

## BAB II TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Penelitian terdahulu ini akan menjadi acuan penulis dalam membuat laporan akhir sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Berikut dalam tabel 2.1 merupakan penelitian terdahulu.

**Tabel 2.1** Penelitian Terdahulu

No.	Judul Referensi	Penulis	Tahun	Hasil Penelitian
1.	Perancangan Aplikasi <i>E-Journal On-line</i> LPPM Universitas Borobudur Menggunakan CMS OJS	Permadi Surya	2017	Sistem <i>online</i> jurnal fakulta ilmu komputer di Universitas Borobudur dibuat menggunakan sistem CMS ( <i>Content Management System</i> ).
2.	Perancangan Aplikasi Jurnal <i>Online</i> dengan <i>Open Journal System</i> (OJS) pada Jurnal STMIK IBBI	Tintin Chandra	2020	Aplikasi Jurnal <i>Online</i> dengan <i>Open Journal System</i> (OJS) berjalan baik mulai dari proses <i>submit</i> artikel oleh <i>user</i> (penuis) sampai kepada proses publikasi.
3.	Pengembangan Sistem <i>Authoring</i>	Didit Setia Budi &	2019	Berdasarkan hasil pengujian <i>Usability</i> , aplikasi

	<i>Tools</i> untuk Publikasi Jurnal	Ardiansyah		<i>Authoring Tools</i> Jurnal Berbasis <i>Web</i>
<b>No.</b>	<b>Judul Referensi</b>	<b>Penulis</b>	<b>Tahun</b>	<b>Hasil Penelitian</b>
	Ilmiah Berbasis <i>Web</i>			mendapatkan skor pengujian sebesar 89.58 (dengan skala 0..100) maka dinyatakan <i>acceptable</i> yang artinya <i>Authoring Tools</i> Jurnal Berbasis <i>Web</i> diterima dan dapat memuaskan kebutuhan pengguna.
4.	Aplikasi Pengeolaan Publikasi Mahasiswa Berbasis Web Di Universitas Muhammadiyah Surakarta	Setyo Andik Cahyo Putro	2013	Pada hasil penelitian menunjukkan bahwa pengelolaan publikasi mahasiswa di UMS dapat dibantu dengan aplikasi yang telah dibangun, yaitu dalam proses mahasiswa upload paper dan cek status paper. Proses biro mendownload paper dan menentukan status paper mahasiswa apakah lolos atau tidak. Dan terpublikasikan pada proses menentukan jenis paper

				oleh redaksi jurnal sampai paper terpublikasikan.
--	--	--	--	---

## **2.2. Rancang Bangun**

Rancang bangun adalah kegiatan menerjemahkan hasil analisa kedalam bentuk paket perangkat lunak, kemudian menciptakan sistem atau memperbaiki sistem yang sudah ada (Jh, A. R., & Prastowo, A. T. 2021)

## **2.3. Publikasi**

Publikasi merupakan tindakan penerbitan, dan juga mengacu pada setiap salinan.

Menurut Astika (2014:28) “Publikasi adalah sesuatu kegiatan yang dilakukan untuk menyediakan atau menyebarkan sebuah informasi kepada masyarakat umum dengan cara bermacam-macam mulai dari buku, teks, gambar, konten audio visual dan website sehingga masyarakat mengetahui informasi yang disediakan untuk konsumen sehingga kegiatan promosi diharapkan mampu mempengaruhi masyarakat untuk mememanfaatkannya”.

Menurut lesly (2014;58) “Publikasi adalah penyebaran pesan yang direncanakan dan dilakukan untuk kepentingan tertentu dari organisasi dan perorangan tanpa pembayaran pada media”.

Dari pengertian di atas dapat disimpulkan bahwa publikasi adalah penyampaian informasi kepada masyarakat melalui media-media komunikasi.

## **2.4. Artikel**

Artikel merupakan karangan faktual (nonfiksi) tentang masalah tertentu yang disajikan dengan lengkap untuk dimuat di surat kabar, majalah, tabloid, buletin, media siber, dan lainnya. (Ibda, 2019).

