

RANCANG SKEMA DATABASE DAN IMPLEMENTASI DATABASE MIGRATION PADA APLIKASI PEMINJAMAN RUANGAN

I.G.D. Dwijayana¹, I.G.A. Wibawa², dan G.A.V.M. Giri³

ABSTRAK

Peminjaman ruangan secara manual pada umumnya memiliki kendala, karena terkadang tidak adanya pencatatan terhadap ruangan yang sedang dipinjam atau digunakan sehingga bisa saja terjadi bentrok pelaksanaan suatu kegiatan yang berbeda di ruangan yang sama dalam waktu yang bersamaan. Sistem informasi yang telah maju dapat membantu dalam kegiatan peminjaman ruangan. Untuk mempermudah melakukan peminjaman ruangan dibuatkanlah Aplikasi Peminjaman Ruangan. Sistem *database* yang baik juga berpengaruh terhadap sistem Aplikasi yang akan dibuat. Tahapan yang digunakan untuk perancangan atau pembuatan skema *database* adalah analisis kebutuhan, perancangan konseptual, perancangan logika dan perancangan fisik. Pembuatan *database* menggunakan fitur *database migration*. Dengan menggunakan *database migration* dapat membuat *database* dengan skema yang sama pada perangkat lokal yang berbeda, hal ini juga memudahkan kerja *developer* aplikasi sehingga tidak membutuhkan waktu lama untuk membuat tabel pada *database* lokal.

Kata kunci : basis data, perancangan, aplikasi peminjaman ruangan, migrasi basis data, skema

ABSTRACT

Manually borrowing rooms in general has problems, because sometimes there is no record of the room being borrowed or used so that there may be clashes in the implementation of different activities in the same room at the same time. Advanced information systems can assist in room lending activities. To make it easier to borrow a room, a room loan application is made. A good database system also affects the application system that will be created. The stages used to design or create a database schema are requirements analysis, conceptual design, logical design and physical design. Database creation using the database migration feature. By using database migration, you can create databases with the same schema on different local devices, this also makes it easier for application developers so that it doesn't take long to create tables in the local database.

Keywords: *database, planning, room loan application, database migration, schema.*

1. PENDAHULUAN

Sering terjadi permasalahan dalam peminjaman ruangan dimana mahasiswa dan dosen kesulitan dalam penjadwalan peminjaman ruangan kampus. Peminjaman ruangan secara manual pada umumnya memiliki kendala, karena terkadang tidak adanya pencatatan terhadap ruangan yang

¹ Program Studi Informatika Fakultas MIPA Universitas Udayana, dwijayanaigede@gmail.com.

² Program Studi Informatika Fakultas MIPA Universitas Udayana, gede.arta@unud.ac.id.

³ Program Studi Informatika Fakultas MIPA Universitas Udayana, vida@unud.ac.id.

Submitted: 7 November 2022

Revised: 25 November 2022

Accepted: 27 November 2022

sedang dipinjam atau digunakan sehingga bisa saja terjadi bentrok pelaksanaan suatu kegiatan yang berbeda di ruangan yang sama dalam waktu yang bersamaan dikarenakan tidak adanya informasi tentang peminjaman atau penggunaan ruangan tersebut. Sistem informasi yang telah maju dapat membantu dalam kegiatan peminjaman ruangan. Untuk mempermudah melakukan peminjaman ruangan dibuatkanlah Aplikasi Peminjaman Ruangan.

Untuk membangun sistem Aplikasi Peminjaman Ruangan diperlukan *database* sebagai tempat penyimpanan data dari sistem tersebut. Sistem *database* yang baik juga berpengaruh terhadap sistem Aplikasi yang akan dibuat. *Database* yang baik haruslah memenuhi semua kebutuhan sistem. Dalam pembuatan *database* diperlukan perancangan terlebih dahulu lalu membangun *database* tersebut sesuai dengan rancangan yang telah dibuat. Tahapan yang digunakan untuk perancangan *database* adalah analisis kebutuhan, perancangan konseptual, perancangan logika dan perancangan fisikal (A. Kurnianti dkk, 2017). Pada pembuatan *database* menggunakan fitur *database migration*. *Database migration* digunakan untuk memudahkan pemindahan *database*. *Database migration* memudahkan pengguna untuk memperbarui teknologi atau fitur dari aplikasi, memindahkan *database* dengan skema yang sama dari suatu perangkat ke perangkat lainnya. *Database Migration* adalah salah satu fitur yang cukup memudahkan dalam pembuatan atau perubahan pada tabel-tabel di *database* aplikasi. Dengan menggunakan *database migration* dapat membuat *database* dengan skema yang sama pada perangkat lokal yang berbeda, hal ini juga memudahkan kerja developer aplikasi sehingga tidak membutuhkan waktu lama untuk membuat tabel pada *database* lokal mereka yang biasanya digunakan untuk *testing* aplikasi yang sedang dibuat.

