**BAB 2**

**LANDASAN TEORI**

**2.1 Rancang Bangun**

Rancang Bangun adalah tahap awal dari membuat gambaran dan bentuk sketsa yang belum pernah dibuat sama sekali lalu dikelolah menjadi gambaran atau sketsa yang memiliki fungsi yang diinginkan (Nurmalina, 2017).

Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi (Nurhayati and Josi, 2017)

Rancang bangun adalah menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tersebut. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada (Maulani, 2018).

Dari pengertian diatas Penulis menyimpulkan bahwa Rancang Bangun adalah tahap awal dari membuat gambaran dan bentuk sketsa yang belum pernah dibuat sama sekali lalu dikelolah menjadi gambaran atau sketsa yang memiliki fungsi yang diinginkan.

**2.4 Pengertian Sistem Informasi**

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Ahmad and Hasti, 2018).

Sistem informasi merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi yang akan mendukung pembuatan keputusan dan melakukan kontrol terhadap jalannya perusahaan (Lutfi, 2017).

Sistem informasi adalah kegiatan dari prosedur yang di organisasikan yang digunakan untuk menyediakan informasi pengambilan keputusan dan pengendalian didalam organisasi (Fridayanthie and Charter, 2016).

Dari pengertian diatas Penulis menyimpulkan bahwa Sistem informasi merupakan suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang tergorganisasi.

**2.3 Pengertian Pelayanan**

Pelayanan adalah kegiatan yang dilakukan oleh seseorang atau sekelompok orang dengan landasan faktor materi melalui sistem, prosedur dan metode tertentu dalam rangka usaha memenuhi kepentingan orang lain sesuai dengan haknya (Moenir, 2018).

Pelayanan adalah pemberian jasa baik oleh pemerintah, pihak swasta atas nama pemerintah ataupun pihak swasta kepada masyarakat, dengan tanpa pembayaran guna memenuhi kebutuhan dan kepentingan masyarakat (Wasistiono, 2016).

Pelayanan adalah setiap tindakan atau kegiatan yang dapat ditawarkan oleh suatu pihak kepada pihak lain yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun (Kotler, 2017).

Dari pengertian diatas Penulis menyimpulkan bahwa Pelayanan ialah menolong untuk menyediakan segala sesuatu yang diperlukan oleh orang lain. Pelayanan dapat diberikan kepada orang lain sebagai pertolongan yang dibutuhkan orang lain itu sendiri. Yang mana dengan pertolongan tersebut dapat membantu orang lain untuk bisa mengatasi masalahnya.

**2.4 Pengertian Jasa**

Jasa adalah suatu proses atau aktivitas, dan aktivitas-aktivitas tersebut tidak berwujud. Jadi pada dasarnya jasa merupakan semua aktivitas ekonomi yang hasilnya tidak merupakan produk dalam bentuk fisik atau kontruksi, yang biasanya dikonsumsi pada saat yang sama dengan waktu yang dihasilkan dan memberikan nilai tambah atau pemecahan atas masalah yang dihadapi konsumen (Asih, 2016).

Jasa merupakan kegiatan ekonomi, melibatkan beberapa interaksi bersama konsumen atau barang-barang milik, tapi tidak menghasilkan pergantian kepemilikan(Septiyanti and Nadjamuddin, 2018).

Dengan demikian dapat disimpulkan bahwa jasa merupakan suatu tindakan atau aktivitas yang ditawarkan pada pihak lain dan tidak berwujud tetapi bisa dinikmati manfaatnya.

**2.5 Pengertian Laundry**

Laundry adalah kata benda yang mengacu pada tindakan mencuci pakaian, tempat dimana mencuci dilakukan. Laundry juga diartikan sebagai kegiatan mencuci pakaian atau bahan tekstil lainnya dan juga sebagai sebuah tempat untuk mencuci pakaian atau bahan tekstil lainnya (Simargolang and Nasution, 2018).

Usaha laundry adalah suatu jenis usaha yang bergerak dibidang jasa cuci dan setrika. Bisnis ini termasuk dalam kategori bisnis dengan perputaran yang cepat. Maksudnya rentang waktu permintaan pelanggan antara permintaan pertama dan permintaan selanjutnya pada jasa ini yang memakan waktu relatif singkat. Lebih jelasnya, pelanggan akan kembali menggunakan jasa ini ketika pakaian yang dikenakan sudah kotor (Purnarini, 2019).

Laundry adalah suatu kegiatan mencuci, mengeringkan, menyetrika dan merawat objek cucian antara lain pakaian seperti kemeja, celana, jaket dan sebagainya, adapun linen meliputi sprei, handuk, taplak meja dan sebagainya (Januar, 2020).

