

**APLIKASI REPOSITORY MANAJEMEN DATA SKRIPSI  
MENGUNAKAN NETBEANS  
STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI**

**PROPOSAL SKRIPSI**

Usulan Penelitian sebagai syarat mengikuti Skripsi

Disusun oleh:  
**JERI RISMAWAN**  
NPM. 311150003



**PROGRAM STRATA 1  
PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG  
BANDUNG**

2019

**LEMBAR PENGESAHAN**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**APLIKASI REPOSITORY MANAJEMEN DATA SKRIPSI  
MENGUNAKAN NETBEANS  
STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI**

Disusun Oleh:  
JERI RISMAWAN / 311150003

disetujui dan disahkan sebagai  
Proposal Skripsi

Bandung, 16 Maret 2019  
Ketua Program Studi Teknik Informatika

Yaya Suharya, S.Kom, M.T  
NIDN. 0407047706

**LEMBAR PENGESAHAN**

**FAKULTAS TEKNOLOGI INFORMASI**

**APLIKASI REPOSITORY MANAJEMEN DATA SKRIPSI  
MENGUNAKAN NETBEANS  
STUDI KASUS DI FAKULTAS TEKNOLOGI INFORMASI**

Disusun Oleh:  
JERI RISMAWAN / 311150003

disetujui dan disahkan sebagai  
Proposal Skripsi

Bandung, 16 Maret 2019  
Dekan FTI UNIIBBA

Yudi Herdiana, ST.MT  
NIK. 0428027501

## **KATA PENGANTAR**

Segala puji bagi ALLAH SWT, yang telah memberikan rahmat dan hidayahnya kepada penulis, sehingga penulis dapat menyelesaikan pembuatan proposal skripsi dan dapat di selesaikan sesuai dengan yang penulis harapkan.

Dalam penyusunan proposal , penulis mendapat banyak bantuan, dukungan dan bimbingan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Yudi Herdiana S.T,.M.T. selaku Dekan Fakultas teknologi informasi.
2. Bapak Yaya Suharya S.Kom,.M.T.Selaku ketua prodi.
3. Bapak Mochammad Ridwan ST. M.Kom yang telah memberikan informasi serta masukan dalam penyusunan propoal.
4. Kedua orang tua yang selalu memberikan motifasi, dorongan serta mendoakan penulis.

Penulis juga menyadari bahwa di dalam penyusunan proposal ini terdapat banyak kekurangan dan kesalahan. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun, sehingga proposal penulis selanjutnya dapat menjadi lebih baik.

Bandung, Maret 2019

Hormat Saya

Penulis

## **ABSTRAK**

Teknologi komputer telah menghasilkan informasi yang lebih cepat, akurat dan lebih relevan bila dibandingkan dengan informasi yang dihasilkan dengan cara konvensional. Dengan perkembangan teknologi informasi seperti sekarang ini kehadiran komputer sangat membantu dalam proses pengolahan data. Komputer merupakan alat pengolah data yang tepat, cepat dan akurat dengan tingkat ketelitian yang tinggi sehingga mampu menjadi alternatif terbaik dalam memenuhi setiap kebutuhan informasi. Studi kasus yang diambil dalam proyek akhir ini yaitu membangun sebuah Aplikasi Repositoty Manajemen Data Skripsi di Fakultas Teknologi Informasi.

Sejalan dengan perkembangan teknologi, penyusun mendapati satu studi kasus yang menarik untuk dikaji di Fakultas Teknologi Informasi yaitu, Proses pengolahan data Skripsi di Fakultas Teknologi informasi belum tersedianya User Interface sebagai media komunikasi berkas digital Sehingga timbul permasalahan seperti Sulitnya temu kenali dokumen baik dokumen berkas fisik maupun berkas digital, karena penyimpanan yang belum terindeks dan tidak tersedianya cloud storage yang berfungsi sebagai media penyimpanan berkas digital (Repository). Untuk itu penulis tertarik untuk mengembangkan sistem informasi manajemen khususnya dalam pembuatan Repository Manajemen Data Skripsi di Fakultas Teknologi Informasi menggunakan Netbeans 8.1 sebagai User interfacenya untuk membantu proses mengolah data Skripsi di Fakultas Teknologi Informasi. Metode yang digunakan dalam penulisan ini berupa penelitian studi pustaka, observasi dan wawancara dengan sumber penulisan. Dalam penulisan ini penulis merancang suatu sistem dengan menggunakan file database NoSQL dengan harapan permasalahan di atas dapat dikurangi, selain itu dapat meningkatkan aktifitas manajemen serta menambah motifasi kerja khususnya dalam informasi di bagian perpustakaan, sehingga data yang diperlukan yang berhubungan dengan data skripsi dapat diketahui dengan cepat.

