

BAB II

LANDASAN TEORI

2.1 Human Resource Management

Menurut (Dessler, 2016), *Human Resource Management* adalah proses dari mengakuisisi, melatih, menilai, dan memberikan kompensasi kepada karyawan, serta untuk memastikan hubungan kerja antar karyawan, kesehatan, keamanan, dan masalah keadilan karyawan. Banyak penulis saling berpendapat dan setuju bahwa terdapat 5 *basic functions* dalam melakukan proses manajemen, diantaranya adalah *planning, organizing, staffing, leading, dan controlling*.

1. *Planning*

Tahap perencanaan atau *planning* adalah menentukan tujuan dan standar, mengembangkan peraturan dan prosedur serta mengembangkan ide dan memprediksinya.

2. *Organizing*

Memberikan masing-masing bawahan dengan tugas tertentu, mendirikan departemen, mendelegasikan wewenang kepada bawahan, serta memastikan komunikasi dapat berjalan tanpa adanya masalah.

3. *Staffing*

Tujuan dari *staffing* adalah menentukan tipe orang yang memang diperlukan untuk direkrut, melakukan *recruiting* untuk memenuhi bagian yang kosong dalam perusahaan, menentukan kinerja standar karyawan, mengkompensasikan karyawan, mengevaluasi kinerja,

melakukan konseling terhadap karyawan, serta untuk melakukan *training* dan *developing* karyawan.

4. *Leading*

Tahap *leading* bertujuan untuk memastikan bahwa pekerjaan yang dilakukan dapat selesai dan untuk memotivasi serta mempertahankan moral para karyawan.

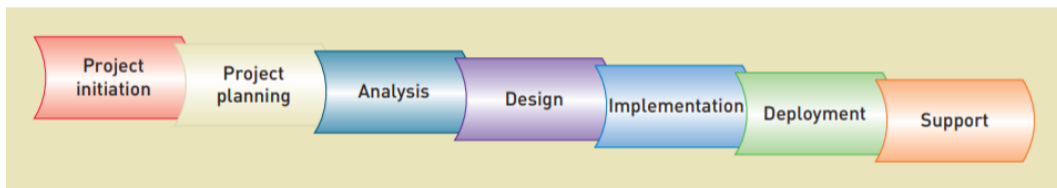
5. *Controlling*

Dalam tahap *controlling* ini dilakukan untuk mengkonfigurasi standar seperti *sales quotas*, *quality standards*, atau *production level*. Dan melakukan pengecekan terhadap kinerja yang sudah dilakukan dengan standar yang sudah ada.

2.2 *Systems Development Lifecycle (SDLC)*

Menurut (Satzinger et al., 2016), *Systems Development Life Cycle* atau yang sering disebut sebagai SDLC adalah keseluruhan aktivitas yang dilakukan dari proses membuat, menyebarkan, menggunakan, dan memperbaiki sebuah sistem informasi. Dalam mengembangkan sebuah sistem informasi, SDLC memiliki 5 tahapan yang dapat digunakan sebagai pedoman, tahapan-tahapan tersebut antara lain adalah *Project Planning Phase*, *Analysis Phase*, *Design Phase*, *Implementation Phase*, dan *Support Phase*.

Ilustrasi mengenai 5 tahapan dari proses pengembangan SDLC dapat dilihat pada gambar 2.1.



Gambar 2. 1 Tahap pengembangan SDLC

Sumber : (Satzinger et al., 2016)

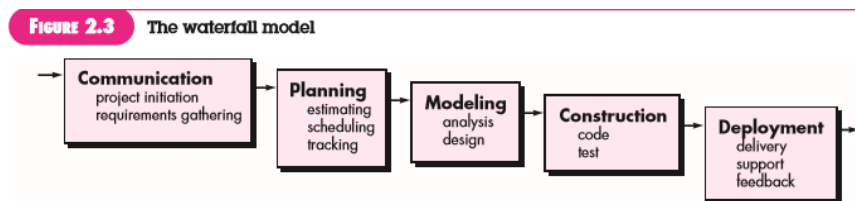
Berdasarkan dari gambar 2.1 mengenai tahapan-tahapan *predictive approaches* dalam SDLC yang dijelaskan oleh (Satzinger et al., 2016), berikut adalah penjelasan untuk setiap tahap yang ada di dalam *predictive approaches* SDLC tersebut:

