

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

Tinjauan umum perusahaan praktek kerja lapangan pada bab ini menjelaskan secara singkat profil perusahaan PT. Duta Transformasi Insani.

2.1.1. Sejarah Singkat Perusahaan

Pendirian perusahaan ini adalah diawali dari ditetapkannya Rencana Strategis Daarut Tauhiid 2022 yang mendorong terjadinya sinergi aktivitas yang ada di lingkungan pesantren Daarut Tauhid, salah satu lini yang paling strategis adalah bidang pelatihan. Sebelumnya terdapat dua lembaga pelatihan dalam lingkungan Daarut Tauhid yaitu Daarut Tauhid Training Center (DTTC) yang merupakan unit kerja di Yayasan Daarut Tauhiid dan LP2ES Learning Center yang merupakan unit usaha Kopontren Daarut Tauhid.

Sebagai upaya untuk mensinergikan dan mengembangkan peran dan kiprah Daarut Tauhid di bidang pelatihan sumber daya manusia, maka pada tanggal 1 Mei 2013, didirikanlah perusahaan berbentuk Perseroan Terbatas (PT) yang merupakan hasil *merger* dari dua unit pelatihan tersebut, dan diberi nama PT. Duta Transformasi Insani (PT DT Insani) dengan Yayasan Daarut Tauhiid, Kopontren Daarut Tauhiid, dan KH Abdullah Gymnastiar sebagai pemegang sahamnya. Bidang usaha dari perusahaan ini adalah (1) Pendidikan dan Pelatihan (2) Konsultan Manajemen, (3) *Event Organizer*, dan (4) Laboratorium Bisnis.

2.1.2. Logo Instansi

Berikut ini adalah logo resmi perusahaan yang digunakan oleh PT. Duta Transformasi Insani :



Gambar 2.1 Logo PT. Duta Transformasi Insani

Bentuk keseluruhan logo ini merupakan penggabungan huruf 'd', 't', 'i' dalam satu karakter. Unsur pita dan berlian melambangkan proses perjalanan (pita) menuju suatu tujuan (berlian). Perjalanan dalam rangka perubahan ini merupakan proses yang berkelanjutan, karena itu tujuan yang dicapai merupakan awal dari proses perjalan berikutnya, yang digambarkan oleh pita yang keluar lagi dari berlian. Ini menggambarkan suatu proses perjalanan yang terus-menerus untuk selalu menjadi lebih baik dan lebih baik lagi.

2.1.3. Visi dan Misi

Perusahaan PT.Duta Transformasi Insani memiliki visi dan misi, yaitu:

1. Visi

Menjadi konsultan manajemen yang terbaik dan terpercaya di Indonesia dalam pengembangan karakter baik dan kuat pada tahun 2020.

2. Misi

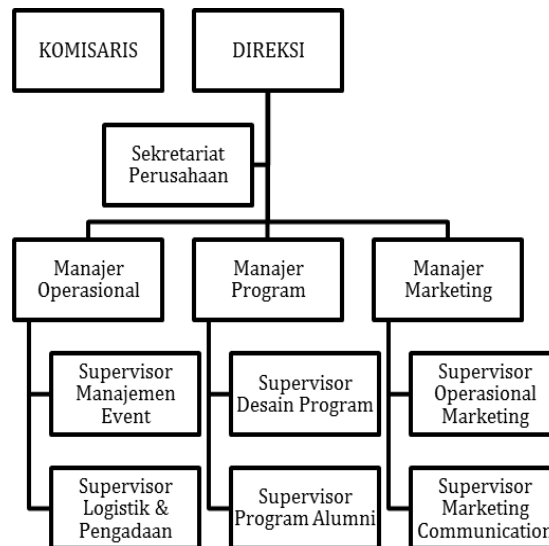
- a. Membangun bangsa yang berkarakter baik dan kuat menuju kebahagiaan dunia dan akhirat.
- b. Memberikan konsultansi manajemen dengan pelayanan terbaik, memuaskan, kreatif inovatif dan berkesinambungan.
- c. Membangun lembaga berkah, sehat, kuat terpercaya, serta berdaya saing yang dikelola oleh sumber daya manusia ahli dzikir, fikir dan ikhtiar.

