

**IMPLEMENTASI FRAMEWORK LARAVEL PADA PERANCANGAN APLIKASI
SISTEM PENDAFTARAN PROGRAMMING COURSE ROBLOX**

LAPORAN PROYEK 2

Ditujukan sebagai salah satu syarat
Untuk memperoleh nilai pada kegiatan proyek 2
Program Studi D-IV Teknik Informatika



Oleh

Daffa Audya Pramana (1214015)
Jasmine Mutiara Bintang (1214012)

**PROGRAM STUDI DIPLOMA IV TEKNIK INFORMATIKA
UNIVERSITAS LOGISTIK DAN BISNIS INTERNASIONAL**

BANDUNG

2023

SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME

Yang bertanda Tangan di bawah ini:

Nama : Daffa Audya Pramana

NPM : 1214015

Program Studi : D-IV Teknik Informatika

Judul Proyek : Implementasi *Framework Laravel* Pada Perancangan Aplikasi Sistem Pendaftaran *Programming Course Roblox*

Menyatakan Bawa:

1. Proyek Implementasi *Framework Laravel* Pada Perancangan Aplikasi Sistem Pendaftaran *Programming Course Roblox* (PROYEK 2) saya ini adalah asli dan belum pernah diajukan untuk memenuhi kelulusan mata kuliah Proyek 2 pada Program Studi Sekolah Vokasi Universitas Logistik Dan Bisnis Internasional maupun di Perguruan Tinggi lainnya.
2. Proyek Implementasi *Framework Laravel* Pada Perancangan Aplikasi Sistem Pendaftaran *Programming Course Roblox* (PROYEK 2) ini adalah murni gagasan, rumusan, dan penelitian saya sendiri tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam Implementasi *Framework Laravel* Pada Perancangan Aplikasi Sistem Pendaftaran *Programming Course Roblox* (PROYEK 2) ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku diperguruan tinggi lain.

Bandung, 07 Februari 2023

Yang Membuat Pernyataan

Daffa Audya Pramana

NPM. 1214015

SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME

Yang bertanda Tangan di bawah ini:

Nama : Jasmine Mutiara Bintang

NPM : 1214012

Program Studi : D-IV Teknik Informatika

Judul Proyek : Implementasi *Framework Laravel* Pada Perancangan Aplikasi Sistem *Programming Course Roblox*

Menyatakan Bawa:

1. Proyek Implementasi *Framework Laravel* Pada Perancangan Aplikasi Sistem Pendaftaran *Programming Course Roblox* (PROYEK 2) saya ini adalah asli dan belum pernah diajukan untuk memenuhi kelulusan mata kuliah Proyek 2 pada Program Studi Sekolah Vokasi Universitas Logistik Dan Bisnis Internasional maupun di Perguruan Tinggi lainnya.
2. Proyek Implementasi *Framework Laravel* Pada Perancangan Aplikasi Sistem Pendaftaran *Programming Course Roblox* (PROYEK 2) ini adalah murni gagasan, rumusan, dan penelitian saya sendiri tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam Implementasi *Framework Laravel* Pada Perancangan Aplikasi Sistem Pendaftaran *Programming Course Roblox* (PROYEK 2) ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku diperguruan tinggi lain.

Bandung, 07 Februari 2023

Yang Membuat Pernyataan

Jasmine Mutiara Bintang

NPM. 1214012

ABSTRAK

Programming course roblox adalah sebuah media pendidikan yang dijalankan kurang lebih 1 tahun dimana kursus ini masih baru dimulai kecil kecilan yang dilakukan secara *online* setiap pertemuan sebagai sarana pembelajaran. Selama ini pendaftaran peserta kursus yang dilakukan pada *programming course roblox* masih dilakukan secara manual yaitu peserta harus mengisi formulir pendaftaran kemudian mengirimkan nya kepada admin secara pribadi dan admin merekap laporan data peserta melalui buku, serta biaya pembayaran kursus dilakukan masih dilakukan secara langsung. Dengan melihat permasalahan tersebut terdapat beberapa kendala diantaranya banyaknya peserta kursus dan berkas berkas data peserta membuat admin kesulitan dalam mengelolanya. Formulir pendaftaran kursus yang dikirim ke admin tentunya akan tersimpan secara manual yakni dibuku, jika buku laporan data calon peserta hilang tentunya data data peserta kursus yang tersimpan didalamnya akan hilang juga. Maka dari itu dirancangnya sebuah aplikasi sistem pendaftaran *programming course roblox* sebagai sebuah solusi dari permasalahan yang terjadi selama berjalannya proses kegiatan pendaftaran kursus tersebut. Tujuan dari perancangan aplikasi sistem pendaftaran *programming course roblox* ini yakni dapat mempermudah dan mempersingkat peserta dalam melakukan registrasi atau pendaftaran kursus melalui *website* secara *online* tanpa harus mengisi dan mengirimkan formulir pendaftaran melalui admin secara langsung. Landasan teori yang mendukung perancangan aplikasi sistem pendaftaran *programming course roblox* ini yaitu bahasa pemrograman Php, *framework* laravel, perancangan, system, PhpMyAdmin, Mysql, XAMPP, *Visual Studio Code*, UML (*Unified Language Modelling*, konsep MVC (*Model, View, Controllers*) dan *code coverage*. Metode yang digunakan dalam perancangan ini adalah dengan metode pengembangan *waterfall*.

Kata kunci: Kursus, Pendaftaran, Bahasa Pemrograman Php, *Framework* Laravel

ABSTRACT

The Roblox programming course is an educational medium that has been running for about 1 year where this course has just started small, which is carried out online at every meeting as a learning tool. So far, the registration of course participants on the Roblox programming course is still done manually, that is, participants have to fill out a registration form and then send it to the admin personally and the admin recaps the participant data reports through a book, and course payment fees are still done manually. By looking at these problems there are several obstacles including the large number of course participants and participant data files making it difficult for the admin to manage them. The course registration form sent to the admin will of course be stored manually, namely in a book, if the data report book for prospective participants is lost, of course the course participant data stored in it will also be lost. Therefore, an application for the Roblox course registration system was designed as a solution to the problems that occurred during the course registration process. The purpose of making the Roblox programming course registration system application is to simplify and shorten participants in registering or registering for courses through the website online without having to fill out and send registration forms directly through the admin. The theoretical basis that supports the preparation of the roblox programming course registration system application is the Php programming language, laravel framework, design, system, PhpMyAdmin, Mysql, XAMPP, Visual Studio Code, UML (Unified Language Modeling), the concept of MVC (Model, View, Controllers) and Code Coverage. The method used in this design is the waterfall.

Keywords: Course, Registration, Php Programming Language, Laravel Framework

KATA PENGANTAR

Puji syukur terpanjatkan kepada Allah SWT. yang telah melimpahkan rahmat dan karunia-Nya, sehingga laporan proyek 2 ini dengan judul "Implementasi *Framework* Laravel Pada Perancangan Aplikasi Sistem Pendaftaran *Programming Course Roblox*" dapat diselesaikan. Penulisan laporan ini dilakukan dalam rangka memenuhi salah satu syarat untuk mendapat nilai pada mata kuliah proyek 2 Program Studi D-IV Teknik Informatika Universitas Logistik dan Bisnis Internasional.

Dalam penulisan laporan ini tentunya banyak kendala yang dihadapi. Oleh karena itu, saya menyadari bahwa penulisan laporan ini belum sempurna dan tanpa bantuan serta bimbingan dari berbagai pihak, mungkin penulisan ini belum bisa diselesaikan. Saya mengharapkan kritik dan saran dari pembaca yang sifatnya berguna kedepannya. Dengan terselesaiannya penulisan laporan ini, saya mengucapkan terima kasih kepada:

Allah SWT. yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulisan laporan ini dapat terselesaikan.

Woro Isti Rahayu, S.T., M.T., SFPC selaku Dosen Pembimbing Universitas Logistik dan Bisnis Internasional yang telah menyediakan waktu, tenaga, dan pikiran untuk membimbing dan mengarahkan dalam penulisan laporan ini.

Nisa Hanum Harani, S.Kom., M.T., CDSP, SFPC selaku Koordinator Proyek 2 dan yang telah menyediakan waktu, tenaga, dan pikiran untuk mengatur dan mengarahkan dalam berjalannya proyek 2 ini.

Semoga tulisan ini dapat bermanfaat bagi pengembangan ilmu, para pembaca dan diri sendiri. Akhir kata, diharapkan agar laporan proyek 2 ini dapat bermanfaat bagi diri sendiri dan orang lain.

Bandung, 07 Februari 2022

Penulis

DAFTAR ISI

SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME	i
SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME	ii
ABSTRAK.....	iii
<i>ABSTRACT</i>	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xvi
DAFTAR SIMBOL	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	2
1.2 Identifikasi Masalah.....	2
1.3 Tujuan Pembuatan Penelitian.....	2
1.4 Manfaat Penelitian	3
1.5 Ruang Lingkup.....	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI.....	5
2.1 <i>Related Research</i>	5
2.2 Tinjauan Pustaka	7
2.2.1 <i>Programming Course Roblox</i>	7
2.2.2 Implementasi.....	7
2.2.3 <i>Framework Laravel</i>	8
2.2.4 Perancangan	8
2.2.5 Sistem.....	8
2.2.6 Bahasa Pemrograman Php	9
2.2.7 PhpMyAdmin.....	9
2.2.8 MySql.....	9
2.2.9 XAMPP.....	10
2.2.10 <i>Visual Studio Code</i>	10
2.2.11 <i>UML (Unified Modelling Language)</i>	10

2.2.12 Konsep MVC (<i>Model, View, Controllers</i>).....	10
2.2.13 <i>Code Coverage</i>	11
2.2.14 Metode <i>Waterfall</i>	11
BAB III ANALISIS DAN PERANCANGAN	13
3.1 Analisis.....	13
3.1.1 Analisis Sistem Berjalan (<i>Current System</i>).....	13
3.1.1.1 Analisis Sistem Yang Sedang Berjalan Pada Pendaftaran Kursus .	14
3.1.2 Analisis Sistem Yang Akan Dibangun	114
3.1.2.1 Analisis sistem yang akan dibangun pada prosedur <i>register</i>	14
3.1.2.2 Analisis sistem yang akan dibangun pada prosedur <i>login</i>	15
3.1.2.3 Analisis sistem yang akan dibangun pada prosedur pendaftaran....	16
3.1.2.4 Analisis sistem yang akan dibangun pada prosedur pembayaran	17
3.1.2.5 Analisis sistem yang akan dibangun pada prosedur pengumuman..	18
3.1.2.6 Analisis sistem yang akan dibangun pada prosedur kelola data pengguna	19
3.1.2.7 Analisis sistem yang akan dibangun pada prosedur kelola data sekolah	20
3.1.2.8 Analisis sistem yang akan dibangun pada prosedur kelola data program kursus.....	21
3.1.2.9 Analisis sistem yang akan dibangun pada prosedur kelola data pendaftaran.....	22
3.1.2.10 Analisis sistem yang akan dibangun pada prosedur kelola data pembayaran	23
3.1.2.11 Analisis sistem yang akan dibangun pada prosedur kelola data pengumuman.....	24
3.1.2.12 Analisis sistem yang akan dibangun pada prosedur kelola data profil.....	25
3.1.2.13 Analisis sistem yang akan dibangun pada prosedur <i>logout</i>	26
3.1.3 Analisis Dokumen Yang Digunakan.....	26
3.1.4 Analisis Kebutuhan Fungsional	28
3.1.5 Analisis Dokumen Non Fungsional	29
3.1.5.1 Analisis Kebutuhan Perangkat Lunak.....	29
3.1.5.2 Analisis Kebutuhan Perangkat Keras.....	30
3.1.6 Analisis Pengguna.....	30
3.2 Perancangan	32

3.2.1 UML (<i>Unified Modelling Language</i>).....	32
3.2.1.1 <i>Use Case Diagram</i>	32
3.2.1.1.1 Definisi Aktor	33
3.2.1.1.2 Definisi <i>Use Case</i>	33
3.2.1.1.3 Skenario <i>Use Case</i>	34
3.2.1.2 <i>Activity Diagram</i>	39
3.2.1.2.1 <i>Activity Diagram Register</i>	39
3.2.1.2.2 <i>Activity Diagram Login</i>	40
3.2.1.2.3 <i>Activity Diagram</i> Pendaftaran.....	41
3.2.1.2.4 <i>Activity Diagram</i> Pembayaran	42
3.2.1.2.5 <i>Activity Diagram</i> Pengumuman.....	43
3.2.1.2.6 <i>Activity Diagram</i> Kelola Data Pengguna.....	44
3.2.1.2.7 <i>Activity Diagram</i> Kelola Data Sekolah.....	45
3.2.1.2.8 <i>Activity Diagram</i> Kelola Data Program Kursus	46
3.2.1.2.9 <i>Activity Diagram</i> Kelola Data Pendaftaran	47
3.2.1.2.10 <i>Activity Diagram</i> Kelola Data Pembayaran.....	48
3.2.1.2.11 <i>Activity Diagram</i> Kelola Data Pengumuman.....	49
3.2.1.2.12 <i>Activity Diagram</i> Kelola Data Profil	50
3.2.1.2.13 <i>Activity Diagram Logout</i>	51
3.2.1.3 <i>Sequence Diagram</i>	51
3.2.1.3.1 <i>Sequence Diagram Register</i>	52
3.2.1.3.2 <i>Sequence Diagram Login</i>	53
3.2.1.3.3 <i>Sequence Diagram</i> Pendaftaran.....	54
3.2.1.3.4 <i>Sequence Diagram</i> Pembayaran	55
3.2.1.3.5 <i>Sequence Diagram</i> Pengumuman.....	56
3.2.1.3.6 <i>Sequence Diagram</i> Kelola Data Pengguna.....	57
3.2.1.3.7 <i>Sequence Diagram</i> Kelola Data Sekolah.....	58
3.2.1.3.8 <i>Sequence Diagram</i> Kelola Data Program Kursus.....	59
3.2.1.3.9 <i>Sequence Diagram</i> Kelola Data Pendaftaran.....	61
3.2.1.3.10 <i>Sequence Diagram</i> Kelola Data Pembayaran	62
3.2.1.3.11 <i>Sequence Diagram</i> Kelola Data Pengumuman.....	63
3.2.1.3.12 <i>Sequence Diagram</i> Kelola Data Profil.....	63
3.2.1.3.13 <i>Sequence Diagram Logout</i>	64

3.2.1.4 <i>Class Diagram</i>	64
3.3 Perancangan <i>Database</i>	66
3.3.1 <i>Conceptual Data Model (CDM)</i>	66
3.3.2 <i>Physical Data Model (PDM)</i>	67
3.3.3 <i>Entity Relationship Diagram (ERD)</i>	68
3.4 Perancangan Struktur Menu	69
3.4.1 Struktur Menu Peserta.....	69
3.4.2 Struktur Menu Admin	70
3.5 Antarmuka.....	70
3.5.1 Rancangan Antarmuka Peserta	71
3.5.1.1 Rancangan Antarmuka Halaman <i>Register</i>	71
3.5.1.2 Rancangan Antarmuka Halaman <i>Login</i>	71
3.5.1.3 Rancangan Antarmuka Halaman Beranda.....	72
3.5.1.4 Rancangan Antarmuka Halaman Kelola Data Profil.....	72
3.5.1.5 Rancangan Antarmuka Halaman Pendaftaran	73
3.5.1.6 Rancangan Antarmuka Halaman Pembayaran.	74
3.5.1.7 Rancangan Antarmuka Halaman Detail Pendaftaran.	75
3.5.1.8 Rancangan Antarmuka Halaman Pengumuman.	76
3.5.1.9 Rancangan Antarmuka Halaman <i>Logout</i>	76
3.5.2 Rancangan Antarmuka Admin.....	77
3.5.2.1 Rancangan Antarmuka Halaman <i>Login</i>	77
3.5.2.2 Rancangan Antarmuka Halaman Beranda.....	77
3.5.2.3 Rancangan Antarmuka Halaman Kelola Data Profil.....	78
3.5.2.4 Rancangan Antarmuka Halaman Kelola Data Pengguna.	79
3.5.2.5 Rancangan Antarmuka Halaman Kelola Data Sekolah.....	79
3.5.2.6 Rancangan Antarmuka Halaman Kelola Data Program Kursus....	80
3.5.2.7 Rancangan Antarmuka Halaman Kelola Data Pendaftaran.....	80
3.5.2.8 Rancangan Antarmuka Halaman Kelola Data Pembayaran.	81
3.5.2.9 Rancangan Antarmuka Halaman Kelola Data Pengumuman.....	81
3.5.2.10 Rancangan Antarmuka Halaman Detail Pengguna.	82
3.5.2.10 Rancangan Antarmuka Halaman Detail Pendaftaran.	83
3.5.2.10 Rancangan Antarmuka Halaman Detail Pengumuman.	84
3.5.2.11 Rancangan Antarmuka Halaman <i>Logout</i>	84
3.6 Logika Fungsi/ Algoritma Fungsi	85

BAB IV IMPLEMENTASI DAN PENGUJIAN.....	87
4.1 Implementasi dan Pembahasan Hasil Implementasi.....	87
4.1.1 Implementasi Aplikasi Sistem Pendaftaran <i>Programming Course Roblox</i>	87
4.1.2 Implementasi <i>Source Code</i>	100
4.1.2.1 <i>Model</i>	131
4.1.2.2 <i>View</i>	131
4.1.2.3 <i>Controllers</i>	131
4.2 Pengujian dan Hasil Pengujian.....	131
4.2.1 Pengujian.....	131
4.2.1.1 Rencana Pengujian <i>Role Peserta</i>	131
4.2.1.2 Rencana Pengujian <i>Role Admin</i>	131
4.2.1.3 Pengujian Aplikasi	131
4.2.1.4 Pengujian <i>Code Coverage</i>	144
4.2.2 Hasil Pengujian <i>Role Peserta</i> dan Admin	144
BAB V KESIMPULAN DAN SARAN	150
5.1 Kesimpulan	150
5.2 Saran.....	150
DAFTAR PUSTAKA	151
LAMPIRAN.....	151

DAFTAR GAMBAR

Gambar 3.1 Alur Metode <i>Waterfall</i>	11
Gambar 3.1 <i>Flowmap</i> Analisis Sistem Yang Sedang Berjalan Pada Prosedur Pendaftaran Kursus	14
Gambar 3.2 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur <i>Register</i>	15
Gambar 3.3 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur <i>Login..</i>	16
Gambar 3.4 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Pendaftaran.....	17
Gambar 3.5 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Pembayaran	18
Gambar 3.6 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Pengumuman.....	19
Gambar 3.7 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Pengguna	20
Gambar 3.8 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Sekolah	21
Gambar 3.9 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Program Kursus	22
Gambar 3.10 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Pendaftaran	23
Gambar 3.11 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Pembayaran	24
Gambar 3.12 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Pengumuman	25
Gambar 3.13 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Profil	25
Gambar 3.14 <i>Flowmap</i> Analisis Sistem Yang Akan Dibangun Pada Prosedur <i>Logout</i>	26
Gambar 3.15 <i>Use Case Diagram</i>	32

Gambar 3.16 <i>Activity Diagram Register</i>	39
Gambar 3.17 <i>Activity Diagram Login</i>	40
Gambar 3.18 <i>Activity Diagram Pendaftaran</i>	41
Gambar 3.19 <i>Activity Diagram Pembayaran</i>	42
Gambar 3.20 <i>Activity Diagram Pengumuman</i>	43
Gambar 3.21 <i>Activity Diagram Kelola Data Pengguna</i>	44
Gambar 3.22 <i>Activity Diagram Kelola Data Sekolah</i>	45
Gambar 3.23 <i>Activity Diagram Kelola Data Program Kursus</i>	46
Gambar 3.24 <i>Activity Diagram Kelola Data Pendaftaran</i>	47
Gambar 3.25 <i>Activity Diagram Kelola Data Pembayaran</i>	48
Gambar 3.26 <i>Activity Diagram Kelola Data Pengumuman</i>	49
Gambar 3.27 <i>Activity Diagram Kelola Data Profil</i>	50
Gambar 3.28 <i>Activity Diagram Logout</i>	51
Gambar 3.29 <i>Sequence Diagram Register</i>	52
Gambar 3.30 <i>Sequence Diagram Login</i>	53
Gambar 3.31 <i>Sequence Diagram Pendaftaran</i>	54
Gambar 3.32 <i>Sequence Diagram Pembayaran</i>	55
Gambar 3.33 <i>Sequence Diagram Pengumuman</i>	56
Gambar 3.34 <i>Sequence Diagram Kelola Data Pengguna</i>	57
Gambar 3.35 <i>Sequence Diagram Kelola Data Sekolah</i>	58
Gambar 3.36 <i>Sequence Diagram Kelola Data Program Kursus</i>	59
Gambar 3.37 <i>Sequence Diagram Kelola Data Pendaftaran</i>	61
Gambar 3.38 <i>Sequence Diagram Kelola Data Pembayaran</i>	62
Gambar 3.39 <i>Sequence Diagram Kelola Data Pengumuman</i>	63
Gambar 3.40 <i>Sequence Diagram Kelola Data Profil</i>	63
Gambar 3.41 <i>Sequence Diagram Logout</i>	64
Gambar 3.42 <i>Class Diagram</i>	65
Gambar 3.43 <i>Conceptual Data Model (CDM)</i>	66
Gambar 3.44 <i>Physical Data Model (PDM)</i>	67
Gambar 3.45 <i>Entity Relationship Diagram (ERD)</i>	68
Gambar 3.46 Struktur Menu Peserta	69
Gambar 3.47 Struktur Menu Admin	70

Gambar 3.48 Antarmuka Halaman <i>Register</i>	71
Gambar 3.49 Antarmuka <i>Login</i>	71
Gambar 3.50 Antarmuka Halaman Beranda	72
Gambar 3.51 Antarmuka Halaman Kelola Data Profil	72
Gambar 3.52 Antarmuka Halaman Pendaftaran	73
Gambar 3.53 Antarmuka Halaman Pembayaran.....	74
Gambar 3.54 Antarmuka Halaman Detail Pendaftaran.....	75
Gambar 3.55 Antarmuka Halaman Pengumuman.....	76
Gambar 3.56 Antarmuka Halaman <i>Logout</i>	76
Gambar 3.57 Antarmuka Halaman <i>Login</i>	77
Gambar 3.58 Antarmuka Halaman Beranda	77
Gambar 3.59 Antarmuka Halaman Kelola Data Profil	78
Gambar 3.60 Antarmuka Halaman Kelola Data Pengguna.....	79
Gambar 3.61 Antarmuka Halaman Kelola Data Sekolah	79
Gambar 3.62 Antarmuka Halaman Kelola Data Program Kursus	80
Gambar 3.63 Antarmuka Halaman Kelola Data Pendaftaran	80
Gambar 3.64 Antarmuka Halaman Kelola Data Pembayaran.....	81
Gambar 3.65 Antarmuka Halaman Kelola Data Pengumuman	81
Gambar 3.66 Antarmuka Halaman Detail Pengguna	82
Gambar 3.67 Antarmuka Halaman Detail Pendaftaran.....	83
Gambar 3.68 Antarmuka Halaman Detail Pengumuman.....	84
Gambar 3.69 Antarmuka Halaman <i>Logout</i>	84
Gambar 3.70 <i>Flowchart</i> Aplikasi.....	85
Gambar 4.1 Menu Daftar (<i>Register</i>)	87
Gambar 4.2 Menu <i>Login</i>	88
Gambar 4.3 Menu Beranda (<i>Dashboard Admin</i>).....	88
Gambar 4.4 Menu Data <i>Master</i> (Pengguna)	89
Gambar 4.5Menu Detail Pengguna	89
Gambar 4.6 Menu Data <i>Master</i> (Sekolah)	90
Gambar 4.7 Menu Data <i>Master</i> (Program Kursus)	91
Gambar 4.8 Menu Data <i>History</i> (Pendaftaran)	91
Gambar 4.9 Menu Data <i>History</i> (Pembayaran).....	92

Gambar 4.10 Menu Pengumuman.....	93
Gambar 4.11 Menu Profil Admin.....	93
Gambar 4.12 Menu <i>Logout</i>	94
Gambar 4.13 (<i>Dashboard</i> Peserta).....	94
Gambar 4.14 Menu Beranda (<i>Dashboard</i> Peserta)	95
Gambar 4.15 Menu Beranda (<i>Dashboard</i> Peserta)	95
Gambar 4.16 Menu Pendaftaran.....	96
Gambar 4.17 Menu Pendaftaran.....	96
Gambar 4.18 Halaman Pembayaran Peserta	97
Gambar 4.19 Detail Pendaftaran Peserta.....	97
Gambar 4.20 Halaman Kartu Pendaftaran Peserta	98
Gambar 4.21 Halaman Menu Lihat Pengumuman	98
Gambar 4.22 Halaman Pengumuman.....	99
Gambar 4.23 Menu Profil Peserta	99
Gambar 4.24 <i>Source Code User.php</i>	101
Gambar 4.25 <i>Source Code</i> Pendaftaran.php	102
Gambar 4.26 <i>Source Code</i> Pembayaran.php.....	103
Gambar 4.27 <i>Source Code</i> Pengumuman.php	105
Gambar 4.28 <i>Source Code</i> Pengguna.php.....	105
Gambar 4.29 <i>Source Code</i> Sekolah.php.....	107
Gambar 4.30 <i>Source Code</i> Prodi.php.....	108
Gambar 4.31 <i>Source Code</i> Register.php	109
Gambar 4.32 <i>Source Code</i> Login.php	110
Gambar 4.33 <i>Source Code</i> Data-pendaftaran.php.....	121
Gambar 4.34 <i>Source Code</i> Data-pembayaran.php	123
Gambar 4.35 <i>Source Code</i> Data-pengumuman.php	125
Gambar 4.36 <i>Source Code</i> Data-programkursus.php.....	126
Gambar 4.37 <i>Source Code</i> Data-user.php.....	127
Gambar 4.38 <i>Source Code</i> Profil.php	128
Gambar 4.39 <i>Source Code</i> Dashboard.php	129
Gambar 4.40 <i>Source Code</i> Controllers	130
Gambar 4.41 Pengujian <i>Register</i>	134

Gambar 4.42 Pengujian <i>Login</i> Gagal Peserta.....	134
Gambar 4.43 Pengujian <i>Login</i> Berhasil Peserta.....	135
Gambar 4.44 Pengujian Pendaftaran Peserta	135
Gambar 4.45 Pengujian Pembayaran Peserta.....	136
Gambar 4.46 Pengujian Pengumuman Peserta	136
Gambar 4.47 Pengujian Profil Peserta	137
Gambar 4.48 Pengujian <i>Login</i> Gagal Admin	137
Gambar 4.49 Pengujian <i>Login</i> Berhasil Admin	138
Gambar 4.50 Pengujian <i>Dashboard</i> Admin.....	138
Gambar 4. 51 Pengujian Menu Penguna Admin.....	139
Gambar 4.52 Pengujian Menu Sekolah Admin.....	139
Gambar 4.53 Pengujian Menu Kursus Admin	140
Gambar 4.54 Pengujian Menu Pendaftaran Admin	140
Gambar 4.55 Pengujian Menu Pembayaran Admin.....	141
Gambar 4.56 Pengujian Menu Pengumuman Admin.....	141
Gambar 4.57 Pengujian Menu Profil Admin	142
Gambar 4.58 Pengujian <i>Code Coverage</i>	142
Lampiran 1. Poster Infografis.....	154
Lampiran 2. Submit Jurnal Teknik Informatika.....	142

DAFTAR TABEL

Tabel 3.1 Dokumen Pendaftaran Kursus.....	27
Tabel 3.2 Dokumen Pengumuman Kursus.....	27
Tabel 3.3 Dokumen Pembayaran Kursus	27
Tabel 3.4 Spesifikasi Perangkat Lunak	29
Tabel 3.5 Spesifikasi Perangkat Keras	30
Tabel 3.6 Analisis Kebutuhan Pengguna	30
Tabel 3.7 Definisi Aktor.....	33
Tabel 3.8 Definisi <i>Use Case</i>	33
Tabel 3.9 Skenario <i>Use Case Register</i>	34
Tabel 3.10 Skenario <i>Use Case Login</i>	35
Tabel 3.11 Skenario <i>Use Case</i> Pendaftaran	35
Tabel 3.12 Skenario <i>Use Case</i> Pembayaran.....	35
Tabel 3.13 Skenario <i>Use Case</i> Pengumuman	36
Tabel 3.14 Skenario <i>Use Case</i> Kelola Data Pengguna.....	36
Tabel 3.15 Skenario <i>Use Case</i> Kelola Data Sekolah	36
Tabel 3.16 Skenario <i>Use Case</i> Kelola Data Program Kursus	37
Tabel 3.17 Skenario <i>Use Case</i> Kelola Data Pendaftaran	37
Tabel 3.18 Skenario <i>Use Case</i> Kelola Data Pembayaran	37
Tabel 3.19 Skenario <i>Use Case</i> Kelola Data Pengumuman	37
Tabel 3.20 Skenario <i>Use Case</i> Kelola Data Profil	38
Tabel 3.21 Skenario <i>Use Case Logout</i>	38
Tabel 4.1 Rencana Pengujian <i>Role</i> Peserta	131
Tabel 4.2 Rencana Pengujian <i>Role</i> Admin.....	132
Tabel 4.3 Hasil Pengujian <i>Role</i> Peserta	144
Tabel 4.4 Hasil Pengujian <i>Role</i> Admin.....	146

DAFTAR SIMBOL

Simbol Flowchart dan Flowmap

Simbol	Nama	Keterangan
	Terminal <i>points</i>	Menyatakan <i>symbol</i> untuk memulai dan mengakhiri program
	Proses	Menyatakan proses, pekerjaan atau tindakan yang dilakukan pada data sehingga data berubah, disimpan, atau didistribusikan
	<i>Input/ Output</i>	Mempresentasikan <i>input</i> dan <i>output</i> , untuk memasukkan data dan menunjukkan keluaran yang dihasilkan
	Keputusan	Kondisi ini menyatakan sebuah pilihan yang dapat menghasilkan sebuah keputusan dalam program
	Arus	Menunjukan arah proses

Lanjutan

	Anak Panah	Mempresentasikan sebuah alur kerja program yang berjalan
---	------------	--

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada era digital saat ini, perkembangan zaman sangat memengaruhi dalam bidang ilmu pengetahuan dan teknologi. Teknologi yang begitu pesat membawa perubahan dalam sendi kehidupan salah satunya adalah sebagai sarana pembelajaran dan media hiburan. Hal itu membuat berbagai kalangan membuat sebuah media pendidikan kursus dengan memanfaatkan teknologi informasi khususnya media pendidikan *programming course roblox*. *Programming course roblox* adalah sebuah media pendidikan berbasis IT yakni memberikan pembelajaran kursus dalam bidang pemrograman *game roblox*, kursus ini juga yang dijalankan berupa kursus yang masih baru dimulai kecil kecilan setiap pertemuan sebagai sarana pembelajaran. Di era globalisasi yang semuanya sudah serba cepat dan mudah, *website* sebagai sarana komunitas dan *marketing* adalah cara paling jitu yang digunakan untuk meningkatkan *value*. Dengan adanya sarana digital, maka semakin mudah dan cepatlah seseorang untuk melakukan pendaftaran apabila ingin mengikuti suatu kursus.

Maraknya perkembangan media pendidikan kursus di indonesia yakni tercatat data yang diperoleh dari DPKK terdapat 13.446 lembaga pendidikan kursus yang tersebar di Indonesia. Salah satunya adalah media pendidikan *programming course roblox*. Kursus *programming course roblox* yang berada di Bandung ini sistem pendaftaran peserta kursusnya dilakukan secara manual yaitu peserta harus mengisi formulir pendaftaran kemudian mengirimkannya kepada admin secara pribadi dan admin merekap data pendaftaran peserta melalui buku, dan biaya pembayaran kursus dilakukan masih secara langsung. Selain itu kursus tersebut belum dikenal secara luas dikalangan masyarakat. Dengan melihat permasalahan tersebut terdapat beberapa kendala diantaranya banyaknya peserta kursus dan berkas berkas data peserta membuat admin kesulitan dalam memanajemen data data peserta kursus tersebut. Formulir pendaftaran kursus yang dikirim ke admin tentunya akan tersimpan secara manual yakni dibuku, jika buku laporan data calon peserta hilang tentunya data data peserta kursus yang tersimpan didalamnya akan hilang juga.

Maka dari itu dirancangnya sebuah aplikasi sistem pendaftaran *programming course* roblox sebagai sebuah solusi dari permasalahan yang terjadi selama berjalannya proses kegiatan pendaftaran dikursus tersebut. Perancangan aplikasi sistem pendaftaran *programming course* roblox berbasiskan *web* dan akan menggunakan sebuah *framework*, *framework* tersebut adalah *framework* laravel. Tujuan dari perancangan aplikasi sistem pendaftaran *programming course* roblox ini yakni dapat mempermudah dan mempersingkat peserta dalam melakukan kegiatan pendaftaran kursus secara *online* tanpa harus mengisi dan mengirimkan formulir pendaftaran melalui admin. Selain itu, adanya sistem pendaftaran *programming course* roblox ini juga mempermudah admin dalam membuat dan mengelola data laporan peserta kursus *online*. Metode yang digunakan dalam perancangan ini adalah dengan metode pengembangan perangkat lunak *waterfall*. Perancangan aplikasi menggunakan metode *waterfall*, karena metode ini lebih spesifik terhadap pengembangan perangkat lunak.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas ada beberapa masalah ataupun hambatan dengan dibuatnya perancangan aplikasi sistem pendaftaran *programming course* roblox ini:

1. Bagaimana mengelola data berkas pendaftaran kursus tersebut?
2. Bagaimana membuat *system* pendaftaran secara *online* pada aplikasi sistem pendaftaran *programming course* roblox tersebut?
3. Bagaimana memberikan informasi kepada masyarakat umum mengenai kursus *programming course* roblox?

1.3 Tujuan Pembuatan Penelitian

Adapun tujuan pembuatan aplikasi ini adalah:

1. Memudahkan admin dalam mengelola berkas data peserta kursus *programming course* roblox dengan menerapkan *database* Mysql.
2. Membuat aplikasi *system* pendaftaran *programming course* roblox menggunakan *framework* laravel.

3. Agar memudahkan dan mempersingkat waktu dalam proses pendaftaran peserta kursus melalui *website* secara *online* supaya peserta tidak perlu mengirim surat formulir kepada admin secara langsung.

1.4 Manfaat Penelitian

Bagi Admin:

1. Dapat membangun sebuah aplikasi *system* pendaftaran *programming course* roblox yang dapat digunakan oleh peserta kursus untuk melihat informasi, maupun masyarakat yang ingin mendaftar kursus.
2. Menambah pengetahuan dan wawasan tentang hal hal baru mengenai pembuatan aplikasi tersebut.

Bagi Peserta Kursus atau Calon Peserta Kursus:

1. Dapat memanfaatkan teknologi sebagai salah satu media memperoleh penyampaian informasi mengenai *programming course* roblox tersebut secara *online*.
2. Lebih mudah dan praktis untuk melakukan kegiatan mendaftar kursus dan melihat informasi mengenai *programming course* roblox tersebut, sehingga tidak perlu menempuh jarak jauh untuk mendaftar kursus secara langsung.

1.5 Ruang Lingkup

Untuk menghindari pembahasan diluar permasalahan, maka berikut ini adalah ruang lingkupnya:

1. Aplikasi *system* pendaftaran *programming course* roblox dikhkususkan untuk kegiatan pendaftaran bagi masyarakat, peserta maupun admin.
2. Aplikasi *system* pendaftaran *programming course* roblox hanya menggunakan Bahasa pemrograman Php.
3. Aplikasi *system* pendaftaran *programming course* roblox menggunakan *framework* laravel.

1.6 Sistematika Penulisan

Tujuan dari adanya sistematika penulisan ini ialah untuk mempermudah para pembaca dalam memahami apa saja yang terdapat pada isi laporan ini, sekaligus memberikan gambaran alur yang sistematis dalam menuliskan bab, maka penulisan laporan ini terdiri dari 5 (lima) bab diantaranya adalah sebagai berikut:

BAB 1 PENDAHULUAN

Pada bab I ini merupakan bab pendahuluan yang berisikan: latar belakang masalah, identifikasi masalah, tujuan pembuatan penelitian, ruang lingkup, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab II ini merupakan bab landasan teori yang berisikan mengenai kajian pustaka berdasarkan studi *literature* menurut para peneliti. Landasan teori pada bab ini meliputi: *Related Research*, *Programming Course Roblox*, Implementasi, *Framework Laravel*, Perancangan, Sistem, Bahasa Pemrograman Php, Php My Admin, Mysql, XAMPP, *Visual Studio Code*, UML (*Unified Language Modelling*), Konsep MVC (*Model, View, Controlers*), *Code Coverage*, dan Metode *waterfall*.

BAB III ANALISIS DAN PERANCANGAN

Pada bab III analisis dan perancangan ini akan menjelaskan dan menganalisis proses yang sedang berjalan pada aplikasi sistem pendaftaran *programming course roblox*, untuk mengetahui kebutuhan sistem yang akan dibuat.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab IV implementasi dan pengujian berisikan tentang memaparkan hasil implementasi aplikasi yang dijalankan dan pengujian terhadap aplikasi yang dihasilkan untuk mengukur sejauh mana aplikasi dapat berjalan.

BAB V KESIMPULAN DAN SARAN

Pada bab V kesimpulan dan saran ini menjelaskan kesimpulan yang didapat dalam mengerjakan proyek yang telah dibuat dan saran berupa perbaikan perbaikan yang mendukung dalam pembuatan dan pengembangan aplikasi.

BAB II

LANDASAN TEORI

2.1 Related Research

Dalam penelitian yang penulis lakukan, tentunya memiliki tinjauan studi literatur dari beberapa jurnal yang didapatkan dan melalui *related research* terdapat studi penelitian yang hampir relevan dengan penelitian yang penulis teliti saat ini, diantaranya adalah sebagai berikut:

Penelitian yang dilakukan oleh Zulia Karini, S.S., M.Hum, dan Devi Brilliane pada tahun 2017 dengan judul penelitian “Perancangan *E-Learning* Bahasa Inggris Berbasis *Web* di UPT Bahasa STMIK Amikom Purwokerto”. Penelitian ini bertujuan untuk terwujudnya *e-learning* bahasa Inggris, sehingga nantinya *e-learning* ini mampu meningkatkan kemampuan aspek kognitif siswa. Penelitian ini memakai sebuah metode *model SDLC (System Development Life Cycle) waterfall* dengan tahapan penelitian yaitu *Analyst, Design, Coding and Testing*. Hasil penelitian ini adalah berwujud aplikasi *e-learning* bahasa Inggris berbasis *web*.[2]

Penelitian yang dilakukan oleh Dyah Rhetno Wardhani dan Adhityo Kuncoro pada tahun 2019 dengan judul penelitian “Perancangan Sistem Pembelajaran *E-Learning* pada kursus *online* dengan metode “*Web Based Learning*” menggunakan Dokeos di SMK Nasional Depok”. Penelitian ini bertujuan untuk mempermudah interaksi antara siswa dengan bahan materi, siswa dengan pengajar, maupun antara sesama siswa. Metode penelitian yang dilakukan peneliti menggunakan metode penelitian dan pengembangan (*Research and Development*). Penelitian ini menghasilkan produk berupa sistem pembelajaran secara jarak jauh untuk SMK Nasional Depok.[3]

Penelitian yang dilakukan oleh Dwi Budi Srisulistiwati pada tahun 2021 dengan judul penelitian “Perancangan Sistem Informasi Kursus Di Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika”. Penelitian ini bertujuan memudahkan calon siswa untuk melihat dan memilih program yang tersedia tanpa harus mendatangi tempat kursus. Metode penelitian yang digunakan adalah metode deskriptif analitis yaitu dengan mengolah dan memproses data kemudian dianalisis sehingga dapat

diambil kesimpulan dengan menggunakan alat bantu UML (*Unified Language Modelling*). Hasil dari penelitian yang dilakukan berupa aplikasi berbasis *web* yang berisikan *system* informasi mengenai kursus di sekolah tinggi ilmu komputer cipta karya informatika.[4]

Penelitian yang dilakukan oleh Arisantoso, Bambang Mulyatno, dan Zahid Al Haris pada tahun 2017 dengan judul penelitian “Analisis Dan Rancang Bangun Aplikasi Kursus *Online* Menggunakan Pendekatan Model Diagram Alir Data (Studi Kasus: Universitas Islam Attahiriyyah)”. Penelitian ini bertujuan sebagai wadah bagi tenaga pengajar (dosen) khususnya di Universitas Islam Attahiriyyah untuk memberikan materi pembelajaran yang dapat dipelajari oleh setiap mahasiswa ataupun orang yang ingin menambah pengetahuan diluar mata pelajaran diprogram studinya. Hasil dari penelitian ini adalah setiap tenaga pengajar (dosen) dan mahasiswa dapat menggunakan aplikasi kursus *online* ini sebagai wadah mengajar dan belajar yang lebih fleksibel tanpa perlu terikat waktu dan ruang kuliah, sehingga dapat memberikan motivasi dan kontribusi yang lebih baik dalam proses kegiatan pembelajaran.[5]

Penelitian yang dilakukan oleh Ayu Ramadhani, Hashfi Qaedi Yusman, Ilham Vanny Putra, dan Imam Asrowardipada tahun 2019 dengan judul penelitian “Aplikasi *Online Course Waroeng Inggris Berbasis Web Dengan Menggunakan Framework Codeigniter*”. Penelitian ini bertujuan untuk dapat memudahkan pengguna dalam mengakses konten digital karna dapat diakses di berbagai *device* seperti komputer dan *smartphone* serta digital content disajikan dengan visualisasi yang baik berupa video pembelajaran, gambar dan *text*. Metode pada penelitian ini yakni “*Rapid Application Development (RAD)*” dan dilakukan pengujian menggunakan metode *blackbox testing*. Hasil dalam penelitian ini menunjukan aplikasi dapat berjalan dengan baik sesuai dengan hasil yang di inginkan.[6]

Perbedaan antara penelitian ini dengan penelitian terdahulu yang telah disebutkan di atas adalah penelitian yang dilakukan saat ini lebih memfokuskan pada penggunaan aplikasi yang dapat digunakan dengan mudah oleh pengguna atau pelanggan tanpa perlu adanya pendaftaran kursus yang dilakukan secara datang langsung kelokasi. Aplikasi ini juga sangat fleksibel dan dilakukan secara *online*, jadi peserta kursus hanya perlu menyiapkan persyaratan yang dibutuhkan pada saat pendaftaran kursus.

2.2 Tinjauan Pustaka

2.2.1 Programming Course Roblox

Programming course roblox adalah salah satu kursus untuk anak-anak yang bergerak di bidang pemrograman, dimana siswa diajarkan untuk membuat *game* sederhana dalam aplikasi roblox. Aplikasi roblox adalah sebuah aplikasi berkategori *game* yang didesain lintas *platform* yang artinya banyak dimana beberapa orang lebih dari beberapa pengguna mancanegara yang memainkan *game* ini secara *online*. Menurut Artasasmita (dalam Hatimah dan Sadri, 2008:44) kursus adalah sebuah kegiatan pendidikan yang berlangsung di dalam masyarakat yang dilakukan secara sengaja, terorganisir, dan sistematis untuk memberikan materi pelajaran tertentu kepada orang dewasa, anak-anak atau remaja dalam waktu yang *relative* singkat agar dapat memperoleh pengetahuan, keterampilan, dan sikap yang dapat dimanfaatkan untuk mengembangkan potensi diri.

2.2.2 Implementasi

Menurut Cleaves dalam Wahab (2008: 187), yang secara tegas menyebutkan bahwa implementasi mencakup proses bergerak menuju kebijakan dengan cara langkah administratif dan politik. Keberhasilan atau kegagalan implementasi sebagai demikian dapat dievaluasi dari sudut kemampuannya secara nyata dalam mengoprasionalkan program program yang telah dirancang sebelumnya. Terdapat tahapan dalam implementasi yakni diantaranya:

- A. Pertama, (*Planning*) merupakan proses penetapan tujuan, mengembangkan strategi, dan menguraikan tugas dan jadwal untuk mencapai tujuan yang terarah dan sistematis agar tujuan dapat dicapai secara efektif dan efisien.
- B. Kedua (*Explary*) merupakan mensyaratkan komitmen untuk memberikan contoh terbaik dalam setiap tingkah laku seorang yang mau jujur dengan tidak dibuat buat, tetapi asli muncul kepermukaan sebagai sebuah kepribadian.
- C. Ketiga (*Pelaksanaan*) merupakan suatu tindakan dari sebuah rencana yang disusun secara matang dan terperinci.

2.2.3 Framework Laravel

Laravel adalah *framework* PHP dimana agar dapat bersifat kode terbuka (*open source*) dengan sebuah konsep suatu yaitu MVC (*Model View Controllers*) yang digunakan untuk membangun aplikasi berbasis *website*. Laravel merupakan sebuah kerangka kerja pemrograman yang berbasis *open source* yang dipakai oleh banyak *developer* dari seluruh dunia. *Framework* laravel memiliki fungsi yakni melakukan sebuah proses *migrate* atau pindah sehingga keuntungannya agar dapat lebih mudah digunakan dalam melakukan proses pengelolahan basis datanya. *Framework* laravel memiliki *templating engine* yang dapat membantu membangun tampilkan *front end* lebih efisien dengan fungsi *blade* yang telah disediakan oleh laravel. Laravel menjadi primadona dalam beberapa tahun terakhir (Mulyadi, 2015). Laravel juga menjadi salah satu *framework* yang dapat membantu *developer* untuk memaksimalkan penggunaan PHP didalam proses pengembangan *website*.

2.2.4 Perancangan

Perancangan merupakan salah satu hal yang sangat penting dalam membuat suatu program dalam kehidupan sehari hari. Perancangan atau rancang merupakan beberapa urutan proses langkah langkah dimana berfungsi untuk mengartikan hasil sebuah analisis dari suatu sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan sebuah detail detail dari komponen komponen sebuah sistem dalam perangkat lunak agar dapat di terapkan. Dengan demikian pengertian perancangan dapat diartikan adalah kegiatan kegiatan yang dapat menerjemahkan hasil analisis suatu proses kegiatan didalam suatu program kedalam suatu paket paket perangkat lunak.

2.2.5 Sistem

Sistem juga merupakan suatu kesatuan, baik objek nyata atau abstrak yang terdiri dari berbagai komponen atau unsur yang saling berkaitan, saling tergantung, saling mendukung, dan secara keseluruhan bersatu dalam sebuah kesatuan yang dapat ditentukan agar dapat lebih efektif dan efisien digunakan di berbagai perangkat. Jadi,

dapat disimpulkan bahwa sistem adalah komponen komponen yang saling berkaitan dan bekerjasama untuk mencapai suatu tujuan tertentu.

2.2.6 Bahasa Pemrograman Php

Php adalah bahasa *server side scripting* dimana dapat berguna untuk membuat suatu desain agar dapat digunakan untuk sebuah *web*. Maksud dari *server side scripting* adalah sebuah *script script function* yang mana didefinisikan sebagai serangkaian suatu proses perintah yang di berikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada dokumen *HTML* biasa. Php menawarkan konektifitas yang baik dengan beberapa basis data antara lain *Oracle*, *Sybase*, *MySql*, dan *PostgreSQL*.

2.2.7 PhpMyAdmin

PhpMyAdmin adalah sebuah program aplikasi yang mana dapat bersifat *open source* yang memiliki fungsi untuk memudahkan dalam mengatur suatu *database MySql*. Dengan menggunakan PhpMyAdmin, dapat membuat *database*, membuat tabel atau *insert*, menghapus dan *update* data terasa lebih mudah, tanpa perlu mengetikkan perintah sql secara manual. PhpMyAdmin adalah sebuah aplikasi atau program komputer yang digunakan dengan fungsi agar dapat memudahkan dan menyederhanakan dalam melakukan proses pengelolahan dalam basis data.

2.2.8 MySql

MySql adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi atau sebuah sistem yang berfungsi agar dapat menjalankan proses proses pengolahan suatu basis data. MySql adalah sebuah *software* atau perangkat lunak yang bersifat terbuka atau dapat diakses oleh banyak orang fungsinya adalah digunakan untuk membuat sebuah *database*. Sql dapat disebut sebagai singkatan dari *Structured Query Language*. MySql berfungsi untuk menambahkan, mengubah, dan menghapus data dalam *database*, hal ini sangat berguna bagi pekerjaan yang membutuhkan data yang cukup luas.

2.2.9 XAMPP

Menurut Kartini (2013:27-26) Xampp merupakan *tools* yang menyediakan paket perangkat lunak ke dalam satu buah sistem. Dengan kata lain Xampp merupakan paket berbasis *open source*, yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi.

2.2.10 Visual Studio Code

Visual studio code adalah sebuah *tools* dengan *editor* yang bersifat *open source code* yang dikembangkan oleh *Microsoft* untuk *Windows*, *Linux* dan *MacOS*. Didalam *visual studio code* ini juga termasuk untuk layanan mengaplikasikan berbagai kode program dengan masing masing *extension* nya yang sudah ada dan terminal nya tersendiri. Biasanya juga dalam *visual studio code* digunakan untuk merancang sebuah *website* maupun aplikasi. *Visual studio code* dalam hal ini, digunakan untuk merancang sebuah kode program untuk *application* dimana *tools visual studio code* ini sangat berguna untuk merancang sebuah kode program secara terstruktur, cepat, efisien dan dapat menyesuaikan dengan sendirinya.

2.2.11 UML (*Unified Modelling Language*)

UML (*Unified Modeling Languange*) adalah sebuah sistem yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun dan pendokumentasi dari sebuah sistem pengembangan *software* berbasis OO (*Object Oriented*). UML (*Unified Modeling Languange*) terdapat bagiannya masing masing, diantaranya terdapat 4 jenis seperti: *Use case diagram*, *Activity diagram*, *Sequence diagram*, dan *Class diagram*.

2.2.12 Konsep MVC (*Model, View, Controllers*)

Konsep yang digunakan dalam membangun aplikasi menggunakan konsep MVC (*Model View, Controllers*) dimana dalam *model view controll* ini mempunyai bagian diantaranya adalah: *model*, *view*, dan *controll*. *Model* mempunyai fungsi untuk mengatur dan mengkoordinasikan suatu data data yang ada dari dalam *database* sesuai

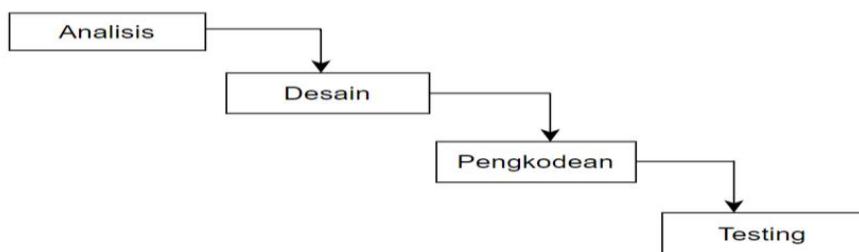
dengan instruksi dari adanya *controllers*. *View* mempunyai fungsi untuk menyediakan layanan berupa informasi dengan kata lain *view* ini merupakan desain gambaran yang dihasilkan dari proses *controllers*. Sedangkan *controllers* ini mempunyai tugas untuk mengatur jalannya proses yang akan dilakukan oleh suatu *system* dan menerima permintaan dari *client* atau *user* sehingga *outputnya* akan menampilkan berupa *view*.

2.2.13 Code Coverage

Code coverage adalah sebuah pengujian dengan menitik beratkan terhadap kode program yang akan diuji atau dilakukan pengujian dari segi *source code* nya, kemudian *source code* yang telah dibuat dapat dilakukan proses testing dengan minimal untuk pengujian *code coverage* yakni 80%. Karena dalam pengujian kode program harus diketahui terlebih dahulu *input* dan *output* yang dijalankan apakah sudah sesuai atau belum.

2.2.14 Metode Waterfall

Metode yang digunakan dalam merancang aplikasi ini ialah metode *waterfall*. *Metode waterfall* adalah metode yang digunakan untuk tahap pengembangan suatu sistem dengan model alur hidup atau dapat dikatakan sebagai (*Classic Cycle*). Dalam metode pembuatan aplikasi ini menerapkan sebuah metode *waterfall* dimana metode *waterfall* banyak digunakan untuk pengembangan suatu *website*. Dalam kata lain, metode *waterfall*. Terdapat tahapan proses yang terlibat dalam metode *waterfall* dimana dalam metode ini dispesifikasikan terhadap perancangan perangkat lunak suatu sistem khususnya dalam sistem pendaftaran *programming course* roblox, yaitu ada 4 tahapan dalam merancang pembuatan aplikasi ini, seperti pada gambar berikut:



Gambar 2.1 Alur Metode *Waterfall*

1. Analisis Kebutuhan Perangkat Lunak

Pada tahap *analisis ini adalah* tahap awal ini terdapat kegiatan pengumpulan kebutuhan perangkat lunak yang diperlukan untuk menggambarkan sebuah sistem pendaftaran *programming course* roblox yang akan dibangun secara fungsional dari aplikasi yang dirancang. Dalam tahap ini juga dilakukan langkah langkah berupa: mendefinisikan masalah, menentukan tujuan pembuatan aplikasi sistem pendaftaran *programming course* roblox, dan mengidentifikasi kendala kendala.

2. *Design* (Perancangan)

Tahap *design* merupakan bagian dari perancangan aplikasi agar sesuai dengan kebutuhan dari penggunanya yaitu berdasarkan pada fase kedua ini. Pada fase ini dilakukan kegiatan perancangan sistem berdasarkan *Use Case Diagram*, *Class Diagram*, *Activity Diagram* dengan menggunakan pemodelan UML (*Unified Model Diagram*). Selain itu, alur kerja, dan basis data juga termasuk pada penelitian ini.

3. *Coding* (Pengkodean)

Tahap *coding* ini dilakukan proses pembuatan kode program pada aplikasi *programming course* roblox pada perangkat lunak yang digunakan dalam pengembangan aplikasi yakni dengan perangkat *visual studio code* dimana pada *visual studio code* ini memiliki fungsi untuk melakukan kode program baik berbasiskan *website* maupun aplikasi android.

