



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IS184853

**RANCANG BANGUN SISTEM RAPOR YANG
TERHUBUNG DENGAN APLIKASI PERMAINAN
ANDROID (STUDI KASUS: PAUD CILUK-BAA)**

***DEVELOPMENT OF REPORT CARD SYSTEM THAT
CONNECTED WITH ANDROID GAME (CASE
STUDY: CILUK-BAA ECE)***

**KRISNA FAWWASTAQI RAHEKO
NRP. 0521164000020**

**Dosen Pembimbing
Nisfu Asrul Sani, S.Kom, M.Sc.**

**DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020**



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IS184853

RANCANG BANGUN SISTEM RAPOR YANG TERHUBUNG DENGAN APLIKASI PERMAINAN ANDROID (STUDI KASUS: PAUD CILUK-BAA)

KRISNA FAWWASTAQI RAHEKO
NRP. 05211640000020

Dosen Pembimbing
Nisfu Asrul Sani, S.Kom, M.Sc.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan



ITS
Institut
Teknologi
Sepuluh Nopember

UNDERGRADUATE THESIS - IS184853

DEVELOPMENT OF REPORT CARD SYSTEM THAT CONNECTED WITH ANDROID GAME (CASE STUDY: CILUK-BAA ECE)

KRISNA FAWWASTAQI RAHEKO
NRP. 052116400000020

Dosen Pembimbing
Nisfu Asrul Sani, S.Kom, M.Sc.

DEPARTMENT OF INFORMATION SYSTEMS
Faculty of Electrical and Intelligent Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN**Rancang Bangun Sistem Rapor yang Terhubung dengan Aplikasi Permainan Android (Studi Kasus: PAUD Ciluk-Baa)****TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer (S.Kom)
pada

Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas (ELECTICS)
Institut Teknologi Sepuluh Nopember

Oleh

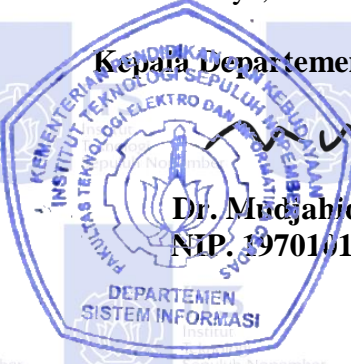
Krisna Fawwastaqi Raheko

05211640000020

Surabaya, 14 Agustus 2020

Kepala Departemen Sistem Informasi

**Dr. Mudjahidin, ST., MT.
NIP. 197010102003121001**



Halaman ini sengaja dikosongkan

LEMBAR PERSETUJUAN

RANCANG BANGUN SISTEM RAPOR YANG TERHUBUNG DENGAN APLIKASI PERMAINAN ANDROID (STUDI KASUS: PAUD CILUK-BAA)

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh :

KRISNA FAWWASTAQI RAHEKO
NRP. 0521164000020

Disetujui Tim Penguji : Tanggal Ujian : 09 Juli 2020
Periode Wisuda : September 2020

Nisfu Asrul Sani, S.Kom., M.Sc.

(Pembimbing I)

Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.

(Penguji I)

Bekti Cahyo Hidayanto, S.Si., M. Kom.

(Penguji II)

Halaman ini sengaja dikosongkan

RANCANG BANGUN SISTEM RAPOR YANG TERHUBUNG DENGAN APLIKASI PERMAINAN ANDROID (STUDI KASUS: PAUD CILUK-BAA)

Nama : Krisna Fawwastaqi Raheko
NRP : 0521164000020
Departemen : Sistem Informasi ITS
Pembimbing I : Nisfu Asrul Sani, S.Kom., M.Sc.

ABSTRAK

PAUD Ciluk-Baa adalah PAUD (Pendidikan Anak Usia Dini) swasta yang berlokasi di Tulungagung dan menggunakan rapor dalam bentuk buku cetak. Kegiatan berkaitan rapor yang dilakukan secara konvensional menimbulkan masalah yaitu: guru melewatkan kesempatan mengamati perkembangan anak, guru lupa menuliskan hasil pengamatan pada rapor, dan wali murid tidak dapat meminjam serta mengembalikan rapor di luar jam kerja.

Rancang bangun sistem rapor adalah pengembangan aplikasi berbasis web untuk melakukan proses bisnis PAUD Ciluk-Baa yang berkaitan dengan rapor. Sistem ini dikembangkan untuk mengatasi masalah mengenai batasan jam dan hari kerja. Guru dapat mengisi rapor kapan saja dan dimana saja dengan mengakses sistem melalui jaringan. Serta wali murid dapat melihat rapor kapan saja.

Aplikasi permainan yang dikembangkan didesain untuk memenuhi indikator berhitung pada rapor. Aplikasi permainan akan dihubungkan menggunakan RESTful API untuk mengisi rapor pada setiap akhir permainan.

Jadi terdapat dua modul yang akan dikembangkan melalui dua metode yang berbeda secara berdampingan yaitu sistem rapor dan aplikasi permainan Android. Modul sistem rapor akan

dikembangkan menggunakan SDLC (Software Development Life Cycle) iterative dengan 5 tahapan yaitu: analisis kebutuhan, desain, pengembangan, pengujian sistem, dan evaluasi. Modul aplikasi permainan Android akan dikembangkan menggunakan GDLC (Game Development Life Cycle) dengan 5 tahapan yaitu: inisiasi, pra produksi, produksi, pengujian permainan, dan beta. Keduanya memiliki jumlah tahapan yang sama sehingga dapat dijalankan bersamaan dalam 5 fase. Setelah 5 fase selesai dan telah diterima oleh pihak PAUD Ciluk-Baa maka pengembangan diakhiri dan berlanjut ke penyelesaian dokumen.

Penelitian tugas akhir ini berhasil memberikan luaran berupa sistem rapor berbasis web yang sesuai dengan proses bisnis PAUD Ciluk-Baa dan dapat menerima masukan manual melalui web maupun otomatis dari aplikasi permainan Android. Keduanya telah diuji ke pengguna yang “Sangat Setuju” terhadap pernyataan positif untuk kriteria efektifitas dan kepuasan.

Kata kunci: aplikasi permainan Android, PAUD, RESTful API, sistem rapor

DEVELOPMENT OF REPORT CARD SYSTEM THAT CONNECTED WITH ANDROID GAME (CASE STUDY: CILUK-BAA ECE)

Name : Krisna Fawwastaqi Raheko
NRP : 0521164000020
Department : Information Systems ITS
Supervisor I : Nisfu Asrul Sani, S.Kom., M.Sc.

ABSTRACT

Ciluk-Baa ECE is a private ECE (Early Childhood Education) located in Tulungagung and uses report cards in the form of printed books. Report-related activities carried out conventionally cause problems, namely: the teacher miss the opportunity to observe the child's development, the teacher forgets to write down the observations to the report card, and the student's guardian cannot borrow and return the report card outside working hours.

The design of the report card system is the development of web-based applications for conducting the Ciluk-Baa ECE business processes related to report cards. This system was developed to overcome the problem of limited hours and working days. Teachers can fill out report cards anytime and anywhere by accessing the system through the network. And guardians of students can see the report card at any time.

The game developed is designed to meet the numeracy indicators on report cards. Game will be linked using the RESTful API to fill in report cards at the end of each stage.

So there are two modules that will be developed through two different methods side by side, namely the report card system and the Android game. Report card system modules will be developed using an iterative SDLC (Software Development Life Cycle) with 5 stages: requirements analysis, design,

development, system testing, and evaluation. The Android game application module will be developed using GDLC (Game Development Life Cycle) with 5 stages: initiation, pre production, production, game testing, and beta. Both have the same number of stages so they can be run together in 5 phases. After the 5 phases are completed and have been accepted by the ECE Ciluk-Baa, the development is ended and continues to complete the document.

This research succeeded in providing an output in the form of a web-based report card system that is in line with the Ciluk-Baa ECE business process and can receive manual input via the web or automatically from Android game. Both have been tested to users who "Strongly Agree" to the positive statements for the criteria of effectiveness and satisfaction.

Keywords: Android game, ECE, report card system, RESTful API

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertandatangan di bawah ini:

Nama : Krisna Fawwastaqi Raheko
NRP : 05211640000020
Tempat / Tanggal lahir : Tulungagung / 02 Nopember 1997
Fakultas / Departemen : FTEIC / Sistem Informasi
Nomor Telp/Hp/email : 082264270301/
krisnafawwastaqi@gmail.com

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian/makalah/tugas akhir saya yang berjudul

RANCANG BANGUN SISTEM RAPOR YANG TERHUBUNG DENGAN APLIKASI PERMAINAN ANDROID (STUDI KASUS: PAUD CILUK-BAA)

Bebas Dari Plagiarisme Dan Bukan Hasil Karya Orang Lain.

Apabila dikemudian hari ditemukan seluruh atau sebagian penelitian/makalah/tugas akhir tersebut terdapat indikasi plagiarisme, maka saya bersedia menerima sanksi sesuai peraturan dan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya.

Surabaya, Juni 2020



METERAI
TEMPEL
TGL. 20
8F691AFF616453743
6000
ENAM RIBU RUPIAH

Krisna Fawwastaqi Raheko
NRP.05211640000020

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan kekuatan serta hidayah-Nya kepada penulis sehingga penulis dapat menyelesaikan tugas akhir ini yang merupakan salah satu syarat kelulusan di Departemen Sistem Informasi Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada :

1. Kedua orang tua serta adik penulis yang terus mendukung kelancaran pengerjaan tugas akhir ini.
2. Bapak Dr. Mudjahidin, S.T., M.T. selaku Kepala Departemen Sistem Informasi ITS.
3. Bapak Nisfu Asrul Sani, S. Kom., M.Sc. selaku dosen pembimbing yang telah meluangkan waktu untuk mengarahkan dengan memberikan petunjuk dalam penyelesaian tugas akhir ini.
4. Bapak Dr.Eng. Febriliyan Samopa, S. Kom., M. Kom. dan Bapak Bakti Cahyo Hidayanto, S.Si., M. Kom. selaku dosen penguji yang telah memberikan kritik dan saran dalam penyempurnaan tugas akhir ini.
5. Pihak PAUD Ciluk-Baa diantaranya pengelola, guru, dan wali murid yang menjadi studi kasus untuk tugas akhir ini.
6. Shabrina Luthfiani Khanza yang telah membuatkan aset game tugas akhir ini.
7. Teman-teman Sistem Informasi angkatan 2016 (Artemis) yang senantiasa menemani dan memberikan motivasi bagi penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.

8. Seluruh pihak-pihak lainnya yang tidak dapat disebutkan satu per satu, yang telah membantu penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.

Penyusunan laporan tugas akhir ini masih jauh dari kata sempurna, sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, Juni 2020

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	V
LEMBAR PERSETUJUAN	VII
ABSTRAK	IX
ABSTRACT	XI
KATA PENGANTAR.....	XV
DAFTAR ISI	XVII
DAFTAR GAMBAR.....	XXI
DAFTAR TABEL	XXV
DAFTAR KODE	XXVII
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Relevansi.....	4
2 BAB II TINJAUAN PUSTAKA.....	5
2.1 Studi Literatur	5
2.2 Dasar Teori.....	7
2.2.1 Rapor PAUD Ciluk-Baa.....	7
2.2.2 Proses <i>Iconix</i>	8
2.2.3 Laravel	9
2.2.4 RESTful API.....	9
2.2.5 Aplikasi Permainan Android.....	10
3 BAB III METODOLOGI	11
3.1 Fase 1.....	12
3.1.1 Analisis Kebutuhan	12
3.1.2 Inisiasi	12
3.2 Fase 2.....	13
3.2.1 Desain.....	13
3.2.2 Pra Produksi	14
3.3 Fase 3.....	14
3.3.1 Pengembangan	14
3.3.2 Produksi	14
3.4 Fase 4.....	14

3.4.1	Pengujian Sistem	15
3.4.2	Pengujian Permainan	15
3.5	Fase 5.....	15
3.5.1	Evaluasi	15
3.5.2	Beta.....	16
3.6	Dokumen Tugas Akhir.....	16
4	BAB IV PERANCANGAN	17
4.1	Fase 1.....	17
4.1.1	Analisis Kebutuhan (Sistem Rapor)	17
4.1.1.1	Diagram Proses Bisnis.....	18
4.1.1.2	Daftar Kebutuhan Fungsional	22
4.1.1.3	Domain Model.....	23
4.1.1.4	Storyboard	24
4.1.2	Inisiasi (Aplikasi Permainan)	30
4.1.2.1	Konsep permainan	30
4.2	Fase 2.....	30
4.2.1	Desain (Sistem Rapor).....	30
4.2.1.1	Use Case Diagram	30
4.2.1.2	Desain RESTful API	40
4.2.1.3	Robustness Diagram	43
4.2.1.4	Sequence Diagram	49
4.2.1.5	Class Diagram.....	59
4.2.2	Pra Produksi (Aplikasi Permainan)	60
4.2.2.1	Game Design	60
4.2.2.2	Mockup Game	63
5	BAB V IMPLEMENTASI.....	69
5.1	Lingkungan Implementasi.....	69
5.2	Fase 3.....	70
5.2.1	Pengembangan (Sistem Rapor)	70
5.2.1.1	Implementasi Database	70
5.2.1.2	Implementasi Model	73
5.2.1.3	Implementasi View.....	76
5.2.1.4	Implementasi Controller	80
5.2.1.5	Implementasi Route.....	86
5.2.2	Produksi (Aplikasi Permainan).....	88
5.2.2.1	Implementasi Login	88
5.2.2.2	Implementasi Menu	90
5.2.2.3	Implementasi Game	93
5.3	Lingkungan Pengujian.....	98
5.4	Fase 4.....	99
5.4.1	Pengujian Sistem (Sistem Rapor).....	99

5.4.1.1	Unit Test.....	99
5.4.1.2	System Test.....	106
5.4.2	Pengujian Permainan (Aplikasi Permainan).....	109
5.4.2.1	Functional Test.....	109
6	BAB VI HASIL DAN PEMBAHASAN.....	111
6.1	Fase 5.....	111
6.1.1	Evaluasi (Sistem Rapor).....	112
6.1.2	Beta (Aplikasi Permainan).....	114
7	BAB VII KESIMPULAN DAN SARAN.....	117
7.1.	Kesimpulan.....	117
7.2.	Saran.....	117
8	DAFTAR PUSTAKA.....	119
	BIODATA PENULIS.....	127

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1 Halaman Rapor yang Telah Terisi.....	8
Gambar 2.2 Proses Iconix	9
Gambar 3.1 Tahapan Pelaksanaan Tugas Akhir	11
Gambar 4.1 Desain Sistem.....	17
Gambar 4.2 Diagram Proses Mencetak Rapor	18
Gambar 4.3 Diagram Proses Pendaftaran Murid.....	18
Gambar 4.4 Diagram Proses Pembelajaran Harian	19
Gambar 4.5 Diagram Proses Rekapitulasi Rapor.....	20
Gambar 4.6 Diagram Proses Serah Terima Rapor	20
Gambar 4.7 Diagram Proses Rapat Bulanan.....	21
Gambar 4.8 Diagram Proses Perubahan Kurikulum	21
Gambar 4.9 Domain Model.....	23
Gambar 4.10 Storyboard Halaman Masuk.....	24
Gambar 4.11 Storyboard Halaman Daftar Murid.....	24
Gambar 4.12 Storyboard Halaman Profil Murid.....	25
Gambar 4.13 Storyboard Halaman Ubah Profil Murid	25
Gambar 4.14 Storyboard Halaman Mendaftar	26
Gambar 4.15 Storyboard Halaman Daftar Murid Kelas.....	26
Gambar 4.16 Storyboard Halaman Rapor	27
Gambar 4.17 Storyboard Halaman Isi Rapor	27
Gambar 4.18 Storyboard Halaman Daftar Indikator	28
Gambar 4.19 Storyboard Modal Tambah Indikator	28
Gambar 4.20 Storyboard Modal Ubah Indikator.....	29
Gambar 4.21 Storyboard Halaman Ubah Password.....	29
Gambar 4.22 Use Case Diagram Pengguna Tata Usaha	30
Gambar 4.23 Use Case Diagram Pengguna Murid	31
Gambar 4.24 Use Case Diagram Pengguna Kelas	31
Gambar 4.25 Use Case Diagram Pengguna Pengelola.....	31
Gambar 4.26 Use Case Diagram Pengguna Admin	31
Gambar 4.27 Use Case Diagram API.....	41
Gambar 4.28 Robustness Diagram Masuk.....	43
Gambar 4.29 Robustness Diagram Keluar.....	43
Gambar 4.30 Robustness Diagram Melihat Data Murid	43
Gambar 4.31 Robustness Diagram Mengisi Data Murid	44
Gambar 4.32 Robustness Diagram Mengubah Data Murid ...	44

Gambar 4.33 Robustness Diagram Menghapus Data Murid ..	44
Gambar 4.34 Robustness Diagram Melihat dan Mencetak Rapor	45
Gambar 4.35 Robustness Diagram Mengisi Rapor	45
Gambar 4.36 Robustness Diagram Mengubah Rapor	45
Gambar 4.37 Robustness Diagram Menaikkan Murid	46
Gambar 4.38 Robustness Diagram Mengeluarkan Murid	46
Gambar 4.39 Robustness Diagram Melihat Indikator	46
Gambar 4.40 Robustness Diagram Menambahkan Indikator	47
Gambar 4.41 Robustness Diagram Mengubah Indikator	47
Gambar 4.42 Robustness Diagram Menghapus Indikator	47
Gambar 4.43 Robustness Diagram Masuk Melalui Game	48
Gambar 4.44 Robustness Diagram Mengisi Rapor Melalui Game.....	48
Gambar 4.45 Robustness Diagram Menyimpan Hasil Bermain	48
Gambar 4.46 Robustness Diagram Mengirim Hasil Bermain	48
Gambar 4.47 Sequence Diagram Masuk	49
Gambar 4.48 Sequence Diagram Keluar	49
Gambar 4.49 Sequence Diagram Melihat Data Murid	50
Gambar 4.50 Sequence Diagram Mengisi Data Murid	50
Gambar 4.51 Sequence Diagram Mengubah Data Murid	51
Gambar 4.52 Sequence Diagram Menghapus Data Murid	51
Gambar 4.53 Sequence Diagram Melihat Rapor (Murid)	52
Gambar 4.54 Sequence Diagram Melihat Rapor (Kelas)	52
Gambar 4.55 Sequence Diagram Mencetak Rapor.....	52
Gambar 4.56 Sequence Diagram Mengisi Rapor	53
Gambar 4.57 Sequence Diagram Mengubah Rapor	53
Gambar 4.58 Sequence Diagram Menaikkan Murid	54
Gambar 4.59 Sequence Diagram Mengeluarkan Murid	54
Gambar 4.60 Sequence Diagram Melihat Indikator	55
Gambar 4.61 Sequence Diagram Menambahkan Indikator.....	55
Gambar 4.62 Sequence Diagram Mengubah Indikator	56
Gambar 4.63 Sequence Diagram Menghapus Indikator.....	56
Gambar 4.64 Sequence Diagram Mengubah Password.....	57
Gambar 4.65 Sequence Diagram Masuk Melalui Game	57
Gambar 4.66 Sequence Diagram Mengisi Rapor Melalui Game	58

Gambar 4.67 Sequence Diagram Menyimpan Hasil Bermain	58
Gambar 4.68 Sequence Diagram Mengirim Hasil Bermain...	58
Gambar 4.69 Class Diagram Sistem Rapor.....	59
Gambar 4.70 Mockup Login	63
Gambar 4.71 Mockup Menu	63
Gambar 4.72 Mockup Game Tipe 1	64
Gambar 4.73 Mockup Game Tipe 2.....	65
Gambar 4.74 Mockup Game Tipe 3.....	65
Gambar 4.75 Mockup Game Tipe 4.....	66
Gambar 5.1 File Migration Sistem Rapor	71
Gambar 5.2 File Seeder Sistem Rapor	71
Gambar 5.3 File Model Sistem Rapor.....	74
Gambar 5.4 File Blade View Sistem Rapor	77
Gambar 5.5 Halaman Masuk.....	79
Gambar 5.6 Halaman Tambah Rapor.....	79
Gambar 5.7 Halaman Rapor.....	79
Gambar 5.8 Halaman Daftar Murid Kelas	80
Gambar 5.9 File Controller Sistem Rapor.....	80
Gambar 5.10 Implementasi Tampilan Login	88
Gambar 5.11 Implementasi Tampilan Menu.....	90
Gambar 5.12 Implementasi Tampilan Game Tipe 1	93
Gambar 5.13 Implementasi Game Tipe 2 Bagian 3	97
Gambar 5.14 Implementasi Game Tipe 3 Bagian 4	98
Gambar 5.15 Implementasi Game Tipe 4	98
Gambar 5.16 File Unit Test.....	100
Gambar 5.17 Hasil Unit Test.....	105
Gambar 5.18 Pengujian API Masuk di Postman.....	108
Gambar 5.19 Pengujian API Masuk di Debug Consruct 3...	108
Gambar 5.20 Pengujian Mengelola Bagian pada Tampilan Menu	110
Gambar 5.21 Pengujian Mengakhiri Bagian pada Tampilan Game	110
Gambar 6.1 UAT Skenario Mengisi Rapor.....	113
Gambar 6.2 Pernyataan Skenario Mengisi Rapor	113

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Studi literatur.....	5
Tabel 4.1 Kebutuhan Fungsional Sistem Rapor	22
Tabel 4.2 Use Case Description Masuk	32
Tabel 4.3 Use Case Description Keluar	32
Tabel 4.4 Use Case Description Melihat Data Murid	33
Tabel 4.5 Use Case Description Mengisi Data Murid.....	33
Tabel 4.6 Use Case Description Mengubah Data Murid.....	34
Tabel 4.7 Use Case Description Menghapus Data Murid.....	35
Tabel 4.8 Use Case Description Melihat Rapor	35
Tabel 4.9 Use Case Description Mencetak Rapor.....	36
Tabel 4.10 Use Case Description Mengisi Rapor	36
Tabel 4.11 Use Case Description Mengubah Rapor	37
Tabel 4.12 Use Case Description Menaikkan Murid	37
Tabel 4.13 Use Case Description Mengeluarkan Murid	38
Tabel 4.14 Use Case Description Melihat Indikator	38
Tabel 4.15 Use Case Description Menambahkan Indikator...	38
Tabel 4.16 Use Case Description Mengubah Indikator	39
Tabel 4.17 Use Case Description Menghapus Indikator	39
Tabel 4.18 Use Case Description Mengubah Password.....	40
Tabel 4.19 Use Case Description Masuk Melalui Game	41
Tabel 4.20 Use Case Description Mengisi Rapor Melalui Game	41
Tabel 4.21 Use Case Description Menyimpan Hasil Bermain	42
Tabel 4.22 Use Case Description Mengirim Hasil Bermain ..	42
Tabel 4.23 Informasi Bagian pada Permainan	61
Tabel 4.24 Kebutuhan Fungsional Aplikasi Permainan	62
Tabel 4.25 Mekanisme Game Edukasi Android Komersil.....	66
Tabel 6.1 Nilai Interval	111
Tabel 6.2 Pernyataan UAT Sistem Rapor	112
Tabel 6.3 Hasil UAT Sistem Rapor	114
Tabel 6.4 Hasil UAT Aplikasi Permainan ke Guru.....	115
Tabel 6.5 Hasil UAT Aplikasi Permainan ke Pendamping..	116
Tabel 8.1 Rekapitulasi UAT Sistem Rapor	121
Tabel 8.2 Rekapitulasi UAT Aplikasi Permainan ke Guru ..	123

Tabel 8.3 Rekapitulasi UAT Aplikasi Permainan ke Wali Murid
.....125

DAFTAR KODE

Kode 5.1 Skema Migrasi Tabel Rapor	71
Kode 5.2 PenggunaSeeder.....	72
Kode 5.3 RaporSeeder.....	72
Kode 5.4 MuridSeeder	72
Kode 5.5 IndikatorSeeder.....	73
Kode 5.6 HasilSeeder.....	73
Kode 5.7 Model Pengguna.....	74
Kode 5.8 Model Rapor.....	75
Kode 5.9 Model Murid.....	76
Kode 5.10 Model Indikator	76
Kode 5.11 Model Hasil	76
Kode 5.12 Form Select2 Indikator Halaman Tambah Rapor ..	77
Kode 5.13 Halaman Masuk.....	78
Kode 5.14 Halaman Rapor.....	78
Kode 5.15 Halaman Daftar Murid.....	78
Kode 5.16 Fungsi IsExist PenggunaController	81
Kode 5.17 Fungsi GetTipe dan ShowPage PenggunaController	81
Kode 5.18 Fungsi GetRaporKelas RaporController.....	82
Kode 5.19 Fungsi Tambah RaporController	82
Kode 5.20 Fungsi CetakRapor RaporController	83
Kode 5.21 Fungsi IsExist ApiController.....	83
Kode 5.22 Fungsi GetTipe ApiController.....	84
Kode 5.23 Fungsi TambahRapor ApiController	84
Kode 5.24 Fungsi Load ApiController.....	85
Kode 5.25 Fungsi Save ApiController	85
Kode 5.26 Route untuk Web	86
Kode 5.27 Gate Kelas dan Murid.....	86
Kode 5.28 Konfigurasi Auth	87
Kode 5.29 Route untuk API.....	87
Kode 5.30 Global Variable Event Login.....	88
Kode 5.31 Fungsi Masuk	89
Kode 5.32 Fungsi SelectMurid.....	91
Kode 5.33 Fungsi LoadScore	91
Kode 5.34 Tombol Bagian	92

Kode 5.35 Fungsi SetProblem.....	94
Kode 5.36 Angka dan Kotak Jawaban.....	95
Kode 5.37 Fungsi CheckAnswer.....	95
Kode 5.38 Fungsi SaveScore.....	96
Kode 5.39 Fungsi SendRapor.....	96
Kode 5.40 Indikator setiap Bagian.....	97
Kode 5.41 Test Halaman Masuk PenggunaControllerTest ..	100
Kode 5.42 Test Masuk Data Tidak Lengkap PenggunaControllerTest.....	101
Kode 5.43 Test Tipe Pengguna Kelas PenggunaControllerTest	101
Kode 5.44 Test Tambah Rapor RaporControllerTest.....	102
Kode 5.45 Test Semua Murid ApiControllerTest.....	102
Kode 5.46 Test Tambah Rapor ApiControllerTest.....	103
Kode 5.47 Test Load Hasil ApiControllerTest.....	103
Kode 5.48 Test Save Hasil ApiControllerTest	104

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Pada bagian ini akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bagian ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir ini dapat dipahami.

1.1 Latar Belakang Masalah

Rapor dalam standar penilaian Pendidikan Anak Usia Dini (PAUD) adalah pelaporan hasil penilaian berupa deskripsi capaian perkembangan anak disusun secara tertulis yang disampaikan dalam kurun waktu semester untuk ditindaklanjuti dalam kegiatan berikutnya[1].

PAUD Ciluk-Baa adalah PAUD swasta yang berlokasi di Tulungagung[2]. PAUD ini terdaftar dengan nama lembaga PAUD “CILUK-BAA” dengan rumpun pendidikan Kelompok Bermain[3]. Sehingga PAUD ini dapat melayani anak usia 2-4 tahun dan 4-6 tahun[1]. Terdapat layanan kelas Junior dan Kelas Senior serta Readiness Class[4].

PAUD ini menggunakan rapor dalam bentuk buku cetak. Perkembangan anak diamati dan dicatat setiap hari kerja dalam catatan harian lalu dituliskan ke rapor. Selain dibagikan setiap akhir semester, rapor dapat dipinjam oleh wali murid pada hari kerja dengan meminta kepada guru kelas. Kegiatan berkaitan rapor yang dilakukan secara konvensional menimbulkan masalah yaitu: guru melewatkan kesempatan mengamati perkembangan anak, guru lupa menuliskan hasil pengamatan pada rapor, dan wali murid tidak dapat meminjam serta mengembalikan rapor diluar jam kerja.

Masalah pengamatan dan pencatatan perkembangan ini sangat penting. Penelitian Dr. Benyamin S. Bloom menyatakan bahwa

pada usia 0-4 tahun yang merupakan masa usia dini, perkembangan otak anak mencapai 50% dan mencapai 80% saat berusia 8 tahun[5]. Hal ini menunjukkan betapa cepatnya anak berkembang pada rentang usia tersebut. Meskipun sebagian besar anak berkembang dengan lancar, bertahap, dan langkah demi langkah, ada sebagian anak-anak lain yang berkembang dengan kecepatan lebih tinggi atau lebih rendah. Terdapat anak-anak yang mengalami penyimpangan dalam proses perkembangan. Tidak semua anak mampu mencapai tingkat perkembangan yang sama pada usia yang sama[6]. Karena itulah pengamatan dan pencatatan perkembangan masing-masing anak diperlukan.

Mengenai masalah peminjaman rapor yaitu waktu untuk meminjam dan mengembalikan yang kurang fleksibel dan menambah beban guru. Kejadian yang lebih parah adalah wali murid lupa mengembalikan rapor atau rapor yang rusak saat dipinjam sehingga berujung kepada pembuatan ulang rapor.

Rancang bangun sistem rapor adalah pengembangan aplikasi berbasis web untuk melakukan proses bisnis PAUD Ciluk-Baa yang berkaitan dengan rapor. Sistem ini dikembangkan untuk mengatasi masalah mengenai batasan jam dan hari kerja. Guru dapat mengisi rapor kapan saja dan dimana saja dengan mengakses sistem melalui jaringan. Serta wali murid dapat melihat rapor kapan saja.

