

PERANCANGAN SISTEM INFORMASI PEMESANAN LAPANGAN FUTSAL BERBASIS WEB MENGGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT (RAD)*

Agung Riyanto Pratama¹

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pamulang, Tangerang Selatan, Banten
Email : Agungriyantoprataa@gmail.com

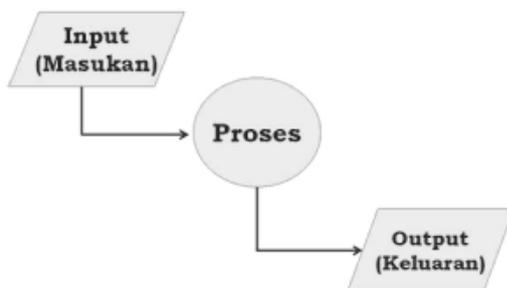
ABSTRAK

Perancangan sistem informasi pemesanan lapangan futsal ini merupakan aplikasi pemesanan berbasis web. Mempermudah penyewa lapangan mengetahui jadwal lapangan yang kosong, untuk memberikan kemudahan dalam melakukan proses penyewaan jadwal tanpa harus datang ke lokasi. Aplikasi ini diimplementasikan menggunakan bahasa pemrograman PHP dan database MySQL sebagai media penyimpanan data. Tujuan dari makalah ini adalah menghasilkan sistem informasi penyewaan lapangan melalui website. Sistem ini dapat dikembangkan lebih lanjut untuk kepentingan customer dalam melakukan pemesanan lapangan.

Kata kunci : Berbasis web, Penyewaan, PHP, MySQL, Sistem Informasi

PENDAHULUAN

Perkembangan teknologi informasi telah membawa dampak dalam kehidupan masyarakat. Sejak ditemukannya komputer pada tahun 1955, peradaban dunia telah memasuki era informasi. Pemrosesan informasi berbasis komputer mulai dikenal orang hingga saat ini sudah banyak *software* yang telah digunakan orang sebagai alat pengelolaan data untuk menghasilkan informasi.



Gambar 1 Komponen Sistem Informasi

Metode Pengembangan Aplikasi

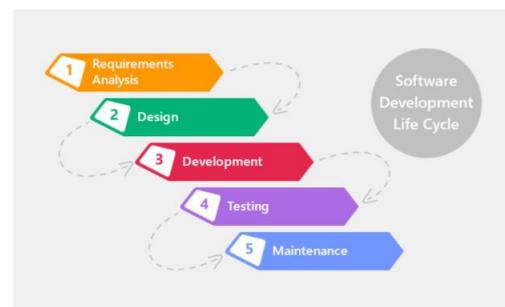
Metodologi dalam perangkat lunak digunakan untuk merancang atau membangun suatu perangkat lunak, dengan perkembangan teknologi yang sangat pesat, metodologi perangkat lunak juga terjadi perubahan atau penambahan requirements.

1. Waterfall

a. Pengertian Waterfall

Metode *waterfall* merupakan suatu metode dalam pengembangan software dimana pengerjaannya harus dilakukan secara berurutan yang dimulai dari tahap perencanaan konsep, pemodelan (*design*), implementasi, pengujian dan pemeliharaan (DOT Intern : 2019)

b. Tahapan Waterfall



Gambar 2 tahapan *waterfall*

1) Requirement Analysis

Metode pengumpulan informasi ini dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei, wawancara, dan sebagainya. Informasi yang diperoleh kemudian diolah dan dianalisa sehingga didapatkan data atau informasi yang lengkap mengenai spesifikasi kebutuhan pengguna akan perangkat lunak yang akan dikembangkan.

2) System and Software Design

Tahap ini juga akan membantu pengembang untuk menyiapkan kebutuhan hardware dalam pembuatan arsitektur sistem perangkat lunak yang akan dibuat secara keseluruhan.

3) Implementation and Unit Testing

Tahap implementation and unit testing merupakan tahap pemrograman. Pembuatan perangkat lunak dibagi menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya.

4) Integration and System Testing

Seluruh unit atau modul yang dikembangkan dan diuji di tahap implementasi selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.

5) **Operation dan Maintenance**

Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan.