Kata kunci: Repository , Skripsi, Java.

## DAFTAR ISI

|  |     |
|--|-----|
| KATA PENGANTAR.....                              | iii |
| ABSTRAK .....                                    | iv  |
| DAFTAR ISI.....                                  | v   |
| BAB I PENDAHULUAN .....                          | 1   |
| 1.1    Latar Belakang .....                      | 1   |
| 1.2    Rumusan Masalah .....                     | 1   |
| 1.3    Batasan Masalah.....                      | 2   |
| 1.4    Tujuan Penelitian.....                    | 2   |
| BAB II TINJAUAN PUSTAKA.....                     | 3   |
| 2.1    Landasan Teori .....                      | 3   |
| 2.1.1    DatabaseNoSQL.....                      | 3   |
| 2.2    Dasar Teori .....                         | 3   |
| 2.2.1    JAVA .....                              | 3   |
| 2.2.2    JSON(Java Script Object Notation) ..... | 4   |
| 2.2.3    NetBeans IDE.....                       | 4   |
| 2.2.4    UML.....                                | 5   |
| BAB III METODOLOGI PENELITIAN.....               | 8   |
| 1.4    Waktu dan Tempat Penelitian .....         | 8   |
| 1.5    Alat dan Bahan .....                      | 8   |
| 1.6    Metode Penelitian.....                    | 8   |
| 1.7    Metode Pengumpulan Data .....             | 9   |
| BAB IV JADWAL PENELITIAN.....                    | 10  |
| DAFTAR PUSTAKA .....                             | vi  |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Skripsi/Tugas Akhir adalah tulisan ilmiah yang wajib ditulis oleh Mahasiswa sebagai bagian dari persyaratan akhir akademisnya. Dengan diajukan terlebih dahulu ke pihak Universitas khususnya pada setiap Fakultas dan Jurusannya.

FTI adalah salah Satu Fakultas yang terdapat di UNIBBA, Sekretariat Fakultas Teknologi Informasi yang berlokasi di Jl.R.A.A Wiranatakusuma No. 7 Baleendah. Pada divisi perpustakaan, lebih tepatnya pada pengelolaan management data skripsi ada beberapa pokok permasalahan yang ada pada sistem pengolahan data yang ada sekarang, yang diantaranya adalah:

1. Sulitnya temu kenali dokumen baik dokumen berkas fisik maupun berkas digital, karena penyimpanan yang belum terindeks
2. Tidak tersedianya cloud storage yang berfungsi sebagai media penyimpanan berkas digital (Repository).
3. Tidak tersedianya User Interface sebagai media komunikasi berkas digital.

Dengan permasalahan di atas, penulis mengambil solusi yaitu harus adanya metode penyimpanan data yang baik dan aplikasi yang dapat menyimpan data laporan skripsi.

Berdasarkan uraian di atas, penulis mengambil kesimpulan bahwa diperlukan adanya Aplikasi Repository pengolahan data Skripsi/Tugas Akhir yang harus dapat mengakomodasi permasalahan diatas.

### **1.2 Rumusan Masalah**

1. Bagaimana cara mengelola data-data laporan skripsi lebih cepat, tepat, dan akurat?
2. Bagaimana cara mengelola data menjadi laporan yang mudah di lihat pegawai Tata Usaha dan Dosen?
3. Bagaimana cara membuat aplikasi yang dapat mempermudah dalam proses pengarsipan data skripsi?