Namun, masalah pengamatan yang terlewatkan belum teratasi jika sistem rapor hanya diisi manual oleh guru. Diperlukan cara agar pengamatan dan pengisian dilakukan secara otomatis. Salah satu cara yang dapat dilakukan adalah dengan menghubungkan sistem rapor dengan aplikasi permainan Android. Aplikasi permainan yang dikembangkan akan didesain untuk memenuhi indikator perkembangan pada rapor yang akan didetailkan pada batasan masalah. Aplikasi permainan akan mengisi rapor pada setiap akhir permainan. Aplikasi permainan ini akan digunakan pada jam belajar sebagai lembar kerja anak interaktif. Android dipilih sebagai

basis aplikasi permainan karena PAUD Ciluk-Baa memberikan akses ke perangkat Android secara terbatas untuk pembelajaran anak.

Berdasarkan uraian di atas maka tujuan penelitian ini adalah mengembangkan sistem rapor berbasis web yang sesuai dengan proses bisnis PAUD Ciluk-Baa dan dapat menerima masukan manual melalui web maupun otomatis dari aplikasi permainan Android sehingga menghasilkan tugas akhir dengan judul “RANCANG BANGUN SISTEM RAPOR YANG TERHUBUNG DENGAN APLIKASI PERMAINAN ANDROID (STUDI KASUS: PAUD CILUK-BAA)”.

1.2 Rumusan Masalah

Adapun rumusan masalah yang ingin diselesaikan berdasarkan paparan latar belakang di atas adalah sebagai berikut:

1. Bagaimana proses bisnis untuk rapor konvensional diterjemahkan ke dalam sistem rapor?
2. Bagaimana hasil rancang bangun sistem rapor?
3. Bagaimana hasil rancang bangun aplikasi permainan Android?
4. Bagaimana menghubungkan antara sistem rapor berbasis web dengan aplikasi permainan Android?

1.3 Batasan Masalah

Adapun batasan masalah terkait penelitian tugas akhir ini adalah sebagai berikut:

1. Sistem rapor dikembangkan dengan mengacu pada proses bisnis dan kondisi PAUD Ciluk-Baa yang berkaitan dengan rapor.
2. Aplikasi permainan Android akan ditargetkan pada murid PAUD Ciluk-Baa yang berusia 5-6 tahun atau murid Readiness Class.
3. Aplikasi permainan Android mengacu pada poin 1-3 usia 5-6 tahun pada lingkup perkembangan berpikir simbolik di lampiran 1 Permendikbud No. 137 Tahun 2014 yaitu “menyebutkan lambang bilangan 1-10”,

“menggunakan lambang bilangan untuk berhitung”, dan “mencocokkan bilangan dengan lambang bilangan”[7].

4. Aplikasi permainan Android akan digunakan untuk menilai indikator rapor PAUD Ciluk-Baa dengan kode 3.3.2.2., 3.3.2.3., 3.3.4.1., dan 3.3.4.2.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah yang telah disebutkan sebelumnya, adapun tujuan dari penelitian tugas akhir ini adalah mengembangkan sistem rapor berbasis web yang sesuai dengan proses bisnis PAUD Ciluk-Baa dan dapat menerima masukan manual melalui web maupun otomatis dari aplikasi permainan Android.

1.5 Manfaat Penelitian

Adapun manfaat yang diharapkan dari hasil penelitian tugas akhir ini adalah sistem rapor yang dikembangkan akan menjadi dasar untuk penerapan sistem rapor yang sepenuhnya menerima masukan otomatis. Sehingga mengurangi beban administrasi guru PAUD Ciluk-Baa dalam pengisian rapor supaya guru dapat fokus pada tugas utamanya yaitu mengajar.

1.6 Relevansi

Tugas akhir ini disusun untuk memenuhi salah satu syarat kelulusan sebagai Sarjana Komputer di Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas (FTEIC) Institut Teknologi Sepuluh Nopember (ITS). Penelitian tugas akhir ini memiliki relevansi terhadap mata kuliah yang ada pada Departemen Sistem Informasi FTEIC ITS, yaitu Manajemen Proses Bisnis dan Rancang Bangun Perangkat Lunak. Serta penelitian ini masuk ke dalam topik Laboratorium Infrastruktur dan Keamanan Teknologi Informasi (IKTI) yaitu Transfer Data.

BAB II TINJAUAN PUSTAKA

Pada bagian ini akan menjelaskan mengenai studi literatur terhadap penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini.

2.1 Studi Literatur

Dalam proses pengerjaan tugas akhir ini, dilakukan pencarian penelitian-penelitian yang sudah dilakukan untuk dijadikan sebagai referensi dalam pengerjaan penelitian tugas akhir ini. Penelitian-penelitian tersebut dirangkum dalam Tabel 2.1 di bawah:

Tabel 2.1 Studi literatur

1) Pembuatan Sistem Informasi Pendidikan dan Pelatihan dalam Jabatan Berbasis Web pada Bagian Pengembangan Pegawai Direktorat Jenderal Perbendaharaan[8]
Penulis; Tahun R. Gerhaantara dan F. Samopa; 2013
Pembahasan Penelitian ini bertujuan merancang dan mengembangkan sistem berbasis web yang mengambil sumber dari proses bisnis studi kasus penulis. Penelitian ini menggunakan iconix process untuk melakukan perancangan.
Keterkaitan Penelitian tugas akhir yang dilakukan berkaitan dengan merancang dan mengembangkan sistem rapor berbasis web dengan mengacu pada proses bisnis studi kasus penulis. Penelitian ini juga akan memanfaatkan iconix process saat melakukan perancangan.

<p>2) Restful API Architecture Based on Laravel Framework[9]</p>
<p>Penulis; Tahun Xianjun Chen et al; 2017</p>
<p>Pembahasan Penelitian ini bertujuan merancang arsitektur Representational State Transfer RESTful Application Program Interface (API) secara efektif. Rancangan arsitektur berdasarkan pada bahasa PHP: Hypertext Preprocessor (PHP) dan framework Laravel. Penelitian ini menyoroti kelas dan pengaturan pada Laravel yang berkaitan dengan RESTful API. Penelitian ini juga memberikan solusi untuk melakukan Cross-Origin Resource Sharing (CORS) pada Laravel.</p>
<p>Keterkaitan Penelitian tugas akhir yang dilakukan berkaitan dengan merancang dan mengembangkan sistem rapor berbasis web menggunakan framework Laravel. Selain menerima masukan pengguna secara konvensional dari web, sistem rapor akan menerima masukan secara otomatis dari aplikasi permainan Android. Untuk menghubungkan antara sistem rapor dengan aplikasi permainan Android maka akan digunakan RESTful API.</p>
<p>3) Digital Game-based Learning for Early Childhood A State of the Art Report[10]</p>
<p>Penulis; Tahun N. Peirce; 2013</p>
<p>Pembahasan Laporan ini memaparkan tentang bagaimana desain game yang sesuai untuk rentang usia ini dengan mempertimbangkan kemampuan kognitif serta bagaimana</p>

praktek terbaik untuk desain pada perangkat dengan layar sentuh.

Keterkaitan

Penelitian tugas akhir ini akan mengembangkan aplikasi permainan Android untuk pengguna anak usia dini. Praktek terbaik yang dicantumkan pada laporan akan dimanfaatkan untuk mendesain game.

2.2 Dasar Teori

Berikut merupakan dasar-dasar teori yang dijadikan sebagai landasan ilmiah pengembangan tugas akhir ini.

2.2.1 Rapor PAUD Ciluk-Baa

Rapor dalam standar penilaian PAUD adalah pelaporan hasil penilaian berupa deskripsi catatan perkembangan anak secara tertulis yang disampaikan dalam kurun waktu semester untuk ditindaklanjuti dalam kegiatan berikutnya[1].

Rapor PAUD Ciluk-Baa berupa buku cetak dengan tabel berisi kode indikator, indikator kemampuan anak, pencapaian pertama, dan frekuensi pencapaian teramati seperti pada gambar 2.1. Indikator kemampuan anak adalah deskripsi dari kode indikator dan menyatakan apa indikatornya. Pencapaian pertama adalah tanggal dimana guru pertama kali mengamati keberhasilan anak mencapai indikator tersebut. Frekuensi pencapaian teramati akan terisi 1 setiap kali anak teramati memenuhi indikator dan ketercapaian indikator dianggap valid setelah frekuensi mencapai 4.

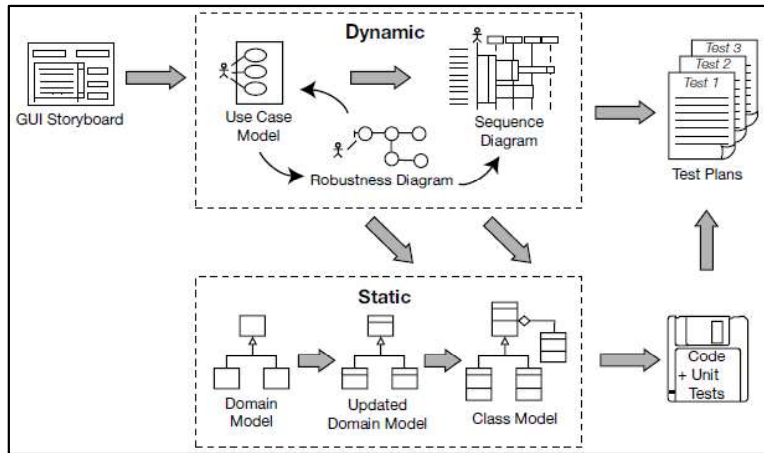
Kode	Indikator Kemampuan Anak	Pencapaian Pertama	Frekuensi Pencapaian Teramat			
3.1.7.3.	Membaca buku cerita bergambar dengan teks sedikit	19-9-16	1	1		
3.3.1.	Membilang					
3.3.1.1.	Menyebutkan bilangan 1, 2					
3.3.1.2.	Menyebutkan bilangan 1-5, kadang tidak urut atau terlompati					
3.3.1.3.	Menyebutkan bilangan 1-5 berurutan					
3.3.1.4.	Menyebutkan bilangan 1-10 berurutan	1-11-16	1	1		
3.3.2.	Menguasai konsep bilangan					
3.3.2.1.	Menyebutkan bilangan 1-5 berurutan sambil menghitung benda					
3.3.2.2.	Menyebutkan bilangan 1-10 berurutan sambil menghitung benda					
3.3.2.3.	Menghubungkan/memasangkan lambang bilangan dengan jumlah benda 1-5 (bukan menuliskan lambang bilangan)	19-8-16	1	1		
3.3.3.	Menguasai konsep himpunan					
3.3.3.1.	Menunjukkan dua kumpulan benda yang memiliki ciri tertentu (sama jumlahnya, tidak sama jumlahnya, lebih banyak, lebih sedikit)	5-4-17	1			
3.3.4.	Menguasai operasi bilangan					
3.3.4.1.	Menghitung dan menyebutkan hasil penambahan (menggabungkan dua kumpulan benda)					
3.3.4.2.	Menghitung dan menyebutkan hasil pengurangan (memisahkan kumpulan benda)					
3.3.4.3.	Menghitung dan menyebutkan hasil pembagian (memisahkan kumpulan benda dengan jumlah yang sama setiap kumpulan)	10-10-16	1	1		
3.3.5.	Mengenal lambang bilangan dan lambang operasi bilangan					
3.3.5.1.	Menghubungkan lambang bilangan dengan sekelompok benda dengan jumlah tertentu sesuai dengan bilangan yang dilambangkan: misal satu jeruk dihubungkan dengan angka 1					
3.3.5.2.	Mengenal lambang operasi bilangan: "+" untuk penambahan, "-" untuk pengurangan, dan "/" untuk pembagian	10-10-16	1			
3.3.5.3.	Menggunakan lambang operasi bilangan untuk berhitung	27-10-16	1	1		
3.4.1.	Membangun menggunakan balok unit					
3.4.1.1.	Bermain balok tanpa membuat bangunan					
3.4.1.2.	Susunan vertical, anak menumpuk balok keatas					

Gambar 2.1 Halaman Rapor yang Telah Terisi

2.2.2 Proses Iconix

Unified Modelling Language (UML) adalah bahasa permodelan pada bidang pengembangan perangkat lunak yang digunakan untuk menentukan standar bagaimana cara menggambarkan desain untuk sebuah sistem[11]. Semua diagram UML secara teori bermanfaat, namun dengan menggunakan proses *iconix*

secara praktis cukup digunakan beberapa diagram yaitu *use case*, *robustness*, *sequence*, dan *class*[12]. Proses *iconix* dapat dilihat pada Gambar 2.2 di bawah:



Gambar 2.2 Proses Iconix

2.2.3 Laravel

Laravel adalah kerangka kerja *open source* untuk pengembangan aplikasi web yang ditulis menggunakan PHP dengan pola arsitektur *Model-View-Controller* (MVC)[9]. Laravel memudahkan pengembangan aplikasi web dengan menyediakan berbagai fitur seperti *template*, *routing*, dan API dalam satu paket.

2.2.4 RESTful API

Representational State Transfer Application Programming Interface secara nonteknis adalah ‘kontrak’ yang menyatakan jika ‘klien’ mengirimkan permintaan dengan format spesifik kepada ‘peladen’ maka akan didapatkan respon dengan format spesifik atau bisa juga melakukan kegiatan yang sudah didefinisikan. REST adalah arsitektur dengan kondisi *stateless* yang berarti semua informasi yang diperlukan untuk memproses permintaan telah tercantum pada permintaan

tersebut[13]. Sehingga RESTful API dapat diartikan sebagai layanan API yang menggunakan arsitektur REST.

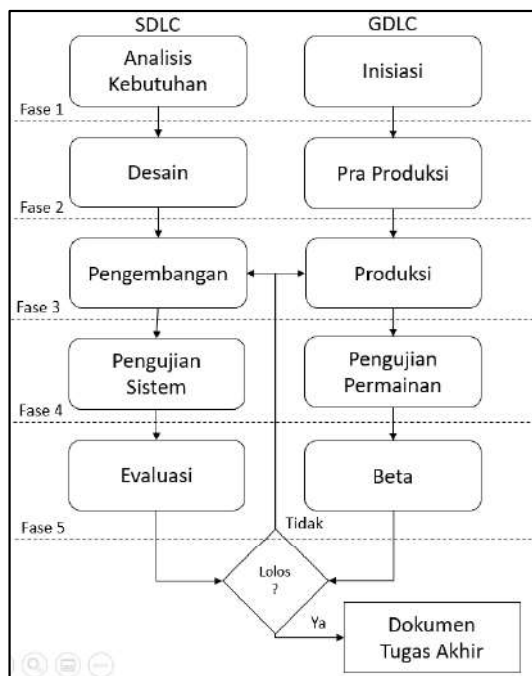
2.2.5 Aplikasi Permainan Android

Aplikasi permainan atau *game* adalah sebuah sistem dimana pemain terlibat dalam sebuah konflik buatan yang didefinisikan oleh peraturan dengan hasil akhir yang dapat terukur[14]. Android adalah sistem operasi *open source* yang dikembangkan oleh Google untuk perangkat bergerak atau *mobile phone*[15]. Aplikasi permainan Android akan dapat dijalankan melalui sistem operasi Android. Aplikasi permainan akan dikembangkan menggunakan *Game Development Life Cycle* (GDLC) yang diusulkan[16].

BAB III METODOLOGI

Pada bagian metodologi akan menjelaskan bagaimana tahap pengerjaan tugas akhir dengan disertakan deskripsi untuk masing-masing tahapan pengerjaan tugas akhir.

Penelitian ini bertujuan menghasilkan dua modul yaitu sistem rapor dan aplikasi permainan Android. Metodologi yang digunakan dalam mengembangkan sistem rapor adalah *System Development Life Cycle (SDLC) Iterative* dan metodologi yang digunakan untuk mengembangkan aplikasi permainan Android adalah *Game Development Life Cycle (GDLC)*. Keduanya akan dilakukan secara berdampingan dengan tahapan yang ditunjukkan pada Gambar 3.1 di bawah.



Gambar 3.1 Tahapan Pelaksanaan Tugas Akhir

3.1 Fase 1

Pada tahapan ini dilakukan penggalian kebutuhan pengguna melalui wawancara dan mengamati dokumen terkait rapor. Kegiatan yang dilakukan untuk masing-masing modul adalah sebagai berikut:

3.1.1 Analisis Kebutuhan

a. Membuat diagram proses bisnis

Diagram proses bisnis dihasilkan dari penggalian kebutuhan. Diagram ini menampilkan terdapat aktifitas apa saja dan siapa saja yang terlibat dalam suatu proses.

b. Membuat daftar kebutuhan fungsional

Kebutuhan fungsional didapatkan dari diagram proses bisnis. Daftar kebutuhan fungsional menentukan apa saja yang dapat dilakukan oleh sistem rapor.

c. Membuat *domain model*

Domain model dibuat berdasarkan kebutuhan fungsional. *Domain model* berisi istilah-istilah yang digunakan dalam pengembangan dan hubungan antara istilah yang ada.

d. Membuat *storyboard*

Storyboard didapatkan dari daftar kebutuhan fungsional. *Storyboard* menceritakan bagaimana pengguna akan berinteraksi dengan sistem rapor dengan alur dasar.

3.1.2 Inisiasi

Membuat konsep permainan

Dengan didasari indikator dari rapor dan didukung oleh dokumen pembelajaran terkait indikator yang dipilih maka akan dibuat konsep permainan berupa kalimat atau yang biasa disebut dengan *high concept statement*.

3.2 Fase 2

Pada tahapan ini dilakukan desain dengan mengacu pada fase sebelumnya, kegiatan yang dilakukan untuk masing-masing modul sebagai berikut:

3.2.1 Desain

a. Membuat use case diagram

Use case diagram dibuat berdasarkan daftar kebutuhan fungsional dan storyboard dari tahapan sebelumnya. Selanjutnya dibuatlah use case description yang berisi alur dasar dan alur alternatif.

b. Membuat desain RESTful API

Desain RESTful API dibuat dengan menggambarkan aplikasi permainan Android sebagai aktor yang berinteraksi dengan sistem dalam use case diagram.

c. Membuat robustness diagram

Robustness diagram dibuat berdasarkan use case diagram. Diagram ini menggambarkan hubungan antara obyek entity, boundary, dan controller.

d. Membuat sequence diagram

Sequence diagram dibuat berdasarkan robustness diagram. Diagram ini menggambarkan bagaimana interaksi antara obyek yang ada pada robustness diagram.

e. Membuat class diagram

Class diagram dibuat berdasarkan domain model dan sequence diagram. Diagram ini menggambarkan hubungan antar kelas serta terdapat atribut dan fungsi apa saja di dalamnya.

3.2.2 Pra Produksi

a. Membuat game design

Game design berfokus dalam menentukan genre, gameplay, mekanisme, cerita, karakter, tantangan, dan aspek teknis yang dituliskan dalam Game Design Document (GDD).

b. Membuat mock-up game

Gambaran awal dari game dibuat berdasarkan detail dalam GDD berupa mock-up.

3.3 Fase 3

Pada tahapan ini kode sumber dituliskan dengan mengacu pada desain di tahapan sebelumnya, kegiatan yang dilakukan untuk masing-masing modul sebagai berikut:

3.3.1 Pengembangan

Menuliskan kode

Kode sumber ditulis berdasarkan *class diagram* dan *sequence diagram*.

3.3.2 Produksi

Membuat aset dan kode sumber

Aset dibuat dan kode sumber dituliskan sehingga menghasilkan *prototype game* yang siap untuk diuji.

3.4 Fase 4

Pada tahapan ini modul yang telah dikembangkan akan diuji fungsionalitasnya oleh pengembang, kegiatan yang dilakukan untuk masing-masing modul sebagai berikut:

3.4.1 Pengujian Sistem

a. Unit test

Unit test dilakukan dengan menguji fungsi yang ada pada class diagram. Menguji apakah setiap fungsi yang dipanggil secara individu dapat menerima input dan menghasilkan output yang sesuai.

b. System test

System test dilakukan dengan menguji apakah sistem sudah mampu memenuhi test yang bersumber dari use case.

3.4.2 Pengujian Permainan

Functional test

Functional test dilakukan dengan cara menguji kualitas fungsional melalui pengujian permainan dengan tujuan menemukan bug dari fungsi atau fitur yang ada pada permainan.

3.5 Fase 5

Pada tahapan ini modul yang telah lolos pengujian pada fase 4 akan diuji kembali. Namun, pengujian ini dilakukan oleh pihak pengguna yaitu PAUD Ciluk-Baa. Kegiatan yang dilakukan untuk masing-masing modul sebagai berikut:

3.5.1 Evaluasi

Acceptance test

Pihak pengguna kecuali murid akan diberikan akses ke sistem lalu diberikan beberapa skenario tes berdasarkan proses bisnis. Kemudian akan dipilih pengguna yang terdiri dari 3 guru, 3 wali murid, dan 2 pengelola untuk melakukan penilaian melalui kuisioner yang berisi tingkat pemahaman tiap skenario dengan skala likert.

3.5.2 Beta

Closed beta

Pemain yang mencoba permainan adalah dari pihak pengguna yaitu murid yang didampingi wali murid dan atau guru. Pengujian ini berfokus untuk mendapatkan masukan mengenai permainan. Kemudian pendamping akan melakukan penilaian melalui kuisisioner dengan skala likert.

Selanjutnya adalah menghitung hasil penilaian kuisisioner. jika belum didapatkan nilai diatas 80% maka tahapan akan dikembalikan ke fase 3. Jika sudah maka pengembangan dianggap selesai lalu kedua modul diserahkan dan penelitian dilanjutkan ke tahap pembuatan dokumen.

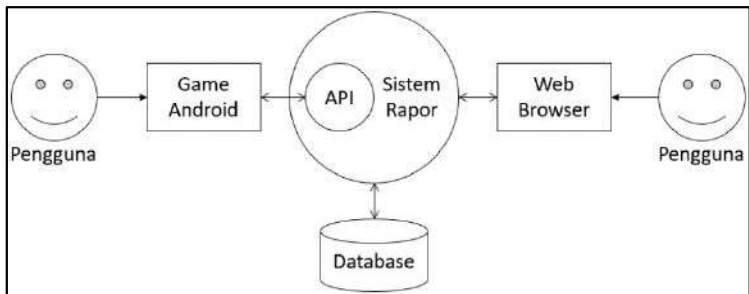
3.6 Dokumen Tugas Akhir

Pada tahapan ini hasil dari kegiatan penelitian dituliskan dalam dokumen tugas akhir. Kemudian menuliskan kesimpulan dan saran untuk penelitian selanjutnya.

BAB IV PERANCANGAN

Pada bagian ini akan dijelaskan mengenai proses perancangan terhadap sistem rapor dan aplikasi permainan yang akan digunakan sebagai acuan dalam proses implementasi. Bagian ini melakukan fase 1 dan 2 yang dijelaskan pada metodologi.

Sistem rapor akan terhubung dengan aplikasi permainan dengan menggunakan API. Desain sistem secara keseluruhan adalah sebagai berikut.



Gambar 4.1 Desain Sistem

Pengguna akan berinteraksi menggunakan browser untuk menggunakan sistem rapor. Pengguna juga dapat berinteraksi dengan aplikasi permainan yang terhubung dengan API pada sistem rapor.

4.1 Fase 1

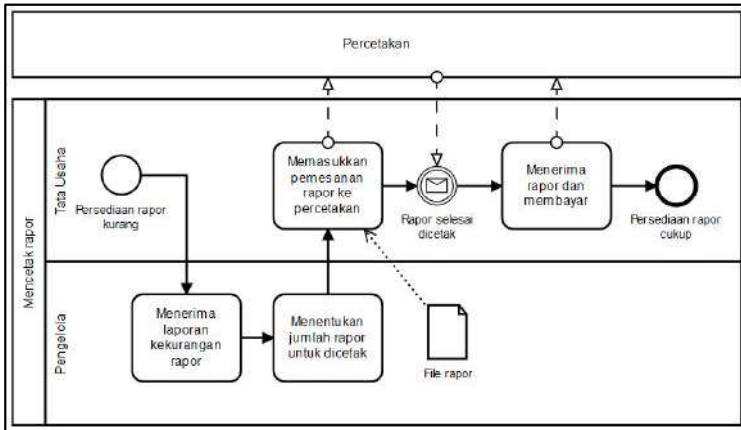
Pada tahapan ini dilakukan penggalan kebutuhan pengguna melalui wawancara dan mengamati dokumen terkait rapor.

4.1.1 Analisis Kebutuhan (Sistem Rapor)

Analisis kebutuhan dilakukan untuk mengetahui apa saja yang dibutuhkan pada sistem rapor.

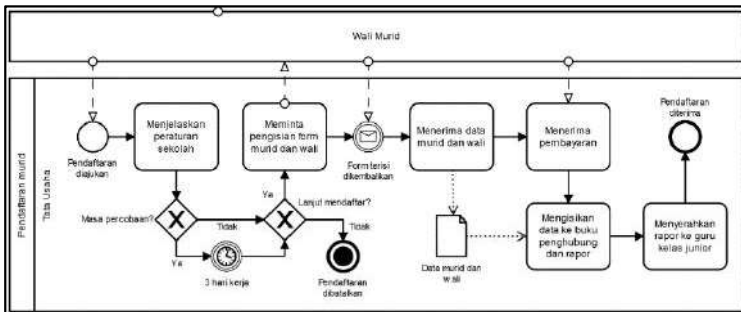
4.1.1.1 Diagram Proses Bisnis

Setelah wawancara dengan pihak PAUD Ciluk-Baa dan mengamati dokumen terkait, didapatkan hasil berupa proses bisnis saat ini yang terkait dengan rapor yang dapat digambarkan dalam diagram proses bisnis sebagai berikut.



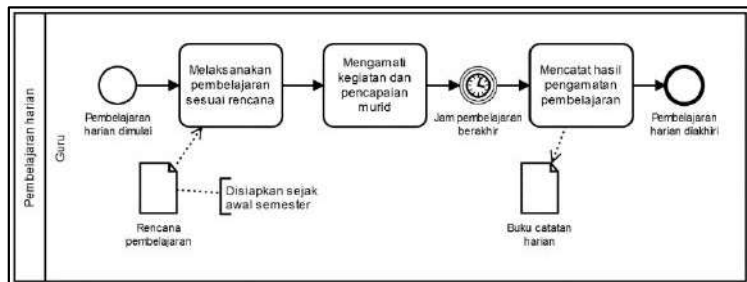
Gambar 4.2 Diagram Proses Mencetak Rapor

- Mencetak rapor.** Pegawai Saat persediaan rapor dirasa kurang (sekitar 10 kopi), pegawai Tata Usaha (TU) akan memberi laporan kekurangan rapor ke pengelola. Pengelola akan menentukan jumlah rapor untuk dicetak (biasanya sekitar 50 kopi). TU lalu membawa rapor dalam bentuk file ke percetakan dan mencetak sesuai kebutuhan.



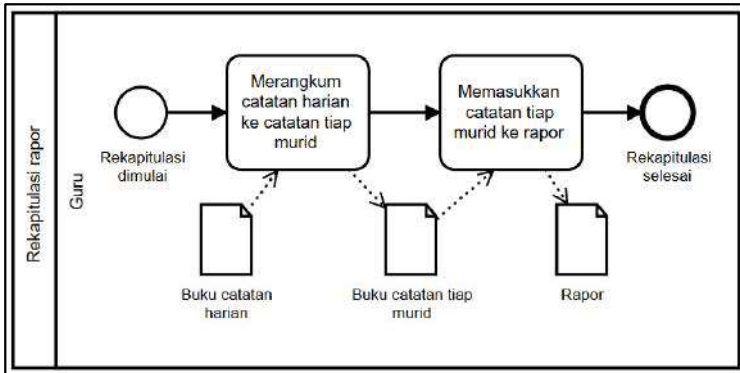
Gambar 4.3 Diagram Proses Pendaftaran Murid

- **Pendaftaran murid.** Wali murid datang dan mengajukan pendaftaran ke TU. TU akan menjelaskan peraturan sekolah dan menawarkan masa percobaan selama 3 hari. Jika wali murid menolak masa percobaan maka TU akan meminta pengisian form pendaftaran ke wali murid. Jika wali murid menerima masa percobaan maka form pendaftaran akan diberikan setelah masa percobaan jika wali murid ingin melanjutkan. Setelah TU menerima data wali murid dan murid beserta pembayarannya maka TU akan mengisikan data tersebut ke buku penghubung dan buku rapor. Murid dan buku rapor akan dialokasikan ke kelas junior.



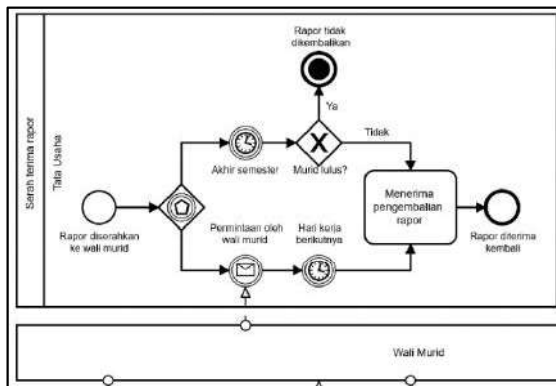
Gambar 4.4 Diagram Proses Pembelajaran Harian

- **Pembelajaran harian.** Saat pembelajaran harian, guru akan melaksanakan pembelajaran sesuai rencana yang sudah disiapkan sejak awal semester dengan cara mengarahkan aktifitas harian murid dan setelah pembelajaran. Saat pembelajaran, selain mengarahkan, guru juga mengamati pencapaian muridnya. Setelah pembelajaran harian usai maka guru akan mencatat pengamatan murid hari itu pada buku catatan harian. Catatan berisi apa aktifitas hari itu dan siapa saja murid yang berhasil melakukannya.