## 2.5. Penelitian

Menurut (Dr. Sandu Siyoto, SKM., 2015) Penelitian adalah suatu penyelidikan terorganisasi atau penyelidikan yang hati-hati dan kritis dalam mencari fakta untuk menentukan sesuatu. Kata penelitian adalah terjemahan dari kata *research* yang berasal dari bahasa Inggris. Kata *research* terdiri dari dua kata yaitu *re* yang berarti kembali dan *to search* yang berarti mencari. Jadi dapat disimpulkan bahwa pengertian penelitian (*research*) adalah mencari kembali suatu pengetahuan.

## 2.6. Database

*Database* atau basis data adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir sehingga mampu dan disimpan secara terintegrasi dengan menggunakan metode tertentu sehingga mampu memenuhi informasi dengan memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna. (Marlinda, 2004).

*Database* adalah kumpulan file yang berelasi, relasi tersebut biasanya ditunjukkan dengan kunci tiap file yang ada, satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu file terdapat *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*.

Istilah yang sering digunakan pada pembahasan *database*, yaitu :

1. DBMS, yaitu sistem perangkat lunak yang menyediakan layanan bagi user untuk membuat, mengontrol, dan mengakses *database*.

2. Tabel, merupakan kumpulan data yang diorganisasikan dalam baris (*record*) dan kolom (*field*). Dalam satu database biasanya terdiri dari beberapa tabel.
3. *Field*, merupakan kolom pada tabel yang memiliki ukuran dan tipe data tertentu.
4. *Record*, merupakan baris data yang memiliki nilai untuk masing-masing field pada tabel. Satu tabel biasanya terdiri dari banyak *record*.
5. *Key*, merupakan suatu kolom (*field*) yang dapat dijadikan kunci dalam operasi tabel.
6. SQL, merupakan singkatan dari *Structured Query Language*, yaitu bahasa yang digunakan untuk mengakses dan memanipulasi *database*. (Abdulloh, 2018).

## 2.7. MySQL

*MySQL* merupakan sebuah *Database Management System* atau sering disingkat DBMS yang dijalankan menggunakan perintah SQL (*Strutured Query Language*) yang populer digunakan untuk pembuatan *website*. Selain itu SQL dipuji karena kesederhanaan *syntax* yang pendek dan mudah dipahami. *MySQL* juga termasuk ke dalam RDBMS atau *Relational Database Management System*, dimana didalam struktur databasenya sehingga ketika proses pengambilan data menggunakan metode *relational database*. Yang juga menjadi penghubung antara perangkat lunak dan *database* server.

Jika kita akan mengakses *database MySQL* menggunakan PHP sebagai salah satu perintah/fungsi utama dari PHP untuk mengambil data adalah *Mysql query*), *query* ini akan menjadi permanen bagi fungsi *mysql query* tersebut. (Haris, 2012).

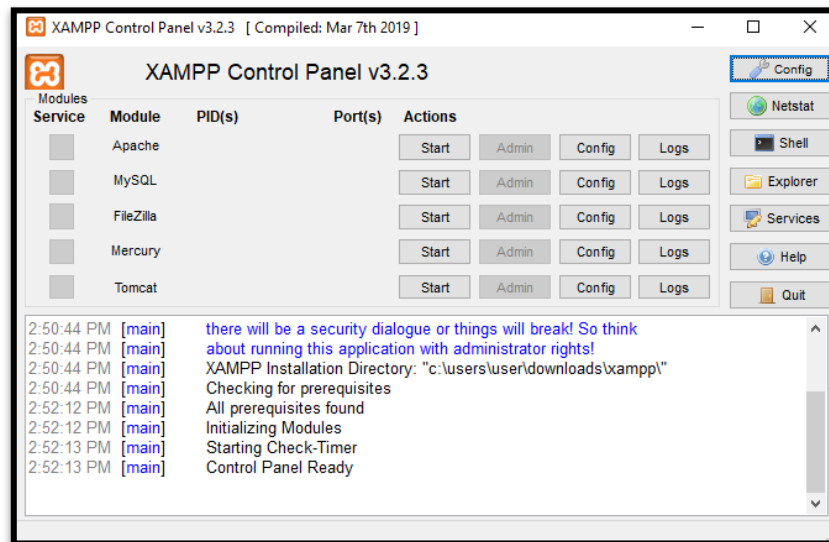
Perintah atau *syntax query MySQL* dimulai dengan salah satu perintah berikut:

1. *CREATE*, perintah untuk membuat database baru.
2. *INSERT*, perintah untuk menambahkan data pada tabel *database*.
3. *SELECT*, perintah untuk menampilkan data dari tabel *database*.
4. *UPDATE*, perintah untuk melakukan perubahan data pada tabel *database*.
5. *DELETE*, perintah untuk menghapus data pada tabel *database*.
6. *DROP*, perintah untuk menghapus tabel dan *database*.
7. *ALTER*, perintah untuk melakukan perubahan struktur tabel yang telah dibuat pada *database*.

Query MySQL memiliki banyak kelonggaran dalam penulisannya. Spasi dan enter tidak begitu diperhatikan dalam penulisannya. Artinya kalau kelebihan salah tulisan enter atau spasi, maka tidak menjadi masalah atau error ketika perintah itu dijalankan. Namun, untuk menjadi catatan ketika akan membuat database adalah harus memberikan tanda titik koma (;) pada setiap akhir penulisan perintah MySQL.

## **2.8. XAMPP**

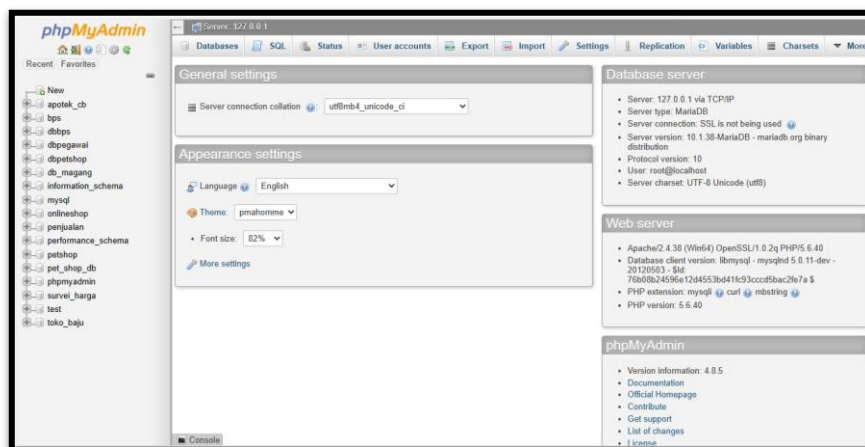
Pengertian XAMPP secara akronim berasal dari singkatan masing-masing program, yakni X (*Cross Operationg System*), A (Apache), M (MySQL-MariaDB), P (PHP) dan P (Perl).



Gambar 2.1 Tampilan XAMPP

## 2.9. *phpMyAdmin*

*phpMyAdmin* merupakan suatu perangkat lunak yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL (keperluan manajemen basis data) melalui jejaring internet. *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya untuk pengelolaan basis data, tabel-tabel, bidang (fields), relasi (*relations*), *indeks*, pengguna (*users*), perizinan (*permissions*), dan lain-lain. (Suryadi, Pratama, and Phonna 2020).



**Gambar 2.2** Tampilan phpMyAdmin

## **2.10. Bahasa Pemrograman**

Bahasa pemrograman adalah *software* bahasa komputer yang digunakan dengan cara merancang atau membuat program yang sesuai dengan struktur dan metode yang dimiliki oleh bahasa pemrograman itu sendiri. Komputer mengerjakan informasi berdasarkan kumpulan perintah program yang telah dibuat oleh program itu sendiri. Kumpulan perintah ini harus dimengerti oleh komputer, berstruktur tertentu (*syntax*) dan bermakna. Bahasa pemrograman merupakan notasi untuk memberikan secara tepat program komputer. (Dipraja, 2013).

