



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Dokumen**

Dokumen menurut kamus besar Bahasa Indonesia merupakan surat yang tertulis atau tercetak yang biasa digunakan sebagai bukti atau keterangan (Badan Pengembangan dan Pembinaan Bahasa Kementerian Pendidikan dan Kebudayaan, 2016).

#### **2.2. Dokumen Elektronik**

Dokumen elektronik adalah setiap informasi elektronik yang dibuat, diteruskan, dikirimkan, diterima atau disimpan dalam bentuk analog, *digital*, elektromagnetik, optikal, atau sejenisnya yang dapat dilihat, ditampilkan atau didengar melalui *computer* atau aplikasi elektronik (P, 2016).

#### **2.3. *Document Management System* (DMS)**

DMS (*Document Management System*) adalah sebuah repositori untuk menyimpan dokumen dan memungkinkan *user* untuk mengambil informasi yang diperlukan. DMS lebih lanjut dikatakan sebagai alur daripada kontrol dokumen *paper-based* dan elektronik untuk digunakan secara efisien. Alur dokumen dapat mencakup penyimpanan, pengambilan dokumen, pemrosesan, pencetakan, dan distribusi dokumen (Ahmad et al., 2017).

DMS merupakan sebuah aplikasi untuk membantu merampingkan operasi dalam penerbitan, pengindeksan, penyimpanan, dan pengambilan sumber daya data elektronik dalam suatu organisasi. DMS dirancang untuk membantu suatu organisasi dalam mengelola, penyimpanan dokumen dengan menyediakan repositori terpusat (Purwanti, 2019).

DMS adalah aplikasi yang digunakan untuk mengelola dokumen-dokumen berbentuk *digital*. DMS menjadikan seluruh dokumen akan berada dalam satu tempat penyimpanan yang dapat diakses oleh orang yang memiliki hak akses. DMS merupakan suatu aplikasi yang bekerja di belakang suatu modul atau fitur aplikasi untuk pengolahan dokumen (Hardiansyah & Dalimunthe, 2016).

DMS memiliki beberapa konsep atau fungsi diantaranya (Roebuck, 2011):

1. *Workflow*: Kolaborasi dalam DMS difasilitasi melalui workflow. Kolaborasi dokumen bisa berupa bentuk assignment atau approval.
2. *Versioning*: DMS menyimpan histori dari setiap versi suatu dokumen.
3. *Security*: Mampu untuk membatasi hak akses user.
4. *Indexing*: Mampu untuk mencari dokumen yang dibutuhkan.

Dari pengertian diatas dapat disimpulkan bahwa DMS merupakan sebuah aplikasi yang bertujuan untuk mengelola dokumen agar dapat lebih mudah diakses, dicari, dan didistribusikan serta memiliki manfaat untuk mengurangi penggunaan kertas atau dokumen fisik sehingga memberikan ruang penyimpanan yang lebih luas.

#### **2.4. *Web***

*Web* secara singkat berarti berbagi informasi, dokumen, dan sumber daya antara pengguna melalui internet. *Website* pertama kali terdiri dari kumpulan teks yang dibuat dengan format *HyperText Markup Language (HTML)* (Jacksi & Abass, 2019). Sedangkan *Web-based platform application* merupakan *platform* yang terhubung dengan internet yang memfasilitasi aplikasi agar dapat diakses dimanapun selama terhubung dengan jaringan internet (Murad, Basukiputra, et al., 2019).

#### **2.5. *HyperText Markup Language (HTML)***

*HyperText Markup Language (HTML)* merupakan sebuah bahasa markah yang digunakan untuk membangun halaman *web*. Bahasa markah merupakan bahasa komputer yang dibuat untuk bekerja menggunakan teks. Bahasa markah menyediakan format untuk dokumen teks (Harris, 2018).

#### **2.6. *PHP***

*PHP: HyperText Preprocessor* merupakan sebuah bahasa *computer* yang lebih fleksibel dan memiliki lebih banyak fitur dibandingkan dengan *HTML* sehingga saat ini *PHP* lebih banyak digunakan oleh para pengembang *website* (Smyth, 2010).