1. *Project Initiation phase* adalah sekumpulan aktivitas yang dilakukan untuk mengidentifikasi masalah dan membuat persetujuan terhadap *client* untuk mengembangkan sebuah sistem baru.
2. *Project planning phase* dilakukan untuk mengidentifikasi jangkauan atau cakupan yang ada pada sistem baru, memastikan bahwa proyek layak dikerjakan, dan untuk mengembangkan jadwal, rencana sumber daya, dan *budget* untuk membiayai proyek.
3. *Analysis phase* adalah untuk memahami dokumen mengenai detail dari kebutuhan bisnis dan proses *requirement* dari sistem baru.
4. *Design phase* adalah untuk mendesain solusi dari sistem berdasarkan dari kebutuhan yang telah didefinisikan dan keputusan yang telah dilakukan pada saat tahapan *analysis*.

5. *Implementation phase* adalah untuk membangun, menguji, dan menginstal sebuah sistem informasi yang dapat digunakan oleh *user* dan dapat memberikan manfaat seperti yang diharapkan dari penggunaan sistem yang baru.
6. *Deployment phase* merupakan aktivitas untuk melakukan instalasi dan konfigurasi untuk memasukkan sistem agar sistem tersebut dapat berjalan atau beroperasi.
7. *Support phase* dilakukan untuk memastikan bahwa sistem dapat berjalan secara produktif dan dapat berjalan dengan baik sampai dengan masa yang akan datang.

2.3 Metode Waterfall

Metode Waterfall sering disebut sebagai *the classic life cycle*, yang menyarankan pendekatan secara sistematis dan berurutan untuk melakukan pengembangan perangkat lunak yang dimulai dengan menspesifikasi kebutuhan *customer* dan berkembang melalui perencanaan, pemodelan, konstruksi, dan penyebaran yang berpuncak pada dukungan berkelanjutan dari *ongoing support* sampai dengan *completed software* (Suryan, 2014).



Gambar 2. 2 Proses Waterfall Model

Sumber : (Suryan, 2014)

2.4 Metode Prototyping

Menurut (Suryan, 2014), metode Prototyping sering digunakan pada saat seorang *customer* atau pemegang saham mendefinisikan tujuan umumnya pada suatu *software*, namun tidak menjelaskan secara detail mengenai kebutuhan untuk *function* dan fiturnya. Metode Prototyping dimulai dengan melakukan komunikasi, dimana para *developer* bertemu dengan para target sistem atau pemegang saham untuk mendefinisikan objektif secara keseluruhan yang akan dibutuhkan oleh sistem, dan untuk mengidentifikasi kebutuhan sistem yang masih belum terwujudkan.

Prototyping dapat menghasilkan versi kerja awalan yang akan dibangun dengan cepat berdasarkan dari sistem informasi yang dibutuhkan atau biasa disebut sebagai *prototype*. Masukan atau *feedback* dari pengguna sangat penting dari setiap proses yang ada di dalam metode Prototyping, karena dengan metode ini melibatkan pengguna dalam setiap proses pengembangan dalam memeriksa model, *output*, *input*, *user interface*, serta proses secara akurat. Dengan kata lain, metode Prototyping dilakukan untuk melakukan validasi terhadap kebutuhan *user* dengan sistem yang akan dihasilkan nantinya (B.Shelly & J.Rosenblatt, 2012).

