

# Permodelan Perangkat Lunak Pemilihan Calon Pendoror Darah Potensial Berbasis Object Oriented

<sup>1</sup>Rusdi Efendi, <sup>2</sup>Aan Erlansari, <sup>3</sup>M. Reza Herliansyah

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.

<sup>1</sup>rusdi.efendi.unib.ac.id, <sup>2</sup>aan\_erlanhari@gmail.com, <sup>3</sup>rezaherliansyah1000@gmail.com

---

## Article Info

### Article history:

Received, 05/11/2021

Revised, 15/12/2021

Accepted, 25/01/2022

---

### Kata Kunci:

donor darah  
sistem informasi  
Prototype

---

### Keywords:

Blood donor  
Information System  
Prototype

---

## ABSTRAK

Palang Merah Indonesia (PMI) Kota Bengkulu memiliki peranan yang penting dalam mengumpulkan darah dari pendonor dan menyalurkannya kepada pasien yang membutuhkan di rumah sakit di Kota Bengkulu. Selama ini pendonor melakukan donor darah dengan cara menghubungi petugas PMI atau datang ke rumah sakit yang dapat melakukan donor darah. Belum terdapat basis data yang menghimpun pendonor yang potensial yang dapat digunakan pada saat donor darah dibutuhkan. Pada penelitian ini menghasilkan sebuah sistem informasi yang dapat menghimpun dan melakukan pemilihan pendonor darah potensial. Untuk mencapai hal tersebut, dalam penelitian ini dilakukan pemodelan prototype dengan diagram UML yang menghasilkan satu diagram usecase dan 12 diagram activity.

---

## ABSTRACT

The Indonesian Red Cross (PMI) of Bengkulu City has a vital role in collecting blood from donors and distributing it to patients in need at hospitals in Bengkulu City. So far, donors donate blood by contacting PMI officers or coming to hospitals that can donate blood. No database collects potential donors that can be used when blood donation is needed. This research produces an information system to collect and select potential blood donors. To achieve this, in this study, prototype modeling with UML diagrams resulted in one use case diagram and 12 activity diagrams.

*This is an open access article under the [CC BY-NC-ND](https://creativecommons.org/licenses/by-nc-nd/4.0/) license.*



---

## Penulis Korespondensi:

Rusdi Efendi,  
Program Studi Teknik Informatika,  
Fakultas Teknik,  
Universitas Bengkulu.  
Email: <sup>1</sup>rusdi.efendi.unib.ac.id

---

## 1. PENDAHULUAN

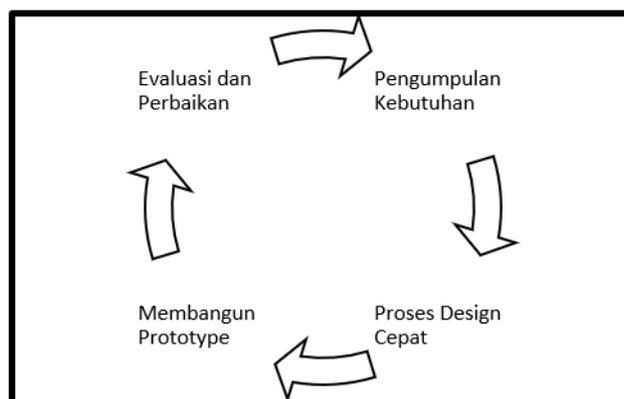
Darah adalah kendaraan untuk transport masal jarak jauh dalam tubuh untuk berbagai bahan antara sel dan lingkungan eksternal antara sel-sel itu sendiri[1]. Donor darah adalah kegiatan memberikan sebagian darah kepada seseorang secara sukarela[2],[3]. Dengan melakukan donor darah secara rutin, dapat meningkatkan regenerasi darah, oksidasi menjadi lebih lambat, dan aliran darah menjadi lebih lancar[3],[4]. Donor darah diatur dalam Peraturan Pemerintah N0. 7/ 2011 tentang Pelayanan Darah. Dalam Peraturan Pemerintah itu disebutkan penyelenggaraan donor darah dan pengolahan darah dilakukan oleh Unit Donor Darah (UDD) yang diselenggarakan oleh organisasi sosial dengan tugas pokok dan fungsinya di bidang Kepalangmerahan atau dalam hal ini Palang Merah Indonesia (PMI)[5]. Setiap tahunnya, PMI menargetkan hingga 4,5 juta kantong darah sesuai dengan kebutuhan darah nasional[6].

Jumlah populasi pada Kota Bengkulu terdapat 369.500 jiwa[7]. Dengan jumlah populasi yang ada, seharusnya kebutuhan darah yang ada di Kota Bengkulu akan tercukupi. Unit tranfusi darah PMI kota Bengkulu sering sekali mengalami kelangkaan stok darah ketika masyarakat membutuhkan. Unit tranfusi darah PMI kota Bengkulu belum memiliki basis data pendonor darah yang potensial yang dapat dihubungi pada saat kebutuhan darah diperlukan. Metode yang digunakan ketika stok darah dibutuhkan adalah menghubungi beberapa orang secara acak atau tidak terpolo untuk melakukan donor darah dan siap didistribusikan ke masyarakat yang membutuhkannya. Metode tersebut dirasa kurang efektif karena data calon pendonor yang terkumpul seringkali tidak memenuhi persyaratan donor darah.

