

BAB II

LANDASAN TEORI

Dalam penyelesaian Tugas Akhir ini menggunakan landasan teori yang digunakan dalam menyelesaikan permasalahan.

2.1 Sistem Aplikasi

Menurut Jogianto (2005), sistem merupakan satu kesatuan komponen atau element-element yang digabungkan menjadi satu untuk memudahkan aliran informasi, materi atau energi. Sistem merupakan beberapa komponen yang saling terhubung, untuk mencapai tujuannya komponen-komponen tersebut bekerja sama dengan menerima *input* dan menghasilkan *output*. Definisi sistem tersebut menghasilkan kesimpulan bahwa sistem adalah suatu jaringan yang memiliki hubungan dan keterkaitan antara bagian dan prosedur-prosedur yang terkumpul dalam suatu organisasi untuk melakukan kegiatan dan mencapai tujuan tertentu.

2.2 Pendayagunaan Aset Tetap PT PLN (Persero) Berupa Tanah dan Bangunan

Ada beberapa istilah menurut Keputusan Direksi PT PLN Persero (2013), diantaranya yaitu :

1. Aset tetap merupakan aktiva tetap atau aset milik PLN berupa tanah dan bangunan, tidak dimanfaatkan untuk dijual dalam rangka kegiatan normal PLN dan memiliki masa manfaat lebih dari satu tahun. Begitu pula dengan kelompok Aset Tetap Tidak Beroperasi (ATTB) dan Aset Tetap yang Belum Dimanfaatkan (ATBM).

2. Pendayagunaan aset tetap merupakan pemanfaatan tanah dan bangunan yang dioptimalkan, baik dikelola sendiri maupun melalui kerjasama dengan mitra.
3. Sewa merupakan pendayagunaan aset tetap oleh mitra dalam jangka waktu tertentu dan PLN menerima imbalan uang tunai.

Pendayagunaan aset tetap PLN dilakukan dengan beberapa cara diantaranya yaitu:

- a. Bangunan Guna Serah (*Build Operate and Transfer*).
- b. Bangunan Serah Guna (*Build Transfer and Operate*).
- c. Kerjasama Operasi (KSO)
- d. Kerjasama Usaha (KSU)
- e. Sewa
- f. Pinjam Pakai
- g. Sewa Kelola

PLN berhak mendapatkan imbalan uang tunai dari mitra berupa sewa bulanan atau tahunan yang dibayarkan sekaligus dimuka yang dituangkan dalam perjanjian.

2.3 PT PLN (Persero)

PT PLN (Persero) Distribusi Jawa Timur merupakan salah satu unit pelaksana induk dibawah PT PLN (Persero) yang merupakan Badan Usaha Milik Negara (BUMN) yang ditunjuk sebagai Pemegang Izin Usaha Ketenagalistrikan Untuk Kepentingan Umum (PIUKU) melalui Undang-undang Nomor 30 tahun 2009.

2.4 Formula Sewa Barang Milik Kekayaan Negara

Menurut Peraturan Menteri Keuangan (2007), besarnya biaya sewa Barang Milik Negara dihitung dengan formula sebagai berikut:

1. Formula Sewa Tanah Kosong

$$St = 3,33 \% \times (Lt \times \text{Nilai tanah})$$

Keterangan:

a. St = Sewa tanah

$$Lt = \text{Luas tanah (M}^2\text{)}$$

Nilai Tanah = Nilai tanah berdasarkan hasil penilaian dengan estimasi terendah menggunakan NJOP (per M²) yaitu Rp.394.000.

b. Luas tanah dihitung berdasarkan pada gambar situasi/peta tanah atau sertifikat tanah dalam meter persegi.

2. Sewa Tanah dan Bangunan

$$Stb = (3,33\% \times Lt \times \text{Nilai tanah}) + (6,64\% \times Lb \times Hs \times Nsb)$$

Keterangan:

a. Lb = Luas lantai Bangunan (M²)

Hs = Harga satuan bangunan standar dalam keadaan baru (Rp/M²), berdasarkan NJOP yaitu Rp.429.000.

NSB = Nilai sisa bangunan (%) :

1. Penyusutan untuk bangunan permanen = 2 % / tahun.
2. Penyusutan untuk bangunan semi permanen = 4 % / tahun.
3. Penyusutan untuk bangunan darurat = 10 % / tahun.
4. Penyusutan maksimal 80 %.