Dari pengertian diatas Penulis menyimpulkan bahwa Laundry adalah salah satu jenis usaha yang mengacu pada tindakan mencuci, menyetrika yang memberikan layanan jasa kepada mereka yang membutuhkan.

**2.6 Pengertian Progressive Web App**

Pada dasarnya, *Progressive Web App (PWA)* adalah sebuah website yang dibangun menggunakan teknologi web modern, namun dapat berlaku seperti sebuah Mobile App. Pada tahun 2015, *Google Engineer* Alex Russel dan Frances Berriman memberikan istilah *PWA* pada konsep *web apps* yang dapat memberikan pengalaman pengguna dalam keandalan *(reliability*), kecepatan (*speed*), dan keterlibatan pengguna (*user engagement*) (Santoso, 2019).

Google berusaha memastikan bahwa aplikasi web progresif ini dapat memberikan pengalaman ke pengguna seperti aplikasi *native*. PWA memiliki beberapa kelebihan, yaitu memiliki respon seperti *mobile app*, *responsive layout,* tersedia di mode *offline*, memiliki ikon yang dapat ditambahkan ke layar *home screen* perangkat, dapat mengirim dan menerima notifikasi (Faisol and Noertjahjono, 2019).

Teknologi yang umum digunakan pada *PWA* meliputi *Web App Manifest*, *Service Worker*, *App Shell* yang semuanya memerlukan web browser modern terbaru. Berikut fitur-fitur pendukung *PWA* :

1. Web App Manifest

*Web App Manifest* merupakan *W3C specification* yang mendefinisikan *JSON-based* manifest untuk memberikan developer tempat untuk meletakkan metadata dari aplikasi web, terdiri dari :

1. Nama aplikasi web
2. Link ke icon atau obyek gambar web app
3. Target URL untuk membuka (launch) web app
4. Data konfigurasi web app seperti display, orientasi, dan lainnya

Dengan setting file *web manifest*, developer memungkinkan pengguna untuk menciptakan pengalaman seperti aplikasi mobile native melalui aplikasi web *progresif*. Pembahasan detail tentang web app manifest akan dibahas tersendiri.

2. Service Worker

Aplikasi mobile *native* memberikan pengalaman yang kaya dan kinerja tinggi, namun membutuhkan ruang penyimpanan (ROM), kurangnya real-time update, dan visibilitas mesin pencari yang rendah. Sementara aplikasi web tradisional tidak membutuhkan native compiled executable (seperti .exe di Windows, .apk di Android atau .ips di IOS), namun sangat tergantung pada koneksi jaringan yang tidak dapat diandalkan dan berpotensi lambat. Disinilah *service worker* berperan untuk memberikan aplikasi web progresif yang terbaik. *Service worker* adalah script yang berjalan di belakang browser pengguna. Service worker tidak membutuhkan sebuah halaman ataupun interaksi dari pengguna untuk menjalankan tugasnya, dengan begitu *service worker* akan terus berjalan walaupun halaman webnya tidak terbuka.

3. Arsitektur Application Shell

PWA menggunakan pendekatan arsitektur yang dinamakan *App Shell Model*. Supaya loading web cepat, service worker menyimpan antarmuka pengguna dasar (*basic UI*) atau “shell” dari aplikasi web desain responsif. Shell ini menyediakan bingkai statis di awal, tata letak (*layout*) atau arsitektur di mana konten dimuat secara progresif serta dinamis. Ini memungkinkan user untuk terlibat dengan aplikasi dalam kondisi jaringan web bagaimanapun.

Secara teknis, shell adalah bundel kode yang disimpan secara lokal di *cache browser* perangkat *mobile*. Pembahasan lebih lanjut tentang application shell ini akan dijelaskan pada bab selanjutnya.

4. Web Browser Modern

Web browser popular saat ini, seperti *Google Chrome*, *Mozilla Firefox*, *Opera*, *Samsung Internet,* *Apple Safari* dan *Microsoft Edge* sudah mulai mendukung teknologi web progresif. Di website Jake Archibald11 https://jakearchibald.github. io/isserviceworkerready/ ini dapat melihat versi web browser apa saja yang dapat menjalankan PWA dengan sempurna.

Kelebihan yang dimiliki oleh PWA adalah sebagai berikut :

1. Respon seperti Mobile App, Pada *PWA s*emuanya di-cache didalam “App Shells” sehingga tampak seperti aplikasi Mobile App.

2. Layout Responsif, PWA menggunakan responsive layout

3. Tersedia di Mode Offline, Pada saat membuka PWA di browser, cache situs web akan disimpan secara otomatis di web browser.