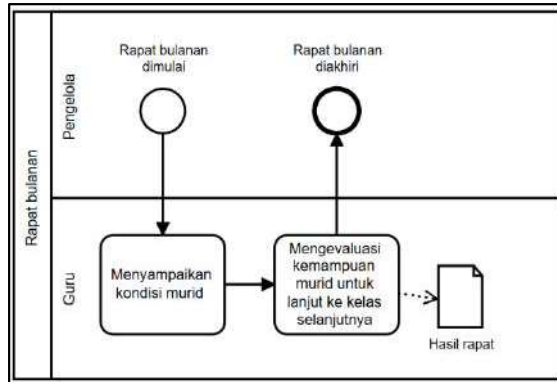
Gambar 4.5 Diagram Proses Rekapitulasi Rapor

- **Rekapitulasi rapor.** Guru merangkum catatan harian ke catatan setiap murid lalu memasukkan catatan setiap murid ke rapor.



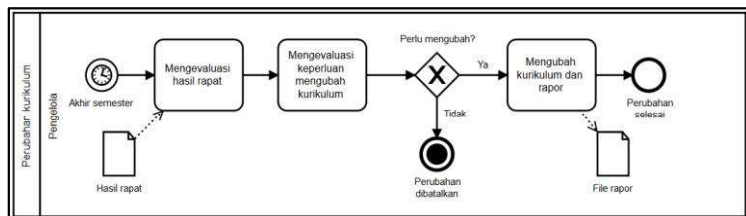
Gambar 4.6 Diagram Proses Serah Terima Rapor

- **Serah terima rapor.** Normalnya rapor diserahkan pada setiap akhir semester dan diterima kembali pada pertemuan pertama semester berikutnya jika murid belum lulus. Namun, wali murid dapat meminjam rapor kapanpun pada hari kerja dan diterima kembali oleh TU pada hari kerja berikutnya.



Gambar 4.7 Diagram Proses Rapat Bulanan

- **Rapat bulanan.** Pengelola berkumpul bersama guru dan membahas mengenai kondisi masing-masing murid yang diajar. Kondisi ini menjadi evaluasi kemampuan murid untuk menentukan lanjut ke kelas.



Gambar 4.8 Diagram Proses Perubahan Kurikulum

- **Perubahan kurikulum.** Pada akhir semester, setelah mengevaluasi hasil rapat maka akan dievaluasi keperluan mengubah **indikator** kurikulum. Jika ditemukan perlu ada perubahan pada kurikulum maka pengelola akan mengubah indikator dan memperbarui file rapor.

PAUD Ciluk-Baa memiliki 4 jenjang kelas yaitu Junior (usia 2-3 tahun), Senior (usia 3-4 tahun), RC-A (usia 4-5 tahun), dan RC-B (usia 5-6 tahun). Serta memiliki 3 guru yang dapat mengajar semua kelas.

4.1.1.2 Daftar Kebutuhan Fungsional

Berdasarkan diagram proses bisnis, maka didapatkan daftar kebutuhan fungsional sebagai berikut:

Tabel 4.1 Kebutuhan Fungsional Sistem Rapor

Kode	Pengguna	Kebutuhan	Deskripsi
FR01	Tata Usaha	Mengelola data murid	Mengisi, ubah, atau menghapus data murid
FR02	Murid, Kelas	Melihat rapor	Melihat data rapor murid
FR03	Murid, Kelas	Mencetak rapor	Mencetak rapor murid
FR04	Kelas	Mengelola rapor	Mengisi atau mengubah data rapor
FR05	Kelas	Mengelola kelas	Menaikkan atau mengeluarkan murid
FR06	Pengelola	Mengelola indikator	Menambah, hapus, atau mengubah indikator
FR07	Admin	Mengubah password	Mengubah password pengguna
FR08	Game	Mengisi rapor	Mengisi data rapor melalui game
FR09	Game	Mengelola hasil bermain	Menyimpan dan mengirimkan data hasil bermain

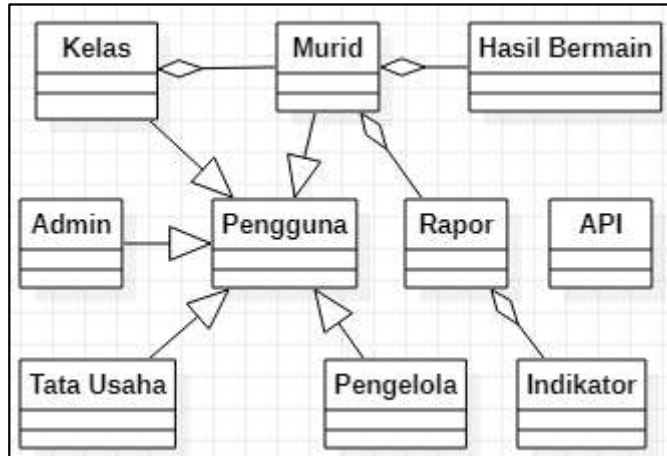
Pengguna Guru berubah menjadi Kelas karena guru dapat mengelola setiap kelas.

Pengguna Admin adalah peran yang ditambahkan untuk sistem rapor. Kebutuhan mengubah password hanya berada di tipe pengguna admin karena tipe pengguna Tata Usaha, Kelas, dan Pengelola adalah pengguna berbagi dan bukan perorangan.

Pengguna Game bukanlah tipe pengguna namun perwakilan dari aplikasi permainan Android.

4.1.1.3 Domain Model

Berdasarkan kebutuhan fungsional didapatkan *domain model* sebagai berikut:

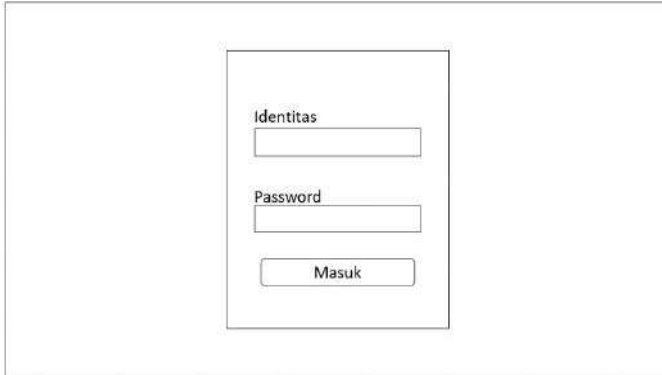


Gambar 4.9 Domain Model

Pengguna adalah orang yang menggunakan sistem dan terdiri dari 5 tipe: Kelas, Murid, Admin, Tata Usaha, dan Pengelola. Setiap jenjang kelas memiliki murid. Setiap murid memiliki rapor yang tersusun dari indikator. Setiap murid memiliki laporan semester yang diisi pada setiap semester. Setiap murid memiliki hasil bermain dari aplikasi permainan Android. API adalah obyek yang akan digunakan untuk berkomunikasi dengan aplikasi permainan Android.

4.1.1.4 Storyboard

Berdasarkan kebutuhan fungsional dapat digambarkan *storyboard* sebagai berikut:



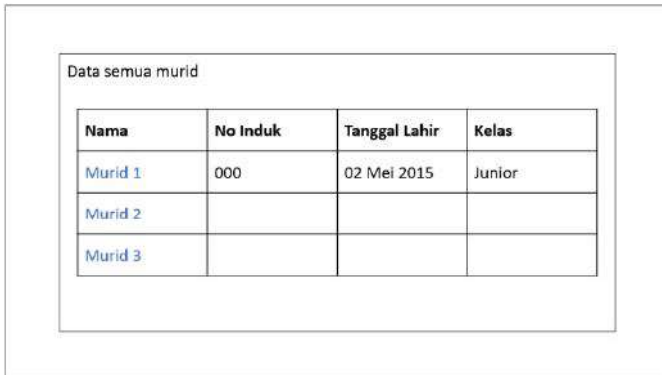
The storyboard shows a login form with the following elements:

- A label "Identitas" above a text input field.
- A label "Password" above a text input field.
- A button labeled "Masuk" below the password field.

Gambar 4.10 Storyboard Halaman Masuk

Halaman pertama yang akan dilihat seluruh pengguna saat mengakses sistem rapor. Pengguna akan memasukkan identitas dan password pada kolom yang sesuai lalu menekan tombol Masuk. Selanjutnya, pengguna akan diarahkan sesuai tipenya.

Pengguna Tata Usaha

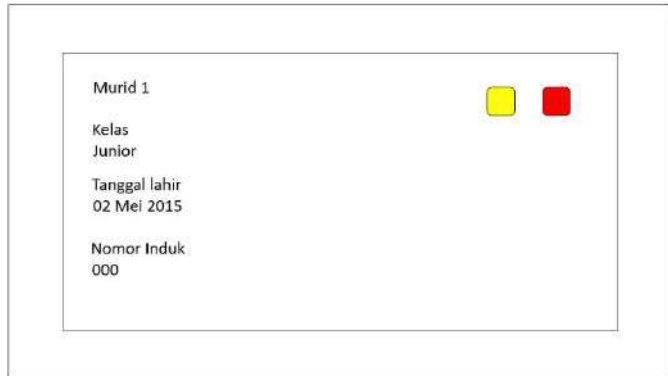


The storyboard shows a table titled "Data semua murid" with the following data:

Nama	No Induk	Tanggal Lahir	Kelas
Murid 1	000	02 Mei 2015	Junior
Murid 2			
Murid 3			

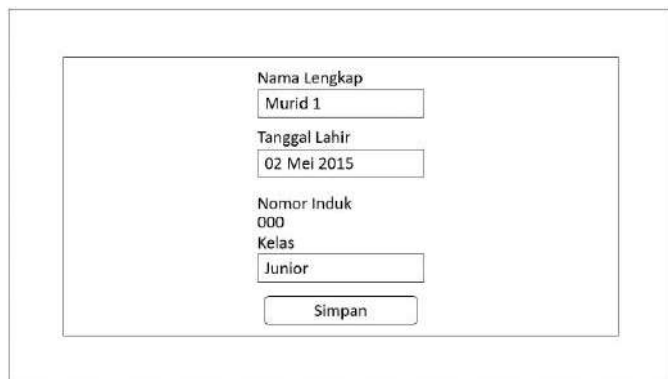
Gambar 4.11 Storyboard Halaman Daftar Murid

Halaman berisi daftar seluruh murid. Terdapat 2 tambahan tipe kelas selain 4 kelas utama, yaitu Keluar dan Lulus. Pengguna dapat menekan nama murid untuk melihat profil murid.



Gambar 4.12 Storyboard Halaman Profil Murid

Halaman berisi data murid dan tombol untuk mengubah (warna kuning) dan menghapus (warna merah). Jika tombol hapus ditekan maka data murid akan dihapus. Jika tombol ubah ditekan maka halaman ubah profil murid akan ditampilkan.



Gambar 4.13 Storyboard Halaman Ubah Profil Murid

Halaman berisi kolom data murid untuk diubah, kecuali nomor induk. Pengguna menekan tombol Simpan untuk menyimpan.

Storyboard for the registration page. It contains three text input fields labeled "Nama Lengkap", "Tanggal Lahir", and "Nomor Induk". Below these fields is a button labeled "Daftar".

Gambar 4.14 Storyboard Halaman Mendaftar

Halaman untuk menambahkan murid. Pengguna mengisi data sesuai kolom yang disediakan lalu menekan tombol Daftar.

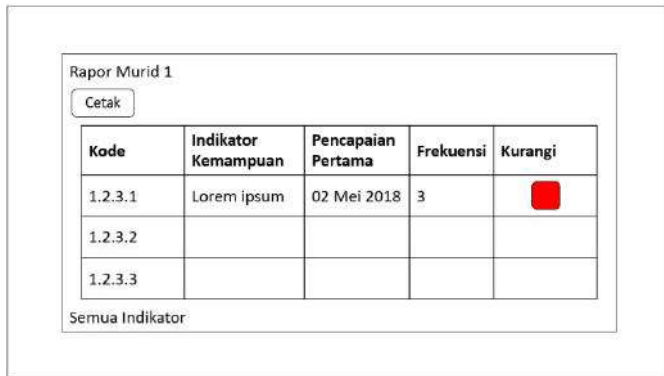
Pengguna Kelas

Storyboard for the class student list page. It features a table titled "Data Murid Kelas Junior". The table has five columns: "Nama", "No Induk", "Rapor", "Naik", and "Keluar". The first row contains "Murid 1", "000", a yellow square, a green square, and a red square. The second and third rows are labeled "Murid 2" and "Murid 3" respectively and are currently empty.

Nama	No Induk	Rapor	Naik	Keluar
Murid 1	000			
Murid 2				
Murid 3				

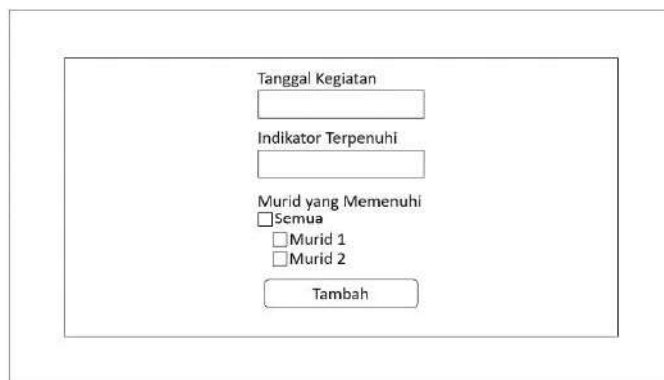
Gambar 4.15 Storyboard Halaman Daftar Murid Kelas

Halaman berisi daftar murid kelas junior. Untuk melihat kelas lain pengguna dapat memilih menu Daftar Murid → [Nama Kelas]. Tabel berisi nama, nomor induk, tombol rapor, tombol naik, dan tombol keluar.



Gambar 4.16 Storyboard Halaman Rapor

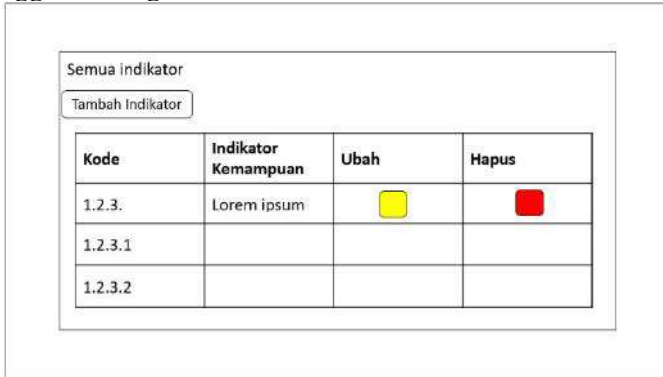
Halaman rapor diakses melalui tombol rapor. Halaman berisi daftar indikator yang telah dipenuhi murid dan daftar seluruh indikator. Jika tombol kurangi ditekan maka frekuensi akan berkurang 1 dan jika mencapai 0 maka baris indikator akan hilang. Jika tombol Cetak ditekan maka akan muncul permintaan mengunduh file pdf berisi tabel daftar indikator yang telah dipenuhi. Halaman ini juga merupakan halaman pertama yang dilihat pengguna Murid setelah masuk, tanpa ada tombol kurangi. Pengguna Murid juga dapat mengakses halaman ini melalui menu Lihat Rapor.



Gambar 4.17 Storyboard Halaman Isi Rapor

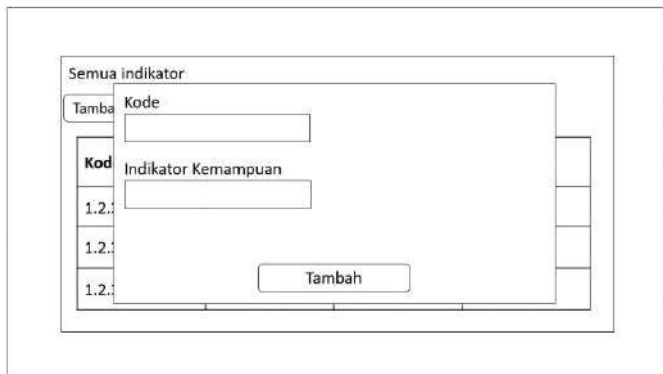
Halaman untuk mengisi rapor berdasarkan kelas, dapat diakses melalui menu Isi rapor → [Nama Kelas]. Pengguna memilih tanggal kegiatan lalu indikator apa saja yang terpenuhi dan siapa saja murid yang memenuhi lalu menekan tombol Tambah.

Pengguna Pengelola



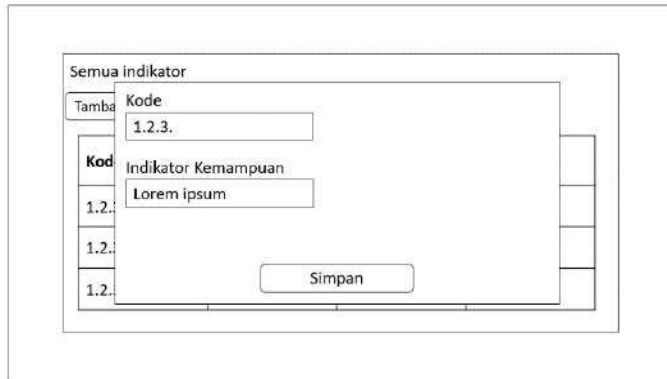
Gambar 4.18 Storyboard Halaman Daftar Indikator

Halaman berisi daftar seluruh indikator. Terdapat tombol Tambah Indikator, Ubah, dan Hapus. Jika tombol Tambah Indikator ditekan maka akan muncul form tambah indikator. Jika tombol Ubah ditekan maka akan muncul form ubah indikator. Jika tombol Hapus ditekan maka indikator tersebut akan dihapus.



Gambar 4.19 Storyboard Modal Tambah Indikator

Form untuk menambah indikator. Pengguna mengisikan kode dan indikator kemampuan lalu menekan tombol Tambah.




The storyboard shows a modal window titled "Semua indikator". On the left, there is a sidebar with a "Tamba" button and a "Kod" section containing a list of "1.2." items. The main area of the modal contains a form with two input fields: "Kode" with the value "1.2.3." and "Indikator Kemampuan" with the value "Lorem ipsum". A "Simpan" button is located at the bottom right of the form area.

Gambar 4.20 Storyboard Modal Ubah Indikator

Form untuk mengubah indikator. Pengguna mengubah deskripsi indikator lalu menekan tombol Simpan.

Pengguna Admin



The storyboard shows a form for changing a user's password. It contains two input fields: "Pegguna" and "Password baru". Below the input fields is a button labeled "Ubah password".

Gambar 4.21 Storyboard Halaman Ubah Password

Halaman untuk mengubah password pengguna di sistem. Admin memilih pengguna yang akan diubah passwordnya lalu memasukkan password baru pada kolom di bawahnya. Selanjutnya menekan tombol Ubah Password.

4.1.2 Inisiasi (Aplikasi Permainan)

Inisiasi untuk memulai pengembangan aplikasi permainan.

4.1.2.1 Konsep permainan

Berdasarkan batasan masalah dan wawancara dengan pihak PAUD Ciluk-Baa didapatkan judul permainan “Toko Bibi” dengan konsep:

“Toko Bibi adalah game yang dikembangkan untuk mendukung kebutuhan pengisian sistem rapor untuk sebagian indikator yang terkait lambang bilangan dan operasinya. Mekanisme utama yang digunakan adalah drag & drop untuk memberikan label angka sesuai jumlah buah. Setelah pemain menyelesaikan satu level maka game akan melakukan penilaian dan meneruskan hasilnya kepada sistem rapor.”.

4.2 Fase 2

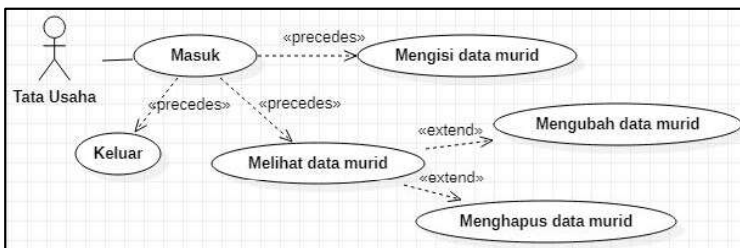
Pada tahapan ini dilakukan desain dengan mengacu pada fase sebelumnya.

4.2.1 Desain (Sistem Rapor)

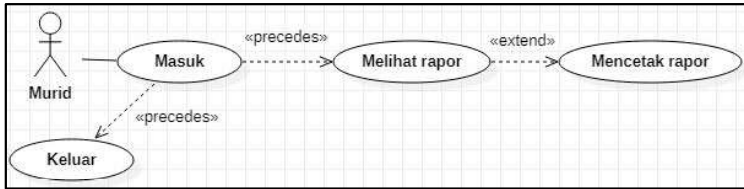
Desain dilakukan berdasarkan hasil analisis kebutuhan.

4.2.1.1 Use Case Diagram

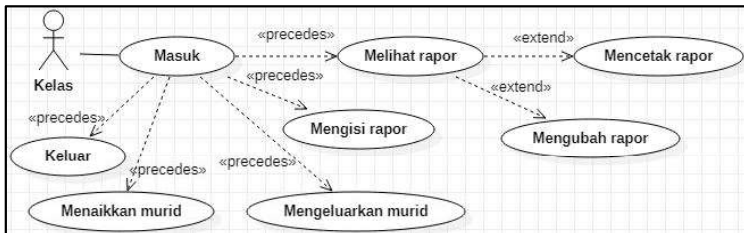
Berdasarkan daftar kebutuhan fungsional dan storyboard dari tahapan sebelumnya dapat dituliskan use case diagram berikut.



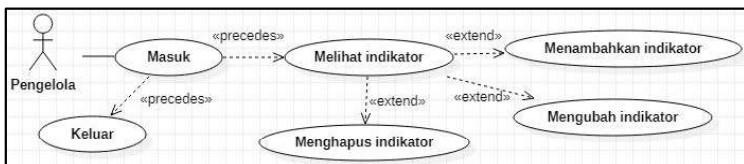
Gambar 4.22 Use Case Diagram Pengguna Tata Usaha



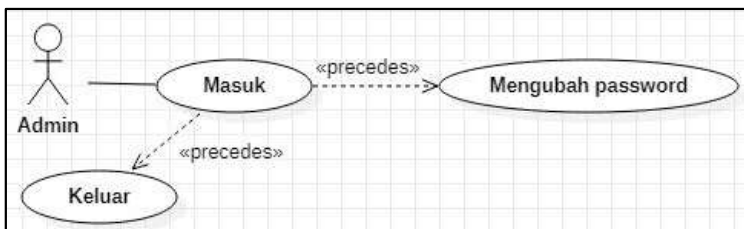
Gambar 4.23 Use Case Diagram Pengguna Murid



Gambar 4.24 Use Case Diagram Pengguna Kelas



Gambar 4.25 Use Case Diagram Pengguna Pengelola



Gambar 4.26 Use Case Diagram Pengguna Admin

Dari diagram di atas dapat dituliskan use case description sebagai berikut.

Tabel 4.2 Use Case Description Masuk

Kode	UC1
Use Case	Masuk
Actor	Tata Usaha, Murid, Kelas, Pengelola, Admin
Pre-Condition	Data pengguna sudah ada.
Basic Course	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman masuk. 2. Pengguna memasukkan id dan password lalu menekan tombol masuk. 3. Sistem mencari menggunakan id lalu mencocokkan dengan password. 4. Sistem menampilkan halaman berdasarkan tipe pengguna: Tata Usaha → Halaman daftar murid. Murid → Halaman rapor. Kelas → Halaman daftar murid kelas. Pengelola → Halaman daftar indikator. Admin → Halaman ubah password.
Alternate Course	<ol style="list-style-type: none"> 2.1. Pengguna menekan tombol masuk tanpa mengisi id dan atau password. 2.2. Sistem menampilkan pesan "Mohon lengkapi data". 3.1. Sistem tidak menemukan id pengguna atau password yang dimasukkan tidak sesuai. 3.2. Sistem menampilkan pesan "ID atau password salah".

Tabel 4.3 Use Case Description Keluar

Kode	UC2
Use Case	Keluar
Actor	Tata Usaha, Murid, Kelas, Pengelola, Admin
Pre-Condition	Aktor telah masuk.
Basic Course	<ol style="list-style-type: none"> 1. Pengguna menekan tombol keluar. 2. Sistem menampilkan halaman masuk.

Alternate Course	-
-------------------------	---

Tabel 4.4 Use Case Description Melihat Data Murid

Kode	UC3
Use Case	Melihat data murid
Actor	Tata Usaha
Pre-Condition	Aktor telah masuk. Data murid sudah ada.
Basic Course	1. Sistem menampilkan halaman daftar murid. 2. Pengguna memilih salah satu murid. 3. Sistem menampilkan halaman profil murid.
Alternate Course	-

Tabel 4.5 Use Case Description Mengisi Data Murid

Kode	UC4
Use Case	Mengisi data murid
Actor	Tata Usaha
Pre-Condition	Aktor telah masuk.
Basic Course	1. Pengguna memilih menu mendaftar. 2. Sistem menampilkan halaman mendaftar. 3. Pengguna memasukkan nama, tanggal lahir, dan nomor induk murid lalu menekan tombol daftar. 4. Sistem mengecek kesamaan nomor induk dengan data murid yang telah tersimpan. 5. Sistem menampilkan nama, tanggal lahir, dan nomor induk yang telah dimasukkan pengguna sebagai konfirmasi mendaftar. 6. Pengguna menekan tombol konfirmasi. 7. Sistem menyimpan data murid dalam database. 8. Sistem menampilkan pesan "Berhasil mendaftarkan". 9. Sistem menampilkan halaman mendaftar

Alternate Course	<p>3.1. Pengguna menekan tombol daftar tanpa melengkapi data.</p> <p>3.2. Sistem menampilkan pesan "Mohon lengkapi data".</p> <p>4.1. Sistem menemukan nomor induk yang sama pada database.</p> <p>4.2. Sistem menampilkan konfirmasi nomor induk.</p> <p>4.3. Pengguna menekan tombol Kembali.</p> <p>4.4. Sistem menutup konfirmasi nomor induk.</p> <p>4.3.1. Pengguna menekan tombol Lihat Murid.</p> <p>4.3.2. Sistem menampilkan halaman profil murid dengan nomor induk tersebut.</p> <p>6.1. Pengguna menekan tombol batal</p> <p>6.2. Sistem menutup konfirmasi mendaftar.</p>
-------------------------	---

Tabel 4.6 Use Case Description Mengubah Data Murid

Kode	UC5
Use Case	Mengubah data murid
Actor	Tata Usaha
Pre-Condition	<p>Aktor telah masuk.</p> <p>Aktor berada pada halaman profil murid.</p> <p>Data murid sudah ada.</p>
Basic Course	<p>1. Pengguna menekan tombol ubah.</p> <p>2. Sistem menampilkan halaman ubah profil.</p> <p>3. Pengguna menekan tombol simpan.</p> <p>4. Sistem menyimpan perubahan.</p> <p>5. Sistem menampilkan pesan "Berhasil mengubah".</p> <p>6. Sistem menampilkan halaman profil murid.</p>
Alternate Course	<p>3.1.1. Pengguna menekan tombol simpan tanpa melengkapi data.</p> <p>3.1.2. Sistem menampilkan pesan "Mohon lengkapi data".</p> <p>3.2.1. Pengguna menekan tombol kembali.</p> <p>3.2.2. Sistem menampilkan halaman profil murid.</p>

Tabel 4.7 Use Case Description Menghapus Data Murid

Kode	UC6
Use Case	Menghapus data murid
Actor	Tata Usaha
Pre-Condition	Aktor telah masuk. Aktor berada pada halaman profil murid. Data murid sudah ada.
Basic Course	<ol style="list-style-type: none"> 1. Pengguna menekan tombol hapus. 2. Sistem menampilkan konfirmasi menghapus. 3. Pengguna menekan tombol konfirmasi. 4. Sistem menghapus data murid tersebut dari database. 5. Sistem menutup konfirmasi menghapus. 6. Sistem menampilkan pesan "Berhasil menghapus". 7. Sistem menampilkan halaman daftar murid.
Alternate Course	<ol style="list-style-type: none"> 5.1. Pengguna menekan tombol batal. 5.2. Sistem menutup konfirmasi menghapus.

Tabel 4.8 Use Case Description Melihat Rapor

Kode	UC7
Use Case	Melihat rapor
Actor	Kelas, Murid
Pre-Condition	Aktor telah masuk. Data murid sudah ada.
Basic Course	<ol style="list-style-type: none"> 1. Pengguna merupakan tipe Kelas. 2. Sistem menampilkan halaman daftar murid kelas. 3. Pengguna menekan tombol rapor salah satu murid. 4. Sistem menampilkan halaman rapor.
Alternate Course	<ol style="list-style-type: none"> 1.1. Pengguna merupakan tipe Murid. 1.2. Sistem menampilkan halaman rapor.

Tabel 4.9 Use Case Description Mencetak Rapor

Kode	UC8
Use Case	Mencetak rapor
Actor	Kelas, Murid
Pre-Condition	Aktor telah masuk. Aktor berada pada halaman rapor. Data murid sudah ada.
Basic Course	1. Pengguna menekan tombol cetak. 2. Sistem mengubah halaman rapor cetak ke pdf. 3. Sistem meminta pengguna mengunduh pdf tersebut.
Alternate Course	-

Tabel 4.10 Use Case Description Mengisi Rapor

Kode	UC9
Use Case	Mengisi rapor
Actor	Kelas
Pre-Condition	Aktor telah masuk. Data murid sudah ada. Data indikator sudah ada.
Basic Course	1. Pengguna memilih menu isi rapor. 2. Sistem menampilkan halaman isi rapor. 3. Pengguna memilih tanggal. 4. Pengguna memilih indikator. 5. Pengguna memilih murid lalu menekan tombol tambah. 6. Sistem menampilkan konfirmasi isi rapor. 7. Pengguna menekan tombol konfirmasi. 8. Sistem menyimpan data dalam database. 9. Sistem menutup konfirmasi isi rapor. 10. Sistem menampilkan pesan "Rapor berhasil ditambahkan".
Alternate Course	5.1. Pengguna menekan tombol simpan tanpa melengkapi data.