4. *Testing* (Pengujian)

Tahap pengujian berfungsi untuk menguji apakah aplikasi yang dibangun sesuai dengan fitur dan fungsionalitas yang diharapkan oleh pengguna. Selain itu dilakukan juga pengujian kode program yang bertujuan untuk mengurangi waktu pengembangan dan meminimalisir adanya *bugs* atau *error*. Pada penngujian aplikasinya menggunakan tools selenium IDE, dan juga menggunakan *whitebox testing* dengan menerapkan Php unit sebagai rujukan untuk mengecek kode program nya.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis

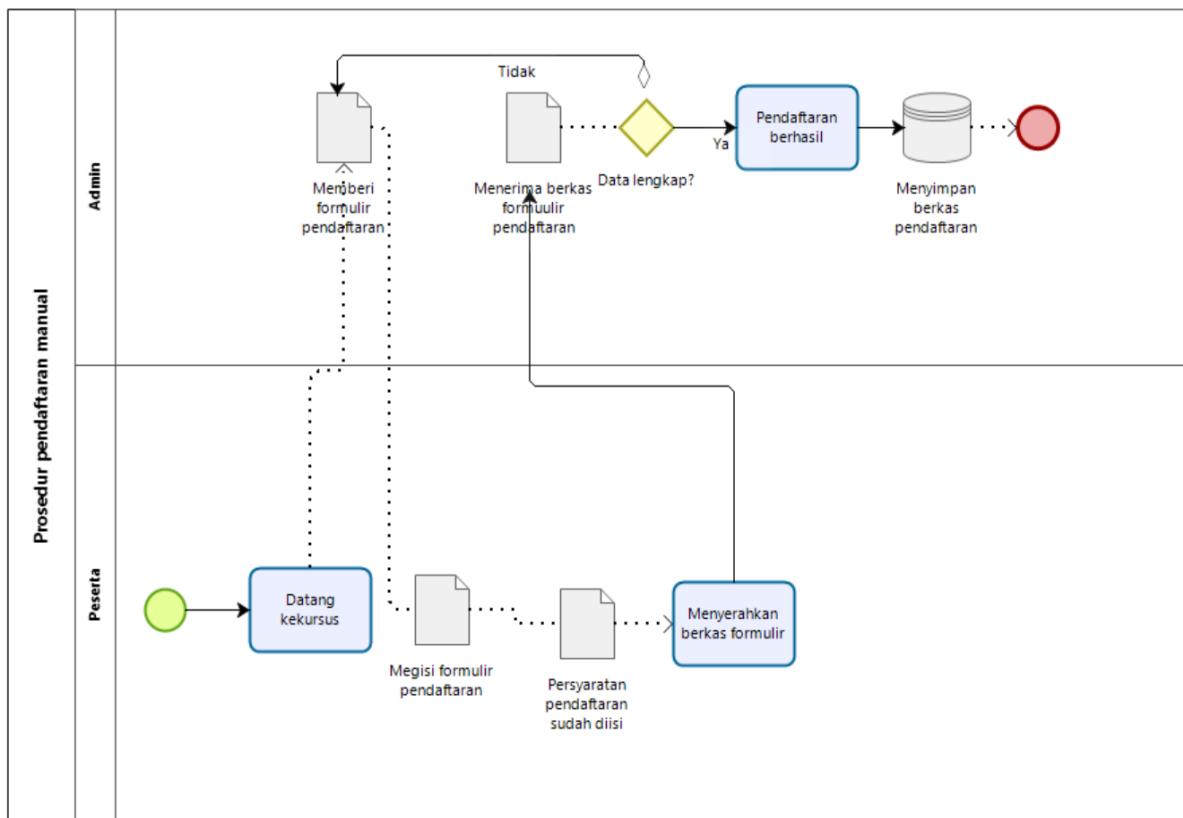
Analisis merupakan kegiatan proses yang menguarikan dari serangkaian proses dari suatu sistem informasi yang disusun secara utuh dengan tujuan melihat permasalahan yang ada dan mengusulkan perbaikannya. Dengan kata lain, analisis adalah tahap yang sangat penting karena suatu kesalahan dalam tahap ini akan mempengaruhi pada tahap berikutnya. Tahap Analisis merupakan kegiatan pemecahan dari suatu sistem informasi yang utuh ke dalam bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan yang terjadi sehingga dapat diusulkan perbaikannya. Pada tahap ini berisi hal-hal yang berhubungan dengan analisis sistem berjalan, prosedur kerja deskripsi dokumentasi, kebutuhan pengguna sistem, kebutuhan perangkat keras dan lunak, dengan maksud untuk mengidentifikasikan segala permasalahan atau hambatan hambatan yang terjadi. Dengan kata lain, tahap analisis bertujuan untuk mengetahui cara kerja dari sistem dengan melakukan beberapa pengamatan dan pengujian.

3.1.1 Analisis Sistem Berjalan (*Current System*)

Analisis Pada sub bab ini akan dijelaskan mengenai sistem pendaftaran *programming course* roblox yang sedang berjalan. Adapun sistem yang berjalan saat ini terdiri dari 1 prosedur yaitu pendaftaran kursus.

3.1.1.1 Analisis Sistem Yang Berjalan Pada Pendaftaran Kursus

Saat ini, prosedur pendaftaran kursus terdiri dari 2 entitas atau *actor* dimana diantaranya terdapat peserta dan juga admin. Prosedur pendaftaran yang dilakukan peserta dengan cara manual yakni mengirimkan berkas pendaftaran nya kepada admin atau melalui datang ke kursus, kemudian admin memverifikasi data data tersebut seperti data kelengkapan berkas pendaftaran dan juga berkas bukti pembayaran. Serta terdapat *symbol database* yang masih manual tersimpan di laporan buku, Berikut adalah uraian penjelasan mengenai prosedur pendaftaran yang sedang berjalan pada *programming course* roblox:



Gambar 3.1 *Flowmap* Analisis Sistem Yang Sedang Berjalan Pada Prosedur Pendaftaran Kursus

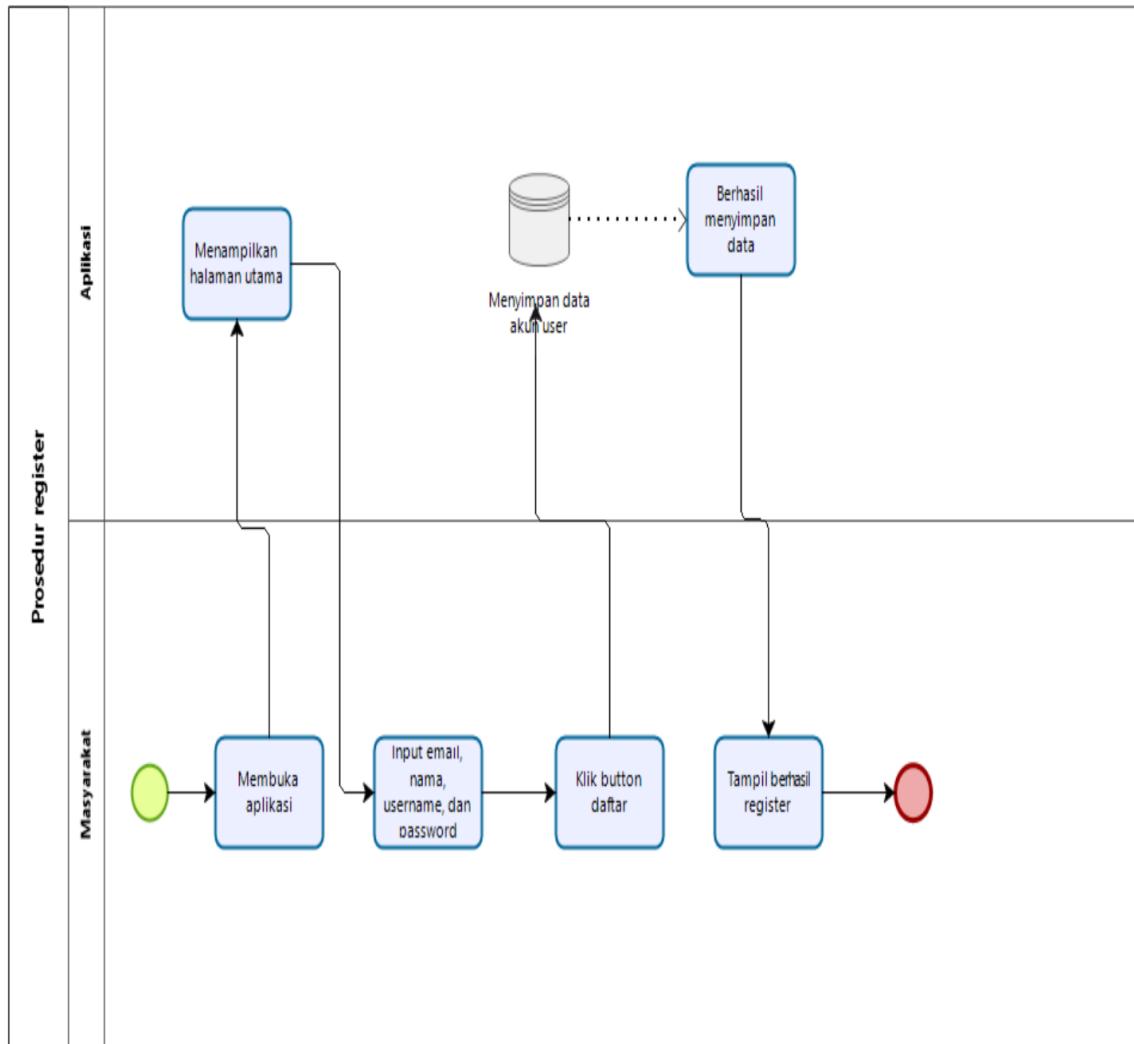
3.1.2 Analisis Sistem Yang Akan Dibangun

Analisa sistem yang akan dibangun disini bertujuan untuk mengetahui lebih jelas bagaimana sistem yang akan dibangun tersebut dari tahap ke tahap serta masalah yang dihadapi sistem untuk dapat dijadikan landasan usulan perancangan analisa sistem yang sedang berjalan. Analisis sistem yang akan dibangun berdasarkan urutan kegiatan yang ada dan dari urutan kegiatan tersebut dapat dibuat berupa analisis *flowmap* mulai dari prosedur *register*, *login*, pendaftaran, pembayaran, pengumuman, kelola data pengguna, kelola data sekolah, kelola data program kursus, kelola data pendaftaran, kelola data pembayaran, kelola data pengumuman, kelola data profil dan *logout* adalah sebagai berikut:

3.1.2.1 Analisis sistem yang akan dibangun pada prosedur *register*

Pada prosedur *register* terdapat 2 entitas dimana antara lain masyarakat dan aplikasi. Setiap entitas atau *actor* tersebut memiliki kewajiban atau hak dalam aplikasi yang akan dibangun tersebut. Kewajiban atau hak dari masyarakat disini ialah memasukkan *username*, *password* dan nama sebagai pelengkap *register* yang dibutuhkan sehingga

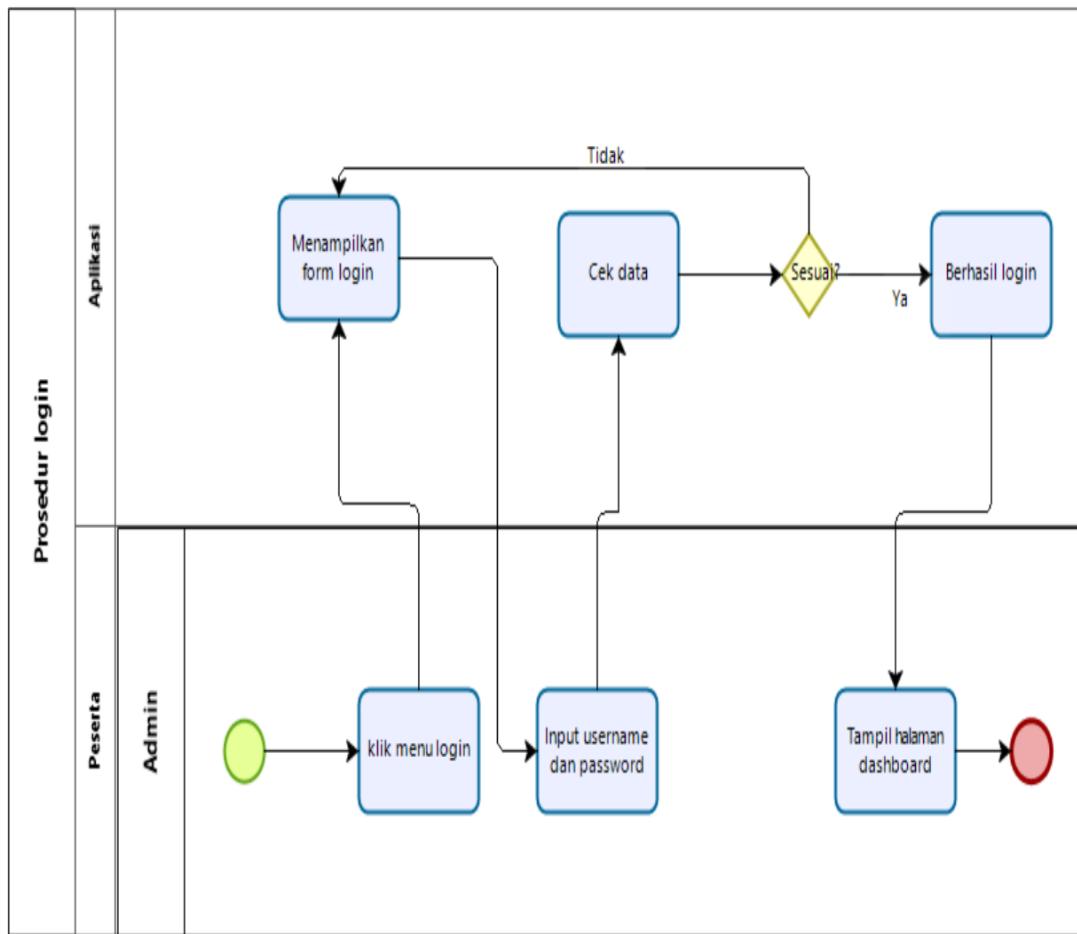
masyarakat dapat masuk ke aplikasi *system* tersebut dengan akun *username* dan *password* tersebut. Adapun *Flowmap* dari prosedur *system* yang akan dibangun pada *register* adalah sebagai berikut:



Gambar 3.2 *Flowmap* Analisis Sistem Yang Akan Dibangun Pada Prosedur *Register*

3.1.2.2 Analisis sistem yang akan dibangun pada prosedur *login*

Pada prosedur *login* ini terdapat 3 entitas yang terlibat didalamnya, diantaranya adalah admin, peserta, dan aplikasi. Setiap *actor* atau entitas tersebut memiliki kewajiban dalam sistem pendaftaran *programming course roblox* tersebut. Kewajiban admin dan peserta dalam prosedur *login* ini memasukkan *username* dan *password* sehingga peserta dan admin dapat masuk ke aplikasi sistem tersebut dengan akun *username* dan *password* tersebut. Adapun *flowmap* dari prosedur *system* yang akan dibangun pada *login* adalah sebagai berikut:

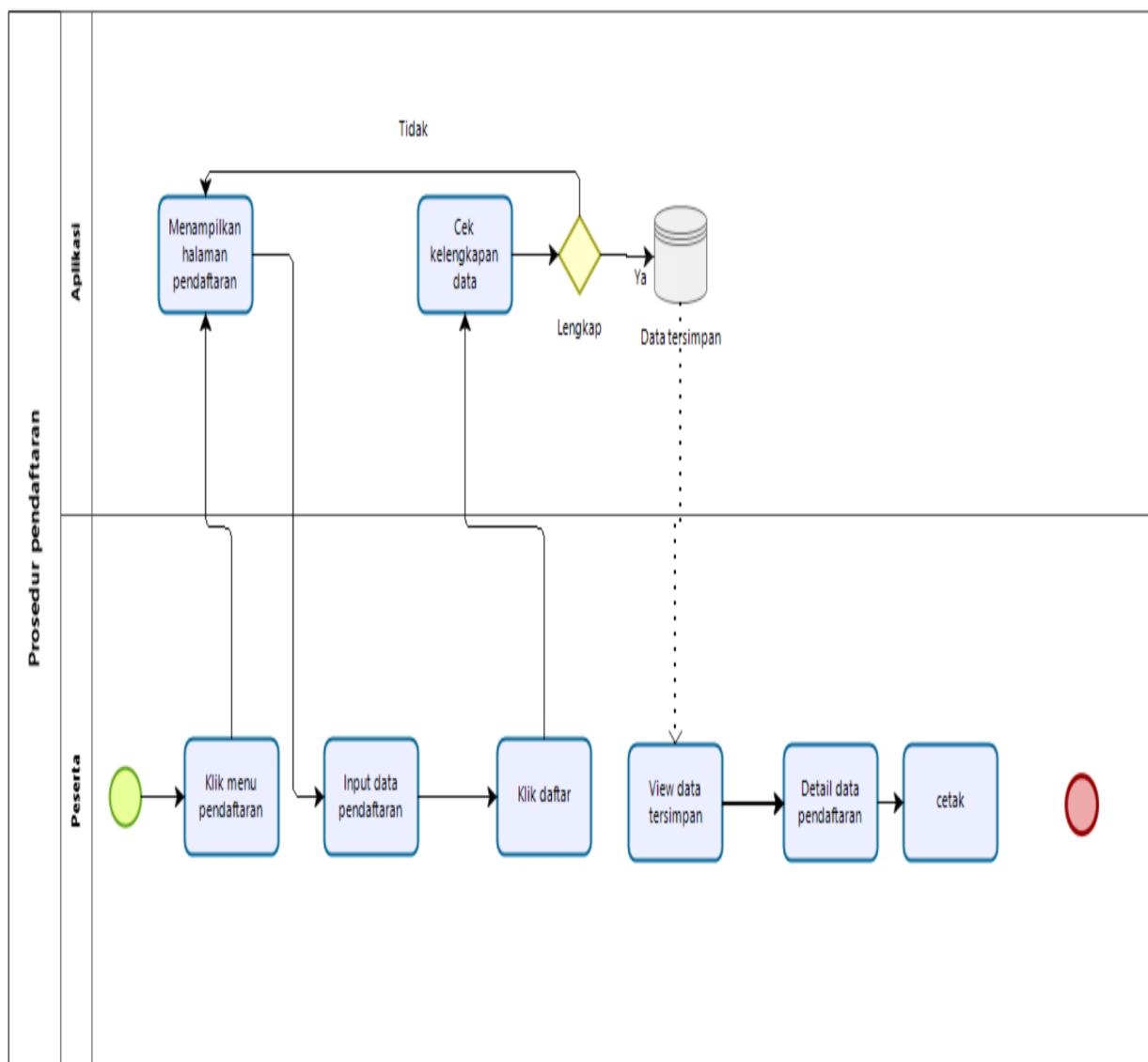


Gambar 3.3 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Login

3.1.2.3 Analisis sistem yang akan dibangun pada prosedur pendaftaran

Pada prosedur pendaftaran kursus ini, akan melibatkan diantaranya seperti 2 entitas yakni peserta dan aplikasi. Setiap *actor* atau entitas tersebut memiliki kewajiban dalam sistem *programming course roblox* tersebut. Kewajiban atau hak dari peserta disini ialah peserta dapat mendaftar dengan melengkapi data data kursus tersebut seperti data pribadi, data sekolah, data pilihan jenis kursus, data orang tua serta data pendidikan. Untuk mendapatkan gambaran mengenai analisis *system* yang akan dibangun pada prosedur pendaftaran dapat dilihat pada uraian dibawah ini:

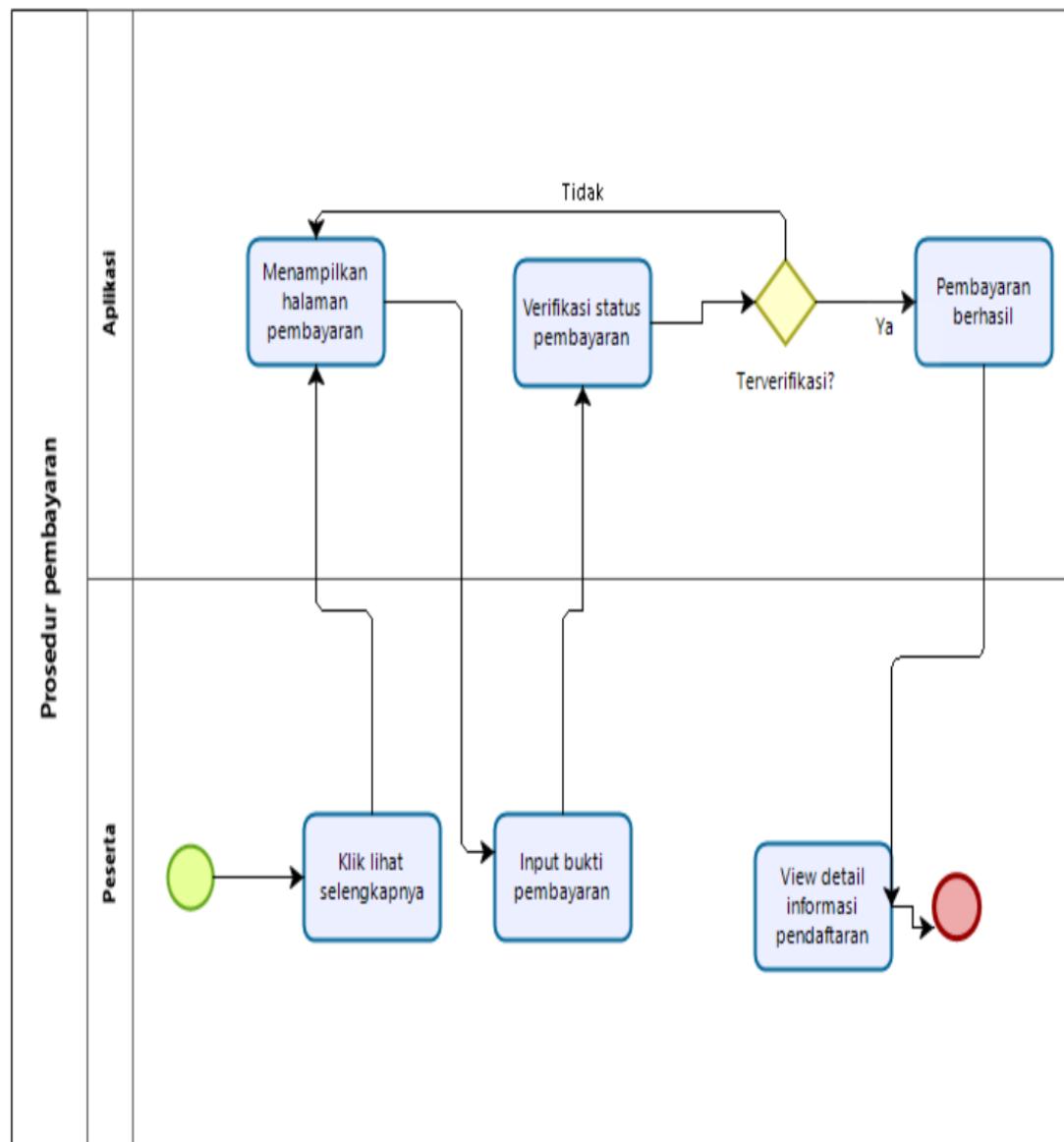
1. Peserta melakukan serangkaian pengisian pendaftaran sesuai dengan persyaratan dimana pendaftaran ini dilakukan pada *form* pendaftaran.
2. Apabila selesai mengisi data pendaftaran maka aplikasi akan memunculkan notifikasi “Data berhasil tersimpan” dan diarahkan ke halaman beranda.



Gambar 3.4 *Flowmap* Analisis Sistem Yang Akan Dibangun Pada Prosedur Pendaftaran

3.1.2.4 Analisis sistem yang akan dibangun pada prosedur pembayaran

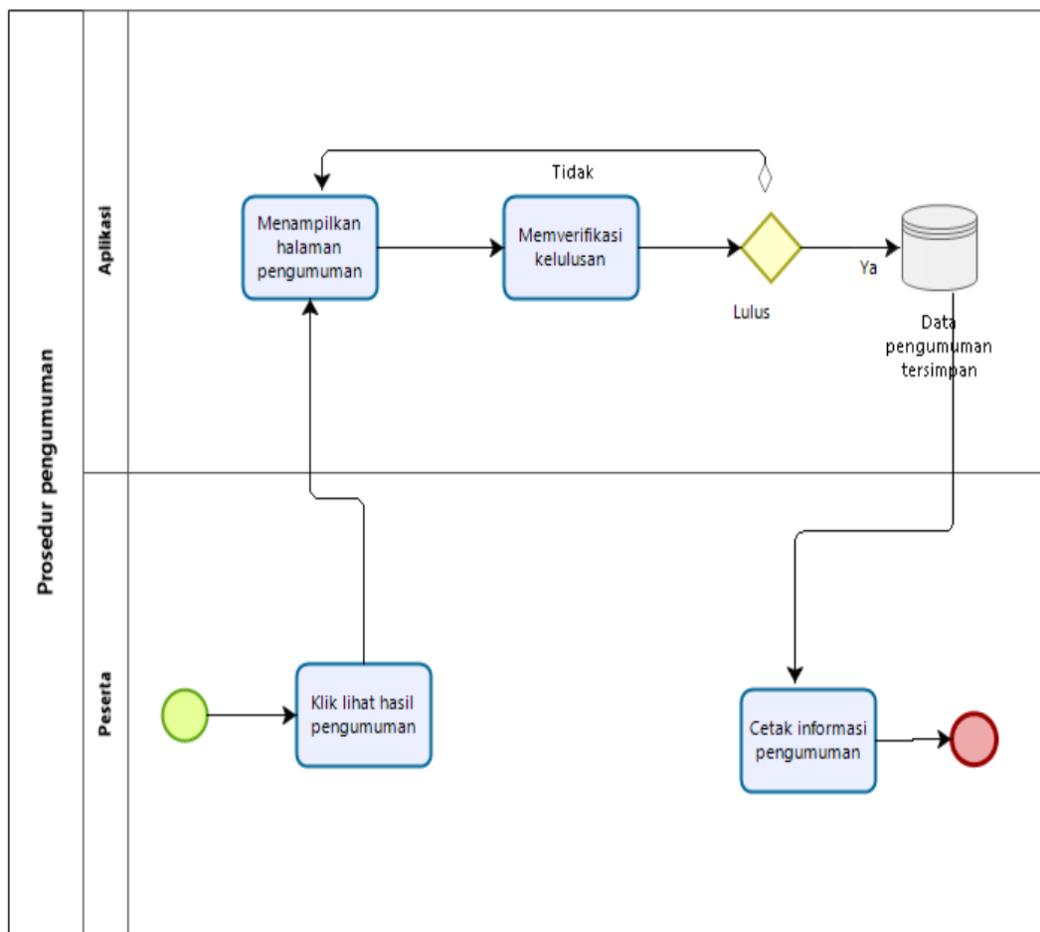
Pada prosedur pengumuman kursus ini, akan melibatkan diantaranya seperti 2 entitas yakni peserta dan aplikasi. Setiap *actor* atau entitas tersebut memiliki kewajiban dalam kursus tersebut. Kewajiban peserta adalah melakukan pembayaran kursus dengan memasukkan bukti pembayaran yang telah dilakukan serta dapat melihat status pembayaran yang telah dilakukan. Kemudian aplikasi dapat menyimpan data pembayaran tersebut. Untuk mendapatkan gambaran mengenai analisis *system* yang akan dibangun pada prosedur pembayaran dapat dilihat pada uraian dibawah ini:



Gambar 3.5 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Pembayaran

3.1.2.5 Analisis sistem yang akan dibangun pada prosedur pengumuman

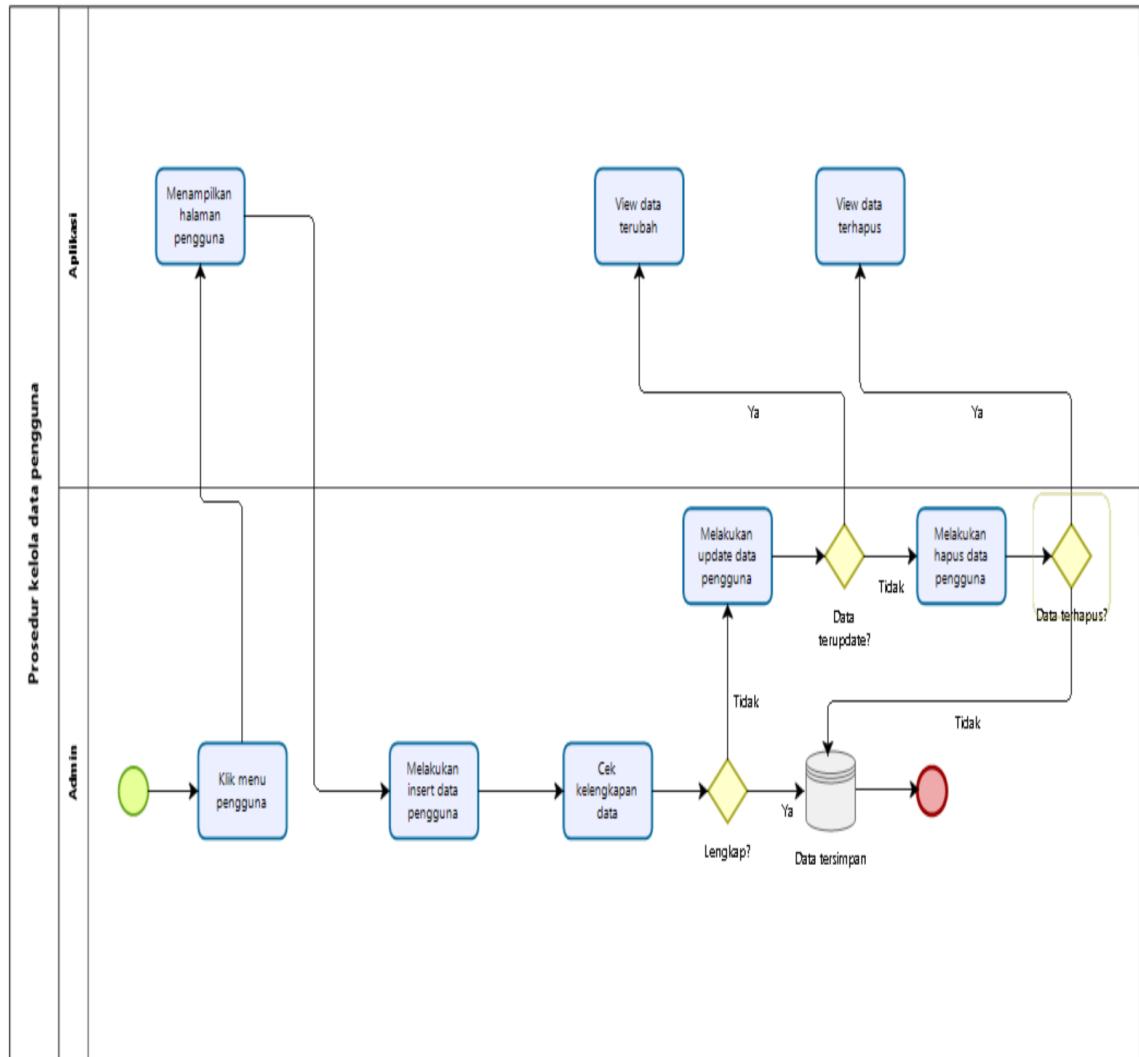
Pada prosedur pengumuman kursus ini, akan melibatkan diantaranya seperti 2 entitas yakni peserta dan aplikasi. Setiap *actor* atau entitas tersebut memiliki kewajiban dalam kursus tersebut. Kewajiban peserta dalam prosedur terdapat pada *form* pengumuman adalah melihat informasi dari pengumuman tersebut dan mencetak hasil pengumuman. Kemudian pada aplikasi akan melakukan simpan data di *database* untuk data pengumuman. Untuk mendapatkan gambaran mengenai analisis *system* yang akan dibangun pada prosedur pengumuman dapat dilihat pada uraian gambar dibawah ini:



Gambar 3.6 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Pengumuman

3.1.2.6 Analisis sistem yang akan dibangun pada prosedur kelola data pengguna

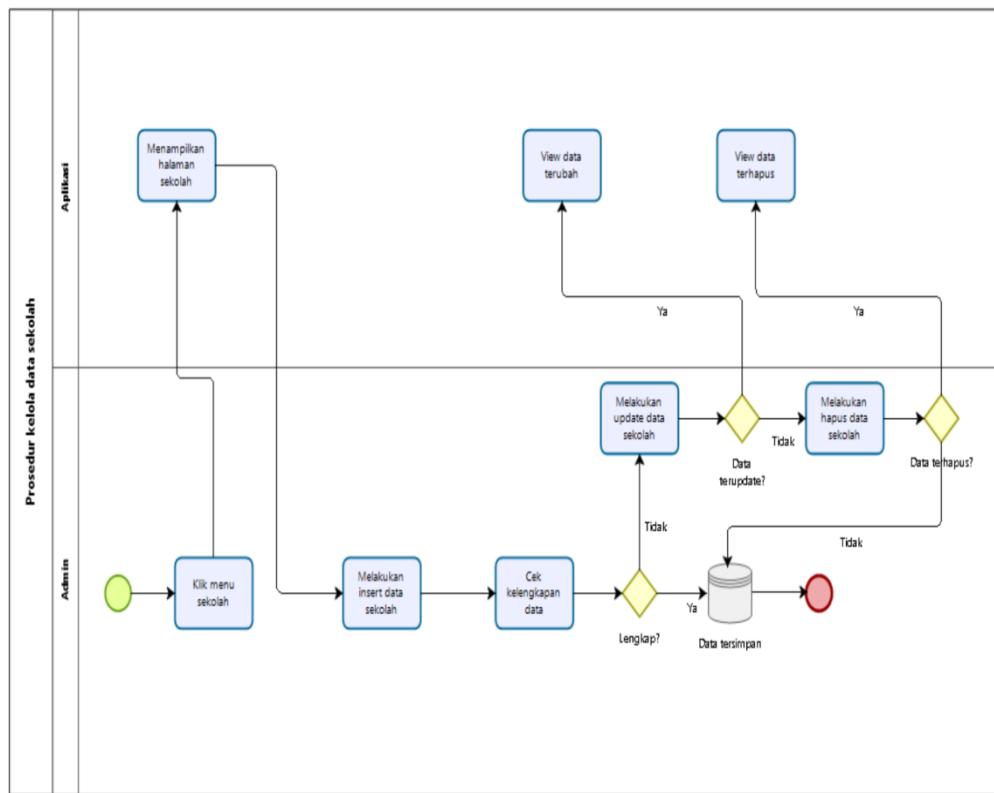
Pada prosedur pengumuman kursus ini, akan melibatkan diantaranya seperti 2 entitas yakni admin dan aplikasi. Kewajiban admin dalam prosedur kelola data pengguna ini ialah melihat dan mengelola data pengguna kursus yang sudah mendaftar kursus, serta admin juga dapat melakukan proses menambahkan data, menghapus serta memperbarui data pengguna. Admin dapat mengelola data pengguna yang mendaftar kursus seperti nama, alamat, nomor telepon dan lain sebagainya dan admin dapat menambahkan pengguna baru, menghapus atau dapat juga mengupdate data peserta kursus yang telah mendaftar. Sedangkan perbedaan pada peserta, dimana peserta hanya dapat melakukan pengubahan data atau perbarui data. Untuk mendapatkan gambaran mengenai analisis *system* yang akan dibangun pada prosedur kelola data pengguna dapat dilihat pada uraian gambar dibawah ini:



Gambar 3.7 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Pengguna

3.1.2.7 Analisis sistem yang akan dibangun pada prosedur kelola data sekolah

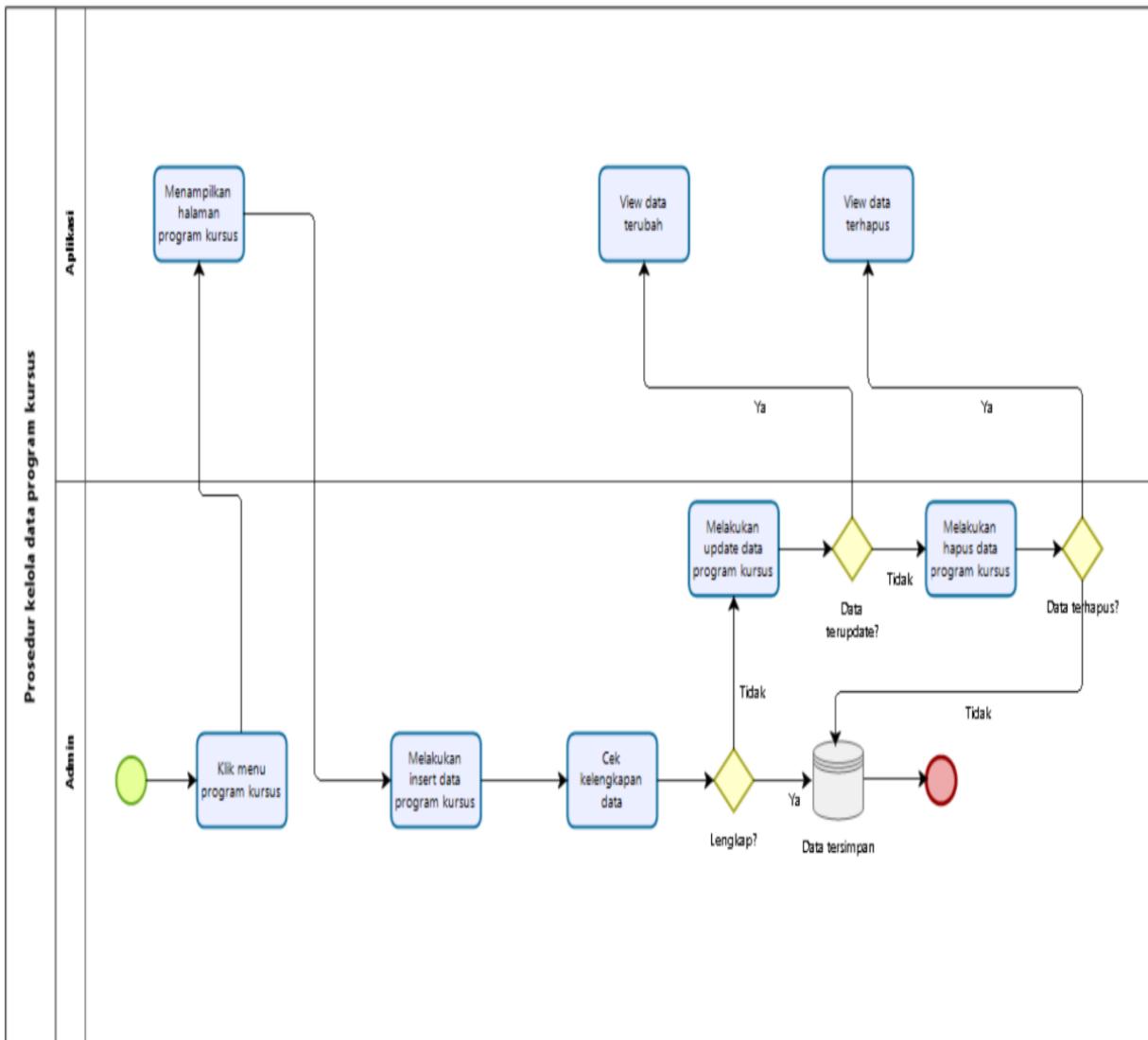
Kewajiban admin dalam prosedur kelola data sekolah ini ialah mengelola data sekolah yang mana akan tertera pada *form* pendaftaran sebagai salah satu persyaratan kelengkapan data pendaftaran yakni mengisi data sekolah. Admin berkewajiban mengelola dan memasukkan seperti nama sekolah, alamat, kota, NPSN sekolah dari berbagai macam sekolah yang ada, serta admin juga dapat melakukan proses menambahkan, data menghapus serta memperbarui data sekolah. Untuk mendapatkan gambaran mengenai analisis *system* yang akan dibangun pada prosedur kelola data sekolah dapat dilihat pada uraian dibawah ini:



Gambar 3.8 *Flowmap* Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Sekolah

3.1.2.8 Analisis sistem yang akan dibangun pada prosedur kelola data program kursus

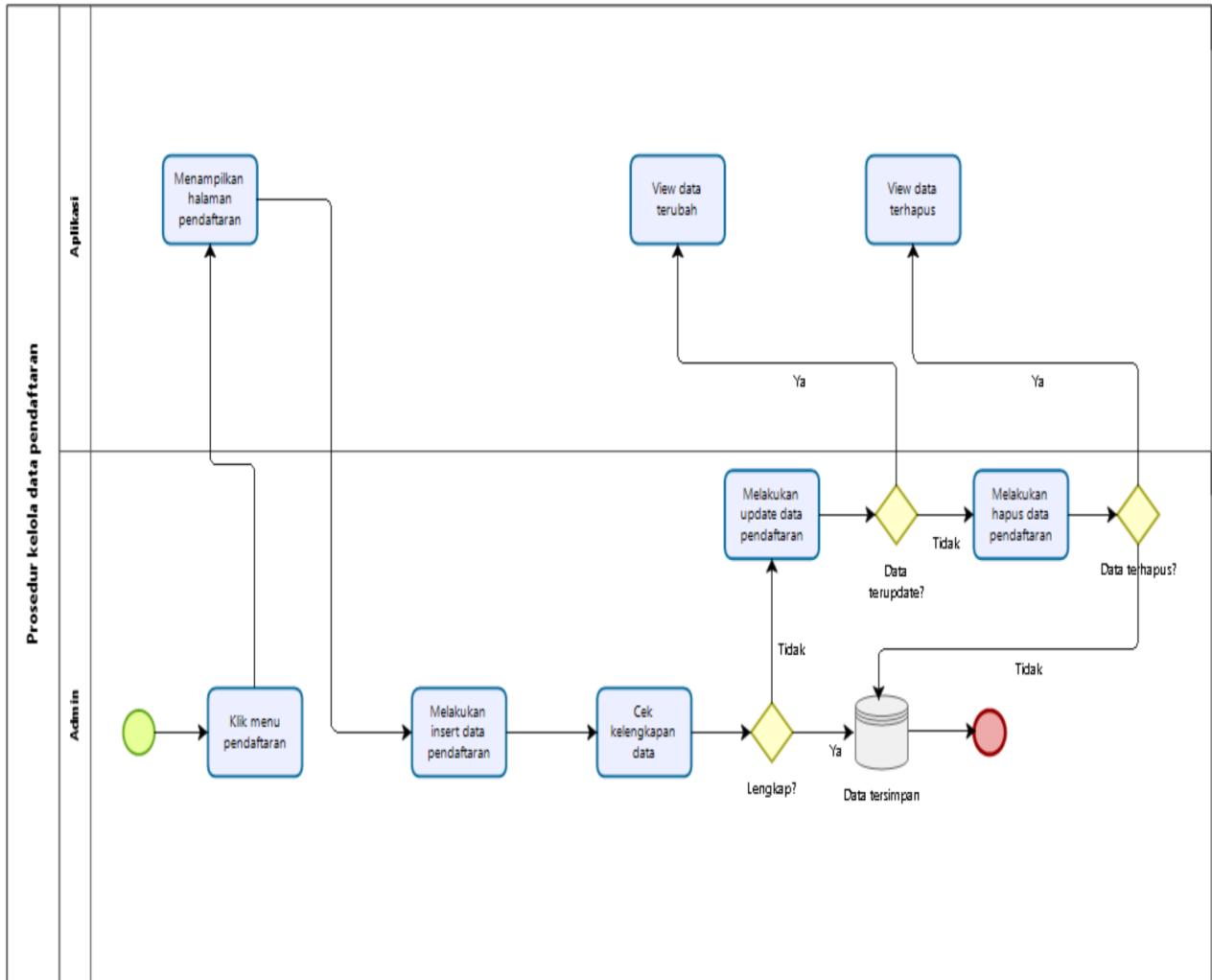
Terdapat 2 entitas atau *actor* yang mewakili proses kelola data program kursus ini, diantaranya yakni admin dan aplikasi. Kewajiban admin dalam prosedur kelola data program kursus ini ialah mengelola data program kursus serta admin juga dapat melakukan proses menambahkan, data menghapus serta memperbaharui data program kursus. Data data program kursus ini harus dipilih oleh peserta sebab, sebagai salah satu persyaratan pendaftaran oleh peserta. Program kursus yang telah tersedia ialah seperti: *Puzzle game* (program kursus 1), *Fighting game* (program kursus 2), *Sport game* (program kursus 3), dan *Education game* (program kursus 4). Diantara program kursus tersebut memiliki beberapa pengajar, *link demo game*, harga, jam, dan hari yang sudah tertera. Data data program kursus tersebut tersimpan oleh aplikasi yakni di *database*. Untuk mendapatkan gambaran mengenai analisis *system* yang akan dibangun pada prosedur kelola data program kursus dapat dilihat pada uraian dibawah ini:



Gambar 3.9 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Program Kursus

3.1.2.9 Analisis sistem yang akan dibangun pada prosedur kelola data pendaftaran

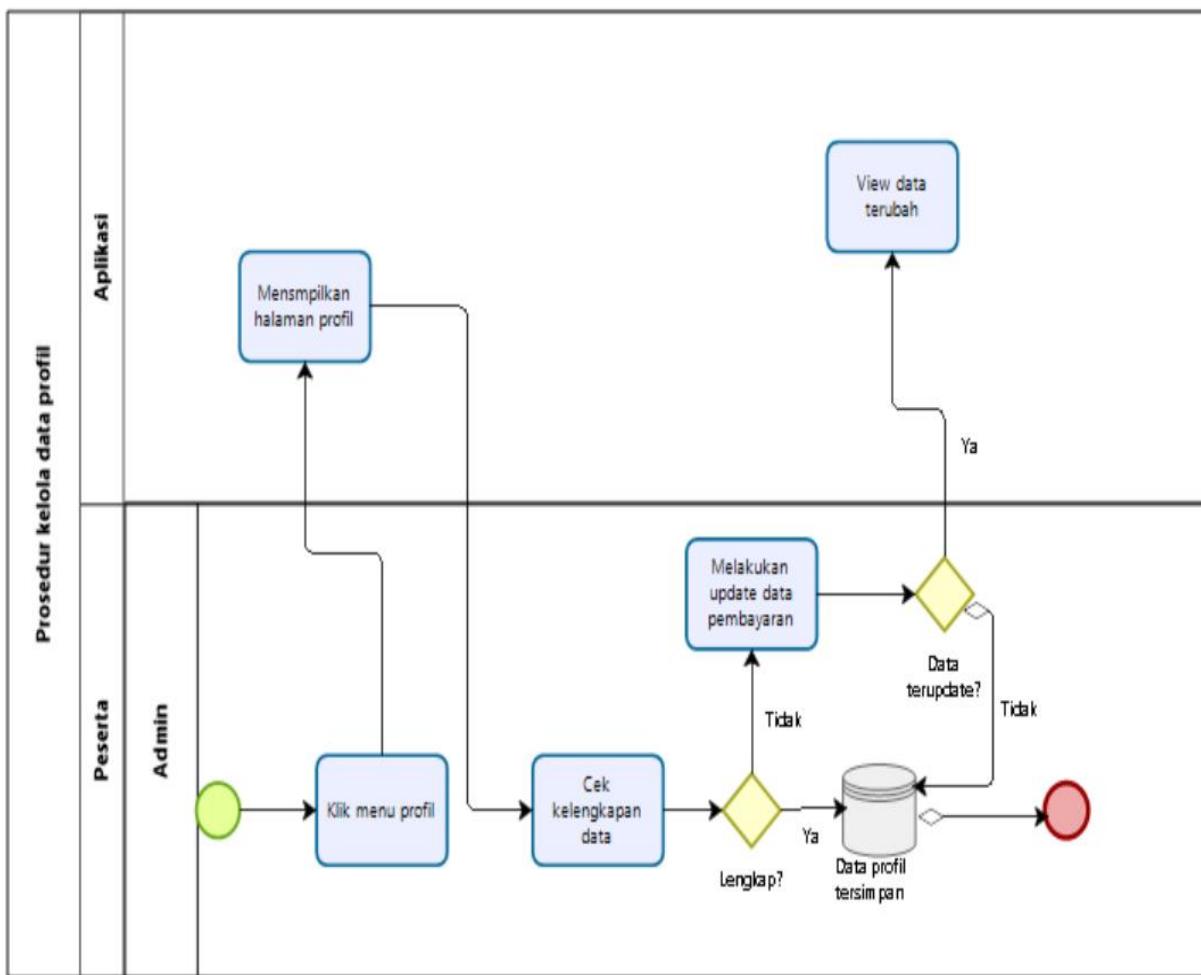
Kewajiban admin dalam prosedur kelola pendaftaran ini ialah mengelola data pendaftaran, admin juga dapat melakukan proses menambahkan, data menghapus serta memperbaharui data pendaftaran seperti: No, id pendaftaran, data orangtua, data pribadi, dan data Pendidikan peserta apabila peserta ingin didaftarkan. Data data pendaftaran tersebut disimpan oleh aplikasi di *database* nya. Untuk mendapatkan gambaran mengenai analisis prosedur *system* yang akan dibangun pada kelola data pendaftaran dapat dilihat pada uraian dibawah ini:



Gambar 3.10 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Pendaftaran

3.1.2.10 Analisis sistem yang akan dibangun pada prosedur kelola data pembayaran

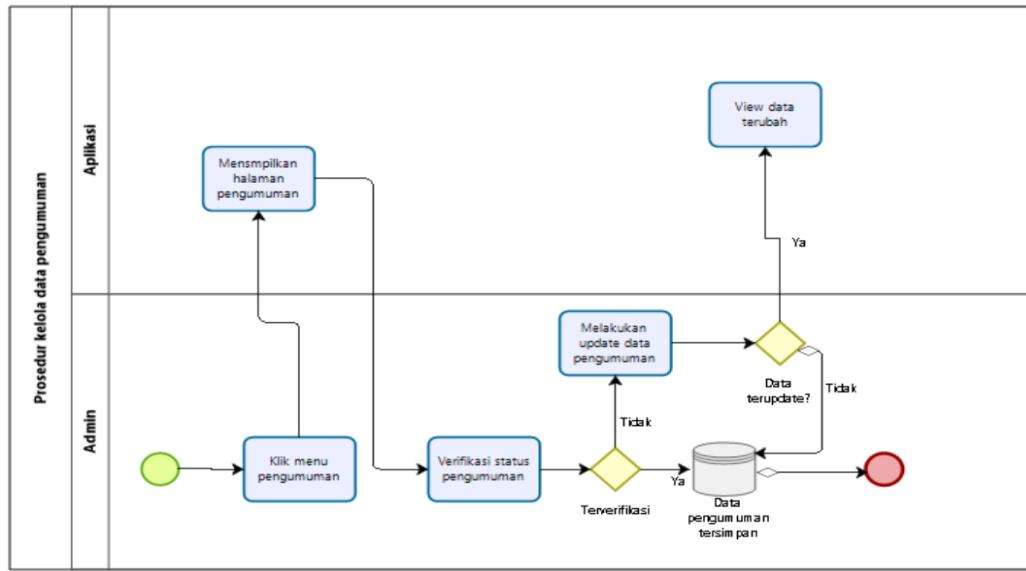
Terdapat 2 entitas atau *actor* yang mewakili proses kelola data pembayaran, diantaranya yakni admin dan aplikasi. Kewajiban admin dalam prosedur pembayaran di kursus ini ialah mengecek status pembayaran dari peserta kursus, jika sudah membayar maka admin dapat memverifikasi status pembayaran tersebut. Kemudian aplikasi dapat melakukan penyimpanan data pembayaran seperti bukti pembayaran yang dilakukan oleh peserta. Admin juga dapat melakukan proses *update* pembayaran. Untuk mendapatkan gambaran mengenai analisis *system* yang akan dibangun pada prosedur kelola data pembayaran dapat dilihat pada uraian dibawah ini:



Gambar 3.11 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Pembayaran

3.1.2.11 Analisis sistem yang akan dibangun pada prosedur kelola data pengumuman

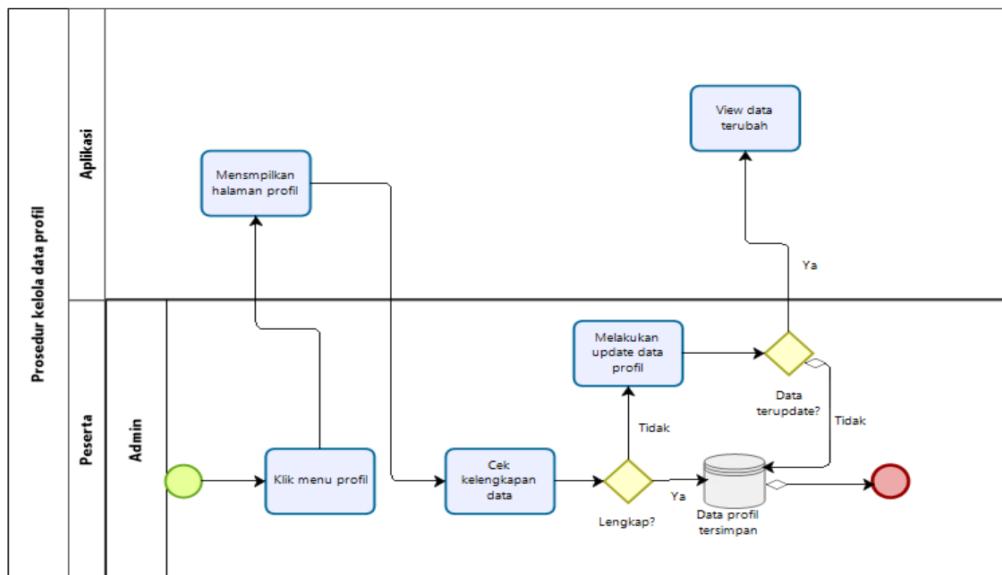
Pada proses kelola data pengumuman ini terdapat 2 entitas atau *actor* yang memiliki kewajiban masing masing diantaranya yakni admin dan aplikasi. Kewajiban admin dalam prosedur pengumuman ini ialah mengupdate hasil pengumuman yakni dari pendaftaran yang telah dilakukan peserta apakah lulus atau tidak lulus serta dapat mencetak hasil pengumuman. Kemudian aplikasi dapat melakukan penyimpanan data pengumuman yakni hasil pengumuman pendaftaran yang dilakukan yang dilakukan seperti: No pendaftaran, nama, NISN, asal sekolah, pilihan program kursus penerima, pengajar, jam dan hari. Untuk mendapatkan gambaran mengenai analisis prosedur *system* yang akan dibangun pada kelola data pengumuman dapat dilihat pada uraian dibawah ini:



Gambar 3.12 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Pengumuman

3.1.2.12 Analisis sistem yang akan dibangun pada prosedur kelola data profil

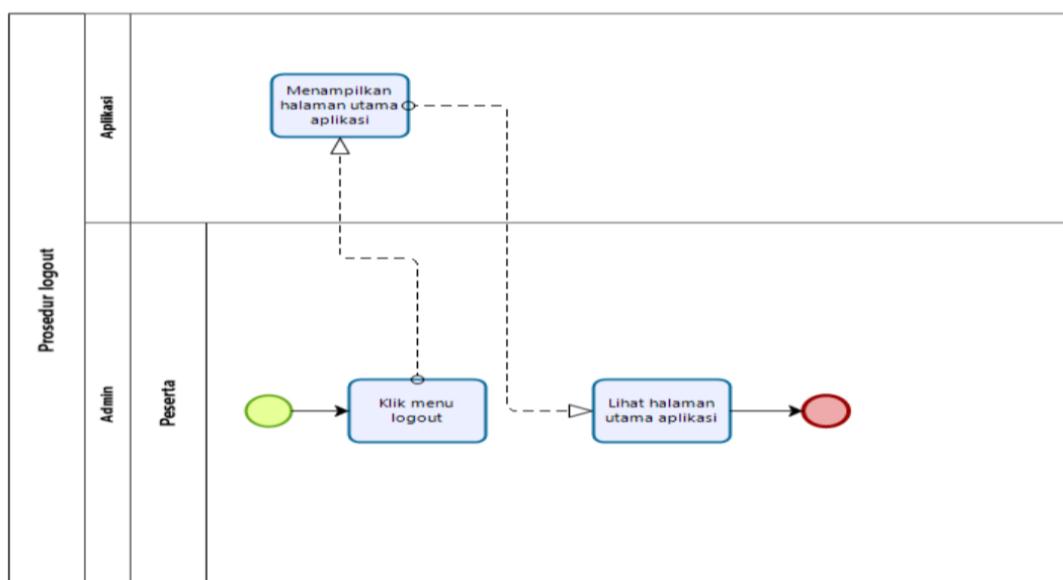
Pada proses kelola data profil ini terdapat 3 entitas atau *actor* yang memiliki kewajiban masing masing diantaranya yakni admin, peserta dan aplikasi. Kewajiban admin dan peserta disini adalah melakukan *update* data. Kemudian aplikasi dapat melakukan proses menyimpan data profil yang telah dilakukan dari proses tersebut kedalam *database*. Untuk mendapatkan gambaran mengenai analisis prosedur sistem yang akan dibangun pada kelola data profil dapat dilihat pada uraian dibawah ini:



Gambar 3.13 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur Kelola Data Profil

3.1.2.13 Analisis sistem yang akan dibangun pada prosedur *logout*

Pada proses atau prosedur *logout* terdapat 3 entitas atau *actor* yang memiliki kewajiban masing masing diantaranya yakni admin, peserta dan aplikasi. Kewajiban admin dan peserta dalam prosedur *logout* ini dengan melakukan klik *button logout*, maka secara langsung aplikasi akan memproses permintaan proses tersebut dan menampilkan halaman utama. Untuk mendapatkan gambaran mengenai analisis prosedur *logout* yang akan dibangun pada proses *logout* dapat dilihat pada uraian dibawah ini:



Gambar 3.14 Flowmap Analisis Sistem Yang Akan Dibangun Pada Prosedur *Logout*

3.1.3 Analisis Dokumen Yang Digunakan

Analisis dokumen merupakan sebuah bagian dari analisis data dimana terdapat sebuah kegiatan mengumpulkan data data yang berhubungan dengan kursus yang telah ada dan sistem yang akan dibangun. Analisis dokumen merupakan suatu cara yang akan berlanjut dalam membuat perancangan sistem selanjutnya. Adapun analisis dokumen yang terdapat di *programming course roblox* adalah sebagai berikut:

- 1.Dokumen pendaftaran kursus
- 2.Dokumen pengumuman kursus
- 3.Dokumen pembayaran kursus

Tabel 3.1 Dokumen Pendaftaran Kursus

No.	Dokumen Pendaftaran Kursus	Penjelasan
1.	Dibuat oleh	Admin
1.	Dibuat untuk	Peserta
2.	Isi	Data calon peserta kursus seperti data pribadi, data orang tua, data pendidikan, dan data pilihan kursus
3.	Tujuan	Sebagai data pelengkap dari calon peserta kursus yang melakukan pendaftaran

Tabel 3.2 Dokumen Pengumuman Kursus

No.	Dokumen Pengumuman Kursus	Penjelasan
1.	Dibuat oleh	Admin
2.	Dibuat untuk	Peserta
3.	Isi	Pengumuman hasil pendaftaran yang telah dilakukan oleh peserta kursus seperti: No pendaftaran, Nama, NISN, Asal sekolah, program kursus penerima, jam, hari, pengajar
4.	Tujuan	Menyediakan informasi mengenai informasi pengumuman jenis kursus yang telah dipilih oleh peserta kursus sehingga dapat dilihat informasinya lulus atau tidak lulus

Tabel 3.3 Dokumen Pembayaran Kursus

No.	Dokumen Bukti Pembayaran Kursus	Penjelasan
1.	Dibuat oleh	Admin
1.	Dibuat untuk	Peserta
2.	Isi	Bukti pembayaran
3.	Tujuan	Menyediakan bukti catatan mengenai pembayaran kursus yang dilakukan oleh peserta

3.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah tahap analisis yang dibuat berdasarkan kebutuhan yang berisi proses apa saja yang nantinya akan dilakukan untuk suatu sistem dapat dikembangkan selanjutnya. Sistem pendaftaran *programming course* roblox ini secara fungsional merupakan suatu kursus kecil kecilan yang baru dimulai khusus nya dibidang pendidikan. Adapun kebutuhan fungsional yang akan dibangun yaitu pengelolahan data terdiri dari 13 proses sesuai dengan urutannya sebagai berikut:

1. *Register* Masyarakat
2. *Login*
3. Pendaftaran
4. Pembayaran
5. Pengumuman
6. Kelola data pengguna
7. Kelola data sekolah
8. Kelola data program kursus
9. Kelola data pendaftaran
10. Kelola data pembayaran
11. Kelola data pengumuman
12. Kelola data profil admin dan peserta
13. *Logout*

Ke-13 proses diatas memiliki alur dari masing masing kegiatan yang dilakukan, dan tentunya saling terhubung dan berkaitan dengan *database* yang telah dibuat sebelumnya. Setiap proses yang ada diatas, saling berkaitan juga dengan tiap entitasnya baik peserta maupun admin.

