

## Sistem Berbasis *Cloud Computing* Untuk Identifikasi Resep Dokter “BarSep”

Andi Saryoko<sup>1</sup>, Ganda Wijaya<sup>2</sup>, Irwansyah Saputra<sup>3</sup>, Meilynda Trisiana<sup>4</sup>, Asep Mulyana<sup>5</sup>,  
Dandi Yusbial Bayani<sup>6</sup>, Dharma Winata<sup>7</sup>, Vilsafa Khoirunnisak<sup>8</sup>  
<sup>1, 2, 3, 4, 5, 6, 7, 8</sup> Department of Information System, STMIK Nusa Mandiri, Indonesia

---

### Article Info

### ABSTRACT

---

#### Article history:

Received Sep 17, 2020

Revised Jan 31, 2021

Accepted Feb 6, 2021

---

#### Keywords:

Prescription

Recipe

Medicine

QR Code

Cloud Computing

A doctor's prescription is a doctor's written request to the pharmacist to prepare and give medicine to the patient. Prescriptions are made according to the needs of the patient after the doctor has examined and diagnosed the patient. However, doctor's writing on a prescription that considered unclear can cause errors when compounding/preparing the drug and using prescribed drugs. In fact, the cure rate and life expectancy of patients is directly proportional to the administration of the right medicine. This study aims to prevent errors in the process of identification of prescription drugs by pharmacists. The technology used is cloud computing with the implementation of QR Code. The QR Code contains patient examination information including patient data, prescription drugs, and diagnoses, so that when the pharmacist scans the QR Code, the system will display all patient information that has been inputted by the doctor at the time of the examination. The results obtained from the implementation of this application at the Rapha Farma Pharmacy is the applications effective for tackling errors in reading doctor's prescriptions that can save patients from medication errors.

Copyright © 2020 Universitas Indraprasta PGRI.  
All rights reserved.

---

#### Corresponding Author:

Andi Saryoko,

Department of Information System,

STMIK Nusa Mandiri,

Jl. Margonda Raya No. 545, Pondok Cina, Kecamatan Beji, Kota Depok 16424.

Email: [andi.asy@nusamandiri.ac.id](mailto:andi.asy@nusamandiri.ac.id)

---

### 1. PENDAHULUAN

Keselamatan pasien merupakan suatu disiplin baru dalam pelayanan kesehatan yang mengutamakan pelaporan, analisis, dan pencegahan *medical error* yang sering menimbulkan Kejadian Tak Diharapkan (KTD) dalam pelayanan kesehatan. Kegiatan skrining resep yang dilakukan tenaga kefarmasian adalah untuk mencegah terjadinya kesalahan pengobatan (*Medication error*) [1]. *Medication error* adalah suatu kejadian yang tidak hanya dapat merugikan pasien tetapi juga dapat membahayakan keselamatan pasien yang dilakukan oleh petugas kesehatan. *Medication error* dapat terjadi pada tahapan *prescribing*, *transcribing*, *dispensing*, dan *administering*. Kesalahan dalam tahap *transcribing* meliputi kesalahan dalam mengartikan resep [2]. Resep merupakan perwujudan akhir dari kompetensi, pengetahuan dan keahlian dokter dalam menerapkan pengetahuannya dalam bidang farmakologi dan terapi. Resep yang ditulis dengan tidak jelas akan menimbulkan terjadinya kesalahan saat peracikan/penyiapan obat dan penggunaan obat yang diresepkan. *National Academies of Science's Institute of Medicine (IOM)* Amerika Serikat pada tahun 1999 melaporkan bahwa sekitar 44.000-98.000 orang meninggal karena *medical error*, kemudian pada

tahun 2006 setidaknya 1,5 juta orang mengalami cedera dan 7.000 orang meninggal dunia setiap tahunnya akibat kesalahan membaca resep yang ditulis tangan dokter [3].

Seperti yang dimuat dalam kanal berita CNN, pada Desember 2018 terdapat laporan medis yang telah dipublikasikan jurnal *BMJ Case Reports*, seorang perempuan di Skotlandia harus dilarikan ke Instalasi Gawat Darurat (IGD) karena kesalahan pemakaian obat disfungsi ereksi pada mata [4]. Hal ini dikarenakan apoteker salah membaca resep dokter yang seharusnya Vita-Pos (Obat untuk mata kering dan erosi kornea) tetapi apoteker membacanya sebagai Vitaros (Obat disfungsi ereksi). Contoh kasus kesalahan pembacaan resep juga terjadi di Indonesia. Dokter di puskesmas Buleleng Tiga, Bali memberikan resep obat tetes mata kepada pasien, tetapi bagian farmasi membaca resep tersebut sebagai obat tetes telinga. Kasus yang sama juga terjadi di Kota Bandar Lampung. Seorang gadis berusia 18 tahun mengalami kebutaan karena diberikan obat tetes telinga akibat kesalahan pembacaan resep oleh apoteker. Kasus lain dialami oleh Jean Dowd (68 Tahun) di Inggris meninggal karena kanker dan mendapatkan pengobatan yang salah akibat hasil pemeriksaannya tertukar dengan pasien lain [5]. Kasus-kasus tersebut merupakan contoh kecil dari banyaknya *medication error* pada resep obat. Padahal, tingkat kesembuhan dan harapan hidup pasien berbanding lurus dengan pemberian obat yang tepat. Oleh karena itu, kesalahan-kesalahan tersebut seharusnya dapat dicegah seiring berkembangnya teknologi.

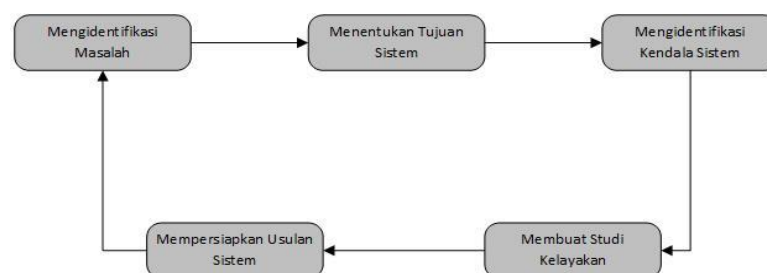
Teknologi atau sistem yang dapat menghubungkan antara dokter dengan pihak terkait guna mencegah terjadinya kesalahan proses identifikasi resep obat diusulkan sebagai solusi dari permasalahan diatas. Teknologi ini diberi nama “Sistem Berbasis *Cloud Computing* untuk Identifikasi Resep Dokter “BarSep””. Cara kerja dari sistem ini adalah dokter memasukkan data pasien dan resep kedalam sistem kemudian pasien menerima *print out* kertas resep berisi daftar obat (dalam bahasa medis) yang harus ditebus berikut *QR Code*. Setelah menerima *print out*, resep tersebut diberikan kepada apoteker untuk di cek keabsahannya dengan memindai *QR Code* menggunakan komputer *desktop* maupun *smartphone*. Proses pemindaian ini dilakukan untuk menampilkan informasi yang terkandung pada resep sehingga data pasien dan daftar obat yang tertulis di resep dapat disesuaikan dengan informasi yang tampil pada komputer atau *smartphone*. Sistem ini diharapkan dapat menanggulangi kesalahan membaca tulisan dokter pada resep, sehingga tidak menimbulkan korban jiwa karena kesalahan identifikasi resep di masa yang akan datang.

