

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Perangkat Lunak

Menurut Sukamto dan Shalahuddin (2018:2), “Perangkat Lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Sedangkan Kadir (2017:2), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa Perangkat lunak adalah elemen kunci dalam evolusi sistem dan produk yang terasosiasi dengan dokumentasi perangkat lunak berupa program untuk mengatur hubungan antara hardware dan brainware.

##### 2.1.2 Pengertian Komputer

Menurut Harmayani, dkk. (2021:2), “Definisi Komputer adalah alat elektronik yang terdiri dari rangkaian berbagai komponen yang saling terhubung sehingga membentuk suatu sistem kerja”.

Sedangkan Fauzi (2018:1), “Komputer adalah suatu peralatan elektronik yang dapat menerima input, mengolah input (*processing*), memberikan informasi dengan menggunakan suatu program yang tersimpan di memori komputer dan dapat menyimpan program dari hasil pengolahan yang bekerja secara otomatis”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian komputer adalah alat elektronik yang dapat menerima informasi input dan menghasilkan informasi output sesuai dengan program yang tersimpan.

##### 2.1.3 Pengertian Data

Data merupakan sekumpulan keterangan atau fakta mentah berupa simbol, angka, kata-kata, atau citra, yang didapatkan melalui proses pengamatan atau pencarian ke sumber-sumber tertentu. Definisi data adalah kumpulan keterangan-keterangan atau deskripsi dasar dari suatu hal (objek atau kejadian) yang diperoleh

dari basil penga-matan (observasi) dan dapat diolah menjadi bentuk yang lebih kompleks, seperti; in-formasi, database, atau solusi untuk masalah tertentu (Pane, Sari, & Wicaksono, 2020:7).

#### **2.1.4 Pengertian Database**

Menurut Pamungkas (2017:2), “Basis Data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu”.

Menurut Rachmadi (2020:1), “Basis data terdiri dari kata basis dan data. Basis dapat diartikan sebagai markas atau gudang. Sedangkan data adalah catatan atas kumpulan fakta dunia nyata yang mewakili objek seperti manusia, barang, hewan, konsep, peristiwa dan sebagainya yang diwujudkan dalam bentuk huruf, angka. Simbol, gambar, teks, bunyi atau kombinasinya”

Berdasarkan pengertian di atas dapat disimpulkan bahwa, Basis Data (database) merupakan kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.

#### **2.1.5 Pengertian Sistem**

Menurut Sukanto dan Shalahuddin (2018:1), “Sistem merupakan kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai”.

Menurut Kadir dikutip Rianto (2019:63), “sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan”.

Berdasarkan pendapat diatas, dapat disimpulkan bahwa sistem adalah kumpulan komponen yang saling berkaitan atau terpadu untuk mencapai suatu tujuan.

#### **2.1.6 Metode Pengembangan Sistem**

Menurut Sukanto dan Shalahudin (2018:28), model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequensial linier*) atau

hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengajuan, dan tahap pendukung (*support*). Tahapan :

#### 1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

#### 2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk data, arsitektur perangkat lunak, representasi antar muka dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

#### 3. Pembuatan kode program

Desain harus ditranlasikan kedalam program perangkat lunak. Hasil dari tahapan ini adalah program komputer sesuai dengan desain yang telah ditentukan pada tahap desain.

#### 4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logika dan fungsional dan memastikan bahwa semua bagian telah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

#### 5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau sistem harus beradaptasi dengan lingkungan baru. Pada pendukung dan pemeliharaan dapat mengulangi

proses pengembangan mulai dari analisis spesifikasi untuk perubahan sistem yang ada, tapi tidak untuk sistem yang baru.

## **2.2 Pengertian Judul**

### **2.2.1 Pengertian Aplikasi**

Menurut Indrajani (2018:3), “Aplikasi adalah program yang menentukan aktivitas pemrosesan informasi yang dibutuhkan untuk penyelesaian tugas-tugas khusus dari pemakai komputer”.

Menurut Pane, Zamzam, & Fadillah (2020:53), “Aplikasi adalah suatu perangkat lunak (software) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu”.

### **2.2.2 Pengertian Akta Cerai**

Akta cerai merupakan akta otentik yang dikeluarkan oleh pengadilan agama sebagai bukti telah terjadi perceraian. Akta cerai bisa diterbitkan jika gugatan dikabulkan oleh majelis hakim dan perkara tersebut telah memperoleh kekuatan hukum tetap (inkracht).

### **2.2.3 Pengertian Pengadilan**

“Pengadilan menurut bahasa adalah dewan atau majelis yang mengadili perkara, mahkamah, proses mengadili keputusan hakim ketika mengadili perkara (bangunan tempat mengadili perkara)” (Bahasa, 1990:7).