4. Add To Home Screen (A2HS), dapat menambahkan PWA ke layar Home untuk akses cepat.

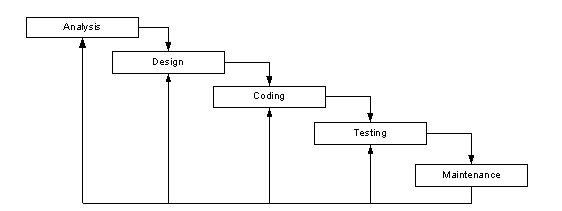
5. Push Notification, PWA memliki kemampuan untuk mendapatkan pesan dan peringatan cepat dari situs web.

**2.6 Metode *Waterfall***

Adapun metode pengembangan sistem yang digunakan dalam penelitian adalah model *waterfall.* Model SDLC air terjun (*waterfall*) sering disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*).

Metode *Waterfall* merupakan contoh dari tahapan yang plan-driven, yaitu secara prinsip pengembang harus merencanakan dan menjadwalkan semua kegiatan tahapan sebelum mulai mengembangkan perangkat lunak. Komitmen dibuat di awal tahapan sehingga membuat sulit untuk merespon ketika ada perubahan pada sebuah customer. Model air terjuan menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau berurutan dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*) (Sasmito, 2017).

Berikut adalah gambar model air terjun :



(Sumber : Fadhilah and Budiyanto, 2020)

**Gambar 2.1** *Metode Waterfall*

Gambar 2.1 *Metode Waterfall* diatas dalah gambaran dari tahapan-tahapan metode pengembangan sistem informasi *waterfall* model dalam proses penelitian dan perancangan suatu sistem informasi. tahapan tersebut memiliki struktur terurut terdiri dari analisis sistem, desain, implementasi sistem, testing, dan maintenence yang memiliki peran masing-masing dalam suatu proses penelitian dan perancangan suatu sistem.

Deskripsi dari tahapan-tahapan pada Metode *Waterfall* sebagai berikut :

1. Analisis (*Analysis*)

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. Desain (Design)

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. Implementasi (*Implementation*)

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. Pengujian sistem (*System Testing*)

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirim ke customer.

5. Pemeliharaan *(Maintenance*)

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan layanan sistem sebagai kebutuhan baru.

**2.20 Pengertian Perancangan Sistem**

Tahapan perancangan sistem adalah tahapan untuk memberikan gambaran mengenai sistem informasi baru yang akan dibuat, dan telah dianalisis ke dalam bentuk yang mudah dimengerti oleh calon pemakai (IpanRipai, 2017)

**2.21 Pengertian Prosedur**

Prosedur adalah suatu bagian sistem yang merupakan rangkaian, tindakan yang menyangkut beberapa orang dalam satu atau beberapa bagian yang ditetapkan untuk menjamin agar suatu kegiatan usaha atau transaksi dapat terjadi berulangkali dan dilaksanakan secara beragam (Hasanudin, 2018).

**2.22 Pengertian Flowchart**

*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dengan hubungan antara suatu proses (*instruksi*) dengan proses lainnya dalam suatu program (Wibawanto and W, 2017). Adapun simbol-simbol *Flowchart* adalah sebagai berikut:

**Tabel 2.5** *Simbol-Simbol FlowChart*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | | Simbol | | Keterangan | |
| 1. | |  | | Simbol Start atau End yang mendefinisikan awal atau akhir dari sebuah flowchart. | |
| 2. | |  | | Simbol pemrosesan yang terjadi pada sebuah alur kerja. | |
| 3. | |  | | Simbol Input / Output yang mendefinisikan masukkan dan keluaran proses. | |
| 4. | |  | | Simbol untuk memutuskan proses lanjutan dari kondisi tertentu. | |
| 5. | |  | | Simbol konektor untuk menyambung proses pada lembar kerja yang sama. | |
| 6. | |  | | Simbol konektor yang menyambung proses pada lembar kerja yang berbeda. | |
| 7. | |  | | Simbol untuk menghubungkan antar proses atau antar symbol. | |

***Tabel Lanjutan 2.5*** *Simbol-Simbol FlowChart*

|  |  |  |
| --- | --- | --- |
| 8. |  | Simbol yang mendefinisikan proses yang dilakukan secara manual. |
| 9. |  | Simbol masukkan atau keluaran dari atau ke sebuah dokumen. |
| 10. |  | Simbol yang menyatakan bagian dari program (sub - program) |
| 11. |  | Simbol masukkan atau keluaran dari atau ke sebuah pita magnetik. |
| 12. |  | Simbol database atau basis data. |
| 13. |  | Simbol yang menyatakan piranti keluaran, seperti layar monitor, printer, dll. | |

(Sumber : Bio Rekes, 2018)

**2.23 Pengertian DFD (Data Flow Diagram)**

Diagram aliran data atau data *flow diagram* digunakan untuk menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output* (Hasanudin, 2018).