### **1.3 Batasan Masalah**

Batasan-batasan dari aplikasi ini adalah :

1. Aplikasi ini hanya sebatas pengimputan data data skripsi.
2. Aplikasi berbasis Dekstop dengan menggunakan NetBeans IDE 8.1.
3. Menggunakan standar format JSON (Java Script Object Notation)
4. Menggunakan database NoSQL.

### **1.4 Tujuan Penelitian**

Membangun aplikasi yang dapat :

1. Membangun Aplikasi Repository Manajement data skripsi di Fakultas Teknologi Informasi.
2. Mengelola inputan yang ada di dalam penyimpanan data.
3. Mengefisienkan proses pencarian data skripsi di Fakultas Teknologi Informasi Universitas Bale Bandung
4. Mengelola data menjadi laporan yang mudah di lihat pegawai Tata Usaha dan dosen.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### **2.1.1 DatabaseNoSQL**

Menurut Wikipedia NoSQL adalah istilah yang dikenal dalam teknologi komputasi untuk merujuk kepada kelas yang luas dari sistem manajemen basis data yang di identifikasikan dengan tidak mematuhi aturan pada model sistem manajemen basis data relasional yang banyak digunakan. NoSQL tidak dibangun terutama dengan table dan umumnya tidak menggunakan SQL untuk memanipulasi data, sehingga sering ditafsirkan sebagai “tidak hanya SQL” Selama berpuluh tahun, model data utama yang digunakan untuk pengembangan aplikasi adalah model data relasional yang digunakan oleh databaserelasional seperti Oracle, DB2, SQL Server, MySQL, dan PostgreSQL. Hingga pada pertengahan hingga akhir tahun 2000 model data lain mulai mendapatkan adopsi dan penggunaan yang signifikan. Untuk membedakan dan mengategorikan kelas database dan model data baru ini, istilah “NoSQL” diciptakan. Sering kali istilah “NoSQL” digunakan secara bergantian dengan “nonrelasional.”.

#### **2.2 Dasar Teori**

##### **2.2.1 JAVA**

Menurut Rickyanto (2003). Java adalah suatu teknologi di dunia software komputer. Selain merupakan suatu bahasa pemrograman, Java juga merupakan suatu platform. Java merupakan teknologi dimana teknologi tersebut mencakup java sebagai bahasa pemrograman yang memiliki sintaks dan aturan pemrograman sendiri. Juga mencakup java sebagai platform dimana teknologi ini memiliki virtual machine dan library yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemrograman Java.

Java merupakan suatu teknologi yang unik dan revolusioner dan merupakan teknologi pertama di dunia software yang memiliki semboyan “write once, run anywhere”. Semboyan tersebut telah terbukti karena banyak program java dapat dijalankan diberbagai platform sistem operasi seperti Linux, Windows maupun Unix. Adapun karakteristik Java menurut Rickyanto (2003) adalah:

- a. Sederhana: Java tidak memiliki sintaks yang aneh tetapi banyak menggunakan sintaks c++ yang sudah banyak dikenal, sehingga java tidak menyulitkan bagi para programmer. Bahkan java memberikan banyak keunggulan dan kemudahan dibanding C++
- b. Berorientasi objek: Java merupakan pemrograman berorientasi objek yang murni. Dalam pemrograman Java semua adalah objek, terkecuali tipe primitif.

Dapat didistribusikan Dengan mudah: Sifat distribusi dari Java sangat tampak sebagai applet dan library yang mampu bekerja dalam jaringan dan bekerja dengan objek terdistribusi (RMI) dengan sangat baik.

- c. Aman: Program Java memiliki library security serta policy yang membatasi akses applet di komputer client.
- d. Diinterpretasi oleh interpreter: Java memerlukan virtual machine yang bertindak sebagai interpreter yang menterjemahkan bytecode (file class) menjadi bahasa mesin yang dimengerti oleh komputer host.
- e. Portabel: Java dapat dijalankan diberbagai platform tanpa perubahan kode.
- f. Multi threading: Java memiliki banyak kemampuan untuk menangani dan menjalankan banyak thread sekaligus.
- g. Dinamis: Java merupakan teknologi yang terus berkembang dan hal ini tampak nyata sekali dengan library yang terus ditingkatkan kemampuan dan kelengkapannya.
- h. Netral terhadap arsitektur hardware: Java dapat dijalankan dengan baik pada komputer yang memiliki arsitektur berbeda-beda.
- i. Robust: Java merupakan teknologi yang mampu menolong programmer untuk menghasilkan program secara cepat dan handal karena Java mencegah adanya memori leaking, meniadakan pointer serta mencegah berbagai eror yang mungkin terjadi dengan adanya proses pengecekan awal pada kompilasi.

