

IMPLEMENTASI APLIKASI e-BOARDING HOUSE DI KOTA LHOKSEUMAWE MENGGUNAKAN ALGORITMA LEVENSHTTEIN DISTANCE

Agami Iqbal¹, Nurdin²

Teknik Informatika Universitas Malikussaleh Lhokseumawe
Jl. Cot Tgk Nie-Reulet, Aceh Utara, 141 Indonesia
email : nurdin_um@ymail.com

ABSTRAK

Seiring dengan kebutuhan masyarakat akan teknologi informasi sangatlah penting, semakin berkembangnya teknologi informasi, semakin besar pula kebutuhan atau ketergantungan masyarakat akan teknologi informasi tersebut. Maka dengan sewajarnya akan memaksa para masyarakat dalam memanfaatkan teknologi tersebut dengan semaksimal mungkin dari segala segi kehidupan. Informasi tentunya bermacam-macam, salah satunya adalah untuk para mahasiswa yang akan menjalani studi tour ataupun menjalani kuliah ditempat yang jauh dari tempat tinggal. Untuk itu jelas pada saat tersebut membuat mahasiswa tersebut harus segera mencari tempat tinggal sementara untuk dijadikan domisili mereka sampai studi selesai. Algoritma *Levenshtein Distance* adalah salah satu algoritma pencocokan string yang dapat digunakan untuk karakter/string. Dengan menggunakan algoritma *levenshtein distance* dapat memudahkan menentukan pencarian berdasarkan String secara efektif dan efisien . Munculnya *e-Boarding House* berbasis web sangat memudahkan pengguna dalam memperoleh informasi mengenai rumah sewa, diantaranya informasi lokasi dan jarak.

Kata Kunci : *e-Boarding House, Levenshtein Distance.*

1. PENDAHULUAN

1.1 Latar Belakang

Dewasa ini kebutuhan masyarakat akan kebutuhan teknologi informasi sangat penting, karena semakin berkembangnya teknologi informasi, semakin besar pula kebutuhan atau ketergantungan masyarakat akan teknologi informasi, maka dengan sewajarnya akan memaksa para

masyarakat dalam memanfaatkan teknologi tersebut dengan semaksimal mungkin dari segala segi kehidupan. Informasi tentunya bermacam-macam, salah satunya adalah untuk para mahasiswa yang akan menjalani studi tour ataupun menjalani kuliah ditempat yang jauh dari tempat tinggal. Untuk itu jelas pada saat tersebut membuat mahasiswa tersebut harus segera mencari tempat tinggal sementara untuk dijadikan domisili mereka sampai studi selesai.

Latar belakang tersebut yang memberikan inspirasi penulis membuat solusi untuk mempermudah para teman-teman mahasiswa untuk lebih mudah mencari tempat tinggal sementara, atau dengan kata lain Tempat Sewa / Kontrak. Untuk itu penulis mengambil judul "Implementasi Aplikasi *e-Boarding House* Menggunakan Metode *Levenshtein Distance*". Kebingungan, tidak tahu wilayah kota dan tidak memiliki teman, itulah alasan kenapa sistem ini dibuat yang tujuannya tidak lain adalah untuk mempermudah. Dan Karena penulis juga merupakan mahasiswa yang pernah merasakan kebingungan dalam mencari tempat sewa yang pas dan ideal dan dengan harga yang sesuai, yang sampai saat ini harus berkeliling wilayah untuk mencari lokasi tempat sewa yang sesuai. Maka dengan adanya sistem ini maka hal itu tidak perlu dilakukan, karena tempat sewa yang tergabung dengan sistem ini dapat langsung dilihat fasilitas berikut harga dari tempat sewa tersebut. Dan juga dengan adanya sistem ini dapat membantu pemilik sewa dalam mempromosikan rumah sewanya melalui media internet. Yang tujuannya agar bisa maksimal dalam tindakan promosi.

Sistem ini dikembangkan untuk mempermudah mahasiswa dalam mencari tempat sewa, melihat informasi-informasi serta lokasi alamatnya dengan system pencarian. Dengan begitu mahasiswa yang belum tahu tentang wilayah tempat studi mereka berada, dapat mengakses system ini. Dan jika yang bersangkutan tertarik dengan tempat sewa tersebut dapat segera memesannya dengan mengirimkan sms ke sistem dan selanjutnya dapat segera menghubungi pemilik sewa.

2.TINJAUAN PUSTAKA

2.1. Sistem Informasi

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen – komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem Informasi adalah seperangkat komponen yang saling berhubungan yang berfungsi mengumpulkan, memproses, menyimpan dan

mendistribusikan informasi untuk mendukung pembuatan keputusan dan pengawasan dalam organisasi (Eka Prasetya Adhy Sugara : 2011).

Sistem Informasi dapat digambarkan sebagai sistem yang terdiri dari berbagai komponen. Komponen ini dapat dianalogikan sebagai blok bangunan (*building block*) yang terdiri dari : Blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*), dan blok kendali (*control block*).

Ada beberapa hal yang termasuk dalam komponen Sistem Informasi diantaranya sebagai berikut :

- 1 Blok masukan yaitu : Input mewakili data yang masuk ke dalam sistem informasi.
- 2 Blok model yaitu : Blok ini terdiri dari kombinasi prosedur, Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.
- 3 Blok keluaran yaitu : Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
- 4 Blok teknologi yaitu : Teknologi merupakan “kotak alat” (tool box) dalam system informasi. Gunanya untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sitem secara keseluruhan.
- 5 Blok basis data yaitu : Basis data (database) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.
- 6 Blok kendali yaitu : Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.2 Rumah Sewa

Rumah sewa merupakan suatu tempat tinggal yang disewakan kepada pihak lain dengan fasilitas-fasilitas tertentu dengan harga yang lebih terjangkau daripada di hotel/penginapan. Rumah sewa lebih akrab digunakan sebagai domilisi, karena kebanyakan rumah sewa disewa dalam

jangka waktu yang cukup lama dari pada hotel atau penginapan yang menggunakan hitungan hari. Dan juga istilah rumah sewa sangatlah berdampingan dengan mahasiswa, karena pada umumnya rumah sewa disewakan untuk mahasiswa walaupun tidak jarang juga rumah sewa yang disewakan untuk umum. Rumah sewa sangatlah bermacam-macam, dari cara penyewaannya, fasilitas-fasilitas dan harga yang bervariasi. Dan rumah sewa ini adalah merupakan suatu investasi yang cukup menjanjikan yang dimana kita dapat menghitung biaya perbulan dengan yang dihasilkan disetiap bulannya.

Biasanya alasan memilih rumah sewa sebagai tempat tinggal adalah sebagai berikut :

1. Studi/kuliah dalam jangka pendek ataupun lama
2. Biaya yang terjangkau dibandingkan dari pada ke hotel/ penginapan
3. Rumah sewa sudah sangat banyak dan mudah dijangkau yaitu yang terdapat dilokasi kampus, perkantoran, sekolah, dll.
4. Cara penyewaan yang bervariasi, bergantung kebutuhan. Perhari, perbulan, per enam bulan atau pertahun.

Dari survey yang dilakukan, penulis menyimpulkan beberapa hal yang terkait daripada rumah sewa tersebut.