## 2.7. *MySQL*

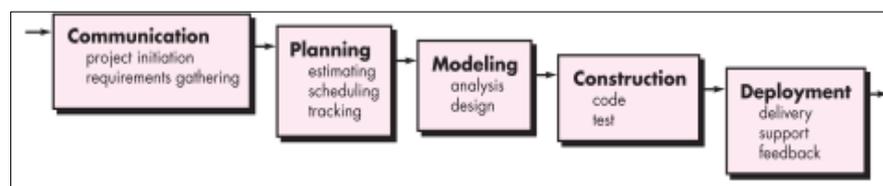
*MySQL* merupakan sebuah *platform* terbuka untuk mengelola relasi *database* dan bisa digunakan berbagai *platform* seperti *Windows*, *Linux*, *MacOS*, dan sebagainya (College, 2019).

## 2.8. *Database*

*Database* merupakan sekumpulan data aplikasiatis. *Database* mendukung penyimpanan dan manipulasi data. *Database* membuat pengolahan data menjadi lebih mudah (College, 2019).

## 2.9. *Waterfall*

Metode *waterfall* merupakan metode yang sering disebut dengan *classic life cycle*, metode ini merupakan metode pengembangan perangkat lunak yang paling banyak dikenal dan sering digunakan di dunia pendidikan maupun dunia industri. Pada metode *waterfall* dokumen pengembangan aplikasi lebih terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum dilanjutkan ke fase berikutnya. Terdapat 5 tahapan dalam pengembangan aplikasi menggunakan metode *waterfall* (Pressman, 2014) yang dapat dilihat pada gambar 2.1.



**Gambar 2.1 *Waterfall Model***

Sumber: (Pressman, 2014)

Untuk memperjelas tahapan dari metode *waterfall* pada gambar 2.2, maka akan dijelaskan lebih lanjut sebagai berikut:

**1. *Communication***

Pada tahap ini dilakukan *requirement gathering* untuk mencari kebutuhan apa saja yang diperlukan oleh *stakeholder*.

**2. *Planning***

Pada tahap ini dibuat *plan*, apa saja yang nantinya bisa dilakukan oleh aplikasi yang akan dibuat.

**3. *Modeling***

Pada tahap ini aplikasi dibuat tampilan *user interface* untuk bisa melihat gambaran besar dari aplikasi yang akan dibuat.

**4. *Construction***

Pada tahap ini aplikasi mulai dibangun fungsi-fungsinya sesuai dengan kriteria kebutuhan yang sudah didapat dan dilakukan *testing* untuk mengetahui apakah aplikasi sudah berjalan sesuai dengan semestinya.

**5. *Deployment***

Pada tahap ini aplikasi akan diimplementasikan guna mendapatkan *feedback* dari *user*.

**2.10. 8 Golden Rules User interface Design**

*8 Golden Rules* merupakan salah satu metode yang digunakan sebagai pedoman untuk membuat sebuah *user interface*. *8 Golden Rules* memiliki 8 tahapan, diantaranya (Shneiderman et al., 2016).

1. *Strive For Consistency*

Urutan tindakan harus konsisten didalam situasi yang serupa.

2. *Seek Universal Usability*

Melihat kebutuhan dari variasi pengguna, memberikan pintasan kepada pengguna yang sudah mengerti terhadap aplikasi.

3. *Offer Informative Feedback*

Memberikan umpan balik kepada pengguna pada saat melakukan setiap tindakan.

4. *Design Dialogs To Yield Closure*

Memberikan panduan yang jelas pada tindakan dan memberikan umpan balik jika tahap tersebut sudah selesai.

5. *Prevent Errors*

Memberikan keterangan yang jelas jika terjadi *error*.

6. *Permit Easy Reversal Of Actions*

Memberikan kemudahan untuk melakukan perubahan terhadap tindakan yang sudah dilakukan.

7. *Keep Users in Control*

Memberikan kebebasan kepada *user* untuk mengontrol menu yang tersedia.

8. *Reduce Short-Term Memory Load*

Memberikan menu yang sederhana untuk mengurangi beban memori jangka pendek agar mudah diingat oleh *user*.

## 2.11. *Unified Modeling Language (UML)*

*UML (Unified Modelling Language)* adalah salah satu alat bantu yang sangat handal di dunia pengembangan aplikasi yang berorientasi obyek. Hal ini disebabkan karena *UML* menyediakan bahasa permodelan visual yang memungkinkan bagi pengembang aplikasi untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2018).