3.1.5 Analisis Kebutuhan *Non Fungsional*

Analisis kebutuhan *non fungsional* adalah suatu proses dimana akan menggambarkan suatu perangkat lunak, analisis dan kebutuhan perangkat keras serta kebutuhan yang harus dipenuhi dalam perancangan perangkat lunak atau aplikasi yang akan dibangun. Pada analisis kebutuhan *non fungsional* ini berdasarkan kebutuhan pendukung, yaitu kebutuhan operasional seperti perangkat lunak, perangkat keras serta analisis pengguna. Adapun analisis kebutuhan fungsional yang akan dibuat adalah sebagai berikut:

3.1.5.1 Analisis Kebutuhan Perangkat Lunak

Analisis ini dibutuhkan agar mengetahui spesifikasi minimum yang dibutuhkan untuk menjalankan aplikasi sistem *programming course* roblox. Dibawah ini adalah analisis kebutuhan perangkat lunak.

Tabel 3.4 Spesifikasi Perangkat Lunak

No	<i>Tool/ Software</i>	Fungsi
1.	<i>Windows 10</i>	Standar operasi
2.	Xampp 1.7.3	<i>Server basis data</i>
3.	<i>Visual studio code</i>	<i>Software</i> membuat aplikasi
3.	Roblox studio	<i>Software</i> membuat <i>game</i> roblox
4.	Php	Bahasa pemrograman yang digunakan dalam merancang aplikasi atau <i>system</i>
5.	Php My Admin	<i>Software</i> pengolah data dari aplikasi yang dirancang
6.	Mysql	Membuat dan mengelola informasi yang ada pada <i>database</i>
7.	<i>Google Chrome/Mozilla firefox</i>	<i>Web browser</i>
8.	PDF	Dokumen yang digunakan untuk mencetak (<i>print</i>)

3.1.5.2 Analisis Kebutuhan Perangkat Keras

Diantara analisis kebutuhan perangkat lunak, terdapat beberapa spesifikasi minimum perangkat keras *mobile* yang dibutuhkan agar dapat menjalankan aplikasi sistem *programming course* roblox dapat dilihat pada tabel dibawah ini:

Tabel 3.5 Spesifikasi Perangkat Keras

No	Perangkat Keras	Disarankan	Alasan
1.	<i>Processor</i>	2005+ (1.6 Ghz)	Karena dalam membuat sebuah program aplikasi, <i>processor</i> ini akan membantu mengeksekusi program program yang ada dikomputer melalui instruksi
2.	<i>Memory</i>	4 GB	Karena membangun aplikasi membutuhkan penyimpanan yang cukup besar sehingga membutuhkan kapasitas minimal 4gb
3.	Kecepatan HDD & Penyimpanan HDD	5400 RPM & 4 GB	Karena agar terhindar dari kegagalan dalam menjalankan aplikasi yang dibuat tidak lambat dalam melakukan proses. Kecepatan yang diasarkan minimal 5400 RPM
4.	<i>Hardisk</i>	320 GB	Karena untuk menyimpan segala jenis <i>file</i> dan data dari aplikasi minimal sebesar 320 GB

3.1.6 Analisis Pengguna

Analisis kebutuhan pengguna pada aplikasi sistem pendaftaran *programming course* roblox memiliki 3 pengguna diantaranya adalah sebagai berikut:

Tabel 3.6 Analisis Kebutuhan Pengguna

No.	Pengguna	Hak	Pengguna	Hak
1.	Masyarakat	Masyarakat dapat mendaftar akun atau <i>register</i> dengan memasukkan nama, <i>email</i> , <i>username</i> dan <i>password</i>	Admin	Admin dapat menambahkan pengguna, menghapus pengguna, dan <i>update</i> data pengguna
2.	Peserta	Peserta dapat melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> yang sudah didaftarkan	Admin	Admin dapat melakukan <i>login</i> dengan memasukkan <i>username</i> dan <i>password</i> yang dimiliki

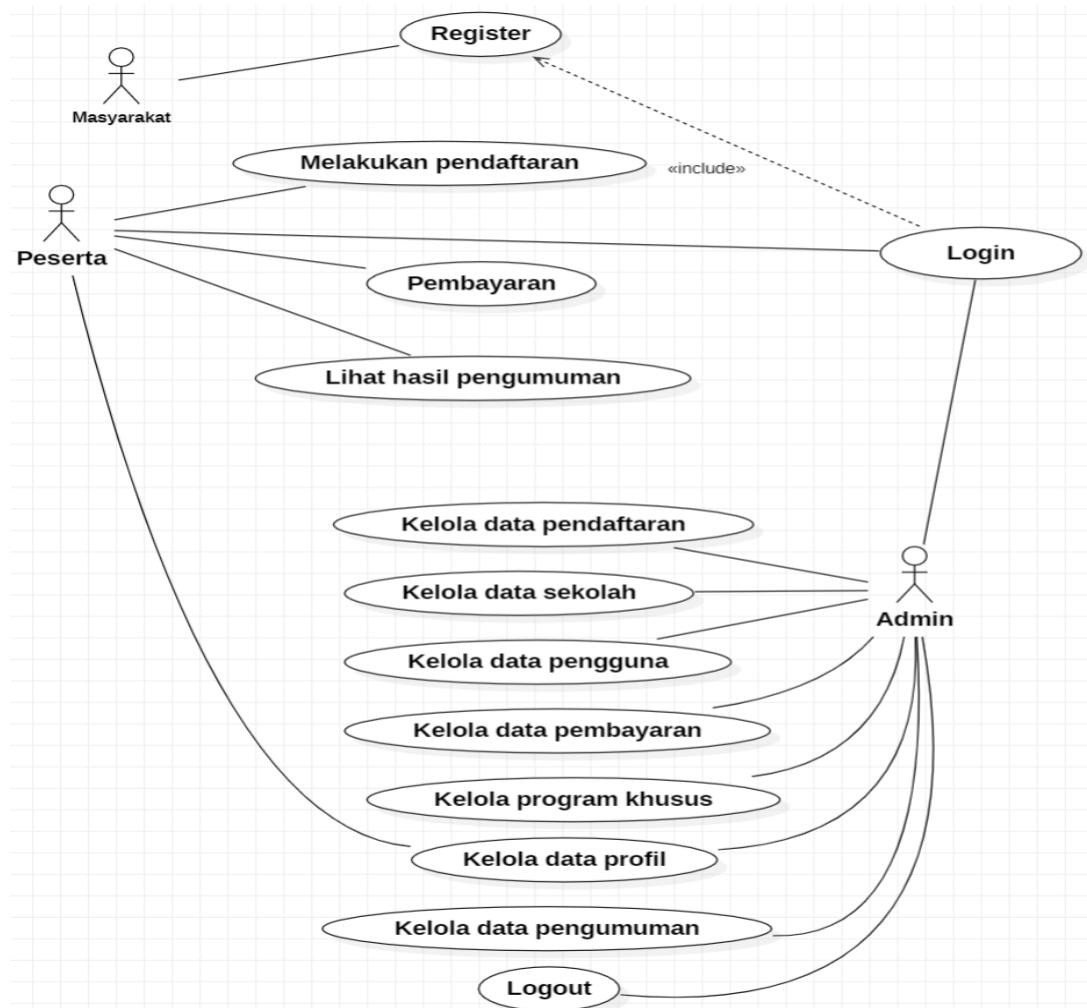
3.	Peserta	Peserta dapat melakukan pendaftaran dengan persyaratan yang diperlukan seperti memasukkan data pribadi, data sekolah, data orang tua, data program kursus dan data Pendidikan, serta peserta dapat mengecek status pendaftarannya	Admin	Admin dapat melihat kelengkapan data dari peserta, apakah sesuai dengan persyaratan atau tidak, dan admin juga memiliki hak untuk memasukkan maupun menghapus, dan mengubah data pendaftaran
4.	Peserta	Peserta dapat melihat informasi program kursus, jadwal, pengajar, dan harga yang telah tersedia	Admin	Admin dapat menambahkan daftar program program mengenai pilihan kursus pada sistem
5.	Peserta	Peserta dapat mengedit atau mengupdate data di menu profil peserta	Admin	Admin dapat mengupdate data di profil
6.	Peserta	Peserta dapat melakukan pembayaran dan menyertakan bukti pembayaran yang sah	Admin	Admin dapat mengecek dan memverifikasi status pembayaran yang telah dilakukan oleh peserta serta juga melakukan update
7.	Peserta	Peserta dapat melihat informasi mengenai hasil pengumuman dari pendaftaran yang dilakukan yang berisi pilihan program kursus penerima, dan jadwal kursus tersebut, serta dapat mencetak hasil pengumuman	Admin	Admin dapat update hasil pengumuman berdasarkan keseluruhan berkas pendaftaran yang telah dilengkapi oleh peserta, dan memverifikasi status kelulusan peserta hingga tahap mencetak hasil pengumuman
8.	Peserta	Peserta dapat keluar (<i>logout</i>) dari aplikasi ketika kegiatan di aplikasi sudah selesai	Admin	Admin dapat keluar (<i>logout</i>) dari aplikasi ketika kegiatan di aplikasi sudah selesai

3.2 Perancangan

3.2.1 UML (*Unified Modelling Language*)

3.2.1.1 Use Case Diagram

Use case diagram adalah salah satu jenis dari bagian kategori UML (*Unified Modelling Language*) yang menggambarkan hubungan komunikasi antara sistem dan aktor. *Use case* dapat mendeskripsikan mengenai hal hal apa saja yang akan dibangun oleh suatu *system*, dengan kata lain *use case diagram* ini merupakan gambaran proses proses. Simbol lingkaran pada gambar *use case diagram* tersebut mengenai proses *system*. *Use case diagram* menggambarkan hubungan dan interaksi antara *use case* dan *actor* dalam satu *diagram*. Berikut ini adalah proses proses yang digambarkan dalam *use case diagram*:



Gambar 3.15 *Use Case Diagram*

3.2.1.1.1 Definisi Aktor

Tabel 3.7 Definisi Aktor

No.	Actor	Deskripsi
1.	Masyarakat	➢ <i>Register</i>
2.	Peserta	➢ <i>Login</i> ➢ Melakukan pendaftaran ➢ Melakukan pembayaran ➢ Melihat hasil pengumuman ➢ Kelola Data Profil ➢ <i>Logout</i>
3.	Admin	➢ <i>Login</i> ➢ Kelola Data Pengguna ➢ Kelola Data Program Kursus ➢ Kelola Data Sekolah ➢ Kelola Data Pendaftaran ➢ Kelola Data Pembayaran ➢ Kelola Data Pengumuman ➢ Kelola Data Profil ➢ <i>Logout</i>

3.2.1.1.2 Definisi Use Case

Tabel 3.8 Definisi Use Case

No.	Actor	Deskripsi
1.	<i>Register</i>	Merupakan aktivitas yang dilakukan masyarakat sebelum masuk ke aplikasi dengan membuat akun aplikasi menggunakan <i>email</i> , <i>username</i> , dan <i>password</i>
2.	<i>Login</i>	Merupakan aktivitas <i>login</i> yang dilakukan oleh admin dan peserta. Sebelum admin dan peserta masuk ke aplikasi harus melakukan <i>login</i> terlebih dahulu dengan memasukkan <i>username</i> (<i>email</i>) dan <i>password</i>
3.	Melakukan Pendaftaran	Merupakan aktivitas melakukan pendaftaran yang dilakukan oleh peserta dengan mengisi persyaratan pendaftaran yang dibutuhkan
4.	Melakukan Pembayaran	Merupakan aktivitas melakukan pembayaran yang dilakukan oleh peserta dan setelah tahap melakukan pendaftaran, dengan cara mengupload bukti pembayaran
5.	Melakukan Pengumuman	Merupakan aktivitas melihat pengumuman, setelah proses pembayaran diselesaikan maka melanjutkan untuk tahap melihat hasil kelulusan.
6.	Kelola Data Pengguna	Merupakan aktivitas mengelola data pengguna yang dilakukan oleh admin. Data peserta yang telah mendaftar akun di aplikasi, dapat diolah oleh admin
7.	Kelola Data Sekolah	Merupakan aktivitas mengelola data sekolah yang dilakukan oleh admin. Data sekolah yang diolah admin merupakan salah satu persyaratan pendaftaran untuk peserta

9.	Kelola Data Pendaftaran	Merupakan aktivitas mengelola data pendaftaran yang dilakukan oleh admin. Peserta yang telah melakukan pendaftaran selanjutnya berkas dokumen pendaftarannya diolah oleh admin
10.	Kelola Data Pembayaran	Merupakan aktivitas mengelola data pembayaran yang dilakukan oleh admin. Setelah melakukan upload pembayaran oleh peserta, maka data berkas dokumen pendaftarannya diolah oleh admin
11.	Kelola Data Pengumuman	Merupakan aktivitas mengelola data pengumuman yang dilakukan oleh admin. Admin mengelola hasil kelulusan yakni di pengumuman setelah aktivitas pendaftaran dan pembayaran yang dilakukan oleh peserta
12.	Kelola Data Profil	Merupakan aktivitas mengelola data profil yang dilakukan oleh admin dan peserta
13.	<i>Logout</i>	Merupakan aktivitas keluar aplikasi atau <i>logout</i> yang dilakukan oleh admin dan peserta setelah melakukan pekerjaan di aplikasi

3.2.1.1.3 Skenario Use Case

Skenario *use case* adalah gambaran cara kerja dari masing masing proses yang terlibat didalam *system* atau aplikasi tersebut. Selain itu juga diberikan deksripsi yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang dilakukan oleh *actor*. Setiap *use case* yang ada didalam *system* aplikasi tersebut tentunya akan diberikan sebuah *scenario* yang akan menjelaskan secara detail interaksi yang ada di dalamnya. Berikut ini adalah *scenario* yang akan dijelaskan *step by step* dari tiap *use case* dalam aplikasi:

Tabel 3.9 Skenario Use Case Register

Identifikasi	
Nama	<i>Register</i>
Tujuan	Mendaftar akun
Aktor	Masyarakat
Kondisi awal	Tampilan <i>register</i> akun
Aksi aktor	Memasukkan data identitas seperti nama, <i>password</i> dan <i>email</i>
Aksi sistem	Menyimpan data akun masyarakat
Kondisi akhir	Masyarakat berhasil <i>register</i>

Tabel 3.10 Skenario Use Case Login

Identifikasi	
Nama	<i>Login</i>
Tujuan	Masuk ke aplikasi
Aktor	Peserta dan admin
Kondisi awal	Tampilan halaman <i>login</i>
Aksi aktor	Memasukkan <i>username, password</i> dan email
Aksi sistem	Berhasil <i>login</i>
Kondisi akhir	Tampilan halaman beranda

Tabel 3.11 Skenario Use Case Pendaftaran

Identifikasi	
Nama	Pendaftaran
Tujuan	Mendaftar ke aplikasi
Aktor	Peserta
Kondisi awal	Tampilan halaman pendaftaran
Aksi aktor	Memasukkan data berkas pendaftaran yang diperlukan seperti: data pribadi, data pendidikan, data pilihan program kursus dan data orang tua
Aksi sistem	Menyimpan data pendaftaran peserta
Kondisi akhir	Berhasil melakukan pendaftaran

Tabel 3.12 Skenario Use Case Pembayaran

Identifikasi	
Nama	Pembayaran
Tujuan	Sebagai salah satu syarat menyelesaikan pendaftaran
Aktor	Peserta
Kondisi awal	Tampilan halaman <i>upload</i> bukti pembayaran
Aksi aktor	Memasukkan bukti pembayaran

Aksi sistem	Menyimpan data pembayaran peserta
Kondisi akhir	Data Tersimpan

Tabel 3.13 Skenario *Use Case* Pengumuman

Identifikasi	
Nama	Pengumuman
Tujuan	Melihat hasil pengumuman setelah menyelesaikan pendaftaran
Aktor	Peserta
Kondisi awal	Tampilan halaman pengumuman
Aksi aktor	Melihat hasil pengumuman dan mencetak hasil pengumuman
Aksi sistem	Mencetak dokumen dalam bentuk pdf
Kondisi akhir	Berhasil mencetak dokumen

Tabel 3.14 Skenario *Use Case* Kelola Data Pengguna

Identifikasi	
Nama	Kelola data pengguna
Tujuan	Mengelola data pengguna
Aktor	Admin
Kondisi awal	Tampilan halaman data pengguna
Aksi aktor	Melakukan <i>insert, update, read, dan delete</i> data
Aksi sistem	Menyimpan data pengguna ke <i>database</i>
Kondisi akhir	Berhasil menyimpan data

Tabel 3.15 Skenario *Use Case* Kelola Data Sekolah

Identifikasi	
Nama	Kelola data sekolah
Tujuan	Sebagai syarat dalam pelengkap berkas pendaftaran peserta
Aktor	Admin
Kondisi awal	Tampilan halaman data sekolah
Aksi aktor	Melakukan <i>insert, update, read, dan delete</i> data
Aksi sistem	Menyimpan data sekolah ke <i>database</i>
Kondisi akhir	Berhasil menyimpan data

Tabel 3.16 Skenario *Use Case* Kelola Data Program Kursus

Identifikasi	
Nama	Kelola data program kursus
Tujuan	Sebagai syarat dalam pelengkap berkas pendaftaran peserta
Aktor	Admin
Kondisi awal	Tampilan halaman data program kursus
Aksi aktor	Melakukan <i>insert, update, read, dan delete</i> data
Aksi sistem	Menyimpan data program kursus ke <i>database</i>
Kondisi akhir	Berhasil menyimpan data

Tabel 3.17 Skenario *Use Case* Kelola Data Pendaftaran

Identifikasi	
Nama	Kelola data pendaftaran
Tujuan	Mengelola berkas pendaftaran peserta
Aktor	Admin
Kondisi awal	Tampilan halaman data pendaftaran
Aksi aktor	Melakukan <i>insert, update, read, dan delete</i> data
Aksi sistem	Menyimpan data pendaftaran ke <i>database</i>
Kondisi akhir	Berhasil menyimpan data

Tabel 3.18 Skenario *Use Case* Kelola Data Pembayaran

Identifikasi	
Nama	Kelola data pembayaran
Tujuan	Memverifikasi status pembayaran peserta
Aktor	Admin
Kondisi awal	Tampilan halaman data pembayaran
Aksi aktor	Melakukan <i>update</i> data
Aksi sistem	Menyimpan data berkas dokumen bukti pembayaran ke <i>database</i>
Kondisi akhir	Berhasil menyimpan data

Tabel 3.19 Skenario *Use Case* Kelola Data Pengumuman

Identifikasi	
Nama	Kelola data pengumuman

Tujuan	Memverifikasi status kelulusan peserta antara lulus atau tidak lulus
Aktor	Admin
Kondisi awal	Tampilan halaman data pengumuman
Aksi aktor	Melakukan <i>update</i> data
Aksi sistem	Menyimpan data pengumuman ke <i>database</i>
Kondisi akhir	Berhasil menyimpan data

Tabel 3.20 Skenario *Use Case* Kelola Data Profil

Identifikasi	
Nama	Kelola data profil
Tujuan	Mengelola data profil
Aktor	Admin dan peserta
Kondisi awal	Tampilan halaman profil
Aksi aktor	Melakukan <i>update</i> data
Aksi sistem	Menyimpan data profil ke <i>database</i>
Kondisi akhir	Berhasil menyimpan data

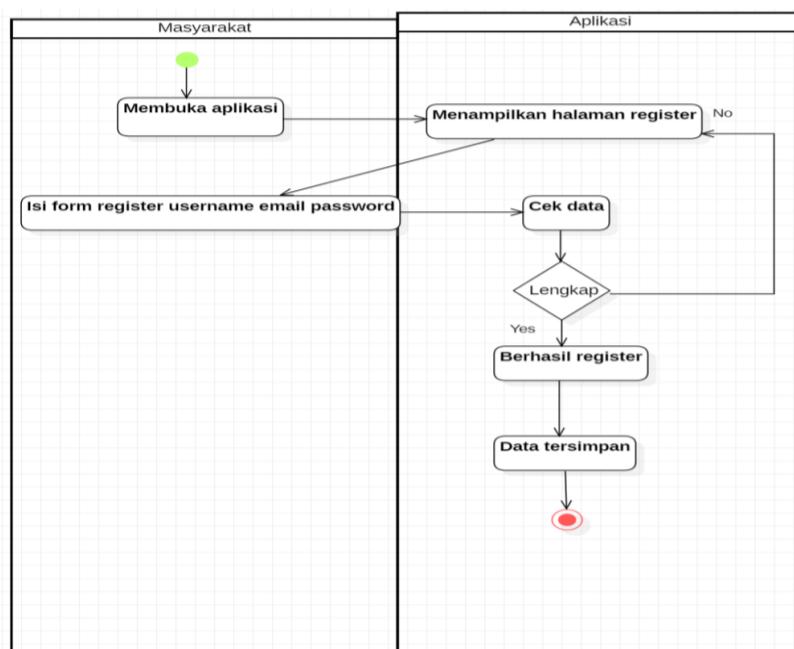
Tabel 3.21 Skenario *Use Case Logout*

Identifikasi	
Nama	<i>Logout</i>
Tujuan	Keluar dari aplikasi
Aktor	Admin dan peserta
Kondisi awal	Tampilan halaman beranda
Aksi aktor	Menekan tombol <i>logout</i>
Aksi sistem	Melakukan proses keluar atau <i>logout</i> dari aplikasi
Kondisi akhir	Tampilan halaman utama

3.2.1.2 Activity Diagram

Activity diagram merupakan bagian dalam UML (*Unified Modelling Language*) yang dapat didefinisikan aktivitas yang menggambarkan tentang berbagai proses atau kegiatan yang berjalan pada suatu sistem. *Activity diagram* adalah suatu urutan proses atau langkah-langkah yang digunakan untuk mendeskripsikan proses bisnis serta aliran kerja aplikasi. Berikut ini adalah proses proses *activity diagram* dalam perancangan aplikasi:

3.2.1.2.1 Activity Diagram Register

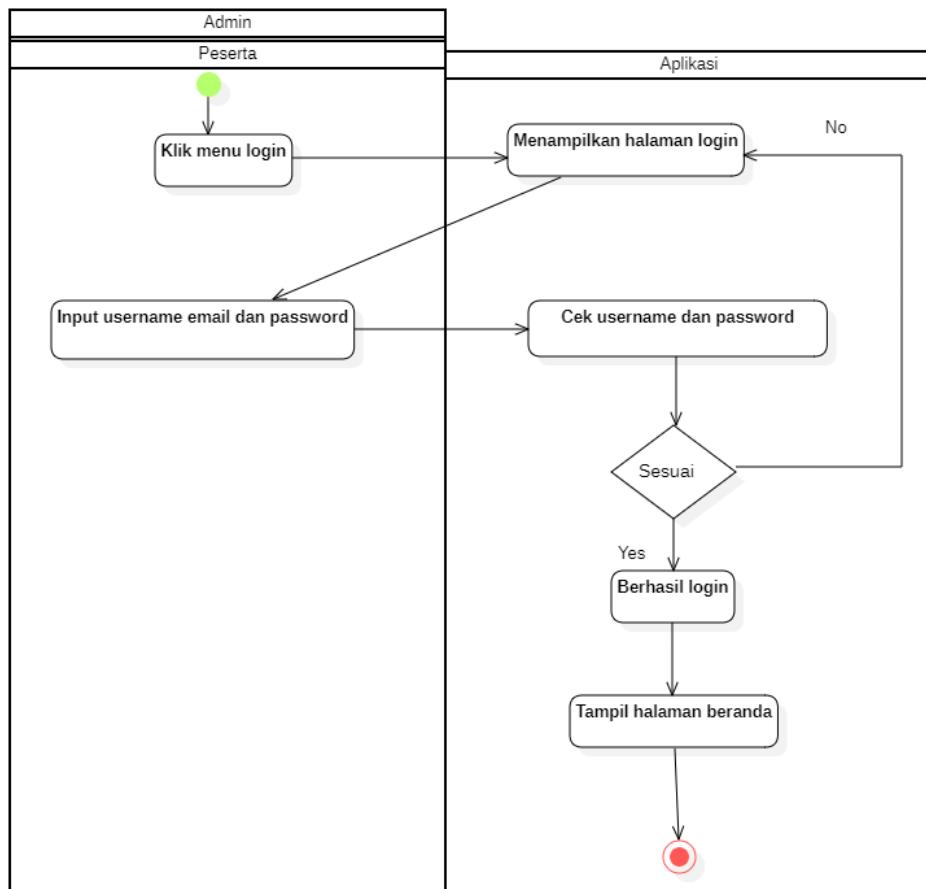


Gambar 3.16 Activity Diagram Register

Keterangan:

- 1) Masyarakat membuka aplikasi.
- 2) Aplikasi menampilkan halaman *register*.
- 3) Masyarakat *input form register* username email dan password.
- 4) Aplikasi mengecek data.
- 5) Apabila lengkap maka berhasil *register* dan menyimpan data.
- 6) Aplikasi menampilkan halaman gagal atau kembali jika peserta datanya tidak lengkap dan tidak berhasil *register*

3.2.1.2.2 Activity Diagram Login

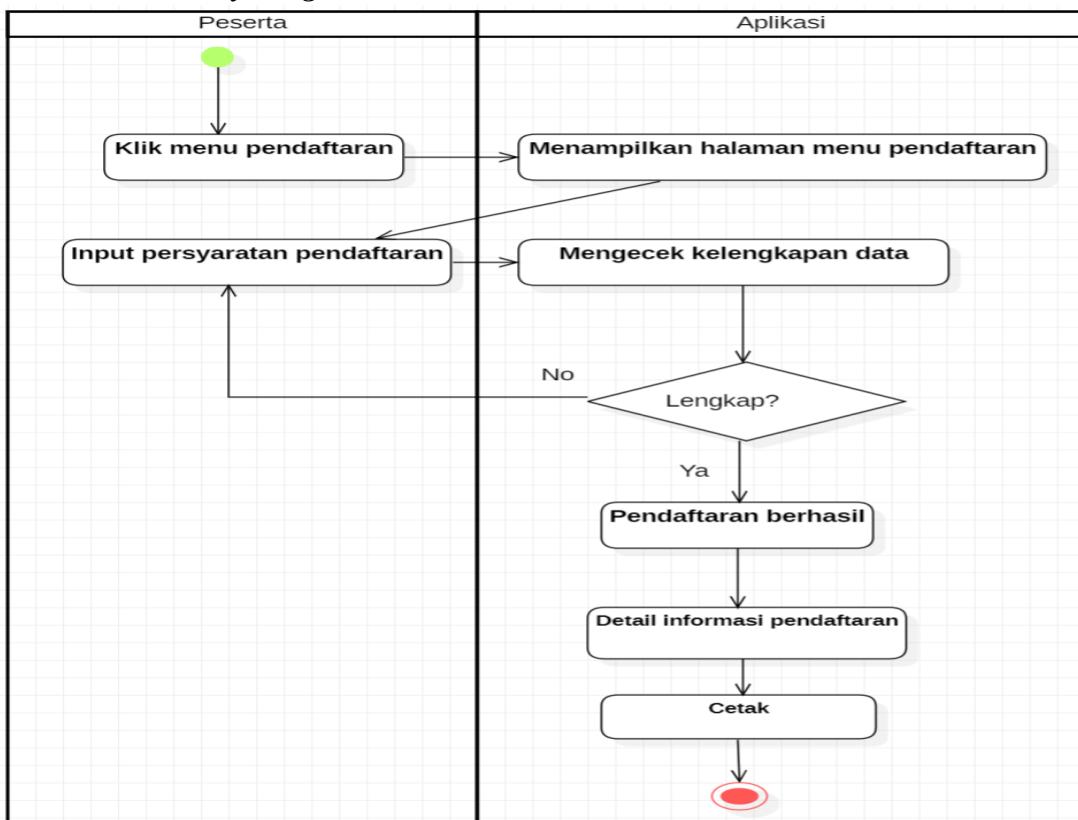


Gambar 3.17 Activity Diagram Login

Keterangan:

- 1) Peserta membuka aplikasi.
- 2) Aplikasi menampilkan halaman *login*.
- 3) Peserta memasukkan email, *username*, dan *password*.
- 4) Aplikasi mengecek *username*, email dan *password* yang dimasukkan jika sesuai maka berhasil *login* dan menampilkan halaman beranda.
- 5) Jika tidak berhasil aplikasi akan menampilkan kembali halaman *login*.

3.2.1.2.3 Activity Diagram Pendaftaran

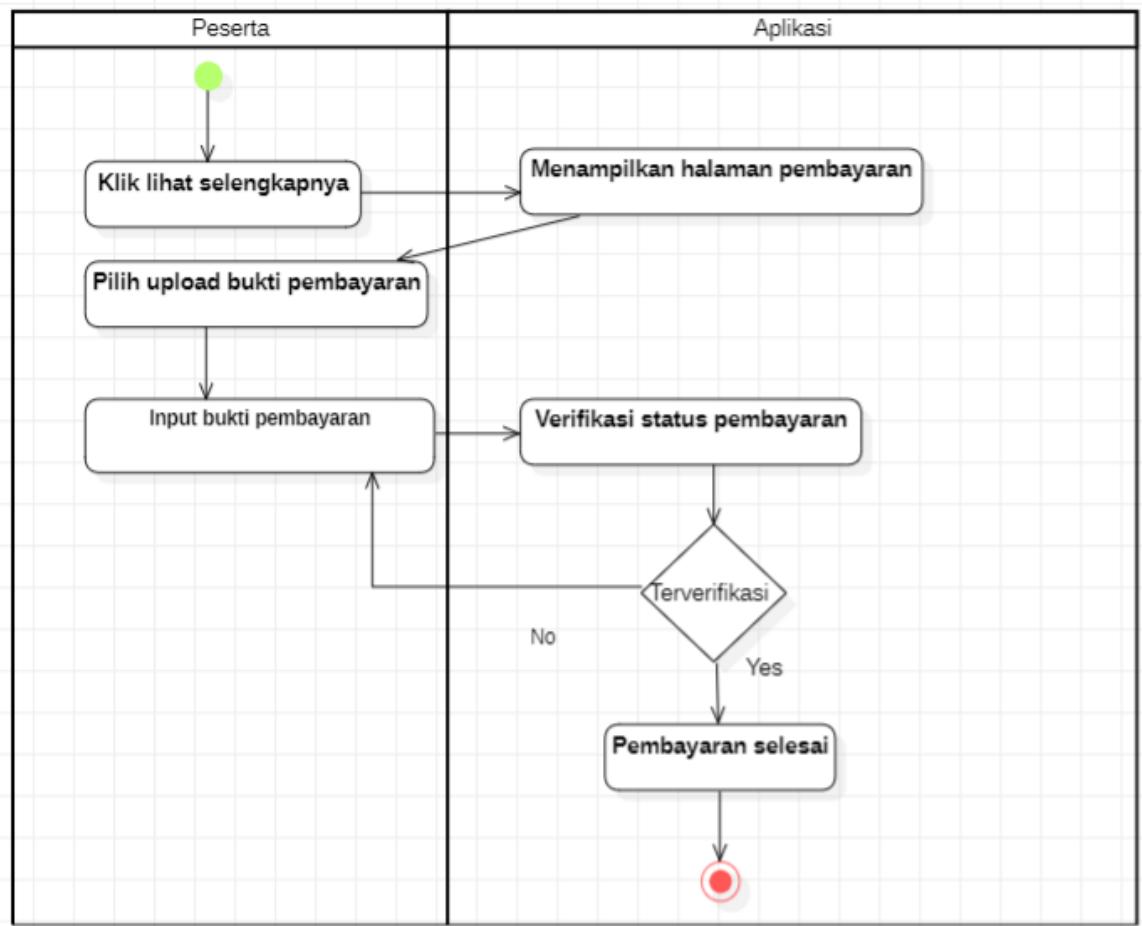


Gambar 3.18 Activity Diagram Pendaftaran

Keterangan :

- 1) Peserta mengklik menu pendaftaran.
- 2) Aplikasi menampilkan halaman menu pendaftaran.
- 3) Peserta memasukkan dan mengisi persyaratan pendaftaran.
- 4) Aplikasi memverifikasi status pendaftaran, jika sudah lengkap maka pendaftaran berhasil.
- 5) Jika tidak berhasil peserta harus memasukkan dan mengisi persyaratan pendaftaran kembali.
- 6) Peserta melihat informasi detail pendaftaran jika sudah berhasil mendaftar.

3.2.1.2.4 Activity Diagram Pembayaran

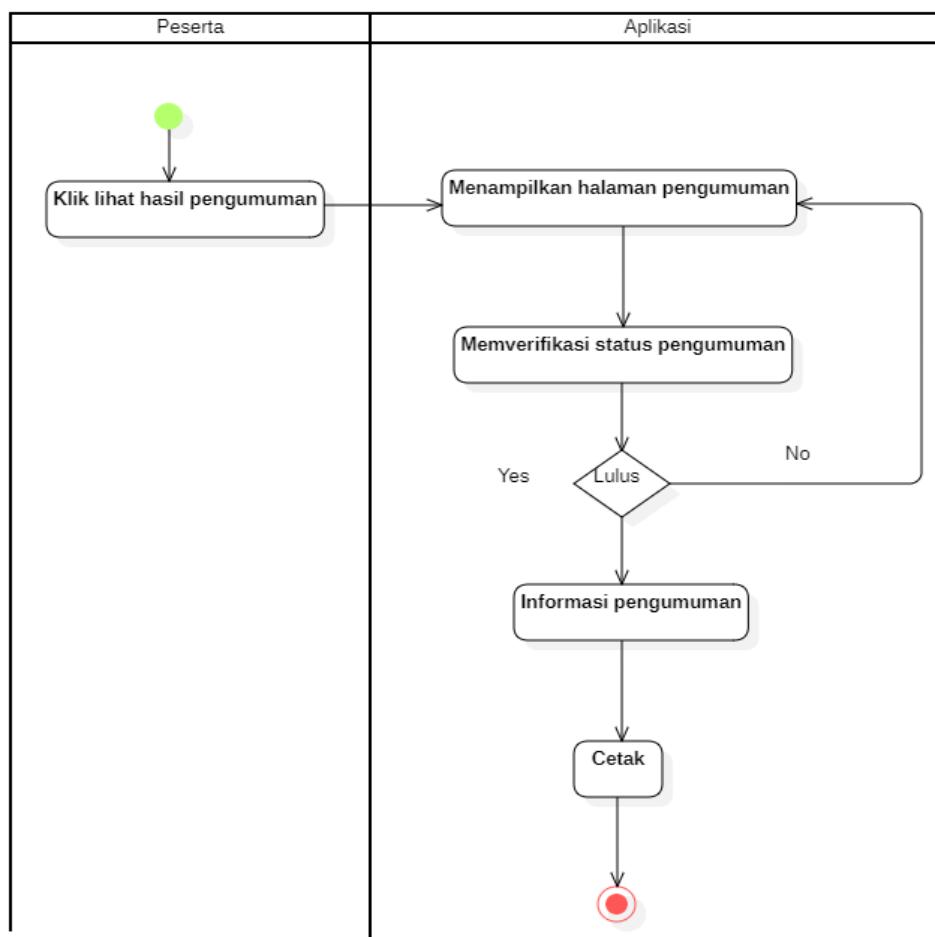


Gambar 3.19 Activity Diagram Pembayaran

Keterangan :

- 1) Peserta mengklik *button* pembayaran.
- 2) Aplikasi menampilkan halaman pembayaran.
- 3) Peserta memasukkan bukti pembayaran yang telah dilakukan.
- 4) Aplikasi memverifikasi status pembayaran.
- 5) Jika pembayaran terverifikasi maka peserta dapat melihat status pembayaran.

3.2.1.2.5 Activity Diagram Pengumuman

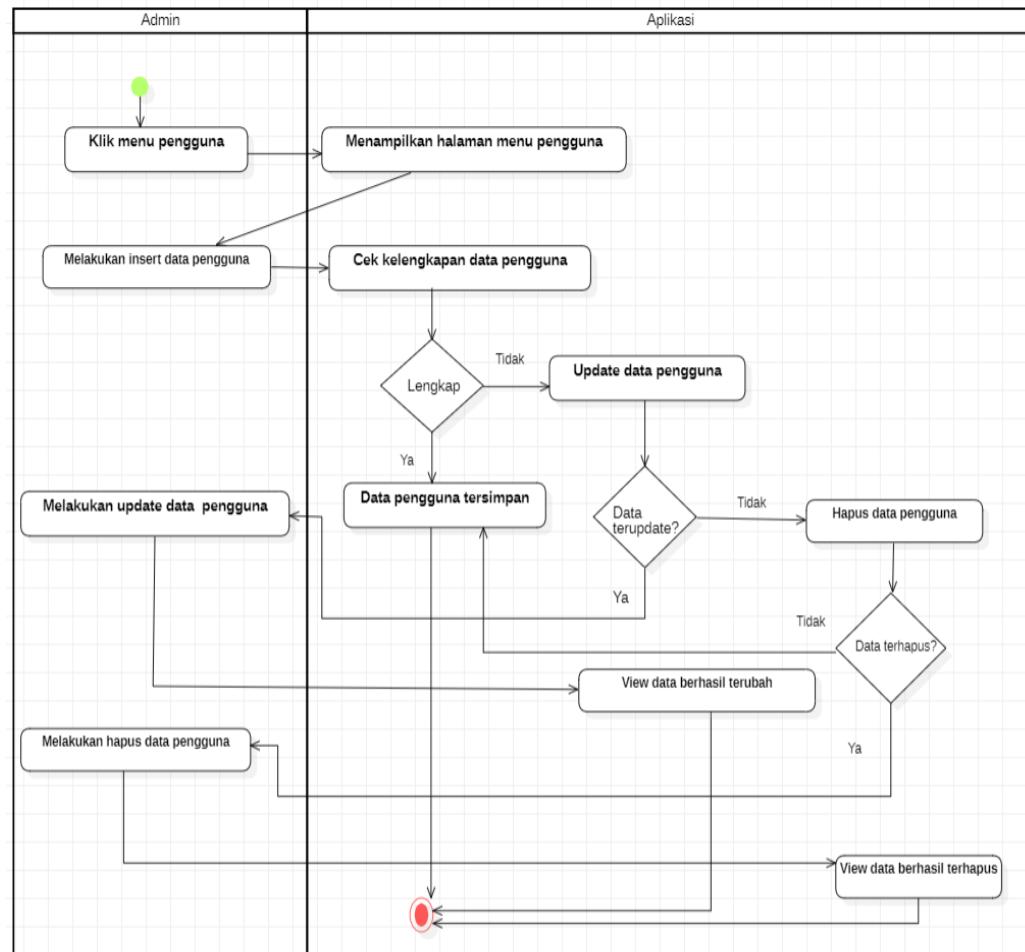


Gambar 3.20 Activity Diagram Pengumuman

Keterangan :

- 1) Peserta mengklik menu pengumuman.
- 2) Aplikasi akan menampilkan halaman pengumuman.
- 3) Peserta melihat informasi pengumuman.
- 4) Aplikasi memverifikasikan status kelulusan.
- 5) Jika lulus maka data pengumuman secara otomatis tersimpan.
- 6) Peserta dapat mengecek informasi kelulusan setelah diverifikasi kelulusan.
- 7) Peserta mencetak hasil pengumuman.
- 8) Jika tidak lulus maka peserta dapat mengklik menu pengumuman kembali.

3.2.1.2.6 Activity Diagram Kelola Data Pengguna

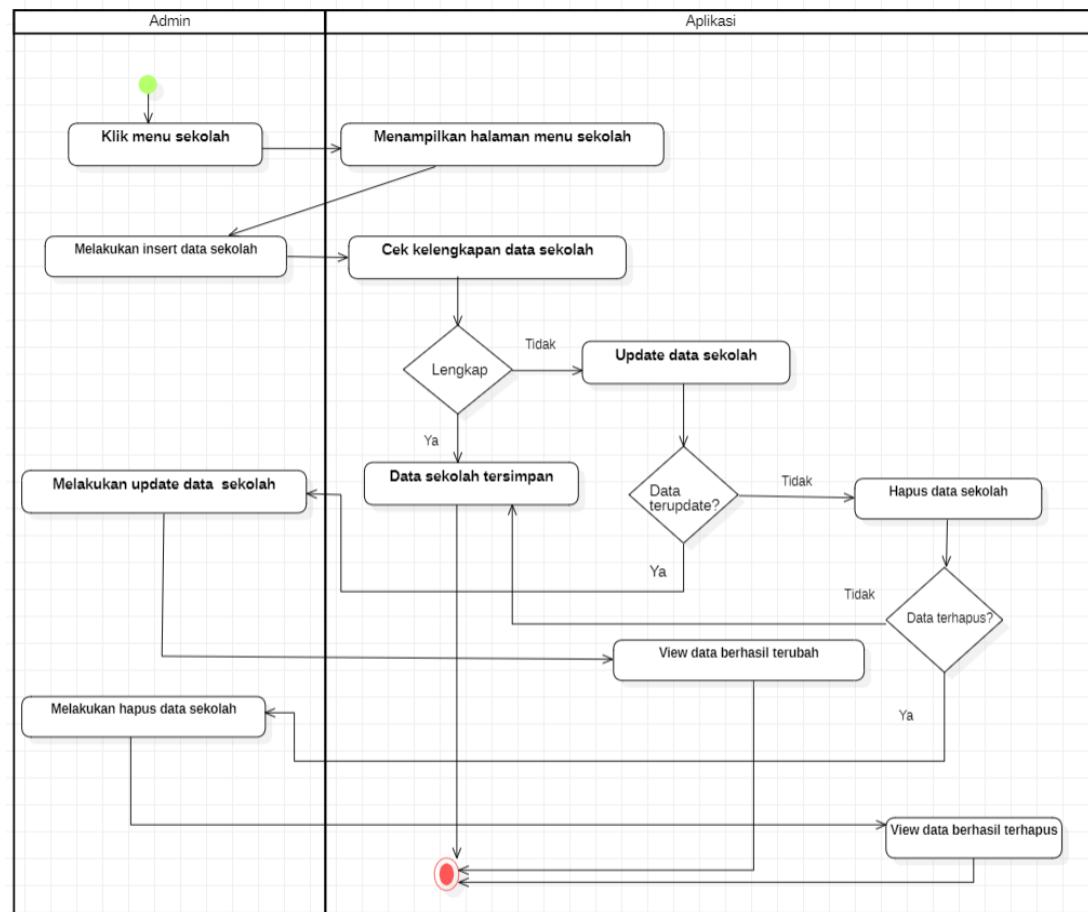


Gambar 3.21 Activity Diagram Kelola Data Pengguna

Keterangan :

- 1) Admin mengklik menu pengguna.
- 2) Aplikasi menampilkan halaman menu pengguna.
- 3) Admin melakukan *insert* data.
- 4) Aplikasi mengecek kelengkapan data pengguna.
- 5) Jika data lengkap maka data pengguna akan tersimpan.
- 6) Tetapi apabila data pengguna tidak lengkap, maka admin harus mengupdate data tersebut.
- 7) Admin melakukan hapus data pengguna.
- 8) Aplikasi menyimpan data pengguna ke dalam *database*.

3.2.1.2.7 Activity Diagram Kelola Data Sekolah

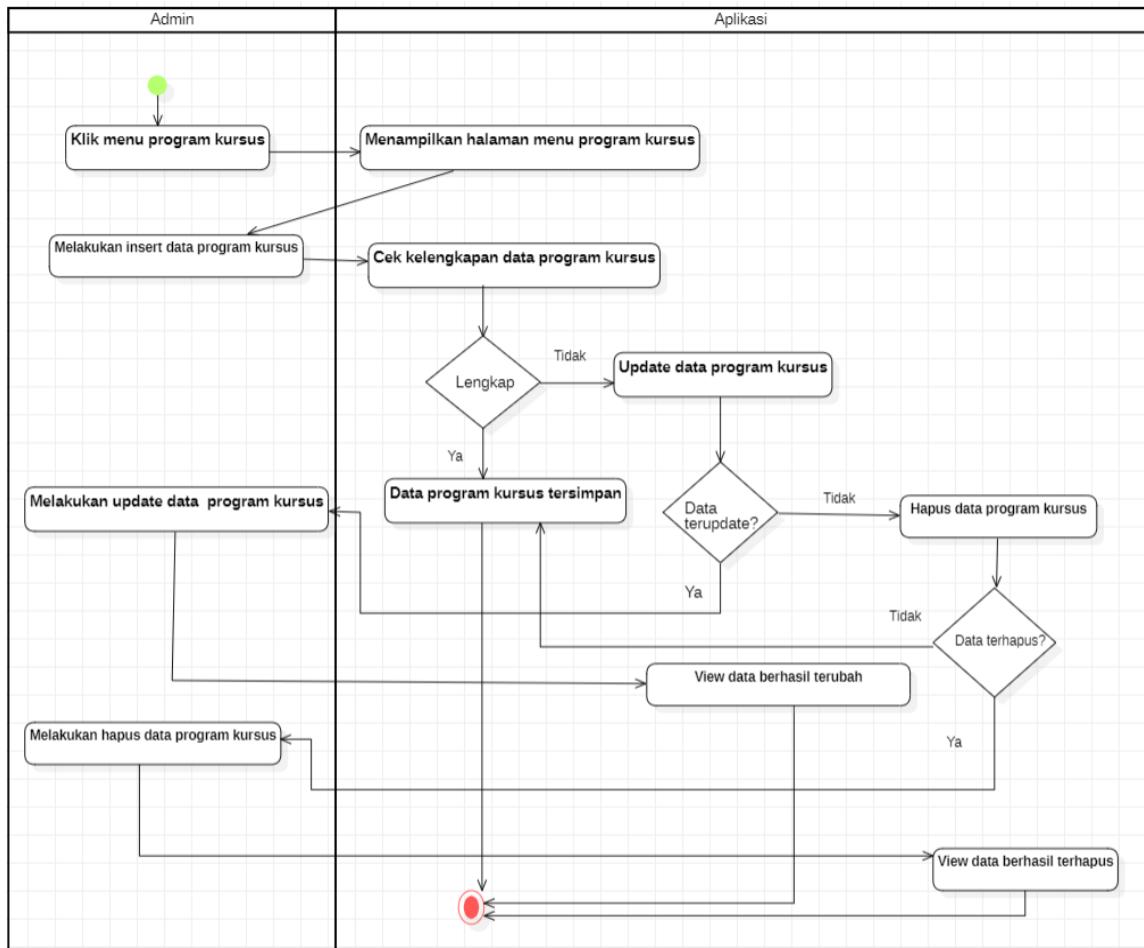


Gambar 3.22 Activity Diagram Kelola Data Sekolah

Keterangan :

- 1) Admin mengklik menu sekolah.
- 2) Aplikasi menampilkan halaman menu sekolah.
- 3) Admin melakukan *insert* data.
- 4) Aplikasi mengecek kelengkapan data sekolah.
- 5) Jika data lengkap maka data sekolah akan tersimpan.
- 6) Tetapi apabila data sekolah tidak lengkap, maka admin harus mengupdate data tersebut.
- 7) Admin melakukan hapus data sekolah.
- 8) Aplikasi menyimpan data sekolah ke dalam *database*.

3.2.1.2.8 Activity Diagram Kelola Data Program Kursus

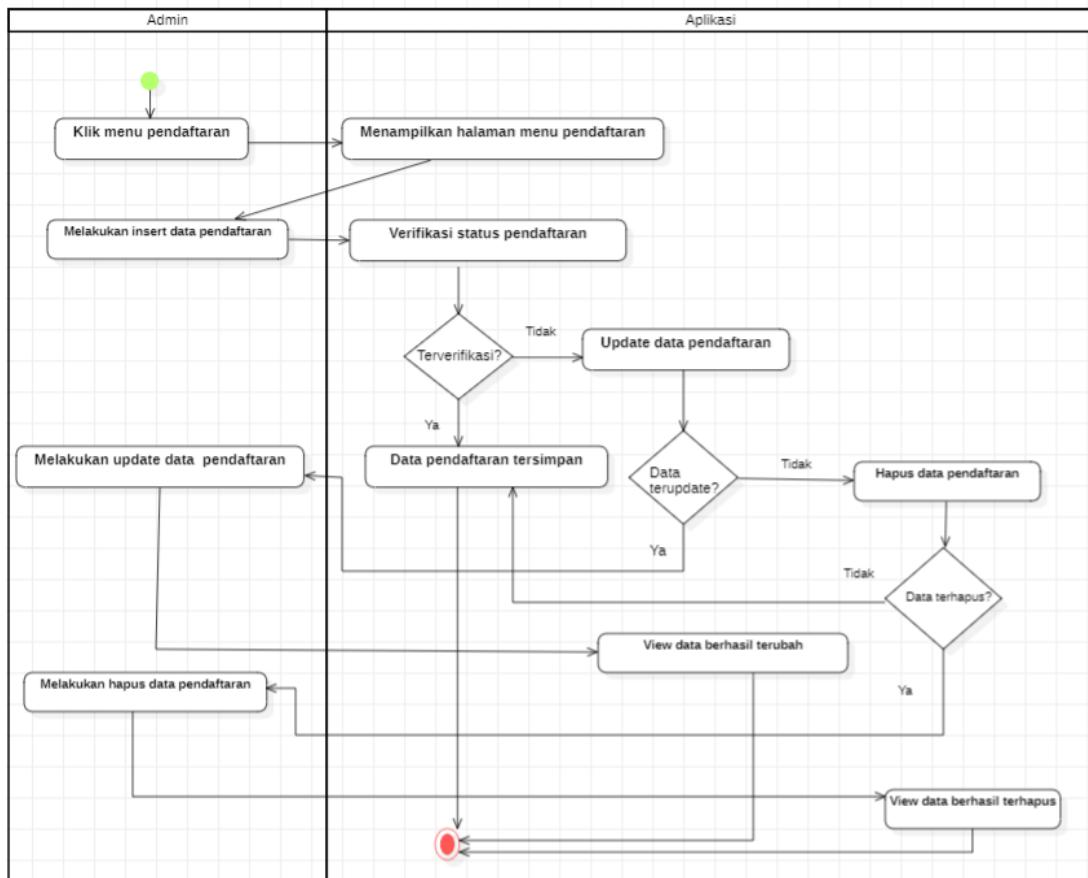


Gambar 3.23 Activity Diagram Kelola Data Program Kursus

Keterangan :

- 1) Admin mengklik menu program kursus.
- 2) Aplikasi menampilkan halaman menu program kursus.
- 3) Admin melakukan *insert* data.
- 4) Aplikasi mengecek kelengkapan data program kursus.
- 5) Jika data lengkap maka data program kursus akan tersimpan.
- 6) Tetapi apabila data pengguna tidak lengkap, maka admin harus mengupdate data tersebut.
- 7) Admin melakukan hapus data program kursus.
- 8) Aplikasi menyimpan data program kursus ke dalam *database*.