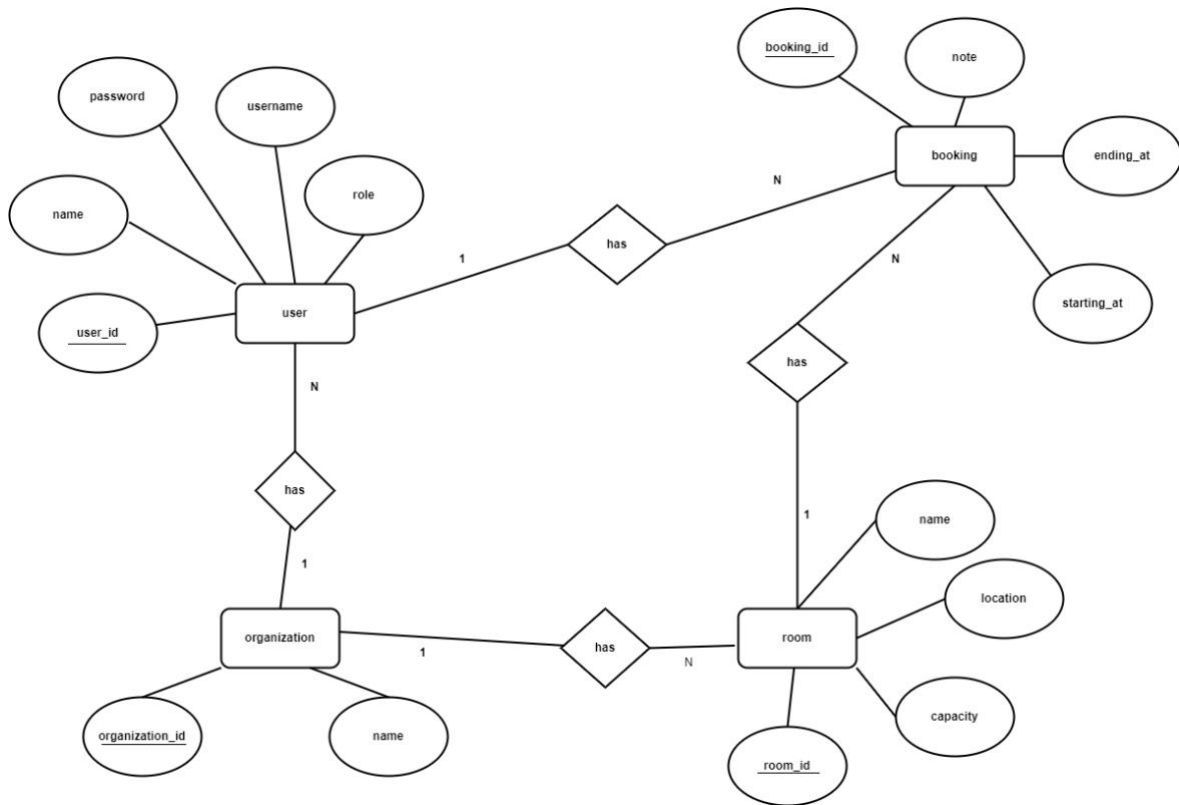
2. METODE PELAKSANAAN

Perancangan dan migrasi *database* dilakukan dengan metode *Waterfall SDLC* (*Software Development Life Cycle*) dengan analisis kebutuhan pengguna yang bertujuan untuk mengetahui kebutuhan sebelum merancang skema *database*, membuat rancangan atau skema dari *database* yang akan digunakan, implementasi rancangan *database* dengan membuat *database* tersebut menggunakan *database migration*. Dalam metode perancangan *database* ada 3 tahap yang harus dilakukan yaitu:

1. Perancangan *Database Level* Konseptual
Perancangan *database level* konseptual bertujuan untuk membuat skema dari batasan-batasan serta hubungan antar entitas pada *database* yang akan dibuat.
2. Perancangan *Database Level* Logika
Perancangan *level* logika bertujuan untuk memetakan rancangan konseptual yang telah dibuat ke dalam model *database*.
3. Perancangan *Database Level* Fisikal
Perancangan *database level* fisikal bertujuan untuk mengimplementasikan hasil dari rancangan level konseptual dan level logika, untuk mendapatkan rancangan *database* yang digunakan.

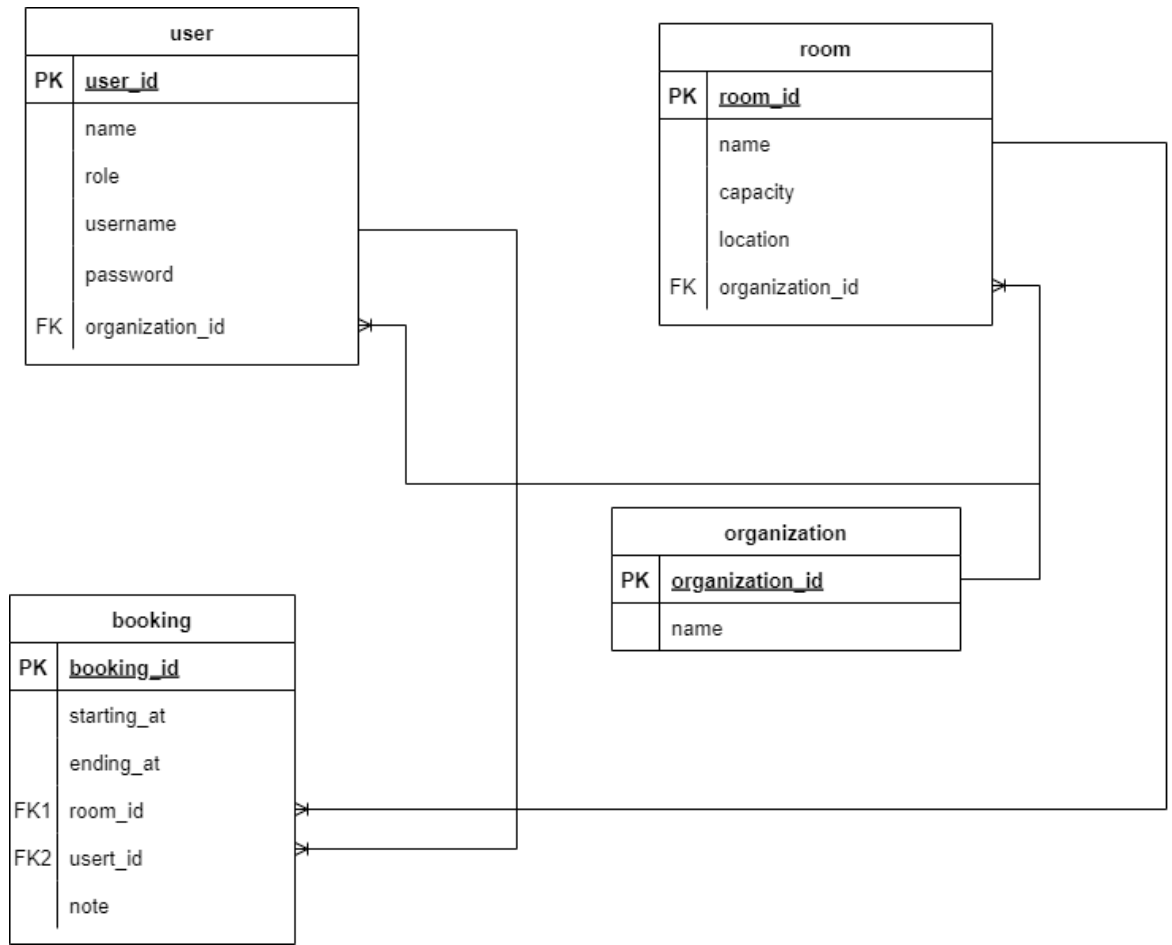
3. HASIL DAN PEMBAHASAN