Berdasarkan masalah yang ada, pada penelitian ini akan dikembangkan sebuah perangkat lunak yang dapat memberikan informasi daftar calon pendonor darah potensial. Untuk mendukung proses tersebut dilakukan sebuah perancangan berbasis pemodelan *object oriented*. Pemanfaatan *prototype* berfungsi untuk mendekatkan ketidaktahuan pengguna dan pengembang maka harus ada penggambar dan alat presentasi pemikiran yang berupa *object oriented*[8] menggunakan diagram *Unified Modeling Language (UML)*. Penelitian ini melakukan perancangan diagram UML dengan menggunakan diagram usecase dan diagram activity. Penelitian ini menghasilkan rancangan diagram usecase dan 12 activity diagram yakni activity diagram berdasarkan user, activity diagram lihat permintaan, activity diagram lihat respon, activity diagram tambah permintaan, activity diagram tambah respon, activity diagram lihat histori, activity diagram berdasarkan operator, activity diagram manajemen pendonor, activity diagram manajemen permintaan, activity diagram manajemen rumah sakit, activity diagram manajemen respon, activity diagram manajemen history.

**2. METODE PENELITIAN**

*Prototyping* merupakan metode pengembangan perangkat lunak, yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem[8]. Dengan metode *prototyping* ini akan dihasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem informasi[9]. Agar proses pembuatan *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan pada tahap awal, yaitu pengembang dan pengguna harus satu pemahaman bahwa *prototype* dibangun untuk mendefinisikan kebutuhan awal. *Prototype* akan dihilangkan atau ditambahkan pada bagiannya sehingga sesuai dengan perencanaan dan analisis yang dilakukan oleh pengembang sampai dengan uji coba dilakukan secara simultan seiring dengan proses pengembangan[10]. *Prototyping* dimulai dengan pengumpulan kebutuhan, melibatkan pengembang dan pengguna sistem untuk menentukan tujuan, fungsi dan kebutuhan operasional sistem. Langkah-langkah dalam *prototyping* adalah sebagai berikut



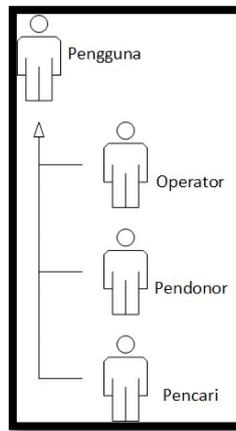
Gambar 1 Prototype

Sesuai pada gambar 1 mengumpulkan kebutuhan melibatkan pertemuan antara pengembang dan pelanggan untuk menentukan keseluruhan tujuan dibuatnya perangkat lunak; mengidentifikasi kebutuhan berupa garis besar kebutuhan dasar dari sistem yang akan dibuat. Desain berfokus pada representasi dari aspek perangkat lunak dari sudut pengguna; ini mencakup input, proses dan format output[11]. Desain cepat mengarah ke pembangunan prototipe, prototipe dievaluasi oleh pengguna dan bagian analisis desain dan digunakan untuk menyesuaikan kebutuhan perangkat lunak yang akan dikembangkan. prototipe diatur untuk memenuhi kebutuhan pengguna, dan pada saat itu pula pengembang memahami secara lebih jelas dan detil apa yang perlu dilakukannya[10].

Unified Modeling Language (UML) adalah salah satu dari metode analisis berorientasi objek dan design berorientasi objek yang muncul pada tahun 1990[12]. UML merupakan gabungan dari metode OMT dan jacobson. Tetapi UML mencakup lebih besar daripada OOAD[13]. Pada pengembangan UML, dilakukan standarisasi Object Management Group (OMG) dengan harapan UML dapat menjadi bahasa standar permodelan prototype[14]. Pemodelan dengan UML terdiri dari banyak tipe diagram yang berbeda untuk memodelkan sistem perangkat lunak, masing-masing diagram UML didesain untuk menunjukkan satu sisi dari bermacam-macam sudut pandang (perspektif) dan terdiri dari tingkat abstraksi yang berbeda, diantaranya yaitu: *Use case, Class, Object, State, Activity, Sequence, Collaboration, Component dan Deployment Diagram*[15]. Pada penelitian ini hanya digunakan dua diagram yakni *Use case dan Activity*.