3.2.1.2.9 Activity Diagram Kelola Data Pendaftaran

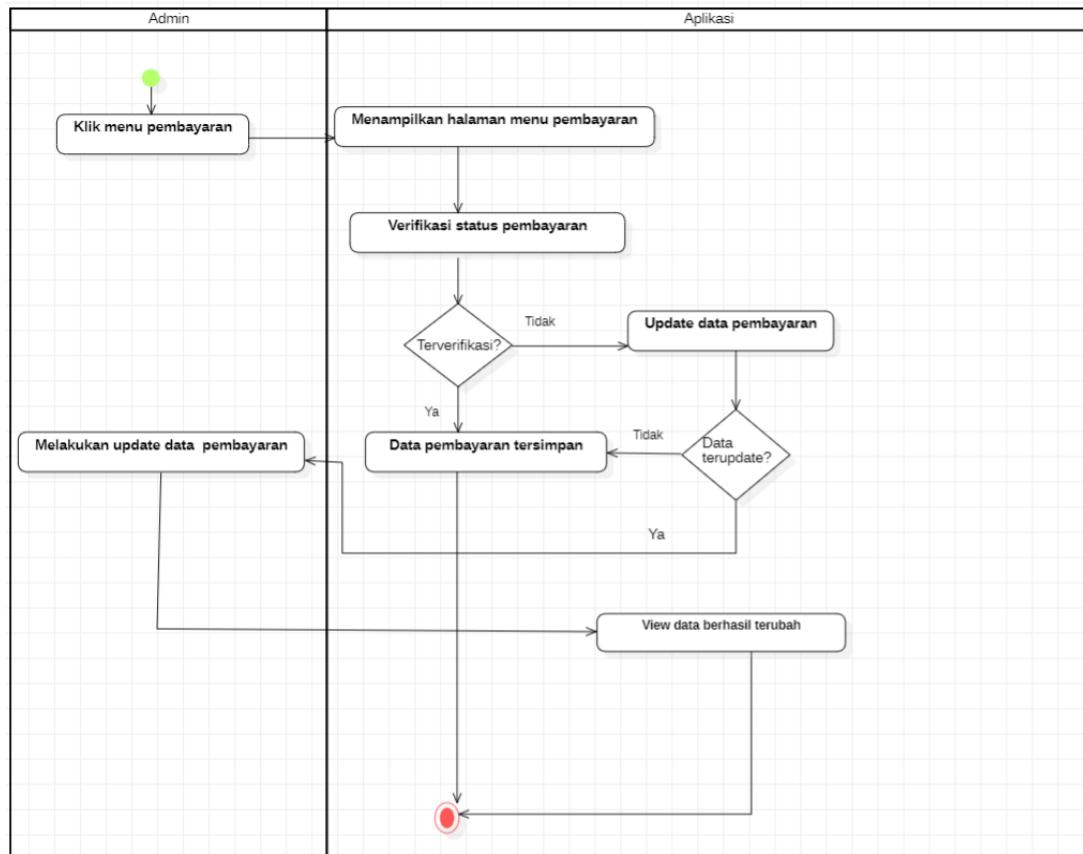


Gambar 3.24 Activity Diagram Kelola Data Pendaftaran

Keterangan :

- 1) Admin mengklik menu pendaftaran.
- 2) Aplikasi menampilkan halaman menu pendaftaran.
- 3) Admin melakukan *insert* data.
- 4) Aplikasi mengecek memverifikasi data pendaftaran.
- 5) Jika data terverifikasi maka data pendaftaran akan tersimpan.
- 6) Tetapi apabila data pendaftaran tidak terverifikasi, maka admin harus mengupdate data tersebut lalu aplikasi menampilkan “data berhasil diupdate”.
- 7) Admin melakukan hapus data pendaftaran.
- 8) Aplikasi menampilkan “data berhasil terhapus”.

3.2.1.2.10 Activity Diagram Kelola Data Pembayaran

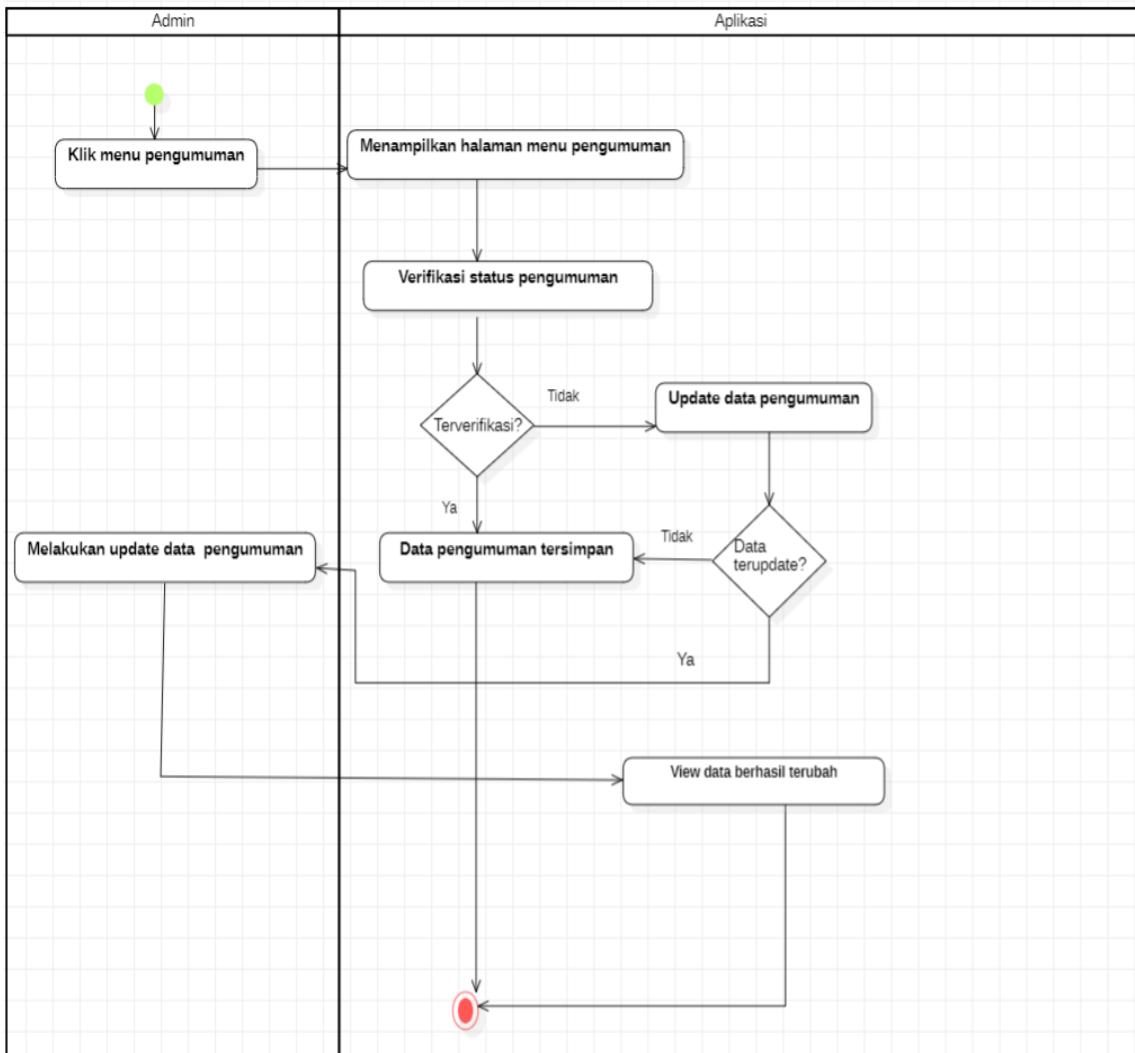


Gambar 3.25 Activity Diagram Kelola Data Pembayaran

Keterangan :

- 1) Admin mengklik menu pembayaran.
- 2) Aplikasi menampilkan halaman data pembayaran.
- 3) Aplikasi memverifikasi data pembayaran.
- 4) Apabila terverifikasi maka data tersimpan.
- 5) Apabila tidak maka *update* data pembayaran.
- 6) Apabila *true* atau setuju data *diupdate* maka admin melakukan *update* data pembayaran kemudian aplikasi menampilkan “data berhasil *diupdate*”.
- 7) Apabila tidak, maka aplikasi menyimpan *update* data pembayaran ke dalam *database*.

3.2.1.2.11 Activity Diagram Kelola Data Pengumuman

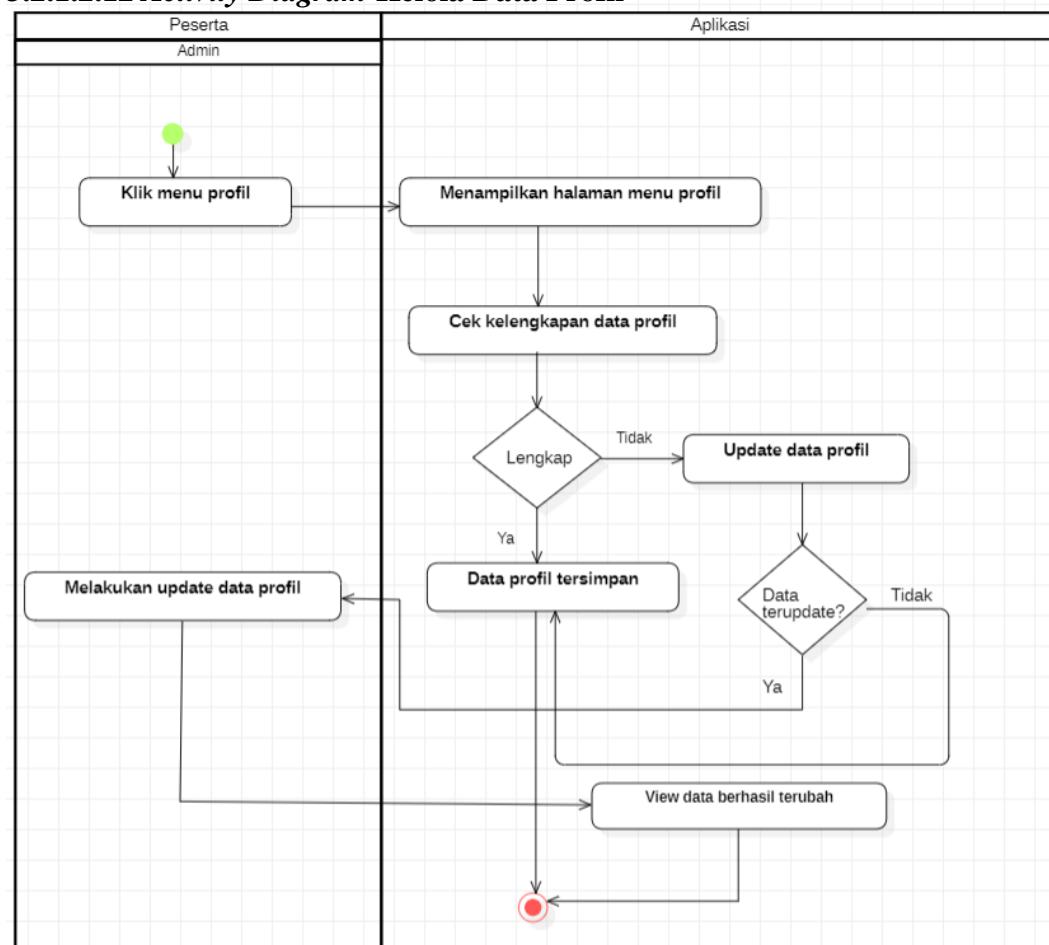


Gambar 3.26 Activity Diagram Kelola Data Pengumuman

Keterangan :

- 1) Admin mengklik menu pengumuman.
- 2) Aplikasi menampilkan halaman data pengumuman.
- 3) Mengecek verifikasi data pengumuman.
- 4) Apabila terverifikasi maka data pengumuman tersimpan.
- 5) Apabila *true* maka admin melakukan *update* data dan aplikasi menampilkan “data berhasil diupdate”.
- 6) Apabila tidak terupdate, maka data akan tersimpan.

3.2.1.2.12 Activity Diagram Kelola Data Profil

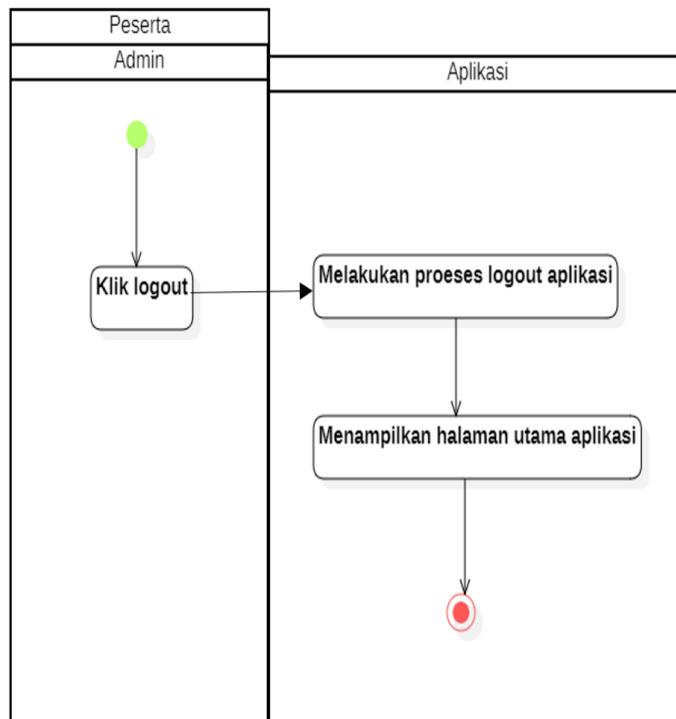


Gambar 3.27 Activity Diagram Kelola Data Profil

Keterangan :

- 1) Admin dan peserta mengklik menu profil.
- 2) Aplikasi menampilkan halaman profil.
- 3) Mengecek kelengkapan data profil.
- 4) Apabila lengkap data profil tersimpan.
- 5) Apabila tidak lengkap admin atau peserta melakukan *update* data profil
- 6) Admin melakukan *update* data profil dan menampilkan “data berhasil terupdate”.
- 7) Apabila tidak maka data tersimpan ke *database*.

3.2.1.2.13 Activity Diagram Logout



Gambar 3.28 Activity Diagram Logout

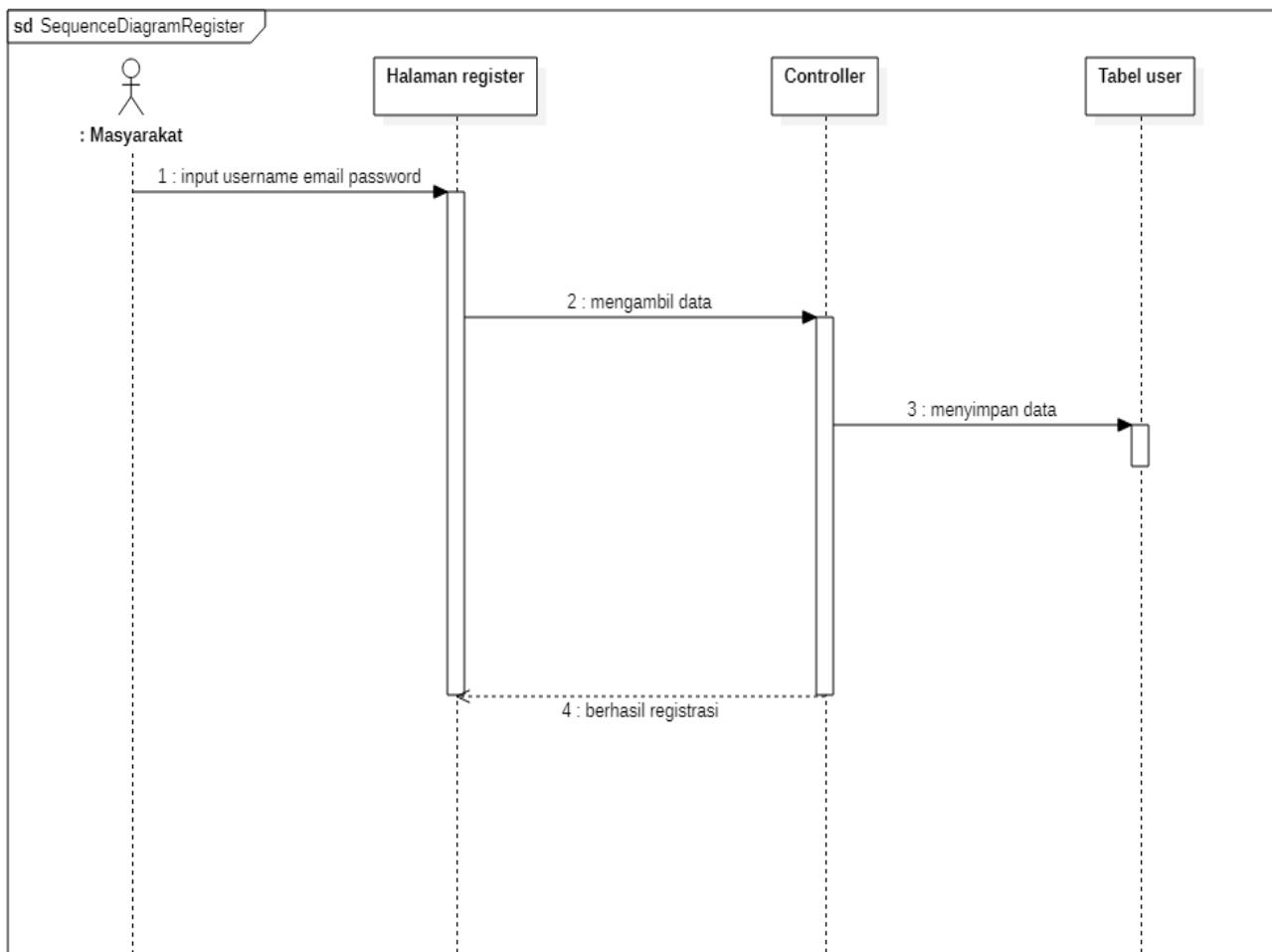
Keterangan :

- 1) Admin atau peserta mengklik menu *logout*.
- 2) Aplikasi melakukan proses *logout* aplikasi.
- 3) Aplikasi menampilkan halaman utama aplikasi.

3.2.1.3 Sequence Diagram

Sequence diagram adalah salah satu bagian dalam UML (*Unified Language Modelling*) yang mendeskripsikan serta menjelaskan suatu interaksi objek yang berdasarkan urutan waktu. Interaksi tersebut diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. *Sequence diagram* dapat digunakan untuk menggambarkan mengenai proses atau prosedur yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram*. Berikut ini adalah proses proses *sequence diagram* dapat dilihat pada gambar dibawah ini:

3.2.1.3.1 Sequence Diagram Register

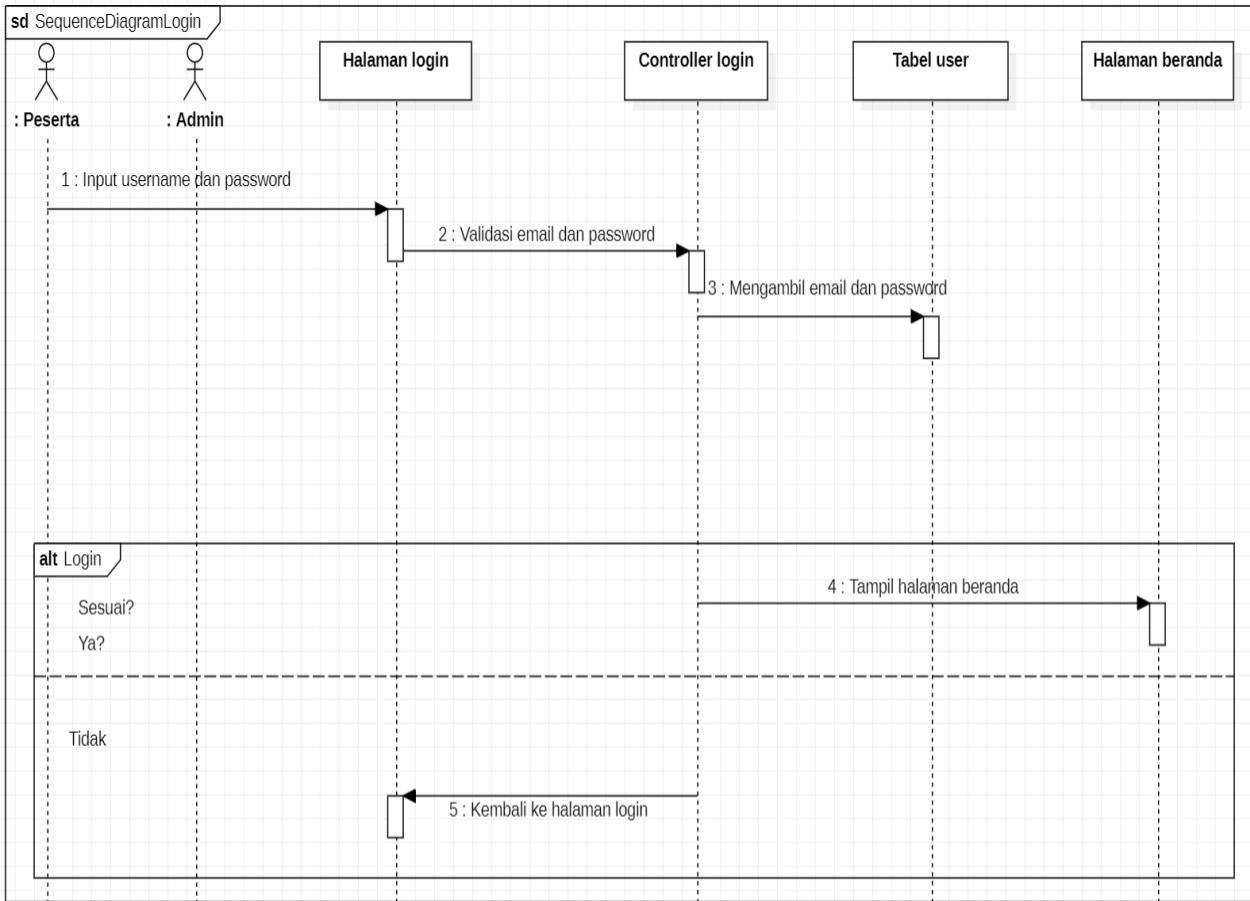


Gambar 3.29 Sequence Diagram Register

Keterangan:

- 1.) Masyarakat melakukan *register* dengan memasukkan *username* atau email, dan *password* nya.
- 2.) Kemudian dilanjutkan ke halaman *login* dengan mengambil data tersebut dan menampilkan halaman *login*.
- 3.) Dari halaman *login* mengambil data, kemudian *controller* mengambil data tersebut dan menyimpan data *register* kedalam *database* atau tabel *user* yang terbentuk.
- 4.) Dari *controller* ke *user interface outputnya* berupa berhasil *register*.

3.2.1.3.2 Sequence Diagram Login

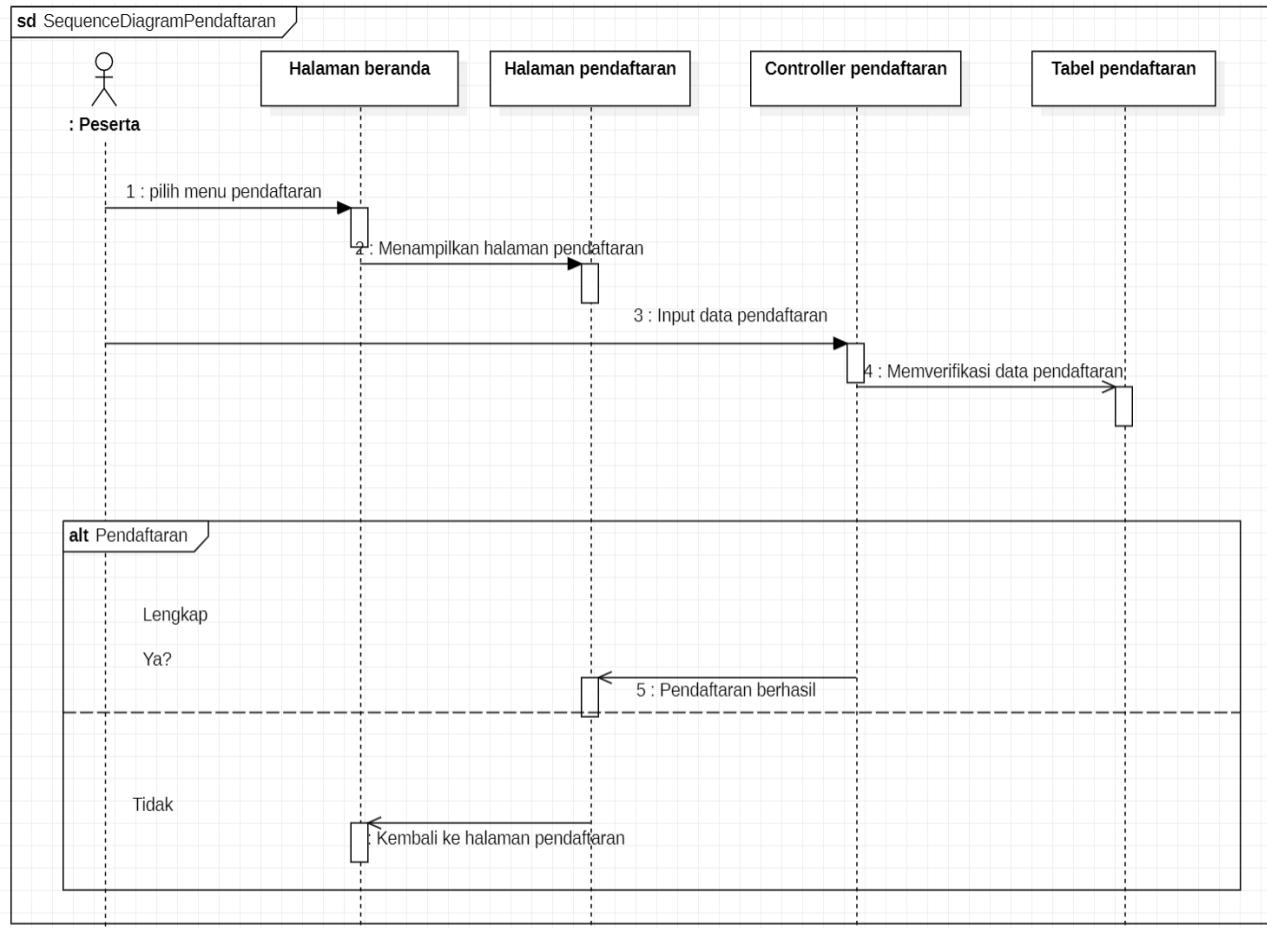


Gambar 3.30 Sequence Diagram Login

Keterangan:

- 1.) Peserta melakukan *login* dengan memasukkan *username* atau email, dan *password* nya.
- 2.) Kemudian dilanjutkan validasi *username* atau email dan *password* diteruskan ke *controller login*.
- 3.) *Controller login* mengambil data ke *database* data peserta untuk validasi *login*.
- 4.) Jika *username* dan *password* valid pada data, admin maka akan menampilkan halaman beranda.
- 5.) Jika *username* dan *password* tidak valid pada data maka akan ditampilkan halaman *login* kembali.

3.2.1.3.3 Sequence Diagram Pendaftaran

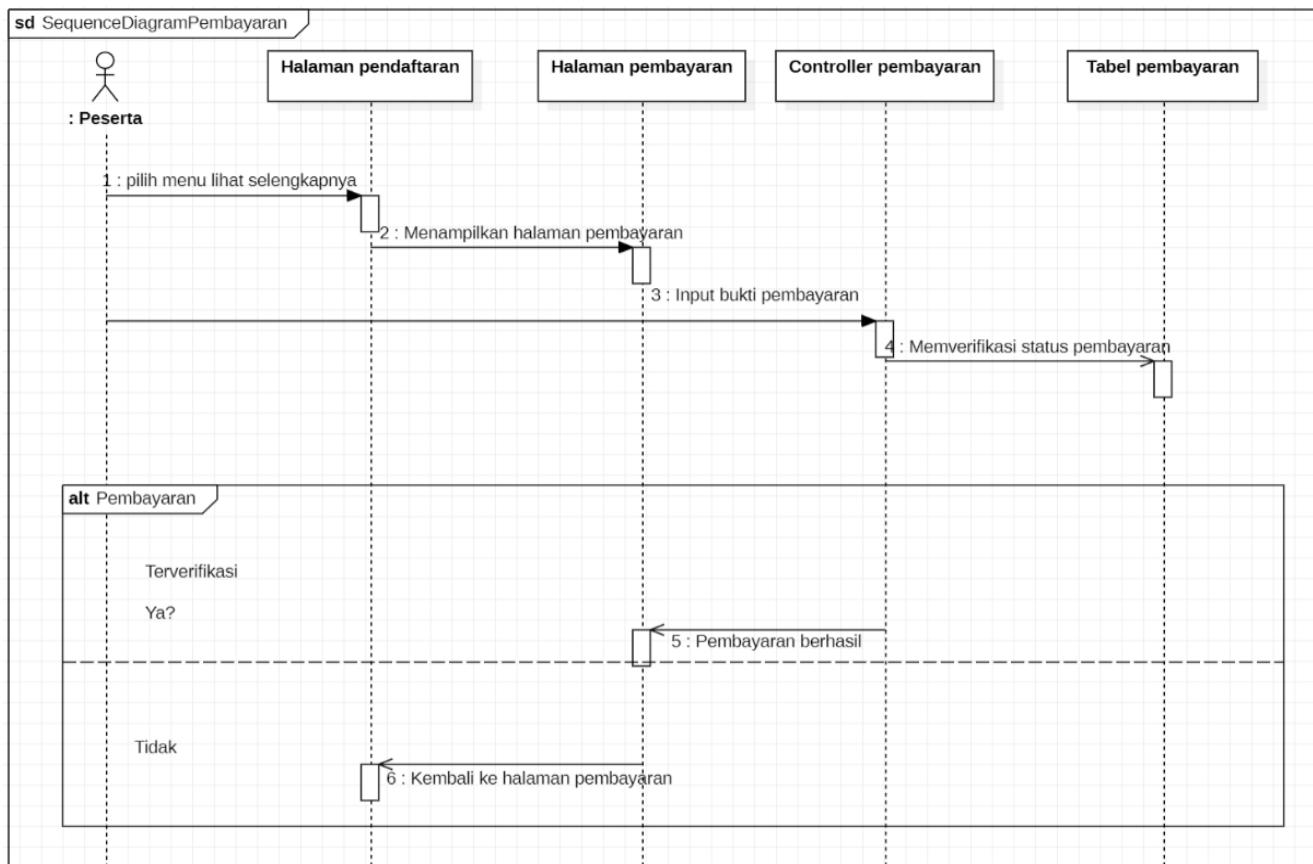


Gambar 3.31 Sequence Diagram Pendaftaran

Keterangan:

- 1.) Peserta memilih menu pendaftaran.
- 2.) Aplikasi menampilkan halaman pendaftaran.
- 3.) Kemudian dilanjutkan peserta memasukkan pendaftaran yang diperlukan.
- 4.) Dari *controller* memverifikasi data pendaftaran.
- 5.) Apabila data lengkap, maka pendaftaran berhasil.
- 6.) Apabila tidak lengkap maka kembali ke halaman pendaftaran.

3.2.1.3.4 Sequence Diagram Pembayaran

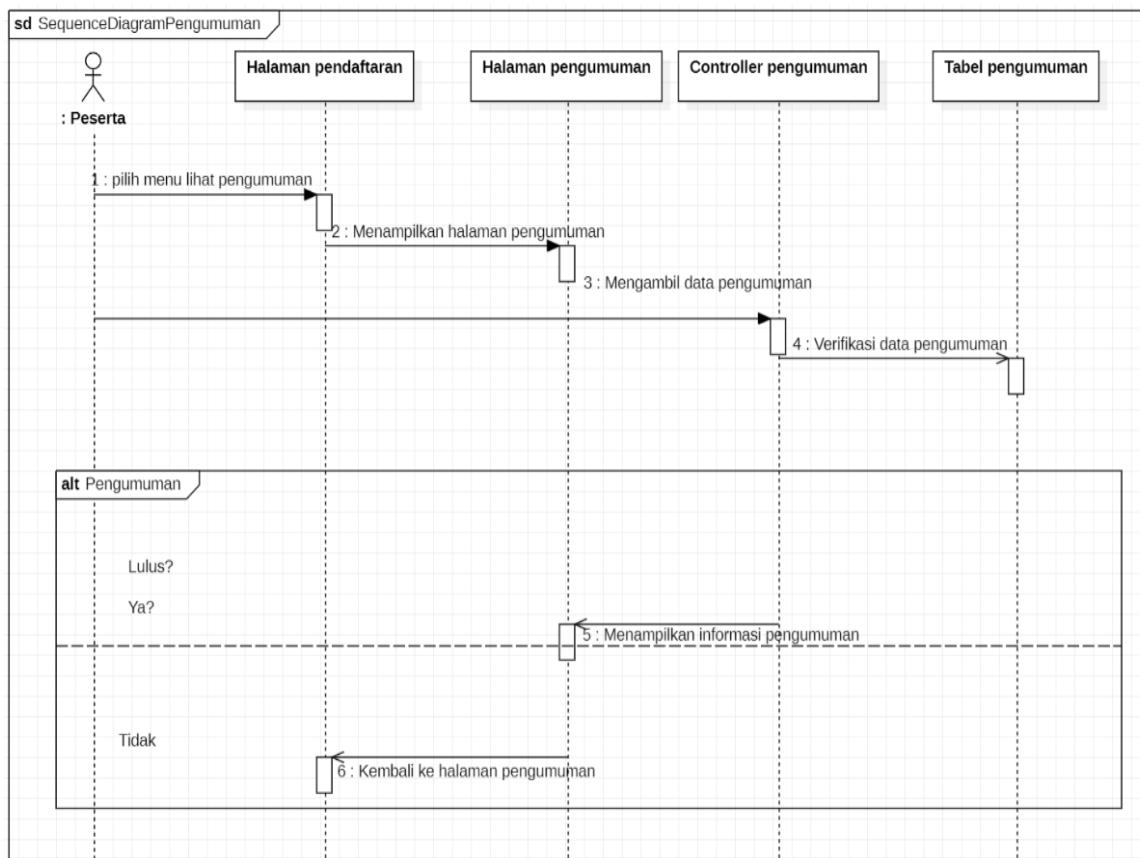


Gambar 3.32 Sequence Diagram Pembayaran

Keterangan:

- 1.) Peserta memilih menu lihat selengkapnya.
- 2.) *User interface* menampilkan halaman pembayaran.
- 3.) Kemudian dilanjutkan memasukkan bukti pembayaran.
- 4.) *Controller* memverifikasi status pembayaran.
- 5.) Apabila terverifikasi maka pembayaran berhasil.
- 6.) Apabila tidak terverifikasi maka kembali ke halaman pembayaran.

3.2.1.3.5 Sequence Diagram Pengumuman

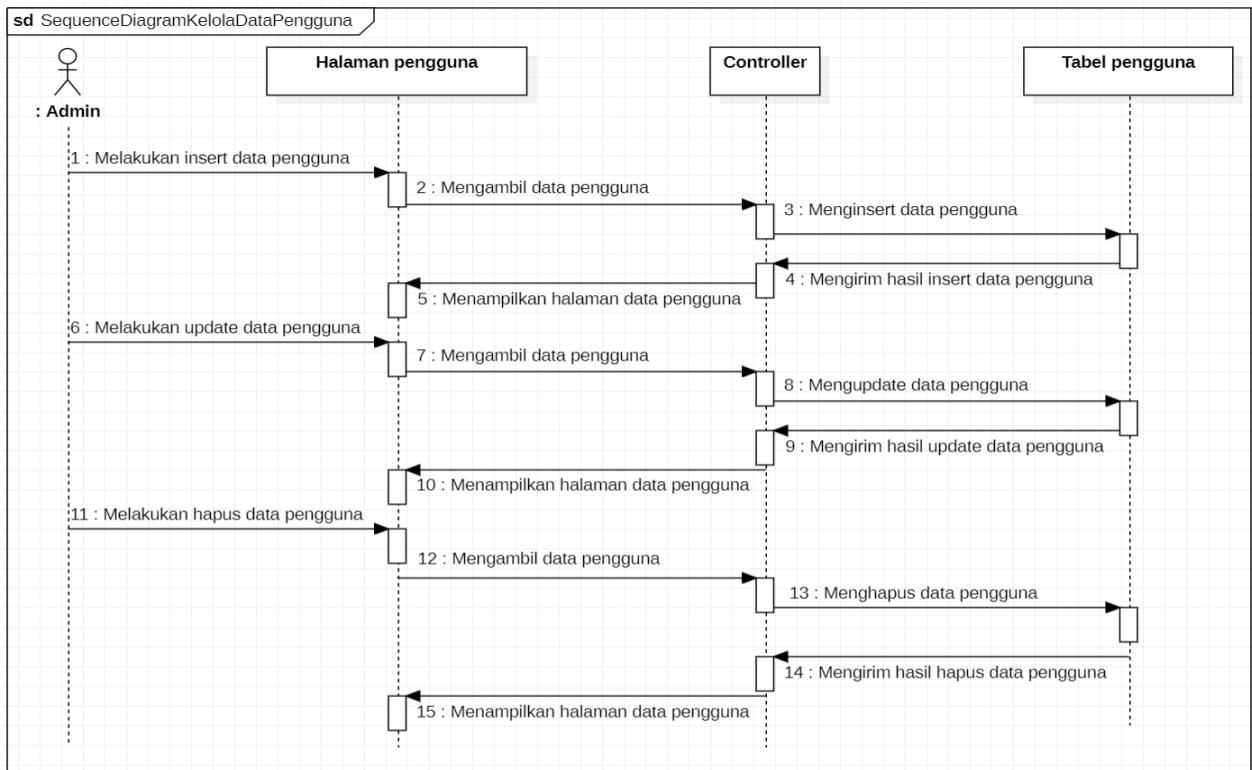


Gambar 3.33 Sequence Diagram Pengumuman

Keterangan:

- 1.) Peserta memilih menu lihat hasil pengumuman.
- 2.) *User interface* menampilkan halaman pengumuman.
- 3.) *Controller* mengambil data pengumuman.
- 4.) Dari *controller* memverifikasi data pengumuman kemudian mencocokkan dengan *database*.
- 5.) Apabila lulus, *user interface* akan menampilkan informasi pengumuman.
- 6.) Namun, apabila tidak lulus *user interface* akan kembali ke menu pengumuman.

3.2.1.3.6 Sequence Diagram Kelola Data Pengguna



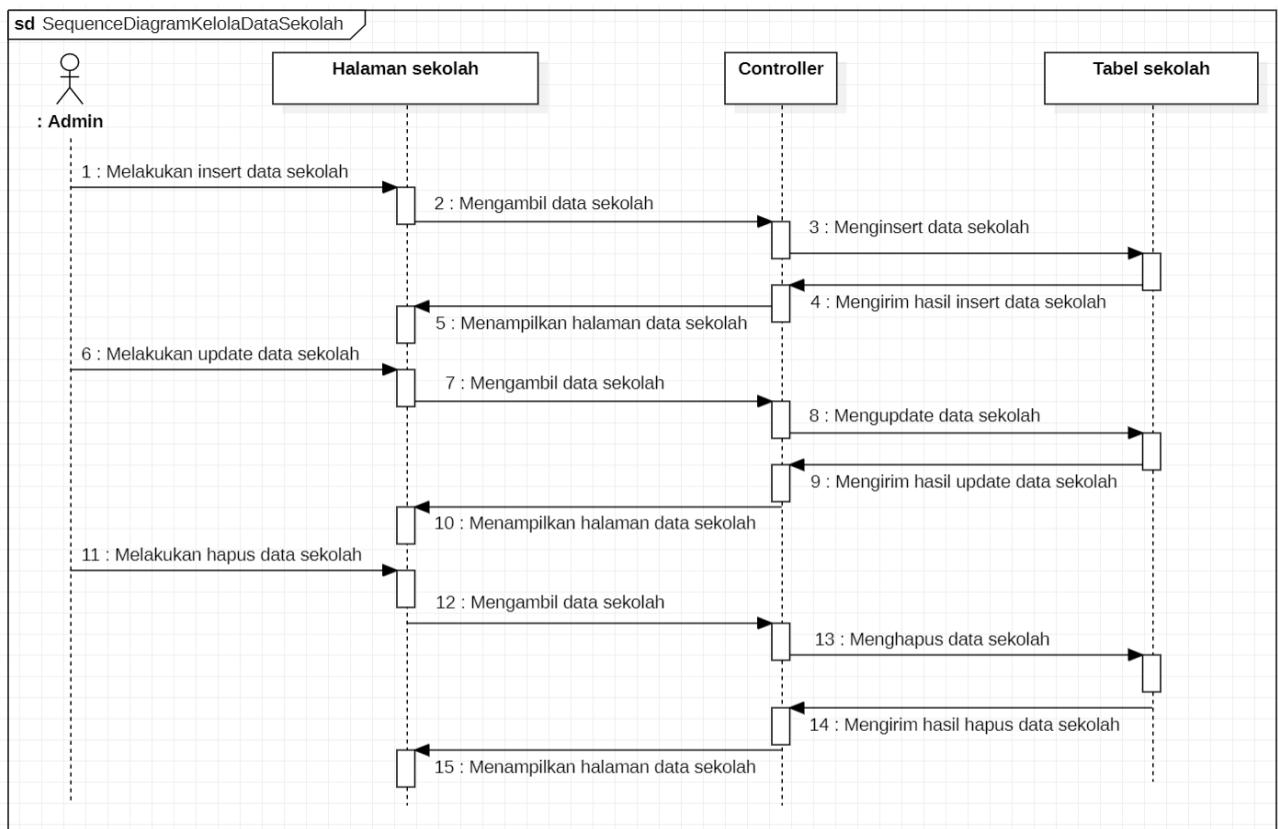
Gambar 3.34 Sequence Diagram Kelola Data Pengguna

Keterangan:

- 1.) Admin melakukan *insert* data pengguna.
- 2.) *User interface* mengambil data pengguna.
- 3.) Dari *controller* ke tabel melakukan proses *insert* data pengguna sesuai perintah admin.
- 4.) Dari *controller* ke *user interface* mengirim hasil *insert* data pengguna.
- 5.) *User interface* menampilkan halaman data pengguna.
- 6.) Admin melakukan *update* data pengguna.
- 7.) *User interface* mengambil data pengguna.
- 8.) Dari *controller* ke tabel melakukan proses *update* data pengguna sesuai perintah admin.
- 9.) Dari *controller* ke *user interface* mengirim hasil *update* data pengguna.
- 10.) *User interface* menampilkan halaman data pengguna.

- 11.) Admin melakukan hapus data pengguna.
- 12.) *User interface* mengambil data pengguna.
- 13.) Dari *controller* ke tabel melakukan proses hapus data pengguna sesuai perintah admin.
- 14.) Dari *controller* ke *user interface* mengirim hasil hapus data pengguna.
- 15.) *User interface* menampilkan halaman data pengguna.

3.2.1.3.7 Sequence Diagram Kelola Data Sekolah



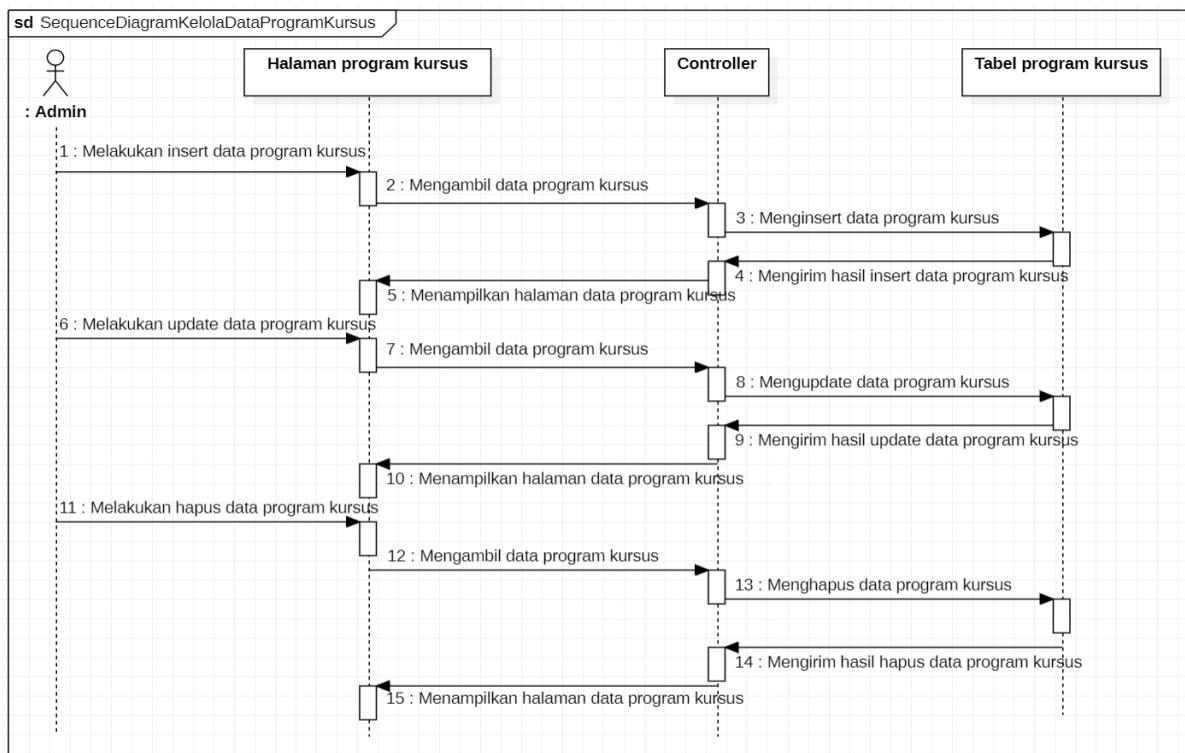
Gambar 3.35 Sequence Diagram Kelola Data Sekolah

Keterangan:

- 1.) Admin melakukan *insert* data sekolah.
- 2.) *User interface* mengambil data sekolah.
- 3.) Dari *controller* ke tabel melakukan proses *insert* data sekolah sesuai perintah admin.
- 4.) Dari *controller* ke *user interface* mengirim hasil *insert* data sekolah.

- 5.) *User interface* menampilkan halaman data sekolah.
- 6.) Admin melakukan *update* data sekolah.
- 7.) *User interface* mengambil data sekolah.
- 8.) Dari *controller* ke tabel melakukan proses *update* data sekolah sesuai perintah admin.
- 9.) Dari *controller* ke *user interface* mengirim hasil *update* data sekolah.
- 10.) *User interface* menampilkan halaman data sekolah.
- 11.) Admin melakukan hapus data sekolah.
- 12.) *User interface* mengambil data sekolah.
- 13.) Dari *controller* ke tabel melakukan proses hapus data sekolah sesuai perintah admin.
- 14.) Dari *controller* ke *user interface* mengirim hasil hapus data sekolah.
- 15.) *User interface* menampilkan halaman data sekolah.

3.2.1.3.8 Sequence Diagram Kelola Data Program Kursus

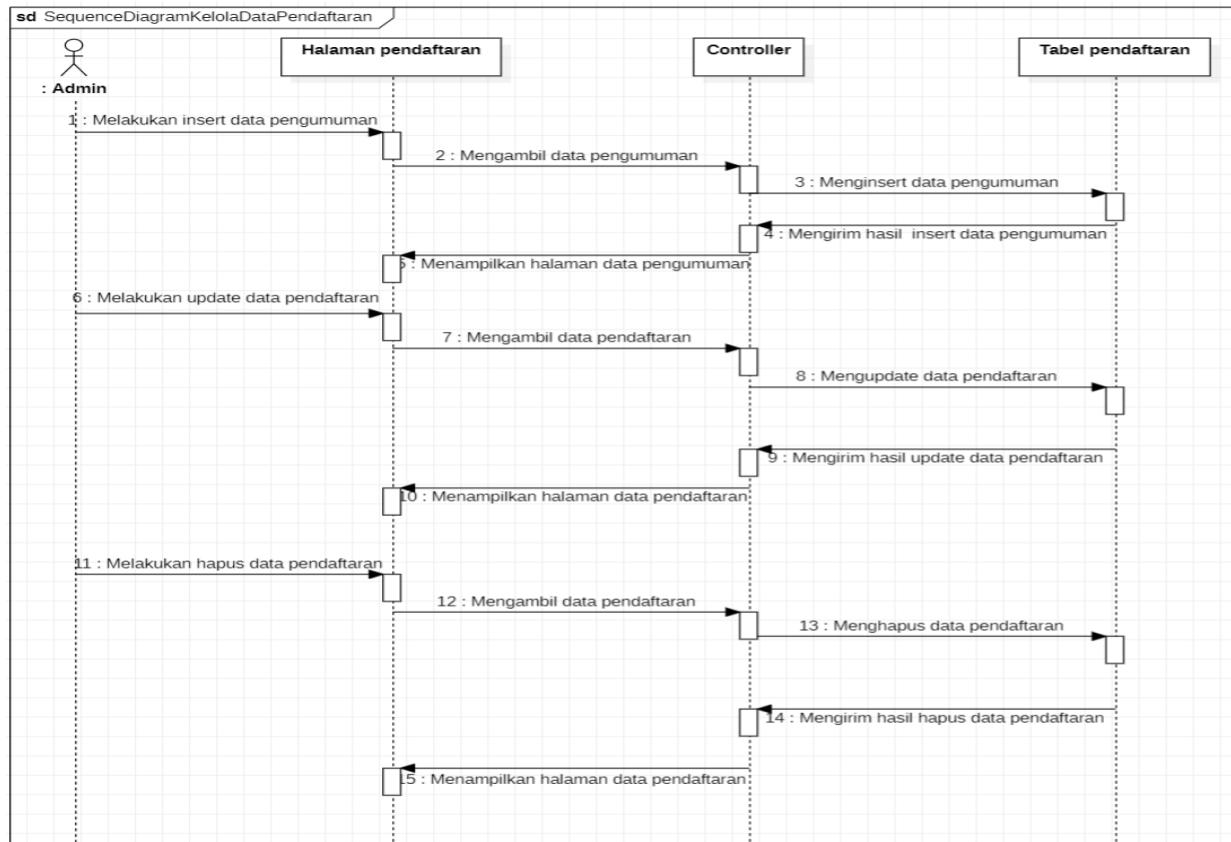


Gambar 3.36 Sequence Diagram Kelola Data Program Kursus

Keterangan:

- 1.) Admin melakukan *insert* data program kursus.
- 2.) *User interface* mengambil data program kursus
- 3.) Dari *controller* ke tabel melakukan proses *insert* data program kursus sesuai perintah admin.
- 4.) Dari *controller* ke *user interface* mengirim hasil *insert* data program kursus.
- 5.) *User interface* menampilkan halaman data program kursus.
- 6.) Admin melakukan *update* data program kursus.
- 7.) *User interface* mengambil data program kursus.
- 8.) Dari *controller* ke tabel melakukan proses *update* data program kursus sesuai perintah admin.
- 9.) Dari *controller* ke *user interface* mengirim hasil *update* data program kursus.
- 10.) *User interface* menampilkan halaman data program kursus.
- 11.) Admin melakukan hapus data program kursus.
- 12.) *User interface* mengambil data program kursus.
- 13.) Dari *controller* ke tabel melakukan proses hapus data program kursus sesuai perintah admin.
- 14.) Dari *controller* ke *user interface* mengirim hasil hapus data program kursus.
- 15.) *User interface* menampilkan halaman data program kursus.

3.2.1.3.9 Sequence Diagram Kelola Data Pendaftaran



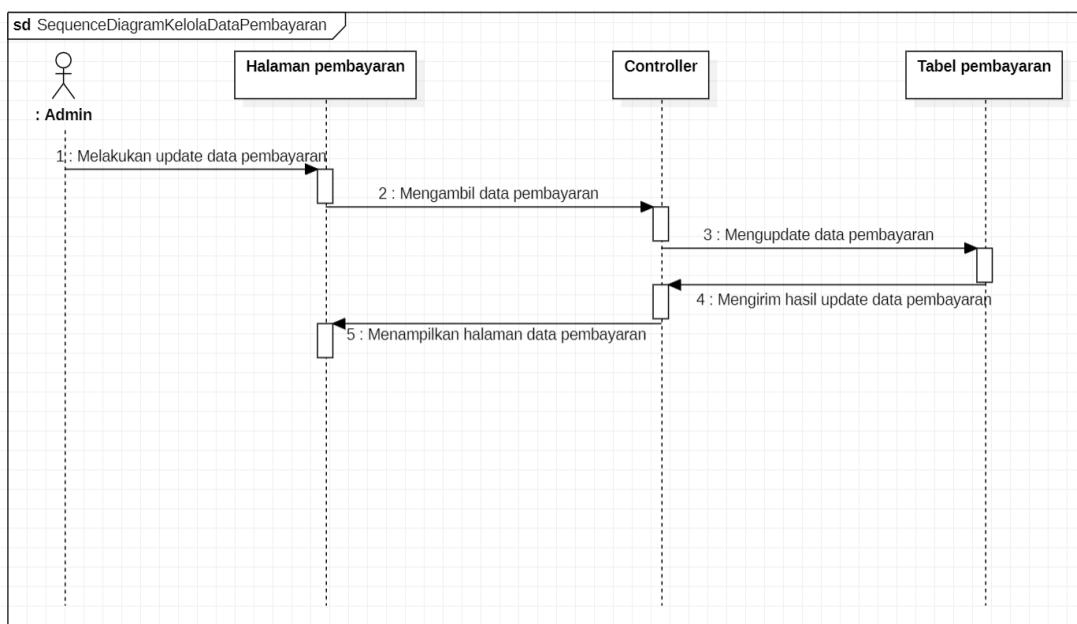
Gambar 3.37 Sequence Diagram Kelola Data Pendaftaran

Keterangan:

- 1.) Admin melakukan *insert* data pendaftaran.
- 2.) *User interface* mengambil data pendaftaran.
- 3.) Dari *controller* ke tabel melakukan proses *insert* data pendaftaran sesuai perintah admin.
- 4.) Dari *controller* ke *user interface* mengirim hasil *insert* data pendaftaran.
- 5.) *User interface* menampilkan halaman data pendaftaran.
- 6.) Admin melakukan *update* data pendaftaran.
- 7.) *User interface* mengambil data pendaftaran.
- 8.) Dari *controller* ke tabel melakukan proses *update* data *pendaftaran* sesuai perintah admin.
- 9.) Dari *controller* ke *user interface* mengirim hasil *update* data pendaftaran.

- 10.) *User interface* menampilkan halaman data pendaftaran.
- 11.) Admin melakukan hapus data pendaftaran.
- 12.) *User interface* mengambil data pendaftaran.
- 13.) Dari *controller* ke tabel melakukan proses hapus data pendaftaran sesuai perintah admin.
- 14.) Dari *controller* ke *user interface* mengirim hasil hapus data pendaftaran.
- 15.) *User interface* menampilkan halaman data pendaftaran.

