapat perusahaan yang mempertimbangkan penggunaan

ERP dalam proses bisnis mereka dengan pertimbangan dari segi pengeluaran yang perlu dikeluarkan untuk menggunakan ERP. Saat ini terdapat banyak sistem ERP yang tersedia di pasaran, baik yang berbayar (seperti SAP, Oracle, Peoplesoft, dll.) maupun yang bersifat *open source* (seperti Odoo) sehingga memungkinkan dilakukannya kustomisasi lebih lanjut sesuai dengan kebutuhan perusahaan.

2.3. Odoo

Odoo adalah sistem ERP yang didistribusikan secara *open source* dengan berbagai aplikasi bisnis yang terintegrasi di dalamnya, antara lain *CRM, Sales, Accounting, Manufacturing, Inventory, Marketing, Finance, Project Management, HR, MRP*, dll. [12]. Odoo diimplementasikan sebagai ERP yang berbasis aplikasi web yang dibangun dengan menggunakan Bahasa pemrograman *Python, XML, JavaScript*, dan *PostgreSQL* sebagai basis data. Dalam perkembangannya, Odoo beberapa kali mengalami pergantian nama, dari TinyERP, kemudian menjadi OpenERP, dan akhirnya menjadi Odoo [13]. Sifat *open source* pada Odoo membuat sistem ERP ini tidak hanya bisa digunakan pada perusahaan-perusahaan besar saja, tetapi bisa digunakan juga pada perusahaan kecil dan independen. Aplikasi bisnis yang terintegrasi dalam Odoo berbentuk modul-modul yang siap untuk diunduh dan digunakan dan sebagian besar bisa didapatkan secara gratis [4]. Selain itu, Odoo juga memungkinkan kustomisasi modul yang sudah ada maupun membangun modul baru, hal ini menunjukkan tingkat fleksibilitas Odoo yang tinggi.

2.4. Agile

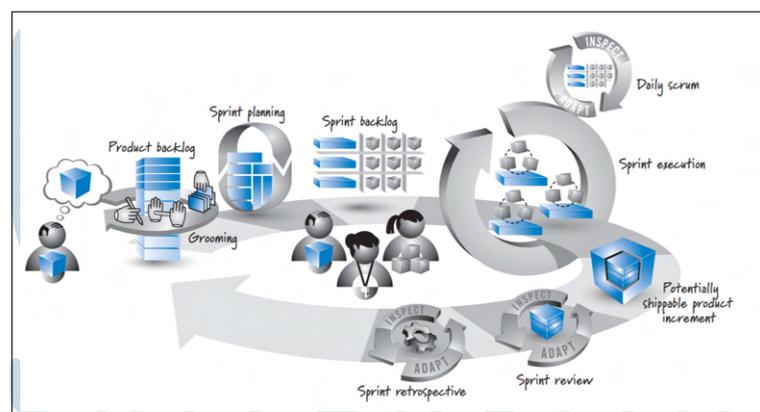
Agile adalah metodologi yang memungkinkan pembangunan perangkat lunak dengan lebih cepat dan efisien serta memiliki respons yang tinggi terhadap perubahan [14]. Tujuan utama dari metodologi ini adalah untuk mengurangi risiko dengan memecah proyek menjadi bagian-bagian kecil dengan waktu pengerjaan yang terbatas. Setiap bagian kecil tersebut dibangun seperti proyek mandiri dalam skala yang lebih kecil dan masing-masing hanya berlangsung selama beberapa minggu [15]. Selain itu, metodologi ini memiliki siklus hidup yang pendek dan sangat menekankan pada kepuasan pelanggan dengan terus menghadirkan perangkat lunak fungsional secara terus menerus dalam bentuk iteratif dari setiap bagian-bagian yang telah dibangun.

2.5. Scrum

Scrum merupakan salah satu dari metodologi Agile yang mengikuti pendekatan empiris dengan menerapkan fleksibilitas, kemampuan beradaptasi, dan produktivitas [14]. Selain itu, Scrum juga mengikuti pendekatan inkremental dan bersifat iteratif yang digunakan untuk mengembangkan perangkat lunak dengan requirement yang sering berubah [5]. Peran yang terdapat dalam metode Scrum menurut [16] antara lain:

1. **Product Owner:** Bertanggung jawab atas apa yang akan dikembangkan dan sesuai urutannya.
2. **Scrum Master:** Bertanggung jawab untuk membimbing tim dalam menciptakan dan mengikuti prosesnya dalam menyelesaikan pekerjaan.
3. **Development Team:** Bertanggung jawab untuk menghasilkan produk perangkat lunak, dengan mengatur dan mengelola pekerjaannya secara mandiri

Berdasarkan [16], aktivitas yang dilakukan dalam Scrum dapat digambarkan seperti pada gambar 2.1, dan diuraikan sebagai berikut:



Gambar 2.1. Scrum Framework

1. **Product Backlog.** Pada tahapan awal akan ditentukan urutan pekerjaan-pekerjaan yang terdapat dalam proyek dan dikomunikasikan dalam bentuk daftar prioritas yang dikenal sebagai *Product Backlog*. *Product backlog* dapat berisi fitur baru, perubahan fitur yang sudah ada, cacat yang perlu diperbaiki, peningkatan teknis, dll. *Product backlog* merupakan artefak yang

terus berkembang dimana item dapat ditambahkan, dihapus, dan direvisi saat kebutuhan bisnis berubah, atau seiring berkembangnya pemahaman tim tentang produk yang diperoleh melalui *feedback* setiap *sprint*. Aktivitas membuat, menyempurnakan, dan memprioritaskan item-item dalam *product backlog* dikenal sebagai *Grooming*.