### 3. HASIL DAN ANALISIS

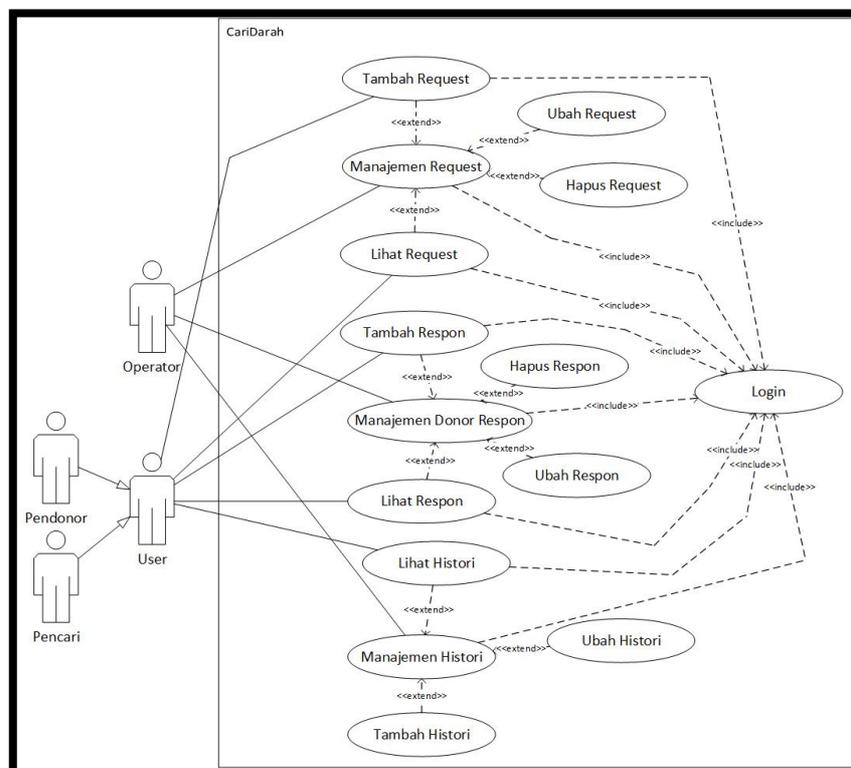
Pengguna pada perangkat lunak ini meliputi petugas unit tranfusi darah PMI Kota Bengkulu dan masyarakat kota Bengkulu. Dalam hal ini petugas PMI Kota Bengkulu berperan sebagai operator dan masyarakat Kota Bengkulu berperan sebagai pendonor dan pencari darah. Gambar 2 menunjukkan struktur generalisasi dari pengguna yang memiliki akses ke perangkat lunak.



Gambar 2 Generalisasi Pengguna Perangkat Lunak

#### 3.1 Tampilan Use Case

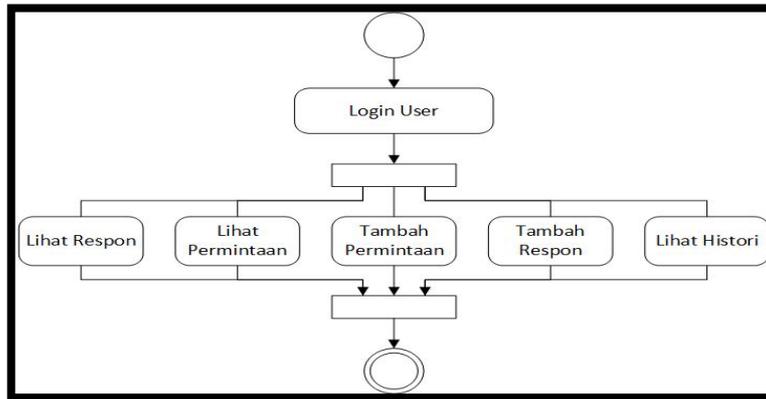
Perangkat lunak dalam pemilihan calon pendonor darah potensial akan diterapkan pada bagian user dan operator. Pengguna sistem informasi ini adalah user (masyarakat kota Bengkulu/pendonor) dan operator (pegawai PMI Kota Bengkulu). Gambar 3 menjelaskan generalisasi dari penggunaan yang dimiliki aktor ke sistem.



Gambar 3 Tampilan Use Case

3.2 Tampilan activity diagram berdasarkan user

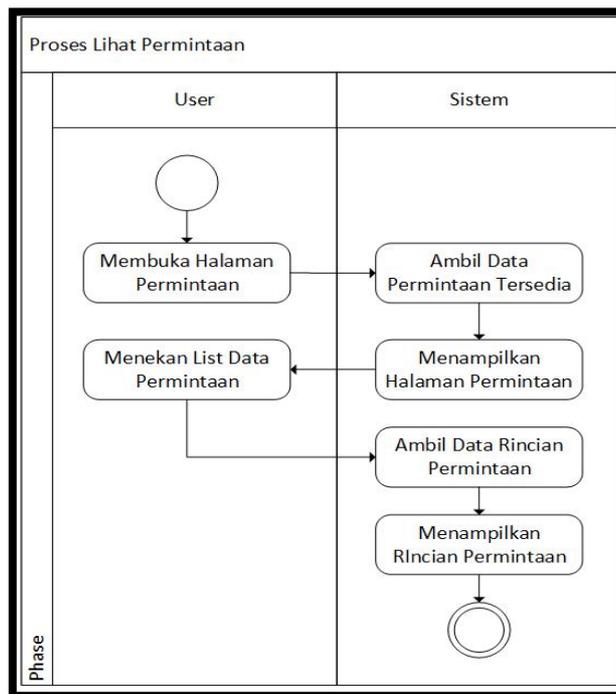
Tampilan activity diagram yang dilakukan user meliputi login, lihat respon, lihat permintaan, tambah permintaan, tambah respon, dan lihat histori. Aktivitas user tersebut dirinci menjadi 12 diagram activity yakni yakni activity diagram berdasarkan user, activity diagram lihat permintaan, activity diagram lihat respon, activity diagram tambah permintaan, activity diagram tambah respon, activity diagram lihat histori, activity diagram berdasarkan operator, activity diagram manajemen pendonor, activity diagram manajemen permintaan, activity diagram manajemen rumah sakit, activity diagram manajemen respon, activity diagram manajemen history. Gambar 4 menjelaskan tentang activity diagram berdasarkan user.