3.2.1.3.10 Sequence Diagram Kelola Data Pembayaran

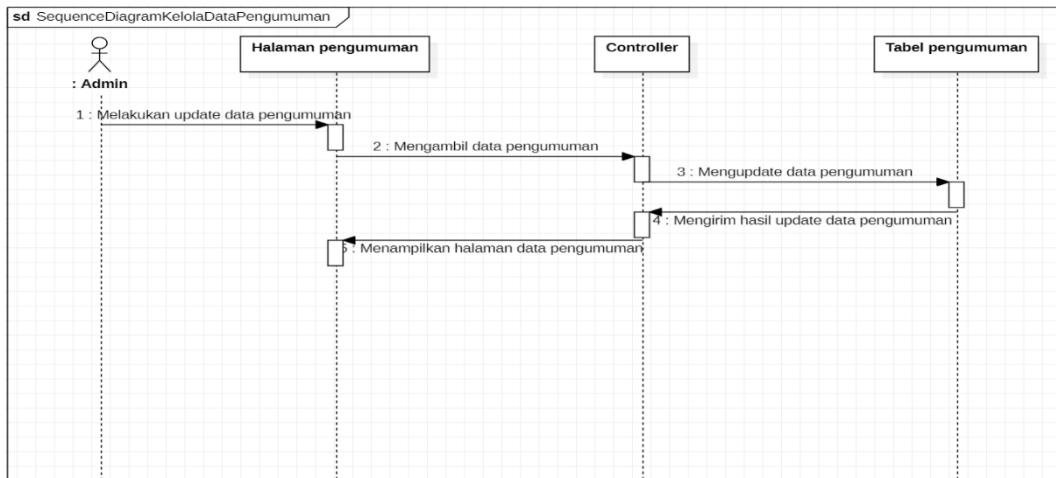


Gambar 3.38 Sequence Diagram Kelola Data Pembayaran

Keterangan:

- 1.) Admin melakukan *update* data pembayaran.
- 2.) *User interface* mengambil data pembayaran.
- 3.) Dari *controller* ke tabel melakukan proses *update* data pembayaran sesuai perintah admin.
- 4.) Dari *controller* ke *user interface* mengirim hasil *update* data pembayaran
- 5.) *User interface* menampilkan data pembayaran.

3.2.1.3.11 Sequence Diagram Kelola Data Pengumuman

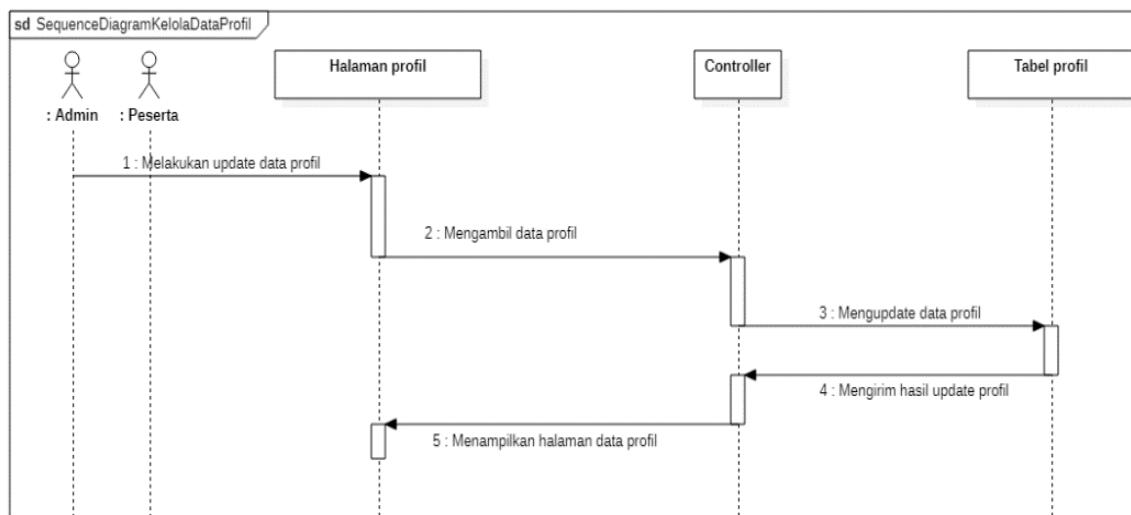


Gambar 3.39 Sequence Diagram Kelola Data Pengumuman

Keterangan:

- 1.) Admin melakukan *update* data pengumuman.
- 2.) *User interface* mengambil data pengumuman.
- 3.) Dari *controller* ke tabel melakukan proses *update* data pengumuman sesuai perintah admin.
- 4.) Dari *controller* ke *user interface* mengirim hasil *update* data pengumuman.
- 5.) *User interface* menampilkan halaman data pengumuman.

3.2.1.3.12 Sequence Diagram Kelola Data Profil

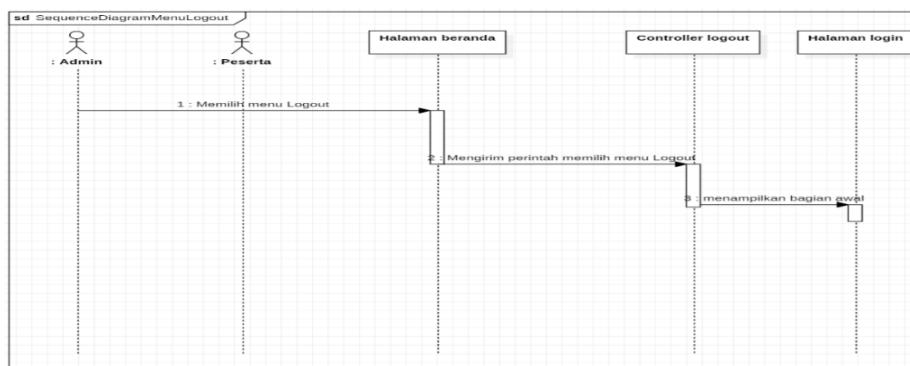


Gambar 3.40 Sequence Diagram Kelola Data Profil

Keterangan:

- 1.) Admin dan peserta melakukan *update* data profil.
- 2.) *User interface* mengambil data *profile*.
- 3.) Dari *controller* ke tabel melakukan proses *update* data pembayaran sesuai perintah admin.
- 4.) Dari *controller* ke *user interface* mengirim hasil *update* data profil.
- 5.) *User interface* menampilkan halaman data profil.

3.2.1.3.13 Sequence Diagram Logout



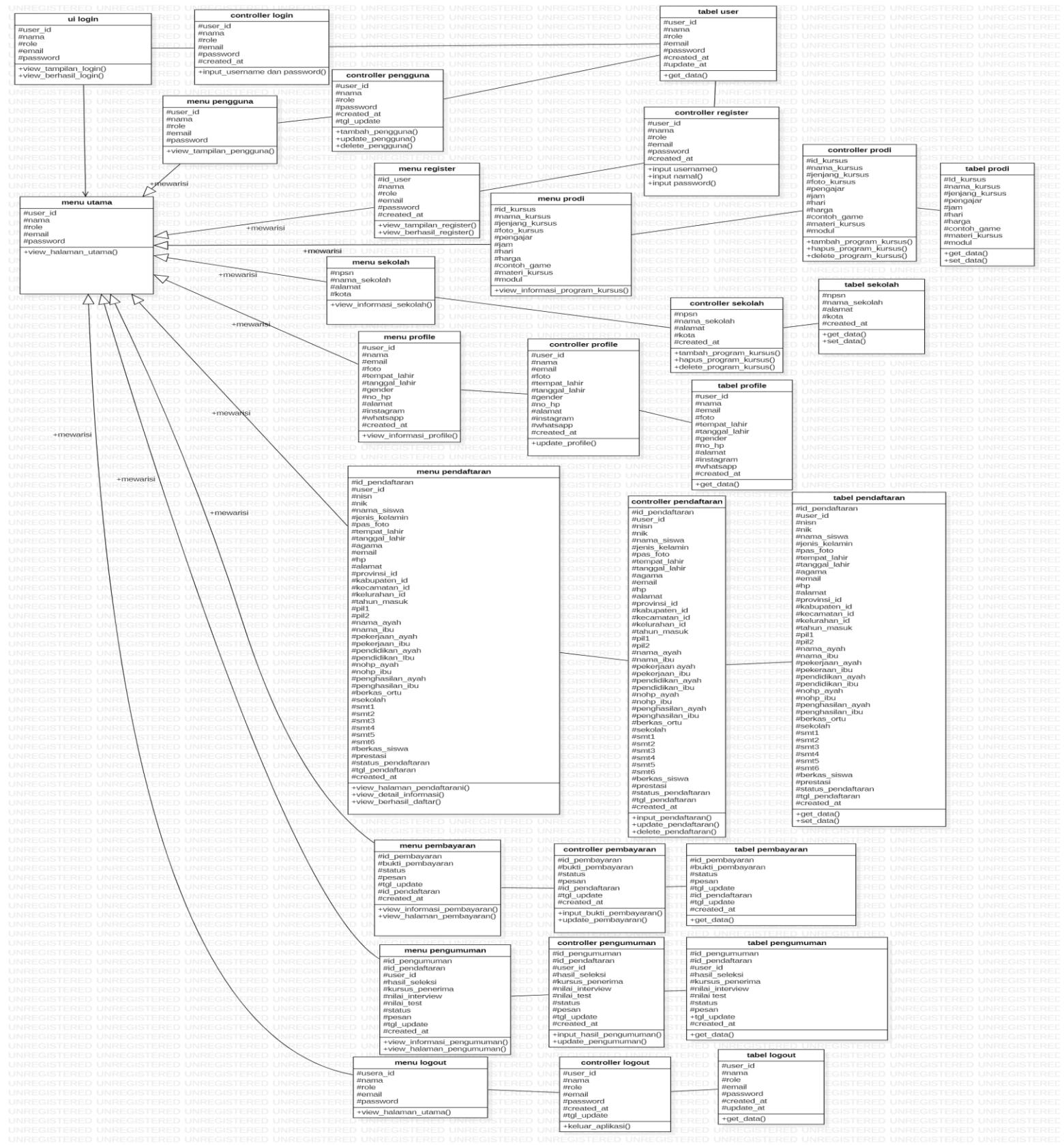
Gambar 3.41 Sequence Diagram Logout

Keterangan:

- 1.) Admin dan peserta memilih menu *logout*.
- 2.) *User interface* mengirim perintah memilih menu *logout* ke *controller*.
- 3.) Dari *controller* ke *user interface* menampilkan utama *login*.

3.2.1.4 Class Diagram

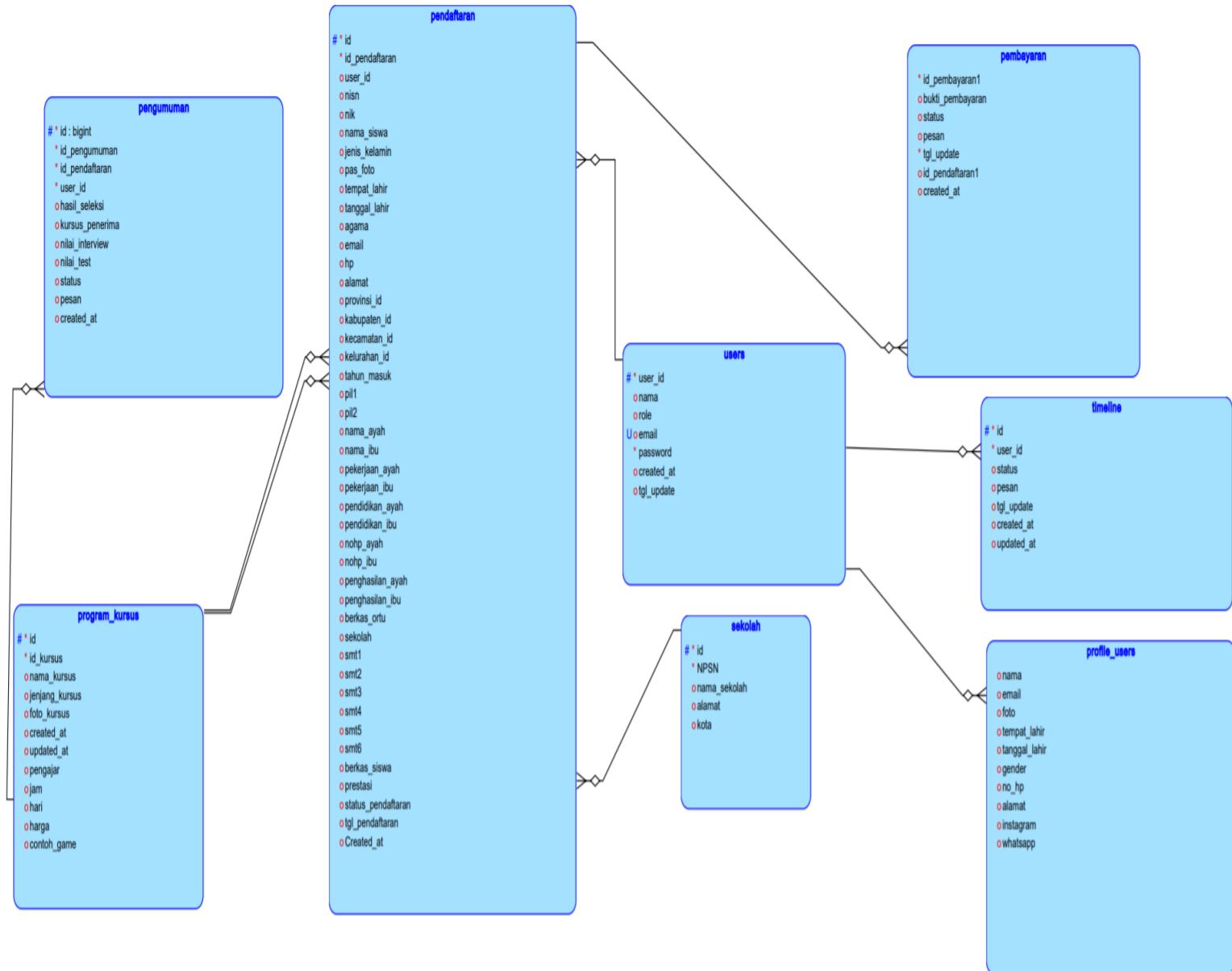
Class diagram adalah suatu bagian dari diagram UML (*Unified Modelling Language*) yang menjelaskan antara suatu struktur dari sebuah sistem yang didalamnya terdapat bagian dari kelas-kelas dengan relasi relasinya (hubungan). *Class diagram* ini dapat menunjukkan suatu proses antara hubungan entitas 1 dengan entitas yang lainnya yang saling terhubung dikarenakan adanya sebuah hubungan yang ditunjukkan dalam *symbol* relasi. Dalam *class diagram* ini terdapat berbagai *table*, atribut dalam *class* yang digambarkan seperti dibawah ini



Gambar 3.42 Class Diagram

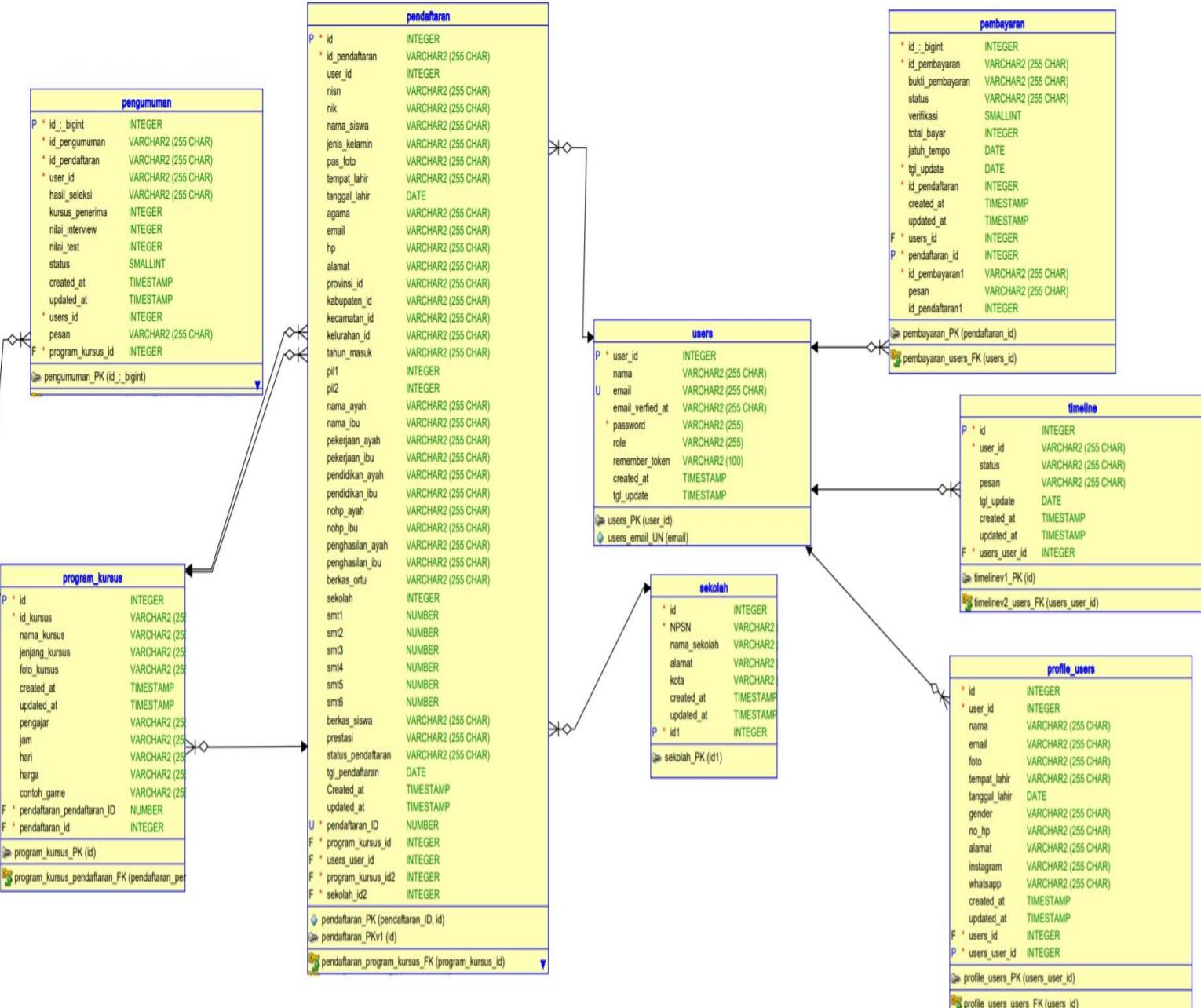
3.3 Perancangan Database

3.3.1 Conceptual Data Model (CDM)



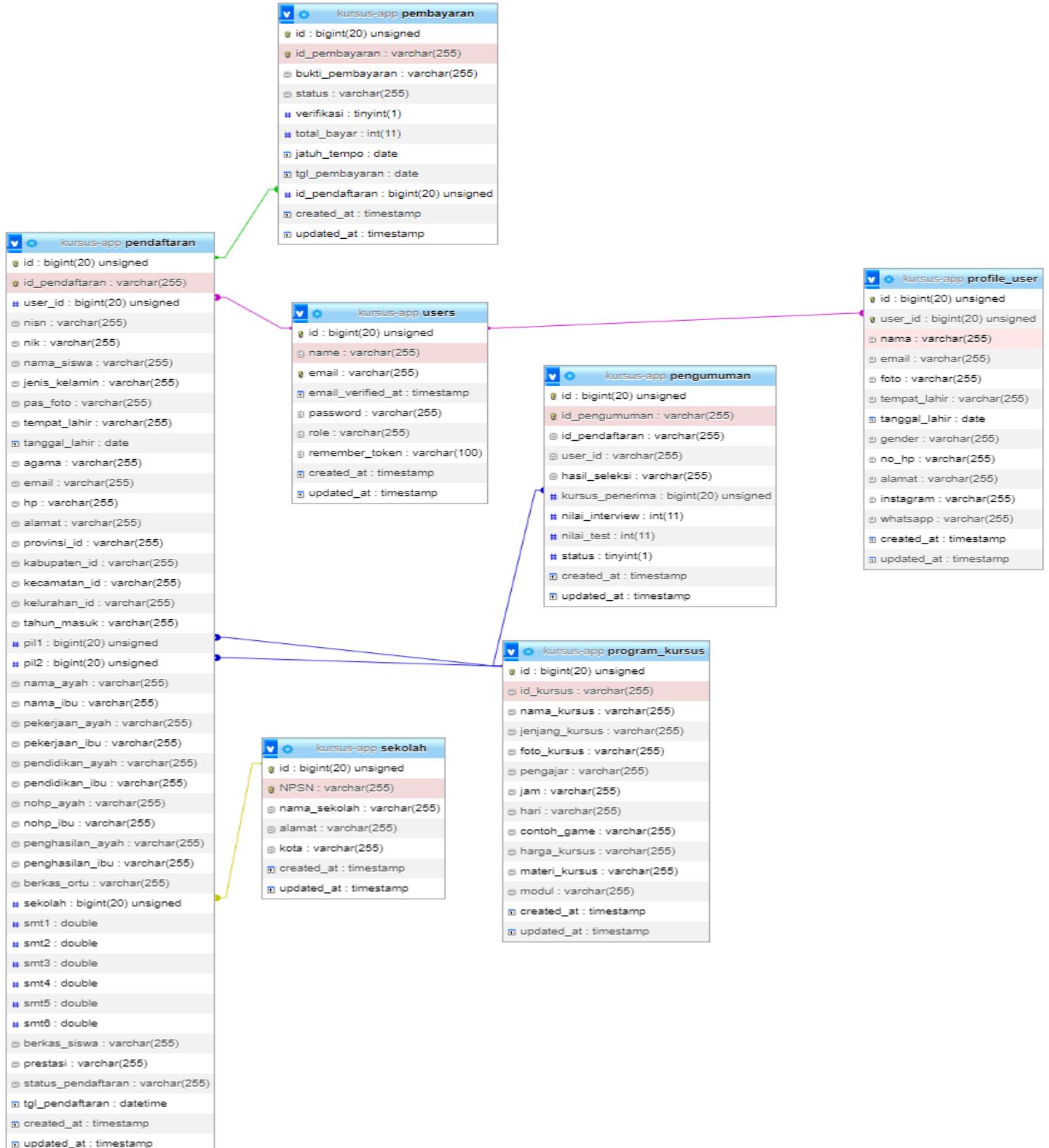
Gambar 3.43 Conceptual Data Model (CDM)

3.3.2 Physical Data Model (PDM)



Gambar 3.44 Physical Data Model (CDM)

3.3.3 Entity Relationship Diagram (ERD)

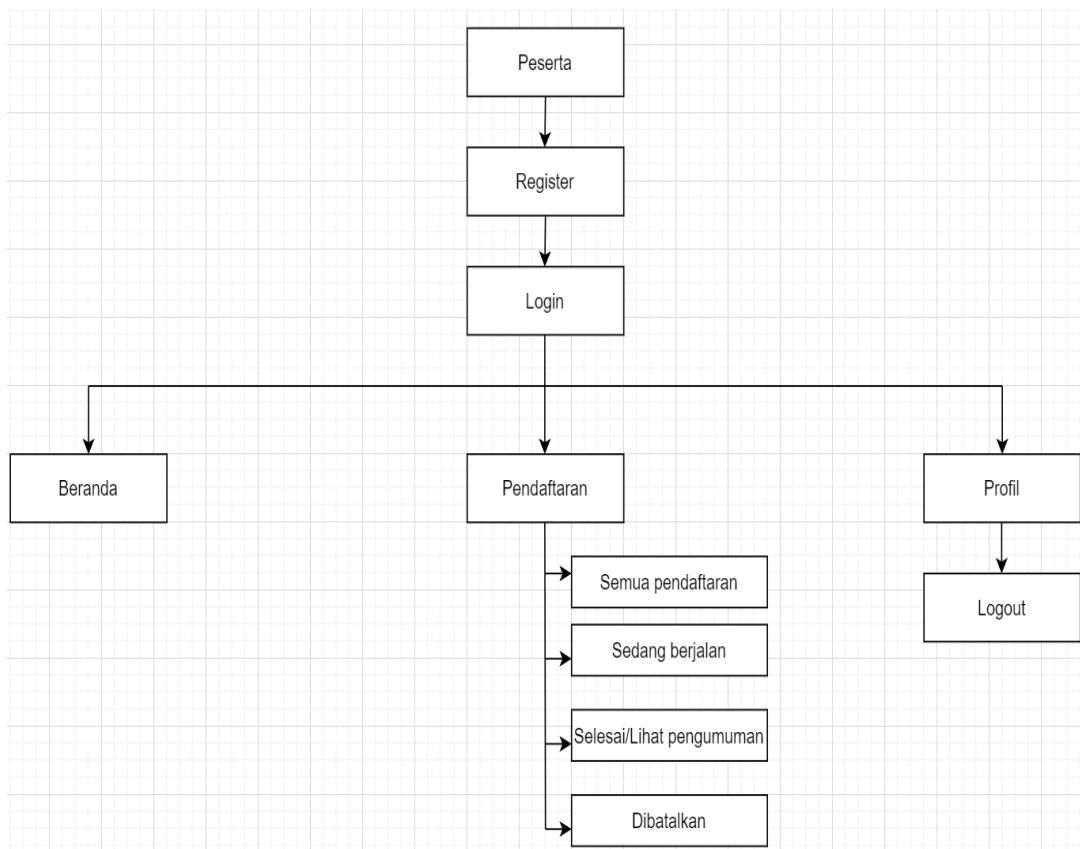


Gambar 3.45 Entity Relationship Diagram (ERD)

3.4 Perancangan Struktur Menu

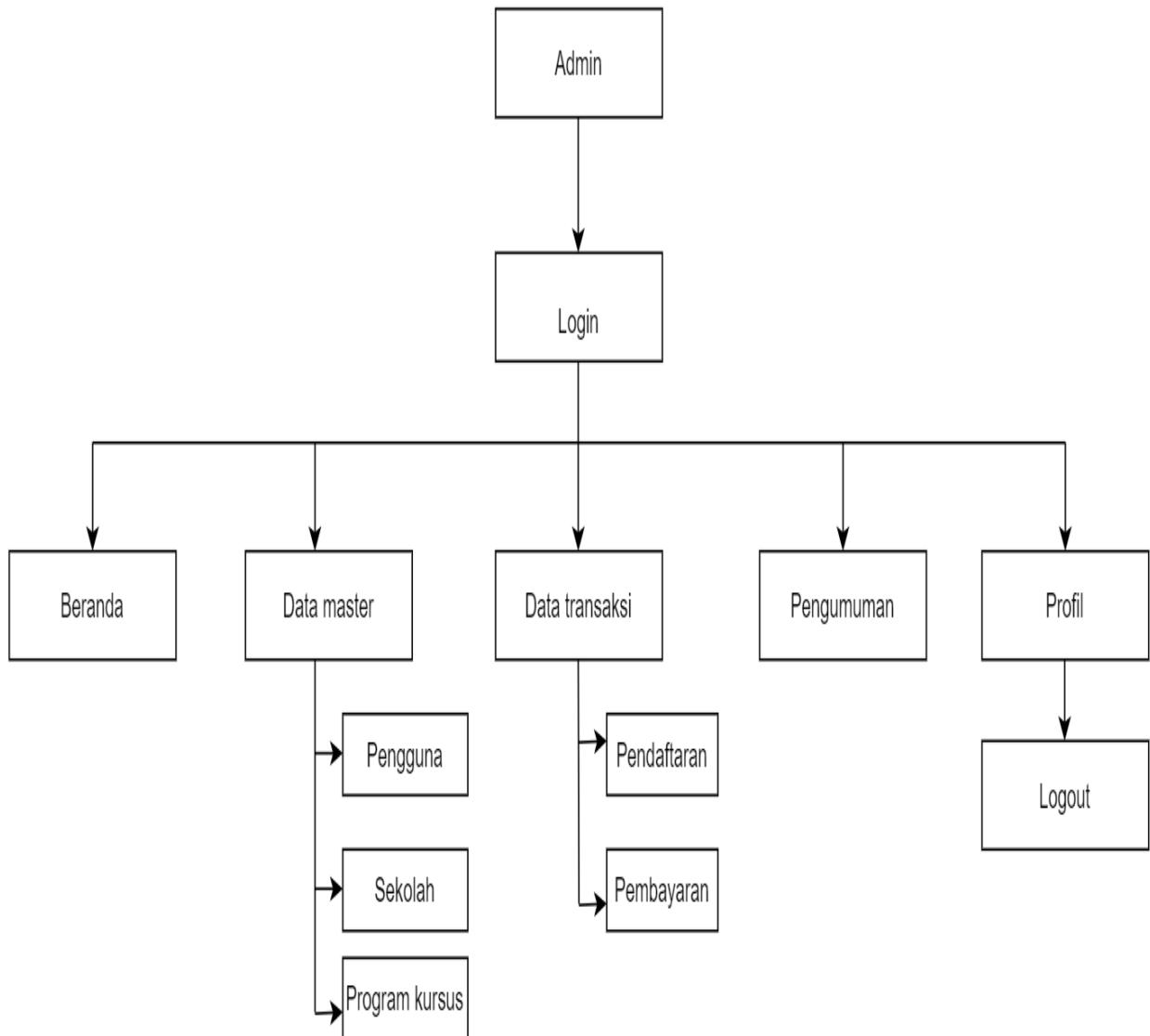
Struktur menu adalah suatu gambaran umum dari sebuah rancangan yang dibuat untuk memudahkan pengguna dalam mengetahui informasi melalui struktur. Struktur menu ini berisikan mengenai menu setiap entitas peserta dan admin. Tujuan dari perancangan struktur menu ini adalah untuk mengetahui gambaran lebih jauh mengenai menu dan *tools* yang terdapat pada aplikasi tersebut. Berikut ini adalah struktur menu pada aplikasi sistem pendaftaran *programming course* roblox yang terbagi atas dua struktur yakni struktur menu admin dan struktur menu peserta.

3.4.1 Struktur Menu Peserta



Gambar 3.46 Struktur Menu Peserta

3.4.2 Struktur Menu Admin



Gambar 3.47 Struktur Menu Admin

3.5 Antarmuka

Antarmuka disini adalah salah satu bagian dari perancangan design aplikasi, dimana tujuan dari antarmuka ini ialah untuk mengetahui lebih detail setiap bagian bagian dari aplikasi tersebut. Rancangan antarmuka menunjukkan bagaimana sistem berinteraksi atau berkomunikasi dengan *computer*. Berikut ini adalah halaman antarmuka pada aplikasi sistem pendaftaran *programming course* roblox:

3.5.1 Rancangan Antarmuka Peserta

3.5.1.1 Rancangan antarmuka halaman *register*

Rancangan antarmuka halaman *register* yang terdiri dari:

- Tombol "Daftar" di bagian atas.
- Label "Nama" di dekat inputan pertama.
- Inputan pertama (text box).
- Label "Username" di dekat inputan kedua.
- Inputan kedua (text box).
- Label "Password" di dekat inputan ketiga.
- Inputan ketiga (text box).
- Tombol "Daftar" di bagian bawah.

Gambar 3.48 Antarmuka Halaman *Register*

Pada tulisan daftar diatas merupakan rancangan *register* untuk mendaftar atau membuat akun bagi masyarakat yang belum memiliki akun. Apabila peserta belum memiliki akun maka wajib mendaftar akun dengan memasukan nama, *username*, email, dan *password*.

3.5.1.2 Rancangan antarmuka halaman *login*

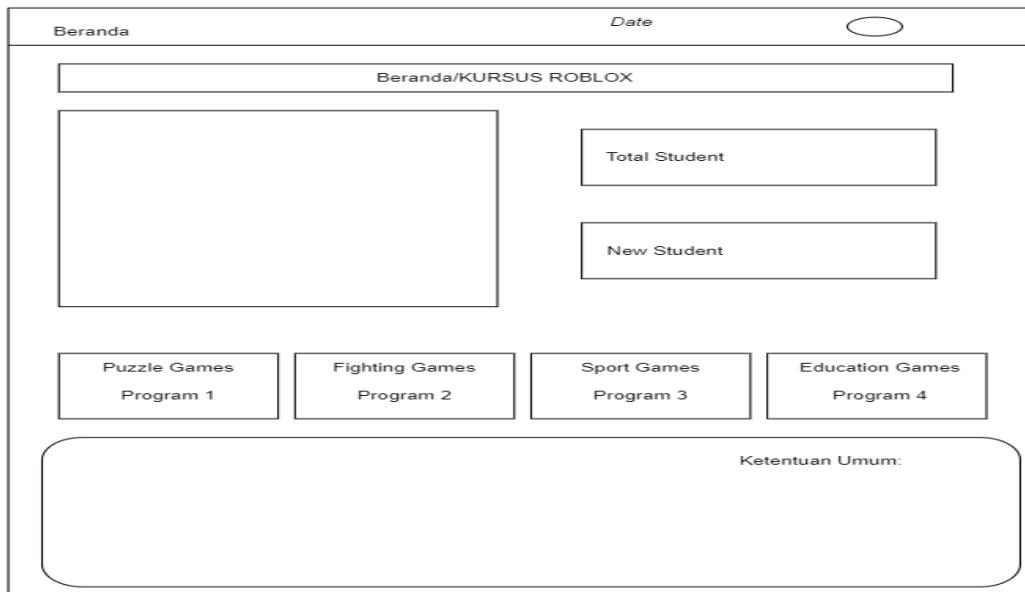
Rancangan antarmuka halaman *login* yang terdiri dari:

- Tombol "Daftar" di bagian kiri atas.
- Label "Username" di dekat inputan pertama.
- Inputan pertama (text box).
- Label "Password" di dekat inputan kedua.
- Inputan kedua (text box).
- Tombol "Daftar" di bagian bawah.

Gambar 3.49 Antarmuka Halaman *Login*

Pada tulisan daftar diatas merupakan rancangan *login* untuk masuk aplikasi. Apabila sudah memiliki akun maka peserta dapat melakukan *login* dengan memasukkan *username* dan *password* yang sudah dimiliki dari proses *register*/daftar akun sebelumnya.

3.5.1.3 Rancangan antarmuka halaman beranda



Gambar 3.50 Antarmuka Halaman Beranda

Pada gambar diatas merupakan rancangan beranda atau dalam hal ini adalah halaman utama apabila peserta telah melakukan maka selanjutnya menampilkan beranda yang berisi berbagai informasi mengenai kursus tersebut.

3.5.1.4 Rancangan antarmuka halaman kelola data profil

Rancangan antarmuka halaman kelola data profil yang ditunjukkan dalam diagram wireframe. Di bagian atas kiri terdapat tautan 'Profil Pengguna'. Di bagian atas kanan terdapat tautan 'Date' dan simbol lingkaran. Di bagian tengah kiri terdapat foto placeholder dengan tulisan 'Calon Peserta'. Di bagian tengah kanan terdapat tiga tab: 'Profil', 'Pengaturan', dan 'Perbaharui Sandi'. Tab 'Profil' menunjukkan formularis untuk 'Informasi Pribadi' yang mencakup 'Nama', 'Jenis Kelamin', 'Tempat Lahir', 'Tanggal Lahir', dan 'Alamat'. Tab 'Pengaturan' menunjukkan formularis untuk 'Kontak Pribadi' dan 'Sosial Media'. Tab 'Perbaharui Sandi' menunjukkan formularis untuk 'Nama' dan 'Sandi'.

Gambar 3.51 Antarmuka Halaman Kelola Data Profil

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman profil peserta. Simbol persegi panjang tersebut ialah berisikan mengenai informasi data pribadi peserta seperti tertera pada gambar diatas. Pada lingkaran bulat seperti *symbol* gambar diatas ialah *photo* pribadi peserta.

3.5.1.5 Rancangan antarmuka halaman pendaftaran

Data Pendaftaran		Date																
<p>Data Pribadi</p> <table border="1"> <tr> <td>NISN <input type="text"/></td> <td>Jenis Kelamin <input type="text"/></td> <td>NIK <input type="text"/></td> </tr> <tr> <td>Nama <input type="text"/></td> <td>Tanggal Lahir <input type="text"/></td> <td>Agama <input type="text"/></td> </tr> <tr> <td>Tempat Lahir <input type="text"/></td> <td>Pas Photo <input type="text"/></td> <td></td> </tr> <tr> <td>Alamat <input type="text"/></td> <td></td> <td></td> </tr> <tr> <td>Email <input type="text"/></td> <td>No Hp/ Whatsapp <input type="text"/></td> <td></td> </tr> </table>				NISN <input type="text"/>	Jenis Kelamin <input type="text"/>	NIK <input type="text"/>	Nama <input type="text"/>	Tanggal Lahir <input type="text"/>	Agama <input type="text"/>	Tempat Lahir <input type="text"/>	Pas Photo <input type="text"/>		Alamat <input type="text"/>			Email <input type="text"/>	No Hp/ Whatsapp <input type="text"/>	
NISN <input type="text"/>	Jenis Kelamin <input type="text"/>	NIK <input type="text"/>																
Nama <input type="text"/>	Tanggal Lahir <input type="text"/>	Agama <input type="text"/>																
Tempat Lahir <input type="text"/>	Pas Photo <input type="text"/>																	
Alamat <input type="text"/>																		
Email <input type="text"/>	No Hp/ Whatsapp <input type="text"/>																	
Data Pilihan Program Kursus																		
<p>Pilihan 1 <input type="text"/></p>																		
<p>Data Orang Tua</p> <table border="1"> <tr> <td>Nama Ayah <input type="text"/></td> <td>Pekerjaan Ayah <input type="text"/></td> <td>No Telepon Ayah <input type="text"/></td> </tr> <tr> <td>Nama Ibu <input type="text"/></td> <td>Pekerjaan Ibu <input type="text"/></td> <td>No Telepon Ibu <input type="text"/></td> </tr> <tr> <td>Penghasilan Ayah <input type="text"/></td> <td>Penghasilan Ibu <input type="text"/></td> <td>Slip Gaji <input type="text"/></td> </tr> </table>				Nama Ayah <input type="text"/>	Pekerjaan Ayah <input type="text"/>	No Telepon Ayah <input type="text"/>	Nama Ibu <input type="text"/>	Pekerjaan Ibu <input type="text"/>	No Telepon Ibu <input type="text"/>	Penghasilan Ayah <input type="text"/>	Penghasilan Ibu <input type="text"/>	Slip Gaji <input type="text"/>						
Nama Ayah <input type="text"/>	Pekerjaan Ayah <input type="text"/>	No Telepon Ayah <input type="text"/>																
Nama Ibu <input type="text"/>	Pekerjaan Ibu <input type="text"/>	No Telepon Ibu <input type="text"/>																
Penghasilan Ayah <input type="text"/>	Penghasilan Ibu <input type="text"/>	Slip Gaji <input type="text"/>																
<p>Data Pendidikan</p> <table border="1"> <tr> <td>Nama Sekolah <input type="text"/></td> <td>Jurusan <input type="text"/></td> <td></td> <td></td> </tr> <tr> <td>Semester 1 <input type="text"/></td> <td>Semester 2 <input type="text"/></td> <td>Semester 3 <input type="text"/></td> <td>Semester 4 <input type="text"/></td> <td>Semester 5 <input type="text"/></td> </tr> <tr> <td>Raport <input type="text"/></td> <td>Prestasi <input type="text"/></td> <td></td> <td></td> <td></td> </tr> </table>				Nama Sekolah <input type="text"/>	Jurusan <input type="text"/>			Semester 1 <input type="text"/>	Semester 2 <input type="text"/>	Semester 3 <input type="text"/>	Semester 4 <input type="text"/>	Semester 5 <input type="text"/>	Raport <input type="text"/>	Prestasi <input type="text"/>				
Nama Sekolah <input type="text"/>	Jurusan <input type="text"/>																	
Semester 1 <input type="text"/>	Semester 2 <input type="text"/>	Semester 3 <input type="text"/>	Semester 4 <input type="text"/>	Semester 5 <input type="text"/>														
Raport <input type="text"/>	Prestasi <input type="text"/>																	
Buat Pendaftaran																		

Gambar 3.52 Antarmuka Halaman Pendaftaran

Pada gambar diatas ialah gambaran antarmuka secara umum dari menu pendaftaran, dimana pada menu pendaftaran ini terdiri dari beberapa persyaratan yang dilengkapi oleh peserta seperti data pribadi, data pilihan kursus, data orang tua, dan data Pendidikan.

3.5.1.6 Rancangan antarmuka halaman pembayaran

The wireframe diagram illustrates a user interface for a payment page. At the top left, there is a 'Detail Pendaftaran' button. To its right is a placeholder for a date ('Date') and a circular icon. Below this header, there are three buttons: 'Data Pendaftaran' (highlighted with a thick border), 'Upload Pembayaran' (disabled state, greyed out), and 'Terverifikasi' (disabled state, greyed out). A large rectangular area contains a sub-form titled 'Kirim bukti pembayaran'. Inside this sub-form, there is a label 'Pilih dokumen' above a file input field, and two buttons at the bottom labeled 'Tutup' and 'Perbaharui data'.

Gambar 3.53 Antarmuka Halaman Pembayaran

Pada gambar diatas ialah halaman antarmuka pembayaran. Pada menu pembayaran tersebut terdiri dari *button upload* bukti pembayaran yang harus dimasukkan oleh peserta setelah melalui tahap pengisian menu pendaftaran sebelumnya. Kemudian digambar tersebut juga terdapat status pendaftaran dari peserta, jika sudah terverifikasi maka peserta dapat melakukan tahap proses melakukan pembayaran yakni dengan melakukan *transfer* ke *via bank* yang sudah tertera di halaman beranda atau *dashboard* dengan jumlah biaya yang harus dibayar adalah bergantung dari pilihan program kursus yang dipilih peserta. Setelah mengupload, kemudian terdapat *button* perbaharui data dan *button* tutup.

3.5.1.7 Rancangan antarmuka halaman detail pendaftaran

Detail Pendaftaran

Date _____

Data Pendaftaran

PROFIL SISWA

ID Pendaftaran	<input type="text"/>
Nama	<input type="text"/>
Jenis kelamin	<input type="text"/>
Tempat tanggal lahir	<input type="text"/>
Agama	<input type="text"/>
NISN	<input type="text"/>
NIK	<input type="text"/>
Alamat	<input type="text"/>
Email	<input type="text"/>
Telepon/ Whatsapp	<input type="text"/>

DATA ORANG TUA

Nama ayah	<input type="text"/>	Nama ibu	<input type="text"/>
Pekerjaan ayah	<input type="text"/>	Pekerjaan ibu	<input type="text"/>
No. Handphone	<input type="text"/>	No. Handphone	<input type="text"/>
Gaji	<input type="text"/>	Tanggungan	<input type="text"/>
Dokumen	<input type="text"/>		

DATA REGISTRASI

Tahun	<input type="text"/>	Pilihan 1	<input type="text"/>
DATA PENDIDIKAN		Jurusan	<input type="text"/>
Nama sekolah	<input type="text"/>	Semester 1	<input type="text"/>
Semester 3	<input type="text"/>	Semester 2	<input type="text"/>
		Semester 4	<input type="text"/>
		Semester 5	<input type="text"/>

Selesai

Gambar 3.54 Antarmuka Halaman Detail Pendaftaran

Pada gambar diatas merupakan antarmuka halaman detail pendaftaran. Pada hal detail pendaftaran berisikan mengenai informasi dari peserta, mulai dari profil peserta, data pribadi, data orang tua, data pilihan kursus, dan data Pendidikan.

3.5.1.8 Rancangan antarmuka halaman pengumuman

Hasil Pengumuman Date _____

Pengumuman Pendaftaran Peserta Kursus Programming Course
didaftarkan oleh... pada 2022-12-04 20:22:4

PROGRAMMING COURSE ROBLOX
Jalan Sariashih IV, No.54, Sarjadi, Kota Bandung, Jawa Barat, 40151 Indonesia

Data Pendaftar

No. Pendaftaran	<input type="text"/>
NISN Siswa	<input type="text"/>
Nama Siswa	<input type="text"/>
Asal Sekolah	<input type="text"/>

Pas Photo

Selamat! Anda dinyatakan LULUS sebagai peserta kursus di programming course roblox 2022/2023

Program Kursus Penerima:	<input type="text"/>
Hari:	<input type="text"/>
Jam:	<input type="text"/>
Pengajar:	<input type="text"/>

Gambar 3.55 Antarmuka Halaman Pengumuman

Pada gambar diatas merupakan antarmuka halaman pengumuman. Pada hal pengumuman berisikan mengenai informasi dan hasil pendaftaran yang telah dilakukan oleh peserta, baik dinyatakan lulus maupun tidak lulus. Jadi, pada halaman antarmuka pengumuman juga memiliki *button print* yang fungsinya agar peserta dapat mencetak hasil pengumuman kursus tersebut.

3.5.1.9 Rancangan antarmuka halaman *logout*

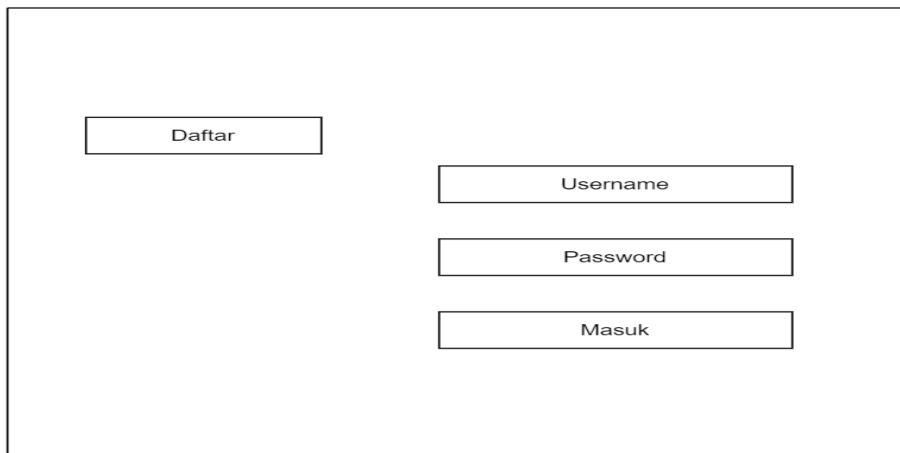
Username

Password

Gambar 3.56 Antarmuka Halaman *Logout*

3.5.2 Rancangan Antarmuka Admin

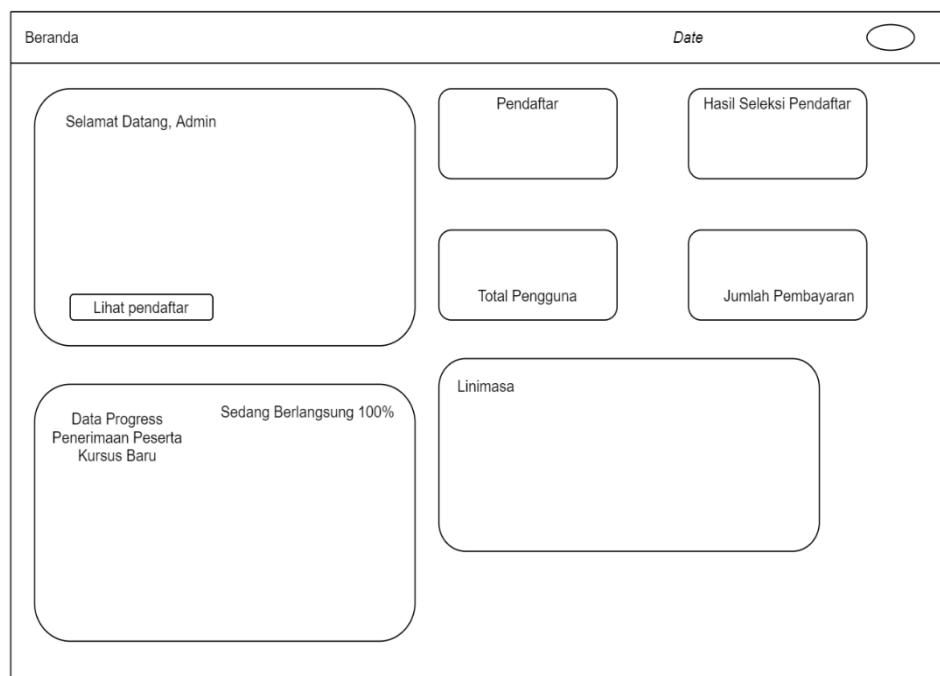
3.5.2.1 Rancangan antarmuka *login*



Gambar 3.57 Antarmuka Halaman Login

Pada rancangan antarmuka diatas adalah rancangan antarmuka *login*. Pada menu ini, admin dapat melakukan *login* dengan memasukkan *username* dan *password* yang sudah dimiliki. *Username* yang digunakan pada rancangan menu *login* ini menggunakan *username* email pribadi milik admin.

3.5.2.2 Rancangan antarmuka halaman beranda



Gambar 3.58 Antarmuka Halaman Beranda

Pada rancangan antarmuka diatas adalah rancangan antarmuka beranda admin. Pada menu halaman beranda peserta ini terdiri dari berbagai informasi seperti: informasi mengenai pendaftar, hasil seleksi pendaftar, total pengguna, jumlah pembayaran, data *progress* penerimaan peserta kursus baru serta data linimasa. Pada menu beranda admin ini juga memiliki *button* “Lihat pendaftar” untuk melihat yang sudah mendaftar kursus baru baru ini.

3.5.2.3 Rancangan antarmuka halaman kelola data profil

Profil	Pengaturan	Perbaharui Sandi
Informasi Pribadi	<input type="text"/>	<input type="text"/>
Nama	<input type="text"/>	<input type="text"/>
Jenis Kelamin	<input type="text"/>	<input type="text"/>
Tempat Lahir	<input type="text"/>	<input type="text"/>
Tanggal Lahir	<input type="text"/>	<input type="text"/>
Alamat	<input type="text"/>	<input type="text"/>
Kontak Pribadi	<input type="text"/>	<input type="text"/>
Sosial Media	<input type="text"/>	

Gambar 3.59 Antarmuka Halaman Kelola Data Profil

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman profil admin. Simbol persegi panjang tersebut ialah berisikan mengenai informasi data pribadi admin seperti tertera pada gambar diatas. Pada lingkaran bulat seperti *symbol* gambar diatas ialah *photo* pribadi admin.

3.5.2.4 Rancangan antarmuka halaman kelola data pengguna

Rancangan antarmuka halaman kelola data pengguna. Tampilan ini menampilkan daftar pengguna dengan 10 entri. Setiap baris daftar mengandung kolom untuk Nama, Jenis Kelamin, Telepon, Email, dan Aksi. Di bagian atas, terdapat tombol Print dan Tambahkan Pengguna, serta input Search. Di bagian bawah, terdapat tombol << 1 >>.

User Data				
Show	10	entries	Search:	
<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="radio"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Aksi				
<input type="radio"/>				
<input type="radio"/>				
<input type="radio"/>				
<< 1 >>				

Gambar 3.60 Antarmuka Halaman Kelola Data Pengguna

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman kelola data pengguna. Pada menu kelola data pengguna ini admin dapat melihat data dan informasi mengenai siapa saja yang telah mendaftarkan akun pada *system pendaftaran programming course roblox*.

3.5.2.5 Rancangan antarmuka halaman kelola data sekolah

Rancangan antarmuka halaman kelola data sekolah. Tampilan ini menampilkan daftar sekolah dengan 10 entri. Setiap baris daftar mengandung kolom untuk No, NPSN, Nama Sekolah, Alamat, Kota, dan Aksi. Di bagian atas, terdapat tombol Print dan Tambah Sekolah, serta input Search. Di bagian bawah, terdapat tombol << 1 >>.

Data Sekolah					
Show	10	entries	Search:		
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="radio"/>
Aksi					
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<< 1 >>					

Gambar 3.61 Antarmuka Halaman Kelola Data Sekolah

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman kelola data sekolah. Pada menu kelola data sekolah ini admin dapat membuat

atau menambahkan serta menghapus data sekolah seperti nama sekolah, NPSN, jenis kelamin, alamat dan kota sebagai salah satu kelengkapan persyaratan pendaftaran peserta.

3.5.2.6 Rancangan antarmuka halaman kelola data program kursus

Rancangan antarmuka halaman kelola data program kursus. Tampilan ini menunjukkan daftar program kursus dengan berbagai kolom: No, ID, Nama Kursus, Biaya, Pengajar, Jadwal, Gambar, dan Aksi. Terdapat tombol Print dan Tambah Kursus di bagian atas kanan, serta tombol Search dan tombol navigasi << 1 >> di bagian bawah kanan.

Gambar 3.62 Antarmuka Halaman Kelola Data Program Kursus

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman kelola data program kursus. Pada menu kelola data program kursus ini admin dapat mengubah, menghapus atau menambahkan data program kursus seperti jenis program kursus, hari, jam, pengajar, dan harga sebagai salah satu kelengkapan persyaratan pendaftaran peserta.

3.5.2.7 Rancangan antarmuka halaman kelola data pendaftaran

Rancangan antarmuka halaman kelola data pendaftaran. Tampilan ini menunjukkan daftar pendaftar dengan berbagai kolom: No, No Peserta, Nama, Program Kursus, Jenis Kelamin, Tanggal Daftar, Status, dan Aksi. Terdapat tombol Print dan Tambah Pendaftaran di bagian atas kanan, serta tombol Search dan tombol navigasi << 1 >> di bagian bawah kanan.

Gambar 3.63 Antarmuka Halaman Kelola Data Pendaftaran

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman kelola data pendaftaran. Pada menu kelola data pendaftaran ini berfungsi agar dapat melihat siapa saja yang telah mengisi data pendaftaran.

3.5.2.8 Rancangan antarmuka halaman kelola data pembayaran

Rancangan antarmuka halaman kelola data pembayaran. Tampilan ini menunjukkan daftar pembayaran dengan kolom-kolom: No, Nama Pendaftar, Program Kursus, Biaya Pendaftaran, Tanggal Pembayaran, Status, Bukti Pembayaran, dan Aksi. Terdapat tombol Print di bagian atas kanan, tombol Search di bagian tengah kanan, dan tombol << 1 >> di bagian bawah kanan.

No	Nama Pendaftar	Program Kursus	Biaya Pendaftaran	Tanggal Pembayaran	Status	Bukti Pembayaran	Aksi
<input type="checkbox"/>							<input type="radio"/>
<input type="checkbox"/>							<input type="radio"/>
<input type="checkbox"/>							<input type="radio"/>

Gambar 3.64 Antarmuka Halaman Kelola Data Pembayaran

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman kelola data pembayaran. Pada menu kelola pembayaran ini admin dapat melihat informasi mengenai status pembayaran dari peserta beserta informasi bukti pembayaran yang dilakukan oleh peserta.

3.5.2.9 Rancangan antarmuka halaman kelola data pengumuman

Rancangan antarmuka halaman kelola data pengumuman. Tampilan ini menunjukkan daftar pengumuman dengan kolom-kolom: No, Id Pendaftaran, Nama Pendaftar, Hasil, Kursus Penerima, Nilai Rata Rata, Nilai Tes, dan Aksi. Terdapat tombol Print di bagian atas kanan, tombol Search di bagian tengah kanan, dan tombol << 1 >> di bagian bawah kanan.

No	Id Pendaftaran	Nama Pendaftar	Hasil	Kursus Penerima	Nilai Rata Rata	Nilai Tes	Aksi
<input type="checkbox"/>							<input type="radio"/>
<input type="checkbox"/>							<input type="radio"/>
<input type="checkbox"/>							<input type="radio"/>

Gambar 3.65 Antarmuka Halaman Kelola Data Pengumuman

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman kelola data pengumuman. Pada menu kelola data pengumuman menampilkan informasi mengenai pengumuman hasil pendaftaran yang telah dilakukan oleh peserta.

3.5.2.10 Rancangan antarmuka detail pengguna

Detail Pengguna		Date
 Calon Peserta	Kirim Pesan	
Nama: <input type="text"/>	Pengaturan Profil Informasi Pribadi Nama <input type="text"/> Jenis Kelamin <input type="text"/> Tempat Lahir <input type="text"/> Tanggal Lahir <input type="text"/> Alamat <input type="text"/>	Pengaturan Kata Sandi <input type="text"/>
No. Hp: <input type="text"/>	Kontak Pribadi <input type="text"/>	<input type="text"/>
E-mail: <input type="text"/>	Sosial Meda <input type="text"/>	

Gambar 3.66 Antarmuka Halaman Detail Pengguna

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman detail pengguna. Pada menu tersebut berisikan mengenai informasi dari pengguna seperti profil, pengaturan profil, dan pengaturan kata sandi. Di pengaturan profil terdapat beberapa informasi yang dapat *diupdate* oleh pengguna misalnya data diri. Selain itu, dipengaturan kata sandi, admin dapat memperbaharui kata sandi pengguna apabila pengguna lupa *password*.

