

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam proses sistem informasi jasa *laundry* pada Keisya *Laundry* Karawang diperlukan penjelasan atau teori yang dapat mendukung dan menunjang keberhasilan dalam pembuatan sistem informasi jasa *laundry* untuk memberikan penjelasan kepada pemilik keisya laundry bagaimana fungsi dan manfaat dari sistem informasi jas *laundry* ini.

A. Konsep Dasar Sistem Informasi

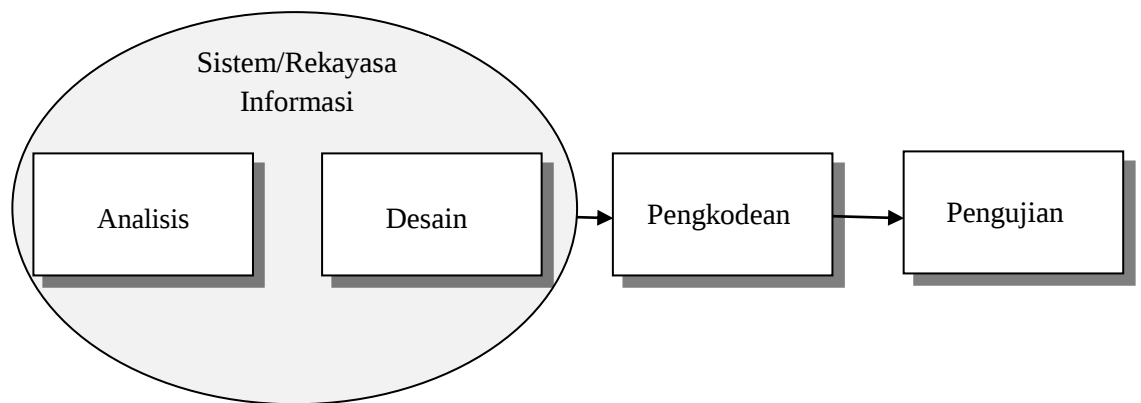
Pengertian Sistem Menurut Jogianto (2014:1) mengemukakan bahwa “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk mnyelesaikan suatu sasaran yang tertentu”.

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Informasi memiliki tingkat kualitas yang ditentukan, diantaranya tingkatan keakuratan, tepat waktu, dan relevan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.(Agus Mulyanto, 2009).

Dari beberapa makna tersebut dapat disimpulkan bahwa, menurut Saputra dan Feni Agustin (2013:14) “Sistem Informasi adalah kegiatan dari prosedur-prosedur yang diorganisasikan, bilamana dieksekusi akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam organisasi”.

B. Konsep Dasar Model SDLC (*Software Development Life Cycle*)

Menurut Sukamto dan M. Shalahuddin (2013:28) “Model SDLC air terjun (*Waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). Berikut adalah gambar model air terjun:



Sumber: Sukamto dan M. Shalahuddin (2013:29)

Gambar II.1.
Model *Waterfall*

Tahapan-tahapan dalam model *waterfall* terdiri dari:

1. Analisis kebutuhan perangkat lunak

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

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur

perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program/*Coding*

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian/*Testing*

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

5. Pendukung (*Support*) atau pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

C. Konsep Dasar Pemrograman

Pemrograman adalah proses mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dengan menggunakan suatu bahasa pemrograman (Binarso, dkk, 2012:76-77). Pemrograman terstruktur merupakan proses mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program yang memiliki rancang bangun terstruktur dan tidak berbelit-belit sehingga mudah ditelusuri, dipahami dan dikembangkan oleh siapa saja. Konsep pemrograman terstruktur memegang peranan penting dalam merancang, menyusun, memelihara, dan mengembangkan suatu program khususnya program aplikasi yang besar dan komplek.

Ciri-ciri teknik pemrograman terstruktur adalah:

1. Teknik pemecahan masalahnya tepat dan benar.
2. Algoritma pemecahan masalahnya sederhana, standard an efektif.
3. Struktur logikanya benar dan mudah dipahami.
4. Memiliki 3 (tiga) struktur dasar yaitu:
 - a) *Sequence Structure* (Struktur Urut)
 - b) *Selection Structure* (Struktur Keputusan/Kondisi)
 - c) *Looping Structure* (Struktur Perulangan)
5. Menghindari penggunaan *statement* GOTO (peralihan proses tanpa syarat tertentu).
6. Membutuhkan biaya testing yang rendah.
7. Memiliki dokumen yang baik.
8. Membutuhkan biaya perawatan dan pengembangan yang rendah.

