

APLIKASI GENERATOR TURNAMEN SEPAK BOLA



الجامعة الإسلامية
الاستد بالاندونيسية

Disusun Oleh:

N a m a : Muhammad Radin Pradipta

NIM : 15523026

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

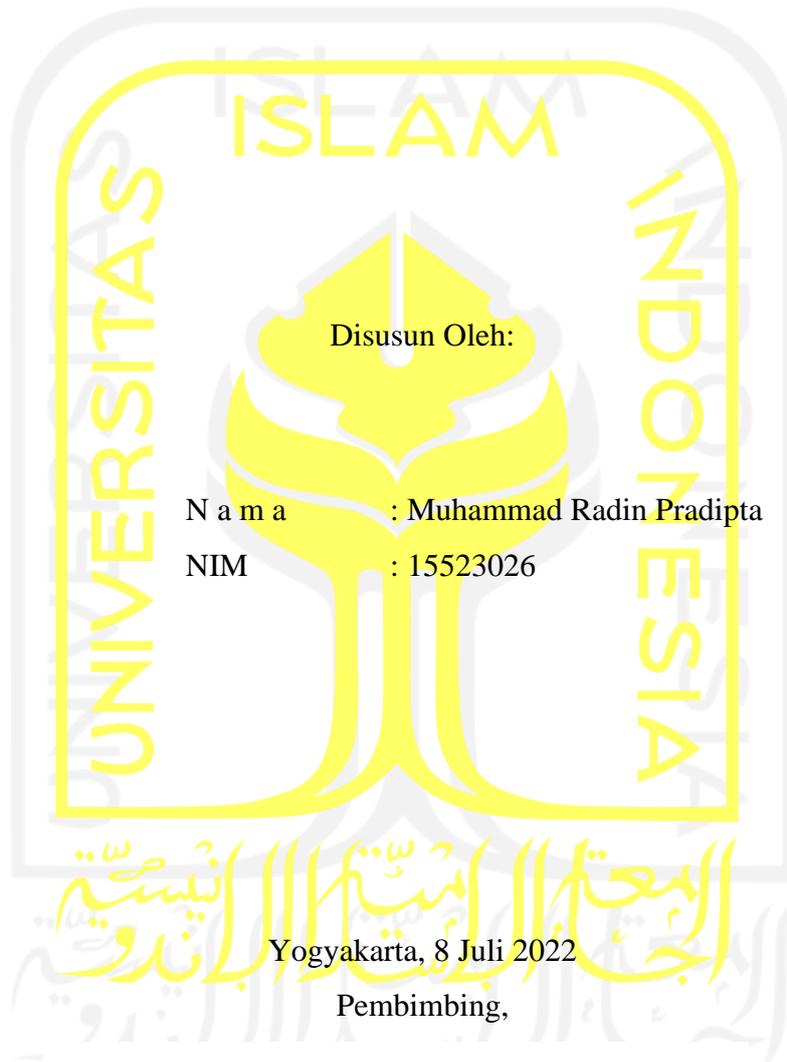
UNIVERSITAS ISLAM INDONESIA

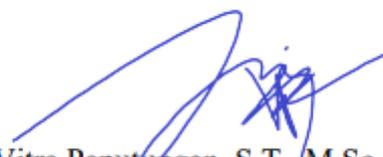
2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

APLIKASI GENERATOR TURNAMEN SEPAK BOLA

TUGAS AKHIR




05/09/2022
(Irving Vitra Paputungan, S.T., M.Sc., Ph.D.)

HALAMAN PENGESAHAN DOSEN PENGUJI

APLIKASI GENERATOR TURNAMEN SEPAK BOLA

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, 21 Juli 2022

Tim Penguji

05/09/2022

Irving Vitra Papatungan, M.Sc., Ph.D.

ANGGOTA 1

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

ANGGOTA 2

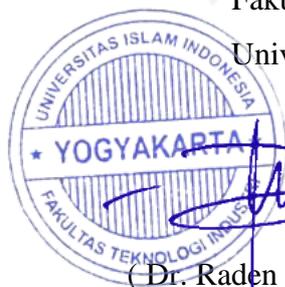
Galang Prihadi Mahardhika, S.Kom., M.Kom.

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 : Muhammad Radin Pradipta

NIM : 15523026

Tugas akhir dengan judul:

APLIKASI GENERATOR TURNAMEN SEPAK BOLA

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, 4 Juli 2022



(Muhammad Radin Pradipta)

HALAMAN PERSEMBAHAN

Alhamdulillah Saya panjatkan kepada kehadiran Allah SWT, atas segala rahmat dan juga kesempatan dalam menyelesaikan tugas akhir saya yang berjudul “Aplikasi Generator Turnamen Sepak Bola” dengan segala kekurangannya. Dan Segala puji dan syukur bagi Allah SWT, karena berkat rahmat dan hidayahNya Saya dihadirkan dengan individu-individu yang senantiasa selalu memberikan doa dan semangatnya sehingga tugas akhir saya ini dapat diselesaikan dengan baik dan pada waktunya. Tugas akhir ini saya persembahkan untuk:

1. Ayah Kurnianto Trubus Pranowo dan Ibu Umi Mutmainah Sri Susilowati.
2. Rekan seperjuangan Saya di jurusan Informatika FTI UII.
3. Dosen pembimbing bapak Irving Papatungan, S.T., M.Sc.
4. Jurusan Informatika Universitas Islam Indonesia.
5. Serta seluruh pihak yang tidak bisa saya sebutkan satu-persatu.



HALAMAN MOTO

“Don't judge a book by its cover”

(George Eliot)



KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmat dan karunia-Nyalah penulis dapat menyelesaikan laporan tugas akhir ini sebagai salah satu syarat untuk dapat mengikuti ujian pendaran di Jurusan Informatika Fakultas Teknik Industri Universitas Islam Indonesia. Penulis menyadari, tanpa dukungan motivasi eksternal laporan ini tidak akan dapat selesai tepat waktu. Oleh karena itu, penulis menyampaikan rasa terimakasih yang sebanyak-banyaknya kepada:

1. Ayah Kurnianto Trubus Pranowo dan Ibu Umi Mutmainah Sri Susilowati.
2. Rekan-rekan Saya di jurusan Informatika Fakultas Teknik Industri UII.
3. Dosen pembimbing bapak Irving Papatungan, S.T., M.Sc.
4. Jurusan Informatika Universitas Islam Indonesia.
5. Serta seluruh pihak yang tidak bisa saya sebutkan satu-persatu.

Penulis menyadari bahwa penyusunan skripsi ini jauh dari kata sempurna, semoga Allah SWT memberikan balasan yang berlipat ganda kepada pihak-pihak yang telah turut membantu penulis dalam menyelesaikan penulisan laporan tugas akhir ini.

Yogyakarta, 4 Juli 2022



(Muhammad Radin Pradipta)

SARI

Salah satu fungsi dan tujuan utama dari teknologi informasi adalah untuk memecahkan suatu masalah dan meningkatkan efektivitas dan efisiensi pekerjaan manusia. Namun pemanfaatan teknologi informasi dalam hal pengelolaan turnamen sepak bola di Indonesia masih belum adekuat. Hal ini bisa dilihat dari masih belum banyak dan bahkan belum adanya sebuah sistem atau aplikasi yang mampu mengelola jalannya sebuah turnamen secara otomatis padahal sepak bola adalah olah raga paling populer di Indonesia. Pada penelitian ini penulis berusaha memfasilitasi dan memudahkan pengelolaan sebuah turnamen sepak bola dengan cara mengembangkan sebuah aplikasi berbasis web yang dapat *men-generate* secara otomatis bagan dari turnamen-turnamen yang telah dibuat yang nantinya dapat dikelola untuk memudahkan jalannya sebuah turnamen sepak bola. Hasil penelitian ini berupa aplikasi berbasis web yang menggunakan arsitektur *MVC (Model View Controller)* berbasiskan pada bahasa pemrograman PHP dan dikembangkan menggunakan *framework* PHP Codeigniter yang mampu membuat bagan turnamen eliminasi tunggal.

Kata kunci: teknologi informasi, turnamen, sepak bola, sistem informasi, aplikasi generator, *MVC*, Codeigniter, PHP

GLOSARIUM

Bracket	bagan berupa diagram berbentuk pohon yang berisi detail skor pertandingan pertandingan yang sedang berjalan pada sebuah turnamen.
Controller	salah satu bagian dari arsitektur MVC atau <i>model view controller</i> yang bertugas untuk megnhubungkan <i>view</i> dengan <i>model</i>
Model	salah satu bagian dari arsitektur MVC atau <i>model view controller</i> yang bertugas untuk manipulasi database
View	salah satu bagian dari arsitektur MVC atau <i>model view controller</i> yang bertugas untuk menampilkan informasi dengan antarmuka grafis (GUI)
Waterfall	metode pengembangan perangkat lunak dimana fase-fase pengembangannya menyereupai air terjun
Turnamen	pertandingan (tenis dan sebagainya) yang diikuti oleh beberapa regu
Generator	sebuah aplikasi yang mampu men- <i>generate</i> sebuah fungsi untuk tujuan tertentu