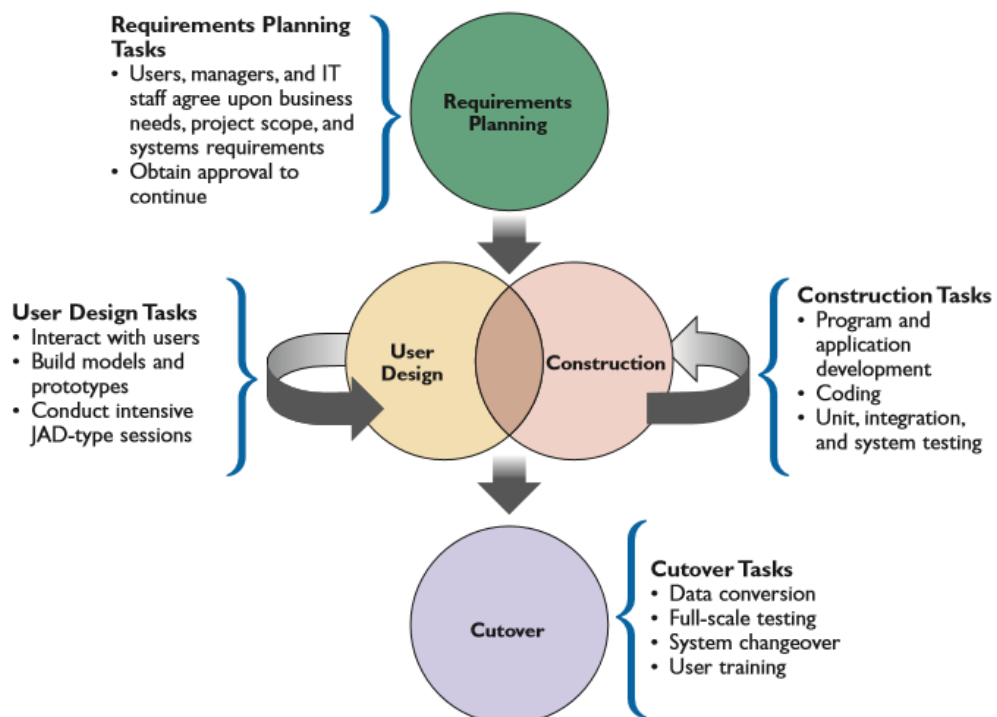


Gambar 2. 3 Proses Prototyping

Sumber : (B.Shelly & J.Rosenblatt, 2012)

2.5 Metode Rapid Application Development (RAD)

Menurut (B.Shelly & J.Rosenblatt, 2012), Rapid Application Development atau sering disebut sebagai RAD adalah teknik berbasis tim yang dapat mempercepat proses pengembangan sistem informasi dan dapat memproduksi sebuah sistem informasi yang berkualitas. Banyak dari perusahaan yang menggunakan metode RAD untuk mengurangi *cost* dan mempercepat waktu pengembangan sistem serta untuk meningkatkan peluang kesuksesan dari sistem. RAD juga merupakan metodologi yang lengkap dengan menggunakan 4 tahapan *life-cycle* yang berasal dari tahapan tradisional SDLC.



Gambar 2. 4 Proses Metode Rapid Application Development

Sumber : (B.Shelly & J.Rosenblatt, 2012)

Berdasarkan pada gambar 2.4, berikut adalah penjelasan mengenai proses dari metode RAD:

1. *Requirements Planning* merupakan tahapan mengkombinasikan elemen dari perencanaan sistem dengan tahapan analisis sistem di dalam SDLC. *Users, managers,* dan anggota staf IT saling berdiskusi untuk menyetujui mengenai kebutuhan bisnis, cakupan proyek, *constraint,* dan *systems requirements.* Tahap *requirement planning* berakhir pada saat tim menyetujui masalah-masalah utama dan mendapatkan otorisasi manajemen untuk melanjutkan proyek.
2. *User Design* merupakan interaksi *users* dengan *systems analyst* dan mengembangkan model serta *prototype* untuk mewakili keseluruhan proses, *output* dan *input* yang telah diberikan.
3. *Construction,* tahapan ini berfokus pada program dan tugas pengembangan aplikasi yang hampir sama dengan SDLC. Di dalam RAD, *users* tetap berpartisipasi dan tetap bisa memberikan saran perubahan atau peningkatan mengenai sistem yang dibuat atau laporan yang sedang dikembangkan.
4. *Cutover* adalah tahapan yang menyerupai tahap akhir yang ada di dalam SDLC tahap implementasi, seperti *data conversion,* pengujian, pergantian menggunakan sistem baru, dan melakukan pelatihan kepada *user.* Sebagai hasilnya, sistem baru telah berhasil dibangun, diberikan, dan diletakkan ke dalam operasi lebih cepat.

