**PROPOSAL**

**IMPLEMENTASI *STRING MATCHING* ALGORITMA RAITA**

**UNTUK PENCARIAN DATA DALAM PEMBUATAN APLIKASI MANAJEMEN ARSIP PENGALAMAN BIMBINGAN LAPANGAN (PBL)**

**(Study Kasus : Fakultas Kesehatan Masyarakat Universitas Halu Oleo).**

**Diajukan untuk memenuhi salah satu syarat memperoleh derajat Sarjana Teknik**



**OLEH :**

**R.M.ALFIAN APRIANSYA DIPONEGORO**

**E1E1 16 073**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HALU OLEO**

**KENDARI**

**2020**

# BAB I

# PENDAHULUAN

# 1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat di era globalisasi saat ini tidak bisa dihindari pengaruhnya terhadap dunia pendidikan. Tuntutan global menuntut dunia pendidikan untuk selalu dan senantiasa menyesuaikan perkembangan teknologi terhadap usaha dalam peningkatan mutu pendidikan, terutama penyesuaian penggunaannya teknologi informasi bagi instansi Pendidikan

Kegiatan instansi juga memerlukan peran teknologi informasi agar mampu mengelola data dan informasi yang baik serta demi tercapainya peningkatkan manajemen sumber daya sebuah instansi dan juga manajemen informasi yang lebih terpercaya. Ini pula berkaitan dengan kegiatan instansi di sebuah Lembaga Pendidikan.

Di fakultas kesehatan masyarakat Universitas Halu Oleo terdapat matakuliah wajib yaitu laporan Pengalaman Belajar Lapangan (PBL) yang berfokus pada kemampuan mahasiswa dibidang kesehatan masyarakat. Matakuliah PBL yang telah ditempuh oleh mahasiswa kemudian dibuatkan sebuah laporan sebagai arsip di perpustakaan sebagai data awal untuk penelitian tugas akhir mahasiswa. Namun pencarian arsip PBL masih dilakukan secara manual sehingga menyulitkan mahasiswa dan dosen dalam proses pencarian arsip karena membutuhkan waktu yang cukup lama untuk mendapatkan data PBL.

Permasalahan *string matching* adalah untuk mencari beberapa karakter (*pattern*) dalam sejumlah besar teks (Hartoyo, Vembrina, & Meilana, 2011). Algoritma Raita merupakan bagian dari Algoritma *exact string matching* yaitu pencocokan string secara tepat dengan susunan karakter dalam string yang dicocokan memiliki jumlah maupun urutan karakter dalam string yang sama.

Algoritma Raita merancang sebuah Algoritma dengan membandingkan

karakter yang terakhir dari pola yang diawali dari karakter paling kanan

dari“jendela”.Jika mereka cocok, kemudian karakter pertama dari pola teks paling kiri dari jendela juga dibandingkan. Jika mereka cocok, maka akan dibandingkan karakter tengah pola dengan karakter teks tengah jendela. Pada akhirnya, jika mereka benar-benar cocok, maka Algoritma membandingkan karakter lain mulai dari pola karakter kedua ke karakter kedua terakhir, dan akan membandingkan dengan karakter tengah lagi (Charras & Lecroq, 2004).

Berdasarkan permasalahan diatas, maka peneliti tertarik melakukan penelitian berjudul “Implementasi *String Matching* Algoritma Raita Untuk Pencarian Data Dalam Pembuatan Aplikasi Manajemen Arsip PBL (Praktek Bimbingan Lapangan) (study kasus : Fakultas Kesehatan Masyarakat Universitas

Halu Oleo)”.

# 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang diuraikan sebelumnya maka rumusan masalah penelitian ini adalah:

1. Bagaimana membangun Aplikasi Manajemen Arsip PBL di Fakultas Kesehatan Masyarakat Universitas Halu Oleo.
2. Bagaimana mengimplementasikan Algoritma Raita dalam pencarian data di Aplikasi Manajemen Arsip PBL.

# 1.3 Batasan Masalah

Berdasarkan latar rumusan masalah yang diuraikan diatas maka batasan masalah dalam tugas akhir ini adalah :

1. Penelitian ini dilakukan bertempat di Fakultas Kesehatan Masyarakat Universitas Halu Oleo.
2. Pembuatan aplikasi pencarian data berkas PBL menggunakan Algoritma

Raita.

1. Pencarian ini hanya dapat mencari satu kalimat dalam satu kali pencarian.
2. Pengambilan data berkas PBL diambil dari tahun 2012-2017.
3. Aplikasi ini menggunakan DBMS *MySQL* untuk penyimpanan Arsip PBL.
4. Aplikasi ini menggunakan *Framework CodeIgniter* sebagai *tools* pembuatan Arsip PBL .

# 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah membangun sebuah aplikasi yang dapat membantu dalam mengelola arsip Pengalaman Belajar Lapangan (PBL) di Fakultas Kesehatan Masyarakat Universitas Halu Oleo menggunakan Algoritma Raita.

# 1.5 Manfaat Penelitian

Adapun manfaat penelitian yakni agar dapat membantu dan mempermudah dosen dan mahasiswa dalam memenmukan dan mengelola Arsip Pengalaman Belajar Lapangan (PBL) dengan menggunakan sistem terkomputerisasi.

# 1.6 Sistematika Penulisan

Untuk mempermudah dan memperjelas pembahasan, maka tugas akhir ini disusun dalam sistematika sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan laporan dan tinjau penelitian sejenis.

BAB II LANDASAN TEORI

Bab ini berisi tentang pembahasan mengenai teori-teori yang berhubungan dengan penelitian yang akan dilakukan meliputi penjelasan tentang Pengalaman Belajar Lapangan (PBL), Manajemen Sistem Informasi, Arsip, Pengertian Algoritma, *String Matching,* Algortima Raita, *Framework* dan hal-hal lain yang berhubungan dengan pembuatan sistem.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang metode penelitian yang digunakan dan analisis perancangan sistem.

# 1.7 Tinjau Pustaka

Penelitian ini dibuat berdasarkan penelitian sebelumnya yang pernah dilakukan untuk menjadi acuan dan referensi dalam penyusunan tugas akhir . Penulis mengambil jurnal dan penelitian antara lain sebagai berikut :

**Mamta Culkari Puding**, ditahun 2019. Dengan judul “Perbandingan Algoritma Horspool Dan Algoritma Raita Pada Aplikasi Istilah Psikologi Berbasis Android”. Berdasarkan pengujian pencarian dengan menggunakan 3 kata pada masing-masing istilah diperoleh hasil rata-rata waktu untuk Algoritma Horspool adalah 11,95 ms, sedangkan untuk Algoritma Raita adalah 11,46 ms. Berdasarkan hasil tersebut pada pengujian pencarian dengan meggunakan 3 kata, Algoritma Raita lebih cepat dibandingkan Algoritma Horspool dengan selisih waktu kecepatan

0,56 ms.

Kemudia**n Anisya Sonita**, pada tahun 2018. Dengan Judul “Implementasi Algoritma Sequential Searching Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik”. Pertama Aplikasi Arsip Elektronik dapat di jalankan dengan baik. Dari hasil pengujian dari berbagai aspek aplikasi telah sesuai dengan perencanaan dan tujuan penelitian. Kedua Algoritma Sequential Searching dapat di terapkan pada pencarian arsip berdasarkan nomor arsip dan berjalan dengan baik sesuai perencanaan. Ketiga Hampir semua permasalahan pada sistem arsip manual dapat teratasi dengan adanya sistem arsip elektronik ini. Dari segi penyimpanan, pengolahan dan proses menjadi lebih efisien.

Setelah itu **Ryan Ridho Valba**, ditahun 2018. Dengan judul “Perbandingan Algoritma Horspool dan Algoritma Raita pada Aplikasi Kamus Bahasa Indonesia-Jerman Berbasis Web”. Pertama, Algoritma Horspool melakukan pergeseran karakter sebanyak satu karakter paling kanan apabila cocok melakukan pergeseran karakter berikutnya ke kiri. Sedangkan Algoritma Raita melakukan pergeseran karakter sebanyak satu karakter paling kanan apabila cocok melakukan pergeseran ke awal karakter dan tengah karakter. Hal ini menunjukkan bahwa kedua Algoritma tersebut memiliki persamaan dalam pencarian karakter pada pencocokan karakter paling kanan dari teks. Rata – rata hasil running time Algoritma Horspool untuk pencarian kata adalah 0,2381 ms. Pada Algoritma Raita rata – rata hasil running time 0,2336 ms. Kedua Hasil Kompleksitas Algoritma Horspool adalah 𝑇(𝑛) = Ɵ(𝑚𝑛). Sedangkan Hasil kompleksitas Algoritma Raita adalah 𝑇(𝑛) = Ɵ(𝑚). Maka pencarian dengan menggunakan Algoritma Raita lebih cepat dibandingkan dengan Algoritma Horspool. Walaupun terkadang kedua algoritma ini tidak terlihat jauh berbeda saat diimplementasikan. Ketiga Pada *running time* dan kompleksitas dari kedua Algoritma menghasilkan bahwa Algoritma Raita lebih efektif dari Algoritma Horspool pada pencarian kata.

Kemudian selanjutnya **Ratih Anggraini**, pada tahun 2018. Dengan judul “Implementasi Algoritma Raita Search Pada Aplikasi Filsafat Berbasis

Android”. Penerapan Algoritma raita dapat mempercepat proses penemuan hasil pencarian arti istilah filsafat dan Aplikasi filsafat berbasis android yang dibangun pada penelitian ini dapat mempermudah masyarakat umum untuk memperoleh informasi mengenai arti istilah filsafat dengan mudah.

Setelah itu **Sukron Amin**, ditahun 2016. Dengan Judul “Analisis Dan Perancangan Sistem Informasi Manajemen Arsip Berbasis Web Pada Sekolah

Tinggi Ilmu Tarbiyah ( STIT ) Kabupaten Tebo”. Penelitian ini menghasilkan prototype rancangan sistem informasi manajemen arsip berbasis web bagi Sekolah Tinggi Ilmu Tarbiyah (STIT) Kabupaten Tebo yang dapat diimplementasikan sesuai dengan kebutuhan yang ada, terutama dalam memanajemen arsip yang dikelola oleh staf prodi STIT Kabupaten Tebo. Prototipe sistem informasi manajemen arsip berbasis web ini menampilkan informasi-informasi yang berkaitan dengan pengelolaan arsip dan peminjaman arsip.

Kemudian **Agustina Simangunsong**, pada tahun 2018. Dengan Judul “Sistem Informasi Pengarsipan Dokumen Berbasis Web”. Berdasarkan pembahasan tentang sistem informasi pengarsipan dokumen berbasis web pada perumnas- regional I Medan, maka diambil kesimpulan sebagai berikut : Pertama Sistem Pengarsipan yang dibangun di sub bagian SDM telah berbasis web, Kedua Penataan dokumen tersusun rapi dan menggunakan waktu yang lebih efektif; Ketiga Dengan adanya sistem ini, maka dapat memudahkan proses pekerjaan di dalam mencari dokumen yang dibutuhkan suatu waktu dengan cepat dan terperinci, keempat Dokumen terpelihara dan aman.

Berdasarkan tinjauan pustaka diatas maka peneliti menyimpukan bahwa pentingnya pembuatan sebuah aplikasi yang berhubungan dengan kearsipan serta algoritma raita merupakan algoritma yang baik dalam pencarian kata.

# BAB II

# LANDASAN TEORI

# 2.1 Pengalaman Bimbingan Lapangan (PBL)

Fakultas kesehatan masyarakat Universitas Halu Oleo merupakan fakultas yang berdiri sejak tahun 2012. Saat ini *program* studi kesehatan masyarakat mempunyai 6 peminatan yakni Administrasi Kebijakan Kesehatan, Kesehatan Lingkungan, Keselamatan dan kesehatan kerja, Epidemiologi, Promosi Kesehatan, dan Gizi. Dalam upaya pembinaan profesional dalam bidang promotif dan preventif yang mengarah kepada pemahaman permasalahan-permasalahan kesehatan masyarakat, untuk selanjutnya dapat dilakukan pengembangan *program* atau intervensi menuju perubahan pola pikir dan perilaku masyarakat yang diinginkan. Salah satu bentuk konkret upaya tersebut dengan melakukan pengalaman belajar lapangan (PBL).

Pengalaman Belajar Lapangan (PBL) adalah proses belajar untuk mendapatkan kemampuan dibidang kesehatan masyarakat. Pengalaman Belajar Lapangan (PBL) mempunyai peranan yang sangat penting dan strategis. Maka dari itu PBL harus dilaksanakan secara benar dan berkesinambungan. Setelah PBL yang dilakukan oleh mahasiswa Kesehatan Masyarakat maka selanjutnya adalah pembuatan laporan yang terdiri dari dua laporan dimana laporan pertama berisi data yang telah di kumpulkan melalu kuesioner dan juga data implementasi *program* yang di buat di tempat PBL, laporan kedua berisikan data evaluasi *program* (Fakultas Kesehatan Masyarakat, 2019).

# 2.2 Manajemen Sistem Informasi

Sistem Informasi Manajemen merupakan salah satu bidang kajian yang saat ini sedang berkembang secara pesat seiring dengan perkembangan dunia bisnis dan kemajuan teknologi informasi sebagai pendukung dari aktivitas bisnis. Semakin kompleksnya sistem dan perannya di berbagai bidang kehidupan membutuhkan

7

kajian Sistem Informasi Manajemen melalui pendekatan sosioteknik. Selama ini, pembahasan dan diskusi mengenai sistem informasi masih banyak asumsi umum yang menggiring pemahaman bahwa sistem informasi hanya berkaitan dengan pendekatan teknis saja.(Kom et al., 2018).