### **3.10.1. Hypertext Markup Language (HTML)**

HTML singkatan dari *Hypertext Markup Language* dan berguna untuk menampilkan halaman *web*. Berdasarkan penjelasan di atas, penulis menyimpulkan. HTML adalah bahasa pemrograman standar yang digunakan untuk menampilkan teks, gambar, video, atau audio pada halaman *web*. Elemen HTML dimulai dengan tag awal, yang diikuti dengan isi elemen dan tag akhir. Tag terakhir termasuk simbol/diikuti oleh tipe elemen, misalnya `</head>`. Sebuah dokumen HTML standar terlihat seperti ini :

```
<html>
  <head>
    <title> ini judul dokumen html </title>
  </head>
  <body>
    Teks ini adalah teks yang akan muncul
  </body>
</html>
```



### **3.10.2. Hypertext Preprocessor (PHP)**

PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membagun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di-*parsing* di dalam *web server* oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali *web server*. Karena pemrosesan program PHP dilakukan didalam lingkungan *web browser*, PHP dikatakan sebagai bahasa sisi server (*server-side*). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “*View Source*” pada *web browser* yang mereka gunakan.

### **3.10.3. Cascading Style Sheet (CSS)**

*Cascading Style Sheet* atau yang biasa disingkat dengan CSS berguna untuk menyederhanakan kodingan yang dimana untuk mendesain halaman *website* atau tampilan *website*. CSS memiliki banyak fungsi yang dapat membantu dalam pembuatan *website*, dengan fungsi mengatur jenis *font*, warna teks, baris antar paragraf, ukuran kolom, ukuran baris, dan dapat mengganti *background* yang digunakan. CSS sangat populer digunakan pada seluruh *website* yang ada di dunia internet. Selain desainnya yang kompleks, CSS sendiri juga mudah untuk dipelajari dan digunakan. (Andy Febrianto, Achmadi Sentot, Agung Panji Sasmito, 2021).

### **3.10.4. JavaScript**

*JavaScript* adalah bahasa yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan pada sebuah browser menjadi lebih interaktif, tidak sekedar indah saja. *JavaScript* memberikan beberapa fungsionalitas ke dalam halaman *web*, sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antar muka *web*.

## 2.11. *Framework*

*Framework* atau kerangka kerja adalah komponen siap pakai yang digunakan *developer* untuk menangani berbagai permasalahan dalam pemrograman, seperti pemanggilan variabel, file, koneksi ke *database* dan sebagainya. Supaya lebih fokus dan mampu menyelesaikan *software* menjadi lebih cepat dan efektif (Jh, A. R., & Prastowo, A. T. 2021).

*Framework* adalah kumpulan intruksi-intruksi yang yang dikumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam memanggilnya tanpa harus menuliskan *syntax* program yang sama berulang-ulang serta dapat menghemat waktu (Sallaby & Kanedi, 2020).

*Framework* juga bisa diartikan sebagai komponen – komponen pemrograman yang sudah jadi dan siap untuk digunakan kapan saja, sehingga pengembang aplikasi tidak perlu lagi membuat scrip yang sama untuk tugas – tugas yang sama (Kristianto, G. A. 2019).

### 2.11.1. *CodeIgniter*

Menurut A Budiman, S Sunariyo, J Jupriyadi (2021) mengatakan bahwa *CodeIgniter* adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman PHP bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis web. *CodeIgniter* memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya. *CodeIgniter* bersifat *open source* dan menggunakan model basis MVC (*Model View Controller*), yang merupakan model konsep modern saat ini. *CodeIgniter* juga menawarkan banyak *library* yang dapat digunakan.