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.....	VIII
GLOSARIUM.....	IX
DAFTAR ISI.....	X
DAFTAR TABEL	XII
DAFTAR GAMBAR	XIII
BAB I PENDAHULUAN	14
1.1 Latar Belakang Masalah	14
1.2 Rumusan Masalah.....	15
1.3 Batasan Masalah	15
1.4 Tujuan Penelitian	15
1.5 Manfaat Penelitian	15
1.6 Sistematika Penulisan	16
BAB II LANDASAN TEORI.....	17
2.1 Sepak Bola	17
2.2 Turnamen	18
2.2.1 <i>Knockoff / Elimination</i>	18
2.2.2 <i>Challenge</i>	18
2.2.3 <i>Round Robin</i>	20
2.2.4 <i>Multi-stage Tournament</i>	20
2.3 Metodologi Pengembangan Sistem.....	20
2.3.1 <i>Waterfall</i>	21
2.3.2 <i>Agile Development</i>	22
2.4 <i>Parser</i>	23
2.5 Aplikasi Generator	24
BAB III ANALISIS DAN PERANCANGAN	25
3.1 Analisis Kebutuhan.....	27

3.1.1	Analisis Kebutuhan Fungsional	27
3.1.2	Analisis Kebutuhan Non Fungsional	27
3.1.3	Analisis Kebutuhan Antar Muka	27
3.2	Perancangan	27
3.2.1	Perancangan <i>Use Case Diagram</i>	27
3.2.2	<i>Activity diagram</i>	28
3.2.3	Perancangan Data.....	30
BAB IV IMPLEMENTASI DAN PEMBAHASAN		35
4.1	Implementasi.....	35
4.1.1	Implementasi Antarmuka.....	35
4.1.2	Implementasi Kode Program	39
4.2	Skenario Pembuatan Turnamen	42
4.2.1	Bracket turnamen non komplit.....	42
4.2.2	Bracket turnamen komplit.....	45
4.3	Pengujian.....	46
4.3.1	Pengujian pembuatan turnamen.....	47
4.3.2	Pengujian kelola turnamen.....	47
BAB V KESIMPULAN DAN SARAN.....		49
5.1	Kesimpulan	49
5.2	Saran	49
DAFTAR PUSTAKA.....		50

DAFTAR TABEL

Tabel 3.1 Tabel turnamen	30
Tabel 4.1 Tabel pengujian pembuatan turnamen	47
Tabel 4.2 Tabel pengujian lihat turnamen	47



DAFTAR GAMBAR

Gambar 2.1 Bagan Lapangan Sepak Bola (IFAB, 2021).....	17
Gambar 2.2 Turnamen Piramida.....	19
Gambar 2.3 Bagan <i>round robin tournament</i>	20
Gambar 2.4 <i>Waterfall development</i>	21
Gambar 3.1 Tahapan-tahapan Pengembangan Aplikasi	25
Gambar 3.2 Use Case Diagram Aplikasi Turnamen Sepakbola	28
Gambar 3.3 <i>Activity diagram</i> Buat Turnamen	29
Gambar 3.4 <i>Activity diagram</i> Mengelola turnamen.....	29
Gambar 3.5 JSON turnamen 4 tim.....	31
Gambar 3.6 JSON turnamen 8 tim.....	32
Gambar 3.7 JSON turnamen 16 tim.....	33
Gambar 3.8 JSON turnamen 15 tim.....	34
Gambar 4.1 Tampilan antarmuka halaman utama	35
Gambar 4.2 Tampilan menu pembuatan turnamen dan daftar turnamen.....	36
Gambar 4.3 Tampilan antarmuka buat turnamen	36
Gambar 4.4 Tampilan antarmuka daftar turnamen	37
Gambar 4.5 Tampilan antarmuka detail turnamen	38
Gambar 4.6 Kode program <i>create tournament</i>	39
Gambar 4.7 Kode program untuk menampilkan <i>bracket</i>	40
Gambar 4.8 Kode program penampilan jadwal	41
Gambar 4.9 Kode program penyimpanan detail turnamen.....	42
Gambar 4.10 <i>Bracket</i> 3 tim.....	43
Gambar 4.11 <i>Bracket</i> 7 tim.....	43
Gambar 4.12 <i>Bracket</i> 13 tim.....	44
Gambar 4.13 <i>Bracket</i> 4 tim.....	45
Gambar 4.14 <i>Bracket</i> 8 tim.....	45
Gambar 4.15 <i>Bracket</i> 16 tim.....	46
Gambar 4.16 Contoh entri turnamen yang tidak valid.....	48
Gambar 4.17 <i>Bug</i> pada tampilan nama tim.....	48

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Sebagai olahraga paling populer di dunia dengan ratusan bahkan miliaran penggemar, banyak turnamen maupun liga sepak bola yang diselenggarakan di dunia. Turnamen–turnamen dan liga-liga tersebut khususnya yang diselenggarakan di Inggris dan neger–negeri Eropa mempunyai banyak sekali penggemar yang ingin selalu ter-*update* perihal tim kesayangannya seperti: skor pertandingan terakhir, pertandingan selanjutnya, *starting line-up* yang akan dimainkan, siapa saja tim lawan, dan lain sebagainya. Pada penelitian ini penulis berusaha memberikan solusi dengan membuat sebuah aplikasi yang bertujuan untuk memudahkan para fans tersebut untuk dapat ter-*update* dengan tim kesayangannya.

Sepak bola adalah olahraga yang paling disukai dan digemari oleh masyarakat Indonesia dan dunia. Hal ini dapat dilihat dari jumlah penonton pertandingan, baik secara langsung di stadion - stadion maupun melalui siaran televisi langsung ataupun siaran televisi ulang. Oleh karena itu masyarakat penggemar sepak bola membutuhkan akses informasi tentang liga -liga atau turnamen - turnamen sepak bola dengan cepat dan efisien. Sehubungan dengan hal tersebut diperlukanlah suatu sistem informasi atau aplikasi yang mampu mendukung dan memfasilitasi pengelolaan informasi dan hasil pertandingan liga -liga maupun turnamen - turnamen sepak bola. Pengelolaan data yang baik dan sistematis, dapat menciptakan informasi yang relevan. Dengan cara tersebut, proses manajemen dan pengelolaan dapat dilakukan dengan lebih efektif dan efisien untuk menghasilkan informasi yang bermanfaat dan relevan.

Penjadwalan pertandingan sangat mempengaruhi tingkat kesuksesan dan mutu suatu kompetisi serta kepuasan para penggemar. Kompetisi sepakbola setiap tahunnya menganut sistem *double round-robin* atau biasa disebut dengan kompetisi penuh, masing-masing klub bertanding dengan klub yang sama sebanyak dua kali: sekali home atau bermain sebagai tuan rumah dan sekali away atau bermain sebagai tamu. Sedangkan untuk sistem setengah kompetisi, umumnya digunakan pada babak penyisihan, seperti Piala Dunia, Liga Champions dan sebagainya.

Kompetisi atau turnamen sepakbola di dunia digelar setiap tahunnya. Banyak orang yang ingin mengetahui informasi terkini mengenai update score disaat pertandingan berlangsung, klasemen sementara dan jadwal pertandingan. Namun permasalahan yang terjadi yaitu informasi ini seringkali terhambat penyampaiannya karena penggemar tidak sempat

menyaksikan langsung atau karena sedang dalam perjalanan serta terbatasnya akses media seperti televisi, tabloid atau majalah olahraga. Adapun sebagian penggemar sepakbola telah menyediakan jadwal seluruh pertandingan pada suatu turnamen atau liga sepakbola, namun dalam kesibukan pekerjaan atau aktifitas yang menuntut sehingga mengakibatkan penggemar tersebut kurang update pertandingan terkini. Turnamen yang diselenggarakan selama satu periode biasanya memakan waktu antara 6 sampai dengan 12 bulan dan memerlukan update data tiap pekannya. Untuk meningkatkan pengelolaan data demi menghasilkan informasi yang tepat, akurat, dan relevan serta terpercaya dibutuhkanlah sebuah sistem informasi dan komputer untuk mendukung proses tersebut.

1.2 Rumusan Masalah

Dengan latar belakang yang telah diuraikan di atas maka dapat dirumuskan permasalahan sebagai berikut:

- a. Bagaimana cara membuat sebuah aplikasi yang dapat membuat bagan/*bracket* turnamen sepakbola secara otomatis?
- b. Bagaimana mendesain aplikasi yang minimalis, lengkap, mudah diakses, dan mudah dibaca oleh pengguna?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

- a. Aplikasi ini hanya menggunakan sistem turnamen eliminasi tunggal.
- b. Aplikasi ini tidak memiliki sistem login.
- c. Asumsi hanya terdapat 1 lapangan.

1.4 Tujuan Penelitian

Tujuan dibuatnya karya tulis atau penelitian ini adalah membangun sebuah aplikasi generator turnamen sepak bola dengan desain yang minimalis yang dapat dengan mudah membuat, mengelola, dan menjadwalkan secara otomatis jalannya sebuah turnamen sepak bola.

1.5 Manfaat Penelitian

Dengan dibuatnya karya tulis ini diharapkan dapat memberi manfaat antara lain:

- a. Mempermudah pengadaan dan penyelenggaraan sebuah turnamen sepak bola.

- b. Mendorong lebih banyak lagi diadakannya turnamen sepak bola karena penyelenggaraanya sudah dipermudah dengan kehadiran aplikasi ini.

1.6 Sistematika Penulisan

Dalam penyusunan, karya tulis ini akan terbagi menjadi beberapa bab, yaitu:

- a. Bab I Pendahuluan

Bab ini berisi pemaparan latar belakang, perumusan masalah, batasan masalah, serta tujuan dari penelitian ini.

- b. Bab II Landasan Teori

Bab ini berisi landasan teori-teori dan tinjauan pustaka yang digunakan pada penelitian ini.

- c. Bab III Analisis dan Perancangan

Bab ini berisi analisis kebutuhan dan perancangan aplikasi yang akan dikembangkan.

- d. Bab IV Hasil dan Pembahasan

Bab ini berisi pemaparan dan pembahasan hasil capaian pada penelitian ini.