# 2.3 Arsip

Arsip merupakan memori korporat bagi organisasi yang menciptakannya. Ia memberikan bukti bagi tindakan, keputusan dan komunikasi, serta merupakan bahan akuntabilitas dari instansi yang memilikinya. Arsip lebih dari sekadar berisi data karena arsip merupakan bukti dari tindakan dan keputusan. Untuk dianggap sebagai arsip, suatu dokumen harus memiliki isi, struktur dan konteks. Suatu arsip yang memiliki atribut ini disebut arsip yang lengkap. Namun demikian, agar dapat dijadikan bukti, arsip tidak hanya harus lengkap, tetapi juga dapat diakses, reliabel, otentik, akurat, dan tidak dapat diganggu gugat (Rustam, 2014).

# 2.4 Pengertian Algoritma

Kata “Algoritma‟ diturunkan dari nama belakang seorang tokoh matematikawan Persia bernama Muhammad ibn Musa Al-Khuwarizmi. Secara umum, Algoritma merupakan metode umum yang digunakan untuk menyelesaikan kasus-kasus tertentu (Ananda, et al. 2009). Secara spesifik, Algoritma adalah sekumpulan prosedur komputasi yang terdefinisi dengan baik dan memiliki beberapa nilai atau himpunan nilai sebagai nilai masukan dan menghasilkan nilai atau himpunan nilai sebagai nilai keluaran. Sebuah Algoritma pada dasarnya adalah urutan langkah-langkah komputasi yang mengubah nilai *input*  menjadi nilai *output* (Charras, et al. 2001).

# 2.5 String Matching

*String matching* adalah pencarian sebuah *pattern* pada sebuah teks (Charras, et al. 2001). Konsep pencocokan *string* atau *string* *matching* dapat diartikan sebagai sebuah cara untuk mencari string yang sama didalam satu kumpulan teks atau basis data (*database*). Konsep pencocokan *string* ini mirip dengan fungsi *Find* dalam aplikasi pengolah kata atau fungsi *Query* di dalam basis data (Dewanto, 2007). Pencocokan *string* juga merupakan subjek yang sangat penting dalam domain yang lebih luas dari pengolahan Pencocokan *string* adalah *subjek* yang sangat penting dalam domain yang lebih luas dari pengolahan teks. Algoritma pencocokan *string* adalah komponen dasar yang digunakan dalam implementasi *software* praktis yang ada di bawah sistem operasi. Selain itu, mereka menekankan metode pemrograman yang berfungsi sebagai paradigma dalam medan lain dari ilmu komputer (sistem atau desain perangkat lunak). Akhirnya, mereka juga memainkan peran penting dalam ilmu komputer teoritis dengan memberikan masalah menantang (Charras, et al. 2001).

# 2.6 Klasifikasi Pencocokan String

Klasifikasi pencocokan *string* Pencocokan *string* (*string* *matching*) secara garis besar dapat dibedakan menjadi dua yaitu:

1. *Exact string matching*, merupakan pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokkan memiliki jumlah maupun urutan karakter dalam *string* yang sama. Contoh: kata *step* akan menunjukkan kecocokan hanya dengan kata *step*.
2. *Inexact* *string* *matching* atau *Fuzzy* *string* *matching*, merupakan pencocokan *string* secara samar, maksudnya pencocokan *string* dimana *string* yang dicocokkan memiliki kemiripan dimana keduanya memiliki susunan karakter yang berbeda (mungkin jumlah atau urutannya) tetapi *string*-*string* tersebut memiliki kemiripan baik kemiripan *tekstual*/penulisan (*approximate* *string* *matching*) atau kemiripan ucapan (*phonetic* *string* *matching*).

# 2.7 Algortima Raita

Algoritma Raita merupakan bagian dari Algoritma *exact string matching* yaitu pencocokan *string* secara tepat dengan susunan karakter dalam *string* yang dicocokan memiliki jumlah maupun urutan karakter dalam *string* yang sama. Raita merancang sebuah Algoritma dengan membandingkan karakter yang terakhir dari pola yang diawali dari karakter paling kanan dari “jendela”. Jika mereka cocok, kemudian karakter pertama dari pola teks paling kiri dari jendela juga dibandingkan. Jika mereka cocok, maka akan dibandingkan karakter tengah pola dengan karakter teks tengah jendela. Pada akhirnya, jika mereka benar-benar cocok, maka Algoritma membandingkan karakter lain mulai dari pola karakter kedua ke karakter kedua terakhir, dan akan membandingkan dengan karakter tengah lagi (Charras & Lecroq, 2001).

Adapun tahap yang digunakan dalam pencarian pada Algoritma Raita adalah sebagai berikut :

a. Membuat tabel pergeseran pola yang dicari sebagai kata yang akan dicari pada teks.

b Jika dalam proses pembandingan terjadi ketidakcocokan antara pasangan karakter pada akhir pola dengan karakter teks, pergeseran dilakukan sesuai nilai karakter pada Tabel BmBc.

1. Jika dalam proses pembandingan akhir pola terjadi ketidakcocokan lagi maka karakter akan digeser lagi sesuai Tabel BmBc.
2. Jika karakter akhir pola dengan karakter pada teks yang sedang dibandingkan cocok, maka posisi karakter pada pola dan teks akan memiliki nilai (0), dan dilanjutkan pencocokan pada karakter awal pola. Jika cocok maka dilanjutkan pencocokan dengan karakter tengah pola.
3. Jika akhir, awal dan tengah pola telah cocok. Pencocokan dilanjutkan dengan bagian kanan dari awal karakter pada pola, jika cocok maka dicocokkan pada bagian kanan tengah pola.

Algoritma Raita memiliki pola yang baik dalam pencarian, terutama dalam karakter kata tengah memeriksa dari pola yang diberikan dengan membandingkan karakter terakhir dari kedua pola (𝑚 − 1) dengan yang kedua karakter dari panjang karakter. Karakter tengah pola dibandingkan dua kali untuk karakter teks yang sesuai. prosedur ini diulang sampai tidak mencapai 𝑡 − 𝑚 + 1.

Algoritma Raita yang menganalisis untuk mencari file dapat diasumsikan sebagai pencarian teks, lihat contoh proses di bawah ini.

dalam pencarian *pattern* **Aprian** pada Teks **R.M.Alfian Apriansya D**.

Maka diketahui pula perhitungannya sebagai berikut :

𝑡 = Panjang pola m = Teks yang akan dicari Maka :

𝑡 = 𝑹. 𝑴. 𝑨𝒍𝒇𝒊𝒂𝒏 𝑨𝒑𝒓𝒊𝒂𝒏𝒔𝒚𝒂 𝑫

𝑚 = 𝑨𝒑𝒓𝒊𝒂𝒏 diketahui bahwa : 𝑚 = 6

Dibuatlah tabel BmBc untuk melakukan perhitungan de ngan persamaan sebagai berikut:

𝑚 – 2 . . . . . . . . . . . . . . (2.1)

6 – 2 = 4

Mencari nilai BmBc (𝑎) 𝑚 – 1 – 𝑖 . . . . . . . . (2.2)

Sebagai pencarian nilai karakter pada tabel BmBc

### Tabel 2.1 Perhitungan tabel BmBc

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **I** | **0** | **1** | **2** | **3** | **4** | **5** |
| **A** | A | P | R | I | A | N |
| **BmBc(a)** | **5** | **4** | **3** | **2** | **1** | **6** |

Pencarian nilai BmBc (a) dengan menggunakan rumus persamaan

𝑚 – 1 – 𝑖 . . . . . (2)

6 – 1 – 0 = 5 maka nilai diletakkan pada indeks 𝑘𝑒 − 0 dengan krakter A

6 – 1 – 1 = 4 maka nilai diletakkan pada indeks 𝑘𝑒 − 1 dengan karakter P

6 – 1 – 2 = 3 maka nilai diletakkan pada indeks 𝑘𝑒 − 2 dengan karakter R

6 – 1 – 3 = 2 maka nilai diletakkan pada indeks 𝑘𝑒 − 3 dengan karakter I

6 – 1 – 4 = 1 maka nilai diletakkan pada indeks 𝑘𝑒 − 4 dengan karakter A

Nilai 𝑁 adalah 6 sesuai dengan panjang pola, karena abjad yang tidak ada pada tabel maka diinisialisasikan dengan tanda (∗) kemudian nilainya sesuai dengan panjang pola. Jadi, untuk perhitungan Algoritma raita sesuai tabel BmBc adalah sebagai berikut:

### Tabel 2.2 Hasil BmBc(a)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Karakter**  **Teks** | **A** | **P** | **R** | **I** | **A** | **\*** |
| **BmBc(a)** | **5** | **4** | **3** | **2** | **1** | **6** |

Langkah selanjutnya adalah melakukan pencarian menggunakan Algoritma Raita dengan tahap-tahap berikut ini :

1. Tahap pertama, yaitu mencocokan pola akhir pada teks. Jika tidak cocok maka pola akan bergeser kekanan sebanyak nilai teks.

### Tabel 2.3 Proses Pencarian Pada Teks Ke-1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Teks** | **R** | **.** | **M** | **.** | **A** | **L** | **F** | **I** | **A** | **N** |  | **A** | **P** | **R** | **I** | **A** | **N** | **S** | **Y** | **A** | **.** | **D** |
| **BmBc**  **(a)** | **A** | **P** | **R** | **I** | **A** | **N** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Pada proses dinyatakan terjadi ketidakcocokan pada teks, maka Pada pola terakhir teks terdapat huruf terakhir yaitu N. Pada tabel hasil BmBc bahwa L bernilai 6 karakter.

2. Tahap kedua, yaitu pola akan begeser sebanyak 6 karakter

### Tabel 2.4 Proses Pencarian Pada Teks ke-2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Teks** | **R** | **.** | **M** | **.** | **A** | **L** | **F** | **I** | **A** | **N** |  | **A** | **P** | **R** | **I** | **A** | **N** | **S** | **Y** | **A** | **.** | **D** |
| **BmBc**  **(a)** |  |  |  |  |  |  | **A** | **P** | **R** | **I** | **A** | **N** |  |  |  |  |  |  |  |  |  |  |

Pada proses dinyatakan terjadi ketidakcocokan pada teks, maka Pada pola terakhir teks terdapat huruf terakhir yaitu N. Pada tabel hasil BmBc bahwa A bernilai 1 karakter.

3. Tahap ketiga, yaitu pola akan begeser sebanyak 1 karakter

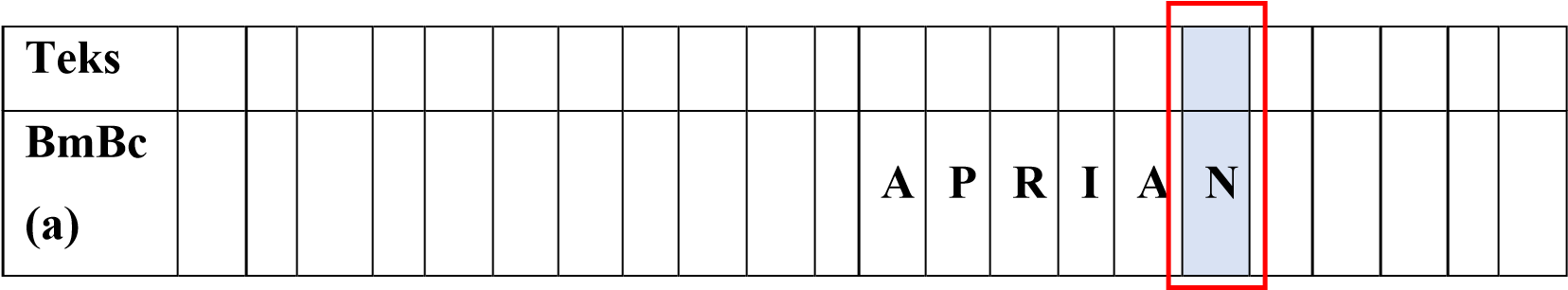
### Tabel 2.5 Proses Pencarian Pada Teks ke-3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Teks** | **R** | **.** | **M** | **.** | **A** | **L** | **F** | **I** | **A** | **N** |  | **A** | **P** | **R** | **I** | **A** | **N** | **S** | **Y** | **A** | **.** | **D** |
| **BmBc**  **(a)** |  |  |  |  |  |  |  | **A** | **P** | **R** | **I** | **A** | **N** |  |  |  |  |  |  |  |  |  |

Pada proses dinyatakan terjadi ketidakcocokan pada teks, maka Pada pola terakhir teks terdapat huruf terakhir yaitu N. Pada tabel hasil BmBc bahwa P bernilai 4 karakter.

4. Tahap keempat, yaitu jika pergeseran pola terdapat ketidakcocokan, maka pola akan terus bergeser sebanyak 4 karakter

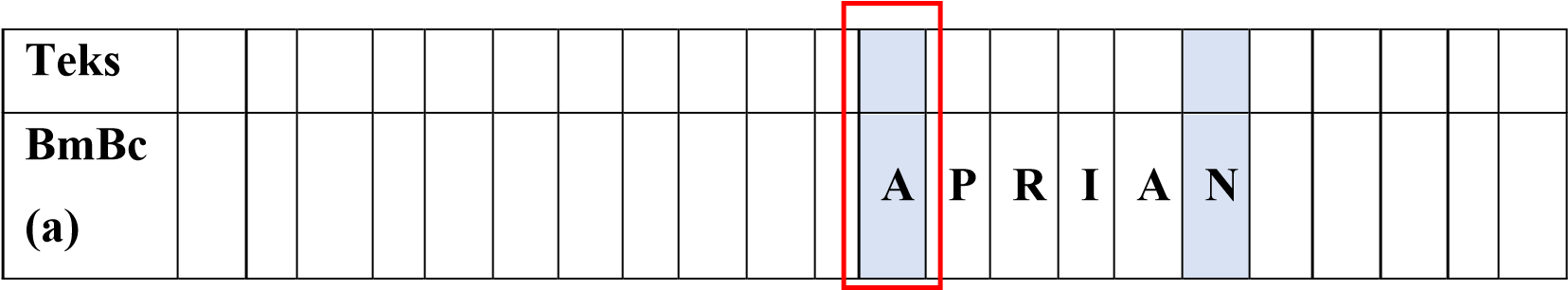
### Tabel 2.6 Proses Pencarian Pada Teks ke-4

**R . M . A L F I A N A P R I A N S Y A . D**

Dilihat pada Tabel 1.5 terjadi kecocokan antara teks dan pola pada kolom text yaitu N, dan pada kolom BmBC(a) yaitu N. sehingga selanjutnya proses pengecekan index ke sebelah kiri kolom BmBc(a).