## 2. METODE

Pelaksanaan pengembangan sistem “BarSep” dilakukan dengan menggunakan metode *waterfall*. Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*”, dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak [6]. Metode *waterfall* berkembang secara sistematis dari satu tahap ke tahap lain layaknya air terjun [7]. Disebut air terjun karena seperti air terjun yang jatuh satu demi satu sehingga penyelesaian tahap sebelumnya kemudian dapat dilanjutkan ke tahap berikutnya dan berjalan-urut. Kelebihan menggunakan metode air terjun (*waterfall*) adalah metode ini memungkinkan untuk departementalisasi dan kontrol. proses pengembangan model fase *one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi [8]. Dalam pengembangan *software* metode *waterfall* dimana pengerjaannya harus dilakukan secara berurutan yang dimulai dari tahap perencanaan, pemodelan (*design*), perancangan, pengujian (*testing*) dan pemeliharaan [7].

### 1. Tahap Perencanaan

Dimulai dengan menyadari dan mengidentifikasi adanya masalah yaitu belum ada teknologi pembantu untuk identifikasi resep, sehingga memungkinkan terjadinya kesalahan dalam membaca resep dokter [9]. Dilanjutkan dengan menentukan tujuan sistem, yaitu untuk membantu bagian apotek dalam membaca tulisan tangan pada resep dokter [10]. Mengidentifikasi kendala sistem. Dalam pengoperasian sistem, tidak lepas dari berbagai kendala. Beberapa kendala yang mungkin muncul adalah kendala perangkat keras yaitu *QR Code Scanner* yang tidak tersedia di apotek, kendala waktu untuk pengembangan sistem, dan kendala keadaan berupa pandemic sehingga tidak memungkinkan untuk lebih banyak observasi secara langsung [11]. Langkah selanjutnya adalah membuat studi kelayakan. Studi kelayakan meninjau faktor-faktor yang mempengaruhi kemampuan sistem untuk mencapai tujuan yang diinginkan [12]. Langkah terakhir adalah mempersiapkan usulan sistem. Usulan sistem dibuat setelah mendapat solusi dari kendala sistem dan studi kelayakan [13].



Gambar 1. Tahap Perencanaan Sistem

### 2. Tahap Desain dan Sistem Aplikasi

Tahap desain dan sistem aplikasi dibuat dengan menganalisa prosedur sistem berjalan Apotek Rapha Farma dan membuat prosedur sistem yang diusulkan dalam bentuk *use case diagram* (admin, dokter, apoteker) dan *activity diagram* (BarSep).

3. Tahap Perencanaan Program

Pada tahap perancangan program akan ditampilkan *user interface* dari sistem yang diusulkan. Tampilan *user interface* yang dibuat antara lain : Menu *Login, Dashboard Admin, Dashboard Doctor, Dashboard Employee, Menu User, Add New User, Edit User, Menu Medicines, Add New Medicine, Edit Medicine, Delete Medicines, Menu Patients, Add New Patient, Edit Patient, Delete Patient, Menu Recipes, Add New Recipe, Edit Recipe, Delete Recipe, Menu Records, Add New Records, Print QR Code, Edit Record, Delete Record, Menu Recipes Akun Employee, Edit Recipes Akun Employee, Hasil Pemindaian Resep Dokter.*

4. Tahap Implementasi

Tahap ini menjabarkan tentang spesifikasi dari *hardware* dan *software* yang digunakan. *Hardware* yang digunakan antara lain : CPU (*Central Processing Unit*), Monitor, *Keyboard, Mouse, Laptop, QR Code Scanner, Smartphone.* *Software* yang digunakan antara lain : Bahasa Pemrograman PHP 7.1.3, Database MySQL, Framework Laravel, Composer, Code Editor Visual Code, Xampp.

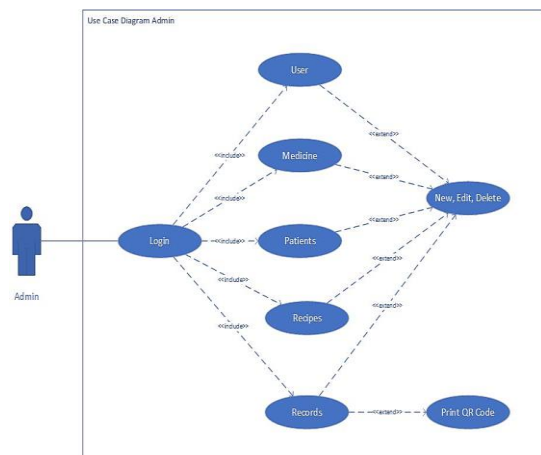
5. Tahap *Testing* dan Evaluasi

Menampilkan hasil pengujian menyeluruh untuk setiap *form* dan menu pada sistem yang dibangun dengan menggunakan *black box testing*. Ketentuan pengujian menyertakan nama pengujian, skenario pengujian, hasil yang diharapkan, hasil pengujian, dan kesimpulan [14].