- e. Bab V Kesimpulan dan Saran

Pada bab ini akan dipaparkan kesimpulan dari pencapaian penelitian ini dan saran yang dapat digunakan untuk penelitian selanjutnya.

BAB II LANDASAN TEORI

2.1 Sepak Bola

Sepak bola merupakan salah satu cabang olahraga yang dimainkan dengan 2 tim, setiap tim dalam permainan sepak bola terdiri dari 11 orang. Sepak bola dimainkan di lapangan berbentuk persegi empat dengan gawang di setiap ujungnya. Pada gambar 2.1 dapat dilihat bagan dari sebuah lapangan sepak bola.



Gambar 2.1 Bagan Lapangan Sepak Bola (IFAB, 2021).

Selain para pemain terdapat juga beberapa orang yang bertugas menjaga jalannya permainan agar sesuai dengan peraturan yang berlaku di antaranya: 1 orang wasit utama, 2 orang asisten wasit, 1 orang *fourth official* atau wasit cadangan, 2 orang asisten wasit tambahan,

1 orang cadangan asisten wasit, 1 orang *Video Assitant Referee (VAR)*, dan setidaknya 1 orang *Assistant VAR (AVAR)* (IFAB, 2021).

Alur permainan sepak bola secara garis besar adalah setiap tim berusaha untuk memasukkan bola ke gawang lawan sebanyak-banyaknya dalam kurun waktu yang ditentukan biasanya 90 menit. Setiap bola yang masuk ke dalam gawang lawan disebut sebagai gol dan dihitung sebagai 1 skor untuk tim yang memasukkan bola tersebut. Bola boleh dimainkan dengan seluruh anggota badan kecuali kedua tangan pemain. 1 pemain dari setiap tim bertugas sebagai penjaga gawang, pemain tersebut boleh menggunakan tangannya untuk memainkan bola namun hanya sebatas di dalam area yang sudah ditentukan. Tim yang memasukkan gol lebih banyak sebelum pertandingan berakhir keluar sebagai pemenang.

2.2 Turnamen

Menurut Kamus Besar Bahasa Indonesia (KBBI) (2022) turnamen adalah pertandingan (tenis dan sebagainya) yang diikuti oleh beberapa regu. Regu-regu atau pemain-pemain tersebut akan bertanding satu sama lainnya sesuai dengan format turnamen yang diikuti. Pemenang akan ditentukan jika terdapat regu yang telah mengalahkan semua lawan atau memperoleh poin tertinggi, tergantung dengan format turnamen. Berikut merupakan format-format turnamen yang ada

2.2.1 *Knockoff / Elimination*

Knockoff atau sistem gugur dalam Bahasa Indonesia adalah sistem turnamen dimana peserta yang kalah akan langsung tereliminasi dari turnamen. Terdapat 2 jenis sistem gugur, yang pertama adalah *single elimination* dimana peserta akan bermain 1 kali dan peserta yang kalah akan langsung gugur dari turnamen, dan yang kedua adalah *double elimination* dimana peserta akan bermain sebanyak 2 kali dan peserta yang lebih banyak kalah dalam pertandingan akan di eliminasi dari turnamen (Sziklaia et al., 2021).

2.2.2 *Challenge*

Terdapat 2 jenis tipe turnamen *Challenge*, yang pertama adalah *ladder* (Sportmeets, 2017), peserta akan diacak dan diatur sedemikian rupa sehingga menyerupai anak tangga.

Pada turnamen *ladder* peserta dalam hal ini penantang dapat mengajukan tantangan yang sebaiknya tidak ditolak kepada peserta lain 1 atau 2 tingkat di atas penantang tersebut. Pada gambar 2.2, dapat dilihat pada baris keempat dari atas Rafael Nada dapat mengajukan

tantangan pada Roger Federo yang berada di baris yang sama (keempat) ataupun Isabelle Welldone yang berada di baris ketiga. Jika penantang menang maka penantang akan bertukar posisi dengan yang peserta yang kalah. Pemenang turnamen adalah peserta yang berada di posisi paling tinggi ketika batas waktu yang sudah ditentukan sudah habis.

Tipe turnamen *challenge* yang kedua adalah *pyramid* atau piramida (Byl, 2005). Hampir mirip dengan *ladder* peserta akan diacak dan diatur sedemikian rupa posisinya sehingga akan menyerupai piramida.



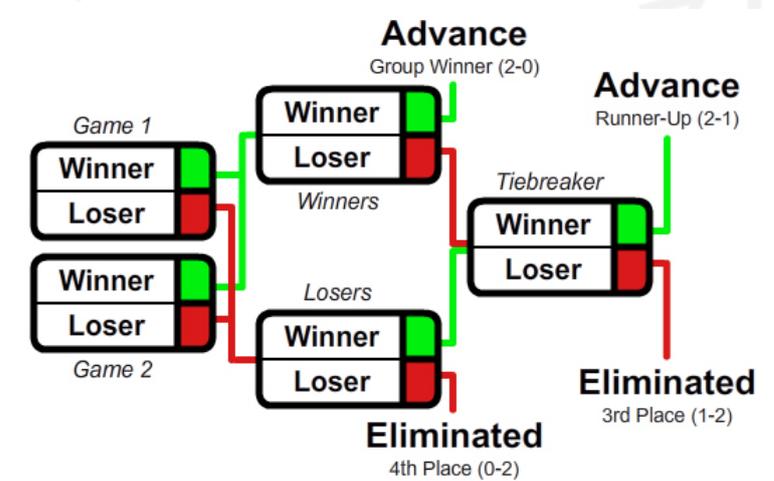
Gambar 2.2 Turnamen Piramida

Pada turnamen format piramida peserta pada lapisan bawah hanya dapat mengajukan tantangan pada peserta yang berada di lapisan tepat di atas penantang tersebut dan apabila menang akan bertukar posisi dengan peserta yang kalah. Sama dengan turnamen *ladder*, pemenang turnamen adalah peserta yang berada di posisi paling tinggi ketika batas waktu yang sudah ditentukan sudah habis.

Turnamen format *challenge* biasanya digunakan untuk olahraga: Tenis, Badminton, dan Skuas.

2.2.3 Round Robin

Biasa juga disebut *league* atau liga, setiap peserta akan bertanding dengan semua peserta turnamen sebanyak 1 atau 2 kali. Hasil kalah, menang, maupun imbang akan mendapatkan poin. Pemenang adalah peserta yang memperoleh poin tertinggi (Sziklaia et al., 2021). Liga biasa digunakan pada olahraga sepak bola. Gambar 2.3 dibawah merupakan bagan dari turnamen menggunakan *format round-robin*.



Gambar 2.3 Bagan *round robin tournament*

2.2.4 Multi-stage Tournament

Merupakan gabungan dari beberapa format turnamen, biasanya digunakan jika peserta turnamen cenderung banyak. Contoh dari turnamen format multi-stage adalah FIFA World Cup yang merupakan gabungan antara format knockoff dengan format round robin (IFAB, 2021).

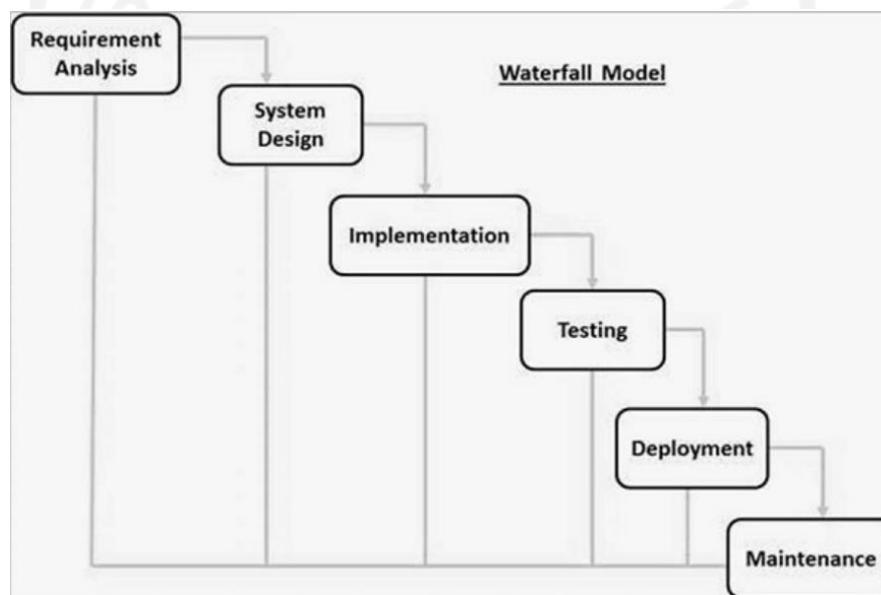
2.3 Metodologi Pengembangan Sistem

Metodologi pengembangan sistem adalah sebuah pendekatan formal untuk pengimplementasian SDLC atau *System Life Development Cycle*. Terdapat beberapa cara untuk mengkategorikan sebuah metodologi yaitu yang metodologi terfokus pada proses bisnis atau yang terfokus pada data yang mendukung bisnis. Pada penelitian ini penulis akan menggunakan

metodologi berorientasi objek untuk menggabungkan fokus proses bisnis dan data menjadi satu buah model yang efisien.

2.3.1 Waterfall

Teknik Waterfall, juga dikenal sebagai model air terjun, adalah proses pengembangan berurutan yang jatuh seperti air terjun melalui semua fase proyek (misalnya: analisis, desain, pemrograman, dan pengujian). Pada gambar 2.4 dibawah merupakan fase-fase pada pengembangan sistem menggunakan Teknik *waterfall*.



