

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Beberapa penelitian terkait telah dilakukan pada kajian-kajian terdahulu tentang rancang bangun sistem kepegawaian dengan metode waterfall. Penelitian pertama dilaksanakan oleh Maesaroh et al (2017) yang berjudul "Rancang Bangun Sistem Informasi Kepegawaian (SIMPEG) Dengan SDLC Metode Waterfall Studi Kasus Di Kantor BKPLD Kabupaten Tasikmalaya" yang membandingkan metode perancangan sistem informasi paling optimal untuk Kantor BKPLD Kabupaten Tasikmalaya diantara tiga metode yaitu metode *waterfall*, metode *prototyping*, dan metode *rapid application development* (RAD). Hasilnya metode *waterfall* merupakan metode yang paling sesuai dengan sistem informasi kepegawaian dan lingkungan pembangunan sistem (BKPLD).

Kemudian penelitian selanjutnya dilakukan oleh Widiyanto (2018) berjudul "Analisa Metodologi Pengembangan Sistem Dengan Perbandingan Model Perangkat Lunak Sistem Informasi Kepegawaian Menggunakan Waterfall Development Model, Model Prototype, Dan Model Rapid Application Development (RAD)" yang mengemukakan kelebihan dan kekurangan tiga metode perancangan sistem informasi yaitu metode *waterfall*, metode *prototyping*, dan metode *rapid application development* (RAD). Hasilnya terdapat persamaan kepada ketiga metode tersebut yaitu tahapan dilakukan secara sekuensial artinya setiap tahapan harus diselesaikan secara berturut-turut. Kemudian Cocok digunakan untuk produk software/program yang sudah jelas kebutuhannya di awal, sehingga minim kesalahannya.

Menurut Adawiyah (2020) dalam penelitiannya yang berjudul "Perancangan Sistem Informasi Sekolah Bidang Kepegawaian Berbasis Web Pada SMK Negeri 1 Cibadak Dengan Metode Waterfall" Sistem Informasi digunakan untuk menampilkan informasi data cepat tanpa adanya edukasi yang tidak diperlukan dan menghindari kesalahan yang disebabkan oleh *user*. Pada penelitian yang dilakukan Adawiyah

rancangan sistem informasi kepegawaian berbasis web SMK Negeri 1 Cibadak digunakan metode *waterfall*. Berdasarkan hasil analisa sistem yang didokumentasikan melalui *unified modelling language* (UML), pengguna aplikasi dimudahkan dalam pencarian data dan dokumen guru dan staf. Lebih lanjut aplikasi tersebut mengurangi tingkat kesalahan yang disebabkan *user* seperti pencatatan ganda data kepegawaian dan presensi pegawai.

2.2 ASN Fakultas Teknik

Fakultas Teknik Universitas Tanjungpura memiliki 207 ASN yang terdiri dari 169 ASN Dosen dan 38 ASN Tenaga Kependidikan (Tendik), dengan berbagai jenjang pendidikan, pangkat/golongan ruang dan jabatan fungsional/jabatan akademik. Berikut disajikan profil ASN Dosen dan Tendik di lingkungan Fakultas Teknik Universitas Tanjungpura.

Tabel 2.1 Jumlah ASN Fakultas Teknik Berdasarkan Jenis Kelamin

No.	Jenis Kelamin	Jumlah ASN	
		Dosen	Tendik
1	Lak-laki	108	29
2	Perempuan	61	9
	Total	169	38

Tabel 2.2 Jumlah ASN Fakultas Teknik Berdasarkan Jenjang Pendidikan

No.	Jenjang Pendidikan	Jumlah ASN	
		Dosen	Tendik
1	S3 Doktor	39	-
2	S2 Magister	130	2
3	S1 Sarjana	-	14
4	DIII	-	1
5	SMA/SMK	-	19
6	SMP	-	1
7	SD	-	1

