

ANALISA DAN PERANCANGAN SISTEM APLIKASI “VACCINE-LY”

LAPORAN

Dibuat untuk memenuhi Ujian Akhir Semester
mata kuliah Analisa dan Perancangan Sistem Lanjutan



Oleh :

- | | |
|--------------------|-------------|
| 1. Alexander Toar | 00000051653 |
| 2. Aljevan Komala | 00000044016 |
| 3. Jerrell Susilo | 00000045370 |
| 4. Thomas Januardy | 00000046001 |
| 5. Vannes Lie | 00000045860 |

**JURUSAN SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA
TANGERANG
2020**

KATA PENGANTAR

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa yang telah menganugerahkan banyak hikmat dan pengetahuan sehingga kami dapat membuat dan menyelesaikan laporan ini dengan judul “Analisa dan Perancangan Sistem Aplikasi ‘Vaccine-ly’” dengan tepat waktu. Dengan segala kelancaran dan keselamatan yang diberikan-Nya dalam pembuatan laporan ini, menjadikan tugas akhir ini dapat diterima dengan baik oleh dosen mata kuliah.

Tujuan dari projek laporan tersebut adalah untuk memenuhi tugas mata kuliah Information Systems Analysis and Design. Laporan tersebut bukan hanya untuk memenuhi kewajiban kami sebagai mahasiswa, melainkan untuk menambah dan mengembangkan pengetahuan kami maupun pembaca terkait perancangan sistem dalam mata kuliah tersebut. Dalam penyusunan laporan ini, kami menyadari bahwa hasil laporan yang kami berikan ini masih jauh dari kata sempurna. Walaupun demikian, apa yang kami tulis dalam maha karya ini adalah hasil terbaik yang kami dapat berikan kepada pembaca.

Dalam menyelesaikan penyusunan laporan ini penulis banyak mendapatkan bimbingan, pengarahan dan bantuan yang sangat berharga dari berbagai pihak. Pada kesempatan ini penulis tidak lupa menyampaikan terima kasih terutama kepada:

1. Johan Setiawan, S.Kom., M.B.A dan Wira Munggana, S.Si., M.Sc. selaku dosen mata kuliah Analisis dan Perancangan Sistem Lanjutan
2. Orang tua tercinta kami yang telah memberikan dukungan baik berupa moral maupun materi serta doa demi kelancaran pembuatan laporan ini,
3. Teman-teman seperjuangan jurusan Sistem Informasi yang telah membantu dan memberikan semangat dalam penyusunan laporan ini,
4. Serta kepada semua pihak yang telah memberikan motivasi serta semangat yang tidak dapat penulis sebutkan satu per satu.

Diharapkan laporan ini dapat bermanfaat bagi penulis pada khususnya dan pembaca pada umumnya. Kami menyadari dalam menyusun laporan ini masih banyak

terdapat kekurangan, baik dari segi materi maupun dalam penyampaian materi. Oleh karena itu, penyusun mengharapkan saran dari semua pihak yang sifatnya membangun yang akan dapat berguna untuk kesempurnaan laporan ini.

Tangerang, 6 Januari 2016

Penulis

DAFTAR ISI

DAFTAR ISI	3
DAFTAR GAMBAR.....	5
DAFTAR TABEL.....	6
BAB I.....	7
1.1 The Situation	7
1.2 System Definition.....	10
BAB II	11
2.1 Class	11
2.1.1 Find Candidate.....	11
2.1.2 Select Candidate	11
2.2 Structure	17
2.2.1. Generalization	17
2.2.3. Aggregation.....	18
2.2.4. Association	19
2.2.5. Class Diagram	22
2.4. Behaviour	23
2.4.1 Statechart diagram	23
2.4.2 Behavioral Pattern.....	26
2.5. Relational and Role Pattern.....	27
2.5.1. Relational Pattern	27
2.5.2. Role Pattern	28
BAB III	30
3.1 Usage	30
3.1.1. Use Case.....	30
3.1.2. Actor Tables	35
3.1.3. Activity Diagram	37
3.2 Function	39

3.3 Interface.....	41
BAB IV	42
4.1. <i>Criteria</i>	42
4.2. <i>Component Architecture</i>	45
4.3. <i>Process Architecture</i>	47
BAB V.....	49
5.1 Model component	49
5.2 Function component	56
5.3 Connecting Component.....	58
BAB VI	66
LAMPIRAN.....	68
USER INTERFACE.....	68

DAFTAR GAMBAR

Gambar 1 Rich Picture User.....	8
Gambar 2 Rich Picture Admin	9
Gambar 3 Generalization	18
Gambar 4 Aggregation.....	19
Gambar 5 Association Structure 1	20
Gambar 6 Association Structure 2	20
Gambar 7 Association Structure 3	21
Gambar 8 Association Structure 4.....	21
Gambar 9 Association Structure 5	22
Gambar 10 Class Diagram	22
Gambar 11 Statechart User	23
Gambar 12 Statechart Dokter	23
Gambar 13 Statechart Nakes	24
Gambar 14 Statechart Lokasi	24
Gambar 15 Statechart Sertifikat Vaksin.....	25
Gambar 16 Statechart Jenis Vaksin	25
Gambar 17 Statechart Jadwal.....	26
Gambar 18 Statechart Administrator	26
Gambar 19 Relational Pattern	28
Gambar 20 Role Pattern.....	29
Gambar 21 Use Case Diagram	32
Gambar 22 Activity Diagram User	38
Gambar 23 Activity Admin	39
Gambar 24 Model Component	50
Gambar 25 User.....	50
Gambar 26 Jenis Vaksin.....	51
Gambar 27 Revised User Class Diagram.....	54
Gambar 28 Statechart user	55
Gambar 29 StatechartJenis Vaksin	55
Gambar 30 Statechart Jadwal	56
Gambar 32 Jadwal-Dokter-User-Nakes	59
Gambar 33 User-Admin	60
Gambar 34 User-Jadwal.....	60
Gambar 35 User-JenisVaksin	61
Gambar 36 User-Lokasi	61
Gambar 37 Dokter-Jadwal	62
Gambar 38 User-Sertifikat	63

Gambar 39 Class Diagram setelah Revisi	65
--	----

DAFTAR TABEL

Tabel 1 Candidate Class & Event	11
Tabel 2 Class & Event	12
Tabel 3 Event Table	17
Tabel 4 Behavioral Pattern	27
Tabel 5 Actor Tables	35
Tabel 6 Tabel Function	41
Tabel 7 Criteria	43
Tabel 8 User dan Jadwal	51
Tabel 9 Function List	58

BAB I

PENDAHULUAN

1.1 The Situation

Vaccine-Ly merupakan sebuah aplikasi yang berbasis vaksinasi. Vaccine-Ly memiliki beberapa fitur yang dapat dimanfaatkan oleh penggunanya. Diantaranya mendaftar untuk melakukan vaksin, tes Covid, mencetak sertifikat vaksin, serta mengecek ketersediaan vaksin. Aplikasi tersebut juga dapat digunakan untuk menunjukkan pengguna bahwa telah melakukan vaksinasi dalam memenuhi protokol kesehatan ketika ingin pergi ke tempat umum. Vaccine-Ly tersebut dapat membantu *user* dalam mencari lokasi terkait vaksinasi serta membantu menyusun jadwal vaksinasi. Aplikasi yang dapat memudahkan dalam mengetahui *track record* pengguna selama masa pandemi terkait proses vaksinasi. Dalam beberapa menu aplikasi tersebut, pengguna diberikan beberapa artikel maupun data mengenai vaksinasi seperti contoh jumlah penduduk indonesia yang sudah melakukan vaksin, dll.

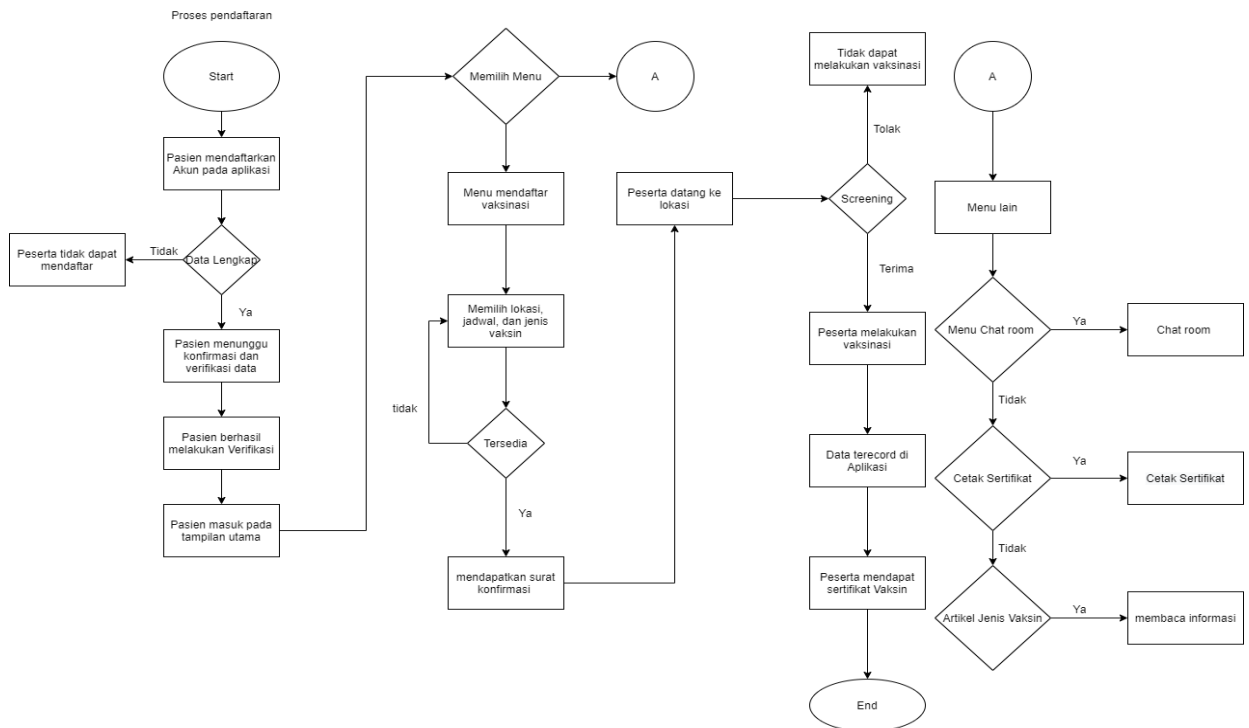
Ketika pengguna masuk ke dalam aplikasi, mereka akan melihat banyak opsi yang bisa mereka lakukan. Salah satu nya adalah mereka dapat mendaftar vaksinasi dengan cara memasukkan nik serta nama lengkap mereka. Kemudian mereka dapat memilih tempat serta jenis vaksin yang diinginkan serta memilih jadwal yang sesuai. Selain itu, pengguna aplikasi juga dapat melihat peta persebaran lokasi vaksinasi yang telah disediakan. Di tiap lokasi dicantumkan keterangan mengenai info vaksinasi maupun kuota yang masih tersedia, sehingga bisa terus diakumulasikan dengan ketersediaan vaksin terkininya. Dalam mendaftar menjadi pengguna maupun peserta vaksinasi, juga diperlukan konfirmasi kebenaran data agar data yang didaftarkan ke sistem merupakan data valid. Hal ini juga mempermudah dalam hal *scanning* dan *screening* saat proses vaksinasi.

Proses Bisnis

Ketika pengguna masuk ke dalam aplikasi, mereka akan melihat banyak opsi yang bisa mereka lakukan. Salah satu nya adalah mereka dapat mendaftar vaksinasi dengan cara memasukkan nik serta nama lengkap mereka. Kemudian mereka dapat memilih tempat serta jenis vaksin yang diinginkan serta memilih jadwal yang sesuai. Selain itu, pengguna aplikasi juga dapat melihat peta persebaran lokasi vaksinasi yang telah disediakan. Di tiap lokasi dicantumkan keterangan mengenai info vaksinasi maupun kuota yang masih tersedia, sehingga bisa terus diakumulasikan dengan ketersediaan vaksin terkininya. Dalam mendaftar menjadi pengguna maupun peserta vaksinasi, juga diperlukan konfirmasi kebenaran data agar data yang didaftarkan ke sistem merupakan data valid. Hal ini juga mempermudah dalam hal *scanning* dan *screening* saat proses vaksinasi.

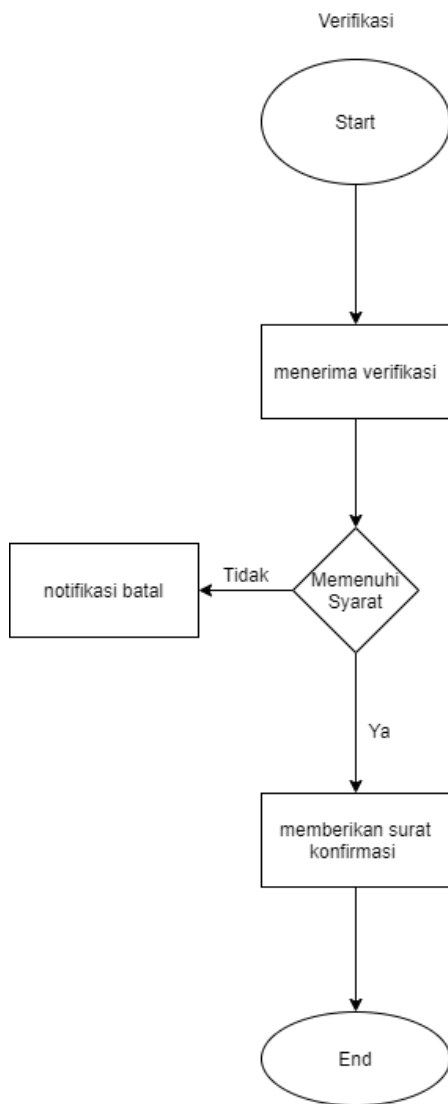
Rich Picture

User :



Gambar 1 Rich Picture User

Admin :



Gambar 2 Rich Picture Admin

1.2 System Definition

Sistem ini dirancang untuk memenuhi 6 syarat :

a. Functionally

Aplikasi yang memiliki fungsi sebagai yang memudahkan user/masyarakat dalam mencari lokasi terkait vaksinasi serta membantu menyusun jadwal vaksinasi

b. Application Domain

Melakukan screening pasien, data diri, konfirmasi vaksinasi, serta sertifikat vaksinasi.

c. Condition

Tenaga Kesehatan maupun peserta vaksin wajib mematuhi protokol kesehatan dan ketentuan/syarat vaksin, untuk menjaga keamanan dalam melakukan vaksinasi

d. Technology

Application, Data, Medical Tools serta gadget penunjang lainnya

e. Object

Aplikasi, Dokter, Nakes, Pasien, sertifikat, Administrator

f. Responsibility

Menyediakan aplikasi yang udah untuk dipahami dan digunakan, serta pelayanan yang memadai dari segi peralatan medis, tenaga medis, dan pelaksanaan dari vaksin itu sendiri.

BAB II

PROBLEM DOMAIN

2.1 Class

2.1.1 Find Candidate

Event	Class
Mendaftar	User
Verifikasi	Dokter
Membeli	Nakes
Mengganti	Lokasi
Vaksinasi	Sertifikat Vaksin
Mencetak	Jenis Vaksin
Membatalkan	Jadwal
Memilih	Track Record
Screening	Administrator
Membayar	Pembayaran
Menggunakan	Cash

Tabel 1 Candidate Class & Event

2.1.2 Select Candidate

Event	Class
Mendaftar	User
Verifikasi	Dokter
Screening	Nakes
Mengganti	Lokasi
Vaksinasi	Sertifikat Vaksin
Mencetak	Jenis Vaksin
Membatalkan	Jadwal
Memilih	Track Record
	Administrator

Tabel 2 Class & Event

Class

(yang terpilih)

User:

User merupakan pengguna dari aplikasi tersebut. User dapat melakukan banyak hal seperti mendaftar, verifikasi, Screening, melakukan vaksinasi, dan mencetak sertifikat. User tidak berhak melakukan pembatalan pada aplikasi tersebut karena yang akan melakukan pembatalan adalah tenaga kesehatan.

Dokter:

Dokter adalah orang yang melakukan vaksinasi terhadap user atau pasien. Dokter adalah orang memiliki hak atas vaksin yang diberikan. Selama proses vaksinasi, setiap pasien akan masuk ke dalam bilik vaksinasi yang

disediakan dan setiap bilik akan ada dokter-dokter yang sudah siap untuk melakukan vaksinasi.

Nakes :

Nakes atau Tenaga Kesehatan adalah orang yang memiliki tugas sebagai ujung tombak perawatan pasien dan pelaksana vaksinasi. Nakes juga memiliki peran sebagai edukator kepada masyarakat mengenai bahaya covid dan juga pentingnya melakukan vaksinasi. Selama proses vaksinasi, mereka yang melakukan pengecekan terhadap kondisi calon penerima vaksin yang biasa disebut observasi. Kemudian mereka juga yang menjadi penjaga selama kegiatan vaksinasi berlangsung.

Lokasi:

Lokasi adalah tempat dimana user menentukan tempat dia akan melakukan vaksinasi. Pada class lokasi tersebut dapat diganti. Pada aplikasi tersebut, lokasi akan diperlihatkan kepada user yang terdekat.

Sertifikat Vaksin:

Sertifikat vaksin adalah hal yang akan terus digunakan oleh user dalam melakukan protokol kesehatan. User dapat menampilkan sertifikat vaksin apabila dibutuhkan.

Jenis Vaksin :

Jenis Vaksin sesuai dengan namanya artinya adalah Jenis Vaksin yang bisa dipilih oleh calon pasien. Vaksin yang dapat dipilih adalah vaksin yang sesuai dengan ketersediaan di setiap lokasi vaksinasi. Contohnya jika kita memilih lokasi Rumah Sakit UI, dan di lokasi tersebut terdapat Vaksin Sinovac dan Astrazeneca. Maka Jenis Vaksin yang dapat anda pilih adalah Sinovac atau AstraZeneca.