Gambar 2.4 *Waterfall development*

Menurut Fagarasan et al. (2021) berikut merupakan fase-fase pada *Waterfall* SDLC (*System Development Life Cycle*):

a. *Requirement Analysis*

Pengumpulan dan Analisis Kebutuhan Semua kemungkinan persyaratan sistem yang akan dikembangkan ditangkap dalam fase ini dan didokumentasikan dalam dokumen spesifikasi kebutuhan.

b. *System Design*

Kebutuhan sistem dari fase pertama dipelajari dalam fase ini dan desain sistem disiapkan. Desain sistem ini membantu dalam menentukan perangkat keras dan

persyaratan sistem dan membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

c. *Implementation*

Dengan masukan dan saran dari desain sistem, sistem pertama kali dikembangkan dalam program kecil yang disebut unit, yang diintegrasikan pada fase berikutnya. Setiap unit dikembangkan dan diuji fungsinya, yang disebut sebagai Unit Testing.

d. *Testing*

Semua unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah setiap unit sudah diuji.

e. *Deployment*

Penerapan sistem Setelah pengujian fungsional dan non-fungsional selesai; produk akan di rilis ke *end-user* atau di rilis ke pasar.

f. *Maintenance*

Jika ada beberapa masalah yang muncul setelah rilis, *patch* akan dirilis. Juga untuk meningkatkan produk beberapa versi yang lebih baik juga akan dirilis. Pemeliharaan dilakukan untuk menyampaikan perubahan ini pada *end-user*.

Teknik Waterfall adalah metodologi manajemen proyek yang terdefinisi dengan baik dan tidak rumit. Semua fase ini mengalir satu sama lain di mana kemajuan terlihat mengalir terus ke bawah (seperti air terjun). Fase berikutnya dimulai jika goal-goal tertentu pada fase sebelumnya telah tercapai. Karena persyaratan ditulis dari awal, setiap kontributor tahu persis apa yang perlu dilakukan kapan dan dapat merencanakan waktunya secara efektif untuk jalannya sebuah proyek.

2.3.2 *Agile Development*

Agile merupakan pendekatan teriterasi untuk manajemen sebuah proyek dan pengembangan perangkat lunak yang membantu tim merilis aplikasi kepada kostumer secara lebih cepat. Alih-alih mempertaruhkan segalanya pada satu peluncuran utama, tim *agile* memberikan aplikasi melalui rilis-rilis kecil, tetapi dapat langsung digunakan. Kebutuhan, rencana, dan hasil dievaluasi terus menerus sehingga tim memiliki mekanisme alami untuk merespons perubahan dengan cepat. Menurut Beck et al. (2001), agile memiliki 12 prinsip dasar, berikut ke 12 prinsip dasar tersebut:

- a. Software dikembangkan secara kontinyu dan di rilis lebih cepat demi kepuasan kostumer.
- b. Revisi kebutuhan tidak masalah bahkan saat proyek sudah mencapai tahap akhir pada pengembangannya.
- c. Merilis *software* yang terupdate setiap harinya secara berkala.
- d. Adanya kerjasama dan kesinambungan antara kostumer dan pengembang dalam penyelesaian masalah yang dihadapi.
- e. *Software* dikembangkan oleh individu – individu yang termotivasi dan terfasilitasi dengan baik.
- f. Tatap muka secara langsung merupakan cara komunikasi terbaik.
- g. *Progress* di ukur dari sejauh mana *software* telah memenuhi kebutuhan kostumer.
- h. *Working pace* atau laju kerja yang konsisten harus dipertahankan.
- i. Atensi yang terus menerus diberikan terhadap keunggulan teknis dan desain yang baik dapat meningkatkan *agility* dari sebuah proyek.
- j. Kesederhanaan adalah hal yang diutamakan. Hindari melakukan hal – hal yang tidak diperlukan dan justru malah akan mempersulit proyek.
- k. Tim yang dapat menorganisir dirinya sendiri akan dapat menghasilkan arsitektur, desain, dan kebutuhan yang terbaik.
- l. Tim harus dan akan berusaha secara terus menerus untuk dapat meningkatkan efektivitas dengan cara penyesuaian cara kerja selama ini.

Metode *agile* lebih terfokus pada perampingan proses pengembangan sebuah *software* dengan cara menghilangkan dan menghindari pemodelan dan dokumentasi dan waktu yang dibutuhkan untuk menyelesaikan *task-task* tersebut. Pada metode *agile* proses pengembangan dibuat sesederhana mungkin walaupun juga melewati fase-fase tradisional dari proses pengembangan sebuah sistem (SLDC).

2.4 Parser

Parser adalah komponen *compiler* atau *interpreter* yang memecah data menjadi elemen-elemen yang lebih kecil agar mudah diterjemahkan ke dalam bahasa lain. Sebuah parser mengambil input dalam bentuk urutan token, perintah interaktif, atau instruksi program dan memecahnya menjadi bagian-bagian yang dapat digunakan oleh komponen lain dalam sebuah program (Technopedia, 2020).

Agar kode yang ditulis dalam bentuk yang dapat dibaca manusia dapat dipahami juga oleh mesin, kode tersebut harus diubah ke dalam bahasa mesin terlebih dahulu. Pekerjaan ini biasanya dilakukan oleh sebuah *translator* (*interpreter* atau *compiler*). Parser umumnya digunakan sebagai komponen penerjemah yang mengatur teks linier ke dalam struktur yang dapat dengan mudah dimanipulasi (pohon *parse*). Untuk melakukannya, *parser* mengikuti seperangkat aturan yang sudah ditetapkan yang biasa disebut "*grammar*" (Technopedia, 2020).

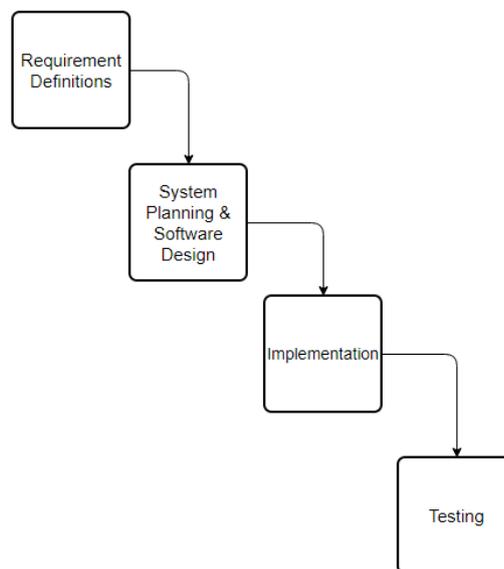
2.5 Aplikasi Generator

Aplikasi generator adalah sebuah perangkat lunak yang mampu membuat berbagai program atau aplikasi dalam domain-domain tertentu. Program yang dihasilkan akan dikonfigurasi dengan informasi yang diberikan oleh *user* yang menggunakan aplikasi generator tersebut. Domain di mana aplikasi generator sering ditemui termasuk: simulasi, proses kontrol, dan perangkat lunak antarmuka pengguna (Encyclopedia.com, 2022).

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas proses-proses perancangan sebuah penelitian, proses-proses tersebut dapat didefinisikan sebagai pembuatan, perencanaan dan penggambaran draft awal atau penyesuaian dari elemen-elemen yang terpisah untuk kemudian selanjutnya akan dibentuk menjadi sebuah kesatuan yang komplit dan mampu dialihfungsikan sebagai perancangan sebuah sistem. Disini peneliti memutuskan untuk memilih metode penelitian dengan merancang sebuah aplikasi bernama *toernig* menggunakan metode air terjun atau *waterfall*. Metode *waterfall* yang sering juga disebut dengan *classic life cycle* merupakan suatu metode pengembangan perangkat lunak yang berurutan, di mana *progress* dipandang sebagai terus mengalir ke bawah seperti air terjun secara bertahap. Tahapan-tahapan tersebut diantaranya: perencanaan, pemodelan, pengimplementasian, dan uji coba. Metode *waterfall* bisa dibilang adalah metode yang kuno akan tetapi justru hal tersebut malah menjadi hal positif karena kematangannya sudah teruji di lapangan, sehingga banyak peneliti yang memilih menggunakan metode *waterfall* ini dalam pengembangan sistem. Bisa dilihat dari gambar 3.1 tahapan-tahapan proses pengembangan *software* menggunakan metode air terjun.



