

# I. PENDAHULUAN

## 1.1 Latar Belakang

PT Darma Digital Solution adalah perusahaan dengan izin operasional sebagai Perseroan Swasta Nasional (PT/Perseroan Terbatas) yang mengkhususkan diri untuk bergerak dibidang Teknologi Informasi dan Komputer. PT Darma Digital Solution atau yang sering disingkat DDS ini memiliki kantor di gedung Darmapala, Jalan Pagar Alam No.61, Kedaton, Kecamatan. Tanjung Karang Barat., Kota Bandar Lampung. Kegiatan yang dilakukan oleh DDS saat ini dibagi menjadi dua (2) yaitu, penyedia jasa (*bussiness application development, web development, mobile development, corporate branding, networking* dan konsultan IT) dan pengembangan produk (sistem manajemen aset, sistem informasi akademik, dll). Beragam kegiatan yang dilakukan oleh PT DDS menyebabkan banyaknya proyek yang tidak di manajemen dengan baik sehingga proses perencanaan dan pengembangan menjadi terlambat dari waktu yang telah ditentukan.

Manajemen merupakan strategi pemanfaatan pikiran dan tenaga orang lain dalam melaksanakan suatu aktivitas dengan tujuan yang telah ditentukan. Didalam manajemen, terdapat berbagai teknik seperti, mempengaruhi mengawasi, kepemimpinan salam mengarahkan, dan mengorganisasikan segala komponen dalam tercapainya tujuan (Sulfemi, 2018).

Proyek merupakan suatu kegiatan pekerjaan yang dilakukan atas dasar permintaan dari seorang pemilik pekerjaan atau pebisnis untuk mencapai tujuan tertentu. Dilaksanakan oleh pelaksana pekerjaan sesuai perintah dari pemilik proyek atau pebisnis dan spesifikasi yang ada. Dalam melaksanakan proyek, pelaksana proyek dan pemilik proyek mempunyai kewajiban dan hak yang harus dilaksanakan sesuai dengan batas waktu yang telah ditentukan (Ediman Manik & Melinta, 2017).

Manajemen Proyek merupakan suatu tata cara untuk mengorganisir dan menyelesaikan suatu proyek. Manajemen proyek dapat digunakan pada jenis proyek apapun yang berskala kecil ataupun besar. Mengerjakan proyek secara tepat waktu merupakan tujuan yang harus dicapai. Namun pada kenyataannya hal tersebut sangat sulit dicapai karena kendala yang menyebabkan keterlambatan. Keterlambatan proyek menyebabkan pembengkakan biaya yang tidak sedikit, belum lagi jika terdapat sanksi apabila pengerjaannya tidak selesai tepat pada waktunya.

PT Darma Digital Solution saat ini masih melakukan *monitoring* proyek dengan mengandalkan *live chat* menggunakan aplikasi *Whatsapp* untuk diskusi antar karyawan dalam mengetahui perkembangan proyek yang sedang dikerjakan. Hal tersebut dirasa kurang efisien dalam pengelolaan proyek-proyek yang rata-rata dikerjakan bersama dengan tim. Oleh karena itu, berdasarkan masalah yang telah diuraikan maka penulis membuat Aplikasi Manajemen Proyek Sederhana Berbasis *Web* Menggunakan *Framework* *Laravel*. Aplikasi ini berupa *monitoring* kinerja tim ataupun pekerjaan pribadi. Keunggulan aplikasi ini pekerjaan bisa di pantau langsung oleh pimpinan atau *client*.