2. ***Sprint Planning***. Pekerjaan-pekerjaan dalam *Scrum* dilakukan dalam iterasi atau siklus yang disebut *Sprint*. Ketika sebuah *sprint* selesai, maka akan dilanjutkan dengan *sprint* baru. Untuk menentukan bagian terpenting dari item *product backlog* yang akan dibangun pada *sprint* mendatang, maka dilakukan *Sprint Planning*. Dengan tujuan menentukan apa yang seharusnya dicapai pada *sprint* mendatang, maka *product backlog* akan ditinjau kembali dan menentukan item prioritas tinggi yang dapat dikerjakan berikutnya. Seringkali dilakukan pemecahan setiap fitur yang ditargetkan menjadi serangkaian tugas yang nantinya dikumpulkan bersamaan dengan *product backlog* yang berkaitan, sehingga membentuk *backlog* kedua yang disebut *Sprint Backlog*. Terdapat beberapa pendekatan yang dapat digunakan, salah satunya adalah dengan memilih item *product backlog*, pecahkan item menjadi tugas, dan tentukan apakah item yang dipilih cocok dalam *sprint* (dalam kombinasi dengan item lain yang ditargetkan pada *sprint* yang sama).
3. ***Sprint Execution***. Setelah *sprint planning* diselesaikan dan konten *sprint* berikutnya telah disetujui, maka item-item yang telah dipilih dalam konten *sprint* akan dikerjakan. Tahapan ini disebut sebagai *Sprint Execution*. Pada tahap pengerjaannya tidak terdapat urutan atau aturan dalam bagaimana anggota tim melakukan pekerjaannya, tetapi anggota tim dibebaskan untuk mengatur sendiri cara yang mereka rasa paling baik untuk mencapai tujuan *sprint*.
4. ***Daily Scrum***. Dalam *sprint*, setiap harinya dilakukan *Daily Scrum* dengan waktu yang dibatasi, yaitu 15 menit atau kurang, untuk memberikan kesempatan bagi setiap anggota tim dalam menjawab pertanyaan yang berguna untuk kepentingan anggota tim lainnya. *Daily scrum* berguna untuk mengomunikasikan status item *sprint backlog* di antara anggota tim. Terdapat tiga pertanyaan yang perlu dijawab, yaitu:

- Apa yang saya capai sejak *scrum* terakhir?
- Apa yang saya rencanakan untuk dikerjakan pada *scrum* berikutnya?

- Apa hambatan atau kesulitan yang menghalangi saya untuk membuat kemajuan?

Dari ketiga pertanyaan tersebut, semua orang akan mengetahui gambaran besar tentang apa yang terjadi, termasuk di dalamnya kemajuan dalam mencapai tujuan sprint, rencana untuk hari mendatang, dan masalah yang perlu ditangani.

5. ***Sprint Review***. Pada akhir sprint, dilakukan *Sprint Review*, yaitu kegiatan untuk menginspeksi dan mengadaptasi produk yang sedang dibangun melalui percakapan antara para peserta yang terlibat, yang difokuskan untuk meninjau fitur yang baru saja selesai dibuat / dikembangkan. Orang-orang yang tidak berada dalam tim dapat menyelaraskan upaya pengembangan dan memandu arah. Sedangkan anggota tim mendapatkan apresiasi yang lebih baik dalam sisi bisnis dan pemasaran produk mereka melalui *feedback* tentang konvergensi produk terhadap kepuasan pengguna.
6. ***Sprint Retrospective***. *Sprint Retrospective* adalah aktivitas yang dilakukan setelah *sprint review* dan sebelum *sprint planning* berikutnya. Pada aktivitas ini para peserta akan membahas terkait peningkatan proses berkelanjutan yang diperlukan untuk membantu tim menjadi lebih baik. Selain itu, tim akan melakukan identifikasi dan berkomitmen pada beberapa tindakan perbaikan yang akan dilakukan pada sprint berikutnya. Setelah *sprint retrospective* selesai, seluruh siklus akan diulangi kembali.