Diagram *konteks* merupakan suatu model yang menjelaskan secara global mengenai bagaimana data digunakan dan dipindahkan untuk proses yang berdasarkan dari database. Diagram *konteks* terdiri dari proses dan menggambarkan ruang lingkup suatu sistem. Diagram *konteks* merupakan tertinggi dari *DFD* yang menggambarkan seluruh *input* ke *system* atau output dari

sistem. Jadi Data flow diagram (*DFD*) merupakan diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari sistem. *DFD* menggambarkan penyimpanan data dan proses yang mentransformasikan data. *DFD* menunjukkan hubungan antara satu dan proses pada sistem (Sismadi and Santoso, 2018). Tingkatan-tingkatan pada *DFD* :

1. Diagram Konteks

Diagram *Konteks* adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram *konteks* merupakan level tertinggi dari *DFD* yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram *konteks* hanya ada satu proses. Tidak boleh ada *store* dalam diagram *konteks.*

2. Diagram Nol/*Zero* (*Overview Diagram*) Diagram

Diagram nol adalah diagram yang menggambarkan proses dari data *flow* diagram. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data, *eksternal entity*.

3. Diagram Rinci (*Level Diagram*)

Diagraam rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram *zero* atau diagram level diatasnya.

Notasi-notasi pada *DFD* adalah sebagai berikut :

**Tabel 2.6** *Simbol-simbol pada DFD*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama** | **Simbol** | **Keterangan** | |
| 1. | Proses (*process)* |  | Proses atau fungsi atau prosedur, pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang seharusnya menjadi fungsi atau prosedur di dalam kode program. | |
| 2. | Penyimpanan data (*Datastore*) |  | File atau basis data atau penyimpanan (*storage*), pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel pada basis data yang dibutuhkan, tabel-tabel pada basis data (*Entity Relationship*) Diagram (*ERD*), Conceptual Data Model (*PDM*), *physical* Data Model (*PDM).*  Catatan:  Nama yang diberikan pada sebuah penyimpanan biasanya kata benda. | |
| 3. | Entitas luar (external entity) |  | Entitas lunak luar (*external entity*) atau masukan (*input*) atau keluaran (*output*) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem yang terkait dengan aliran data dari sistem yang dimodelkan.  Catatan:  Nama yang digunakan pada masukan (*input*) atau keluaran (*output*) biasanya kata benda. |

***Tabel Lanjutan 2.6***  *Simbol-Simbol pada DFD*

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | Aliran data (*Data flow*) |  | Aliran data, merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (*input*) atau keluaran (*output*).  Catatan:  Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata benda, dapat diawali dengan kata data misalnya "data siswa" atau tanpa kata data misalnya "siswa" |

(Sumber Sismadi and Santoso, 2018)

Tabel 2.6 Simbol-simbol pada *DFD* tabel ini menjabarkan nama simbol, bentuk simbol dan arti dari proses pembuatan *DFD* (*Data Flow Diagram*). Setiap simbol yang digunakan pada pembuatan *Data Flow Diagram* memiliki nama dan arti tersendiri dimaksudkan agar peneliti atau perancang suatu sistem dapat mengetahui lebih rinci dari proses sistem informasi yang dipelajari ataupun dirancang.

Penjelasan tentang fungsi dari masing-masing simbol DFD (*Data Flow Diagram*) sebagai berikut :

1. Proses (*Process*)

Proses menggambarkan bagian dari sistem yang mentransformasikan input ke output. Pemberian nama pada komponen proses menggunakan kata kerja yang membutuhkan subyek (transitif).

2. Penyimpanan (*Datastore*)

Data store digunakan untuk membuat model sekumpulan paket data. Data Store biasanya berkaitan dengan penyimpanan seperti file atau database yang berkaitan dengan penyimpanan, baik secara komputerisasi, misalnya file hardisk, maupun manual, misalnya nama dan alamat pada buku alamat. Nama yang diberikan pada data store biasanya menggunakan kata benda jamak.

3. Entitas luar (*external entity*)

Orang, organisasi, atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem. Atau masukan (input) atau keluaran (output) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan.