Menghitung NSB / penyusutan bangunan:

$100\% - ((\text{thn sekarang} - \text{thn bangunan}) \times \text{NSB} (\%))$

- b. Luas bangunan dihitung berdasarkan luas lantai bangunan sesuai gambar dalam meter persegi.
- c. Harga satuan bangunan :
 1. Harga Satuan bangunan per M² sesuai klasifikasi/tipe dalam keadaan baru berdasarkan keputusan pemerintah daerah kabupaten/kota setempat pada tahun yang bersangkutan.
 2. Harga satuan tertinggi rata-rata per M² bangunan bertingkat untuk Bangunan Gedung Negara dapat dilihat pada tabel satu, dua dan tiga.
- d. Dalam hal sisa bangunan menurut umur tidak sesuai dengan kondisi nyata, maka NSB ditetapkan berdasarkan kondisi bangunan sebagai berikut:
 1. baik = 85% s.d. 100 % siap pakai/perlu pemeliharaan awal.
 2. rusak ringan = 70% s.d. < 85% rusak sebagian non struktur.
 3. rusak berat = 55% s.d. < 70% rusak sebagian non struktur/struktur.
 4. rusak berat = 35% s.d. < 55% rusak sebagian besar non struktur/struktur.

Sewa Prasarana Bangunan

$$Sp = 6,64\% \times Hp \times Nsp$$

- a. Keterangan:

Sp = sewa prasarana bangunan (Rp/tahun)

Hp = harga prasarana bangunan dalam keadaan baru (Rp)

N_{sp} = nilai sisa prasarana bangunan (%)

b. Besar penyusutan / tahun dihitung dengan ketentuan:

1. pekerjaan halaman = 5 %
2. mesin/instalasi = 10 %
3. furniture/elektronik = 25 %
4. penyusutan maksimal = 80 %

Harga satuan tertinggi rata-rata per M2 bangunan bertingkat untuk Bangunan Gedung Negara. Data satuan rata-rata bangunan tertinggi ini akan digunakan untuk perhitungan sewa aset bangunan sesuai dengan banyaknya lantai bangunan yang ada. Harga satuan rata-rata bangunan akan dihitung permeter persegi dari luas bangunan tersebut. Berikut daftar tabel harga satuan rata-rata bangunan dapat dilihat pada tabel 2.1.

Tabel 2. 1 Harga Satuan Rata-rata Bangunan

Jumlah lantai bangunan	Harga satuan per M2 tertinggi
Bangunan 1 lantai	Rp.1000 standar harga gedung bertingkat
Bangunan 2 lantai	Rp.1090 standar harga gedung bertingkat
Bangunan 3 lantai	Rp.1120 standar harga gedung bertingkat
Bangunan 4 lantai	Rp.1135 standar harga gedung bertingkat
Bangunan 5 lantai	Rp.1162 standar harga gedung bertingkat
Bangunan 6 lantai	Rp.1197 standar harga gedung bertingkat
Bangunan 7 lantai	Rp.1236 standar harga gedung bertingkat
Bangunan 8 lantai	Rp.1265 standar harga gedung bertingkat
Bangunan 9 lantai	Rp.1299 standar harga gedung bertingkat
Bangunan 10 lantai	Rp.1333 standar harga gedung bertingkat