### **2.2.2 JSON(Java Script Object Notation)**

JSON adalah singkatan dari Javascript Object Notation atau Bahasa Indonesianya Notasi Objek Javascript. JSON juga bisa diartikan format file yang digunakan untuk transfer data baik membaca dan menulis. JSON sangat mudah bagi engine manapun untuk melakukan parsing dan men-generate sebuah data.

Banyak bahasa pemrograman yang mendukung JSON. JSON memiliki fungsi yang sama dengan XML yaitu berfungsi untuk merepresentasikan data. Jika dibanding dengan XML, JSON memiliki kelebihan yaitu mudah dibaca dan tidak perlu menulis tag JSON tetapi justru struktur JSON didefinisikan oleh data. Cara ini lebih baik dan lebih mudah difahami dan mempelajari JSON. Sumber: <https://www.dumetschool.com/blog/Pengertian-JSON>

### **2.2.3 NetBeans IDE**

Netbeans adalah aplikasi Integrated Development Environment (IDE) yang berbasiskan Java. Menurut Nofriadi (2015:4) "Netbeans Merupakan Sebuah aplikasi Integrated Development Environment (IDE) yang Menggunakan Bahasa Pemrograman Java dari Sun Microsystems yang berjalan diatas swing", sedangkan menurut [www.netbeans.org/index\\_id.html](http://www.netbeans.org/index_id.html) (2017:1) "NetBeans IDE adalah sebuah lingkungan pengembangan sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program.

Netbeans IDE ditulis dalam Bahasa Pemrograman Java, namun dapat mendukung bahasa pemrograman lain”. Dari kedua pendapat diatas dapat ditarik kesimpulan bahwa Netbeans adalah Suatu aplikasi untuk membuat suatu program yang dimana bahasa pemrograman yang di pakai adalah java, maupun bahasa pemrograman yang lain dengan mudahnya penggunaan aplikasi netbeans ini maka programmer sering menggunakan aplikasi ini sebagai wadah untuk membuat berbagai macam program yang dimana netbeans IDE ini bersifat Open Source.

#### **2.2.4 UML**

UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan singkatan dari Unified Modeling Language. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu developer ke developer lainnya. Tidak hanya antar developer terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML. Sumber <https://www.codepolitan.com/mengenal-diagram-uml-unified-modeling-language>

UML diciptakan oleh Object Management Group yang diawali dengan versi 1.0 pada Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal yaitu:

1. Object
2. Class
3. Abstraction
4. Encapsulation
5. Inheritance
6. Polymorphism

Dalam UML sendiri terdapat beberapa diagram yang wajib dikuasai yaitu:

1. Structure Diagrams
  - a) Class Diagram, diagram ini terdiri dari class, interface, association, dan collaboration. Diagram ini menggambarkan objek - objek yang ada di sistem.
  - b) Object Diagram, diagram ini menggambarkan hasil instansi dari class diagram. Diagram ini digunakan untuk membuat prototype
  - c) Component Diagram, diagram ini menggambarkan kumpulan komponen dan hubungan antar komponen. Komponen terdiri dari class, interface, atau collaboration
  - d) Deployment Diagram, diagram ini menggambarkan kumpulan node dan hubungan antar node. Node adalah entitas fisik dimana komponen di-

deploy. Entitas fisik ini dapat berupa server atau perangkat keras lainnya.