4. Aliran data (*Data flow*)

Alir Data ini digunakan untuk menerangkan perpindahan data atau satu paket data/informasi dari satu bagian sisitem ke bagian lainnya. Biasanya pemberian nama pada alur data dilakukan dengan menggunakan kata benda.

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD:

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

2. Membuat DFD

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

3. MembuatDFD Level 2

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

4. Membuat DFD Level 3 dan seterusnya

DFD Level 3,4,5, dan seterusnya merupakan breakdwon dan modul pada DFD Level di-atasnya. Brekdwon pada Level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau Level 2.

**2.24 Pengertian ERD (*Entity Relationship Diagram*)**

Entity Relationship Diagram (ERD) adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam system bisnis (Santoso and Nurmalina, 2017).

***Tabel 2.7*** *Simbol-Simbol Entity Relationship Diagram (ERD)*

|  |  |  |
| --- | --- | --- |
| **Nama** | **Simbol** | **Keterangan** |
| Entitas |  | Persegi panjang menyatakan himpunan entitas adalah orang, kejadian, atau berada dimana data akan dikumpulkan. |

***Tabel Lanjutan 2.7*** *Simbol-Simbol Entity Relationship Diagram (ERD)*

|  |  |  |
| --- | --- | --- |
| Atribut |  | Atribut merupakan informasi yang diambil tentang sebuah entitas. |
| Relasi |  | Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas. |
| Link |  | Garis sebagai penghubung antar himpunan, relasi, dan himpunan entitas dengan atributnya |

(Sumber : Nurmalina, 2017)

Tabel 2.7 *Simbol-Simbol Entity Relationship Diagram (ERD)* tabel ini berisi nama, simbol dan arti pada pembuatan *Entity Relationship Diagram (ERD).* Setiap simbol memiliki arti tersendiri yang dapat memberi gambaran tentang entitas, relasi, atribut, asosiasi, dan kardinalitas pada suatu sistem sehingga pengguna maupun perancang mengetahui hubungan dan perbedaan masing-masing objek pada *Entity Relationship Diagram (ERD).*

Beberapa elemen yang ada di dalam ERD adalah sebagai berikut:

1. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

2. *Atribut* secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnnya yang dimaksud entitas maupun *relationship,* sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *relationship*.

3. *Relasi* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relasi*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau pasif).

4. *Link* memiliki fungsi untuk menghubungkan Atribut dengan Entitas dan Entitas dengan Relasi.

Relationship mempunyai tiga tipe. Tiap tipe menunjukkan jumlah record dari setiap tabel yang direlasikan ke record pada tabel lain. Ketiga tipe tersebut adalah sebagai berikut:

1. Hubungan satu ke satu (*One-to-one relationship*)

Hubungan antara file pertama dan file kedua satu berbanding satu. Dalam hubungan ini, tiap record dalam tabel A hanya memiliki satu record yang cocok dalam tabel B dan tiap record dalam tabel B hanya memiliki satu record yang cocok dalam tabel A.

Logika penalaran matematik dari one to one relationship adalah pemetaan dengan “perkawanan satu-satu”. Di ilmu matematika aljabar hal ini lazim disebut sebagai korespondensi satu-satu.

2. Hubungan satu ke banyak (*One-to-many relationship*)

Hubungan antar file pertama dengan file ke dua adalah satu berbanding banyak. Dalam hubungan ini tiap record dalam tabel A memiliki beberapa record yang cocok dalam tabel A. Logika penalaran matematik dari one to many relationship adalah pemetaan dengan “Perkawanan satu- banyak”.

3. Hubungan banyak ke banyak (*Many-to-many Relationship*)

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Dalam hubungan ini tiap record dalam tabel A memilik beberapa record yang cocok dalam tabel B dan tiap record dalam tabel B hanya memiliki satu record yang cocok dalam tabel A. Logika penalaran matematik dari many to many relationship adalah pemetaan “Perkawanan banyak ke banyak”. Contoh hubungan many to many adalah banyak produk mempunyai banyak order dan banyak order mempunyai banyak produk, untuk hal tersebut hubungan ini harus dipecah menjadi hubungan One to many.