## 1.2 Tujuan

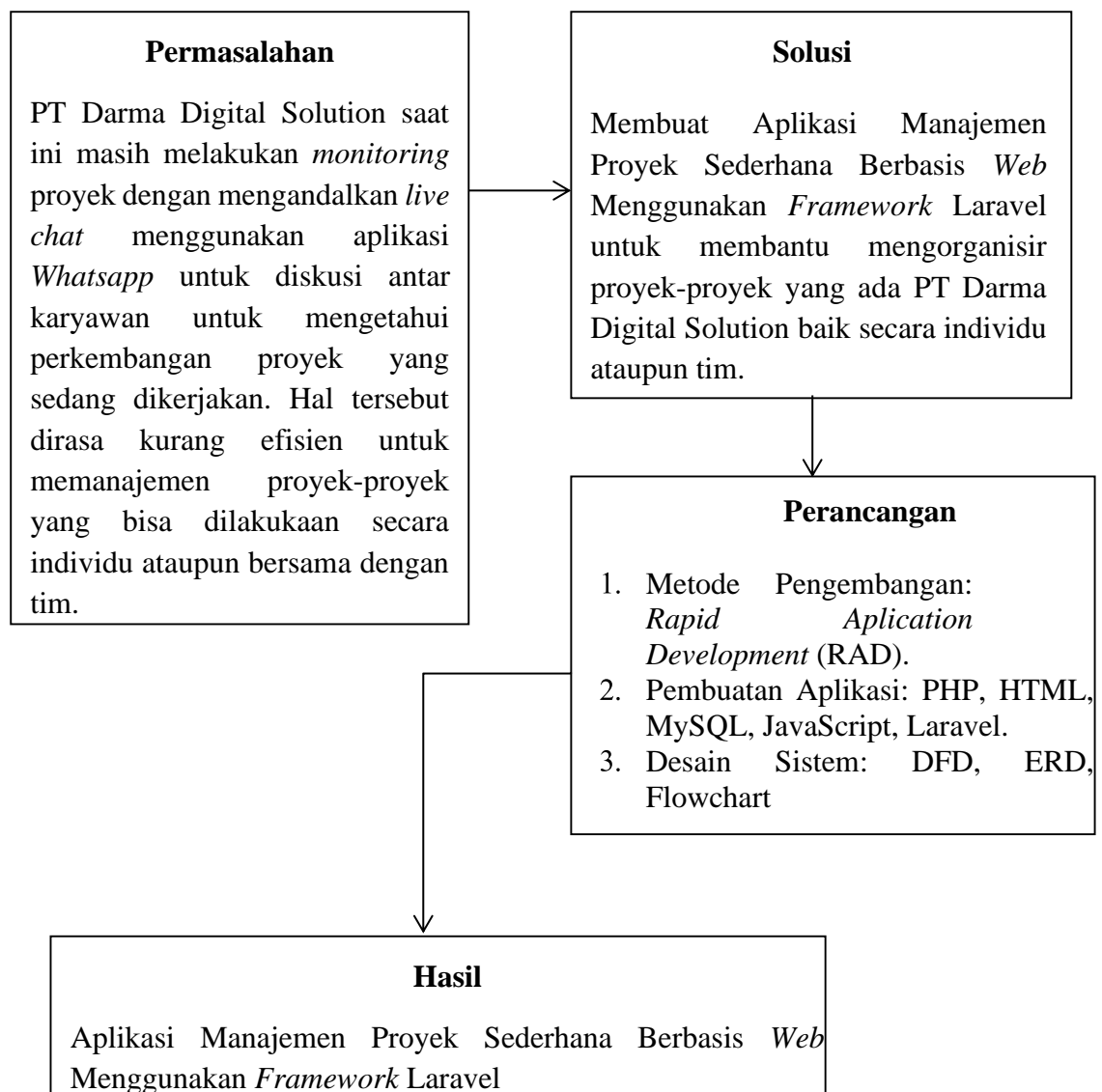
Tujuan dari pembuatan tugas akhir ini adalah membangun Aplikasi Manajemen Proyek Sederhana Berbasis *Web* Menggunakan *Framework* *Laravel* yang dapat proyek-proyek yang ada PT Darma Digital Solution baik secara individu ataupun tim.

## 1.3 Kerangka Pemikiran

Kerangka pemikiran adalah suatu diagram yang menjelaskan secara garis besar alur logika berjalannya sebuah penelitian (*reseach question*), dan mempresentasikan suatu himpunan dari beberapa konsep serta hubungan diantara konsep-konsep tersebut. Masalah yang muncul dari bagian monitoring proyek di PT Darma Digital Solution yaitu *monitoring* masih mengandalkan *live chat* di aplikasi *Whatsapp* untuk diskusi antar karyawan untuk mengetahui perkembangan

proyek. Solusi yang dapat diberikan yaitu dengan membangun “Aplikasi Manajemen Proyek Sederhana Berbasis *Web* Menggunakan *Framework* Laravel”.