Dari hasil analisis kebutuhan, didapat bahwa diperlukan empat tabel entitas yang terlibat dalam sistem *database* Aplikasi Peminjaman Ruangan. Empat tabel tersebut antara lain “*user*” sebagai tabel yang menyimpan data *user*, “*room*” sebagai tabel yang menyimpan data ruangan, “*organization*” sebagai tabel yang menyimpan data organisasi, “*booking*” sebagai tabel untuk menyimpan data peminjaman ruangan. Dari masing-masing tabel harus memiliki atribut yang sesuai untuk kebutuhan fungsional dari sistem *database* aplikasi peminjaman ruangan. Untuk mengetahui setiap atribut dan hubungan dari masing-masing tabel maka dibuatlah perancangan atau skema *database level* konseptual.



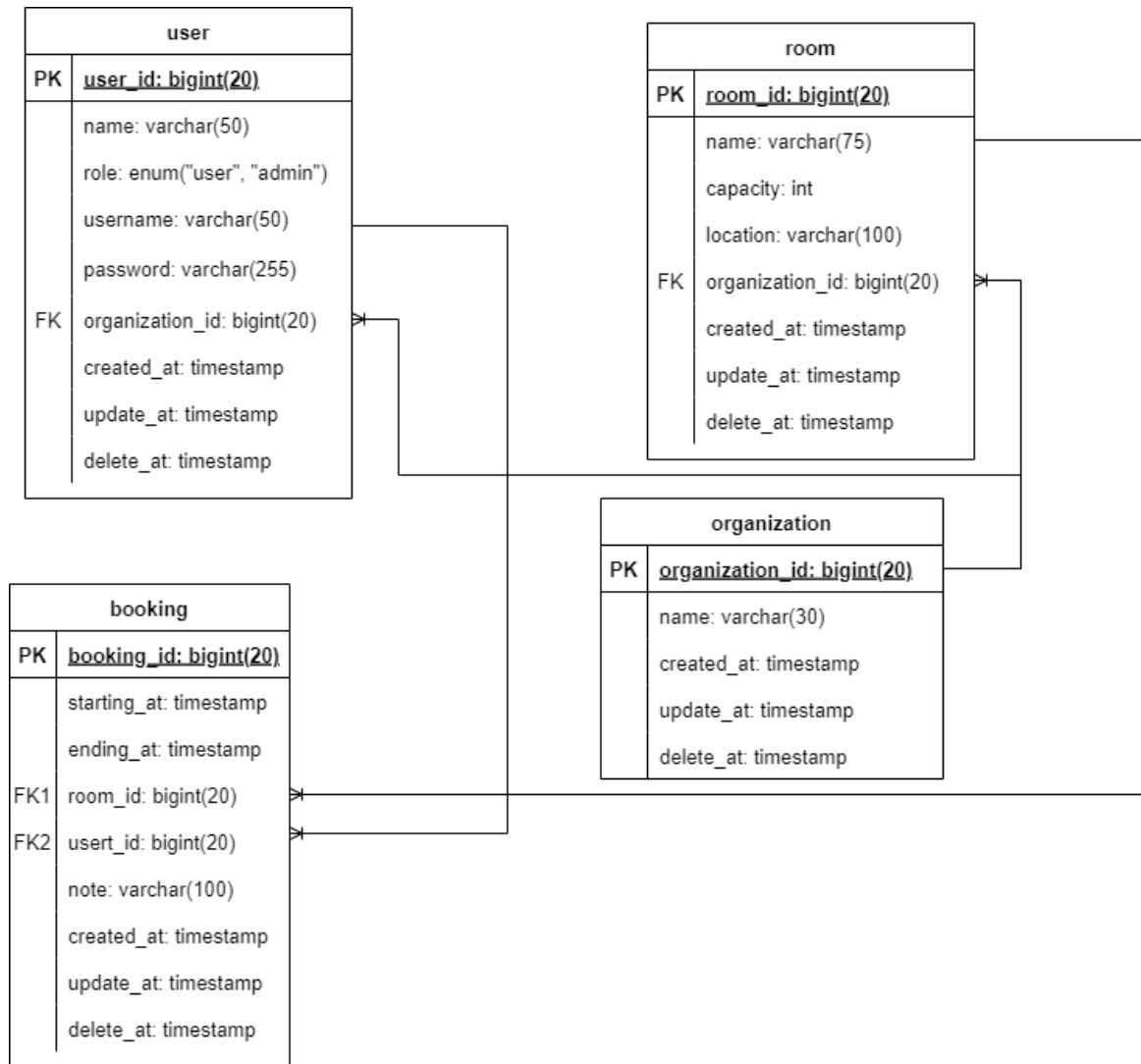
Gambar 3.1. ERD sebagai *level* konseptual skema *database*