Standar program yang baik dalam pemrograman terstruktur dilihat dari teknik pemecahan masalahnya ada beberapa teknik, diantaranya:

1. Teknik *Top-Down* yaitu suatu masalah yang kompleks dibagi-bagi kedalam beberapa kelompok masalah yang lebih kecil.
2. Teknik *Bottom-Up* yaitu pemecahan masalah dilakukan dengan menggabungkan prosedur-prosedur yang ada menjadi satu kesatuan program guna menyelesaikan masalah tersebut.

D. PHP (*Hypertext Preprocessor*)

Munurut Zuliarso (2012:5) menerangkan bahwa:

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman pada sisi *server* yang memperbolehkan *programmer* menyisipkan perintah-perintah perangkat lunak *web server* (Apache, IIS, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke *browser* yang me-request-nya, contohnya adalah bagaimana memungkinkannya memasukkan tanggal sekarang pada sebuah halaman *web* setiap kali tampilan tanggal dibutuhkan.

Sesuai dengan fungsinya yang berjalan di sisi *server* maka PHP adalah bahasa pemrograman yang digunakan untuk membangun teknologi *web application*. PHP telah menjadi bahasa *scripting* untuk keperluan umum yang pada awalnya hanya digunakan untuk pembangunan web yang menghasilkan halaman web dinamis. Untuk tujuan ini, kode PHP tertanam ke dalam dokumen sumber *HTML* dan diinterpretasikan oleh *server web* dengan modul PHP prosesor, yang menghasilkan dokumen halaman *web*. Sebagai bahasa pemrograman untuk tujuan umum, kode PHP diproses oleh aplikasi penerjemah dalam modus baris-baris perintah modus dan melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program dichannel *output* standar. Hal ini juga dapat berfungsi sebagai aplikasi grafis. PHP tersedia sebagai prosesor untuk

server web yang paling modern dan sebagai penerjemah mandiri pada sebagian besar system operasi dan komputer *platform*.

E. MySQL (*My Structure Query Language*)

Munurut Sukadi (2013:3) “MySQL adalah *multiuser database* yang menggunakan bahasa *Structured Query Language* (SQL). MySQL dalam operasi *client server* melibatkan *server daemon* MySQL disisi *server* dan berbagai macam program serta *library* yang berjalan disisi *client*”. MySQL mampu mengangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TEX, mengaku mampu menyimpan data lebih dari 40 *database*, 10.000 tabel, dan sekitar 7.000.000 baris totalnya kurang lebih 100 *Gigabyte* data.

F. UML (*Unified Modeling Language*)

Menurut Dharwiyanti dan Romi Satrio Wahono (2003:2) “UML (*Unified Modeling Language*) adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem”. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Bahasa pemodelan grafis telah ada diindustri perangkat lunak sejak lama. Pemicu utama di balik semuanya adalah bahwa bahasa pemrograman berada pada tingkat

abstraksi yang terlalu tinggi untuk memfasilitasi diskusi tentang desain. Dengan menggunakan notasi-notasi seperti UML, alur logika dari perangkat lunak yang akan dikembangkan bisa mudah untuk dipahami.

Berikut ini definisi dari beberapa model UML (*Unified Modeling Language*) yaitu:

1. Use Case Diagram

Use Case Diagram merupakan inti fungsionalitas koheren yang diekspresikan sebagai transaksi-transaksi yang terjadi antara *actor* dan *system*.

2. Activity Diagram

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

3. Component Diagram

Diagram komponen (*Component Diagram*) digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen *software* sistem. Diagram ini dapat menunjukkan bagaimana kode pemrograman dibagi menjadi modul-modul (komponen).

4. Deployment Diagram

Deployment Diagram menunjukan perangkat keras sistem dan perangkat lunak dalam perangkat keras yang ada. Model *deployment diagram* mewakili pandangan pengembangan sistem sehingga akan hanya ada satu *deployment diagram* untuk satu sistem.