*Codeigniter* adalah sebuah *framework* php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan

*developer* atau *programmer* dalam membangun sebuah aplikasi berbasis web tanpa harus membuatnya dari awal. Dengan menggunakan PHP *CodeIgniter* diharapkan dapat memudahkan *developer* atau *programmer* untuk membuat aplikasi *web* dengan cepat dan mudah dibandingkan dengan membuat sebuah web dari awal (Destiningrum & Adrian, 2017).

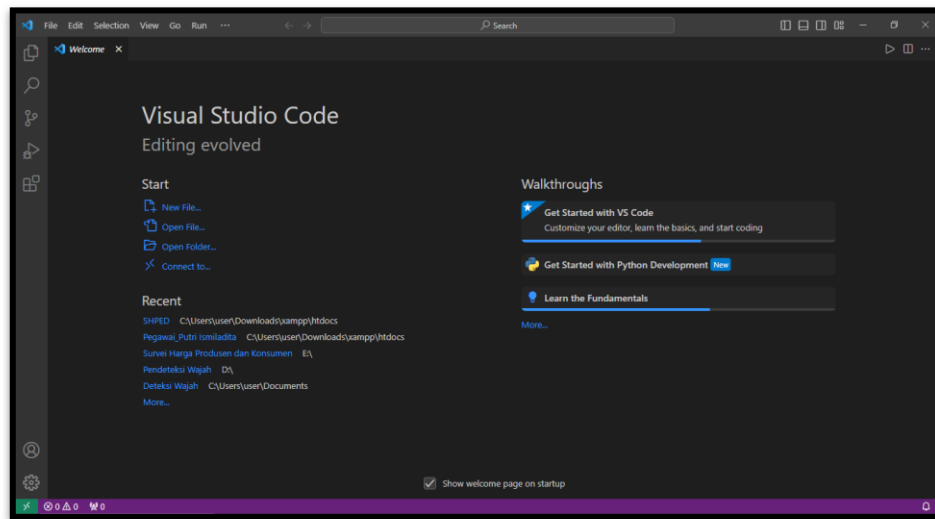
*CodeIgniter* adalah Sebuah *framework* php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuatnya dari awal (Sallaby & Kanedi, 2020).

### **2.11.2. *Bootstrap***

*Bootstrap* adalah paket aplikasi siap pakai untuk membuat halaman *front-end* maupun *back-end* dari sebuah *website*, dan *bootstrap* merupakan template desain web dengan fitur plus. (Setiady dan Syah, 2018).

### **2.12. *Visual Studio Code***

*Visual Studio Code* merupakan aplikasi *cross platform* yang dapat digunakan berbagai sistem operasi seperti windows, Linux, dan Mac os. *Visual Studio Code* termasuk *software* yang ringan namun kuat editor sumbernya dengan deksto. Menggunakan berbagai macam bahasa pemrograman seperti PHP, Java, *JavaScript*, Go, C++, dan masih banyak lainnya.



**Gambar 2.3** Tampilan *Visual Studio Code*


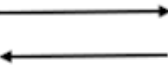


### 2.13. Diagram Konteks

Diagram konteks merupakan sebuah bagian dari *Data Flow Diagram* yang digunakan untuk menentukan konsep dan batasan sistem pada saat pemodelan. Diagram ini mencakup hubungan antara sistem dengan entitas-entitas di luar sistem seperti sistem lain, kelompok organisasi, dan penyimpanan data eksternal. Diagram konteks sering disebut sebagai Level-0 dan merupakan bagian terpenting dalam pembuatan *Data Flow Diagram*.

### 2.14. *Data Flow Diagram* (DFD)

DFD adalah suatu diagram yang menggambarkan aliran data dari sebuah proses yang sering disebut dengan sistem informasi. Di dalam data flow diagram juga menyediakan informasi mengenai *input* dan dari tiap entitas dan proses itu sendiri.