No.	Jenjang Pendidikan	Jumlah ASN	
		Dosen	Tendik
	Total	169	38

Tabel 2.3 Jumlah ASN Dosen Fakultas Teknik Berdasarkan Pangkat/Golongan Ruang

No.	Pangkat / Golongan Ruang	Jumlah Dosen	Persentase
1	Penata Muda, III/a	1	0,59%
2	Penata Muda Tk.I, III/b	60	35,50%
3	Penata, III/c	30	17,75%
4	Penata Tk.I, III/d	24	14,20%
5	Pembina, IV/a	36	21,30%
6	Pembina Tk.I, IV/b	10	5,92%
7	Pembina Utama Muda, IV/c	7	4,14%
8	Pembina Utama Madya Tk.I, IV/d	1	0,59%
9	Pembina Utama, IV/e	0	0,00%
	Total	169	100%

Tabel 2.4 Jumlah ASN Tendik Fakultas Teknik Berdasarkan Pangkat/Golongan Ruang

No.	Pangkat / Golongan Ruang	Jumlah Tendik	Persentase
1	Juru Muda, I/a	0	0,00%
2	Juru Muda Tk.I, I/b	0	0,00%
3	Juru, I/c	1	2,63%
4	Juru Tk.I, I/d	0	0,00%
5	Pengatur Muda, II/a	2	5,26%
6	Pengatur Muda Tk.I, II/b	0	0,00%
7	Pengatur, II/c	1	2,63%
8	Pengatur Tk.I, II/d	4	10,53%
9	Penata Muda, III/a	6	15,79%
10	Penata Muda Tk.I, III/b	11	28,95%
11	Penata, III/c	8	21,05%
12	Penata Tk.I, III/d	4	10,53%
13	Pembina, IV/a	0	0,00%
14	Pembina Tk.I, IV/b	1	2,63%
	Total	38	100%

Tabel 2.5 Jumlah ASN Dosen Fakultas Teknik Berdasarkan Jabatan Akademik

No.	Jabatan Akademik	Jumlah Dosen	Persentase
1	Guru Besar	4	2,37%
2	Lektor Kepala	51	30,18%
3	Lektor	54	31,95%
4	Asisten Ahli	28	16,57%
5	Tenaga Pengajar	32	18,93%
	Total	169	100%

Tabel 2.6 Jumlah ASN Fakultas Teknik Berdasarkan Masa Kerja

No.	Masa Kerja	Jumlah ASN	
		Dosen	Tendik
1	> 30 Tahun	15	14
2	20 - 30 Tahun	76	16
3	10 - 20 Tahun	33	8
4	< 10 Tahun	45	0
	Total	169	38

Tabel 2.7 Jumlah ASN Dosen Fakultas Teknik Berdasarkan Jabatan Akademik dan Masa Kerja

No.	Jabatan Akademik dan Masa Kerja	Jumlah Dosen	Persentase
1	Lektor Kepala \geq 20 Tahun	49	28,99%
2	Lektor \geq 20 Tahun	36	21,30%
3	Asisten Ahli > 10 Tahun	15	8,88%
4	Tenaga Pengajar > 2 Tahun	26	15,38%

Tabel 2.8 Jumlah ASN Dosen Fakultas Teknik Berdasarkan Golongan Ruang dan Masa Kerja

No.	Golongan Ruang dan Masa Kerja	Jumlah Dosen	Persentase
1	IIIa >= 10 Tahun	1	0,59%
2	IIIb >= 15 Tahun	6	3,55%
3	IIIc >= 20 Tahun	14	8,28%
4	IIId >= 25 Tahun	7	4,14%

Tabel 2.9 Jumlah ASN Dosen Fakultas Teknik Dalam Status Tugas Belajar

No.	Nama ASN Dosen	Gol/Ruang	Jenjang TB	Institusi Tempat Studi & Tanggal Berangkat
1	Lusiana, ST, MT	III/c	S3	ITB – Teknik Sipil 08/01/2014
2	Hafzoh Batubara, ST, M.Sc	III/d	S3	UGM – Teknik Industri 09/01/2015
3	Herwani, ST, MT	III/d	S3	ITB – Teknik Sipil 09/01/2014
4	Mira Sophia Lubis, ST, MT	III/c	S3	UI – Arsitektur 09/01/2015
5	Ir. Siti Mayuni, MT	IV/a	S3	UNDIP – Teknik Sipil 01/09/2017
6	Eti Sulandari, ST, MT	IV/a	S3	ITB – Teknik Sipil 09/01/2018
7	Ivan Gunawan, ST, M.Sc	III/b	S3	ITS – Arsitektur 01/09/2019
8	Eddy Kurniawan, S.T, M.Sc	III/b	S3	Pusan National University, South of Korea September 2021
9	Sri Rezeki, S.T, M.Sc	III/b	S3	University of Jena Germany Oktober 2021