G. Dreamweaver CS6

Menurut Sadeli (2011:1) memberikan pengertian bahwa “*Dreamweaver* merupakan suatu perangkat lunak *web editor* keluaran *Adobe System* yang digunakan untuk membangun dan mendesign suatu *website* dengan fitur-fitur yang menarik dan kemudahan dalam penggunaannya”.

Selain mudah digunakan dalam membuat dan mendesign *web*, *dreamweaver* juga menambah fleksibilitas dengan bahasa pemrograman *web* lainnya dan dapat berintegrasi pada beberapa perangkat lunak lainnya.

Adobe Dreamweaver merupakan program membuat atau mengedit *web* yang dibuat *Adobe System* yang dulu dikenal sebagai *Macromedia Dreamweaver* dibuat *Macromedia*. Program ini banyak digunakan oleh para pengembang *web* karena fiturnya yang menarik dan kemudahan untuk menggunakannya. *Macromedia Dreamweaver* berubah menjadi *Adobe Dreamweaver* karena *Macromedia* diakuisi oleh *Adobe System* sehingga seluruh produk yang dibuat oleh *Macromedia* kini diawali dengan kata *Adobe*.

H. Photoshop

Menurut Anhar (2007:1) mengemukakan bahwa “*Adobe Photoshop CS3* merupakan *software* yang digunakan untuk *design graphic* dan mengedit foto. *Adobe Photoshop CS3* ini menyediakan fitur-fitur dan kesempurnaan dari versinya yang lalu”.

Photoshop juga adalah perangkat lunak untuk mengedit gambar dalam format *BITMAP*. Oleh karena itu *photoshop* sering kali digunakan oleh *fotografer*, *desainer* dan standar *professional* seperti perataan layar secara otomatis dan teknis pencampuran komposisi. Saat ini *photoshop* merupakan

perangkat lunak terbaik dikelasnya, hampir tidak ada tandingannya. Adobe photoshop merupakan *software* pengolah gambar yang sangat digemari, salah satunya karya dari Adobe Corporation, setelah merilis beberapa photoshop, versi photoshop yang terbaru adalah photoshop CS5.

Walaupun terdapat berbagai macam program yang dapat digunakan untuk mewarnai ilustrasi anda, Photoshop dengan segala kelebihan dan kekurangannya dapat dimaksimalkan untuk menghasilkan ilustrasi berwarna yang indah. Anda dapat menghasilkan ilustrasi dengan teknik berbeda dengan teman anda walaupun *tool* yang digunakan sama. Anda juga dapat mengerjakan ilustrasi berwarna secara manual untuk kemudian diedit lebih lanjut dengan photoshop. Berikut ini beberapa perubahan baru yang terdapat dalam Adobe Photoshop CS3 yang berbeda dengan versi sebelumnya.

1. Tampilan baru pada jendela dan lembar kerja yang menawan, mempermudah anda untuk mengatur tampilan kerja agar lebih nyaman.
2. Tampilan panel *Tools* yang fleksibel dapat ditampilkan menjadi satu kolom atau dua kolom.
3. Tombol *Quick Mask* tampil sebagai tombol *single* menyerupai tombol *On/Off* untuk mengaktifkan dan menonaktifkan fungsi *Quick Mask*.
4. Panel ditampilkan dalam sebuah *dock* yang secara fleksibel dapat diubah sesuai dengan ukuran yang diinginkan sehingga anda akan lebih leluasa memaksimalkan tampilan lembar kerja.
5. Mode *Full Screen* baru yang diberi nama *Maximize Screen Mode*, berfungsi memaksimalkan ukuran jendela gambar semaksimal mungkin menyesuaikan dengan layer.

I. Database (Basis Data)

Menurut Kadir (2009:14) memberikan pengertian bahwa “*database* sering didefinisikan sebagai kumpulan data yang terkait. Secara teknis, yang berada dalam sebuah *database* adalah sekumpulan tabel atau objek lain (*index*, *view* dan lain-lain)”.

Tujuan utama pembuatan *database* adalah untuk memudahkan dalam mengakses data. Data dapat ditambahkan , diubah, dihapus atau dibaca dengan relatif mudah dan cepat. Sebuah tabel (atau kadang disebut relasi) berisi sejumlah baris dan kolom. Perpotongan baris dan kolom menyatakan sebuah data.