2.6 Unified Modeling Language

Unified Modelling Language (UML) merupakan sebuah *object-based* model dan *diagramming technique* yang cukup banyak untuk dapat membuat sebuah model dari berbagai macam jenis proyek pengembangan sistem dari tahap analisis menuju tahap desain (Dennis et al., 2015)

Menurut (B.Shelly & J.Rosenblatt, 2012) *Unified Modeling Language* merupakan metode yang biasanya digunakan untuk membuat sebuah visualisasi dan dokumentasi mengenai sistem *software design* yang akan dikembangkan. UML menggunakan konsep desain berorientasi objek, namun tidak tergantung pada bahasa pemrograman tertentu dan dapat digunakan untuk mendefinisikan kebutuhan dan proses bisnis secara umum.

Untuk pembuatan desain mengenai sistem yang akan dibuat, UML menyediakan berbagai macam *graphical tools* yang dapat digunakan. Contoh dari *graphical tools* yang dapat digunakan dalam pembuatan desain sistem tersebut adalah *use case diagrams*, *activity diagrams*, *sequence diagrams*, dan *class diagrams*.


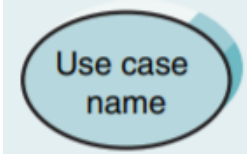
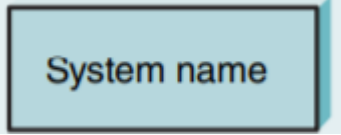
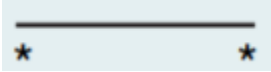
2.6.1 Use Case Diagram

Menurut (Dennis et al., 2015), *use case diagram* adalah ringkasan visual mengenai beberapa *use case* yang saling berhubungan pada sebuah sistem atau subsistem. *Use case diagram* juga dibuat untuk membantu tim pengembang sistem untuk dapat lebih mengerti mengenai tahapan-tahapan apa saja yang diperlukan untuk mencapai tujuan *user*. Setelah dibuat, *use case* dapat digunakan untuk mengidentifikasi *functional requirement* secara

detail kepada sistem yang baru. Dalam pembuatan *use case*, seorang pengembang sistem dapat memulainya dengan melakukan mengulas informasi yang sudah dikumpulkan pada saat tahapan *requirement gathering*, dengan tujuan untuk mengidentifikasi aktor dan fungsi atau proses yang nanti akan berjalan.

Berikut merupakan simbol-simbol yang dapat digunakan dalam membuat *use case diagrams*:

Tabel 2. 1 Simbol Use Case Diagram

Simbol	Desikripsi
 <p>Actor role name</p>	<p>Aktor merupakan seseorang atau sebuah sistem yang berinteraksi dan mendapatkan sebuah nilai dari sistem. Sebuah <i>actor</i> bukanlah <i>user</i> secara spesifik, namun merupakan sebuah posisi dimana <i>user</i> dapat berinteraksi dengan sistem.</p>
 <p>Use case name</p>	<p><i>Use case</i> digambarkan dengan bentuk oval yang merupakan proses utama yang akan dilakukan dalam sistem dengan menggunakan aktor di dalamnya dan dapat diisikan label dengan kata kerja yang deskriptif.</p>
 <p>System name</p>	<p><i>System boundary</i> merupakan sebuah kotak yang mewakili sistem secara jelas dengan menggambarkan bagian diagram mana yang eksternal atau internal.</p>
	<p><i>Association relationship</i> merupakan sebuah penghubung antara <i>actor</i> dengan satu atau lebih <i>use case</i> agar dapat berinteraksi satu sama lain.</p>