	<p>5.2. Sistem menampilkan pesan "Mohon lengkapi data".</p> <p>7.1. Pengguna menekan tombol batal.</p> <p>7.2. Sistem menutup konfirmasi isi rapor.</p>
--	---

Tabel 4.11 Use Case Description Mengubah Rapor

Kode	UC10
Use Case	Mengubah rapor
Actor	Kelas
Pre-Condition	<p>Aktor telah masuk.</p> <p>Aktor berada pada halaman rapor.</p> <p>Data murid sudah ada.</p> <p>Data rapor sudah ada.</p>
Basic Course	<ol style="list-style-type: none"> 1. Pengguna menekan tombol kurangi pada indikator yang ingin dikurangi. 2. Sistem mengurangi frekuensi indikator tersebut sebanyak 1 dari rapor.
Alternate Course	-

Tabel 4.12 Use Case Description Menaikkan Murid

Kode	UC11
Use Case	Menaikkan murid
Actor	Kelas
Pre-Condition	<p>Aktor telah masuk.</p> <p>Aktor berada pada halaman daftar murid kelas.</p> <p>Data murid sudah ada.</p> <p>Data laporan semester sudah ada.</p>
Basic Course	<ol style="list-style-type: none"> 1. Pengguna menekan tombol naikan salah satu murid. 2. Sistem menampilkan konfirmasi kenaikan. 3. Pengguna menekan tombol konfirmasi. 4. Sistem menyimpan data dalam database. 5. Sistem menutup konfirmasi kenaikan. 6. Sistem menampilkan pesan "Berhasil menaikkan".

Alternate Course	3.1. Pengguna menekan tombol batal. 3.2. Sistem menutup konfirmasi kenaikan.
-------------------------	---

Tabel 4.13 Use Case Description Mengeluarkan Murid

Kode	UC12
Use Case	Mengeluarkan murid
Actor	Kelas
Pre-Condition	Aktor telah masuk. Aktor berada pada halaman daftar murid kelas. Data murid sudah ada.
Basic Course	1. Pengguna menekan tombol keluarkan salah satu murid. 2. Sistem menampilkan konfirmasi keluarkan. 3. Pengguna menekan tombol konfirmasi. 4. Sistem menyimpan data dalam database. 5. Sistem menutup konfirmasi keluarkan. 6. Sistem menampilkan pesan "Murid keluar".
Alternate Course	3.1. Pengguna menekan tombol batal. 3.2. Sistem menutup konfirmasi keluarkan.

Tabel 4.14 Use Case Description Melihat Indikator

Kode	UC13
Use Case	Melihat indikator
Actor	Pengelola
Pre-Condition	Aktor telah masuk. Data indikator sudah ada.
Basic Course	1. Sistem menampilkan halaman daftar indikator.
Alternate Course	-

Tabel 4.15 Use Case Description Menambahkan Indikator

Kode	UC14
Use Case	Menambahkan indikator
Actor	Pengelola
Pre-Condition	Aktor telah masuk. Aktor berada pada halaman daftar indikator.

Basic Course	<ol style="list-style-type: none"> 1. Pengguna menekan tombol tambah indikator. 2. Sistem menampilkan form tambah indikator. 3. Pengguna mengisi form indikator lalu menekan tombol tambah. 4. Sistem menyimpan data dalam database. 5. Sistem menutup form tambah indikator.
Alternate Course	<ol style="list-style-type: none"> 3.1.1. Pengguna menekan tombol simpan tanpa melengkapi data. 3.1.2. Sistem menampilkan pesan melengkapi. 3.2.1. Pengguna menekan tombol batal. 3.2.2. Sistem menutup form tambah indikator.

Tabel 4.16 Use Case Description Mengubah Indikator

Kode	UC15
Use Case	Mengubah indikator
Actor	Pengelola
Pre-Condition	<p>Aktor telah masuk. Aktor berada pada halaman daftar indikator. Data indikator sudah ada.</p>
Basic Course	<ol style="list-style-type: none"> 1. Pengguna menekan tombol ubah salah satu indikator. 2. Sistem menampilkan form ubah indikator. 3. Pengguna mengubah data indikator lalu menekan tombol simpan. 4. Sistem menyimpan data dalam database. 5. Sistem menutup form ubah indikator.
Alternate Course	<ol style="list-style-type: none"> 3.1.1. Pengguna menekan tombol simpan tanpa melengkapi data. 3.1.2. Sistem menampilkan pesan melengkapi. 3.2.1. Pengguna menekan tombol batal. 3.2.2. Sistem menutup form ubah indikator.

Tabel 4.17 Use Case Description Menghapus Indikator

Kode	UC16
Use Case	Menghapus indikator
Actor	Pengelola

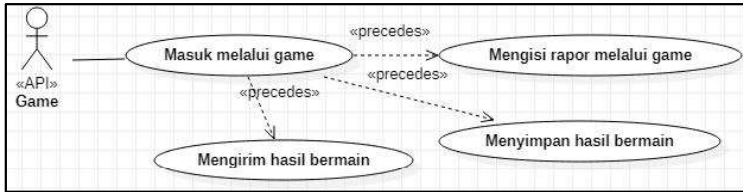
Pre-Condition	Aktor telah masuk. Aktor berada pada halaman daftar indikator. Data indikator sudah ada.
Basic Course	1. Pengguna menekan tombol hapus salah satu indikator. 2. Sistem menampilkan konfirmasi hapus. 3. Pengguna menekan tombol konfirmasi. 4. Sistem menghapus data dari database.
Alternate Course	3.1. Pengguna menekan tombol batal. 3.2. Sistem menutup konfirmasi hapus.

Tabel 4.18 Use Case Description Mengubah Password

Kode	UC17
Use Case	Mengubah password
Actor	Admin
Pre-Condition	Aktor telah masuk.
Basic Course	1. Sistem menampilkan halaman ubah password. 2. Pengguna memilih salah satu akun pengguna. 3. Pengguna memasukkan password baru. 4. Pengguna menekan tombol ubah password. 5. Sistem menyimpan data dalam database.
Alternate Course	4.1. Pengguna menekan tombol ubah password tanpa mengisi password baru. 4.2. Sistem menampilkan pesan melengkapi.

4.2.1.2 Desain RESTful API

Desain RESTful API dibuat dengan menggambarkan aplikasi permainan Android sebagai aktor yang berinteraksi dengan sistem dalam use case diagram dan use case description sebagai berikut.



Gambar 4.27 Use Case Diagram API

Dari diagram di atas dapat dituliskan use case description sebagai berikut.

Tabel 4.19 Use Case Description Masuk Melalui Game

Kode	UC18
Use Case	Masuk melalui game
Actor	Game
Pre-Condition	Data pengguna sudah ada. Data murid sudah ada.
Basic Course	<ol style="list-style-type: none"> 1. Pengguna mengirimkan id dan password. 2. Sistem mencari menggunakan id lalu mencocokkan dengan password. 3. Sistem menemukan jika akun yang digunakan akun kelas. 4. Sistem mengirimkan data semua murid aktif.
Alternate Course	<ol style="list-style-type: none"> 2.1. Sistem tidak menemukan id pengguna atau password yang dimasukkan tidak sesuai. 2.2. Sistem mengirimkan error "Akun tidak ditemukan". 3.1.1. Sistem menemukan jika akun yang digunakan akun murid. 3.1.2. Sistem mengirimkan data murid tersebut. 3.2.1. Sistem menemukan jika akun yang digunakan bukan akun kelas atau murid. 3.2.2. Sistem mengirimkan error "Bukan akun kelas atau murid".

Tabel 4.20 Use Case Description Mengisi Rapor Melalui Game

Kode	UC19
Use Case	Mengisi rapor melalui game

Actor	Game
Pre-Condition	Data murid sudah ada. Data indikator sudah ada.
Basic Course	1. Pengguna mengirimkan no induk, indikator, dan waktu. 2. Sistem menyimpan data dalam database. 3. Sistem mengirimkan pesan sukses.
Alternate Course	-

Tabel 4.21 Use Case Description Menyimpan Hasil Bermain

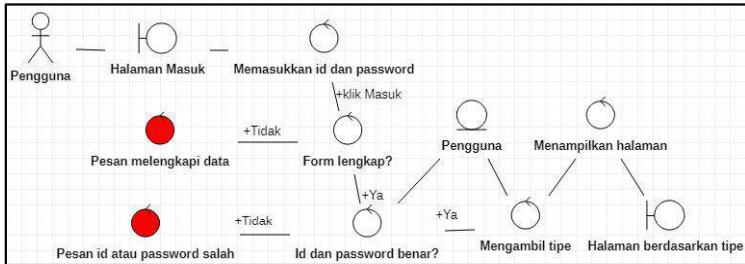
Kode	UC20
Use Case	Menyimpan hasil bermain
Actor	Game
Pre-Condition	Data murid sudah ada.
Basic Course	1. Pengguna mengirimkan nomor induk, game id, level, score. 2. Sistem menyimpan data dalam database. 3. Sistem mengirimkan pesan sukses.
Alternate Course	-

Tabel 4.22 Use Case Description Mengirim Hasil Bermain

Kode	UC21
Use Case	Mengirim hasil bermain
Actor	Game
Pre-Condition	Data hasil bermain sudah ada.
Basic Course	1. Pengguna mengirimkan nomor induk, game id. 2. Sistem mencari hasil permainan melalui nomor induk dan game id. 3. Sistem mengirimkan hasil permainan ke pengguna.
Alternate Course	-

4.2.1.3 Robustness Diagram

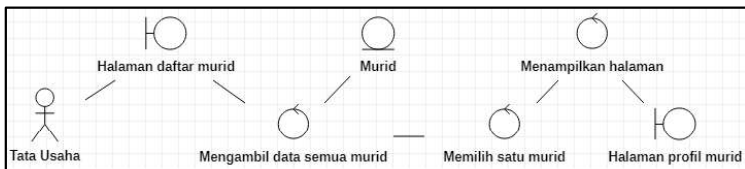
Berdasarkan use case diagram dan use case description pada 4.2.1.1 dan 4.2.1.2 didapatkan robustness diagram sebagai berikut.



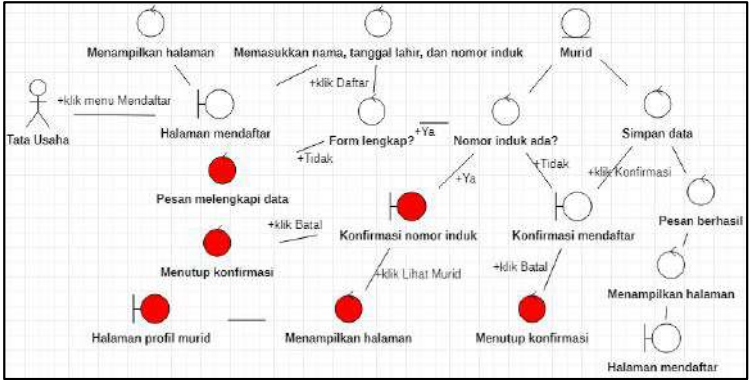
Gambar 4.28 Robustness Diagram Masuk



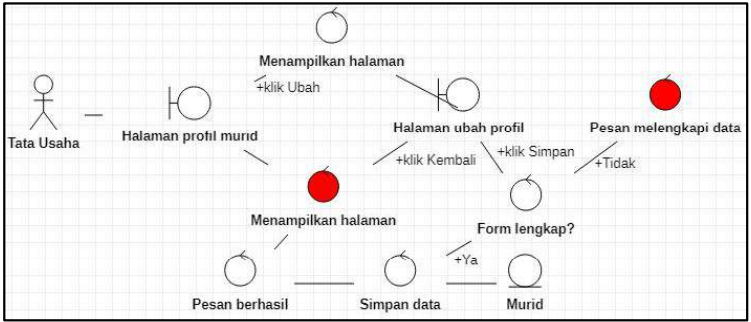
Gambar 4.29 Robustness Diagram Keluar



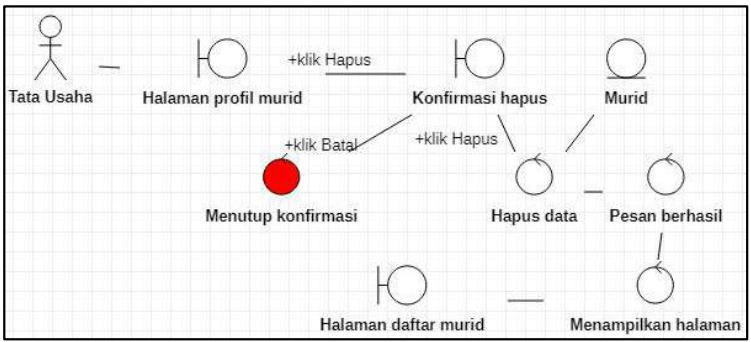
Gambar 4.30 Robustness Diagram Melihat Data Murid



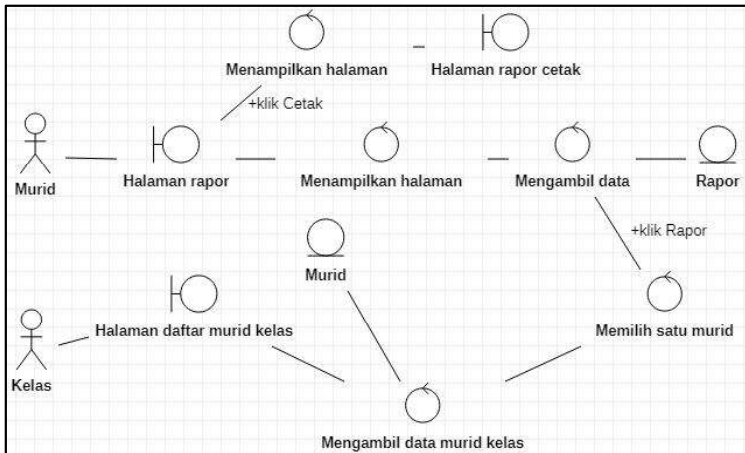
Gambar 4.31 Robustness Diagram Mengisi Data Murid



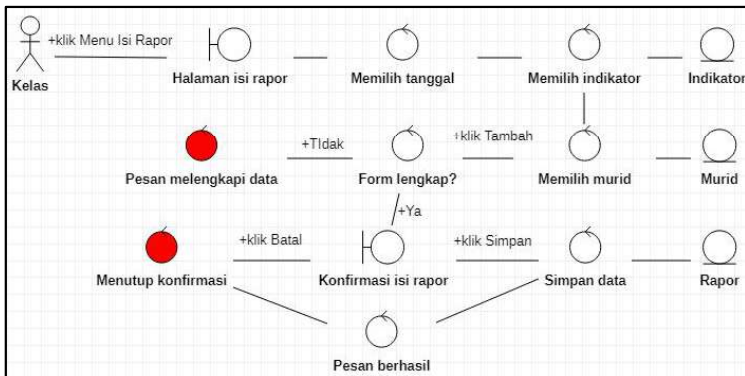
Gambar 4.32 Robustness Diagram Mengubah Data Murid



Gambar 4.33 Robustness Diagram Menghapus Data Murid



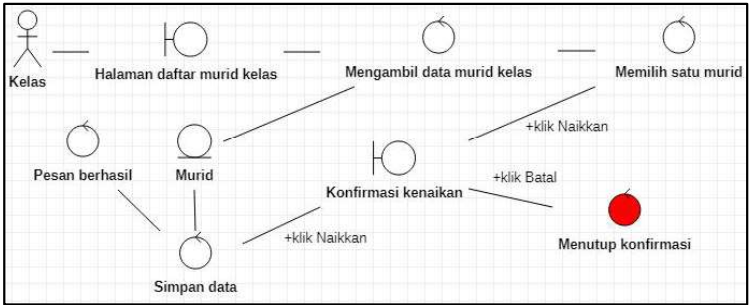
Gambar 4.34 Robustness Diagram Melihat dan Mencetak Rapor



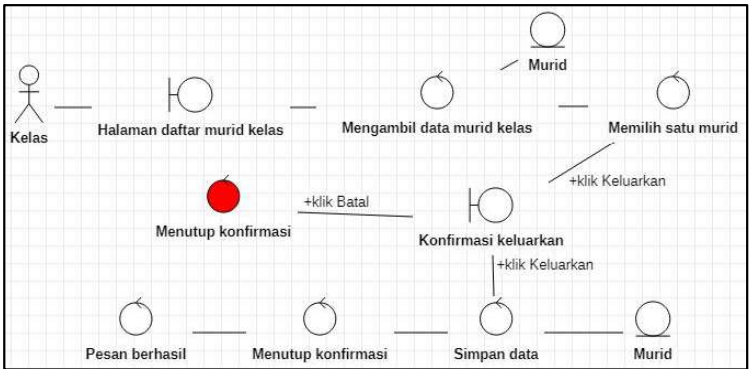
Gambar 4.35 Robustness Diagram Mengisi Rapor



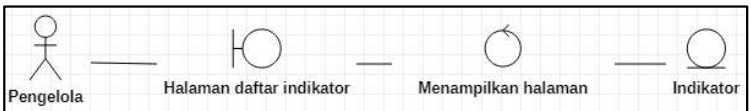
Gambar 4.36 Robustness Diagram Mengubah Rapor



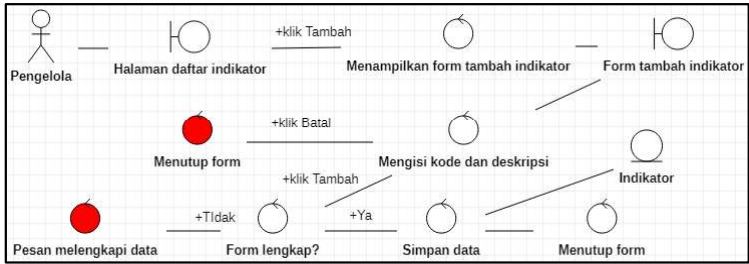
Gambar 4.37 Robustness Diagram Menaikkan Murid



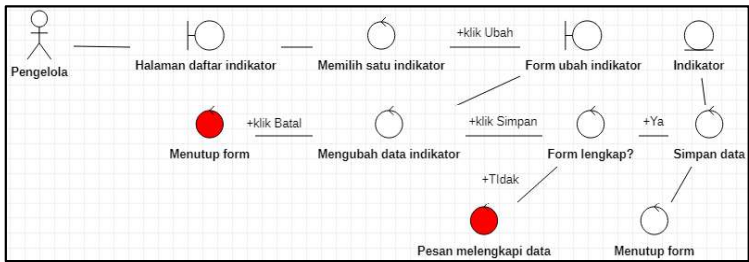
Gambar 4.38 Robustness Diagram Mengeluarkan Murid



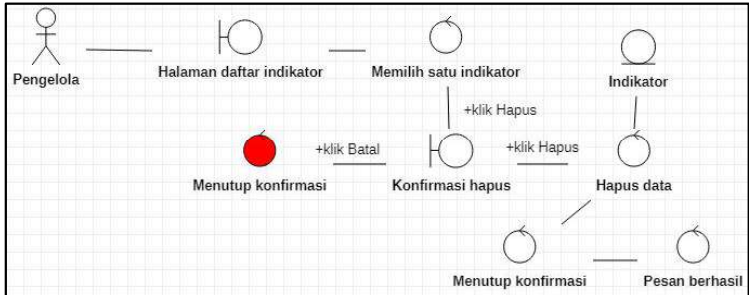
Gambar 4.39 Robustness Diagram Melihat Indikator



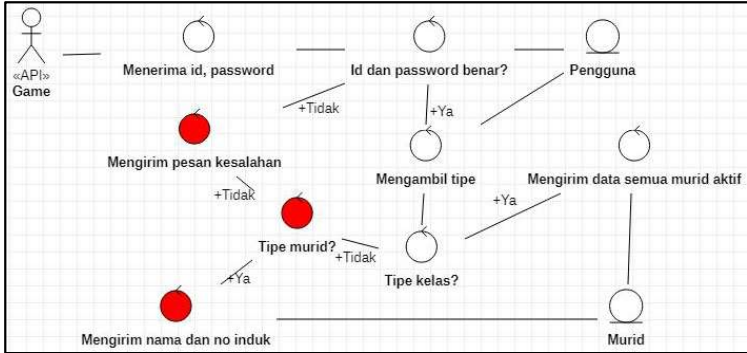
Gambar 4.40 Robustness Diagram Menambahkan Indikator



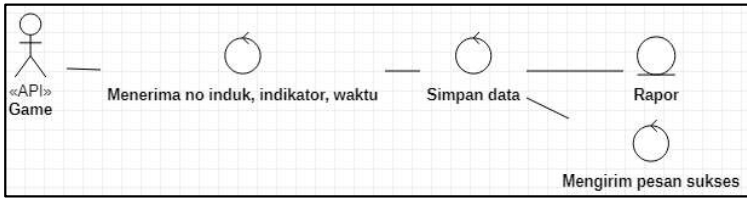
Gambar 4.41 Robustness Diagram Mengubah Indikator



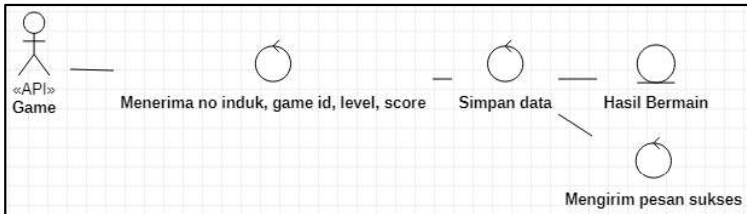
Gambar 4.42 Robustness Diagram Menghapus Indikator



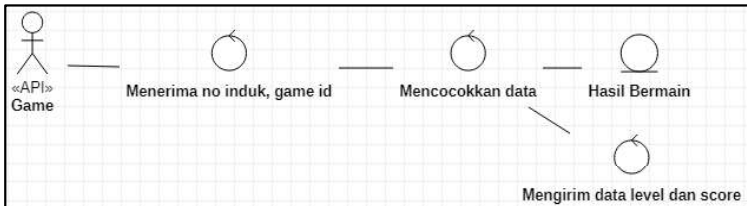
Gambar 4.43 Robustness Diagram Masuk Melalui Game



Gambar 4.44 Robustness Diagram Mengisi Rapor Melalui Game



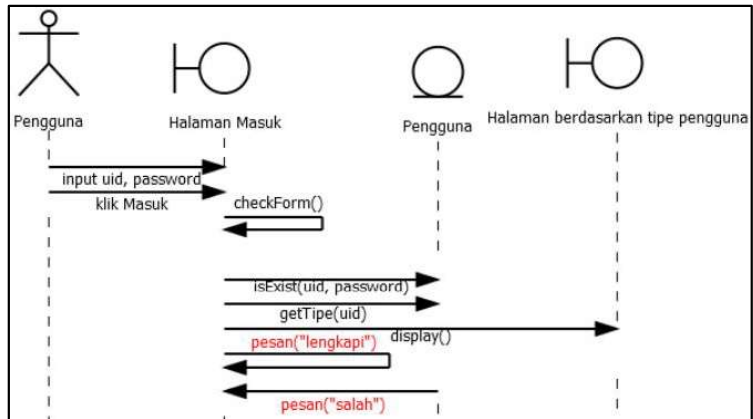
Gambar 4.45 Robustness Diagram Menyimpan Hasil Bermain



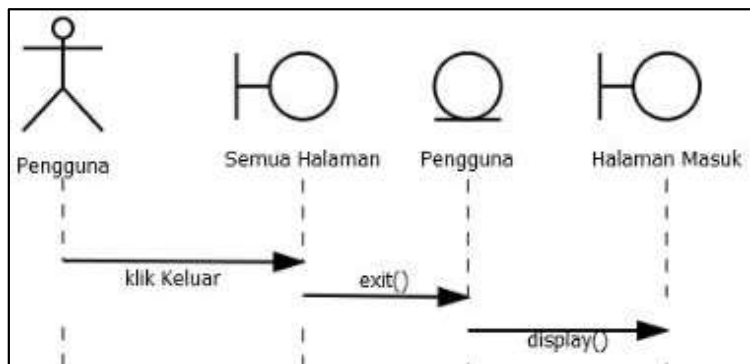
Gambar 4.46 Robustness Diagram Mengirim Hasil Bermain

4.2.1.4 Sequence Diagram

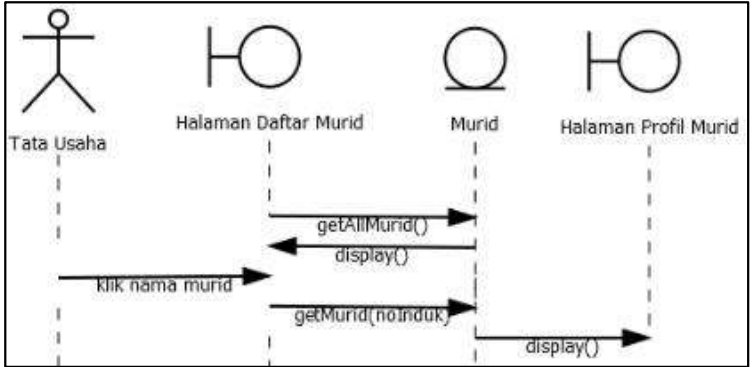
Berdasarkan robustness diagram di atas dapat digambarkan sequence diagram sebagai berikut.



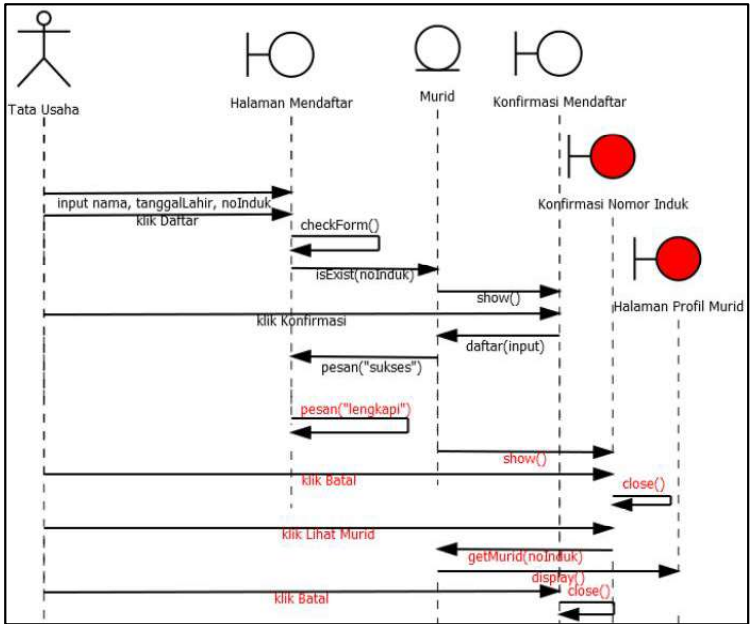
Gambar 4.47 Sequence Diagram Masuk



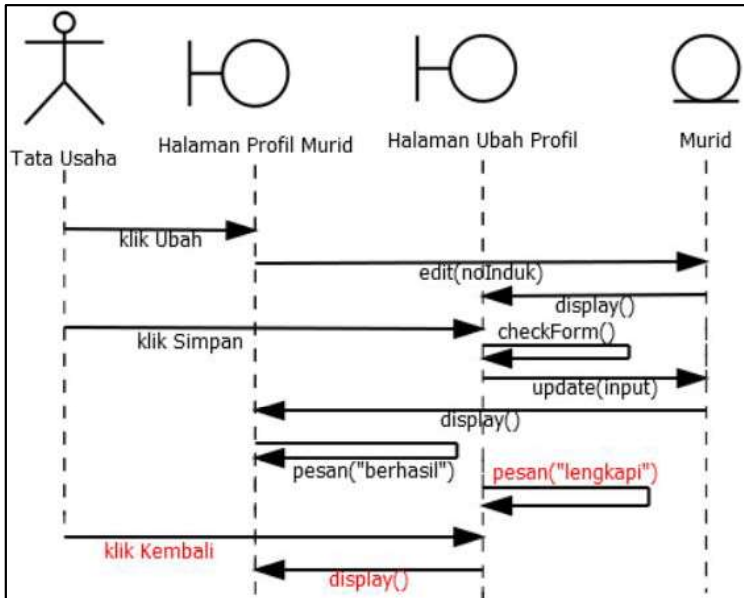
Gambar 4.48 Sequence Diagram Keluar



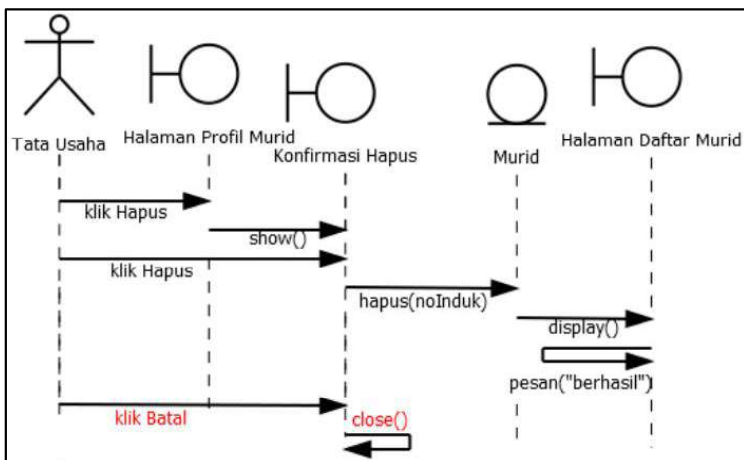
Gambar 4.49 Sequence Diagram Melihat Data Murid



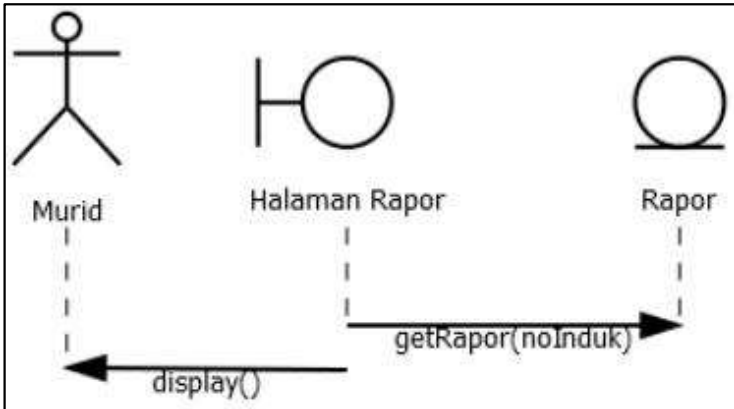
Gambar 4.50 Sequence Diagram Mengisi Data Murid