### **2.2.4 Pengertian Pengadilan Agama**

“Pengadilan agama adalah sebutan (titelateur) resmi bagi salah satu diantara empat lingkungan peradilan negara atau kekuasaan kehakiman yang sah di Indonesia. Pengadilan Agama juga salah satu diantara tiga peradilan khusus di Indonesia. dua peradilan khusus lainnya adalah Peradilan Militer dan Peradilan Tata Usaha Negara. Dikatakan peradilan khusus karena Pengadilan Agama mengadili perkara-perkara tertentu atau mengenai golongan rakyat tertentu (yang beragama Islam)” (Rasyid, 2000:5).

### **2.2.5 Pengertian Website**

Menurut Abdulloh dalam Sa'ad (2020:3), "Website merupakan halamanhalaman yang berisi informasi yang dapat diakses oleh browser dan mampu memberikan informasi yang berguna bagi para pengaksesnya".

Menurut Arief dalam Nofyat dkk (2018:12), "Pengertian Website adalah "kumpulan dari halaman web yang sudah dipublikasikan di jaringan internet dan memiliki domain/URL (Uniform Resource Locator) yang dapat diakses semua pengguna internet dengan cara mengetikan alamatnya. "

### **2.2.6 Pengertian Aplikasi Pengambilan Akta Cerai pada Pengadilan Agama Palembang Kelas IA berbasis Website**

Aplikasi Pengambilan Akta Cerai pada Pengadilan Agama Palembang Kelas IA berbasis *Website* adalah aplikasi yang dibuat dengan berbagai fitur dengan tujuan untuk memudahkan dalam pengambilan akta cerai pada Pengadilan Agama Palembang Kelas IA, berbasis *website*.

## **2.3 Teori Khusus**

### **2.3.1 Pengertian UML (*Unified Modeling Language*)**









"UML adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem" (Mulyani, 2016:42).

### **2.3.2 Pengertian *Use Case Diagram***


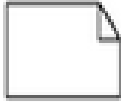
Rahayu dkk (2020:24), "*Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut".

*Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Lanjutan Tabel 2.1 Simbol-simbol pada *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada diatas objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> adalah sumber secara <i>eksplisit</i> .
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Associaton</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasian paket yang menampilkan sistem secara terbatas.
8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

Lanjutan **Tabel 2.1** Simbol-simbol pada *Use Case Diagram*

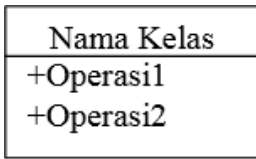
No.	Simbol	Nama	Keterangan
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber: *User Guide Enterprise Architect 7.0*, Modul Workshop UML Bab 2

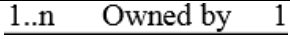

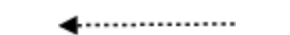
### 2.3.3. Pengertian *Class Diagram*

*Class Diagram* merupakan diagram yang menunjukkan class-class yang terdapat dari sebuah sistem dan hubungannya secara logika. *Class Diagram* memperlihatkan struktur statis dari sebuah sistem yang dijalankan (Fadila, Rahayu, & Saputra, 2020:39). Berikut adalah simbol-simbol yang ada pada diagram kelas:

**Tabel 2.2** Simbol-Simbol pada *Class Diagram*



No.	Simbol	Nama	Keterangan
1.		<i>Class</i>	Blok pembangun pada pemrograman berorientasi objek. Bagian atas adalah bagian dari <i>class</i> . Bagian tengah mendefinisikan <i>property/atribut class</i> . Bagian akhir mendefinisikan <i>method</i> dari sebuah <i>class</i> .

Lanjutan Tabel 2.2 Simbol-simbol *Class Diagram*

No.	Simbol	Nama	Keterangan
2		<i>Assosiation</i>	<i>Relationship</i> paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe- tipe <i>relationship</i> dan juga dapat menampilkan hukum - hukum multiplisitas pada sebuah <i>relationship</i> .
3		<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat bergantung tersebut.
4.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri ( <i>independent</i> ).



Lanjutan **Tabel 2.2** Simbol-simbol *Class Diagram*

No.	Simbol	Nama	Keterangan
5.		<i>Aggregation</i>	Menindikasikan keseluruhan bagian <i>Relationship</i> dan biasanya disebut sebagai relasi.
6.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada diatas objek induk ( <i>ancestor</i> ).



Sumber : *User Guide Enterprise Architect 7.0*, Modul Workshop UML Bab 2

### 2.3.4 Pengertian *Activity Diagram*

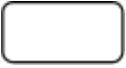



Secara umum *activity diagram* merupakan gambaran alur dari suatu sistem yang dibuat, sehingga pengguna mengerti kegunaan sistem yang akan dibangun. *Activity diagram* merupakan gambaran workflow atau aktivitas dari sebuah sistem dalam proses bisnis (Sidiq, Rd., & Noviana, 2020:110).

*Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum (Habibi, Ferdy, & Ida, 2020:62). Berikut adalah simbol-simbol *activity diagram* :

**Tabel 2.3.** Simbol-Simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Initial</i>	Titik awal, untuk memulai suatu aktivitas.
2.		<i>Dependency</i>	Titik akhir, untuk mengakhiri aktivitas.