3.5.2.11 Rancangan antarmuka detail pendaftaran

Detail Pendaftaran

Date

Data Pendaftaran

PROFIL SISWA

ID Pendaftaran	<input type="text"/>
Nama	<input type="text"/>
Jenis kelamin	<input type="text"/>
Tempat tanggal lahir	<input type="text"/>
Agama	<input type="text"/>
NISN	<input type="text"/>
NIK	<input type="text"/>
Alamat	<input type="text"/>
Email	<input type="text"/>
Telepon/ Whatsapp	<input type="text"/>

DATA ORANG TUA

Nama ayah	<input type="text"/>	Nama ibu	<input type="text"/>
Pekerjaan ayah	<input type="text"/>	Pekerjaan ibu	<input type="text"/>
No. Handphone	<input type="text"/>	No. Handphone	<input type="text"/>
Gaji	<input type="text"/>	Tanggungan	<input type="text"/>
Dokumen	<input type="text"/>		

DATA REGISTRASI

Tahun	<input type="text"/>	Pilihan 1	<input type="text"/>
DATA PENDIDIKAN		Jurusan	<input type="text"/>
Nama sekolah	<input type="text"/>	Semester 1	<input type="text"/>
Semester 3	<input type="text"/>	Semester 2	<input type="text"/>
	<input type="text"/>	Semester 4	<input type="text"/>
	<input type="text"/>	Semester 5	<input type="text"/>

Selesai

Gambar 3.67 Antarmuka Halaman Detail Pendaftaran

3.5.2.12 Rancangan antarmuka detail pengumuman

Hasil Pengumuman	Date	<input type="text"/>																		
<input type="button" value="Print"/>																				
Pengumuman Pendaftaran Peserta Kursus Programming Course didaftarkan oleh... pada 2022-12-04 20:22:4		PROGRAMMING COURSE ROBLOX Jalan Sariashih IV. No.54, Sarijadi, Kota Bandung, Jawa Barat, 40151 Indonesia																		
<p>Data Pendaftar</p> <table> <tr> <td>No. Pendaftaran</td> <td><input type="text"/></td> </tr> <tr> <td>NISN Siswa</td> <td><input type="text"/></td> </tr> <tr> <td>Nama Siswa</td> <td><input type="text"/></td> </tr> <tr> <td>Asal Sekolah</td> <td><input type="text"/></td> </tr> <tr> <td colspan="2" style="text-align: right;"><input type="text"/> Pas Photo</td> </tr> </table> <p>Selamat! Anda dinyatakan LULUS sebagai peserta kursus di programming course roblox 2022/2023</p> <table> <tr> <td>Program Kursus Penerima:</td> <td><input type="text"/></td> </tr> <tr> <td>Hari:</td> <td><input type="text"/></td> </tr> <tr> <td>Jam:</td> <td><input type="text"/></td> </tr> <tr> <td>Pengajar:</td> <td><input type="text"/></td> </tr> </table>			No. Pendaftaran	<input type="text"/>	NISN Siswa	<input type="text"/>	Nama Siswa	<input type="text"/>	Asal Sekolah	<input type="text"/>	<input type="text"/> Pas Photo		Program Kursus Penerima:	<input type="text"/>	Hari:	<input type="text"/>	Jam:	<input type="text"/>	Pengajar:	<input type="text"/>
No. Pendaftaran	<input type="text"/>																			
NISN Siswa	<input type="text"/>																			
Nama Siswa	<input type="text"/>																			
Asal Sekolah	<input type="text"/>																			
<input type="text"/> Pas Photo																				
Program Kursus Penerima:	<input type="text"/>																			
Hari:	<input type="text"/>																			
Jam:	<input type="text"/>																			
Pengajar:	<input type="text"/>																			

Gambar 3.68 Antarmuka Halaman Detail Pengumuman

3.5.2.13 Rancangan antarmuka halaman *logout*

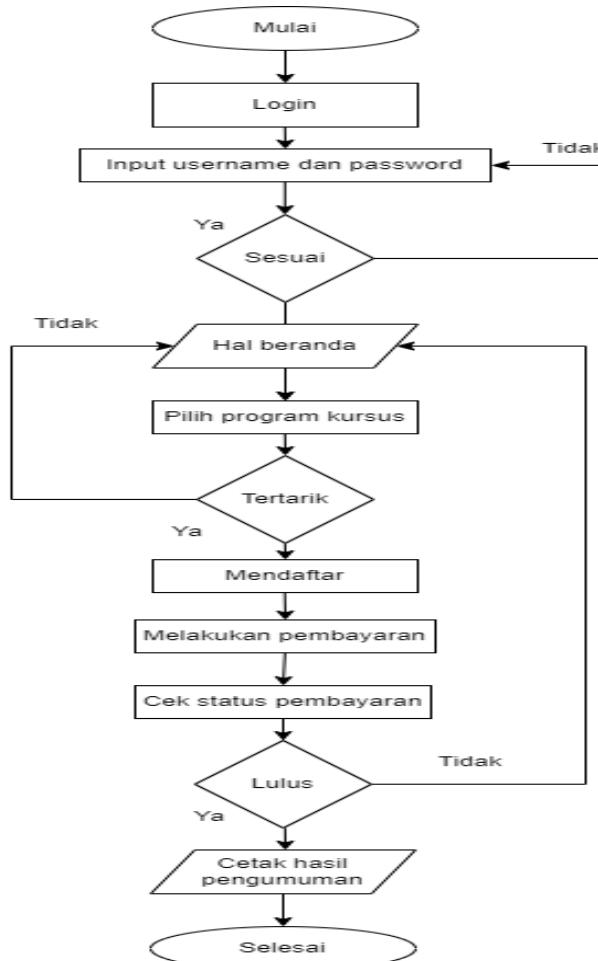
<input type="button" value="Daftar"/>	<input type="text" value="Username"/>
	<input type="text" value="Password"/>
	<input type="button" value="Masuk"/>

Gambar 3.69 Antarmuka Halaman *Logout*

Pada gambaran antarmuka diatas merupakan rancangan antarmuka halaman *logout*, yang mana apabila peserta dan admin mengklik menu *logout* maka aplikasi mengarahkan ke halaman utama (*register* dan *login*).

3.6 Logika Fungsi/ Algoritma Fungsi

Logika fungsi dan algoritma fungsi pada perancangan bertujuan untuk mengetahui algoritma dari masing masing fungsi dari aplikasi yang akan dibuat. Suatu algoritma menggunakan *flowchart* maupun *flowmap* untuk menggambarkan urutan atau langkah langkah yang berisi pernyataan dalam penulisan algoritma agar mengetahui *input* dan *output* oleh sistem. Pada pembahasan kali ini membahas tentang logika fungsi dari aplikasi *system kursus online* yakni *programming course roblox* yang sudah dibuat menggunakan *flowchart*. Adapun rangkaian *flowchart* secara umum pada alur berjalannya aplikasi sistem *programming course roblox* adalah sebagai berikut:



Gambar 3.70 Flowchart Aplikasi

Pada Gambar 3.70 terlihat secara umum alur dalam berjalannya sistem pendaftaran *programming course* roblox secara sederhana yaitu diawali dengan *start* atau mulai, kemudian *login* aplikasi dengan memasukkan *username* dan *password*. Terdapat sebuah simbol *decision tree* yang menentukan keputusan antara bernilai *true* atau *false*. Jika salah memasukkan *username* dan *password* saat *login* maka *output* nya akan berupa *input username* dan *password* kembali. Tetapi jika benar memasukkan *username* dan *password* saat *login* maka *output* nya yakni akan berupa tampilan halaman beranda. Di tampilan halaman beranda tersebut, pengguna disajikan dengan tampilan pilihan program kursus, ada terdapat 4 program kursus yakni program kursus 1 (*puzzle game*), program kursus 2 (*fighting game*), program kursus 3 (*sport game*), dan program kursus 4 (*education game*). Selain itu ditampilkan beranda ini juga disajikan berupa informasi mengenai kursus tersebut seperti jadwal, pengajar, harga, dan *link* demo contoh *game*. Selanjutnya pengguna dapat memilih program kursus jika tidak tertarik kembali ke halaman beranda tetapi jika tertarik maka *output* nya dapat melakukan tahap mendaftar. Setelah mendaftar, proses selanjutnya yakni melakukan pembayaran dengan memasukkan bukti pembayaran yang valid. Setelah melakukan pembayaran selesai, tahap selanjutnya yakni dengan mengecek status pengumuman antara lulus atau tidak lulus. Ditahap terakhir ini juga terdapat sebuah keputusan antara bernilai *true* atau *false*, artinya terdapat dua pilihan yakni jika tidak lulus maka kembali ke tampilan halaman beranda, jika statusnya lulus maka dapat mencetak hasil pengumuman tersebut.

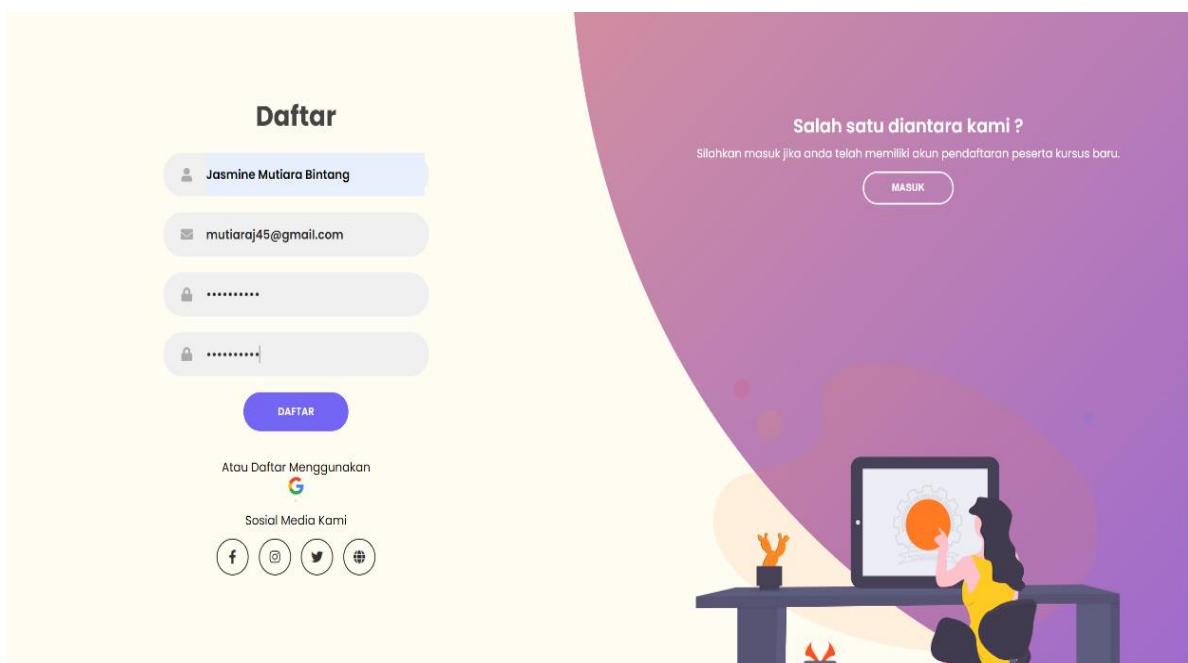
BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi dan Pembahasan Hasil Implementasi

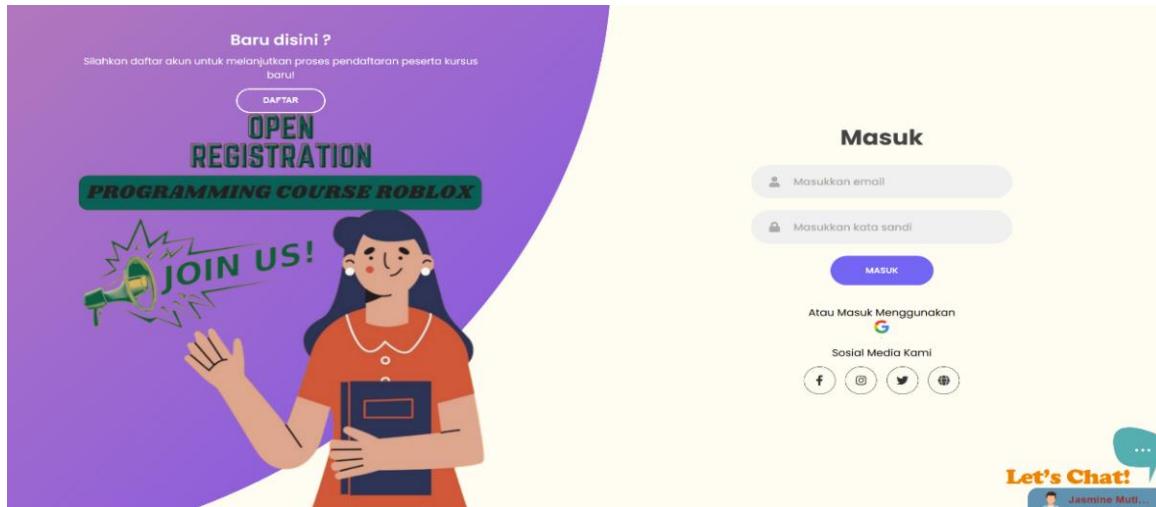
Implementasi merupakan tahap merealisasikan hasil analisis dan perancangan ke tahap penulisan *script* program sesuai rancangan yang telah dilakukan. Tahap pengujian dilakukan setelah semua proses perancangan selesai dilakukan. Hasil pengujian dapat dijadikan sebagai ukuran keberhasilan dalam mengimplementasikan hasil perancangan dengan mengidentifikasi dan melakukan analisa terhadap hasil pengujian aplikasi.

4.1.1 Implementasi Aplikasi Sistem Pendaftaran *Programming Course Roblox*



Gambar 4.1 Menu Daftar (*Register*)

Pada gambar 4.1 merupakan implementasi aplikasi sistem pendaftaran *programming course* roblox pada bagian rancangan menu *register*. Pada gambar menu *register*, masyarakat diminta untuk memasukkan nama, email, *password*, dan ulangi *password* jika belum memiliki akun. Apabila setelah memasukkan keempat tersebut, maka aplikasi memunculkan pesan “Data Tersimpan”.



Gambar 4.2 Menu Login

Pada gambar 4.2 merupakan implementasi aplikasi sistem pendaftaran *programming course* roblox pada bagian rancangan menu *login*. Pada gambar diatas menu *login* terdapat data masukan yang harus dimasukkan oleh peserta dan admin yakni data email, atau *username* dan data *password*. Peserta ataupun admin diminta *login* terlebih dahulu sebelum masuk ke halaman beranda/*dashboard*. Dibagian bawah sebelah kanan juga terdapat fitur *chat online* tawk.to yang dapat digunakan oleh peserta agar dapat berkomunikasi dengan admin.

Gambar 4.3 Menu Beranda (*Dashboard* Admin)

Pada gambar 4.3 merupakan implementasi aplikasi sistem pendaftaran *programming course* roblox pada bagian rancangan menu *dashboard* admin. Pada gambar diatas

terdapat antarmuka halaman admin yakni *dashboard* yang berisi informasi mengenai pendaftar, hasil seleksi pendaftar, dan mengenai linimasa pendaftar.

Nama	Jenis Kelamin	Telepon	Email	Aksi
Jasmine Mutiara Bintang	Perempuan	082363219861	admin@gmail.com	
Ratu SukmaKomala	Perempuan	085789902654	admin2@gmail.com	
Marjuniati Putri	Perempuan	085274889045	user@gmail.com	
Dafa Audya Pramana	Laki-Laki	085278554987	admin3@gmail.com	
Isnaenti Nur Latifah	Perempuan	083278996043	user2@gmail.com	
Lily Andira	Perempuan	0832165789776	lily@gmail.com	

Gambar 4.4 Menu Data Master (Pengguna)

Pada gambar 4.4 merupakan implementasi aplikasi *system* pendaftaran *programming course* roblox pada bagian rancangan menu pengguna. Pada gambar diatas terdapat halaman kelola data pengguna, dimana admin dapat melakukan *create*, *read*, *update*, dan *delete* data pengguna. Pada halaman menu pengguna juga terdapat *button* cetak, yang berfungsi untuk mencetak data data pengguna.

Gambar 4.5 Menu Detail Pengguna

Pada gambar 4.5 merupakan implementasi aplikasi sistem pendaftaran *programming course* roblox pada bagian rancangan menu detail pengguna. Pada gambar diatas terdapat antarmuka halaman detail pengguna, apabila admin mengklik *edit* di menu pengguna maka admin dapat melakukan perubahan pada detail data pengguna tersebut. Selain itu juga apabila kata sandi dari peserta lupa atau tidak bisa *login*, peserta dapat menghubungi atau *chat* admin, dan admin akan melakukan *setting* ulang pengaturan kata sandi dari peserta.

No	NPSN	Nama Sekolah	Alamat	Kota	Aksi
1	2303241	SMAN 3 Purwakarta	Jl. Sisingamaraja, No.15, Purwakarta	Purwakarta	
2	20219179	SMKS MUTIARA BANDUNG	Jl. Moleber Utara, No.37, Bandung	Bandung	
3	20219312	SMAN 4 BANDUNG	Jl. Gardujati, No.20, Bandung	Bandung	
4	20350979	SMKS GARUDA NUSANTARA	Jl. Raya Kuripan, No.89, Karangawen	Cimahi	
5	10210853	SMAN 2 MEDAN	Jl. Karang Sari, No.435, Kota Medan	Medan	
6	20217763	SMAN 2 KARAWANG	Jl. Manunggal, No. 7, Karawang	Karawang	
7	20223138	SMKN 1 KOTA BEKASI	Jl. Bintara, No. 8, Bekasi Barat	Bekasi	
8	20220273	SMKN 1 BOGOR	Jl. Heulang, No.6, Bogor	Bogor	
9	10204063	SMAN 1 LIMAPULUH	Jl. Lima Puluh, No, 5, Batu Bara	Limapuluh	Kirim pesan

Gambar 4.6 Menu Data Master (Sekolah)

Pada gambar 4.6 merupakan implementasi aplikasi sistem pendaftaran *programming course* roblox pada bagian rancangan menu data *master* (sekolah). Pada menu sekolah, admin dapat melakukan atau mengelola data data sekolah yang akan digunakan sebagai salah satu syarat yang akan dimasukkan peserta saat melakukan pendaftaran. Di menu sekolah, admin dapat melakukan tambah, *edit*, dan hapus sesuai dengan kebutuhan. Dimenu sekolah juga terdapat *button* cetak yang digunakan untuk mencetak data sekolah.

Gambar 4.7 Menu Data Master (Program Kursus)

Pada gambar 4.7 merupakan implementasi aplikasi sistem pendaftaran *programming course roblox* pada bagian rancangan menu data *master* (program kursus). Pada menu program kursus, admin dapat melakukan atau mengelola data data program kursus yang akan digunakan sebagai salah satu syarat yang akan dimasukkan peserta saat melakukan pendaftaran. Dimenu program kursus berisi informasi mengenai nama kursus, harga, pengajar, jadwal, gambar kursus, dan contoh *link demo game* tersebut. Admin dapat melakukan CRUD yakni (*create, update* dan *delete*) di menu kelola data program kursus tersebut.

Gambar 4.8 Menu Data History (Pendaftaran)

Pada gambar 4.8 merupakan implementasi aplikasi sistem pendaftaran *programming course roblox* pada bagian rancangan menu pendaftaran. Pada menu pendaftaran, admin dapat melakukan kelola data pendaftaran. Dimenu pendaftaran, admin dapat melihat informasi mengenai pengguna yang melakukan pendaftaran di aplikasi *programming course roblox*. Selain melakukan tambah, *delete*, dan *update* serta mencetak data pendaftaran, admin juga dapat memverifikasi status pendaftaran yang dilakukan oleh pengguna.

No	Nama Pendaftar	Program Kursus	Biaya Pendaftaran	Tanggal Pembayaran	Status	Aksi
1	Inayah Feuziah	PUZZLE GAME	Rp. 60.000	2023-01-22	Dibayar	

Gambar 4.9 Menu Data *History* (Pembayaran)

Pada gambar 4.9 merupakan implementasi aplikasi sistem pendaftaran *programming course roblox* pada bagian rancangan menu pembayaran. Pada menu pembayaran, admin dapat melihat informasi mengenai pengguna yang sudah melakukan pembayaran. Setelah pengguna melakukan pendaftaran pada langkah sebelumnya, maka admin memverifikasi pendaftaran jika setelah memverifikasi tahap selanjutnya yakni pada tahap ini pembayaran. Pengguna akan mendapatkan *notif* masing masing dan harus segera melakukan pembayaran. Admin mengelola pembayaran seperti memverifikasi status pembayaran, melakukan *update* data pembayaran, dan melakukan hapus data pembayaran. Tidak seperti pada rancangan kelola data pendaftaran, di menu pembayaran admin hanya dapat melakukan perintah *edit* data sedangkan untuk tambah (*insert*), dan hapus sudah secara

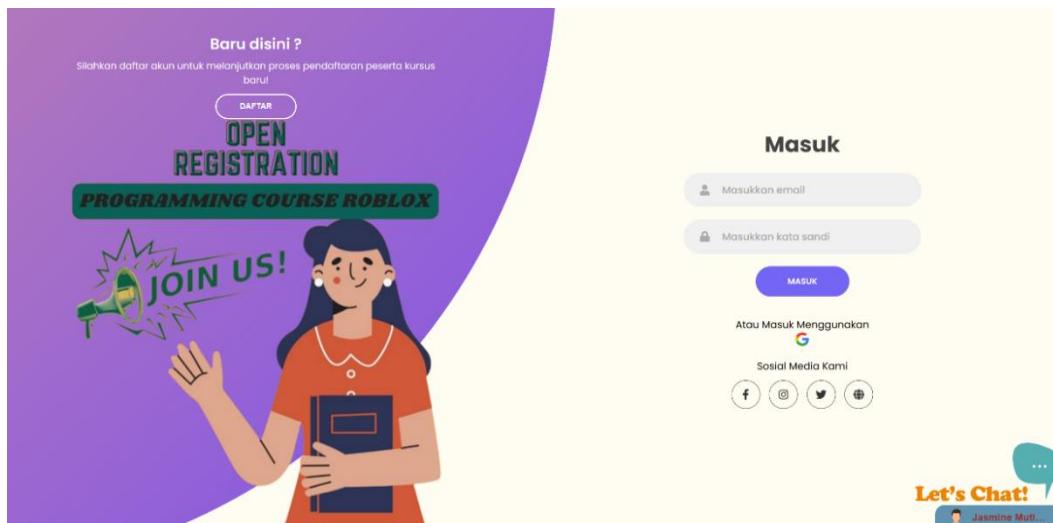
otomatis data dapat dihapus dimenu pendaftaran. Jadi apabila admin hapus data pendaftaran dan pengumuman maka otomatis terhapus.

Gambar 4.10 Menu Pengumuman

Pada gambar 4.10 merupakan implementasi aplikasi *system programming course* roblox pada bagian rancangan menu pengumuman. Pada menu pengumuman, admin dapat melakukan kelola data pengumuman seperti melakukan *update* sama halnya seperti pada menu kelola data pembayaran. Dimenu pengumuman, admin dapat melakukan verifikasi hasil pengumuman seperti lulus, tidak lulus, atau batal.

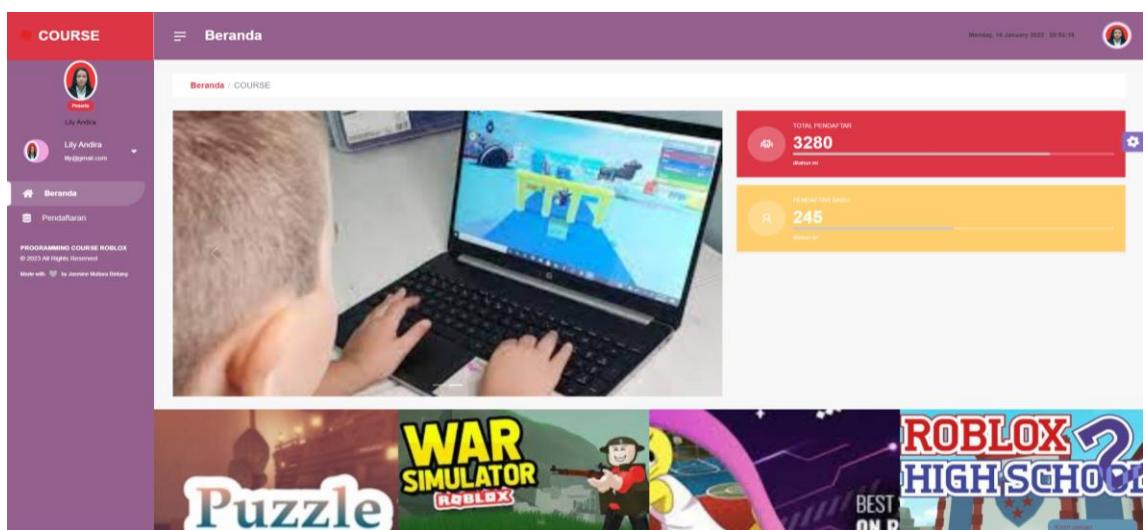
Gambar 4.11 Menu Profil Admin

Pada gambar 4.11 merupakan implementasi aplikasi *system programming course roblox* pada bagian rancangan menu profil admin. Pada menu profil, terdapat informasi data diri dari admin seperti nama, jenis kelamin, tanggal lahir, tempat lahir, alamat, kontak pribadi, email, maupun sosial media. Dimenu profil juga dapat melakukan *edit* seperti *edit* data pribadi, dan *edit* ganti kata sandi. Pada bagian kanan atas juga terdapat menu *logout*.



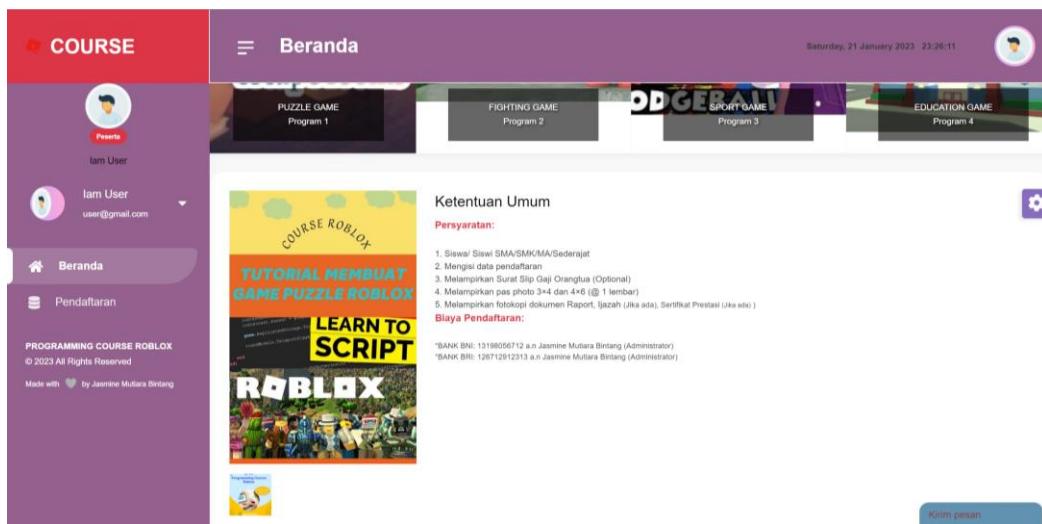
Gambar 4.12 Menu Logout

Pada gambar 4.12 merupakan halaman utama aplikasi sistem pendaftaran *programming course roblox*. Setelah melakukan klik *logout*, admin akan diarahkan keluar ke halaman utama aplikasi.

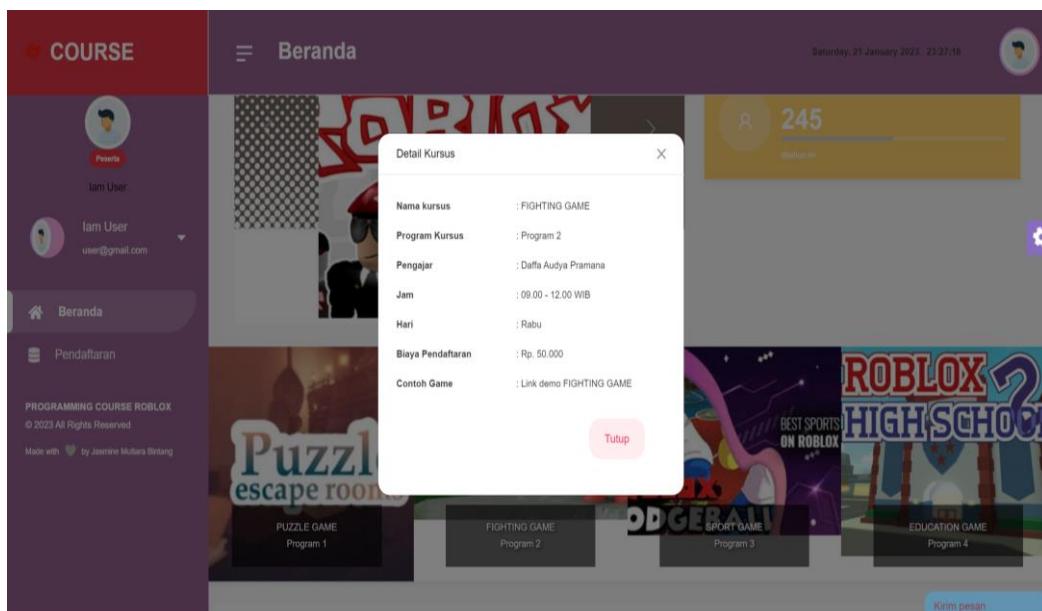


Gambar 4.13 (Dashboard Peserta)

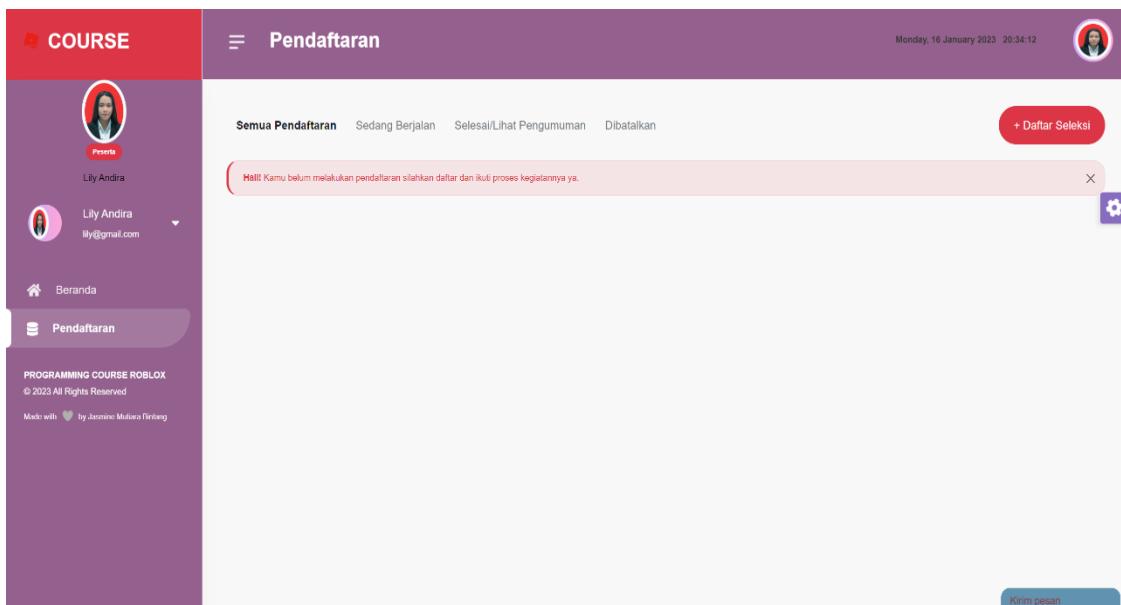
Pada gambar 4.13 merupakan implementasi aplikasi sistem pendaftaran *programming course roblox* pada bagian rancangan menu *dashboard* peserta. Pada halaman *dashboard* peserta terdapat informasi mengenai program kursus seperti: nama kursus, jadwal, harga, pengajar dan contoh *game*. Selain itu, informasi lain yang tersedia seperti ketentuan umum sebelum melakukan pendaftaran seperti: Ijazah, *photo*, *raport* dan lain sebagainya. Di menu *dashboard* ini, peserta juga dapat melihat contoh contoh *game* yang disediakan, jika peserta berminat dan menentukan pilihan program kursus yang diinginkan maka melanjutkan ketahap berikutnya yakni tahap menu pendaftaran.



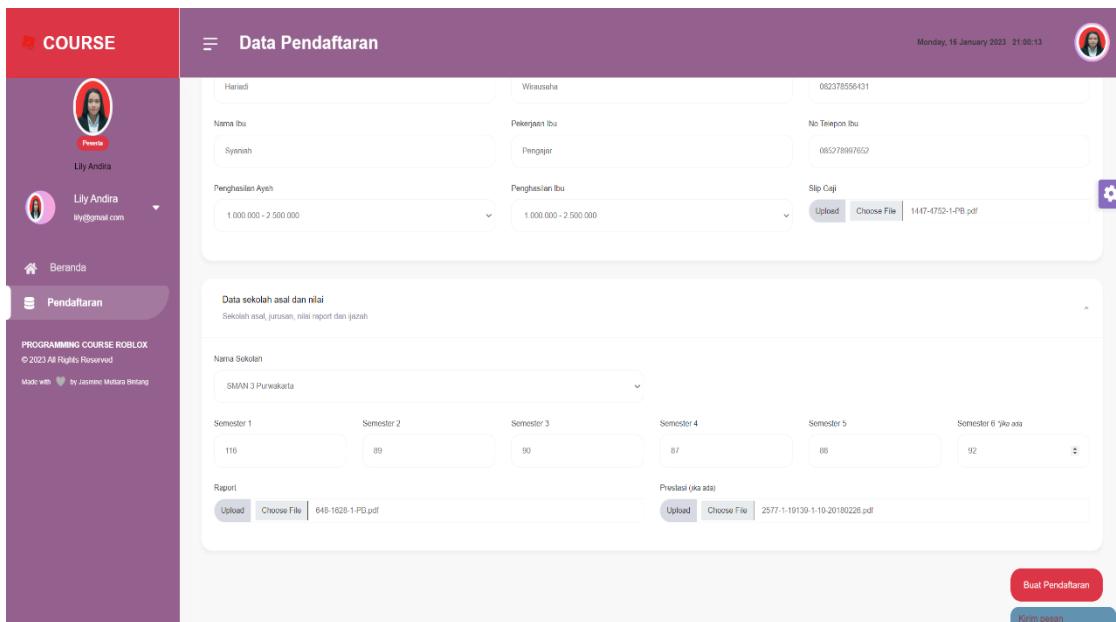
Gambar 4.14 Menu Beranda (*Dashboard* Peserta)



Gambar 4.15 Menu Beranda (*Dashboard* Peserta)

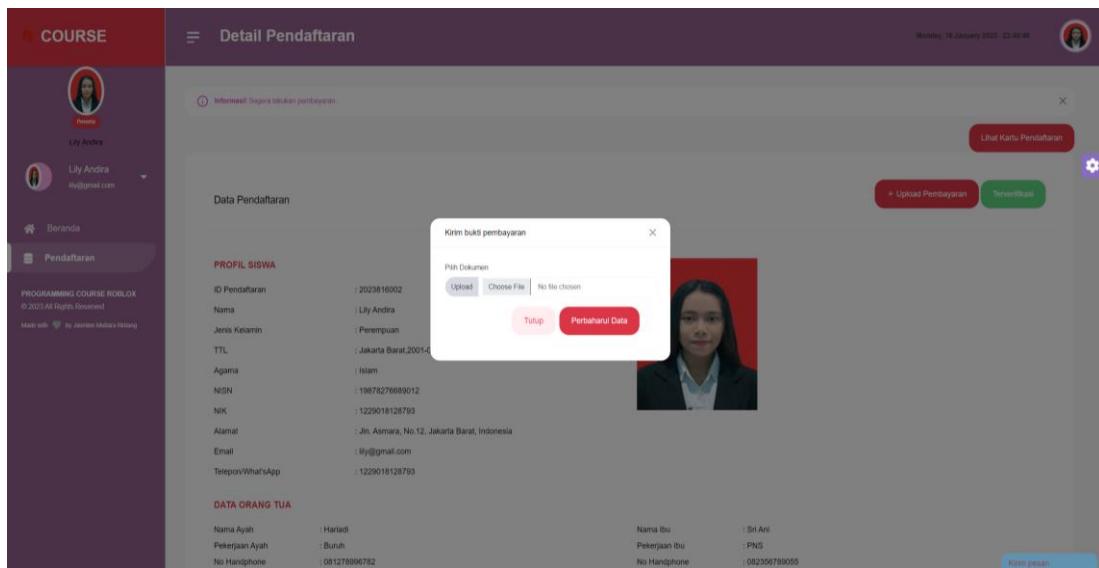


Gambar 4.16 Menu Pendaftaran



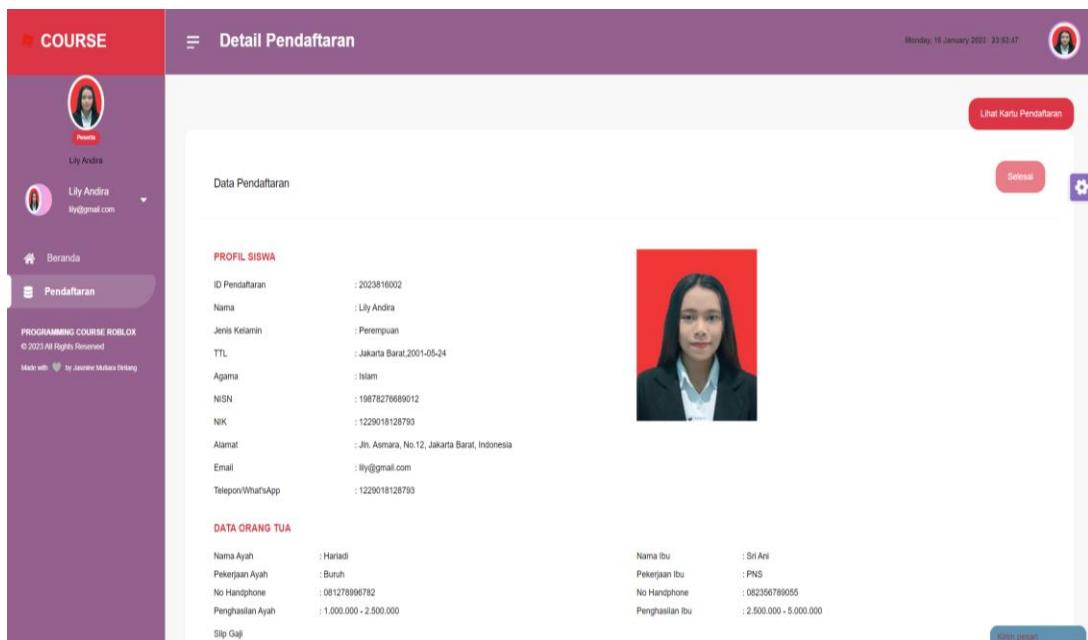
Gambar 4.17 Menu Pendaftaran

Pada gambar 4.17 merupakan implementasi aplikasi *system* pendaftaran *programming course roblox* pada bagian rancangan menu pendaftaran. Pada halaman menu pendaftaran ini, peserta dapat melakukan *input* persyaratan pendaftaran sesuai yang diminta seperti: data pribadi, data sekolah asal dan nilai, data orang tua, dan data pilihan program kursus. Setelah memasukkan persyaratan pendaftaran dengan lengkap selanjutnya mengklik *button* “buat pendaftaran”.



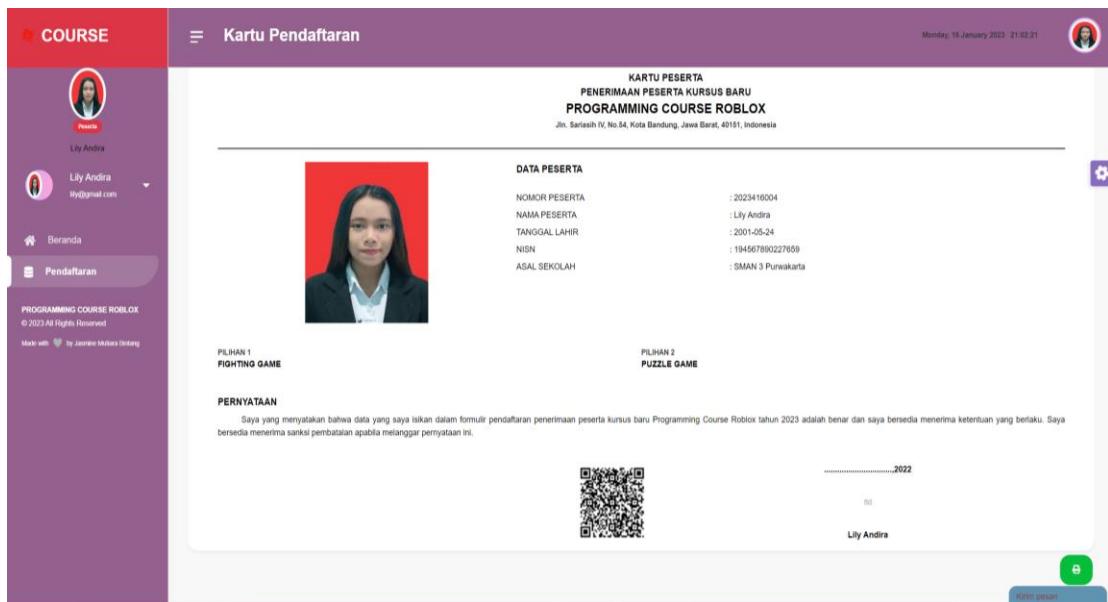
Gambar 4.18 Halaman Pembayaran Peserta

Pada gambar 4.18 merupakan implementasi aplikasi sistem pendaftaran *programming course roblox* pada bagian rancangan halaman pembayaran peserta. Pada tahap sebelumnya, peserta telah melakukan pendaftaran maka langkah berikutnya adalah melakukan pembayaran dengan mengklik “*upload* pembayaran”.

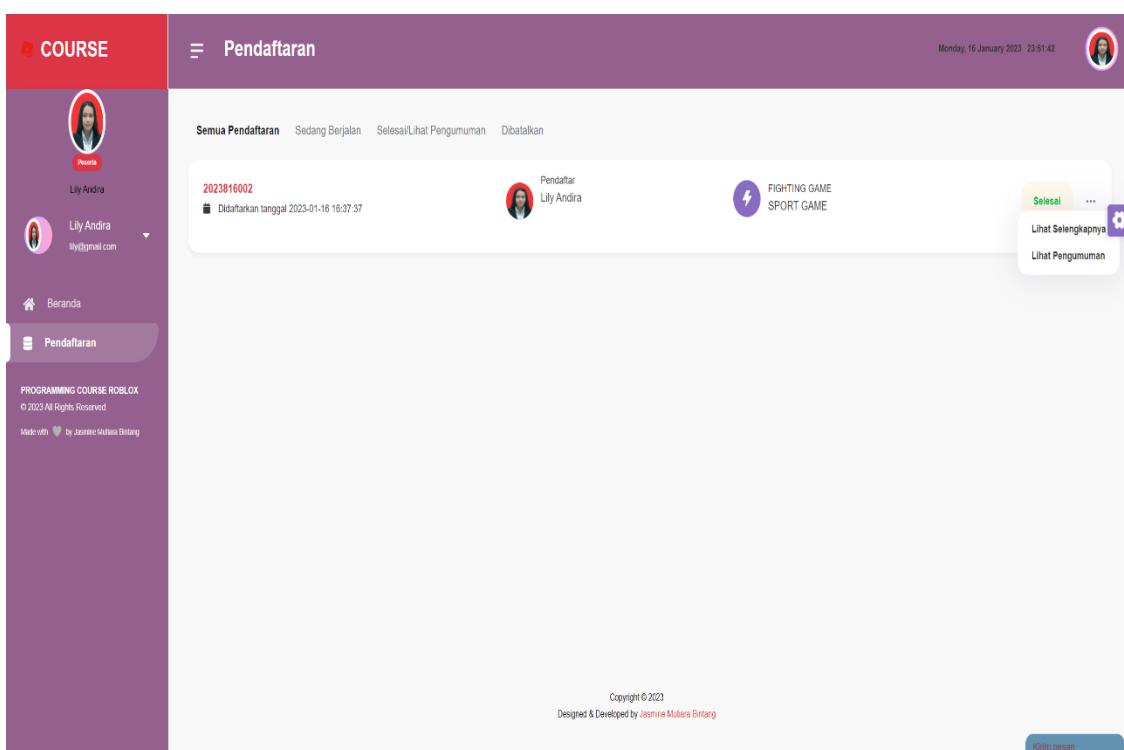


Gambar 4.19 Detail Pendaftaran Peserta

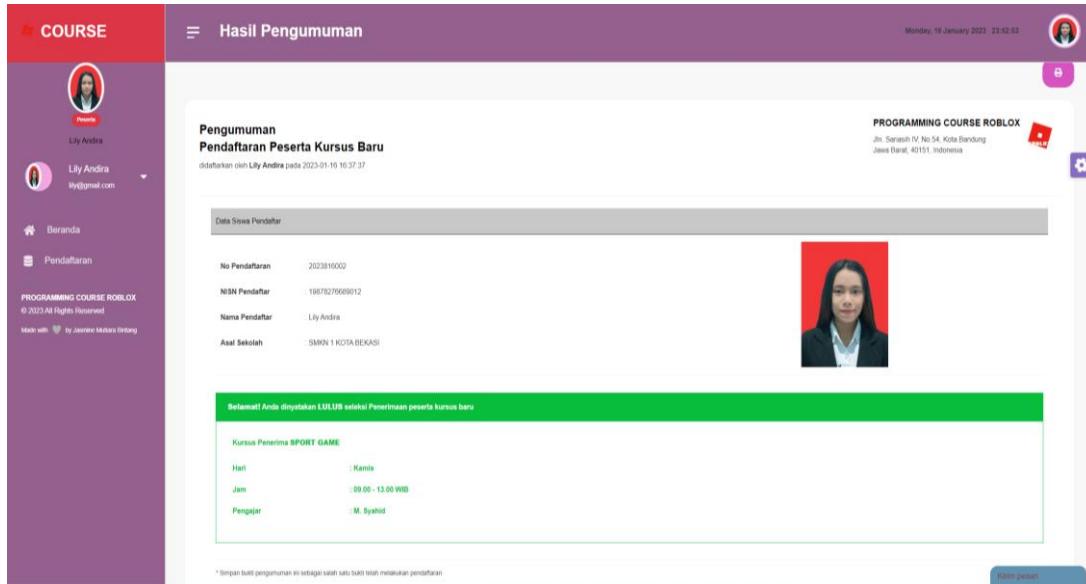
Pada gambar 4.19 merupakan implementasi aplikasi *system* pendaftaran *programming course* roblox pada bagian rancangan halaman detail pendaftaran peserta. Peserta dapat melihat informasi mengenai detail pendaftaran yang telah dilakukan ditahap sebelumnya. Selanjutnya juga dapat “lihat kartu pendaftaran” dan status pendaftaran yakni “selesai”.



Gambar 4.20 Halaman Kartu Pendaftaran Peserta

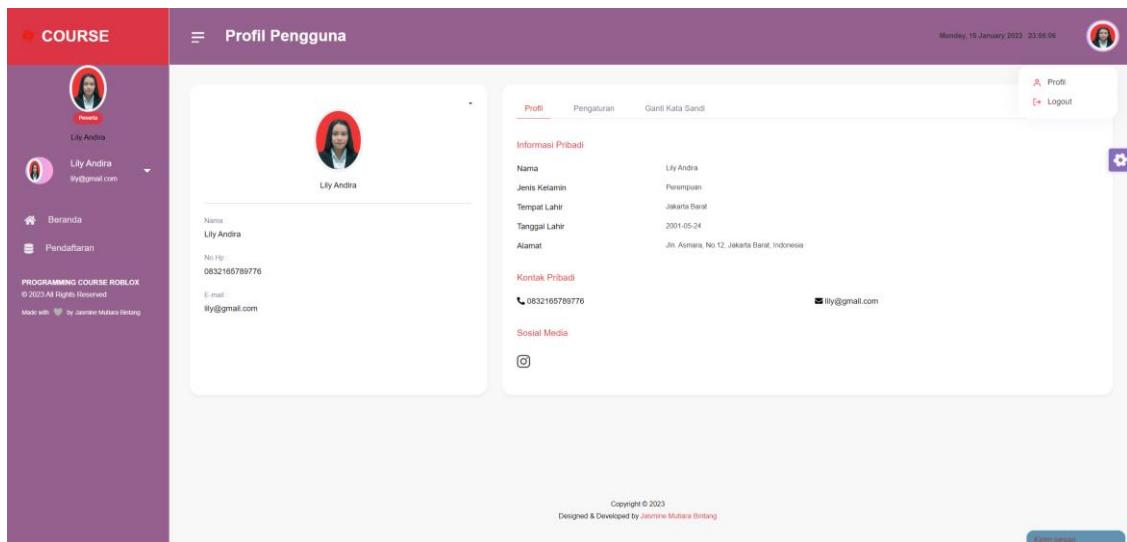


Gambar 4.21 Halaman Menu Lihat Pengumuman



Gambar 4.22 Halaman Pengumuman

Pada gambar 4.22 merupakan implementasi aplikasi sistem pendaftaran *programming course roblox* pada bagian rancangan menu pengumuman peserta. Pada halaman menu pengumuman ini, terdapat beberapa informasi mengenai hasil pengumuman dari peserta seperti: program kursus penerima, hari, jam, dan pengajarnya.



Gambar 4.23 Menu Profil Peserta

Pada gambar 4.23 merupakan implementasi aplikasi *system* pendaftaran *programming course roblox* pada bagian rancangan menu profil peserta. Pada halaman menu profil, terdapat beberapa informasi seperti data pribadi peserta. Selain itu, peserta dapat melakukan *edit* data pribadi dan *edit* kata sandi.