Jadwal:

User atau Calon Pasien dapat memilih jadwal yang sesuai dengan waktu user tersebut. Jadi selain dapat memilih lokasi user juga dapat memilih kapan user mau menerima vaksinnya. Namun jadwal yang dipilih sesuai dengan batch yang disediakan oleh penyedia vaksin.

Administrator:

Administrator atau yang dapat disingkat Admin, adalah orang yang berada dibalik pendataan proses vaksinasi tersebut. Admin bertugas untuk menerima dan melakukan pengecekan data user atau calon pasien yang diisi dalam aplikasi Vaccine-ly ini. Admin juga yang dapat mengecek apakah user dapat menerima vaksin atau tidak dengan bantuan sistem yang ada.

Event

(yang terpilih)

Mendaftar :

Ketika user mendaftarkan diri untuk melakukan vaksinasi, maka datanya akan diproses dan akan menunggu konfirmasi dan verifikasi data. User harus memastikan agar data yang dimasukkan lengkap dan memenuhi syarat. Setelah terverifikasi, user memilih menu mendaftar vaksinasi, memilih lokasi, serta memilih jenis vaksin yang tersedia untuk melakukan vaksinasi tersebut.

Verifikasi :

Ketika user berhasil mendaftarkan dirinya, proses berikutnya adalah verifikasi terhadap data yang diisi. Contohnya ketika mendaftarkan nama dan ktp, maka perlu ada verifikasi seperti melakukan foto diri dengan ktp yang sesuai dengan yang didaftarkan. Tujuannya untuk memastikan bahwa orang yang melakukan registrasi adalah orang yang benar.

Screening :

Ketika user sudah melakukan pendaftaran jadwal vaksinasi dan verifikasi, mereka akan datang ke tempat vaksinasi sesuai dengan jadwal yang ada. Ketika sampai disana, mereka akan melakukan proses pertama yaitu screening. Proses ini adalah sebuah proses pemeriksaan kondisi fisik dari calon penerima vaksin. Nakes akan melakukan anamnesa yang merupakan proses mengingat kembali beberapa penyakit yang pernah dialami oleh calon penerima. Nakes juga akan mengecek hal-hal seperti suhu tubuh dan tekanan darah.

Mengganti:

Pada event mengganti tersebut, lokasi dan jadwal dapat melakukan event class tersebut apabila lokasi dan jadwal dari user tersebut tidak sesuai dan tidak memungkinkan terjadi.

Vaksinasi:

Proses ini adalah proses utama dari event kami, yaitu vaksinasi. Proses ini adalah proses dimana Dokter akan menyuntikkan vaksin yang sudah dipilih oleh penerima vaksin. Vaksin akan diberikan secara intra muskular atau melalui lengan kiri yang sesuai dengan prinsip penyuntikan aman.

Mencetak:

Proses ini adalah proses yang dilakukan setelah user menerima vaksin. Ketika selesai vaksin maka penerima vaksin akan diberikan sebuah sertifikat yang menunjukkan bahwa penerima vaksin telah menerima vaksin. Sertifikat tersebut akan dikirim melalui email atau langsung muncul dalam aplikasi *Vaccine-ly*. Kemudian dalam aplikasi user dapat melakukan pencetakan sertifikat yang ada.

Membatalkan:

Proses ini adalah proses yang dapat dilakukan oleh user dan juga administrator. Dari sisi user, user dapat membatalkan jadwal yang sudah ditentukan oleh user jika ada keadaan yang memaksa user untuk membatalkan jadwal. User juga dapat membatalkan lokasi yang sudah ditentukan dan menggantinya ke lokasi yang lain jika ada satu dua hal yang perlu dilakukan. Dari sisi admin, admin dapat membatalkan jadwal yang ditentukan jika tidak sesuai dengan keadaan yang ada. Pertama, admin dapat membatalkan jadwal user jika ternyata user tidak memenuhi syarat yang diberikan oleh pihak penyedia vaksin. Yang kedua admin dapat membatalkan *appointment* user jika ternyata tempat vaksin atau jadwal yang ditentukan sudah penuh dan mengharuskan adanya penggantian jadwal dan lokasi.

Memilih:

Pada proses memilih, memungkinkan user untuk menentukan jenis vaksin yang tersedia pada saat itu, dengan menampilkan informasi terupdate dari pihak rumah sakit ataupun tempat yang menyediakan vaksinasi. Selain dapat memilih jenis vaksin, user juga dapat memilih tempat atau lokasi yang diinginkan untuk melakukan proses vaksinasi tersebut, kemudian sesudah user memilih lokasi yang diinginkan, baru user diminta untuk memilih jenis vaksin seperti yang sudah dijelaskan sebelumnya.

Event Table

Berdasarkan pemilihan class dan event yang telah kami tentukan, kami membentuk sebuah event table. Class disusun dalam bentuk horizontal sedangkan Event dalam bentuk Vertikal.

Class	User	Dokter	Nakes	Lokasi	Sertifikat	Jenis	Jadwal	Administrator
-------	------	--------	-------	--------	------------	-------	--------	---------------

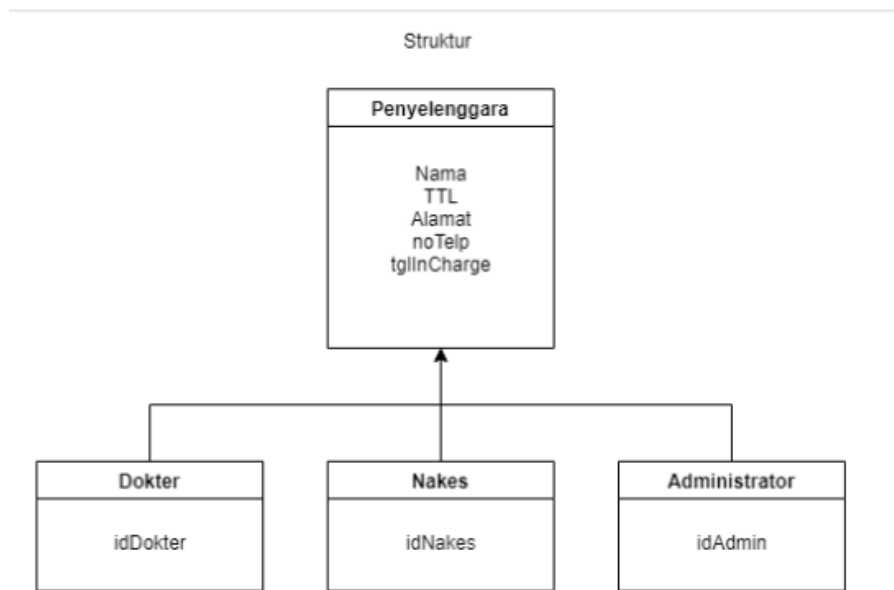
Event					Vaksin	Vaksin		
Mendaftar	√							
Verifikasi	√		√					√
Screening	√	√	√				√	
Mengganti				√			√	
Vaksinasi	√	√	√			√		
Membatalkan			√				√	
Mencetak	√				√			√
Memilih				√		√		

Tabel 3 Event Table

2.2 Structure

2.2.1. Generalization

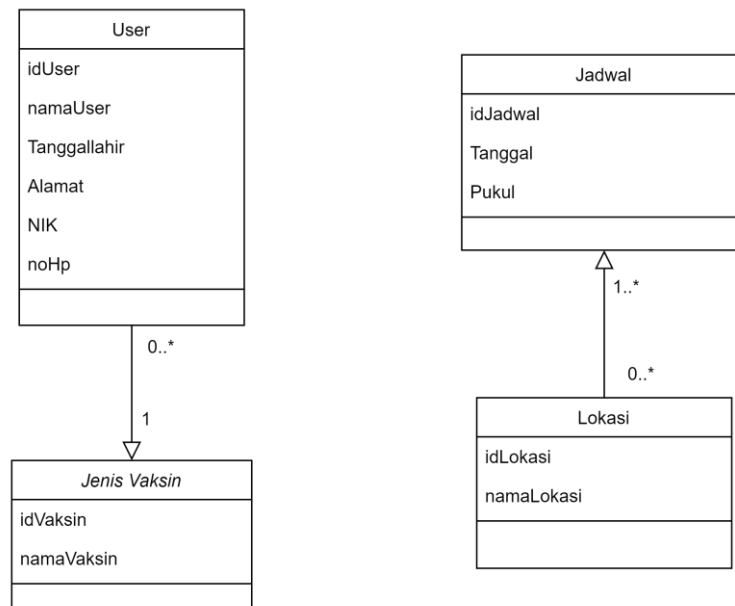
Merupakan kelas umum atau superclass dimana generalisasi menggambarkan sifat-sifat umum untuk sekelompok khusus dari subclass. Generalisasi yang telah kita bentuk terlampir sebagai berikut :



Gambar 3 Generalization

2.2.3. Aggregation

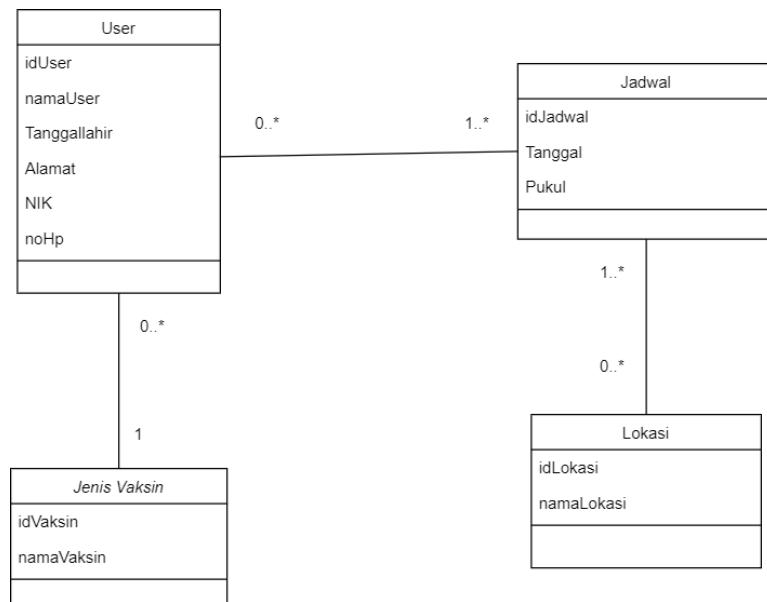
Agregasi menjelaskan keseluruhan objek terdiri dari sejumlah objek/objek bagian-bagian kecil. Berikut merupakan agregasi yang terdapat pada class diagram Vaccine-Ly



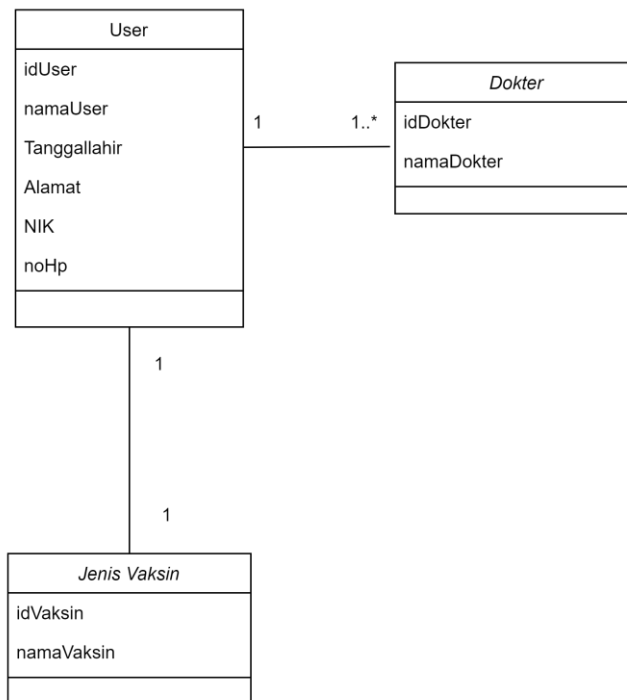
Gambar 4 Aggregation

2.2.4. Association

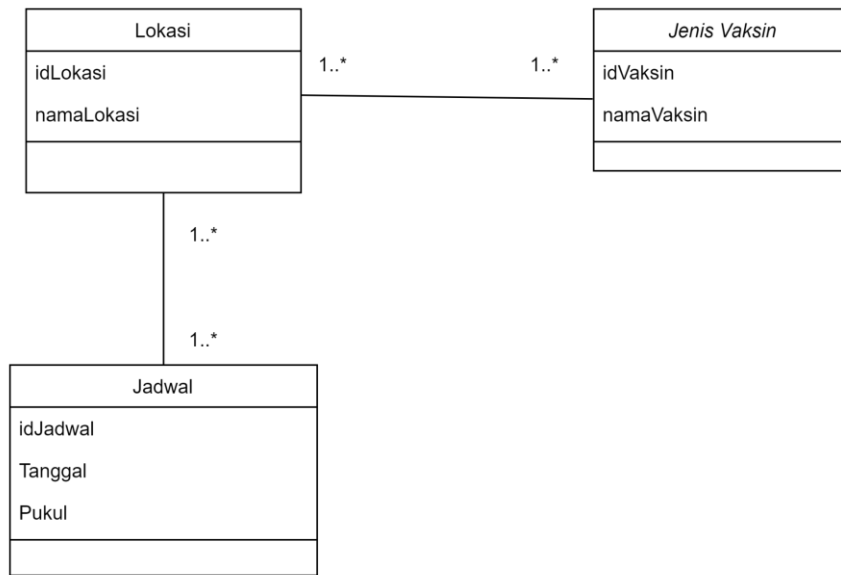
Asosiasi merupakan hubungan berarti yang dimiliki antar sejumlah objek selain generalisasi dan agregasi. Association adalah sebuah hubungan yang tercipta antara sebuah class dengan class lainnya yang memiliki saling keterhubungan. Oleh karena itu, association seringkali disebut sebagai penghubung antar class. Berikut merupakan asosiasi yang terdapat pada class diagram Vaccine-Ly.



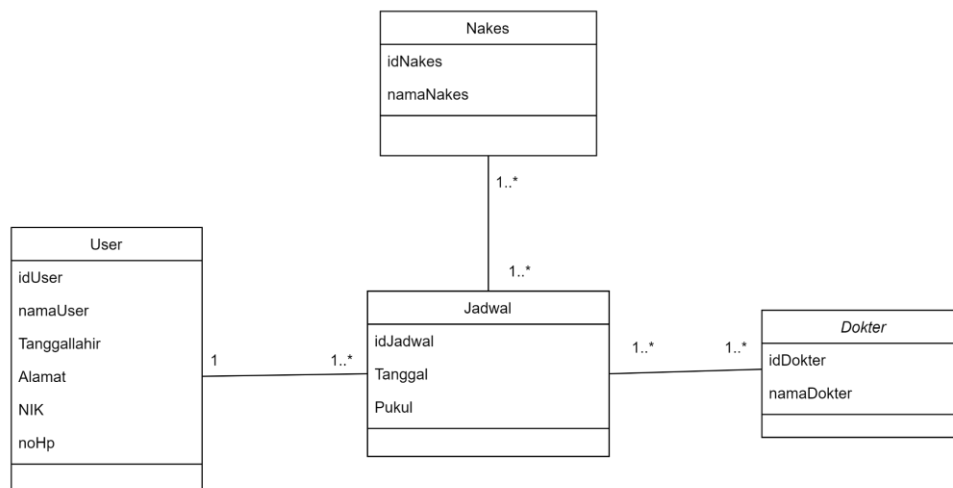
Gambar 5 Association Structure 1



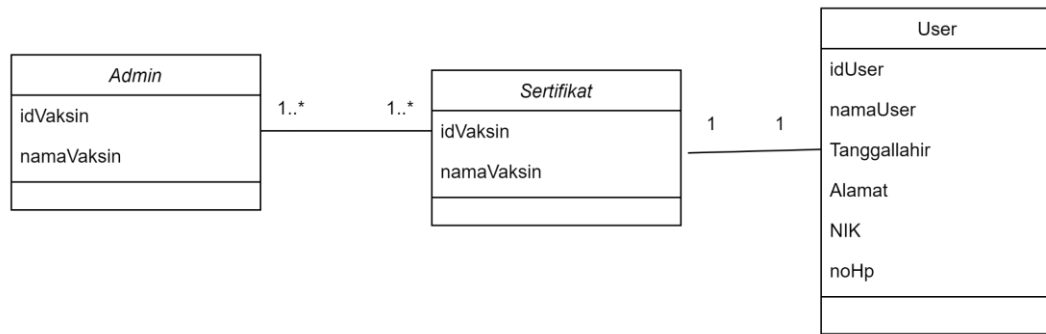
Gambar 6 Association Structure 2



Gambar 7 Association Structure 3



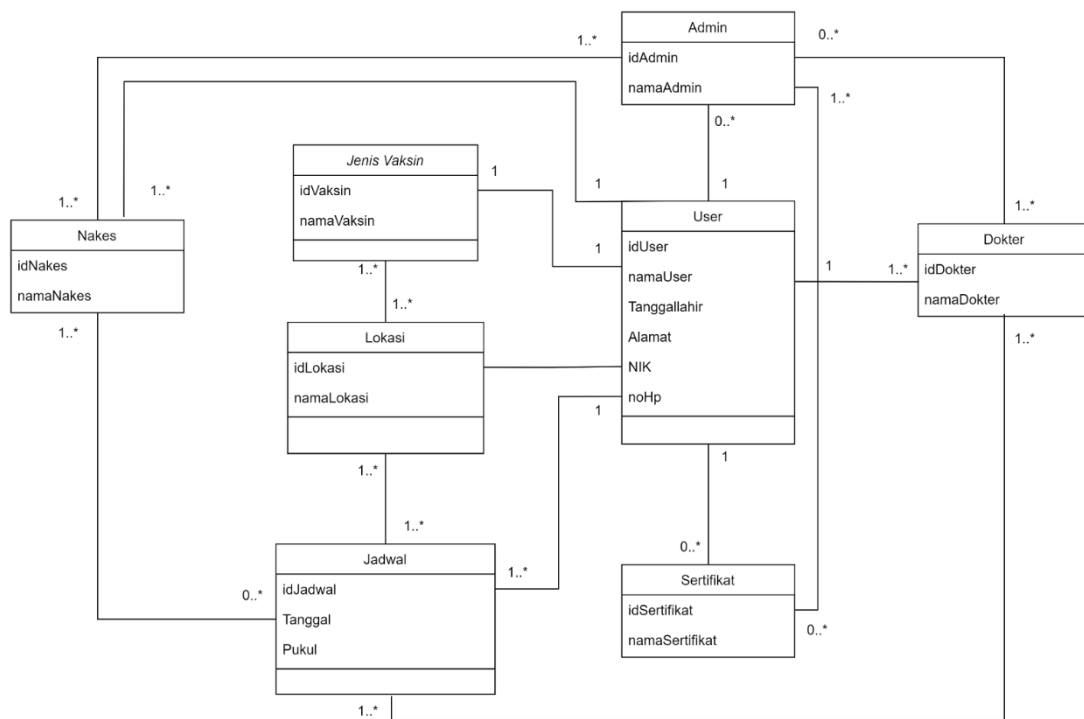
Gambar 8 Association Structure 4



Gambar 9 Association Structure 5

2.2.5. Class Diagram

Class diagram adalah diagram utama dari *Unified Modelling Language* untuk menggambarkan *class* atau blueprint object pada sebuah sistem. Berikut merupakan class diagram sistem Vaccine-Ly yang merupakan rangkaian dari class yang telah dibentuk sebelumnya:



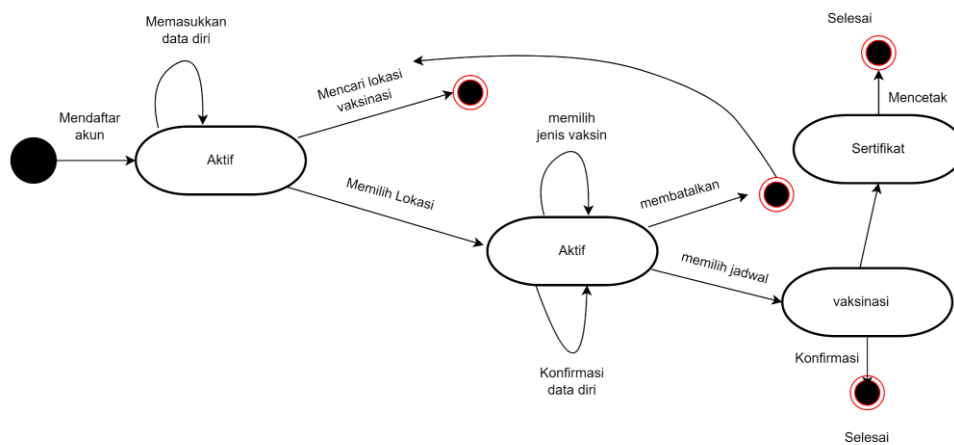
Gambar 10 Class Diagram

2.4. Behaviour

Behaviour adalah keadaan yang terjadi karena adanya hubungan dari sebuah class dengan event. Langkah pertama yang perlu dilakukan untuk membuat sebuah *event table behaviour*, diperlukan adanya *statechart diagram*. *Statechart diagram* tersebut dapat dijadikan acuan untuk menentukan event berjalan.

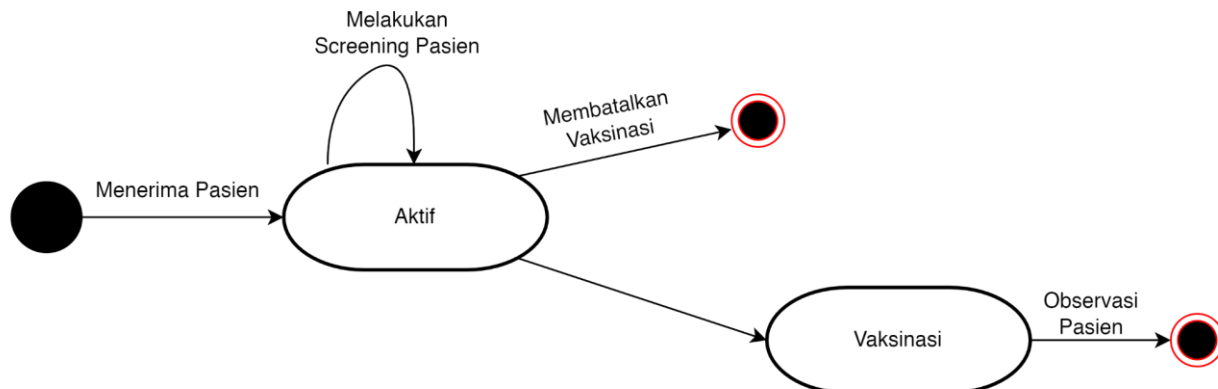
2.4.1 Statechart diagram

- User



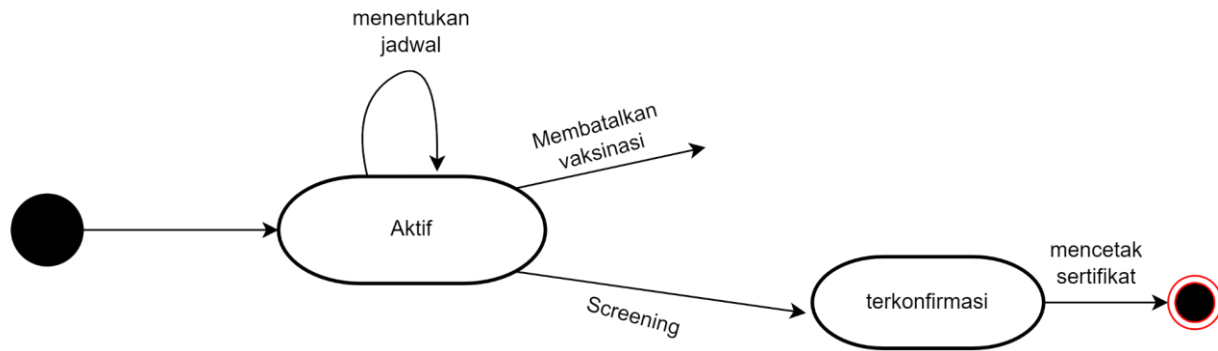
Gambar 11 Statechart User

- Dokter



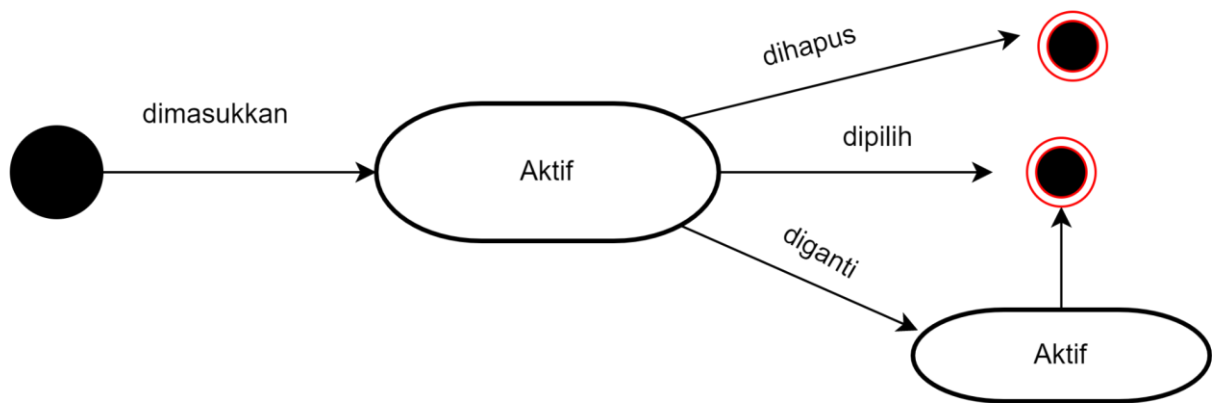
Gambar 12 Statechart Dokter

- **Nakes**



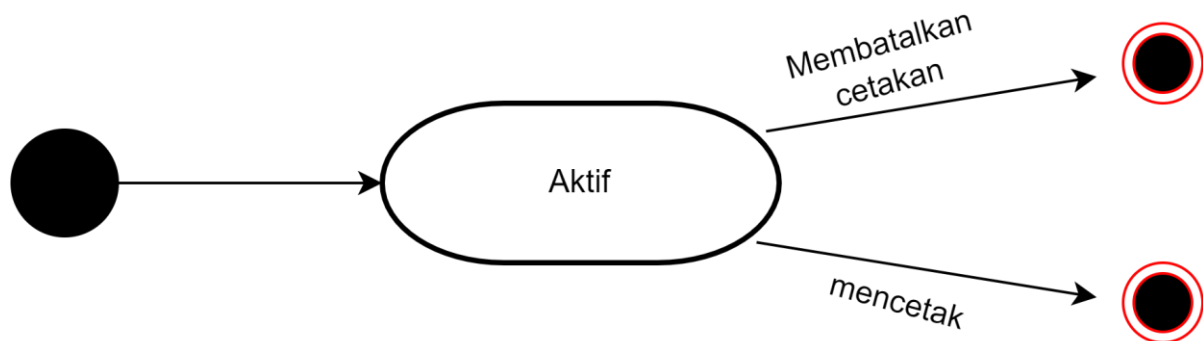
Gambar 13 Statechart Nakes

- **Lokasi**



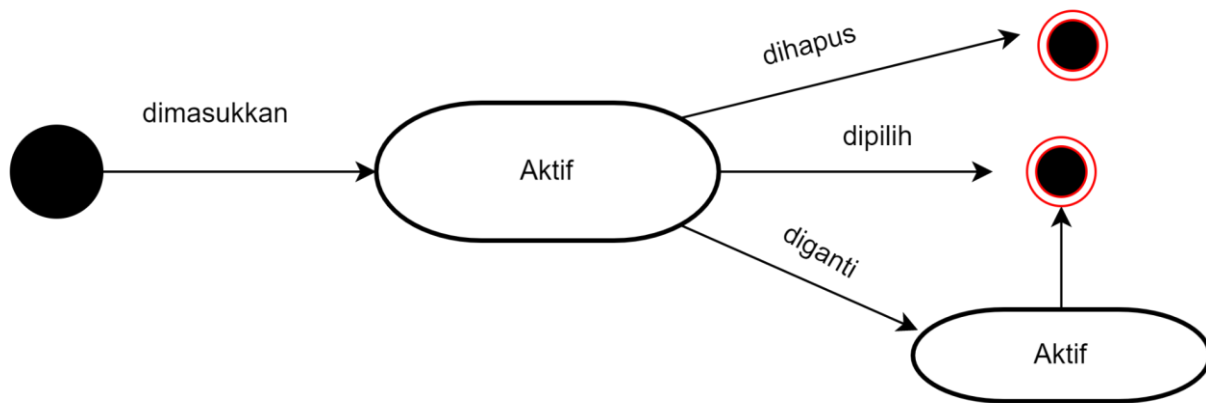
Gambar 14 Statechart Lokasi

- **Sertifikat Vaksin**



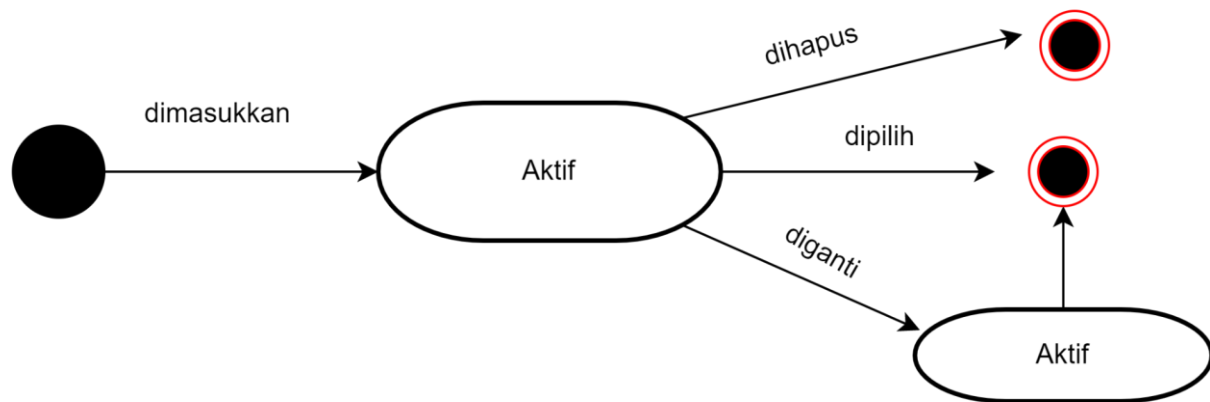
Gambar 15 Statechart Sertifikat Vaksin

- **Jenis Vaksin**



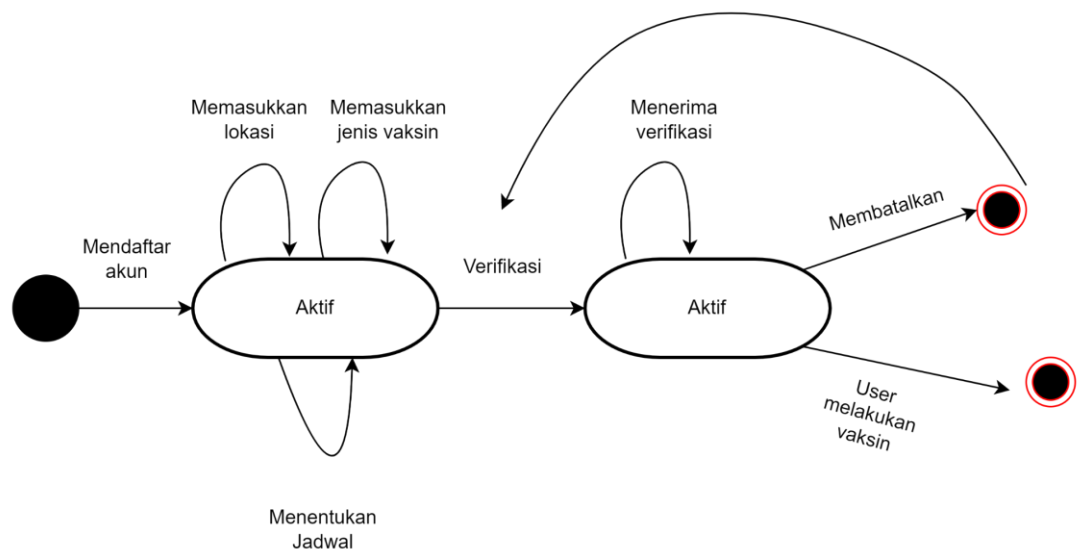
Gambar 16 Statechart Jenis Vaksin

- **Jadwal**



Gambar 17 Statechart Jadwal

- **Administrator**



Gambar 18 Statechart Administrator

2.4.2 Behavioral Pattern

Behavioral Pattern adalah pola yang didalamnya adalah komunikasi dan interaksi yang tercipta dari sekumpulan objek yang ada dalam sebuah sistem. Pola yang ada dalam pattern tersebut merupakan lanjutan pada *statechart diagram*

yang dibuat sebelumnya. Tanda tambah (+) mengindikasikan bahwa objek dari class tersebut terlibat di dalam event spesifik yang terjadi hanya sekali, sedangkan tanda bintang (*) mengindikasikan bahwa objek dari class tersebut terlibat di dalam event spesifik yang terjadi secara berulang kali. Berikut merupakan event table secara behavioral pada sistem Vaccine-Ly:

Class	User	Dokter	Nakes	Lokasi	Sertifikat Vaksin	Jenis Vaksin	Jadwal	Administrator
Mendaftar	+							
Verifikasi	+		*					*
<i>Screening</i>	*	*	*				*	
Mengganti				*			*	
Vaksinasi	+	*	*			+		
Membatal kan			*				*	
Mencetak	*				*			*
Memilih				*		*		

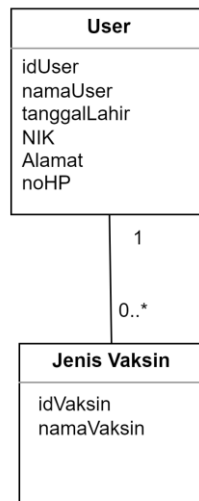
Tabel 4 Behavioral Pattern

2.5. Relational and Role Pattern

2.5.1. Relational Pattern

Relational Pattern adalah sebuah pola yang terbentuk karena sebuah class berhubungan seolah-olah membentuk sebuah hierarki. Pada sistem Vaccine-Ly,

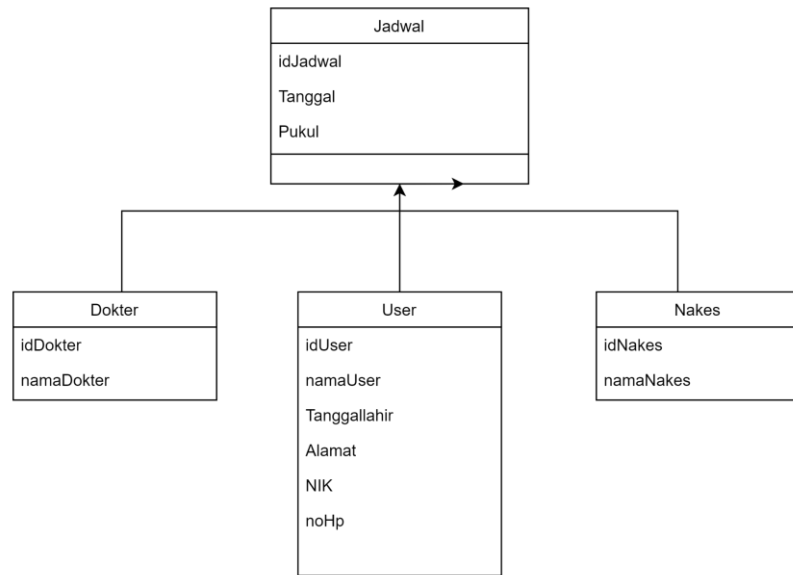
relational pattern terdapat pada class user dengan Jenis Vaksin. Tidak semua class jenis vaksin merupakan User, namun semua User akan membentuk Jenis Vaksin, sehingga User merupakan bagian dari Jenis Vaksin.