Gambar 4 Activity diagram berdasarkan user

3.3 Tampilan activity diagram lihat permintaan

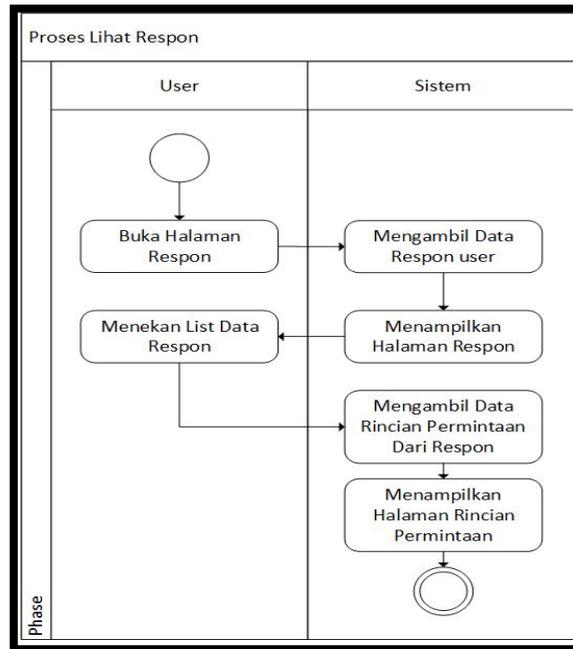
Penjelasan activity diagram berdasarkan use case diagram lihat permintaan terdapat pada gambar 5. Dimana user membuka halaman permintaan dan sistem mengambil data permintaan yang tersedia. Setelah data diambil, sistem menampilkan halaman permintaan. Tahapan selanjutnya user akan menekan list permintaan. Berdasarkan rincian permintaan dari user, sistem akan menampilkan rincian permintaan.



Gambar 5 Tampilan Activity Diagram Lihat Permintaan

### 3.4 Tampilan activity diagram lihat respon

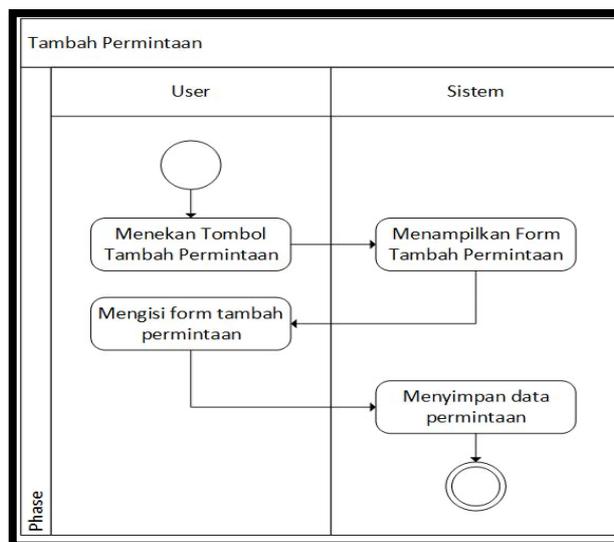
Penjelasan activity diagram berdasarkan use case lihat respon pada gambar 6. User membuka halaman respon lalu sistem mengambil data respon dari user itu sendiri dan menampilkan halaman respon. Pada halaman respon terdapat list data respon yang telah dilakukan oleh user itu sendiri. User dapat menekan list tersebut untuk melihat data permintaan. Setelah user menekan list data respon, sistem mengambil data permintaan dari respon yang ditekan lalu menampilkan halaman permintaan.