1. Periklanan

Dari hasil data survey yang dijadikan sample oleh penulis bahwa pada rumah sewa di Kota Lhokseumawe, para pemilik kos umumnya masih memasarkan rumah sewanya dengan manual. Seperti memasang tulisan 'Terima Kos' didepan rumah. Ini menjadikan yang mengetahui bahwa disitu rumah sewa adalah bagi yang kebetulan melewati daerah tersebut. Dan itu bisa dibilang adalah suatu kebetulan, dan ketika penulis bertanya terhadap beberapa mahasiswa tentang keberadaan rumah sewa. Mereka juga hanya menjawab sekitar daerah yang umum terdapat rumah sewa, yang tentunya telah banyak yang penuh.

2. Sistem Penyewaan

Penyewaan yang dilakukan oleh pemilik kost terhadap pencari kost umumnya dilakukan secara perbulan, tetapi ada juga yang menyewakan berdasarkan periode kuliah (persemester, pertahun). Proses penyewaan kamar kost langsung dilakukan secara langsung oleh pemilik dan penyewa, dengan kisaran harga antara Rp. 200.000 sampai Rp. 2.000.000,

tetapi mungkin ada juga yang melebihi dari harga tersebut dikawasan kost yang bisa dibilang lebih elite. Harga tersebut dapat dikelompokkan juga sesuai jenis biayanya yaitu pembayaran perorang atau perkamar.

3. Fasilitas yang didapat

Fasilitas yang didapat oleh penyewa kost sesuai dengan kondisi kost yang ada. Fasilitas standar yang telah didapat biasanya seperti tempat tidur, almari, meja dan listrik standar. Pada fasilitas lain yang bisa ditambahkan dan penyewa harus membayar tambahan itu seperti, membawa komputer, alat masak, kipas angin, dll. Terutama biasanya jika fasilitas itu membuat pemakaian listrik naik, maka itu termasuk biaya fasilitas tambahan.

4. Tipe Rumah Sewa

Jenis dan tipe rumah sewa yang penulis survei, umumnya berkelompok sesuai dengan jenis dan tipenya. Seperti khusus mahasiswa cewek/cowok, karyawan dan rumah tangga / umum. Tetapi ada juga yang rumah sewa tersebut merupakan perpaduan dari jenis-jenis tersebut, misalkan untuk mahasiswa dan karyawan atau terima kost untuk umum yang artinya semua jenis.

5. Area Universitas

Survey penulis merupakan rumah sewa yang terdapat didaerah universitas, maka pada umumnya para penyewa kost adalah para mahasiswa yang sedang studi pada universitas tersebut. Kemungkinan lain juga terdapat karyawan biasanya terjadi saat lokasi universitas tersebut dekat dengan suatu perkantoran atau market. Sehingga karyawan dari kantor/ market tersebut memilih untuk menyewa rumah sewa dilokasi sekitar tersebut.

2.3 String Matching

Pengertian string menurut Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology(NIST) adalah susunan dari karakter-karakter(angka,alfabet atau karakter yang lain) dan biasanya direpresentasikan sebagai struktur dan array. *String* dapat berupa kata, frase, atau kalimat.Pencocokan string (*string matching*) merupakan bagian penting dari sebuah proses pencarian *string(string searching)* dalam sebuah dokumen. Hasil dari sebuah pencarian *string* dalam dokumen tergantung dari teknik dan cara pencocokan *string* yang digunakan.

2.3.1 Algoritma Levenshtein Distance

Levenshtein Distance dibuat oleh Vladimir Levenshtein pada tahun 1965. Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara

dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B (Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia, (Richard Junedy.S : 2014)

Ada 3 macam operasi utama yang dapat dilakukan oleh perhitungan algoritma *Levenshtein Distance* ini yaitu:

1. Operasi Pengubahan Karakter Operasi pengubahan karakter merupakan operasi menukar sebuah karakter dengan karakter lain contohnya penulis menuliskan string “yang” menjadi “yng”. Dalam kasus ini karakter “m” diganti dengan huruf “n”.
2. Operasi Penambahan Karakter Operasi penambahan karakter berarti menambahkan karakter ke dalam suatu string. Contohnya string “kepad” menjadi string “kepada”, dilakukan penambahan karakter “a” di akhir string. Penambahan karakter tidak hanya dilakukan di akhir kata, namun bias ditambahkan diawal maupun disisipkan di tengah string .
3. Operasi Penghapusan Karakter Operasi penghapusan karakter dilakukan untuk menghilangkan karakter dari suatu string.

2.3.1.1 Implementasi Algoritma *Levenstein Distance*

Algoritma *Levenstein Distance* digunakan untuk mencari perbedaan dari dua buah string. Sebagai contoh membandingkan dua buah string yaitu : CS = abu dan ST = aku. Proses algoritma levenstein distance nya adalah sebagai berikut :

- a. Jika panjang CS adalah 0 maka jarak CS ke ST adalah panjang ST dalam contoh diatas berarti nilai 3.
- b. Jika panjang ST adalah 0 maka jarak CS ke ST adalah panjang CS dalam contoh diatas berarti bernilai 3.
- c. Membuat matrix dengan ukuran (CS+1) x (ST+1).
- d. Melakukan pencocokan dengan melakukan perbandingan dari setiap karakter CS dengan karakter ST.

Gambar 2.1 adalah proses dimana dari iterasi pertama yaitu iterasi (1,1) dimana merupakan iterasi untuk membandingkan karakter awal.

- a. Iterasi (1,1)

	0	a	b	u
		1	2	3
a	1	0		
k	2			
u	3			

Gambar 2.1. Iterasi (1,1)

(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.2, karakter 1 CS dengan karakter 1 ST sama, maka nilai pada cell(1,1) diisi dengan 0 yang merupakan nilai $(x-1,y-1)$.

b. Iterasi (1,2)

0		a	b	u
		1	2	3
a	1	0		
k	2	1		
u	3			

Gambar 2.2. Iterasi (1,2)
(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.3, karakter 1 CS dengan karakter 2 ST berbeda maka nilai cell di dapatkan dari nilai terkecil dari :

- Nilai cell $(x-1,y-1)+1$, adalah 2.
- Nilai cell $(x-1,y)+1$, adalah 3.
- Nilai cell $(x,y-1)+1$, adalah 1.

Maka jarak antara karakter 1 CS dan 2 ST bernilai 1.

c. Iterasi (1,3)

0		a	b	u
		1	2	3
a	1	0		
k	2	1		
u	3	2		

Gambar 2.3. Iterasi (1,3)

(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.4, karakter 1 CS dengan karakter 3 ST berbeda maka nilai cell di dapatkan dari nilai terkecil dari :

- nilai cell $(x-1,y-1)+1$, adalah 3.
- nilai cell $(x-1,y)+1$, adalah 4.
- nilai cell $(x,y-1)+1$, adalah 2.

Maka jarak antara karakter 1 CS dan 3 ST bernilai 2.

d. Iterasi (2,1)

0		a	b	u
		1	2	3
a	1	0	1	
k	2	1		
u	3	2		

Gambar 2.4. Iterasi (2,1)
(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.5, karakter 2 CS dengan karakter 1 ST berbeda maka nilai cell di dapatkan dari nilai terkecil dari :

- nilai cell $(x-1,y-1)+1$, adalah 2.
- nilai cell $(x-1,y)+1$, adalah 1.
- nilai cell $(x,y-1)+1$, adalah 3.