Gambar 19 Relational Pattern

2.5.2. Role Pattern

Adalah sebuah pola yang terbentuk karena adanya sebuah interaksi antar objek yang terfokus pada dimensi horizontal pada class diagram. Sistem Vaccine-Ly memiliki role pattern pada class user. Class tersebut terdiri dari masyarakat dan instansi yang memiliki peran yang berbeda-beda dalam sistem Vaccine-ly. Atribut yang terdapat pada user diwariskan kepada kedua class tersebut.



Gambar 20 Role Pattern

BAB III

APPLICATION DOMAIN

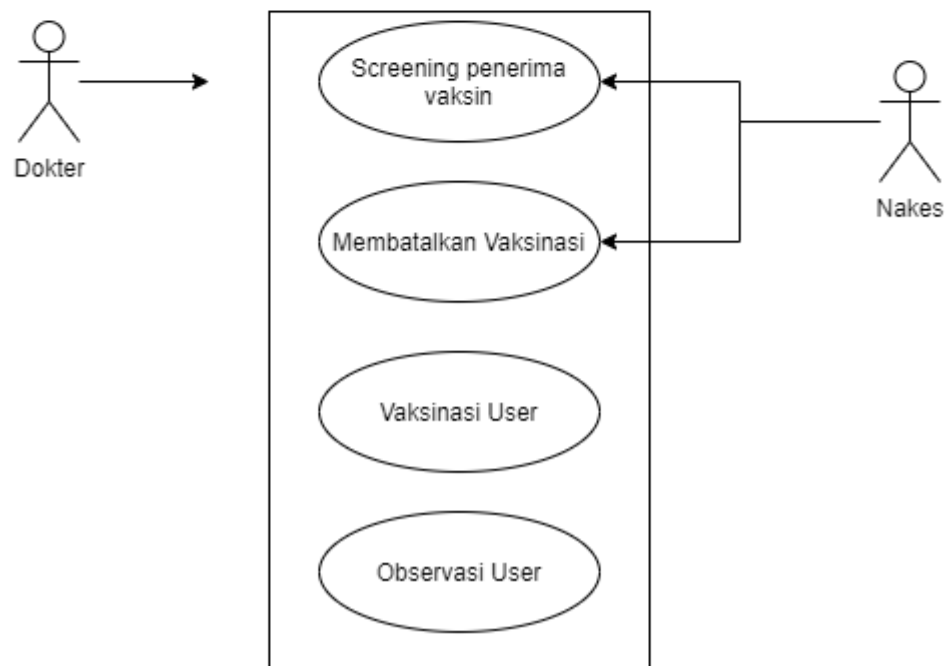
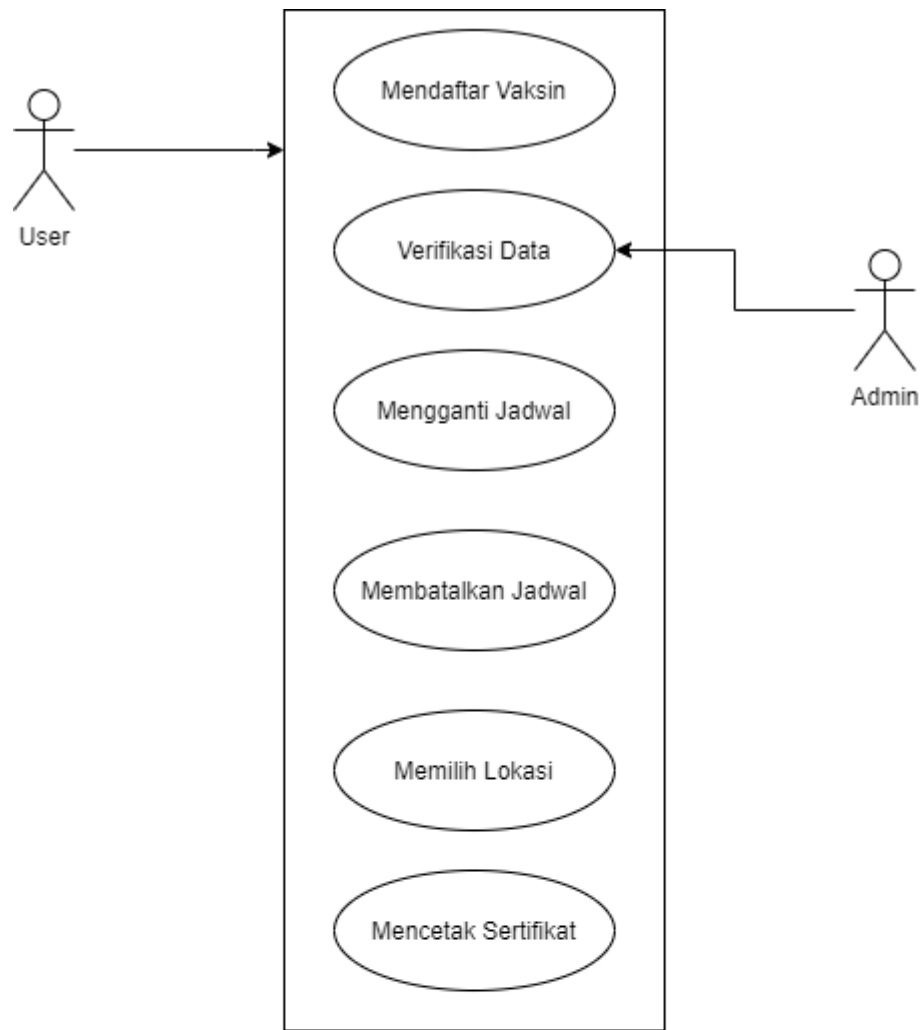
3.1 Usage

Usage memiliki tujuan untuk menentukan bagaimana *actor* berhubungan. Actor dapat berupa user, nakes, dan orang-orang yang berhubungan satu sama lain di dalam sistem yang dibuat.

3.1.1. Use Case

Use case adalah diagram yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use case* ini dapat mendeskripsikan tipe interaksi antara isi pengguna sistem dengan sistemnya. *Use case* ini adalah teknik yang menangkap kebutuhan fungsional dari sebuah sistem baru atau sistem yang diubah. *Use case* terdiri dari satu atau lebih event atau skenario yang menjelaskan bagaimana sistem yang dibuat berinteraksi dengan user atau class lain.

Berikut adalah *Use case diagram* dari Vaccine-ly :



Gambar 21 Use Case Diagram

Deskripsi use case diagram :

User
Pattern : Pada Vaccine-ly, user adalah calon pasien yang ingin melakukan berbagai hal yang berhubungan dengan vaksinasi. User dapat mendaftarkan akun baru yang nantinya dihubungkan dengan data diri, kemudian user dapat melakukan verifikasi data diri yang telah diisi oleh mereka. Kemudian user juga dapat melakukan penggantian jadwal yang telah ditentukan awalnya. User juga dapat membatalkan jadwal yang tadinya sudah ditentukan jika memiliki masalah yang mendadak. User juga bisa memilih lokasi vaksinasi yang disediakan oleh vaksinator. Yang terakhir user dapat mencetak sertifikat vaksin yang menjadi penanda bahwa user sudah di vaksin.
Object : idUser,namaUser,NIK,Alamat,NoTelp,TanggalLahir
Functions : Mendaftar Vaksin,Verifikasi Data, Mengganti Jadwal, Membatalkan Jadwal, Memilih Lokasi, Mencetak Sertifikat

Administrator
Pattern : Administrator adalah pihak yang bertanggung jawab untuk melakukan pendataan baik dari user, lokasi vaksin, jadwal, dan jenis vaksin. Admin memiliki tanggung jawab yang besar di dalam aplikasi, bisa dikatakan mereka adalah pihak yang

mengoperasikan aplikasi yang mengintegrasikan semua pihak. Aktivitas yang dapat dilakukan oleh admin adalah melakukan verifikasi data terhadap user yang mendaftar. Mereka memastikan bahwa user yang mendaftar sudah memenuhi syarat yang ada.

Object :

AdminID, Nama

Functions :

Verifikasi data

Dokter

Pattern :

Dokter adalah orang yang menyuntikan vaksin kepada penerima vaksin. Hanya mereka yang dapat memberikan vaksin. Aktivitas yang mereka bisa lakukan adalah ikut andil dalam melakukan screening terhadap pasien untuk mengetahui penyakit yang dimiliki pasien dan dampaknya ke vaksinasi. Dokter juga dapat membatalkan proses vaksinasi jika menemukan ada hal yang dialami pasien sehingga pasien tidak dapat menerima vaksin. Dokter juga yang melakukan vaksinasi kepada pasien. Dan dokter adalah orang yang melakukan observasi efek samping dari vaksinasi terhadap pasien.

Object :

idDokter, namaDokter

Functions :

Screening, Membatalkan Vaksinasi, Vaksinasi User, Observasi User

Nakes**Pattern :**

Nakes atau Tenaga Kesehatan adalah orang yang bertanggung jawab untuk memastikan penerima vaksin bisa menerima vaksin dengan aman dan nyaman. Aktivitas yang dilakukan adalah melakukan screening terhadap pasien, bisa berupa pengecekan suhu dan tekanan darah. Aktivitas lainnya adalah melakukan pembatalan vaksinasi jika pasien dirasa tidak memenuhi ketentuan yang diminta.

Object :

idNakes,namaNakes

Function :

Melakukan Screening terhadap penerima vaksin, Membatalkan vaksinasi.

3.1.2. Actor Tables

<i>Use Case</i>	<i>Actors</i>			
	User	Admin	Dokter	Nakes
Mendaftar Vaksin	√			
Verifikasi Data	√	√		
Mengganti Jadwal	√			
Membatalkan Jadwal	√			
Memilih Lokasi	√			
Mencetak Sertifikat	√			
Screening			√	√
Membatalkan Vaksinasi			√	√
Vaksinasi User			√	
Observasi User			√	

Tabel 5 Actor Tables

Deskripsi Aktor :

User
<p>Tujuan :</p> <p>User adalah orang yang dapat melakukan segala aktivitas yang ada di dalam aplikasi Vaccine-ly. Mereka juga dapat mengakses informasi terkini mengenai kondisi covid dan juga perkembangan vaksin yang ada.</p> <p>Karakteristik :</p> <p>Mendaftar vaksin, memilih lokasi, memilih jenis vaksin , Mendaftar akun, mengisi data diri, verifikasi data, cetak sertifikat</p> <p>Contoh :</p> <p>User mendaftarkan diri pada aplikasi Vaccine-ly. Setelah itu mengisi data diri yang sesuai dengan KTP kemudian melakukan pendaftaran vaksinasi. Pendaftaran vaksinasi</p>

dilakukan dengan mencari lokasi yang sesuai dengan keinginan kemudian memilih jenis vaksin yang tersedia pada lokasi tersebut. Setelah mendapat lokasi dan jadwalnya, user dapat langsung mendatangi lokasi dan menerima vaksinasi. Setelah menerima vaksinasi, user akan menerima sertifikat yang diterima di dalam aplikasi.

Admin

Tujuan :

Menjadi penanggung jawab atas segala kegiatan yang dilakukan didalam aplikasi. Admin juga bertugas untuk melakukan pemeliharaan sistem aplikasi serta melakukan pendataan.

Karakteristik :

Memverifikasi data user, memelihara sistem

Contoh:

Admin akan melakukan setiap verifikasi data terhadap user yang mengisi datanya, kemudian akan menentukan apakah sudah sesuai dengan ketentuan aplikasi yang ada atau tidak. Admin juga akan terus melakukan monitoring serta pemeliharaan sistem yang ada dan memperbaiki jika ada kesalahan. Jika ada *bug* atau *glitch* yang terjadi di dalam aplikasi, Admin harus selalu siap untuk melakukan perbaikan yang cepat sehingga user akan merasa tetap aman.

Dokter

Tujuan :

Seorang Dokter adalah orang yang bertanggung jawab untuk melakukan vaksinasi kepada penerima Vaksin. Dokter juga bertanggung jawab untuk menjaga agar penerima vaksin menerima efek samping yang minim.

Karakteristik :

Vaksinasi pasien, observasi dan screening pasien

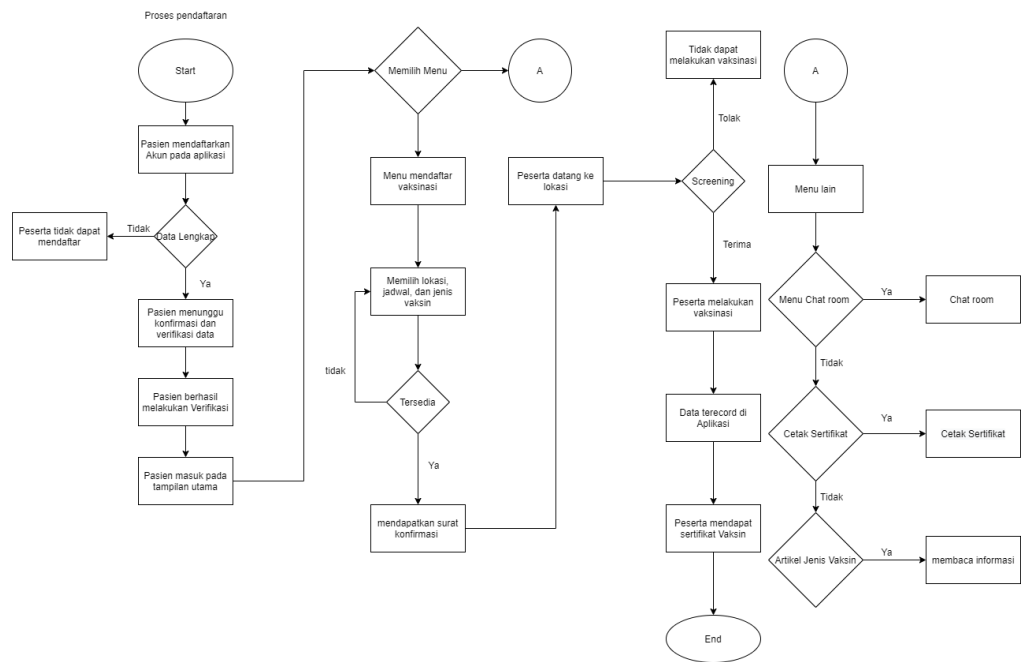
Contoh :

Seorang Dokter akan membantu Nakes dalam melakukan screening penerima vaksin saat sebelum menerima vaksin. Kemudian Dokter akan melakukan vaksinasi ketika pasien sudah menerima konfirmasi dapat menerima vaksin.

3.1.3. Activity Diagram

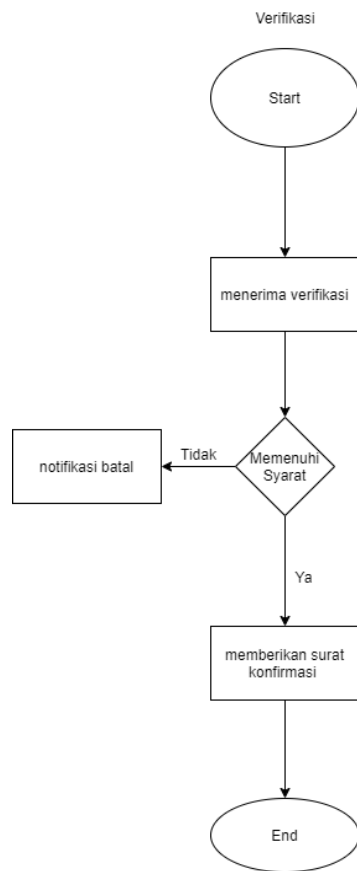
Activity diagram adalah diagram yang dapat memodelkan prose-proses yang terjadi pada sebuah sistem. Activity diagram ini menunjukkan sebuah interaksi yang ada dalam *use case* agar proses bisnis yang ada menjadi kompleks

1. User



Gambar 22 Activity Diagram User

2. Admin



Gambar 23 Activity Admin

3.2 Function

Function adalah fungsi yang paling menentukan sebuah proses atau sistem yang akan dibuat oleh developer. *Function* menentukan apakah kemampuan suatu sistem tersebut dapat memproses proses informasi dalam sistem itu sendiri. *Function* memberikan fasilitas untuk membuat sistem yang berguna bagi aktor aktor yang ada dalam sistem. Beberapa tipe tipe *function* yaitu:

1. Read:

Function read diaktifkan untuk kebutuhan akan informasi oleh aktor yang akan menghasilkan sesuatu yang akan ditampilkan oleh sistem yang relevan.

2. Update:

Fungsi yang aktif jika sebuah event berlangsung dalam problem domain tertentu, dan mengakibatkan pada perubahan status model.

3. Compute:

Function tersebut akan diaktifkan oleh kebutuhan akan informasi dalam tugas aktor dan terdiri dari komputasi yang mencakup informasi yang disediakan oleh aktor atau oleh model. Fungsi yang aktif jika aktor membutuhkan suatu informasi terkait aktivitas yang dikerjakannya.

4. Signal:

Fungsi *signal* diaktifkan pada perubahan yang ada pada status model dan menghasilkan reaksi pada konteks.

Berikut adalah *function list* yang ada dalam aplikasi Vaccine-ly beserta tipe dan kompleksitasnya :

Function	Complexity	Type
Mendaftar Vaksin	Simple	Compute
Verifikasi Data	Simple	Read
Mengganti Jadwal	Medium	Update
Membatalkan Jadwal	Medium	Update
Memilih lokasi	Simple	Read
Mencetak Sertifikat	Simple	Read

Screening penerima Vaksin	Medium	Update
Membatalkan Vaksinasi	Simple	Update
Vaksinasi User	Simple	Update
Observasi user	Simple	Update

Tabel 6 Tabel Function

3.3 Interface

Interface memiliki tujuan untuk menentukan *user interface* dari aplikasi. *Interface* juga menjadi wadah untuk mengimplementasikan model untuk sistem serta fungsi-fungsi yang tersedia bagi *actors* yang sudah disebutkan. Pada *interface* sendiri memiliki 2 fungsi dan 2 *interface* yang berbeda. Yang pertama adalah *user interface* yang merupakan sebuah *interface* yang digunakan oleh user untuk mengakses semua proses aplikasi Vaccine-ly. Yang kedua adalah *interface* yang digunakan untuk menghubungkan banyak sistem dengan sistem Vaccine-ly.