2. **Rapid Application Development (RAD)**

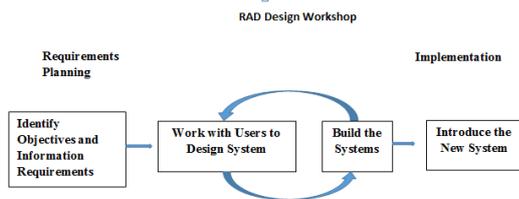
a. **Pengertian Rapid Application Development (RAD)**

Menurut S. Dan Shalahuddin (2018), menjelaskan bahwa “Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek”.

Pengembangan aplikasi yang cepat menggunakan pendekatan iteratif (berulang) untuk mengembangkan sistem. Metode ini bertujuan untuk membuat model kerja sistem di awal tahap pengembangan, mengidentifikasi kebutuhan pengguna, dan kemudian menghilangkannya. Model kerja dapat digunakan sebagai dasar untuk desain dan implementasi sistem akhir.

b. **Tahapan Rapid Application Development (RAD)**

Rapid Application Development (RAD). RAD merupakan gabungan dari teknik prototyping dengan *Join Application Development (JAD)* yang bertujuan untuk menggambarkan alur secara jelas tentang hal - hal yang diperlukan dalam sebuah aplikasi dan mempercepat pembangunan sistem. Tahap metode *Rapid Application Development (RAD)* diantaranya Requirement Planning (Persyaratan Perencanaan), Design Workshop (Pemodelan), Implementation (Penerapan).



Gambar 3 Tahapan RAD

1) **Perencanaan Persyaratan (Requirement Planning)** merupakan proses identifikasi dan analisa kebutuhan sistem yang akan dibangun

untuk mencapai tujuan, umumnya pada fase ini membutuhkan waktu satu sampai empat minggu.

2) **Pemodelan (Design Workshop)** dalam perancangan sistem ini dibuat sesuai fungsionalitas dan user friendly. Menyusun desain secara terperinci dengan menguraikan sesuai kebutuhan aplikasi yang tepat dengan memenuhi syarat yang telah ditentukan. Memahami pembuatan sistem sesuai dengan prosedur yang ada dan mengimplementasikan model dengan keinginan pengguna (user).

3) **Penerapan (Implementation)** dalam pembuatan aplikasi penyewaan lapangan futsal berbasis web dengan metode *Rapid Application Development (RAD)* tahap yang terakhir yaitu implementasi. Tahapan implementasi ini merupakan penerapan metode dalam pemrograman terhadap hasil dari desain yang telah dibuat dan dapat diuraikan secara jelas dalam tahap implementasi database dan coding. Tujuan dari tahap ini yaitu mengkonstruksi sistem dan penerapan metode dalam pemrograman terhadap hasil kebutuhan sistem.

3. **Agile Software Development**

a. **Pengertian Agile Software Development**

Muhammad Robith Adani menyatakan bahwa, “Agile Software Development adalah metodologi pengembangan software yang didasarkan pada proses pengerjaan yang dilakukan berulang di mana tujuh aturan dan solusi yang disepakati dilakukan dengan kolaborasi antar tiap tim secara terorganisir dan terstruktur” (Adani, 2020).

Model agile adalah kombinasi dari model proses iteratif dan inkremental. Langkah-langkah yang terlibat dalam model SDLC Agile adalah:

- 1) *requirement gathering*,
- 2) *requirement analysis*,
- 3) *design*,
- 4) *coding*,
- 5) *unit testing*.