Maka jarak antara karakter 2 CS dan 1 ST bernilai 1.

e. Iterasi (2,2)

0		a	b	u
		1	2	3
A	1	0	1	
K	2	1	1	
U	3	2		

Gambar 2.5. Iterasi (2,2)
(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.6, karakter 2 CS dengan karakter 2 ST berbeda maka nilai cell di dapatkan dari nilai terkecil dari :

- nilai cell $(x-1,y-1)+1$, adalah 1.
- nilai cell $(x-1,y)+1$, adalah 2.
- nilai cell $(x,y-1)+1$, adalah 2.

Maka jarak antara karakter 2 CS dan 2 ST bernilai 1.

f. Iterasi (2,3)

0	a	b	u
a 1	0	1	
k 2	1	1	
u 3	2	2	

Gambar 2.6. Iterasi (2,3)
(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.7, karakter 2 CS dengan karakter 3 ST berbeda maka nilai cell di dapatkan dari nilai terkecil dari :

- nilai cell $(x-1,y-1)+1$, adalah 2.
- nilai cell $(x-1,y)+1$, adalah 3.
- nilai cell $(x,y-1)+1$, adalah 3.

Maka jarak antara karakter 2 CS dan 3 ST bernilai 2

g. Iterasi (3,1)

0	a	b	u
a 1	0	1	2
k 2	1	1	
u 3	2	2	

Gambar 2.7. Iterasi (3,1)
(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.8, karakter 3 CS dengan karakter 1 ST berbeda maka nilai cell di dapatkan dari nilai terkecil dari :

- nilai cell $(x-1,y-1)+1$, adalah 3.
- nilai cell $(x-1,y)+1$, adalah 2.
- nilai cell $(x,y-1)+1$, adalah 4.

Maka jarak antara karakter 3 CS dan 1 ST bernilai 2.

h. Iterasi (3,2)

0	a	b	u
	1	2	3
a 1	0	1	2
k 2	1	1	2
u 3	2	2	

Gambar 2.8. Iterasi (3,2)

(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.9, karakter 3 CS dengan karakter 2 ST berbeda maka nilai cell di dapatkan dari nilai terkecil dari :

- nilai cell $(x-1,y-1)+1$, adalah 2.
- nilai cell $(x-1,y)+1$, adalah 2.
- nilai cell $(x,y-1)+1$, adalah 3.

Maka jarak antara karakter 3 CS dan 1 ST bernilai 2.

i. Iterasi (3,3)

0	a	b	u
	1	2	3
a 1	0	1	2
k 2	1	1	2
u 3	2	2	1

Gambar 2.9. Iterasi (3,3)

(Sumber : Richard Junedy.S : 2014)

Pada gambar 2.10, karakter 3 CS dengan karakter 3 ST berbeda maka nilai cell di dapatkan dari nilai terkecil dari :

- nilai cell $(x-1,y-1)+1$, adalah 1.
- nilai cell $(x-1,y)+1$, adalah 3.
- nilai cell $(x,y-1)+1$, adalah 3.

Maka jarak antara karakter 3 CS dan 1 ST bernilai 1.

Setelah semua proses iterasi selesai maka, didapatkan jarak string asli (CS) dengan string pembanding (ST) pada cell (3,3) adalah sebesar 1.

0	a	b	U
	1	2	3
a 1	0	1	2
k 2	1	1	2
u 3	2	2	1

Gambar 2.10. Proses selesai
(Sumber : Richard Junedy.S : 2014)

2.4 Bahasa Pemrograman

2.4.1 PHP (Hypertext Preprocessor)

PHP (akronim dari PHP Hypertext Preprocessor) yang merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah server-side embedded script language artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di server.

PHP sebagai server-side embedded script language maka server akan melakukan hal-hal sebagai berikut :

1. Membaca permintaan dari client/browser
2. Mencari halaman/page di server
3. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page.
4. Mengirim kembali halaman tersebut kepada client melalui internet atau intranet.

2.5 Basis Data

2.5.1 MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut

tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

2.6 Entity Relation Diagram (ERD)

Entity Relation Diagram yang berisi komponen-komponen entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang merepresentasikan seluruh fakta dari 'dunia nyata' yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan Diagram *Entity Relationship* (Diagram E-R). (Sumber : Alfian Helmi Muhib : 2013)

Pada pembuatan *ERD* perlu diperhatikan penentuan entitas, atribut dan relationship.

1. Entitas (Entity)

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Sebuah kursi yang kita duduki, seseorang yang melintas di depan kita adalah entitas. Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah himpunan entitas (entity set). Sederhananya, entitas menunjuk pada individu suatu objek,

sedang himpunan entitas menunjuk pada rumpun (famili) dari individu tersebut. Seseorang memang dapat menjadi sebuah entitas, tapi dapat berada pada himpunan entitas yang berbeda dengan seseorang yang lain.

2. Atribut (Attributes)

Setiap entitas pasti memiliki atribut yang mendeskripsikan karakteristik dari Entitas tersebut. Penentuan/pemilihan atribut-atribut yang *relevan* bagi sebuah entitas merupakan hal penting lainnya dalam pembentukan model data. Penetapan atribut bagi sebuah entitas umumnya memang didasarkan pada fakta yang ada. Atribut terbagi kedalam beberapa model yaitu atribut deskriptif adalah atribut yang menjadi anggota dari key primer. Atribut sederhana adalah atribut atomic yang tidak dapat dipilah lagi. Atribut bernilai tunggal ditujukan pada atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data. Atribut bernilai banyak ditujukan pada atribut-atribut yang dapat kita isi dengan lebih dari 1 (satu) nilai, tetapi jenisnya sama.

3. Relasi (Relationship)

Relasi menunjukkan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Misalnya, entitas seorang mahasiswa dengan nim= '980001' dan nama_mhs= 'Ali Akbar' mempunyai relasi dengan entitas sebuah mata kuliah dengan kode_kul= 'IF-110' dan nama_kul= 'Struktur Data' tersebut sedang mengambil/mempelajari mata kuliah tersebut di sebuah perguruan tinggi yang kita tinjau. Kumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas-himpunan tersebut membentuk himpunan relasi.

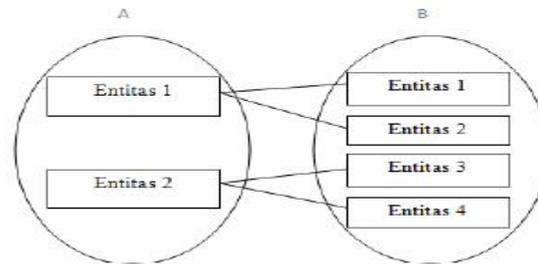
Berikut adalah contoh penggambaran relasi antar himpunan entitas lengkap dengan kardinalitas relasi dan atribut-atributnya:

1. Relasi satu-ke-satu (*one-to-one*)

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan satu entitas pada himpunan entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B dapat berhubungan dengan satu entitas pada himpunan entitas B.

2. Relasi satu-ke-banyak (*one-to-many*)

Pada himpunan entitas A berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A



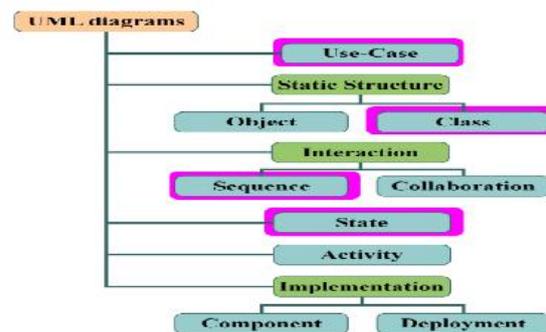
Gambar 2.12 Contoh relasi *one-to-many*
(Sumber : Alfian Helmi Muhibb : 2013)

3. Relasi banyak-ke-banyak (*many-to-many*)

Yang berarti setiap entitas pada himpunan entitas A berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

2.7 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, et. al. 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, et. al. 2004 dalam jurnal Haviluddin, 2011). UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.