*UML* memiliki sintaks dan semantik. Ketika membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang dibuat berhubungan satu dengan lainnya harus mengikuti standar yang ada (Muslihudin et al., 2016).

Pada penelitian ini akan menggunakan beberapa *diagram* yang dimiliki oleh *UML*, diantaranya:

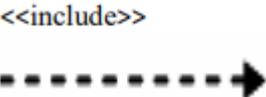
### 1. *Use Case Diagram*

*Use Case* adalah deskripsi fungsi dari sebuah aplikasi dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah aplikasi dengan aplikasinya itu sendiri melalui sebuah cerita bagaimana sebuah aplikasi dipakai.

*Use Case Diagram* digunakan untuk menangkap aspek dinamis dari aplikasi. Secara spesifik, *Use Case Diagram* digunakan untuk mengumpulkan kebutuhan dari sebuah aplikasi baik karena pengaruh

internal maupun eksternal, adapun elemen-elemen yang digunakan dalam *Use Case Diagram* dapat terlihat pada tabel 2.1.

**Tabel 2.1 Elemen *Use Case Diagram***

<b>Nama</b>	<b>Simbol</b>	<b>Keterangan</b>
Aktor		Berfungsi untuk mewakili peran <i>user</i> , aplikasi yang lain, atau alat Ketika berkomunikasi dengan <i>Use Case</i> .
<i>Use Case</i>		Interaksi antara aplikasi dan aktor
Relasi Asosiasi		Relasi antara aktor dengan <i>Use Case</i>
<i>Extend</i>		Berfungsi untuk menunjukkan bahwa suatu <i>Use Case</i> merupakan tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi terpenuhi.
<i>Include</i>		Berfungsi bahwa suatu <i>Use Case</i> merupakan fungsionalitas dari <i>Use Case</i> lainnya.
Relasi Asosiasi Berarah		Asosiasi antara aktor dan <i>Use Case</i> yang mengindikasikan apabila aktor sedang berinteraksi dengan aplikasi.

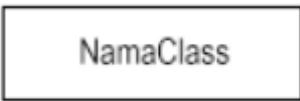
Sumber: (Munawar, 2018)

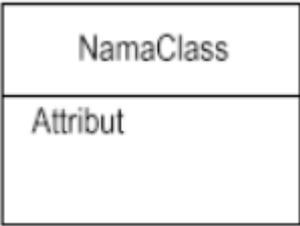
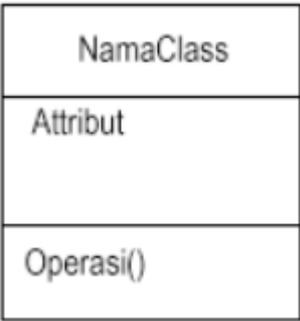
## 2. *Class diagram*

*Class diagram* adalah *diagram* statis. Ini mewakili pandangan statis dari suatu aplikasi. *Class diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek aplikasi tetapi juga untuk membangun kode eksekusi dari aplikasi perangkat lunak.

*Class diagram* menggambarkan atribut, operasi, dan juga *constraint* yang terjadi pada aplikasi. *Class diagram* banyak digunakan dalam pemodelan aplikasi *Object Oriented* karena mereka adalah satu-satunya *diagram UML*, yang dapat dipetakan langsung dengan bahasa berorientasi obyek. Adapun elemen yang terdapat pada *class diagram*, konsep *multiplicity*, dan konsep visibilitas *class diagram* dapat terlihat pada tabel 2.2, tabel 2.3, dan tabel 2.4 dibawah.

**Tabel 2.2 Elemen *Class Diagram***

<b>Notasi</b>	<b>Nama</b>	<b>Deskripsi</b>
	Nama <i>Class</i>	Nama dari <i>Class</i> .

Notasi	Nama	Deskripsi
	<i>Attribute</i>	<i>Property</i> dari sebuah <i>class</i> , berfungsi untuk melukiskan batas nilai yang mungkin ada pada obyek dari <i>class</i> .
	Operasi	Sesuatu yang bisa dilakukan oleh sebuah <i>class</i> atau yang bisa <i>class</i> lain dapat lakukan untuk sebuah <i>class</i> .