Gambar 3.1 Tahapan-tahapan Pengembangan Aplikasi

- a. *Requirement Definitions* menjelaskan bagaimana peneliti melakukan pengembangan perangkat lunak, peneliti harus mengetahui dan memahami bagaimana informasi kebutuhan pengguna terhadap sebuah perangkat lunak. Metode yang diakumulasikan dalam bentuk informasi ini diperoleh dengan berbagai cara yaitu observasi, wawancara, survei, diskusi, dan lainnya. Setelah informasi yang dibutuhkan sudah cukup selanjutnya akan dianalisis dan diolah oleh peneliti terkait perangkat lunak yang akan dikembangkan.
- b. *Planning Systems & Software Design* disini menjelaskan tentang informasi mengenai spesifikasi kebutuhan dari tahap Requirement Definitions dan selanjutnya di analisa, pada tahap ini akan dilanjutkan dengan pengimplementasian ke desain pengembangan. Perancangan desain memiliki tujuan untuk membantu menyampaikan gambaran besar perihal perancangan yang harus dikerjakan. Tahap ini mampu menyokong peneliti untuk mempersiapkan kebutuhan pengguna dalam penyusunan struktur sistem perangkat lunak yang akan dibuat secara menyeluruh. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.
- c. *Implementation* merupakan tahap pemrograman, dalam pembuatan perangkat lunak dibagi menjadi modul-modul yang lebih kecil agar mudah dibangun dan hasil akhirnya akan digabungkan dalam tahap berikutnya. Tahap ini akan dilanjutkan ke proses pengujian dan pengecekan perihal fungsionalitas modul yang sudah dikerjakan, apakah sudah memenuhi tolok ukur yang diharapkan atau belum. Tahapan ini bisa dibidang sebagai tahapan proses penerjemahan dari bentuk desain menjadi kode/program atau bentuk bahasa yang mampu dibaca oleh mesin. Setelah proses pemrograman selesai, maka dilanjutkan ke pemeriksaan terhadap sistem dan juga program yang sudah dibuat. Tujuannya yaitu mencari kesalahan yang mungkin terjadi dan kedepannya bisa diperbaiki melalui update berkala.
- d. *Testing* adalah tahap yang dilakukan untuk pengujian dan pemeriksaan terhadap fungsionalitas modul yang sudah dibuat, apakah pengembangan aplikasi tersebut memenuhi kriteria yang sudah ditentukan atau belum. Setelah proses implementasi selesai, selanjutnya melakukan pengujian terhadap sistem dan juga kode yang sudah dibangun. Tujuannya untuk menemukan kesalahan yang mungkin terjadi disaat tahap pengemplementasian agar nantinya diperbaiki. Tahap terakhir yaitu melakukan pengecekan dan pengujian sistem secara menyeluruh untuk mencari kemungkinan adanya kegagalan dan kesalahan sistem.

3.1 Analisis Kebutuhan

Analisis kebutuhan digunakan sebagai penentuan kebutuhan-kebutuhan *end user* terhadap sistem. Analisis kebutuhan mencakup 3 hal yaitu analisis kebutuhan fungsional, non fungsional, dan antarmuka.

3.1.1 Analisis Kebutuhan Fungsional

Kebutuhan fungsional mencakup kebutuhan input, output dan proses sistem. Pada aplikasi ini terdapat 2 kebutuhan fungsional yaitu:

- a. Menampilkan turnamen yang sedang berjalan.
- b. Membuat turnamen sepakbola, dengan tipe single elimination.

3.1.2 Analisis Kebutuhan Non Fungsional

Kebutuhan nonfungsional jika dapat dikatakan batasan layanan sebuah sistem. Kebutuhan nonfungsional dapat digunakan sebagai acuan kualitas sebuah sistem. Berikut merupakan kebutuhan fungsional pada aplikasi ini:

- a. Sistem memiliki antarmuka yang mudah dipahami dan simpel.

3.1.3 Analisis Kebutuhan Antar Muka

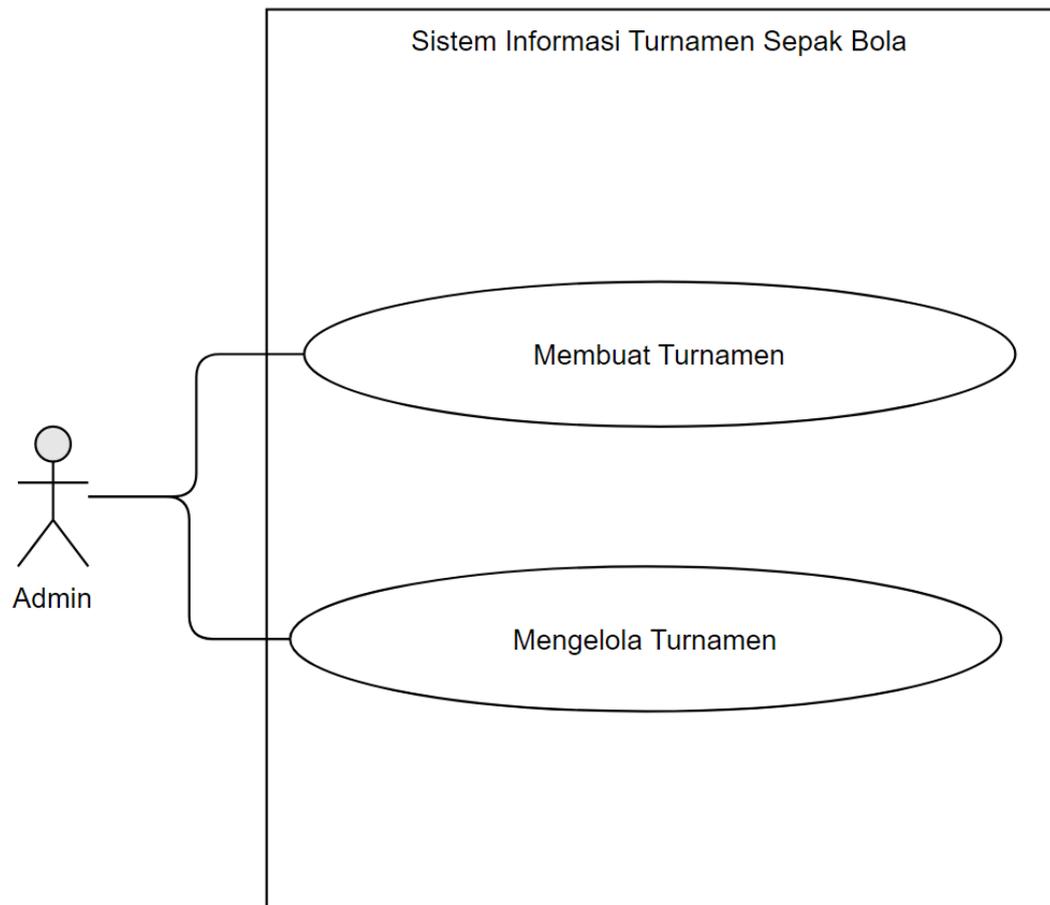
Analisis kebutuhan antarmuka digunakan untuk pembuatan antarmuka yang selaras sekaligus memudahkan penggunaan sistem oleh *end user*. Berikut merupakan kebutuhan antarmuka pada aplikasi ini:

- a. Halaman buat turnamen.
- b. Halaman edit turnamen

3.2 Perancangan

3.2.1 Perancangan *Use Case Diagram*

Setelah dilakukannya analisis kebutuhan, Langkah selanjutnya adalah memodelkan kebutuhan-kebutuhan fungsional tersebut menjadi diagram *use case*. Pada aplikasi ini hanya terdapat 1 aktor yaitu admin aplikasi generator turnamen sepak bola.



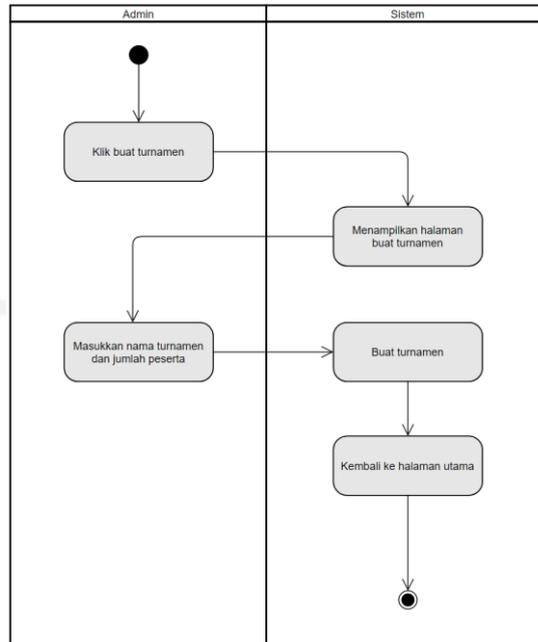
Gambar 3.2 Use Case Diagram Aplikasi Turnamen Sepakbola

Use case pada gambar 3,2 diatas memiliki sebuah *actor* yaitu admin. Admin tidak harus terlebih dahulu login untuk bisa mengelola jadwal dan membuat turnamen.

3.2.2 *Activity diagram*

Pada pemodelan UML, *Activity diagram* dapat digunakan untuk menjelaskan bisnis dan alur kerja operasional secara *step-by-step* dari komponen suatu sistem. *Activity diagram* menunjukkan keseluruhan dari aliran kontrol. Berikut adalah *Activity diagram* dari sistem yang dirancang.

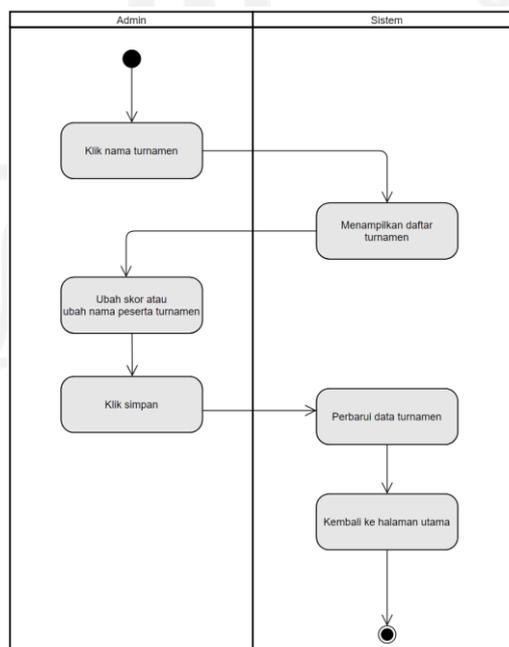
a. *Activity diagram* Membuat Turnamen



Gambar 3.3 *Activity diagram* Buat Turnamen

Jika menu buat turnamen diklik sistem lalu akan menampilkan halaman pembuatan turnamen lalu jika sudah selesai akan Kembali ke halaman utama seperti yang ditunjukkan pada gambar 3.3.

b. *Activity diagram* Mengelola Turnamen



Gambar 3.4 *Activity diagram* Mengelola turnamen

Jika nama sebuah turnamen diklik sistem lalu akan menampilkan halaman yang berisi detail dari turnamen tersebut, lalu admin dapat mengelola berupa mengubah hasil skor pertandingan atau mengubah nama tim pada turnamen tersebut seperti yang ditunjukkan pada gambar 3.4.