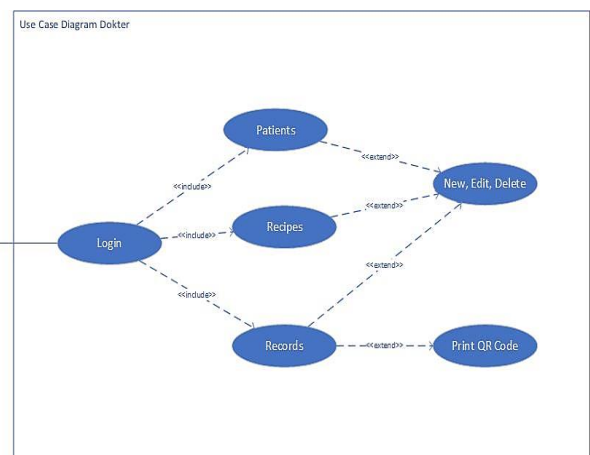
3. HASIL DAN PEMBAHASAN

Memetakan sistem menggunakan *Use Case Diagram, Entity Relationship Diagram, Sequence Diagram,* sebagai acuan untuk membangun sistem serta menampilkan *User Interface Program.*

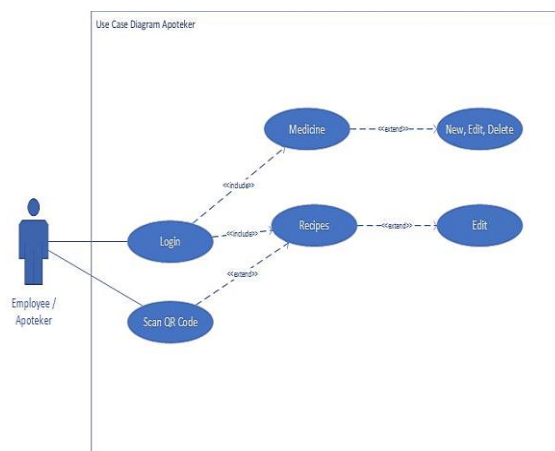
1. *Use Case Diagram & Scenario*



Gambar 2. *Use Case Diagram Admin*



Gambar 3. *Use Case Diagram Doctor*



Gambar 4. *Use Case Diagram Employee*

Tabel 1. *Use Case Scenario Login*

Title	Skenario Melakukan <i>Login</i>
Deskripsi	Aktor melakukan <i>login</i>
Aktor & Interface	Aktor : <i>Admin, Doctor, Employee</i> Interface : <i>Halaman Login, Halaman Admin/Doctor/Employee</i>
Pre-condition	Aktor belum melakukan <i>login</i>
Basic Flow	<ul style="list-style-type: none"> <li>• Aktor membuka halaman <i>Login</i></li> <li>• Aktor memasukkan <i>username &amp; password</i></li> <li>• Sistem memvalidasi akun pengelola</li> <li>• Jika valid, Sistem menampilkan halaman <i>Admin/Doctor/Employee</i></li> <li>• Jika tidak, Sistem mengkonfirmasi bahwa <i>username &amp; password</i> salah di halaman <i>Login</i></li> </ul>
Post-condition	Aktor telah berhasil masuk ke halaman <i>Admin/Doctor/Employee</i>
Alternative(s) Flow	Aktor gagal masuk ke halaman <i>Admin/Doctor/Employee</i>

Tabel 2. Use Case Scenario Login

Title	Skenario Kelola <i>User</i>
Deskripsi	Aktor mengelola daftar & informasi <i>user</i>
Aktor & Interface	Aktor : <i>Admin</i> Interface : <i>Halaman Admin, Halaman User List</i>
Pre-condition	Aktor sudah melakukan <i>login</i>
Basic Flow	<ul style="list-style-type: none"> <li>• Aktor melihat daftar <i>user</i></li> <li>• Aktor dapat menambah <i>user</i> baru</li> <li>• Aktor dapat mengubah informasi <i>user</i> yang sudah terdaftar</li> <li>• Aktor dapat menghapus <i>user</i> yang sudah terdaftar</li> </ul>
Post-condition	<ul style="list-style-type: none"> <li>• Aktor telah berhasil menambah <i>user</i> baru</li> <li>• Aktor telah berhasil mengubah informasi <i>user</i> yang sudah terdaftar</li> <li>• Aktor telah berhasil menghapus <i>user</i> yang sudah terdaftar</li> </ul>
Alternative(s) Flow	<ul style="list-style-type: none"> <li>• Aktor gagal menambah <i>user</i> baru</li> <li>• Aktor gagal mengubah informasi <i>user</i> yang sudah terdaftar</li> <li>• Aktor gagal menghapus <i>user</i> yang sudah terdaftar</li> </ul>

Tabel 3. Use Case Scenario Medicines

Title	Skenario Kelola Daftar Obat
Deskripsi	Aktor mengelola daftar & informasi obat
Aktor & Interface	Aktor : <i>Admin &amp; Employee</i> Interface : <i>Halaman Admin/Employee, Halaman Medicines</i>
Pre-condition	Aktor sudah melakukan <i>login</i>
Basic Flow	<ul style="list-style-type: none"> <li>• Aktor melihat daftar obat</li> <li>• Aktor dapat menambah obat baru</li> <li>• Aktor dapat mengubah informasi obat yang sudah terdaftar</li> <li>• Aktor dapat menghapus obat yang sudah terdaftar</li> </ul>
Post-condition	<ul style="list-style-type: none"> <li>• Aktor telah berhasil menambah obat baru</li> <li>• Aktor telah berhasil mengubah informasi obat yang sudah terdaftar</li> <li>• Aktor telah berhasil menghapus obat yang sudah terdaftar</li> </ul>
Alternative(s) Flow	<ul style="list-style-type: none"> <li>• Aktor gagal menambah obat baru</li> <li>• Aktor gagal mengubah informasi obat yang sudah terdaftar</li> <li>• Aktor gagal menghapus obat yang sudah terdaftar</li> </ul>

Tabel 4. Use Case Scenario Patients

Title	Skenario Kelola Daftar Pasien
Deskripsi	Aktor mengelola daftar & informasi pasien
Aktor & Interface	Aktor : <i>Admin &amp; Doctor</i> Interface : <i>Halaman Admin/Doctor, Halaman Patient List</i>
Pre-condition	Aktor sudah melakukan <i>login</i>
Basic Flow	<ul style="list-style-type: none"> <li>• Aktor melihat daftar pasien</li> <li>• Aktor dapat menambah pasien baru</li> <li>• Aktor dapat mengubah informasi pasien yang sudah terdaftar</li> <li>• Aktor dapat menghapus pasien yang sudah terdaftar</li> </ul>

<i>Post-condition</i>	<ul style="list-style-type: none"> <li>• Aktor telah berhasil menambah pasien baru</li> <li>• Aktor telah berhasil mengubah informasi pasien yang sudah terdaftar</li> <li>• Aktor telah berhasil menghapus pasien yang sudah terdaftar</li> </ul>
<i>Alternative(s) Flow</i>	<ul style="list-style-type: none"> <li>• Aktor gagal menambah pasien baru</li> <li>• Aktor gagal mengubah informasi pasien yang sudah terdaftar</li> <li>• Aktor gagal menghapus pasien yang sudah terdaftar</li> </ul>