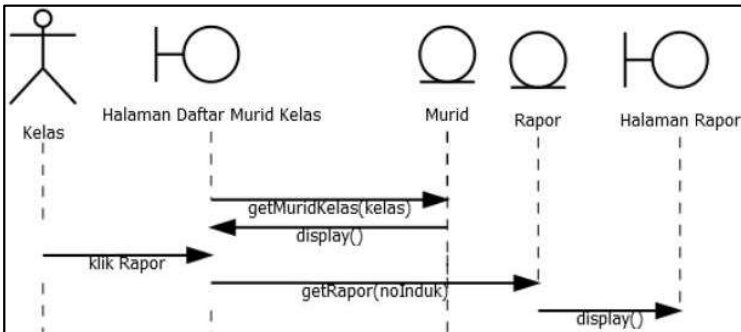
Gambar 4.51 Sequence Diagram Mengubah Data Murid



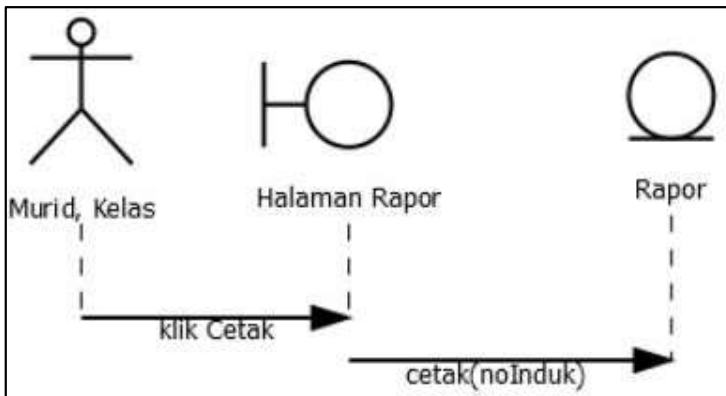
Gambar 4.52 Sequence Diagram Menghapus Data Murid



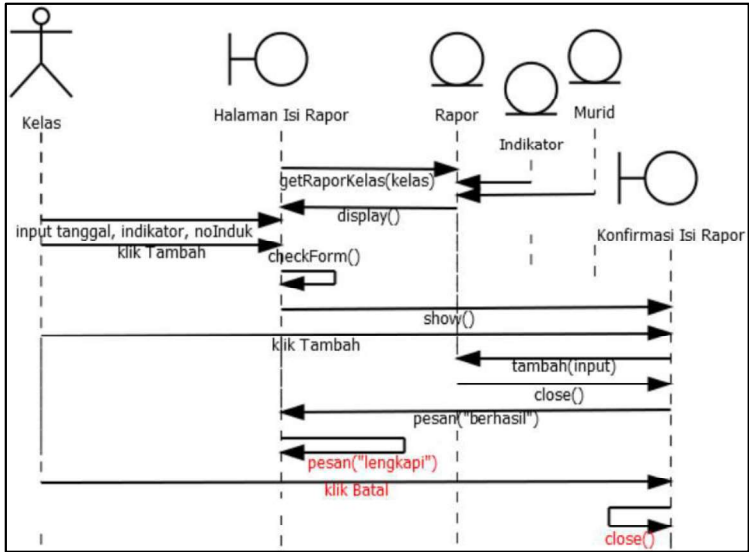
Gambar 4.53 Sequence Diagram Melihat Rapor (Murid)



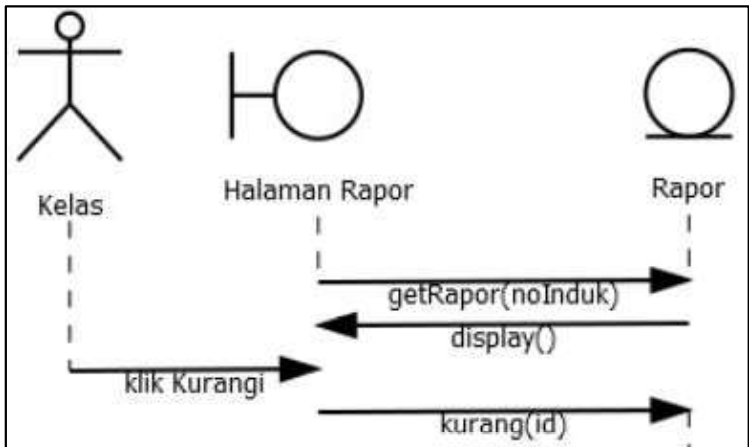
Gambar 4.54 Sequence Diagram Melihat Rapor (Kelas)



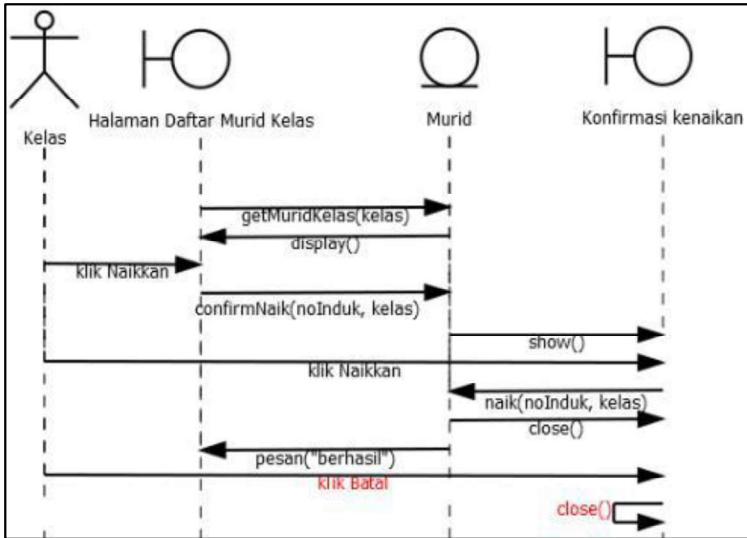
Gambar 4.55 Sequence Diagram Mencetak Rapor



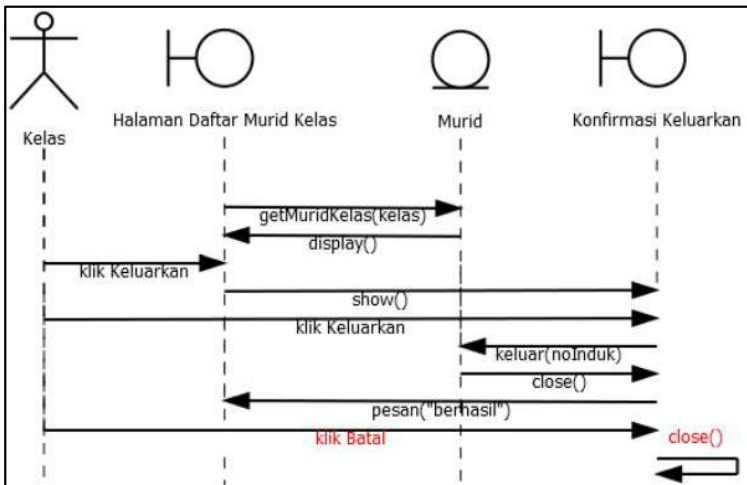
Gambar 4.56 Sequence Diagram Mengisi Rapor



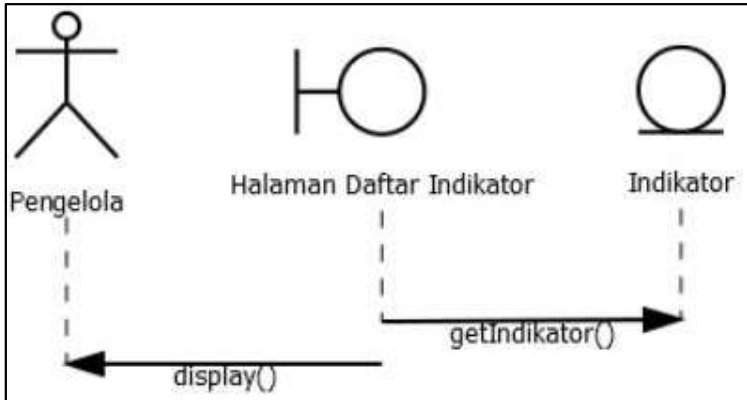
Gambar 4.57 Sequence Diagram Mengubah Rapor



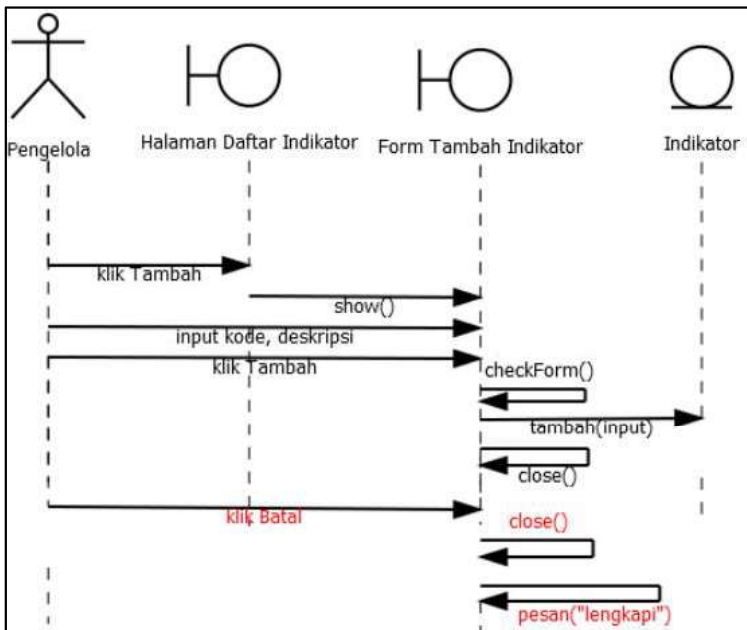
Gambar 4.58 Sequence Diagram Menaikkan Murid



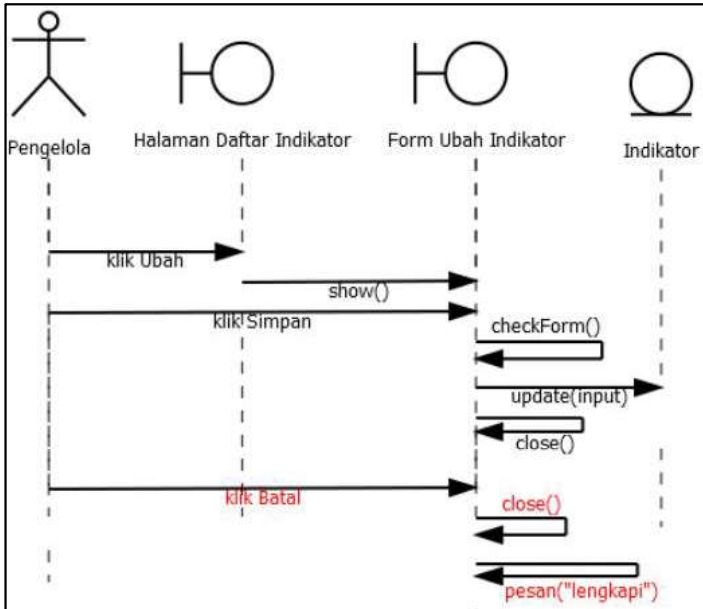
Gambar 4.59 Sequence Diagram Mengeluarkan Murid



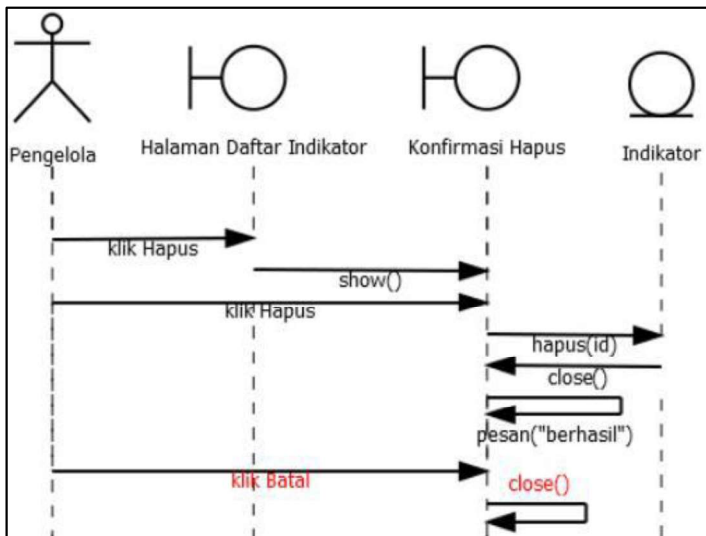
Gambar 4.60 Sequence Diagram Melihat Indikator



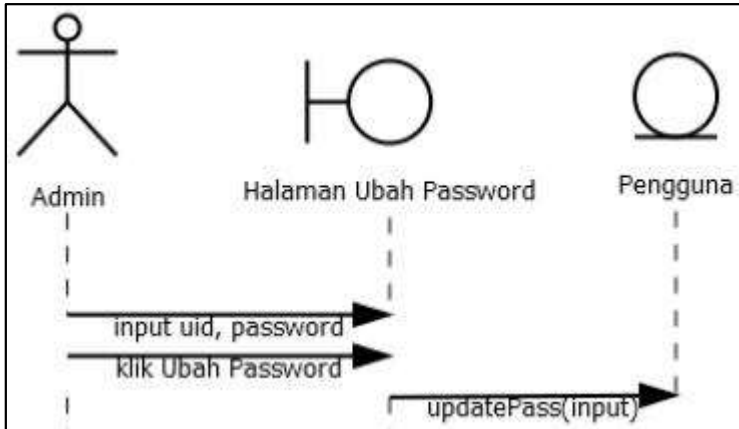
Gambar 4.61 Sequence Diagram Menambahkan Indikator



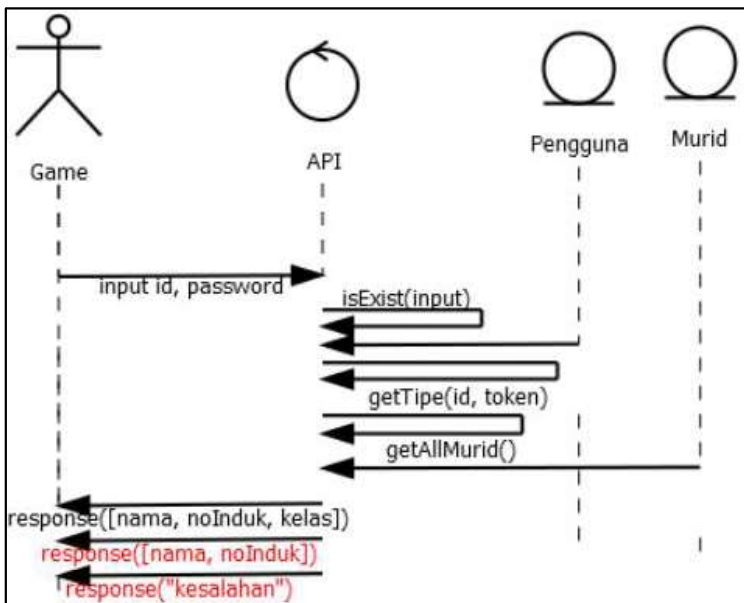
Gambar 4.62 Sequence Diagram Mengubah Indikator



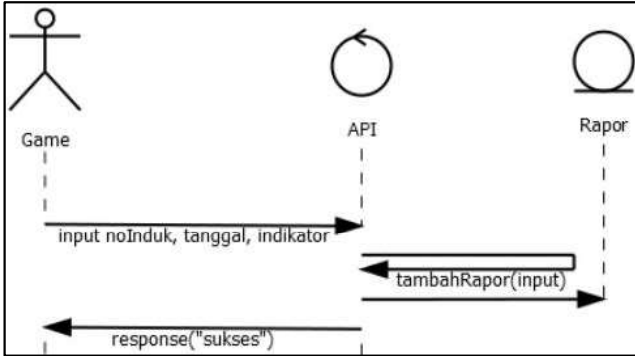
Gambar 4.63 Sequence Diagram Menghapus Indikator



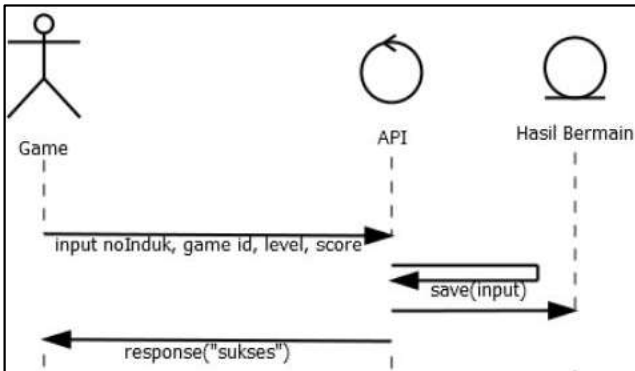
Gambar 4.64 Sequence Diagram Mengubah Password



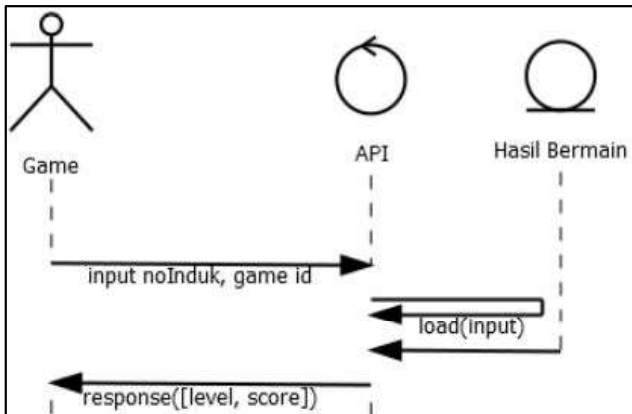
Gambar 4.65 Sequence Diagram Masuk Melalui Game



Gambar 4.66 Sequence Diagram Mengisi Rapor Melalui Game



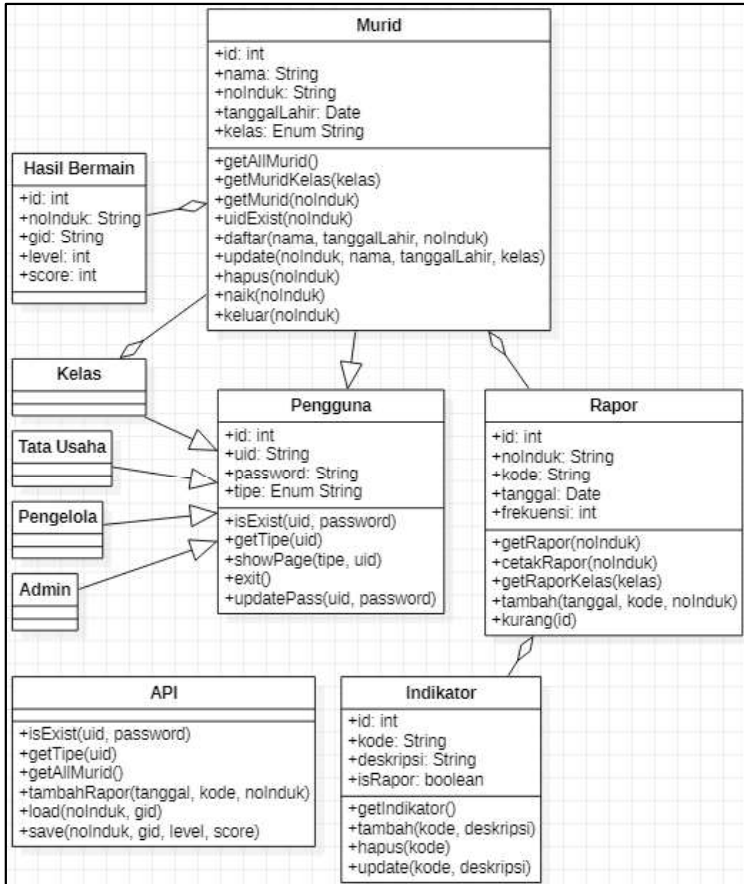
Gambar 4.67 Sequence Diagram Menyimpan Hasil Bermain



Gambar 4.68 Sequence Diagram Mengirim Hasil Bermain

4.2.1.5 Class Diagram

Berdasarkan domain model dan sequence diagram didapatkan class diagram sebagai berikut.



Gambar 4.69 Class Diagram Sistem Rapor

Pada class Hasil Bermain terdapat atribut gid yang merupakan singkatan dari game id. Atribut ini digunakan pada fungsi di class API.

Pada class Indikator terdapat atribut isRapor yang digunakan untuk menyaring kode mana saja yang dapat dimasukkan rapor.

4.2.2 Pra Produksi (Aplikasi Permainan)

Pra produksi mengembangkan konsep dari inisiasi menjadi desain untuk mengembangkan game.

4.2.2.1 Game Design

Berdasarkan konsep permainan pada inisiasi maka dibuatlah poin-poin desain sebagai berikut.

Definisi:

- Toko Bibi adalah game dengan genre kuis dengan tujuan edukasional karena permainannya didorong oleh pertanyaan yang diajukan kepada pemain.
- Toko Bibi dikembangkan untuk sistem operasi Android dan rencananya akan dimainkan menggunakan perangkat tablet milik PAUD Ciluk-Baa.
- Toko Bibi ditargetkan untuk pemain berusia 5-6 tahun atau setara dengan kelas RC-B di PAUD Ciluk-Baa.
- Terdapat seorang karakter berwujud anak kecil bernama Bibi yang menjadi ikon dari game ini dan juga mengajukan pertanyaan dalam permainan.

Mekanik dan peraturan:

- Gameplay dari permainan ini adalah Bibi akan memberikan pertanyaan terkait bilangan lalu pemain menarik angka ke kotak jawaban dan menekan tombol konfirmasi untuk menjawab.
- Jika jawaban pemain benar maka sistem akan memberikan feedback positif dan pemain mendapatkan skor sebanyak 1. Jika jawaban salah maka sistem akan memberikan feedback negatif.
- Terdapat 10 soal pada satu sesi permainan. Setiap kali pemain menekan tombol konfirmasi maka soal akan dilanjutkan ke soal berikutnya.
- Satu bagian permainan berakhir setelah 10 soal selesai dijawab atau pemain kehabisan waktu.
- Setelah permainan berakhir maka skor akan dihitung dan dikirim ke sistem rapor. Jika pemain mendapatkan skor

minimal di bagian tersebut maka game juga akan mengirimkan indikator yang sesuai ke sistem rapor.

- Akan terdapat 4 bagian dimana bagian 2-4 terkunci, pemain dapat memainkan bagian selanjutnya jika telah mendapat skor minimal di bagian yang dimainkan saat ini.
- Terdapat 4 tipe permainan yang dapat dipilih sebelum memulai bagian atau diacak setiap soal diberikan. Jumlah tipe permainan ini didasari oleh kebutuhan validasi sejumlah 4 kali frekuensi pengamatan.
- Skor minimal (Skor Min) ditentukan setelah berdiskusi dengan pihak PAUD Ciluk-Baa.

Tabel 4.23 Informasi Bagian pada Permainan

Ba- gian	Pertanyaan	Indikator	Skor Min
1	Jumlah buah 1-10 secara berurutan.	3.3.2.2. Menyebutkan bilangan 1-10 berurutan sambil menghitung benda.	10
2	Jumlah buah 1-10 secara acak.	3.3.2.3. Memasangkan lambang bilangan dengan jumlah benda 1-10.	10
3	Operasi pengurangan dengan hasil 1-9 secara acak.	3.3.4.1. Menghitung dan menyebutkan hasil pengurangan.	7
4	Operasi penjumlahan dengan hasil 2-10 secara acak.	3.3.4.2. Menghitung dan menyebutkan hasil penjumlahan.	7

Pengguna:

- Pengguna utama Toko Bibi adalah pengajar PAUD Ciluk-Baa atau pengguna Kelas di sistem rapor.
- Pengguna lainnya adalah pengguna Murid di sistem rapor dan pengguna Tamu untuk bermain secara offline.

- Alur permainan ini adalah pemain masuk menggunakan akun Sistem Rapor di tampilan login lalu tampilan menu akan terbuka dan pemain memilih siapa yang bermain dan memainkan bagian apa. Setelah menyelesaikan 1 bagian maka pemain dapat memilih memainkan ulang atau kembali ke tampilan menu.

Berdasarkan poin-poin desain maka didapatkan kebutuhan fungsional sebagai berikut.

Tabel 4.24 Kebutuhan Fungsional Aplikasi Permainan

Kode	Kebutuhan	Deskripsi
FR001	Masuk	Mencocokkan identitas lalu masuk ke tampilan menu
FR002	Memilih murid	Menampilkan pilihan kelas dan murid untuk pengguna kelas
FR003	Menampilkan nama	Menampilkan nama pengguna atau murid yang dipilih
FR004	Mengelola bagian	Mengunci dan membuka bagian
FR005	Membuat pertanyaan	Membuat pertanyaan sesuai bagian
FR006	Memberi feedback	Memberi feedback sesuai jawaban pemain
FR007	Mengisi rapor	Mengisi rapor sesuai indikator
FR008	Mengelola skor	Menghitung, menyimpan dan mengirim data hasil bermain
FR009	Mengakhiri bagian	Mengakhiri bagian setelah soal atau waktu habis
FR010	Memilih tipe	Memilih atau mengacak tipe permainan

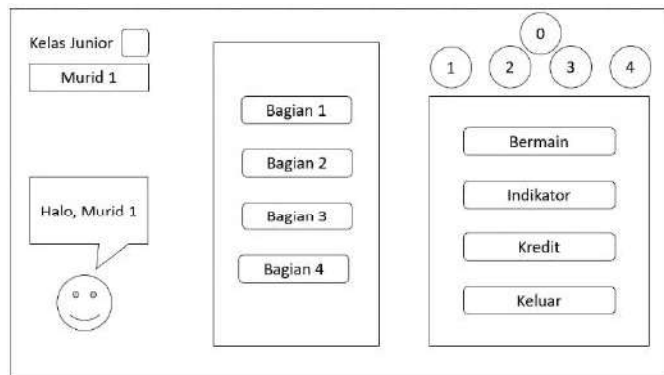
4.2.2.2 Mockup Game

Berikut ini merupakan gambaran awal (mockup) dari game Toko Bibi untuk memenuhi desain.



Gambar 4.70 Mockup Login

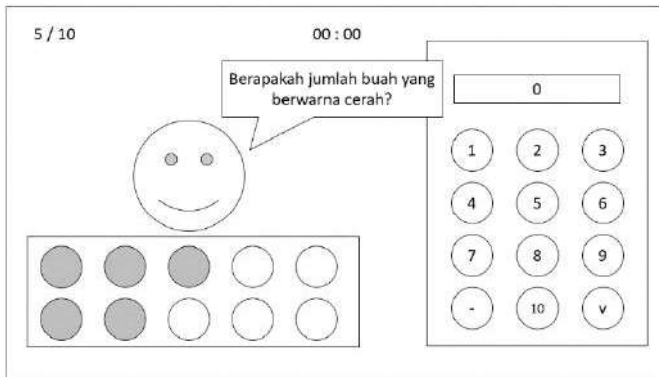
Tampilan pertama yang dilihat oleh seluruh pengguna. Saat online pengguna dapat memasukkan identitas dan password lalu menekan tombol Masuk. Saat offline pengguna tetap dapat masuk menggunakan tombol Masuk sebagai tamu.



Gambar 4.71 Mockup Menu

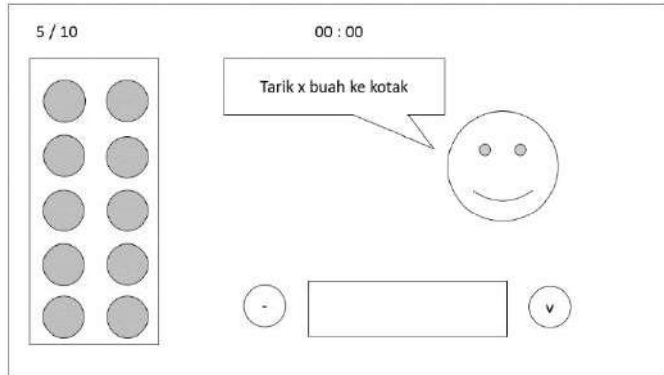
Setelah masuk maka karakter akan menampilkan nama pengguna pada balon teks. Jika pengguna adalah Kelas maka

pengguna dapat memilih murid yang akan bermain lewat daftar yang ada di pojok kiri. Jika bukan maka daftar tersebut tidak ditampilkan. Setelah tombol Bermain ditekan maka daftar bagian akan muncul di tengah. Tombol bagian juga menampilkan skor yang didapat pada bagian itu. Setelah tombol bagian ditekan maka pemain akan dibawa ke tampilan game sesuai pilihan. Tombol indikator akan menampilkan teks mengenai bagian berapa memenuhi indikator apa saja dan sumbernya. Tombol Kredit akan menampilkan siapa saja yang berperan dalam pengembangan game ini. Tombol Keluar akan membawa pemain ke tampilan login. Terdapat tombol dengan angka 0-4, tombol ini menentukan tipe permainan (1-4) dan 0 akan mengacak tipe setiap ganti soal.