Berdasarkan data ASN Fakultas Teknik di atas, dari 169 Dosen aktif, persentase ASN Dosen menempati jabatan akademik Guru Besar adalah 2,37%, Lektor Kepala 30,18%, Lektor 31,95%, Asisten Ahli 16,57% dan belum menempati jabatan akademik

atau masih berstatus Tenaga Pengajar 18,93%. Sedangkan dosen dengan pendidikan S3 adalah 39 orang atau 23,08% dan dosen dengan pendidikan S2 adalah 130 orang atau 76,92%.

Jika data ASN Fakultas Teknik dilihat dari masa kerja, maka pada ASN Dosen terdapat 15 orang atau sebesar 8,88% merupakan dosen dengan masa kerja di atas 30 tahun, 76 atau sebesar 44,97% merupakan dosen dengan masa kerja 20-30 tahun, 33 orang atau sebesar 19,53% merupakan dosen dengan masa kerja 10-20 tahun, dan 45 orang atau sebesar 26,63% merupakan dosen dengan masa kerja dibawah 10 tahun. Sedangkan untuk ASn Tendik, terdapat 14 orang atau sebesar 36,84% merupakan dosen dengan masa kerja di atas 30 tahun, 16 atau sebesar 42,11% merupakan dosen dengan masa kerja 20-30 tahun, dan 8 orang atau sebesar 21,05% merupakan dosen dengan masa kerja 10-20 tahun.

Sedangkan jika dihubungkan jabatan akademik dosen dan masa kerja dosen, maka didapat data sebagai berikut:

- a. Dosen dengan jabatan akademik Lektor Kepala dengan masa kerja 20 tahun ke atas adalah 49 orang atau 28,99% dari total seluruh dosen (yaitu 169 orang).
- b. Dosen dengan jabatan akademik Lektor dengan masa kerja 20 tahun ke atas adalah 36 orang atau 21,30%.
- c. Dosen dengan jabatan akademik Asisten Ahli dengan masa kerja 10 tahun ke atas adalah 15 orang atau 8,88%.
- d. Dosen yang belum menempati jabatan akademik atau masih berstatus Tenaga Pengajar dengan masa kerja 2 tahun ke atas adalah 26 orang atau 15,38.

Berdasarkan data-data tersebut, terlihat bahwa masih banyak ASN Dosen Fakultas Teknik Unviersitas Tanjungpura yang sudah menempuh masa kerja di atas 20 tahun, namun masih menempati jabatan akademik Lektor. Sedangkan terdapat pula ASN dosen yang sudah menempuh masa kerja di atas 10 tahun, namun masih menempati jabatan akademik Asisten Ahli. Begitu pula dengan ASN Dosen dengan status Tenaga Pengajar dengan masa kerja sudah di atas 2 tahun, namun belum menempati jabatan akademik Asisten Ahli.

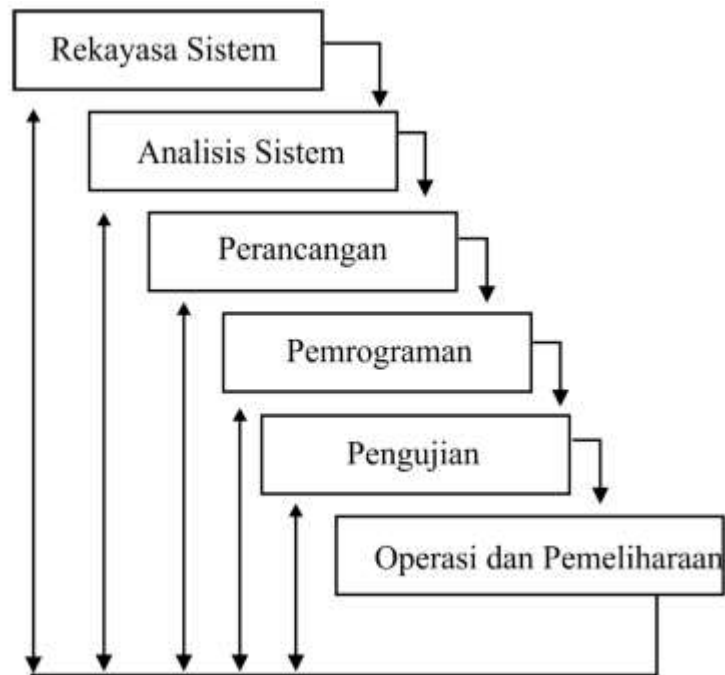
Begitu pula dengan data ASN Dosen Fakultas Teknik berdasarkan golongan ruang dan masa kerja, masih cukup banyak terdapat ASN Dosen dengan masa kerja di atas 20 tahun, namun golongan ruang masih di IIIc dan III d. Selain itu terdapat 1 orang ASN Dosen dengan golongan ruang IIIa dengan masa kerja 32 tahun dan 1 orang dosen dengan golongan ruang IIIb dengan masa kerja 22 tahun.