Gambar 2.14 Diagram *Unified Modeling Language*
(Sumber :Haviluddin, 2011)

Berikut tujuan utama dalam desain UML adalah (Sugrue J. 2009 dalam jurnal *Haviluddin, 2011*) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

2.7.1 Komponen-komponen UML

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik.

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu OOD (*Object-Oriented Design*), OMT (*Object Modelling Technique*), dan OOSE (*Object-Oriented Software Engineering*) (*Haviluddin, 2011*).

3. METODE PENELITIAN

penelitian meliputi dari studi keperpustakaan pengumpulan referensi tentang algoritma *Levenshtein Distance* , teori pendukung serta data keperluan lainnya meliputi kebutuhan hardware dan software yang digunakan serta data objek penelitian, perancangan diagram sistem aplikasi *e-boarding house* menggunakan algoritma *Levenshtein Distance*.

3.1 LANGKAH - LANGKAH PEMBUATAN APLIKASI

3.1.1 Teknik Pengumpulan Data

1. Studi Kepustakaan
Melakukan studi pustaka dan literatur menggunakan buku-buku yang mendukung proses penelitian dan mencari informasi menggunakan sumber-sumber lain yang terpercaya.
2. Observasi

Melakukan observasi terhadap website yang sudah menggunakan teknologi Kecerdasan Buatan.

3. Pengumpulan Data
Mengumpulkan data berupa data sample data user dan data sample video.
4. Pembangunan Sistem
Pengembangan sistem dimulai setelah desain website dan database disetujui dosen pembimbing.
5. Analisis dan Pengujian
Analisis dan pengujian dilakukan untuk menguji apakah sistem sudah berjalan dengan benar atau tidak

3.2 ANALISIS KEBUTUHAN SISTEM

Tahap metode analisis ini merupakan tahap menganalisa sistem yang akan di bangun. Setelah analisa didapatkan maka langkah selanjutnya adalah membuat sebuah hasil analisa. Hasil analisa tersebut akan menjadi acuan dari perancangan sistem yang di bangun.

3.2.2 Analisis Kebutuhan Input

Di dalam sistem yang akan dibangun memiliki kebutuhan data yang diperlukan, yaitu berupa data rumah sewa Se-Kecamatan Kota Lhokseumawe yang diperoleh dari hasil survey lapangan. Data-data tersebut diperlukan untuk dijadikan sumber informasi nantinya.

3.2.3 Analisis Kebutuhan Proses

Pada tahapan analisa proses ini terdapat beberapa proses, yaitu

1. Proses Inputan data ke basis data
2. Proses kinerja web pada browser

3.2.4 Analisis Kebutuhan Output

Di dalam sistem yang akan dibangun menghasilkan sumber informasi mengenai seputaran rumah sewa secara efektif dan efisien.

3.3 SKEMA SISTEM

Adapun langkah-langkah prosesnya seperti berikut:

1. Diawali dengan penginputan katagori atau criteria yang ingin dicari
2. Kemudian Sistem Aplikasi memproses tahap selanjutnya apakah kriteria/Kategori yang dicari ada ataupun tidak.

3. Jika ada maka hasil langsung ditampilkan pada tampilan aplikasi, jika tidak maka aplikasi menggunakan proses algoritma *Levensthein Distance* untuk mencocokkan Kategori yang ada di database dengan yang diinputkan. Sehingga jika ditemukan kesamaan maka akan ditampilkan jika tidak maka akan dilakukan penginputan ulang.

4. HASIL DAN PEMBAHASAN

4.1 ANALISA SISTEM

4.1.1 Analisa Sistem Pencarian

Dalam Penelitian ini, Penulis merancang sistem pengkoreksian kata kunci dengan mengaplikasikan metode *String Matching* algoritma Levenshtein Distance. Sistem ini bertujuan membantu pengguna dalam mengatasi permasalahan yang sering timbul pada saat memanfaatkan fasilitas pencarian pada website yakni hasil pencarian yang tidak optimal dikarenakan terjadi kesalahan penginputan kata kunci. Apabila hal itu terjadi maka sistem diharapkan dapat melakukan pengkoreksian untuk lalu memberikan saran kata kunci yang dapat dipakai untuk melakukan proses pencarian berikutnya. Saran kata kunci diperoleh dengan cara membandingkan kata kunci yang dianggap salah dengan kata-kata terindeks yang berasal dari website. Hasil perbandingan kata yang memiliki nilai jarak terkecil akan terpilih sebagai saran kata kunci.

4.1.2 Analisa Kebutuhan Sistem

Berdasarkan kebutuhan yang akan diterapkan pada sistem ini, terdapat dua jenis kebutuhan yaitu kebutuhan sebagai *admin* dan *user*. Yang masing-masing kebutuhan tersebut memiliki peran penting dalam implementasi sistem *e-Boarding House*.

1. Kebutuhan Admin
 - a. Input Data
 - b. Input admin Baru
 - c. Pencarian *String*
 - d. Lihat hasil koreksi dokumen
2. Kebutuhan *User*
 - a. Pencarian *String*
 - b. Lihat hasil koreksi dokumen

4.2 PERANCANGAN SISTEM

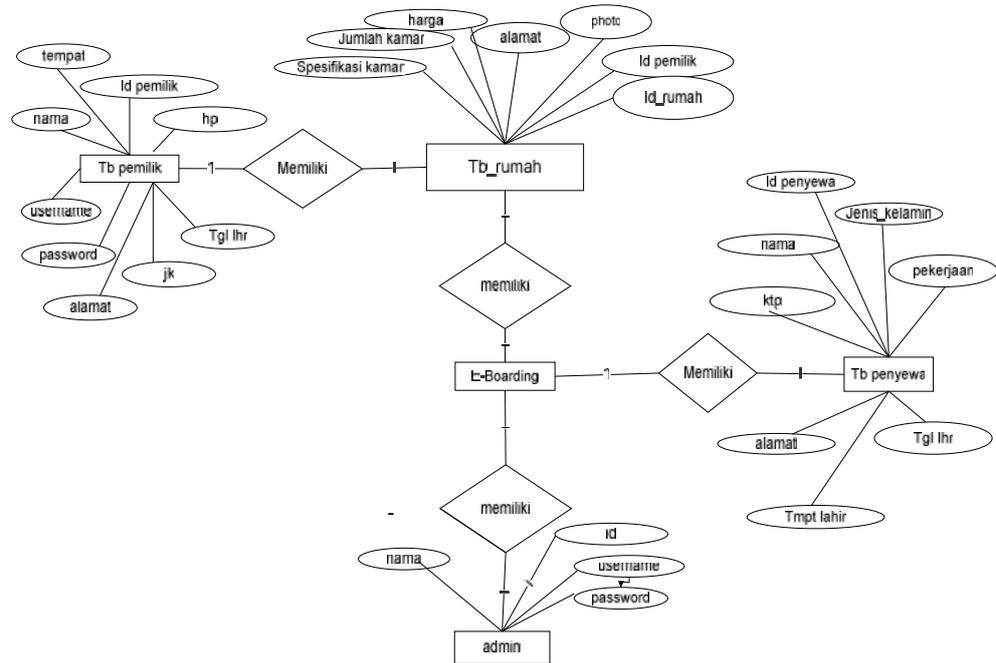
4.2.1 Perancangan Sistem E-Boarding House

Perancangan sistem E-Boarding House terdiri dari beberapa tahapan yaitu tahap perancangan ERD (*Entity Relationship Diagram*), Tabel- tabel yang dibutuhkan dalam proses pembuatan sistem serta UML (*Unified*

Modeling Language) yang terdiri dari use case diagram, activity, diagram komponen serta perancangan user interface.