5. Tahap kelima, yaitu pengecekan pola pada kiri index kolom BmBc(a) yaitu A dan kolom Teks yaitu A.

### Tabel 2.7 Proses Pencarian Pada Teks ke-5

**R . M . A L F I A N A P R I A N S Y A . D**

Dilihat pada Tabel 2.7 terjadi kecocokan menurut table BmBC(a) antara teks dan pola pada kolom text yaitu A bernilai 5 , dan pada kolom BmBC(a) yaitu A bernilai 5. sehingga selanjutnya proses pengecekan index ke tengah.

1. Tahap keenam, yaitu pengecekan pola pada tengah text . jika jumlah text genap maka nilai tengah yang diambil berdekatan dengan nilai karakter paling kanan, dan jika ganjil maka sebaliknya. kolom BmBc(a) yaitu I dan kolom Teks yaitu I.

**Tabel 2.8 Proses Pencarian Pada Teks ke-6**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | | | | | | |  |  | | | | | | |
| **Teks** | **R** | **.** | **M** | **.** | **A** | **L** | **F** | **I** | **A** | **N** |  | **A** | **P** | **R** | **I** | **A** | **N** | **S** | **Y** | **A** | **.** | **D** |
| **BmBc**  **(a)** |  |  |  |  |  |  |  |  |  |  |  | **A** | **P** | **R** | **I** | **A** | **N** |  |  |  |  |  |

Dilihat pada Tabel 2.8 terjadi kecocokan menurut table BmBC(a) antara teks dan pola pada kolom text yaitu I bernilai 3 , dan pada kolom BmBC(a) yaitu I bernilai 3. sehingga selanjutnya proses pengecekan index ke kanan karakter kedua.

1. Tahap ketujuh, yaitu pengecekan pola pada karakter kanan kedua. kolom BmBc(a) yaitu A dan kolom Teks yaitu A.

### Tabel 2.9 Proses Pencarian Pada Teks ke-7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | | | | | | | |  |  | | | | | |
| **Teks** | **R** | **.** | **M** | **.** | **A** | **L** | **F** | **I** | **A** | **N** |  | **A** | **P** | **R** | **I** | **A** | **N** | **S** | **Y** | **A** | **.** | **D** |
| **BmBc**  **(a)** |  |  |  |  |  |  |  |  |  |  |  | **A** | **P** | **R** | **I** | **A** | **N** |  |  |  |  |  |
|  | | | | | | | | | | | | | | | |  |  | | | | | |

Dilihat pada Tabel 2.9 terjadi kecocokan menurut table BmBC(a) antara teks dan pola pada kolom text yaitu A bernilai 1 , dan pada kolom BmBC(a) yaitu A bernilai 1. sehingga selanjutnya proses pengecekan index ke kiri karakter kedua.

8. Tahap kedelapan, yaitu pengecekan pola pada karakter kiri kedua. kolom BmBc(a) yaitu P dan kolom Teks yaitu P.

### Tabel 2.10 Proses Pencarian Pada Teks ke-8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | | | | | | |  |  | | | | | | | | |
| **Teks** | **R** | **.** | **M** | **.** | **A** | **L** | **F** | **I** | **A** | **N** |  | **A** | **P** | **R** | **I** | **A** | **N** | **S** | **Y** | **A** | **.** | **D** |
| **BmBc**  **(a)** |  |  |  |  |  |  |  |  |  |  |  | **A** | **P** | **R** | **I** | **A** | **N** |  |  |  |  |  |
|  |  | | | | | | | | | |  | |  |  | | | | | | | | |

Dilihat pada Tabel 2.10 terjadi kecocokan menurut table BmBC(a) antara teks dan pola pada kolom text yaitu P bernilai 4 , dan pada kolom BmBC(a) yaitu P bernilai 4. sehingga selanjutnya proses pengecekan index ke kiri karakter kedua.

9. Tahap kesembilan, yaitu pengecekan pola pada karakter tengah atau sisa karakter yang belum dicek. kolom BmBc(a) yaitu R dan kolom Teks yaitu R

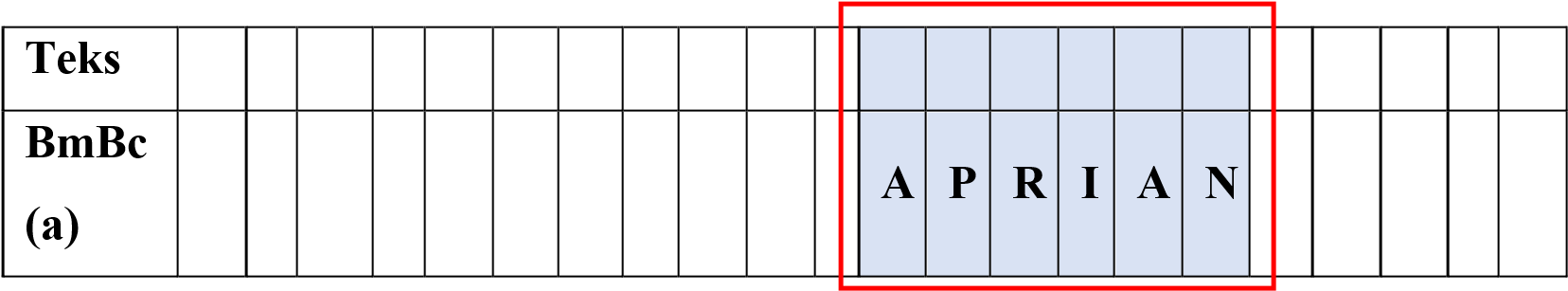
### Tabel 2.11 Proses Pencarian Pada Teks ke-9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Teks** | **R** | **.** | **M** | **.** | **A** | **L** | **F** | **I** | **A** | **N** |  | **A** | **P** | **R** | **I** | **A** | **N** | **S** | **Y** | **A** | **.** | **D** |
| **BmBc**  **(a)** |  |  |  |  |  |  |  |  |  |  |  | **A** | **P** | **R** | **I** | **A** | **N** |  |  |  |  |  |
|  |  | | | | | | | | | |  | | |  |  | | | | | | | |

Dilihat pada Tabel 2.11 terjadi kecocokan menurut table BmBC(a) antara teks dan pola pada kolom text yaitu R bernilai 2 , dan pada kolom BmBC(a) yaitu R bernilai 2. sehingga selanjutnya Seluruh pencocokan karakter menggunakan Algoritma Raita dicek berdasarkan index pada table bmbc(a).

10. Tahap kesepuluh, yaitu pencocokan pola pada table BmBc(a)

### Tabel 2.12 Proses Pencarian Pada Teks ke-10

**R . M . A L F I A N A P R I A N S Y A . D**

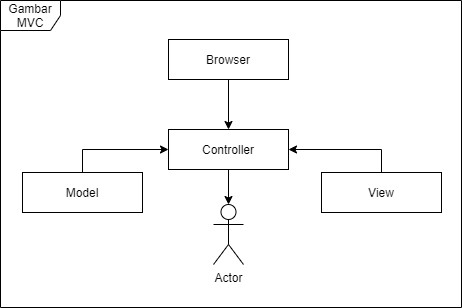
Dilihat pada Tabel 2.12 terjadi kecocokan menurut table BmBC(a) antara teks dan pola pada kolom text telah cocok. sehingga selanjutnya Seluruh pencocokan karakter menggunakan Algoritma Raita telah selesai dan berhenti.

# 2.8 Framework CodeIgniter

*Framework* merupakan perangkat lunak yang mulai menjadi pilihan untuk membuat suatu aplikasi (Marsudi, 2016). Kemudahan-kemudahan yang diberikan menarik orang-orang untuk menggunakannya. Hal ini tidak terlepas dari tingkat efektifitas dan efisiensinya yang lebih baik dalam proses pengembangan suatu perangkat lunak. *Framework* adalah suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi web. (Raharjo 2015)

Proses pengembangan web dapat dilakukan dengan beragam bahasa pemrograman seperti *PHP, Python, Ruby, Perl, C++, JAVA* dan sebagainya. Saat ini, banyak bermuculan *framework* web yang dirancang untuk bahasa pemrograman tersebut. Salah satunya adalah *CodeIgniter*. *CodeIgniter* adalah kerangka kerja PHP yang kuat dengan tapak yang sangat kecil, dibangun untuk pengembang yang membuthkan *toolkit* sederhana dan elegan untuk membuat aplikasi web berfitur lengkap.

CodeIgniter didasarkan pada pola pengembangan *Model-View-Controller* (MVC). MVC adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi. Adapun pola MVC sebagai berikut :



### Gambar 2.1 MVC Pada Codeigniter

Model merupakan struktur data. Secara spesifik *class* model akan mengandung fungsi kode yang akan membantu dalam segala proses yang berhubungan dengan *database* seperti memasukkan, mengedit mendapatkan dan menghapus data dalam sebuah *database*. *View* merupakan informasi yang disampaikan ke pengguna. Sebuah *view* biasanya berupa halaman web, sebuah *view* juga bisa berupa sebuah *fragmen* halaman seperti *header* dan *footer*. *View* juga bisa berupa halaman RSS atau jenis halaman web yang lain. *Controller* merupakan sebuah perantara antara Model dan View dan semua sumber yang dibutuhkan untuk memproses permintaan HTTP dalam membuat halaman web (Prabowo, 2015).

Beberapa keunggulan yang kemudian ditawarkan oleh *CodeIgniter* yaitu sebagai berikut :

1. *CodeIgniter* adalah *framework* yang bersifat *free* dan *open source*.
2. *CodeIgniter* memiliki ukuran yang kecil dibandingkan dengan *framework* lain. Setelah proses instalasi, *framework* *CodeIgniter* hanya berukuran kurang lebih 2 MB. Dokumentasi *CodeIgniter* memiliki ukuran sekitar 6 MB.
3. Aplikasi yang dibuat menggunakan *CodeIgniter* bisa berjalan cepat.
4. *CodeIgniter* menggunakan pola desain *Model-View-Controller* (MVC) sehingga satu file tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara dikemudian hari.
5. *CodeIgniter* dapat diperluas sesuai dengan kebutuhan.
6. *CodeIgniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *CodeIgniter* dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.

# 2.8 Database Management Sistem (DBMS)

*Database Management**Sistem* (DBMS) merupakan perangkat lunak untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data yang berskala besar. Penggunaan DBMS saat ini merupakan hal yang sangat penting dalam segala aspek, baik itu dalam skala yang besar atau kecil. Sebagai contoh media sosial facebook menggunakan DBMS untuk menyimpan data-data pengguna *facebook* yang sangat banyak kedalam DBMS MySQL. Beberapa DBMS yang digunakan adalah MySQL dan MariaDB. Berdasarkan survey yang dilakukan, MySQL dan MariaDB merupakan DBMS yang banyak digunakan sebagai contoh survey yang terdapat pada db-engines.com *DB-Engines Ranking* menempatkan MySQL pada posisi ke-2 sedangkan MariaDB pada posisi ke-20 namun pada *survey* yang terdapat di serverwatch.com *Top 10 Enterprise Database Sistem Of 2016*, MariaDB menempati posisi ke-6 dan MySQL menempati posisi ke-7 (Warman & Ramdaniansyah, 2018).

## 2.8.1 MySQL

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS *Multithread* dan *multi user*. MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam *database* untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan secara mudah dan otomatis. MySQL diciptakan oleh Michael "Monty" Widenius pada tahun 1979, seorang *programmer* komputer asal Swedia yang mengembangkan sebuah sistem *database* sederhana yang dinamakan UNIREG yang menggunakan koneksi *low-level ISAM database engine* dengan *indexing* 1.

1. Kelebihan Dan Kekurangan MYSQL

Adapun kelebihan dari MySQL dalam penggunaanya dalam *database* yakni sebagai berikut :

1. Gratis sehingga MySQL dapat dengan mudah untuk mendapatkannya.
2. MySQL stabil dalam pengoprasiannya.
3. MySQL mempunyai sistem keamanan yang cukup baik.
4. Sangat mendukung transaksi dan mempunyai banyak dukungan dari komunitas.
5. Sangat *fleksibel* dengan berbagai macam *program* Perkembangan dari *MySQL* sangat cepat.

Adapun kelemahan dari MySQL dalam penggunaanya dalam *database* yakni sebagai berikut :

* 1. Sulit untuk diaplikasikan pada intansi atau perusahan dengan *database* yang besar
  2. *Support* yang kurang
  3. Tidak populer untuk aplikasi *game* dan *mobile*

2. Istilah-istilah Didalam *Database*

Ada beberapa istilah dalam *database* yang nantinya akan sering disebut didalam tutorial, sebelum kita masuk ke tutorial selanjutnya maka harus dapat memahami istilah-istilah dibawah ini agar lebih mudah dalam mempelajarinya.

Adapun istilah-istilah tersebut sebagai berikut :

1. Pengertian Tabel

Sebuah tabel merupakan kumpulan data (nilai) yang diorganisasikan ke dalam baris (*record*) dan kolom (*field*). Masing-masing kolom memiliki nama yang spesifik dan unik.

1. Pengertian *FieldField*

*FieldField* merupakan kolom dari sebuah *tabel*. *Field* memiliki ukuran *type* data tertentu yang menentukan bagaimana data nantinya tersimpan.

1. Pengertian *Record* atau Data

*Record* atau Data merupakan sebuah kumpulan nilai yang saling terkait.