3.2.3 Perancangan Data

3.2.3.1 Perancangan Tabel Basis Data

Basis data dibuat dan dirancang untuk menyimpan data yang akan digunakan dan diolah dalam proses pembuatan turnamen. Berikut adalah desain dari tabel yang dibuat di dalam basis data:

a. Tournament

Tournament merupakan tabel yang menyimpan data turnamen yang sedang berjalan. Tabel 3.1 dibawah menunjukkan spesifikasi dari tabel Tournament seperti berikut.

Tabel 3.1 Tabel turnamen

Nama Field	Tipe Data	Ukuran	Keterangan
id	INT	5	Primary Key
tournamentName	VARCHAR	255	
type	INT	1	
slot	INT	2	
json	VARCHAR	255	

3.2.3.2 Perancangan JSON

JSON adalah format pertukaran data berbasis teks yang dapat dibaca manusia yang digunakan untuk bertukar data antara klien dan web server. Format ini mendefinisikan aturan-aturan penataan representasi data yang terstruktur. Pada aplikasi ini file JSON bertugas untuk menyimpan data sebuah turnamen yang nantinya akan di-*parse* dan diubah menjadi bagan pohon turnamen menggunakan *library* jQuery Bernama jQuery Bracket. Berikut merupakan rancangan file JSON untuk turnamen sepak bola.

Pada JSON yang digunakan pada aplikasi ini memiliki 2 *array* utama. *Array teams* merupakan *array* untuk menyimpan nama-nama peserta yang mengikuti turnamen, sedangkan *results* merupakan *array* yang menyimpan hasil skor pertandingan. Pada gambar 3.5, 3.6, dan 3.7 berikut merupakan file JSON untuk turnamen yang beranggotakan komplit (4,8,16).

a. JSON turnamen 4 tim

```

{
  "teams": [
    ["Team 1", "Team 2"],
    ["Team 3", "Team 4"]
  ],
  "results": [
    [
      [
        [null, null],
        [null, null]
      ],
      [
        [null, null],
        [null, null]
      ]
    ]
  ]
}

```

Gambar 3.5 JSON turnamen 4 tim

b. JSON turnamen 8 tim

```

{
  "teams": [
    ["Team 1", "Team 2"],
    ["Team 3", "Team 4"],
    ["Team 5", "Team 6"],
    ["Team 7", "Team 8"]
  ],
  "results": [
    [
      [
        [null, null],
        [null, null],
        [null, null],
        [null, null]
      ],
      [
        [null, null],
        [null, null]
      ]
    ]
  ]
}

```

```

    ],
    [
        [null, null],
        [null, null]
    ]
]
}

```

Gambar 3.6 JSON turnamen 8 tim

c. JSON turnamen 16 tim

```

{
  "teams": [
    ["Team 1", "Team 2"],
    ["Team 3", "Team 4"],
    ["Team 5", "Team 6"],
    ["Team 7", "Team 8"],
    ["Team 9", "Team 10"],
    ["Team 11", "Team 12"],
    ["Team 13", "Team 14"],
    ["Team 15", "Team 16"]
  ],
  "results": [
    [
      [
        [null, null],
        [null, null]
      ],
      [
        [null, null],
        [null, null],
        [null, null],
        [null, null]
      ]
    ]
  ]
}

```



```
[
  [
    [null, null],
    [null, null],
    [null, null],
    [null, null]
  ],
  [
    [null, null],
    [null, null]
  ],
  [
    [null, null],
    [null, null]
  ]
]
```

Gambar 3.8 JSON turnamen 15 tim

BAB IV

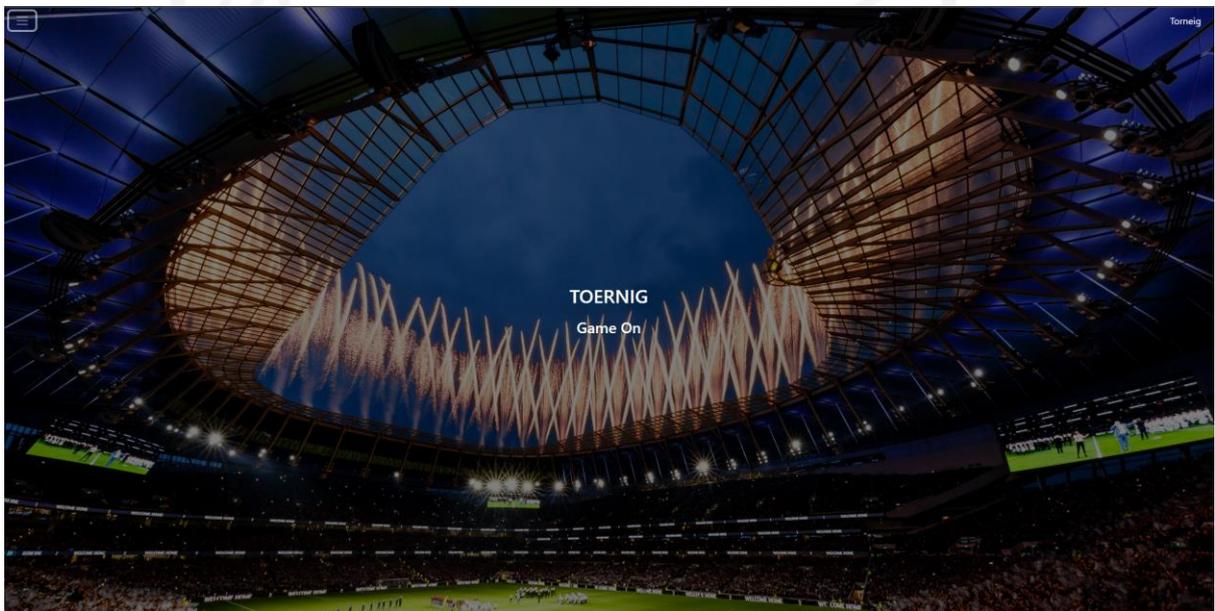
IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

4.1.1 Implementasi Antarmuka

Pada aplikasi ini terdapat beberapa antarmuka utama yaitu: antarmuka halaman utama, antarmuka buat turnamen, antarmuka daftar turnamen, dan antarmuka detail turnamen.

4.1.1.1 Antarmuka Halaman Utama



Gambar 4.1 Tampilan antarmuka halaman utama

Gambar 4.1 menunjukkan tampilan antarmuka halaman utama, lalu pada halaman utama ini terdapat menu untuk pembuatan turnamen dan daftar turnamen yang sedang berjalan seperti yang ditunjukkan pada gambar 4.2 dibawah.



Gambar 4.2 Tampilan menu pembuatan turnamen dan daftar turnamen

4.1.1.2 Antarmuka Buat Turnamen

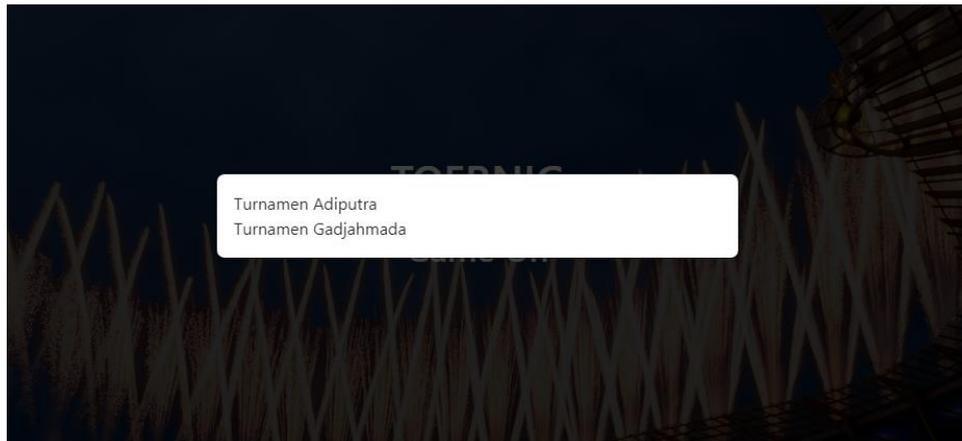
Pada antarmuka buat turnamen pengguna akan diminta untuk memasukkan beberapa detail turnamen seperti: nama turnamen, jumlah slot untuk turnamen tersebut, dan tipe turnamen tersebut (untuk penelitian ini hanya untuk eliminasi tunggal) seperti yang ditunjukkan pada gambar 4.3.

A screenshot of a form titled 'Buat Turnamen'. The form has three input fields: 'Nama Turnamen' with a text input field containing 'Nama Turnamen', 'Slot' with a dropdown menu showing '2-16', and 'Tipe' with a dropdown menu showing 'Tipe turnamen'. At the bottom left of the form is a blue button with the text 'Create'.

Gambar 4.3 Tampilan antarmuka buat turnamen

4.1.1.3 Antarmuka Daftar Turnamen

Pada antarmuka daftar turnamen akan ditampilkan turnamen- turnamen yang telah dibuat seperti yang ditunjukkan pada gambar 4.4 dibawah ini.