Tabel 5. Use Case Scenario Recipes

Title	Skenario Kelola Daftar Resep Dokter
Deskripsi	Aktor mengelola daftar & informasi resep dokter
Aktor & Interface	Aktor : <i>Admin &amp; Doctor</i> Interface : Halaman <i>Admin &amp; Doctor</i> , Halaman <i>Recipes</i>
<i>Pre-condition</i>	Pengelola sudah melakukan <i>login</i>
<i>Basic Flow</i>	<ul style="list-style-type: none"> <li>• Aktor melihat daftar resep dokter</li> <li>• Aktor dapat menambah resep dokter baru</li> <li>• Aktor dapat mengubah informasi resep dokter yang sudah terdaftar</li> <li>• Aktor dapat menghapus resep dokter yang sudah terdaftar</li> </ul>
<i>Post-condition</i>	<ul style="list-style-type: none"> <li>• Aktor telah berhasil menambah resep dokter baru</li> <li>• Aktor telah berhasil mengubah informasi resep dokter yang sudah terdaftar</li> <li>• Aktor telah berhasil menghapus resep dokter yang sudah terdaftar</li> </ul>
<i>Alternative(s) Flow</i>	<ul style="list-style-type: none"> <li>• Aktor gagal menambah resep dokter baru</li> <li>• Aktor gagal mengubah informasi resep dokter yang sudah terdaftar</li> <li>• Aktor gagal menghapus resep dokter yang sudah terdaftar</li> </ul>

Tabel 6. Use Case Scenario Recipes(Employee)

Title	Skenario Kelola Resep Dokter (Employee)
Deskripsi	Aktor mengelola informasi resep dokter
Aktor & Interface	Aktor : <i>Employee</i> Interface : Halaman <i>Employee</i> , Halaman <i>Recipes</i>
<i>Pre-condition</i>	Pengelola sudah melakukan <i>login</i>
<i>Basic Flow</i>	<ul style="list-style-type: none"> <li>• Aktor melihat daftar resep dokter</li> <li>• Aktor dapat mengubah informasi resep dokter yang sudah terdaftar</li> </ul>
<i>Post-condition</i>	Aktor telah berhasil mengubah informasi resep dokter yang sudah terdaftar
<i>Alternative(s) Flow</i>	Aktor gagal mengubah informasi resep dokter yang sudah terdaftar

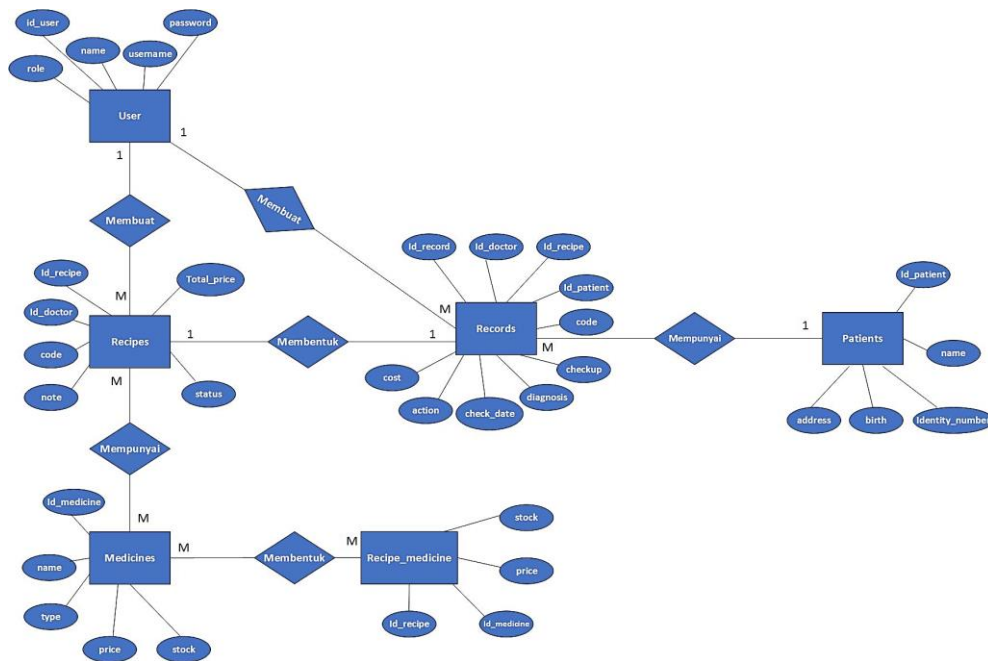
Tabel 7. Use Case Scenario QR Code(Employee)

Title	Skenario Membaca <i>QR Code</i> Resep Dokter
Deskripsi	Aktor membaca <i>QR Code</i> resep dokter
Aktor & Interface	Aktor : <i>Employee</i> Interface : Halaman <i>Employee</i> , Halaman <i>Recipes</i> , Halaman <i>QR Code</i>
<i>Pre-condition</i>	Pengelola sudah melakukan <i>login &amp; scan QR Code</i>
<i>Basic Flow</i>	<ul style="list-style-type: none"> <li>• Aktor melakukan <i>scan QR Code</i></li> <li>• Aktor dapat melihat informasi obat yang sesuai dengan resep dokter dari hasil <i>scan QR Code</i></li> <li>• Aktor dapat mengubah informasi resep dokter yang sudah terdaftar</li> </ul>
<i>Post-condition</i>	<ul style="list-style-type: none"> <li>• Aktor berhasil melakukan <i>scan QR Code</i></li> <li>• Aktor berhasil melihat informasi obat yang sesuai dengan resep dokter dari hasil <i>scan QR Code</i></li> <li>• Aktor telah berhasil mengubah informasi resep dokter yang sudah terdaftar</li> </ul>
<i>Alternative(s) Flow</i>	<ul style="list-style-type: none"> <li>• Aktor gagal melakukan <i>scan QR Code</i></li> <li>• Aktor gagal melihat informasi obat yang sesuai dengan resep dokter dari hasil <i>scan QR Code</i></li> <li>• Aktor telah berhasil mengubah informasi resep dokter yang sudah terdaftar</li> </ul>