Gambar 6 Tampilan Activity DDiagram Lihat Respon

### 3.5 Tampilan activity diagram tambah permintaan

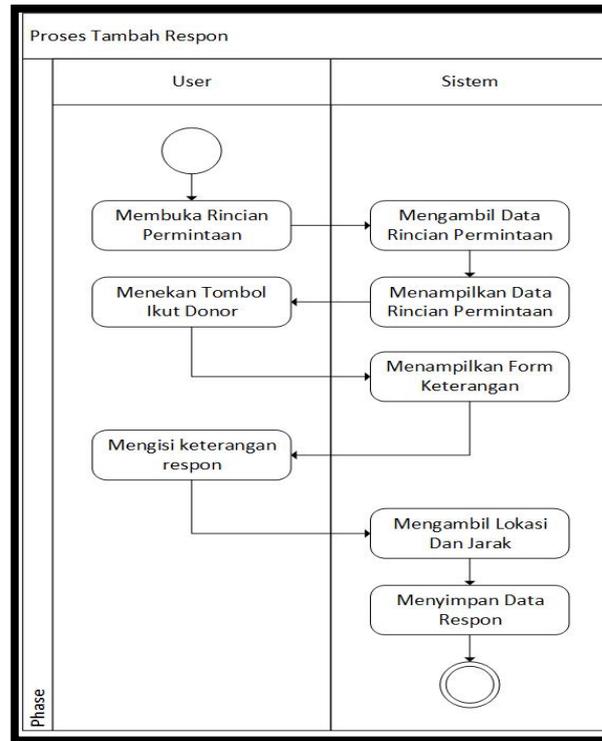
Penjelasan activity diagram pada use case tambah permintaan pada gambar 7, user menekan tombol tambah permintaan pada sistem. Lalu sistem menampilkan form tambah permintaan pada user. User akan mengisi form tambah permintaan yang akan menjadi rincian dari permintaan yang ingin ditampilkan ke user lainnya. Setelah user selesai mengisi data, sistem menyimpan data permintaan tersebut kedalam basisdata.



Gambar 7 Tampilan Activity DDiagram Tambah Respon

3.6 Tampilan activity diagram tambah respon

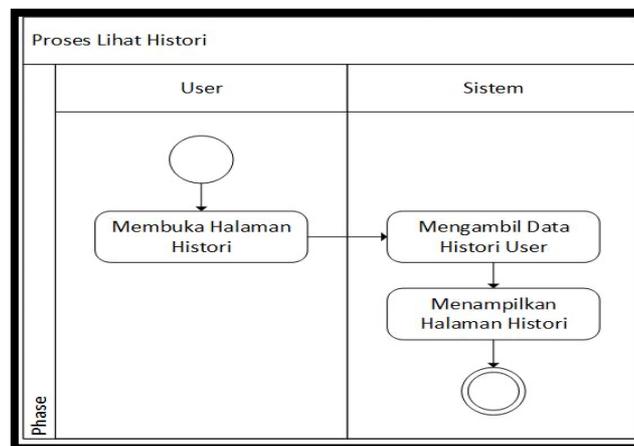
Pada gambar 8 dapat dilihat untuk memulai tambah respon, user membuka halaman rincian permintaan. Sistem akan ngambil data rincian permintaan lalu menampilkan data tersebut. Pada halaman rincian permintaan terdapat tombol ikut donor. Ketika user menekan tombol ikut donor sistem akan menampilkan form keterangan yang berisi pesan dari user yang melakukan respon kepada user yang membuat permintaan. User mengisi form lalu sistem akan mengambil data jarak dan lokasi user. Setelah itu sistem akan menyimpan data di basis data.



Gambar 8 Tampilan Activity Diagram Tambah Respon

3.7 Tampilan activity diagram lihat histori

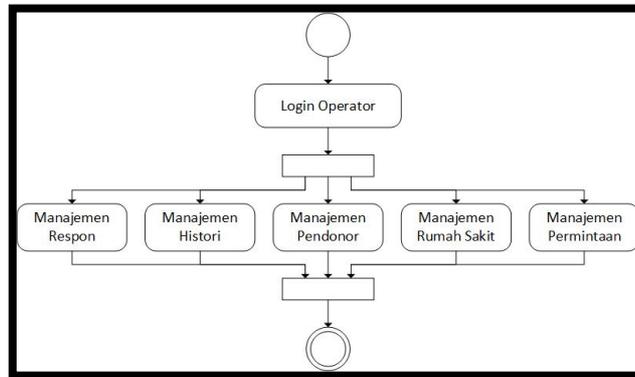
Terdapat pada gambar 9 dijelaskan user menekan tombol tambah permintaan pada sistem. Lalu sistem menampilkan form tambah permintaan pada user. User akan mengisi form tambah permintaan yang akan menjadi rincian dari permintaan yang ingin ditampilkan ke user lainnya. Setelah user selesai mengisi data, sistem menyimpan data permintaan tersebut kedalam basisdata.