**2.7 Pengertian Metode *Unified Modeling Language* (UML)**

Unifield Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam blue print diamna didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem, yaitu :

1. *Use Case* Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat aktor yang merupakan sebuah gambaran entitias dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem. Adapun simbol-simbol *Use Case* adalah sebagai berikut:

***Tabel 2.1*** *Simbol-Simbol Pada Use Case*

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
| Leveling Use Case – Ini WordPress ? | *Actor*, Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
|  | *Use Case*, Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
|  | Generalization, Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek induk (ancestor). |

***Tabel Lanjutan 2.1*** Simbol-Simbol Pada *Use Case*

|  |  |
| --- | --- |
|  | *Include*, menspesifikasikan bahwa use case sumber secara eksplisit. |
|  | *Association*, Apa yang menghubungkan antara objek satu dengan objek lainnya. |
|  | *System*, menspesifikasikan paket yang menampilkan sistem secara terbatas. |
|  | *Use Case*, Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
|  | *Collaboration*, Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
|  | *Note*, Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

(Sumber: M Teguh Prihandoyo, 2018)

Tabel 2.1 *Simbol-Simbol* Pada *Use Case* diatas adalah tabel yang menjabarkan simbol dan arti dari masing-masing simbol yang digunakan pada *Use Case.*

2. *Activity Diagram* Merupakan gambarn alir dari aktivitas-aktivitas didalam sistem yang berjalan.

***Tabel 2.2*** *Simbol-Simbol Activity Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Arti** |
|  | *Activity*, memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
|  | *Action*, state dari sistem yang mencerminkan eksekusi dari suatu aksi. |
|  | *Initial Node*, bagaimana objek dibentuk atau diawali. |
|  | *Final Node,* bagaimana objek dibentuk dan dihancurkan. |
|  | *Fork Note*, satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran. |

(Sumber: M Teguh Prihandoyo, 2018)

Tabel 2.2 *Simbol-Simbol* Pada *Activity Diagram* diatas adalah tabel yang menjabaran simbol dan arti dari masing-masing simbol yang digunakan pada *Activity Diagram*.

3. *Sequence Diagram* Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa message yang digambarkan terhadap waktu.

***Tabel 2.3*** Simbol-Simbol Sequence Diagram

|  |  |
| --- | --- |
| **Simbol** | **Arti** |
|  | *Life Line, objek entity*, antarmuka yang saling berinteraksi. |

***Tabel Lanjutan 2.3*** Simbol-Simbol Sequence Diagram

|  |  |
| --- | --- |
|  | Message, spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |

(Sumber: M Teguh Prihandoyo, 2018)

Tabel 2.3 *Simbol-Simbol* Pada *Sequence Diagram* diatas adalah tabel yang menjabarkan simbol dan arti dari masing-masing simbol yang digunakan pada *Sequence Diagram*.

4. *Class diagram,* Merupakan gambaran struktur dan deskripsi dari class, package, dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

***Tabel 2.4*** *Simbol-Simbol Class Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Arti** |
|  | *Generalization,* hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek induk (ancestor). |
|  | *Collaboration*, deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor. |
|  | *Realization,* operasi yang benar-benar dilakukan oleh suatu objek. |

***Tabel Lanjutan 2.4*** *Simbol-Simbol Class Diagram*

|  |  |
| --- | --- |
|  | Dependency, hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya. |
|  | *Class,* himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
|  | *Association*, apa yang menghubungkan antara objek. |

(Sumber: M Teguh Prihandoyo, 2018)

Tabel 2.4 *Simbol-Simbol* Pada *Class Diagram* diatas adalah table yang menjabarkan simbol dan arti masing-masing simbol yang digunakan pada *Class Diagram*.

**2.8 Pengertian *HTML***

*HTML* (*Hyper Text Mark-up Language*) merupakan bahasa yang digunakan untuk menulis halaman web. Biasanya mempunyai ekstensi .*html, .html*, atau .*shtml* HTML. Tersusun atas tag-tag dan digunakan untuk menentukan tampilan dari dokumen *HTML* yang diterjemahkan oleh browser. Tag HTML tidak *case sensitive*. Jadi bisa menggunakan <HTML> atau <html>. Keduanya menghasilkan *output* yang sama (Muhammad Son Mu, 2015).

**2.9 Pengertian *PHP***

*PHP* (*Hypertext Preprocessor*) adalah bahasa yang dirancang secara khusus untuk penggunaan pada Web. *PHP* adalah tool untuk pembuatan halaman web dinamis. Pada awalnya *PHP* merupakan kependekan dari Personal *Home Page*. *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web. Saat ini *PHP* adalah singkatan dari *PHP (Hypertext Preprocessor*), sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri (Lutfi, 2017).

**2.11 Pengertian MySQL**

*MySQL* (*My Structure Query Language)* adalah sebuah aplikasi atau sistem untuk mengelola database atau manajemen data. Untuk menyimpan segala informasi ke komputer menggunakan data. *MySQL* bertugas mengatur dan mengelola data-data pada database, selain itu *MYSQL* dikenal sebagai sistem yang *efisien* dan *relibel* proses *query* cepat dan mudah, sehingga cocok digunakan untuk aplikasi berbasis web (Sismadi and Santoso, 2018).