Sumber: (Munawar, 2018)

**Tabel 2.3 Class diagram Multiplicity**

Simbol	Keterangan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau satu atau lebih
1..*	Satu atau lebih
0..1	Boleh tidak ada, maksimal satu
n..n	Batasan antara. Misalnya 1..4 mempunyai arti minimal satu maksimal empat

Sumber: (Munawar, 2018)

**Tabel 2.4 Class diagram Visibility**

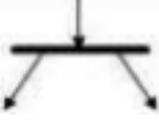
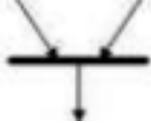
<b>Simbol</b>	<b>Visibility</b>	<b>Keterangan</b>
+	<i>Public</i>	Fitur (sebuah operasi atau atribut) bisa diakses oleh <i>class</i> manapun.
-	<i>Private</i>	Fitur ini hanya boleh digunakan oleh <i>instance</i> dari <i>class</i> .
#	<i>Protected</i>	Fitur ini hanya boleh digunakan oleh <i>instance class</i> dan anak-anaknya.
~	<i>Package</i>	Fitur ini hanya bisa diakses langsung oleh <i>instance</i> sebuah <i>class</i> pada <i>package</i> yang sama

Sumber: (Munawar, 2018)

### **3. Activity diagram**

*Activity diagram* menggambarkan aspek dinamis dari aplikasi. Logika procedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *Activity diagram*. *Activity diagram* bertujuan untuk menangkap tingkah laku dinamis dari aplikasi dengan cara menunjukkan aliran pesan dari satu aktifitas ke aktifitas lainnya. Pada gambar 2.6 terdapat simbol-simbol yang digunakan dalam pembuatan *Activity diagram*.

**Tabel 2.5 Simbol *Activity diagram***

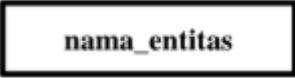
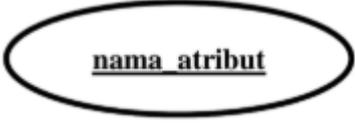
Simbol	Keterangan
	Posisi awal
	Posisi akhir
	<i>Activities</i> (Aktifitas)
	<i>Decision</i> (keputusan)
	<i>Fork Node</i> (Percabangan)
	<i>Join Node</i> (Penggabungan)

Sumber: (Munawar, 2018)

## 2.12. Entity Relationship Diagram

*Entity relationship diagram* adalah pemodelan awal basis data yang dapat memberikan *template* perancangan *database* yang paling banyak digunakan. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan *database* relasional. ERD memiliki beberapa notasi, tujuannya adalah menunjukkan objek data dan hubungan yang ada pada objek tersebut. Pada tabel 2.6 adalah symbol-simbol yang digunakan dalam ERD.

**Tabel 2.6. Simbol Entity Relationship Diagram**

Simbol	Keterangan
	Suatu benda yang eksistensinya terpisah atau berbeda. Contoh: pegawai, guru, mahasiswa, dosen.
	Hubungan yang terjadi antara satu entitas dengan entitas lainnya
	Karakteristik dari entitas atau dari relasi yang menyediakan penjelasan detail tentang entitas atau relasi tersebut.
	Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki kemungkinan jumlah pemakaian

Sumber: (Syahidi, 2018)

Selain simbol yang dijelaskan diatas, terdapat batasan jumlah entitas yang berhubungan melalui satu relasi yang disebut *Cardinality Ratio*. Adapun, jenis-jenisnya sebagai berikut:

1. *One to One (1:1)*

Perbandingan entitas pertama dan kedua berbanding satu berbanding satu.



**Gambar 2.2. one to one**

Sumber: (Syahidi, 2018)

2. *One to Many (1:M)*

Perbandingan entitas pertama dan kedua berbanding satu berbanding banyak.



**Gambar 2.3. one to many**

Sumber: (Syahidi, 2018)

3. *Many To One (M:1)*

Perbandingan entitas pertama dan kedua berbanding banyak berbanding satu.



**Gambar 2.4. *many to one***

Sumber: (Syahidi, 2018)

4. *Many to Many (M:M)*

Perbandingan entitas pertama dan kedua berbanding banyak berbanding banyak.