Pembuatan modul aplikasi dilakukan dengan metode pengembangan sistem *Rapid Application Development* (RAD) yang terdiri dari analisis kebutuhan sistem, desain perancangan sistem, dan implementasi. Berdasarkan latar belakang dapat disusun suatu kerangka pemikiran yang disajikan dalam Gambar 1.



Gambar 1. Kerangka Pemikiran

#### **1.4 Kontribusi**

Adanya Aplikasi Manajemen Proyek Sederhana Berbasis *Web* Menggunakan *Framework* Laravel ini pengelolaan proyek-proyek baik dari perencanaan, pengorganisasian, dan pengawasan dapat dilakukan secara efisien dan kendala-kendala yang ada pada PT Darma Digital Solution dapat terselesaikan.

## II. TINJAUAN PUSTAKA

### 2.1 Aplikasi dan Manajemen

Aplikasi merupakan program yang siap digunakan dan dibuat untuk tujuan melakukan suatu fungsi bagi pengguna aplikasi dan penggunaan aplikasi lain seperti aplikasi berbasis *web* yang dapat diakses penggunanya melalui fungsi *hyperlink* dan digunakan untuk sasaran yang akan dituju (Fansuri, 2015).

Manajemen merupakan strategi pemanfaatan pikiran dan tenaga orang lain dalam melaksanakan suatu aktivitas dengan tujuan yang telah ditentukan. Didalam manajemen, terdapat berbagai teknik seperti, mempengaruhi mengawasi, kepemimpinan salam mengarahkan, dan mengorganisasikan segala komponen dalam tercapainya tujuan (Sulfemi, 2018).

### 2.2 Proyek

Proyek merupakan suatu kegiatan pekerjaan yang dilakukan atas dasar permintaan dari seorang pemilik pekerjaan atau pebisnis untuk mencapai tujuan tertentu. Dilaksanakan oleh pelaksana pekerjaan sesuai perintah dari pemilik proyek atau pebisnis dan spesifikasi yang ada. Dalam melaksanakan proyek, pelaksana proyek dan pemilik proyek mempunyai kewajiban dan hak yang harus dilaksanakan sesuai dengan batas waktu yang telah ditentukan (Ediman Manik & Melinta, 2017).

### 2.3 Komponen Pembangun *Website*

Membangun sebuah *website* terdiri dari beberapa komponen penting yang diperlukan dalam membangun sebuah website, yaitu :

#### a. PHP

PHP merupakan bahasa pemrograman yang memiliki fungsi dalam membangun suatu *web* dinamis. PHP berjalan di sisi *server*, karenanya PHP disebut sebagai *Server Side Scripting*. Perintah PHP dijalankan pada

sisi server yang kemudian dikirim ke *browser*. PHP digunakan pada berbagai OS seperti Windows, Mac OS dan unix (Nugraha dkk., 2017).

**b. *Framework***

*Framework* adalah struktur konseptual dasar yang digunakan dalam menangani atau memecahkan suatu masalah yang kompleks. *Framework* merupakan kerangka kerja atau wadah dari sebuah *website* yang akan dibangun. Untuk menyingkat waktu dalam membuat *website* maka digunakan kerangka tersebut (Naista, 2017). Laravel merupakan *framework* berbasis PHP yang bersifat *open source*, serta menggunakan konsep *model-view-controller*. Laravel berada dibawah lisensi MIT dengan Github sebagai tempat berbagi kode dalam menjalankannya (Naista, 2017).

**c. *Database***

*Database* adalah kumpulan data yang dikelola berdasarkan ketentuan tertentu yang saling berhubungan dan merupakan bagian penting pada sistem aplikasi, semua fitur yang ada, menggunakan *database* untuk melakukan fungsinya (Safaat, 2015). Untuk mengelola database diperlukan DBMS (*Database Management System*) atau sistem manajemen basis data yang merupakan sistem perangkat lunak dalam mengelola, mengakses, dan mengontrol data secara praktis dan efisien (Serepia dkk., 2019). Salah satu DBMS yang banyak digunakan yaitu MySQL merupakan sistem perangkat lunak dalam manajemen dan mengelola *database* dengan bahasa SQL yang bersifat *open source*. MySQL diciptakan oleh Tcx dan menjadi basis data yang digemari *programmer web* karena menyimpan data dengan kuat dan stabil (Lavarino, 2016).