Tabel 8. Use Case Scenario Records

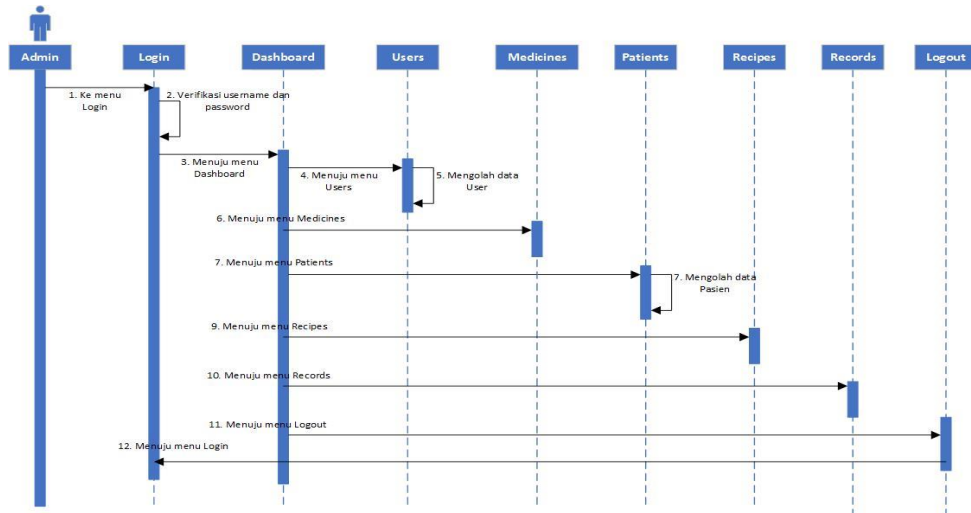
Title	Skenario Kelola Rekam Medis Pasien
Deskripsi	Aktor mengelola rekam medis pasien
Aktor & Interface	Aktor : <i>Admin &amp; Doctor</i> Interface : <i>Halaman Admin/Doctor, Halaman Records, Halaman QR Code</i>
Pre-condition	Pengelola sudah melakukan <i>login</i>
Basic Flow	<ul style="list-style-type: none"> <li>• Aktor melihat daftar rekam medis pasien</li> <li>• Aktor dapat menambah rekam medis pasien</li> <li>• Aktor dapat mengubah informasi rekam medis pasien dokter yang sudah terdaftar</li> <li>• Aktor dapat menghapus rekam medis pasien yang sudah terdaftar</li> <li>• Aktor dapat melihat <i>QR Code</i> berdasarkan rekam medis pasien yang terdaftar</li> <li>• Aktor dapat melakukan pencetakan <i>QR Code</i> berdasarkan rekam medis pasien yang terdaftar</li> </ul>
Post-condition	<ul style="list-style-type: none"> <li>• Pengelola telah berhasil menambah rekam medis pasien baru</li> <li>• Pengelola telah berhasil mengubah informasi rekam medis pasien yang sudah terdaftar</li> <li>• Pengelola telah berhasil menghapus rekam medis pasien yang sudah terdaftar</li> <li>• Pengelola telah berhasil melihat <i>QR Code</i> berdasarkan rekam medis pasien yang terdaftar</li> <li>• Pengelola telah berhasil melakukan pencetakan <i>QR Code</i> berdasarkan rekam medis pasien yang terdaftar</li> </ul>
Alternative(s) Flow	<ul style="list-style-type: none"> <li>• Pengelola gagal menambah rekam medis pasien baru</li> <li>• Pengelola gagal mengubah informasi rekam medis pasien yang sudah terdaftar</li> <li>• Pengelola gagal menghapus rekam medis pasien yang sudah terdaftar</li> <li>• Pengelola gagal melihat <i>QR Code</i> berdasarkan rekam medis pasien yang terdaftar</li> <li>• Pengelola gagal melakukan pencetakan <i>QR Code</i> berdasarkan rekam medis pasien yang terdaftar</li> </ul>

2. Entity Relationship Diagram (ERD)

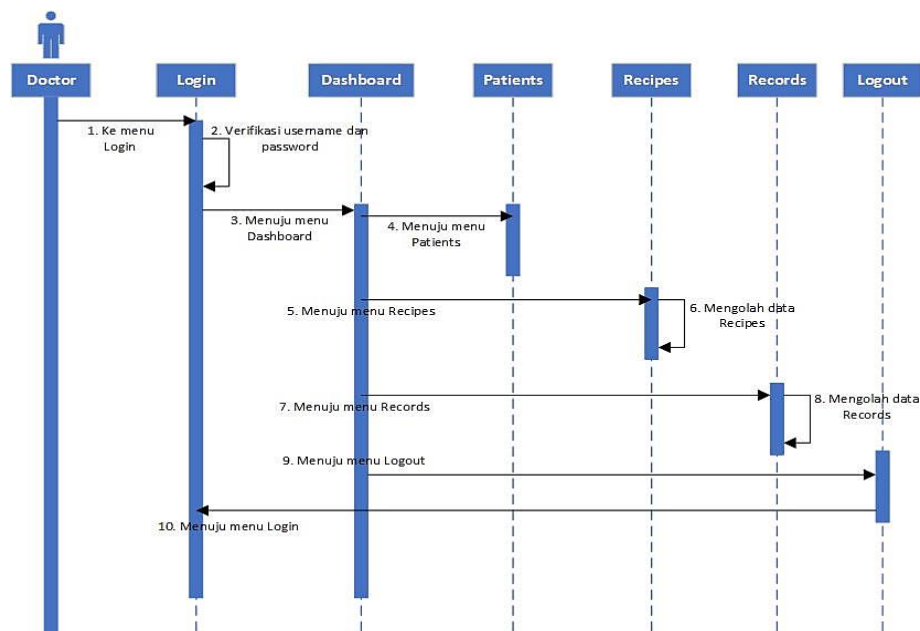


Gambar 5. Entity Relationship Diagram

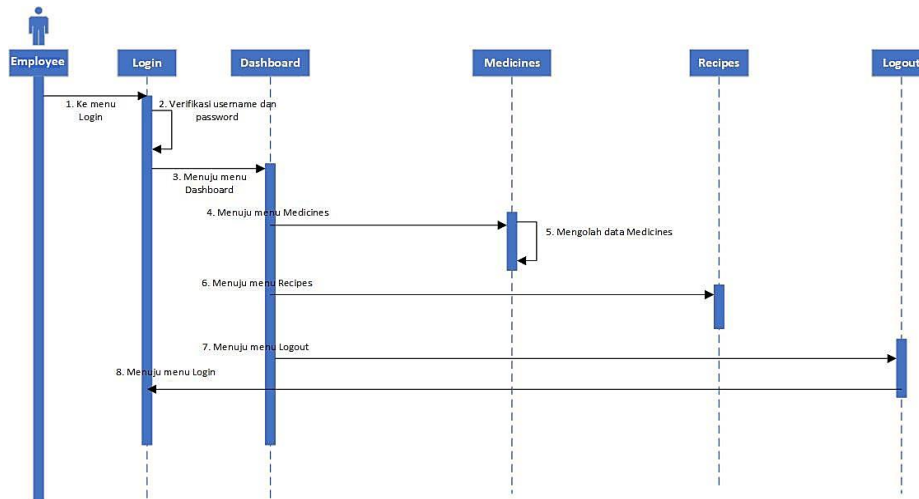
- Terdapat 5 (lima) entitas yang menunjukkan objek pada sistem.
  - Terdapat 4 (empat) relasi yang menunjukkan hubungan antar entitas.
  - Terdapat 30 (tiga puluh) atribut yang merupakan penjelasan detail entitas.
3. Sequence Diagram