Kelebihan yang dimiliki oleh *MySQL* adalah sebagai berikut :

1. Bersifat Open *source*, yang memiliki kemampuan untuk dapat dikembangkan lagi.
2. Menggunakan bahasa *SQL* (*Structure Query Language*), yang merupakan standar bahasa dunia dalam pengolahan data.
3. *Super performance* dan *reliabel*, tidak bisa diragukan pemrosesan databasenya sangat cepat dan stabil.
4. Sangat mudah dipelajari (*ease of use*).
5. Memiliki dukungan support (*grou*p) pengguna *MySQL*.
6. Mampu lintas *Platform*, dapat berjalan diberbagai sistem operasi.
7. *Multiuser*, dimana *MySQL* dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami konflik.

**2.12 Pengertian *Xampp***

Pengertian *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 apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan web *server* yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis (ambarita, 2018).

**2.13 Pengertian Javascript**

JavaScript adalah bahasa script yang ditempelkan pada kode HTML dan proses pada sisi klien, sehingga kemampuan dokumen HTML menjadi lebih luas (IpanRipai, 2017).

Javascript (js) ialah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu web (Ayu and Permatasari, 2018).

**2.26 Penelitian Terdahulu**

Pelaksanaan penelitian terdahulu ini dimaksudkan untuk menggali informasi tentang ruang penelitian yang berkaitan dengan penelitian ini. Dengan penelusuran penelitian ini akan dapat dipastikan sisi ruang yang akan diteliti yang dapat diteliti dalam ruangan ini, dengan harapan penelitian ini tidak tumpang tindih dan tidak terjadi penelitian ulang dengan penelitian terdahulu. Penelitian terdahulu yang berhasil dipilih untuk dikedepankan dapat dilihat ditabel dibawah ini :

***Tabel 2.9*** *Penelitian Terdahulu*

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Peneliti** | **Judul Penelitian** |
| 1. | Ahmad Faisol & Sidik Noertjahjono, 2019. | PERANCANGAN SISTEM PELAYANAN KELURAHAN DIGITAL TERPADU  (SIPERDIT) BERBASIS PROGRESSIVE WEB APP (PWA). |

***Tabel Lanjutan 2.9*** *Penelitian Terdahulu*

|  |  |  |
| --- | --- | --- |
| 2. | Abdul Hafiz Syaputra , Ucuk Darulsalam,  & Winarsih, 2021. | Rancang Bangun Sistem Pengelolaan *Laundry* menggunakan Metode *Waterfall*. |
| 3. | Muhammad Yasin Simargolang &Nurmala Nasution, 2018. | Aplikasi Pelayanan Jasa Laundry Berbasis WEB (Studi Kasus : Pelangi Laundry Kisaran). |
| 4. | Naufal Al - Fikri, 2020. | RANCANG BANGUN APLIKASI OFFICIAL STORE BERBASIS PROGRESIVE WEB APP  (STUDI KASUS: ONE STOP HYDROPONICS SURABAYA) . |
| 5. | Welly Dian Astika, Adam Hendra Brata & Komang Candra Brata, 2019. | Perancangan Aplikasi Asistensi Skripsi Mahasiswa berbasis Progressive Web App dengan Metode Prototyping. |

Tabel 2.9 Penelitian Terdahulu tabel ini berisi kumpulan jurnal utama yang menjadi acuan pembuatan tugas akhir ini berjudul "Rancang Bangun Sistem Informasi Pelayanan Jasa Pada Rumah Laundry 381 Digital dengan Progressive Web App" dalam pembuatan laporan tugas akhir maupun perancangan sistem yang akan dilampirkan beserta laporan. Dalam tabel terdapat nama peneliti, tahun terbit dan judul dari jurnal peneliti tersebut. Tiap judul pada jurnal memiliki teori yang berbeda-beda dan dapat dijadikan literatur dalam pembuatan laporan tugas akhir peneliti.

