

**PERANCANGAN BACK-END PLATFORM KOMPETISI
ESPORTS BERBASIS WEB DENGAN METODE RAPID
APPLICATION DEVELOPMENT**



N a m a : Lalu Ahdiyat Alfian Hadi
NIM : 16523186

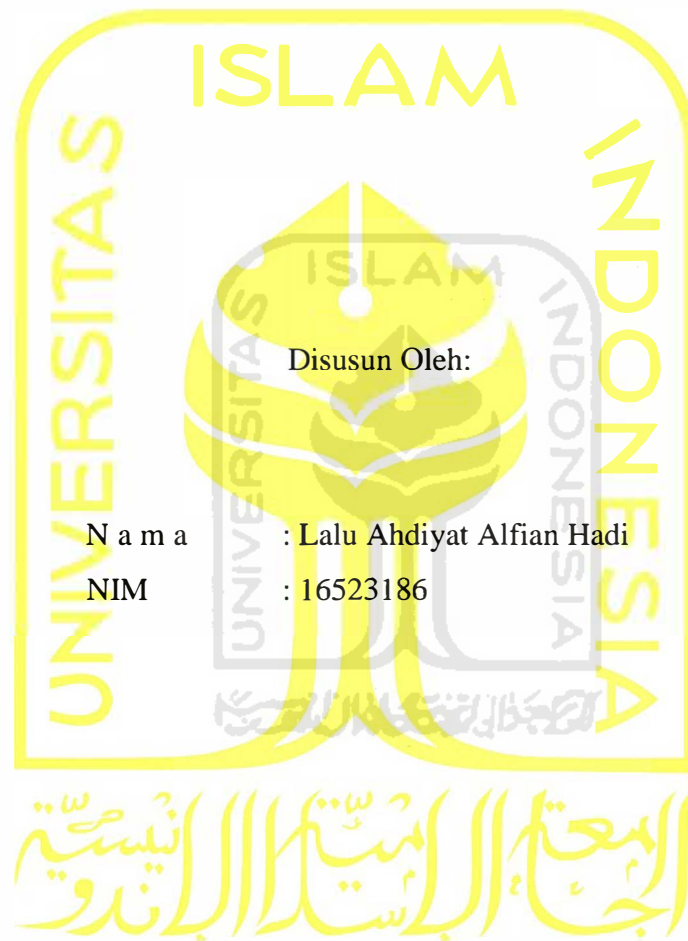
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PERANCANGAN BACK-END PLATFORM KOMPETISI
ESPORTS BERBASIS WEB DENGAN METODE RAPID
APPLICATION DEVELOPMENT**

TUGAS AKHIR



Disusun Oleh:

N a m a : Lalu Ahdiyat Alfian Hadi

NIM : 16523186

Yogyakarta, 12 Juli 2020

Pembimbing,

A handwritten signature in black ink, appearing to be 'Hari Setiaji', written over a light blue horizontal line.

(Hari Setiaji, S.Kom., M.Eng.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PERANCANGAN BACK-END PLATFORM KOMPETISI
ESPORTS BERBASIS WEB DENGAN METODE RAPID
APPLICATION DEVELOPMENT**

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 20 Juli 2020

Tim Penguji

Hari Setiaji, S.Kom., M.Eng.

Anggota 1

Fayruz Rahma, S.T., M.Eng.

Anggota 2

Hanson Prihantoro Putro, S.T., M.T.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Lalu Ahdiyati Alfian Hadi

NIM : 16523186

Tugas akhir dengan judul:

**PERANCANGAN BACK-END PLATFORM KOMPETISI
ESPORTS BERBASIS WEB DENGAN METODE RAPID
APPLICATION DEVELOPMENT**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 20 Juli 2020



(Lalu Ahdiyati Alfian Hadi)

HALAMAN PERSEMBAHAN

Puji dan syukur ke hadirat Allah Subhanahu wa Ta'ala atas segala rahmat, taufik, hidayah dan inayah-Nya yang senantiasa dilimpahkan kepada penulis sehingga dapat menyelesaikan laporan tugas akhir ini dengan baik, semoga keberhasilan ini akan membukakan pintu kesuksesan dalam meraih cita-cita. Shalawat dan salam senantiasa tercurah kepada Rasulullah Shalallaahu 'Alayhi Wasallam yang telah mengantarkan manusia dari zaman kegelapan ke zaman yang terang benderang ini. Dengan rasa syukur mendalam, tugas akhir ini saya persembahkan kepada:

1. Kedua orang tua, ayahanda tercinta Lalu Muslihun dan ibunda tersayang Yuliani, terima kasih telah membesarkan dan mendidik saya dengan sangat baik sehingga saya mencapai titik keberhasilan ini.
2. Kedua kakak tercinta, Lalu Muhammad Yassin Kashagie dan Baiq Lina Widya Sofiana, terima kasih atas segala motivasi dan dukungan kalian selama ini dan seterusnya.
3. Bapak Hari Setiaji, S.Kom., M.Eng., terima kasih telah membimbing dari awal perintisan ide hingga laporan tugas akhir ini selesai.
4. Anggota tim Avior, Heydar Arif Satrio dan Rafi Hidayat Tarmidzi terima kasih atas kerja sama, dukungan dan motivasi selama pengerjaan tugas akhir.
5. Seluruh teman-teman yang telah membantu penulis dalam menyelesaikan tugas akhir ini, terima kasih atas segala dukungannya.

HALAMAN MOTO

“Allah tidak akan membebani seseorang melainkan dengan kesanggupannya”

(QS. Al Baqarah: 286)

“Jika seseorang bepergian dengan tujuan mencari ilmu, maka Allah akan menjadikan perjalanannya seperti perjalanan menuju surga.”

(Nabi Muhammad Shalallaahu 'Alayhi Wasallam)

"Suatu pengetahuan (ilmu), walaupun tidak bermanfaat untukmu, tidak akan membahayakanmu."

(Umar bin Khattab)



KATA PENGANTAR

Puji dan syukur ke hadirat Allah Subhanahu wa Ta'ala atas segala rahmat, taufik, hidayah dan inayah-Nya yang senantiasa dilimpahkan kepada penulis sehingga dapat menyelesaikan laporan tugas akhir yang berjudul “Perancangan Back-End Platform Kompetisi Esports Berbasis Web dengan Metode Rapid Application Development” dengan baik. Shalawat dan salam senantiasa tercurah kepada Rasulullah Shalallaahu 'Alayhi Wasallam yang telah mengantarkan manusia dari zaman kegelapan ke zaman yang terang benderang ini.

Penyusunan skripsi ini dimaksudkan untuk memenuhi syarat guna menyelesaikan pendidikan Strata Satu (S1) di Program Studi Informatika – Program Sarjana, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penulis menyadari bahwa penyelesaian laporan tugas akhir ini tidak terlepas dari bantuan dan dukungan dari berbagai pihak baik moril maupun materil. Oleh karena itu, penulis ucapkan terima kasih kepada semua pihak yang telah membantu dalam menyelesaikan laporan tugas akhir ini terutama kepada:

1. Kedua orang tua, ayahanda tercinta Lalu Muslihun dan ibunda tersayang Yuliani yang selalu memberikan motivasi, doa yang tiada henti-hentinya serta dukungan baik moril maupun materil kepada penulis.
2. Bapak Hari Setiaji, S.Kom., M.Eng. selaku dosen Pembimbing Tugas Akhir I yang telah berkenan membimbing, memberikan pengetahuan serta memberikan solusi atas permasalahan terkait pengerjaan tugas akhir ini.
3. Bapak Beni Suranto, S.T., M.Soft.Eng., selaku dosen Pembimbing Tugas Akhir II yang telah membimbing dan memberikan saran-saran dalam perintisan ide untuk kepentingan tugas akhir ini.
4. Seluruh dosen Jurusan Informatika yang telah memberikan pengetahuan yang sangat bermanfaat selama perkuliahan.
5. Heydar Arif Satrio selaku hipster dalam tim yang telah bekerja sama dengan baik dalam menyelesaikan tugas akhir ini.
6. Rafi Hidayat Tarmidzi selaku hustler dalam tim yang telah bekerja sama dengan baik dalam menyelesaikan tugas akhir ini.
7. Teman-teman Jurusan Informatika Universitas Islam Indonesia yang telah memberikan motivasi serta dukungan.

Penulis menyadari bahwa laporan akhir ini masih jauh dari kata sempurna, maka dari itu penulis mengharpakan segala kritik dan saran yang membangun dari berbagai pihak. Semoga laporan akhir ini dapat bermanfaat bagi para pembaca.

Yogyakarta, 12 Juli 2020



(Lalu Ahdiyat Alfian Hadi)



SARI

Industri *esports* di Indonesia belakangan ini mengalami perkembangan yang cukup pesat. Hal ini ditandai dengan banyaknya bermunculan kompetisi *esports* berskala lokal hingga nasional dan juga mulai bermunculan pemain game *esports* baru. Aktivitas bermain game yang hanya sebagai hiburan semata perlahan mulai menghasilkan dengan mengikuti kompetisi *esports*. Namun demikian, informasi tentang kompetisi *esports* terkadang tidak sampai kepada pemain game. Kekhawatiran juga muncul dari pemain game *esports* tentang kompetisi fiktif yang dapat merugikan pemain. Oleh karena itu, diperlukan sebuah sistem yang memuat informasi seputar kompetisi *esports* yang memiliki sistem validasi terhadap suatu kompetisi sebagai media untuk para penyelenggara kompetisi dalam menyebarkan suatu kompetisi dan untuk memudahkan para pemain game *esports* dalam mengikuti suatu kompetisi. *Website* adalah teknologi yang dipilih sebagai basis dari sistem ini. Metode pengembangan yang digunakan dalam pengembangan sistem ini adalah *rapid application development* (RAD) yang terdiri dari tahap *Requirements Planning*, *Workshop Desain RAD*, *Implementation*. Metode ini menekankan pada siklus pembangunan pendek, singkat, dan cepat serta cocok untuk sistem berskala kecil (Chusyairi, Rudiawan, & Sasongko, 2018). Pengujian *website* dilakukan dengan menguji fungsi atau fitur secara otomatisasi dengan teknik *browser test* menggunakan Dusk. Dari hasil pengujian terhadap beberapa fitur didapatkan hasil dengan status “OK” yang artinya tidak ditemukan kesalahan.

Kata kunci: *Rapid Application Development*, *esports*, web

GLOSARIUM

<i>Back-end</i>	bagian belakang dalam pembangunan web, sebagian besar merujuk pada proses pemrograman. <i>Back-end</i> web menangani bagian basis data, <i>controller</i> , <i>routing</i> dan sebagainya.
<i>Debugging</i>	proses analisis atau mencari kesalahan kode program untuk disempurnakan sehingga program dapat dijalankan seperti apa yang diharapkan.
<i>Ekstensi</i>	aplikasi atau fitur tambahan dari sebuah perangkat lunak. <i>Ekstensi</i> berfungsi untuk meningkatkan kemampuan dari sebuah perangkat lunak dan biasanya dibuat oleh pengembang pihak ketiga.
<i>Esports</i>	olahraga elektronik atau aktivitas bermain <i>video game</i> baik <i>online</i> ataupun <i>offline</i> yang dilakukan secara kompetitif.
<i>Feedback</i>	tanggapan yang diberikan oleh calon pengguna kepada tim pengembang perangkat lunak. Tanggapan ini berguna sebagai pertimbangan dalam membuat suatu keputusan dalam pembangunan perangkat lunak.
<i>Framework</i>	kerangka kerja yang diciptakan untuk mempermudah pengembang dalam proses pembangunan web dengan lebih cepat dan terstruktur.
<i>Incremental</i>	model pembangunan perangkat lunak dengan memecah sistem menjadi beberapa modul pada tahap desain sistem.
<i>Open source</i>	perangkat lunak yang kodenya dapat dilihat dan dikembangkan oleh pengguna.
<i>Organizer</i>	penyelenggara kompetisi <i>esports</i> , biasanya berupa organisasi atau perusahaan.
<i>Player</i>	pemain game <i>esports</i> atau peserta kompetisi <i>esports</i> .
<i>RAD</i>	metode pengembangan perangkat lunak <i>rapid application development</i> .
<i>Software</i>	perangkat lunak atau program pada komputer berupa instruksi-instruksi yang dapat dimengerti oleh komputer.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Metodologi Pengembangan	4
1.7 Sistematika Penulisan	4
BAB II KAJIAN PUSTAKA.....	6
2.1 Startup.....	6
2.2 Esports.....	6
2.3 Online Game	6
2.4 Rapid Application Development.....	7
2.5 Website	7
2.5.1 Basis Data.....	8
2.5.2 HyperText Markup Language (HTML)	9
2.5.3 PHP: Hypertext Preprocessor	9
2.5.4 Cascading Style Sheets (CSS).....	9
2.5.5 JavaScript	10
2.6 Penelitian Terdahulu	10

BAB III METODOLOGI.....	11
3.1 Perintisan Ide	11
3.1.1 Growth Talk: Big Data for Startup.....	11
3.1.2 Kelas Inspirasi: Menjadi Entrepreneur	13
3.1.3 Grow your tech Startup with Effective Brand Strategy	14
3.1.4 Global Startup Weekend Sustainable Revolution	15
3.2 Perancangan Back-End Sistem	17
3.3 Metode Pengembangan	18
3.3.1 Requirements Planning.....	18
3.3.2 Workshop Design RAD.....	19
3.3.3 Implementation.....	19
BAB IV HASIL DAN PEMBAHASAN	20
4.1 Penerapan Rapid Application Development.....	20
4.2 Requirements Planning	20
4.2.1 Rencana Kebutuhan.....	21
4.2.2 Kebutuhan Ekstensi Teks Editor	22
4.2.3 Rencana Kegiatan.....	23
4.3 Workshop Design RAD	24
4.3.1 Use Case Diagram	24
4.3.2 Activity Diagram.....	28
4.3.3 Desain Entity Relationship Diagram.....	34
4.4 Implementation	37
4.4.1 Basis Data.....	37
4.4.2 Integrasi Antarmuka pada Back-End	57
4.4.3 Pengujian.....	70
BAB V KESIMPULAN DAN SARAN	78
5.1 Kesimpulan	78
5.2 Saran	78
DAFTAR PUSTAKA	79

DAFTAR TABEL

Tabel 4.1 Ekstensi dalam Visual Studio Code.....	22
Tabel 4.2 Rencana kegiatan perancangan <i>back-end</i>	23
Tabel 4.3 Relasi entitas	36
Tabel 4.4 Struktur tabel users	39
Tabel 4.5 Struktur tabel organizers	40
Tabel 4.6 Struktur tabel players	41
Tabel 4.7 Struktur tabel banks	42
Tabel 4.8 Struktur tabel platforms	43
Tabel 4.9 Struktur tabel games	44
Tabel 4.10 Struktur tabel teams	45
Tabel 4.11 Struktur tabel tournaments.....	47
Tabel 4.12 Struktur tabel informations	48
Tabel 4.13 Struktur tabel match.....	49
Tabel 4.14 Struktur tabel schedules	50
Tabel 4.15 Struktur tabel organizer_transaction.....	51
Tabel 4.16 Struktur tabel team_transaction	52
Tabel 4.17 Struktur tabel tournament_transaction.....	54
Tabel 4.18 Struktur tabel proofs	55
Tabel 4.19 Struktur tabel player_team.....	56
Tabel 4.20 Struktur tabel team_tournament.....	57

DAFTAR GAMBAR

Gambar 2.1 <i>Rapid Application Development</i>	7
Gambar 2.2 Arsitektur web.....	8
Gambar 3.1 Growth Talk	12
Gambar 3.2 Kelas Inspirasi.....	13
Gambar 3.3 Global Startup Weekend Sustainable Revolution	15
Gambar 3.4 Speakers dan coach Global Startup Weekend.....	16
Gambar 3.5 Aktivitas coaching Global Startup Weekend	16
Gambar 3.6 Juri Global Startup Weekend	17
Gambar 3.7 <i>Rapid Application Development</i>	18
Gambar 4.1 <i>Use Case Diagram Admin</i>	25
Gambar 4.2 <i>Use Case Diagram Organizer</i>	26
Gambar 4.3 <i>Use Case Diagram Player</i>	27
Gambar 4.4 <i>Use Case Diagram Pengunjung</i>	28
Gambar 4.5 <i>Activity diagram</i> membuat kompetisi	29
Gambar 4.6 <i>Activity diagram</i> membuat tim.....	30
Gambar 4.7 <i>Activity diagram</i> pengisian saldo tim.....	32
Gambar 4.8 <i>Activity diagram</i> bergabung kompetisi	34
Gambar 4.9 Entity Relationship Diagram.....	35
Gambar 4.10 Konfigurasi <i>file .env</i>	38
Gambar 4.11 Migrasi tabel users	38
Gambar 4.12 Migrasi tabel organizers	39
Gambar 4.13 Migrasi tabel players	41
Gambar 4.14 Migrasi tabel banks	42
Gambar 4.15 Migrasi tabel platforms	43
Gambar 4.16 Migrasi tabel games	44
Gambar 4.17 Migrasi tabel teams	45
Gambar 4.18 Migrasi tabel tournaments.....	46
Gambar 4.19 Migrasi tabel informations	48
Gambar 4.20 Migrasi tabel matches	49
Gambar 4.21 Migrasi tabel schedules	50
Gambar 4.22 Migrasi tabel organizer_transaction.....	51
Gambar 4.23 Migrasi tabel team_transaction	52

Gambar 4.24 Migrasi tabel tournament_transaction.....	53
Gambar 4.25 Migrasi tabel proofs	54
Gambar 4.26 Migrasi tabel player_team.....	55
Gambar 4.27 Migrasi tabel team_tournament	56
Gambar 4.28 Halaman login	58
Gambar 4.29 Kode login.....	58
Gambar 4.30 Halaman register	59
Gambar 4.31 Kode register player	60
Gambar 4.32 Kode register organzier	60
Gambar 4.33 Halaman beranda.....	62
Gambar 4.34 Kode beranda	63
Gambar 4.35 Halaman tournament	64
Gambar 4.36 Kode tournament.....	64
Gambar 4.37 Halaman detail tournament	65
Gambar 4.38 Kode detail tournament.....	66
Gambar 4.39 Halaman detail team.....	67
Gambar 4.40 Kode detail tim.....	68
Gambar 4.41 Halaman pembayaran midtrans.....	69
Gambar 4.42 Kode transaksi midtrans.....	70
Gambar 4.43 Kode pengujian register.....	71
Gambar 4.44 Kode pengujian login.....	72
Gambar 4.45 Kode pengujian logout.....	73
Gambar 4.46 Kode pengujian pembuatan game	73
Gambar 4.47 Kode pengujian pembuatan tim	74
Gambar 4.48 Kode pengujian bergabung tim	74
Gambar 4.49 Kode pengujian pembuatan kompetisi.....	75
Gambar 4.50 Kode pengujian bergabung kompetisi.....	76
Gambar 4.51 Hasil pengujian	76

BAB I PENDAHULUAN

1.1 Latar Belakang

Industri *esports* beberapa tahun terakhir ini mengalami perkembangan yang cukup pesat di Indonesia. Hal tersebut ditandai dengan hampir di setiap daerah dapat ditemukan kompetisi-kompetisi *esports* baik itu yang berskala lokal hingga nasional. Indonesia merupakan salah satu pasar *video game* terbesar di Asia Pasifik, dengan angka sekitar 13 triliun rupiah (Newzoo, 2019).

Seiring dengan perkembangan industri *esports* ini, bermunculan juga tim-tim baru dan para pemain (*player*) yang bergerak dalam bidang kompetisi *esports*. Para penikmat *esports* ini mulai dari pelajar sampai dengan orang tua baik itu sebagai penonton, pemain, penyelenggara turnamen, ataupun pemilik tim. Kompetisi *esports* mempertandingkan sebuah tim dengan tim lainnya, setiap tim biasanya terdiri dari 4 sampai 5 orang dan juga 1 orang cadangan tergantung pada game yang ditandingkan.

Pada tahun 2019 telah diselenggarakan kompetisi *esports* tingkat nasional khusus mahasiswa oleh Indonesian eSports League (IeSPA) yang didukung oleh Federasi Olahraga Rekreasi Masyarakat Indonesia (FORMI), Kementerian Pemuda dan olahraga (KEMENPORA), dan Asian Electronic eSports Federation (AESF). Pemerintah Republik Indonesia juga ikut andil dalam perkembangan industri *esports* ini dengan mengadakan turnamen *esports* bernama Piala Presiden Esport pada tahun 2018 dan 2019. Piala Presiden Esports merupakan kompetisi *esports* hasil kolaborasi antara Kantor Staf Presiden, Kementerian Pemuda dan Olahraga (KEMENPORA), Kementerian Komunikasi dan Informatika (KEMENKOMINFO), Badan Ekonomi Kreatif (BEKRAF), Indonesia Esport Premier League (IESPL), Bank Central Asia (BCA), dan Blibli.com. Dengan perkembangan *esports* ini, sejak tahun 2018 *esports* mulai dilombakan dalam ajang Asian Games.