3. Tujuan

Memberikan manfaat dan solusi dalam pengembangan sumber daya manusia.

2.1.4. Struktur Organisasi

Organisasi adalah tempat atau wadah orang berkumpul untuk saling bekerjasama untuk mencapai tujuan dan manfaat bersama. Dengan demikian dapat dikatakan bahwa struktur organisasi pada suatu organisasi merupakan kerangka dasar yang menggambarkan alur hubungan antara bagian yang satu dengan yang lainnya, sehingga suatu bagian dalam organisasi tersebut menjadi jelas kedudukan, jabatan, wewenang dan juga tanggung jawabnya.



Gambar 2.2 Struktur Organisasi PT. Duta Transformasi Insani

2.1.5. Deskripsi Kerja

Pada perusahaan ini memiliki beberapa bidang kerja dan pegawai. Adapaun disebut pegawai ialah mulai dari level kepala sekretariat, manajer, supervisor, dan staff, sedangkan direksi dan komisaris tidak termasuk dalam kategori pegawai. Dan berikut ini penjelasan mengenai fungsi pokok dari masing-masing bidang yang dideskripsikan sebagai berikut:

1. Komisaris

Melakukan pengawasan terhadap kebijakan pengurusan Perusahaan yang dilakukan direksi serta memberi nasihat kepada direksi termasuk mengenai rencana pengembangan perusahaan, rencana jangka panjang, rencana kerja dan anggaran perusahaan, pelaksanaan ketentuan-ketentuan anggaran

dasar dan keputusan RUPS (Rapat Umum Pemegang Saham) dan peraturan perundang-undangan yang berlaku.

2. Dewan Direksi

Direksi merupakan organisasi perusahaan yang bertanggung jawab penuh atas pengurusan perusahaan untuk kepentingan dan tujuan Perusahaan serta mewakili perusahaan baik di dalam maupun di luar pengadilan sesuai ketentuan anggaran dasar.

3. Sekretariat

Merencanakan, mengembangkan, dan mengontrol fungsi administrasi, keuangan, dan personalia di perusahaan sehingga dapat memberikan informasi keuangan secara komprehensif dan tepat waktu untuk membantu perusahaan dalam proses pengambilan keputusan yang mendukung pencapaian target keuangan perusahaan.

4. Operasional

Merencanakan, mengarahkan dan mengendalikan semua sumber daya untuk menyelenggarakan operasional event/pelatihan dengan efektif, efisien, memenuhi standar kualitas program/layanan yang persyaratan, dan memberikan sigma kepuasan kepada kastemer.

5. Marketing

Merencanakan, mengarahkan dan mengendalikan seluruh kegiatan pemasaran program/layanan perusahaan serta mengembangkan hubungan dengan kastemer

6. Program

Merencanakan, mengarahkan dan mengendalikan seluruh kegiatan pengembangan program/layanan yang sesuai dengan kebutuhan kastemer dan mendukung pencapaian visi, misi serta tujuan perusahaan.

2.2 Landasan Teori

Sub bab ini berisikan teori - teori pendukung yang digunakan dalam proses analisis dan implementasi pada permasalahan yang ada di PT. Duta Transformasi Insani, dan berikut ini adalah landasan teori yang digunakan penulis, yaitu :

2.2.1 Basis Data

Basis Data terdiri atas lebih 2 kata, yaitu Basis dan Data, Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. [2]

2.2.1.1 Operasi Dasar Basis Data

Operasi – operasi dasar yang dapat kita lakukan berkenaan dengan basis data dapat meliputi [2]:

1. Pembuatan basis data baru (*create database*).
2. Penghapusan basis data (*drop database*).
3. Pembuatan file/table baru ke suatu basis data (*create table*)
4. Penghapusan file/tabel dari suatu basis data (*drop table*).
5. Penambahan/pengisian data baru ke sebuah file/tabel di sebuah basis data (*insert*).
6. Pengambilan data dari sebuah file/tabel (*retrieve/search*).
7. Pengubahan data dari sebuah file/tabel (*update*).
8. Penghapusan data dari sebuah file/tabel (*delete*).