b. **Tahapan Agile Software Development**



Gambar 4 Tahapan Agile

- 1) **Perencanaan** : merupakan langkah di mana tim pengembang dan juga klien merancang apa saja yang dibutuhkan dalam suatu perangkat lunak yang hendak dibuat.
- 2) **Implementasi** : merupakan tahapan di mana para tim pemrogram melakukan pengkodean pada suatu perangkat lunak.
- 3) **Tes Perangkat Lunak** : pada tahap ini, perangkat lunak yang telah diproduksi akan dites atau dicek, yang menjadi tanggung jawab bagian kontrol kualitas supaya *bug* yang masih ditemukan dapat langsung diperbaiki agar kualitas perangkat lunak tersebut tetap terjaga.
- 4) **Dokumentasi** : jika tahap tes perangkat lunak sudah selesai, kemudian dilanjutkan dengan proses dokumentasi yang mana tahap ini dimaksudkan untuk memberi kemudahan terhadap proses pemeliharaan atau *maintenance* ke depannya.
- 5) **Deployment** : merupakan tahap yang dilakukan untuk menjamin kualitas perangkat lunak yang diciptakan dengan menguji kualitas sistem. Jika sistem yang diproduksi telah memenuhi syarat, perangkat lunak tersebut nantinya sudah siap untuk dikembangkan.
- 6) **Pemeliharaan** : tahapan terakhir yang dilakukan dalam Metode Agile adalah pemeliharaan atau *maintenance*. Tahap ini ditujukan supaya tidak ada lagi bug yang mengganggu perangkat lunak. Maka dari itu, pemeliharaan ini merupakan tahap yang sangat penting dan harus dilakukan secara berkala agar kualitas selalu terjaga.

c. Kekurangan Agile Development

- 1) Agile tidak akan berjalan dengan baik jika komitmen tim kurang.
- 2) Tidak cocok dalam skala tim yang besar (>20 orang).
- 3) Perkiraan waktu release dan harga perangkat lunak sulit ditentukan.

4. Perbedaan antar Metodologi

Waterfall	Rapid Application Development	Agile Methodologies
Terstruktur	Berdasarkan pada perancangan prototipe lalu merekayasa ulang menjadi kualitas produksi	Fleksibel
Terbagi dalam fase yang berbeda	Pengembang memisahkan berdasarkan modul, domain dan integrasinya	Memisahkan siklus hidup menjadi sprint
Tidak memerlukan partisipasi pengguna	Mendemonstrasikan prototipe kepada produk owner	Kumpulan dari banyak proyek

Pengujian setelah pemrograman	Membutuhkan kepuasan pengguna Pengujian bersamaan dengan pemrograman
-------------------------------	--

Tabel 1. Perbedaan Metodologi

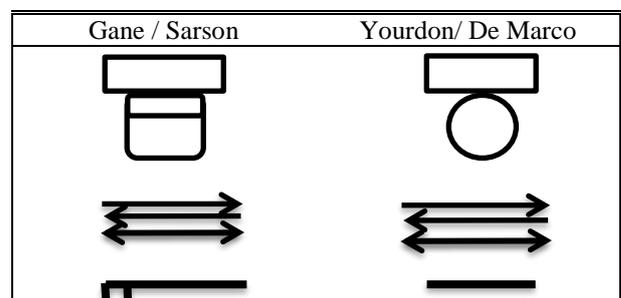
2.1 Pengertian Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah representasi grafis dari aliran data melalui sistem informasi. (Fithrie Soufitri : Ready Star – 2. ISSN: 2686-6641)

DFD dipopulerkan Oleh DeMacro & Yordan (1979) dan Gane & Sarson (1979) dengan menggunakan pendekatan Metoda Analisis Sistem Terstruktur. DFD ini merupakan model proses. Model proses merupakan teknik untuk mengorganisasikan dan mendokumentasikan struktur dan alir data di dalam sistem.

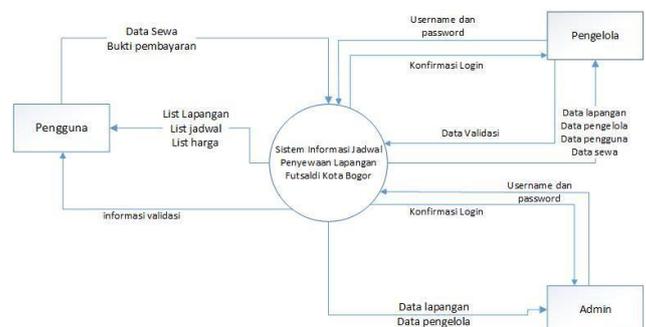
2.2 Komponen Data Flow Diagram (DFD)

Data Flow Diagram (DFD) terdiri dari empat komponen utama: input data, output data, penyimpanan data, dan proses yang memindahkan data di antaranya.