1. Pengertian *Key*

*Key* merupakan suatu *field* yang dapat dijadikan kunci dalam operasi

tabel. Dalam konsep database, key memiliki banyak jenis diantaranya *Primary Key, Foreign Key, Composite Key, dll.*

3. fitur-fitur MYSQL

Adapun fitur-fitur serta kapabilitas yang dimiliki oleh MySQLdalam menjalankanya yakni sebagai berikut :

1. Unjuk kerja yang tinggi dalam memproses *query* sederhana, dalam arti dapat memproses lebih banyak SQL per satuan waktu.
2. Memiliki lebih banyak tipe data seperti : *signed*/*unsigned* *integer* yang memiliki panjang data sebesar 1,2,3,4 dan 8 *byte, Float, Double, Char, Varchar, Text, Blob, Date, Time, Datetime, Timestamp, Year, Set* dan *Tipe Enum.*
3. Mendukung *field* yang dijadikan *Index*, dengan maksimal 32 *index* dalam satu tabel.
4. MYSQL memiliki beberapa lapisan keamanan, seperti *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perijinan yang mendetail serta sandi atau *password* terenkripsi.
5. Konektivitas , MySQL dapat melakukan koneksi dengan *client* menggunakan *protokol* TCP/IP, *Unix soket* (UNIX), atau *Named Pipes* (NT).
6. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik
7. *Command* and *function*, MySQL memiliki fungsi dan operator secara penuh yang mendukung perintah *select* dan *where* dalam *query*.
8. *Structure* *Tabel*, MySQL memiliki struktur tabel yang lebih *fleksibel* dalam menangani *Alter Tabel* dibandingkan DBMS lainnya.
9. Mendukung penuh terhadap kalimat *Sql Group By Dan Order By.* Mendukung terhadap fungsi penuh *( COUNT(),COUNT(), DISTINCT() AVG(), STD(), SUM(), MAX() dan MIN() ).*

## 2.8.2 XAMPP

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB

(sekarang), PHP, dan *Perl*. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS *Linux*, OS *Windows*, Mac OS, dan juga Solaris (andy,2019). Sejarah mencatat, *software* XAMPP pertama kali dikembangkan oleh tim proyek bernama *Apache* *Friends* dan sampai saat ini sudah masuk dalam rilis versi 7.3.9yang bisa didapatkan secara gratis dengan label GNU (*General Public License*).

Jika dijabarkan secara gamblang, masing-masing huruf yang ada di dalam nama XAMPP memiliki arti sebagai berikut ini: a. X artinya *Cross Platform*

Merupakan kode penanda untuk *software cross platform* atau yang bisa berjalan di banyak sistem operasi. b. A artinya *Apache*

*Apache* adalah aplikasi *web server* yang bersifat gratis dan bisa dikembangkan oleh banyak orang (*open* *source*). c. M artinya MySQL / MariaDB

MySQL atau MariaDB merupakan aplikasi *database server* yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui *database*. d. PHP

Huruf “P” yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat *website* dinamis, contohnya dalam *website* berbasis CMS *WordPress*.

e. P artinya *Perl*

Sementara itu, untuk huruf P selanjutnya merupakan singkatan dari bahasa pemrograman *Perl* yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. *Perl* ini bisa berjalan di dalam banyak sistem operasi sehingga sangat fleksibel dan banyak digunakan.

*Program* aplikasi XAMPP berfungsi sebagai *server* lokal untuk mengampu berbagai jenis data *website* yang sedang dalam proses pengembangan. Dalam prakteknya, XAMPP bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam *website* kepada orang lain tanpa harus terkoneksi dengan internet, atau istilahnya *website* *offline* . XAMPP bekerja secara *offline*  layaknya *web hosting* biasa namun tidak bisa diakses oleh banyak orang. Maka dari itu, XAMPP biasanya banyak digunakan oleh para mahasiswa maupun pelajar untuk melihat hasil desain *website* sebelum akhirnya dibuat *online* menggunakan *web hosting* yang biasa dijual dipasaran.

# 2.9 Metode Perancangan UML (Unified Modeling Language)

UML adalah bahasa permodelan untuk merancang sistem berbasis *object- oriented.* UML digunakan untuk pengembangan sebuah sistem yang dapat menyampaikan alur kerja sistem dan menjelaskan tugas setiap *user* dalam sebuah sistem. UML memiliki tiga kategori diagram, yaitu *strucrture diagrams, behavior diagram,* dan *interaction diagram.* Ada beberapa jenis diagram dalam UML :

1.  *Use Case Diagram*

*Use case* atau diagram *use case* merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi- fungsi itu (A. S & Shalahuddin, 2013).

### Tabel 2.13 Simbol-simbol Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. |  | *Use Case* | Fungsionalitasi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama *use case.* |
|  |  | Aktor */ actor* | Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan mengguakan kata benda diawal frase nama *actor*. |
| 3. |  | *Asosiasi / association* | Komuniksi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan *actor*. |
| 4. | << extend >> | *Ekstensi / extend* | *Relasi* *use case* tambahan ke sebuah *use case* dimana *usecase* yang Ditambahkan dapat berdiri sendiri walau tanpa use case tambahan. |
| 5. |  | *Generalisasi*  */generalizati on* | Hubungan *generalisasi* dan  *spesialiasi* (umum - khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 6. | *<<Include>>* | *Include* | Relasi *use case* tambahan ke sebuah use case di mana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya  atau sebagai syarat dijalankan *use case* ini. |

Sumber: AS Rosa dan M. Shalahuddin, 2013.

2.  *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar *programmer* dapat membuat kelas-kelas didalam *program* perangkat lunak sesuai dengan perancangan diagram kelas (A. S & Shalahuddin, 2013). Ada beberapa simbol-simbol yang terdapat dalam diagram kelas, yaitu:

### Tabel 2.14 Simbol-simbol Class Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. | |  | | --- | | **nama\_kelas** | | + atribut | | + operasi() | | Kelas | Kelas pada struktur sistem |
| 2. |  | Antarmuka / *interface* | Semua dengan konsep *interface* dalam  pemrograman b*error* ientasi objek |
| 3. |  | Asosiasi / *association* | Relasi antarkelas dengan makna umum,  asoasi biasanya juga disertai dengan *multiplicity* |
| 4. |  | Asosiasi berarah  /*directed asosiasi* | Relasi antarkelas dengan makna kelas yang satu  digunakan oleh kelas  yang lain, asosiasi biasanya juga  disertai dengan *multiplicity* |
| 5. |  | Generalisasi | Relasi antarmuka dengan makna  generalisasi-spesialisasi  (umum khusus) |
| 6. |  | Kebergantungan / *dependency* | Relasi antarmuka dengan makna  kebergantungan antarkelas |
| 7. |  | Agregasi / *aggregation* | Relasi antarkelas dengan makna semua bagian (*whole- part*) |

Sumber: AS Rosa dan M. Shalahuddin, 2013.

1. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau *menu* yang ada pada perangkat lunak (A. S & Shalahuddin, 2013). Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor.

**Tabel 2.15 Simbol-simbol Activity Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. |  | *Start* / Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal |
| 2. |  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja |
| 3. |  | Percabangan / *desicion* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| 4. |  | Penggabungan / *join* | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5. |  | Status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6. |  | *swimlane* | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

Sumber: AS Rosa dan M. Shalahuddin, 2013.

1. *Sequence Diagram*

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan terima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang dimiliki kelas yang diinstasiasi menjadi objek itu. Menggambar diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case* (A. S & Shalahuddin, 2013).

### Tabel 2.16 Simbol-simbol Sequence Diagram

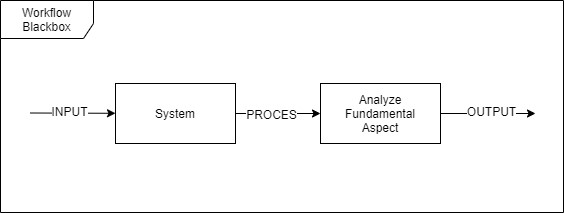
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Keterangan** |
| 1. | atau  Nama aktor | Aktor | Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan mengguakan kata benda diawal frase nama aktor |
| 2. |  | Garis hidup / *lifeline* | Menyatakan kehidupan suatu objek |
| 3. | Nama objek : nama kelas | Objek | Menyatakan objek yang berinteraksi pesan |
| 4. |  | Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya. |
| 5. |  | Pesan tipe *create* | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat |
| 6. |  | Pesan tipe *call* | Menyatakan suatu objek memanggil opeasi/metode yang ada pada objek  lain atau dirinya sendiri |
| 7. |  | Pesan tipe *send* | Menyatakan bahwa suatu objek mengirim data / masukkan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi |
| 8. |  | Pesan tipe *return* | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| 9. |  | Pesan *type* *distroy* | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri,  sebaiknya jika ada create makan ada *destroy* |

Sumber: AS Rosa dan M. Shalahuddin, 2013.

# 2.10 Pengujian Perangkat Lunak Black Box

Pengujian perangkat lunak adalah sangat diperlukan dalam suatu sistem informasi, dimana dengan melakukan suatu pengujian akan ditemukan kesalahan atau *error* yang muncul dari sistem perangkat lunak tersebut. Dengan demikian tentunya seorang *programmer* akan bisa mengetahui dan apa yang harus dikerjakan selanjutnya. pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas perangkat lunak tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi di mana peluang terjadinya kesalahan manusia sangat besar dan arena ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas (Rouf, 2012).

*Black-Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, *tester* dapat mendefinisikan kumpulan kondisi *input*  dan melakukan pengetesan pada spesifikasi fungsional *program* (Rouf, 2012).



### Gambar 3.2 Workflow Black Box

Penggunaan metode dalam pengujian *Black Box testing* terdiri dari 6 yakni sebagai berikut:

1. *Equivalence class partitioning*

Partisi ekuivalensi adalah metode yang membagi domain *input*  dari suatu *program* ke dalam kelas data, menentukan kasus pengujian dengan mengungkapkan kelas-kelas kesalahan sehingga akan mengurangi jumlah keseluruhan kasus pengujian (Gelar, Komputer, & Informatika, 2011). Adapun penentuan kelas Partisi ekuivalensi yakni :

1. Bagi domain *Input*  ke dalam beberapa kelas yang nantinya akan dijadikan sebagai kasus uji.
2. Kelas yang telah terbentuk disajikan sebagai kondisi *input*  dalam kasus uji.
3. Kelas tersebut merupakan himpunan nilai-nilai yang *valid* dan tidak *valid*.
4. Kondisi *input*  bisa merupakan suatu *range*, harga khusus, suatu himpunan, atau suatu *boolean*.
5. Bila kondisi *input*  berupa suatu *range*, maka *input*  kasus ujinya satu valid dan dua yang invalid.
6. Bila kondisi *input*  berupa suatu harga khusus, maka *input*  kasus ujinya satu valid dan dua yang invalid.
7. Bila kondisi *input*  berupa suatu anggota himpunan, maka *input*  kasus ujinya satu valid dan dua yang *invalid*
8. Bila kondisi *input*  berupa suatu anggota *Boolean* , maka *input*  kasus ujinya satu *valid* dan dua yang *invalid*

2. *Sample* *testing*

Adapun penggunaan sample testing pada metode *blackbox* yaitu sebgai berikut:

1. Melibatkan sejumlah nilai yang dipilih dari data masukan kelas ekivalensi
2. Integrasikan nilai tersebut ke dalam kasus uji
3. Nilai yang dipilih dapat berupa konstanta atau variabel *Limit* *Testing*
4. Kasus uji yang memproses nilai batas (atau titik singular)
5. Nilai batas disimpulkan dari kelas ekivalensi dengan mengambil nilai yang sama atau mendekati nilai yang membatasi kelas akivalensi tersebut
6. *Limit* *test* *also* juga melibatkan data keluaran dari ekivalensi kelas
7. Pada kasus segi tiga, misalnya limit testing mencoba untuk mendeteksi apakah a+b >= c dan bukan a + b > c
8. Bila kondisi *input*  menentukan suatu *range*, maka kasus ujinya harus mencakup pengujian nilai batas dari range dan nilai invalid yang dekat dengan nilai batas. Misal bila rangenya antara [-1.0, +1.0], maka *input*  untuk kasus ujinya adalah -1.0, 1.0, - 1.001,1.001
9. Bila kondisi *input* nya berupa harga khusus kasaus ujinya harus mencakup nilai minimum dan maksimum. Misal suatu *file* dapat terdiri dari 1 to 255 record, maka kasus ujinya harus mencakup untuk nilai 0, 1, 255 and 256, atau uji saat keadaan *record* kosong dan *record* penuh.

1. *Limit* *testing*

Data dipilih dari luar *range* yang didefinisikan. Tujuan pengujian ini adalah untuk membuktikan tidak adanya kejadian yang katastropik yang dihasilkan akibat adanya keabnormalan.

1. *Robustness* *testing*

Data dipilih dari luar *range* yang didefinisikan. Tujuan pengujian ini adalah

untuk membuktikan tidak adanya kejadian yang katastropik yang dihasilkan akibat adanya keabnormalan

1. *Behavior* *testing*

Suatu pengujian yang hasilnya hanya dapat dievaluasi per *sub* *program*, tidak bisa dilakukan per modul

1. *Requirement* *Testing*

Adapun penggunaan *Behavior* *testing* pada metode *blackbox* yaitu sebagai berikut:

1. Menyusun kasus uji untuk tiap kebutuhan yang berkorelasi dengan modul / CSU.
2. Tiap kasus uji harus dapat dirunut dengan kebutuhan perangkat lunaknya melalui matriks keterunutuan.

Pengujian *Black Box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input*  yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu *program* (Lumbanraja, 2018). *Black Box* dapat menemukan kesalahan dalam kategori berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam strutur data atau akses basis data *eksternal*.
4. Inisialisasi dan kesalahan terminasi.
5. Validitas fungsional.
6. Kesensitifan sistem terhadap nilai *input*  tertentu.
7. Batasan dari suatu data

**BAB III METODOLOGI PENELITIAN**

### 3.1 Metode Pengumpulan Data

Adapun metode yang digunakan dalam penelitian dan pembuatan Aplikasi Manajemen Berkas PBL adalah sebagai berikut :

1. Kajian Pustaka

Kajian pustaka adalah metode pengumpulan data yang dilakukan dengan cara melakukan pengumpulan data pendukung penelitian yang akan dijadikan referensi, data dapat berupa buku, *paper,* jurnal, skripsi, dan sebagainya.