Gambar 9 Tampilan Activity Diagram

3.8 Tampilan activity diagram berdasarkan operator

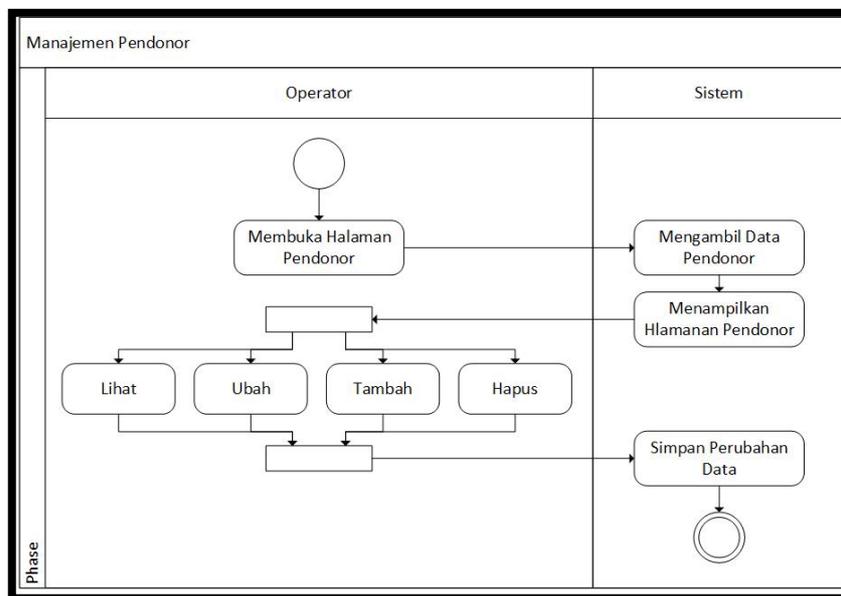
Pada gambar 10 dapat dilihat bahwa operator pertama kali melakukan login. Setelah melakukan login, operator dapat melakukan manajemen respon, manajemen histori, manajemen pendonor, manajemen rumah sakit dan manajemen permintaan.



Gambar 10 Tampilan Activity Diagram Berdasarkan Operator

3.9 Tampilan activity diagram manajemen pendonor

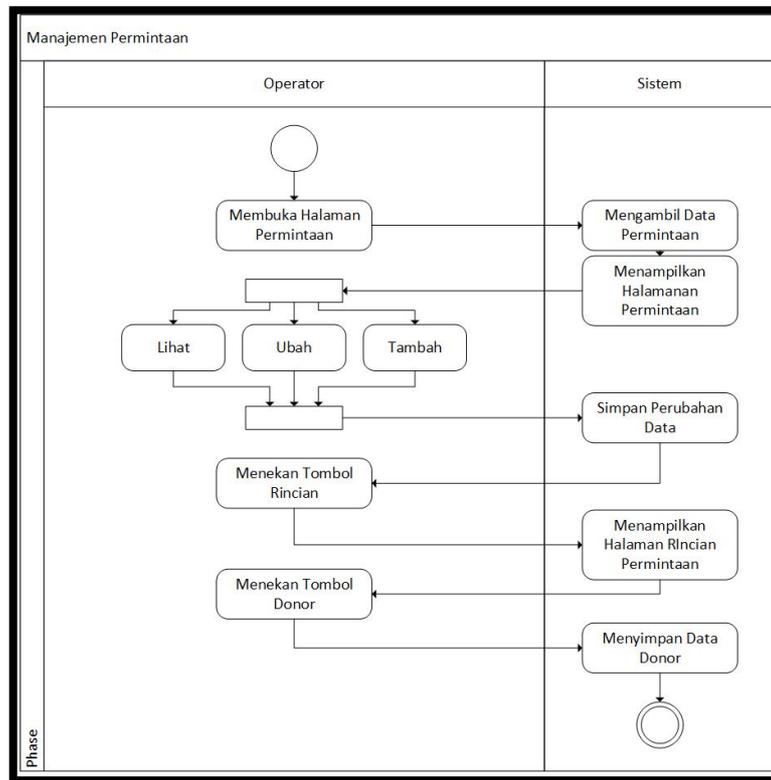
Pada gambar 11 dapat dilihat bahwa pada manajemen pendonor, pendonor membuka halaman pendonor, lalu sistem mengambil data pendonor yang terdapat pada basis data. Setelah mengambil data, sistem menampilkan halaman pendonor. Pada halaman pendonor, operator dapat melakukan lihat, ubah tambah dan hapus data pendonor. Semua perubahan yang dilakukan oleh operator disimpan oleh sistem pada basisdata.