4.2.2. Perancangan Entity Relation Diagram (ERD)

ERD berfungsi untuk menggambarkan hubungan antar entitas yang satu dengan entitas yang lain. Di sistem E-Boarding House ini memiliki beberapa entitas yang entitas tersebut memiliki kaitan dari satu entitas dengan entitas lainnya. Adapun hubungan antar entitas yang terdapat di dalam sistem E-Boarding House ini dapat di gambarkan seperti gambar dibawah berikut:



Gambar 4.1 Entity Relationship Diagram E-Boarding House

ERD Diatas berfungsi untuk menggambarkan hubungan antar entitas yang satu dengan entitas yang lain. ERD pada gambar 4.1 diatas memiliki lima buah entitas, yaitu entitas E-Boarding House, Admin, Penyewa, Pemilik Rumah Sewa dan Rumah Sewa. diantara entitas-entitas tersebut terdapat relasi antar tabelnya, yang berfungsi untuk menghubungkan antara tabel yang satu dengan tabel yang lainnya. Atribut pada entitas berfungsi untuk menjelaskan karakteristik dari entitas tersebut.

4.3. Perancangan Tabel

Tabel-Tabel yang digunakan pada perancangan sistem E-Boarding House terdiri dari 4 table diantaranya Tabel Admin, Tabel Penyewa, Tabel Pemilik Rumah Sewa dan Tabel Rumah Sewa.

4.3.1. Tabel Pemilik Rumah Sewa

Tabel pemilik rumah sewa berisi tentang data pribadi si pemilik rumah sewa yang akan nantinya di promosikan lewat sistem E-Boarding House. Berikut adalah struktur tabel Pemilik sewa.

Tabel 4.1 Tabel Pemilik Rumah Sewa

No	Field	Type	Length	Keterangan
1.	Id	Int	11	Primary key
2.	Nama	Varchar	25	
3.	Tempat	Varchar	25	
4.	Tanggal Lahir	Date	25	
5.	Jenis Kelamin	Varchar	10	
6.	Alamat	Varchar	100	
7.	Hp	Varchar	25	
8.	Username	Varchar	25	
9.	Password	Varchar	25	

4.3.2 Tabel Penyewa Rumah Sewa

Tabel data penyewa rumah sewa berisi tentang data seseorang yang ingin menyewa rumah lewat sistem E-Boarding House. Berikut ini adalah struktur tabel penyewa rumah sewa.

Tabel 4.2. Tabel Penyewa Rumah Sewa

No	Field	Type	Length	Keterangan
1.	ID Penyewa	Int	11	Primary key
2.	Nama	Varchar	25	
3.	Tempat lahir	Varchar	25	
4.	Tanggal Lahir	Varchar	25	
5.	Alamat	Varchar	5	
6	Jenis Kelamin	Varchar	10	
7	HP	Varchar	25	
8	Pekerjaan	Varchar	25	

4.3.3 Tabel Rumah Sewa

Tabel ini digunakan untuk mengisi data spesifikasi rumah sewa. Table ini disikan oleh si pemliki rumah sewa yang diberikan hak akses oleh admin Berikut adalah struktur tabel rumah sewa.

Table 4.3 Tabel Rumah Sewa

No	Field	Type	Length	Keterangan
1.	Id_rumah	Int	25	Primary key
2.	Id Pemilik	Varchar	25	
3.	Alamat	Varchar	50	
4.	Harga	Varchar	25	
5.	Jumlah Kamar	Int	25	
6.	Spesifikasi Kamar	Varchar	200	
7	Photo	VArchar	200	

4.3.4 Tabel Admin

Tabel admin berisi tentang data admin digunakan untuk menyimpan data yang mengatur akses E-Boarding House. Berikut adalah struktur tabel admin.

Tabel 4.4 Tabel Admin

No	Field	Type	Length	Keterangan
1.	Id_Admin	Int	15	Primary key
2.	Username	Varchar	30	
3.	Password	Varchar	30	
4.	Nama	Varchar	50	

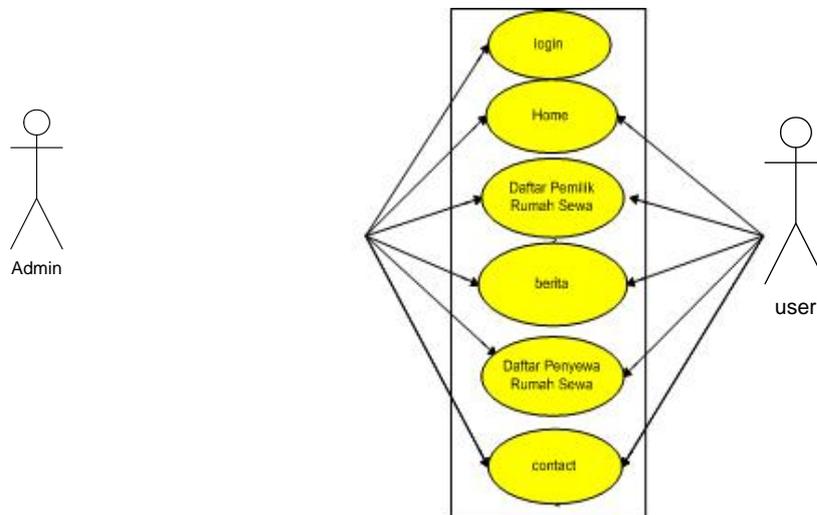
4.4 Perancangan UML(Unified Modelling Language).

Sistem E-Boarding ini menggunakan UML (*Unified Modelling Language*). Perancangan sistem akan digambarkan ke dalam beberapa bentuk diagram sebagai berikut.

4.4.1 Perancangan Use Case Diagram

Use-case diagram adalah gambaran *graphical* dari beberapa atau semua *actor*, *use-case*, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun dalam perancangan. *Use case* utama berfungsi untuk memberikan gambaran

umum mengenai sistem yang telah dibangun. *Use case* utama dari sistem *e-Boarding House* dapat dilihat pada gambar 4.2 berikut ini:



Gambar 4.2 *Use case Diagram* sistem E-Boarding House

Use case Diagram sistem E-Boarding House pada gambar 4.2 menjelaskan tentang aktivitas-aktivitas user dalam menggunakan sistem. Berdasarkan gambar use case diagram diatas, admin level super dapat melakukan pemilihan menu utama seperti melakukan login sebagai admin, maka dapat mengakses Home, Daftar Pemilik Rumah Sewa, Berita, Daftar Penyewa, Contact, Logout sedangkan *User* hanya bisa melihat Halaman Utama, Daftar Pemilik Rumah Sewa, Berita, Daftar Penyewa dan Contact tanpa memiliki hak akses.