1. Kuesioner

Kuesioner adalah salah satu metode pengumpulan data yang dilakuakan dengan cara menyediakan daftar pertanyaan seputar penelitian yang akan dijawab oleh responden guna melakukan penilaian terhadap penelitian.

### 3.2 Metode Pengembangan Sistem

#### 3.2.1 Metode *Waterfall*

Tahapan utama dari *waterfall* model (Sommerville, 2011) langsung mencerminkan aktivitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall* model, yaitu *requirement, analysis and definition, sistem and software design, implementation and unit testing, integration and sistem testing, dan operation and maintenance.*

Berikut ini penjelasan tentang tahapan-tahapan dari *waterfall* model yakni sebagai berikut :

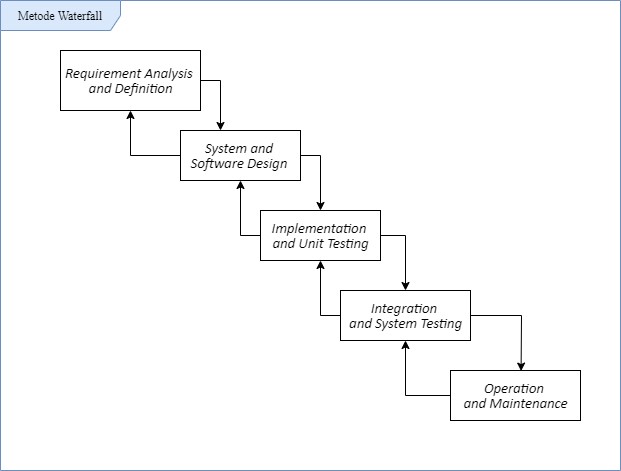
1. *Requirement Analysis and Definition* Merupakan tahapan penetapan fitur, analisa kebutuhan, kendala pembuatan dan tujuan sistem melalui konsultasi dengan pengguna sistem.
2. *Sistem and Software Design* Merupakan tahapan pembentukan arsitektur sistem berdasarkan persyaratan yang telah ditetapkan pada tahapan sebelumnya.

32

33

1. *Implementation and Unit Testing* Gambar 3.1 Diagram Tahapan *Waterfall* Model 8 Merupakan tahapan hasil dari desain perangkat lunak untuk direalisasikan sebagai satu *set* *program* atau unit *program*.
2. *Integration and Sistem Testing* Merupakan tahapan mengintegrasikan setiap unit *program* satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang sudah ditetapkan.
3. *Operation and Maintenance* Merupakan tahapan instalasi dan penerapan sistem. Pada tahap ini juga dilakukan pengujian pada saat sistem dijalankan untuk menemukan dan memperbaiki *error* yang tidak ditemukan pada tahap pembuatan.

Adapun gambar pada tahapan-tahapan dari *waterfall* *model* Sommerville yakni ditunjukan pada gambar 3.1 :



### Gambar 4.1 Diagram Tahapan Waterfall

#### 3.3 Waktu dan Tempat Penelitian

##### 3.3.1 Waktu Penelitian

Adapun waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan februari 2020 sampai dengan Juni 2020. Rincian Kegiatan dapat dilihat pada Tabel 3.1 berikut :

### Tabel 3.1 Gannt Chart Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Uraian** |  |  | | | **Waktu (2020-2021)** | | | | | | | | | | | | | | | |
|  | **Feb** | | | **Maret** | | | | **Apr** | | | | **Mei** | | | | **Jun** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| 1. | Analisis Kebutuhan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Analisis Spesifikasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Desain Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Implemenntasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Integrasi dan  Pengujian |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Pemeliharaan |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

#### 3.3.2 Tempat Penelitian

Adapun tempat pelaksanaan penelitian tugas akhir yang akan dilakukan pada Fakultas Kesehatan Masyarakat Universitas Halu Oleo Kota Kendari Provinsi Sulawesi Tenggara.

**BAB IV**

**ANALISIS DAN PERANCANGAN**

### 4.1 Analisis Sistem

Analisis sistem merupakan tahapan yang bertujuan untuk mengetahui apa saja yang dibutuhkan dalam sistem yang akan dibuat. Pembahasan pada analisis sistem ini yaitu analisis kebutuhan fungsional dan analisis kebutuhan nonfungsional.

### 4.2 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah semua data yang diperlukan oleh sistem agar sistem dapat berjalan sesuai dengan mekanisme yang dibangun melalui perancangan sistem. Adapaun kebutuhan sistem yang akan dibangun, adalah Perancangan diagram sistem menggunakan bahasa permodelan UML (*unified modeling language*) yang meliputi *usecase diagram, activity diagram,* dan *class diagram, sequence diagram.*

### 4.3 Analisis Kebutuhan Nonfungsional

Analisis kebutuhan nonfungsional bertujuan untuk menganalisis sumber daya yang dibutuhkan untuk membangun aplikasi yang akan dibangun. Analisis kebutuhan nonfungsional dibagi dalam dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak. Analisis kebutuhan perangkat keras yaitu merinci perangkat yang dibutuhkan untuk membangunn aplikasi sedangkan analisis kebutuhan perangkat lunak adalah merinci *software* yang dibutuhkan dalam membantu komponen perangkat keras dapat berjalan pada sistem. 1. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk membangun Aplikasi Manajemen Arsip PBL ditunjukan pada tabel 4.1berikut :

35

### Tabel 4.1 Spesifikasi perangkat keras

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** |
| 1. | *Laptop* | *Asus Vivobook* |
| 2. | *Processor* | *Intel Core I7-7500U @ 2.70GHz*  *(4 CPUs)* |
| 3. | *Monitor* | *Monitor 14 inci* |
| 4. | *Memori* | *RAM 8 GB DDR 4* |
| 5. | *Fashdisk* | *16 GB* |
| 6. | *Smartphone opppo* | *RAM 4 GB* |

2. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk membangun Aplikasi Manajemen Arsip PBL ditunjukan pada tabel 3.3 berikut:

### Tabel 4.2 Spesifikasi perangkat lunak

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Perangkat** | **Fungsi** | **Spesifikasi** |
| 1 | *Operation Sistem* | Sistem Operasi adalah perangkat lunak sistem yang mengatur sumber daya dari  perangkat keras (*Hardware*) dan perangkat lunak  (*Software*), serta sebagai jurik  (*daemon*) untuk *program* komputer. Tanpa sistem  operasi, pengguna tidak dapat  menjalankan *program* aplikasi pada komputer mereka, kecuali *program* *booting*. | *Windows* 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| 2. | *Java Developmen*  *Kit* (JDK) | JDK adalah sebuah perangkat lunak yang menyediakan *tools* seperti  Java *Compiler* yang digunakan pada IDE atau  SDK untuk membangun aplikasi Java. | JDK 1.8.0 |
| 4 | *Visual Studio code* | Merupakan aplikasi *text* *editor* yang digunakan untuk  mengetikan kode *program* pada saat pembuatan dan pengembangan *website*. | *Version 1.41* |
| 5. | *CodeIgniter* | Merupakan *framework* dari PHP yaitu bahasa pemrograman yang bersifat *backend*,  digunakan untuk membuat  fungsi- fungsi pada website yang akan dibuat agar bisa digunakan. | *Version*  *3.1.11* |
| 6. | *Boostrap* | Merupakan *framework* dari CSS yaitu Bahasa pemrograman yang bersifat *frontend*,  digunakan untuk membuat  tampilan pada *website* yang akan dibuat. | *Version 1.41* |
| 7. | *Xampp* | Merupakan sebuah aplikasi yang digunakan sebagai *local* *server*, yaitu sebuah simulasi untuk menjalankan *website*  yang sedang dibuat sebelum dihosting pada *server* nyata. | *Version*  *7.3.12* |
| 8. | *Maria DB* | Merupakan *database* versi lanjutan dari MySQL yang  digunakan untuk menyimpan  dann mengintegrasikan datadata pada *website* yang akan dibuat. | *Version*  *10.5.0* |
| 9. | *Android OS* | Merupakan Sistem Operasi yang digunakan untuk menjalankan layanan mobile. | *Marsmello* |

#### 4.4 Deksripsi Sistem

Aplikasi Manajemen Arsip PBL yaitu sistem yang berfungsi untuk mengelola penyimpanan arsip izin secara digital dan untuk mempermudah pencarian dan penemuan kembali arsip yang dibutuhkan. Adapun aktor yang terlibat di dalam sistem tersebut, yaitu :

1. Operator, Operator menjadi admin dalam sistem dengan deskripsi kerja mengelola dan mengatur *user* serta manajemen informasi PBL.
2. Pembimbing Lapangan, mempunyai deksripsi kerja melakukan peng*input* an dan mengelola arsip PBL.
3. Pimpinan, mempunyai deskripsi kerja melihat laporan rekapitulasi PBL.
4. Mahasiswa, mempunyai dekripsi kerja melihat list data arsip dan meng*input*  arsip ketika telah memiliki isin dari Operator.

Adapun prosedur dalam membangun Aplikasi Manajemen Berkas PBL antara lain sebagai berikut:

1. Aplikasi dapat digunakan tanpa login hanya untuk melihat dan melakukan login Sebagai *actor* untuk menggunakan aplikasi.
2. Di dalam sistem ini terdapat 4 area *user* yang berbeda yaitu Operator, Pembimbing Lapangan, Mahasiswa dan Pimpinan.
3. Di dalam Area Operator, Operator mampunyai hak akses untuk mengatur *user* dan hak ases *user* serta mengelola informasi PBL.
4. Di dalam Area Pembimbing Lapangan, Pembimbing Lapangan mempunyai hak akses untuk meng*input*  arsip PBL.
5. Di dalam Area Pimpinan, Pimpinan mampunyai hak akses untuk melihat Selur.
6. Di dalam Area Mahasiswa, *User* mempunyai hak akses untuk meilhat data arsip PBL dan ketika diberi izin oleh operator.
7. Ke empat aktor diatas dapat menggunakan akses pencarian data sesuai dengan masing-masing tugas pokok dan fungsi masing-masing.

#### 4.5 Analisis Perancangan UML

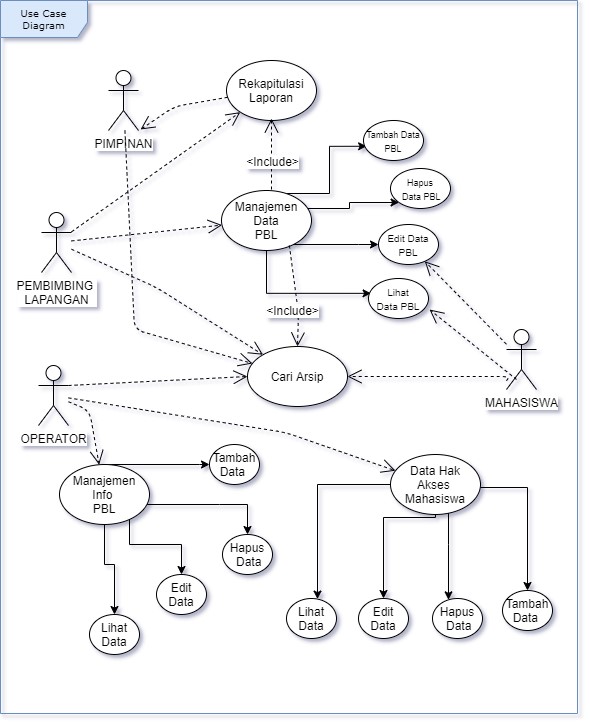
Analisis perancangan sistem yang akan dibangun meliputi, analisis perancangan UML, analisis Algoritma*,* dan peracangan *user interface.*Aplikasi dibangun dengan menggunakan *Unified Modeling Language* (UML). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram yang terdiri dari *Use Case Diagram, Class Diagram, Activty Diagram, State Diagram* dan *Sequence Diagram.*

##### 4.5.1. Use case diagram

*Use case diagram,* digunakan untuk memodelkan fungsionalitas-

fungsionalitas sistem yang dilihat dari pengguna yang ada di luar sistem (aktor).

Berikut adalah *use case* *diagram* ditunjukan pada gambar 4.1 :

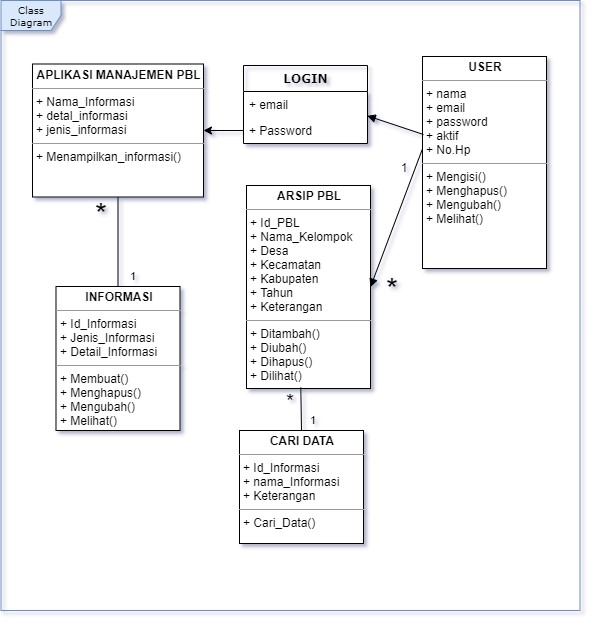


### Gambar 4.1 Use Case Diagram

#### 4.5.2 Class Diagram

*Class* juga memiliki 3 area pokok (utama) yaitu : nama,atribut,dan operasi. Nama berfungsi untuk *member* identitas pada sebuah kelas, atribut fungsinya adalah untuk member karakteristik pada data yang dimiliki suatu objek di dalam kelas, sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek .

Adapun*class diagram* dalam pembuatan Aplikasi manajemen berkas PBL (Praktek Bimbingan Lapangan) ini ditunjukan pada gambar 4.2 :

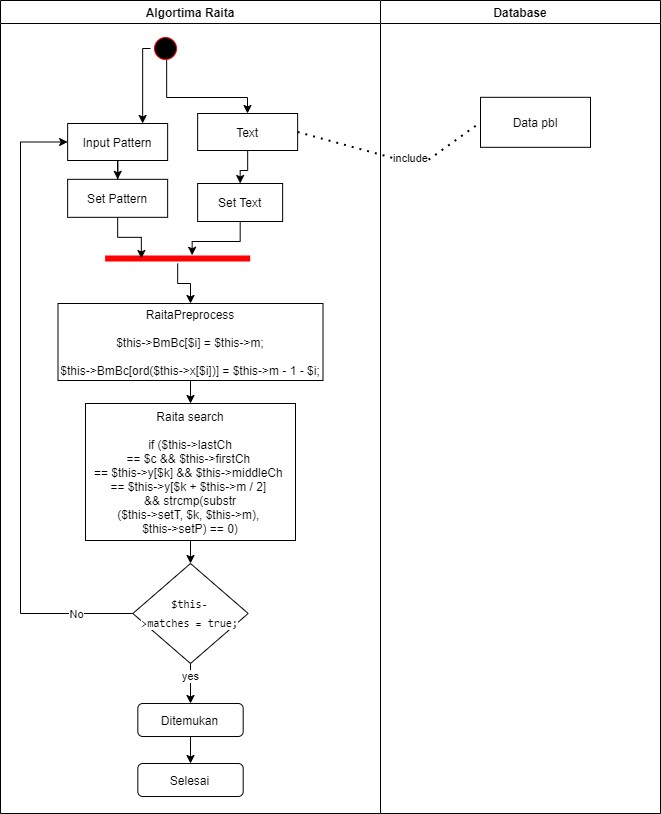


### Gambar 4.2 Class Diagram

#### 4.5.3 Activity Diagram

*Activity Diagram* adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. **a. *Activity Diagram* Pencarian**

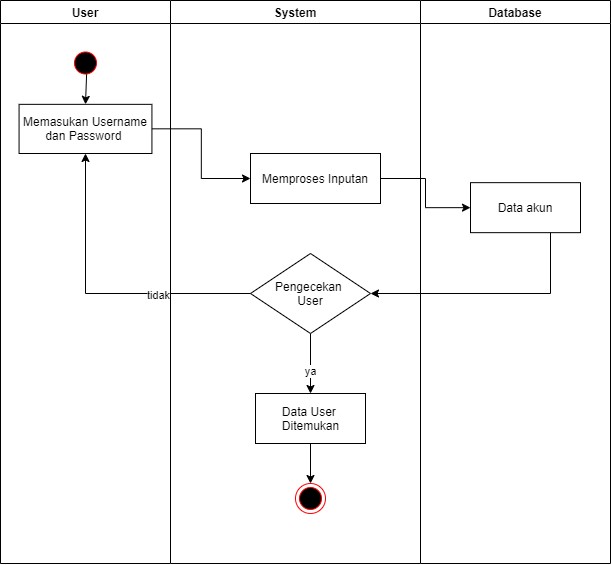
*Activity Diagram* Pencarian adalah sebuah aktivitas yang menggambarkan proses pencarian data menggunakan *string matching* algoritma raita. maka didapatkan rancangan yang ditunjukkan oleh Gambar 4.3 :



### Gambar 4.3 Activity Diagram Pencarian

#### b *Activity Diagram* Login

*Activity Diagram* Login adalah sebuah aktivitas yang menggambarkan proses masuk kedalam sebuah system berdasarkan 4 user yaitu pimpinan, pembimbing lapangan, operator, dan mahasiswa. maka didapatkan rancangan yang ditunjukkan oleh Gambar 4.4:

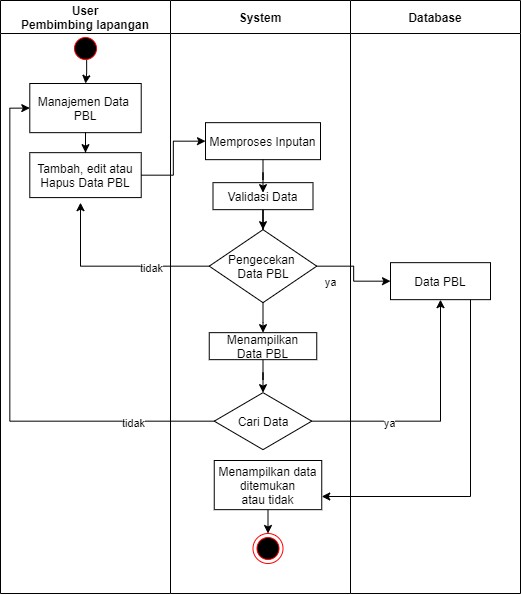


### Gambar 4.4 Activity Diagram Login

#### c *Activity Diagram* Manajemen Data PBL

*Activity Diagram* Manajemen Data PBL adalah sebuah aktivitas yang menggambarkan proses manajemen data PBL mulai dari menambahkan data, menghapus serta mengedit data sebagai data baku aplikasi manajemen PBL.

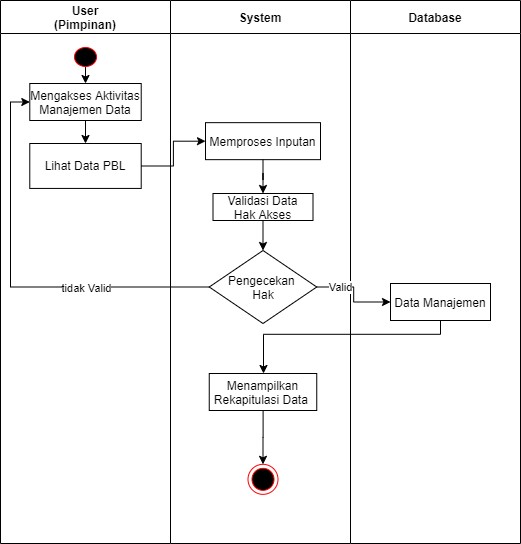
Sehingga didapatkan rancangan yang ditunjukkan oleh Gambar 4.5:



### Gambar 4.5 Activity Diagram Manajemen Data PBL

#### d Activity Diagram Rekapitulasi

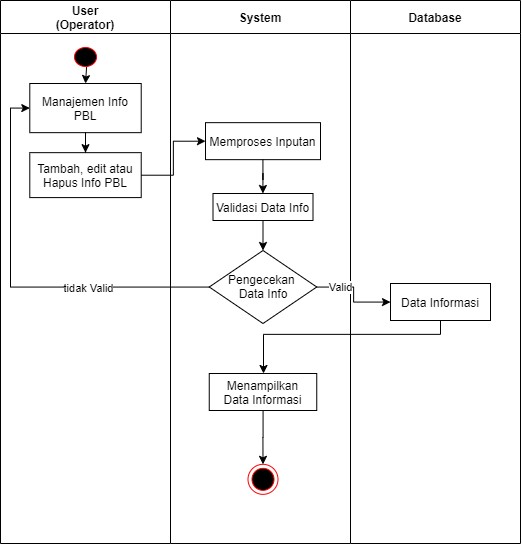
*Activity Diagram* Rekapitulasi Data PBL adalah sebuah aktivitas yang menggambarkan proses manajemen semua aktivitas yang berada dalam system serta mampu melihat hail rekapitulasi jumlah laporan dan jumlah mahasiswa yang telah memiliki hak akses untuk melihat data PBL. Sehingga didapatkan rancangan yang ditunjukkan oleh Gambar 4.6:



**Gambar 4.6 Activity Diagram Pimpinan**

### e Activity Diagram Manajemen Info PBL

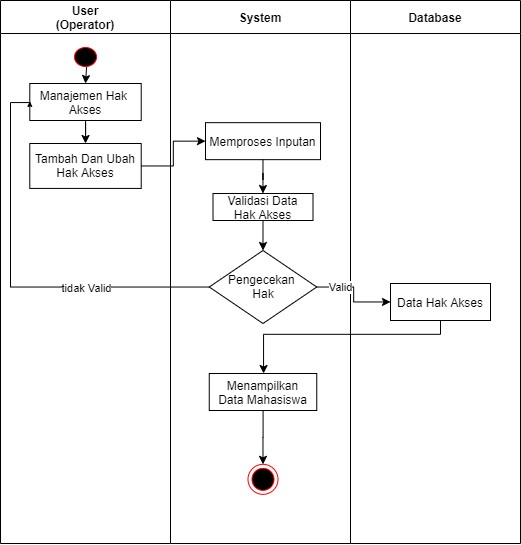
*Activity Diagram* Manajemen Info PBL adalah sebuah aktivitas yang menggambarkan proses manajemen informasi mulai dari membuat informasi, mengedit informasi maupun menghapus informasi. Sehingga didapatkan rancangan yang ditunjukkan oleh Gambar 4.7:



**Gambar 4.7 *Activity* Diagram Manajemen Info PBL**

### f Activity Diagram Manajemen Hak Akses

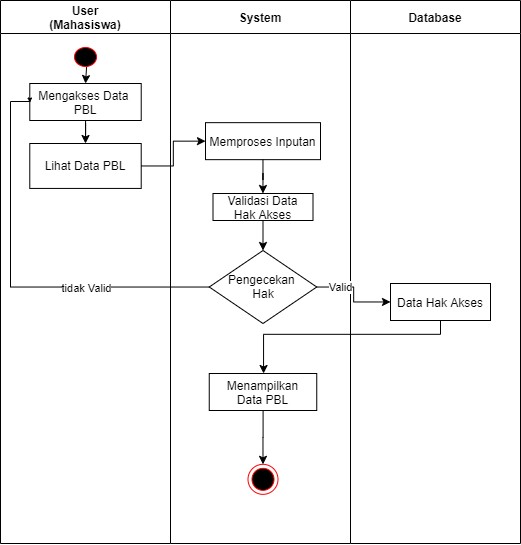
*Activity Diagram* Manajemen Hak Akses adalah sebuah aktivitas yang menggambarkan proses manajemen Hak akses mulai dari membuat akun mahasiswa, menghapus akun mahasiswa, dan mengedit akun serta memberi akses apakah mahasiswa diaktifkan atau di nonaktifkan. Sehingga didapatkan rancangan yang ditunjukkan oleh Gambar 4.8:



### Gambar 4.8 *Activity* Diagram Manajemen Hak Akses

#### g *Activity Diagram* Menu Mahasiswa

*Activity Diagram* Menu Mahasiswa adalah sebuah aktivitas yang menggambarkan proses aktivitas mahasiswwa dari melihat informasi serta melihat daftar laporan PBL yang berada di dalam system. Sehingga didapatkan rancangan yang ditunjukkan oleh Gambar 4.9:

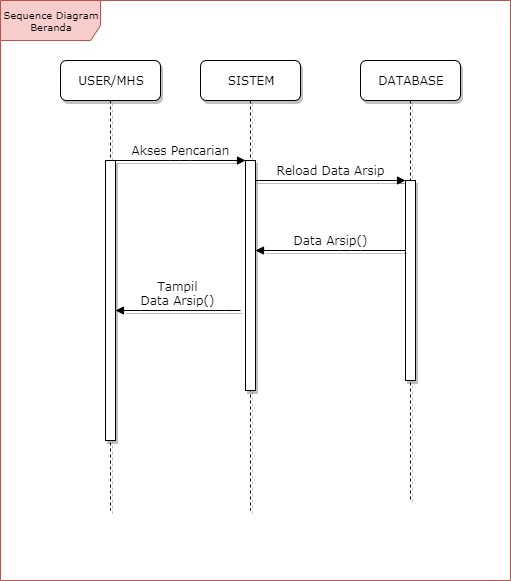


**Gambar 4.9 *Activity* Diagram *Menu Mahasiswa***

**4.5.4 Sequence Diagram**

#### a. *Sequence User* (Mahasiswa)

Setelah menganalisis sistem, maka didapatkan rancangan diagram *sequance* *user* yang ditunjukkan oleh Gambar 4.10 :



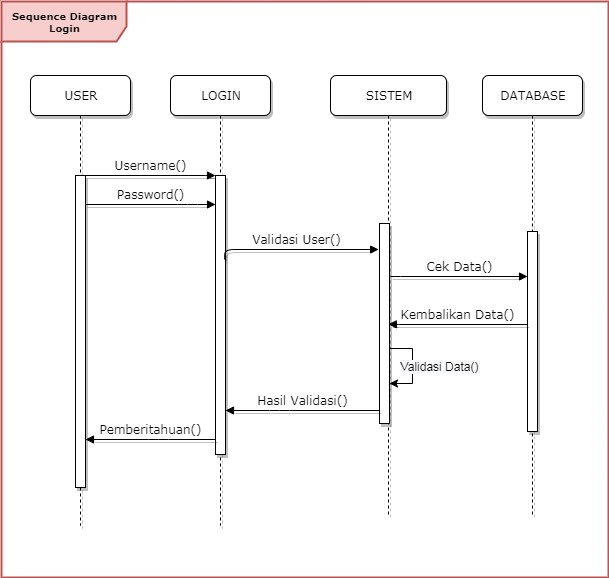
### Gambar 4.10 Sequence Diagram Beranda User

Adapun penjelaasan alur kerja *Sequence Diagram User* (Mahasiswa) adalah sebagai berikut :

1. Mahasiswa langsung mengakses aplikasi tanpa login
2. Setelah aplikasi terbuka mahasiswa dapat melihat seluruh informasi , dan dapat mencari berkas-berkas PBL**.**

#### b. *Sequence Login* 1*. Sequence Login User*

Setelah menganalisis *Sequence Diagram User*, maka didapatkan rancangan diagram *Sequance* *login user* yang ditunjukkan oleh Gambar 4.11 :



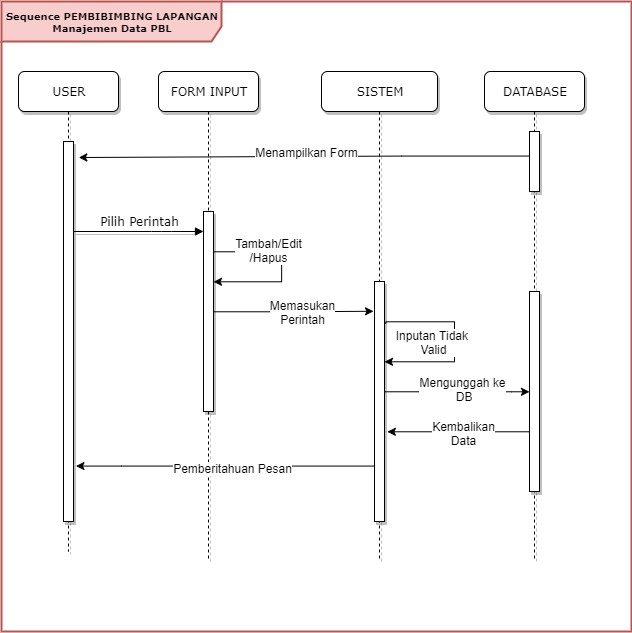
### Gambar 4.11 Sequence Diagram Login User

Adapun penjelaasan alur kerja *Sequence Diagram Login User* adalah sebagai berikut :

1. *User* memasukkan *id* dan *password*.
2. Data akan divalidasi oleh sistem dengan mengecek kecocokan data yang ada pada *database*. Jika data *valid*, maka sistem akan menampilkan halaman beranda . Namun jika data tidak *valid*, maka sistem akan menampilkan pesan *error*, yang meminta *admin* memasukkan ulang *admin id* dan *password*.