Istilah *database* berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukan hal-hal diluar bidang elektronika. Catatan yang mirip dengan *database* sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar kuitansi dan kumpulan data yang berhubungan dengan bisnis.

J. Cascading Style Sheet (CSS)

Menurut Kun (2010:1) menjelaskan bahwa “CSS merupakan salah komponen untuk memperindah *website* yang anda buat, tetapi hanya sebatas mengganti tampilan luarnya saja dan sama sekali tidak menimbulkan animasi yang menakjubkan”.

Menurut Nugroho (2014:27) mengemukakan bahwa “*file css (Cascading Style Sheets* dipakai untuk mengatur tampilan *website*, terutama pengaturan pada desain tabel, teks dan menu dalam halaman *website* dan program admin”.

Ada dua jenis CSS yang dapat digunakan yaitu CSS yang bersifat internal dan eksternal. CSS eksternal lebih mudah digunakan, karena *script CSS* akan

tersimpan berupa *file* CSS tersendiri, jadi akan lebih mudah melakukan pengeditan, karena biasanya dihubungkan dengan link. Jika melakukan perubahan format, cukup melakukan perubahan pada *file* CSS, maka halaman yang terhubung dengan *file* CSS tersebut akan ikut berubah. Jika menggunakan CSS bersifat internal pada halaman, maka bila ada perubahan format harus melakukan banyak pengeditan pada semua file.

K. ERD (*Entity Relationship Diagram*)

ERD (*Entity Relationship Diagram*) adalah model konseptual yang mendeskripsikan hubungan antara penyimpanan (dalam DFD). ERD digunakan untuk memodelkan struktur data dan hubungan antar data (Binarso, dkk, 2012:74).

Ada 2 (dua) komponen utama pementuk Model ERD, yaitu Entitas (*Entity*) dan Relasi (*Relation*). Entitas (*Entity*) ialah individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Himpunan Entitas (*Entity Set*) ialah sekelompok entitas yang sejenis dan berada dalam lingkup yang sama.

Atribut (*Attributes/Properties*) mendeskripsikan karakteristik (properti) dari suatu entitas.

1. Kardinalitas Relasi

Dalam ERD, hubungan (relasi) dapat terdiri dari sejumlah entitas yang disebut derajat relasi. Derajat relasi maksimum disebut dengan kardinalitas, sedangkan derajat minimum disebut dengan modalitas. Jadi kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas

pada himpunan entitas lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa Satu ke satu (*One to One*), Satu ke banyak (*One to Many*), Banyak ke banyak (*Many to Many*).

L. LRS (*Logical Relational Structure*)

Pengetian LRS (*Logical Relational Structure*) menurut Prihatna (2005:65) menjelaskan bahwa “LRS adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas”.

Logical Relational Structure dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan namanya unik. LRS terdiri dari *link-link* diantara tipe *record* yang menunjukkan dari arah satu tipe *record* lainnya. Banyak link dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link* tipe *record*.

Pengembangan LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan ERD dan langsung dikonversi ke LRS.

2.2. Penelitian Terkait

Dalam hal ini, tiga (2) jurnal yang dipergunakan sebagai kutipan hasil penelitian sebelumnya yaitu:

Menurut Okstria, dkk (2011:24) Perkembangan teknologi saat ini yang semakin canggih membuat kebutuhan akan informasi di segala bidang menjadi sangat penting. Hal ini diperoleh dari pengetahuan ataupun teknologi itu sendiri. Dimana perkembangan teknologi informasi tersebut diikuti dengan persaingan di dunia bisnis. Sistem ini dibangun dengan teknologi komputer berbasis web. Sistem informasi *laundry* ini

dibangun dengan menggunakan bahasa pemrograman PHP dengan didukung basis data MySQL.

Menurut Husain, dkk (2013:2) Memberikan pernyataan bahwa: Kehadiran usaha jasa laundry memberikan dampak positif bagi mahasiswa, pekerja, dan rumah tangga, salah satunya dapat meringankan beban pekerja rumah tangga yang semula mencuci pakaian dikerjakan sendiri menjadi tidak dengan adanya jasa laundry, selain itu juga lebih efisien waktu dan tenaga.