Dengan perkembangan industri *esports* di Indonesia ini, berarti membuka peluang bisnis industri *esports* dengan potensi pasar yang bisa dibilang cukup menjanjikan. Salah satu jenis bisnis yang sedang populer belakangan ini yaitu bisnis *startup*. Dalam suatu *startup*, terdapat tiga peran atau karakter utama yaitu *hustler* yang berperan sebagai analis, *hipster* bertugas menangani *front-end*, dan *hacker* bertugas menangani *back-end*.

Untuk membangun sebuah bisnis seperti *startup*, terlebih dahulu harus dilakukan identifikasi masalah ataupun kebutuhan terhadap calon pengguna dan selanjutnya dibuat konsep dan model sistem berdasarkan hasil identifikasi. Selain itu, akan lebih baik jika disertai dengan mengikuti berbagai kegiatan tentang teknologi *startup*.

Salah satu metode dalam mengidentifikasi masalah ataupun kebutuhan calon pengguna yaitu dengan melakukan wawancara atau survei terhadap calon pengguna. Berdasarkan dari wawancara yang telah dilakukan pada beberapa pemain game *esports*, rata-rata para *player* ini mempunyai masalah yaitu seputar informasi sebuah turnamen *esports*. Terkadang para *player* tersebut ketinggalan informasi bahkan tidak mendapatkan informasi tentang sebuah turnamen padahal mereka para *player* tersebut ingin dan atau berpotensi untuk mengikuti turnamen *esports* yang ada. Salah satu penyebabnya yaitu tidak adanya media khusus yang memuat informasi-informasi tentang sebuah turnamen *esports* secara *up-to-date* dan dapat diakses dengan mudah oleh para *player esports*. Selama ini rata-rata para *player esports* mendapatkan informasi tentang suatu turnamen *esports* melalui berbagai media sosial maupun brosur di beberapa tempat tertentu seperti *gamenet* dan *icafe*. Selain itu, penyebab para *player esports* tidak dapat mengikuti kompetisi yaitu karena tidak memiliki tim, sedangkan sebagian besar kompetisi *esports* mempertandingkan antara tim satu dengan tim lainnya.

Kemajuan teknologi pada saat ini diharapkan dapat menjadi solusi dalam penyebaran informasi kompetisi *esports* melalui media *website* sehingga mempermudah para pelaku *esports* khususnya para pemain dan penyelenggara kompetisi *esport* dalam pengaksesan informasi seputar kompetisi *esports*. Dalam hal ini, dibangunlah platform kompetisi *esports* berbasis *website* yang bernama Arenasia. Arenasia menyediakan informasi-informasi tentang turnamen *esports* secara *up-to-date*, para *player esports* dapat mendaftarkan tim mereka pada sebuah kompetisi *esports* secara langsung melalui *website* Arenasia. Selain itu, penyelenggara kompetisi juga dapat mempromosikan kompetisi yang mereka adakan dengan mendaftarkan organisasi mereka dalam *platform* Arenasia. Setelah itu, para penyelenggara dapat memberikan informasi *update* seputar jadwal pertandingan, peserta kompetisi, dan atau informasi apapun untuk para peserta kompetisi.

Salah satu metode yang pengembangan perangkat lunak adalah *rapid application development*. Metode *rapid application development* mengedepankan proses pembangunan yang singkat dan cepat yang dalam pembangunannya menggunakan teknik *incremental* dengan membagi sistem menjadi beberapa modul yang dikerjakan oleh beberapa tim. Maka dari itu,

dalam perancangan sistem berbasis *website* Arenasia modul pengerjaan dibagi menjadi *front-end* dan *back-end*.

Agar *website* Arenasia dapat diakses di luar jaringan lokal secara daring, maka dilakukan perancangan *back-end website* menggunakan *framework* Laravel dan MySQL sebagai basis data.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka didapatkan rumusan masalah pada pembangunan *platform* kompetisi *esports* Arenasia, yaitu:

- a. Bagaimana merancang *back-end website* Arenasia menggunakan metode pengembangan *rapid application development*?
- b. Bagaimana membangun *back-end website* Arenasia menggunakan *framework* Laravel dan basis data MySQL?
- c. Bagaimana memanfaatkan sistem yang telah ada untuk mempercepat proses perancangan *back-end website* Arenasia?

1.3 Batasan Masalah

Agar tugas akhir fokus pada permasalahan dan penyelesaiannya serta tidak keluar dari ruang lingkup topik yang dibahas, maka dibuat batasan masalah. Berikut adalah batasan masalah yang telah ditentukan:

- a. Pilihan *game* yang disediakan hanya *game online* yang sedang populer yaitu Dota 2, Counter Strike: Global Offensive, Mobile Legends: Bang Bang, Playerunknown's Battleground PC dan mobile.
- b. Konten hanya berupa informasi kompetisi *esports*, pendaftaran kompetisi *esport* dan jadwal pertandingan kompetisi *esports* yang sedang berlangsung.
- c. Hanya memuat kompetisi dalam wilayah negara Indonesia.

1.4 Tujuan Penelitian

Tujuan pembuatan tugas akhir ini yaitu membangun *back-end* sebuah *platform* kompetisi *esports* berbasis *website* yaitu Arenasia menggunakan metode *rapid application development* dalam proses pembangunannya dan mengintegrasikan antarmuka *website* dengan basis data sehingga menjadi dinamis dan dapat diakses secara daring dari mana saja.

1.5 Manfaat Penelitian

Manfaat dari pembangunan platform kompetisi *esports* Arenasia ini adalah sebagai berikut:

- a. Membuat antarmuka ataupun konten *website* dapat diakses oleh pengguna melalui *browser*.
- b. Membuat *website* terintegrasi dengan basis data.
- c. Menjadikan *website* dapat diakses di luar jaringan lokal secara daring.

1.6 Metodologi Pengembangan

Metode yang digunakan dalam membangun platform kompetisi *esports* berbasis web ini adalah metode Rapid Application Development (RAD). Metode ini menekankan pada siklus pembangunan pendek, singkat, dan cepat serta cocok untuk sistem berskala kecil. *Rapid application development* merupakan metode pengembangan perangkat lunak yang terdiri dari fase *Requirements Planning*, *Workshop Design RAD* dan *Implementation*.

1.7 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini terdiri dari lima bab. Setiap bab memiliki beberapa subbab yang dibuat untuk mempermudah pembahasan dan penyampaian. Berikut penjelasan tiap bab tersebut:

BAB I PENDAHULUAN

Pada bab ini, dijelaskan secara singkat mengenai tugas akhir ini. Bab ini terdiri dari latar belakang dibangunnya *platform* kompetisi *esports* Arenasia, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan tugas akhir.

BAB II KAJIAN PUSTAKA

Bab ini berisi istilah yang digunakan pada tugas akhir ini serta penjelasan mengenai dasar teori yang digunakan dalam pembangunan *platform* kompetisi *esports* berbasis web, seperti penjelasan tentang *website* dan pembangunannya. Selain itu, pada bab ini dimuat beberapa penelitian terdahulu sebagai referensi dalam penyusunan tugas akhir.

BAB III METODOLOGI

Pada bab ini, dijelaskan metode beserta tahapan yang digunakan dalam membangun *platform* kompetisi *esports* Arenasia mulai dari perancangan sistem, desain sistem dan implementasi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi pembahasan mengenai hasil akhir yang didapatkan selama pembuatan tugas akhir ini, seperti penjelasan proses pembuatan basis data, integrasi tampilan *website* dengan *back-end* beserta kode yang menanganinya dan hasil pengujian.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari tugas akhir yang telah dilakukan dengan membangun *platform* kompetisi *esports*. Selain itu, bab ini berisi saran bagi penelitian sejenis selanjutnya untuk perbaikan ke depannya sebagai pengembangan dari tugas akhir ini agar menjadi lebih baik.



BAB II KAJIAN PUSTAKA

2.1 Startup

Startup adalah perusahaan atau proyek yang diprakarsai oleh seorang wirausahawan untuk mencari, mengembangkan, dan memvalidasi model bisnis yang dapat diukur secara efektif (Robehmed, 2013). *Startup* menghadapi ketidakpastian yang tinggi (Schmitt, Rosing, Zhang, & Leatherbee, 2018) dan memiliki tingkat kegagalan yang tinggi, tetapi sebagian kecil dari *startup* berhasil dan berpengaruh (Errin, 2004).

Terdapat tiga jenis peran dengan kompetisi yang spesifik dalam *startup* yaitu *hustler*, *hipster* dan *hacker*. *Hustler* adalah peran yang memiliki kompetensi dalam bidang analisis, pemasaran dan bisnis. *Hipster* adalah peran yang memiliki kompetensi dalam bidang desain. *Hacker* adalah peran yang memiliki kompetensi dalam bidang teknologi (Zhong, Liu, Zhong, & Xiong, 2008). Ketiga peran tersebut harus berkolaborasi dengan baik untuk memaksimalkan kinerja *startup*.

2.2 Esports

Menurut (Wagner, 2006), *esports* merupakan suatu kegiatan olahraga di mana orang melatih dan mengembangkan mental atau fisik dalam penggunaan teknologi informasi dan komunikasi. Kegiatan olahraga yang dimaksud adalah aktivitas memainkan game baik itu di komputer ataupun konsol, *game esports* seringkali melibatkan koneksi internet (Anthony K., 2012).

Istilah "Olahraga elektronik" atau "*esports*" berawal dari akhir tahun sembilan puluhan. Ketika salah satu sumber terpercaya pertama yang menggunakan istilah "*esports*" adalah pada siaran pers tahun 1999 tentang peluncuran Online Gamers Association (OGA). Munculnya *esports* di kultur masa kini menjadi aspek bisnis dari dalam industri *video game* (The OGA, 1999).

2.3 Online Game

Online game adalah *game* yang dimainkan secara daring dengan koneksi internet (Young, 2009). *Online game* sendiri memiliki beberapa *genre*, namun *genre online game* yang

paling banyak ditemukan pada suatu kompetisi di antaranya yaitu Multiplayer Online Battle Arena (MOBA), First-Person Shooter (FPS) dan Battle Royale.

2.4 Rapid Application Development

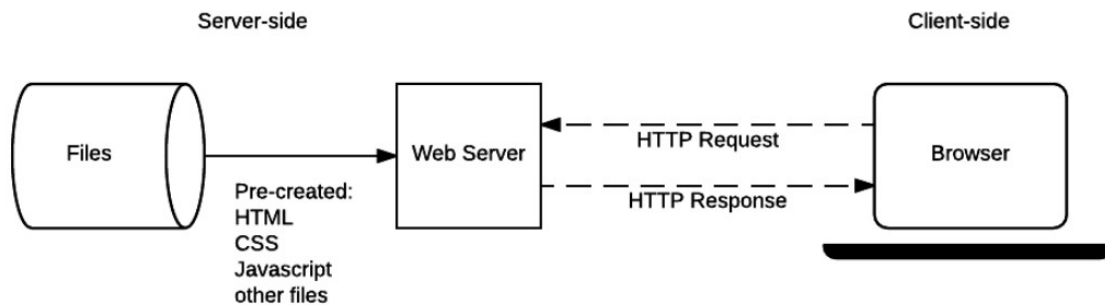


Gambar 2.1 *Rapid Application Development*

Rapid application development (RAD) adalah suatu metode pengembangan perangkat lunak yang tergolong dalam teknik *incremental* dan menggunakan metode iteratif (berulang) dalam mengembangkan sistem. Waktu yang singkat adalah batasan yang penting karena metode ini menekankan pada siklus pembangunan pendek, singkat, dan cepat. *Rapid application development* menggunakan metode iteratif dalam mengembangkan sistem, di mana pada tahap awal pengembangan ditetapkan *working model* (model bekerja) dengan tujuan untuk menetapkan kebutuhan pengguna dan selanjutnya disingkirkan (Britton & Doake, 2001). Metode ini terdiri dari 3 tahap yaitu *requirements planning*, *workshop design RAD* dan *implementation* (Kendall, 2010).

2.5 Website

Website atau biasa disingkat web adalah kumpulan halaman atau info dokumen pada suatu domain internet yang bisa diakses menggunakan *browser* melalui jaringan lokal maupun jarak jauh (Hakim & Musalini, 2004).



Gambar 2.2 Arsitektur web

Sumber: Mozilla (2020)

Arsitektur web baik yang sederhana maupun kompleks berjalan pada mekanisme *client-server* seperti pada gambar 2.1. Sisi *client* umumnya merupakan sebuah *browser* yang tertanam pada sebuah perangkat elektronik seperti komputer, *handphone*, *smartphone* dan sebagainya, sedangkan *server* merupakan sebuah sistem komputer yang tertanam *web server* di dalamnya. Kedua hal tersebut berkomunikasi melalui suatu jaringan baik itu *localhost*, LAN maupun internet dengan metode *request* dan *response* (Suehring, 2002). Pada dasarnya, suatu *website* disusun oleh beberapa komponen utama, seperti basis data, HyperText Markup Language (HTML), PHP: Hypertext Preprocessor, Cascading Style Sheet (CSS) dan JavaScript.

2.5.1 Basis Data

Basis data merupakan kumpulan dari beberapa tabel yang saling terhubung dan memuat informasi. Basis data menyimpan informasi dalam sebuah tabel. Tabel adalah struktur dari database yang terdiri dari setidaknya satu kolom, tetapi biasanya lebih dari satu kolom. Basis data tidak hanya untuk menyimpan data, namun juga dapat berfungsi dalam penyortiran, pengindeksan, memanggil kueri, dan mengatur data untuk membantu *server database* (Suehring, 2002).

MySQL bukan merupakan basis data, melainkan sebuah perangkat lunak untuk membuat dan mengolah basis data (Suehring, 2002). MySQL memiliki 2 versi yaitu MySQL Enterprise Edition (berbayar) dan MySQL Community Edition (gratis). Untuk mengakses suatu basis data diperlukan MySQL Client, salah satu yang paling populer yaitu phpMyAdmin walaupun MySQL telah menyertakan MySQL Client bawaan berbasis terminal pada paket instalasi (Solichin, 2016).

2.5.2 HyperText Markup Language (HTML)

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* untuk membuat dokumen atau halaman pada suatu web. Dokumen dapat mencakup gambar bergerak, teks, tampilan grafis dan suara (United States of America Patent No. 5,982,445, 1999).

2.5.3 PHP: Hypertext Preprocessor

PHP adalah sebuah bahasa pemrograman yang digunakan untuk memproses permintaan dalam sebuah web. Ketika client mengirim sebuah permintaan ke *server*, PHP akan mengeksekusi dan menafsirkan permintaan tersebut menjadi dokumen HTML agar *client* dapat mengaksesnya melalui *browser* (Pomaska, 2012). PHP dapat disisipkan dalam dokumen HTML sehingga membuat dokumen HTML menjadi dinamis.

Laravel merupakan sebuah *framework* (kerangka kerja) PHP yang digunakan untuk membangun sebuah web. Laravel menerapkan konsep *model-view-controller* (MVC) dalam penggunaannya serta didesain dengan sintaks yang bersih dan fungsional yang dapat mengurangi waktu dan biaya pengembangan. Menurut pencipta Laravel, Taylor Otwell, alasannya membuat *framework* adalah kurangnya beberapa fungsi penting, seperti autentikasi pengguna dalam *framework* CodeIgniter (Surguy, 2013).

Laravel memiliki dokumentasi yang lengkap dan komunitas yang banyak, termasuk di Indonesia. Selain itu, Laravel juga bersifat *open source*. Web yang dibangun dengan Laravel memiliki skalabilitas yang kuat sehingga dapat meningkatkan efisiensi pengembangan (Di, 2014) (Zheng, 2014).

2.5.4 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) adalah bahasa yang digunakan untuk mendeskripsikan tampilan dan pemformatan pada dokumen HTML. CSS telah banyak digunakan dalam praktik pengembangan web dan seluler. Bahasa ini memiliki fitur yang kompleks, seperti warisan, kaskade, dan kekhasan (Mazanian, Tsantalís, & Mesbah, 2014). CSS memungkinkan penggunaanya dalam memodifikasi sebuah element pada HTML, seperti penempatan objek dan properti huruf (Lie, Wium, & Bos, 2005).

Bootstrap adalah sebuah *framework* (kerangka kerja) CSS yang dibuat oleh Mark Otto dan Jacob Thornton ketika bekerja di Twitter (Spurlock, 2013). Bootstrap memudahkan pengguna ketika membangun antarmuka web, karena menyediakan templat berbasis HTML dan CSS dengan komponen antarmuka seperti tabel, formulir, tombol, tipografi, *dropdown*,

navigasi dan banyak fitur lainnya. Bootstrap juga memiliki fitur agar tata letak web menjadi fleksibel dan responsif (Shah, 2015).

2.5.5 JavaScript

JavaScript adalah bahasa pemrograman tingkat tinggi dan dinamis, yang dimodelkan dengan kemampuan berorientasi objek (OO). Secara sintaksis, bahasa inti JavaScript menyerupai C, C++, dan Java, dengan konstruksi pemrograman seperti pernyataan dengan *if*, *loop* dengan *while* dan operator menggunakan simbol seperti “&&”. JavaScript umumnya digunakan pada web *browser* untuk berinteraksi dengan pengguna, mengontrol *browser* web, dan mengubah konten dokumen yang muncul di dalam jendela web *browser*. Bahasa JavaScript mendukung tipe data primitif seperti *number*, *string*, dan nilai *Boolean* (Flanagan, 2006).

2.6 Penelitian Terdahulu

Dalam penyusunan tugas akhir ini, digunakan tinjauan studi terdahulu yang berkaitan dengan judul yang diambil yaitu tentang perancangan web *Esports* sebagai pendukung penyusunan tugas akhir. Berikut tiga referensi terdahulu yang digunakan:

1. Ade Asrianto dalam tugas akhir yang berjudul “Sistem Informasi Komunitas Pecandu *Game* Berbasis Web” pada tahun 2014. Tugas akhir ini membahas tentang pembangunan *website* informasi seputar *game esports* yang ditujukan untuk para pecandu *game* agar lebih mudah bertukar informasi sesama pecandu *game* dan untuk meningkatkan pengetahuan seputar *game esports*.
2. Fawwaz Ashrafuddin dalam tugas akhir yang berjudul “Sistem Informasi Portal Berita *Esports* Berbasis Web” pada tahun 2017. Tugas akhir ini membahas tentang perancangan dan desain portal berita *esports* berbasis web dengan menggunakan bahasa pemrograman PHP dan basis data MySQL yang ditujukan untuk para penggemar *esports* atau masyarakat agar lebih mudah mendapatkan informasi tentang *esports*.
3. Galang Sapto Manggolo dalam tugas akhir yang berjudul “Rancang Bangun Aplikasi Penjadwalan *Coaching* Berbasis *Website* pada Akademi *Esports* Indonesia” pada tahun 2020. Tugas akhir ini membahas tentang perancangan dan pembangunan *website* penjadwalan *coaching* pada Akademi *esports* Indonesia yang ditujukan untuk para manajer dalam menyusun dan merencanakan jadwal *coaching* antara pelatih dan *gamers*.

BAB III METODOLOGI

Dalam merancang *website* Arenasia, penulis dan tim pada awalnya mengikuti berbagai kegiatan pada bidang teknologi *startup* dalam rangka mendapatkan ide dan pengalaman. Berdasarkan pada ide yang disepakati yaitu platform kompetisi *esports*, selanjutnya *hustler* membuat konsep dan desain awal sistem. Konsep dan desain awal sistem Arenasia yang telah dirancang oleh *hustler* tersebut selanjutnya dieksekusi oleh *hipster* dan *hacker* yang masing-masing bertugas menangani bagian *front-end* dan *back-end* sistem.

3.1 Perintisan Ide

Dalam merintis sebuah bisnis khususnya pada bidang teknologi seperti *startup*, untuk mendapatkan inspirasi atau ide maka dibutuhkan pengalaman baik itu dengan mengikuti seminar maupun *workshop*. Selama tahun 2019, penulis telah mengikuti beberapa acara yaitu seminar "Growth Talk: Big Data for Startup" oleh Ibisma UII, Kelas Inspirasi "Menjadi Entrepreneur", seminar "Grow your tech Startup with Effective Brand Strategy" oleh CoLearn, "Global Startup Weekend Sustainable Revolution" edisi Surabaya dan "Business Plan Competition: from Code to Gold" oleh Magister Informatika UII.