#### c*. Sequence Diagram* Pembimbing Lapangan 1. *Sequence* Pembimbing Lapangan (Manajemen Data PBL)

Setelah didapatkan rancangan diagram *Sequance* *login user*, maka didapatkan rancangan *Sequence* *Diagram* Pembimbing Lapangan (Manajemen Data PBL) yang ditunjukkan oleh Gambar 4.12 :



### Gambar 4.12 Sequence Diagram Manajemen Data PBL

Adapun penjelaasan alur kerja *Sequence* Diagram Pembimbing Lapangan (Manajemen Data PBL) adalah sebagai berikut :

1. *Admin* memasukkan p*erintah* Manajemen *(*Tambah, Edit, Hapus, Lihat*).*
2. Data akan diproses dan divalidasi oleh sistem dengan mengecek form yang kosong serta kecocokan data yang ada pada *database*. Jika data *valid*, maka sistem akan menampilkan pesan sukses dan kembali ke beranda. Namun jika

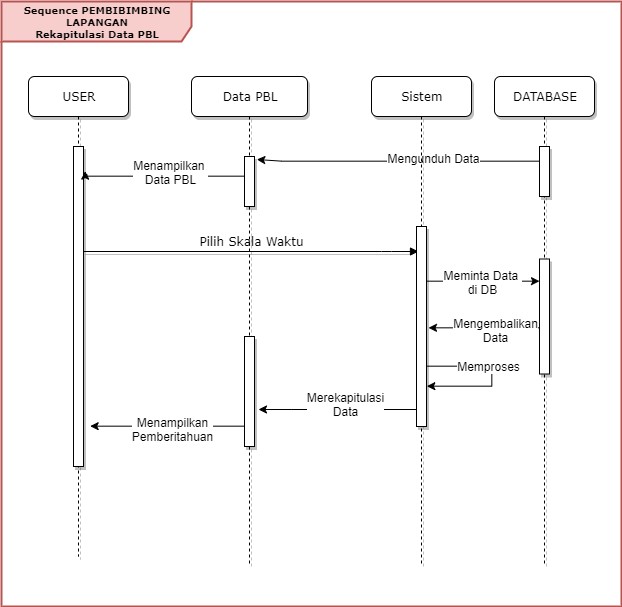
data tidak *valid* kembali ke halaman beranda

#### 2*. Sequence* Pembimbing Lapangan(Rekapitulasi Laporan)

Setelah menganalisis *Sequence* *Diagram* Pembimbing Lapangan

(Manajemen Data PBL) , maka didapatkan rancangan diagram *Sequence*

Pembimbing Lapangan(Rekapitulasi Laporan)yang ditunjukkan oleh Gambar 4.13



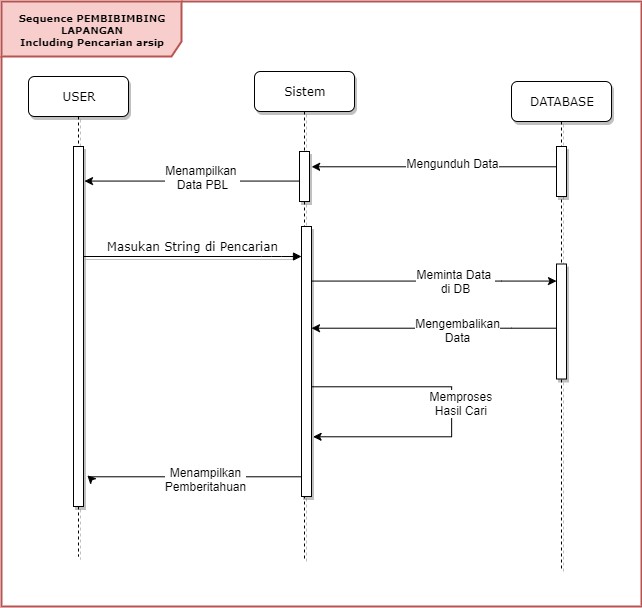
### Gambar 4.13 Sequence Diagram Rekapitulasi Laporan

Adapun penjelaasan alur kerja *Sequence* *Diagram* Pembimbing Lapangan(Rekapitulasi Laporan) adalah sebagai berikut :

1. Pembimbing lapangan memasukkan perintahlihat rekapitulasi PBL*.*
2. Data akan diproses dan divalidasi oleh sistem dengan mengecek *form* yang kosong serta kecocokan data yang ada pada *database*. Jika data *valid*, maka sistem akan menampilkan pesan sukses dan kembali ke beranda. Namun jika data tidak *valid* kembali ke halaman beranda.

#### 3*. Sequence* Pembimbing Lapangan(Pencarian Arsip)

Setelah menganalisis *Sequence* *Diagram* Pembimbing Lapangan (Rekapitulasi Laporan), maka didapatkan rancangan diagram *Sequence* Pembimbing Lapangan(Pencarian Arsip)yang ditunjukkan oleh Gambar 4.14 :



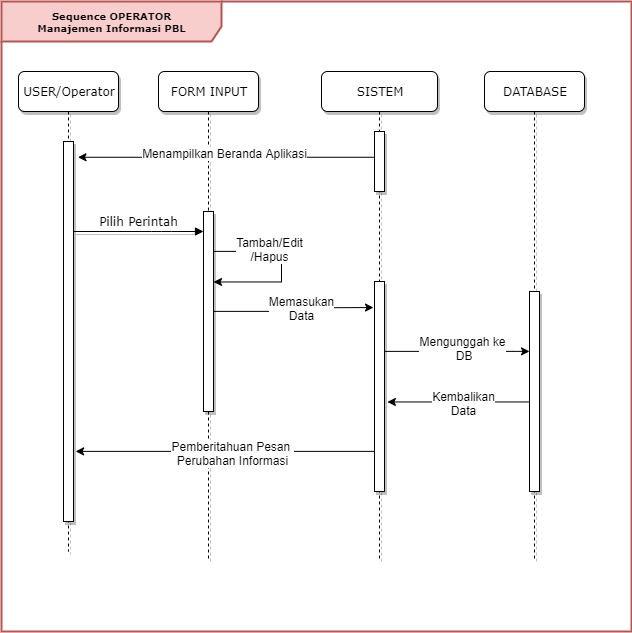
### Gambar 4.14 Sequence Diagram Pembimbing Lapangan (Pencarian Arsip)

Adapun penjelaasan alur kerja *Sequence* *Diagram* Pembimbing Lapangan (Manajemen Data PBL) adalah sebagai berikut :

1. Pembimbing lapangan mengisi form pencarian*.*
2. Data akan diproses dan divalidasi oleh sistem dengan mengecek form yang kosong serta kecocokan data yang ada pada *database*. Jika data *valid*, maka sistem akan menampilkan pesan sukses dan kembali ke beranda. Namun jika data tidak *valid* kembali ke halaman beranda

#### d*. Sequence Diagram* Operator 1. *Sequence* Operator (Manajemen Informasi PBL)

Setelah menganalisis *Sequence* Pembimbing Lapangan, maka didapatkan rancangan diagram *Sequance* Operator (Manajemen Informasi PBL)yang ditunjukkan oleh Gambar 4.15 :



### Gambar 4.15 Sequence Diagram Manajemen Informasi PBL

Adapun penjelaasan alur kerja *Sequence* *Diagram* Pembimbing Lapangan (Manajemen Data PBL) adalah sebagai berikut :

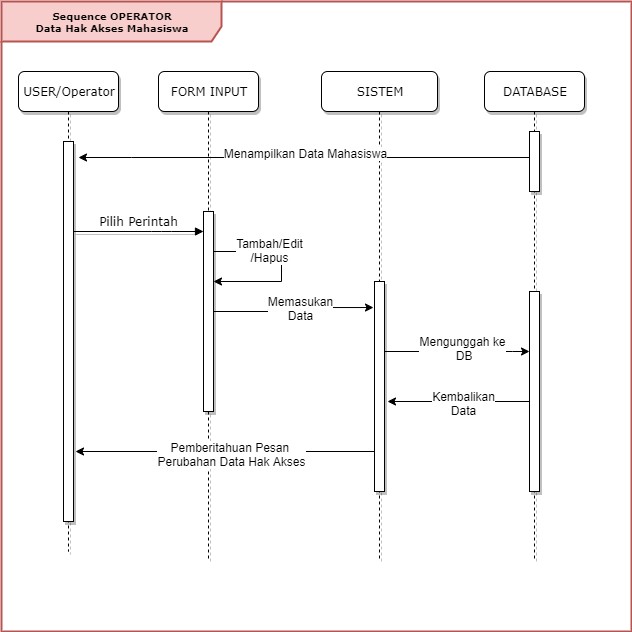
1. *Admin* memasukkan p*erintah* Manajemen *(*Tambah,edit,hapus,lihat*).*
2. Data akan diproses dan divalidasi oleh sistem dengan mengecek form yang kosong serta kecocokan data yang ada pada *database*. Jika data *valid*, maka

*valid*, maka sistem akan kembali menampilkan pesan gagal dan

kembali ke halaman beranda.

#### 2. *Sequence* Operator (Data Hak Akses Mahasiswa)

Setelah menganalisis *Sequance* Operator (Manajemen Informasi PBL), maka didapatkan rancangan diagram *Sequance* Operator (Data Hak Akses Mahasiswa) yang ditunjukkan oleh Gambar 4.16 :



### Gambar 4.16 Sequance Operator (Data Hak Akses Mahasiswa)

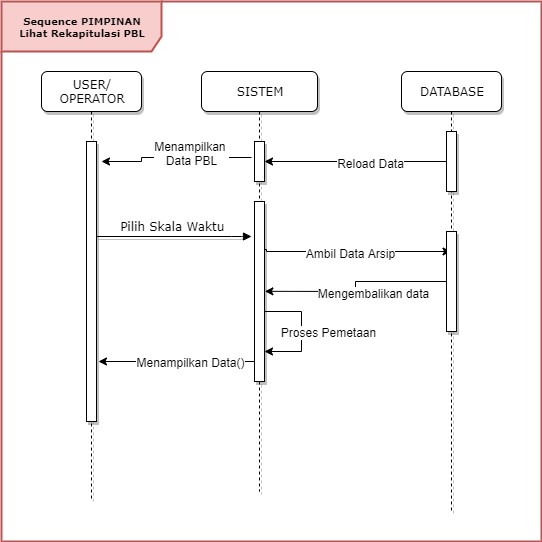
Adapun penjelaasan alur kerja *Sequence* Operator (Data Hak Akses Mahasiswa) adalah sebagai berikut :

1. *Admin* memasukkan p*erintah* Manajemen *(*Tambah,edit,hapus,lihat*).*
2. Data akan diproses dan divalidasi oleh sistem dengan mengecek form yang kosong serta kecocokan data yang ada pada *database*. Jika data *valid*, maka *valid*, maka sistem akan kembali menampilkan pesan gagal.

#### e*. Sequence Diagram* Pimpinan

##### 1. *Sequence* Pimpinan (Lihat Rekapitulasi PBL)

Setelah menganalisis *Sequence* *Diagram* Operator, maka didapatkan rancangan diagram *Sequence* Pimpinan (Lihat Rekapitulasi PBL)yang ditunjukkan oleh Gambar 4.17



### Gambar 4.17 Sequence Pimpinan (Lihat Rekapitulasi PBL)

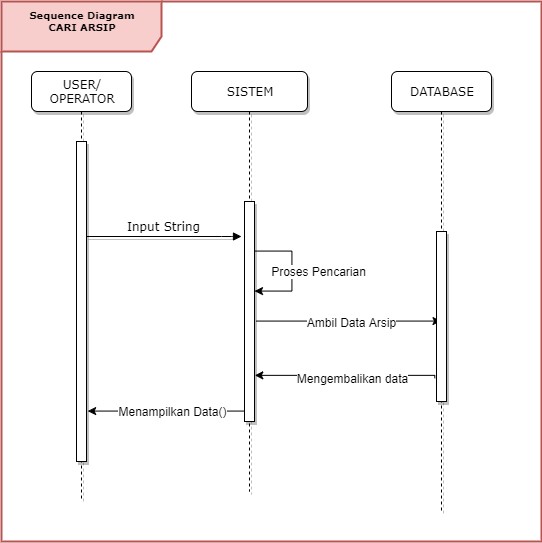
Adapun penjelaasan alur kerja *Sequence* Pimpinan (Lihat Rekapitulasi PBL) adalah sebagai berikut :

1. Pimpinanmemasukkan p*erintah* lihat rekapitulasi PBL.
2. Data akan diproses dan divalidasi oleh sistem dengan mengecek form yang kosong serta kecocokan data yang ada pada *database*. Jika data *valid*, maka *valid*, maka sistem akan kembali menampilkan pesan gagal dan

kembali ke halaman beranda.