Sumber : (Dennis et al., 2015)



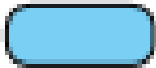
2.6.2 Activity Diagram


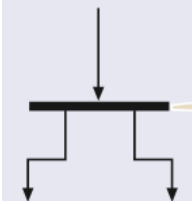
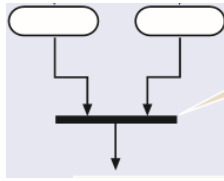

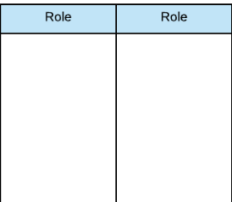
Menurut (B.Shelly & J.Rosenblatt, 2012), *Activity diagram* merupakan kumpulan dari *flowchart* horizontal yang menunjukkan bagaimana sebuah *action* atau *event* bisa terjadi. *Activity diagram* juga menunjukkan secara terurut mengenai bagaimana *action* itu terjadi dan mengidentifikasi hasilnya.

(Dennis et al., 2015) mengatakan bahwa, *activity diagram* menggambarkan berbagai aktivitas pengguna atau sistem, seseorang yang melakukan aktivitas dengan gambar *sequential flow* untuk setiap aktivitasnya.

Berikut adalah simbol-simbol yang digunakan dalam membuat sebuah *activity diagram*:

Tabel 2. 2 Simbol Activity Diagram

Simbol	Deskripsi
	<i>Initial node</i> atau <i>start point</i> menandakan bahwa <i>activity diagram</i> akan dimulai, yang digambarkan dengan simbol lingkaran hitam.
	<i>Final node</i> atau <i>endpoint</i> merupakan tanda yang menunjukkan akhir dari suatu aktivitas dalam <i>activity diagrams</i> . <i>Final node</i> digambarkan dengan simbol lingkaran dengan lingkaran hitam di dalamnya.
	<i>Action nodes</i> digunakan untuk menggambarkan aktivitas atau proses yang ada di dalam <i>activity diagram</i> .

Simbol	Deskripsi
	<p>Simbol tersebut digunakan untuk menghubungkan antara setiap <i>action nodes</i> yang ada di dalam <i>activity diagram</i>.</p>
	<p><i>Synchronization bar split</i> merupakan simbol yang digunakan untuk memisahkan sebuah aktivitas dalam 2 atau lebih <i>nodes</i>.</p>
	<p><i>Synchronization bar join</i> merupakan simbol yang digunakan untuk menggabungkan aktivitas setelah <i>synchronization bar split</i> dilakukan.</p>
	<p>Simbol <i>decision</i> digunakan untuk menggambarkan sebuah pilihan yang tersedia dan bagaimana hasil dari setiap pilihan yang ada.</p>
	<p><i>Swimlane</i> merupakan area persegi panjang yang ada pada <i>activity diagram</i> yang mewakili aktivitas dari suatu posisi tertentu.</p>

Sumber : (Dennis et al., 2015)

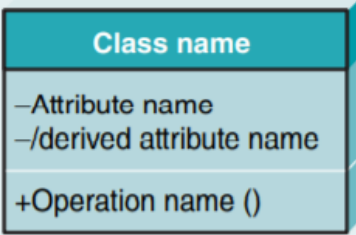
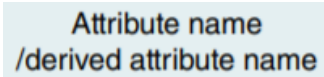
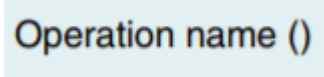
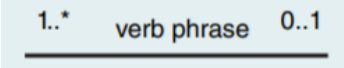
2.6.3 Class Diagrams

Menurut (Dennis et al., 2015), *class diagrams* merupakan model logika yang berkembang menjadi model fisik dan dapat menjadi sebuah sistem informasi yang dapat berfungsi dengan baik. Dalam *class diagrams*,

setiap *class* muncul dengan simbol persegi dengan memiliki nama di atasnya dan mempunyai atribut dan metode *class* didalamnya. Class diagrams memiliki arti lain yaitu sebuah *static model* yang dapat mendukung tampilan statis dari sistem yang masih dalam tahap pengembangan. Di dalam *class diagram* ditampilkan *classes* dan hubungan antara *classes* tersebut dengan sebuah sistem.