3.1.1 Growth Talk: Big Data for Startup

Seminar Growth Talk "Big data for Startup" Ibisma UII di The Growth Hub membahas tentang *data science*, *data analytics* serta *artificial intelligence* dengan pembicaranya yaitu Tuti Purwaningsih S.Stat., M.Si. (Co-Founder Data Science Indonesia) dan Ridho Rahmadi S.Kom., M.Sc., Ph.D (Director of Centre of Data Science-Informatics UII).



Gambar 3.1 Growth Talk

Gambar 3.1 merupakan sesi pembahasan tentang *data science* oleh pembicara pertama yaitu Tuti Purwaningsih. Ia berbicara tentang awal ketika mendirikan Data Science Indonesia, bagaimana penggunaan *big data* pada suatu perusahaan dalam membaca *trend* pasar hingga apa saja yang diperlukan jika ingin menjadi seorang *Data Science/Analytic*. Masih dengan tema *big data*, pembicara kedua yaitu Ridho Rahmadi berbicara tentang perkembangan teknologi khususnya *artificial intelligence*, *machine learning*, *deep learning* dan pemanfaatan *data science* di berbagai bidang.

Poin penting pada seminar ini yaitu pengetahuan tentang *data analytic* sangat penting dalam sebuah *startup* yang secara garis besar berfungsi sebagai *analytic*, *tracking*, *validation* dan *features*. Pembicara juga membagikan langkah awal dalam memulai *startup*, di antaranya memahami target pasar, menganalisis perkembangan pasar, memanfaatkan platform yang ada serta mengoptimalkan sosial media sebagai media pemasaran produk.

3.1.2 Kelas Inspirasi: Menjadi Entrepreneur



Gambar 3.2 Kelas Inspirasi

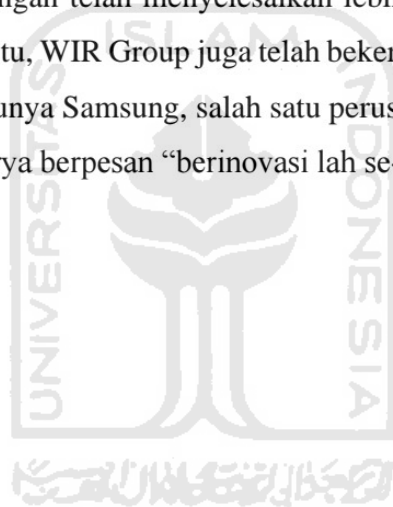
Sumber: Instagram Magister Informatika UII (2019)

Seminar Kelas Inspirasi “Menjadi Entrepreneur” yang diadakan di Auditorium FTI seperti pada gambar 3.2 menghadirkan dua pembicara yaitu Novi Wahyuningsih (CEO CakraTalk) dan Aryo Wiryawan (JALA). Pembicara pertama, Novi Wahyuningsih menceritakan pengalamannya dalam membangun perusahaan *startup* mulai dari menjadi penjaga warung internet, bekerjasama dengan perusahaan-perusahaan asing hingga sekarang ia mempunyai perusahaan sendiri yaitu PT. Callind Cakra Nusantara dengan produknya yaitu CakraTalk yang sudah tersedia di Playstore. Selanjutnya, pembicara kedua yaitu Aryo

Wiryawan menceritakan pengalamannya membangun JALA yang merupakan perusahaan pemberdayaan petambak ikan dengan teknologi. Ia menceritakan cara mengelola perusahaannya, melihat peluang bisnis hingga cara memasarkan produk pada pasar yang tepat.

3.1.3 Grow your tech Startup with Effective Brand Strategy

Acara yang diselenggarakan oleh CoHive ini mengusung tema “Grow your tech Startup with Effective Brand Strategy”. Menghadirkan pembicara Daniel Surya (CEO of WIR Group), WIR Group merupakan perusahaan teknologi digital AR, VR dan AI asal Indonesia yang berskala global, selain itu WIR Group juga merupakan pioneer perusahaan AR di Indonesia. Ia menceritakan awal ketika dia merintis bisnis di bidang *augmented reality* beberapa tahun lalu ketika teknologi tersebut masih asing di Indonesia. Banyak yang meremehkan idenya pada saat itu, namun ia membuktikannya dengan telah menyelesaikan lebih dari 1000 proyek di lebih dari 20 negara di dunia. Tak hanya itu, WIR Group juga telah bekerja sama dengan perusahaan-perusahaan raksasa dunia salah satunya Samsung, salah satu perusahaan elektronik terbesar di dunia. Pada akhir acara, Daniel Surya berpesan “berinovasi lah se-gila mungkin”.



3.1.4 Global Startup Weekend Sustainable Revolution



Gambar 3.3 Global Startup Weekend Sustainable Revolution

Menjelang akhir semester, semua tim perintisan bisnis diikutsertakan dalam acara Global Startup Weekend di GECO Surabaya selama 3 hari. Event tersebut diikuti oleh lebih dari 50 peserta seperti pada gambar 3.3. Para peserta yang hadir yaitu mulai dari siswa SMA/SMK, mahasiswa hingga para perintis *startup* yang telah bergerak pada bidangnya masing-masing. Dengan mengusung tema Sustainable Revolution, seluruh peserta dibimbing untuk menciptakan suatu produk/konsep bisnis *startup* yang bersifat *zero-waste*.

Pada hari pertama, seluruh peserta membentuk kelompok untuk melakukan *brainstorming* ide. Kelompok penulis sendiri menghasilkan ide yaitu pemanfaatan sampah dapur atau sisa makanan yang nanti akan diolah menjadi pupuk hingga pakan ternak. Selanjutnya ide tersebut dikonversikan ke dalam bisnis model kanvas khusus dengan tema *sustainable* yang sudah didesain oleh panitia.

Speakers and Coaches:



Tony Dwi Susanto
(Head of AISINDO)



Ricky Elson
(Founder Lentera Bumi Nusantara)



Agus Wiyono
(Head of EJEJ)



Dian Mukti Wicaksono
(Executive VP ICSB Surabaya)



Mufid Wahyudi
(Governor of ICSB JATIM)



Yoki Tyas Riesmana
(TopCoder MVP)



Avianto Tyo
(IOT Inovator)



Dedhy Trunoyudho
(COO Garda Pangan)



Ratu Fitri
(Founder SUBstitute)



Halim Sugianto
(CEO SEVIMA)

Gambar 3.4 Speakers dan coach Global Startup Weekend

Pada hari kedua, aktivitas lebih fokus pada pemantapan, *sharing* materi serta *coaching*. Pada acara ini didatangkan beberapa *coach* dari berbagai bidang seperti pada gambar 3.4. Setiap kelompok berkesempatan mendapatkan *coaching* 5 sampai 10 menit dari setiap *coach*. Kelompok penulis mendapatkan banyak sekali saran, terlebih kepada cara mendesain proses produksi yang efisien hingga mendapatkan *income*. Salah satu *coach* yang melakukan *coaching* pada kelompok penulis yaitu Ibu Ratu Fitri seperti pada gambar 3.5.



Gambar 3.5 Aktivitas coaching Global Startup Weekend

Hari terakhir Global Startup Weekend semua tim akan diberi kesempatan untuk melakukan *pitching deck* selama 3 menit dan tanya jawab dengan juri selama 3 menit. Juri yang dihadirkan pada kali ini yaitu Bapak Emil Elestianto Dardak, Bapak Jatmiko Wahyu Nugroho dan Bapak Achmad Zulkarnain seperti pada gambar 3.6. Tiga tim terbaik akan diinkubasi dan berkesempatan untuk ikut dalam acara Global Startup Weekend di Paris, Perancis.



Gambar 3.6 Juri Global Startup Weekend

Bertemu para tokoh hebat dan mendapatkan materi dari mereka merupakan pengalaman yang sangat berharga bagi kami ke depannya untuk mengembangkan ide bisnis khususnya dalam bidang *startup*.

3.2 Perancangan Back-End Sistem

Dalam membangun *back-end* Arenasia, konsep dan desain yang telah dirancang oleh *hustler* tersebut selanjutnya dikonversi menjadi *use case diagram*, *activity diagram* dan *entity relationship diagram*. Di sisi lain, hasil rancangan *front-end* oleh *hipster* juga akan diintegrasikan ke dalam *back-end* agar dapat menerima atau membaca data dari basis data dan dapat diakses secara daring di luar jaringan lokal oleh pengguna.

Arenasia yang merupakan *platform* kompetisi *esports* berbasis web didesain memiliki 3 jenis pengguna yaitu “*admin*” sebagai pengelola sistem, “*organizer*” sebagai pihak penyelenggara kompetisi dan “*player*” sebagai peserta kompetisi. *Organizer* didesain dapat mengadakan suatu kompetisi, mengelola suatu kompetisi dan melakukan transaksi penarikan saldo sedangkan *player* didesain dapat membuat tim, bergabung dengan tim, melakukan transaksi pengisian saldo tim dan mengikuti kompetisi. Sumber pendapatan utama *website* Arenasia berasal dari pemotongan saldo kompetisi sebesar 5% yang dibebankan kepada pihak *organizer*.

3.3 Metode Pengembangan

Rapid application development (RAD) merupakan metode pengembangan perangkat lunak yang digunakan oleh penulis dalam membangun web Arenasia. *Rapid application development* digunakan karena metode ini cocok dalam pembangunan web berskala kecil dan dalam waktu yang relatif singkat.



Gambar 3.7 *Rapid Application Development*

Sumber: (Kendall, 2010)

Gambar 3.7 merupakan skema metode *rapid application development*. Metode *rapid application development* memiliki 3 tahapan utama yaitu *Requirements Planning* (perencanaan syarat-syarat), *Workshop Design RAD* dan *Implementation*.

3.3.1 Requirements Planning

Pada fase ini dilakukan pengumpulan data dari pengguna sistem. Selain itu, dilakukan penyusunan rencana tentang hal-hal yang dibutuhkan dalam proses pembangunan sistem seperti *software* yang akan digunakan, pemilihan *framework* untuk mempermudah pembangunan, bahasa pemrograman yang dipakai dan informasi-informasi yang dibutuhkan sebagai dalam sistem. Adapun tahapan yang dilakukan pada fase ini yaitu:

- a. Penyusunan rencana kebutuhan untuk pembangunan web Arenasia seperti kebutuhan *software*, *framework*, bahasa pemrograman, informasi dan kebutuhan tambahan lainnya.

- b. Pemilihan ekstensi teks editor yang bertujuan untuk mempercepat aktivitas pemrograman.
- c. Penyusunan rencana kegiatan agar proses pembangunan terstruktur dan selesai dalam waktu yang diharapkan.

3.3.2 Workshop Design RAD

Workshop design RAD merupakan fase di mana perancangan sistem dilakukan, seperti perancangan *use case diagram*, *activity diagram*, dan *entity relationship diagram*. Pada tahap ini pengguna dapat dilibatkan dalam proses perancangan dan dapat memberikan *feedback* kepada pengembang sehingga memungkinkan terjadinya *pivot* pada fase ini. Hal ini dilakukan agar hasil akhir dari sistem sesuai dengan apa yang diharapkan oleh pengguna (Kendall, 2010). Berikut tahapan-tahapan yang dilakukan pada fase ini adalah sebagai berikut:

- a. Perancangan *use case diagram* untuk menentukan fungsi dan aktor yang terlibat pada sistem.
- b. Perancangan *activity diagram* untuk memetakan alur kerja setiap fungsi
- c. Perancangan *entity relationship diagram* sebagai langkah awal dalam perancangan skema basis data.

3.3.3 Implementation

Pada fase *implementation* ini dilakukan implementasi hasil dari perancangan sistem yang telah dilakukan pada fase *workshop design RAD* sebelumnya seperti implementasi skema basis data dalam bentuk *entity relationship diagram* menjadi basis data yang bisa menyimpan data pada sistem, menghubungkan antarmuka *website* dengan *controller* dan juga pengujian fungsional sistem dilakukan pada fase ini.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Penerapan Rapid Application Development

Ketika menggunakan metode *Rapid Application Development* (RAD) untuk membangun sebuah sistem, hal-hal yang perlu diperhatikan sekaligus menjadi batasan metode ini adalah meminimalisir biaya selama proses pembangunan, menyusun rencana kebutuhan yang dapat mempercepat pembangunan, sebisa mungkin memanfaatkan sistem yang sudah ada, bekerja dalam tim dengan membagi pekerjaan menjadi beberapa modul dan memperhatikan kecepatan pembangunan dengan rentang waktu 60 sampai 90 hari. Berikut adalah penerapan metode *rapid application development* dalam perancangan *back-end* Arenasia:

- a. Untuk meminimalisir biaya pembangunan, maka penulis menggunakan *software* gratis dan atau yang bersifat *open source*.
- b. Pada awal perancangan akan dipilih beberapa *software* dan *ekstensi* yang dapat mempercepat proses pembangunan sistem terutama untuk kebutuhan pemrograman.
- c. Mengintegrasikan beberapa sistem yang sudah ada dan atau *package* dalam *back-end* untuk mempercepat proses pembangunan.
- d. Pengerjaan web Arenasia dibagi menjadi 2 bagian yaitu *front-end* dan *back-end* yang kemudian digabungkan pada akhir pembangunan.
- e. Pembangunan web Arenasia direncanakan membutuhkan waktu 9 minggu, mulai dari penyusunan kebutuhan sampai pengujian *back-end* sistem.

4.2 Requirements Planning

Pada fase ini dilakukan konsultasi kepada calon pengguna dengan menyampaikan ide atau gambaran umum sistem ini. Pada fase ini penulis dan tim melakukan survei singkat kepada calon pengguna platform kompetisi *esports* yaitu para pemain game aktif menggunakan media Google Formulir. Dari hasil survei tersebut, didapatkan kesimpulan sebagai berikut:

- a. Memprioritaskan game dengan perangkat mobile dan PC.
- b. Memilih Mobile Legends, Dota 2, Counter Strike: Global Offensive dan Playerunknown's Battlegrounds mobile sebagai game yang dimuat pada platform kompetisi *esports*.
- c. Menyediakan fitur membentuk dan mencari tim.

4.2.1 Rencana Kebutuhan

Untuk membangun sebuah web dibutuhkan beberapa *software*, *framework*, bahasa pemrograman dan informasi, maka dari itu penulis membuat daftar rencana kebutuhan yang nantinya akan digunakan selama pembangunan *platform* kompetisi *esports* berbasis web. Tiap rencana kebutuhan yang dipilih didasarkan pada kemudahan dalam penggunaan dan efisiensi waktu. Rencana kebutuhan tersebut adalah sebagai berikut:

- f. Kebutuhan *Software*
 - 4. Windows 10
 - 5. Google Chrome
 - 6. Visual Studio Code
 - 7. XAMPP Control Panel
 - 8. MySQL
 - 9. Lucidchart
- g. Kebutuhan *Framework*
 - 1. Laravel
 - 2. Bootstrap
- h. Kebutuhan Bahasa Pemrograman
 - 1. HTML
 - 2. CSS
 - 3. PHP
 - 4. JavaScript
- i. Kebutuhan Tambahan
 - 1. Spatie
 - 2. Socialite
 - 3. Midtrans
 - 4. Dusk
- j. Kebutuhan Informasi
 - 1. Informasi kompetisi *esports* di Indonesia
 - 2. Informasi game *esports*
 - 3. Data provinsi beserta kabupaten di Indonesia



4.2.2 Kebutuhan Ekstensi Teks Editor

Selain itu, guna mempercepat proses penulisan kode pemrograman pada saat pembangunan web, maka pada teks editor Visual Studio Code dipasangkan beberapa *ekstensi* yang dapat dilihat pada tabel 4.1 berikut.

Tabel 4.1 Ekstensi dalam Visual Studio Code

<i>Ekstensi</i>	Fungsi
Laravel Artisan	Sebagai alternatif dalam mempercepat menjalankan perintah <i>php artisan</i> .
Laravel Snippets	Mempercepat penulisan kode <i>framework</i> Laravel tanpa harus menuliskan kode secara lengkap, hanya dengan menuliskan <i>prefix</i> seperti Auth, Config, DB, Route, View dan lainnya.
Laravel Blade Snippets	Mempercepat penulisan sintaks-sintaks pada bagian blade dalam Laravel.
Laravel Goto View	Mempercepat pengaksesan <i>file view</i> dalam Laravel hanya dengan melakukan klik pada baris kode yang merujuk pada <i>file view</i> .
Laravel Goto Controller	Mempercepat pengaksesan <i>file controller</i> dalam Laravel hanya dengan melakukan klik pada baris kode yang merujuk pada <i>file controller</i> .
PHP Intelephense	Mempercepat penulisan sintaks PHP dengan fitur <i>code completion</i> yang disertai saran sintaks.
MJML	Memudahkan dalam penulisan dan desain kerangka <i>email MJML</i> karena dapat langsung melihat hasilnya di dalam Visual Studio Code.

4.2.3 Rencana Kegiatan

Dalam metode *rapid application development* siklus pembangunan yang singkat merupakan batasan yang penting. Agar proses pembangunan *platform* kompetisi *esports* ini selesai dalam kurun waktu antara 60 sampai 90 hari (Kendall, 2010) merujuk pada metode *rapid application development*, maka dibuat penjadwalan rencana kegiatan seperti pada tabel 4.2 berikut.

Tabel 4.2 Rencana kegiatan perancangan *back-end*

No.	Proses	Bulan/Minggu											
		Maret				April				Mei			
		1	2	3	4	1	2	3	4	1	2	3	4
1.	<i>Requirements Planning</i>	■											
2.	<i>Workshop design RAD</i>												
	1. Perancangan <i>use case diagram</i> dan <i>activity diagram</i>		■										
	2. Perancangan <i>entity relationship diagram</i>			■									
3.	<i>Implementation</i>												
	1. Pembuatan <i>autentikasi login, logout</i> dan <i>register</i>					■							
	2. Pembuatan fitur untuk <i>admin</i>						■						
	3. Pembuatan fitur untuk <i>organizer</i>							■					
	4. Pembuatan fitur untuk <i>player</i>								■				
	5. Integrasi antarmuka dengan <i>controller</i>									■			
	6. Pengujian										■		

Rencana kegiatan disusun berdasarkan fase pada metode *rapid application development* yaitu *requirements planning*, *workshop design RAD* dan *implementation*. Fase *requirements planning* direncanakan membutuhkan waktu pengerjaan selama 1 minggu yang meliputi proses perencanaan kebutuhan, perencanaan ekstensi dan perencanaan kegiatan. Fase *workshop design RAD* direncanakan membutuhkan waktu pengerjaan selama 3 minggu yang meliputi proses perancangan desain awal *back-end* berdasarkan analisis model. Hasil perancangan desain awal *back-end* selanjutnya ditinjau oleh tim pengembang, jika desain belum disepakati

maka akan dilakukan perbaikan desain pada fase ini. Fase *implemmentation* direncanakan membutuhkan waktu pengerjaan selama 5 minggu yang meliputi proses pemrograman hingga pengujian sistem.

4.3 Workshop Design RAD

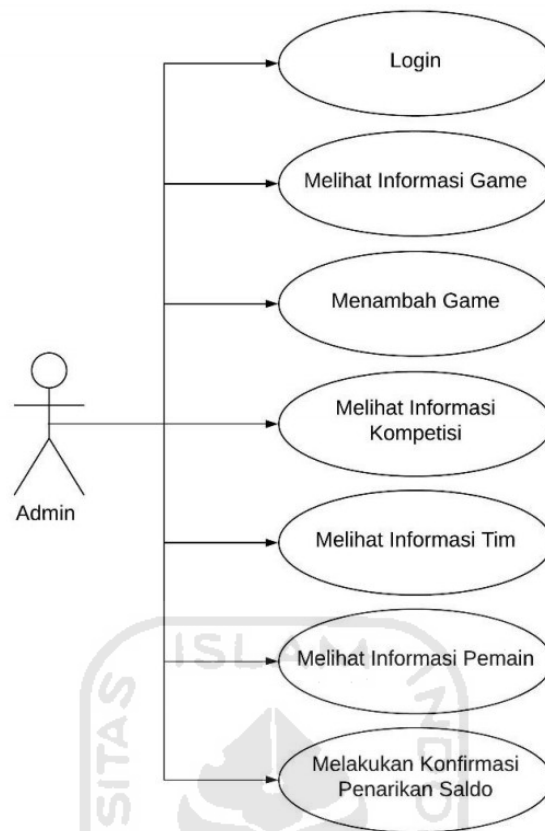
Pada fase *Workshop Design RAD*, dilakukan perancangan desain sistem yaitu *use case diagram*, *activity diagram* dan desain *entity relationship diagram*. Hal ini bertujuan untuk memudahkan proses pembangunan karena hasil-hasil dari fase ini menjadi dasar atau pedoman pembangunan sehingga membuat pembangunan lebih terstruktur dan tidak keluar dari pokok permasalahan.