Dengan demikian, terlihat jelas bahwa proses kenaikan pangkat ASN maupun jabatan akademik Dosen sangat lambat. Banyak faktor penyebab, diantaranya, ketidaktahuan informasi terkait proses kenaikan jabatan akademik dan pangkat/golongan ruang, ketidak inginan dari ASN bersangkutan untuk memproses atau mengurus berkas kenaikan pangkat dan jabatan, hingga tidak adanya monitoring dan pengawasan terhadap jenjang pangkat ASN dan jabatan akademik Dosen dari pimpinan baik di tingkat Fakultas maupun Universitas.

Demikian pula dengan seringnya keterlambatan dalam pengajuan penghargaan Karya Satya bagi ASN yang sudah menempuh masa bakti (masa kerja) selama 10, 20, dan 30 tahun. Hal ini disebabkan tidak adanya basis data kepegawaian yang dapat dikelola dengan baik, sehingga informasi seperti ini tidak dapat dimonitor oleh pimpinan.

2.3 Metode Waterfall

Metode yang digunakan dalam pembuatan Sistem Database adalah metode terstruktur yang didasari pada pengembangan Model *Waterfall*. Metode *Waterfall* merupakan metode pengembangan perangkat lunak tertua sebab sifatnya yang natural. Metode *Waterfall* merupakan pendekatan SDLC paling awal yang digunakan untuk pengembangan perangkat lunak. Urutan dalam Metode *Waterfall* bersifat serial yang dimulai dari proses perencanaan, analisa, desain, dan implementasi pada sistem. Metode ini dilakukan dengan pendekatan yang sistematis, mulai dari tahap kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Langkah demi langkah yang dilalui harus diselesaikan satu per satu (tidak dapat meloncat ke tahap berikutnya) dan berjalan secara berurutan, oleh karena itu di sebut *waterfall* (Air Terjun) (Hidayat, 2021).



Gambar 2.1. Tahapan Metode *Waterfall*

1. Rekayasa Sistem Informasi: Perangkat lunak menjadi penting ketika harus berkomunikasi dengan *hardware*, manusia dan basis data. Rekayasa dan permodelan sistem menekan pada pengumpulan kebutuhan pada level sistem dengan sedikit perancangan dan analisis.
2. Analisis Sistem: disebut juga Analisis Kebutuhan Perangkat untuk memutuskan fitur mana yang ada di dalam Aplikasi Database Aparatur Sipil Negara (ASN) Fakultas Teknik Universitas Tanjungpura, peneliti membuat analisis yaitu dengan mengumpulkan dan meneliti semua aplikasi berbasis web yang sudah dinilai dari berbagai aspek, baik aspek fitur, kekurangan, kelebihan, sumber data.
3. Perancangan: Pada tahap ini gambaran Aplikasi Database Aparatur Sipil Negara (ASN) Fakultas Teknik Universitas Tanjungpura akan dibuat dengan alat perancangan terdiri dari *Data Flow Diagram* (DFD) sebagai alat perancangan system, *Entity Relationship Diagram* (ERD) sebagai alat perancangan basis data, *Flowchart*, dan Kamus Data (*Data Dictionary*).

