



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

Kadir (2013:2) Komputer merupakan peralatan elektronis yang biasa dipakai orang untuk membantu pelaksanaan pekerjaan. Sebagai contoh, mahasiswa menggunakan program statistik seperti SPSS untuk mengolah hasil penelitian dan sekretaris menggunakan pengolahan kata seperti *Microsoft Word* atau *Open Office Writer* untuk membuat surat dan dokumen.

Sopandi (2005:5) Komputer saat ini merupakan komponen yang sangat penting dalam menyelesaikan proses sistem administrasi perkantoran. Hal ini bisa kita lihat bagaimana percepatan arus informasi baik dari dalam maupun diluar kantor.

Jadi, komputer adalah sebuah alat elektronik yang dapat memproses data sesuai dengan perintah-perintah yang telah diurutkan dan diprogramkan.

##### 2.1.2 Pengertian Data

Wahyudi (2004:1) Kata data diambil dari bahasa Inggris yang berasal dari bahasa Yunani *datum* yang berarti fakta. Makna kata data bagi manusia adalah segala sesuatu yang dapat diterima oleh indera manusia dari rangsangan-rangsangan yang ada disekitarnya, baik secara tersurat maupun tersirat.

##### 2.1.3 Pengertian Pengolahan Data

Jogiyanto (1993:5) Pengolahan data adalah suatu proses yang menerima data sebagai masukan (input), diproses (*processing*) oleh program tertentu yang tersimpan dan mengeluarkan hasil proses data tersebut dalam bentuk informasi (*output*).



## 2.2 Teori Khusus

### 2.2.1 Pengertian Database

Nugroho (2004:69) *database* atau basis data merupakan sekumpulan data yang sangat kompleks yang memiliki hubungan satu dengan yang lainnya. Di dalam sebuah *database* data *dimanage* dengan menggunakan sebuah pengelompokkan menjadi beberapa bagian yang berupa *field*.

Sukamto dan Shalahuddin (2013:43) Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.

Jadi, *database* adalah media yang diakses dengan mudah dan cepat untuk menyimpan data.

### 2.2.2 Pengertian DFD (*Data Flow Diagram*)

Pressman (2002:364) Diagram aliran data atau *data flow diagram* (DFD) adalah buah teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* ke *output*.

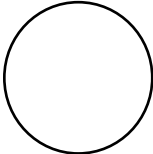
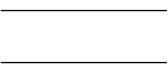
Sukamto dan Shalahuddin (2013:70) DFD dapat digunakan untuk mempresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk mempresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

DFD tidak sesuai untuk memodelkan sistem perangkat lunak yang akan dibangun menggunakan pemrograman berorientasi objek. Paradigma pemrograman terstruktur dan pemrograman berorientasi objek merupakan hal yang berbeda. Jangan mencampuradukan pemrograman terstruktur dan pemrograman berorientasi objek.





Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut:

**Tabel 2.1** Simbol-simbol DFD (*Data Flow Diagram*)

Simbol	Keterangan
	<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya di dalam kode program</p> <p>Catatan: Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
	<p>File atau basisdata atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity relationship diagram (ERD)</i>, <i>Conceptual Data Model (CDM)</i>, <i>Physical Data Model (PDM)</i>).</p> <p>Catatan: Nama yang diberikan pada sebuah penyimpanan biasanya kata benda.</p>



**Lanjutan Tabel 2.1** Simbol-simbol DFD (*Data Flow Diagram*)

	<p>Entitas luar (<i>external entity</i>) atau masukan (input) atau keluaran (output) atau orang berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.</p> <p>Catatan: Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
	<p>Aliran data merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (input) atau keluaran (output).</p> <p>Catatan: Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa” .</p>

**Sumber:** Sukamto dan Shalahuddin (2013:71)

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga Context Diagram . DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.



## 2. Membuat DFD Level 1

Dfd Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

## 3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modeul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

## 4. Membuat DFD Level 3 dan seterusnya

DFD Level 3,4,5 dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdwon* pada level 3,4,5 dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

Pada satu diagram DFD sebaliknya jumlah modul tidak boleh lebih dari 20 buah. Jika lebih dari 20 buah, diagram akan terlihat rumit dan susah dibaca sehingga menyebabkan sistem yang dikembangkan juga menjadi rumit.