4.3.1 Use Case Diagram

Use case diagram berfungsi untuk mendefinisikan aktor-aktor dengan relasi terhadap fungsi-fungsi yang ada pada sistem sebagai alur proses kerja sistem. Pada sistem *platform* kompetisi *esports* ini terdapat 3 aktor yaitu *admin*, *organizer* dan *player*. Setiap aktor mempunyai hak akses masing-masing terhadap suatu fungsi dalam sistem. Perancangan *use case diagram* dilakukan menggunakan *software* *lucidchart*.

a. *Use Case Diagram* untuk *Admin*

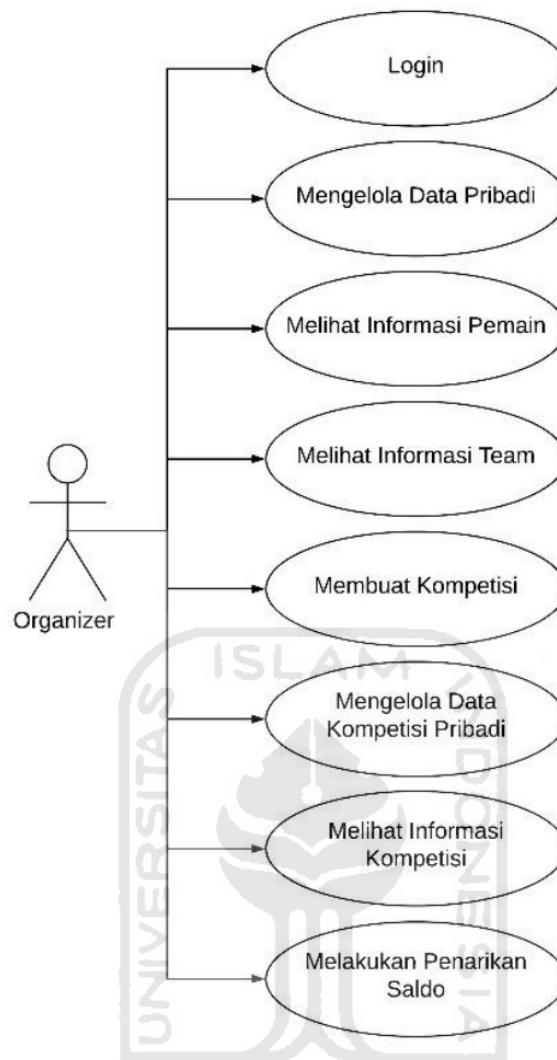
Admin mempunyai hak akses untuk beberapa fungsi atau halaman pada web Arenasia yaitu melakukan *login*, melihat informasi *game*, menambah *game*, melihat informasi kompetisi, melihat informasi tim, melihat informasi pemain dan melakukan konfirmasi penarikan saldo yang dilakukan oleh *organizer*. *Diagram use case admin* dapat dilihat pada gambar 4.1 berikut.



Gambar 4.1 *Use Case Diagram Admin*

b. *Use Case Diagram untuk Organizer*

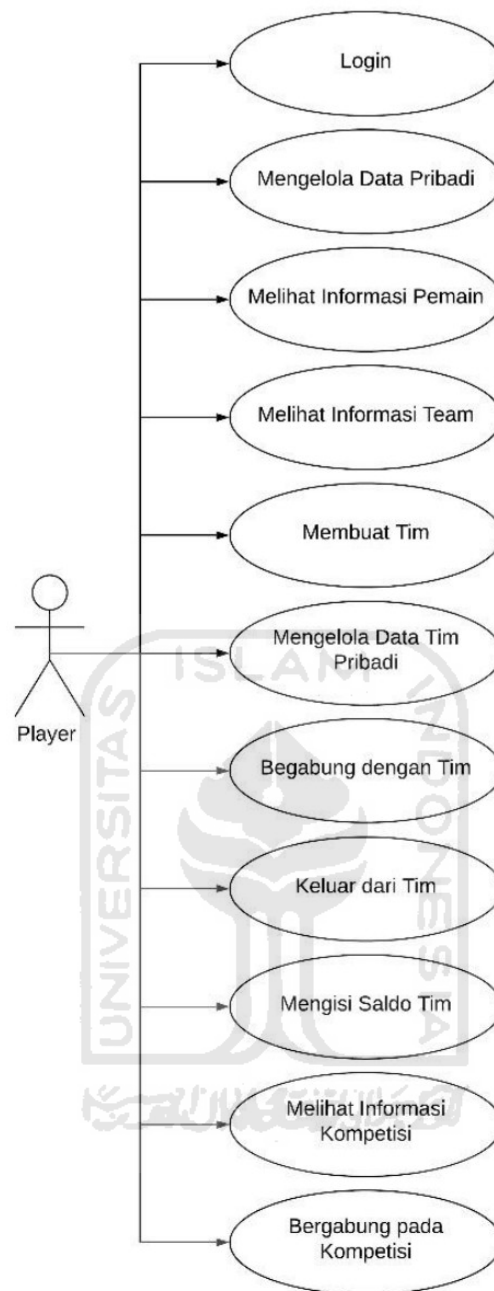
Organizer mempunyai hak akses untuk beberapa fungsi atau halaman pada web Arenasia yaitu melakukan *login*, mengelola data pribadi, melihat informasi kompetisi, melihat informasi tim, membuat kompetisi, mengelola data kompetisi pribadi, melihat informasi kompetisi dan melakukan penarikan saldo. *Diagram use case organizer* dapat dilihat pada gambar 4.2 berikut.



Gambar 4.2 Use Case Diagram Organizer

c. Use Case Diagram untuk Player

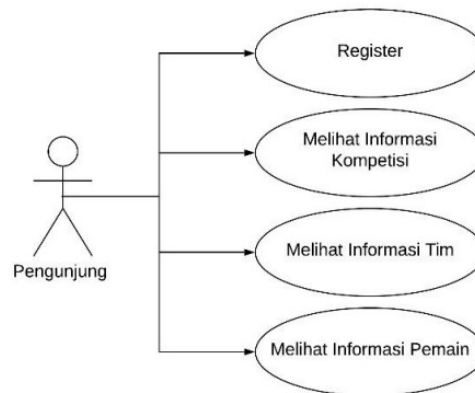
Player mempunyai hak akses untuk beberapa fungsi atau halaman pada web Arenasia yaitu melakukan *login*, mengelola data pribadi, melihat data pemain, melihat informasi tim, membuat tim, mengelola data tim pribadi, bergabung dengan tim, keluar dari tim, mengisi saldo tim, melihat informasi kompetisi dan bergabung pada kompetisi. Diagram use case player dapat dilihat pada gambar 4.3 berikut.



Gambar 4.3 *Use Case Diagram Player*

d. *Use Case Diagram* untuk Pengunjung

Pengunjung web Arenasia mempunyai hak akses untuk beberapa fungsi atau halaman pada web Arenasia yaitu melakukan registrasi, melihat informasi kompetisi, dan melihat informasi tim dan melihat informasi pemain. *Diagram use case* pengunjung dapat dilihat pada gambar 4.4 berikut.



Gambar 4.4 *Use Case Diagram* Pengunjung

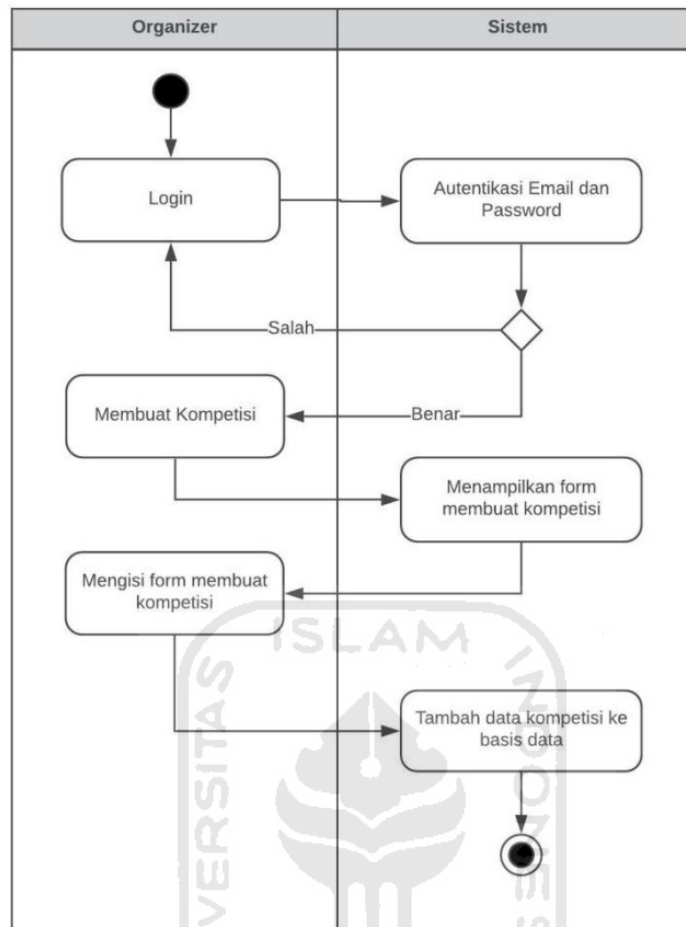
4.3.2 Activity Diagram

Activity diagram merupakan diagram yang menggambarkan proses alur kerja suatu fungsi pada sistem secara urut. *Activity diagram* dapat dimodelkan secara terpisah, artinya tiap *activity diagram* memodelkan masing-masing fitur pada sebuah sistem.

Pembangunan sistem Arenasia memanfaatkan beberapa fitur atau fungsi bawaan dari Laravel dan *package* pihak ketiga seperti *login* dengan provider, *register*, *logout*, *reset password* dan lainnya. Maka *activity diagram* yang perlu dirancang adalah fitur yang dibangun secara manual, berikut beberapa fitur tersebut:

a. Membuat Kompetisi

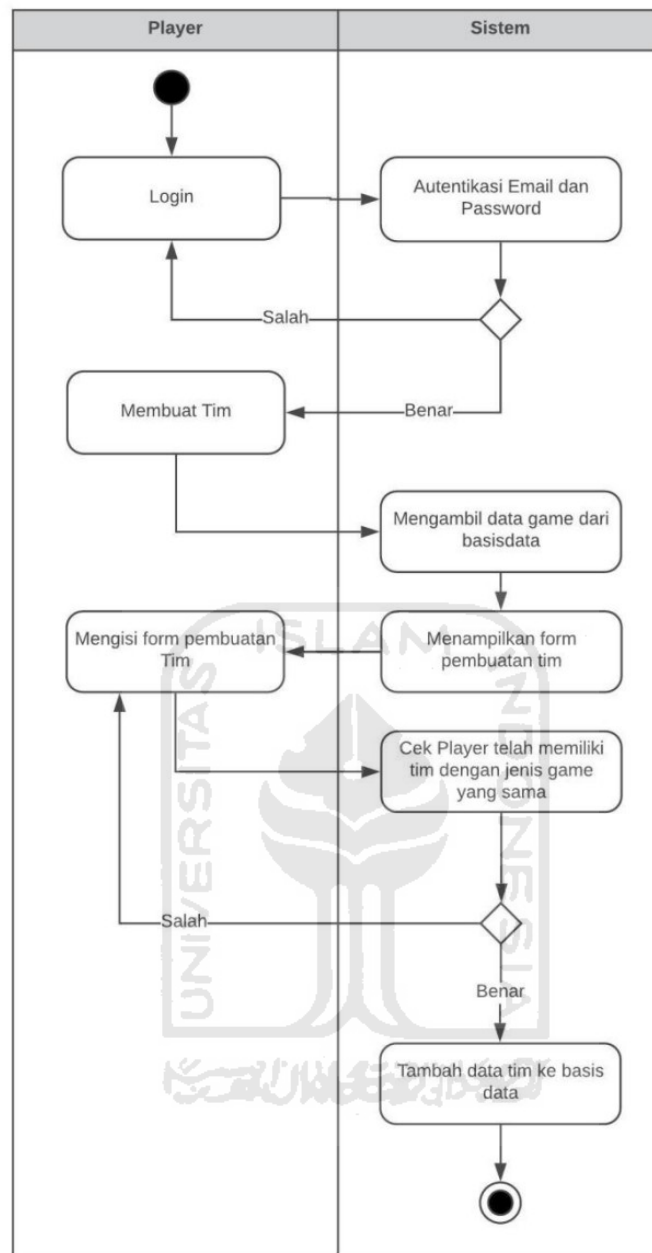
Untuk membuat suatu kompetisi, *organizer* harus melakukan *login* terlebih dahulu untuk mendapatkan hak akses terhadap fungsi membuat kompetisi. Jika *autentikasi login* gagal maka sistem akan menampilkan halaman *login* dan jika benar maka sistem akan mengizinkan *organizer* untuk melakukan aksi membuat kompetisi. Ketika *organizer* memilih menu membuat kompetisi, maka sistem akan menampilkan *form* pembuatan kompetisi yang selanjutnya harus diisi oleh *organizer*. Setelah berhasil maka sistem akan menambahkan data kompetisi ke basis data. *Activity diagram* proses membuat kompetisi dapat dilihat pada gambar 4.5 berikut.



Gambar 4.5 Activity diagram membuat kompetisi

b. Membuat Tim

Jika seseorang ingin membuat tim, harus melakukan *login* terlebih dahulu sebagai *player* untuk mendapatkan hak akses membuat tim. Jika *autentikasi login* gagal sistem akan menampilkan halaman *login* dan jika benar maka sistem akan mengizinkan *player* untuk melakukan aksi membuat tim. Ketika *player* memilih menu membuat tim, sistem akan menampilkan *form* pembuatan tim yang selanjutnya harus diisi oleh *player*. Selanjutnya *form* yang telah diisi oleh *player* akan diperiksa oleh sistem. Jika *player* telah memiliki tim dengan jenis *game* yang sama maka aksi membuat tim gagal dan sistem akan mengarahkan *player* kembali ke halaman *form* pembuatan tim dan jika berhasil maka sistem akan menambahkan data tim pada basis data. Activity diagram proses membuat tim dapat dilihat pada gambar 4.6 berikut.



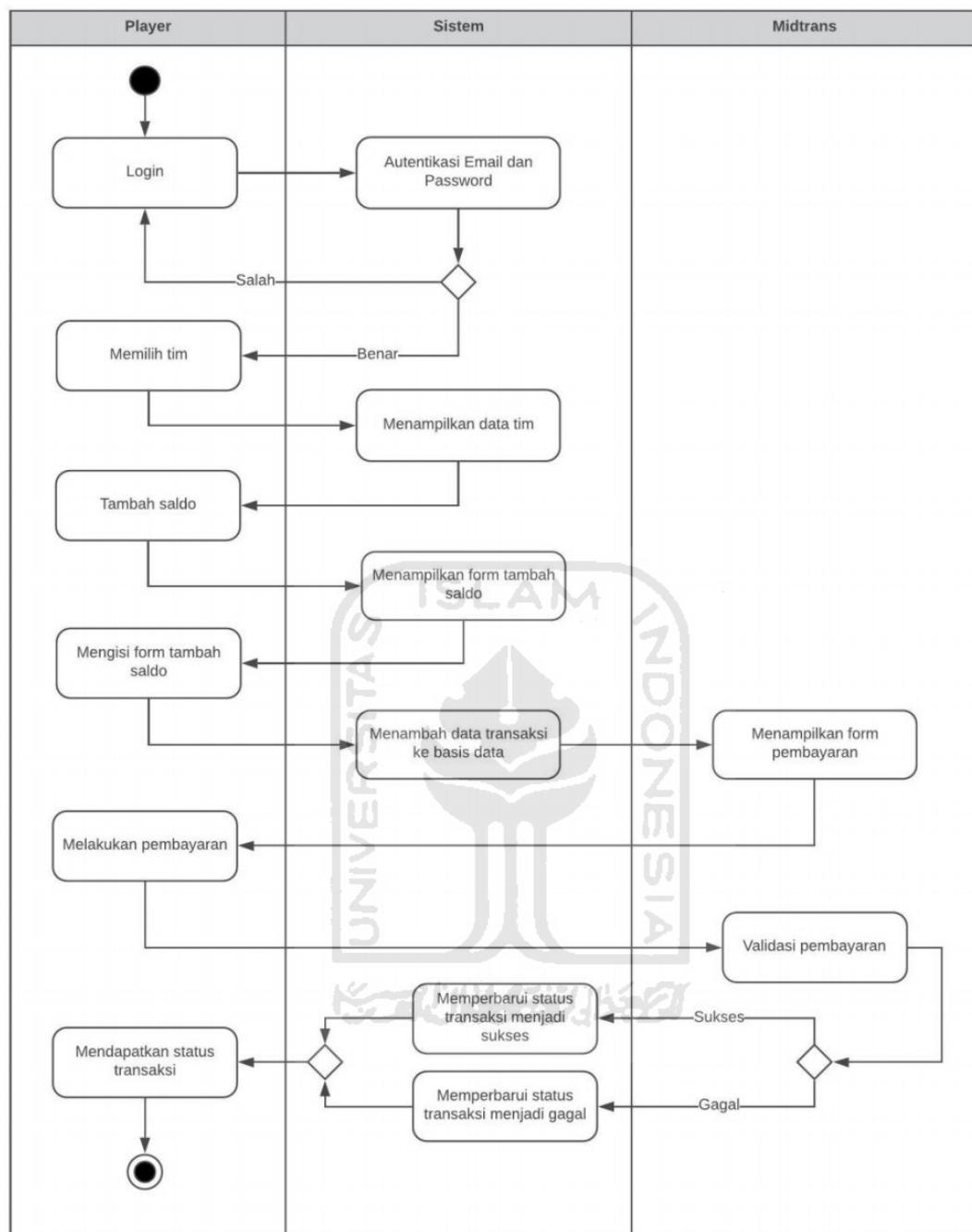
Gambar 4.6 Activity diagram membuat tim

c. Melakukan Pengisian Saldo Tim

Jika seseorang ingin mengisi saldo tim, maka harus melakukan *login* terlebih dahulu sebagai *player* untuk mendapatkan hak akses pengisian saldo tim. Jika *autentikasi login* gagal maka sistem akan menampilkan halaman *login* dan jika benar maka sistem akan mengizinkan *player* untuk melakukan aksi pengisian tim. Untuk mengisi saldo tim, *player* terlebih dahulu memilih tim yang dimiliki dan sistem akan menampilkan data tim

yang dipilih. Selanjutnya *player* dapat melakukan aksi pengisian saldo tim dan sistem akan menampilkan *form* pengisian saldo yang harus diisi oleh *player*. Setelah itu sistem akan menambahkan data transaksi pada basis data dengan status *pending* dan mengarahkan *player* untuk menyelesaikan transaksi melalui *payment gateway* Midtrans, aktivitas transaksi akan ditangani oleh Midtrans dan akan mengembalikan status transaksi kepada sistem untuk diperbarui pada basis data. *Activity diagram* proses pengisian saldo tim dapat dilihat pada gambar 4.7 berikut.