**Gambar 2.5. *many to many***

Sumber: (Syahidi, 2018)

**2.13. *Black-box Testing***

Menurut Tri Sandhika Jaya, *Black-box Testing* merupakan teknik pengujian perangkat lunak yang berfokus pada fungsional dari perangkat lunak tersebut. Cara *Black-box Testing* memungkinkan *developer* perangkat lunak untuk membuat kumpulan input untuk melatih seluruh syarat fungsional suatu program (Jaya, 2018).

*Black-box Testing* menurut (Febiharsa et al., 2018) merupakan pengujian *software* yang tidak mengacu kepada struktur internal atau tidak memerlukan

pengetahuan khusus pada *code* yang diterapkan pada suatu aplikasi dan pengetahuan tentang pemrograman, jadi hanya berfokus kepada fungsionalitas dari suatu *software* yang sedang diuji.

Menurut (Astuti, 2018) *Black-box Testing* merupakan teknik pengujian yang hanya dilakukan dengan mengamati hasil eksekusi melalui data uji dan fungsional dari *software* yang diuji. Jadi pengujian hanya dilakukan dengan melihat penampilan luarnya saja tanpa melihat bagian dalamnya.

*Black-box Testing* merupakan pengujian perangkat lunak dari sisi spesifikasi fungsionalitas tanpa menguji *code* dan *design* program. Proses *testing* dilakukan hanya untuk mengetahui apakah fungsi, *input* dan *output* dari perangkat lunak yang diuji sesuai dengan spesifikasi yang dibutuhkan. Metode *testing Black-box* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang diinginkan banyaknya data yang diuji dapat dihitung dari banyaknya kolom data yang akan diuji (Cholifah et al., 2018). Metode *Black-box Testing* memiliki beberapa keuntungan yaitu (Jaya, 2018):

1. *Tester* tidak harus memiliki pengetahuan tentang bahasa pemrograman tertentu.
2. *Testing* dilakukan dari sisi *user*, hal ini membantu mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.

Dari beberapa pengertian diatas dapat disimpulkan bahwa *Black-box Testing* merupakan sebuah metode pengujian perangkat lunak yang hanya berfokus kepada

sisi fungsionalitas dari perangkat lunak tersebut tanpa harus melihat sisi *design* dan *code* dari perangkat lunak yang diuji.

Karena metode ini pengujian hanya melihat dari sisi fungsional, maka penguji tidak harus memiliki pengetahuan yang mendalam tentang bahasa pemrograman tertentu. Oleh karena itu, pada penelitian kali ini *Black-box Testing* merupakan metode yang dipilih sebagai pedoman untuk melakukan pengujian terhadap *aplikasi* yang dibuat.

#### **2.14. Skala Likert**

Skala *likert* merupakan salah satu skala pengukuran yang dikembangkan oleh Likert. Skala pengukuran ini memiliki empat atau lebih pertanyaan-pertanyaan yang dapat dikombinasikan agar membentuk sebuah nilai akhir atau skor untuk merepresentasikan sifat individu atau perilaku. Skala *likert* merupakan suatu skala yang umum digunakan dalam kuisioner (Maryuliana et al., 2016).

Pada skala ini biasanya menggunakan beberapa pilihan dengan format:

1. Sangat Setuju
2. Setuju
3. Tidak Setuju
4. Sangat Tidak Setuju

Skala diatas dapat diberi skor untuk menentukan nilai kuantitatif.

## **2.15. Wawancara**

Wawancara adalah sebuah metode untuk mengumpulkan informasi yang akan digunakan untuk pengembangan kegiatan edukasi atau komunikasi (Aubel, 2011).