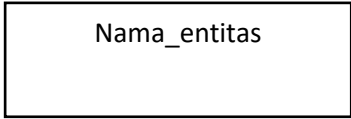
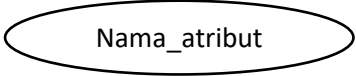
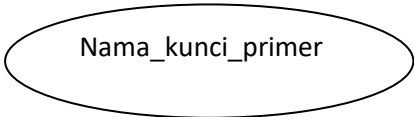
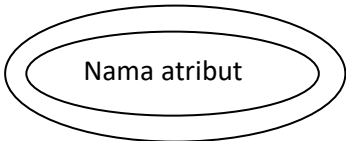
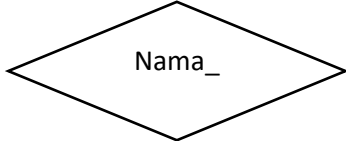
### 2.2.3 Pengertian ERD (*Entity Relationship Diagram*)

Sukamto dan Shalahuddin (2013:50) Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD), ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain.

Namun yang banyak digunakan pada ERD dengan notasi Chen, sebagai berikut:

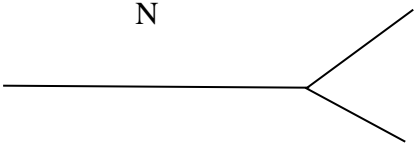


**Tabel 2.2** Simbol-simbol *Entity Relationship Diagram (ERD)*

No	Simbol	Deskripsi
1.	Entitas/Entity 	Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimoan dalam suatu entitas.
3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4.	Atribut multivalai / <i>multivalue</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.



**Lanjutan Tabel 2.2** Simbol-simbol *Entity Relationship Diagram (ERD)*

6.	Asosiasi / <i>association</i>  	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan yang lain disebut kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas dan entitas B.
----	------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



**Sumber:** Sukamto dan Shalahuddin (2013:50)

#### 2.2.4 Block Chart

Wahyono (2005:75) *Block Chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.

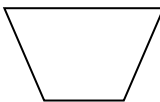
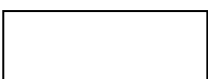
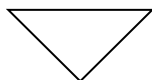


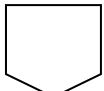
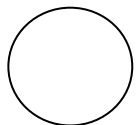
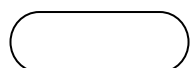
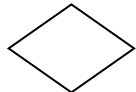


Adapun simbol-simbol yang sering digunakan dalam *Blockchart* dapat dilihat pada tabel sebagai berikut:

**Tabel 2.3** Simbol-Simbol *Block Chart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam formulir, buku/bendel/berkas atau cetakan
2.		Multi Dokumen



**Lanjutan Tabel 2.3** Simbol-Simbol *Block Chart*

3.		Proses Manual
4.		Proses yang dilakukan oleh komputer
5.		Menandakan dokumen yang diarsipkan (Arsip Manual)
6.		Data Penyimpanan (data storage)
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik
8.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran
11.		Pengambilan keputusan (decision)
12.		Layar peraga (monitor)
13.		Pemasukan data secara manual

**Sumber:** Wahyono (2005:75)







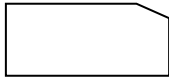


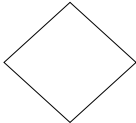
### 2.2.5 Flowchart

Kadir (2013:20) mengemukakan bahwa, *flowchart* (diagram alir) dapat digunakan sebagai alternatif untuk menyajikan algoritma. Sebagai contoh perhitungan luas lingkungan berdasarkan jari-jari dapat dinyatakan dalam bentuk *flowchart*.

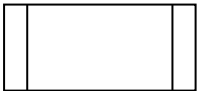
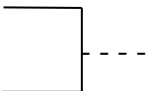
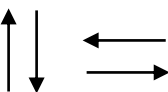
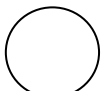


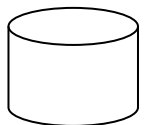
Jogiyanto (1993:20) *Flowchhart* adalah suatu skema atau bagan yang menggambarkan urutan kegiatan dan suatu program dari awal sampai akhir.

Untuk menggambarkan bagan alir ini, digunakan simbol-simbol tertentu, seperti sebagai berikut:

**Tabel 2.4** Simbol-simbol *Flowchart*

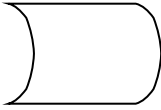
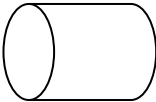
No.	Simbol	Keterangan
1.		<i>Terminal</i> - menunjukkan awal mulai dan akhir dari kegiatan.
2.		<i>Input/output</i> - menunjukkan operasi pembacaan input atau percetakan output.
3.		<i>Input/output kartu plong</i> - menunjukkan media input/output (I/O) menggunakan kartu plong.
4.		<i>Input/output dokumentasi</i> - digunakan untuk pembacaan input melalui optical scanner atau percetakan output pada printer.
5.		<i>Pengolahan</i> - menunjukkan suatu pemrosesan.
6.		<i>Keputusan</i> - menunjukkan suatu seleksi yang harus dikerjakan.