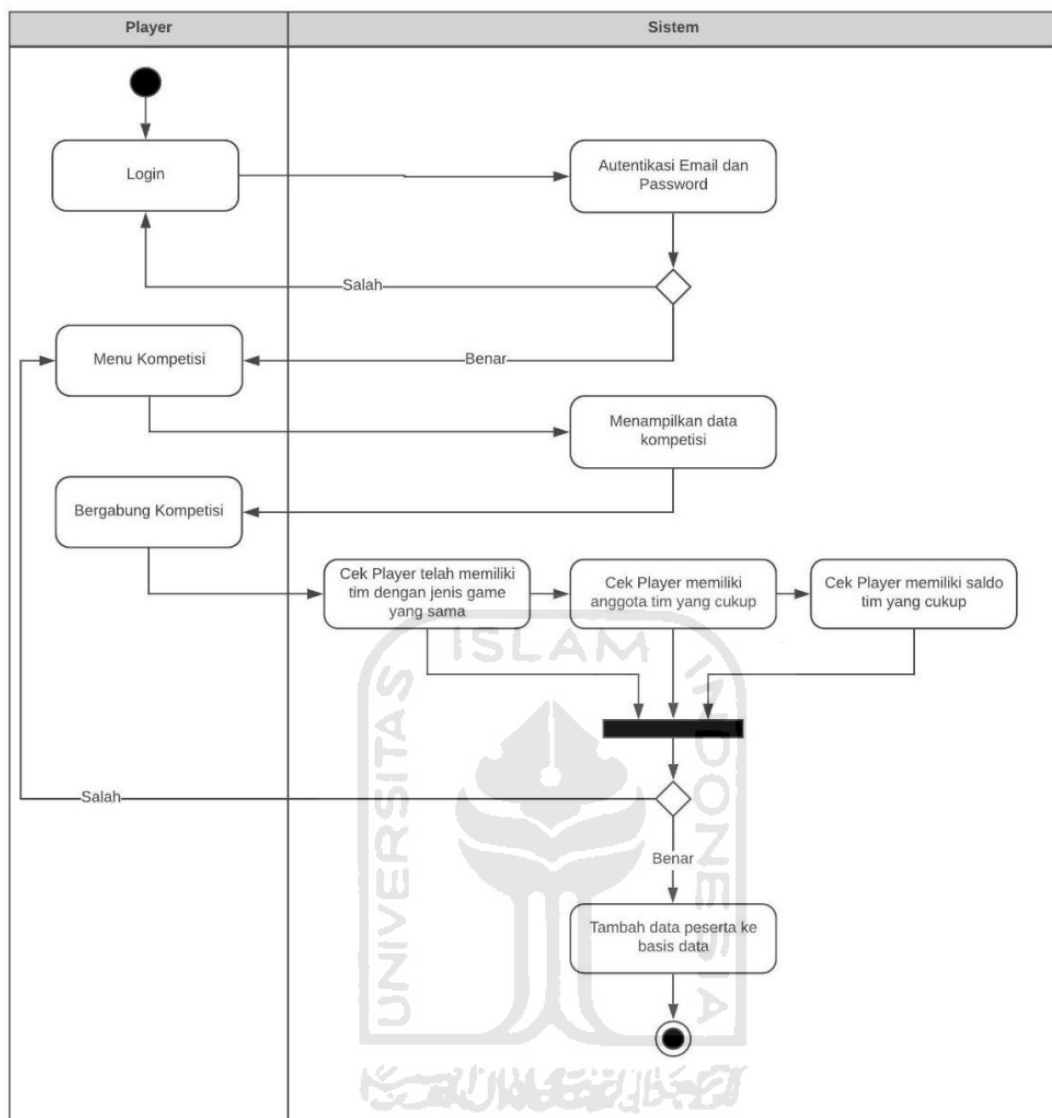
Gambar 4.7 Activity diagram pengisian saldo tim

d. Bergabung Kompetisi

Jika seseorang ingin mengikuti suatu kompetisi, maka harus melakukan *login* terlebih dahulu sebagai *player* untuk mendapatkan hak akses bergabung kompetisi. Jika *autentikasi login* gagal maka sistem akan menampilkan halaman *login* dan jika benar maka sistem akan mengizinkan *player* untuk melakukan aksi bergabung kompetisi.

Untuk mengikuti suatu kompetisi, terlebih dahulu *player* harus memilih suatu kompetisi dan sistem akan menampilkan data kompetisi yang dipilih. Ketika *player* melakukan aksi bergabung kompetisi, maka sistem akan memvalidasi *player* dengan 3 kondisi yaitu, memeriksa apakah *player* memiliki tim dengan jenis *game* yang dikompetisikan, apakah *player* memiliki anggota yang cukup dalam timnya dan tim memiliki saldo yang cukup untuk melakukan pendaftaran. Jika salah satu kondisi gagal maka sistem akan mengarahkan *player* kembali ke halaman memilih kompetisi dan jika berhasil maka sistem akan menambahkan data peserta pada basis data. *Activity diagram* proses membuat tim dapat dilihat pada gambar 4.8 berikut.

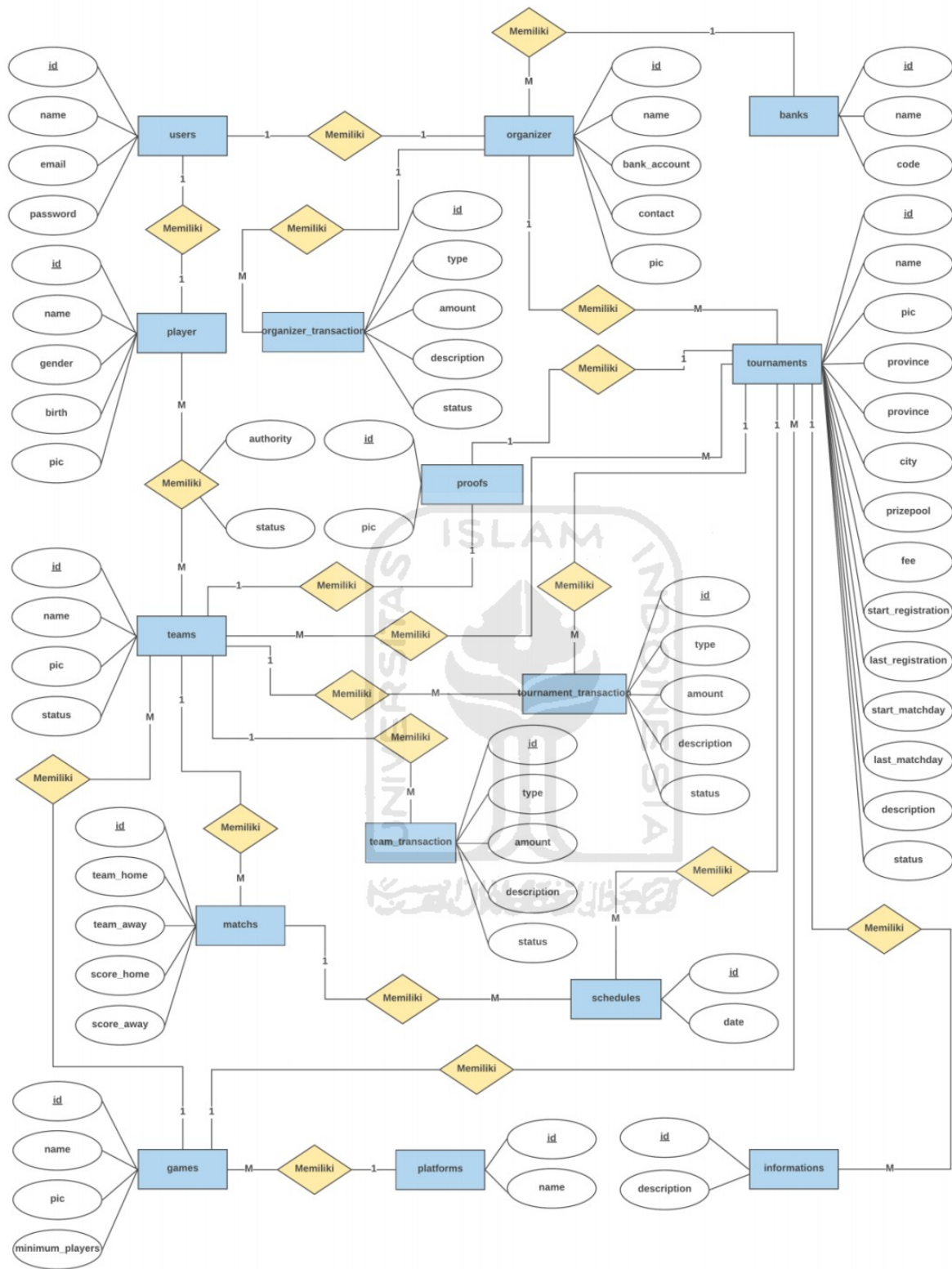




Gambar 4.8 Activity diagram bergabung kompetisi

4.3.3 Desain Entity Relationship Diagram

Entity relationship diagram merupakan diagram yang memuat fungsi-fungsi utama pada suatu sistem yang didefinisikan dalam bentuk entitas. Masing-masing entitas memiliki atribut dan terdapat relasi yang menghubungkan entitas satu dengan entitas lainnya. Perancangan *entity relationship diagram* ini berfungsi untuk memudahkan dalam pembuatan basis data nantinya. Perancangan *diagram entity relationship* dilakukan menggunakan *software* Lucidchart. *Diagram entity relationship* dapat dilihat pada gambar 4.9 berikut.



Gambar 4.9 Entity Relationship Diagram

Terdapat 14 entitas dengan atributnya masing-masing. Entitas-entitas tersebut yaitu users, players, organizers, games, platforms, teams, tournaments, informations, schedules,

matches, proofs, organizer_transaction, teams_transaction dan tournaments_transaction. Setiap entitas memiliki relasi dengan entitas lainnya, relasi tersebut seperti pada tabel 4.3 berikut.

Tabel 4.3 Relasi entitas

Entitas 1	Entitas 2	Relasi
users	organizers	One to one
	players	One to one
organizers	users	One to one
	tournaments	One to many
	organizer_transaction	One to many
	banks	Many to one
players	users	One to one
	teams	Many to many
banks	organizers	One to many
platforms	games	One to many
games	platforms	Many to one
	teams	One to many
	tournaments	One to many
teams	players	Many to many
	tournaments	Many to many
	tournament_transaction	One to many
	team_transaction	One to many
	match	One to many
	games	Many to one
	proofs	One to one
tournaments	organizers	Many to one
	informations	One to many
	games	Many to one
	schedules	One to many
	tournament_transaction	One to many
	teams	Many to many
	proofs	One to one

informations	tournaments	Many to one
matchs	teams	Many to one
	schedules	One to many
schedules	tournaments	Many to one
	teams	Many to one
organizer_transaction	organizers	Many to one
proofs	tournaments	One to one
	teams	One to one
tournament_transaction	tournaments	Many to one
	teams	Many to one
team_transaction	teams	Many to one

4.4 Implementation

Setelah proses desain sistem selesai dilakukan dengan persetujuan oleh calon pengguna, penulis melakukan perancangan *back-end platform* kompetisi *esports* berbasis web ini menggunakan *framework* PHP Laravel. Mulai dari implementasi desain *entity relationship diagram* ke dalam bentuk basis data MySQL, penjelasan antarmuka beserta kode *back-end* web dan pengujian fungsional sistem yang dilakukan ketika sistem telah selesai dibangun secara keseluruhan menggunakan Dusk.

Dalam membangun web Arenasia, penulis menggunakan *framework* Laravel. Di mana pembangunan web dengan *framework* Laravel menggunakan konsep *Model-View-Controller*. *Model* merupakan bagian yang menangani koneksi dengan tabel-tabel pada basis data, *view* merupakan bagian yang menangani halaman-halaman yang akan ditampilkan pada web dan *controller* merupakan bagian yang menangani logika pada web seperti *request*, *response* dan *query*. Selain bersifat *open source*, Laravel juga memiliki dokumentasi yang lengkap, struktur yang mudah dimengerti serta memiliki banyak *package* yang dapat membantu mempercepat proses pembangunan sistem.

4.4.1 Basis Data

Merujuk pada Entity Relationship Diagram yang telah dirancang pada tahap *workshop design RAD*, maka penulis membuat basis data MySQL bernama Arenasia. Untuk menghubungkan basis data Arenasia dengan proyek Laravel maka dilakukan

pengkonfigurasiannya pada *file .env*. Isi dari *file .env* yang telah dikonfigurasi seperti pada gambar 4.10 berikut.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=skripsi
DB_USERNAME=root
DB_PASSWORD=
```

Gambar 4.10 Konfigurasi *file .env*

Gambar 4.10 menjelaskan bahwa tipe basis data yang digunakan yaitu MySQL dengan *host* 127.0.0.1 dan *port* 3306 sebagai *localhost* untuk mengakses proyek. Pada MySQL digunakan basis data bernama Arenasia dengan pengaturan *username* root tanpa password. Selanjutnya tiap-tiap entitas pada *entity relationship diagram* dikonversikan menjadi tabel pada basis data Arenasia menggunakan fitur migrasi pada Laravel. Hasil dari migrasi adalah sebagai berikut:

a. Tabel users

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

Gambar 4.11 Migrasi tabel users

Gambar 4.11 merupakan isi dari file migrasi untuk tabel users yang menghasilkan tabel users pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.4 berikut.

Tabel 4.4 Struktur tabel users

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
name	VARCHAR	191	
email	VARCHAR	191	
email_verified_at	DATE		
password	VARCHAR	191	
remember_token	VARCHAR	100	
created_at	DATE		
updated_at	DATE		

Tabel users berfungsi untuk menyimpan data khusus untuk autentikasi seluruh pengguna yaitu *admin*, *organizer* dan *player*. Tabel users memiliki beberapa kolom yaitu 'id' sebagai *primary key*, 'email' yang berfungsi untuk menyimpan *email*, 'email_verified_at' berfungsi untuk menyimpan status terkonfirmasi *email* dalam bentuk tanggal, 'password' berfungsi untuk menyimpan sandi yang telah melalui proses *hash* sehingga *autentikasi* menjadi lebih aman, *created_at* berfungsi menyimpan tanggal dibuatnya data dan *updated_at* untuk menyimpan tanggal terakhir data dimodifikasi.

b. Tabel organizers

```
public function up()
{
    Schema::create('organizers', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('user_id');
        $table->foreign('user_id')->references('id')->on('users')
            ->onDelete('cascade');
        $table->string('name')->unique();
        $table->unsignedBigInteger('bank_id');
        $table->foreign('bank_id')->references('id')->on('banks')
            ->onDelete('cascade');
        $table->string('bank_account');
        $table->string('contact');
        $table->string('slug');
        $table->string('pic')->default('default.jpg');
        $table->timestamps();
    });
}
```

Gambar 4.12 Migrasi tabel organizers

Gambar 4.12 merupakan isi dari file migrasi untuk tabel organizers yang menghasilkan tabel organizers pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.5 berikut.

Tabel 4.5 Struktur tabel organizers

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
user_id	BIGINTEGER	20	Foreign key
name	VARCHAR	191	
bank_id	BIGINTEGER	20	Foreign key
bank_account	VARCHAR	191	
contact	VARCHAR	191	
pic	VARCHAR	191	
slug	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel organizers berfungsi untuk menyimpan data users yang memiliki role *organizers* dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'user_id' sebagai *foreign key* dari tabel users, 'name' berfungsi untuk menyimpan nama dari *organizers*, 'bank_id' sebagai *foreign key* dari tabel banks, 'bank_account' berfungsi untuk menyimpan nomor rekening *organizer*, 'pic' berfungsi untuk menyimpan nama *file* dari foto profil dengan *default file* bernama 'default.jpg', 'slug' merupakan hasil *parsing* dari kolom name yang digunakan untuk mempercantik URL konten, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

c. Tabel players

```

public function up()
{
    Schema::create('players', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('user_id');
        $table->foreign('user_id')->references('id')->on('users')
            ->onDelete('cascade');
        $table->string('name');
        $table->enum('gender', ['Male', 'Female']);
        $table->date('birth');
        $table->string('pic')->default('default.jpg');
        $table->string('slug');
        $table->timestamps();
    });
}

```

Gambar 4.13 Migrasi tabel players

Gambar 4.13 merupakan isi dari file migrasi untuk tabel players yang menghasilkan tabel players pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.6 berikut.

Tabel 4.6 Struktur tabel players

Atribut	Type Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
users_id	BIGINTEGER	20	Foreign key
name	VARCHAR	191	
gender	ENUM	('Male', 'Female')	
birth	DATE		
pic	VARCHAR	191	
slug	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel players berfungsi untuk menyimpan data *users* yang memiliki role *organizers* dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'users_id' sebagai *foreign key* dari tabel users, 'name' berfungsi untuk menyimpan nama dari organizers, 'gender' berfungsi untuk menyimpan jenis kelamin, 'birth' berfungsi untuk menyimpan tanggal lahir, 'pic' berfungsi untuk menyimpan nama *file* dari foto profil *organizers* dengan *default file*

bernama 'default.jpg', 'slug' merupakan hasil *parsing* dari kolom name yang digunakan untuk mempercantik URL konten, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

d. Tabel banks

```
public function up()
{
    Schema::create('banks', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('code')->unique();
        $table->timestamps();
    });
}
```

Gambar 4.14 Migrasi tabel banks

Gambar 4.14 merupakan isi dari file migrasi untuk tabel banks yang menghasilkan tabel banks pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.7 berikut.

Tabel 4.7 Struktur tabel banks

Atribut	Type Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
name	VARCHAR	191	
code	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel banks berfungsi untuk menyimpan data bank dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'name' berfungsi untuk menyimpan nama dari bank, 'code' berfungsi untuk menyimpan kode bank, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

e. Tabel platforms

```

public function up()
{
    Schema::create('platforms', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('slug')->unique();
        $table->softDeletes();
        $table->timestamps();
    });
}

```

Gambar 4.15 Migrasi tabel platforms

Gambar 4.15 merupakan isi dari file migrasi untuk tabel games yang menghasilkan tabel games pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.8 berikut.

Tabel 4.8 Struktur tabel platforms

Atribut	Type Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
name	VARCHAR	191	
slug	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel platforms berfungsi untuk menyimpan data *platform* dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'name' berfungsi untuk menyimpan nama dari *game*, 'slug' merupakan hasil *parsing* dari kolom name yang digunakan untuk mempercantik URL konten, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

f. Tabel games

```

public function up()
{
    Schema::create('games', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('slug');
        $table->unsignedBigInteger('platform_id');
        $table->foreign('platform_id')->references('id')
            ->on('platforms');
        $table->string('pic');
        $table->integer('minimum_players');
        $table->timestamps();
    });
}

```

Gambar 4.16 Migrasi tabel games

Gambar 4.16 merupakan isi dari file migrasi untuk tabel games yang menghasilkan tabel games pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.9 berikut.

Tabel 4.9 Struktur tabel games

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
platform_id	BIGINTEGER	20	Foreign key
name	VARCHAR	191	
pic	VARCHAR	191	
minimum_players	INTEGER	11	
slug	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel games berfungsi untuk menyimpan data *game* yang dikompetisikan oleh Arenasia dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'platform_id' sebagai *foreign key* dari tabel platforms, 'name' berfungsi untuk menyimpan nama dari *game*, 'pic' berfungsi untuk menyimpan nama *file* dari foto *game*, 'minimum_players' berfungsi untuk menyimpan jumlah pemain minimal dari suatu *game* untuk mengikuti kompetisi dengan *game* terkait, 'slug' merupakan hasil *parsing* dari kolom name yang digunakan untuk

mempercantik URL konten, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

g. Tabel teams

```
public function up()
{
    Schema::create('teams', function (Blueprint $table) {
        $table->id();
        $table->string('name')->unique();
        $table->string('slug');
        $table->string('pic');
        $table->unsignedBigInteger('game_id');
        $table->foreign('game_id')->references('id')->on('games')
            ->onDelete('cascade');
        $table->string('status')->default('public');
        $table->timestamps();
    });
}
```

Gambar 4.17 Migrasi tabel teams

Gambar 4.17 merupakan isi dari file migrasi untuk tabel teams yang menghasilkan tabel teams pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.10 berikut.

Tabel 4.10 Struktur tabel teams

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
name	VARCHAR	191	
slug	VARCHAR	191	
pic	VARCHAR	191	
game_id	BIGINTEGER	20	Foreign key
status	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel teams berfungsi untuk menyimpan data dari tim yang dibuat oleh *player* pada web dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'name' berfungsi untuk menyimpan nama dari suatu tim, 'pic' berfungsi untuk menyimpan nama *file* dari logo tim, 'game_id' sebagai *foreign key* dari tabel games, 'status' berfungsi untuk menyimpan

status apakah tim tersebut membuka atau menutup akses bergabung terhadap seorang player, 'slug' merupakan hasil *parsing* dari kolom name yang digunakan untuk mempercantik URL konten, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

h. Tabel tournaments