Dapat dilihat pada Gambar 3.1 yaitu *Entity Relationship Diagram*, tabel *user* memiliki atribut yaitu “*user_id*” untuk penanda unik data *user*, “*name*” sebagai alternatif nama dari *user*, “*role*” yaitu peran *user* bisa sebagai *user* atau *admin*, “*username*” sebagai nama dari *user* yang *value*-nya akan digunakan untuk autentikasi, dan “*password*” untuk menyimpan *password user*. Selanjutnya atribut dari tabel *room* yaitu “*name*” untuk menyimpan nama ruangan, “*location*” untuk menyimpan data lokasi ruangan, “*capacity*” untuk menyimpan data kapasitas ruangan tersebut dan “*room_id*” sebagai penanda unik data ruangan. Atribut pada tabel *organization* hanya ada 2 yaitu “*organization_id*” dan “*name*”. Tabel yang terakhir adalah tabel *booking* yang memiliki atribut yaitu “*starting_at*” untuk menyimpan data waktu peminjaman dimulai, “*ending_at*” untuk menyimpan data waktu kapan peminjaman berakhir, “*note*” untuk menyimpan catatan, dan “*booking_id*” untuk penanda unik dari data *booking*. Selain atribut dari masing-masing tabel, terdapat juga hubungan antar tabel. Contoh relasi dari *organization* dengan *user* adalah 1toN yang artinya satu *organization* dapat memiliki beberapa atau banyak *user* dan sebaliknya banyak *user* dapat memiliki *organization* yang sama. Relasi antar tabel lainnya dapat dilihat pada Gambar 3.1. Langkah selanjutnya adalah membuat skema *database level* logika.



Gambar 3.2. LDM sebagai level logika skema database

Gambar 3.2 berupa *Logical Data Model* memperlihatkan bahwa ada penambahan atribut dari setiap tabel. Atribut tersebut berfungsi sebagai *foreign key* untuk menghubungkan tabel satu dengan tabel lainnya yang memiliki relasi. Ditentukan juga id dari setiap tabel merupakan *primary key*. Langkah selanjutnya adalah membuat skema *database level* fisik.



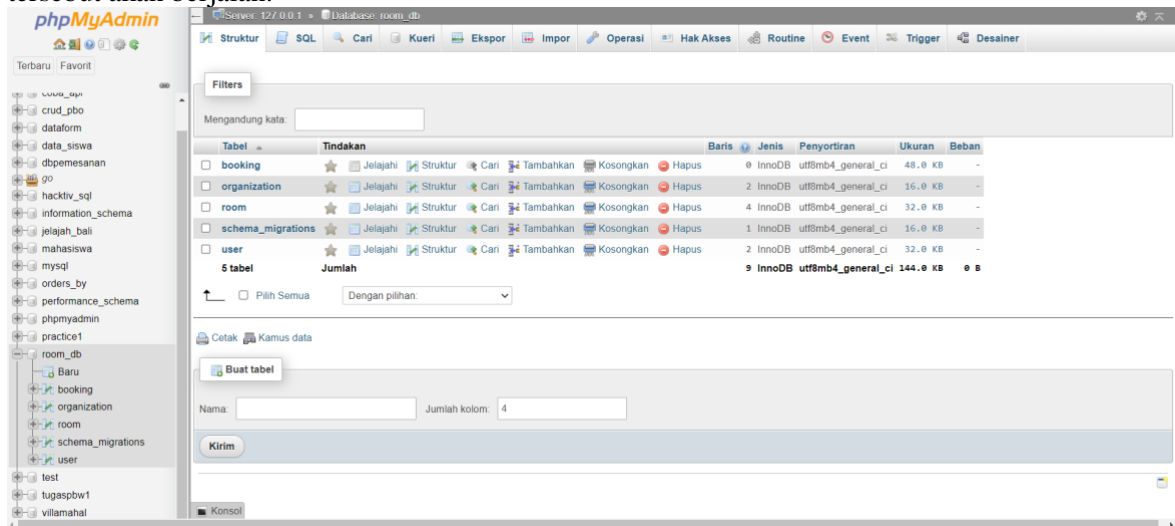
Gambar 3.3. PDM sebagai level fisik skema database