Tabel 2. komponen Data Flow Diagram (DFD)

2.3 Contoh Gambar Data Flow Diagram



Gambar 5. Contoh Data Flow Diagram (DFD)

Penjelasan dari Data Flow Diagram pada gambar 5, sebagai berikut :

DFD diatas menampilkan proses yang menghubungkan tiga aktor yaitu admin, pengelola dan pengguna. Admin dapat melihat, menambah, merubah dan menghapus data pengelola dan lapangan. Pengelola akan menerima data sewa dari pengguna untuk divalidasi oleh pengelola lapangan. Sedangkan pengguna dapat langsung mendaftarkan diri untuk langsung memesan lapangan yang tersedia, dan kemudian menunggu validasi dari pengelola lapangan.

3.1 Pengertian Diagram Konteks

Context diagram atau diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari data flow diagram (DFD) yang menggambarkan seluruh input ke dalam sistem atau output dari sistem yang memberi gambaran tentang keseluruhan sistem. (Jurnal REKAVASI, Vol. 7, No. 2, 2019 : 15 – 24).

3.2 Simbol Diagram Konteks

Simbol	Arti	Contoh
	Terminator	Mahasiswa
	Aliran Data/ Data flow	Informasi mahasiswa baru
	Proses/Process	Membuat record mahasiswa

Tabel 2.2 Simbol Context Diagram

3.3 Contoh Pembuatan Diagram Konteks



Gambar 6. Diagram Konteks Lapangan Futsal
Penjelasan Diagram Konteks gambar 6 sebagai berikut :

- a. Customer sebagai terminator melakukan aliran data seperti booking lapangan dan cek jadwal, lalu di proses oleh aplikasi dan diteruskan kembali ke customer berupa kwitansi maupun pengelola berupa laporan lapangan dan laporan keuangan

4.1 UML

1.1 Usecase Diagram

Use case diagram adalah satu dari berbagai jenis diagram *Unified Modelling Language* (UML) yang menggambarkan hubungan interaksi antara aktor dan sistem.

(socs.binus.ac.id/2019)

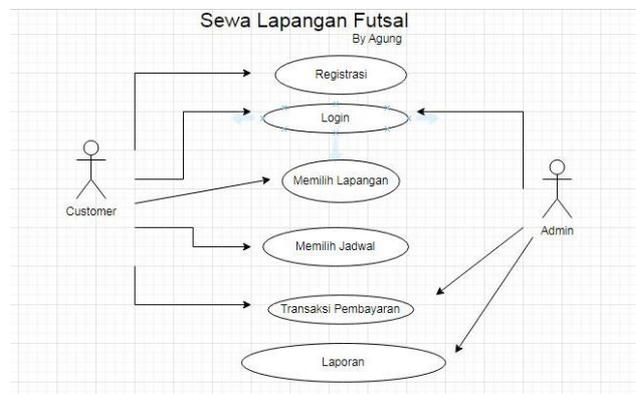
Usecase Diagram mengasosiasikan aktor dengan fungsi yang sesuai dengan kebutuhan (Merdekawati, 2019)

1. Simbol Usecase Diagram

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan use case
	Use case : Abstraksi dan interaksi antara sistem dan aktor
	Association : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

Tabel 4.1 simbol – simbol usecase diagram

2. Contoh Usecase diagram



Gambar 7. contoh usecase diagram

Penjelasan gambar 7. sebagai berikut :

1. Customer
 - a. Pertama customer mengisi data diri untuk melakukan pendaftaran seperti nama, alamat, username, password. Setelah melakukan pendaftaran customer baru bisa login ke dalam aplikasi
 - b. Jika login berhasil, customer dapat melakukan pencarian jadwal lapangan futsal. Jika jadwal yang dipesan kosong maka customer dapat melakukan pemesanan lapangan dan jadwal lapangan
2. Admin

- a. Melakukan login
Mendata transaksi pembayaran serta laporan, agar dapat di cek setiap bulannya

mendaftarkan akun terlebih dahulu.
Jika login berhasil, penyewa dapat melakukan pemesanan dan pembayaran