## 2.4 RAD

*Rapid Application Development* (RAD) menurut (Aswati & Siagian, 2016) adalah proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan, dengan waktu yang singkat. Dengan menggunakan metode *iteratif* (berulang) dalam pengembangannya, dimana model bekerja sistem dikonstruksikan diawal tahap pengembangan, hal ini dengan tujuan menetapkan

kebutuhan (*requirement*) pengguna kemudian disingkirkan. Model RAD mempunyai 3 tahapan sebagai berikut :

a. Rencana Kebutuhan (*Requirement Planning*)

*Analyst* dan *user* melakukan pertemuan dengan maksud mengidentifikasi tujuan dari kebutuhan informasi dan sistem dalam mencapai tujuan. Pada tahapan ini adalah hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.

b. Proses Desain Sistem (*Design System*)

Tahapan ini adalah keaktifan *user* yang menentukan dalam mencapai tujuan, karena pada proses ini dilakukan proses desain dan melakukan perbaikan apabila terdapat ketidaksesuaian desain antara *analyst* dan *user*. *User* dapat langsung berkomentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahap ini adalah spesifikasi *software* yang meliputi struktur data dan organisasi sistem secara umum.

c. Implementasi (*Implementation*)

Pada tahap ini *programmer* yang mengembangkan desain suatu program yang sudah disetujui oleh analis dan *user*. Sebelum diaplikasikan pada suatu organisasi, dilakukan proses pengujian pada program tersebut, apakah terdapat kesalahan atau tidak. Pada tahap ini *user* akan memberikan tanggapan pada sistem yang sudah dibuat dan mendapat persetujuan mengenai sistem tersebut.

## 2.5 *Blackbox Testing*

*Blackbox testing* atau pengujian kotak hitam menurut (Pressman, 2012) adalah suatu pengujian yang dilakukan hanya untuk kebutuhan fungsional sistem atau nama lainnya yaitu pengujian *behavior* (prilaku). Merupakan pengujian yang didapat pada serangkaian kondisi *input* dengan keseluruhan yang dapat menjalankan seluruh persyaratan fungsional kepada suatu program. Menurut (Jaya, 2018) penggunaan metode *blackbox testing* memiliki kelebihan dan kekurangan sebagai berikut :

1. Keuntungan
  - a. Pengujian dilihat dari sudut pandang pengguna
  - b. Tester dan *programmer* saling bergantung satu sama lain
  - c. Penguji tidak perlu mengetahui bahasa pemrograman tertentu
2. Kekurangan
  - a. Memiliki kemungkinan dalam pengulangan tes yang sudah dilakukan oleh *programmer*
  - b. Uji kasus sulit tanpa adanya spesifikasi yang jelas
  - c. Pada beberapa bagian back end tidak dilakukan pengujian sama sekali.

## 2.6 Visual Studio Code

Visual Studio Code merupakan sebuah teks editor yang handal dan ringan yang diciptakan oleh Microsoft untuk sistem operasi multiplatform, yang tersedia pada versi Mac, Linux dan Windows. Visual Studio Code mendukung bahasa pemrograman Typescript, Javascript dan NodeJS, selain itu terdapat bahasa pemrograman lainnya dengan bantuan plugin yang dapat di gunakan via marketplace Visual Studio Code seperti C#, C++, Python, Java, Go, dll).



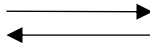




Terdapat banyak fitur yang disediakan oleh Visual Studio Code, seperti Git, Intellisense, Debugging dan Integration. Fitur-fitur yang ada akan terus bertambah seiring dengan bertambahnya versi dari Visual Studio Code. Visual Studio Code melakukan pembaharuan setiap bulannya, sehingga ini yang membedakan Visual Studio Code dengan teks editor lainnya. Teks editor Visual Code bersifat open source dimana kode sumbernya dapat dilihat dan kita dapat berkontribusi dalam mengembangkannya (Permana & Romadlon, 2019).



### 2.15 Mapping Chart

*Mapping chart* adalah penggambaran secara grafik urutan dan langkah-langkah prosedur dari suatu program. *Mapping chart* efektif dalam menggambarkan proses ataupun prosedur dalam sebuah organisasi. *Mapping chart* digunakan dalam menggambarkan proses atau urutan prosedur kerja dalam pembuatan sistem (Widarma & Kumala, 2017). Berikut simbol *Mapping Chart* pada Tabel 1.