## 2. Behavioral Diagram

- a) Use case Diagram, diagram ini menggambarkan kumpulan use case, aktor, dan hubungan mereka. Use case adalah hubungan antara fungsionalitas sistem dengan aktor internal/eksternal dari sistem.
- b) Sequence Diagram, diagram ini menggambarkan interaksi yang menjelaskan bagaimana pesan mengalir dari objek ke objek lainnya.
- c) Collaboration Diagram, diagram ini merupakan bentuk lain dari sequence diagram. Diagram ini menggambarkan struktur organisasi dari sistem dengan pesan yang diterima dan dikirim.
- d) Statechart Diagram, diagram ini menggambarkan bagaimana sistem dapat bereaksi terhadap suatu kejadian dari dalam atau luar. Kejadian (event) ini bertanggung jawab terhadap perubahan keadaan sistem.
- e) Activity Diagram, menggambarkan aliran kontrol sistem. Diagram ini digunakan untuk melihat bagaimana sistem bekerja ketika dieksekusi.

UML umum digunakan di dunia kerja dan dunia pendidikan. Dunia kerja menggunakan UML untuk menggambarkan sistem dan aplikasi berbasis objek yang sudah ada sebelumnya atau merancang sistem dan aplikasi di awal pengembangan sebelum memasuki tahap coding. Beberapa posisi pekerjaan yang sering bersinggungan dengan UML antara lain system analyst, software architect, dan technical writer. System analyst selain perlu memiliki kemampuan coding, sudah tentu wajib menguasai UML, jika akan mengembangkan aplikasi atau sistem berbasis objek agar dapat menjadi acuan formal bagi para developer yang bekerja sama dengannya. Tidak hanya itu system analyst pun terkadang dituntut untuk menguasai pembuatan diagram lain seperti data flow diagram, flowchart, entity relationship diagram, dan lainnya. Sedangkan UML menjadi kemampuan wajib bagi seorang technical writer. Pekerjaan di posisi ini cukup banyak dibutuhkan oleh beberapa perusahaan yang kesulitan dalam membuat dokumentasi sistem atau aplikasi yang mereka kembangkan. Selain membuat user guide, technical writer diperlukan untuk membangun keseluruhan dokumentasi dalam UML dari aplikasi atau sistem, sehingga keberadaannya dapat menambah nilai aplikasi menjadi lebih tinggi.

UML pun digunakan sebagai alat dalam beberapa mata kuliah di perguruan tinggi yang membuka jurusan ilmu komputer, teknik informatika, sistem informasi, manajemen informatika, dan komputerisasi akuntansi. Mata kuliah yang biasa diberikan bernama Metodologi Berorientasi Objek dan Pemrograman Berorientasi Objek. Bahasa pemrograman yang digunakan umumnya Java, C#, dan PHP, kemudian mahasiswa akan mempelajari juga bagaimana merancang sebuah sistem atau aplikasi menggunakan UML terhadap suatu kasus dan membuat aplikasinya ketika memasuki tahap coding. Tidak hanya sebatas kuliah, beberapa perguruan tinggi yang mempunyai jurusan informatika, kadang mewajibkan penggunaan UML untuk digunakan sebagai alat bantu di skripsi, tesis, dan disertasi. Karena

tidak semua akademisi dapat membaca semua kode program yang datang dari berbagai bahasa pemrograman, maka UML menjadi sarana untuk memahami aplikasi dan sistem yang dikembangkan akademisi lainnya.

Berikut ini adalah beberapa alat yang membantu kita untuk menggunakan UML disebut dengan CASE (computer aided software engineering). Ada beberapa CASE yang umum digunakan oleh para profesional yang menggunakan UML, diantaranya:

- a) Sybase Power Designer
- b) Rational Rose
- c) Microsoft Visio
- d) Enterprise Architect

Ada juga beberapa CASE yang open source dan dapat digunakan secara bebas tanpa terbentur dengan masalah pembayaran lisensi seperti berikut:

- a) StarUML
- b) DIA Diagram Editor
- c) Umbrello
- d) ArgoUML
- e) Software Ideas Modeller
- f) Eclipse UML2

Beberapa alat yang berbayar mempunyai kemampuan untuk melakukan forward engineering dan reverse engineering. Forward engineering adalah sebuah fitur dari suatu CASE dimana UML yang telah dirancang dapat menghasilkan kode program yang akan menjadi pijakan dan panduan untuk mengawali pengembangan aplikasi. Sehingga koherensi antara aplikasi dan UML dapat terjaga. Reverse engineering adalah fitur dimana suatu CASE dapat membaca kode program dan menghasilkan UML dari kode program yang telah ada atau sedang dibuat.