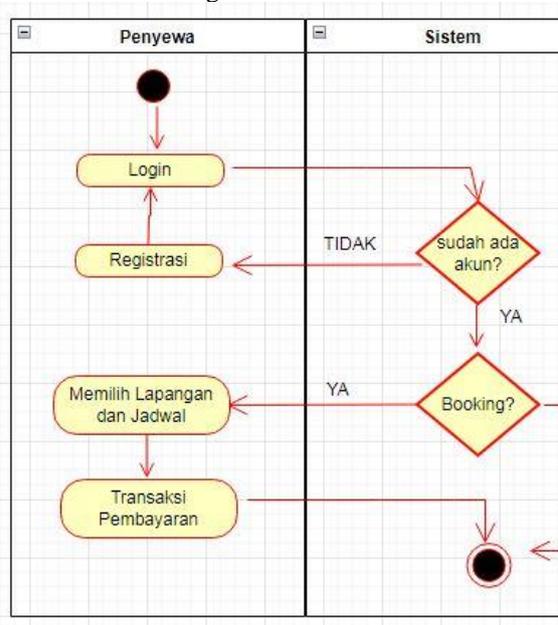
1.2 Activity Diagram

Menurut Rosa dan Shalahuddin (2018:161), Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

1. Komponen Activity Diagram

No.	Nama elemen	Fungsi	Notasi
1.	Action	Untuk menggambarkan perilaku yang sederhana dan bersifat <i>non-decomposable</i> .	
2.	Activity	Untuk mewakili kumpulan aksi (<i>action</i>)	
3.	Object Node	Untuk mewakili objek yang terhubung dengan kumpulan <i>object flow</i> .	
4.	Control Flow	Menunjukkan rangkaian pelaksanaan	
5.	Object Flow	Menunjukkan aliran sebuah objek dari sebuah aktivitas atau aksi ke aktivitas atau aksi lainnya.	
6.	Initial Node	Menandakan awal dari kumpulan aksi atau aktivitas.	
7.	Final-Activity Node	Untuk menghentikan seluruh <i>control flows</i> atau <i>object flows</i> pada sebuah aktivitas (atau aksi).	
8.	Final-Flow Node	Untuk menghentikan <i>control flow</i> atau <i>object flow</i> tertentu.	
9.	Decision Node	Untuk mewakili suatu kondisi pengujian yang bertujuan untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> hanya menuju ke satu arah.	
10.	Merge Node	Untuk menyatukan kembali <i>decision path</i> yang dibuat dengan menggunakan <i>decision node</i> .	

Gambar 8. komponen atau simbol pada activity diagram



Gambar 9. Contoh Diagram Activity Sewa Lapangan Futsal

Penjelasan Activity Diagram pada gambar 9, sebagai berikut :

- a. Penyewa mencoba melakukan login, jika sudah memiliki akun akan diteruskan ke tahap berikutnya oleh sistem, namun jika belum akan ditolak dan dianjurkan

1.1 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi di antara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. (Indonesian Journal On Information System. E-ISSN : 2548 – 6438. Vol.4, No.1 (2019))

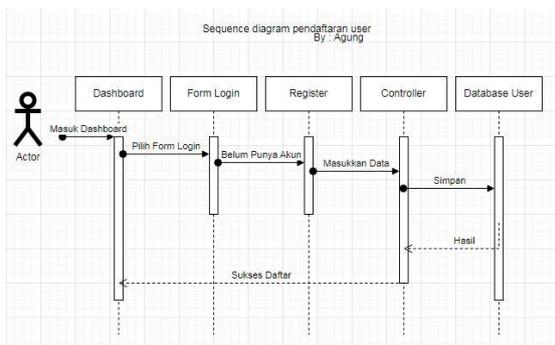
1. Simbol Sequence Diagram

No	Simbol	Keterangan
1		Actor adalah pesan dari seseorang atau sistem lain yang bertukar informasi dengan sistem yang lainnya.
2		Object Life Line menunjukkan kesatuan dari sebuah objek terhadap waktu. Yaitu objek dibuat atau dihilangkan selama suatu periode waktu diagram ditampilkan, kemudian <i>lifeline</i> berhenti atau mulai pada titik yang tepat.
3		Activation menampilkan periode waktu selama sebuah objek atau aktor melakukan aksi. Dalam <i>object lifeline</i> , <i>activation</i> berada diatas <i>lifeline</i> dalam bentuk kotak persegi panjang, bagian atas dari kotak merupakan inialisasi waktu dimulainya suatu kegiatan dan yang dibawah merupakan akhir dari waktu.
4		Message adalah komunikasi antar objek yang membawa informasi dan hasil pada sebuah aksi. Message menyampaikan dari <i>lifeline</i> sebuah objek kepada <i>lifeline</i> yang lain, kecuali pada kasus sebuah <i>message</i> dari objek kepada objek itu sendiri, atau dengan kata lain <i>message</i> dimulai dan berakhir pada <i>lifeline</i> yang sama.
5		Boundary digunakan untuk menggambarkan sebuah <i>form</i> .