Gambar 4.4 Tampilan antarmuka daftar turnamen

4.1.1.4 Antarmuka Detil Turnamen

Pada antarmuka ini akan ditampilkan *bracket* atau bagan dari turnamen yang telah dibuat, pada antarmuka ini pengguna juga dapat mengubah skor pertandingan dan mengubah nama tim yang bertanding. Pada gambar 4.5 dibawah merupakan tampilan antarmuka detil turnamen.



Gambar 4.5 Tampilan antarmuka detail turnamen

4.1.2 Implementasi Kode Program

Aplikasi ini menggunakan *library* jQuery yang bernama jQuery Bracket. *Library* ini bekerja dengan cara mem-*parse* objek javascript berupa JSON yang kemudian akan diubah menjadi bagan turnamen seperti pada gambar 4.5.

Berikut merupakan beberapa penggalan kode-kode yang menjadi kebutuhan utama dari aplikasi ini.

a. Fungsi create tournament

```
public function Create_Tournament(){
    $this->load->model('login_model');
    $namaTurnamen = $this->input->post('namaTurnamen');
    $s = $this->input->post('slot');
    $t = $this->input->post('tipe');
    $json = file_get_contents('./assets/json/template/.'.$s.$t.'.json');
    file_put_contents('./assets/json/.'.$namaTurnamen.'.json', $json);
    $path = './assets/json/.'.$namaTurnamen.'.json';
    $data = array(
        'tournamentName' => $namaTurnamen,
        'slot' => $s,
        'type' => $t,
        'json' => $path
    );
    $this->login_model->insert($data);
    redirect(base_url());
}
```

Gambar 4.6 Kode program *create tournament*

Kode program pada gambar 4.6 tersebut bertujuan untuk memasukkan data-data turnamen yang telah dimasukkan oleh pengguna dan dilanjutkan dengan membuat file JSON yang akan diparse oleh JQuery bracket library nantinya. Kode pada controller ini menangkap inputan user pada halaman utama yang di-submit menggunakan form dan menggunakan metode *POST*.

b. Fungsi tampilkan *bracket* menggunakan library JQuery bracket

```

1  var saveData = <?php echo $json; ?>;
2  console.log(saveData);
3  function saveFna(data, userData) {
4  var json = JSON.stringify(data)
5  infoWhereWhen();
6  $.ajax({
7  type:"POST",
8  data:{id:json}
9  });
10 }
11 $(function () {
12 var container = $('#demo')
13 container.bracket({
14 disableToolbar:true,
15 matchMargin: 40,
16 init: saveData,
17 save: saveFna,
18 userData: "load.json"
19 })
20 infoWhereWhen();
21 var data = container.bracket('data')
22 })

```

Gambar 4.7 Kode program untuk menampilkan *bracket*

Kode program pada gambar 4.7 ini merupakan kebutuhan utama dari sistem. Library JQuery bracket akan mem-parse objek saveData (baris 1) yang sebelumnya sudah terlebih dahulu mendapatkan data JSON dari database. File dari JSON itu sendiri berisi detail-detail turnamen seperti: nama-nama tim, hasil skor dari pertandingan, dan slot dari turnamen itu sendiri. Setelah JSON di load dari database maka data tersebut akan di parse menggunakan fungsi utama untuk membuat bracket itu sendiri yaitu pada fungsi init pada baris 16. Sedangkan fungsi yang mengatur agar bracket dapat diubah-ubah sesuai kebutuhan adalah fungsi save pada baris 17. Fungsi save ini sendiri akan memanggil fungsi saveFna setiap kali terjadi perubahan yang kemudian akan mengirimkan data POST menggunakan JQuery ajax untuk diproses menggunakan fungsi penyimpanan data turnamen.

c. Fungsi tampilkan jadwal

Fungsi `infoWhereWhen()` disini berfungsi untuk menampilkan jadwal pertandingan dan pada gambar 4.8 berikut merupakan kode untuk fungsi tersebut

```
function infoWhereWhen() {
    const matches =
document.getElementsByClassName("teamContainer");
    const hari = "Hari ke ";

    for (let i = 0; i < matches.length; i++) {

        const div = document.createElement("div");
        div.style.position = "relative"
        div.style.width = "0"
        div.style.height = "0"

        const label = document.createElement("label");
        label.style.position = "absolute"
        label.style.width = "250px"
        label.style.fontSize = "85%"
        label.style.left = "10px"
        label.style.top = "-20px"
        label.style.padding = "0"
        label.style.color = 'rgba(255,255,255,0.6) '
        label.innerHTML = hari + (i + 1);

        div.appendChild(label)
        matches[i].appendChild(div)
    }
}
```

Gambar 4.8 Kode program penampilan jadwal

d. Fungsi penyimpanan data turnamen

Lalu untuk penyimpanan hasil sunting sendiri akan menggunakan fungsi JQuery ajax pada baris program `$.ajax({...})` pada gambar 4.7 yang selanjutnya akan diproses dengan kode program seperti pada gambar 4.9 berikut:

```

if(isset($_POST['id'])) {
    $myfile = fopen($path, "w") or die("Unable to open
file!");
    $txt = $_POST['id'];
    fwrite($myfile, $txt);
    fclose($myfile);
}

```

Gambar 4.9 Kode program penyimpanan detail turnamen

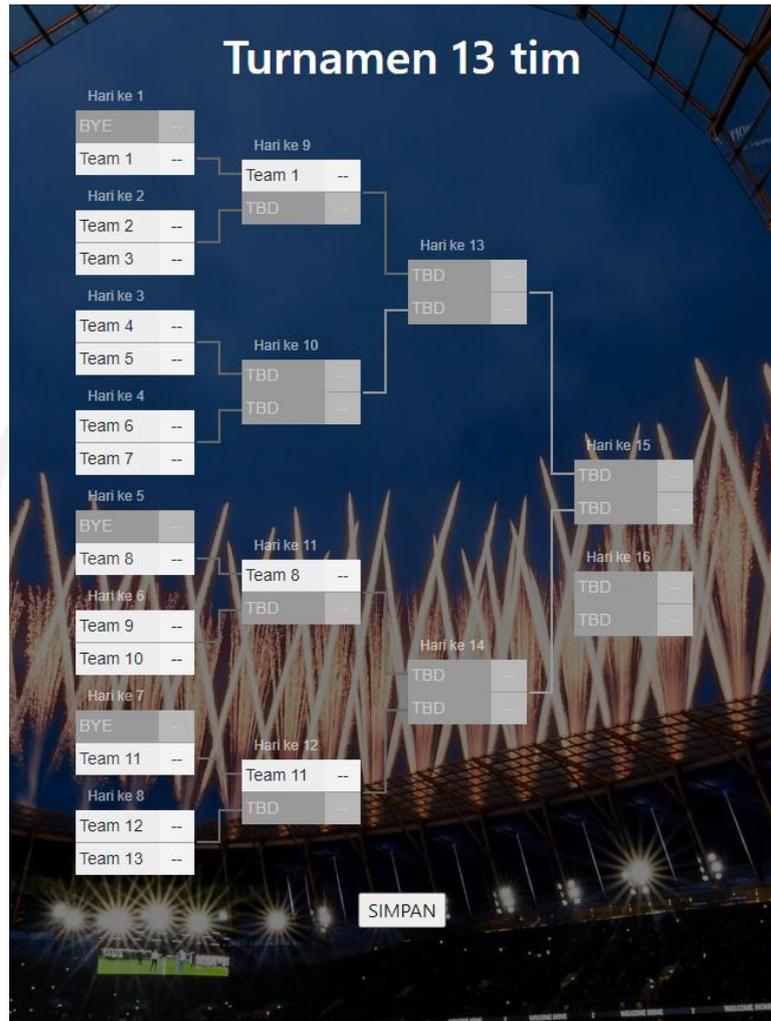
4.2 Skenario Pembuatan Turnamen

Untuk pembuatan *bracket* yang tidak umum contohnya *bracket* untuk turnamen yang terdiri dar 3 atau 7 tim, *bracket* akan dibuat mengacu pada *bracket* turnamen komplit yang tedekat. Turnamen komplit itu sendiri adalah turnamen yang memiliki tim berjumlah 2^n (2,4,8,16,dst). Jadi semisal untuk pembuatan *bracket* turnamen 3 tim, maka *bracket* akan dibuat mengikuti *bracket* untuk 4 tim, dimana 1 tim akan langsung maju ke babak selanjutnya. Berikut ini beberapa skenario hasil *bracket* yang telah berhasil dibuat menggunakan aplikasi ini.

4.2.1 Bracket turnamen non komplit

Pada gambar 4.11, 4.12, dan 4.13 berikut ini merupakan beberapa contoh *bracket* untuk turnamen non komplit

Gambar 4.10 *Bracket 3 tim*Gambar 4.11 *Bracket 7 tim*



Gambar 4.12 *Bracket* 13 tim

Bisa dilihat dari gambar-gambar di atas bila turnamen yang diselenggarakan memiliki tim yang tidak komplit atau penuh (bukan 2,4,8,16), maka akan ada beberapa tim yang otomatis maju ke babak selanjutnya tanpa harus mengikuti babak pertama. Biasanya tim yang langsung lolos ke babak selanjutnya adalah tim yang pernah menjadi juara turnamen sebelumnya. Namun sebenarnya tidak ada aturan yang mengatur tentang hal tersebut, jadi persoalan tim-tim mana saja yang berhak langsung maju ke babak selanjutnya itu semua kembali kepada keputusan dari penyelenggara turnamen.