## 2.16. Penelitian Terdahulu

**Tabel 2.7 Penelitian Terdahulu**

Penulis	Judul	Permasalahan	Penelitian Terdahulu	Hasil Penelitian	Yang Diadopsi Dari Jurnal
Denny Hardiansyah, Nurmaini Dalimunthe	Analisis Penerimaan Pengguna Terhadap <i>Document Management System</i> (DMS) Dengan Model TAM (Hardiansyah & Dalimunthe, 2016) <i>Volume 2 No.1</i>	Kurangnya sosialisasi penggunaan DMS terutama untuk pegawai baru dalam penelusuran dokumen yang dihasilkan oleh unit kerja intern maupun referensi.	Menganalisis pengaruh faktor kemudahan dan manfaat terhadap penerimaan DMS.	Secara kuantitatif, korelasi variabel kemudahan dan manfaat mempunyai hubungan baik terhadap penerimaan pengguna DMS.	Menerapkan DMS sebagai sarana untuk mengkolaborasi para guru untuk berbagi dokumen.
Analy R. Mendoza, Cristina Alvarez, Henriette Mae Castillo, Maricel Manongsong, Andromeda Santiago	<i>Electronic Document Management System Implementing Internet Of Things.</i> (Mendoza et al., 2019) <i>Volume 10, No.2</i>	Penggunaan kertas yang tinggi dan membutuhkan tempat penyimpanan yang luas.	Pengembangan <i>Document Management System</i> untuk meningkatkan efisiensi, mengurangi penggunaan kertas, serta memperluas ruang penyimpanan.	<i>Document Management System</i> untuk mengelola dokumen.	Menerapkan <i>Document Management System</i> sebagai media untuk menyimpan dan berbagi dokumen.

Penulis	Judul	Permasalahan	Penelitian Terdahulu	Hasil Penelitian	Yang Diadopsi Dari Jurnal
Fuadz Hasyim, Andi Wijaya	Peningkatan Mutu Akreditasi Perguruan Tinggi Menggunakan Aplikasi Manajemen Dokumen Elektronik <i>Volume 4 No.2</i>	Sulitnya melakukan inventarisasi terhadap data dan informasi pendukung yang menjadi syarat kelengkapan Menyusun boring akreditasi.	Menganalisis bagaimana DMS dapat mempengaruhi pengelolaan arsip dokumen pendukung akreditasi.	Penggunaan DMS untuk mengelola arsip dokumen pendukung akreditasi.	Melakukan wawancara untuk mengetahui kebutuhan dari aplikasi yang akan dibangun.
Fatin Hazwani Kiplie, Syaiful Farik Mat Yatin, Maizurah Angutim, Nur Hanani Abdul Hamid	<i>System Development for Document Management System Volume 8 No. 9</i>	Bagaimana menentukan pengembangan aplikasi yang baik untuk mengembangkan DMS.	Menganalisis pengembangan aplikasi yang baik untuk pengembangan DMS.	Pemilihan cara pengembangan aplikasi yang baik akan menghasilkan aplikasi yang bisa dimengerti oleh pengguna.	Melakukan pengujian setelah aplikasi selesai dibangun.
I Putu Susila Handika, I Gede Totok Suryawan	Rancang Bangun <i>Document Management System Untuk Mengelola Dokumen Standard Operational Procedure. Volume 4 No. 3</i>	Proses pencarian dan proses distribusi dokumen membutuhkan waktu yang cukup lama dikarenakan masih menggunakan cara tradisional.	Merancang dan membangun <i>document management</i> aplikasi untuk mengelola dokumen SOP.	Aplikasi DMS berbasis <i>web</i> .	Menerapkan aplikasi DMS berbasis <i>web</i> .

<b>Penulis</b>	<b>Judul</b>	<b>Permasalahan</b>	<b>Penelitian Terdahulu</b>	<b>Hasil Penelitian</b>	<b>Yang Diadopsi Dari Jurnal</b>
Eni Purwanti	Pemanfaatan <i>Document Management System</i> dalam Penyimpanan dan Temu Kembali Arsip di Fakultas Teknik Universitas Gadjah Mada. <i>Volume 5 No. 3</i>	Dampak dari penggunaan DMS di Fakultas Teknik UGM.	Mengkaji implementasi DMS di Fakultas teknik Universitas Gadjah Mada.	Penggunaan DMS memiliki keunggulan dan kemanfaatan yang besar dalam pengaturan dokumen di Fakultas Teknik UGM.	Menerapkan manfaat <i>Document Management System</i> untuk mengelola dokumen di SMAIT Asy-Syukriyyah.

Berdasarkan penelitian terdahulu dari (Hasyim & Wijaya, 2019), (Kiplie et al., 2018), dan (Handika & Suryawan, 2017) pada tabel 2.7, membantu dalam penyelesaian penelitian “Rancang Bangun *Document Management System* Berbasis Web Untuk Smaıt Asy-Syukriyyah” dengan mengadopsi yaitu pembuatan aplikasi DMS berbasis *web*, dengan menggunakan *UML* sebagai media pemodelannya.