### f. *Sequence* Cari Arsip

Setelah menganalisis *Sequence Diagram* Pimpinan, Maka didapatkan rancangan diagram *sequance* pemilihan dosenyang ditunjukkan oleh Gambar 4.18



### Gambar 4.18 Cari Arsip

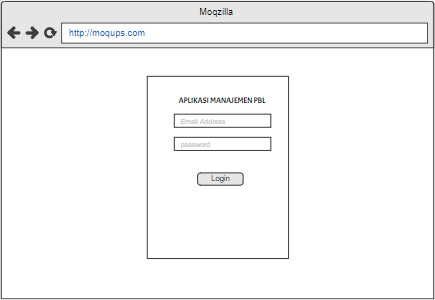
Adapun penjelaasan alur kerja *Sequence Diagram* Cari Arsip yadalah sebagai berikut :

1. *User* membuka pilihan menu yang akan dicari.
2. *User* memasukan string di kolom pencarian.
3. Jika proses pencarian berhasil maka pemberitahuan berhasil maka data berhasl di temukan, jika gagal maka pemberitahuan gagal dan data tidak ditemukan.

#### 4.6 Perancangan *Desain* *Interface*

Desain Antarmuka Pengguna (*User Interface Design*) atau rekayasa antarmuka pengguna adalah desain untuk komputer, peralatan, mesin, perangkat komunikasi mobile, aplikasi perangkat lunak, dan situs *web* yang berfokus pada pengalaman pengguna (*User* *Experience*) dan interaksi. **a. *Login* *User***

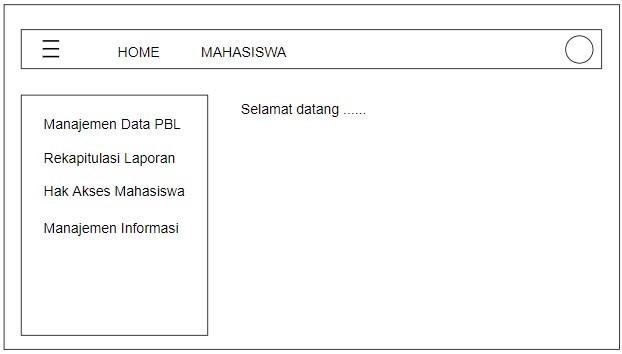
*Login User* merupakan rancangan antar muka *form* login tampilan *web* yang didalamnya berisi *textfield* *username* dan *textfield* *password*, serta tombol *login* untuk masuk kedalam sistem ditunjukan oleh gambar 4.19



**Gambar 4.19 Login User**

### b. Tampilan Beranda

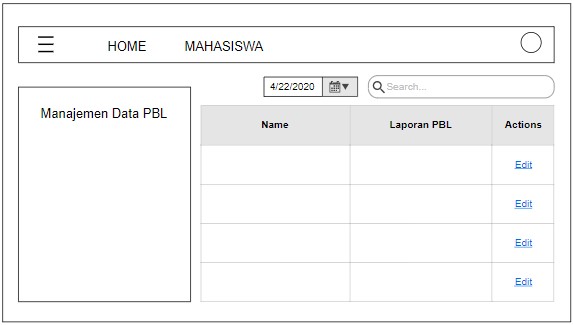
Tampilan Beranda merupakan rancangan antar muka *beranda* tampilan *web* yang didalamnya berisi tampilan selamat datang dan beberapa menu yang dapat diakses oleh *user* yang telah *login*. Berikut rancangan desain *interface* tampilan beranda ditunjukan oleh gambar 4.20:



### Gambar 4.20 Tampilan Beranda

#### c. Tampilan Manajemen Data PBL

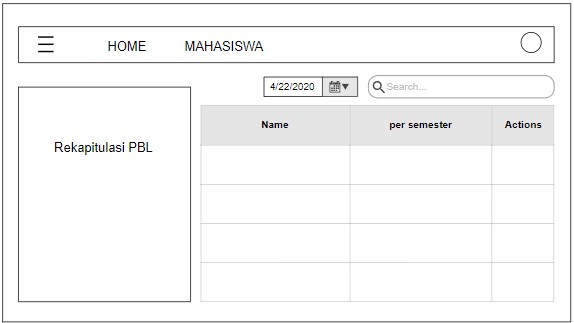
Tampilan Manajemen Data PBL merupakan rancangan antar muka manajemen data PBL tampilan *web* yang didalamnya berisi tabel data PBL dan tombol *search* untuk pencarian data PBL. Berikut rancangan desain *interface* tampilan manajemen data PBL ditunjukan oleh gambar 4.21



### Gambar 4.21 Tampilan manajemen data PBL

#### d. Tampilan Rekapitulasi Data PBL

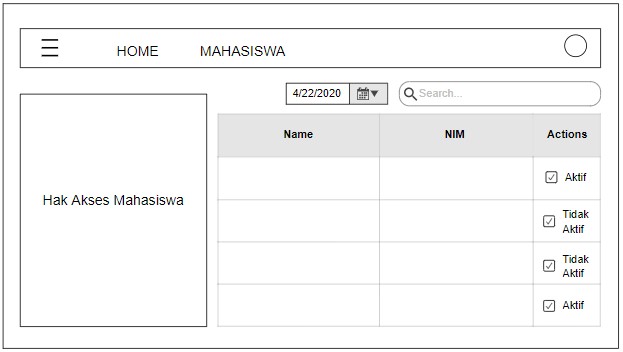
Tampilan Rekapitulasi Data PBL merupakan rancangan antar muka tampilan rekapitulasi data PBL tampilan *web* yang didalamnya berisi *tabel* data informasi PBL dan tombol *search* untuk pencarian data PBL. Berikut rancangan desain *interface* tampilan rekapitulasi data PBL ditunjukan oleh gambar 4.22



**Gambar 4.22 Tampilan rekapitulasi data PBL**

### e. Tampilan Hak Akses Mahasiswa

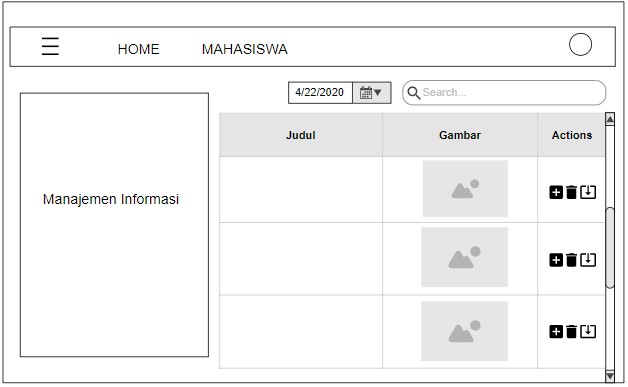
Tampilan Hak Akses Mahasiswa merupakan rancangan antar muka tampilan hak akses mahasiswa tampilan *web* yang didalamnya berisi tabel data akses mahasiswa serta checkbox sebagai perintah untuk mengaktifkan atau tidak dan tombol *search* untuk pencarian data mahasiswa. Berikut rancangan desain *interface* tampilan hak akses mahasiswa ditunjukan oleh gambar 4.23 :



**Gambar 4.23 Tampilan hak akses mahasiswa**

### f. Tampilan Manajemen Informasi

Tampilan Manajemen Informasi merupakan rancangan antar muka tampilan manajemen informasi PBL tampilan *web* yang didalamnya berisi *tabel* data informasi PBL serta *icon add, delete, update, view* sebagai perintah untuk mengelola data dan tombol *search* untuk pencarian data PBL. Berikut rancangan desain tampilan manajemen informasi PBL ditunjukan oleh gambar 4.23 :



### Gambar 4.23 Tampilan manajemen informasi PBL

#### 4.7 Rancangan *Black Box*

Rancangan Perangkat lunak merupakan proses rancangan serta mengeksekusi suatu *program* atau sistem dengan maksud menemukan atau, melibatkan setiap kegiatan yang bertujuan untuk mengevaluasi atribut atau kemampuan suatu *program* atau sistem dan menentukan bahwa itu memenuhi hasil yang dibutuhkan sebuah instansi yang menggunakan.

Penelitian ini menggunakan metode *Equivalence Partitions*, hal ini dilakukan untuk mendapatkan dataset, kemudian dataset ini akan dihitung efektifitasnya.Adapun model Rancanganperangkat lunak menggunakan *Black Box* untuk mendeteksi kesalahan pada fungsi dapat dilihat pada Tabel 3.4

### Tabel 4.3 Rancangan Black Box

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***No*** | **Nama *Form*** | **Detail Uji** | **Output** | **Keterangan hasil uji** |
| 1 | Pengujian Login | Menu Login | 0 | Menggunakan *Blackbox* |
| 2 | Pengujian Menu  Pembimbing  Lapangan | Manajemen PBL  (Tambah Data) | 0 | Menggunakan *Blackbox* |
| Manajemen PBL  (Edit Data) | 0 | Menggunakan *Blackbox* |
| Manajemen PBL  (Hapus Data) | 0 | Menggunakan *Blackbox* |
| Manajemen PBL  (Lihat Data) | 0 | Menggunakan *Blackbox* |
| Rekapitulasi  Laporan | 0 | Menggunakan *Blackbox* |
| Cari Data | 0 | Menggunakan *Blackbox* |
| 3 | Pengujian Menu  Operator | Manajemen Hak  Akses  (Tambah  Mahasiswa) | 0 | Menggunakan *Blackbox* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Manajemen Hak  Akses  (Edit  Mahasiswa) | 0 | Menggunakan *Blackbox* |
| Manajemen Hak  Akses  (Hapus  Mahasiswa) | 0 | Menggunakan *Blackbox* |
| Manajemen Hak  Akses  (Lihat  Mahasiswa) | 0 | Menggunakan *Blackbox* |
| Manajemen  Informasi  (Tambah Info) | 0 | Menggunakan *Blackbox* |
| Manajemen  Informasi  (Edit Info) | 0 | Menggunakan *Blackbox* |
| Manajemen  Informasi  (Hapus Info) | 0 | Menggunakan *Blackbox* |
| Manajemen  Informasi  (Lihat Info) | 0 | Menggunakan *Blackbox* |
| Cari arsip | 0 | Menggunakan *Blackbox* |
| 4 | Pengujian Menu  Pimpinan | Lihat  Rekapitulasi  Laporan PBL | 0 | Menggunakan *Blackbox* |
| 5 | Pengujian Menu  Mahasiswa | Lihat Data PBL | 0 | Menggunakan *Blackbox* |
| Edit Data PBL | 0 | Menggunakan *Blackbox* |
| Cari Arsip | 0 | Menggunakan *Blackbox* |

**DAFTAR PUSTAKA**

A.S.. Rosa Dan Shalahuddin. M. 2013. Rekayasa Perangkat Lunak Terstruktur Dan B*error* Ientasi Objek. Informatika. Bandung.

Baharudin. R. (2010). Keefektifan Media Belajar Berbasis Teknologi Informasi Dan Komunikasi. Tadrîs. 5(1) . 112–127. Charras. C. & Lecroq. T. 2004. Handbook Of Exact String-Maching Algorithms. College Publications.

Gelar. M.. Komputer. S.. & Informatika. J. T. (2011). *Pengujian Perangkat Lunak Menggunakan Metode Black Box Studi Kasus Exelsa Universitas Sanata Dharma*.

Indrayana. I. N. E. (2017). Perancangan Aplikasi Sistem Pengelolaan Arsip Jurusan

Teknik Elektro Politeknik Negeri Bali. *Logic : Jurnal Rancang Bangun Dan*

*Teknologi*. *16*(2). 75. Https://Doi.Org/10.31940/Logic.V16i2.156

Kom. S.. Kom. M.. Fauzi. R.. Kom. S.. Kom. M.. Studi. P.. & Informatika. T. (2018). *Universitas Putera Batam*.

Lumbanraja. H. D. (N.D.). *Perancangan Sistem Informasi Akademik Online Menggunakan Black Box Testing Pada Sekolah Design Of Online Academic Information Sistem Using Black Box Testing At Sekolah Tinggi Ilmu*. 9–18.

Manalu. D. R. (2014). *Perancangan Aplikasi Kelola Berkas Digital Kampus*. *4*. 53–

63.

Marsudi. D. (2016). Politeknik Negeri Sriwijaya 4. *Pembangkitan Energi Listrik*. *7*(1). 4–31.

Prabowo. D. (2015). Website E-Commerce Menggunakan Model View Controller

( Mvc ) Dengan Framework Codeigniter Studi Kasus : Toko Miniatur. *Data Manajemen Dan Teknologi Informasi (Dasi)*. *16*(1). 23.

Rouf. A. (2012). *Pengujian Perangkat Lunak Dengan Menggunakan Metode White*

*Box Dan Back Box*. *Vol 8 No1*. 1–7. Retrieved From

Http://Www.Ejournal.Himsya.Ac.Id/Index.Php/Himsyatech/Article/View/28/ 27

Rustam. M. (2014). *Pengantar Pengelolaan Arsip Elektronik*. 1–85.

Sutanto. C.. Listyorini. T.. Jazuli. A.. Studi. P.. Informatika. T.. Teknik. F.. & Kudus. U. M. (2015). *Aplikasi Web. Arsip Dinamis. Yii Framework. 1.* 377– 380.

Sommerville. Ian. 2011. Software Engineering (Rekayasa Perangkat Lunak). Jakarta: Erlangga.

Warman. I.. & Ramdaniansyah. R. (2018). Analisis Perbandingan Kinerja Query

Database Management Sistem (Dbms) Antara Mysql 5.7.16 Dan Mariadb

10.1. *Jurnal Teknoif*. *6*(1). 32–41.

Https://Doi.Org/10.21063/Jtif.2018.V6.1.32-41