Gambar 6. *Sequence Diagram Admin*

- Terdapat 1 (satu) actor yaitu admin.
- Terdapat 8 (delapan) *life line* antar muka yang saling berinteraksi yaitu *login*, *dashboard*, *medicines*, *patients*, *receipes*, *record*, dan *logout*.
- Terdapat 12 (dua belas) *message* dari komunikasi antar objek berisi aktivitas yang terjadi. Admin memiliki hak akses untuk menuju semua *life line* antar muka tetapi sesuai kode etik hanya melakukan pengolahan data pada antar muka *users* dan *patients*.

Gambar 7. *Sequence Diagram Doctor*

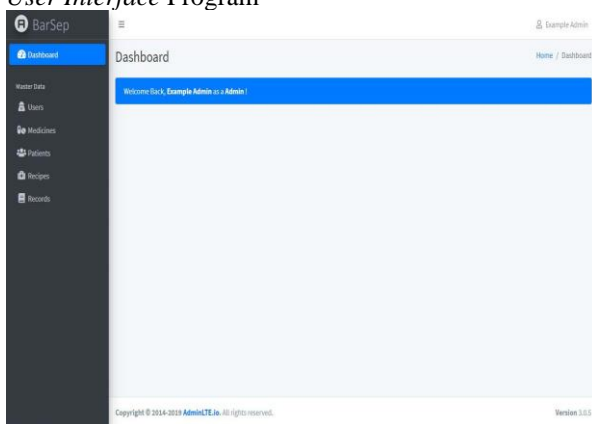
- Terdapat 1 (satu) actor yaitu *doctor*.
- Terdapat 6 (enam) *life line* antar muka yang saling berinteraksi yaitu *login*, *dashboard*, *patients*, *receipes*, *record*, dan *logout*.
- Terdapat 11 (sepuluh) *message* dari komunikasi antar objek berisi aktivitas yang terjadi. Dokter hanya memiliki hak akses terhadap enam *life line* antar muka tersebut.



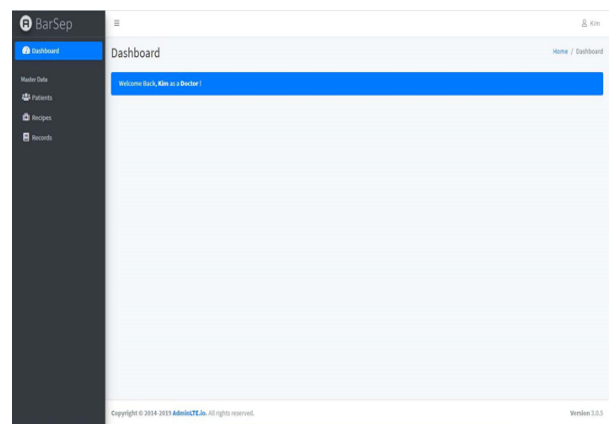
Gambar 8. *Sequence Diagram Employee*

- g. Terdapat 1 (satu) actor yaitu *employee*.
- h. Terdapat 5 (lima) *life line* antar muka yang saling berinteraksi yaitu *login*, *dashboard*, *medicines*, *receipes*, dan *logout*.
- i. Terdapat 8 (delapan) *message* dari komunikasi antar objek berisi aktivitas yang terjadi. Apoteker hanya memiliki hak akses terhadap lima *life line* antar muka tersebut.

4. *User Interface Program*

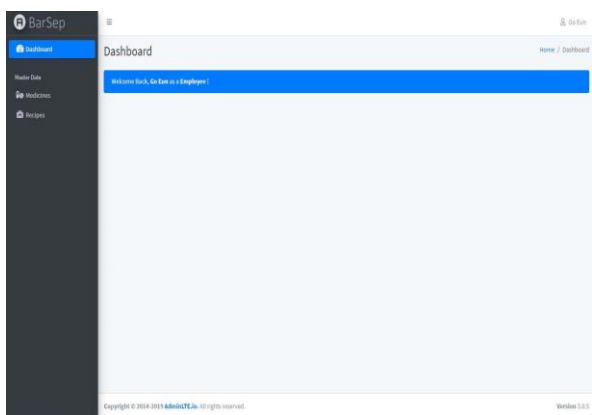


Gambar 9. Tampilan *Dashboard Admin*



Gambar 10. Tampilan *Dashboard Doctor*

Gambar 9 apabila login sebagai admin, maka menu yang muncul dan dapat diakses adalah menu *Users*, *Medicines*, *Patients*, *Recipes*, dan *Records*. Sedangkan Gambar 10 apabila login sebagai dokter, maka akan muncul menu *Patients*, *Recipes*, dan *Records*.



Gambar 11. Tampilan *Dashboard Employee*

#	Role	Username	Name	Actions
1	admin	admin	Example Admin	[Edit] [Delete]
2	employee	employee	Example Employee	[Edit] [Delete]
3	doctor	doctor	Example Doctor	[Edit] [Delete]
4	doctor	dr.kim	Kim	[Edit] [Delete]
5	employee	gusam	Go Sam	[Edit] [Delete]
6	employee	park.harta	Park	[Edit] [Delete]
7	admin	Rahju	Rahju	[Edit] [Delete]
8	doctor	Sari	Sulawesi	[Edit] [Delete]
9	employee	Devi	Devi Sukmawati	[Edit] [Delete]

Gambar 12. Tampilan Menu *User*

Gambar 11 apabila login sebagai *employee* / apoteker, maka hanya dapat mengakses menu *Medicines* dan *Recipes*. Dan Gambar 12 berupa menu *user* berisi data-data *user* yang telah dibuat. Dalam menu *user* terdapat menu *New* yang diakses dari ikon *gear* untuk menambah *user*, menu *edit* diakses dari ikon *checklist* untuk mengubah data *user*, dan menu *delete* diakses melalui ikon *delete* untuk menghapus *user*.