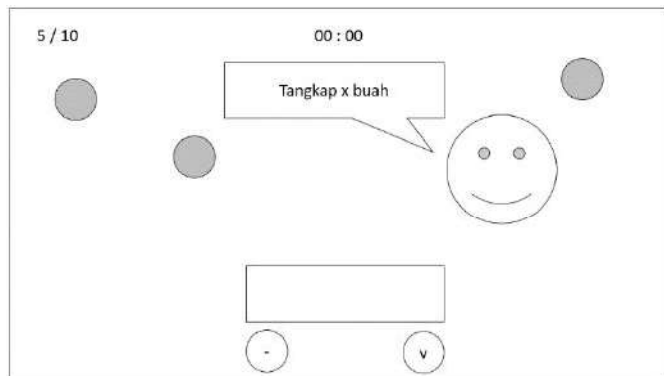
Gambar 4.72 Mockup Game Tipe 1

Untuk menjawab pertanyaan tipe 1, pemain menyeret angka pada bagian kanan ke kotak jawaban di atasnya. Pemain lalu menekan tombol centang untuk konfirmasi jawaban. Pemain dapat menghapus jawaban dengan menekan tombol kurang. Pertanyaan yang diajukan pada bagian 1 dan 2 sama. Pada bagian 3 pertanyaan berubah menjadi soal cerita mengenai pengurangan dan pada bagian 4 pertanyaan berubah menjadi soal cerita mengenai penjumlahan. Terdapat tampilan soal ke berapa dan sisa waktu pada bagian atas.



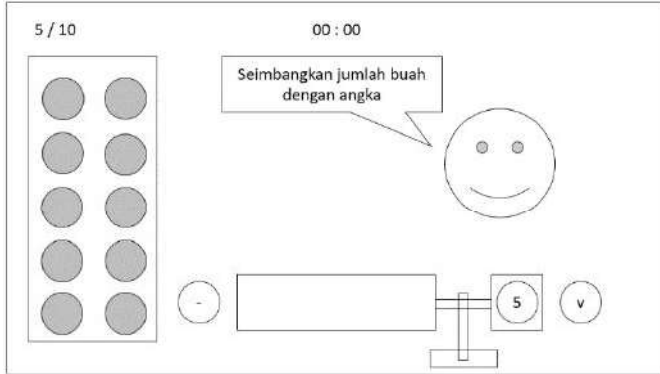
Gambar 4.73 Mockup Game Tipe 2

Untuk menjawab pertanyaan game tipe 2, pemain menarik buah ke dalam kotak sesuai permintaan. Tombol minus akan mengembalikan 1 buah dari kotak ke tempat asalnya.



Gambar 4.74 Mockup Game Tipe 3

Untuk menjawab pertanyaan game tipe 3, pemain menangkap buah yang berjatuhan dengan menggeser kotak. Buah akan tertangkap jika bersentuhan dengan kotak saat kotak digeser. Tombol minus menghapus 1 buah dari kotak.



Gambar 4.75 Mockup Game Tipe 4

Untuk menjawab pertanyaan game tipe 4, pemain menarik buah ke kotak sampai jumlahnya seimbang dengan angka yang diminta. Tombol minus mengembalikan 1 buah dari kotak ke tempat asalnya.

Desain tipe permainan didapatkan dari membandingkan mekanisme dari beberapa game android komersil (COTS) yang ada pada Google PlayStore sebagai berikut.

Tabel 4.25 Mekanisme Game Edukasi Android Komersil

Nama Game	Mekanisme
Little Panda Math Genius	Operasi bilangan dimana pertanyaan berupa lambang bilangan dengan visualisasi benda dan dijawab dengan menekan lambang bilangan
Preschool and Kindergarten Learning Games	Membilang dimana beberapa buah diberi lambang bilangan dan diminta menekan secara berurutan
	Operasi bilangan dimana pertanyaan berupa lambang bilangan dengan visualisasi benda dan dijawab dengan menekan lambang bilangan

Belajar Angka - Learn Numbers with Marbel	Membilang dimana instruksi berupa suara yang menyebutkan lambang bilangan dan dijawab dengan menekan lambang bilangan yang diminta
	Mencari 2 pasang lambang bilangan dari beberapa kartu yang tertutup
	Memasangkan siluet lambang bilangan dengan lambang bilangan
Marbel Kids Learn To Count	Membilang dengan memasang jumlah benda dengan lambang bilangan
	Operasi bilangan dimana pertanyaan berupa lambang bilangan dengan visualisasi benda dan dijawab dengan menekan lambang bilangan
	Menentukan kurang dari dan lebih dari lambang bilangan menggunakan timbangan
Marbel Preschool	Membilang dimana instruksi berupa suara yang menyebutkan lambang bilangan dan dijawab dengan memilih lambang bilangan yang bergerak secara vertikal

Beberapa mekanisme aplikasi permainan pada tabel dipilih dan diadaptasi sesuai kebutuhan dan kemampuan pengembangan aplikasi permainan untuk PAUD Ciluk-Baa.

Halaman ini sengaja dikosongkan

BAB V IMPLEMENTASI

Pada bagian ini akan menjelaskan mengenai proses implementasi terhadap sistem rapor dan aplikasi permainan berdasarkan proses perancangan sebelumnya. Bagian ini melakukan fase 3 dan 4 yang dijelaskan pada metodologi.

5.1 Lingkungan Implementasi

Pengembangan sistem rapor dan aplikasi permainan menggunakan komputer dengan spesifikasi yang tertera pada Tabel 5.1

Tabel 5.1 Spesifikasi Komputer Implementasi

<i>Type</i>	ASUS X555LF
<i>Processor</i>	Intel® Core™ i5-5200U CPU @2.2GHz
<i>Memory (RAM)</i>	8 GB
<i>Sistem Operasi</i>	Windows 10 Education 64Bit
<i>Graphic Card</i>	Nvidia GeForce 930M

Sistem rapor dikembangkan dengan menggunakan beberapa teknologi seperti IDE (*Integrated Development Environment*), bahasa pemrograman, *database*, *framework*, *bootstrap*, dan *library* yang terdapat pada Tabel 5.2.

Tabel 5.2 Teknologi Pengembangan Sistem Rapor

<i>IDE</i>	Visual Studio Code 1.45.1
<i>Bahasa Pemrograman</i>	PHP 7.4.3
<i>Database</i>	MySQL 5.0.2
<i>Framework</i>	Laravel 7.2.2
<i>Bootstrap</i>	SB Admin 2

Library	Sweet Alert (untuk memperindah alert) Laravel DomPdf (untuk membuat pdf) JWT Auth (untuk autentikasi API) Select2 (untuk form pilihan)
----------------	---

Aplikasi permainan dikembangkan dengan menggunakan beberapa teknologi seperti *game engine* dan bahasa yang terdapat pada Tabel 5.3.

Tabel 5.3 Teknologi Pengembangan Aplikasi Permainan

Game Engine	Construct 3 r197.2 Personal Edition
Bahasa	HTML5

5.2 Fase 3

Tahapan ini mengimplementasikan desain dari fase sebelumnya menjadi sistem dan aplikasi permainan.

5.2.1 Pengembangan (Sistem Rapor)

Sistem rapor dikembangkan menggunakan *framework* Laravel, karena itu hal pertama yang dilakukan adalah instalasi Laravel lalu dilanjutkan dengan memasang *bootstrap* dan *library*. Kode sumber yang ditampilkan hanya potongan dari beberapa kode sumber yang dipilih.

5.2.1.1 Implementasi Database

Database diimplementasi berdasarkan *class diagram* dimana hanya class yang memiliki atribut yang dibuat tabelnya menggunakan perintah membuat *migration*. Perintah ini akan membuat *file migration* pada “database/migrations” seperti pada Gambar 5.1 yang berisi tabel untuk disesuaikan dengan kebutuhan berdasarkan atribut pada class seperti pada Kode 5.1.

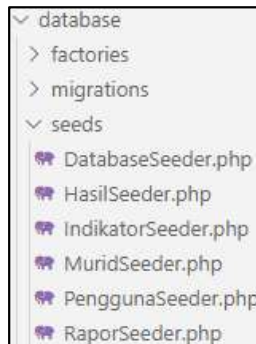


Gambar 5.1 File Migration Sistem Rapor

```
public function up()
{
    Schema::create('rapor', function (Blueprint $table) {
        $table->id();
        $table->string('noInduk');
        $table->string('kode');
        $table->date('tanggal');
        $table->integer('frekuensi');
        $table->timestamps();
    });
}
```

Kode 5.1 Skema Migrasi Tabel Rapor

Dilanjutkan dengan perintah membuat *seeder*. Perintah ini akan membuat *file seeder* pada “database/seeds” seperti pada Gambar 5.2 untuk mengisi tabel yang dibuat pada migrasi dengan mock *data* seperti pada Kode 5.2. Data ini akan digunakan saat pengujian.



Gambar 5.2 File Seeder Sistem Rapor

```
DB::table('pengguna')->insert([
    'uid' => 'kelas',
    'password' => Hash::make('secret'),
    'tipe' => 'Kelas',
]);

DB::table('pengguna')->insert([
    'uid' => '123456789',
    'password' => Hash::make('cobamurid'),
    'tipe' => 'Murid',
]);
```

Kode 5.2 PenggunaSeeder

```
public function run()
{
    DB::table('rapor')->insert([
        'noInduk' => '202001250',
        'kode' => '1.2.3.4.',
        'tanggal' => '2020-07-01',
        'frekuensi' => '2',
    ]);
}
```

Kode 5.3 RaporSeeder

```
DB::table('murid')->insert([
    'nama' => '-',
    'noInduk' => '0',
    'tanggalLahir' => '2019-08-01',
    'kelas' => 'RC-B',
]);

DB::table('murid')->insert([
    'nama' => 'Uji Coba Junior',
    'noInduk' => '123456789',
    'tanggalLahir' => '2019-08-01',
    'kelas' => 'Junior',
]);
```

Kode 5.4 MuridSeeder

Pada MuridSeeder terdapat murid dengan noInduk 0 di masing-masing kelas. Murid-murid ini berfungsi sebagai placeholder.

```
public function run()
{
    DB::table('indikator')->insert([
        'kode' => '1.2.3.4.',
        'deskripsi' => 'Indikator ini 1.2.3.4.',
    ]);

    DB::table('indikator')->insert([
        'kode' => '1.2.3.5.',
        'deskripsi' => 'Indikator ini 1.2.3.5.',
    ]);
}
```

Kode 5.5 IndikatorSeeder

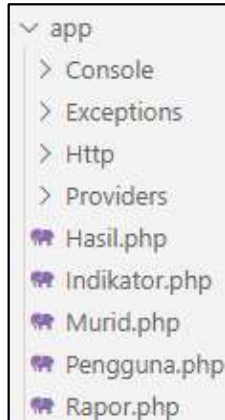
```
public function run()
{
    DB::table('hasilBermain')->insert([
        'noInduk' => '123456789',
        'gid' => '1',
        'level' => '1',
        'score' => '0',
    ]);
}
```

Kode 5.6 HasilSeeder

Perintah *migrate fresh seed* akan membuat tabel dan isinya pada *database* yang sesuai dengan pengaturan pada *file .env*.

5.2.1.2 Implementasi Model

Model pada Laravel adalah obyek pada *class diagram* yang sama dengan yang telah diimplementasikan pada *database* pada bagian sebelumnya. Dengan perintah membuat *model*, maka *file model* akan dibuat pada 'app' seperti yang terlihat pada Gambar 5.3.



Gambar 5.3 File Model Sistem Rapor

```

class Pengguna extends Authenticatable implements JWTSubject
{
    protected $table = 'pengguna';

    protected $fillable = [
        'uid', 'password',
    ];

    protected $attributes = [
        'tipe' => 'Murid',
    ];

    public function getJWTIdentifier()
    {
        return $this->getKey();
    }

    public function getJWTCustomClaims()
    {
        return [];
    }
}

```

Kode 5.7 Model Pengguna

Model pengguna mendefinisikan obyek Pengguna menggunakan tabel bernama 'pengguna' pada database. Atribut yang dapat diubah atau diisi adalah 'uid' dan 'password'.

Atribut ‘tipe’ memiliki bawaan isi ‘Murid’. Sisanya adalah fungsi untuk menggunakan fitur JSON Web Token (JWT).

```

class Rapor extends Model
{
    protected $table = 'rapor';

    protected $fillable = [
        'noInduk', 'kode', 'tanggal', 'frekuensi',
    ];

    public function getTanggalAttribute()
    {
        return \Carbon\Carbon::parse($this->attributes['tanggal'])
            ->format('d F Y');
    }

    public function setTanggalAttribute($value)
    {
        $this->attributes['tanggal'] = \Carbon\Carbon::parse($value)->format('Y-m-d');
    }

    public function murid(){
        return $this->belongsTo('App\Murid', 'noInduk', 'noInduk');
    }
}

```

Kode 5.8 Model Rapor

Model rapor memiliki Carbon yang digunakan untuk mengubah format pada atribut tanggal. Atribut tanggal memiliki format (‘d F Y’) saat diberikan dan format (‘Y-m-d’) saat disimpan. Terdapat pula fungsi dengan nama *model* yang lain yaitu *model* murid, fungsi ini digunakan untuk menghubungkan kedua *model* berdasarkan atribut ‘noInduk’ dimana ‘belongsTo’ menggambarkan hubungan one-to-many (satu murid memiliki banyak rapor).

Berikut ini merupakan potongan kode dari beberapa model.

```

class Murid extends Model
{
    protected $table = 'murid';

    protected $fillable = [
        'nama', 'noInduk', 'tanggalLahir', 'kelas',
    ];

    public function getTanggalLahirAttribute()
    {
        return \Carbon\Carbon::parse($this->attributes['tanggalLahir'])
            ->format('d F Y');
    }

    public function setTanggalLahirAttribute($value)
    {
        $this->attributes['tanggalLahir'] = \Carbon\Carbon::parse($value)->format('Y-m-d');
    }
}

```

Kode 5.9 Model Murid

```

class Indikator extends Model
{
    protected $table = 'indikator';

    protected $fillable = [
        'kode', 'deskripsi', 'isRapor',
    ];
}

```

Kode 5.10 Model Indikator

```

class Hasil extends Model
{
    protected $table = 'hasilBermain';

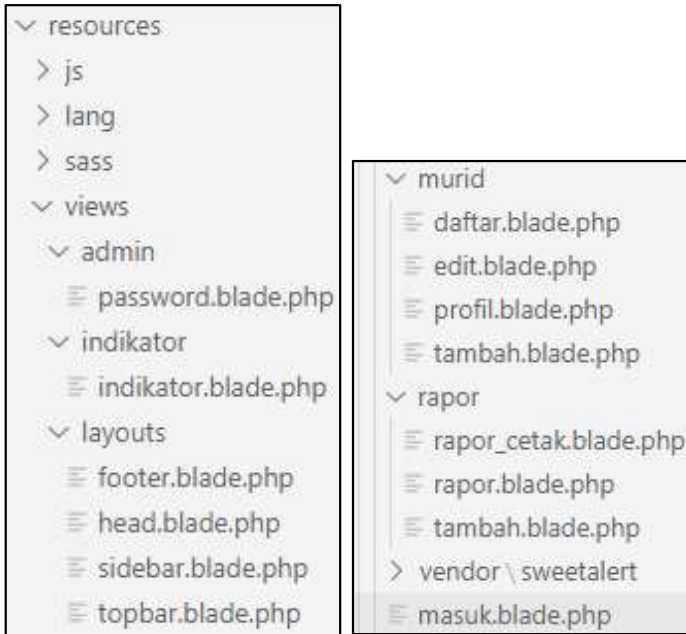
    protected $fillable = [
        'noInduk', 'gid', 'level', 'score',
    ];
}

```

Kode 5.11 Model Hasil

5.2.1.3 Implementasi View

View adalah kode yang ditulis untuk membuat tampilan sesuai desain pada *storyboard* dengan bantuan *bootstrap*. *View* tersimpan pada 'resources/views' dan memiliki tambahan 'blade' pada nama file seperti pada Gambar 5.4.



Gambar 5.4 File Blade View Sistem Rapor

```

<h6 class="text-gray-900">Indikator Terpenuhi</h6>
<div class="form-group">
  <select class="form-control form-control-user select2" name="indikator[]"
    multiple="multiple">
    @foreach ($indikators as $indikator)
      <option value="{{ $indikator->kode }}" {{ $indikator->deskripsi }}"
        {{ (collect(old('indikator'))->contains($indikator->kode." ")).$indikator->deskripsi}}
        {{ $indikator->kode }} {{ $indikator->deskripsi }}</option>
    @endforeach
  </select>
</div>
@error('indikator')
<div class="alert alert-danger">{{ $message }}</div>
@enderror

```

Kode 5.12 Form Select2 Indikator Halaman Tambah Rapor

Halaman Tambah Rapor meminta input indikator menggunakan *form* dengan bantuan *library* *select2* sehingga memungkinkan dilakukan pencarian indikator dan memilih beberapa indikator.

Berikut ini merupakan potongan kode dari beberapa *view*.

```

<div class="text-center">
  <h1 class="h4 text-gray-900 mb-4">Sistem Rapor Ciluk-baa</h1>
</div>
<form class="user" method="POST" action="{{ route('masuk') }}">
  @csrf
  <div class="form-group">
    <input type="text" class="form-control form-control-user" name="uid"
      placeholder="Identitas" value="{{ old('uid') }}">
  </div>
  @error('uid')
  <div class="alert alert-danger">{{ $message }}</div>
  @enderror

```

Kode 5.13 Halaman Masuk

```

@extends('layouts.sidebar')

@section('title', 'Rapor Murid')

@section('customCSS')
<link href="{{ asset('vendor/datatables/dataTables.bootstrap4.min.css') }}" rel="stylesheet">
@endsection

@section('content')
<div class="container-fluid">
  <h1 class="h3 mb-2 text-gray-800">Rapor {{ $murid->nama }}</h1>
  <a @can('isKelas') href="{{ route('cetakRapor', $murid->noInduk) }}" @endcan @can('isMurid')
    href="{{ route('cetakRapor.murid', $murid->noInduk) }}" @endcan class="btn btn-primary">
    <span class="icon text-white-50">
      <i class="fas fa-print"></i>
    </span>
    <span class="text">Cetak</span>
  </a>

```

Kode 5.14 Halaman Rapor

```

@extends('layouts.sidebar')

@section('title', 'Daftar Murid')

@section('customCSS')
<link href="{{ asset('vendor/datatables/dataTables.bootstrap4.min.css') }}" rel="stylesheet">
@endsection

@section('content')
<div class="container-fluid">
  <div class="card shadow mb-4">
    <div class="card-header py-3">
      @can('isTU')
      <h6 class="m-0 font-weight-bold text-primary">Data Semua Murid</h6>
      @elsecan('isKelas')
      <h6 class="m-0 font-weight-bold text-primary">Data Semua Murid Kelas {{ $murid->kelas }}</h6>
      <p id="kelas" hidden>{{ $murid->kelas }}</p>
      @endcan
    </div>

```

Kode 5.15 Halaman Daftar Murid

Berikut ini beberapa tampilan halaman hasil implementasi kode view.

Gambar 5.5 Halaman Masuk

Gambar 5.6 Halaman Tambah Rapor

Kode	Indikator Kemampuan Anak	Pencapaian Pertama	Probandal	Kurang
1.2.3.4.	Indikator no 1.2.3.4.	05 May 2020	3	-

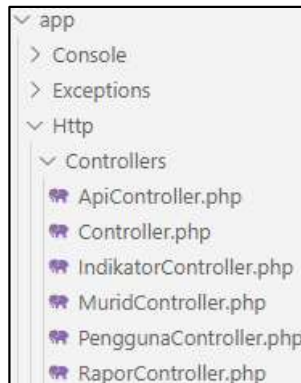
Gambar 5.7 Halaman Rapor

Nama	No. Induk	Rapor	Nilai	Kembalikan
Anshika Pramadi	302001210			
Aji Litta Junior	113406100			

Gambar 5.8 Halaman Daftar Murid Kelas

5.2.1.4 Implementasi Controller

Controller adalah kode yang ditulis untuk mengolah logika dan data sesuai desain *sequence diagram*. *Class* yang dibuat controllernya adalah yang memiliki fungsi. Dengan perintah membuat *controller*, maka *file controller* akan dibuat pada ‘app/http/controller’ seperti pada Gambar 5.9.



Gambar 5.9 File Controller Sistem Rapor

Berikut ini merupakan potongan kode dari beberapa *controller*.

```

class PenggunaController extends Controller{
  public function isExist(Request $request){
    $request->validate([
      'uid' => 'required',
      'password' => 'required',
    ],
    [
      'uid.required' => 'Identitas tidak boleh kosong',
      'password.required' => 'Password tidak boleh kosong',
    ]
    );
    $uid = $request->uid;
    $credential = [
      'uid' => $uid,
      'password' => $request->password
    ];
    if (Auth::attempt($credential)) {
      return $this -> getTipe($uid);
    }
    else return back()->withInput()->withErrors(['wrong' =>
      'Identitas atau password salah']);
  }
}

```

Kode 5.16 Fungsi IsExist PenggunaController

Fungsi isExist terdapat *validate* yang berfungsi untuk mengatur batasan dari *input* dan pesan kesalahannya. Jika pengguna gagal masuk maka tampilan akan kembali ke halaman saat ini dengan pesan kesalahan. Jika pengguna berhasil masuk maka fungsi *getTipe* akan dijalankan.

```

public function getTipe($uid){
  $tipe = Pengguna::where('uid', $uid)->value('tipe');
  return $this -> showPage($tipe, $uid);
}

public function showPage($tipe, $uid){
  if($tipe == "Admin"){
    return redirect()->route('admin');
  }
  if($tipe == "Tata Usaha"){
    return redirect()->route('daftarMurid');
  }
  if($tipe == "Kelas"){
    $kelas = 'junior';
    return redirect()->route('daftarMuridKelas', $kelas);
  }
  if($tipe == "Pengelola"){
    return redirect()->route('daftarIndikator');
  }
  if($tipe == "Murid"){
    return redirect()->route('rapor.murid', $uid);
  }
}

```

Kode 5.17 Fungsi GetTipe dan ShowPage PenggunaController

Fungsi `getTipe` akan mengambil atribut 'tipe' dari pengguna dan menjalankan `showPage` untuk mengarahkan pengguna ke halaman berdasarkan tipenya.

```
public function getRaporKelas($kelas){
    if($kelas != 'keluar' and $kelas != 'lulus'){
        if($murid = Murid::where('kelas', $kelas)->first()){
            $murids = Murid::where([[ 'kelas', $kelas], [ 'noInduk', '!=', '0' ] ])->get();
            $indicators = Indikator::where('isRapor', true)
            ->orderBy('kode', 'asc')
            ->get();
            return view('rapor/tambah', compact('murid', 'murids', 'indicators'));
        }
        else{
            alert()->error('Kelas ini tidak dapat diakses!');
            return back();
        }
    }
}
```

Kode 5.18 Fungsi `GetRaporKelas` `RaporController`

Fungsi `getRaporKelas` akan menggunakan *input* 4 kelas PAUD Ciluk-Baa yaitu Junior, Senior, RC-A, dan RC-B. Fungsi ini mengecek *input* lalu mengecek apakah ada murid di kelas tersebut untuk menghindari *input* tidak sesuai melalui URL. Setiap kelas memiliki satu murid *placeholder* dengan `noInduk` 0, fungsi ini menyaring murid tersebut terlebih dahulu. Selanjutnya fungsi ini menyaring indikator untuk dipilih melalui atribut `isRapor`. Terakhir fungsi ini menampilkan *view* `rapor/tambah` dengan data yang telah disaring sebelumnya.

```
public function tambah(Request $request){
    $indicators = $request->input('indikator');
    $murids = $request->input('murid');
    foreach ($murids as $murid) {
        $murid = substr($murid,0,strlen($murid));
        foreach ($indicators as $indikator) {
            $indikator = substr($indikator,0,strlen($indikator));
            if(Rapor::where([[ 'noInduk', $murid], [ 'kode', $indikator ] ])->exists()){
                $rapor = Rapor::where([[ 'noInduk', $murid], [ 'kode', $indikator ] ])->first();
                $rapor->frekuensi = $rapor->frekuensi+1;
                $rapor->save();
            }
            else{
                $rapor = new Rapor;
                $rapor->noInduk = $murid;
                $rapor->kode = $indikator;
                $rapor->tanggal = $request->tanggal;
                $rapor->frekuensi = 1;
                $rapor->save();
            }
        }
    }
}
```

Kode 5.19 Fungsi `Tambah` `RaporController`

Fungsi tambah adalah fungsi yang dilakukan setelah konfirmasi *input* untuk menambah rapor. Konfirmasi menampilkan *input* dengan format tambahan untuk mempermudah pengguna membaca data, seperti '[kode] [deskripsi]' untuk indikator dan '[noInduk] [nama]' untuk murid. Karena itulah perlu dilakukan pemisahan terlebih dahulu. Setelah terpisah maka kode akan dimasukkan ke rapor dengan syarat, jika kode sudah ada maka frekuensi akan ditambah satu, dan jika belum ada maka buat baris rapor baru.

```
public function cetakRapor($no){
    $murid = Murid::where('noInduk', $no)->first();
    $rapor = Rapor::with(['indikator'])
    ->where(['noInduk', $no], ['frekuensi', ">", "0"], [])
    ->orderBy('kode', 'asc')
    ->get();
    $pdf = PDF::loadview('rapor/rapor_cetak', compact('murid', 'rapor'));
    return $pdf->download('Rapor '.$murid->nama);
}
```

Kode 5.20 Fungsi CetakRapor RaporController

Fungsi cetakRapor mengambil murid melalui noInduk dan rapor yang memiliki frekuensi lebih dari 0 lalu dengan bantuan *library* DomPDF, halaman rapor cetak dengan data yang diambil diubah menjadi pdf dan meminta pengguna mengunduh pdf tersebut.

```
public function isExist(Request $request){
    $uid = $request->uid;
    $credential = [
        'uid' => $uid,
        'password' => $request->password
    ];
    if ($token = auth('api')->attempt($credential)) {
        return $this -> getTime($uid, $token);
    }
    else return response()->json(['error' => 'Akun tidak ditemukan'], 401);
}
```

Kode 5.21 Fungsi IsExist ApiController

Fungsi isExist untuk API tidak menggunakan validasi karena validasi akan dilakukan oleh game. Jika pengguna tidak ditemukan maka responnya adalah pesan dan kode kesalahan.

Jika ditemukan, maka token JWT akan dibuat dan fungsi `getTipe` dijalankan.

```
public function getTipe($uid, $token){
    $tipe = Pengguna::where('uid', $uid)->value('tipe');
    if($tipe == 'Kelas'){
        $data = [
            'token' => $token,
            'murids' => $this->getAllMurid(),
        ];
        return response()->json($data);
    }
    if($tipe == 'Murid'){
        $pengguna = Pengguna::where('uid', $uid)->first();
        $murid = Murid::where('noInduk', $uid)->first();
        $data = [
            'token' => $token,
            'nama' => $murid->nama,
            'noInduk' => $pengguna->uid,
        ];
        return response()->json($data);
    }
    else return response()->json(['error' => 'Bukan akun kelas atau murid'], 401);
}
```

Kode 5.22 Fungsi GetTipe ApiController

Fungsi `getTipe` hanya menerima dua tipe pengguna yaitu Kelas dan Murid. Jika tipe pengguna adalah Murid maka data berupa token, `noInduk`, dan nama akan dikirim. Jika tipe pengguna adalah Kelas maka data berupa token dan data (`noInduk` dan nama) semua murid aktif yaitu murid yang berada pada kelas utama akan dikirim.

```
public function tambahRapor(Request $request){
    $indikators = $request->input('indikator');
    $noInduk = $request->uid;
    foreach ($indikators as $indikator) {
        if(Rapor::where([[ 'noInduk', $noInduk], ['kode', $indikator]])->exists()){
            $rapor = Rapor::where([[ 'noInduk', $noInduk], ['kode', $indikator]])->first();
            $rapor->frekuensi = $rapor->frekuensi+1;
            $rapor->save();
        }
        else{
            $rapor = new Rapor;
            $rapor->noInduk = $noInduk;
            $rapor->kode = $indikator;
            $rapor->tanggal = date("Y-m-d");
            $rapor->frekuensi = 1;
            $rapor->save();
        }
    }
    return response()->json(['message' => 'Berhasil menambahkan']);
}
```

Kode 5.23 Fungsi TambahRapor ApiController

Fungsi tambahRapor sama seperti fungsi dengan nama yang sama pada RaporController hanya saja tanpa format tambahan.

```
public function load(Request $request){
    $noInduk = $request->uid;
    $gid = $request->gid;
    if(!Hasil::where([[ 'noInduk', $noInduk], [ 'gid', $gid]]->exists())){
        $hasil = new Hasil;
        $hasil->noInduk = $noInduk;
        $hasil->gid = $gid;
        $hasil->level = 0;
        $hasil->score = 0;
        $hasil->save();
    }
    $hasil = Hasil::where([[ 'noInduk', $noInduk], [ 'gid', $gid]])
    ->select('level', 'score')->orderBy('level', 'asc')->get();
    $data = [
        'hasil' => $hasil,
    ];
    return response()->json($data);
}
```

Kode 5.24 Fungsi Load ApiController

Fungsi load menerima permintaan untuk mengirimkan data hasil bermain. Jika tidak ditemukan data yang sesuai dengan permintaan maka akan dibuat baris baru pada tabel hasil bermain. Jika ditemukan maka data akan dikirim setelah diurutkan berdasarkan *level*.

```
public function save(Request $request){
    $noInduk = $request->uid;
    $gid = $request->gid;
    $level = $request->level;
    $score = $request->score;
    if(Hasil::where([[ 'noInduk', $noInduk], [ 'gid', $gid], [ 'level', $level]]->exists())){
        $hasil = Hasil::where([[ 'noInduk', $noInduk], [ 'gid', $gid], [ 'level', $level]]->first();
        $hasil->score = $score;
        $hasil->save();
    }
    else{
        $hasil = new Hasil;
        $hasil->noInduk = $noInduk;
        $hasil->gid = $gid;
        $hasil->level = $level;
        $hasil->score = $score;
        $hasil->save();
    }
    return response()->json(['message' => 'Berhasil menyimpan']);
}
```

Kode 5.25 Fungsi Save ApiController

Fungsi save menerima permintaan menyimpan data hasil bermain dimana jika data sudah ada maka data akan diperbarui dan jika data belum ada maka akan dibuat baris baru.