4.1.2 Implementasi *Source Code*

4.1.2.1 *Model*

Dalam folder *model* terdapat beberapa *file* didalamnya seperti: *User.php*, *pendaftaran.php*, *pembayaran.php*, *pengumuman.php*, *pengguna.php*, *sekolah.php*, dan *prodi.php*. Berikut ini adalah ulasan penjelasan singkat beserta *source code file* tersebut:

a. *User.php*

```
<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $primaryKey= "id";
    protected $fillable = [
        'name',
        'email',
        'password',
        'role',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];
}
```

```

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    /**
     * @return \Illuminate\Database\Eloquent\Relations\HasMany
     */
    public function socialAccounts()
    {
        return $this->hasMany(SocialAccount::class);
    }

    public function profile()
    {
        return $this->hasOne(ProfileUsers::class);
    }

    public function timeline(){
        return $this->hasMany(Timeline::class);
    }
}

```

Gambar 4.24 Source Code User.php

Pada gambar diatas merupakan *source code* data *user*. Pada *source code* data *user* terdapat beberapa atribut diantaranya yakni *nama*, *email*, *password*, *role*, dan *id_user*. Beberapa atribut diatas merupakan berasal dari tabel yakni tabel *user*.

b. Pendaftaran.php

```

public static function id()
{
    $data = DB::table('pendaftaran')->orderBy('id_pendaftaran', 'DESC')-
>first();
    $kodeakhir5 = substr($data->id_pendaftaran,-4);
    $kodeku= (int)$kodeakhir5;
    $addNol = '';
    $kode = (int)$kodeku + 1;

    if (strlen($kode) == 1) {
        $addNol = "0000";
    } elseif (strlen($kode) == 2) {

```

```

        $addNol = "000";
    } elseif (strlen($kode) == 3) {
        $addNol = "00";
    } elseif (strlen($kode) == 4) {
        $addNol = "0";
    }
    $kodeBaru = now()->format('Y').now()->format('m').$addNol.$kode;
    return $kodeBaru;
}

public function user()
{
    return $this->belongsTo(User::class, 'user_id');
}

public function pembayaran()
{
    return $this->hasMany(Pembayaran::class);
}

public function pengumuman()
{
    return $this->hasMany(pengumuman::class);
}

public function skolah(){
    return $this->belongsTo(Sekolah::class,'sekolah');
}

public function jadwal(){
    return $this->belongsTo(JadwalKegiatan::class,'gelombang');
}

public function pilihan1(){
    return $this->belongsTo(ProgramStudi::class,'pil1');
}

public function pilihan2(){
    return $this->belongsTo(ProgramStudi::class,'pil2');
}
}

```

Gambar 4.25 *Source Code* Pendaftaran.php

Pada gambar diatas merupakan *source code* pendaftaran. Beberapa atribut dalam tabel pendaftaran yakni: id_pendaftaran, id_user, nisn, nama_siswa, jenis_kelamin, pas_foto, tempat_lahir, tanggal_lahir, agama, alamat, email, nohp, pil1, nama_ayah,

nama_ibu, pekerjaan_ayah, pekerjaan_ibu, nohp_ayah, noibu_ibu, gaji, tanggungan, slip_gaji, kk, id_sekolah, smt1, smt2, smt3, smt4, smt5, smt6, prestasi, dan status_pendaftaran. Tabel berikut merupakan tabel pendaftaran, apabila peserta mengklik menu pendaftaran maka *system* akan *loading* data data yang ada didalam *database*.

c. Pembayaran.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use DB;

class Pembayaran extends Model
{
    use HasFactory;
    protected $table = "pembayaran";
    protected $fillable =
    ["id_pembayaran", "bukti_pembayaran", "status", "verifikasi", "tgl_pembayaran", "id_pendaftaran"];
    public $timestamps = false;
    public $incrementing = false;
    protected $primaryKey= "id";

    public function pendaftaran()
    {
        return $this->belongsTo(Pendaftaran::class, 'id_pendaftaran');
    }
    public static function id()
    {
        $data = DB::table('pembayaran')->orderBy('id_pendaftaran', 'DESC')->first();
        $kodeakhir5 = substr($data->id_pendaftaran,-3);
        $kodeku= (int)$kodeakhir5;
        $addNol = '';
        $kodetb = 'TAG';
        // $kode = str_replace($kodetb, "", $kode);
        $kode = (int)$kodeku + 1;
        $incrementKode = $kode;

        if (strlen($kode) == 1) {
```

```

        $addNol = "000";
    } elseif (strlen($kode) == 2) {
        $addNol = "00";
    } elseif (strlen($kode) == 3) {
        $addNol = "0";
    } elseif (strlen($kode) == 4) {
        $addNol = "";
    }
    $kodeBaru = $kodetb->now()->format('y').$addNol.$incrementKode;
    $id_pengumuman = 'ANN'.now()->format('y').$addNol.$incrementKode;
    return $kodeBaru;
}
}

```

Gambar 4.26 Source Code Pembayaran.php

Pada gambar diatas merupakan *source code* pembayaran. Pada *source code* pembayaran terdapat beberapa atribut diantaranya yakni: id_pembayaran, bukti_pembayaran, status_pembayaran, tgl_pembayaran, dan id_pendaftaran. Apabila peserta mengklik *button upload* pembayaran maka *system* akan *loading* informasi yang ada dalam tabel.

d. Pengumuman.php

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use DB;

class Pengumuman extends Model
{
    use HasFactory;
    protected $table = "pengumuman";
    protected $fillable =
    ["id_pengumuman","user_id","id_pendaftaran","hasil_seleksi","kursus_penerima","nila
    i_interview","nilai_test","status"];
    public $timestamps = false;
    public $incrementing = false;
    protected $primaryKey= "id";

    public static function id()
    {
        $data = DB::table('pembayaran')->orderBy('id_pendaftaran', 'DESC')->first();
        $kodeakhir5 = substr($data->id_pendaftaran, -3);
    }
}

```

```

$kodeku= (int)$kodeakhir5;
$addNol = '';
$kodetb = 'TAG';
//$kode = str_replace($kodetb,"", $kode);
$kode = (int)$kodeku + 1;
$incrementKode = $kode;

if (strlen($kode) == 1) {
    $addNol = "000";
} elseif (strlen($kode) == 2) {
    $addNol = "00";
} elseif (strlen($kode) == 3) {
    $addNol = "0";
} elseif (strlen($kode) == 4) {
    $addNol = "";
}
$kodeBaru = 'ANN'.now()->format('y').$addNol.$incrementKode;
return $kodeBaru;
}

public function pendaftaran()
{
    return $this->belongsTo(Pendaftaran::class, 'id_pendaftaran');
}
public function kursus()
{
    return $this->belongsTo(ProgramStudi::class, 'kursus_penerima');
}
public function user()
{
    return $this->belongsTo(\App\Models\User::class, 'user_id');
}
}

```

Gambar 4.27 *Source Code* Pengumuman.php

Pada gambar diatas merupakan *source code* pengumuman. Pada *source code* pengumuman terdapat beberapa atribut diantaranya yakni: id_pengumuman, id_pendaftaran, hasil_seleksi, prodi_penerima, nilai_rata_rata, nilai_kelengkapan data e. Pengguna.php

```

<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;

```

```

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $primaryKey= "id";
    protected $fillable = [
        'name',
        'email',
        'password',
        'role',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    /**
     * @return \Illuminate\Database\Eloquent\Relations\HasMany
     */
    public function socialAccounts()
    {
        return $this->hasMany(SocialAccount::class);
    }
}

```

```

    }
    public function profile()
    {
        return $this->hasOne(ProfileUsers::class);
    }

    public function timeline(){
        return $this->hasMany(Timeline::class);
    }
}

```

Gambar 4.28 Source Code Pengguna.php

Pada gambar diatas merupakan *source code* pengguna.php. Fungsi tabel pengguna ini yakni untuk memunculkan data data informasi mengenai pengguna yang terdaftar pada aplikasi *system* pendaftaran *programming course* tersebut. Pada *source code* pengguna terdapat beberapa tabel dan atributnya diantaranya yakni: *id_user*, *nama*, *email*, *password*, dan *role*.

f. Sekolah.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Sekolah extends Model
{
    use HasFactory;
    protected $table = "sekolah";
    protected $primaryKey= "npsn";
    protected $fillable = ["npsn","nama_sekolah","alamat","kota"];
    public $timestamps = false;
    public $incrementing = false;

    public function pendaftaran(){
        return $this->hasMany(Pendaftaran::class);
    }
}

```

Gambar 4.29 Source Code Sekolah.php

Pada gambar diatas merupakan *source code* sekolah.php. Terdapat tabel yang ada didalam *model* sekolah.php yakni tabel sekolah. Fungsi tabel sekolah ini yakni untuk memunculkan data data informasi tentang sekolah yang tersedia untuk di menu pendaftaran. Dengan kata lain *source code* sekolah.php ini berisi tabel sekolah, lalu data sekolah ini akan dimunculkan juga di menu pendaftaran. Pada *source code*

sekolah.php ini juga terdapat beberapa atributnya diantaranya yakni: NPSN, nama_sekolah, alamat, dan kota.

g. Prodi.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Support\Facades\DB;

class ProgramStudi extends Model
{
    use HasFactory;
    protected $table = "program_kursus";
    protected $primaryKey= "id";
    protected $fillable =
    ["id_kursus","nama_kursus","jenjang_kursus","foto_kursus","pengajar","jam","hari",
    "harga_kursus","contoh_game"];
    public $timestamps = false;
    public $incrementing = false;

    public static function id()
    {
        $kode = DB::table('program_kursus')->max('id');
        $addNol = '';
        $kode = str_replace("KRS", "", $kode);
        $kode = (int) $kode + 1;
        $incrementKode = $kode;

        if (strlen($kode) == 1) {
            $addNol = "00";
        } elseif (strlen($kode) == 2) {
            $addNol = "0";
        }
        $kodeBaru = "KRS".$addNol.$incrementKode;
        return $kodeBaru;
    }

    public function pengumuman()
    {
        return $this->belongsTo(Pengumuman::class);
    }
}
```

```

public function pendaftaran(){
    return $this->hasMany(Pendaftaran::class);
}
}

```

Gambar 4.30 *Source Code* Prodi.php

Pada gambar diatas merupakan *source code* dari *script* prodi.php. Terdapat tabel yang ada didalam *file* prodi.php yakni tabel prodi. Prodi disini dimaksudkan adalah tabel program kursus, dimana berisi data data mengenai atribut yang ada didalam tabel program kursus. Tabel prodi ini merupakan tabel yang akan dimunculkan di halaman pendaftaran, sebagai syarat kelengkapan data pendaftaran. Tabel prodi memiliki atribut didalamnya yakni: id_prodi, nama_kursus, hari, jam, pengajar, dan harga.

4.1.2.2 View

a. Register.php

```

<h2 class="title">Daftar</h2>
<div class="input-field">
<i class="fas fa-user"></i>
<input type="text" placeholder="Nama Lengkap" name="name" value="{{ old('name') }}"
autocomplete='off' />
</div>
<div class="input-field">
<i class="fas fa-envelope"></i>
<input type="email" placeholder="Email" name="email"
value="{{ old('email') }}" autocomplete='off' />
</div>
<div class="input-field">
<i class="fas fa-lock"></i>
<input type="password" placeholder="Kata Sandi"
name="password" autocomplete='off' />
</div>
<div class="input-field">
<i class="fas fa-lock"></i>
<input type="password" placeholder="Ulangi Kata Sandi" name="password_confirmation"
autocomplete='off' />
</div>
<input type="submit" class="btn" value="DAFTAR" />
<br>

```

Gambar 4.31 *Source Code* Register.php

Pada gambar diatas merupakan *source code* dari *script register.php*. Masyarakat atau pengguna baru apabila ingin *register* atau mendaftar akun maka harus mengisi nama lengkap, email, kata sandi, dan masukan ulang kata sandi. Apabila nama lengkap, email, dan kata sandi tidak diisi secara lengkap, maka akan memunculkan `{ $message }` yang berarti pesan *error* atau “*alert alert-warning*” yang menandakan *register* harus diisi secara lengkap.

b. *Login.php*

```
<h2 class="title">Masuk</h2>
    <div class="input-field">
        <i class="fas fa-user"></i>
        <input type="text" placeholder="Masukkan email"
name="email" value="{{ old('email') }}" autocomplete='off' />

    </div>
    <div class="input-field">
        <i class="fas fa-lock"></i>
        <input type="password" placeholder="Masukkan kata sandi"
name="password" autocomplete='off'/>

    </div>
    <input type="submit" value="MASUK" class="btn solid" />

<br>
<p class="text-center">Atau Masuk Menggunakan</p>
<div class="google-btn">
    <div class="google-icon-wrapper">
        <a href="/auth/google">
            
        </a>
    </div>
</div>
<hr>
<p class="social-text">Sosial Media Kami</p>
<div class="social-media">
    <a href="#" class="social-icon">
        <i class="fab fa-facebook-f"></i>
    </a>
    <a href="#" class="social-icon">
        <i class="fab fa-instagram"></i>
    </a>
    <a href="#" class="social-icon">
        <i class="fab fa-twitter"></i>
    </a>
</div>
```

```

        </a>
        <a href="#" class="social-icon">
            <i class="fas fa-globe"></i>
        </a>
    </div>
</form>

<form method="POST" action="/register" class="sign-up">
    @csrf
    @if ($session()->has('loginError'))
        <div class="alert alert-danger alert-dismissible fade show">
            <svg viewBox="0 -6 24 24" width="24" height="24" stroke="currentColor" stroke-width="2" fill="none" stroke-linecap="round" stroke-linejoin="round" class="me-2"><polygon points="7.86 2 16.14 2 22 7.86 22 16.14 16.14 22 7.86 22 2 16.14 2 7.86 7.86 2"></polygon><line x1="15" y1="9" x2="9" y2="15"></line><line x1="9" y1="9" x2="15" y2="15"></line></svg>
            <strong>Peringatan!</strong> {{session('loginError')}}
        </div>
    @endif
    @error('name')
        <div class="alert alert-warning alert-dismissible fade show">
            <svg viewBox="0 -6 24 24" width="24" height="24" stroke="currentColor" stroke-width="2" fill="none" stroke-linecap="round" stroke-linejoin="round" class="me-2"><path d="M10.29 3.86L1.82 18a2 2 0 0 0 1.71 3h16.94a2 2 0 0 0 1.71-3L13.71 3.86a2 2 0 0 0-3.42 0z"></path><line x1="12" y1="9" x2="12" y2="13"></line><line x1="12" y1="17" x2="12.01" y2="17"></line></svg>
            <strong>Peringatan!</strong> {{ $message }}
        </div>
    @enderror
    @error('email')
        <div class="alert alert-warning alert-dismissible fade show">
            <svg viewBox="0 -6 24 24" width="24" height="24" stroke="currentColor" stroke-width="2" fill="none" stroke-linecap="round" stroke-linejoin="round" class="me-2"><path d="M10.29 3.86L1.82 18a2 2 0 0 0 1.71 3h16.94a2 2 0 0 0 1.71-3L13.71 3.86a2 2 0 0 0-3.42 0z"></path><line x1="12" y1="9" x2="12" y2="13"></line><line x1="12" y1="17" x2="12.01" y2="17"></line></svg>
            <strong>Peringatan!</strong> {{ $message }}
        </div>
    @enderror
    @error('password')
        <div class="alert alert-warning alert-dismissible fade show">

```

```

        <svg viewBox="0 -6 24 24" width="24" height="24"
stroke="currentColor" stroke-width="2" fill="none" stroke-linecap="round" stroke-
linejoin="round" class="me-2"><path d="M10.29 3.86L1.82 18a2 2 0 0 0 1.71 3h16.94a2
2 0 0 1.71-3L13.71 3.86a2 2 0 0 0-3.42 0z"></path><line x1="12" y1="9" x2="12"
y2="13"></line><line x1="12" y1="17" x2="12.01" y2="17"></line></svg>
        <strong>Peringatan!</strong> {{ $message }}</div>
    @enderror
    @error('password_confirmation')
        <div class="alert alert-warning alert-dismissible fade
show">
            <svg viewBox="0 -6 24 24" width="24" height="24"
stroke="currentColor" stroke-width="2" fill="none" stroke-linecap="round" stroke-
linejoin="round" class="me-2"><path d="M10.29 3.86L1.82 18a2 2 0 0 0 1.71 3h16.94a2
2 0 0 1.71-3L13.71 3.86a2 2 0 0 0-3.42 0z"></path><line x1="12" y1="9" x2="12"
y2="13"></line><line x1="12" y1="17" x2="12.01" y2="17"></line></svg>
            <strong>Peringatan!</strong> {{ $message }}</div>
        @enderror
        <h2 class="title">Daftar</h2>
        <div class="input-field">
            <i class="fas fa-user"></i>
            <input type="text" placeholder="Nama Lengkap" name="name"
value="{{ old('name') }}" autocomplete='off' />
        </div>
        <div class="input-field">
            <i class="fas fa-envelope"></i>
            <input type="email" placeholder="Email" name="email"
value="{{ old('email') }}" autocomplete='off' />
        </div>
        <div class="input-field">
            <i class="fas fa-lock"></i>
<input type="password" placeholder="Kata Sandi" name="password" autocomplete='off'
/>
        </div>
        <div class="input-field">
            <i class="fas fa-lock"></i>
            <input type="password" placeholder="Ulangi Kata Sandi"
name="password_confirmation" autocomplete='off' />
        </div>
        <input type="submit" class="btn" value="DAFTAR" />
    <br>
    <p class="text-center">Atau Daftar Menggunakan</p>
    <div class="google-btn">
        <div class="google-icon-wrapper">
            <a href="/auth/google">

```

```

        
    </a>
</div>
</div>
<hr>
<p class="social-text">Sosial Media Kami</p>
<div class="social-media">
    <a href="#" class="social-icon">
        <i class="fab fa-facebook-f"></i>
    </a>
    <a href="#" class="social-icon">
        <i class="fab fa-instagram"></i>
    </a>
    <a href="#" class="social-icon">
        <i class="fab fa-twitter"></i>
    </a>
    <a href="#" class="social-icon">
        <i class="fas fa-globe"></i>
    </a>
</div>
</form>
</div>
</div>

<div class="panels-container">
    <div class="panel left-panel">
        <div class="content">
            <h3>Baru disini ?</h3>
<p>Silahkan daftar akun untuk melanjutkan proses pendaftaran peserta kursus baru
</p>
            <button class="btn transparent" id="sign-up-btn">
                DAFTAR
            </button>
        </div>
        
    </div>
    <div class="panel right-panel">
        <div class="content">
            <h3>Salah satu diantara kami ?</h3>
<p>Silahkan masuk jika anda telah memiliki akun pendaftaran peserta kursus baru.
</p>
            <button class="btn transparent" id="sign-in-btn">Masuk</button>
        </div>
        
    </div>

```

```

        </div>
    </div>

```

Gambar 4.32 Source Code Login.php

Pada gambar diatas merupakan *source code* dari *script login.php*. Peserta atau pengguna baru apabila ingin masuk ke aplikasi maka harus *login* terlebih dahulu dengan memasukkan *username* dan *password*. Apabila *username* dan kata sandi tidak diisi secara lengkap, maka akan memunculkan {{ \$message }} yang berarti pesan *error* atau “*alert alert-warning*” yang menandakan *form login* harus diisi secara lengkap.

c. Data-pendaftaran.php

```

<div class="row">
    <div class="col-lg-4">
        <div class="mb-3 mb-4">
            <label class="form-label" for="personal-data-name">Nama</label>
            @if ( auth()->user()->profile->nama != null)
                <input type="text" class="form-control" id="basicpill" name="nama"
                    placeholder="Masukkan Nama Lengkap" value="{{ auth()->user()->profile->nama }}"
                    required>
            @else
                <input type="text" class="form-control" id="personal-data-name"
                    name="nama" placeholder="Masukkan Nama Lengkap"
                    value="{{ old('nama') }}" required>
            @endif
            @error('nama')
                <div class="alert alert-warning" role="alert">
                    <strong>Peringatan!</strong>
                    {{ $message }}
                </div>
            @enderror
        </div>
        <div class="col-lg-4">
            <div class="mb-3 mb-4">
                <label class="form-label" for="personal-data-gender">Jenis Kelamin</label>
                @if (auth()->user()->profile->gender != null)
                    @if (auth()->user()->profile->gender == 'Perempuan')
                        <select class="form-control wide" name="jk"
                            value="{{ old('jk') }}"
                            >
                            <option value="{{ auth()->user()->profile->gender }}" selected>
                                {{ auth()->user()->profile->gender }}</option>
                            <option value="Laki-laki">Laki-laki</option>
                        </select>
                    @else

```

```

<select class="form-control wide" name="jk"
        value="{{ old('jk') }}">
<option value="{{ auth()->user()->profile->gender }}" selected>
{{ auth()->user()->profile->gender }}</option>
<option value="Perempuan">Perempuan</option>
</select>
@endif
@else
<select class="form-control wide" name="jk"
<option value="{{ old('jk') }}" disabled selected>Pilih Jenis Kelamin </option>

<option value="Laki-laki">Laki-aki</option>

<option value="Perempuan">Perempuan</option>

</select>

@endif

@error('jk')
<div class="alert alert-warning" role="alert">
<strong>Peringatan!</strong>
{{ $message }}
</div>
enderror
</div>
</div>
<div class="col-lg-4">
<label class="form-label" for="personal-data">Agama</label>
<select class="form-control wide" name="agama"

value="{{ old('agama') }}>

<option value="{{ old('agama') }}" disabled selected>Pilih agama

</option>

<option value="Islam">Islam</option>

<option value="Kristen">Kristen</option>

<option value="Hindu">Hindu</option>

<option value="Budha">Budha</option>

<option value="Kong Hu Chu ">Kong Hu Chu</option>

```

```

<option value="Lainnya">Etc</option>

</select>

@error('agama')

<div class="alert alert-warning" role="alert">

<strong>Peringatan!</strong>

{{ $message }}

</div>

@enderror

</div>

</div>
</div>
<div class="row">
<div class="col-lg-4">
<div class="mb-4 mb-lg-0">
<label class="form-label">Tempat lahir</label>
@if (auth()->user()->profile->tempat_lahir != null)
<input type="text" class="form-control" id="basicpill"
name="tempatlahir" placeholder="Masukkan Tempat Lahir"
value="{{ auth()->user()->profile->tempat_lahir }}" required>
@else
<input type="text" class="form-control" id="basicpill"
name="tempatlahir" placeholder="Masukkan Tempat Lahir"
value="{{ old('tempatlahir') }}" required>
@endif
@error('tempatlahir')
<div class="alert alert-warning" role="alert">
<strong>Peringatan!</strong>
{{ $message }}
</div>
@enderror
</div>
</div>
<div class="col-lg-4">
<div class="mb-4 mb-lg-0">
<label class="form-label" for="billing-city">Tanggal lahir</label>
@if (auth()->user()->profile->tanggal_lahir != null)
<input type="date" class="form-control" id="basicpill"
name="tanggallahir" placeholder="Masukkan Tanggal Lahir"
value="{{ auth()->user()->profile->tanggal_lahir }}"
required>
@endif
@error('tanggal_lahir')
<div class="alert alert-warning" role="alert">
<strong>Peringatan!</strong>
{{ $message }}
</div>
@enderror
</div>
</div>

```

```

value="{{ auth()->user()->profile->tanggal_lahir }}" required>
    @else
        <input type="date" class="form-control" id="basicpill"
            name="tanggallahir" placeholder="Masukkan Tanggal Lahir"
            old('tanggallahir') }> required>
    @endif
    @error('tanggallahir')
        <div class="alert alert-warning" role="alert">
            <strong>Peringatan!</strong>
            {{ $message }}
        </div>
    @enderror
</div>

<div class="col-lg-4">
    <div class="mb-0">
        <label class="form-label" for="zip-code">Pas Photo</label>
        <div class="input-group">
            <span class="input-group-text">Upload</span>
            <div class="form-file">
                <input type="file" class="form-file-input form-control"
                    name="foto" value="{{ old('foto') }}" accept="image/png, image/jpg, image/jpeg"
                    required>
            </div>
        </div>
        @error('foto')
            <div class="alert alert-warning" role="alert">
                <strong>Peringatan!</strong>
                {{ $message }}
            </div>
        @enderror
    </div>
    <div class="mb-4">
        <label class="form-label" for="billing-address">Alamat</label>

        @if (auth()->user()->profile->alamat != null)
            <textarea class="form-control" id="billing-address" rows="3" name="alamat"
            required placeholder="Masukkan alamat lengkap">{{ auth()->user()->profile->alamat
            }}</textarea>
        @else
            <textarea class="form-control" id="billing-address" rows="3" name="alamat"
            required
            placeholder="Masukkan alamat lengkap">{{ old('alamat') }}</textarea>
        @endif
        @error('alamat')
    
```

```

<div class="alert alert-warning" role="alert">
    <strong>Peringatan!</strong>
    {{ $message }}
</div>
    @enderror
</div>
<div class="row">
    <div class="col-lg-6">
        <div class="mb-3 mb-4">
<label class="form-label" for="personal-data-nisn">Email</label>
<input type="email" class="form-control" id="personal-data-nisn"
        name="email" placeholder="Masukkan email"
        value="{{ auth()->user()->email }}" required>
        @error('email')
            <div class="alert alert-warning" role="alert">
<strong>Peringatan!</strong>
{{ $message }}
            </div>
        @enderror
    </div>
</div>
<div class="col-lg-6">
<div class="mb-3 mb-4">
<label class="form-label" for="personal-data-nik">No
Hp/WhatsApp</label>
@if (auth()->user()->profile->no_hp != null)
<input type="number" class="form-control" id="basicpill" name="nohp"
placeholder="Masukkan Tanggal Lahir" value="{{ auth()->user()->profile->no_hp }}" required>
@else
<input type="number" class="form-control" id="basicpill" name="nohp"
placeholder="Masukkan nomor telepon" value="{{ old('nohp') }}" required>
@endif
@error('nohp')
<div class="alert alert-warning" role="alert">
<strong>Peringatan!</strong>
{{ $message }}
</div>
@enderror
</div>
</div>
</div>
</div>
</div>
<div class="card">
<a href="#registration-data" class="collapsed text-dark" data-bs-toggle="collapse">

```

```

<div class="p-4">
<div class="d-flex align-items-center">
<div class="flex-shrink-0 me-3" style="font-size: 24px; color: #007bff; font-weight: bold;"><i class="uil uil-truck text-primary h2"></i>
```

```

        <div class="flex-shrink-0 me-3"> <i class="uil uil-bill text-primary h2"></i>
        </div>
<div class="flex-grow-1 overflow-hidden">
    <h5 class="font-size-16 mb-1">Data Orang Tua</h5>
    <p class="text-muted text-truncate mb-0">Data orang tua, keuangan dan data.</p>
</div>
<div class="mdi mdi-chevron-up accor-down-icon font-size-24"></i> </div>
</div>
</div>
</a>
<div id="parental-data" class="collapse">
    <div class="p-4 border-top">
        <div class="row">
            <div class="col-lg-4">
                <div class="mb-3 mb-4">
                    <label class="form-label"
for="personal-data-name">Nama Ayah</label>
<input type="text" class="form-control" id="personal-data-name"
name="ayah" placeholder="Masukkan Nama Ayah"
value="{{ old('ayah') }}" required
@error('ayah')
<div class="alert alert-warning" role="alert">
    <strong>Peringatan!</strong>
        {{ $message }}
</div>
@enderror
</div>
</div class="col-lg-4">
                <div class="mb-3 mb-4">
                    <label class="form-label"
for="personal-data-gender">Pekerjaan Ayah</label>
<input class="form-control" list="datalistOptionsOccupation"
id="exampleDataList" placeholder="Masukkan Jenis Pekerjaan..."
name="pekerjaanayah" value="{{ old('pekerjaanayah') }}" required>
<datalist id="datalistOptionsOccupation">
    <option value="Karyawan Swasta"></option>
    <option value="Karyawan BUMN"></option>
    <option value="Karyawan BUMD"></option>
    <option value="Karyawan Honorer"></option>
    <option value="PNS"></option>
    <option value="Wirausaha"></option>
    <option value="PNS"></option>
    <option value="Buruh"></option>

```

```

<option value="Asisten Rumah Tangga"></option>
<option value="Dokter"></option>
<option value="Perawat"></option>
</div>
<div class="col-lg-4">
    <div class="mb-3 mb-4">
<label class="form-label" for="personal-data-nik">No HP Ayah</label>
<input type="number" class="form-control" id="personal-data-no"
name="noayah" placeholder="Masukkan Telepon Ayah" required
value="{{ old('noayah') }}"
@error('pekerjaanayah')
    <div class="alert alert-warning" role="alert">
        <strong>Peringatan!</strong>
{{ $message }}
    </div>
@enderror
</div>
<div class="flex-grow-1 overflow-hidden">
    <h5 class="font-size-16 mb-1">Data sekolah asal dan nilai</h5>
    <p class="text-muted text-truncate mb-0">Sekolah asal, jurusan, nilai
raport dan ijazah</p>
</div>
    <div class="flex-shrink-0"><i
class="mdi mdi-chevron-up accor-down-icon font-size-24"></i></div>
        </div>
    </div>
</a>
<div id="school-data" class="collapse">
<div class="p-4 border-top">
<div class="row">
<div class="col-lg-6">
<div class="mb-3 mb-4">
<label class="form-label" for="billing-address">Nama
Sekolah</label>
<select name="sekolah" class="form-control" id="">
@foreach ($viewSekolah as $z)
<option value="{{ $z->id }}>
{{ $z->nama_sekolah }}</option>
@endforeach
</select>
<div class="row my-4">
    <div class="col">
        <div class="text-end mt-2 mt-sm-0">
            <button type="submit" name="add" class="btn btn-
primary">Buat Pendaftaran</button>
        </div>
    </div>
</div>

```

Gambar 4.33 Source Code Data-pendaftaran.php

Pada gambar diatas merupakan *source code* dari *script* pendaftaran.php. Peserta atau pengguna baru apabila ingin melakukan pendaftaran maka harus mengisi data pribadi, data pendidikan, data pilihan program kursus, dan data orangtua. Apabila data pribadi seperti NISN dan NIK tidak diisi secara lengkap, maka akan memunculkan {{ \$message }} yang berarti pesan *error* atau “*alert alert-warning*” yang menandakan NIK dan NISN harus diisi secara lengkap. Jika tidak mengisi NIK dan NISN maka data pendaftaran peserta tidak akan tersimpan kedalam *database*.

D. Data-pembayaran.php

```
<button type="button" class="btn btn-primary mb-4" data-bs-toggle="modal" data-bs-target=".upload"
style="margin-bottom: 1rem;"><i class="mdi mdi-plus me-1"></i>Upload
Pembayaran </button>
@endif

<button class="btn btn-success mb-4" style="margin-bottom: 1rem;" disabled>Terverifikasi</button>
@elseif ($viewData->status_pendaftaran == "Selesai")
<button class="btn btn-primary mb-4" style="margin-bottom: 1rem;" disabled>Selesai</button>
@else
<span class="badge badge-danger">Data Tidak Sah</span>
@endif
</div>
</div>
<div class="modal fade upload" tabindex="-1" role="dialog"
aria-labelledby="mySmallModalLabel" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Kirim bukti pembayaran</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close">
</button>
</div>
<div class="modal-body">
<form action="{{route('upload-payment')}}" method="POST" enctype="multipart/form-data">
{{ csrf_field() }}
<input type="hidden" name="userid" value="{{ auth()->user()->id}}>
<div class="form-group">
<input type="hidden" name="id_pendaftaran" id="nama" class="form-control"
value="{{ $viewData->id_pendaftaran }}>
```

```

<div class="row">
<div class="col-xl-12">
<label for="iduser">Pilih Dokumen</label>
<div class="input-group">
<span class="input-group-text">Upload</span>
<div class="form-file">
<input type="file" class="form-file-input form-control"
name="pem" >
<input type="hidden" class="form-file-input form-control"
name="pathnya">
</div>
</div>
</div>
</div>
<div class="modal-footer border-top-0 d-flex">
<button type="button" class="btn btn-danger light"
data-bs-dismiss="modal">Tutup</button>
<button type="submit" name="add"
class="btn btn-primary">Perbaharui
Data</button>
</div>

```

Gambar 4.35 *Source Code* Data-pembayaran.php

Pada gambar diatas merupakan *source code* dari *script* pembayaran.php. Pada *source code* tersebut menggunakan *button class primary*, jadi menu pembayaran ini menggunakan *button* saja, apabila ingin melakukan pembayaran hanya perlu mengklik *button* tersebut. Pada kodingan diatas juga terdapat *method POST*, yang berarti perintah untuk mengirimkan data secara langsung ke *file* lain. Disini data yang dikirimkan yakni data *file* bukti pembayaran.

e. Data-pengumuman.php

```

<br>
<div class="p-4 border-top">
<table class="table mb-0">

<thead class="table-light col-lg-12">
<tr>
<td colspan="2">Data Siswa Pendaftar</td>
</tr>
</thead>
<tbody border="5">

<tr border="5">
<td scope="row" width="70%">

```

```

<table>
<tr border="5">
<th scope="row" width="50%">No Pendaftaran</th>
<td width="50%">: {{ $viewIdPendaftaran->id_pendaftaran }}</td>
<tr>
<tr border="5">
<th scope="row" width="50%">NISN Pendaftar</th>
<td width="50%">: {{ $viewIdPendaftaran->nisn }}</td>
</tr>
<tr border="5">
<th scope="row" width="50%">Nama Pendaftar</th>
<td width="50%">: {{ $viewIdPendaftaran->nama_siswa }}</td>
</tr>
<tr border="5">
<th scope="row" width="50%">Asal Sekolah</th>
<td width="50%">
{{-- {$viewIdPendaftaran->skolah->nama_sekolah} --}}
@foreach ($viewSekolah as $item)
@if ($viewIdPendaftaran->sekolah == $item->id)
: {{ $item->nama_sekolah }}
@endif
@endforeach
</td>
</tr>
</table>
</td>
<td colspan="4"></td>
<tr>

<tr border="5">
<th colspan="3" scope="row">
<br>
@foreach ($viewData as $x)
@if($x->hasil_seleksi == 'TIDAK LULUS' && $x->id_pendaftaran== $viewIdPendaftaran->id)


<strong>Semangat!</strong> Anda TIDAK LULUS seleksi Penerimaan peserta kursus baru, Silahkan coba kembali, Jangan Menyerah Ya!
</div>
@elseif ($x->hasil_seleksi == 'LULUS' && $x->id_pendaftaran== $viewIdPendaftaran->id)


<strong>Selamat!</strong> Anda dinyatakan <strong>LULUS</strong> seleksi Penerimaan peserta kursus baru
</div>


```

```

<div class="alert alert-outline-success alert-dismissible fade show" style="border-radius: 0%; margin-top:-1rem">
<table>
<tr>
<th scope="row">Kursus Penerima <b>{{ $x->kursus->nama_kursus }}</b></th>
</tr>
<tr style="padding: 0px; margin:0px">
<td>
<p style="padding: 0px; margin:0px">Hari</p><br>
<p style="padding: 0px; margin:0px">Jam</p><br>
<p style="padding: 0px; margin:0px">Pengajar</p><br>
</td>
<td>
<p style="padding: 0px; margin:0px">: {{ $x->kursus->hari }}</p><br>
<p style="padding: 0px; margin:0px">: {{ $x->kursus->jam }}</p><br>
<p style="padding: 0px; margin:0px">: {{ $x->kursus->pengajar }}</p><br>
</td>
</tr>
</table>
</div>
@endif
@endforeach
</th>
</tr>
<tr>
<td scope="row" style="margin-top:-50px"><small>* Simpan bukti pengumuman ini sebagai salah satu bukti telah melakukan pendaftaran</small></td>
</tr>

```

Gambar 4.33 *Source Code* Data-pengumuman.php

Pada gambar diatas merupakan *source code* dari *script* data-pengumuman.php. Pada kodingan diatas menggunakan perintah *if else*, perintah *if else* ini digunakan apabila terdapat sebuah keputusan antara bernilai *true* atau *false*. Misalnya pada kodingan diatas apabila *if \$x->hasil_seleksi* “TIDAK LULUS” maka memunculkan *alert* pesan “Semangat!, Anda TIDAK LULUS, Silahkan coba Kembali, Jangan Menyerah Ya!. Namun apabila *elseif \$x->hasil_seleksi* “LULUS” maka akan memunculkan *alert* pesan “Selamat! Anda dinyatakan LULUS sebagai peserta kursus *programming course* roblox 2022/2023!. Kemudian akan mengembalikan *view_data* yakni tabel program kursus yakni seperti: Program kursus penerima, hari, jam dan pengajar.

f. Data-programkursus.php

```

<h4 class="card-title">Data jenis kursus</h4>
<!-- center modal -->
<div>
    <button class="btn btn-info waves-effect waves-light mb-4" onclick="printDiv('cetak')"><i class="fa fa-print"></i></button>
    <button type="button" class="btn btn-primary mb-4" data-bs-toggle="modal" data-bs-target=".modal" style="margin-bottom: 1rem;"><i class="mdi mdi-plus me-1"></i>Tambah jenis kursus</button>
</div>

<div class="modal fade modal" tabindex="-1" role="dialog" aria-labelledby="mySmallModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-scrollable">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Tambah jenis kursus</h5>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <form action="save-studyProgram" method="POST" enctype="multipart/form-data">
                    {{ csrf_field() }}
                    <input type="hidden" name="userid" value="{{ auth()->user()->id_user }}>
                    <div class="form-group">
                        <label for="iduser">Nama jenis kursus</label>
                        <input type="text" class="form-control" id="nama" placeholder="Enter Course Program Name" name="nama" required>
                    </div>
                    <div class="form-group">
                        <label for="iduser">Hari</label>
                        <input type="text" class="form-control" id="hari" placeholder="Enter Course Program Name" name="hari" required>
                    </div>
                    <div class="form-group">
                        <label for="iduser">Jam</label>
                        <input type="text" class="form-control" id="jam" placeholder="Enter Course Program Name" name="jam" required>
                    </div>
                    <div class="form-group">
                        <label for="iduser">Pengajar</label>
                        <input type="text" class="form-control" id="pengajar" placeholder="Enter Course Program Name" name="pengajar" required>
                    </div>
                    <div class="form-group">
                        <label for="iduser">Harga</label>
                        <input type="text" class="form-control" id="harga" placeholder="Enter Course Program Name" name="harga" required>
                    </div>
                    <div class="modal-footer border-top-0 d-flex">
                        <button type="button" class="btn btn-danger light" data-bs-dismiss="modal">Tutup</button>
                        <button type="submit" name="add" class="btn btn-primary">Tambah Data</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

No	ID Program Kursus	Program Kursus	Hari	Jam	Pengajar	Harga	Aksi
1							<!--<th>Aksi coba coba</th>-->

```

</tr>
</thead>
<tbody>
    @php $no = 1; @endphp
    @foreach ($viewData as $x)
        <tr>
            <td>{{ $no++ }}</td>
            <td>{{ $x->id_prodi }}</td>
            <td>{{ $x->nama_kursus }}</td>
            <td>{{ $x->hari }}</td>
            <td>{{ $x->jam }}</td>
            <td>{{ $x->pengajar }}</td>
            <td>{{ $x->harga }}</td>
            <td>
                <div class="d-flex">
                    <a class="btn btn-primary shadow btn-xs sharp me-1" title="Edit" data-bs-toggle="modal" data-bs-target=".edit{{ $x->id_prodi }}"><i class="fa fa-pencil-alt"></i></a>
                    <a class="btn btn-danger shadow btn-xs sharp"><i class="fa fa-trash"></i></a>
                    <div class="modal fade delete delete{{ $x->id_prodi }}" tabindex="-1" role="dialog" aria-hidden="true">
                        <div class="modal-dialog modal-sm">
                            <div class="modal-content">
                                <div class="modal-header">
                                    <h5 class="modal-title">Hapus Data</h5>
                                    <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
                                </div>
                                <div class="modal-body text-center"><i class="fa fa-trash"></i><br> Apakah anda yakin ingin menghapus data ini?<br> {{ $x->id_prodi }}</div>
                                <div class="modal-footer">
                                    <button type="button" class="btn btn-danger light" data-bs-dismiss="modal">Batalkan</button>
                                    <a href="delete-studyProgram/{{ $x->id_prodi }}"><button type="submit" class="btn btn-danger shadow">Ya, Hapus Data!</button></a>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </td>
        </tr>
    @endforeach

```

Gambar 4.36 Source Code Data-programkursus.php

Pada gambar diatas merupakan *source code* dari *script* data-programkursus.php. Pada *source code* diatas dimulai dengan *insert* data program kursus, yakni: Nama jenis kursus, hari, jam, pengajar, dan harga. Kemudian *button insert* “Tambah Data”. Setelah itu, perintah *delete* terlebih dahulu kemudian tabel tersebut mengembalikan *view* data dan memilih tabel prodi yang akan dihapus.

g. Data-user.php

```


##### Tambah Pengguna



{{ csrf_field() }}
          
          <div class="form-group">
            <div class="row">
              <div class="col-12">
                <label for="iduser">Nama</label>
                <input type="text" class="form-control" id="nama" placeholder="Masukkan Nama" name="nama" required>
              </div>
            </div>
            <div class="form-group">
              <label for="iduser">Email</label>
              <input type="email" class="form-control" id="email" placeholder="Masukkan Email" name="email" required>
            </div>
            <div class="form-group">
              <label for="iduser">Kata Sandi</label>
              <input type="password" class="form-control" id="password" placeholder="Masukkan Kata Sandi" name="password" required>
            </div>
            <div class="form-group">
              <div class="row">
                <div class="col-6">
                  <label for="iduser">Jenis Kelamin</label>
                  <select class="default-select form-control wide" title="Jenis Kelamin" name="gender" required>
                    <option value="">Pilih Jenis Kelamin</option>
                    <option value="Perempuan">Female</option>
                    <option value="Laki-laki">Male</option>
                  </select>
                </div>
                <div class="col-6">
                  <label for="iduser">Telepon</label>
                  <input type="number" class="form-control" placeholder="Enter Telepon" name="nohp" required>
                </div>
              </div>
            </div>
            <div class="form-group">
              <div class="row">
                <div class="col-6">
                  <label for="iduser">Role Pengguna</label>
                  <select class="default-select form-control wide" title="Country" name="level" required>
                    <option value="">Pilih Role</option>
                    <option value="Administrator">Administrator</option>
                    <option value="Calon Peserta">Calon Peserta</option>
                  </select>
                </div>
                <div class="col-6">
                  <label for="iduser">Foto Profil</label>
                  <div class="input-group mb-3">
                    <span class="input-group-text">Upload</span>
                    <div class="form-file">
                      <input type="file" class="form-file-input form-control" name="foto" required>
                    </div>
                  </div>
                </div>
              </div>
            </div>
            <div class="modal-footer border-top-0 d-flex">
              <button type="button" class="btn btn-danger light" data-bs-dismiss="modal">Tutup</button>
              <button type="submit" name="add" class="btn btn-primary">Tambah Data</button>
            </div>
          </form>


```

```

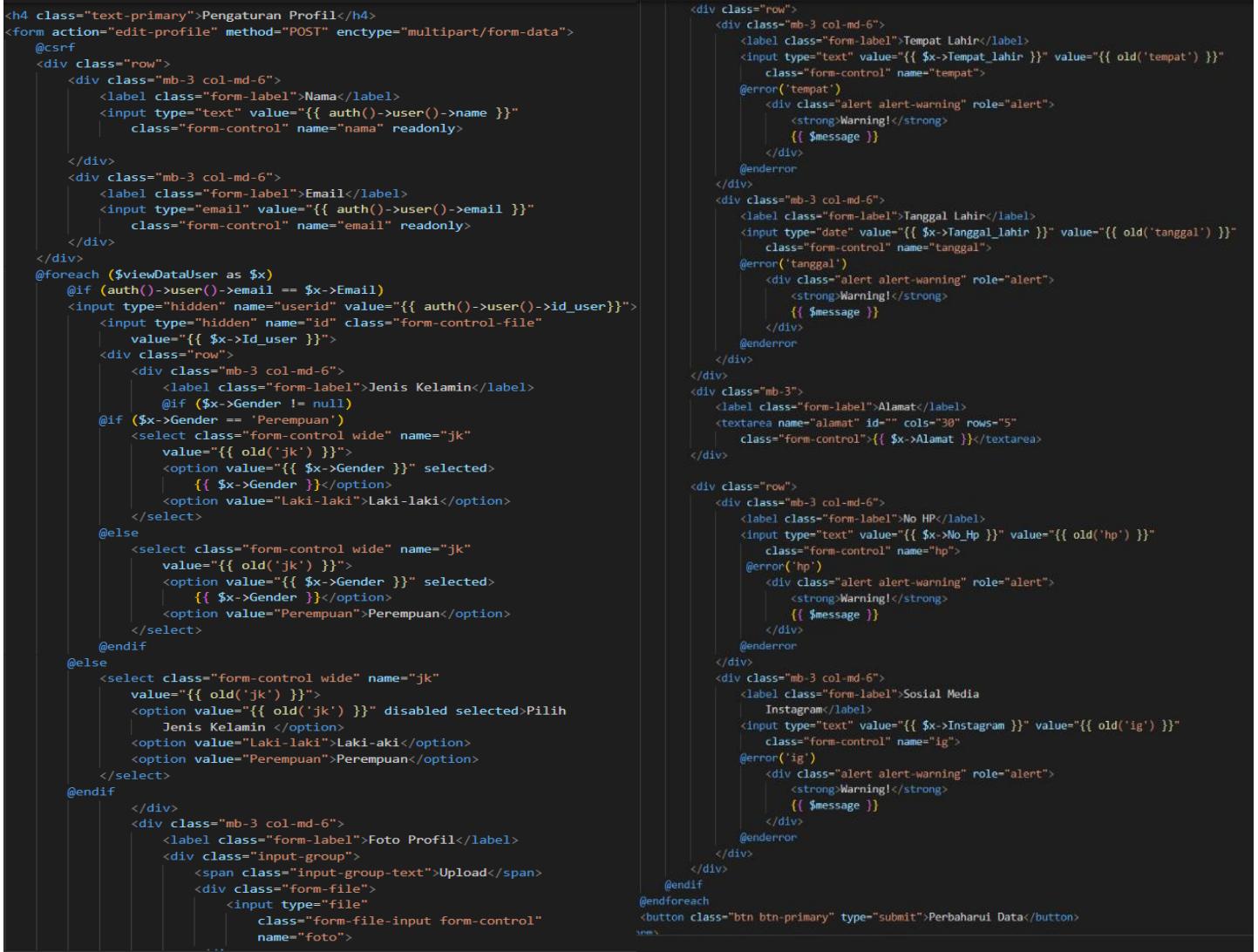
  <div class="card-body">
    <div class="table-responsive" id="cetak">
      {{ csrf_field() }}
      <table id="example3" class="display" style="min-width: 845px">
        <thead>
          <tr>
            <th></th>
            <th>Nama</th>
            <th>Jenis Kelamin</th>
            <th>Telepon</th>
            <th>Email</th>
            <th>Aksi</th>
          </tr>
        </thead>
        <tbody>
          @foreach ($viewData as $x)
            <tr>
              <td>
                @if ($x->Foto != null)
                  
                @else
                  
                @endif
              </td>
              <td>{{ $x->Nama }}</td>
              <td>
                @if ($x->Gender == 'Perempuan')
                  <span class="badge badge-secondary">Perempuan</span>
                @elseif($x->Gender == 'Laki-laki')
                  <span class="badge"
                        style="background-color: #rgb(81, 171, 255)">Laki-Laki</span>
                @endif
              </td>
              <td><strong>{{ $x->No_Hp }}</strong></td>
              <td><a href="javascript:void(0)"><strong>{{ $x->Email }}</strong></a></td>
              <td>
                <div class="d-flex">
                  <a class="btn btn-light shadow btn-xs sharp me-1" href="#" title="Data Registration">i
                    class="fa fa-file-alt"</a>
                  <a class="btn btn-primary shadow btn-xs sharp me-1" href="#" title="Edit">i
                    class="fa fa-pencil-alt"</a>
                  <a href="#">Delete User</a>
                  <div class="btn btn-danger shadow btn-xs sharp">i
                    class="fa fa-trash"</i></a>
                </div>
                <div class="modal fade delete{{ $x->Id_user }}" tabindex="-1" role="dialog" aria-hidden="true">
                  <div class="modal-dialog modal-sm">
                    <div class="modal-content">
                      <div class="modal-header">
                        <h5 class="modal-title">Hapus Data</h5>
                        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
                      </div>
                      <div class="modal-body text-center">i
                        class="fa fa-trash"</i><br> Apakah anda yakin ingin menghapus data ini?<br> {{ $x->Id_user }}</div>
                      <div class="modal-footer">
                        <button type="button" class="btn btn-danger light" data-bs-dismiss="modal">Batalkan</button>
                        <a href="#">Delete User</a>
                        <button type="submit" class="btn btn-danger shadow">Ya, Hapus Data!</button>
                      </div>
                    </div>
                  </div>
                </div>
              </td>
            </tr>
          @endforeach
        </tbody>
      </table>
    </div>
  
```

Gambar 4.37 Source Code Data-user.php

Pada gambar diatas merupakan *source code* dari *script* data-user.php. Pada *source code* diatas merupakan perintah tambah dan hapus data. Kodingan nya dimulai dari tambah dengan mencocokan *\$Post* yang berarti akan mengembalikan data dalam bentuk *view*, kemudian ditampilkan data *user* nya sesuai di *database* yakni: Nama, email, kata sandi, jenis kelamin (*female* atau *male*), telepon, *role* pengguna (administrator dan peserta), foto profil (*upload*). Setelah itu terdapat *button* tambah

data. Setelah itu juga terdapat *button* “cetak”. Disebelahnya, terdapat *script* atau kodingan yakni *delete* data dimulai dengan *display* atau menampilkan data dalam tabel *user* seperti yang sudah dijelaskan sebelumnya. Perintah *if else* pada data-user.php. Jika ingin menghapus data maka `{{$x->id_user}}` yang dipilih akan terhapus.

h. Profil.php



```

<h4 class="text-primary">Pengaturan Profil</h4>
<form action="edit-profile" method="POST" enctype="multipart/form-data">
@csrf
<div class="row">
    <div class="mb-3 col-md-6">
        <label class="form-label">Nama</label>
        <input type="text" value="{{ auth()->user()->name }}" class="form-control" name="nama" readonly>
    </div>
    <div class="mb-3 col-md-6">
        <label class="form-label">Email</label>
        <input type="email" value="{{ auth()->user()->email }}" class="form-control" name="email" readonly>
    </div>
</div>
@foreach ($viewDataUser as $x)
@if (auth()->user()->email == $x->Email)
<input type="hidden" name="userid" value="{{ auth()->user()->id_user }}>
<input type="hidden" name="id" class="form-control-file" value="{{ $x->Id_user }}>
<div class="row">
    <div class="mb-3 col-md-6">
        <label class="form-label">Jenis Kelamin</label>
        @if ($x->Gender != null)
        @if ($x->Gender == 'Perempuan')
            <select class="form-control wide" name="jk" value="{{ old('jk') }}"
                ><option value="{{ $x->Gender }}" selected>{{ $x->Gender }}</option>
                <option value="Laki-laki">Laki-laki</option>
            </select>
        @else
            <select class="form-control wide" name="jk" value="{{ old('jk') }}"
                ><option value="{{ $x->Gender }}" selected>{{ $x->Gender }}</option>
                <option value="Perempuan">Perempuan</option>
            </select>
        @endif
    @else
        <select class="form-control wide" name="jk" value="{{ old('jk') }}"
            ><option value="{{ old('jk') }}" disabled selected>Pilih Jenis Kelamin </option>
            <option value="Laki-laki">Laki-aki</option>
            <option value="Perempuan">Perempuan</option>
        </select>
    @endif
    </div>
    <div class="mb-3 col-md-6">
        <label class="form-label">Foto Profil</label>
        <div class="input-group">
            <span class="input-group-text">Upload</span>
            <div class="form-file">
                <input type="file" class="form-file-input form-control" name="foto">
            </div>
        </div>
    </div>
</div>
<div class="mb-3 col-md-6">
    <label class="form-label">Tempat Lahir</label>
    <input type="text" value="{{ $x->Tempat_lahir }}" value="{{ old('tempat') }}"
        class="form-control" name="tempat">
    @error('tempat')
        <div class="alert alert-warning" role="alert">
            <strong>Warning!</strong>
            {{ $message }}
        </div>
    @enderror
</div>
<div class="mb-3 col-md-6">
    <label class="form-label">Tanggal Lahir</label>
    <input type="date" value="{{ $x->Tanggal_lahir }}" value="{{ old('tanggal') }}"
        class="form-control" name="tanggal">
    @error('tanggal')
        <div class="alert alert-warning" role="alert">
            <strong>Warning!</strong>
            {{ $message }}
        </div>
    @enderror
</div>
<div class="mb-3">
    <label class="form-label">Alamat</label>
    <textarea name="alamat" id="" cols="30" rows="5"
        class="form-control">{{ $x->Alamat }}</textarea>
</div>
<div class="row">
    <div class="mb-3 col-md-6">
        <label class="form-label">No HP</label>
        <input type="text" value="{{ $x->No_Hp }}" value="{{ old('hp') }}"
            class="form-control" name="hp">
        @error('hp')
            <div class="alert alert-warning" role="alert">
                <strong>Warning!</strong>
                {{ $message }}
            </div>
        @enderror
    </div>
    <div class="mb-3 col-md-6">
        <label class="form-label">Sosial Media Instagram</label>
        <input type="text" value="{{ $x->Instagram }}" value="{{ old('ig') }}"
            class="form-control" name="ig">
        @error('ig')
            <div class="alert alert-warning" role="alert">
                <strong>Warning!</strong>
                {{ $message }}
            </div>
        @enderror
    </div>
    @endforeach
</div>
<button class="btn btn-primary" type="submit">Perbarui Data</button>

```

Gambar 4.38 Source Code Profil.php

Pada gambar diatas merupakan *source code* dari *script* profil.php. Pada *source code* diatas merupakan bagian pengaturan profil terdapat isinya yakni: Nama, email, jenis kelamin (laki laki atau perempuan, serta foto profil (*upload*), tempat lahir, tanggal lahir, alamat, no hp, *social media* (*instagram*). Terdapat *button* “Perbarui Data”. Data data yang ada dimenu profil tidak boleh kosong. Sebab terdapat *script* {{

\$message } } warning !. Hal itu berarti data yang ada di profil tidak boleh tidak terisi seperti: Tempat lahir, tanggal lahir, alamat, no hp, dan sosial media.

i. Dashboard.php

```
</div>
<div class="row">
    <div class="col-xl-3 col-lg-6 col-sm-6" style="padding: 0px">
        <div class="card" style="border-radius: 0%;">
            <div class="new-arrival-product">
                <div class="new-arrivals-img-contnent">
                    
                    <div class="carousel-caption d-none d-md-block">
                        <button type="button" class="btn btn-transparent text-white" data-bs-toggle="modal" data-bs-target="#myModal1">
                            Puzzle Games</button>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="col-xl-3 col-lg-6 col-sm-6" style="padding: 0px">
        <div class="card" style="border-radius: 0%;">
            <div class="new-arrival-product">
                <div class="new-arrivals-img-contnent">
                    
                    <div class="carousel-caption d-none d-md-block">
                        <button type="button" class="btn btn-transparent text-white" data-bs-toggle="modal" data-bs-target="#myModal2">
                            Fighting Games</button>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="col-xl-3 col-lg-6 col-sm-6" style="padding: 0px">
        <div class="card" style="border-radius: 0%;">
            <div class="new-arrival-product">
                <div class="new-arrivals-img-contnent">
                    
                    <div class="carousel-caption d-none d-md-block">
                        <button type="button" class="btn btn-transparent text-white" data-bs-toggle="modal" data-bs-target="#myModal3">
                            Sport Games</button>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="col-xl-3 col-lg-6 col-sm-6" style="padding: 0px">
        <div class="card" style="border-radius: 0%;">
            <div class="new-arrival-product">
                <div class="new-arrivals-img-contnent">
                    
                    <div class="carousel-caption d-none d-md-block">
                        <button type="button" class="btn btn-transparent text-white" data-bs-toggle="modal" data-bs-target="#myModal4">
                            Education Games</button>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

Gambar 4.39 Source Code Dashboard.php

Pada gambar 4.39 merupakan *source code* dari *script dashboard.php*. Pada *source code* diatas berisikan kodingan untuk desain halaman beranda agar memiliki kesan menarik seperti warna, lebar, tinggi, gambar dan lain sebagainya. Di halaman beranda terdapat *mymodal1* (program 1), *mymodal2* (program 2), *mymodal3* (program 3), dan *mymodal4* (program 4). Dimana *mymodal1* berisi untuk tampilan *puzzle game* (program 1), *mymodal2* untuk tampilan *fighting game* (program 2), *mymodal3* untuk tampilan *sport game* (program 3), serta *mymodal4* untuk tampilan *education game* (program 4).

4.1.2.3 Controllers

```

//data pembayaran komplit
public function datapembayaran(){
    $dataUser = Pengguna::all();
    $data = Pembayaran::all();
    $dataid = Pendaftaran::all();
    return view ('data-pembayaran-admin',[ 'viewDataUser' => $dataUser,'viewData' => $data,'viewIdPendaftaran' => $dataid]);
}

public function simpanpembayaran(Request $a)
{
    try{
        // $dataUser = Pengguna::all();
        $kode = Pembayaran::id();
        $file = $a->file('bukti');
        $kodependaftaran = $a->id_pendaftaran;
        $nama_file = "payment-".time() . "-" . $file->getClientOriginalName();
        $namaFolder = 'data pendaftar/'.$kodependaftaran;
        $file->move($namaFolder,$nama_file);
        $pathBukti = $namaFolder."/". $nama_file;
        Pembayaran::create([
            'id_pembayaran' => $kode,
            'bukti_pembayaran' => $pathBukti,
            'status_pembayaran'=> $a->status,
            'id_pendaftaran' =>$a->id_pendaftaran
        ]);
        Timeline::create([
            'id_user' => $a->userid,
            'status' => "Memperbaharui Pembayaran"
        ]);
        return redirect('/data-payment')->with('success', 'Data Tersimpan!!');
    } catch (\Exception $e){
        return redirect()->back()->with('error', 'Data Tidak Berhasil Disimpan!');
    }
}
public function updatepembayaran(Request $a, $id_pembayaran){
    // $dataUser = Pengguna::all();
    try{
        $file = $a->file('bukti');
        if(file_exists($file)){
            $kodependaftaran = $a->id_pendaftaran;
            $nama_file = "payment-".time() . "-" . $file->getClientOriginalName();
            $namaFolder = 'data pendaftar/'.$kodependaftaran;
            $file->move($namaFolder,$nama_file);
            $pathBukti = $namaFolder."/". $nama_file;
        } else {
            $pathBukti = $a->pathnya;
        }

        Pembayaran::where("id_pembayaran", "$id_pembayaran")->update([
            'bukti_pembayaran' => $pathBukti,
            'status_pembayaran'=> $a->status,
            'id_pendaftaran' =>$a->id_pendaftaran
        ]);
        Timeline::create([
            'id_user' => $a->userid,
            'status' => "Memperbaharui Pembayaran"
        ]);
        return redirect('/data-payment')->with('success', 'Data Terubah!!');

    } catch (\Exception $e){
        return redirect()->back()->with('error', 'Data Tidak Berhasil Diubah!');
    }
}

```

Gambar 4.40 Source Code Controllers

Pada gambar diatas merupakan *source code* dari *script controllers*. Pada *source code* konsep untuk *controllers* berisikan kodingan untuk mengontrol seluruh kegiatan yang ada didalam proses proses di aplikasi. Mulai dari proses *read, insert, update, and delete* hingga proses tampilan isi data juga *controllers* yang mengatur atau mengontrol berjalannya proses layanan *client* dan aplikasi yang saling berkomunikasi, selain itu di *controllers* ini juga ditandai dengan adanya perintah *try()*, *catch()*, dan *return*.

4.2 Pengujian dan Hasil Pengujian

4.2.1 Pengujian

Pengujian program perlu dilakukan karena merupakan bagian penting dari suatu pengujian, sebab tanpa adanya pengujian maka aplikasi yang dirancang tidak akan berhasil uji coba, tidak hanya itu pengujian juga dilakukan agar mengetahui kesalahan kesalahan pada program agar dapat memastikan bahwa aplikasi yang dirancang telah sesuai dengan apa yang diharapkan. Pada tahap pengujian ini menggunakan pengujian *whitebox*. Pengujian *whitebox* ditargetkan untuk dapat lebih menitikberatkan kepada *source code* apa saja yang ada di program, dan *input* atau *output* apa yang akan di tampilkan dapat berjalan.