1. Pada jurnal yang ditulis Ahmad Faisol & Sidik Noertjahjono, 2019, berjudul Perancangan Sistem Pelayanan Kelurahan Digital Terpadu (Siperdit) Berbasis Progressive Web App (PWA). Penelitian ini membangun sebuah sistem informasi pelayanan keluhan digital (Siperdit) terpadu masyarakat yang menjadi tanggung jawab petugas Kelurahan atau Desa dikarenakan Sistem Pelayanan Kelurahan masyarakat masih semi manual dimana warga masih harus datang ke lokasi pelayanan yang terkadang sampai berkali-kali karena ketidaktahuan persyaratan yang harus dipenuhi, petugas yang tidak ada di tempat, atau karena alasan lain. Karena pada sistem pelayanan yang dibangun Ahmad Faisol & Sidik Noertjahjono berbasis Progressive Web App dan membahas pelayanan jasa keluhan masyarakat, penulis menjadikan sebagai *literatur* alur sistem pelayanan penulis pada sistem ini.
2. Pada jurnal yang ditulis oleh Abdul Hafiz Syaputra , Ucuk Darulsalam & Winarsih, 2021, berjudul Rancang Bangun Sistem Pengelolaan Laundry menggunakan Metode Waterfall. Jurnal ini membahas tentang perancangan dan pembangunan sistem pengelolaan laundry yang menggunakan metode waterfall, dimana sistem yang baru memiliki tampilan menu-menu yang lebih menarik, yang memperbaiki sistem menjadi lebih optimal dan terintegrasi dengan baik. Penulis tugas akhir ini menjadikan jurnal Abdul Hafiz Syaputra , Ucuk Darulsalam, & Winarsih sebagai literatur arsitektur sistem yang akan digunakan pada sistem yang akan dibangun penulis tugas akhir ini.
3. Jurnal berjudul Aplikasi Pelayanan Jasa Laundry Berbasis Web (Studi Kasus : Pelangi Laundry Kisaran) adalah jurnal yang ditulis oleh Muhammad Yasin Simargolang &Nurmala Nasution, 2018. Pada jurnal ini memaparkan tentang pembuatan Aplikasi Pelayanan Jasa Laundry berbasis website pada pelangi laundry kisaran sebagai solusi dari permasalahan yang ada dikarenakan pada sistem yang sedang berjalan masih belum memiliki sistem informasi sendiri dan masih menggunakan sistem manual, laundry ini juga belum mempunyai sistem penyimpanan database yang akurat sehingga semua data masih disimpan kedalam sebuah buku. Disamping itu para pelanggan juga tidak mendapatkan informasi yang up-to-date tentang jasa laundry yang ditawarkan dan juga pelanggan yang merasa kerepotan pada saat mengantar bahkan mengambil laundry yang telah selesai dikarenakan tidak memiliki banyak waktu untuk pergi ke laundry dengan alasan kesibukan. Penulis tugas akhir ini menjadikan jurnal dari Muhammad Yasin Simargolang &Nurmala Nasution sebagai literatur penulisan tugas akhir dikarenakan kesamaan pada proses analisis dan perancangan yaitu menggunakan *Use Case Diagram, Activity Diagram, Class Diagram,* *Sequence Diagram.*
4. Rancang Bangun Aplikasi Official Store Berbasis Progressive Web App (Studi Kasus : One Stop Hydroponics Surabaya) adalah jurnal yang ditulis oleh Naufal Al - Fikri, 2020. Jurnal ini berisi tentang pembuatan aplikasi berbasis Progressive Web App pada toko One Stop Hydroponics Surabaya yang diharapkan dapat memudahkan dalam proses penyampaian informasi produk terbaru dari toko serta dapat membantu dalam penjulan toko karena dapat digunakan kapanpun dan dimanapun. Dalm proses pembuatan aplikasi Naufal Al - Fikri menggunakan bahasa pemograman Javascript,CSS dan database MySql. Penulis tugas akhir ini menjadikan jurnal Naufal Al - Fikri sebagai literatur untuk memahami lebih jauh bahasa pemograman dan penggunaan database MySql dalam pembuatan aplikasi.
5. Welly Dian Astika, Adam Hendra Brata & Komang Candra Brata, 2019, menulis sebuah jurnal berjudul Perancangan Aplikasi Asistensi Skripsi Mahasiswa berbasis Progressive Web App dengan Metode Prototyping. Jurnal ini membahas tentang aplikasi perancangan aplikasi asistensi skripsi mahasiswa pada Fakultas Ilmu Komputer Universitas Bandung (FILKOM UB) yang akan melakukan asistensi skripsi di kampus. Karena adanya kesamaan sistem pada judul yaitu berbasis progressive web app maka tugas akhir ini menjadikan Welly Dian Astika, Adam Hendra Brata & Komang Candra Brata sebagai literatur untuk lebih memahami proses dan perancangan dari PWA yang dilakukan oleh mahasiswa FILKOM UB.

**2.27 Pengertian Spesifikasi File**

Spesifikasi file menjelaskan tentang file-file yang akan digunakan dalam pengolahan database, serta sebagai media penyimpanan data (Ardiyah & palasara (2018:1),2018).