```
public function up()
{
    Schema::create('tournaments', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('organizer_id');
        $table->foreign('organizer_id')->references('id')
            ->on('organizers')->onDelete('cascade');
        $table->string('name')->unique();
        $table->string('slug');
        $table->unsignedBigInteger('game_id');
        $table->foreign('game_id')->references('id')->on('games')
            ->onDelete('cascade');
        $table->string('pic');
        $table->string('province');
        $table->string('city');
        $table->string('prizepool');
        $table->integer('fee');
        $table->tinyInteger('slot');
        $table->date('start_registration');
        $table->date('last_registration');
        $table->date('start_matchday');
        $table->date('last_matchday');
        $table->text('description');
        $table->string('status');
        $table->timestamps();
    });
}
```

Gambar 4.18 Migrasi tabel tournaments

Gambar 4.18 merupakan isi dari file migrasi untuk tabel tournaments yang menghasilkan tabel tournaments pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.11 berikut.

Tabel 4.11 Struktur tabel tournaments

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
organizer_id	BIGINTEGER	20	Foreign key
name	VARCHAR	191	
slug	VARCHAR	191	
game_id	BIGINTEGER	20	Foreign key
pic	VARCHAR	191	
province	VARCHAR	191	
city	VARCHAR	191	
prizepool	VARCHAR	191	
fee	INTEGER	11	
slot	INTEGER	11	
start_registration	DATE		
last_registration	DATE		
start_matchday	DATE		
last_matchday	DATE		
description	TEXT		
status	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel tournaments berfungsi untuk menyimpan data suatu kompetisi yang telah dibuat oleh *organizer* pada web Arenasia dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'organizer_id' sebagai *foreign key* dari tabel organizers yang berfungsi untuk menyimpan id pembuat kompetisi, 'name' berfungsi untuk menyimpan nama kompetisi, 'slug' merupakan hasil *parsing* dari kolom name yang digunakan untuk mempercantik URL konten, 'game_id' sebagai *foreign key* dari tabel games, 'pic' berfungsi untuk menyimpan nama *file* dari poster kompetisi, 'province' berfungsi untuk menyimpan lokasi provinsi kompetisi, 'city' berfungsi untuk menyimpan lokasi kabupaten kompetisi, 'prizepool' berfungsi untuk menyimpan total hadiah yang disediakan dalam suatu kompetisi, 'fee' berfungsi untuk menyimpan biaya pendaftaran kompetisi, 'slot' berfungsi

untuk menyimpan jumlah tim yang dapat mengikuti kompetisi, 'start_registration' berfungsi untuk menyimpan tanggal mulai pendaftaran kompetisi, 'last_registration' berfungsi untuk menyimpan tanggal terakhir pendaftaran kompetisi, 'start_matchday' berfungsi untuk menyimpan tanggal mulai kompetisi, 'last_matchday' berfungsi untuk menyimpan tanggal terakhir kompetisi, 'description' berfungsi untuk menyimpan deskripsi kompetisi, 'status' berfungsi untuk menyimpan status kompetisi apakah kompetisi sedang dalam fase pendaftaran, kompetisi sedang berlangsung, kompetisi telah selesai atau kompetisi mendatang, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

i. Tabel informations

```
public function up()
{
    Schema::create('informations', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('tournament_id');
        $table->foreign('tournament_id')->references('id')
            ->on('tournaments')->onDelete('cascade');
        $table->text('description');
        $table->timestamps();
    });
}
```

Gambar 4.19 Migrasi tabel informations

Gambar 4.19 merupakan isi dari file migrasi untuk tabel information yang menghasilkan tabel informations pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.12 berikut.

Tabel 4.12 Struktur tabel informations

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
tournament_id	BIGINTEGER	20	Foreign key
description	TEXT		
created_at	DATE		
updated_at	DATE		

Tabel information berfungsi untuk menyimpan informasi-informasi lanjutan dari suatu kompetisi yang nantinya akan diisi oleh pembuat kompetisi (*organizer*). Tabel

informations memiliki beberapa kolom yaitu 'id' sebagai *primary key*, 'tournament_id' sebagai *foreign key* dari tabel tournaments, 'description' berfungsi untuk menyimpan keterangan informasi, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

j. Tabel matches

```
public function up()
{
    Schema::create('matches', function (Blueprint $table) {
        $table->id();
        $table->date('date');
        $table->unsignedBigInteger('teams_id_home');
        $table->foreign('team_id_home')->references('id')
            ->on('teams')->onDelete('cascade');
        $table->unsignedBigInteger('team_id_away');
        $table->foreign('team_id_away')->references('id')
            ->on('teams')->onDelete('cascade');
        $table->string('score_home');
        $table->string('score_away');
        $table->timestamps();
    });
}
```

Gambar 4.20 Migrasi tabel matches

Gambar 4.20 merupakan isi dari file migrasi untuk tabel matches yang menghasilkan tabel matches pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.13 berikut.

Tabel 4.13 Struktur tabel match

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
date	DATE		
team_id_home	BIGINTEGER	20	Foreign key
team_id_away	BIGINTEGER	20	Foreign key
score_home	VARCHAR	191	
score_away	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel matches berfungsi untuk menyimpan data pertandingan yang terjadi pada suatu kompetisi dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'date' berfungsi untuk menyimpan tanggal pertandingan, 'team_id_home' sebagai *foreign key* dari tim pertama, 'team_id_away' sebagai *foreign key* dari id tim kedua (lawan), 'result_home' berfungsi untuk menyimpan hasil pertandingan tim pertama, 'result_away' berfungsi untuk menyimpan hasil pertandingan tim kedua, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

k. Tabel schedules

```

public function up()
{
    Schema::create('schedules', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('tournament_id');
        $table->foreign('tournament_id')->references('id')
            ->on('tournaments')->onDelete('cascade');
        $table->unsignedBigInteger('match_id');
        $table->foreign('match_id')->references('id')-
        >on('matches')
            ->onDelete('cascade');
        $table->timestamps();
    });
}

```

Gambar 4.21 Migrasi tabel schedules

Gambar 4.21 merupakan isi dari file migrasi untuk tabel schedules yang menghasilkan tabel schedules pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.14 berikut.

Tabel 4.14 Struktur tabel schedules

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
tournament_id	BIGINTEGER	20	Foreign key
match_id	BIGINTEGER	20	Foreign key
created_at	DATE		
updated_at	DATE		

Tabel schedules berfungsi untuk menyimpan data jadwal pertandingan dari sebuah kompetisi dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'tournament_id' sebagai

foreign key dari tabel tournaments, 'match_id' sebagai *foreign key* dari tabel matches, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

1. Tabel organizer_transaction

```

public function up()
{
    Schema::create('organizer_transaction', function (Blueprint $table)
    {
        $table->id();
        $table->unsignedBigInteger('organizer_id');
        $table->string('type');
        $table->integer('amount');
        $table->string('description');
        $table->string('status');
        $table->timestamps();

        $table->foreign('organizer_id')->references('id')
            ->on('organizers')->onDelete('cascade');
    });
}

```

Gambar 4.22 Migrasi tabel organizer_transaction

Gambar 4.22 merupakan isi dari file migrasi untuk tabel organizer_transaction yang menghasilkan tabel organizer_transaction pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.15 berikut.

Tabel 4.15 Struktur tabel organizer_transaction

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
organizer_id	BIGINTEGER	20	Foreign key
amount	INTEGER	11	
type	VARCHAR	191	
description	VARCHAR	191	
status	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel organizer_transaction berfungsi untuk menyimpan data transaksi yang dilakukan oleh *organizer* dengan beberapa kolom yaitu 'id' sebagai *primary key*,

'organizer_id' sebagai *foreign key* dari tabel organizers, 'amount' berfungsi untuk menyimpan nominal transaksi, 'type' berfungsi untuk menyimpan tipe transaksi yaitu keluar atau masuk, 'description' berfungsi untuk menyimpan keterangan transaksi, 'status' berfungsi untuk menyimpan status transaksi, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

m. Tabel team_transaction

```

public function up()
{
    Schema::create('team_transaction', function (Blueprint $table)
    {
        $table->id();
        $table->unsignedBigInteger('team_id');
        $table->foreign('team_id')->references('id')->on('teams')
            ->onDelete('cascade');
        $table->integer('amount');
        $table->string('type');
        $table->string('description');
        $table->string('status');
        $table->timestamps();
    });
}

```

Gambar 4.23 Migrasi tabel team_transaction

Gambar 4.23 merupakan isi dari file migrasi untuk tabel team_transaction yang menghasilkan tabel team_transaction pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.16 berikut.

Tabel 4.16 Struktur tabel team_transaction

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
team_id	BIGINTEGER	20	Foreign key
amount	INTEGER	11	
type	VARCHAR	191	
description	VARCHAR	191	
status	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel `team_transaction` berfungsi untuk menyimpan data transaksi yang terjadi pada suatu tim dengan beberapa kolom yaitu `'id'` sebagai *primary key*, `'team_id'` sebagai *foreign key* dari tabel `teams`, `'amount'` berfungsi untuk menyimpan nominal transaksi, `'type'` berfungsi untuk menyimpan tipe transaksi yaitu keluar atau masuk, `'description'` berfungsi untuk menyimpan keterangan transaksi, `'status'` berfungsi untuk menyimpan status transaksi, `'created_at'` berfungsi menyimpan tanggal dibuatnya data dan `'updated_at'` untuk menyimpan tanggal terakhir data dimodifikasi.

n. Tabel `tournament_transaction`

```
public function up()
{
    Schema::create('tournament_transaction',
        function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('tournament_id');
            $table->unsignedBigInteger('team_id');
            $table->string('type');
            $table->integer('amount');
            $table->string('description');
            $table->string('status');
            $table->timestamps();

            $table->foreign('tournament_id')->references('id')
                ->on('tournaments')->onDelete('cascade');
            $table->foreign('team_id')->references('id')->on('teams')
                ->onDelete('cascade');
        });
}
```

Gambar 4.24 Migrasi tabel `tournament_transaction`

Gambar 4.24 merupakan isi dari file migrasi untuk tabel `tournament_transaction` yang menghasilkan tabel `tournament_transaction` pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.17 berikut.

Tabel 4.17 Struktur tabel tournament_transaction

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
tournament_id	BIGINTEGER	20	Foreign key
team_id	BIGINTEGER	20	Foreign key
amount	INTEGER	11	
type	VARCHAR	191	
description	VARCHAR	191	
status	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel tournament_transaction berfungsi untuk menyimpan data transaksi yang terjadi pada sebuah kompetisi dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'tournament_id' sebagai *foreign key* dari tabel tournaments, 'team_id' sebagai *foreign key* dari tabel teams, 'amount' berfungsi untuk menyimpan nominal transaksi, 'type' berfungsi untuk menyimpan tipe transaksi yaitu keluar atau masuk, 'description' berfungsi untuk menyimpan keterangan transaksi, 'status' berfungsi untuk menyimpan status transaksi, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

o. Tabel proofs

```

public function up()
{
    Schema::create('proofs', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('tournament_id');
        $table->unsignedBigInteger('winner_id');
        $table->string('pic');
        $table->timestamps();

        $table->foreign('tournament_id')->references('id')
            ->on('tournaments');
        $table->foreign('winner_id')->references('id')
            ->on('teams');
    });
}

```

Gambar 4.25 Migrasi tabel proofs

Gambar 4.25 merupakan isi dari file migrasi untuk tabel proofs yang menghasilkan tabel proofs pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.18 berikut.

Tabel 4.18 Struktur tabel proofs

Atribut	Tipe Data	Panjang/Nilai	Keterangan
id	BIGINTEGER	20	AUTO_INCREMENT
tournament_id	BIGINTEGER	20	Foreign key
winner_id	BIGINTEGER	20	Foreign key
pic	VARCHAR	191	
created_at	DATE		
updated_at	DATE		

Tabel proofs berfungsi untuk menyimpan data bukti kompetisi telah selesai dengan beberapa kolom yaitu 'id' sebagai *primary key*, 'tournament_id' sebagai *foreign key* dari tabel tournaments, 'winner_id' sebagai *foreign key* dari tabel teams, 'pic' berfungsi untuk menyimpan nama *file* untuk bukti kompetisi yang berupa gambar, 'created_at' berfungsi menyimpan tanggal dibuatnya data dan 'updated_at' untuk menyimpan tanggal terakhir data dimodifikasi.

p. Tabel player_team

```
public function up()
{
    Schema::create('player_team', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('team_id');
        $table->unsignedBigInteger('player_id');
        $table->string('authority')->default('Member');
        $table->string('status')->default('Pending');
        $table->timestamps();

        $table->foreign('team_id')->references('id')->on('teams')
            ->onDelete('cascade');
        $table->foreign('player_id')->references('id')
            ->on('players')->onDelete('cascade');
    });
}
```

Gambar 4.26 Migrasi tabel player_team

Gambar 4.26 merupakan isi dari file migrasi untuk tabel `player_team` yang menghasilkan tabel `player_team` pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.19 berikut.

Tabel 4.19 Struktur tabel `player_team`

Atribut	Tipe Data	Panjang/Nilai	Keterangan
<code>id</code>	BIGINTEGER	20	AUTO_INCREMENT
<code>team_id</code>	BIGINTEGER	20	Foreign key
<code>player_id</code>	BIGINTEGER	20	Foreign key
<code>authority</code>	VARCHAR	191	
<code>status</code>	VARCHAR	191	
<code>created_at</code>	DATE		
<code>updated_at</code>	DATE		

Tabel `player_team` merupakan tabel *pivot* antara tabel `players` dan tabel `teams` yang berfungsi untuk menyimpan data anggota dari suatu tim dengan beberapa kolom yaitu '`id`' sebagai *primary key*, '`team_id`' sebagai *foreign key* dari tabel `teams`, '`player_id`' sebagai *foreign key* dari tabel `players`, '`authority`' berfungsi untuk menyimpan status jabatan *player* pada suatu tim, '`status`' berfungsi untuk status *player* pada suatu tim, '`created_at`' berfungsi menyimpan tanggal dibuatnya data dan '`updated_at`' untuk menyimpan tanggal terakhir data dimodifikasi.

q. Tabel `team_tournament`

```
public function up()
{
    Schema::create('team_tournament', function (Blueprint $table)
    {
        $table->id();
        $table->unsignedBigInteger('tournament_id');
        $table->unsignedBigInteger('team_id');
        $table->timestamps();
        $table->foreign('tournament_id')->references('id')
            ->on('tournaments')->onDelete('cascade');
        $table->foreign('team_id')->references('id')->on('teams')
            ->onDelete('cascade');
    });
}
```

Gambar 4.27 Migrasi tabel `team_tournament`

Gambar 4.27 merupakan isi dari file migrasi untuk tabel `team_tournament` yang menghasilkan tabel `team_tournament` pada basis data MySQL Arenasia dengan struktur tabel seperti pada tabel 4.20 berikut.

Tabel 4.20 Struktur tabel `team_tournament`

Atribut	Tipe Data	Panjang/Nilai	Keterangan
<code>id</code>	BIGINTEGER	20	AUTO_INCREMENT
<code>tournament_id</code>	BIGINTEGER	20	Foreign key
<code>team_id</code>	BIGINTEGER	20	Foreign key
<code>created_at</code>	DATE		
<code>updated_at</code>	DATE		

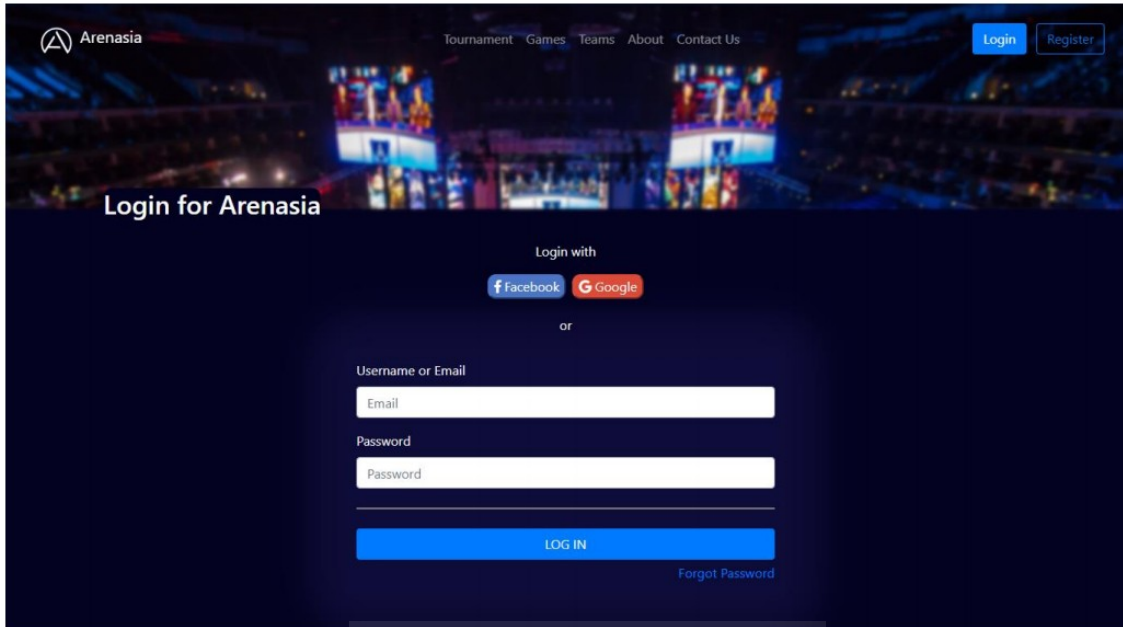
Tabel `team_tournament` merupakan tabel *pivot* antara tabel `teams` dan tabel `tournaments` yang berfungsi untuk menyimpan data peserta dari suatu kompetisi dengan beberapa kolom yaitu '`id`' sebagai *primary key*, '`tournament_id`' sebagai *foreign key* dari tabel `tournaments`, '`team_id`' sebagai *foreign key* dari tabel `teams`, '`created_at`' berfungsi menyimpan tanggal dibuatnya data dan '`updated_at`' untuk menyimpan tanggal terakhir data dimodifikasi.

4.4.2 Integrasi Antarmuka pada Back-End

Agar tampilan antarmuka yang telah dibuat menggunakan *framework bootstrap* oleh desainer dapat berjalan pada *framework* Laravel serta dapat menampilkan hasil *query* dari basis data, maka tiap-tiap antarmuka harus dihubungkan dengan *controller*.

a. Login

Halaman login memuat *form* email dan password yang harus diisi ketika ingin *login* pada web Arenasia. Halaman login ini dapat digunakan untuk autentikasi semua *role* baik itu *admin*, *organizer* ataupun *player*. Sistem secara otomatis akan memvalidasi *email* dan *password* dari seorang pengguna yang melakukan *login*, selanjutnya pengguna dengan *email* terkait akan *diautentikasi* berdasarkan *role* yang terdaftar pada basis data. Tampilan antarmuka halaman login dapat dilihat pada gambar 4.28 serta kode *controllernya* seperti pada gambar 4.29 berikut.



Gambar 4.28 Halaman login

```

public function login(Request $request)
{
    $this->validateLogin($request);

    if (method_exists($this, 'hasTooManyLoginAttempts') &&
        $this->hasTooManyLoginAttempts($request)) {
        $this->fireLockoutEvent($request);

        return $this->sendLockoutResponse($request);
    }

    if ($this->attemptLogin($request)) {
        return $this->sendLoginResponse($request);
    }