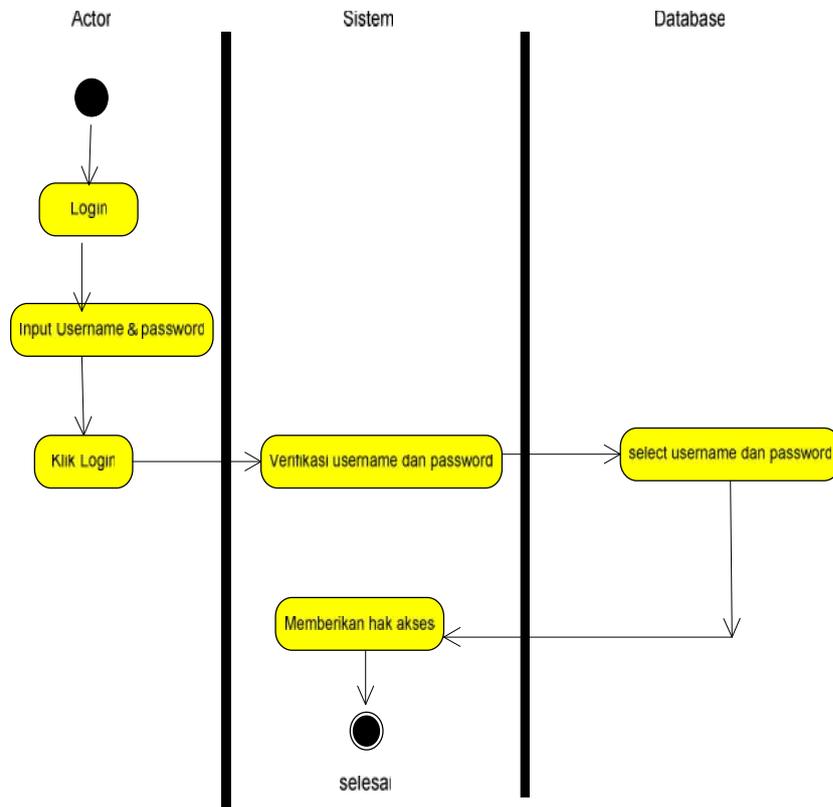
4.4.2. *Activity Diagram*

Activity diagram di gunakan untuk memodelkan *workflow* atau jalur kerja, memodelkan operasi , bagaimana object-object bekerja, aksi-aksi dan pengaruh terhadap objek-objek bekerja.

1. Diagram Activity menu login

Use case Diagram sistem E-Boarding House selanjutnya menjelaskan tentang aktivitas-aktivitas user dalam menggunakan sistem E-Boarding House ini. Dalam urutan aktifitas kerja dan proses dari diagram *login* dimulai dari *state* mulai yang awalnya harus mengisi username dan password. Setelah mengisi username dan password kemudian klik sehingga memunculkan imbauan bahwa anda telah berhasil login. Dalam memverifikasi password dan username yang diinputkan sedikit memakan waktu. Kemudian apabila berhasil login maka sistem akan menampilkan

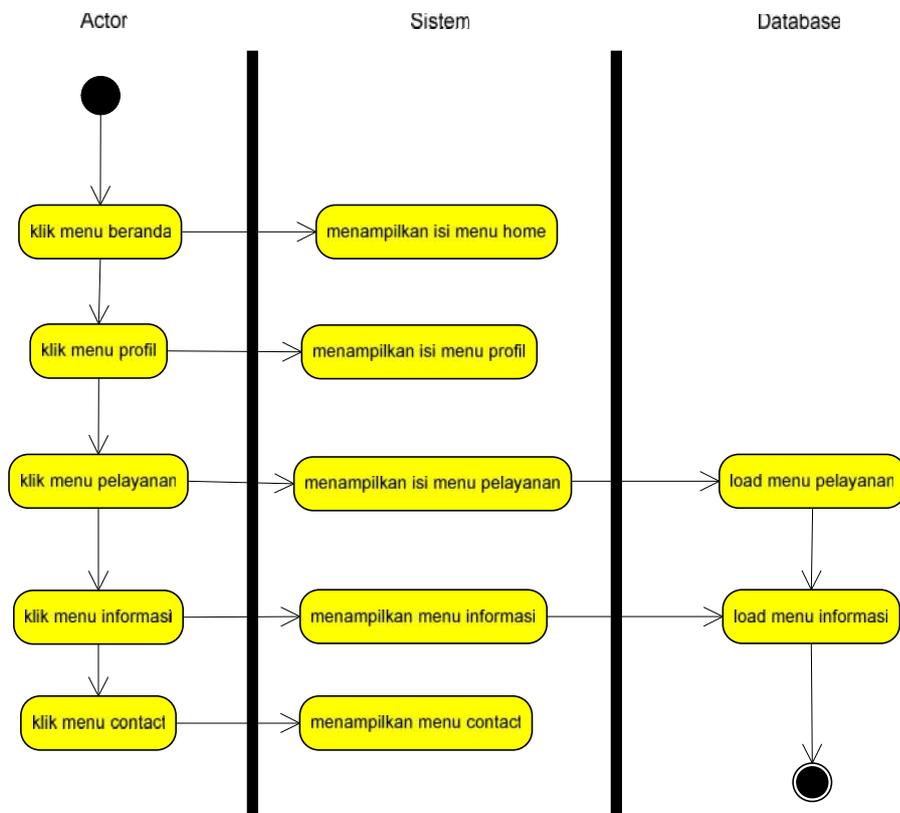
balasan dan memberikan hak akses ke user untuk login sampai dengan *state* selesai berdasarkan arah panah. Untuk lebih jelasnya diagram *activity* menu login dibawah ini dapat dilihat dengan seksama pada gambar 4.3 dibawah ini :



Gambar 4.3 Diagram Activity Menu Login

2. Diagram Activity Halaman Utama

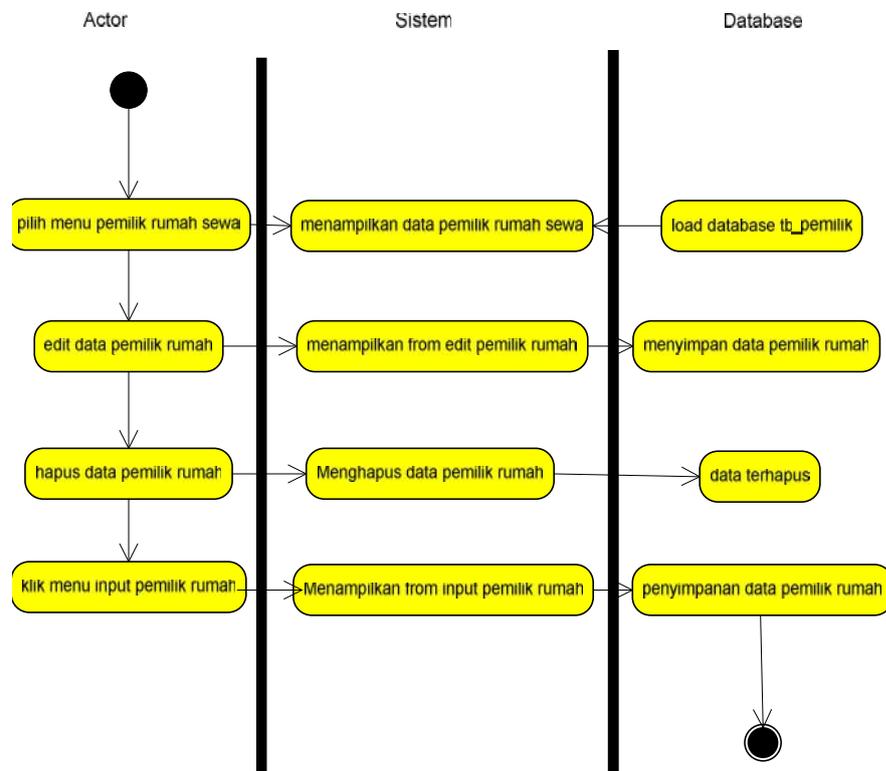
Diagram *activity* menu utama berfungsi untuk menjelaskan aktivitas admin dan user pada tampilan utama sistem yang telah dibangun. Pada gambar 4.4 berikut ini merupakan diagram *activity* Halaman Utama.