Gambar 3.3 adalah *Physical Data Model* yang memperlihatkan bahwa ada penambahan 3 atribut di setiap tabel. Tiga atribut ini adalah “*created_at*” untuk menyimpan data waktu pembuatan data, “*update_at*” untuk menyimpan data waktu data tersebut di-update, dan “*delete_at*” untuk menyimpan data waktu data tersebut di-delete. Pada *physical data model* juga telah dicantumkan tipe data dan ukuran maksimal dari setiap atribut yang ada pada setiap tabel. Atribut yang menyimpan nilai *id* akan diberi tipe data berupa *bigint*. Atribut yang menyimpan *string* akan diberi tipe data *varchar*, dan atribut yang menyimpan nilai waktu akan diberi tipe data *timestamp*. Khusus untuk atribut *role* pada *user*, diberi tipe data *enum* yaitu tipe data konstan yang hanya memiliki nilai *string* “*user*” atau “*admin*”.

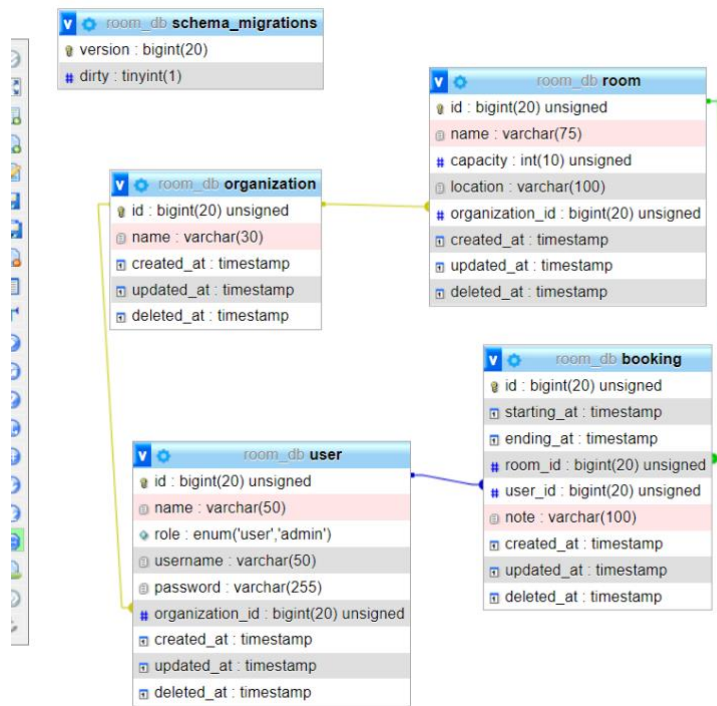
Setelah membuat perancangan *database* melalui 3 tahapan perancangan *database*, selanjutnya mengimplementasikan rancangan *database* tersebut dengan membangun *database* tersebut menggunakan sistem *database* MySQL dengan *database migration*. *Database migration* digunakan untuk memudahkan pemindahan *database*. *Database migration* memudahkan pengguna untuk memperbarui teknologi atau fitur dari aplikasi, baik untuk efisiensi atau keamanan data. *Database migration* adalah salah satu fitur yang cukup memudahkan dalam pembuatan atau perubahan pada tabel-tabel di *database* aplikasi. Contohnya penambahan kolom, *index*s dan lainnya, dengan

Rancang Skema Database Dan Implementasi Database Migration Pada Aplikasi Peminjaman Ruangan

database migration memungkinkan developer untuk membuat tabel-tabel tanpa harus membuka aplikasi administrasi database. Langkah selanjutnya adalah membuat perintah query dari skema database untuk membangun database. Setelah itu menjalankan perintah ``migrate -source file://path/to/schema -database mysql://localhost:3306/database up 2`` perintah ini adalah perintah migrate untuk bahasa pemrograman Go. Path adalah lokasi file query schema database berada, mysql sebagai sistem database yang digunakan dan localhost:3306 sebagai port dimana database tersebut akan berjalan.