6		Control digunakan untuk menghubungkan <i>boundary</i> dengan tabel.
7		Entity digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

Tabel 4.2 simbol *sequence diagram*

2. Contoh Sequence Diagram Lapangan Futsal



Gambar 10. *Sequence Diagram* Pendaftaran Penyuwa

Penjelasan *sequence diagram* pada gambar 10 sebagai berikut :

- Penyuwa masuk ke dashboard aplikasi
- Menuju form login, lalu lakukan registrasi jika belum mempunyai akun
- Sistem akan menyimpan data customer di dalam data base

1.3 Class Diagram

1. Pengertian Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem (Ameldi & Ahsyar, 2018)

2. Komponen Class Diagram



Gambar 11. komponen *Class Diagram*

- Komponen atas

Komponen yang membuat nama class diagram itu sendiri. Tiap class sudah pasti

akan memiliki nama yang berbeda. Istilah lain untuk penamaan ini adalah simple name atau nama sederhana.

- Komponen tengah

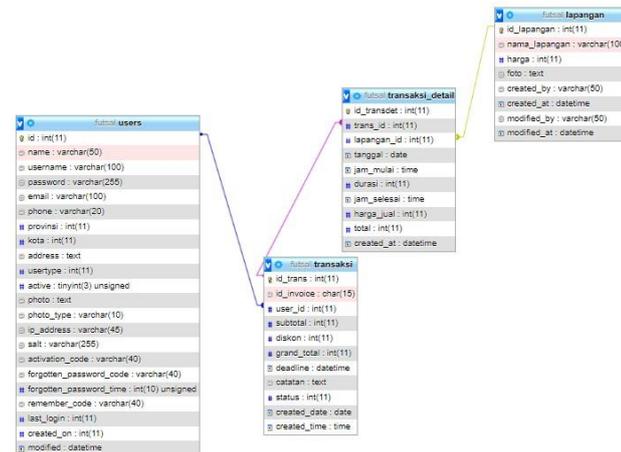
Komponen tengah, komponen ini memuat atribut class yang berguna untuk menjabarkan kualitas dari class. Atribut bisa menjelaskan lebih detail dengan cara menginput tipe nilainya.

- Komponen bawah

Bagian ini merupakan komponen *class diagram* yang menyertakan operasi class yang ditampilkan dalam format daftar. Sehingga setiap operasi mengambil barisnya sendiri.

Bukan hanya itu, komponen ini juga bekerja untuk menggambarkan bagaimana masing-masing class berinteraksi dengan data itu sendiri.

3. Contoh Class Diagram



Gambar 12. contoh *Class Diagram* aplikasi Penyewaan Futsal

Penjelasan dari *class diagram* pada gambar 12 Sebagai berikut :

- Kelas User berelasi dengan kelas transaksi yang menunjukkan kegiatan transaksi oleh user lalu akan dimunculkan detail transaksi tersebut pada kelas transaksi detail, di dalam transaksi detail juga dimunculkan pilihan lapangan yang akan digunakan sehingga berhubungan dengan kelas lapangan
- Entity Relationship Diagram

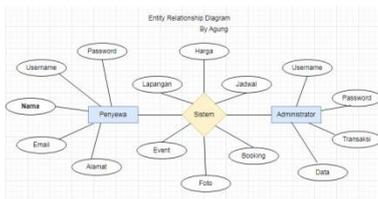
Entity Relationship Diagram (ERD) Data model didasarkan pada persepsi terhadap dunia nyata dan tersusun atas kumpulan objek objek dasar yang disebut entitas dan hubungan antara objek. Entitas adalah suatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain (Febriani Safitri, Kursehi Falgenti, Ambar Tri