Lanjutan Tabel 2.4 Simbol-simbol *Flowchart*

7.		<p><i>Proses terdefinisi</i></p> <ul style="list-style-type: none"> <li>- menunjukkan sejumlah proses yang detailnya tidak ditunjukkan di sini, tetapi terpisah terdefinisi sendiri, berupa program bagian (subroutine).</li> </ul>
8.		<p><i>Komentar proses</i></p> <ul style="list-style-type: none"> <li>- menunjukkan keterangan atau komentar untuk memperjelas maksud isi dari simbol-simbol <i>flow-chart</i> yang hendak dijelaskan.</li> </ul>
9.		<p><i>Garis alir</i></p> <ul style="list-style-type: none"> <li>- garis untuk menghubungkan arah tujuan simbol <i>flow-chart</i> yang satu dengan yang lainnya.</li> </ul>
10.		<p><i>Penghubung halaman yang sama</i></p> <ul style="list-style-type: none"> <li>- bila <i>flow-chart</i> terpotong dan masih mempunyai sambungan dalam halaman yang sama, digunakan simbol ini.</li> </ul>
11.		<p><i>Penghubung halaman berikutnya</i></p> <ul style="list-style-type: none"> <li>- bila <i>flow-chart</i> terpotong dan masih mempunyai sambungan pada halaman berikutnya, digunakan simbol ini.</li> </ul>
12.		<p><i>Magnetic tape</i></p> <ul style="list-style-type: none"> <li>- menunjukkan alat input/output berupa tape magnetis.</li> </ul>
13.		<p><i>Magnetic disk</i></p> <ul style="list-style-type: none"> <li>- menunjukkan alat input/output berupa disk magnetis (hard disk).</li> </ul>



**Lanjutan Tabel 2.4** Simbol-simbol *Flowchart*

14.		<i>Mini disk (diskette)</i> - menunjukkan alat input/output berupa mini disk atau diskette atau floppy disk atau flexibel disk.
15.		<i>Magnetic drum</i> - menunjukkan alat input/output berupa drum magnetis.

**Sumber:** Jogiyanto (1993:20)

### 2.2.6 Kamus Data (*Data Dictionary*)

Pressman (2002:388) Kamus data merupakan sebuah daftar yang terorganisasi dari elemen data yang berhubungan dengan sistem, dengan definisi yang tegas dan teliti sehingga pemakai dan analis sistem akan memiliki pemahaman yang umum mengenai input, output, komponen penyimpanan dan bahkan kalkulasi inter-mediate. Saat ini kamus, kamus data hampir selalu diimplementasikan sebagai bagian dari sebuah “peranti desain dan analisis terstruktur” CASE. Meskipun format kamus bervariasi dari peranti satu ke peranti yang lain, sebagian besar berisi informasi berikut ini:

1. *Name* adalah nama sebenarnya dari data atau item kontrol, penyimpanan data, atau entitas eksternal.
2. *Aliasi* adalah nama lain yang digunakan untuk entri pertama.
3. *Where-used/how-used* adalah suatu daftar dari proses yang menggunakan data atau item kontrol dan bagaimana dia digunakan.
4. *Content description* adalah suatu notasi untuk mempresentasikan isi.
5. *Supplementary information* adalah informasi lain mengenai tipe data, harga preset, barisan, dll.



Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

**Tabel 2.5** Simbol-Simbol Kamus Data

Konstruk data	Notasi	Arti
	=	Disusun atas
	+	Dan
Berurutan	[]	Baik ini – atau
Pilihan	{ } <sup>n</sup>	Pengulangan ke-n dari
Pengulangan	()	Data opsional
	*...*	Komentar tidak dibatasi

**Sumber:** Pressman (1997:389)

## 2.3 Teori Program

### 2.3.1 Pengenalan PHP

Nugroho (2004:140) PHP merupakan bahasa standar yang digunakan dalam dunia web site. PHP adalah bahasa pemrograman yang berbentuk *script* yang diletakkan di dalam server web. Jika kita lihat dari sejarah, mulanya PHP diciptakan dari ide Rasmus Lerdof yang membuat sebuah *script* perl. *Script* tersebut sebenarnya dimaksudkan untuk digunakan sebagai program untuk dirinya sendiri. Akan tetapi, kemudian dikembangkan lagi sehingga menjadi sebuah bahasa yang disebut “Personal Home Page”. Inilah awal mula munculnya PHP sampai saat ini.

Sidik (2006:3) PHP merupakan *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara on the fly, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML.