Berikut simbol-simbol yang dapat digunakan dalam pembuatan *class diagram*:

Tabel 2. 3 Simbol *Class Diagrams*

Simbol	Deskripsi
	<p><i>Class</i> dapat berupa seseorang, tempat, atau sesuatu barang dimana informasi tersebut harus diambil dan disimpan oleh sistem. <i>Class</i> juga mempunyai daftar atribut di bagian tengah dan daftar operasi di bagian bawah kotak tersebut.</p>
	<p><i>Attribute</i> digunakan sebagai properti untuk menggambarkan keadaan dari suatu objek. <i>Attribute</i> juga dapat diturunkan dari <i>attribute</i> lain, ditunjukkan dengan menempatkan garis miring di depan nama <i>attribute</i> yang dituju.</p>
	<p><i>Method</i> merupakan sebuah <i>actions</i> atau <i>functions</i> dari sebuah <i>class</i> yang dapat dijalankan. <i>Method</i> dapat diklasifikasikan sebagai <i>constructor</i>, <i>query</i>, atau <i>update operation</i>.</p>
	<p><i>Association</i> merupakan sebuah hubungan antara banyak <i>classes</i> atau dengan sebuah <i>class</i> itu sendiri.</p>

Sumber : (Dennis et al., 2015)

2.7 XAMPP

Menurut (Robertson, 2011), XAMPP adalah solusi *software* yang bersifat *free and open source software* (FOSS) yang merupakan *cross platform* web *solution stack* yang dibuat oleh Apache. Dalam *software* XAMPP berisikan hal-hal sebagai berikut:

1. Apache HTTP Server
2. MariaDB *Database formerly* yang dikenal sebagai MySQL
3. *Interpreter* untuk PHP *Script*
4. *Interpreter* untuk Perl *Script*

Software XAMPP dapat digunakan pada sistem operasi Windows 32-bit ataupun 64-bit dan Linux. XAMPP dapat di unduh langsung dari *website* resminya dengan mengunjungi <https://www.apachefriends.org>.

2.8 HTML

HTML atau *Hypertext Markup Language* merupakan standar internasional yang ditetapkan oleh W3C dengan maksud untuk memastikan bahwa setiap halaman web yang ditulis sudah sesuai dengan standar dan akan tampak sama di seluruh web *browser*. Pada dasarnya HTML digunakan untuk mengkonstruksi *layout* dari sebuah halaman web dengan menggunakan *code* ataupun *instruction tags*, seperti membuat judul, paragraf, gambar, video, dan lain-lain (Chaffey & Ellis-Chadwick, 2016).

2.9 CSS

Cascading Style Sheets atau yang sering disebut sebagai CSS merupakan sebuah *code* yang dapat digunakan oleh web *designer* untuk menentukan *style* dari *website* mereka, seperti mengubah *font*, *spacing*, dan warna ke dalam *Hypertext Markup Language* (HTML) yang sudah di didefinisikan. CSS dapat menyederhanakan dalam pembuatan dan pemeliharaan web karena *style* yang dapat diubah dengan mudah (Chaffey & Ellis-Chadwick, 2016).

2.10 Javascript

Javascript dapat digunakan untuk membuat web *programming* menjadi lebih baik dan dapat meningkatkan *user experience* seseorang ketika menggunakan web tersebut. Program Javascript sudah bisa dijalankan langsung pada web *browser user* tanpa melakukan instalasi (Steven & Janet, 2013).