4. Pemrograman: Hasil perancangan sebelumnya harus diubah menjadi bentuk yang dimengerti mesin (komputer), sehingga dilakukan langkah penulisan program atau pemrograman (*coding*). Peneliti menggunakan Bahasa pemrograman web PHP, *JavaScript*, CSS, dan HTML, dengan menggunakan beberapa *software* pendukung seperti program aplikasi *text editor* yaitu *Sublime Text*.
5. Pengujian: Setelah penulisan program menggunakan bahasa pemrograman dan *software* pendukung dalam pembuatan Aplikasi Database Aparatur Sipil Negara (ASN) Fakultas Teknik Universitas Tanjungpura selesai, program dapat berjalan, maka tahapan pengujian dilakukan dengan model pengujian *black box testing* dan pengujian *User Acceptance Test (UAT)*.
6. Operasi dan Pemeliharaan: Operasional dan pemeliharaan aplikasi database akan diserahkan kepada pengguna terakhir (*end user*) yaitu Sub Bagian Keuangan dan Kepegawaian Fakultas Teknik Universitas Tanjungpura yang diawasi oleh Wakil Dekan II dan Kasubbag Keuangan dan Kepegawaian.

2.4 Alat Bantu Perancangan Sistem

2.4.1 Data Flow Diagram (DFD)

DFD merupakan suatu cara atau metode untuk membuat rancangan sebuah sistem yang mana berorientasi pada alur data yang bergerak pada sebuah sistem nantinya. *Data Flow Diagram (DFD)* adalah sebuah alat yang menggambarkan aliran data sampai sebuah sistem selesai, dan kerja atau proses dilakukan dalam sistem tersebut (Indrajani, 2011). Ada beberapa simbol yang digunakan pada DFD : (Jogiyanto, 2001).

1. Kesatuan Luar (*External Entity*)

Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lain yang berada pada lingkungan luarnya yang memberikan input atau menerima output dari sistem.

2. Arus Data (*Data Flow*)

Arus Data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir di antara proses, simpan data dan kesatuan luar. Arus data ini menunjukkan arus dari

data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.


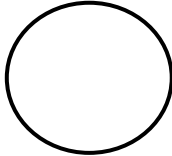
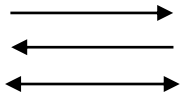

3. Proses (*Process*)

Proses (*process*) menunjukkan pada bagian yang mengubah input menjadi output, yaitu menunjukkan bagaimana satu atau lebih input diubah menjadi beberapa output. Setiap proses mempunyai nama, nama dari proses ini menunjukkan apa yang dikerjakan proses.

4. Simpanan Data (*Data Store*)

Data Store merupakan simpanan dari data yang dapat berupa suatu dokumen atau basis data pada sistem komputer.

Tabel 2.10 Simbol DFD

Simbol DFD	Nama
	Entitas / Kesatuan luar
	Proses
	Arus data / Aliran data
	<i>Data store</i> / Simpanan data


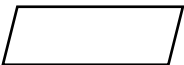

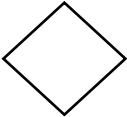

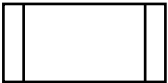
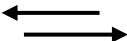
2.4.2 Flowchart

Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan

urutan prosedur suatu program. Biasanya mempermudah penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut (Indrajani, 2011).

Tabel 2.11 Simbol *flowchart*

Simbol <i>Flowchart</i>	Keterangan
Terminal 	Memulai dan Mengakhiri suatu proses
<i>Input / Output</i> 	Digunakan untuk proses data maupun input data
Proses 	Menampilkan suatu proses
Keputusan 	Untuk menyeleksi kondisi didalam program. (percabangan)
Persiapan 	Digunakan untuk pemberian nilai awal suatu besaran
Proses pendefinisian 	Menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain
Garis alir 	Garis suatu proses atau menunjukkan arus proses yang berjalan

2.4.3 Entity Relationship Diagram (ERD)


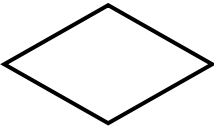
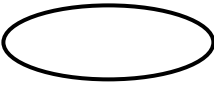

Entity Relationship Diagram (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan

hubungan antar data dengan menggunakan kardinalitas relasi. Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain (Ladjamudin, 2005).