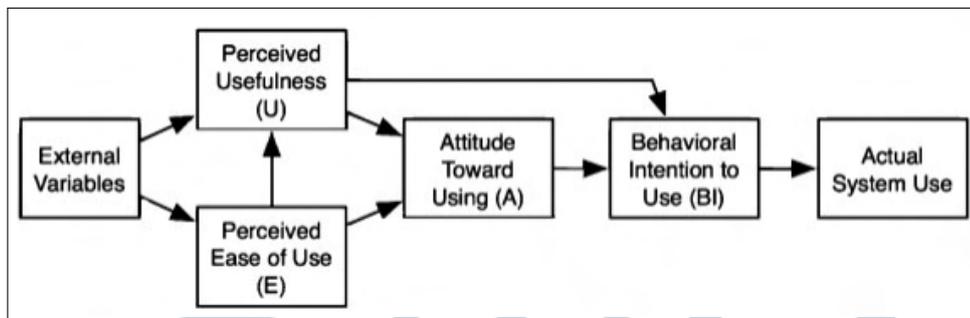
2.6. ***Black Box Testing***

Black box digunakan untuk mewakili perilaku suatu sistem aplikasi ataupun bagian dari suatu sistem aplikasi yang dapat diamati secara eksternal [17]. Pengujian ini memperlakukan perangkat lunak sebagai “kotak hitam”, tanpa pengetahuan tentang kerja internal dan hanya memeriksa aspek fundamental dari sistem [18]. Teknik *Black Box Testing* digunakan untuk memastikan bahwa semua masukan yang dibutuhkan oleh sistem diterima dengan cara yang ditentukan dan memberikan keluaran yang benar. Teknik ini cocok digunakan dimana pengujian didasarkan pada kebutuhan pengguna dan untuk segmen kode yang besar [19]. Keuntungan dari *black box testing* ini adalah pengujian bisa menguji tanpa perlu memiliki pengetahuan pada bidang pemrograman, dan pengujian juga dilakukan

dari sudut pandang pengguna. Selain itu, tidak ada ketergantungan antara pengujian dan pengembang. Pengujian ini juga bisa membantu mengungkap ambiguitas atau ketidakkonsistenan dalam spesifikasi kebutuhan [20].

2.7. *Technology Acceptance Model (TAM)*

Technology Acceptance Model (TAM) merupakan salah satu model penelitian populer yang dikembangkan oleh Davis (1989) untuk memprediksi penggunaan dan penerimaan sistem informasi dan teknologi oleh pengguna individu. Dalam TAM, terdapat dua faktor penentu terpenting terhadap penggunaan suatu sistem. Faktor-faktor tersebut adalah *usefulness*, yaitu kegunaan yang dirasakan oleh pengguna individu bahwa penggunaan sistem aplikasi tersebut bisa meningkatkan kinerja hidup atau pekerjaannya, dan *ease of use*, yaitu kemudahan bagi pengguna dalam penggunaan sistem aplikasi tanpa mengeluarkan banyak usaha. Kedua faktor tersebut berpengaruh pada *attitude* dan *behavioral intention* pengguna individu. *Attitude* berkaitan dengan evaluasi pengguna tentang keinginan menggunakan aplikasi sistem informasi tertentu. Sedangkan *behavioral intention* adalah ukuran kemungkinan seseorang menggunakan aplikasi tersebut [21]. Secara keseluruhan hubungan dalam TAM menurut [22] ditunjukkan pada gambar 2.2.



Gambar 2.2. *Technology Acceptance Model*

2.8. *Skala Likert*

Skala Likert merupakan salah satu metode penskalaan paling fundamental dan populer yang digunakan dalam penelitian ilmu sosial [23]. Metode penskalaan ini dikembangkan tahun 1932 oleh Rensis Likert untuk mengukur sikap, umumnya dengan skala ordinal berkisar 5 atau 7 poin yang digunakan oleh responden untuk menilai seberapa besar mereka setuju atau tidak setuju dengan pernyataan [24].

Variabel yang akan diukur dijabarkan menjadi indikator variabel sebagai titik tolak untuk menyusun item-item instrumen. Jawaban dari item instrumen ini akan mempunyai gradasi nilai dari sangat positif sampai sangat negatif [25]. Gradasi nilai dengan skala ordinal 5 poin dapat diuraikan sebagai berikut:

1. Sangat setuju (skor 5)
2. Setuju (skor 4)
3. Netral (skor 3)
4. Tidak setuju (skor 2)
5. Sangat tidak setuju (skor 1)

Persentase penerimaan dari skala likert dapat dihitung dengan rumus sebagai berikut [25]:

$$P = \frac{(SS * 5) + (S * 4) + (N * 3) + (TS * 2) + (STS * 1)}{(JumlahPertanyaan * 5 * n)} * 100\% \quad (2.1)$$

dimana,

P = persentase penerimaan

n = jumlah responden

SS = jumlah jawaban sangat setuju

S = jumlah jawaban setuju

N = jumlah jawaban netral

TS = jumlah jawaban tidak setuju

STS = jumlah jawaban sangat tidak setuju

Hasil perhitungan yang diperoleh dari rumus 2.1 dapat dikelompokkan dalam interval dengan persentase 0% hingga 100% dengan jarak antar interval adalah 20%.

Tabel 2.1. Interval Persentase Penerimaan Skala *Likert*

No.	Keterangan	Interval
1.	Sangat Tidak Setuju	0% - 19.99%
2.	Tidak Setuju	20% - 39.99%
3.	Netral	40% - 59.99%
4.	Setuju	60% - 79.99%
5.	Sangat Setuju	80% - 100%