User Interface dari aplikasi Vaccine-ly untuk menjalankan model bisnis ini dilampirkan dalam lampiran.

BAB IV

ARCHITECTURAL DESIGN

4.1. *Criteria*

Kriteria pada sebuah proses desain merupakan tahapan dimana diperlukan adanya penentuan prioritas terlebih dahulu untuk mengetahui mana yang lebih dibutuhkan. Pada kriteria, tahapan ini dapat membantu meningkatkan efisiensi sehingga dapat menganalisis kemungkinan resiko atau biaya kedepannya.

Criteria	Very Important	Important	Less Important	Irrelevant	Easily Fulfilled
Usable	✓				
Secure	✓				
Correct	✓				
Reliable	✓				
Maintenable	✓				
Testable		✓			
Comperhensible		✓			

Reusable				✓	
Portable				✓	

Tabel 7 Criteria

1. Very Important

a. Usable

Sesuai dengan arti usable itu sendiri bahwa fungsi dari apa yang dibuat harus mudah untuk dipahami dan mudah untuk dioperasikan. Mengapa kita mengkategorikan usable *very important*, kembali pada tujuan utama dari aplikasi vaccine-ly itu sendiri yaitu untuk membantu dan mempermudah masyarakat untuk melakukan vaksinasi, sehingga vaccine-ly harus mudah dipahami dalam pengoperasiannya dan nyaman untuk digunakan.

b. Secure

Kita harus menjamin keamanan pada aplikasi kami ini, sebagaimana kita tahu bahwa pada aplikasi kami ini akan membutuhkan banyak sekali data pribadi dari masyarakat. Untuk itu, diperlukan perhatian khusus serta dapat menjamin bahwa tidak terjadi kebocoran atau kehilangan data. Dengan hal tersebut, maka akan menguntungkan kedua belah pihak, dimana tidak terjadi kesulitan pada memproses data bagi pihak rumah sakit, dan rasa aman terhadap datanya bagi pihak yang ingin melakukan vaksinasi. Itulah sebab kami menempatkan pada peringkat *very important*.

c. Correct

Kebenaran dan kejelasan data harus benar-benar diperhatikan dan dikedepankan, karena sebagaimana kita tahu bahwa mayoritas masyarakat, terutama di Indonesia masih kurang paham dan mendalami fungsi maupun dampak yang diberikan dari masing-masing vaksin, sehingga begitu pentingnya bagi kami untuk menyebarkan informasi yang benar dan aktual, sehingga akan mengedukasi masyarakat dan tidak membuat kebingungan kepada orang yang ingin melakukan vaksinasi. Selain dari segi

jenis vaksin, lokasi, jadwal, dll harus benar-benar akurat, sehingga masyarakat pun akan merasa nyaman saat menggunakan vaccine-ly. Untuk itu kami menempatkan *correct* pada peringkat *very important*.

d. Reliable

Seperti yang sudah disebutkan sebelumnya bahwa aplikasi kami ini harus benar-benar dapat dipercaya oleh masyarakat, dengan cara menyebarkan berita-berita serta informasi yang edukatif serta aktual, sehingga tidak ada pihak yang merasa dirugikan selama proses vaksinasi berjalan. Selain itu, aplikasi kami ini harus mampu mempertanggungjawabkan segala bentuk hasil maupun proses yang berjalan. Oleh karena itu, kami menempatkan reliable pada level *very important*, karena sulitnya untuk menjaga kepercayaan seseorang.

e. Maintainable

Tidak dapat dihindari bahwa pada kenyataannya, kesalahan ataupun *error* akan sering ditemukan pada sebuah aplikasi. Karena aplikasi vaccine-ly ini yang sifatnya harus menangkap serta mendata informasi yang sangat banyak, yang artinya sistem atau proses manajemen data akan selalu bekerja. Untuk itu, maka aplikasi kami ini harus maintainable dimana kami harus menjaga kestabilan dari memproses data sehingga tidak terjadi kekeliruan data yang tentu akan merugikan kedua belah pihak. Oleh karena itu, kami menempatkan *maintainable* pada level *very important*.

2. Important

a. Testable

Tentunya pada setiap sistem perlu adanya uji coba demi meminimalisir kejadian hal yang tidak dikehendaki. *Testing* nantinya juga dapat digunakan sebagai evaluasi terhadap kekurangan pada sistem sehingga dapat menjadi sistem yang dapat bekerja dengan baik.

b. Comprehensible

Sistem yang kami rancang diharapkan dapat dengan mudah dipahami dan mudah untuk digunakan agar pada proses pendaftaran vaksin tidak menyulitkan para pengguna. Sehingga *Comprehensible* penting demi menunjang efektivitas laju vaksinasi baik bagi tenaga medis dan masyarakat.

3. *Less Important*

a. Reusable

Reusable dinilai tidak relevan dengan aplikasi kami, aplikasi Vaccine-ly bukan sebagai aplikasi daur ulang untuk dapat didaur developer/sistem lain. Vaccine-ly membantu menemukan lokasi vaksinasi terdekat dengan mengoptimalkan efisiensi sebaik mungkin

b. Portable

Aplikasi Vaccine-ly baru direncanakan untuk digunakan pada aplikasi smartphone saja. Portable untuk aplikasi ini dianggap kurang relevan karena terbatas.

4.2. *Component Architecture*

Pada dasarnya, komponen utama dari arsitektur sebuah sistem adalah perangkat lunak (software) dan perangkat keras (hardware). Sebuah sistem yang dikembangkan dalam komponen software harus diidentifikasi untuk selanjutnya dialokasikan pada komponen hardware lain tempat sistem tersebut beroperasi. Komponen hardware dan software memiliki fungsi dasar yang berbeda-beda. Komponen software dibagi menjadi empat fungsi dasar, yaitu data storage, data access logic, application logic, dan presentation logic. Sedangkan, pada hardware terdapat tiga komponen utama, yaitu client computer, server, dan network. Component architecture berisikan struktur-struktur yang tercipta akibat adanya sebuah ikatan antar komponen yang terdapat dalam sebuah sistem. Pembuatan component architecture design harus memperhatikan tingkat kepentingan setiap komponen yang ada. *Component Architecture* Memiliki beberapa pola yang dapat digunakan dalam merancang arsitektur tersebut, antara lain :

1. *Layered Architecture Pattern*

Pada pola ini, memiliki bentuk yang paling umum pada *software*, yaitu terdiri dari beberapa komponen yang dibentuk menjadi beberapa lapisan yang mirip dengan prinsip OSI Layer pada model jaringan. Lapisan yang berada di atas tergantung kepada lapisan yang berada dibawahnya dan sebaliknya. Layered Architecture berguna untuk memecah sistem menjadi beberapa komponen.

2. *Generic Architecture Pattern*

Pada pola ini, *Generic architecture* dapat digunakan untuk menguraikan sistem dasar yang terdiri dari *interface*, *function*, dan *model component*. *Interface* berada di layer paling atas dan dilanjut oleh *function* dan *model component* dibawahnya.

3. *Client Server Architecture Pattern*

Pola ini memiliki fungsi untuk mengatasi sistem yang tersebar pada beberapa proses. Pada pola ini terdiri dari sebuah server dan beberapa client. Server memiliki kumpulan operation yang dapat digunakan oleh client sedangkan client menggunakan server nya secara independen.

Berdasarkan pola yang telah dijabarkan, Vaccine-ly menggunakan pola *Client-server*. Pemilihan pola tersebut dilatarbelakangi oleh beberapa alasan, yakni pertama Vaccine-ly merupakan aplikasi vaksinasi yang memiliki *flow* user dan *admin* dimana keduanya memiliki ketergantungan atau terintegrasi dalam sebuah *server* dimana *client* dari server tersebut adalah admin serta membutuhkan jaringan. Kedua dengan menggunakan model arsitektur client-server dapat dengan jelas membagi batasan fungsi fitur yang disertakan dalam aplikasi Vaccine-ly. Masyarakat sebagai prinsipal dapat memperoleh informasi terkait vaksinasi yang diberikan oleh institusi dan mengisi data tersebut. Selain itu, organisasi yaitu klien berperan dalam memberikan informasi dan mengolah data yang diterima dari klien. Di sisi lain, keberadaan penulis sebagai server dari aplikasi ini berperan dalam menjaga kestabilan data dari kedua arah dan mengintegrasikan data tersebut ke dalam database sehingga dapat saling terhubung.

4.3. Process Architecture

Process Architecture adalah proses struktural secara umum, yang merupakan struktur logis yang terdapat pada *component architecture*. Proses yang terdapat pada *process architecture* yaitu berdasarkan pada dua tingkat abstraksi, dimana tingkat pertama yaitu tingkat secara menyeluruh dalam mendefinisikan distribusi komponen program pada sistem prosesor yang ada, dan tingkat kedua dimana proses menjadi lebih mendalam yaitu dengan ada kaitannya dengan proses yang memiliki struktur kolaborasi antara keberadaan objek selama proses eksekusi.

Dengan merancang sistem secara mandiri, maka aktivitas proses yang berlangsung dapat diselesaikan secara efektif dan efisien. Namun, terlepas dari proses yang dapat diselesaikan secara cepat, kerumitan dari process architecture design itu sendiri dapat ditingkatkan tergantung kegunaan pada monitoring dan control system, embedded system, ataupun sistem yang memiliki interaksi signifikan dengan sistem lain. Apabila kita lihat dari bentuk daripada process architecture itu sendiri, terdapat 3 bentuk umum dari process architecture, yaitu *centralized architecture*, *distributed architecture*, *decentralized architecture*. Untuk penjelasannya lebih lanjut, sebagai berikut:

a. Centralized Architecture

Pola ini bekerja dengan menyimpan semua data di server dan fungsionalitas yang diperlukan untuk mendukung proses bisnis, tetapi klien hanya dapat melihat antarmuka pengguna. Keuntungan dari arsitektur terpusat adalah implementasi klien yang hemat biaya, konsistensi data karena berada di satu tempat, struktur yang mudah dipahami, biasanya mudah diimplementasikan, dan biasanya kemacetan jaringan yang wajar.

b. Distributed Architecture

Pola ini bekerja dengan hanya mendistribusikan model yang diperbarui ke klien. Tentu dari pola ini mempunyai keuntungan maupun kerugian. Keuntungannya salah satunya adalah cepat untuk diakses dan tidak mengganggu lalu lintas jaringan,

dan kerugiannya adalah dapat mengakibatkan duplikasi data dan persyaratan teknis yang tinggi, serta lebih susah untuk digunakan dalam mengaplikasikannya.

c. Decentralized Architecture

Pola ini bekerja dengan cara pengguna menyimpan datanya masing-masing, namun server hanya menyimpan data yang umum untuk klien. Sama seperti pola sebelumnya dimana mempunyai keuntungan maupun kerugian. Keuntungannya adalah tidak memungkinkan adanya duplikasi data, mengurangi tekanan pada lalu lintas server dan kerugian dari pola terdistribusi adalah membutuhkan perangkat keras yang dipergunakan untuk menangani hal-hal yang kompleks yang akan berpengaruh pada biaya.

Berdasarkan ketiga pola pada *Process Architecture*, aplikasi Vaccine-ly menggunakan pola *Decentralized Architecture* karena pada aplikasi Vaccine-ly hanya terdapat 1-2 kali kemungkinan untuk mendaftar vaksinasi pada lokasi tertentu sehingga ini sesuai dengan fungsi Decentralize Architecture, dimana memungkinkan untuk mengurangi duplikasi data. Selanjutnya aplikasi Vaccine-ly sesuai dengan pola ini karena pada aplikasi memiliki banyak client seperti rumah sakit/posko dengan 1 server sebagai pusatnya.

BAB V

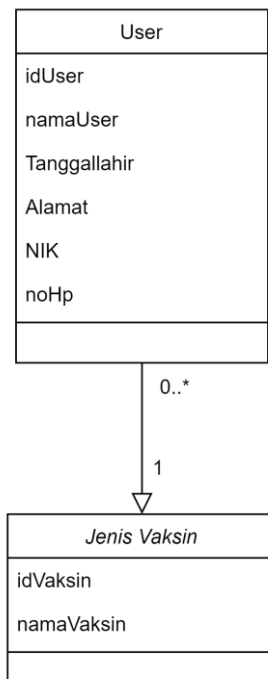
COMPONENT DESIGN

Component adalah kumpulan bagian-bagian yang mengolah keseluruhan program dan memiliki tanggung jawab yang jelas dan spesifik. Komponen dalam sistem informasi dapat diartikan sebagai sebuah kesatuan program-program di dalam sistem yang setiap tanggung jawabnya sudah terdefinisi dengan baik sehingga penggambaran sistem menjadi semakin jelas dan mudah dimengerti.

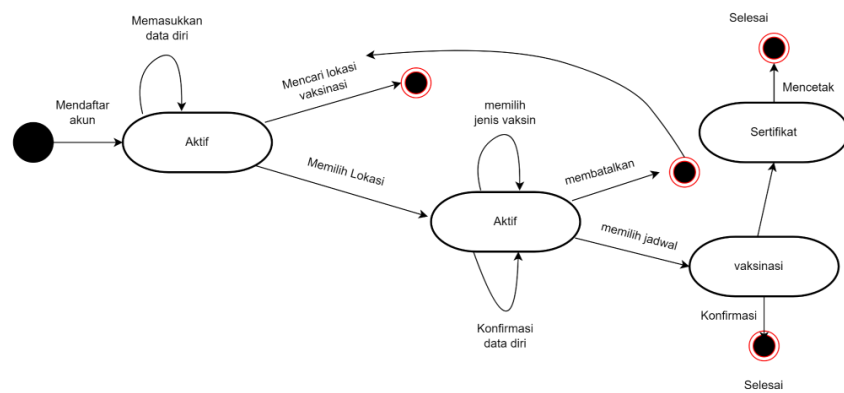
Component design terdiri dari tiga aktivitas yaitu:

5.1 Model component

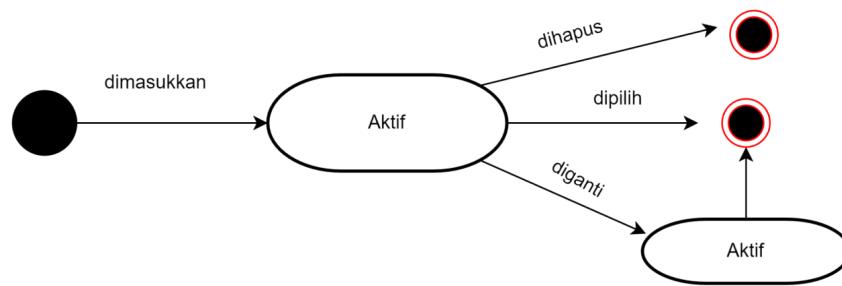
Merupakan aktivitas bagaimana sebuah model direpresentasikan sebagai class dalam sistem. Konsep dalam aktivitas ini meliputi model component dan attribute. Fungsi dari Component Model adalah menyimpan objek di dalam sistem yang merepresentasikan Problem Domain . Pada permasalahan domain program mungkin masih ada beberapa aktivitas yang belum teruraikan dengan baik sehingga diperlukan perubahan dan juga penyesuaian lanjutan terhadap masing-masing aktivitas. Dalam penerapan analisis model component perlu dilakukan penyesuaian serta penggabungan dari Class Diagram dan juga Statechart yang sudah ada. Berikut adalah penguraian Statechart dan Class Diagram terkait. Pembentukan event table bertujuan untuk mengidentifikasi private event yang terjadi pada sebuah object tertentu saja. Pembentukan event table didasarkan pada pemecahan class diagram yang sebelumnya telah dibentuk. Berdasarkan pemecahan tersebut, class yang diperoleh terdiri dari dua yaitu: User dan Jadwal. Pada sistem yang diterapkan di dalam aplikasi Vaccine-Ly.



Gambar 24 Model Component



Gambar 25 User



Gambar 26 Jenis Vaksin

Class	User	Jadwal
Event		
Mendaftar	+	
Verifikasi	+	
Screening	*	*
Mengganti		*
Vaksinasi	+	
Membatalkan		*
Mencetak	*	
Memilih		*

Tabel 8 User dan Jadwal

Pada event mendaftar dalam class User, user sebagai pengguna aplikasi hanya dapat melakukan satu kali saja yang ditandai dengan simbol (+). Hal ini dikarenakan saat mereka melakukan pendaftaran tersebut hanya membutuhkan data diri yang hanya

diisi satu kali. User tidak perlu mendaftarkan diri secara berulang ketika mendownload aplikasi pada *device* yang berbeda.

Pada Event verifikasi pada class user pula hanya perlu dilakukan sekali. Hal tersebut dikarenakan hanya membutuhkan sekali saja dalam memverifikasi data pengguna pada sistem. Hal ini juga ditujukan untuk mencegah terjadinya duplikasi data. Jadi pada event verifikasi ini, dilakukan pada saat pengguna masuk pertama kali pada aplikasi kami dan dimintai data pribadi yang diperlukan untuk keperluan aplikasi, dan pada saat itu, setelah pengguna mengisi seluruh data yang diminta, maka akan masuk pada event verifikasi data, dimana pada event ini harus memastikan bahwa seluruh data sudah terisi dengan benar.