Terdapat 3 macam kardinalitas relasi yaitu sebagai berikut:

1. *One to One* (1:1): Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.
2. *One to many* (1:M / *Many*): Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.
3. *Many to Many* (M:M): Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

Tabel 2.12 Simbol ERD

Simbol ERD	Keterangan
	Entitas : Memberikan identitas yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain.
	Relasi : Menunjukkan hubungan diantara sejumlah entitas yang berbeda.
	Atribut : Properti yang dimiliki oleh suatu entitas, dimana dapat mendeskripsikan karakteristik dari entitas tersebut.
	Alur : Garis yang memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi.

2.5 Kamus Data (*Data Dictionary*)

Kamus data merupakan tempat penyimpanan dari aliran-aliran data, *file* dan proses-proses dalam sebuah sistem. Kamus data digunakan untuk menjelaskan semua

data yang mengalir atau digunakan dalam sistem, yaitu mengenai arus data yang masuk ke dalam sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

Kamus Data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. *Data dictionary* tidak menggunakan notasi grafik sebagaimana halnya *data flow diagram*. Kamus data berfungsi untuk membantu user agar dapat mengerti aplikasi secara rinci (Sutabri, 2004).

2.6 Alat Pemrograman

2.6.1 Hypertext Markup Language (HTML)

HTML atau *Hypertext Markup Language* merupakan bahasa yang paling umum digunakan dalam pembangunan web. Sesuai dengan namanya, markup language yang berarti bahasa penandaan, HTML digunakan untuk menandai secara khusus bagian-bagian yang terdapat dalam sebuah halaman web. Penandaan pada setiap bagian menggunakan tag (Duckett, 2010).

Pada penggunaan tag HTML tanda kurung siku “< >” untuk memulai sebuah *code* serta adanya kata atau huruf di dalamnya. Dalam sebuah struktur file HTML pada setiap bagian menandakan yang mana *header*, *title*, *body*, *paragraph*, *tabel*, dan lainnya. Untuk memulai menulis sebuah kode HTML, bisa menggunakan program aplikasi *text editor* seperti *Notepad*, *Sublime Text*, *Adobe Dreamweaver*, dan sebagainya.

2.6.2 Cascading Style Sheet (CSS)

Kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih menarik dan terlihat elegan. *Cascading Style Sheet* atau lebih dikenal dengan CSS dalam pemrograman web berfungsi untuk mengatur *style* dari halaman *web* (Duckett, 2010).

Dalam penulisan kode CSS memerlukan sebuah *rule* yang dapat diatur sesuai keinginan *programmer*. *Rule* dibagi menjadi dua bagian :

1. *Selector*:

Menunjuk atau memilih elemen mana yang akan diberikan *rule* CSS.

2. *Declaration:*

Menetapkan bagaimana sebuah elemen yang dipilih akan ditata. *Declaration* dibagi menjadi dua bagian yang dipisahkan dengan tanda titik dua “;” yaitu :

a. *Property*

Digunakan untuk memilih properti apa yang akan digunakan untuk memberikan efek kepada elemen yang telah dipilih.

b. *Value*

Memberikan nilai secara spesifik bagi *property* yang telah dipilih. Oleh karena itu jika file HTML tidak memiliki *rule* CSS, maka tampilan yang diberikan HTML akan tampak tidak rapi atau kurang beraturan. Jika diterapkan *rule* CSS pada *elemen* atau *tag* pada *file* HTML maka tampilan yang diberikan menjadi lebih baik.

2.6.3 JavaScript

JavaScript merupakan bahasa pemrograman yang digunakan dalam pemrograman web (Duckett, 2010).

Penulisan *code* JavaScript ditandai dengan *tag* ‘<script type=”JavaScript”></script>’ pada halaman *HTML*. *Code JavaScript* dapat dimasukkan secara internal ataupun eksternal.

Penulisan secara internal cukup dengan memasukan *code* di antara *tag script*. Sedangkan secara eksternal, *code* berupa *file* yang disimpan berlainan folder ataupun dari *link* internet dituliskan seperti berikut: <script type = ”JavaScript” src =”scripts/jquery.js”> </script>.