Jumlah lantai bangunan	Harga satuan per M2 tertinggi
Bangunan 11 lantai	Rp.1364 standar harga gedung bertingkat
Bangunan 12 lantai	Rp.1393 standar harga gedung bertingkat
Bangunan 13 lantai	Rp.1420 standar harga gedung bertingkat
Bangunan 14 lantai	Rp.1445 standar harga gedung bertingkat
Bangunan 15 lantai	Rp.1468 standar harga gedung bertingkat
Bangunan 16 lantai	Rp.1489 standar harga gedung bertingkat
Bangunan 17 lantai	Rp.1508 standar harga gedung bertingkat
Bangunan 18 lantai	Rp.1525 standar harga gedung bertingkat
Bangunan 19 lantai	Rp.1541 standar harga gedung bertingkat
Bangunan 20 lantai	Rp.1556 standar harga gedung bertingkat
Bangunan 21 lantai	Rp.1570 standar harga gedung bertingkat
Bangunan 22 lantai	Rp.1584 standar harga gedung bertingkat
Bangunan 23 lantai	Rp.1597 standar harga gedung bertingkat
Bangunan 24 lantai	Rp.1610 standar harga gedung bertingkat
Bangunan 25 lantai	Rp.1622 standar harga gedung bertingkat
Bangunan 26 lantai	Rp.1634 standar harga gedung bertingkat
Bangunan 27 lantai	Rp.1654 standar harga gedung bertingkat
Bangunan 28 lantai	Rp.1656 standar harga gedung bertingkat
Bangunan 29 lantai	Rp.1666 standar harga gedung bertingkat
Bangunan 30 lantai	Rp.1676 standar harga gedung bertingkat
Bangunan 31 lantai	Rp.1686 standar harga gedung bertingkat
Bangunan 32 lantai	Rp.1695 standar harga gedung bertingkat

Jumlah lantai bangunan	Harga satuan per M2 tertinggi
Bangunan 33 lantai	Rp.1704 standar harga gedung bertingkat
Bangunan 34 lantai	Rp.1713 standar harga gedung bertingkat
Bangunan 35 lantai	Rp.1722 standar harga gedung bertingkat
Bangunan 36 lantai	Rp.1730 standar harga gedung bertingkat
Bangunan 37 lantai	Rp.1738 standar harga gedung bertingkat
Bangunan 38 lantai	Rp.1746 standar harga gedung bertingkat
Bangunan 39 lantai	Rp.1754 standar harga gedung bertingkat
Bangunan 40 lantai	Rp.1761 standar harga gedung bertingkat

2.5 System Development Life Cycle (SDLC)

Metode yang digunakan untuk membuat perancangan aplikasi ini adalah SDLC. Sebelum menjelaskan metode perancangan aplikasi ini saya akan menjelaskan beberapa metode pengembangan perangkat lunak yang lain diantaranya yaitu:

1. *Waterfall*

“*Classic Life Cycle*” atau model *Waterfall* merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). Model ini menggunakan pendekatan secara sistematis dan runtut dimulai dari kebutuhan level sistem selanjutnya menuju tahap analisis, desain, *coding*, *testing / verification*, dan *maintenance*. Model ini disebut dengan *waterfall* dikarenakan tahap pada model ini harus dilakukan secara berurutan.

2. *Prototype*

Prototyping merupakan pendekatan rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana sebuah perangkat lunak atau

komponen-komponen perangkat lunak bekerja dalam lingkungannya sebelum tahapan konstruksi actual dilakukan.

3. RAD

Rapid application development (RAD) atau *rapid prototyping* merupakan Model pembangunan perangkat lunak yang tergolong dalam teknik bertingkat. *Rapid application development* (RAD) lebih menekankan pada siklus pembangunan pendek, singkat, dan cepat. Pada *Rapid application development* (RAD) waktu yang singkat merupakan batasan yang penting untuk model ini. *Rapid application development* menggunakan metode iteratif (berulang) dalam mengembangkan sistem dimana *working model* (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) user. *Rapid application development* (RAD) mengadopsi model *waterfall* dan pembangunan dalam waktu singkat dicapai dengan menerapkan *component based construction*.

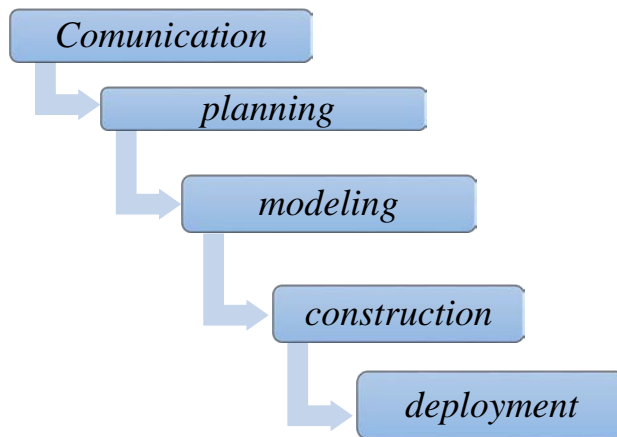
4. Agile software development

Agile merupakan jenis pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dan memerlukan pengembang terhadap perubahan dalam bentuk apapun. Dalam *Agile Software Development* ada beberapa hal yang dapat diperhatikan yaitu interaksi lebih penting dari pada proses dan alat, *software* yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klien lebih penting dari pada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana.