2.2.1.2 Objektif Basis Data

Pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) seperti berikut ini: [2]

1. Kecepatan dan Kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan muda.

2. Efisiensi Ruang Penyimpanan (*Space*)

Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena kita dapat melakukan penekanan jumlah redundansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi antar kelompok data yang saling berhubungan.

3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aluran/batasan (*constraint*) tipe data, domain data, keunikan data, dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

4. Ketersediaan (*Availability*)

Pertumbuhan data (baik dari sisi jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu kita gunakan/butuhkan. Karena itu kita dapat memilah adanya data utama/master/referensi, data transaksi, dan histori hingga data kadaluarsa.

5. Kelengkapan (*Completeness*)

Lengkap/tidaknya data yang kita kelola dalam sebuah basis data bersifat relatif (baik terhadap kebutuhan pemakai maupun terhadap waktu). Bila seorang pemakai sudah menganggap bahwa data yang dipelihara sudah lengkap, maka pemakai yang lain belum tentu berpendapat sama. Atau, yang sekarang dianggap sudah lengkap, belum tentu dimasa yang akan datang juga demikian.

6. Keamanan (*Security*)

Dengan ini kita dapat menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data beserta objek-objek didalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan.

7. Kebersamaan Pemakaian (*Sharability*)

Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan *multiuser*, akan dapat memenuhi kebutuhan ini, tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti

inkonsistensi data(karena data yang sama diubah oleh banyak pemakai pada saat yang bersamaan) atau kondisi *deadlock* (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

2.2.2 *Object Oriented Programming*

Pemrograman berorientasi objek atau sering disebut *Object Oriented Programming* (OOP) merupakan paradigma pemrograman yang berorientasi kepada objek. Semua data dan fungsi di dalam paradigm ini dibungkus ke dalam kelas-kelas atau objek-objek.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih kemudahan mengubah program dan digunakan luas dalam teknik piranti lunak skala besar [3].

2.2.2.1 Konsep Dasar *Object Oriented Programming*

Dalam menggunakan pemrograman berorientasi objek terdapat beberapa konsep dasar yang harus diketahui, yaitu [3]:

A. Kelas

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh '*class of dog*' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah *class* adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah *class* secara tipikal sebaiknya dapat dikenali oleh seorang *non-programmer* sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP).

Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

B. Objek

Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer, objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

C. Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

D. Enkapsulasi

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak, hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

E. Polimorfisme

Melalui pengiriman pesan ini tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan, metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari.

Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut *polimorfisme* karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat

memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama.

2.2.3 Framework

Kerangka kerja atau sering disebut *framework* merupakan suatu kerangka kerja yang berupa sekumpulan *folder* yang memuat *file-file* php yang menyediakan *class libraries*, *helpers*, *plugin*, dan lainnya. *Framework* menyediakan konfigurasi dan teknik *coding* tertentu [3].

2.2.3.1 Model View Controll (MVC)

Model View Controll merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi *web*, berawal dari *Small Talk*, MVC memisahkan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi, yaitu [3]:

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. *Model*, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

2.2.3.2 Codeigniter

Codeigniter (CI) adalah salah satu *framework* yang digunakan dalam bahasa pemrograman PHP berbasis *web*. CI memiliki kelebihan dibandingkan dengan *framework* lainnya [3], yaitu :

1. Performa sangat cepat : salah satu alasan tidak menggunakan framework adalah karena eksekusinya yang lebih lambat daripada PHP from the scratch, tapi Codeigniter sangat cepat bahkan mungkin bisa dibilang

codeigniter merupakan *framework* yang paling cepat dibanding *framework* yang lain.