Hal yang didapat jika menggunakan *JavaScript*:

1. Membaca dan menuliskan (*read & write*) elemen dan teks.
2. Memanipulasi atau memindahkan teks.
3. Menjalankan fungsi perhitungan pada data.
4. Bereaksi terhadap *events*, seperti saat pengguna menekan sebuah *button*.
5. Membaca waktu dan tanggal di komputer pengguna.

6. Menetapkan ukuran tampilan beserta resolusi layar dan *web browser version* pada komputer pengguna.
7. Melakukan pengecekan pada *input* yang diberikan oleh pengguna, seperti pada pengecekan *form validation*.

2.6.4 PHP

PHP (*Hypertext Processor*) adalah bahasa pemrograman yang memungkinkan para *web developer* untuk membuat aplikasi web yang dinamis dengan cepat dan mudah. PHP dirintis dan diperkenalkan pertama kali sekitar tahun 1994 oleh Rasmus Lerdorf melalui situsnya untuk mengetahui siapa saja yang telah mengakses ringkasan *online*-nya (Gunawan, 2010).

PHP disebut bahasa pemrograman *server-side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser (client)*.

PHP dirancang untuk membentuk web dinamis, gunanya membentuk suatu tampilan berdasarkan permintaan terkini. Untuk sebuah web jika menggunakan bahasa pemrograman PHP maka akan menjadi lebih dinamis dan rapi. kode PHP biasanya di sisipkan kedalam *file HTML*, karena itu PHP disebut juga sebagai *Scripting Language* atau bahasa pemrograman *script*.

PHP memiliki kelebihan dibanding dengan bahasa *scripting* lainnya. Berikut ini kelebihannya (Welling & Thomson, 2009) :

1. *Performance*: PHP dapat bekerja cepat dengan performa yang sangat baik.
2. *Scalability*: Skala komoditas yang besar.
3. Integrasi ke *database*: PHP memiliki *native connection* yang tersedia untuk berbagai macam *database system*.
4. *Built-in libraries*: PHP memiliki banyak fungsi *built-in* untuk melakukan tugas yang berhubungan dengan *web task*.
5. *Low cost*: PHP merupakan bahasa *scripting open source*, sehingga siapapun dapat menggunakannya secara gratis.

6. *Ease of learning and use*: PHP mudah untuk dipelajari. Bahasa pemrograman PHP didasari pada bahasa pemrograman pada umumnya, seperti C, C++, Perl, dan Java.
7. *Strong object-oriented (OOP) support*: PHP versi 5 sudah didukung dengan fitur OOP.
8. *Flexibility of development approach*: PHP memberikan kemudahan bagi *developer* dalam implementasi. PHP sangat mudah untuk beradaptasi dengan *framework* seperti CodeIgniter.
9. *Availability of source code*: *Developer* memiliki akses langsung dengan *source code* PHP.
10. *Availability of support and documentation*: PHP memberikan dokumentasi yang jelas dan tersedia komunitas yang siap membantu jika *developer* mengalami kesulitan dalam menggunakan PHP.

2.6.5 XAMPP

XAMPP adalah paket program web lengkap yang dapat dipakai untuk belajar pemrograman web, khususnya PHP dan MySQL (Nugroho, 2013).

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi. XAMPP merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*) yang terdiri dari program Apache HTTP *Server*, MySQL *database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas. XAMPP merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Bagian penting XAMPP yang biasa digunakan pada umumnya:

1. XAMPP *Control Panel Application* berfungsi mengelola layanan (*services*) XAMPP, seperti mengaktifkan layanan (*start*) dan menghentikan layanan (*stop*).
2. Htdocs adalah *folder* tempat meletakkan berkas-berkas yang akan dijalankan.
3. PHPMyOperator biasa merupakan bagian untuk mengelola *database*.

2.6.6 MySQL

MySQL adalah sebuah *relational database server* yang sangat cepat dan dapat menyesuaikan dengan kebutuhan pengguna. MySQL merupakan sebuah basis data yang dapat menyimpan, mencari, melakukan pengurutan, dan mengambil data (Welling & Thomson, 2009).

Server MySQL mengontrol akses data sehingga dapat diakses oleh banyak pengguna secara bersamaan, menyediakan akses yang cepat, dan memastikan hanya pengguna yang terotorisasi yang dapat mengakses basis data. MySQL menggunakan SQL (*Structured Query Language*).