Screening dapat dilakukan berkali-kali dikarenakan pada saat proses screening dapat dilakukan secara berkali-kali, karena pada event ini, pihak yang melakukan vaksinasi akan mengecek suhu, tekanan darah, dll kepada masyarakat, yang tentunya harus memenuhi syarat yang ditetapkan untuk dapat melakukan vaksinasi. Disaat seseorang belum memenuhi syarat, seperti ditemukan tensi darah yang tinggi, maka orang tersebut biasanya akan diminta untuk menunggu beberapa waktu dan kemudian akan melakukan screening kembali, sampai memenuhi syarat. Untuk itu kami memberi simbol * karena event ini dapat dilakukan berkali-kali.

Mengganti pada *class* jadwal kami berikan simbol * dikarenakan, jadwal yang ditentukan oleh *user* dapat diganti secara berkali-kali, sesuai dengan waktu yang diinginkan *user*. Namun tidak semena-mena *user* dapat menentukan jadwal nya sendiri, melainkan user harus menentukan jadwal yang sudah disediakan oleh pihak yang bertanggung jawab dalam melakukan vaksinasi.

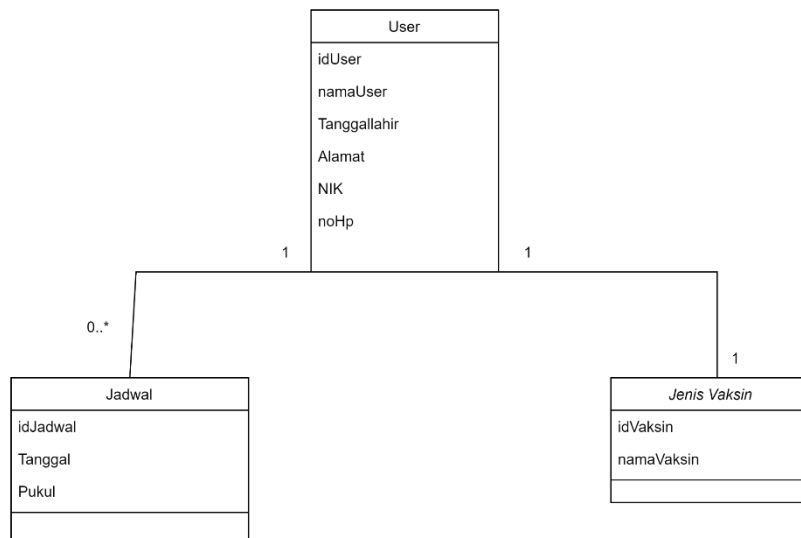
Pada event vaksinasi, *user* hanya dapat melakukan vaksinasi sebanyak satu kali saja sesuai dengan ketentuan yang ditentukan. Vaksin akan dilakukan sebanyak dua kali. Disaat *user* melakukan vaksin yang pertama kali, maka data *user* sudah secara otomatis ter-*update* pada sistem sehingga pada saat *user* ingin melakukan vaksinasi

yang kedua, tidak akan terjadi kejadian kesalahan informasi dimana vaksin yang diberikan terhitung baru vaksin yang pertama, yang tentunya akan berakibat sangat fatal kedepannya.

Membatalkan jadwal, kami memberi simbol * karena pada dasarnya fungsi nya menyerupai dengan *event* mengganti dimana *user* dapat mengganti jadwal yang sudah dipilih sebelumnya dan menggantinya ke jadwal yang baru. Untuk mengganti jadwal yang sudah dipilih sebelumnya, langkah yang harus diambil pertama kali yaitu membatalkan jadwal yang sudah dipilih sebelumnya, baru *user* dapat mengganti ke jadwal yang baru. Untuk itu, kami memberikan simbol * pada event membatalkan, karena dapat dilakukan secara berulang kali.

Mencetak kami berikan simbol * karena *user* dapat mencetak hasil vaksinasi sebanyak dua kali, sesuai dengan jumlah vaksin yang sudah diterima oleh *user*. Disaat *user* menerima vaksin nya yang pertama, maka data *user* akan ter-*update*, dan menyatakan bahwa *user* telah melaksanakan vaksin pertamanya. Saat itu juga, *user* dapat mencetak hasil dari vaksinasi pertamanya, dan hal yang sama pada vaksinasi keduanya.

Memilih jadwal kami berikan simbol * karena fungsi kerja nya sama pada event membatalkan dan mengganti, dimana *user* dapat secara berulang kali memilih jadwal yang disediakan oleh pihak yang melakukan vaksinasi.



Gambar 27 Revised User Class Diagram

Gambar-gambar di atas adalah revised class diagram dan statechart dari pemecahan class *user*. Data diri ini diisi saat masyarakat pertama kali masuk dalam aplikasi dan mendaftarkan diri. Namun terdapat revisi pada bagian ini, dikarenakan tidak semua yang melakukan pendaftaran pada aplikasi kami ini.

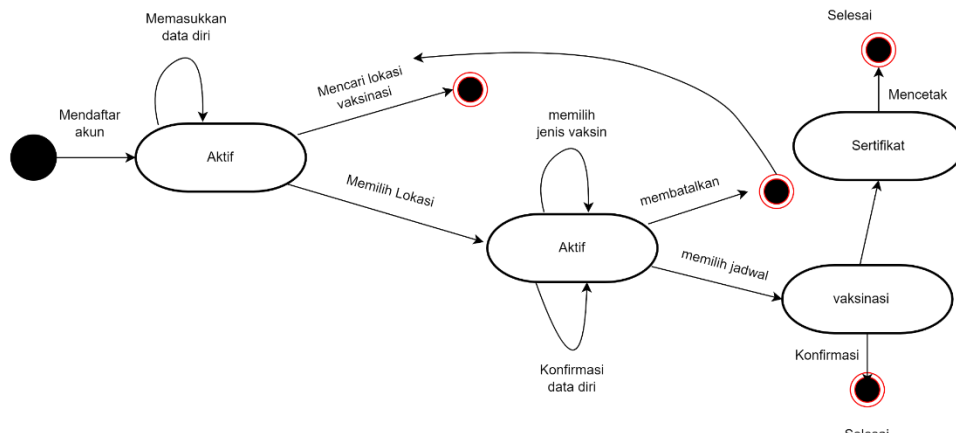
Pada class *user*, terdapat atribut `idUser`, `namaUser`, `Tanggalahir`, `Alamat`, `NIK` dan `noHp`. `idUser` merupakan atribut yang berisikan kode unik yang dimiliki oleh setiap masyarakat yang melakukan pendaftaran vaksinasi melalui aplikasi ini, dimana pada setiap kode unik tersebut mewakili satu saja orang yang mendaftar. Keberadaan atribut ini dimaksudkan agar sistem memiliki satu tabel penyimpanan khusus yang berisikan kode-kode dan informasi lainnya yang hanya berkaitan dengan aktivitas mendaftar vaksinasi.

Pada class *jadwal*, terdiri dari `idJadwal`, `Tanggal` dan `Pukul` (waktu yang ditentukan untuk melakukan vaksinasi). `idJadwal` pada *class* *jadwal* merupakan kode unik yang diberikan pada *user* saat sudah menentukan jadwal yang diinginkan. Setelah itu, akan ditunjukan tanggal serta waktu pelaksanaan vaksinasi oleh pihak rumah

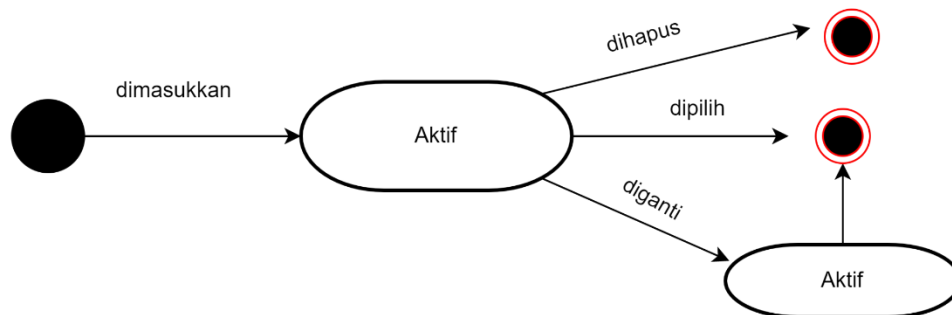
sakit/vaksinator kepada masyarakat yang sudah mendaftar melalui aplikasi Vaccine-Ly.

Pada class Jenis Vaksin, berisikan idVaksin serta namaVaksin, dimana idVaksin merupakan kode unik yang dimiliki oleh setiap nama vaksin yang ada. *User* akan memilih vaksin yang diminati, yang kemudian akan terhubung dengan tabel lainnya melalui Id yang dimiliki oleh nama vaksin yang dipilih

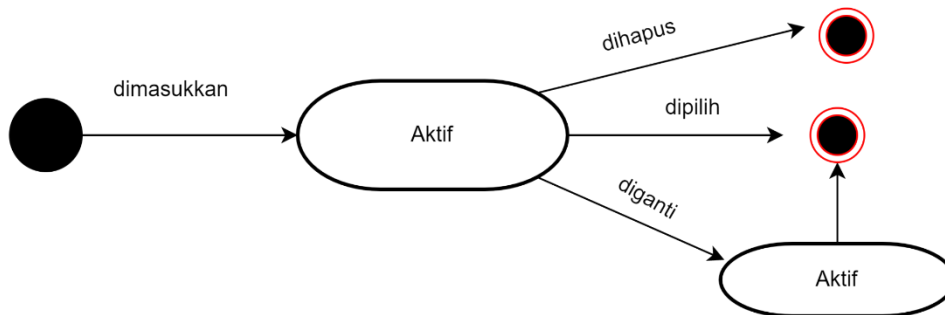
Struktur yang dapat dilihat dari ketiga tabel diatas membentuk struktur agregasi, dimana *class user* memiliki objek-objek dibawahnya yaitu jadwal dan jenis vaksin.



Gambar 28 Statechart user



Gambar 29 StatechartJenis Vaksin



Gambar 30 Statechart Jadwal

5.2 Function component

Function component diperuntukan untuk menjelaskan serta menggambarkan kepada kita mengenai fungsi-fungsi yang ada pada sebuah sistem yang nantinya akan digunakan oleh pengguna daripada sebuah aplikasi. Dalam function component ini dibagi menjadi dua bagian yaitu function component itu sendiri dan operasi dari component yang ada. Dari segi function, berfungsi atau digunakan untuk melihat serta merancang proses yang ditujukan untuk dapat digunakan oleh pengguna aplikasi dimana semua hal tersebut dilakukan dengan adanya sistem yang telah dirancang sebelumnya. Dan yang kedua yang operasi yang dilakukan harus berdasarkan pada fungsi yang digunakan pada masing-masing aplikasi. Dimana pada vaccine-ly ini kami mempunyai 4 fungsi yang akan dioperasikan yaitu

1. *Read*: bertugas untuk menangkap dan mengolah data yang diterima dari pengguna aplikasi yang akan digunakan untuk keperluan lain nya dalam prosesnya. Selain itu *read* juga berfungsi untuk menampilkan bagian-

bagian tertentu dari sistem, berdasarkan pada permintaan pengguna aplikasi

2. *Update*: berfungsi untuk melakukan pembaruan terbaru pada sistem yang disebabkan oleh inputan baru yang masuk dari pengguna aplikasi. Disaat pengguna memasukan data pada aplikasi, maka data tersebut seharusnya masuk dalam model sistem. Hal tersebut dapat direalisasikan apabila terdapat fungsi *update*.
3. *Compute*: *Compute* berfungsi pada saat pengguna ingin melihat aktivitas pada aplikasinya, seperti melihat *history*, informasi-informasi, dll. Maka *compute* akan bekerja dengan menampilkan informasi yang sebelumnya sudah diinput oleh pengguna aplikasi ataupun aktivitas pengguna dalam aplikasi tersebut ke dalam bentuk yang lebih sederhana yang tentunya sudah ter-olah yang artinya menghasilkan hasil akhir.
4. *Signal*: *signal* berfungsi saat terjadi perubahan status pada model. *Signal* juga akan fungsi pengawasan atau kontrol terhadap model, sehingga perancang sistem akan mendapatkan kemudahan untuk mencari tahu kapan harus memberikan tindakan pada perubahan problem domain.

Berdasarkan revised class diagram, terdapat 3 class yang mengalami revisi yaitu User, Jadwal, dan Jenis Vaksin.

Function	Complexity	Type
USER		
Mendaftar Vaksin	Simple	Compute

Verifikasi Data	Simple	Read
Jadwal		
Mengganti Jadwal	Medium	Update
Membatalkan Jadwal	Medium	Update
Jenis Vaksin		
Memilih jenis vaksin	Simple	Read

Tabel 9 Function List

5.3 Connecting Component

Pada connecting component dikenal sebuah istilah “coupling”. Coupling dapat diartikan sebagai sebuah negative property yang harus dicari dan diminimalkan. Coupling adalah alat ukur yang menghitung ketergantungan antar elemen dalam sebuah sistem. Jika ketergantungan yang tercipta antar objek semakin besar, maka perubahan yang terjadi pada bagian desain dapat menyebabkan perubahan di bagian yang lainnya juga (Dennis, Wixom, & Tegarden, 2015). Terdapat 4 macam coupling dalam sebuah sistem, yaitu

1. *Outside coupling*

Sebuah class atau komponen mengacu ke public properties dari class atau komponen yang berbeda secara langsung.

2. *Inside coupling*

Sebuah operasi mengacu ke private properties secara langsung dari class yang sama.

3. *Coupling from below*

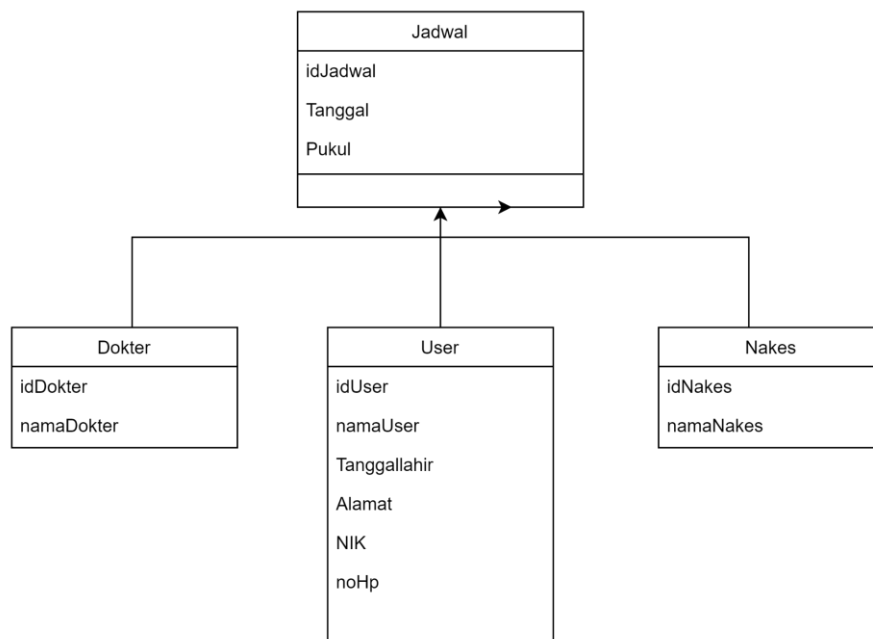
Sebuah class spesial yang mengacu ke private properties yang ada di dalam *super class* secara langsung. Komponen yang terpanggil hanya pada private properties yang terdapat pada super classnya.

4. *Sideways coupling*

Sebuah class mengacu langsung ke private properties yang terdapat pada class lain dengan tingkatan yang sejajar.

Berikut adalah penjabaran hubungan antar-class diagram yang terdapat dalam aplikasi Vaccine-Ly berdasarkan coupling”.

1. Class Jadwal - Class Dokter, User, dan Nakes

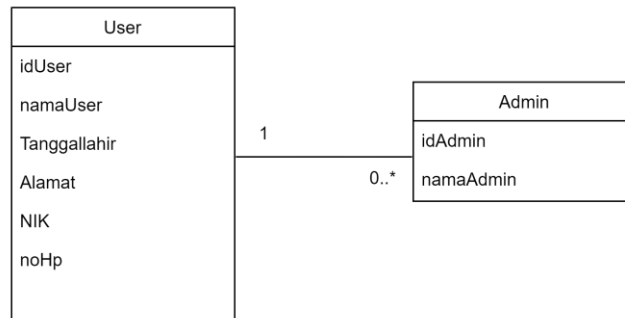


Gambar 31 Jadwal-Dokter-User-Nakes

Berdasarkan proses bisnis yang berlangsung di dalam aplikasi tersebut, *system analyst* membagi class Jadwal menjadi tiga, yaitu Dokter, User, dan Nakes. Oleh karena itu, hubungannya disebut *coupling from below*. Hal tersebut dikarenakan Dokter, User, dan Nakes merupakan subclass dari Jadwal yang merupakan *superclass*,

sehingga *subclass* mengakses private properti-properti yang terdapat dalam superclass (class Jadwal).

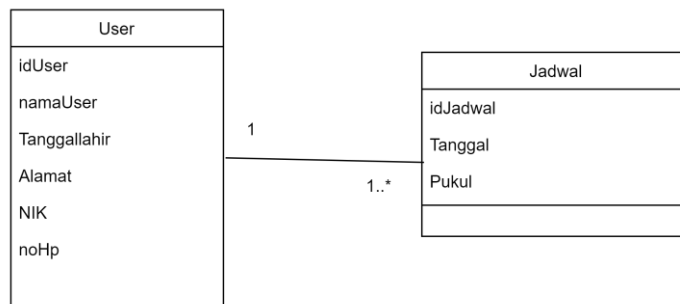
2. Class User - Class Admin



Gambar 32 User-Admin

Antara Class Admin dan Class User terbentuk sebuah hubungan yang memiliki sideways coupling dimana hal tersebut dapat ditunjukan pada private atribut yang terdapat pada Class User yang hanya dapat diakses oleh Class Admin.