Gambar 4.4 Diagram Activity Halaman Utama

2. Diagram Activity Menu Pemilik Rumah Sewa

Diagram activity menu pemilik rumah sewa menggambarkan aktifitas user pada saat diberikan hak akses ke menu pemilik rumah sewa. Admin memberikan hak akses untuk memanipulasi menu pemilik rumah sewa. Diagram activity menu pemilik rumah sewa dapat dilihat pada gambar 4.5 berikut ini:



Gambar 4.5 Diagram Activity Menu Pemilik Rumah Sewa

4.5. IMPLEMENTASI SISTEM

4.5.1 Tampilan Sistem Login

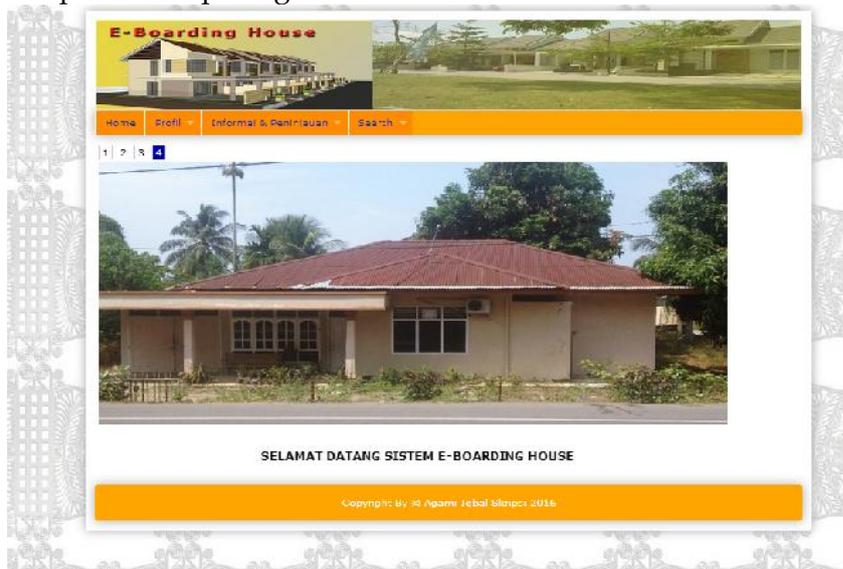
Implementasi sistem merupakan tahap akhir dari pengembangan perangkat lunak. Berikut ini akan ditampilkan halaman login yaitu login Administrator dari sistem yang telah dibangun. Tampilan form login pada gambar 4.8 berikut ini merupakan menu login.



Gambar 4.8 Tampilan Halaman Login

4.5.2 Tampilan Halaman Utama

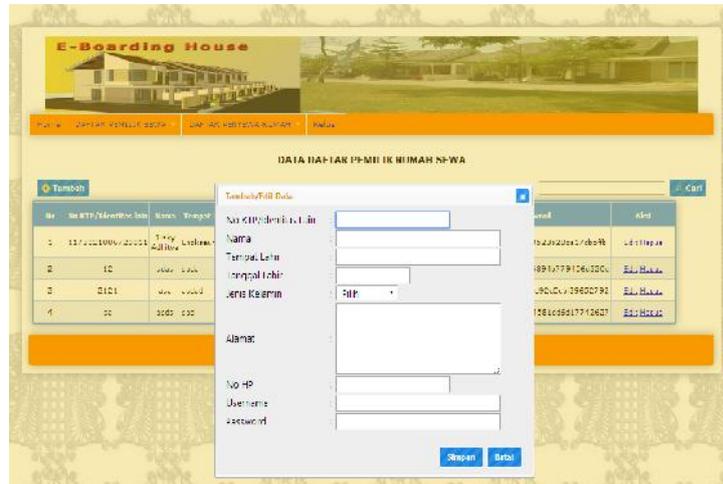
Pada saat admin memilih Halaman Utama maka yang akan ditampilkan pada Halaman Utama berupa data profil Sistem, Informasi dan Pelayanan. Tampilan Halaman Utama ini merupakan tampilan awal pada sistem *E-Boarding House*. Tampilan di desain bersifat user Friendly agar user dapat dengan mudah menggunakan sistem *E-Boarding House* ini. dapat dilihat pada gambar 4.9 berikut ini :



Gambar 4.9 Tampilan Halaman Utama

4.5.3 Tampilan Menu Data Pemilik Sewa

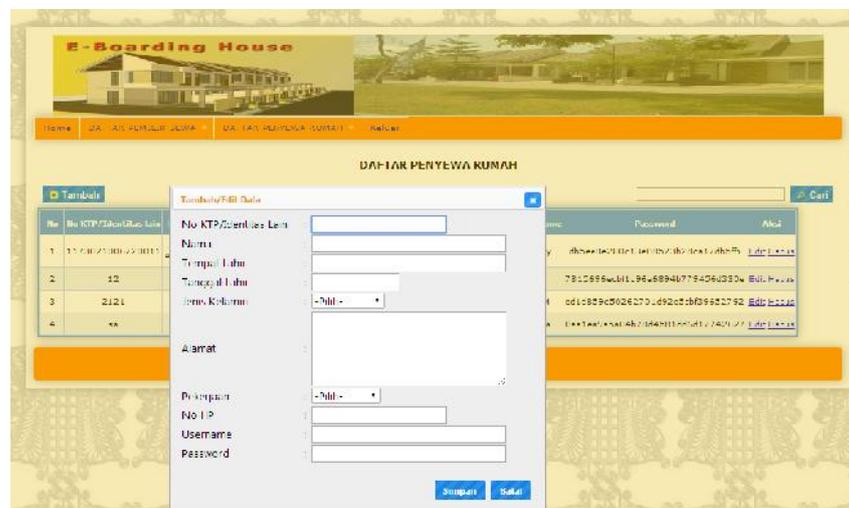
Tampilan Menu pemilik sewa akan muncul pada saat admin memilih menu data pemilik sewa. Untuk menampilkan Inputan data Pemilik sewa, maka admin harus memilih terlebih dahulu kategori tahap pemilik sewa dari menu daftar milik sewa yang disediakan, karena pada tampilan data Pemilik sewa ada beberapa tahap yang akan ditampilkan sesuai dengan tahap yang dipilih. Untuk Menu Pemilik sewa maka user harus memilih menu daftar pemilik sewa dan kemudian pilih sub menu pemilik rumah sewa. Setelah memilih sub menu register maka akan tampil form data Pemilik rumah sewa. Kemudian Klik Button Biru di sudut kiri maka form inputan identitas Pemilik sewa pun akan muncul. Untuk lebih jelas tampilan data Pemilik sewa dapat dilihat pada gambar 4.10 Dibawah ini.