MySQL dan PHP merupakan sistem yang saling terintegrasi. Maksudnya adalah pembuatan database dengan menggunakan sintak PHP dapat di buat. Sedangkan input yang di masukkan melalui aplikasi web yang menggunakan *script serverside* seperti PHP dapat langsung dimasukkan ke database MySQL yang ada di server dan tentunya web tersebut berada di sebuah *web server*. Adapun perangkat pendukung yang terhubung dengan MySQL terdiri dari dua perangkat yaitu:

1. *PhpMyOperator biasa*

PhpMyOperator biasa adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan *phpMyOperator biasa*, user dapat membuat database, membuat tabel, memasukkan, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah tanpa perlu mengetikkan perintah SQL secara manual.

Beberapa fitur dalam *phpMyOperator biasa* :

- 1) Antarmuka berbasis web
- 2) Impor data dari CSV dan SQL
- 3) Ekspor data dari berbagai format
- 4) Membuat grafik PDF dari tampilan dan basis data user
- 5) Membuat kompleks query menggunakan *Query-by-example* (QBE)
- 6) Transformasi data disimpan ke dalam format yang menggunakan satu set yang telah ditetapkan seperti menampilkan data blob-data atau download-link.

2. Basis Data

Basis Data adalah kumpulan data (*elementer*) yang secara logika berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. Basis data adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi (Hariyanto, 2004).

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur, dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organisasi data, dan juga *update* yang rumit.

Proses memasukkan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*Database Management System / DBMS*). Tujuan utama dari DBMS adalah untuk memberikan tinjauan abstrak data kepada user (pengguna). Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara, dan tetap dapat diambil (akses) secara efisien.

2.6.7 Sublime Text

Sublime Text merupakan suatu *text editor* yang sering digunakan oleh programmer khususnya seorang *web developer*. Aplikasi ini menjadi sesuatu yang sangat penting bagi *web developer* sebagai senjata koding. Banyak *web developer* yang menggunakan aplikasi *Sublime Text* untuk mendukung *website* Mereka. Warna dari tulisan koding yang sangat variasi dan interaktif tentu sangat menarik. Anda juga dapat mengubah warnanya temanya. Selain itu, warna koding yang bervariasi dan sangat menarik mampu membuat Anda lebih betah dalam menjalankan proses koding. Selain

itu, ada banyak fitur yang tersedia di *Sublime Text* yang memiliki banyak kegunaan (IDCloudHost, 2020).

2.7 Pengujian Perangkat Lunak

2.7.1 Black Box Testing

Black box Testing atau pengujian *black box* merupakan metode uji coba yang memfokuskan pada keperluan fungsional dari *software*. Karna itu uji coba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program.

Black box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program (Pressman, 2010).

Black box testing berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

Black box testing berusaha untuk menemukan kesalahan dalam kategori berikut :

1. Fungsi yang tidak benar atau fungsi yang hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan perilaku (*behavior*) atau kesalahan kinerja
5. Inisialisasi dan pemutusan kesalahan

Kategori *error* yang akan diketahui melalui *black box testing* :

1. Fungsi yang hilang atau tak benar
2. *Error* dari antar-muka
3. *Error* dari struktur data atau akses eksternal *database*
4. *Error* dari kinerja atau tingkah laku
5. *Error* dari inisialisasi dan terminasi

2.7.2 User Acceptance Testing (UAT)

Pengujian UAT atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa *software* yang telah dikembangkan oleh telah dapat diterima pengguna, apabila hasil pengujian (*testing*) sudah bisa dianggap memenuhi kebutuhan dari pengguna.

User Acceptance Testing merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah *staff* / karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya (Perry, 2006).

Maka dari definisi tersebut, dapat dikatakan bahwa UAT merupakan pengujian yang dilakukan oleh pengguna dari sistem tersebut untuk memastikan fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna. Proses dalam UAT adalah pemeriksaan dan pengujian terhadap hasil pekerjaan. Diperiksa apakah item-item yang ada dalam dokumen requirement sudah ada dalam software yang diuji atau tidak. Diuji apakah semua item yang telah ada telah dapat memenuhi kebutuhan penggunanya.