    $this->incrementLoginAttempts($request);

    return $this->sendFailedLoginResponse($request);
}

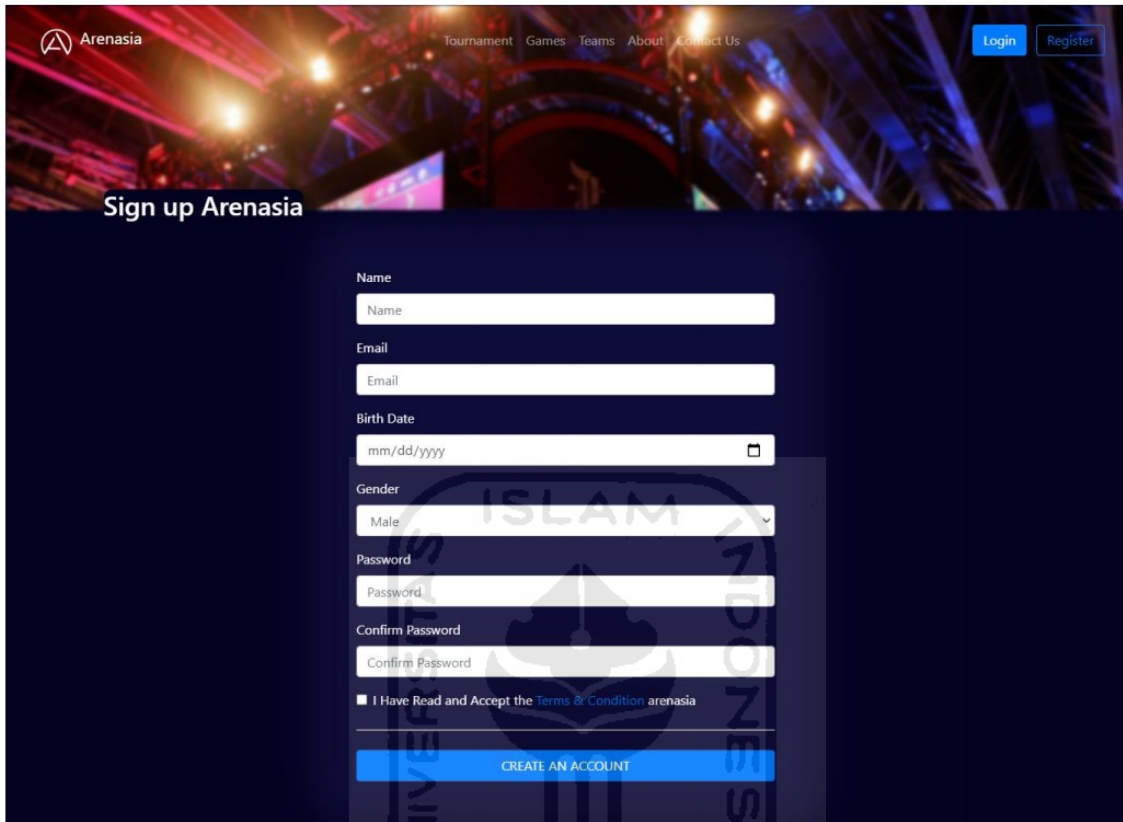
```

Gambar 4.29 Kode login

b. Register

Pada web Arenasia terdapat 2 halaman register yaitu masing-masing untuk *player* dan *organizer*. Halaman register memuat *form* pendaftaran untuk pengguna baru web Arenasia. *Player* harus mengisi beberapa *form* yaitu nama, email, tanggal lahir, jenis kelamin, kontak dan password. Sedangkan *organizer* harus mengisi beberapa *form* yaitu nama, email, bank, nomor rekening dan password. Tampilan antarmuka halaman register

dapat dilihat pada gambar 4.30 serta kode *controllernya* seperti pada gambar 4.31 dan 4.32 berikut.



The image shows a registration form for 'Arenasia'. The form is set against a dark blue background. At the top left, there is a logo for 'Arenasia' and a navigation menu with links for 'Tournament', 'Games', 'Teams', 'About', and 'Contact Us'. To the right of the navigation menu are two buttons: 'Login' and 'Register'. Below the navigation is a large heading 'Sign up Arenasia'. The registration form consists of several input fields: 'Name', 'Email', 'Birth Date' (with a date picker icon), 'Gender' (a dropdown menu currently showing 'Male'), 'Password', and 'Confirm Password'. Below these fields is a checkbox labeled 'I Have Read and Accept the Terms & Condition arenasia'. At the bottom of the form is a prominent blue button with the text 'CREATE AN ACCOUNT'.

Gambar 4.30 Halaman register

```

protected function create(array $data)
{
    $create = User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);

    $create->assignRole('player');

    Player::create([
        'user_id' => $create->id,
        'name' => $data['name'],
        'birth' => $data['birth'],
        'gender' => $data['gender'],
        'contact' => $data['contact'],
        'slug' => Str::slug($data['name']),
    ]);

    return $create;
}

```

Gambar 4.31 Kode register player

```

public function organizerStore(Request $request)
{
    $request->validate([
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255',
'unique:users'],
        'bank_id' => ['required', 'numeric'],
        'bank_account' => ['required', 'numeric'],
        'contact' => ['required', 'numeric'],
        'password' => ['required', 'string', 'min:8',
'confirmed'],
    ]);

    $create = User::create([
        'name' => $request->name,
        'email' => $request->email,
        'password' => Hash::make($request->password),
    ]);

    $create->assignRole('organizer');

    Organizer::create([
        'user_id' => $create->id,
        'name' => $request->name,
        'bank_id' => $request->bank_id,
        'bank_account' => $request->bank_account,
        'contact' => $request->contact,
        'slug' => Str::slug($request->name),
    ]);

    return $create;
}

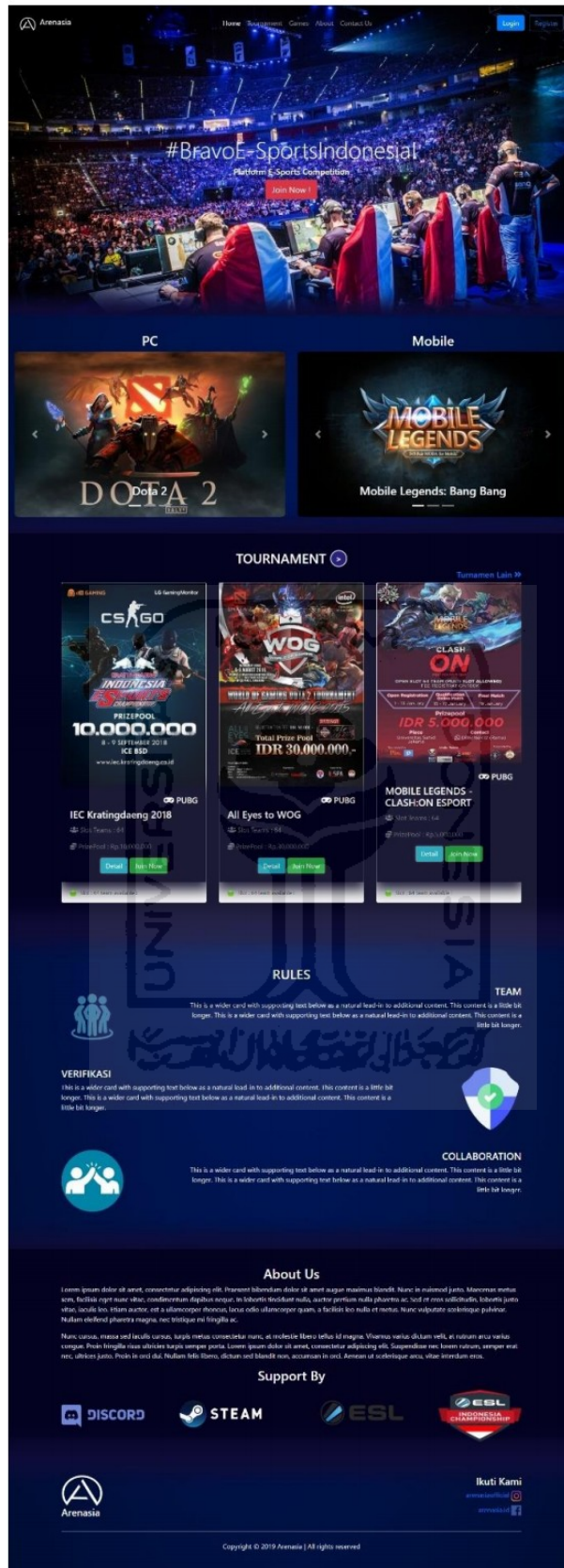
```

Gambar 4.32 Kode register organizer

c. Beranda

Halaman beranda merupakan halaman utama ketika mengakses web Arenasia. Pada halaman ini terdapat beberapa menu dan atau informasi yaitu kompetisi, *game*, tim, *rules*, tentang dan kontak. Tampilan antarmuka halaman beranda dapat dilihat pada gambar 4.34 serta kode *controllernya* seperti pada gambar 4.34 berikut.





Gambar 4.33 Halaman beranda

```

public function home()
{
    $pc = Platform::whereName('PC')->first();
    $mobile = Platform::whereName('Mobile')->first();
    $tournament = Tournament::whereStatus('Registration')->get()
        ->sortByDesc('created_at');

    if (Auth::check() && Auth::user()->hasRole('player')) {
        $user = Player::whereUserId(Auth::user()->id)->first();
    } else {
        $user = false;
    }

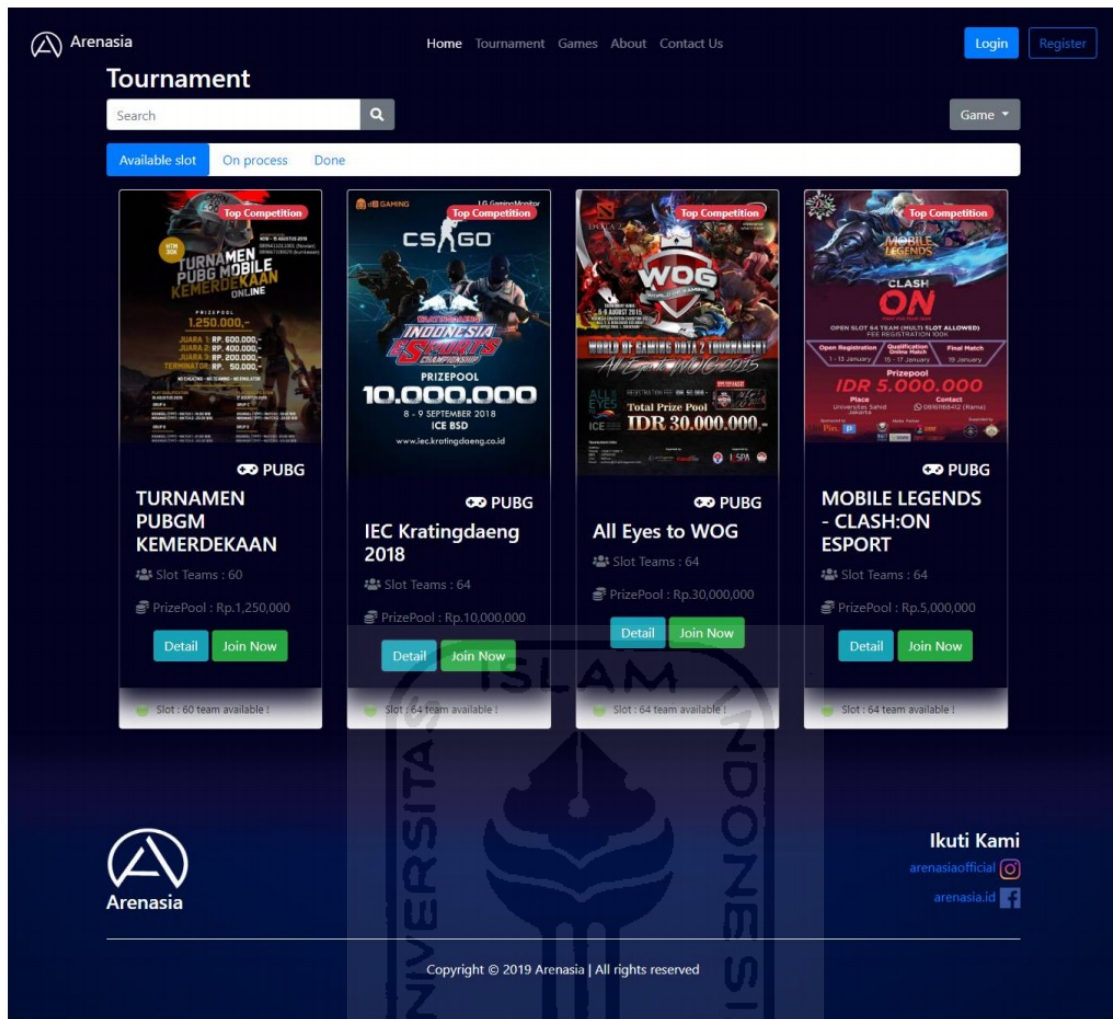
    return view('pages.home', [
        'pc'=>$pc,
        'mobile'=>$mobile,
        'tournament'=>$tournament,
        'user'=>$user,
    ]);
}

```

Gambar 4.34 Kode beranda

d. Tournament

Halaman tournament memuat daftar kompetisi yang telah dibuat oleh *organizer*. Kompetisi dibagi menjadi 4 bagian yaitu kompetisi dengan status registrasi (*registration*), sedang berlangsung (*running*), selesai (*done*) dan yang akan datang (*upcoming*). Pada halaman ini juga terdapat fitur pencarian menurut nama dan *game*. Tampilan antarmuka halaman beranda dapat dilihat pada gambar 4.35 serta kode *controllernya* seperti pada gambar 4.36 berikut.



Gambar 4.35 Halaman tournament

```

public function index()
{
    $tournament = Tournament::all();

    $game = Game::all();

    if (Auth::check() && Auth::user()->hasRole('player')) {
        $user = Player::whereUserId(Auth::user()->id)->first();
    } else {
        $user = false;
    }

    return view('pages.tournament.index', [
        'tournament' => $tournament,
        'game'=>$game,
        'user'=>$user
    ]);
}

```

Gambar 4.36 Kode tournament

e. Detail Tournament

Halaman detail tournament memuat informasi keseluruhan suatu kompetisi. Halaman ini memiliki 4 sub menu yaitu *description* yang berisi deskripsi suatu kompetisi, *news* yang berisi informasi tambahan yang dibuat oleh *organizer* dari suatu kompetisi, *participant* yang berisi daftar tim yang mengikuti suatu kompetisi dan *schedule* yang berisi informasi jadwal pertandingan. Tampilan antarmuka halaman beranda dapat dilihat pada gambar 4.37 serta kode *controllernya* seperti pada gambar 4.38 berikut.

The screenshot displays the Arenasia website interface for a tournament. At the top, there is a navigation bar with 'Home', 'Tournament', 'Games', 'About', and 'Contact Us' links, along with 'Login' and 'Register' buttons. The main content area features a large banner for 'Wolfgank Family' with a 'PLAYERUNKNOWN'S' logo. Below the banner, a sidebar on the left contains navigation options: 'Description', 'News', 'Participant', and 'Schedule'. The main content area shows the following details:

- Tournament Name:** Wolfgank Family
- Slot Teams:** 6/6
- Start Tournament:** 15 Maret 2020
- Game:** PUBG Mobile
- Fee:** Rp.150,000
- Total Prizes:** Rp.20,000,000

The **Description** section includes the following information:

- Detail Turnamen:**
 - Match Day : 5 - 6 Oktober 2019
 - Pendaftaran akan ditutup tanggal 2 Oktober 2019
 - TM tanggal 3 Oktober 2019
 - Biaya pendaftaran IDR 150.000,- / Slot
 - Venue : Aula Gedung E UDINUS - Jalan Imam Bonjol No.207, Pendrikan Kidul, Semarang
 - Mobile Only (Max 6,5 Inch)
 - No Controller | No Trigger | No L1/R1 | No Game Pad
 - No Cheat | No Spion | No Teaming
 - No Voice All
 - No SARA
 - Squad Mode
- Hadiah:**
 - Juara 1 : Rp. 9.000.000,- + Plakat + Sertifikat + Voucher Varuna Entertainment
 - Juara 3 : Rp. 3.500.000,- + Plakat + Sertifikat + Voucher Varuna Entertainment
 - Juara 4 : Rp. 1.750.000,- + Plakat + Sertifikat + Voucher Varuna Entertainment
 - Juara 5 : Rp. 1.000.000,- + Plakat + Sertifikat + Voucher Varuna Entertainment
 - Terminator : Rp. 750.000,- + Plakat + Sertifikat + Voucher Varuna Entertainment
 - Fun Games & Mini Games : Voucher Varuna Entertainment
 - *Hadiah FIX tidak berubah

At the bottom of the page, there is an 'Ikuti Kami' section with social media links for 'arenasiaofficial' (Instagram) and 'arenasia.id' (Facebook). The footer contains the Arenasia logo and the text 'Copyright © 2019 Arenasia | All rights reserved'.

Gambar 4.37 Halaman detail tournament


```

public function show(Tournament $tournament)
{
    $organizer = Organizer::whereUserId(Auth::user()->id)
        ->first();

    $info = Information::whereTournamentId($tournament->id)
        ->get();

    $hasJoined = false;

    if (Auth::check() && Auth::user()->hasRole('player')) {
        $player = Player::whereUserId(Auth::user()->id)->first();

        $team = $player->teams()->where('game_id',
            $tournament->game_id)->first();

        $participant = $tournament->teams()->pluck('team_id');

        $hasJoined = $participant->contains($team->id);
    }

    return view('pages.tournament.detail', [
        'tournament'=>$tournament,
        'organizer'=>$organizer,
        'info'=>$info,
        'hasJoined'=>$hasJoined
    ]);
}

```

Gambar 4.38 Kode detail tournament

f. Detail Team

Halaman detail team memuat informasi suatu tim. Jika pengguna yang bukan merupakan *player* dari tim terkait mengunjungi halaman ini maka informasi yang ditampilkan hanya deskripsi tim beserta *player* yang tergabung pada tim tersebut, namun jika yang mengakses merupakan *player* dari tim terkait maka terdapat tambahan informasi yaitu kompetisi yang diikuti, permintaan bergabung dari *player* dan saldo tim. Tampilan antarmuka halaman beranda dapat dilihat pada gambar 4.39 serta kode *controllernya* seperti pada gambar 4.40 berikut.

The screenshot displays the 'Evos' team page on the Arenasia website. The page features a navigation bar with 'Home', 'Tournament', 'Games', 'About', and 'Contact Us'. The main header includes the 'ESPORT' logo and a user profile for 'Titin Yolanda'. The team's logo is a stylized wolf head with 'EVOS' written below it. The founder is listed as 'Titin Yolanda' and the games played are 'PUBG Mobile'. A table lists five players with their IDs and names. Below this, there are three sections: 'Tournament' showing a registration for 'Wolfgank Family' by 'Titin Yolanda' with a status of 'Done'; 'Request' with a status of 'Open' and an empty list; and 'Balance' showing a current balance of 'Rp.0' and a transaction history with two entries: 'Register tournament: Wolfgank Family' for a debit of 'Rp.150,000' and 'make a deposit.' for a credit of 'Rp.150,000'. The footer includes the Arenasia logo, social media links for 'Ikuti Kami' (Instagram and Facebook), and a copyright notice for 2019.

Evos
 Founder : Titin Yolanda
 Games : PUBG Mobile

No	Players
1	Titin Yolanda
2	Harsanto Prabowo
3	Paiman Yolanda
4	Samiah Hutasoit
5	Farah Hidayat

Tournament

Name	Registrant	Status
Wolfgank Family	Titin Yolanda	Done

Request
 status : Open
 empty

Balance
 Rp.0 [Add Balance](#)

Date	Description	Balance
29 April 2020	Register tournament: Wolfgank Family	Rp.150,000
29 April 2020	make a deposit.	Rp.150,000

Ikuti Kami
[arenasiaofficial](#)
[arenasia.id](#)

Copyright © 2019 Arenasia | All rights reserved

Gambar 4.39 Halaman detail team

```

public function show(Team $team)
{
    $accepted = $team->players()->wherePivot('status','Accepted')
        ->get();
    $pending = $team->players()->wherePivot('status','Pending')
        ->get();

    $isMember = false;
    $haveTeam = false;

    $tournament = $team->tournaments()->get();

    if (Auth::check() && Auth::user()->hasRole('player')) {
        $player = Player::whereUserId(Auth::user()->id)
            ->first();

        $isMember = $team->players()
            ->wherePivot('player_id',$player->id)
            ->first();

        $checkTeam = $player->teams()
            ->wherePivot('status', 'Accepted')
            ->pluck('game_id');

        if ($checkTeam->contains($team->game_id)) {
            $haveTeam = true;
        }
    }

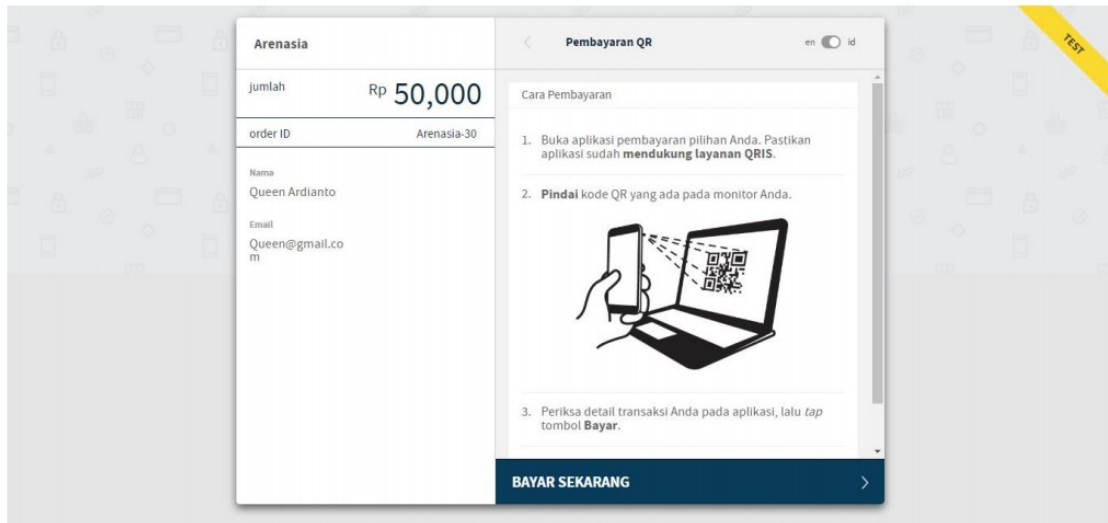
    return view('pages.team.detail', [
        'team'=>$team,
        'accepted'=>$accepted,
        'pending'=>$pending,
        'isMember'=>$isMember,
        'haveTeam'=>$haveTeam,
        'tournament'=>$tournament
    ]);
}

```

Gambar 4.40 Kode detail tim

g. Transaksi

Salah satu transaksi yang ada pada web Arenasia yaitu pengisian saldo tim. Pengisian saldo tim hanya dapat dilakukan oleh anggota tim dengan mengisi *form* pada halaman checkout. Selanjutnya sistem akan mengarahkan *player* ke halaman midtrans untuk menyelesaikan transaksi. Tampilan antarmuka halaman pembayaran midtrans dapat dilihat pada gambar 4.41 serta kode *controllernya* seperti pada gambar 4.42 berikut.



Gambar 4.41 Halaman pembayaran midtrans