2. Konfigurasi yang sangat minim (*nearly zero configuration*) : tentu saja untuk menyesuaikan dengan *database* dan keleluasaan routing tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan *codeigniter* dengan setting standard, anda hanya perlu mengubah sedikit saja file pada folder config.
3. Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
4. Dokumentasi yang sangat lengkap : Setiap paket instalasi *codeigniter* sudah disertai user guide yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

2.2.4 Unified Modelling Language

UML adalah salah satu standar bahasa yang digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. [1]

Dalam pemodelan menggunakan UML dapat dikategorikan menjadi tiga bagian yaitu:

a. Structure Diagram

Kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Diagram tersebut yaitu *Class Diagram*, *Object Diagram*, *Component Diagram*, *Composite Diagram*, *Package Diagram*, dan *Deployment Diagram*.

b. Behavior Diagram

Kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Diagram tersebut yaitu *Use Case Diagram*, *Activity Diagram*, dan *State Diagram*.

c. *Interaction Diagram*

Kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar sub sistem pada suatu sistem. Diagram tersebut yaitu *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, dan *Interaction Diagram*.

Berikut ini penjelasan dari masing-masing diagram UML yang digunakan untuk menganalisa kebutuhan fungsional dan perancangan sistem, yaitu :

2.2.4.1 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. [1]

2.2.4.2 Class Diagram

Class diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. [1]

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh kelas.

2.2.4.3 Deployment Diagram

Diagram *deployment* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut: [1]

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
2. Sistem *client/server*.
3. Sistem terdistribusi murni.
4. Rekayasa ulang aplikasi.

2.2.4.4 Activity Diagram

Activity Diagram atau diagram aktivitas menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem. [1]

Diagram aktivitas banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- d. Rancangan menu yang ditampilkan pada perangkat lunak.

2.2.4.5 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstalasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*. [1]

2.2.5 PHP HyperText Preprocessor

PHP (*PHP Hypertext Preprocessor*) merupakan *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML (*Hypertext Markup Language*) secara *on the fly*, dimana *script* ini disisipkan pada dokumen HTML. Dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat menggunakan editor teks atau editor HTML. [4]

2.2.6 *Hypertext Markup Language*

HTML (*Hypertext Markup Language*) merupakan teks murni yang sering disebut dengan *web page* dan dokumen ini berekstensi “.htm” atau “.html”. Semua halaman *web* yang dibuat menggunakan HTML didalamnya mengandung 3 unsur, yaitu *elements*, *attributes*, dan *entities*. [5]

- a. *Elements* adalah isi dari sebuah halaman *web*, dalam elemen berisi *tag*. *Tag* sendiri terdiri dari dua jenis yaitu *tag tunggal* dan *tag berpasangan*.
- b. *Attributes* digunakan untuk mengatur elemen.
- c. *Entities* digunakan untuk menampilkan karakter-karakter yang bukan ASCII, misal karakter ©.

2.2.7 *Structured Query Language*

SQL (*Structured Query Language*) merupakan suatu bahasa permintaan yang terstruktur, dimana dalam penggunaannya SQL memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI, sehingga SQL merupakan bahasa yang melekat pada *database* atau DBMS (*Database Mangement System*) tertentu, seperti MySQL, PostgreSQL, Interbase, dan Oracle. [6]

Dalam penggunaannya SQL dikategorikan menjadi tiga sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*), DCL (*Data Control Language*).

Ada tiga perintah yang termasuk dalam DDL, yaitu : *create*, *alter*, dan *drop*, kemudian ada empat perintah yang digunakan dalam DML, yaitu : *insert*, *select*, *update*, dan *delete*, serta ada dua perintah dalam DCL, yaitu : *grand* dan *revoke*.