### 2.3.1.1 Script PHP

Nugroho (2004:151-152) PHP adalah program yang fleksibel, artinya script-script PHP dapat Anda tuliskan pada sela-sela tag HTML. Karena PHP memiliki sifat yang dapat berinteraksi dengan program lain, maka kita tidak usah bingung-bingung dalam menuliskan scriptnya. Seperti contohnya sebagai berikut:

```
<HTML>
<HEAD>
<TITLE>Contoh Pertama</TITLE>
</HEAD>
<?php
print ("Halo mas Bunafit Nugroho , Apa Kabar ");
?>
<BODY>
</BODY></HTML>
```

### 2.3.1.2 Beberapa Kegunaan PHP

Nugroho (2004:142) Hampir seluruh aplikasi berbasis web dapat dibuat dengan PHP ini, namun fungsi PHP yang paling utama adalah untuk menghubungkan database dengan web . Dengan PHP, membuat aplikasi web yang terkoneksi ke database menjadi sangat mudah. Sistem database yang telah didukung oleh PHP adalah:

- a. Oracle
- b. Sybase
- c. mSQL
- d. MySQL
- e. Solid
- f. Generic ODBC
- g. PostgresSQL

PHP juga mendukung komunikasi dengan layanan lain melalui protokol IMAP, SNMP, NNTP, dan POP3 atau HTTP.



### 2.3.2 Pengertian HTML

Sulhan (2006:23) *Hypertext Markup Language* (HTML) adalah suatu sistem untuk menambahkan dokumen dengan tabel yang menandakan bagaimana teks di dokumen harus disajikan dan bagaimana dokumen dihubungkan bersamasama.

Wahyono (2005:20) HTML adalah ringkasan daripada Hypertext Markup Language. Fungsi HTML di dalam sebuah dokumen Web adalah untuk mengatur struktur tampilan dokumen tersebut dan juga untuk menampilkan “links” atau sambungan ke lokasi di internet yang lain.

### 2.3.3 Pengertian MYSQL

Nugroho (2004:29) MYSQL (*My Struktur Query Language*) atau yang biasa dibaca “mai-se-kuel” adalah sebuah program pembuat database yang bersifat open source, artinya siapa saja boleh menggunakannya dan tidak dicekal. Saat kita mendengar open source, kita ingat dengan sistem operasi handal keturunan Unix, yaitu Linux.

Utomo (2014:58) MYSQL merupakan salah satu database yang banyak digunakan oleh para pengguna komputer di seluruh dunia.

Jadi MYSQL adalah bahasa standar yang digunakan untuk mengakses database.

**Gambar 2.1** Logo MySQL



Sumber: <http://www.pelicanweb.net>



### 2.3.4 Pengertian Xampp

Wibowo (2007:5) XAMPP merupakan paket aplikasi yang memudahkan Anda dalam menginstalasi Modul PHP, Apache Web Server, dan MySQL Database. Selain itu XAMPP dilengkapi dengan berbagai fasilitas lain yang akan memberikan kemudahan dalam mengembangkan situs web berbasis PHP.

Riyanto (2011:1) XAMPP merupakan paket PHP dan MySQL berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP.

### 2.3.5 Pengertian phpMyAdmin

Nugroho (2004:65) PhpMyAdmin adalah suatu program open source yang berbasis web yang dibuat menggunakan aplikasi PHP. Program ini digunakan untuk mengakses database MySQL. Program ini mempermudah dan mempersingkat kerja kita. Dengan Kelebihannya, para pengguna awam tidak harus paham syntax-syntax SQL dalam pembuatan database dan tabel.

## 2.4 Teori Judul

### 2.4.1 Web

Sulhan (2006:7) *Website* awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan surfer (sebutan bagi pemakai komputer yang melakukan penelusuran informasi di internet).

Sidik (2006:1) *Situs web (website)* awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan *surfer* (sebutan bagi pemakai komputer yang melakukan penelusuran informasi di Internet).



#### **2.4.2 Aplikasi**

Kadir (2013:3) Istilah Program biasa digunakan di lingkungan orang yang bekerja di bidang teknologi informasi, untuk menyatakan hasil karya mereka yang berupa instruksi-instruksi untuk mengendalikan komputer. Di sisi pemakai, hal seperti itu biasa disebut sebagai aplikasi.

#### **2.4.3 Perancangan**

Ladjamudin (dalam Sari, dkk 2018:29) menyebutkan bahwa, Perancangan adalah suatu kegiatan yang memiliki tujuan untuk mendesign sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik.

#### **2.4.4 Cuti**

Prakoso (1987:352) Cuti adalah keadaan tidak masuk kerja yang diizinkan dalam jangka tertentu. Tujuan cuti adalah untuk menjamin kesegaran jasmani dan rohani serta untuk kepentingan Pegawai.