```

public function callback()
{
    Config::$serverKey = config('midtrans.serverKey');
    Config::$isProduction = config('midtrans.isProduction');
    Config::$isSanitized = config('midtrans.isSanitized');
    Config::$is3ds = config('midtrans.is3ds');

    $notif = new Notification();

    $id = explode('-', $notif->order_id);

    $transaction = $notif->transaction_status;
    $type = $notif->payment_type;
    $order_id = $id[1];
    $fraud = $notif->fraud_status;

    $teamTransaction = TeamTransaction::find($order_id);

    if ($transaction == 'capture') {
        if ($type == 'credit_card') {
            if ($fraud == 'challenge') {
                $teamTransaction->update
                    (['status' => 'Pending']);
            } else {
                $teamTransaction->update
                    (['status' => 'Accepted']);
            }
        }
    } else if ($transaction == 'settlement') {
        $teamTransaction->update(['status' => 'Accepted']);
    } else if ($transaction == 'pending') {
        $teamTransaction->update(['status' => 'Pending']);
    } else if ($transaction == 'deny') {
        $teamTransaction->delete();
    } else if ($transaction == 'expire') {
        $teamTransaction->delete();
    } else if ($transaction == 'cancel') {
        $teamTransaction->delete();
    }
}

```

Gambar 4.42 Kode transaksi midtrans

4.4.3 Pengujian

Pengujian web dilakukan untuk mengetahui apakah terdapat kesalahan atau *error* yang terjadi ketika mengakses suatu halaman dan beberapa fungsi pada suatu web. Dalam melakukan pengujian penulis menggunakan Dusk, yaitu sebuah *package* pada Laravel yang digunakan untuk melakukan pengujian pada web secara otomatis.

Proses pengujian otomatis dengan Dusk bergantung pada *test case* yang dibuat. Sistem akan menjalankan setiap *test case* untuk melakukan pengujian secara otomatis melalui Google Chrome untuk membandingkan hasil yang diharapkan dengan hasil yang sebenarnya. Jika hasil

tidak sesuai harapan berarti terdapat kesalahan pada sistem dan sistem akan memberikan *output* “Error”, sebaliknya jika berhasil sistem akan memberikan *output* “OK”.

Pengujian dilakukan pada fungsi-fungsi utama di web Arenasia seperti membuat akun, autentikasi, menambah *game*, membuat tim, membuat kompetisi, bergabung dengan tim dan bergabung pada kompetisi.

a. Register

Pengujian register dilakukan untuk menguji pengguna yang belum terdaftar pada sistem dapat membuat akun pada web Arenasia, dalam hal ini dilakukan pengujian pada *player* dan *organizer*. Pengguna diarahkan ke halaman register dan mengisi *form* pembuatan akun untuk melakukan *register*. Pengujian dilakukan dengan membuat suatu *test case* seperti pada gambar 4.43 berikut.

```
public function testRegister()
{
    $this->browse(function ($first, $second) {
        $first->visitRoute('register')
            ->type('name', 'Muhammad Ali')
            ->type('email', 'muhammadali@gmail.com')
            ->type('birth', '04/25/1991')
            ->select('gender', 'Male')
            ->type('contact', '0878676868866')
            ->type('password', 'qweqweqwe')
            ->type('password_confirmation', 'qweqweqwe')
            ->click('#register')
            ->assertSee('Muhammad Ali');

        $second->visitRoute('register.organizer')
            ->type('name', 'Bayu Aji')
            ->type('email', 'bayuaji@gmail.com')
            ->select('bank_id', '1')
            ->type('bank_account', '456234546')
            ->type('contact', '0821234324234')
            ->type('password', 'qweqweqwe')
            ->type('password_confirmation', 'qweqweqwe')
            ->click('#register')
            ->assertSee('Bayu Aji');

    });
}
```

Gambar 4.43 Kode pengujian register

b. Login

Pengujian login dilakukan untuk menguji pengguna yang telah terdaftar pada sistem dapat melakukan *login* pada web Arenasia sesuai dengan *role* mereka. Pengujian dilakukan menggunakan 3 *browser* yang masing-masing untuk menguji login *admin*,

organizer dan *player*. Pengguna diarahkan ke halaman login dan mengisi *form* untuk melakukan *login*. Pengujian dilakukan dengan membuat suatu *test case* seperti pada gambar 4.44 berikut.

```
public function testLogin()
{
    $this->browse(function ($first, $second, $third) {
        $first->visit('/login')
            ->type('email', 'rizky@gmail.com')
            ->type('password', 'qweqweqwe')
            ->press('Login')
            ->assertPathIs('/player');

        $second->visit('/login')
            ->type('email', 'ahdiyat@arenasia.masuk.id')
            ->type('password', 'qweqweqwe')
            ->press('Login')
            ->assertPathIs('/organizer');

        $third->visit('/login')
            ->type('email', 'admin@arenasia.masuk.id')
            ->type('password', 'qweqweqwe')
            ->press('Login')
            ->assertPathIs('/admin');
    });
}
```

Gambar 4.44 Kode pengujian login

c. Logout

Pengujian logout dilakukan untuk menguji pengguna yang *login* dalam web Arenasia dapat melakukan *logout*. Pengujian dilakukan menggunakan 3 *browser* yang masing-masing untuk menguji logout *admin*, *organizer* dan *player*. Sistem akan mengasumsikan pengguna telah *login* menurut *role* masing-masing, selanjutnya dilakukan *logout* dan sistem akan mengarahkan pengguna ke halaman utama. Pengujian dilakukan dengan membuat suatu *test case* seperti pada gambar 4.45 berikut.

```

public function testLogout()
{
    $this->browse(function ($first, $second, $third) {
        $first->loginAs (User::find(1))
            ->visit('/')
            ->click('#nav')
            ->click('#logout')
            ->assertPathIs ('/');
        $second->loginAs (User::find(2))
            ->visit('/')
            ->click('#nav')
            ->click('#logout')
            ->assertPathIs ('/');
        $third->loginAs (User::find(5))
            ->visit('/')
            ->click('#nav')
            ->click('#logout')
            ->assertPathIs ('/');
    });
}

```

Gambar 4.45 Kode pengujian logout

d. Menambah Game

Pengujian menambah game dilakukan untuk menguji *admin* dapat melakukan penambahan kategori *game* di web Arenasia. Sistem akan mengasumsikan pengguna telah *login* sebagai *admin*, setelah itu *admin* mengakses menu pembuatan *game* dan mengisi *form* yang tersedia, setelah berhasil sistem akan memastikan *admin* diarahkan ke halaman index game. Pengujian dilakukan dengan membuat suatu *test case* seperti pada gambar 4.46 berikut.

```

public function testAddGame()
{
    $this->browse(function (Browser $browser) {
        $browser->loginAs (User::find(1))
            ->visitRoute ('admin.game.create')
            ->type ('name', 'PUBG Mobile')
            ->select ('platform_id', 2)
            ->type ('minimum_players', 4)
            ->attach ('pic',
storage_path ('app/public/games/pubgm.png'))
            ->click ('#createGame')
            ->assertRouteIs ('admin.game.index');
    });
}

```

Gambar 4.46 Kode pengujian pembuatan game

e. Membuat Tim

Pengujian pembuatan tim dilakukan untuk menguji *player* yang belum memiliki tim dengan *game* sejenis dapat membuat tim di web Arenasia. Sistem akan mengasumsikan pengguna telah *login* sebagai *player*, setelah itu *player* mengakses menu pembuatan tim dan mengisi *form* yang tersedia, setelah berhasil sistem akan memastikan *player* diarahkan ke halaman index tim. Pengujian dilakukan dengan membuat suatu *test case* seperti pada gambar 4.47 berikut.

```
public function testCreateTeam()
{
    $this->browse(function (Browser $browser) {
        $browser->loginAs(User::find(6))
            ->visitRoute('player.team.create')
            ->type('name', 'Natus Vincere')
            ->select('#platform', 1)
            ->select('game', 3)
            ->attach('pic',
                storage_path('app/public/teams/navi.png'))
            ->click('#createTeam')
            ->assertRouteIs('player.team.index');
    });
}
```

Gambar 4.47 Kode pengujian pembuatan tim

f. Bergabung dengan Tim

Pengujian bergabung dengan tim dilakukan untuk menguji *player* yang belum memiliki tim dengan *game* sejenis dapat bergabung pada suatu tim di web Arenasia. Sistem akan mengasumsikan pengguna telah *login* sebagai *player*, setelah itu *player* memilih salah satu tim yang tersedia dan melakukan permintaan bergabung, setelah berhasil sistem akan memastikan *player* melihat tulisan 'Request Pending'. Pengujian dilakukan dengan membuat suatu *test case* seperti pada gambar 4.48 berikut.

```
public function testJoinTeam()
{
    $this->browse(function (Browser $browser) {
        $browser->loginAs(User::find(6))
            ->visitRoute('player.team.show', 'rex-regum-
qeon')
            ->click('#join')
            ->assertSee('Request Pending');
    });
}
```

Gambar 4.48 Kode pengujian bergabung tim

g. Membuat Kompetisi

Pengujian membuat kompetisi dilakukan untuk menguji *organizer* dapat melakukan pembuatan kompetisi di web Arenasia. Sistem akan mengasumsikan pengguna telah *login* sebagai *organizer*, setelah itu *organizer* mengakses menu pembuatan kompetisi dan mengisi *form* pembuatan tim, setelah berhasil sistem akan memastikan *organizer* dialihkan ke halaman index tournament. Pengujian dilakukan dengan membuat suatu test case seperti pada gambar 4.49 berikut.

```
public function testCreateTournament()
{
    $this->browse(function (Browser $browser) {
        $browser->loginAs(User::find(2))
            ->visitRoute('admin.tournament.create')
            ->type('name', 'Test Tournament')
            ->select('#platform', 1)
            ->select('#game', 1)
            ->select('#province', 'Aceh')
            ->select('#city', 'Kota Sabang')
            ->type('prizepool', 2000000)
            ->type('fee', 50000)
            ->type('slot', 16)
            ->type('start_registration', '06/29/2020')
            ->type('last_registration', '06/30/2020')
            ->type('start_matchday', '07/04/2020')
            ->type('last_matchday', '07/05/2020')
            ->attach('pic', storage_path
                ('app/public/tournament/test_tournament.jpg'
                ))
            ->click('#createTournament')
            ->assertRouteIs('organizer.tournament.index');
    });
}
```

Gambar 4.49 Kode pengujian pembuatan kompetisi

h. Bergabung pada Kompetisi

Pengujian bergabung pada kompetisi dilakukan untuk menguji *player* yang telah memiliki tim dapat bergabung pada suatu kompetisi dengan jenis *game* yang sama dengan tim yang dimiliki. Sistem akan mengasumsikan pengguna telah *login* sebagai *player*, setelah itu *player* diarahkan oleh sistem menuju halaman utama. Pada bagian kompetisi, *player* memilih salah satu kompetisi dan melakukan aksi bergabung pada kompetisi tersebut. Setelah itu sistem akan mengarahkan *player* ke halaman pembayaran untuk menyelesaikan proses bergabung. Setelah berhasil, sistem akan memastikan *player* berada pada halaman kompetisi yang diikutinya. Pengujian dilakukan dengan membuat suatu *test case* seperti pada gambar 4.50 berikut.

```

public function testJoinTournament()
{
    $this->browse(function (Browser $browser) {
        $browser->loginAs(User::find(6))
            ->visit('/')
            ->click('#detail-tournament-test')
            ->click('#join-tournament-test')
            ->assertRouteIs('tournament.checkout')
            ->assertSee('Process Transaction')
            ->assertRouteIs('tournament.show',
                'tournament-test');
    });
}

```

Gambar 4.50 Kode pengujian bergabung kompetisi

Berdasarkan pada pengujian yang dilakukan terhadap 8 *test case* yaitu pengujian register, login, logout, menambah game, membuat tim, bergabung dengan tim, membuat kompetisi dan bergabung dengan kompetisi. Didapatkan hasil pengujian seperti pada gambar 4.51 berikut.

```

C:\WINDOWS\system32\cmd.exe
c:\xampp\htdocs\arenasia>php artisan dusk
Warning: TTY mode is not supported on Windows platform.
PHPUnit 8.5.5 by Sebastian Bergmann and contributors.

DevTools listening on ws://127.0.0.1:49338/devtools/browser/fc93c6d0-4f13-4af3-ac67-d06bb31797ab
.
DevTools listening on ws://127.0.0.1:49385/devtools/browser/15cab151-5a77-4aef-9251-9cd55214801c
.
DevTools listening on ws://127.0.0.1:49429/devtools/browser/023bb17e-0fca-430c-909d-a2a2c500fea1
.
DevTools listening on ws://127.0.0.1:49467/devtools/browser/6da9aa1d-b776-41ca-9ebb-a51ab743e5f2
.
DevTools listening on ws://127.0.0.1:49507/devtools/browser/92c83021-d7fc-4fd2-b707-9f4fb3df0026
.
DevTools listening on ws://127.0.0.1:49575/devtools/browser/6268d3f8-a7c7-48e4-a6c7-7eae0c531229
.
DevTools listening on ws://127.0.0.1:49584/devtools/browser/fa96ebf5-9fa0-4c00-9d73-478e2d6934ea
.
DevTools listening on ws://127.0.0.1:49593/devtools/browser/169da9fc-9b76-4fdf-96a9-10bc324ee47b
.
DevTools listening on ws://127.0.0.1:49673/devtools/browser/eb58c78b-9561-4cc2-a696-3a2cb77a066b
.
DevTools listening on ws://127.0.0.1:49682/devtools/browser/c54ded6f-00e9-4ede-8eaa-3274806f9082
.
DevTools listening on ws://127.0.0.1:49693/devtools/browser/805c0264-780b-4f74-ac13-39c02458be89
.
DevTools listening on ws://127.0.0.1:49864/devtools/browser/c9c9fbba-150b-4361-8df0-20d9f9125768
.
DevTools listening on ws://127.0.0.1:49873/devtools/browser/1d8dea5f-adf3-4fa1-aff3-b64dae3d0af1
.
8 / 8 (100%)

Time: 2.14 minutes, Memory: 24.00 MB

OK (8 tests, 15 assertions)

c:\xampp\htdocs\arenasia>

```

Gambar 4.51 Hasil pengujian

Pengujian terhadap 8 *test case* membutuhkan waktu 2,14 menit dengan konsumsi memori sebesar 24 *megabytes*. Hasil pengujian menunjukkan status “OK” yang menandakan bahwa pengujian ini berhasil dilakukan tanpa *error*, “8 test” yang berarti telah dilakukan pengujian

terhadap 8 *test case*, dan “15 assertions” yang berarti selama proses pengujian berhasil membuktikan 15 kondisi benar.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pembangunan *platform* kompetisi *esports* Arenasia berbasis web menggunakan metode Rapid Application Development (RAD) dapat diambil beberapa kesimpulan sebagai berikut:

- a. Berdasarkan pada proses perancangan back-end yang telah dilakukan, mulai dari perancangan *use case diagram* hingga proses pengujian, pembangunan web Arenasia telah diselesaikan dalam kurun waktu 9 minggu menggunakan metode *rapid application development*.
- b. Pembangunan website Arenasia dilakukan dengan menggunakan *framework* Laravel dan basis data MySQL dan dapat diakses melalui alamat <http://arenasia.masuk.id>.
- c. Berdasarkan pada hasil implementasi, fitur migrasi Laravel mempermudah serta mempercepat penulis dalam pengimplementasian *entity relationship diagram* ke dalam basis data MySQL. Selain itu, penggunaan *payment gateway* Midtrans mempersingkat proses transaksi dalam sistem baik itu dari segi proses bisnis ataupun pemrograman.
- d. Berdasarkan pada hasil pengujian, tidak ditemukan masalah pada *back-end* web Arenasia.

5.2 Saran

Penulis menyadari bahwa pengembangan *platform* kompetisi *esports* berbasis web menggunakan metode *rapid application development* ini masih jauh dari kata sempurna. Oleh sebab itu, saran untuk pengembangan atau penelitian selanjutnya agar mendapat hasil yang lebih baik adalah:

- a. Melakukan normalisasi pada desain basis data.
- b. Mengintegrasikan sistem dengan API setiap game untuk mendapatkan informasi seperti statistik *player*, *match* ataupun informasi pembaruan *game*.
- c. Menggunakan metode atau *software* berbeda dalam mengembangkan web, yang mendukung kecepatan pembangunan yang lebih baik.

DAFTAR PUSTAKA

- Britton, C., & Doake, J. (2001). *Object-Oriented Systems Development*. Singapore: Mcgraw Hill.
- Chusyairi, A., Rudiawan, B., & Sasongko, E. (2018). Layanan Laporan Kehilangan (E-Report) Dengan Metode Rapid Application Development. *Jurnal Telematika*, 11.
- Di, N. (2014). Design of hospital website based on PHP technology [J]. In *China health vision: medical edition* (Vol. III, pp. 175-176).
- Errin, G. (2004, September 25). *Why startups fail, according to their founders*. Retrieved from Fortune Magazine.
- Eyer, M. K., & Field, M. (1999, November). *United States of America Patent No. 5,982,445*.
- Flanagan, D. (2006). *JavaScript: the definitive guide*. O'Reilly Media, Inc.
- Hakim, L., & Musalini, U. (2004). *Cara Mudah Memadukan Web Design dan Web Programming*. Jakarta: PT Elex Media Komputindo.
- Kendall, J. E. (2010). *Analisis dan Perancangan Sistem*. Jakarta.
- Lie, Wium, H., & Bos, B. (2005). *Cascading style sheets: Designing for the web, Portable Documents*. Addison-Wesley Professional.
- Mazinanian, D., Tsantalis, N., & Mesbah, A. (2014). Discovering refactoring opportunities in cascading style sheets. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, (pp. 496-506).
- Pomaska, G. (2012). *Webseiten-Programmierung*. Wiesbaden: Springer Vieweg.
- Robehmed, N. (2013, December 16). *What is A Startup?* Retrieved from Forbes.
- Schmitt, A., Rosing, K., Zhang, S. X., & Leatherbee, M. (2018). A dynamic model of entrepreneurial uncertainty and business opportunity identification: exploration as a mediator and entrepreneurial self-efficacy as a moderator. *Entrepreneurship Theory and Practice*, 835-859.
- Shah, M. (2015). *Responsive web development using the Twitter Bootstrap framework*.
- Solichin, A. (2016). *Pemrograman web dengan PHP dan MySQL*. Penerbit Budi Luhur.
- Spurlock, J. (2013). *Bootstrap: Responsive Web Development*. O'Reilly Media, Inc.
- Suehring, S. (2002). *MySQL bible*. John Wiley & Sons, Inc.
- Surguy, M. (2013). *History of Laravel PHP framework, Eloquence emerging*. Retrieved from Maxoffsky.

- Wagner, M. G. (2006). On the Scientific Relevance of eSports. *International conference on internet computing*, (pp. 437-442).
- Young, K. (2009). Understanding online gaming addiction and treatment issues for adolescents. *The American Journal of Family Therapy*, 355-372.
- Zheng, H. (2014). Design and Realization of Online Voting System through PHP [J]. In *Jilin Normal University Journal (Natural Science Edition)* (Vol. I, pp. 78-80).
- Zhong, H., Liu, C., Zhong, J., & Xiong, H. (2008). Which startup to invest in: a personalized portfolio strategy. *Annals of Operations Research*, 339-360.