### BAB III

## METODOLOGI PENELITIAN

### 1.4 Waktu dan Tempat Penelitian

Lokasi penelitian di laksanakan di Fakultas teknologi informasi Universitas Bale Bandung, waktu penelitian di lakukan mulai bulan maret 2019.

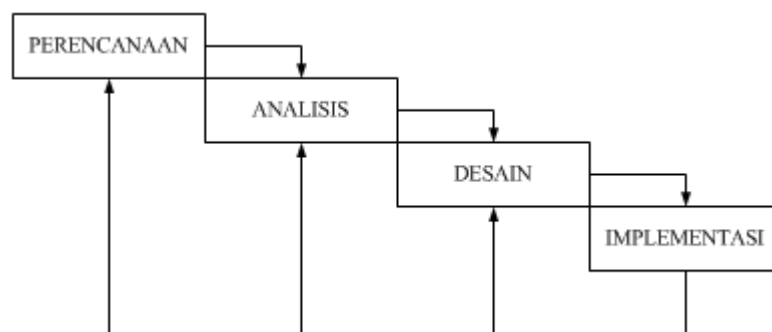
### 1.5 Alat dan Bahan

1. Perangkat Lunak (Software)
  - a. Windows 10 sebagai sistem operasi
  - b. NetBeans IDE 8.1 Sebagai Bahasa pemrograman
  - c. Database NoSQL Sebagai Database.
2. Perangkat Keras

|           |  |
|-----------|--|
| Laptop    | : Asus X454W Series  |
| Processor | : Intel (R) Coleron(R) CPU N3050 @1.60Ghz (2CPUs), ~1.6Ghz |
| Memory    | : 2048 MB RAM  |
| Harddisk  | : 500 GB   |
| Name      | : Intel(R) HD Graphics                                     |

### 1.6 Metode Penelitian

Pada penelitian ini metode yang digunakan adalah metode Rapid Aplication Development (RAD). RAD adalah salah satu metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat. Untuk pembangunan suatu sistem informasi biasanya memerlukan waktu 180 hari, akan tetapi dengan menggunakan metode RAD suatu sistem dapat di selesaikan hanya dalam waktu 30-90 hari. RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat perangkat lunak.



## **1.7 Metode Pengumpulan Data**

Teknik pengumpulan data di lakukan dengan cara :

1. Observasi  
Penelitian di lakukan dengan cara mengumpulkan data-data di fakultas teknologi informasi khususnya di bagian perpustakaan yang akan di jadikan bahan dasar dalam perancangan aplikasi.
2. Wawancara  
Penelitian di lakukan dengan proses tanya jawab kepada Bapak Ridwan selaku Dosen di fakultas teknologi informasi yang mengetahui permasalahan yang akan di bahas.
3. Studi kepustakaan
  - a) Mencari laporan hasil-hasil penelitian baru atau merupakan kelanjutan penelitian selanjutnya di perpustakaan fakultas teknoogi informasi.
  - b) Mengumpulkan jurnal yang memuat artikel-artikel yang relevan dengan masalah