Tabel 1. Simbol dan keterangan *Mapping Chart*

Simbol	Nama	Deskripsi
	Dokumen	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas ataupun output dicetak di kertas.
	<i>Input/Output</i>	Menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
	<i>Data Flow</i>	Simbol ini digunakan untuk menggambarkan aliran data yang berjalan.
	<i>Process</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
	<i>Decision Node</i>	Simbol keputusan, ya atau tidak dalam proses pengolahan data.
	<i>Manual Operation</i>	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	<i>Terminator</i>	Simbol untuk permulaan ( <i>start</i> )

Sumber: (Verawati & Liksha, 2018).

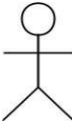
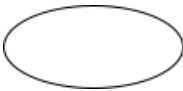





## 2.16 Unified Modeling Language (UML)

*Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan dari sebuah sistem pengembangan *software* berbasis *object oriented* (Manalu, 2015), diantaranya yaitu:

### a. Use Case Diagram

*Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan. (Manalu, 2015) Simbol-simbol beserta keterangan yang digunakan dalam menggambarkan *Use Case Diagram* disajikan pada Tabel 2.

Tabel 2. Simbol-simbol *Use Case Diagram*

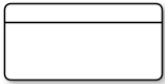
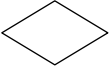



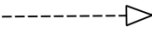
Nama	Simbol	Keterangan
<i>Actor</i>		Sistem, proses, ataupun orang yang berhubungan dengan sistem informasi yang dibangun di luar sistem..
<i>Use Case</i>		Uraian dari susunan aksi-aksi direpresentasikan sistem untuk mewujudkan hasil telah terukur bagi sistem.
<i>Include</i>		Perincian <i>use case</i> sumber secara spesifik.
<i>Associaton</i>		Relasi antar satu objek dengan objek yang lain.
<i>Collaboration</i>		Hubungan aturan-aturan dan elemen lainnya utuk bekerja sama menyajikan integritas lebih besar.
<i>Note</i>		Komponen fisik yang tetap ada pada saat aplikasi dijalankan.
<i>System</i>		Rincian paket yang mempresentasikan sistem dengan terbatas.

Sumber: Havaluddin, (2016).

### b. *Class Diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan (Hendini, 2016).

Tabel 3. Simbol-simbol *Class Diagram*








Simbol	Nama	Keterangan
	<i>Class</i>	Gabungan objek-objek yang berbagi atribut dan juga operasi yang serupa.
	<i>N-Ary Association</i>	Usaha untuk menolak asosiasi lebih dari dua objek.
	<i>Collaboration</i>	Penjelasan dari rangkaian aksi-aksi yang dijelaskan oleh sistem dengan mewujudkan hasil yang terukur bagi aktor.
	<i>Association</i>	Relasi antar objek satu dengan yang lainnya.
	<i>Generalization</i>	Relasi <i>descendent</i> membagikan kegiatan dan bentuk data dari sebuah objek yang terdapat pada <i>ancestor</i> .
	<i>Depedency</i>	Relasi dimana elemen mandiri ( <i>independent</i> ) berubah yang mempengaruhi elemen yang bergantung pada elemen lain.

Sumber : (Haviluddin, 2016).

### c. *Activity Diagram*

*Activity Diagram* menggambarkan *workflow* (aluran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Hendini, 2016). Berikut simbol yang terdapat pada *Activity Diagram* disajikan pada Tabel 4.

Tabel 4. Simbol-simbol *Activity Diagram*

Nama	Simbol	Keterangan
<i>Activity</i>		Kegiatan yang dilakukan oleh sistem.
<i>Fork</i>		Menunjukkan suatu aktivitas paralel
<i>Join</i>		Asosiasi gabungan yaitu satu atau lebih aktivitas digabung menjadi satu
<i>Decision</i>		Asosiasi percabangan yaitu jika terdapat sebuah pilihan aktivitas yang berjumlah lebih dari satu.
<i>Start Point</i>		Proses sebuah objek dibangun ataupun dimulai.
<i>End Point</i>		Proses sebuah objek dibangun dan diselesaikan.
<i>Swinlane</i>		Pemisahan organisasi bisnis yang bertanggung jawab pada kegiatan yang terjadi

Sumber : (Suendri, 2018).