Dalam diagram alir data juga tidak mempunyai kontrol terhadap *flow*-nya, sehingga tidak adanya aturan terkait keputusan atau pengulangan. Bentuk penggambaran berupa data flowchart dengan skema yang lebih spesifik.

Simbol	Keterangan Fungsi
	<i>External entity</i> /Entitas luar. Simbol ini menunjukkan orang, organisasi, atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem.
	<i>Data Flow</i> diberi simbol panah. Simbol ini menunjukkan satu data tunggal atau kumpulan logis suatu data, selalu diawali atau diakhiri pada suatu proses.
	Proses adalah aktivitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, bisa berupa manual maupun terkomputerisasi.
	<i>Data Store</i> adalah kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam <i>data store</i> .



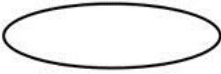

**Gambar 2.4 DFD**

### 2.15. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram (ERD)* adalah sebuah gambar atau diagram yang digunakan untuk menggambarkan hubungan antara entitas (objek) dalam sebuah *database*. Dalam ERD, entitas (objek) direpresentasikan sebagai kotak dengan atribut-atribut yang terkait dengan entitas tersebut. Hubungan antara entitas ditunjukkan oleh tanda panah atau garis yang menghubungkannya. ERD memungkinkan database untuk memvisualisasikan struktur database dengan jelas dan memahami bagaimana entitas saling terkait.

Bentuknya seperti diagram yang menjelaskan hubungan antar objek data. menggambarannya dibutuhkan:

1. Notasi ialah seperangkat lambang yang menggambarkan data.
2. Simbol sebagai lambang sebagai penanda.
3. Bagan merupakan rancangan atau skema untuk mempermudah penafsiran.
4. Dan lain sebagainya.

Simbol	Keterangan
	Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik
	Relasi, yaitu hubungan yang terjadi antara salah satu lebih entitas. Jenis hubungan antara lain. one to one, One to many, dan many to many.
	Atribut, yaitu karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas.
	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya.

**Gambar 2.5 ERD**

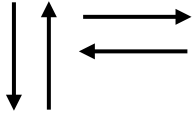
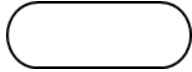



### 2.16. *Flowchart*







*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.

Dalam penulisan *flowchart* dikenal dua model yaitu *flowchart* sistem dan *flowchart* program. *Flowchart* sistem merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antara peralatan tersebut. *Flowchart* program merupakan diagram alir yang menggambarkan suatu logika dari suatu prosedur pemecahan masalah.


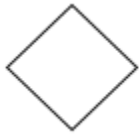



Simbol-simbol yang digunakan dalam *flowchart* disertai dengan keterangan fungsinya dapat dilihat pada tabel 2.2 berikut:

**Tabel 2.2 Simbol Diagram *Flowchart***

NO	Simbol	Keterangan
1.	<p><i>Flow Direction</i> <i>Symbol</i></p> 	<p>Simbol ini digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses.</p>
2.	<p>Terminal (mulai atau berhenti)</p> 	<p>Simbol ini digunakan untuk menunjukkan awal kegiatan (<i>start</i>) atau akhir dari suatu kegiatan (<i>stop</i>).</p>
3.	<p><i>Input dan Output</i></p> 	<p>Untuk menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.</p>
4.	<p>Proses (Pengolahan)</p> 	<p>Untuk menunjukkan pengolahan yang dilakukan oleh komputer atau PC.</p>
5.	<p><i>Connector</i></p> 	<p>Simbol suatu keluaran atau masukan prosedur atau proses dalam lembar atau halaman yang sama.</p>

6.	<i>Offline Connector</i> 	Simbol untuk keluaran atau masukan prosedur atau proses dalam lembar atau halaman yang berbeda.
7.	<i>Document</i> 	Untuk menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.
<b>NO</b>	<b>Simbol</b>	<b>Keterangan</b>
8.	<i>Manual Input</i> 	Berfungsi untuk pemasukan data secara <i>manual on-line keyboard</i> .
9.	<i>Preparation</i> 	Berfungsi untuk mempersiapkan penyimpanan yang sedang atau akan digunakan sebagai tempat pengolahan di dalam <i>storage</i> .
10.	<i>Manual Operation</i> 	Berfungsi untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer atau PC.
11.	<i>Multiple Document</i> 	Sama seperti simbol <i>document</i> , hanya saja dokumen yang digunakan lebih dari satu dalam simbol ini.