5.2.1.5 Implementasi Route

Route adalah kode yang digunakan untuk memetakan pemanggilan fungsi pada *controller*. *Route* untuk web terletak pada 'routes/web'.

```
Route::get('/', function () {return view('masuk');});
Route::post('/', 'PenggunaController@isExist')->name('masuk');
Route::get('/', 'PenggunaController@exit')->name('keluar');

Route::middleware('auth:web')->group(function () {
    Route::middleware('can:isAdmin')->group(function () { ...
    });
    Route::middleware('can:isTU')->group(function () { ...
    });

    Route::middleware('can:isKelas')->group(function () {
        Route::get('/daftar-murid/{kelas}', 'MuridController@getMuridKelas')->name('daftarMuridK');
        Route::get('/rapor/murid/{no}', 'RaporController@getRapor')->name('rapor');
        Route::get('/rapor/cetak/murid/{no}', 'RaporController@cetakRapor')->name('cetakRapor');
        Route::get('/rapor-murid/{kelas}', 'RaporController@getRaporKelas')->name('raporKelas');
        Route::post('/validRapor', 'RaporController@valid')->name('validRapor');
        Route::post('/menambahRapor', 'RaporController@tambah')->name('tambah');
```

Kode 5.26 Route untuk Web

Route dengan tipe GET umumnya digunakan untuk fungsi yang memproses data dari sistem ke pengguna, sedangkan *route* dengan tipe POST digunakan untuk fungsi yang memproses data yang dikirim oleh pengguna ke sistem. *Route* dapat diberi nama untuk memudahkan pemanggilan.

```
Gate::define('isKelas', function($pengguna) {
    return $pengguna->tipe == 'Kelas';
});

Gate::define('isMurid', function($pengguna) {
    return $pengguna->tipe == 'Murid';
```

Kode 5.27 Gate Kelas dan Murid

Route web memiliki beberapa *middleware*, salah satunya adalah 'can:isKelas'. *Middleware gate* ini digunakan untuk mencegah

pengguna yang tidak sesuai mengakses *route*. *Gate* dapat diatur pada ‘App/Providers/AuthServiceProvider’.

```
'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'pengguna',
    ],

    'api' => [
        'driver' => 'jwt',
        'provider' => 'pengguna',
        'hash' => false,
    ],
]
```

Kode 5.28 Konfigurasi Auth

Selain *gate*, terdapat juga *middleware auth* pada *route*. Ini digunakan setelah mendefinisikan *file* pada ‘config/auth’. Konfigurasi *auth* menampilkan jika otorisasi tipe *web* menggunakan *session* sedangkan tipe API menggunakan *jwt*. Keduanya menggunakan *model* pengguna sebagai acuan otorisasi.

```
Route::post('/masuk', 'ApiController@isExist');

Route::middleware('auth:api')->group(function () {
    Route::post('/load', 'ApiController@load');
    Route::post('/save', 'ApiController@save');
    Route::post('/rapor', 'ApiController@tambahRapor');
});
```

Kode 5.29 Route untuk API

Route untuk API terletak pada ‘routes/api’. Untuk menggunakan API, dapat melalui ‘[nama-web]/api/[route]’. Penggunaannya harus melalui jaringan sehingga sistem rapor di-*hosting* dengan alamat <https://ciluk-baa.id>.

Laravel 7.x sudah mengatasi masalah Cross-Origin Resource Sharing (CORS) secara bawaan.

5.2.2 Produksi (Aplikasi Permainan)

Produksi *game* dilakukan menggunakan Construct 3 sehingga untuk memulai perlu dibuat proyek baru dengan pengaturan layar 16:9 (landscape).

5.2.2.1 Implementasi Login

Semua obyek diletakkan sesuai *mockup login* pada *layout login* dan gambar obyek disesuaikan dengan aset yang dibuat untuk mengimplementasikan tampilan *login*.



Gambar 5.10 Implementasi Tampilan Login

```

🌐 Global constant number GameID = 1
🌐 Global constant string URL = https://ciluk-baa.id/api/
🌐 Global string Token =
🌐 Global string UserID =
🌐 Global string TipeAkun =
🌐 Global string Nama =

```

Kode 5.30 Global Variable Event Login

Global variable dapat diakses di seluruh *game* sehingga dapat digunakan untuk menyimpan variabel yang sering digunakan.

On function **Masuk** + Add action + Add

Local string **id** =
Local string **password** =

UIText	Pick instance with UID 24	UIText	Set visibility Visible
		UIText	Typewriter text "Mahon menunggu..." over 1 seconds
		+ Add action	+ Add
Textinp...	Pick instance with UID 5	System	Set id to <code>URLEncode(TextInput.Text)</code>
		+ Add action	+ Add
Textinp...	Pick instance with UID 22	System	Set password to <code>URLEncode(TextInput.Text)</code>
		+ Add action	+ Add
System	id = ""	+ Add action	+ Add
System	OR password = ""	+ Add action	+ Add
UIText	Pick instance with UID 24	UIText	Set text to "Identitas dan kata sandi tidak boleh kosong"
		UIText	Set visibility Visible
System	id = ""	AJAX	Send " uid = " <code>&id&</code> "& password = " <code>&password</code> " to URL URL& " <code>masuk</code> " (method "POST", tag " <code>U</code>)
System	password = ""	System	Wait for previous actions to complete
		JSON	Parse JSON string AJAX.LastData
		+ Add action	+ Add
JSON	Has key " error "	+ Add action	+ Add
JSON	Has key " token "	System	Set Token to <code>JSON.Get("token")</code>
		+ Add action	+ Add
JSON	Has key " murids "	System	Set TipeAkun to "Kelas"
		ArrayM...	Set size to <code>(JSON.ArraySize("murids"), 3, 1)</code>
		+ Add action	+ Add
JSON	For each entry in " murids "	ArrayM...	Set value at <code>(int/JSON.CurrentKey, 0)</code> to <code>JSON.Get("murids."&JSON.CurrentKey&".nama")</code>
		ArrayM...	Set value at <code>(int/JSON.CurrentKey, 1)</code> to <code>JSON.Get("murids."&JSON.CurrentKey&".noInduk")</code>
		ArrayM...	Set value at <code>(int/JSON.CurrentKey, 2)</code> to <code>JSON.Get("murids."&JSON.CurrentKey&".kelas")</code>
		+ Add action	+ Add
JSON	Has key " murids "	System	Set TipeAkun to "Murid"
		System	Set UserID to <code>JSON.Get("noInduk")</code>
		System	Set Nama to <code>JSON.Get("nama")</code>

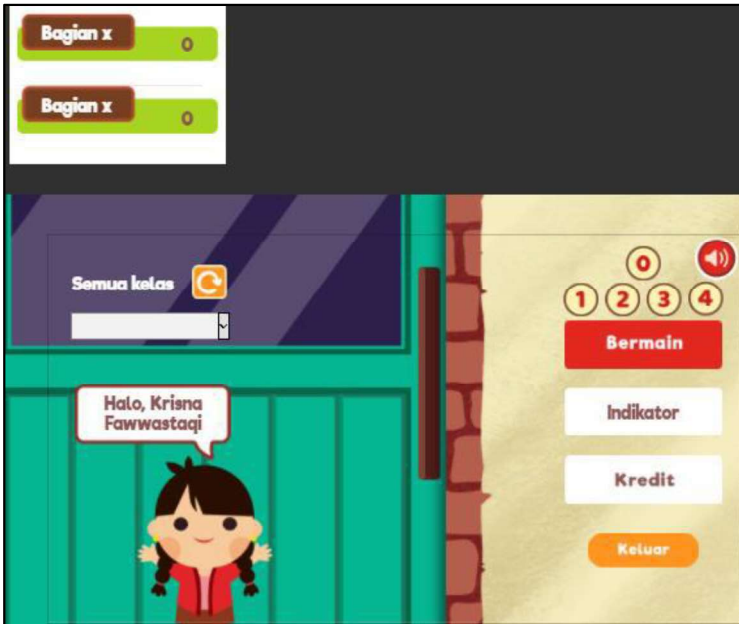
Kode 5.31 Fungsi Masuk

Fungsi Masuk akan dijalankan saat pemain menekan tombol masuk. Identitas dan kata sandi yang diketik pada obyek `TextInput` akan dimasukkan ke variabel lokal. Jika variabel ini kosong maka pesan kesalahan akan tampil. Jika variabel ini ada isinya maka obyek `AJAX` akan memanggil API masuk, hasilnya akan di parse `JSON`.

Token yang dikirimkan dimasukkan ke variabel global untuk transaksi selanjutnya. Jika terdapat key bernama 'murids' maka seluruh isi key tersebut dimasukkan ke obyek `ArrayMurid`. Selanjutnya variabel lain diisi sesuai data yang diterima dan tampilan menu dibuka.

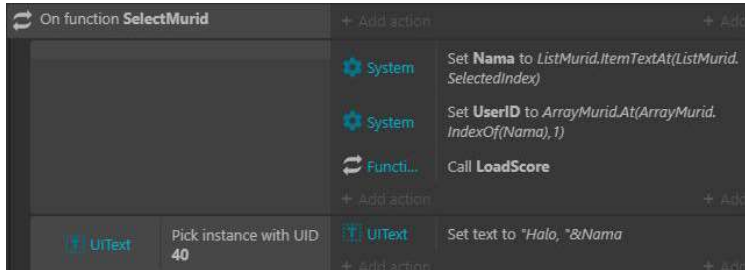
5.2.2.2 Implementasi Menu

Semua obyek diletakkan sesuai *mockup menu* pada *layout menu* dan gambar obyek disesuaikan dengan aset yang dibuat untuk mengimplementasikan tampilan *menu*.



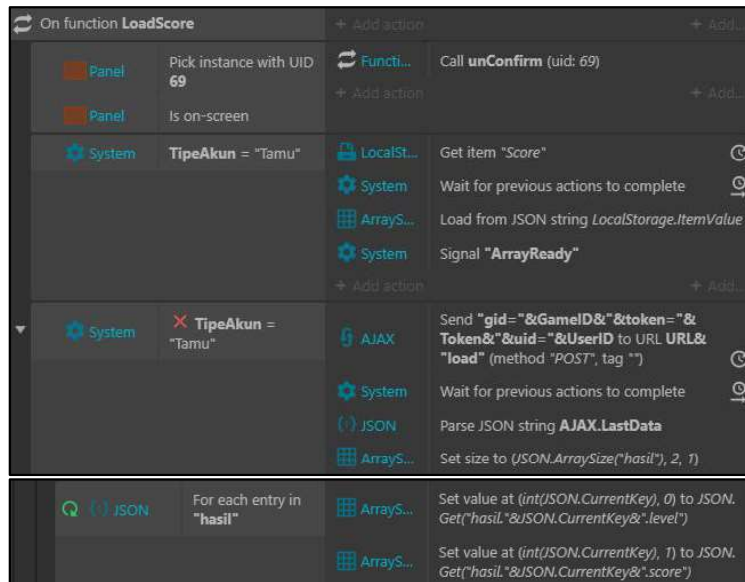
Gambar 5.11 Implementasi Tampilan Menu

Pada tampilan Menu terdapat panel berwarna putih yang berada di luar layout. Panel ini berisi tombol pilihan bagian untuk dimainkan.



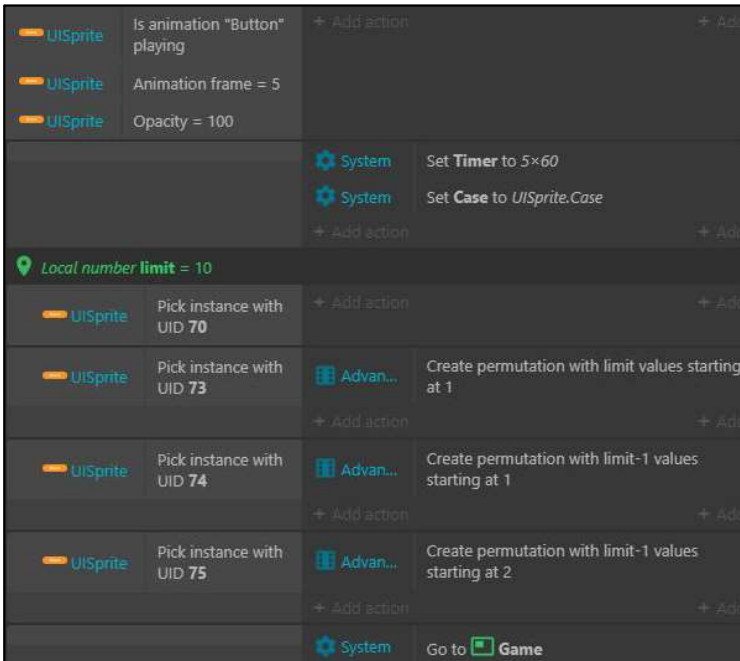
Kode 5.32 Fungsi SelectMurid

Fungsi SelectMurid dijalankan saat pemain memilih murid dari obyek ListMurid. Fungsi ini akan mengisi variabel UserID dengan nomor induk murid yang dipilih dan memanggil fungsi LoadScore.



Kode 5.33 Fungsi LoadScore

Fungsi LoadScore akan menutup panel berisi bagian jika panel tersebut terlihat di layar. Jika pengguna bertipe tamu maka obyek LocalStorage akan mengambil simpanan skor lalu memasukkan skor ke obyek ArrayScore. Jika pengguna bukan tamu maka obyek AJAX akan memanggil API load yang hasilnya di parse dan disimpan pada ArrayScore.

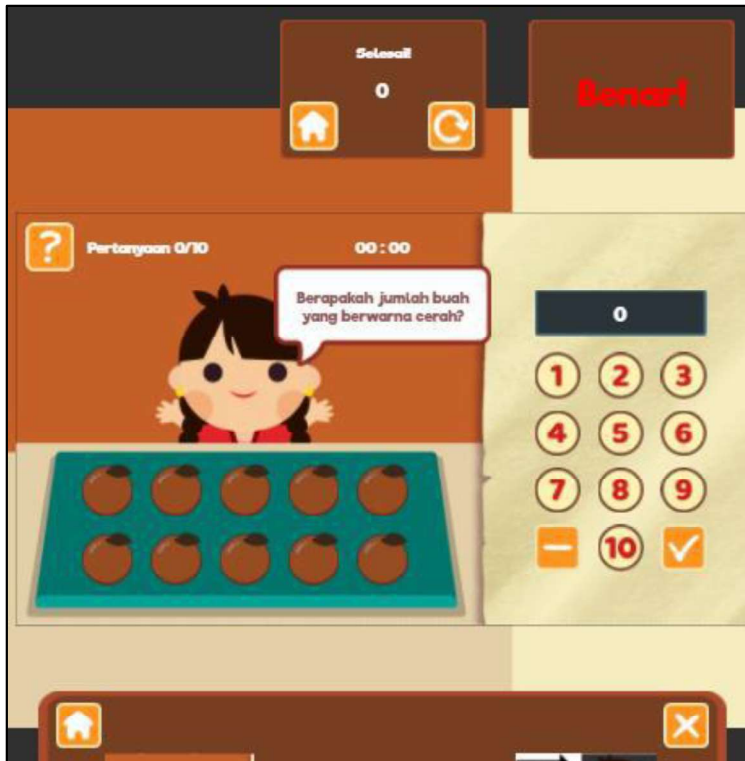


Kode 5.34 Tombol Bagian

Saat tombol pada panel bagian ditekan maka hanya bagian yang tidak transparan yang akan merespon. Variabel case akan diubah sesuai tombol bagian yang dipilih. Obyek AdvancedRandom akan membuat permutasi dengan batasan yang diatur sesuai bagian dan variabel 'limit'. Selanjutnya tampilan *game* akan dibuka dengan membawa data permutasi yang dibuat.

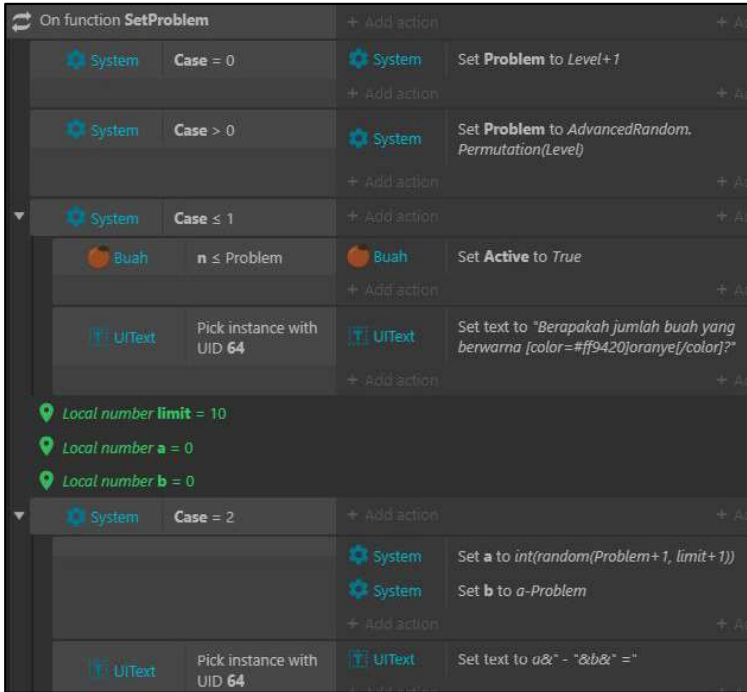
5.2.2.3 Implementasi Game

Semua obyek diletakkan sesuai *mockup game* pada *layout game* dan gambar obyek disesuaikan dengan aset yang dibuat untuk mengimplementasikan tampilan *game*.



Gambar 5.12 Implementasi Tampilan Game Tipe 1

Pada tampilan Game *panel* yang berada di luar *layout* dari atas kiri secara berturut-turut digunakan untuk: menampilkan skor dan pilihan saat permainan berakhir, menampilkan *feedback* setelah pemain menjawab, dan penjelasan cara bermain.

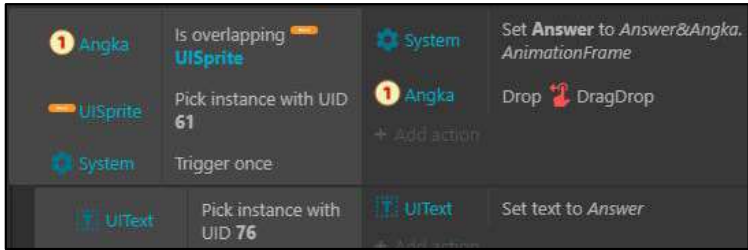


Kode 5.35 Fungsi SetProblem

Fungsi SetProblem berfungsi untuk mendefinisikan jawaban di satu soal. Fungsi ini dipanggil saat bagian permainan dimulai dan setiap soal terjawab. Pada bagian di atas 1 jawaban didapatkan dari permutasi yang dibuat sebelum memasuki tampilan game.

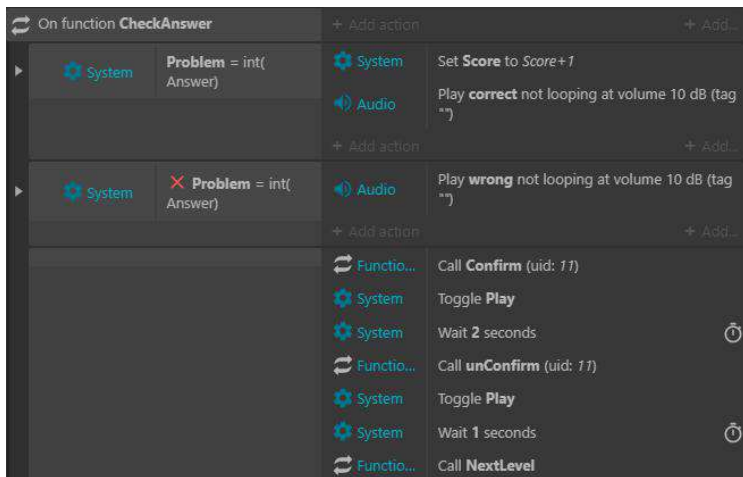
Untuk bagian 1 dan 2, buah yang memiliki urutan kurang dari sama dengan jawaban akan aktif dan berwarna oranye. Pertanyaan akan diatur menjadi jumlah buah yang berwarna oranye.

Untuk bagian 3 terdapat variabel bantuan untuk menentukan soal. Pertanyaan akan diatur menjadi operasi pengurangan berdasarkan soal yang ditentukan. Bagian 4 juga demikian namun, pertanyaan berupa operasi penjumlahan.



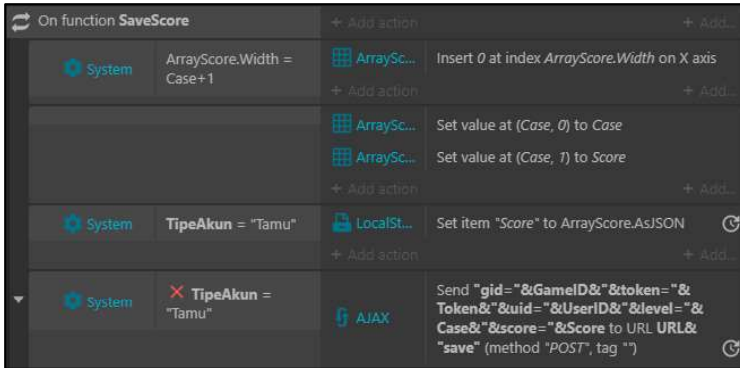
Kode 5.36 Angka dan Kotak Jawaban

Saat angka bersentuhan dengan kotak jawaban maka variabel ‘Answer’ akan disambung dengan frame dari angka dimana frame dari sprite angka sesuai dengan angka yang ditampilkan.



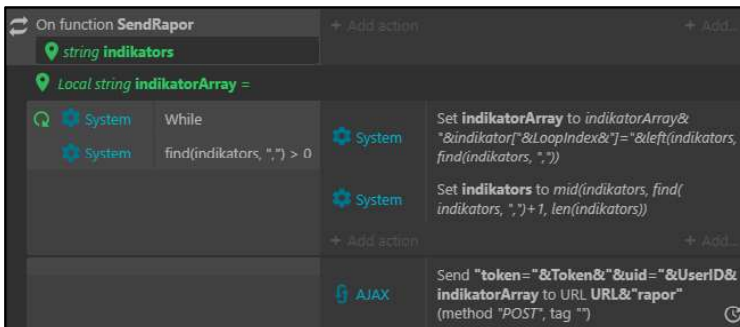
Kode 5.37 Fungsi CheckAnswer

Fungsi CheckAnswer dijalankan saat pemain menekan tombol konfirmasi (centang). Jawaban pemain akan dicocokkan dengan jawaban yang benar dan diberi *feedback* yang sesuai. Setelah itu fungsi NextLevel dipanggil untuk menuju ke soal berikutnya atau mengakhiri permainan jika sudah berada pada soal ke 10.



Kode 5.38 Fungsi SaveScore

Fungsi SaveScore dijalankan saat permainan berakhir. Jika tipe pengguna adalah tamu maka skor disimpan ke ArrayScore lalu obyek LocalStorage akan menyimpan skor secara lokal. Jika tipe pengguna bukan tamu maka API save akan dipanggil.



Kode 5.39 Fungsi SendRapor

Fungsi SendRapor dijalankan setelah skor disimpan. API rapor membutuhkan masukan kode indikator dengan format “indikator[x]=x.x.x.x.”. Fungsi ini menyesuaikan indikator yang terpenuhi sebagai parameter sesuai format tersebut dan memanggil API rapor.

System	Case = 0	+ Add action	+ Add
System	Score ≥ 10	Funcio... Call SendRapor (indikator: "3.3.2.2,")	+ Add
System	Case = 1	+ Add action	+ Add
System	Score ≥ 10	Funcio... Call SendRapor (indikator: "3.3.2.3,")	+ Add
System	Case = 2	+ Add action	+ Add
System	Score ≥ 7	Funcio... Call SendRapor (indikator: "3.3.4.1,")	+ Add
System	Case = 3	+ Add action	+ Add
System	Score ≥ 7	Funcio... Call SendRapor (indikator: "3.3.4.2, 3.3.5.2, 3.3.5.3,")	+ Add

Kode 5.40 Indikator setiap Bagian

Setelah skor disimpan dan jika skor sudah melebihi batas minimal skor di satu bagian maka fungsi SendRapor akan dipanggil dengan parameter kode indikator yang sesuai.

Berikut ini implementasi tipe *game* yang lainnya.



Gambar 5.13 Implementasi Game Tipe 2 Bagian 3



Gambar 5.14 Implementasi Game Tipe 3 Bagian 4



Gambar 5.15 Implementasi Game Tipe 4

5.3 Lingkungan Pengujian

Pengujian sistem rapor dan aplikasi permainan menggunakan komputer dan perangkat dengan spesifikasi sebagai berikut.

Tabel 5. 4 Spesifikasi Komputer Pengujian

Tipe	ASUS X555LF
Processor	Intel® Core™ i5-5200U CPU @2.2GHz

Memory (RAM)	8 GB
Sistem Operasi	Windows 10 Education 64Bit
Graphic Card	Nvidia GeForce 930M

Tabel 5. 5 Spesifikasi Perangkat Android Pengujian

Tipe	Samsung Galaxy Tab A (2016) Pen
Processor	Octa-core 1.6GHz
Memory (RAM)	3 GB
Sistem Operasi	Android 8.1.0

5.4 Fase 4

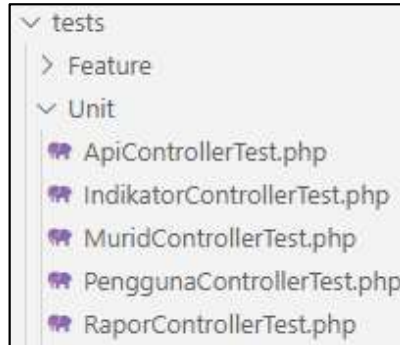
Tahapan ini adalah pengujian sistem dan aplikasi permainan yang dilakukan oleh pengembang.

5.4.1 Pengujian Sistem (Sistem Rapor)

Pengujian terhadap sistem.

5.4.1.1 Unit Test

Laravel menyediakan fitur *unit test* secara bawaan, dengan perintah membuat *test* maka *file test* akan dibuat pada 'tests/Unit'. Unit test ditulis untuk menguji kesesuaian *input* dan *output* pada fungsi yang ada di *controller*. File unit test dibuat sejumlah *controller* yang ada dan diberi nama sesuai *controller* yang diuji seperti pada Gambar 5.13.



Gambar 5.16 File Unit Test

Berikut ini merupakan potongan kode dari beberapa *unit test*.

```
class PenggunaControllerTest extends TestCase
{
    use RefreshDatabase;

    public function test_halaman_masuk(){
        $response = $this->get('/');

        $response->assertViewIs('masuk');
    }
}
```

Kode 5.41 Test Halaman Masuk PenggunaControllerTest

Pada awal setiap *unit test* terdapat `RefreshDatabase` yang digunakan supaya *database* tidak dipengaruhi *file unit test* lain. Test halaman masuk menguji jika alamat `'/'` dibuka maka *view* yang ditampilkan harus `'masuk'`.

```

public function test_masuk_data_tidak_lengkap(){
    $response = $this->post(route('masuk'), [
        'uid' => $uid = '111111',
        'password' => '',
    ]);

    $response->assertSessionHasErrors('password');
    $response->assertSessionHasInput('uid', $value = $uid);

    $response = $this->post(route('masuk'), [
        'uid' => '',
        'password' => 'asdfg',
    ]);

    $response->assertSessionHasErrors('uid');
}

```

Kode 5.42 Test Masuk Data Tidak Lengkap PenggunaControllerTest

Test masuk data tidak lengkap menguji jika rute ‘masuk’ dilakukan dengan *input* yang tidak lengkap maka akan terdapat *error* pada *input* yang kosong.

```

public function test_tipe_pengguna_kelas(){
    $this->seed(PenggunaSeeder::class);

    $response = $this->post(route('masuk'), [
        'uid' => 'kelas',
        'password' => 'secret',
    ]);

    $response->assertRedirect(route('daftarMuridKelas', 'junior'));
}

```

Kode 5.43 Test Tipe Pengguna Kelas PenggunaControllerTest

Test tipe pengguna kelas menjalankan PenggunaSeeder terlebih dahulu lalu masuk menggunakan identitas pengguna Kelas. Tes menguji apakah pengguna diarahkan ke rute daftar murid kelas.

```

public function test_tambah_rapor(){
    $pengguna = factory(Pengguna::class)->create([
        'tipe' => 'Kelas',
    ]);
    $this->seed(MuridSeeder::class);
    $this->seed(IndikatorSeeder::class);
    $response = $this->actingAs($pengguna)
->post(route('tambah'), [
        'tanggal' => '02-05-2020',
        'indikator' => ['1.2.3.4.) Indikator ini 1.2.3.4.'],
        'murid' => ['202001250) Andhika Pramesti'],
    ]);
    $response->assertSuccessful();
    $no = '202001250';
    $response = $this->actingAs($pengguna)
->get(route('rapor', $no));
    $response->assertSee('<td>1.2.3.4.</td>', false);
    $response->assertSee('<td>1</td>', false);
}

```

Kode 5.44 Test Tambah Rapor RaporControllerTest

Test tambah rapor menginisiasi pengguna dengan tipe Kelas lalu seeder Murid dan Indikator. Tes melakukan rute ‘tambah’ lalu menguji apakah sukses. Selanjutnya tes membuka halaman rapor siswa tersebut dan menguji apakah terdapat sel tabel berisi rapor yang baru saja ditambahkan.

```

public function test_semua_murid(){
    $this->seed(PenggunaSeeder::class);
    $this->seed(MuridSeeder::class);
    $response = $this->json('POST', 'api/masuk', [
        'uid' => 'kelas',
        'password' => 'secret',
    ]);
    $response->assertJson(['murids'=>array()]);
    $response->assertJsonFragment(['nama'=>'Uji Coba Junior']);
}

```

Kode 5.45 Test Semua Murid ApiControllerTest

Test semua murid menginisiasi seeder Pengguna dan Murid lalu memanggil API masuk dengan pengguna Kelas. Tes ini menguji apakah terdapat array JSON ‘murids’ dan sebagian JSON dengan nama murid yang ada pada seeder.

```

public function test_tambah_rapor(){
    $this->seed(PenggunaSeeder::class);
    $this->seed(MuridSeeder::class);
    $response = $this->json('POST', 'api/masuk', [
        'uid' => 'kelas',
        'password' => 'secret',
    ]);
    $response = $this->json('POST', 'api/rapor', [
        'uid' => '123456789',
        'indikator' => ['1.2.3.4.', '1.2.3.5.'],
    ]);
    $response->assertJson(['message' => 'Berhasil menambahkan']);
}

```

Kode 5.46 Test Tambah Rapor ApiControllerTest

Test tambah rapor memanggil API rapor lalu menguji apakah ada pesan berhasil.

```

public function test_load_hasil(){
    $this->seed(PenggunaSeeder::class);
    $this->seed(MuridSeeder::class);
    $this->seed(HasilSeeder::class);
    $response = $this->json('POST', 'api/masuk', [
        'uid' => 'kelas',
        'password' => 'secret',
    ]);
    $response = $this->json('POST', 'api/load', [
        'uid' => '123456789',
        'gid' => '1',
    ]);
    $response->assertJson(['hasil'=>array()]);
    $response->assertJsonFragment(['level'=>'1', 'score'=>'0']);
}

```

Kode 5.47 Test Load Hasil ApiControllerTest

Test load hasil menginisiasi tambahan seeder Hasil lalu memanggil API load dan menguji apakah terdapat array JSON 'hasil' dan sebagian JSON dengan hasil yang ada pada *seeder*.

```

public function test_save_hasil(){
    $this->seed(PenggunaSeeder::class);
    $this->seed(MuridSeeder::class);
    $response = $this->json('POST', 'api/masuk', [
        'uid' => 'kelas',
        'password' => 'secret',
    ]);
    $response = $this->json('POST', 'api/save', [
        'uid' => '202001250',
        'gid' => '1',
        'level' => '2',
        'score' => '10',
    ]);
    $response->assertJson(['message' => 'Berhasil menyimpan']);
}

```

Kode 5.48 Test Save Hasil ApiControllerTest

Test save hasil memanggil API save lalu menguji apakah ada pesan berhasil.