Gambar 11 Tampilan Activity Diagram Manajemen Pendonor

3.10 Tampilan activity diagram manajemen permintaan

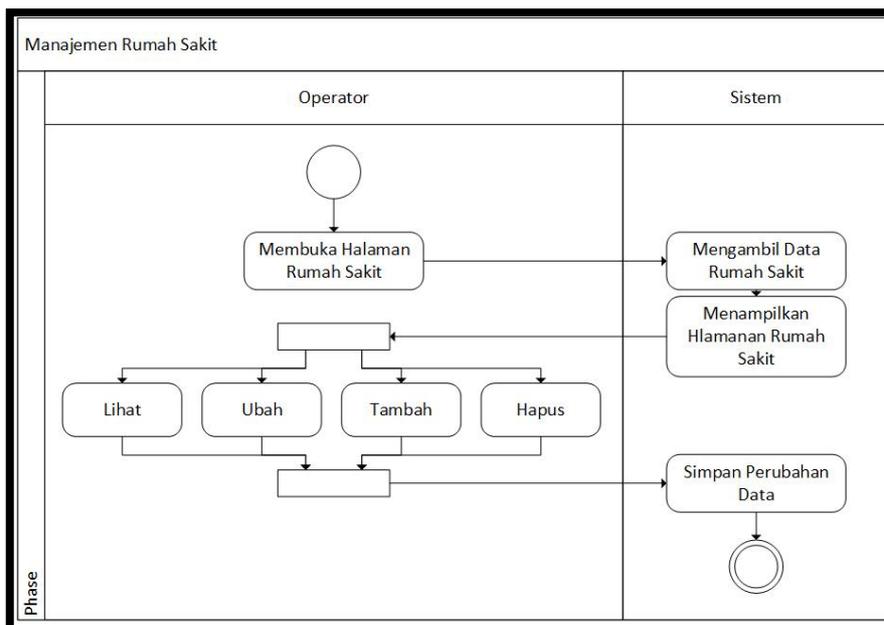
Pada gambar 12 operator membuka halaman permintaan. Sistem mengambil data permintaan yang ada lalu menampilkan di halaman permintaan. Pada halaman permintaan operator dapat melakukan manajemen data permintaan dengan melakukan lihat, tambah dan ubah. Setelah itu sistem menyimpan perubahan data. Setelah itu operator menekan tombol rincian pada halaman permintaan. Sistem menampilkan halaman rincian permintaan. Pada halaman rincian permintaan terdapat data permintaan dan data respon dari permintaan tersebut. Dari pendonor yang melakukan respon, jika melakukan donor maka operator dapat menekan tombol donor. Setelah itu sistem akan menyimpan data donor.



Gambar 12 Tampilan Activity Diagram Manajemen Permintaan

### 3.11 Tampilan activity diagram manajemen rumah sakit

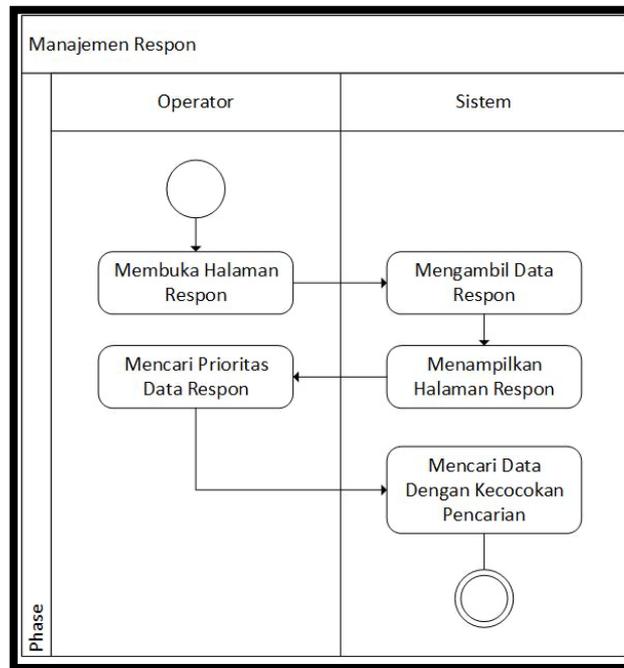
Gambar 13 menjelaskan bahwa pada manajemen rumah sakit, operator membuka halaman rumah sakit terlebih dahulu. Setelah itu sistem akan mengambil data rumah sakit dan menampilkan pada halaman rumah sakit. Pada halaman rumah sakit operator dapat melakukan lihat, ubah, tambah dan hapus. Setelah itu sistem akan menyimpan semua perubahan yang dibuat oleh operator.



Gambar 13 Tampilan Activity Diagram Manajemen Rumah Sakit

3.12 Tampilan activity diagram manajemen respon

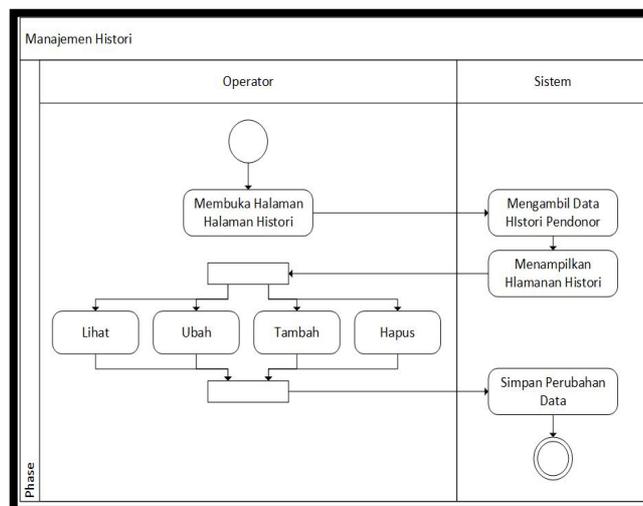
Pada gambar 14 operator dijelaskan melakukan manajemen respon. Pada operator membuka halaman respon. Sistem mengambil data respon lalu menampilkan halaman respon. Pada halaman respon terdapat form pencarian respon dengan menggunakan fuzzy tahani. Opeator mencari data prioritas dengan mengisi form pencarian. Sistem menampilkan data respon.