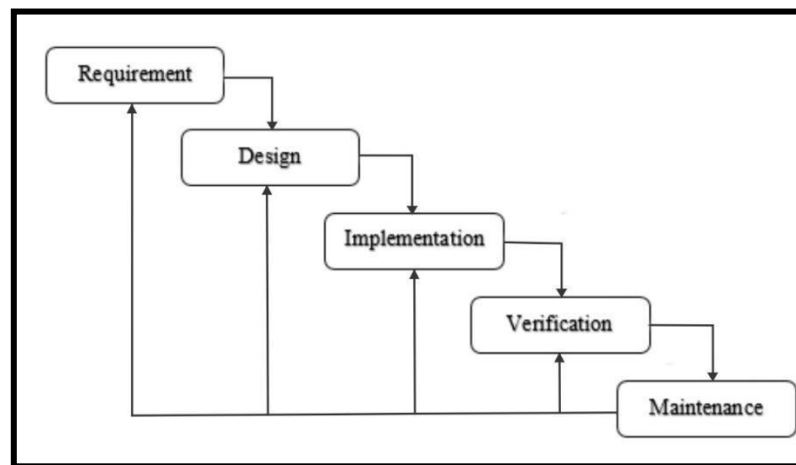
12.	<i>Predefined</i> 	Untuk pelaksanaan suatu bagian (subprogram) atau prosedur.
13.	<i>Decision (Keputusan)</i> 	Menunjukkan suatu perbandingan yang harus dibuat bila hasilnya “ya”, maka alir data menunjukkan ke suatu tempat, bila “tidak” maka akan menuju ke tempat lain.
14	<i>Display</i> 	Simbol yang digunakan untuk menyatakan perangkat <i>output</i> yang digunakan.
<b>NO</b>	<b>Simbol</b>	<b>Keterangan</b>
15.	<i>Database</i> 	Simbol yang digunakan untuk menyatakan data disimpan dalam <i>database</i> .
16.	<i>On-line storage</i> 	Simbol yang menyatakan <i>input</i> yang berasal dari <i>disk</i> atau disimpan ke <i>disk</i> .

### 2.17. Metode Pengembangan Sistem

Metode *waterfall* merupakan model air terjun yang menyediakan pendekatan alur hidup perangkat lunak secara seluasial atau terurut dimulai dari analisa, desain pengkodean, pengujian dan tahap pendukung atau *support*. (Rosa dan Shalaluddin, 2018).

Model *waterfall* pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). saat ini model *waterfall* merupakan model pengembangan perangkat lunak yang sering digunakan. Model penggunaan ini melakukan pendekatan secara sistematis dan berurutan. Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesai tahap sebelumnya dan berjalan berurutan. Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan.

Tahapan dari metode *waterfall* dapat dilihat pada gambar dibawah ini :



**Gambar 2.6** Tahapan dari metode Waterfall

### **2.17.1. Analisis Kebutuhan (*Requirement*)**

Seluruh kebutuhan software didapatkan pada fase ini, termasuk didalamnya kegunaan software yang diharapkan pengguna dan batasan software. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya. Dalam membangun sebuah

website tentu diperlukan persiapan serta menganalisa kebutuhan dari software yang akan diteliti dan dibangun.