4.2.2 Bracket turnamen komplet

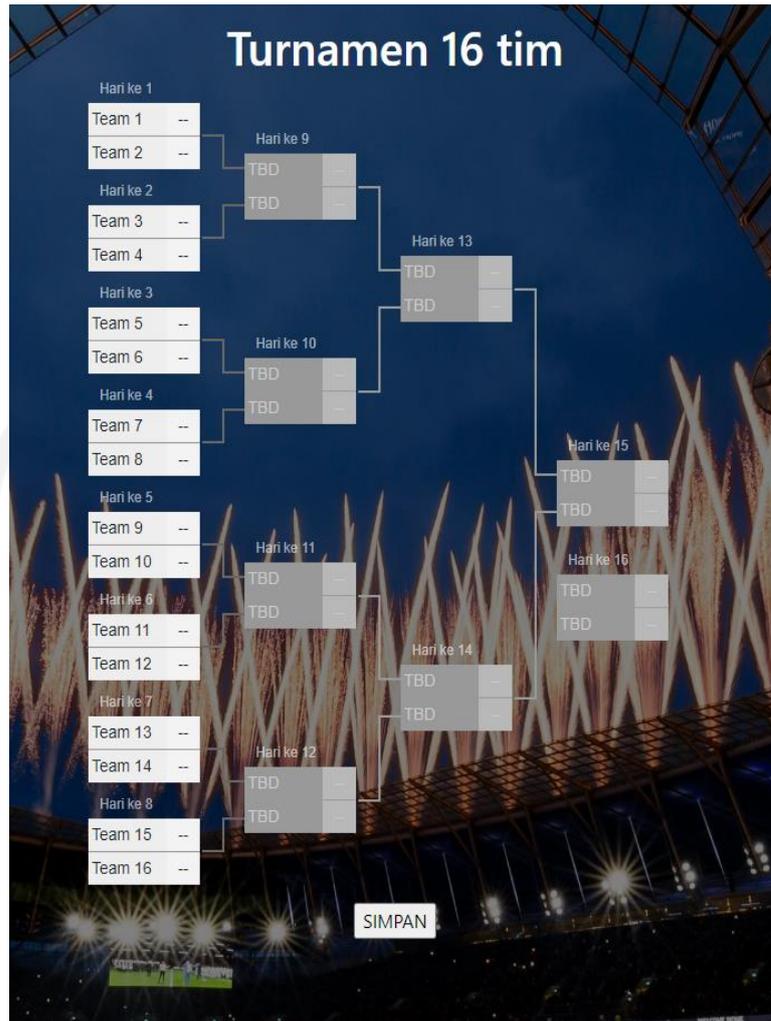
Pada gambar 4.13, 4.14, dan 4.15 berikut ini merupakan semua *bracket* untuk turnamen penuh atau komplet.



Gambar 4.13 *Bracket* 4 tim



Gambar 4.14 *Bracket* 8 tim

Gambar 4.15 *Bracket* 16 tim

4.3 Pengujian

Untuk pengujian disini penulis menggunakan metode pengujian *Blackbox Testing*. *Blackbox Testing* disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik (Setiawan, 2021).

Terdapat 2 pengujian utama pada aplikasi generator turnamen sepak bola ini, diantaranya: pengujian fungsionalitas pembuatan turnamen, dan pengujian fungsionalitas kelola turnamen. Pada tabel 4.1 dan 4.2 berikut merupakan hasil pengujian menggunakan metode *blackbox testing* pada kedua fungsionalitas di atas.

4.3.1 Pengujian pembuatan turnamen

Tabel 4.1 Tabel pengujian pembuatan turnamen

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Masukkan nama turnamen; jumlah slot dan jenis turnamen dipilih sesuai dengan pilihan yang ada di dropdown.	Turnamen akan dibuat di database.	Sesuai harapan.	Valid
2	Masukkan nama turnamen; jumlah slot dan jenis turnamen dipilih tidak sesuai dengan pilihan yang ada di dropdown.	Masukkan ditolak dan <i>user</i> akan diminta untuk memasukkan data lagi.	Turnamen masih terbuat di database.	Tidak Valid

4.3.2 Pengujian kelola turnamen

Tabel 4.2 Tabel pengujian lihat turnamen

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Masukkan hasil skor pertandingan.	Tim dengan skor yang lebih tinggi akan maju ke babak selanjutnya.	Sesuai harapan.	Valid
2	Ubah nama tim.	Nama tim akan berubah pada semua babak.	Sesuai harapan	Valid
3	Ubah hasil skor pertandingan.	Hasil babak selanjutnya atau sebelumnya akan berubah.	Sesuai harapan	Valid
4	Tekan tombol simpan.	Data akan tersimpan di database dan akan tampilan akan kembali ke halaman utama.	Sesuai harapan	Valid
5	Masukkan nama tim melebihi area yang telah disediakan.	Kotak teks akan menyesuaikan ukurannya.	Nama tim akan terpotong	Tidak Valid

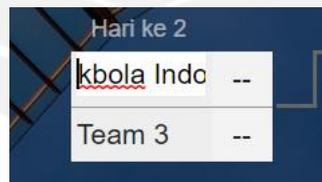
Sayangnya setelah dilakukan pengujian masih ditemukan beberapa bug yang terdapat pada sistem. Pada pengujian fungsionalitas pembuatan turnamen ditemukan sebuah bug dimana jika admin memasukkan masukan yang tidak sesuai dengan yang sudah disiapkan oleh sistem, turnamen akan tetap ter-generate pada database, padahal hasil yang diharapkan yaitu sistem akan menolak masukan dan admin akan diminta untuk memasukkan masukan yang sesuai. Imbasnya ketika turnamen dengan masukan yang tidak sesuai tersebut ter-generate di

sistem, turnamen tersebut akan menghasilkan entri yang tidak valid seperti yang ada pada gambar 4.16.



Gambar 4.16 Contoh entri turnamen yang tidak valid

Sementara itu pada pengujian fungsionalitas kelola turnamen ditemukan sebuah bug dimana jika admin nama tim yang dimasukkan melebihi dari kotak teks yang disediakan maka nama tim tersebut akan terpotong seperti pada gambar 4.17 berikut.



Gambar 4.17 Bug pada tampilan nama tim

Untuk fungsionalitas lainnya seperti: pembuatan turnamen, ubah hasil skor pertandingan, ubah nama tim, masukkan hasil skor pertandingan, dll seperti yang tercantum pada tabel 4.1 dan 4.2 hasil pengujian sudah valid dan keluaran sudah sesuai harapan. 2 bug yang ditemukan nantinya akan diperbaiki lagi pada aplikasi versi selanjutnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pengembangan aplikasi ini sudah berjalan dengan lancar dan kebutuhan utama untuk dapat membuat turnamen eliminasi tunggal untuk turnamen yang tidak penuh maupun penuh partisipannya sudah dipenuhi. Sistem mampu melakukan secara otomatis tim-tim mana saja yang akan langsung maju ke babak selanjutnya. Namun masih terdapat beberapa *bug* yang masih terdeteksi setelah dilakukan *blackbox testing* dengan seksama pada bab 4. Fitur-fitur yang diberikan juga masih dirasa belum lengkap, selain itu batasan penggunaan lapangan yang hanya satu lapangan dirasa masih menyulitkan pengguna dalam hal pengelolaan turnamen. Randomisasi atau pengacakan untuk tim-tim yang akan bertanding juga masih belum terpenuhi kebutuhannya. Memang untuk pengisian tim-tim yang akan bertanding sendiri masih harus secara manual dan bukan secara langsung memilih dari *database*, namun jika dilihat dari sudut pandang yang berbeda penginputan tim secara manual justru malahan membuat aplikasi turnamen sepak bola ini menjadi lebih fleksibel. Misalnya ketika terjadi perubahan nama tim atau terdapat tim yang dihapus, untuk mengubah entri pada *database* tidaklah mudah karena terdapat konstrain-konstrain lain yang harus dipikirkan dan tidak dapat seenaknya menghapus sebuah entri tersebut.

5.2 Saran

Dalam pengembangan aplikasi generator turnamen sepak bola ini dapat menggunakan beberapa saran sebagai berikut:

- a. Penambahan stadion sehingga dapat lebih dari satu.
- b. Randomisasi tim sehingga lebih *fair*.
- c. Dapat membuat jenis turnamen selain eliminasi tunggal seperti kompetisi penuh atau setengah kompetisi.
- d. Penambahan fitur melihat detail tim-tim yang akan bertanding.
- e. Penambahan fitur memasukkan secara otomatis tim-tim yang akan bertanding.

DAFTAR PUSTAKA

- Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Manifesto for Agile System Development*. Retrieved from Agile Manifesto: <https://agilemanifesto.org/>
- Byl, J. (2006). *Organizing Successful Tournaments - 3rd Edition*. Human Kinetics.
- Encyclopedia.com. (2022, Agustus 25). "*application generator*", *A Dictionary of Computing*". Diambil kembali dari <https://www.encyclopedia.com/computing/dictionaries-thesauruses-pictures-and-press-releases/application-generator>
- Fagarasan, C., Pop, O., Pislă, A., & Cristea, C. (2021). IOP Conference Series: Materials Science and Engineering. *Agile, waterfall and iterative approach in information technology projects*, 012025.
- IFAB. (2021, Juli). Laws of the Game 2021/2022. Zurich, Switzerland.
- KBBI. (2022). Diambil kembali dari <https://kbbi.kemdikbud.go.id/entri/turnamen>
- Setiawan, R. (2021, November 17). *Black Box Testing Untuk Menguji Perangkat Lunak*. Diambil kembali dari Dicoding: <https://www.dicoding.com/blog/black-box-testing/>
- Sziklaia, B. R., Biró, P., & Csató, L. (2021). *The efficacy of tournament designs*.
- Technopedia. (2020, September 18). *What is a Parser?* Diambil kembali dari Technopedia: <https://www.techopedia.com/definition/3854/parser>