Selanjutnya dengan menggunakan perintah ‘php artisan test’ didapatkan hasil sebagai berikut.

```

PASS Unit\ApiControllerTest
✓ masuk akun tidak ada
✓ masuk akun salah tipe
✓ masuk kelas
✓ semua murid
✓ masuk murid
✓ tambah rapor
✓ load hasil
✓ save hasil

PASS Unit\IndikatorControllerTest
✓ halaman indikator
✓ tambah indikator
✓ hapus indikator
✓ ubah indikator

```

<pre>PASS Unit\MuridControllerTest ✓ halaman daftar murid ✓ halaman profil murid ✓ halaman tambah murid ✓ tambah murid data tidak lengkap ✓ tambah murid sudah ada ✓ tambah murid valid ✓ tambah murid daftar ✓ halaman edit ✓ ubah murid update ✓ hapus murid ✓ halaman daftar murid kelas ✓ keluarkan murid ✓ naikkan murid</pre>
<pre>PASS Unit\PenggunaControllerTest ✓ halaman masuk ✓ masuk data tidak lengkap ✓ masuk salah uid atau password ✓ masuk terautentikasi ✓ tipe pengguna admin ✓ tipe pengguna tatusaha ✓ tipe pengguna kelas ✓ tipe pengguna pengelola ✓ tipe pengguna murid ✓ pengguna keluar ✓ halaman admin ✓ update password</pre>
<pre>PASS Unit\RaporControllerTest ✓ halaman rapor ✓ cetak rapor ✓ halaman tambah rapor ✓ tambah rapor data tidak lengkap ✓ tambah rapor valid ✓ tambah rapor ✓ kurang rapor ✓ halaman rapor murid ✓ cetak rapor murid</pre>

Gambar 5.17 Hasil Unit Test

Semua pengujian pada setiap *unit test* terpenuhi.

5.4.1.2 System Test

Pengujian kebutuhan fungsional (kode FR) sistem rapor dilakukan berdasarkan *use case* (kode UC) *description* dengan kondisi sistem telah *online* dengan hasil sebagai berikut.

Tabel 5. 6 Hasil System Test

Kode FR	Kode UC	Use Case	Ekspektasi	Hasil
	UC01	Masuk	Pengguna diarahkan ke halaman sesuai tipe	OK
	UC02	Keluar	Pengguna diarahkan ke halaman masuk	OK
FR01	UC03	Melihat data murid	Sistem menampilkan halaman profil murid	OK
	UC04	Mengisi data murid	Data murid tersimpan	OK
	UC05	Mengubah data murid	Data murid diubah	OK
	UC06	Menghapus data murid	Data murid terhapus	OK
FR02	UC07	Melihat rapor	Sistem menampilkan halaman rapor murid	OK
FR03	UC08	Mencetak rapor	Sistem meminta pengguna mengunduh pdf	OK
FR04	UC09	Mengisi rapor	Data rapor ditambahkan	OK
	UC10	Mengubah rapor	Frekuensi rapor berkurang 1 dan hilang jika 0	OK

FR05	UC11	Menaikkan murid	Kelas murid berubah ke jenjang berikutnya	OK
	UC12	Mengeluarkan murid	Kelas murid berubah menjadi keluar	OK
FR06	UC13	Melihat indikator	Sistem menampilkan halaman daftar indikator	OK
	UC14	Menambahkan indikator	Data indikator ditambahkan	OK
	UC15	Mengubah indikator	Data indikator diubah	OK
	UC16	Menghapus indikator	Data indikator terhapus	OK
FR07	UC17	Mengubah password	Data password pada pengguna berubah	OK
FR08	UC18	Masuk melalui game	Sistem mengirimkan data seluruh murid	OK
	UC19	Mengisi rapor melalui game	Data rapor ditambahkan	OK
FR09	UC20	Menyimpan hasil bermain	Data hasil bermain ditambahkan	OK
	UC21	Mengirim hasil bermain	Sistem mengirimkan data hasil bermain	OK

Use case dengan kode UC01 sampai UC17 diuji dengan menggunakan *web browser*. Sisanya diuji menggunakan Postman dan fitur *debug* Construct 3.

5.4.2 Pengujian Permainan (Aplikasi Permainan)

Pengujian terhadap game.

5.4.2.1 Functional Test

Pengujian berdasarkan kebutuhan fungsional dengan cara memainkan game pada perangkat yang telah disiapkan.

Tabel 5. 7 Hasil Functional Test

Kode	Kebutuhan	Ekspektasi	Hasil
FR001	Masuk	Tampilan menu terbuka	OK
		Terdapat fitur pilih murid untuk pengguna kelas	OK
FR002	Memilih murid	Murid dapat dipilih oleh pengguna kelas	OK
FR003	Menampilkan nama	Nama pengguna tampil	OK
		Nama murid tampil sesuai yang dipilih oleh pengguna kelas	OK
FR004	Mengelola bagian	Bagian selanjutnya terbuka jika skor minimal bagian ini tercapai	OK
FR005	Membuat pertanyaan	Soal muncul sesuai ketentuan bagian	OK
FR006	Memberi feedback	Muncul feedback positif jika jawaban benar dan negatif jika jawaban salah	OK
FR007	Mengisi rapor	Jika skor minimal tercapai, rapor bagian itu ditambahkan	OK
FR008	Mengelola skor	Data skor tersimpan setelah bermain secara lokal maupun online	OK
		Data skor dapat dimuat kembali secara lokal maupun online	OK

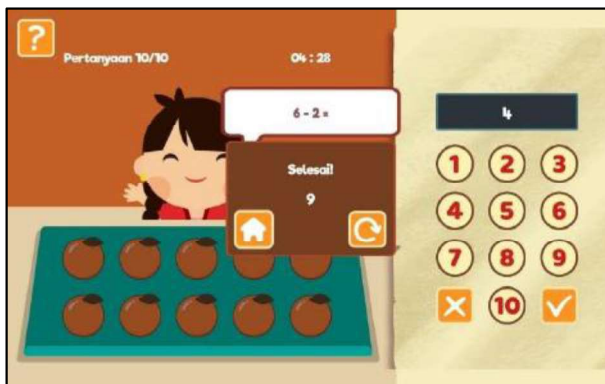
FR009	Mengakhiri bagian	Panel game over muncul jika waktu habis atau semua soal selesai	OK
FR010	Memilih tipe	Tipe permainan muncul sesuai pilihan	OK

Pengujian FR007 dilakukan dengan membandingkan halaman rapor murid yang diuji pada sistem rapor sebelum dan sesudah memainkan game.

Berikut ini beberapa hasil tangkap layar saat uji fungsionalitas.



Gambar 5.20 Pengujian Mengelola Bagian pada Tampilan Menu



Gambar 5.21 Pengujian Mengakhiri Bagian pada Tampilan Game

BAB VI HASIL DAN PEMBAHASAN

Pada bagian ini akan menjelaskan mengenai hasil dan pembahasan dari pengembangan sistem rapor dan aplikasi permainan. Bagian ini melakukan fase 5 yang dijelaskan pada metodologi.

6.1 Fase 5

Setelah pengujian sistem rapor dan aplikasi permainan dilakukan oleh pengembang dan tidak ditemukan *bug* yang maka pengujian dilakukan oleh pihak PAUD Ciluk-Baa untuk menentukan penerimaan pengguna.

User Acceptance Test (UAT) dilakukan dengan menggunakan media google form untuk merekam jawaban. Pernyataan yang diajukan pada *form* akan mengukur kriteria efektifitas dan kepuasan pengguna[17].

Pengukuran dilakukan dengan skala likert dengan angka 1-5 dengan nilai interval sebagai berikut.

Tabel 6.1 Nilai Interval

Nilai	Keterangan
1,00 – 1,79	Sangat Tidak Setuju
1,80 – 2,59	Tidak Setuju
2,60 – 3,39	Ragu-ragu
3,40 – 4,19	Setuju
4,20 – 5,00	Sangat Setuju

Nilai interval diperoleh dari perhitungan ((nilai tertinggi – nilai terendah)/banyak kelas). Sehingga didapatkan $((5-1)/5) = 0,8$ sebagai interval.

6.1.1 Evaluasi (Sistem Rapor)

Form dibuat sebanyak tipe pengguna dengan *section* sejumlah *use case* pada tipe pengguna tersebut. Setiap *section* memiliki skenario berdasarkan alur dasar *use case*, pengguna dituntut atau diperlihatkan bagaimana cara menyelesaikan skenario tersebut.

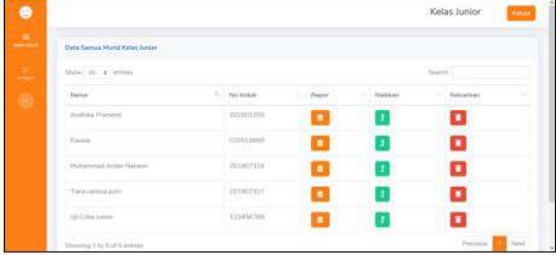

Pada bagian bawah skenario terdapat pernyataan berikut.

Tabel 6.2 Pernyataan UAT Sistem Rapor

Pernyataan	Kriteria
Saya merasa sistem dapat memenuhi kebutuhan kegiatan [Nama Skenario]	Efektifitas
Saya tidak merasa kebingungan atau kesulitan saat melakukan kegiatan [Nama Skenario]	Kepuasan

Berikut ini merupakan contoh skenario yang dimaksud.

Mengisi rapor

Kondisi mulai	Halaman daftar murid kelas <div style="border: 1px solid gray; padding: 5px; margin-top: 5px;">  </div>
Langkah yang perlu dilakukan	Tekan tombol Isi Rapor  <div style="border: 1px solid gray; padding: 5px; margin-top: 10px; width: fit-content;"> KELAS Junior Senior RC-A RC-B </div> <p style="margin-top: 10px;">Pilih kelas dari murid yang dibuat sebelumnya Pilih tanggal hari ini pada tanggal kegiatan Pilih 3 indikator pada indikator terpenuhi Centang murid-murid yang dibuat sebelumnya</p>



Gambar 6.1 UAT Skenario Mengisi Rapor

Saya merasa sistem dapat memenuhi kebutuhan kegiatan Mengisi rapor

1 2 3 4 5

Sangat tidak setuju Sangat setuju

...

Saya tidak merasa kebingungan atau kesulitan saat melakukan kegiatan Mengisi rapor

1 2 3 4 5

Sangat tidak setuju Sangat setuju

Gambar 6.2 Pernyataan Skenario Mengisi Rapor

Berikut ini merupakan tabel rangkuman evaluasi pengujian sistem rapor yang dilakukan terhadap: 3 guru sebagai pengguna Tata Usaha dan Kelas, 2 pengelola sebagai pengguna Pengelola dan Admin, dan 5 wali murid sebagai pengguna Murid.

Tabel 6.3 Hasil UAT Sistem Rapor

Kode UC	Use Case	E ^(*)	S ^(*)	Mean
UC01	Masuk	4,53	4,80	4,67
UC02	Keluar	5,00	4,80	4,90
UC03	Melihat data murid	5,00	4,67	4,83
UC04	Mengisi data murid	5,00	5,00	5,00
UC05	Mengubah data murid	5,00	5,00	5,00
UC06	Menghapus data murid	5,00	5,00	5,00
UC07	Melihat rapor	4,88	4,75	4,81
UC08	Mencetak rapor	4,88	4,63	4,75
UC09	Mengisi rapor	5,00	4,67	4,83
UC10	Mengubah rapor	5,00	5,00	5,00
UC11	Menaikkan murid	4,33	4,67	4,50
UC12	Mengeluarkan murid	4,00	4,33	4,17
UC13	Melihat indikator	4,50	4,50	4,50
UC14	Menambahkan indikator	4,50	4,50	4,50
UC15	Mengubah indikator	5,00	4,50	4,75
UC16	Menghapus indikator	5,00	5,00	5,00
UC17	Mengubah password	4,50	4,00	4,25
Mean		4,77	4,69	

***Keterangan:**

E: Effectivity (Efektifitas)

S: Satisfaction (Kepuasan)

Pengguna merasa “**Sangat Setuju**” dengan pernyataan pada kriteria efektifitas dan kepuasan.

6.1.2 Beta (Aplikasi Permainan)

Form dibuat sebanyak 2 jenis yaitu untuk guru dan wali murid. Beta test aplikasi permainan yang dilakukan oleh guru akan berfokus pada efektifitas setiap bagian dengan indikator dan skor minimal yang terkait. Beta test ini dilakukan dengan memandu guru memainkan semua bagian lalu menjawab pernyataan berikut.

Tabel 6.4 Hasil UAT Aplikasi Permainan ke Guru

Pernyataan	Mean
Anak akan diberi 10 pertanyaan mengenai lambang bilangan yang muncul secara berurutan mulai dari 1-10. Anak menjawab dengan memasangkan jumlah benda dan lambang bilangan. Anak akan mendapat skor 1 setiap kali menjawab pertanyaan dengan benar. Jika anak mendapat skor minimal 10, maka anak akan dianggap memenuhi indikator 3.3.2.2. Menyebutkan bilangan 1-10 berurutan sambil menghitung benda.	4,67
Anak akan diberi 10 pertanyaan mengenai lambang bilangan yang muncul secara acak mulai dari 1-10. Anak menjawab dengan memasangkan jumlah benda dan lambang bilangan. Anak akan mendapat skor 1 setiap kali menjawab pertanyaan dengan benar. Jika anak mendapat skor minimal 10, maka anak akan dianggap memenuhi indikator 3.3.2.3. Memasangkan lambang bilangan dengan jumlah benda 1-10.	4,67
Anak akan diberi 10 pertanyaan cerita mengenai operasi pengurangan. Anak menjawab dengan memasangkan jumlah benda dan lambang bilangan hasil operasi. Anak akan mendapat skor 1 setiap kali menjawab pertanyaan dengan benar. Jika anak mendapat skor minimal 7, maka anak akan dianggap memenuhi indikator 3.3.4.1. Menghitung dan menyebutkan hasil pengurangan.	4,33
Anak akan diberi 10 pertanyaan cerita mengenai operasi penjumlahan. Anak menjawab dengan memasangkan jumlah benda dan lambang bilangan hasil operasi. Anak akan mendapat skor 1 setiap kali menjawab pertanyaan dengan benar. Jika anak mendapat skor minimal 7, maka anak akan dianggap memenuhi indikator 3.3.4.2. Menghitung dan menyebutkan hasil penjumlahan	4,67

Anak akan diberi batasan waktu 5 menit untuk menjawab 10 pertanyaan di setiap bagian. Batasan waktu ini diperlukan saat pengujian untuk membatasi giliran setiap anak.	4,67
Terdapat 4 tipe permainan yaitu: menarik angka ke kotak jawaban, menarik buah ke kotak jawaban, menggeser kotak jawaban untuk menangkap buah, dan menyeimbangkan buah pada kotak jawaban dengan angka. Keempatnya memiliki tingkat kesulitan yang setara dan dapat digunakan untuk memenuhi frekuensi 1 indikator sampai 4 kali yaitu kondisi dimana indikator tersebut dinyatakan valid.	4,33
Mean	4,56

Guru merasa “**Sangat Setuju**” dengan kriteria efektifitas pada aplikasi permainan.

Beta test aplikasi permainan yang dilakukan oleh pendamping akan berfokus ke kepuasan. Beta test dilakukan dengan cara mendampingi saat anak bermain lalu menjawab pernyataan.

Tabel 6.5 Hasil UAT Aplikasi Permainan ke Pendamping

Pernyataan	Mean
Saat memainkan bagian 1 anak tidak mengalami kesulitan untuk menjawab.	4,40
Saat memainkan bagian 2 anak tidak mengalami kesulitan untuk menjawab.	4,00
Saat memainkan bagian 3 anak tidak mengalami kesulitan untuk menjawab.	4,67
Saat memainkan bagian 4 anak tidak mengalami kesulitan untuk menjawab.	4,67
Anak tidak mengalami kesulitan untuk menjawab dengan menekan tombol centang.	4,40
Mean	4,43

Pendamping merasa “**Sangat Setuju**” dengan kriteria kepuasan pada aplikasi permainan.

BAB VII

KESIMPULAN DAN SARAN

Pada bagian ini akan menjelaskan mengenai kesimpulan dari pengerjaan tugas akhir ini dan saran untuk pengembangan perangkat lunak selanjutnya yang relevan.

7.1. Kesimpulan

Berikut merupakan kesimpulan yang dapat diambil dari pengembangan sistem rapor dan aplikasi permainan yang telah dilakukan.

1. Dengan menggunakan proses iconix, proses bisnis rapor konvensional dapat diterjemahkan ke dalam sistem rapor.
2. Sistem rapor telah melalui pengujian dengan *unit test* untuk setiap *controller*-nya sehingga dapat dinyatakan setiap fungsi yang telah diuji bebas *bug*.
3. Sistem rapor dan aplikasi permainan telah memenuhi kebutuhan fungsional berdasarkan hasil *system test* dan *functional test*.
4. Sistem rapor dan aplikasi permainan telah melalui UAT dan pengguna “Sangat Setuju” dengan pernyataan positif untuk kriteria efektifitas dan kepuasan.
5. Sistem rapor memiliki *controller* untuk API yang dapat dipanggil melalui rute ‘/api/’ dan aplikasi permainan dapat memanggil API tersebut menggunakan obyek AJAX.

7.2. Saran

Berikut merupakan saran untuk pengembangan sistem lebih lanjut.

1. Menelusuri alasan mengapa beberapa poin UAT sistem rapor rata-ratanya lebih rendah dari lainnya.
2. Menambah tipe permainan supaya lebih banyak indikator yang dapat diamati secara otomatis.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Menteri Pendidikan dan Kebudayaan Republik Indonesia, “Permendikbud No. 137 Tahun 2014 tentang Standar Nasional PAUD,” 2014.
- [2] Kemendikbud, “Data Referensi Pendidikan.” [Online]. Available:
<https://referensi.data.kemdikbud.go.id/tabs.php?npsn=69815779>. [Accessed: 21-Oct-2019].
- [3] Dinas Pendidikan Pemuda dan Olahraga Kabupaten Tulungagung, “Izin Menyelenggarakan Pendidikan Anak Usia Dini (PAUD),” 188/0046/104.040/2019, 2019.
- [4] PAUD Ciluk-Baa, “Pedoman Program Pembelajaran,” 2018.
- [5] H. Purnomo, “Peran Orang Tua dalam Optimalisasi Tumbuh Kembang Anak untuk Membangun Karakter Anak Usia Dini,” *Pros. Semin. Nas. Parent.*, pp. 35–36, 2013.
- [6] E. B. Hurlock, *Child Development*, Terjemahan. Jakarta: PT. Gelora Aksara Pratama, 1991.
- [7] Menteri Pendidikan dan Kebudayaan Republik Indonesia, “Permendikbud No. 137 Tahun 2014. Lampiran I,” 2014.
- [8] R. Gerhaantara and F. Samopa, “Pembuatan Sistem Informasi Pendidikan dan Pelatihan dalam Jabatan Berbasis Web pada Bagian Pengembangan Pegawai Direktorat Jenderal Perbendaharaan,” *Tek. POMITS*, vol. 2, no. 2, p. A-376-A-380, 2013.
- [9] Xianjun Chen et al, “Restful API Architecture Based on Laravel Framework,” *J. Phys. Conf. Ser.* 910 012016, 2017.
- [10] N. Peirce, “Digital Game-based Learning for Early Childhood A State of the Art Report,” 2013.
- [11] Addison-Wesley, *The Unified Modeling Language User Guide*, 2nd ed. 2015.
- [12] D. Rosenberg and M. Stephens, *Use Case Driven Object*

- Modeling with UML Theory and Practice*. United States of America, 2007.
- [13] M. L. Braunstein, *Health Informatics on FHIR: How HL7's New API is Transforming Healthcare*. 2018.
- [14] E. Zimmerman and K. Salen, *Rules of Play: Game Design Fundamentals*. Cambridge: The MIT Press, 2003.
- [15] S. Holla and M. M. Katti, "ANDROID BASED MOBILE APPLICATION DEVELOPMENT and its Security," *Int. J. Comput. Trends Technol.*, vol. 3, no. 3, pp. 486–490, 2012.
- [16] R. Ramadan and Y. Widyani, "Game Development Life Cycle Guidelines," *ICACSI*, pp. 95–100, 2013.
- [17] S. A. Sualim, N. M. Yassin, and R. Mohamad, "Comparative Evaluation of Automated User Acceptance Testing Tool for Web Based Application," *Int. J. Softw. Eng. Technol. 02(2) 1-6 Int.*, vol. 02, no. 2, pp. 1–6, 2016.

LAMPIRAN A
REKAPITULASI UJI COBA PENGGUNA

Tabel 8.1 Rekapitulasi UAT Sistem Rapor

Kode	Use Case		Form Admin		Form Pengelola		Form Tata Usaha			Form Kelas			Form Murid				
			P1	P2	P1	P2	G1	G2	G3	G1	G2	G3	M1	M2	M3	M4	M5
UC01	Masuk	E	4	4	4	5	5	5	5	5	5	5	4	4	4	4	5
		S	5	4	5	5	5	5	5	5	5	5	4	5	5	4	5
UC02	Keluar	E	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
		S	5	4	5	5	5	5	5	5	5	5	5	4	5	4	5
UC03	Melihat data murid	E					5	5	5								
		S					5	5	4								
UC04	Mengisi data murid	E					5	5	5								
		S					5	5	5								
UC05	Mengubah data murid	E					5	5	5								
		S					5	5	5								
UC06	Menghapus data murid	E					5	5	5								
		S					5	5	5								
UC07	Melihat rapor	E								5	5	5	5	5	5	4	5
		S								5	5	5	5	4	4	5	5
UC08	Mencetak rapor	E								5	5	4	5	5	5	5	5
		S								5	5	5	4	4	5	5	4
UC09	Mengisi rapor	E								5	5	5					

		S								4	5	5					
UC10	Mengubah rapor	E								5	5	5					
		S								5	5	5					
UC11	Menaikkan murid	E								5	4	4					
		S								4	5	5					
UC12	Mengeluarkan murid	E								3	4	5					
		S								4	4	5					
UC13	Melihat indikator	E			4	5											
		S			4	5											
UC14	Menambahkan indikator	E			5	4											
		S			5	4											
UC15	Mengubah indikator	E			5	5											
		S			4	5											
UC16	Menghapus indikator	E			5	5											
		S			5	5											
UC17	Mengubah password	E	5	4													
		S	4	4													

Keterangan:

P: Pengelola

G: Guru

M: Wali Murid

E: Efektifitas

S: Kepuasan

Tabel 8.2 Rekapitulasi UAT Aplikasi Permainan ke Guru

Pernyataan	G1	G2	G3
Anak akan diberi 10 pertanyaan mengenai lambang bilangan yang muncul secara berurutan mulai dari 1-10. Anak menjawab dengan memasangkan jumlah benda dan lambang bilangan. Anak akan mendapat skor 1 setiap kali menjawab pertanyaan dengan benar. Jika anak mendapat skor minimal 10, maka anak akan dianggap memenuhi indikator 3.3.2.2. Menyebutkan bilangan 1-10 berurutan sambil menghitung benda.	4	5	5
Anak akan diberi 10 pertanyaan mengenai lambang bilangan yang muncul secara acak mulai dari 1-10. Anak menjawab dengan memasangkan jumlah benda dan lambang bilangan. Anak akan mendapat skor 1 setiap kali menjawab pertanyaan dengan benar. Jika anak mendapat skor minimal 10, maka anak akan dianggap memenuhi indikator 3.3.2.3. Memasangkan lambang bilangan dengan jumlah benda 1-10.	4	5	5
Anak akan diberi 10 pertanyaan cerita mengenai operasi pengurangan. Anak menjawab dengan memasangkan jumlah benda dan lambang bilangan hasil operasi. Anak akan mendapat skor 1 setiap kali menjawab pertanyaan dengan benar. Jika anak mendapat skor minimal 7, maka anak akan dianggap memenuhi indikator 3.3.4.1. Menghitung dan menyebutkan hasil pengurangan.	4	4	5
Anak akan diberi 10 pertanyaan cerita mengenai operasi penjumlahan. Anak menjawab dengan memasangkan jumlah benda dan lambang bilangan hasil operasi. Anak akan mendapat skor 1 setiap kali menjawab pertanyaan dengan benar. Jika anak mendapat skor minimal 7, maka anak akan dianggap memenuhi indikator 3.3.4.2. Menghitung dan menyebutkan hasil penjumlahan	4	5	5

Anak akan diberi batasan waktu 5 menit untuk menjawab 10 pertanyaan di setiap bagian. Batasan waktu ini diperlukan saat pengujian untuk membatasi giliran setiap anak.	4	5	5
Terdapat 4 tipe permainan yaitu: menarik angka ke kotak jawaban, menarik buah ke kotak jawaban, menggeser kotak jawaban untuk menangkap buah, dan menyeimbangkan buah pada kotak jawaban dengan angka. Keempatnya memiliki tingkat kesulitan yang setara dan dapat digunakan untuk memenuhi frekuensi 1 indikator sampai 4 kali yaitu kondisi dimana indikator tersebut dinyatakan valid.	4	4	5

Keterangan:

G: Guru

Tabel 8.3 Rekapitulasi UAT Aplikasi Permainan ke Wali Murid

Pernyataan	M1	M2	M3	M4	M5
Saat memainkan bagian 1 anak tidak mengalami kesulitan untuk menjawab.	5	5	4	5	3
Saat memainkan bagian 2 anak tidak mengalami kesulitan untuk menjawab.	5	2	4	5	4
Saat memainkan bagian 3 anak tidak mengalami kesulitan untuk menjawab.	4	5	5		
Saat memainkan bagian 4 anak tidak mengalami kesulitan untuk menjawab.	4	5	5		
Anak tidak mengalami kesulitan untuk menjawab dengan cara menekan tombol centang.	4	4	5	5	4

Keterangan:

M: Wali Murid

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Penulis bernama Krisna Fawwastaqi Raheko, lahir di Tulungagung pada tanggal 02 Nopember 1997, merupakan anak pertama dari dua bersaudara. Penulis telah menempuh jenjang pendidikan formal di beberapa sekolah, yaitu : SD Negeri Kampungdalem 1 Tulungagung (2004 - 2010), SMP Negeri 2 Tulungagung (2010 – 2013), dan SMA Negeri 1 Kedungwaru (2013 - 2016). Penulis melanjutkan pendidikan sarjana di Departemen

Sistem Informasi Fakultas Teknologi Elektro dan Informatika Cerdas (FTEIC) Institut Teknologi Sepuluh Nopember (ITS) pada tahun 2016 dan terdaftar dengan NRP 0521164000020.

Selama menjadi mahasiswa, penulis aktif mengikuti organisasi kemahasiswaan, yaitu Himpunan Mahasiswa Sistem Informasi pada tahun 2018-2019, serta kegiatan kepanitiaan, seperti Information Systems Expo 2017 & 2018. Penulis juga pernah meraih berbagai penghargaan pada lomba berskala nasional bidang pengembangan aplikasi permainan seperti “Lomba Karya Cipta PAUD” 2016 & 2018, “Lomba Aplikasi Mobile Ki Hajar” 2018, dan “Hology” 2019.

Penulis mengikuti mayoritas perlombaan dengan tim 3 orang dimana penulis berperan sebagai *game designer* atau *producer*.

Halaman ini sengaja dikosongkan