## BAB IV JADWAL PENELITIAN

| NO             | URAIAN PEKERJAAN     | BOBOT (%) | BULAN KE 1 |     |     |     | BULAN KE 2 |     |     |     | BULAN KE 3 |      |      |      | BULAN KE 4 |      |      |      | BULAN KE 5 |      |      |      | BULAN KE 6 |      |      |       | BOTBOT (%) |
|----------------|----------------------|-----------|------------|-----|-----|-----|------------|-----|-----|-----|------------|------|------|------|------------|------|------|------|------------|------|------|------|------------|------|------|-------|------------|
|                |                      |           | 1          | 2   | 3   | 4   | 5          | 6   | 7   | 8   | 9          | 10   | 11   | 12   | 13         | 14   | 15   | 16   | 17         | 18   | 19   | 20   | 21         | 22   | 23   | 24    |            |
|                |                      |           |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            |      |      | 100%  |            |
| 1              | Pendaftaran          | 0,7       | 0,7        |     |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            |      |      |       |            |
| 2              | Pembuatan Proposal   | 0,7       |            | 0,7 |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            |      |      |       |            |
| 3              | Persetujuan Proposal | 0,7       |            |     | 0,3 | 0,3 |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            |      |      |       |            |
| 4              | Penentuan            | 1,3       |            |     |     | 0,7 | 0,7        |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            |      |      |       |            |
| 5              | identifikasi         | 3,3       |            |     |     |     | 0,8        | 0,8 | 0,8 | 0,8 |            |      |      |      |            |      |      |      |            |      |      |      |            |      |      |       |            |
| 6              | pengumpulan data     | 3,3       |            |     |     |     |            | 0,7 | 0,7 | 0,7 | 0,7        | 0,7  |      |      |            |      |      |      |            |      |      |      |            |      |      |       |            |
| 7              | Analisis             | 6,7       |            |     |     |     |            |     | 1,1 | 1,1 | 1,1        | 1,1  | 1,1  | 1,1  |            |      |      |      |            |      |      |      |            |      |      |       |            |
| 8              | Pembekalan           | 2,0       |            |     |     |     |            |     |     |     |            |      |      | 1,0  | 1,0        |      |      |      |            |      |      |      |            |      |      |       |            |
| 9              | Presentasi #1        | 3,3       |            |     |     |     |            |     |     |     |            |      |      |      |            | 1,7  | 1,7  |      |            |      |      |      |            |      |      | 50%   |            |
| 10             | Desain               | 13,3      |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      | 2,7  | 2,7        | 2,7  | 2,7  | 2,7  |            |      |      |       |            |
| 11             | Koding               | 26,7      |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      | 4,4  | 4,4        | 4,4  | 4,4  | 4,4  | 4,4        |      |      |       |            |
| 12             | Implementasi         | 20,0      |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      | 10,0 | 10,0       |      |      |       |            |
| 13             | Pemberkasan          | 2,0       |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      | 1,0        | 1,0  |      |       |            |
| 14             | Presentasi #2        | 3,3       |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            | 1,7  | 1,7  |       |            |
| 15             | Laporan              | 6,7       |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            | 3,3  | 3,3  |       |            |
| 16             | Lain-lain            | 6,0       |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            |      | 6,0  |       |            |
|                |                      |           |            |     |     |     |            |     |     |     |            |      |      |      |            |      |      |      |            |      |      |      |            |      |      | 0%    |            |
| JUMLAH BOBOT   |                      | 100%      | 0,7        | 0,7 | 0,3 | 0,3 | 0,7        | 1,5 | 1,5 | 2,6 | 2,6        | 1,5  | 1,5  | 1,1  | 2,1        | 1,0  | 1,7  | 1,7  | 7,1        | 7,1  | 7,1  | 7,1  | 17,1       | 15,4 | 6,0  | 11,0  |            |
| BOBOT MINGGUAN |                      |           | 0,7        | 1,3 | 1,7 | 2,0 | 2,7        | 4,2 | 5,7 | 8,3 | 10,8       | 12,7 | 14,4 | 15,6 | 17,7       | 18,7 | 20,3 | 22,0 | 29,1       | 36,2 | 43,3 | 50,4 | 67,6       | 83,0 | 89,0 | 100,0 |            |



## DAFTAR PUSTAKA

- Arya Febiyan ,2014, *Pengertian JSON*,  
<https://www.dumetschool.com/blog/Pengertian-JSON>
- NetBeans IDE, 2017 *NetBeans IDE*, [https://www.netbeans.org/index\\_id.html](https://www.netbeans.org/index_id.html)
- Nofriadi, 2015, *Java Fundamental Dengan Netbeans 8.0.2*, Deepublish,  
Yogyakarta
- Rickyanto, 2003, *Dasar Pemrograman Berorientasi Objek Dengan Java 2 (JDK 1.4 )*, ANDI, Yogyakarta.
- Ridwan Fajar, 2016, *Mengenal Diagram UML (Unified Modeling Language)*.  
<https://www.co>  
[depolitan.com/mengenal-diagram-uml-unified-modeling-language](https://www.co)
- Wikipedia, *Database NoSQL*, <https://id.m.wikipedia.org/wiki/NoSQL>