2.6 Model Waterfall

Model SDLC yang digunakan yaitu model *Waterfall* karena pengerjaan proyek akan lebih mudah terjadwal dengan baik dan mudah dikontrol, selain itu model ini bersifat sistematis dan berurutan dalam pengerjaannya sehingga penulis harus mengerjakan setiap fase secara berurutan. Menurut Pressman (2015), nama lain dari Model *Waterfall* adalah Model Air Terjun kadang dinamakan sebagai siklus hidup klasik (*classic life cycle*), dimana hal ini merupakan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak, yang dimulai dengan spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak lengkap yang dihasilkan.

Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*. Model ini disebut dengan *waterfall* dikarenakan tahap pada model ini harus dilakukan secara berurutan. Berikut adalah gambar yang menunjukkan tahapan umum dari model proses *waterfall*, dapat dilihat pada gambar 2.1.



Gambar 2. 1 Model *Waterfall* Menurut Pressman (2015)

Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* menurut Pressman (2015):

a. *Communication*

Langkah pertama diawali dengan komunikasi kepada pelaksana aset pada PT PLN (Persero) Distribusi Jawa Timur. Langkah awal ini merupakan langkah penting karena menyangkut pengumpulan informasi tentang kebutuhan konsumen/pengguna.

b. *Planning*

Setelah proses *communication* ini, kemudian menetapkan rencana untuk pengerjaan *software* yang meliputi tugas-tugas teknis yang akan dilakukan, risiko yang mungkin terjadi, sumber yang dibutuhkan, hasil yang akan dibuat, dan jadwal pengerjaan.

c. *Modeling*

Pada proses *modeling* ini menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, *representasi interface*, dan detail (algoritma) prosedural.

d. *Construction*

Construction merupakan proses membuat kode (*code generation*). *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

e. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

2.7 Website

Menurut Kadir (2008), Website merupakan kumpulan halaman web (*web page*) yang pada umumnya merupakan bagian dari suatu nama *domain* atau *sub domain* di *World Wide Web* di *Internet*. *WWW* terdiri dari seluruh situs web yang diakses dari sebuah URL, *WWW* juga disebut sebagai *homepage* (halaman induk atau halaman muka) dan umumnya disimpan pada *server* yang sama. Sebuah website biasanya dibuat oleh individual, bisnis atau organisasi berdasarkan topik dan tujuan tertentu.

Website dapat ditulis atau dirubah secara dinamis menjadi HTML (*Hyper Text Markup Language*) dan diakses dengan menggunakan *software* yang disebut *Internet Browser*. Website dikenal juga dengan sebutan *HTTP client*. *Web page* dapat diakses dan dilihat dari berbagai macam alat, diantaranya *desktop computer*, *laptop computer*, PDA ataupun *cell phone* yang semuanya mempunyai koneksi *internet*. Sebuah website ditampung dalam sebuah sistem *computer* yang disebut *web-server*, dikenal juga dengan sebutan *HTTP server*. *Server* ini menggunakan *software* yang berfungsi dan mengirim respon *web page* terhadap perintah yang dilakukan oleh pengakses website.

Unsur-unsur pendukung yang dipersiapkan untuk membangun website sebagai berikut:

1. Nama *Domain* (*Domain name* / *URL – Uniform Resource Locator*).
2. Rumah Website (*Website Hosting*).
3. *Content Management System* (CMS).

2.8 *Testing*

Sesuai model *waterfall* pada tahap ke tiga yaitu *construction* yang menjelaskan tentang pembuatan *software* atau *coding*, selanjutnya dilakukan *testing* untuk menemukan kesalahan-kesalahan dalam sistem tersebut, setelah dilakukan *testing* sistem akan diperbaiki sesuai dengan kesalahan-kesalahan yang ditemukan, maka dari itu penulis menggunakan *testing* untuk pengujian sistem yang akan dibuat. Menurut Romeo (2003), *testing* merupakan proses penguatan kepercayaan pada kinerja program atau sistem yang diharapkan. *Testing Software* merupakan proses pengoperasian *software* yang dikendalikan untuk diverifikasi, mendeteksi adanya *error* dan melakukan validasi.