4.2.1.1 Rencana Pengujian *Role* Peserta

Tabel 4.1 Rencana Pengujian *Role* Peserta

Menu Uji	Butir Uji	Jenis Pengujian	Tool pengujian	Jadwal
<i>Register</i>	Klik <i>button</i> daftar	Sistem	Selenium IDE	15/01/2023
	Data Peserta: email, nama, dan <i>password</i> lengkap	Sistem	Selenium IDE	15/01/2023
<i>Login</i>	Klik <i>button login</i>	Sistem	Selenium IDE	15/01/2023
	Data Peserta: <i>Username</i> dan <i>password</i> salah	Sistem	Selenium IDE	15/01/2023
	Data Peserta: <i>Username</i> dan <i>password</i> benar	Sistem	Selenium IDE	15/01/2023
<i>Pendaftaran</i>	Klik <i>button</i> pendaftaran	Sistem	Selenium IDE	15/01/2023
	<i>Input</i> data pendaftaran lengkap	Sistem	Selenium IDE	15/01/2023
<i>Pembayaran</i>	Klik <i>upload</i> pembayaran	Sistem	Selenium IDE	15/01/2023
	<i>Input</i> bukti pembayaran	Sistem	Selenium IDE	15/01/2023

	Klik <i>button</i> cetak detail pendaftaran	Sistem	Selenium IDE	15/01/2023
Pengumuman	Klik lihat pengumuman	Sistem	Selenium IDE	15/01/2023
	Klik <i>button</i> cetak	Sistem	Selenium IDE	15/01/2023
Profil	Klik menu profil	Sistem	Selenium IDE	15/01/2023
	<i>Update</i> data peserta	Sistem	Selenium IDE	15/01/2023
Logout	Klik menu <i>logout</i>	Sistem	Selenium IDE	15/01/2023

4.2.1.2 Rencana Pengujian *Role Admin*

Tabel 4.2 Rencana Pengujian *Role Admin*

Menu Uji	Butir Uji	Jenis Pengujian	Tool pengujian	Jadwal
Login	Klik <i>button</i> <i>login</i>	Sistem	Selenium IDE	15/01/2023
	Data admin: <i>Username</i> dan <i>password</i> salah	Sistem	Selenium IDE	15/01/2023
	Data admin: <i>Username</i> dan <i>password</i> benar	Sistem	Selenium IDE	15/01/2023
Dashboard	Klik <i>button</i> lihat pendaftar	Sistem	Selenium IDE	15/01/2023
Data Master: Pengguna	Klik menu pengguna	Sistem	Selenium IDE	15/01/2023
	Tambah data pengguna	Sistem	Selenium IDE	15/01/2023
	<i>Update</i> data pengguna	Sistem	Selenium IDE	15/01/2023
	Hapus data pengguna	Sistem	Selenium IDE	15/01/2023
Data Master: Sekolah	Klik menu sekolah	Sistem	Selenium IDE	15/01/2023

	Tambah data sekolah	Sistem	Selenium IDE	15/01/2023
	<i>Update</i> data sekolah	Sistem	Selenium IDE	15/01/2023
	Hapus data sekolah	Sistem	Selenium IDE	15/01/2023
Data <i>Master</i> : Program Kursus	Klik menu program kursus	Sistem	Selenium IDE	15/01/2023
	Tambah data program kursus	Sistem	Selenium IDE	15/01/2023
	<i>Update</i> data program kursus	Sistem	Selenium IDE	15/01/2023
	Hapus data program kursus	Sistem	Selenium IDE	15/01/2023
Data <i>History</i> : Pendaftaran	Klik menu pendaftaran	Sistem	Selenium IDE	15/01/2023
	Tambah data pendaftaran	Sistem	Selenium IDE	15/01/2023
	<i>Update</i> data pendaftaran	Sistem	Selenium IDE	15/01/2023
	Hapus data pendaftaran	Sistem	Selenium IDE	15/01/2023
Data <i>History</i> : Pembayaran	Klik menu pembayaran	Sistem	Selenium IDE	15/01/2023
	<i>Update</i> data pendaftaran	Sistem	Selenium IDE	15/01/2023
Pengumuman	Klik menu pengumuman	Sistem	Selenium IDE	15/01/2023
	<i>Update</i> data pengumuman	Sistem	Selenium IDE	15/01/2023
Profil	Klik menu profil	Sistem	Selenium IDE	15/01/2023
	<i>Update</i> data profil	Sistem	Selenium IDE	15/01/2023
Logout	Klik menu <i>logout</i>	Sistem	Selenium IDE	15/01/2023

4.2.1.3 Pengujian Aplikasi

The screenshot shows a test runner interface for a project named "Sistem_Programming_Course_Roblox.test". The "Tests" tab is selected, displaying a list of test cases. One test case, "register-user", is highlighted in blue. The test log at the bottom shows the following steps:

```

9. click on css=.input-field:nth-child(5) > input OK
10. type on css=.input-field:nth-child(5) > input with value 12345678 OK
11. click on name=password_confirmation OK
12. type on name=password_confirmation with value 12345678 OK
13. click on css=.sign-up-form OK
14. click on css=.btn:nth-child(7) OK
'register-user' completed successfully

```

Gambar 4.41 Pengujian Register

Pada bagian pengujian bagian *register-user*, Peserta melakukan proses mendaftar dengan klik “daftar” data masukan yang dimasukkan yakni: nama, email, *password*, dan ulangi *password*, dan nama. Kemudian klik button “daftar”, maka aplikasi berhasil menyimpan data akun peserta.

The screenshot shows a test runner interface for the same project. A test case, "login-gagal-user", is highlighted in blue. The test log at the bottom shows the following steps:

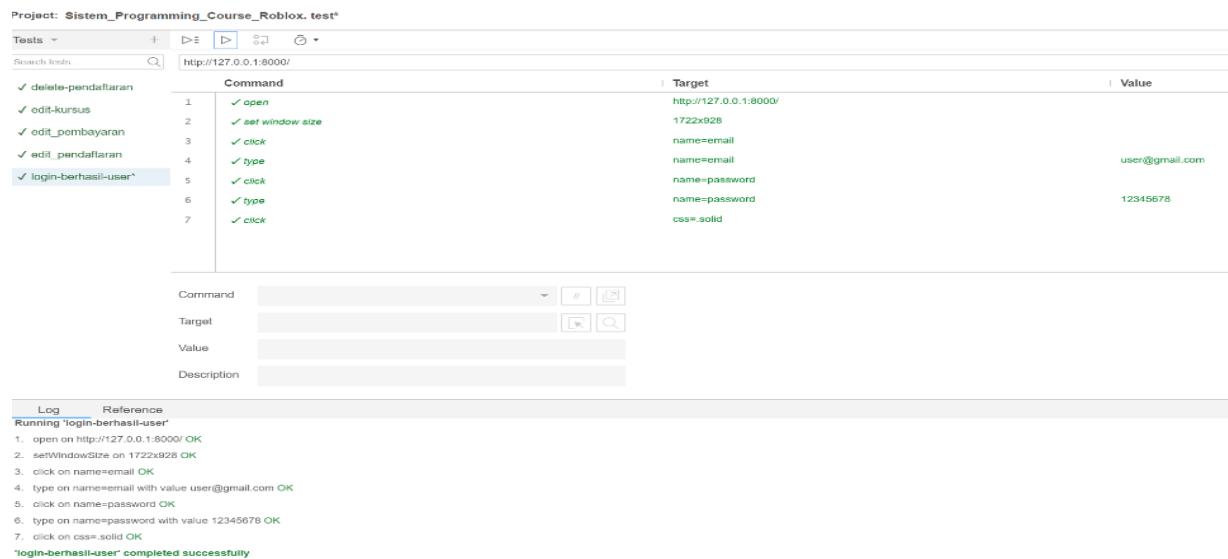
```

1. open on http://127.0.0.1:8000/ OK
2. setWindowSize on 1722x928 OK
3. click on name=email OK
4. type on name=email with value user@gmail.com OK
5. click on name=password OK
6. type on name=password with value usyta42 OK
7. click on css=.solid OK
'registration-gagal-user' completed successfully

```

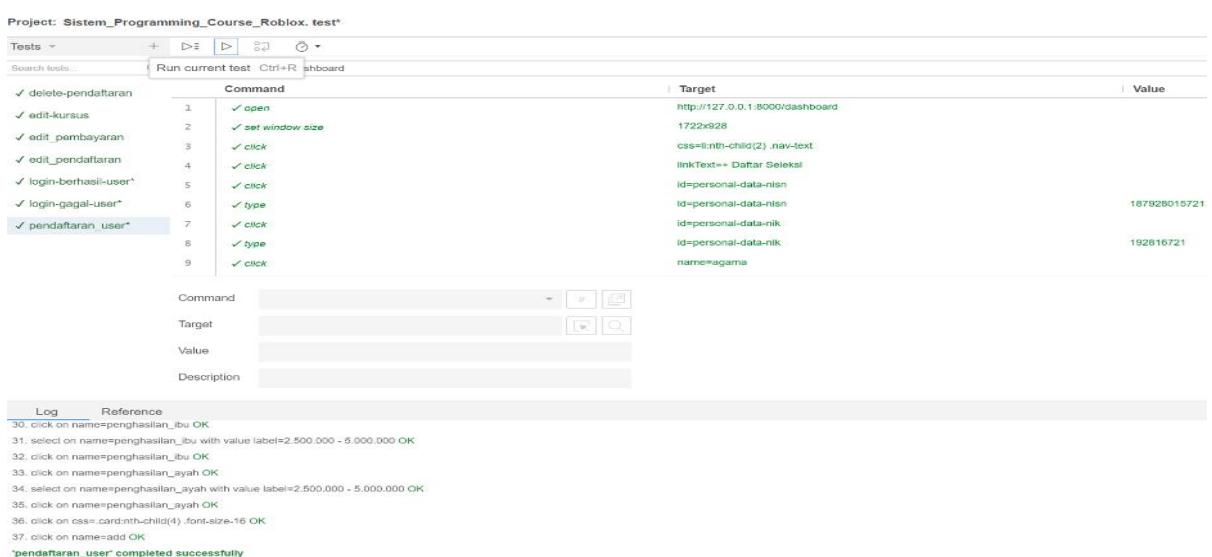
Gambar 4.42 Pengujian Login Gagal Peserta

Pada bagian pengujian bagian *login* gagal, peserta memasukan data akun peserta seperti *username* dan *password*, seperti pada contoh diatas yakni *username = user@gmail.com*, dan *password = ustya42*, peserta selanjutnya mengklik button “*login*”. Maka aplikasi akan menampilkan pesan *error* karena sandi atau *password* tidak sesuai.



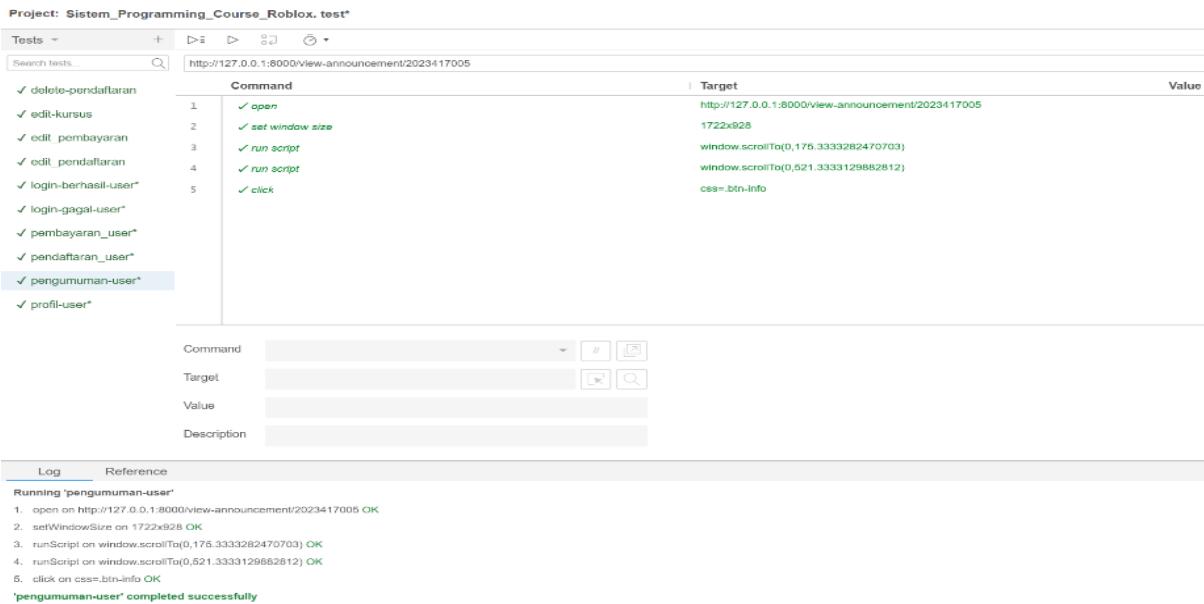
Gambar 4.43 Pengujian *Login* Berhasil Peserta

Pada bagian pengujian bagian *login* berhasil dilakukan proses memasukan data akun peserta seperti *username* dan *password*, seperti pada contoh diatas yakni *username = admin@gmail.com*, dan *password = 12345678*, peserta selanjutnya mengklik button “*login*”. Maka aplikasi akan menampilkan tampilan menu *dashboard*.



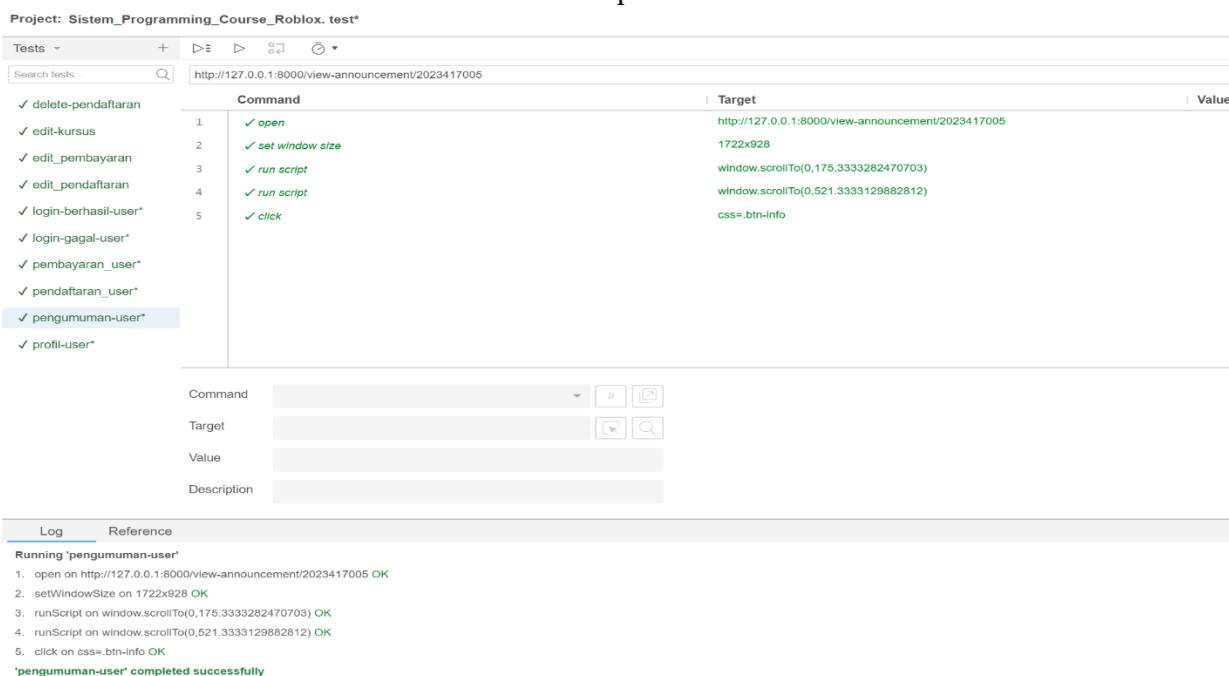
Gambar 4.44 Pengujian Pendaftaran Peserta

Pada bagian pengujian bagian pendaftaran peserta, peserta memasukkan data pendaftaran seperti data pribadi, data nilai dan asal sekolah, data orang tua, dan lain sebagainya. Kemudian peserta mengklik button “buat pendaftaran” maka aplikasi akan menampilkan pesan “berhasil menyimpan data”.



Gambar 4.45 Pengujian Pembayaran Peserta

Pada bagian pengujian bagian pembayaran peserta. Pada awalnya peserta melakukan klik “upload pembayaran” selanjutnya peserta memasukkan bukti pembayaran, dan klik perbaharui data. Maka aplikasi selanjutnya akan memunculkan notifikasi “data tersimpan”.



Gambar 4.46 Pengujian Pengumuman Peserta

Pada bagian pengujian menu pengumuman peserta, peserta mengklik “lihat pengumuman”. Aplikasi selanjutnya menampilkan halaman pengumuman, selanjutnya peserta mengklik “cetak pengumuman”.

The screenshot shows a test project titled "Sistem_Programming_Course_Roblox.test". The "Tests" tab is selected, displaying a list of 13 test steps. The steps are as follows:

- ✓ delete-pendaftaran
- ✓ edit-kursus
- ✓ edit_pembayaran
- ✓ edit_pendaftaran
- ✓ login berhasil-user*
- ✓ login gagal-user*
- pembayaran-user
- ✓ pendaftaran_user*
- pengumuman-user
- ✓ profil-user*

For the last step, "profil-user*", the details are:

Command	Target	Value
✓ click	linkText=Pengaturan	
✓ click	name=tempat	
✓ type	name=tempat	
✓ click	name=alamat	Purwokerto Indonesia
✓ type	name=alamat	Jln. Merdeka, No.09, Purwokerto, Jawa Tengah, Indonesia
✓ click	name=tempat	
✓ type	name=tempat	Purwokerto Dusun IV
✓ click	css=.btn:nth-child(9)	
✓ click	css=.swal2-confirm	

Below the table, there are input fields for "Command", "Target", "Value", and "Description". The "Log" section at the bottom shows the execution results:

```

6. click on name=tempat OK
7. type on name=tempat with value Purwokerto Indonesia OK
8. click on name=alamat OK
9. type on name=alamat with value Jln. Merdeka, No.09, Purwokerto, Jawa Tengah, Indonesia OK
10. click on name=tempat OK
11. type on name=tempat with value Purwokerto Dusun IV OK
12. click on css=.btn:nth-child(9) OK
13. click on css=.swal2-confirm OK
'profil-user' completed successfully

```

Gambar 4.47 Pengujian Profil Peserta

Pada bagian pengujian profil peserta selanjutnya peserta mengklik menu profil, maka aplikasi akan menampilkan berupa halaman profil peserta. Selanjutnya, peserta melakukan *update* data dengan klik “pengaturan” dan memasukkan data yang akan di *edit*. Terakhir, aplikasi akan menampilkan pesan “data terubah”.

The screenshot shows a test project titled "Sistem_Programming_Course_Roblox.test". The "Tests" tab is selected, displaying a list of 7 test steps. The steps are as follows:

- ✓ gagal_login*

For the first step, "gagal_login*", the details are:

Command	Target	Value
✓ open	http://127.0.0.1:8000/	
✓ set window size	1722x928	
✓ click	name=email	
✓ type	name=email	admin@gmail.com
✓ click	name=password	
✓ type	name=password	12jhga
✓ click	css=.solid	

Below the table, there are input fields for "Command", "Target", "Value", and "Description". The "Log" section at the bottom shows the execution results:

```

2. setWindowSize on 1722x928 OK
3. click on name=email OK
4. type on name=email with value admin@gmail.com OK
5. click on name=password OK
6. type on name=password with value 12jhga OK
7. click on css=.solid OK
'gagal_login' completed successfully

```

Gambar 4.48 Pengujian Login Gagal Admin

Pada bagian pengujian bagian *login* gagal, admin memasukan data akun admin seperti *username* dan *password*, seperti pada contoh diatas yakni *username = admin@gmail.com*, dan *password = 12jhga*, admin selanjutnya mengklik button “*login*”. Maka aplikasi akan menampilkan pesan *error* karena sandi atau *password* tidak sesuai.

The screenshot shows a test runner interface for a project named "Sistem_Programming_Course_Roblox.test". The "Tests" dropdown is open, showing three tests: "berhasil_login*", "gagal_login*", and "dashboard_admin*". The "berhasil_login*" test is currently selected. The "Log" tab displays the following steps:

```

2. setWindowSize on 1722x928 OK
3. click on name=email OK
4. type on name=email with value admin@gmail.com OK
5. click on name=password OK
6. type on name=password with value 12345678 OK
7. click on css=.solid OK
'berhasil_login' completed successfully

```

The "Reference" tab shows the expected values for each step. The "Command" column lists the actions taken, and the "Target" and "Value" columns show the expected results. The "Description" field is empty.

Gambar 4.49 Pengujian *Login* Berhasil Admin

Pada bagian pengujian bagian *login* gagal, admin memasukan data akun admin seperti *username* dan *password*, seperti pada contoh diatas yakni *username = admin@gmail.com*, dan *password = 12345678*, admin selanjutnya mengklik button “*login*”. Maka aplikasi akan menampilkan halaman *dashboard* karena sandi dan *password* benar.

The screenshot shows a test runner interface for a project named "Sistem_Programming_Course_Roblox.test". The "Tests" dropdown is open, showing three tests: "berhasil_login*", "dashboard_admin*", and "gagal_login*". The "gagal_login*" test is currently selected. The "Log" tab displays the following steps:

```

2. setWindowSize on 1722x928 OK
3. click on name=email OK
4. type on name=email with value admin@gmail.com OK
5. click on name=password OK
6. type on name=password with value 12jhga OK
7. click on css=.solid OK
'gagal_login' completed successfully

```

The "Reference" tab shows the expected values for each step. The "Command" column lists the actions taken, and the "Target" and "Value" columns show the expected results. The "Description" field is empty.

Gambar 4.50 Pengujian *Dashboard* Admin

Pada bagian pengujian *dashboard* admin, admin mengklik menu *dashboard*, kemudian admin mengklik button “lihat pendaftar”, maka aplikasi akan mengarahkan ke halaman data-*user*.

Project: Sistem_Programming_Course_Roblox. test*			
Tests	+	Run current test Ctrl+R	la-user
Search tests...			
✓ delete-pendaftaran	1.1	✓ click	Target name=password
✓ edit-kursus	1.2	✓ type	name=password Value 12345678
✓ edit_pembayaran	1.3	✓ click	css=.form-group:nth-child(6).nice-select
✓ edit_pendaftaran	1.4	✓ click	css=.open_option:nth-child(3)
✓ login-berhasil-user	1.5	✓ click	css=.form-group:nth-child(6).nice-select
✓ login-gagal-user	1.6	✓ click	css=.open_option:nth-child(2)
✓ pembayaran_user	1.7	✓ click	name=nohp
✓ pendaftaran_user	1.8	✓ type	name=nohp Value 082356789011
✓ pengguna-admin*	1.9	✓ click	css=.form-group:nth-child(7).nice-select
✓ pengumuman-user	2.0	✓ click	css=.open_option:nth-child(3)
✓ profil-user	2.1	✓ click	name=add
✓ sekolah-admin	2.2	✓ click	css=.swal2-popup
	2.3	✓ click	css=.swal2-confirm
Command			
Target			
Value			
Description			
Log Reference			
18. type on name=nohp with value 082356789011 OK			
19. click on css=.form-group:nth-child(7).nice-select OK			
20. click on css=.open_option:nth-child(3) OK			
21. click on name=add OK			
22. click on css=.swal2-popup OK			
23. click on css=.swal2-confirm OK			
'pengguna-admin' completed successfully			

Gambar 4.51 Pengujian Menu Pengguna Admin

Pada bagian pengujian menu pengguna admin, admin mengklik menu pengguna, lalu, melakukan *edit*, *update* dan *delete* sesuai dengan tombol tombol yang ada. Hasil pengujian membuktikan bahwa pada menu pengguna tidak terjadi kesalahan seperti *error* kodingan dan lain sebagainya.

Project: Sistem_Programming_Course_Roblox. test*			
Tests	+	Run current test Ctrl+R	la-sekolah
Search tests...			
✓ delete-pendaftaran	1	✓ open	Target http://127.0.0.1:8000/data-sekolah
✓ edit-kursus	2	✓ set window size	1722x928
✓ edit_pembayaran	3	✓ run script	window.scrollTo(0,400)
✓ edit_pendaftaran	4	✓ click	css=.odd:nth-child(9).btn-primary
✓ login-berhasil-user*	5	✓ click	css=.modal:nth-child(10).col-xl-8 > #nama
✓ login-gagal-user*	6	✓ type	css=.modal:nth-child(10).col-xl-8 > #nama Value SMAN 1 LIMA PULUH
✓ pembayaran_user*	7	✓ click	css=.modal:nth-child(10).bltn:nth-child(2)
✓ pendaftaran_user*	8	✓ click	css=.swal2-confirm
✓ pengumuman-user*			
✓ profil-user*			
✓ sekolah-admin*			
Command			
Target			
Value			
Description			
Log Reference			
3. runScript on window.scrollTo(0,400) OK			
4. click on css=.odd:nth-child(9).btn-primary OK			
5. click on css=.modal:nth-child(10).col-xl-8 > #nama OK			
6. type on css=.modal:nth-child(10).col-xl-8 > #nama with value SMAN 1 LIMA PULUH OK			
7. click on css=.modal:nth-child(10).bltn:nth-child(2) OK			
8. click on css=.swal2-confirm OK			
'sekolah-admin' completed successfully			

Gambar 4.52 Pengujian Menu Sekolah Admin

Pada bagian pengujian *dashboard* admin, admin mengklik menu sekolah, kemudian admin mengklik menu tambah sekolah. Aplikasi akan memunculkan halaman tambah data sekolah, selanjutnya admin juga melakukan *update* dan *delete* dengan menekan tombol *update* dan *delete*.

Project: Sistem_Programming_Course_Roblox. test*

Tests + > D: > 0: > ○ ⊞

Search tests... http://127.0.0.1:8000/data-kursus

Test	Command	Target	Value
✓ kursus-admin*	1 ✓ open	http://127.0.0.1:8000/data-kursus	
	2 ✓ set window size	1722x928	
	3 ✓ click	css=.even:nth-child(4) .btn-primary > .fa	
	4 ✓ click	css=.modal:nth-child(3) #pengajar	
	5 ✓ click	css=.modal:nth-child(5) #pengajar	
	6 ✓ type	css=.modal:nth-child(5) #pengajar	Daffa Audya
	7 ✓ click	css=.modal:nth-child(5) .col-x12:nth-child(3)	
	8 ✓ click	css=.modal:nth-child(5) .btn:nth-child(2)	
	9 ✓ click	css=.swal2-confirm	

Command: open Target: http://127.0.0.1:8000/data-kursus Value: Description:

Log Reference

```

4. click on css=.modal:nth-child(5) #pengajar OK
5. click on css=.modal:nth-child(3) #pengajar OK
6. type on css=.modal:nth-child(5) #pengajar with value Daffa Audya OK
7. click on css=.modal:nth-child(5) .col-x12:nth-child(3) OK
8. click on css=.modal:nth-child(5) .btn:nth-child(2) OK
9. click on css=.swal2-confirm OK
'kursus-admin' completed successfully

```

Gambar 4.53 Pengujian Menu Kursus Admin

Pada bagian pengujian menu kursus, admin mengklik menu kursus, admin klik menu kursus selanjutnya sama halnya seperti menu sebelumnya yakni melakukan *update*, *delete*, dan tambah data kursus. Terlihat hasil, bahwasannya pengujian menu kursus berhasil “sukses”.

Project: Sistem_Programming_Course_Roblox. test*

Tests + > D: > 0: > ○ ⊞

Search tests... Run current test Ctrl+R ta-registration

Test	Command	Target	Value
✓ kursus-admin*	1 ✓ open	http://127.0.0.1:8000/data-registration	
	2 ✓ set window size	1722x928	
	3 ✓ run script	window.scrollTo(0,200)	
	4 ✓ click	css=.odd:nth-child(5) .btn-primary > .fa	
✓ pendaftaran-admin*	5 ✓ run script	window.scrollTo(0,100)	
	6 ✓ run script	window.scrollTo(0,500)	
	7 ✓ click	name=nohp	
	8 ✓ type	name=nohp	82378996034
	9 ✓ click	name=add	
	10 ✓ click	css=.swal2-confirm	

Command: Target: Value: Description:

Log Reference

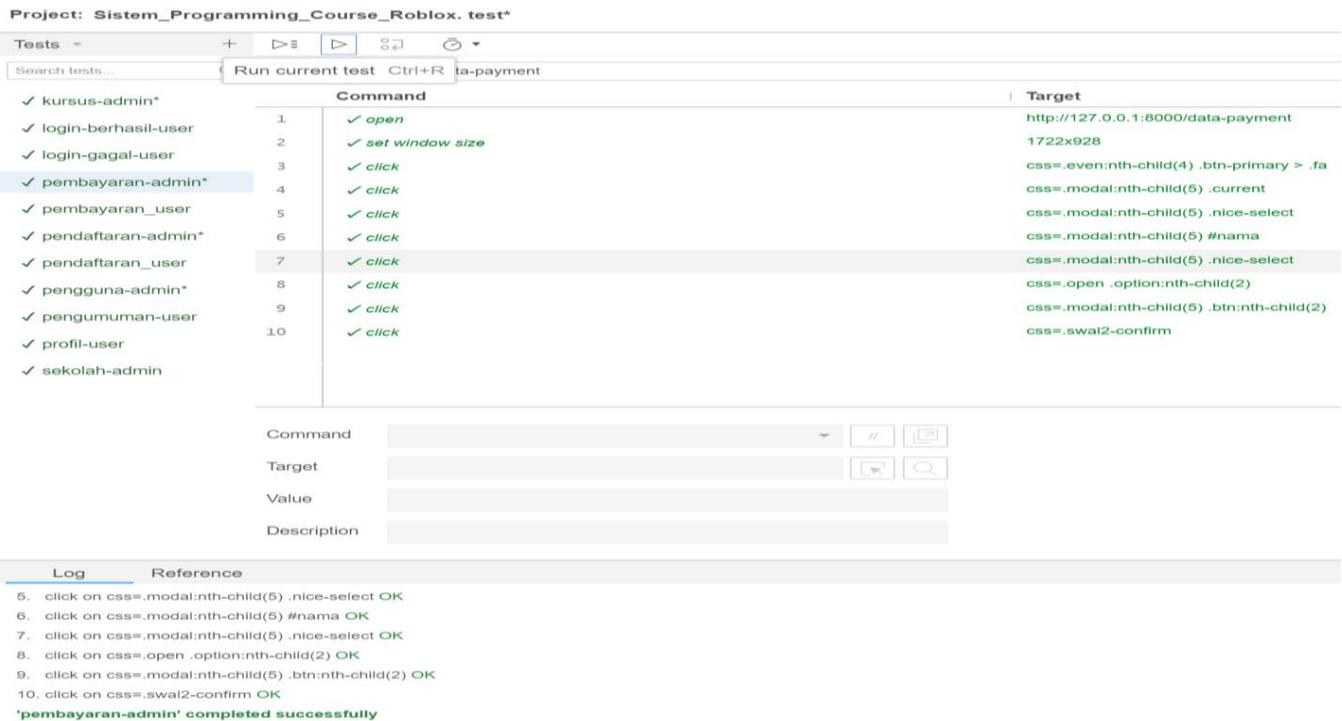
```

5. runScript on window.scrollTo(0,100) OK
6. runScript on window.scrollTo(0,500) OK
7. click on name=nohp OK
8. type on name=nohp with value 82378996034 OK
9. click on name=add OK
10. click on css=.swal2-confirm OK
'pendaftaran-admin' completed successfully

```

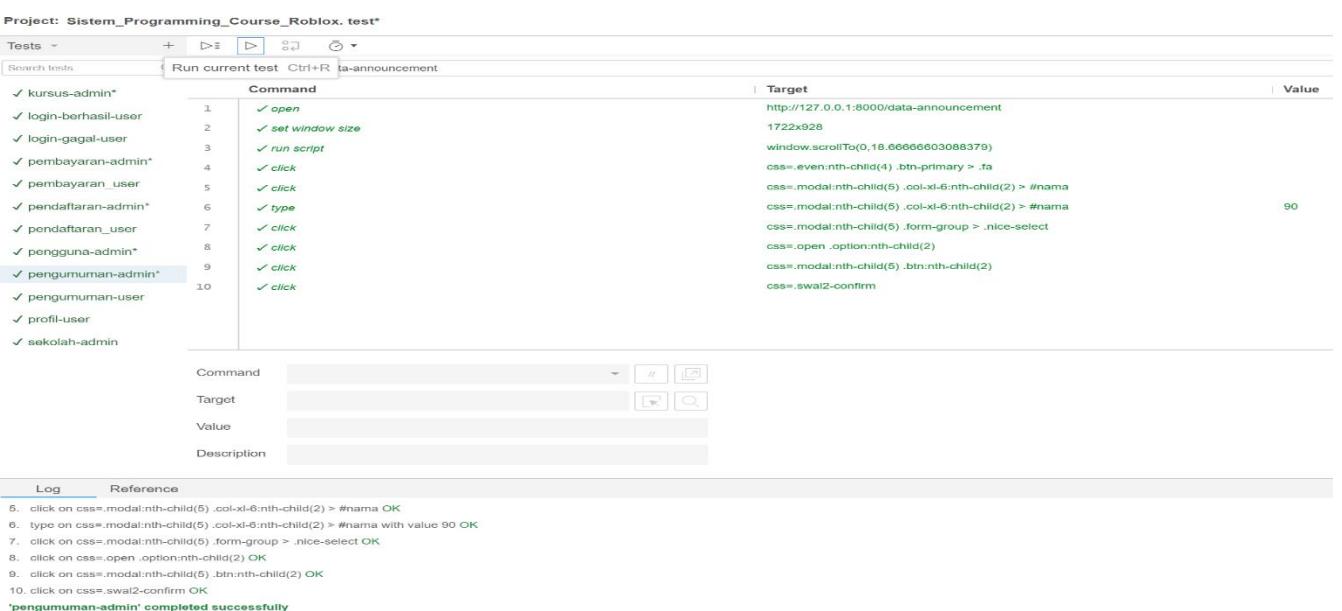
Gambar 4.54 Pengujian Menu Pendaftaran Admin

Pada bagian pengujian menu pendaftaran, admin mengklik menu pendaftaran. Aplikasi memunculkan halaman pendaftaran, selanjutnya admin mengklik tombol seperti *update*, *delete*, dan tambah. Terlihat bahwasannya proses CRUD di menu pendaftaran berhasil “sukses”.



Gambar 4.55 Pengujian Menu Pembayaran Admin

Pada bagian pengujian menu pembayaran, admin mengklik menu pembayaran. Aplikasi memunculkan halaman pembayaran, selanjutnya admin mengklik tombol seperti *edit*. Terlihat bahwasannya proses *edit* menu pembayaran berhasil “sukses”.



Gambar 4.56 Pengujian Menu Pengumuman Admin

Pada bagian pengujian menu pengumuman, admin mengklik menu pengumuman. Aplikasi memunculkan halaman pengumuman, selanjutnya admin mengklik tombol seperti *update*. Terlihat bahwasannya proses *update* di menu pembayaran berhasil “sukses”.

The screenshot shows a test runner interface with the following details:

- Project:** Sistem_Programming_Course_Roblox.test*
- Test Run Status:** The current test is running, indicated by the play button icon in the toolbar.
- Test List:**
 - ✓ konsus-admin*
 - ✓ login-berhasil-user
 - ✓ login-gagal-user
 - ✓ pembayaran-admin*
 - ✓ pembayaran_user
 - ✓ pendaftaran-admin*
 - ✓ pendaftaran_user
 - ✓ pengguna-admin*
 - ✓ pengumuman-admin*
 - ✓ pengumuman-user
 - ✓ profil-admin* (highlighted in blue)
 - ✓ profil-user
- Test Details for profil-admin*:**

Command	Target	Value
1 ✓ open	http://127.0.0.1:8000/profile	
2 ✓ set window size	1722x928	
3 ✓ click	linkText=Pengaturan	
4 ✓ click	name=alamat	
5 ✓ click	name=alamat	
6 ✓ type	name=alamat	Jl. Polonia, No.34, Kota Medan, Sumatera Utara, Indonesia
7 ✓ click	css=.btn:nth-child(9)	
8 ✓ click	css=swal2-confirm	
- Log:**
 - 3. Trying to find linkText=Pengaturan... OK
 - 4. click on name=alamat OK
 - 5. click on name=alamat OK
 - 6. type on name=alamat with value Jl. Polonia, No.34, Kota Medan, Sumatera Utara, Indonesia OK
 - 7. click on css=.btn:nth-child(9) OK
 - 8. click on css=swal2-confirm OK
 - 'profil-admin' completed successfully

Gambar 4.57 Pengujian Menu Profil Admin

Pada bagian pengujian menu pendaftaran, admin mengklik menu profil. Aplikasi memunculkan halaman profil, selanjutnya admin mengedit data dengan mengklik “pengaturan” dan aplikasi memunculkan *notif* “data terubah.”

4.2.1.4 Pengujian *Code Coverage*

```
PS C:\xampp\htdocs\kursus-app> php artisan test
Warning: TTY mode is not supported on Windows platform.

    PASS  Tests\Feature\Auth\PasswordConfirmationTest
✓ confirm password screen can be rendered
✓ password can be confirmed
✓ password is not confirmed with invalid password

    PASS  Tests\Feature\Auth>PasswordResetTest
✓ reset password link screen can be rendered
✓ reset password link can be requested
✓ reset password screen can be rendered
✓ password can be reset with valid token

    PASS  Tests\Feature\Auth\RegistrationTest
✓ registration screen can be rendered

    PASS  Tests\Feature\ExampleTest
✓ example

    PASS  Tests\Feature\FirstTest
✓ example

Tests: 17 passed
Time: 2.25s

PS C:\xampp\htdocs\kursus-app> ./vendor/bin/phpunit
PHPUnit 9.5.27 by Sebastian Bergmann and contributors.

.....
17 / 17 (100%)

Time: 00:02.244, Memory: 36.00 MB

OK (17 tests, 23 assertions)
PS C:\xampp\htdocs\kursus-app>
```

Gambar 4.58 Pengujian *Code Coverage*

Pada gambar diatas dapat diketahui bahwasannya hasil pengujian aplikasi *programming course roblox* sudah berjalan dalam 1 folder yang sudah diuji yakni folder “*kursus-app*”. Jadi dapat disimpulkan, bahwa pengujian yang dititik beratkan terhadap *source code* aplikasi ini sudah berjalan dengan maksimal dengan hasil persentase adalah 100%.

4.2.2 Hasil Pengujian *Role* Peserta dan Admin

Tabel 4.3 Hasil Pengujian *Role* Peserta

Hasil Pengujian <i>Role</i> Peserta				
Menu Uji	Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
<i>Register</i>	Masyarakat klik daftar	Sistem menampilkan halaman <i>register</i>	Sistem berhasil menampilkan halaman <i>register</i>	[x] Berhasil [] Gagal
	Data akun Peserta: email, <i>password</i> , dan nama lengkap	Sistem aplikasi menampilkan pesan “berhasil <i>register</i> ” dan menyimpan data berupa data akun peserta	Sistem berhasil menampilkan pesan “berhasil <i>register</i> ” dan menyimpan data akun peserta	[x] Berhasil [] Gagal
	Data akun Peserta: email, <i>password</i> , dan nama tidak lengkap	Sistem aplikasi menampilkan pesan “Data tidak tersimpan”	Sistem berhasil menampilkan pesan “Data tidak tersimpan”	[x] Berhasil [] Gagal
<i>Login</i>	Klik button <i>login</i>	Sistem akan menampilkan halaman <i>login</i>	Sistem berhasil menampilkan halaman <i>login</i>	[x] Berhasil [] Gagal

	Data akun Peserta: <i>username</i> dan password salah	Sistem menampilkan pesan <i>error</i> “ <i>username</i> dan <i>password</i> salah”	Sistem berhasil menampilkan pesan <i>error</i> “ <i>username</i> dan <i>password</i> salah”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Data akun Peserta: <i>username</i> dan <i>password</i> benar	Sistem mengalihkan pengguna ke halaman beranda	Sistem berhasil mengalihkan pengguna ke halaman beranda	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Pengumuman	Klik menu pengumuman	Sistem menampilkan halaman seleksi pengumuman dan menampilkan informasi pengumuman	Sistem berhasil menampilkan ke halaman seleksi pengumuman dan menampilkan informasi pengumuman	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Cetak hasil pengumuman	Sistem menampilkan <i>pop up</i> untuk melakukan print dan ukuran kertas dalam bentuk pdf	Sistem berhasil menampilkan <i>pop up</i> untuk melakukan print dan ukuran kertas dalam bentuk pdf	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Profil	Klik menu profil	Sistem akan menampilkan informasi profil	Sistem berhasil menampilkan informasi profil	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	<i>Update</i> data profil	Sistem akan menyimpan data	Sistem berhasil menyimpan data	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
<i>Logout</i>	Klik menu <i>logout</i>	Sistem mengalihkan ke halaman utama aplikasi (<i>login</i> dan <i>register</i>)	Sistem menampilkan halaman utama aplikasi (<i>login</i> dan <i>register</i>)	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Tabel 4.4 Hasil Pengujian *Role Admin*

Hasil Pengujian <i>Role Admin</i>				
Menu Uji	Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data Master (Pengguna)	Tambah data sekolah	Sistem akan menampilkan <i>pop up</i> “Tambah Sekolah” dan menyimpan data	Sistem berhasil menampilkan <i>pop up</i> “Tambah Sekolah” dan menyimpan data	[x] Berhasil [] Gagal
	Update data sekolah	Sistem akan menyimpan data perubahan	Sistem berhasil menyimpan data perubahan	[x] Berhasil [] Gagal
Data Master (Sekolah)	Hapus data sekolah	Sistem akan menampilkan pesan “Data Terhapus” dan menyimpannya ke <i>database</i>	Sistem berhasil menampilkan pesan “Data Terhapus” dan menyimpannya ke <i>database</i>	[x] Berhasil [] Gagal
	Tambah data sekolah	Sistem akan menampilkan <i>pop up</i> “Tambah Sekolah” dan menyimpan data	Sistem berhasil menampilkan <i>pop up</i> “Tambah Sekolah” dan menyimpan data	[x] Berhasil [] Gagal

	<i>Update</i> data sekolah	Sistem akan menyimpan data perubahan	Sistem berhasil menyimpan data perubahan	[x] Berhasil [] Gagal
	Hapus data sekolah	Sistem akan menampilkan pesan “Data Terhapus” dan menyimpannya ke <i>database</i>	Sistem berhasil menampilkan pesan “Data Terhapus” dan menyimpannya ke <i>database</i>	[x] Berhasil [] Gagal
Data <i>Master</i> (Program Kursus)	Klik menu program kursus	Sistem mengalihkan ke halaman menu pengguna	Sistem berhasil menampilkan ke halaman menu pengguna	[x] Berhasil [] Gagal
	Tambah program kursus	Sistem akan menampilkan <i>pop up</i> “Tambah program kursus” dan menyimpan data	Sistem berhasil menampilkan <i>pop up</i> “Tambah program kursus” dan menyimpan data	[x] Berhasil [] Gagal
	<i>Update</i> data program kursus	Sistem akan menyimpan data perubahan	Sistem berhasil menyimpan data perubahan	[x] Berhasil [] Gagal
	Hapus data program kursus	Sistem akan menampilkan pesan “Data Terhapus” dan menyimpannya ke <i>database</i>	Sistem berhasil menampilkan pesan “Data Terhapus” dan menyimpannya ke <i>database</i>	[x] Berhasil [] Gagal
Data <i>History</i> : Pendaftaran	Klik menu pendaftaran	Sistem akan menampilkan halaman pendaftaran dan menampilkan informasi pendaftaran	Sistem berhasil menampilkan halaman pendaftaran dan menampilkan informasi pendaftaran	[x] Berhasil [] Gagal

	Tambah data pendaftaran	Sistem akan menyimpan data pendaftaran	Sistem berhasil menyimpan data pendaftaran	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Update data pendaftaran	Sistem akan menyimpan data perubahan pendaftaran	Sistem berhasil menyimpan data perubahan pendaftaran	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Hapus data pendaftaran	Sistem akan menampilkan pesan “Data Terhapus” dan menyimpannya ke database	Sistem berhasil menampilkan pesan “Data Terhapus” dan menyimpannya ke database	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Data History: Pembayaran	Klik menu pembayaran	Sistem mengalihkan ke halaman pembayaran	Sistem berhasil mengalihkan ke halaman pembayaran	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Update data pembayaran	Sistem akan menyimpan data perubahan	Sistem berhasil menyimpan data perubahan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
Pengumuman	Klik menu pengumuman	Sistem akan mengalihkan ke halaman pengumuman dan menampilkan informasi pengumuman	Sistem berhasil mengalihkan ke halaman pengumuman dan menampilkan informasi pengumuman	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Update data pengumuman	Sistem akan menampilkan pesan “Data Terubah” dan menyimpannya kedalam database	Sistem berhasil menampilkan pesan “Data Terubah” dan menyimpannya kedalam database	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Profil	Klik menu profil	Sistem mengalihkan ke halaman profil	Sistem berhasil menampilkan halaman profil	[x] Berhasil [] Gagal
	<i>Update</i> data profil	Sistem akan menampilkan pesan “Data Terubah” dan menyimpannya kedalam <i>database</i>	Sistem berhasil menampilkan pesan “Data Terubah” dan menyimpannya kedalam <i>database</i>	[x] Berhasil [] Gagal
<i>Logout</i>	Klik menu <i>logout</i>	Sistem mengalihkan ke halaman utama	Sistem berhasil menampilkan halaman utama	[x] Berhasil [] Gagal

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah dirancangnya aplikasi sistem pendaftaran *programming course* roblox ini maka dapat diambil kesimpulan sebagai berikut:

- 1.) Dengan adanya aplikasi *system* pendaftaran *programming course* roblox ini, peserta kursus tidak perlu mendaftar kursus secara langsung untuk melakukan pendaftaran karena dapat dilakukan secara *online* melalui *website*.
- 2.) Dengan menerapkan *framework* laravel, dapat diketahui tentang pembuatan aplikasi tersebut beserta cara kerjanya.
- 3.) Dengan menerapkan *database* Mysql, pengelolahan data berkas pendaftaran menjadi lebih terstruktur, dan efisien serta mempersingkat waktu.

5.2 Saran

Dalam membangun aplikasi ini, masih memiliki banyak kekurangan sehingga dapat diusulkan sebagai berikut:

- 1.) Diperlukan pengembangan mengenai tambahan fitur fitur atau menu menu tambahan yang ada didalam aplikasi sehingga terkesan lebih menarik khususnya bagi pengguna yang melihat dan mengaksesnya.
- 2.) Perlu ditingkatkan juga dalam segi media pendidikan berbasis permainan ini, sehingga dapat mengedukasi lebih dalam pengetahuan pemrograman *game*.
- 3.) Perlu adanya penambahan sistem pembayaran melalui *virtual account* yang dikirim ke email peserta agar peserta dapat melihat *notice* informasi pembayarannya.

DAFTAR PUSTAKA

- [1] Zulia Karini, Devi Brilliane, “Perancangan *E-Learning* Bahasa Inggris Berbasis *Web* di UPT BAHASA STMIK Amikom Purwokerto”, vol. 1-3, 2018, <https://doi.org/10.1063/1.5062769>
- [2] Dyah Rhetno Wardhani, Adhityo Kuncoro, “Perancangan Sistem Pembelajaran *E-Learning* Pada Kursus *Online* dengan Metode *Web Based Learning* Menggunakan Dokeos di SMK Nasional Depok”, vol. 18, 2019, <https://doi.org/10.1063/1.5062769>
- [3] Dwi Budhi Srisulistiwati, “Perancangan Sistem Informasi Kursus Di Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika”, vol. 6, 2005, <https://doi.org/10.1063/1.5062769>
- [4] Arisantoso, Bambang Mulyatno, dkk. “Analisis Dan Rancang Bangun Aplikasi Kursus *On-line* Menggunakan Pendekatan Model Diagram Alir Data (Studi Kasus: Universitas Islam Attahiriyah”, vol. 4, Juli 2018, <https://doi.org/10.1063/1.5062769>
- [5] Ayu Ramadhani, Hashfi Qaedi Yusman, dkk. “Aplikasi *Online Course* “Waroeng Inggris” Berbasis *Web* dengan Menggunakan *Framework Codeigniter*”, vol.10, <https://doi.org/10.1063/1.5062769>
- [6] Azhar. 2011. Model Pembelajaran Kewirausahaan Pada PKBM Binaan SKB Kabupaten Temanggung. Jurnal Kependidikan. 41. (1). 11-22.
- [7] Wardani, A.I.K., 2019. Rancang Bangun Sistem Informasi *Raport Online* Berbasis *Website* Menggunakan *Framework Laravel*. Jurnal Manajemen Informatika, 10.
- [8] D. F. Ningtyas and N. Setiyawati, “Implementasi Laravel *Framework* pada Pembangunan Aplikasi *Purchasing Approval Request* Laravel *Framework Implementation in Development Purchasing Approval Request Application*,” J. Janitra Inform. dan Sist. Inf., vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra. v1i1.120.
- [9] M. Singh, A. Verma, A. Parasher, N. Chauhan, and G. Budhiraja, “*Implementation of Database Using Python Laravel Framework*,” International Journal of Engineering and Computer Science, vol. 8, no. 12. pp. 24890– 24893, 2019, doi: 10.18535/ijecs/v8i12.4399.
- [10] Wardani, A.I.K., 2019. Rancang Bangun Sistem Informasi *Raport Online* Berbasis *Website* Menggunakan *Framework Laravel*. Jurnal Manajemen Informatika.

- [11] Huda, M, dan Bunafit K, 2010, Membuat Aplikasi *Database* Dengan Java, Mysql Dan Netbeans, PT. Elexmedia Komputindo, Jakarta. Wahana Komputer, 2010, Pengembangan aplikasi JavaDB dengan NetBeans, C.V Andi Offset, Semarang
- [12] Supriyatna, A. (2018). Metode *Extreme Programming* Pada Pembangunan *Web* Aplikasi Seleksi Peserta Pelatihan Kerja. Jurnal Teknik Informatika, 11(1), 1–18. <https://doi.org/10.15408/jti.v11i1.6628>
- [13] Levi. 2013. Model *Extreme Programming*. [Online]. (Diakses <https://levi91.wordpress.com/2013/03/17/modelExtreme-programmingxp/>, tanggal 23 Maret 2015).
- [14] Keina. 2013. *Extreme Programming*. (Diakses (<https://keinatralala.wordpress.com/2013/12/13/metodologi extreme programming>, tanggal 24 Maret 2015).
- [15] Khamidah, K., & Triyono, R. A. (2013). Pengembangan Aplikasi *E-Learning* Berbasis *Web* Dengan PHP Dan MySQL Studi Kasus SMPN 1 Arjosari. *Indonesian Jurnal on Networking and Security* (IJNS).
- [16] Faridi, Peni Arianti dan Retno Widuri. 2016. Perancangan Sistem Informasi E-Jurnal pada Perguruan Tinggi Berbasis *Web*. Jurnal CERITA. Tangerang: Perguruan Tinggi Raharja. ISSN: 2461-1417. Vol.2 No.2-Agustus 2016.
- [17] Purbadian, Yenda. 2016. Trik Cepat Membangun Aplikasi Berbasis *Web* dengan *Framework CodeIgniter*. Yogyakarta: CV Andi Offset.
- [18] Anggelia, Gressi. Pengembangan Sistem Informasi Administrasi Kursus Bahasa Inggris Berbasis *Web* (Studi Kasus: *Wall Street Institute*, Pondok Indah). Skripsi, Jakarta: Universitas Islam Negeri Syarif Hidayatullah Jakarta, 2011.
- [19] Edi, D., & Betshani, S. (2009). Analisis Data dengan Menggunakan ERD dan Model Konseptual Data *Warehouse*. Jurnal Informatika, 75-76.
- [20] Dewi, R. (2015). Perancangan Aplikasi *E-Learning* Berbasis *Website* Pada SMA/SMK Dharma Bakti Medan. Konferensi Nasional Sistem & Informatika 2015.
- [21] Jenry Masudara, 2014 “Merancang bangun sistem informasi perangkat lunak pemetaan dan pemantauan DAS Tondano di Kota Manado berbasis *web* dan *mobile web*”.
- [22] A. Firman, H. F. Wowor, andX. Najoan. 2016 “Sistem Informasi Perpustakaan Online Berbasis *Web*”, Vol. 5, No. 2. [6] V. N. Nore. 2013” Perancangan Sistem Informasi Penjualan Pemesanan Produk Berbasis *Web*”, Universitas Widyatama, Bandung.

- [23] Dewanto, “*Planning Planning Analysis Analysis Detailed Detailed System System Design Design Implementation Implementation Maintenance Maintenance*,” Fasilkom, vol. 2, no. 1, 2004.
- [24] Islamiyah, M, dan Widayanti, L. (2016). Efektifitas Pemanfaatan *E-Learning* berbasis *Website* Terhadap Hasil Belajar Mahasiswa STMIK Asia Malang Pada Mata Kuliah Fisika Dasar. *Jurnal Ilmiah Teknologi dan Informasi ASIA (JITIKA)*, ISSN: 0852-730X, Vol.10, No.1, hal 41.
- [25] Buana, I Komang Setia. 2014. Jago pemrograman PHP. Dunia Komputer, Jakarta, Indonesia.
- [26] Jeine Enjelina Sarite, 2014 “Perancangan dan Implementasi Sistem Pelayanan Berbasis Teknologi Informasi Komunikasi di Gereja Mawar Sharon Manado “, Vol. No.12, April 2018
- [27] Khamidah, K., & Triyono, R. A. (2013). Pengembangan Aplikasi *E-Learning* Berbasis *Web* Dengan PHP Dan MySql Studi Kasus SMPN 1 Arjosari. *Indonesian Jurnal on Networking and Security (IJNS)*.
- [28] Kusumaningrum, R. (2016). Sistem Pembelajaran *E-Learning* Berbasis *Web*. Informatika, Fakultas Komunikasi dan Informatika.
- [29] Kristanto, Andri, (2008), Perancangan Sistem Informasi dan Aplikasinya, edisi revisi, Yogyakarta: Gava Media
- [30] Salamun. (2017). Sistem Monitoring Nilai Siswa Berbasis Android. *Jurnal Teknologi dan Sistem Informasi Univrab*, ISSN: 2502-819X, Vol. 2, No. 2, hal 213.

LAMPIRAN



Lampiran 1. Poster Infografis

The screenshot shows the submission dashboard for the Jurnal Teknik Informatika. The top navigation bar includes links for English, View Site, and user jasminemutiara03. The main area displays the submission details for entry 2777, titled 'IMPLEMENTASI FRAMEWORK LARAVEL PADA SISTEM PENDAFTARAN PROGRAMMING COURSE ROBLOX'. The workflow tabs are 'Workflow' and 'Publication', with 'Publication' selected. Below these are sub-tabs: 'Submission', 'Review', 'Copyediting', and 'Production'. A 'Submission Files' section lists a file named 'jasminemutiara03, IMPLEMENTASI FRAMEWORK LARAVEL PADA SISTEM PENDAFTARAN PROGRAMMING COURSE ROBLOX.pdf (2)' with a download link. The date 'February 3, 2023' is also shown. A 'Search' button is available in the top right of the submission files section.

Lampiran 2. Submit Jurnal Teknik Informatika