Gambar 4.10 Tampilan Menu Data Pemilik sewa

4.5.4 Tampilan Form Input Data Penyewa Rumah

Form input penyewa rumah donor hanya dapat di akses oleh admin. *Form* ini di pergunakan oleh admin untuk melakukan input data mengenai seseorang yang ingin melakukan penyewaan rumah yang terdiri dari inputan biodata pribadi. Tampilan menu penyewa rumah dapat dilihat pada gambar 4.11 dibawah ini :



Gambar 4.11 Tampilan Form Input Penyewa Rumah

Pada gambar 4.11 diatas merupakan hasil tampilan halaman penyewa rumah, di menu ini langsung diisikan oleh admin yang data sesuai dengan data pribadi si penyewa rumah. Data terdiri dari no identitas, nama, tempat lahir, tanggal lahir, jenis kelamin, alamat, pekerjaan, dan no hp. Setelah selesai menginput maka langsung tekan button simpan, agar data segera tersimpan di database.

4.5.5 Tampilan Menu Rumah Sewa

Tampilan menu informasi Pemilik sewa bisa ditampilkan dengan cara memilih menu informasi & pelayanan. Setelah itu maka akan muncul 4 sub menu salah satunya adalah sub menu informasi rumah sewa. Didalam menu rumah sewa terdapat ditampilkan berbagai rumah sewa yang disediakan. Data yang ditampilkan mendeskripsikan sekilas mengenai rumah yang disewakan. Adapun tampilan informasi mendonor dapat dilihat pada gambar 4.12 dibawah ini.

The screenshot shows a web application interface for 'E-Boarding House'. At the top, there are two images of houses. Below them is a form titled 'ENTRY DATA RUMAH SEWA'. The form contains the following fields and values:

- ID Bangunan Rumah: Dn001
- ID_Pemilik: 112830
- Alamat: 3Ln. Medan-Benda Aceh. aive awc no. 1
- Harga: >1500.000 | Pertahun
- Jumlah Kamar: 4
- Alamat: memiliki 2 kamar mandi, 1 ruang tamu, semcer dan kecur.
- HP: 082378161711
- Photo: Choose File main.png

At the bottom of the form, there are buttons for 'OK', 'Ulangi', and 'Batal', and a link for 'Laporan'. A 'Kembali' button is located at the bottom left of the form area.

Gambar 4.12 Tampilan Menu Rumah Sewa

Informasi yang ditampilkan adalah informasi seputar spesifikasi rumah sewa dan harga sebelum melakukan transaksi penyewaan rumah antara si penyewa dengan si pemilik rumah sewa.

4.5.6 Implementasi Algoritma Levenshtein Distance

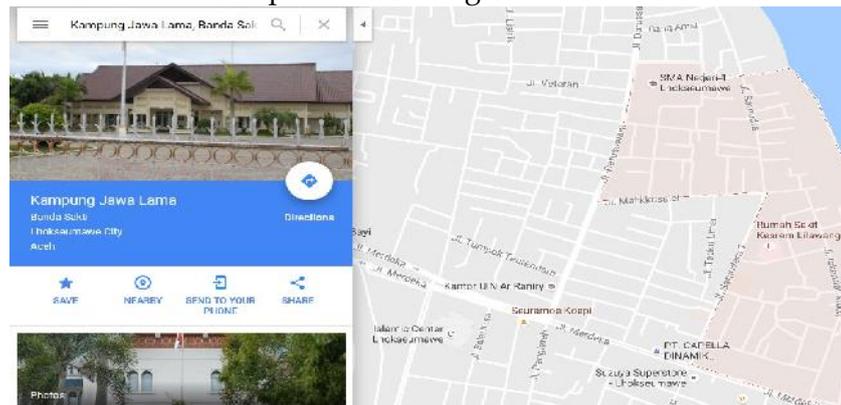
Tahapan ini adalah penerapan algoritma Levenshtein Distance yang digunakan untuk mencari string terdekat dari yang di inputkan melalui menu pencarian. menu ini diperuntukkan untuk user yang ingin

mencari. Berikut merupakan tampilan implementasi algoritma *levenshtein distance* dapat dilihat pada gambar 4.13 berikut ini:



	Lokasi Rumah Sewa	Link Foto	Detail Rumah
1	Jl. Merdeka No. 70, Mon Seudong	Link Foto	Detail Rumah
2	Il. Merdeka, Mon Seudong	Link Foto	Detail Rumah
3	Il. Inang, Inang II, Inapak, Inang, Banda Sakti	Link Foto	Detail Rumah
4	Pusat Baru Lt. IV, Banda Sakti	Link Foto	Detail Rumah
5	Jln. Imam Lirong Melat No. 3, Jawa Baru	Link Foto	Detail Rumah
6	Cp. Cakra No. 10, Banda Maseh, Banda Gakti, Kota Lhokseumawe	Link Foto	Detail Rumah
7	Il. Dalam Putih, Hagu Karat Iaur, Banda Sakti	Link Foto	Detail Rumah
8	Il. Kenari Lirong H. H. Anon Nela	Link Foto	Detail Rumah
9	Jln. Merdeka Putih Lirong 2 No. 30 Hagu Inangkuh	Link Foto	Detail Rumah

Gambar 4.13 Implementasi Program *Levenshtein Distance*



Gambar 4.14 Hasil Tampilan Penelusuran Jarak Rumah Sewa

5. KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Beberapa hal yang dapat disimpulkan dari uraian bab-bab sebelumnya antara lain sebagai berikut :

1. Dengan dibangun sistem E-Boarding House dapat membantu user dalam proses pencarian rumah sewa secara lebih mudah dan akurat.

2. Dalam pengujian fungsional sistem dapat berjalan dengan baik dan hasil yang didapat sesuai dengan proses hasil penelitian.
3. Algoritma Levensthein Distance berhasil diterapkan di pencarian *E-Boarding House*

5.2 SARAN

Beberapa saran dalam pengembangan sistem E-Boarding House :

1. Perlunya penambahan menu transaksi online, agar sistem ini lebih lengkap dan lebih fleksibel dalam proses booking atau pembayarannya.
2. Perlunya penambahan menu pemetaan secara online (GIS) agar lebih memudahkan lagi dalam proses pencariannya.
3. Diharapkan peneliti terkait berikutnya agar dapat menyesuaikan kondisi aplikasi ini dengan kebutuhan di masyarakat.

DAFTAR PUSTAKA

- Alfian Helmi Muhbib. (2013). *Implementasi Desktop Sistem Inventasi Pada Hudi Motor Karangrayung Grobogan*. Jurnal Program Studi Teknik Informatika Universitas Dian Nuswantoro
- Eka Prasetya Adhy Sugara. (2011). *Sistem Informasi Pencarian dan Penjualan Barang BerbasisWeb Pada Toko Bagus*. Jurnal Teknologi Dan Informatika Teknomatika Vol. 1 No. 2 Mei 2011
- Efori Buulolo, (2013), *Implementasi Algoritma String Matching Dalam Pencarian Surat Dan Ayat Dalam Bible Berbasis Android*. Volume III, Maret 2013.
- Haviluddin. (2011). *“Memahami Penggunaan UML” (Unified Modelling Language)*. Jurnal Informatika Mulawarman Vol 6 No. 1 Februari 2011
- Richard Junedy.S. (2014). *Perancangan Aplikasi Deteksi Kemiripan Isi Dokumen Teks Dengan Menggunakan Metode Levenshtein Distance*. Jurnal. Pelita Informatika Budi Darma, Volume : VII, Nomor: 2, Agustus 2014