Verifikasi adalah pengecekan atau pengetesan entitas-entitas pada *software* untuk melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem apakah proses yang telah dituliskan sudah sesuai dengan apa yang dibutuhkan oleh pengguna.

2.8.1 *Black Box Testing*

Menurut Romeo (2003), *Black Box Testing* merupakan pengujian yang dilakukan tanpa adanya suatu pengetahuan tentang detail struktur internal dari sistem atau komponen yang dites, *Black Box Testing* juga disebut sebagai *functional testing*. Berdasarkan pada spesifikasi kebutuhan dari *software black box testing* difokuskan untuk kebutuhan fungsional pada *software*.,

Dengan adanya *black box testing*, perikayasa *software* dapat digunakan untuk kebutuhan fungsional pada suatu program. *Black box testing* dilakukan untuk melakukan pengecekan apakah sebuah *software* telah bebas dari *error* dan fungsi-fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan.

2.8.2 *Usability testing*

Menurut Badre (2002) *Usability testing* atau uji ketergantungan merupakan pengukuran kemudahan digunakan, kemudahan dipelajari oleh pengguna, kemudahan diingat ketika sedang digunakan, memiliki respon yang cepat dan telah memenuhi kebutuhan pengguna. Ada beberapa komponen yang bisa digunakan dalam *usability testing* , diantaranya *Learnability*, *Efficiency*, *Memorability*, *Errors dan Satisfaction*. Berikut adalah contoh pertanyaan yang dapat disampaikan dalam uji coba ketergunaan. Dapat dilihat pada tabel 2.2.

Tabel 2. 2 Pertanyaan *Usability Testing*

No	Pertanyaan	Ya	Tidak
Learnability			
1.	Apakah tulisan teks yang digunakan untuk halaman pada aplikasi sudah jelas bagi anda?		
2.	Apakah menu yang ada di dalam aplikasi mudah untuk dipahami?		
Efficiency			
3.	Apakah menu yang anda pilih atau anda klik pada aplikasi dapat merespon secara cepat?		
4.	Apakah ketika melakukan pencarian data pada menu cari dapat menampilkan hasil pencarian dengan cepat?		
Memoriability			
5.	Apakah anda dapat menggunakan aplikasi dengan lancar hanya dengan 1 kali pelatihan?		
Error			
6.	Apakah anda menemukan error pada menu yang anda klik?		
7.	Apakah anda menemukan menu yang tidak memunculkan respon setelah anda klik?		
Satisfaction			
8.	Apakah menurut anda aplikasi ini telah memenuhi kebutuhan anda?		
9.	Apakah menurut anda informasi yang ditampilkan pada aplikasi ini sudah sesuai?		
10.	Apakah menurut anda desain secara keseluruhan pada aplikasi penyewaan aset tanah dan bangunan sudah terlihat baik?		

Setelah membuat pertanyaan untuk uji coba ketergantungan selanjutnya dilakukan analisa data hasil *usability test* dengan menghitung persentase pertanyaan pada formulir uji ketergantungan. Dari setiap point pertanyaan yang dibuat, dihitung berapa persen yang setuju dengan pernyataan dan berapa persen yang tidak setuju dengan pernyataan.

Dari setiap *point* pertanyaan dihitung subtotal persentase dan dihitung total persentase keseluruhan jawaban yang diberikan responden, baik presentase dari jawaban yang bisa dijawab dan jawaban yang tidak bisa dijawab.

Rumus yang digunakan dalam perhitungan adalah sebagai berikut:

$$\text{Nilai} = \frac{\text{Skor}}{\text{Jumlah Responden}}$$

Untuk mengetahui keberhasilan uji coba ketergunaan digunakan skala dibawah ini:

Tabel 2. 3 Skala Keberhasilan *Usability Testing*

Skor	Kualifikasi	Hasil
85-100%	Sangat Baik	Berhasil
66-84%	Baik	Berhasil
55-65%	Cukup	Tidak Berhasil
0-54%	Kurang	Tidak Berhasil



stikom
SURABAYA