Hapsari, 2019) Simbol Entity Relationship Diagram (ERD)

No	Simbol	Keterangan
1		Entity (Entitas)
2		Relasi (<i>relationship</i>)
3		Atribut
4		Line (Penghubung)

Tabel 5.1 Simbol *Entity relationship Diagram* (ERD)

3.3 Contoh Entity Relationship Diagram



Gambar 13. *Entity Relationship Diagram* Futsal Penjelasan dari *Entity Relationship Diagram* pada gambar 13, sebagai berikut :

- Entity Relationship Diagram* di atas terdapat 2 Entity, yaitu Penyewa dan Administrator
- Penyewa memiliki beberapa atribut seperti, username, password, nama, email dan alamat.
- Administrator memiliki beberapa atribut seperti username, password, transaksi dan data
- Penyewa dan Administrator memiliki hubungan dengan sistem yang mengatur beberapa atribut seperti lapangan, jadwal, biaya, event, foto dan booking.

KESIMPULAN

Olahraga telah menjadi hobi kalangan masyarakat Indonesia sejak lama, salah satunya adalah Sepakbola yang merupakan olahraga paling digemari karena prestasi dan *euforianya*.

keberadaan lapangan menjadi suatu kendala dalam melaksanakan olahraga tersebut dan masyarakat Indonesia pun memilih opsi lain seperti olahraga Futsal (*Football Sala*) yang tidak jauh berbeda dengan sepakbola.

Berbeda dengan sepakbola yang dilakukan di lapangan berukuran luas, futsal dapat dilakukan di ruang tertutup seperti Gelanggang Olahraga (GOR) maupun tempat penyewaan lapangan futsal.

Perancangan sistem informasi pemesanan ataupun booking diharapkan dapat memperbaiki dan membuat pekerjaan menjadi lebih cepat sehingga tidak terjadi hambatan dan kendala

dalam penanganan jadwal sewa lapangan yang disebabkan oleh adanya kelalaian sumber daya manusianya, baik pada saat proses transaksi penjadwalan dan booking lapangan tersebut.

Sistem informasi pemesanan berbasis web pada lapangan futsal ini dapat dijadikan sebagai sarana informasi pemesanan ataupun booking dengan memanfaatkan sistem komputerisasi dengan menggunakan bahasa pemrograman web PHP serta database MySQL.

DAFTAR PUSTAKA

- Destriana, Rachmat dkk. (2021). Diagram UML Dalam Membuat Aplikasi Android Firebase " Studi Kasus Aplikasi Bank Sampah". Yogyakarta:DEEPUBLISH
- Henderi, Untung Rahardja dan Efana Rahwanto.(2021) UML POWERED DESIGN SYSTEM USING VISUAL PARADIGM.Malang:CV. Literasi Nusantara Abadi
- Hikmah,Nur. Agus Suradika. R. Andi Ahmad Gunaedi. “ Metode Agile untuk Meningkatkan Kreativitas Guru Melalui Berbagai Pengetahuan (Knowledge Sharing)”. Jurnal Instruksional. Volume 3, Nomor 1
- Munawar. (2018). Analisa Perancangan Sistem Berorientasi Objek dengan UML. Bandung: Informatika Bandung
- Prabowo,Mei 'M.Kom'. (2020). METODOLOGI PENGEMBANGAN SISTEM. Salatiga: LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT (LP2M)IAIN SALATIGA
- Purwanto,Heri. Fiqri Arya Nugraha. Mochammad Raffnie Prayogha. Rafi Martua Syahputra. (2019) .“ Sistem Informasi Penyewaan Lapangan Futsal Berbasis Web”. Jurnal Informatika dan Rekayasa Perangkat Lunak. E-ISSN : 2685-5518. Vol.3, No 2
- Ridwan. Djunaidi Safi. Hairil K.Siradjun. (2019). “ Perancangan Aplikasi Sewa Lapangan Berbasis Web pada Aziz Futsal Kota Ternate ”. Indonesian Journal On Information System. E-ISSN : 2548 – 6438. Vol.4, No.1
- Rozaq, Abdul (2020). KONSEP PERANCANGAN SISTEM INFORMASI BISNIS DIGITAL. Banjarmasin:POLIBAN PRESS