### **2.17.2. Desain (*Design*)**

Tahap ini dilakukan sebelum melakukan tahapan *coding system*. Tahap ini bertujuan untuk memberikan gambaran seperti apa sistem yang akan dibuat dan bagaimana interface untuk setiap kegiatannya. Tahap ini membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan. Pembuatan desain *website* yang akan dibuat sebelum masuk pada proses pembuatan *coding*. Tujuan diperlukannya tahapan desain ini adalah agar peneliti mempunyai gambaran yang jelas mengenai tampilan dan antarmuka *website* yang kemudian akan dieksekusi menjadi sebuah kode program.

### **2.17.3. Implementation**

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

### **2.17.4. Pengujian (*Verification*)**

Pada tahap ini dilakukan penggabungan modul-modul yang telah dibuat dan dilakukan pengujian untuk mengetahui apakah *software* telah sesuai atau belum sesuai dengan desainnya. Proses ini sering disebut dengan integrasi atau pengujian aplikasi. Pada tahap ini, akan dilakukan penggabungan script/coding *front-end* dan *back-end* serta pengkoneksian ke *database* yang telah dibuat. Setelah proses integrasi selesai, berikutnya masuk pada tahap pengujian aplikasi yang bertujuan untuk mengetahui apakah aplikasi media publikai penelitian dan artikel telah sesuai dengan yang telah dirancang sebelumnya atau belum dan pengujian fungsionalitas dari *website* profil berjalan dengan baik atau tidak.

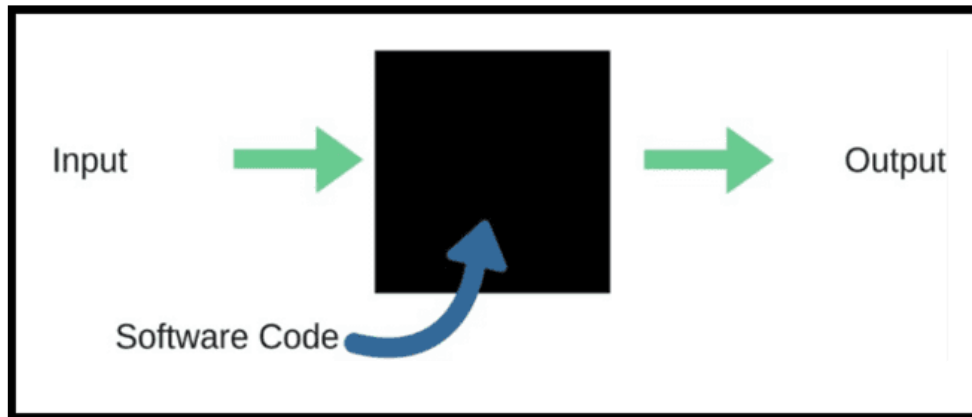
Dengan adanya tahap pengujian, diharapkan dapat mencegah terjadinya kesalahan, bug, atau error pada website profil sebelum digunakan untuk umum. Adapun pengujiannya dilakukan dengan menggunakan server localhost dari XAMPP dimana service Apache dan MySQL di XAMPP Control perlu diaktifkan terlebih dahulu barulah kemudian dapat mengakses dengan URL "*localhost/publikasitekkom*" di web browser yang digunakan. Tampilan hasil pengujian website profil ini akan dibahas pada bab/bahasan selanjutnya . Pengujian fokus pada perangkat lunak secara fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan hasil output yang dihasilkan sesuai dengan yang dibutuhkan.

#### **2.17.5. Maintenance**

Ini adalah tahap akhir dari metode waterfall. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

#### **2.18. Sistem Pengujian Blackbox Testing**

*Black-Box Testing* merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Blackbox testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada Informasi domain. *Black box testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program.



**Gambar 2.7** Sistem Pengujian *Black box testing*

Pengujian *black box* hanya melibatkan *database* antara *input* dan *output*. Pengujian ini menangani kebutuhan pelanggan dari *input* yang *valid* maupun tidak *valid*. Pengujian *black box* memiliki peran penting dalam pengujian perangkat lunak yaitu untuk memvalidasi fungsi keseluruhan sistem apakah telah bekerja dengan baik