#	Name	Type	Price	Stock
1	Paradol	capsules	2000	35
2	Bodrex	tablet	3000	46
3	Test	injections	9000	88
4	Bodrex	liquid	3000	99
5	Acarbose	capsules	5000	100
6	ACE Inhibitor	tablet	6000	100
7	Acetazolamide	liquid	7000	99
8	Acetylcysteine	injections	8000	100
9	Actifed	capsules	9000	100
10	Acyclovir Tablet	tablet	10000	100

Gambar 13. Tampilan Menu *Medicines*

#	Name	Identity Number	Date of Birth	Address
1	Linda	3327015132930004	11/04/1993	Jl. Daan Mogot No.119 Komplek Aldora Blok 4 No 17-18 Jakarta Barat
2	Adha Pratama	3327015132937373	01/01/2020	palemang
3	Mai	3326862128564	22/06/1994	Jl. Jendragilik
4	asepmulyana	3324679810101	18/04/1993	Jakarta Barat, RT.3, RW.3, Wijaya Kusuma, Kec. Gonggoljembatan
5	vifa	11122233444555	10/10/2010	jakarta barat

Gambar 14. Tampilan Menu *Patients*

Gambar 13 menu *medicines* berisi data-data obat yang telah dibuat. Dalam menu *medicines* terdapat menu *new* yang diakses dari ikon *gear* untuk menambah obat, menu *import excel* untuk memasukkan data obat dari microsoft excel sehingga tidak perlu input satu per satu, menu *edit* yang diakses dari ikon *checklist* untuk mengubah data obat, dan menu *delete* diakses melalui ikon *delete* untuk menghapus obat. Dan Gambar 14 data pasien dimasukkan dalam menu *patients*. Dalam menu *patients* terdapat menu *new* yang diakses dari ikon *gear* untuk menambah data pasien, menu *edit* yang diakses dari ikon *checklist* untuk mengubah data pasien, dan menu *delete* diakses melalui ikon *delete* untuk menghapus data pasien.

#	Doctor	Code	Total Price	Status
1	Example Doctor	v1q4e2dln.1532019469	Rp 1.000,00	not_yet_taken
2	Example Doctor	g7hCdp7Y153206811	Rp 3.000,00	not_yet_taken
3	Kim	Wlqghu4qu.1532081275	Rp 25.000,00	not_yet_taken
4	Susilowati	P9h42Wqj1532065721	Rp 22.000,00	not_yet_taken
5	Susilowati	g7CSPKdu.1532085783	Rp 8.500,00	not_yet_taken
6	Susilowati	Sluht2kwa1532055811	Rp 1.500,00	not_yet_taken

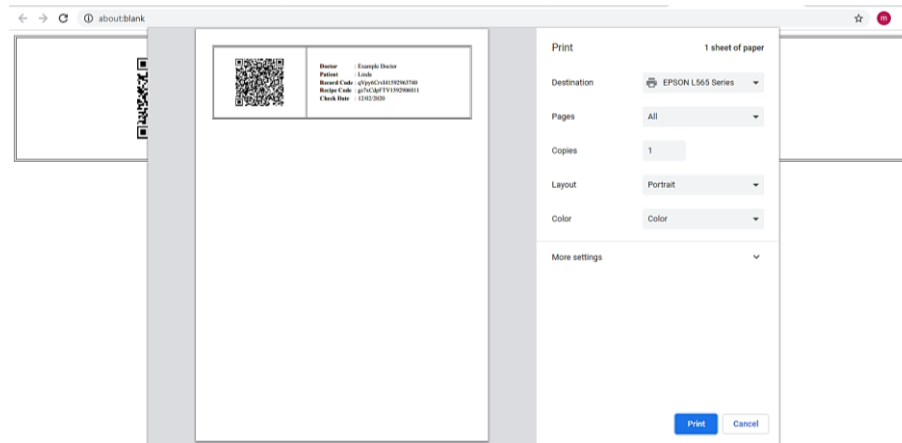
Gambar 15. Tampilan Menu *Recipes*

#	Patient	Doctor	Record Code	Recipe Code	Check Date
1	Linda	Example Doctor	q1qyK14h320293740	g7hCdp7Y153206811	12/02/2020
2	Mai	Kim	16740444444552081374	Wlqghu4qu.1532081275	24/08/2020
3	Mai	Kim	819K17484435208221	Wlqghu4qu.1532081275	24/08/2020
4	vifa	Susilowati	cm24h4D1030307125	Sluht2kwa1532055811	30/01/2020

Gambar 16. Tampilan Menu *Records*

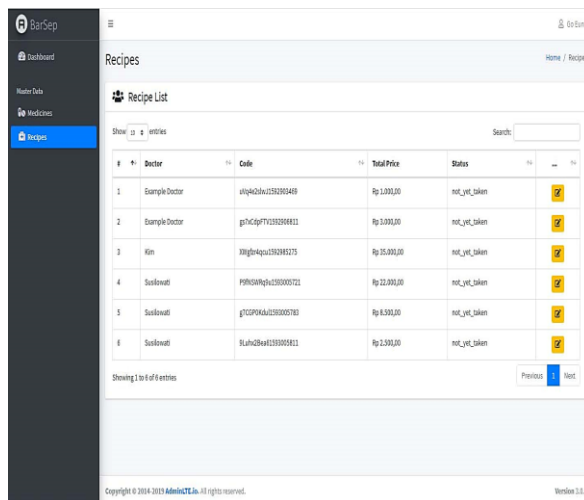
Gambar 15 menu *recipes* dibuat untuk *input* resep yang berisi daftar obat dan catatan dosis obat. Dalam menu *recipes* terdapat menu *new* yang diakses dari ikon *gear* untuk menambah data resep, menu *edit* yang diakses dari

ikon *checklist* untuk mengubah data resep, dan menu *delete* diakses melalui ikon *delete* untuk menghapus data resep. Dan Gambar 16 dari menu *records* inilah akan terbentuk print out resep yang akan diserahkan kepada pasien. Menu *records* berisi data rekam medis pasien. Dalam menu *records* terdapat menu *new* yang diakses dari ikon *gear* untuk menambah data rekam medis, menu *edit* yang diakses dari ikon *checklist* untuk mengubah data rekam medis, dan menu *delete* diakses melalui ikon *delete* untuk menghapus data rekam medis.