3. Class User - Class Jadwal

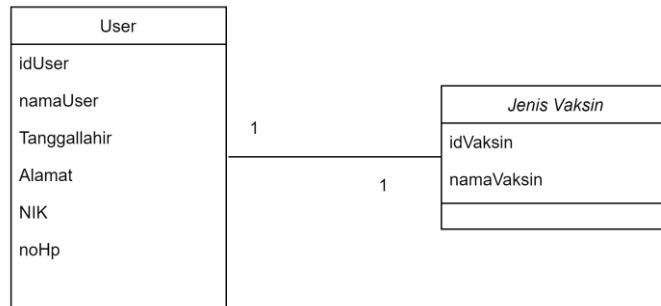


Gambar 33 User-Jadwal

Pada sistem Vaccine-Ly , hubungan yang terbentuk antara class User dengan class Jadwal adalah outside coupling. Hal ini dikarenakan setiap atribut yang terdapat dalam class jadwal mengacu kepada public attribute yang terdapat pada class user. Setiap User yang mencari

informasi vaksinasi dapat memperolehnya melalui class Jadwal yang dibentuk oleh pihak instansi.

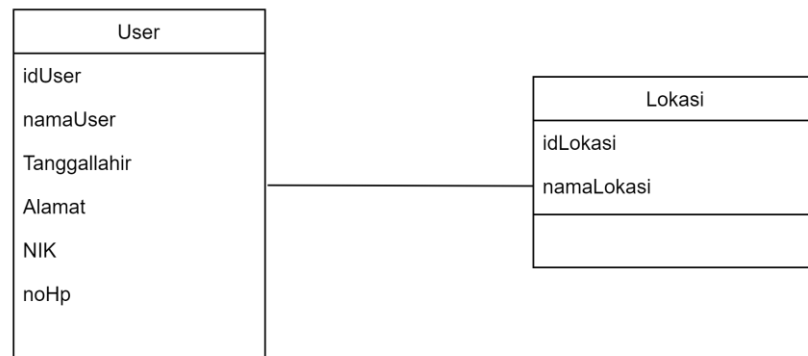
4. Class User - class jenis vaksin



Gambar 34 User-JenisVaksin

Hubungan yang tercipta antara *Class User* dengan *Class jenisVaksin* pada sistem Vaccine-Ly adalah *outside coupling*. Dikatakan demikian karena *Class User* mengacu langsung ke *public property* dari *Class jenisVaksin* yang terdiri dari *idVaksin* dan *namaVaksin*. Melalui *outside coupling*, *class user* dapat mengakses informasi yang terdapat dalam *class jenisVaksin*. Informasi yang diperoleh *user* berupa jenis vaksin yang akan digunakan nanti.

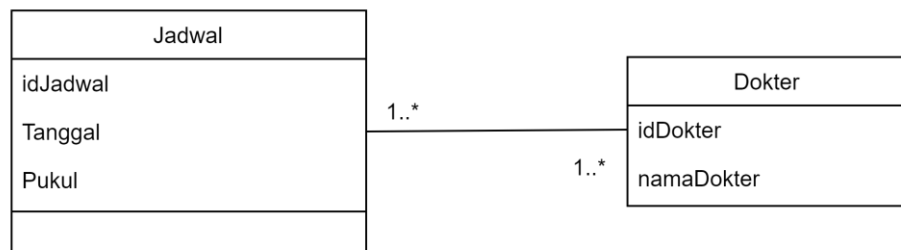
5. Class User - class Lokasi



Gambar 35 User-Lokasi

Proses bisnis yang berlangsung dalam aplikasi Vaccine-Ly ini adalah memungkinkan *user* untuk menentukan lokasi vaksin yang diinginkan, sehingga hubungan yang tercipta antara *Class user* dengan *Class* lokasi adalah *outside coupling*. Dapat dikatakan demikian, karena setiap lokasi yang dipilih oleh *user* akan tercatat pada sistem. Selain itu, *user* juga dapat memasukkan, mengganti dan menghapus lokasi mereka pada sistem Vaccine-ly, itulah alasan mengapa kedua *class* ini saling berhubungan.

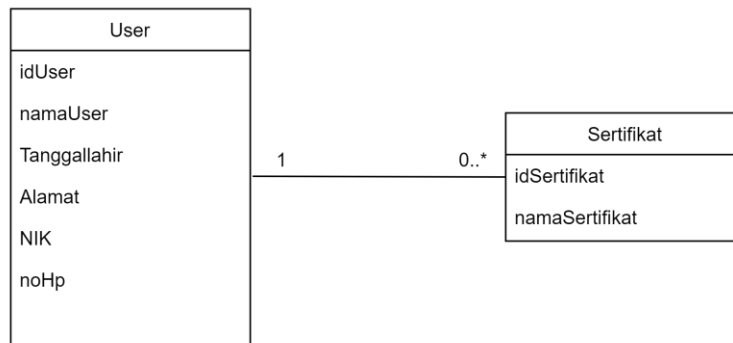
6. Class Dokter - Class Jadwal



Gambar 36 Dokter-Jadwal

Terdapat hubungan sideways coupling pada hubungan antara class jadwal dengan class dokter dimana berdasarkan penjelasan class sideways coupling yaitu sebuah class mengacu langsung ke private properties yang terdapat pada class lain dengan tingkatan yang sejajar. Sehingga jadwal akan menyesuaikan dengan dokter yang terdapat pada lokasi yang telah dipilih sebelumnya oleh user.

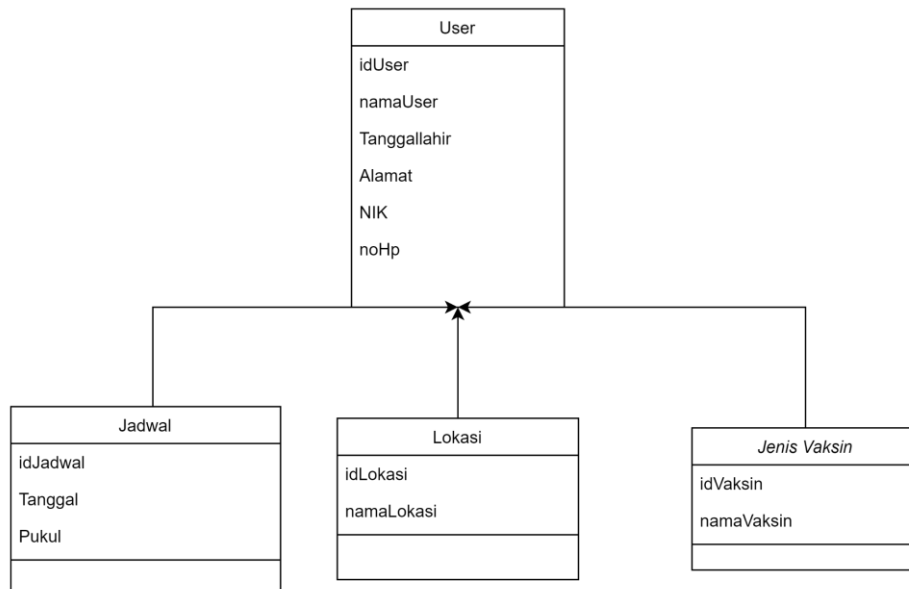
7. Class User - Class Sertifikat



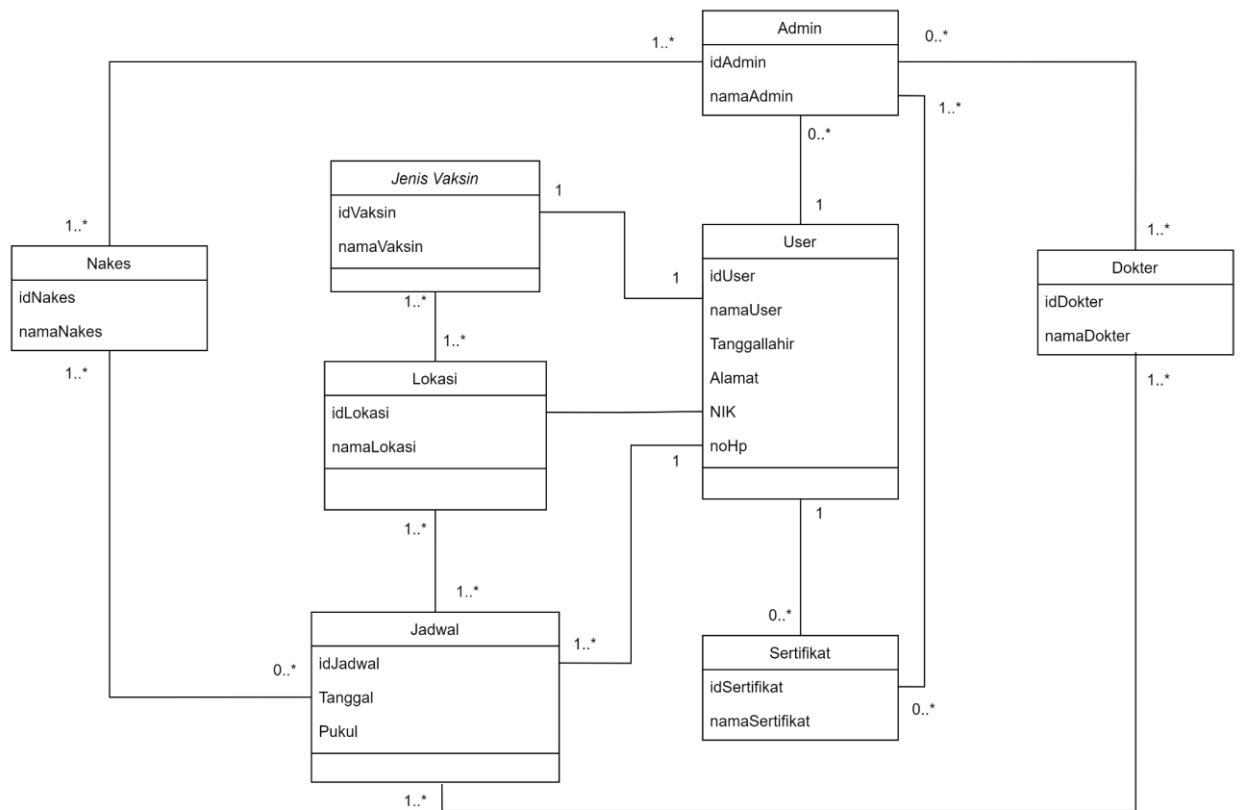
Gambar 37 User-Sertifikat

Terdapat hubungan *outside coupling* antara *class user* dengan *class sertifikat*, hal tersebut dapat ditunjukkan pada *Class User* mengacu langsung ke *Class Sertifikat* yang terdiri dari `idSertifikat` dan `namaSertifikat`. Melalui *outside coupling*, *class sertifikat* dapat mengakses informasi yang terdapat dalam *class user*. Informasi yang diperoleh *user* berupa data-data pribadi yang akan digunakan nanti untuk pencetakan sertifikat.

Jika coupling adalah istilah yang mendeklarasikan hubungan yang terjalin antara satu class dengan class lain. Di lain sisi terdapat cohesion sebagai sebuah istilah yang mendeklarasikan seberapa kuat hubungan atribut-atribut yang terdapat pada sebuah class (Saxena & Kumar, 2015). Pada sistem Vaccine-ly, cohesion yang terbentuk adalah antara class *User*, *Jadwal*, *Lokasi*, dan *Jenis Vaksin*.



Class User, *jadwal*, *lokasi* dan *jenis vaksin* dapat kita lihat bahwa keempat *class* memiliki *cohesion* yang baik karena setiap *class* yang terdapat dalam generalisasi ini terpisah-pisah secara masing-masing, sehingga dapat kita simpulkan bahwa pada setiap atribut yang ada pada *class* tersebut terbagi secara masing-masing dengan baik. Setelah dilakukan revisi, tentunya berdampak pada *class diagram*-nya juga. Berikut adalah *class diagram* yang berada pada sistem Vaccine-ly setelah di revisi..



Gambar 38 Class Diagram setelah Revisi

BAB VI

KESIMPULAN

Vaccine-Ly adalah sebuah aplikasi yang dirancang untuk membantu program pemerintah dalam masa pandemi Covid 19 tersebut. Melalui sistem yang ada pada Vaccine-Ly, diharapkan aplikasi tersebut user dapat menggunakannya untuk keperluan vaksinasi serta untuk memperoleh informasi mengenai Covid 19. Selain itu, aplikasi ini dapat digunakan secara terus menerus dalam membantu program pemerintah dalam program vaksinasi yang digunakan untuk menunjukkan identitas sertifikat vaksin.

Aplikasi Vaccine-Ly memiliki fitur dalam mencari fasilitas pencarian lokasi vaksinasi. Terdapat beberapa jenis pengguna yang dapat mengakses aplikasi Vaccine-Ly, yaitu user yaitu masyarakat sebagai subjek yang mencari informasi lokasi vaksinasi, melakukan vaksinasi, serta mencetak sertifikat dan menggunakannya untuk menunjukkan identitas bahwa telah divaksin. Sebagai User, aktivitas utama yang dilakukan pada aplikasi Vaccine-Ly adalah mencari dan memperoleh informasi vaksinasi. Sedangkan, dari pihak admin, nakes, maupun dokter aktivitas utama yang dilakukan dalam sistem ini adalah memberikan informasi lokasi vaksinasi dan sebagai tenaga yang membantu proses user dalam melakukan vaksinasi.

Secara garis besar, dari sisi user kegiatan yang dilakukan adalah mendaftar akun mengisi data diri dan mencari informasi lokasi vaksinasi, serta kegiatan opsional yang dapat dilakukan adalah mendaftar vaksinasi. Jika user melakukan pendaftaran maka akan dilanjutkan dengan menunggu konfirmasi dari admin kemudian pengisian formulir pendaftaran kemudian melakukan vaksinasi dan berakhir pada mencetak sertifikat. Di sisi lain, bagi pihak selain user adalah sebagai fasilitator dalam membantu user dalam menggunakan aplikasi tersebut.

Aplikasi Vaccine-Ly dirancang dan dianalisis dengan berdasarkan pada konsep Object Oriented Analysis and Design (OOAD). Analisis yang dilakukan dalam perancangan sistem ini yaitu: membuat problem domain, application domain, architectural domain, component design serta perancangan user interface. Pada tahap analisis problem domain, peneliti melakukan penentuan class dan event, structure,

behavior, dan relation. Pada tahap analisis application domain, dilakukan penentuan usage, actor, function dan perancangan interface. Pada tahap analisis architectural domain, kegiatan yang dilakukan adalah menentukan 82 criteria, model architecture, dan process architecture. Terakhir, dalam analisis component design, kegiatan yang dilakukan adalah menentukan model component, function component dan connecting component.

Kami berharap dengan keberhasilan penyelesaian proyek ini, ke depannya kami dapat menyelesaikan permasalahan yang lebih kompleks. Kami juga terbuka dengan kritik maupun saran yang membangun agar dapat meningkatkan wawasan kami di hari mendatang. Akhir kata, kami harap melalui perancangan sistem ini, aplikasi tersebut dapat berguna bagi masyarakat serta sebagai inspirasi dalam membangun sistem program aplikasi terkait vaksinasi.

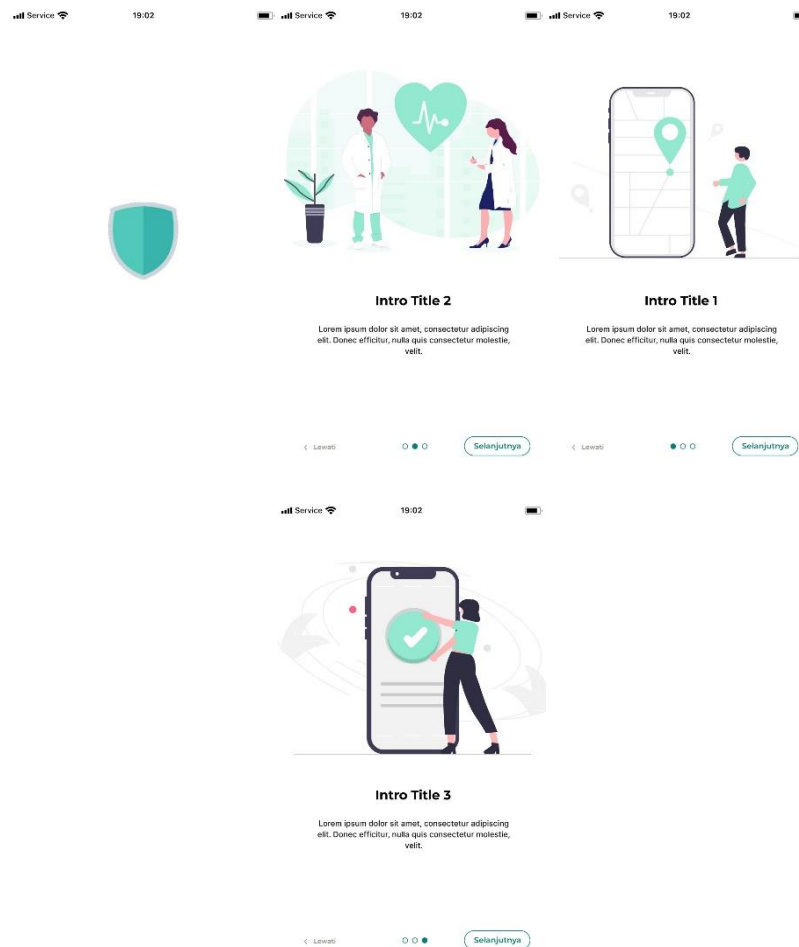
DAFTAR PUSTAKA

- Artina, Nyimas. (2006). Penerapan Analisis Kebutuhan Metode Use Case pada Metode Pengembangan Terstruktur.
- Dennis, A., Wixom, B.H., & Roth. (2014). SYSTEMS ANALYS & DESIGN (6th ed). Wiley.
- Satzinger, J., Jackson, R., & Burd, S. (2014). SYSTEMS ANALYSIS AND DESIGN IN A CHANGING WORLD (7th ed). Cengage Learning.