Lanjutan **Tabel 2.3.** Simbol-Simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
3.		<i>Activity</i>	Menandakan sebuah aktivitas.
4.		<i>Decision</i>	Pilihan untuk mengambil keputusan.
5.		<i>Fork / Join</i>	Menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
6.		<i>Rake</i>	Menunjukkan adanya dekomposisi.

Sumber: *User Guide Enterprise Architect 7.0*, Modul Workshop UML Bab 2

### 2.3.5 Pengertian *Sequence Diagram*

*Sequence Diagram* merupakan diagram yang memperlihatkan interaksi antar objek dan mendefinisikan komunikasi di antara objek tersebut. Diagram ini menunjukkan serangkaian pesan yang dipertukarkan oleh objek yang melakukan aksi tertentu (Sinaga, Rd., & Woro, 2020:89).

*Sequence diagram* merupakan suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek termasuk pengguna, display, dan sebagiannya berupa pesan (Sidiq, Rd., & Noviana, 2020:89). Berikut adalah simbol-simbol yang ada pada *sequence diagram*:

**Tabel 2.4** Simbol-Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.	<u>:Object1</u>	<i>Object</i> (Partisipan)	Merupakan instance dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal.

Lanjutam Tabel 2.4 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
2		<i>Actor</i>	Menspesifikasikan himpunan peran' yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3		<i>Lifeline</i>	Mengindikasikan keberadaan sebuah objek dalam basis waktu.
4		<i>Activation</i>	Mengindikasikan sebuah objek yang akan melakukan sebuah aksi.
5		<i>Object</i>	Menyatakan objek yang berinterkasi dengan pesan.
6		<i>Message</i>	Mengindikasikan komunikasi antara objek.
7.		<i>Self-Message</i>	Mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.
8.		<i>Loop</i>	Mengeksekusi berulang kali dan penjaga menunjukkan dasar iterasi.

Sumber : *Martin Fowler UML Distilled 2005*, Sholiq *Pemodelan Sistem Informasi Berorientasi Objek Dengan UML 2006*, Modul *Workshop UML Bab 2*

### 2.3.6 Pengertian Kamus Data

Menurut Sukamto dan Shalahuddin (2018:73), “Kamus Data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”. Simbol-simbol yang ada dalam kamus data adalah sebagai berikut.

**Tabel 2.5** Simbol-simbol dalam Kamus Data

No.	Simbol	Keterangan
1	=	Terdiri Atas
2	+	Dan
3	[]	Memilih Salah Satu Alternatif
4	()	data opsional
5	**	Komentar
6	@	Identifikasi atribut kunci
7		Pemisah alternatif simbol []

## 2.4 Teori Program

### 2.4.1 Pengertian PHP

Menurut Enterprise (2018:1), “PHP merupakan bahasa pemrograman yang digunakan untuk membuat website dinamis dan interaktif. Dinamis artinya, website tersebut bisa berubah-ubah tampilan dan kontennya sesuai kondisi tertentu.

Menurut Habibi, Putra dan Putri (2020:40), “PHP merupakan salah satu bahasa pemrograman berbasis web. PHP adalah bahasa scripting yang menyatu dengan HTML dan dijalankan pada server side”.

### 2.4.2 Penulisan PHP

Cara penulisan skrip PHP ada dua macam, yaitu Embedded Script dan Non Embedded Script, contoh :

### Embedded Script

```

<html>
    <head>
    </ head >
    <body>
<?php
    Echo "Hallo Dunia";
?>
    </body>
</html>

```

### Non Embedded Script

```

<?php
    Echo "<html>";
    Echo"<head>";
    Echo"<title>Mengenal PHP </title>";
    Echo"<head>";
    Echo"<body>";
    Echo"<p> PHP cukupmenyenangkan</p>";
    Echo"</body>";
    Echo"</html>";
?>

```

### 2.4.3 Pengertian MySQL

Menurut Fitri (2020:2), "MySQL merupakan database engine atau server database yang mendukung bahasa database SQL sebagai bahasa interaktif dalam mengelola data. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread, multi-user".

"MySQL adalah DBMS yang open source dengan dua bentuk lisensi. yaitu Free Software (perangkat lunak bebas) dan Shareware (perangkat lunak berpemilik yang penggunaannya terbatas)" (Setyawan & Pratiwi, 2020:52).

#### 2.4.4 Pengertian XAMPP

“XAMPP adalah perangkat lunak (free software) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai server yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain: Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apa pun), Apache, MySQL, PHP dan Perl” (Haqi, 2019:8).

#### 2.4.5 Pengertian Sublime Text

Menurut Habibi dan Alwan (2020:60) “*Sublime Text* merupakan sebuah aplikasi editor digunakan pada kode dan teks yang dapat berfungsi di berbagai platform operating system (OS) dengan menggunakan teknologi Python API. Aplikasi ini diciptakan karena terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah flasibel dan powerful. Aplikasi ini dapat dikembangkan dengan memanfaatkan *sublime-packages*”.