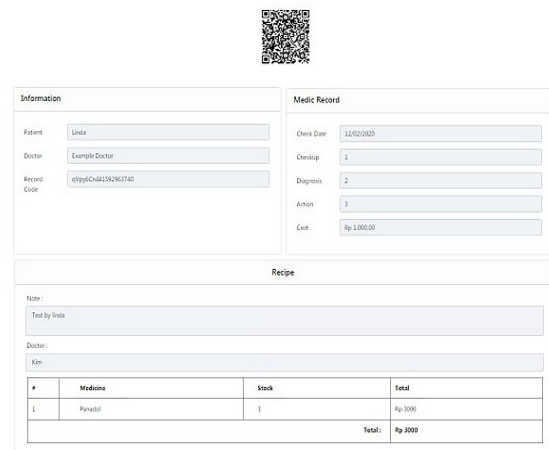


Gambar 17. Tampilan *Print QR Code*

Klik ikon empat kotak berwarna biru, maka akan diarahkan ke tab baru yang berisi tampilan *print preview*. Klik *print* untuk mencetak *QR Code* yang berisi data rekam medis.



Gambar 18. Tampilan Menu *Recipes* Akun *Employee*



Gambar 19. Hasil Pemindaian Resep Dokter

Berbeda dengan menu *recipes* pada akun admin dan dokter, dalam akun *employee* menu *recipes* hanya diberikan hak akses untuk *edit*.

#### 4. PENUTUP

Kesimpulan dari pembuatan aplikasi identifikasi resep dokter “BarSep” pada Apotek Rapha Farma dijabarkan sebagai berikut:

1. Dengan menggunakan aplikasi BarSep, kesalahan pembacaan resep dokter berhasil ditanggulangi karena tulisan dokter yang dinilai kurang jelas dapat dibaca melalui *QR Code* sehingga dapat menyelamatkan pasien dari *medication error*.
2. Aplikasi BarSep menyimpan data pada *cloud* sehingga pembacaan informasi dapat dilakukan secara tepat, cepat dan *real time*. Penyimpanan data di *cloud* juga menjaga agar data tidak mudah hilang.
3. Tampilan yang *user friendly* pada aplikasi BarSep memberikan kemudahan pada pengguna sehingga aplikasi tersebut mudah digunakan pengguna transisi dari sistem manual ke sistem komputerisasi.

**UCAPAN TERIMAKASIH**

1. Ketua Program Studi Sistem Informatika STMIK Nusa Mandiri.
2. Dr. Naomi FeliciaTika dan Dr. Leymena Yusak, selaku owner Apotek Rapha Farma.
3. Saudari Marlenia Isma J, dan semua Staff / karyawan Apotek Rapha Farma.

**DAFTAR PUSTAKA**

- [1] M. A. W. Khairurrijal and N. A. Putriana, "Review : Medication Error Pada Tahap Prescribing, Transcribing, Dispensing, dan Administration," *Farmasetika.com (Online)*, 2018. [Online]. Available: <https://farmasetika.com/2018/02/25/medication-error-pada-tahap-prescribing-transcribing-dispensing-dan-administration/>. [Accessed: 13-Dec-2020].
- [2] S. S. Ulfah and S. R. Mita, "Medication Errors Pada Tahap Prescribing, Transcribing, Dispensing Dan Administering," *Farmaka*, vol. 15, no. 2, pp. 233–240, 2017.
- [3] www.nationalacademies.org, "Medication Errors Injure 1.5 Million People and Cost Billions of Dollars Annually; Report Offers Comprehensive Strategies for Reducing Drug-Related Mistakes," *The Grants Register 2021*, 2020. [Online]. Available: <https://www.nationalacademies.org/news/2006/07/medication-errors-injure-one-point-five-million-people-and-cost-billions-of-dollars-annually-report-offers-comprehensive-strategies-for-reducing-drug-related-mistakes>. [Accessed: 13-Dec-2020].
- [4] N. Avramova and CNN, "Woman accidentally prescribed erectile dysfunction cream for eye condition," *American Academy of Pediatrics*, 2019. [Online]. Available: <https://edition.cnn.com/2019/01/09/health/woman-eye-erectile-dysfunction-cream-intl-study/index.html>. [Accessed: 13-Dec-2020].
- [5] N. A. Tifani, "Nama Tertukar, Pasien Meninggal Karena Salah Obat," *Liputan6.Com*, 2018. [Online]. Available: <https://www.liputan6.com/citizen6/read/3573201/nama-tertukar-pasien-meninggal-karena-salah-obat>. [Accessed: 13-Dec-2020].
- [6] Syafnidawaty, "Metode Waterfall," *raharja.co.id*, 2020. [Online]. Available: <https://raharja.ac.id/2020/04/04/metode-waterfall/>. [Accessed: 13-Dec-2020].
- [7] Elisabet Y. Anggraeni and R. Irviani, *Pengantar Sistem Informasi*. Yogyakarta: CV. ANDI OFFSET, 2017.
- [8] Binus, "Mengenal Metode pembuatan sistem informasi Waterfall," <https://binus.ac.id/bandung/2019/11/mengenal-metode-pembuatan-sistem-informasi-waterfall/>. [Accessed: 13-Dec-2020].
- [9] "Volume 1 No 1 – 2015 Lppm3.bsi.ac.id/jurnal IJSE –," *Indones. J. Softw. Eng.*, vol. 1, pp. 1–10, 2015.
- [10] Suendri, "Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle," *Algoritma. J. Ilmu Komput. dan Inform.*, vol. 03, pp. 1–9, 2018.
- [11] R. L. Rahardian, L. Linawati, and M. Sudarma, "Implementasi Layanan Cloud Computing Software As a Service Pada Usaha Mikro Kecil dan Menengah," *Maj. Ilm. Teknol. Elektro*, vol. 17, no. 3, p. 365, 2018, doi: 10.24843/mite.2018.v17i03.p10.
- [12] M. Jamil, Rosihan, and A. Fuad, *Buku Ajar Cloud Computing*. Yogyakarta: Deepublish, 2016.
- [13] A. G. Grigorescu and O. Cangea, "Designing and Securing Cloud Computing Applications," vol. LXXI, no. 1, pp. 19–28, 2019.
- [14] L. Christiani, "Peluang dan Tantangan Penerapan Cloud Computing (Komputasi Awan) Sebagai Solusi Automasi Kerjasama Antar Perpustakaan," *Anuva*, vol. 2, no. 1, p. 43, 2018, doi: 10.14710/anuva.2.1.43-53.