2.11 PHP

PHP merupakan salah satu bahasa pemrograman dari sisi server (*server-side*) yang dirancang untuk membangun sebuah aplikasi web yang dinamis. Ketika dipanggil dari web *browser*, program yang ditulis dengan PHP akan di-parsing ke dalam web *browser* oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke *browser*. Karena proses dari program PHP juga dilakukan pada lingkungan web server (Nixon, 2014).

2.12 Database

2.12.1 MySQL

Menurut (Nixon, 2014), MySQL merupakan sebuah sistem basis data yang menggunakan *Structured Query Language* untuk menyimpan data dalam sebuah sistem komputer dan mengorganisir data tersebut supaya informasi dapat dicari dan mudah untuk diakses. Bahasa pemrograman yang digunakan oleh MySQL menggunakan bahasa Inggris dan beberapa bahasa pemrograman *database* lainnya seperti Oracle dan Microsoft SQL Server, dimana bahasa tersebut didesain agar mampu memproses *request* dari *database* dengan berbagai macam *command* yang tersedia. *Command* yang biasanya digunakan dalam memproses MySQL adalah seperti *create*, *read*, *update*, dan *delete*.

2.12.2 PhpMyAdmin

Menurut (Delisle, 2012), PhpMyAdmin adalah sebuah aplikasi web *open source* yang dituliskan dengan *syntax* PHP, dimana didalamnya berisikan XHTML, CSS, dan Javascript *client code*. PhpMyAdmin memberikan web *interface* secara keseluruhan untuk mengatur *database* MySQL dan juga telah diakui sebagai aplikasi yang terkemuka pada bidang *database*.

Menurut (Standsyah & N.S Restu, 2017), PhpMyAdmin adalah sebuah aplikasi atau perangkat lunak (*open source*) yang ditulis dalam bahasa pemrograman PHP untuk menangani administrasi *database* MySQL melalui jaringan lokal maupun internet. PhpMyAdmin mendukung berbagai

operasi MySQL, seperti *fields*, relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain.

2.13 Penelitian terdahulu

No	Judul Penelitian	Nama Jurnal	Hasil Penelitian	Kontribusi di dalam penelitian
1	Rancang Bangun Aplikasi Manpower Request Bagian Recruitment pada PT Dynapack Asia. (Listiani & Natalia, 2019)ra	ULTIMA InfoSys Vol x, 2019	Hasil yang berhasil dibuat pada jurnal tersebut adalah sebuah aplikasi yang dapat membuat <i>manpower request</i> berbasis web bagian <i>recruitment</i> untuk divisi <i>Human Resources</i> pada PT Dynapack Indonesia	Penelitian ini bisa menjadi referensi dalam membuat fitur <i>recruitment</i> pada sistem HR yang akan dikembangkan nantinya
2	Human Resources Information System Design for Small, Micro and Medium Enterprises. (Wibawa et al., 2018)	IOP Conference Series: Materials Science and Engineering Vol 407, 2018	Penelitian ini menghasilkan sebuah sistem yang dapat menjalankan proses dari <i>Human Resource</i> untuk perusahaan <i>micro, small</i> dan <i>medium</i>	Penelitian ini menyediakan contoh <i>database</i> dan hasil akhir sistem yang dapat dijadikan referensi untuk membangun sistem HR nantinya.
3	Design and Implementation of Human Resource Information Systems Based on MVC a Case Study Vocational Education In Iraq. (Tariq et al., 2016)	International Journal of u- and e-Service, Science and Technology , Vol 9, 2016	Hasil akhir dan implementasi sistem <i>Human Resource Information System</i> dengan menggunakan MVC model untuk <i>Vocational Education</i> yang ada di negara Iraq	Penelitian tersebut dapat dijadikan referensi dari segi <i>Unified Modelling Language</i> dan hasil akhir sistem yang akan membantu dalam pembuatan sistem HR nantinya.