LAMPIRAN

USER INTERFACE

- Prototype Figma “Vaccine-ly”:
<https://www.figma.com/proto/df9IC5LuUp02F7WrmuPbr6/Vaccine-ly?page-id=0%3A1&node-id=164%3A1908&viewport=241%2C48%2C0.19&scaling=scale-down&starting-point-node-id=162%3A868>
- Introduction dan Splash Screen



Ketika suatu pengguna membuka aplikasi Vaccine-ly, halaman yang akan ditampilkan pertama adalah logo dari aplikasi sebagai tampilan splash screen, selagi aplikasi memuat segala data yang diperlukan agar dapat dijalankan. Jika suatu pengguna baru pertama kali mengunduh dan membuka aplikasi Vaccine-ly, akan terdapat suatu tampilan introduction sebagai awal pengenalan serta penjelasan mengenai aplikasi Vaccine-ly. Tampilan introduction diperlukan agar pengguna baru dapat memahami apa yang aplikasi Vaccine-ly bisa sediakan untuk kebutuhan pengguna dalam memenuhi kebutuhan vaksin maupun mendapatkan berita informasi terkini mengenai covid-19. Terdapat 3 tahapan introduction dengan tombol “Selanjutnya” dan “Lewati”. Pengguna dapat melanjutkan introduction per tahap dengan menggunakan tombol “Selanjutnya. Sebaliknya, pengguna juga dapat melewati tampilan introduction dengan menggunakan tombol “Lewati”.

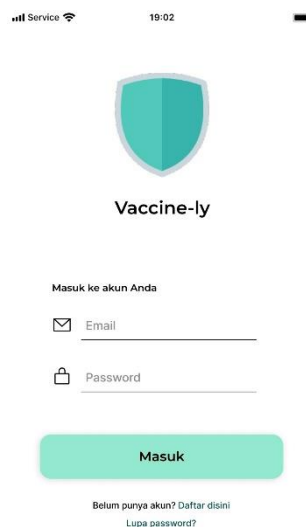
- Menu Pendaftaran Akun

The image displays two sequential screenshots of the Vaccine-ly app's registration process. Both screens have a teal header with the Vaccine-ly logo and a shield icon. The left screenshot, titled 'Daftar akun Anda', shows a registration form with three input fields: 'Email', 'Confirm Email', and 'Password', each preceded by an icon (envelope, envelope, and lock respectively). Below the fields is a teal 'Daftar' button. At the bottom, there is a link: 'Sudah punya akun? Masuk disini'. The right screenshot shows the verification step. It features a teal header with a back arrow and the Vaccine-ly logo. Below the logo, it says 'Kode verifikasi telah dikirimkan ke email Anda. Silakan cek email Anda untuk mendapatkan kode.' followed by a 'Verification Code' input field with a lock icon. Below the field is a link: 'Kirim ulang kode (0:54)'. At the bottom is a teal 'Daftar' button.

Setelah melewati tahap introduction, pengguna akan diminta untuk melakukan formulir pendaftaran akun yang terdapat pada menu pendaftaran akun. Pengguna wajib mengisi kolom email, konfirmasi email, hingga password untuk

melakukan pendaftaran. Jika sudah dirasa benar mengisi seluruh kolom formulir (email dan password), pengguna dapat langsung menggunakan tombol “Daftar” untuk menginput seluruh isi kolom yang telah diisi guna melakukan pendaftaran akun. Nantinya setelah data formulir pendaftaran diinput, pengguna akan diminta kode verifikasi yang telah dikirimkan ke email yang telah didaftarkan untuk memastikan jika email yang digunakan adalah email yang valid. Pengguna juga dapat melakukan “Kirim ulang kode” dengan jarak waktu 1 menit untuk melakukan pengiriman ulang kode. Jika pengguna ternyata sudah mendaftarkan akun Vaccine-ly sebelumnya, pengguna dapat langsung masuk ke akun yang telah dibuat dengan hyperlink “Masuk disini”.

- Menu Masuk Akun



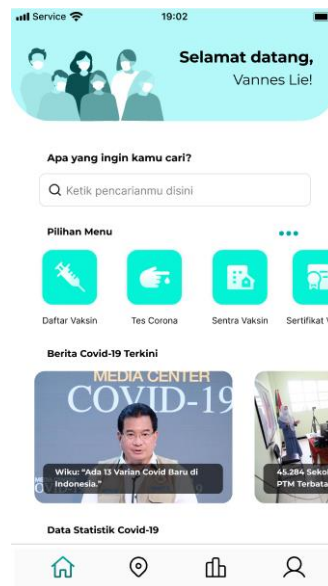
Pengguna dapat langsung memasukkan akun yang telah dipunyai atau dibuat ke menu masuk ini. Terdapat 2 kolom yang perlu diisi: email dan password dan tombol “Masuk” untuk submit seluruh isi kolom data yang telah diisi. Jika pengguna belum memiliki akun pada aplikasi Vaccine-ly, pengguna dapat menggunakan hyperlink “Daftar disini” untuk masuk ke menu pendaftaran dan membuat akun (menu sebelumnya). Disediakan juga mekanisme menu

hyperlink “Lupa password?” jika pengguna lupa password pada akun yang telah dibuat.

Pada halaman menu utama, pengguna akan dibagi menjadi 2 role, antara lain role guest/patient dan role vaksinator/administrator. Menu utama tersebut juga sudah disesuaikan dan memiliki beberapa fungsi yang berbeda satu sama lain sesuai dengan kebutuhannya. Berikut adalah jabaran dan tampilan utama dari masing-masing role dalam mengakses aplikasi Vaccine-ly:

1. Pengguna Guest/Patient

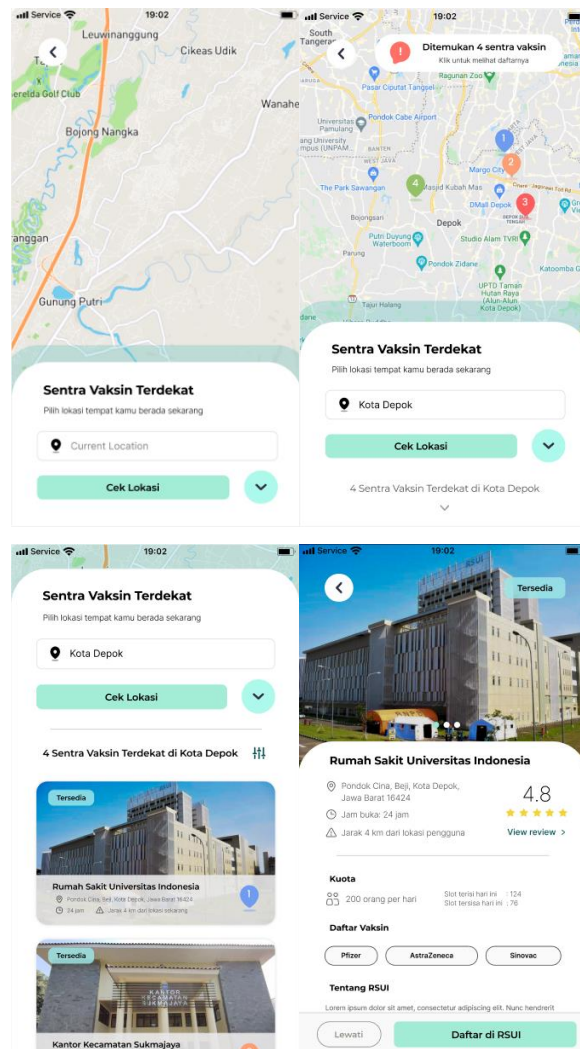
- Menu Beranda



Halaman utama aplikasi Vaccine-ly sebagai guest memiliki format dengan struktur timeline, berisikan seluruh submenu yang terdapat pada aplikasi, seperti tab berita, tab data statistik pasien, hingga tab peta persebaran covid-19. Urutan dalam halaman tersebut diawali dengan fitur search, yang berguna untuk mencari kata kunci dari suatu modul maupun submenu yang terdapat pada aplikasi tersebut. Selanjutnya

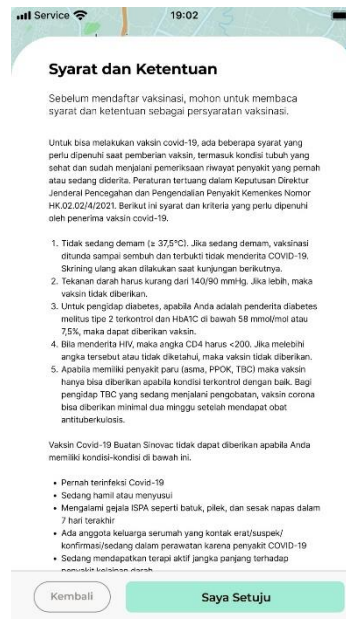
merupakan pilihan menu yang tersedia sebagai guest, antara lain “Daftar Vaksin”, “Tes Corona”, “Sentra Vaksin”, “Sertifikat Vaksin”, dan lain sebagainya. Tampilan dilanjut dengan berbagai tampilan compact dari masing-masing submenu, seperti berita informasi covid-19, data statistik pasien, dan lain sebagainya. Tampilan compact submenu tersebut dapat diatur sesuai keinginan pengguna. Pada bagian bawah, terdapat 4 menu utama yang disediakan, antara lain menu “Beranda”, “Peta”, “Data Statistik Covid-19”, dan “Profile”.

- Menu Peta (Daftar Vaksin)



Saat melakukan mendaftar vaksin di submenu “Daftar Vaksin”, tampilan awal yang ditampilkan adalah pencarian lokasi sentra vaksin terdekat berbentuk tampilan peta dengan sebuah layer yang menampilkan fungsi pencarian lokasi terhadap sentra vaksin terdekat. Ketika menginput suatu nama lokasi pada kolom “Current Location” dan mencarinya menggunakan tombol “Cek Lokasi”, sistem akan mendeteksi seluruh sentra vaksin yang berada pada daerah terdekat pengguna, yang ditampilkan dalam bentuk daftar (bubble chat di atas) dan titik koordinat lokasi dalam urutan angka pada peta. Layer dapat ditarik ke atas atau menggunakan tombol dengan “arah bawah” untuk menampilkan keseluruhan daftar dari seluruh sentra vaksin terdekat yang telah ditemukan urutan angka sesuai pada tampilan di peta. Daftar akan menampilkan total seluruh sentra vaksin yang ditemukan dan dapat diatur menggunakan sorting dan fungsi lainnya. Pada setiap pilihan sentra vaksin, dicantumkan nama lokasi, alamat, jam buka, dan jarak dari lokasi pengguna ke tempat tersebut. Jika dibuka, tampilan deskripsi lokasi akan ditampilkan secara lengkap dengan galeri foto yang tersedia. Deskripsi tersebut menambahkan fitur rating, deskripsi tempat, informasi kuota vaksin, jenis vaksin, dan lain sebagainya.

- Syarat dan Ketentuan



Setelah memilih sentra vaksinasi yang dirasa tepat, tampilan persetujuan syarat dan ketentuan akan muncul sebelum memasuki proses pendaftaran vaksinasi lebih lanjut. Syarat dan ketentuan harus disetujui oleh pengguna terlebih dahulu sebelum menjalani proses vaksinasi. Jika pengguna setuju dengan persyaratan dan ketentuan vaksinasi covid-19, pengguna dapat menggunakan tombol “Saya Setuju” dan berproses ke tahap selanjutnya. Namun sebaliknya, pengguna juga dapat tidak setuju dan kembali ke halaman sebelumnya dengan tombol “Kembali”.

- Proses Pendaftaran Vaksinasi

Pendaftaran Vaksinasi

Isi NIK dan nama lengkap Anda pada kolom di bawah

NIK

Nama Lengkap

Cek Biodata

Halo Vannes!

Mohon isi formulir di bawah ini sesuai identitas Anda

Tempat Lahir Tanggal Lahir

Email

Nomor Handphone

Berikutnya

Cek Ketersediaan Vaksin

Mohon mengisi data vaksin yang ingin dituju di bawah ini untuk proses vaksinasi pasien

Current Location

Pilih Jadwal Vaksin

Pilih Jenis Vaksin

Konfirmasi

Cek Ketersediaan Vaksin

Mohon mengisi data vaksin yang ingin dituju di bawah ini untuk proses vaksinasi pasien

Current Location

Pilih Jenis Vaksin

Pilih Jadwal Vaksin

Konfirmasi

[Kembali ke halaman utama](#)

Selamat, Anda telah terdaftar untuk vaksinasi!

Dengan vaksin AstraZeneca di Sentra Vaksinasi Rumah Sakit Universitas Indonesia pada tanggal 30 September 2021.

Invoice #00000045880

Lihat Rincian

Pada tahap pendaftaran vaksinasi, pengguna akan diminta untuk melengkapi biodatanya terlebih dahulu, mulai dari tampilan awal: NIK dan nama lengkap, tampilan kedua: tempat, tanggal lahir, email, nomor handphone, tampilan ketiga: lokasi sentra vaksin, jadwal vaksin, dan jenis vaksin, tampilan keempat: konfirmasi data pengguna, tampilan kelima: tampilan sukses (berserta invoice vaksin untuk order), jika data yang telah diinput sesuai dan sudah berhasil didaftarkan. Keseluruhan proses tersebut terbagi ke dalam 4 proses berdasarkan tampilannya, antara lain pendaftaran vaksin (NIK dan nama lengkap), biodata lebih lanjut, cek ketersediaan vaksin, dan konfirmasi data. Pemilihan lokasi sentra vaksin pada “Current Location” dapat terisi otomatis jika pengguna memilih sentra vaksinnya terlebih dahulu pada menu peta. Sebaliknya, pengguna juga dapat mengisinya secara manual dengan mengisi biodata terlebih dahulu, dan saat pada tahap cek ketersediaan vaksin, pengguna baru dapat memilih lokasi sentra vaksin. Saat konfirmasi data, setiap kolom data akan diperiksa melalui sistem apakah data yang diinput sudah terdaftar apa belum. Jika belum, logo checklist akan muncul pada masing-masing kolom yang menandakan data tersebut valid dan dapat digunakan.

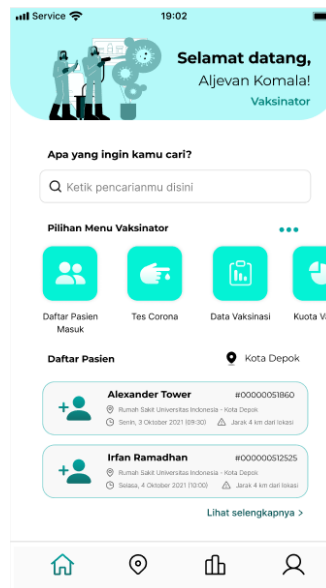
- Submenu Sertifikat Vaksin



Setelah melakukan vaksinasi, pengguna akan diberitahukan melalui aplikasi/email/nomor handphone mengenai sertifikat vaksin jika sertifikat telah dikeluarkan. Pengguna dapat mengakses sertifikat melalui submenu ini yang tersedia pada aplikasi. Sertifikat juga mengacu pada dosis vaksin yang telah diterima oleh pasien. Submenu sertifikat vaksin ini juga menunjukkan judul, tanggal, hingga lokasi dari sertifikat yang telah diterbitkan dan dapat diunduh melalui aplikasi tersebut.

2. Pengguna Vaksinator/Administrator

- Menu Beranda



Berbeda dengan menu beranda sebagai guest/patient, menu beranda sebagai vaksinator/administrator memiliki ilustrasi dan pilihan menu yang berbeda. Ilustrasi menggambarkan visual yang sesuai dengan peran yang pengguna daftarkan di aplikasi Vaccine-ly. Peran juga dijelaskan pada header yang menunjukkan bahwa terdapat role yang dimiliki oleh suatu pengguna di aplikasi ini, tetapi role pengguna guest/patient tidak akan menampilkan informasi role karena perannya yang sebagai guest/patient, bukan petugas. Memiliki urutan dan format yang sama dengan role guest/patient, vaksinator/administrator memiliki fitur search untuk mencari submenu atau modul apa yang tersedia di dalam aplikasi, serta pilihan submenu yang khusus disediakan untuk vaksinator. Pilihan submenu untuk vaksinator berisi menu-menu yang berguna untuk melayani pasien hingga pengecekan ketersediaan vaksin. Pada tampilan di atas, pilihan menu vaksinator terdiri dari “Daftar Pasien Masuk”, “Tes Corona”, “Data Vaksinasi”, dan “Kuota Vaksin”. Lalu, setelah pilihan menu, terdapat tampilan compact dari fitur menu “Daftar Pasien Masuk”, berisikan daftar pasien yang hendak melakukan vaksinasi covid-19. Peran vaksinator disini adalah untuk mengevaluasi

biodata dari setiap pasien sebelum melakukan vaksinasi covid-19. Maka, tampilan yang ditampilkan pada daftar tersebut berisi nama lengkap, invoice, lokasi sentra vaksin, jadwal vaksin, dan jarak lokasi.

- Menu Verifikasi Data Pasien

Service 19:02

1 2 3 4

Verifikasi Data

Alexander Tower #00000051860
Status : Belum vaksin

Dapat melakukan vaksin

Nama Lengkap
Alexander Tower ✓

NIK
32134218412214656 ✓

Tempat/Tanggal Lahir
Kota Depok, 2 Januari 2002 ✓

Email
alexandertower@gmail.com ✓

Nomor Handphone
081234567912 ✓

Lokasi Sentra Vaksin
Rumah Sakit Universitas Indonesia - Kota Depok ✓

Jenis Vaksin
Jenis Vaksinasi

✗ ✓

Ketika vaksinator memilih salah satu pasien dari daftar pasien masuk, akan muncul tampilan verifikasi data dari pasien tersebut. Vaksinator bisa mengevaluasi dan mengecek secara manual masing-masing dari data yang telah diinput pasien, sama seperti guest/patient saat melakukan konfirmasi data mereka untuk vaksinasi. Bedanya ialah tombol yang menentukan vaksinator apakah data pasien tersebut dapat diterima atau tidak (berbeda dengan pengecekan sistem). Jika vaksinator menerima pasien tersebut, vaksinator bisa memilih tombol dengan simbol checklist. Sebaliknya, jika vaksinator menolak pasien tersebut, vaksinator dapat memilih tombol dengan simbol silang, untuk melakukan kegiatan vaksinasi covid-19.