Gambar 14 Tampilan Activity Diagram Manajemen Respon

3.13 Tampilan activity diagram manajemen histori

Dapat dilihat pada gambar 15 operator membuka halaman histori, sistem mengambil data histori pendonor lalu menampilkan pada halaman histori. Pada halaman histori operator dapat melakukan manajemen data histori dengan cara lihat, ubah, tambah dan hapus. Setelah itu sistem menyimpan data perubahan kedalam basis data.



Gambar 15 Tampilan Activity Diagram Manajemen Histori

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dihasilkan pemodelan perangkat lunak pemilihan calon pendonor darah potensial menggunakan diagram Unified Modeling Language (UML) berupa diagram usecase dan 12 diagram activity yakni yakni activity diagram berdasarkan user, activity diagram lihat permintaan, activity diagram lihat respon, activity diagram tambah permintaan, activity diagram tambah respon, activity diagram lihat histori, activity diagram berdasarkan operator, activity diagram manajemen pendonor, activity diagram manajemen permintaan, activity diagram manajemen rumah sakit, activity diagram manajemen respon, activity diagram manajemen history

## UCAPAN TERIMA KASIH

Terima kasih kami ucapkan kepada unit tranfusi darah PMI Kota Bengkulu yang telah memfasilitasi penelitian ini.

## REFRENSI

- [1] K. Fitryadi, "Pengenalan Jenis Golongan Darah Menggunakan Jaringan Syaraf Tiruan Perceptron," *J. Masy. Inform.*, vol. 7, no. 1, p. 113618, 2017, doi: 10.14710/jmasif.v7i1.10794.
- [2] "Tentang Donor Darah | Palang Merah Indonesia." <http://www.pmimedan.or.id/pelayanan/tentang-donor-darah> (accessed Mar. 17, 2021).
- [3] A. Makiyah, "Analisis Persepsi Masyarakat Terhadap pentingnya pengetahuan donor darah bagi kesehatan," *LENTERA J. Ilm. Kesehat. dan Keperawatan*, vol. 1, no. 1, pp. 29–34, 2016.
- [4] J. W. Tang *et al.*, "A febrile blood donor," *Clin. Chem.*, vol. 56, no. 3, pp. 352–355, 2010, doi: 10.1373/clinchem.2009.131623.
- [5] "Tentang Donor Darah: 8 Syarat, Larangan, Cara, dan Manfaatnya." <https://health.detik.com/berita-detikhealth/d-4807214/tentang-donor-darah-8-syarat-larangan-cara-dan-manfaatnya> (accessed Mar. 17, 2021).
- [6] "DONOR DARAH - PMI KOTA SERANG." <https://pmikotaserang.or.id/donor-darah> (accessed Mar. 17, 2021).
- [7] *Bengkulu Dalam Angka 2020*. Percetakan Negara Republik Indonesia, 2020.
- [8] D. Purnomo, "Model Prototyping Pada Pengembangan Sistem Informasi," *JIMP - J. Inform. Merdeka Pasuruan*, vol. 2, no. 2, pp. 54–61, 2017, doi: 10.37438/jimp.v2i2.67.
- [9] "Pengertian Prototype | by Efren Manalu." <https://medium.com/@efrenkun123/prototyping-dan-penerapannya-1d6041e65a82> (accessed Mar. 17, 2021).
- [10] "Prototyping Model in Software Engineering." <https://www.guru99.com/software-engineering-prototyping-model.html> (accessed Mar. 17, 2021).
- [11] Z. Musiafa, *STUDI KASUS PROTOTYPE*. MEDIA SAHABAT CENDEKIA, 2019.
- [12] N. Medvidovic, D. S. Rosenblum, D. F. Redmiles, and J. E. Robbins, "Modeling software architectures in the unified modeling language," *ACM Trans. Softw. Eng. Methodol.*, vol. 11, no. 1, pp. 2–57, 2002, doi: 10.1145/504087.504088.
- [13] P. Sulistyorini, "Pemodelan Visual dengan Menggunakan UML dan Rational Rose," *J. Teknol. Inf. Din. Vol.*, vol. XIV, no. 1, pp. 23–29, 2009.
- [14] A. R. Pratama, "Unified Modeling Language," *CodePolitan*, Jan. 21, 2019. <https://www.codepolitan.com/unified-modeling-language-uml> (accessed Mar. 17, 2021).
- [15] M. Burrakhman, I. F. Astuti, and D. M. Khairina, "Rancang Bangun Sistem Informasi Donor Darah Berbasis Web (Studi Kasus : Unit Kegiatan Mahasiswa Korps Sukarela Universitas Mulawarman)," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 11, no. 1, p. 55, 2016, doi: 10.30872/jim.v11i1.205.