Gambar 3.4. Database yang telah dibuat



Gambar 3.5. Skema database yang sudah dibangun

Hasil dari database yang telah dibuat dapat dilihat dengan database administrator. Dapat dilihat pada Gambar 3.5. bahwa database yang telah dibangun telah sesuai dengan rancangan database yang dibuat sebelumnya. Schema database yang telah dibuat dengan perintah query dapat digunakan oleh developer untuk membuat sistem aplikasi peminjaman ruangan di perangkat yang

berbeda dengan memanfaatkan *database migration*. Tahap Selanjutnya adalah melaksanakan pengujian dari *database* yang telah dibuat. Hasil pengujian *database* dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel Pengujian Database

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Keterangan
1	Melakukan <i>insert data</i> pada setiap tabel pada setiap tabel <i>database</i>	Setelah melakukan <i>insert data</i> tersebut telah disimpan di masing-masing tabel	Semua tabel yang diuji telah berhasil menyimpan data	Valid
2	Melakukan <i>update data</i> pada setiap tabel berdasarkan Id	Semua tabel yang telah di- <i>update</i> berdasarkan Id menyimpan hasil perubahan data	Semua tabel yang diuji telah berhasil menyimpan perubahan data.	Valid
3	Menampilkan semua data yang telah ada pada setiap tabel	Setiap tabel yang diuji menampilkan semua data pada masing-masing tabel tersebut, data yang ditampilkan haruslah sesuai dengan data hasil <i>insert</i> dan <i>update data</i> yang telah dilakukan sebelumnya	Semua tabel yang diuji telah menampilkan data yang sesuai dari hasil <i>insert data</i> dan <i>update data</i> yang telah dilakukan sebelumnya	Valid
4	Menguji hubungan antar tabel dengan membuat salah satu data <i>organization</i> dan membuat data <i>user</i> dengan <i>organization_id</i> yang telah dibuat sebelumnya pada tabel <i>organization</i> . Lalu <i>delete data organization</i> dimana data <i>organization</i> tersebut terhubung ke salah satu data <i>user</i> .	Terjadi <i>error</i> ketika mencoba <i>delete data</i> dari <i>organization</i> , karena data tersebut masih digunakan atau masih memiliki hubungan dengan data yang ada pada tabel <i>user</i> .	<i>Delete</i> tidak berhasil dilakukan, terjadi <i>error</i> pada saat perintah <i>delete</i> dieksekusi. Data pada tabel <i>organization</i> tidak terhapus.	Valid
5	Menguji hubungan antar tabel dengan membuat data <i>organization</i> , lalu membuat data <i>room</i> yang memiliki <i>organization_id</i> sesuai data <i>organization</i> yang dibuat sebelumnya. Melakukan <i>delete data</i> pada tabel <i>organization</i> dimana data tersebut terhubung ke salah satu data <i>room</i>	Terjadi <i>error</i> ketika mencoba <i>delete data</i> dari <i>organization</i> , karena data tersebut masih digunakan atau masih memiliki hubungan dengan data yang ada pada tabel <i>room</i> .	<i>Delete</i> tidak berhasil dilakukan, terjadi <i>error</i> pada saat perintah <i>delete</i> dieksekusi. Data pada tabel <i>organization</i> tidak terhapus.	Valid

6	Menguji hubungan antar tabel dengan menghapus salah satu data dari <i>room</i> dimana data tersebut masih terhubung pada salah satu data tabel <i>booking</i>	Terjadi <i>error</i> ketika mencoba <i>delete</i> data dari <i>room</i> , karena data tersebut masih digunakan atau masih memiliki hubungan dengan data yang ada pada tabel <i>booking</i> .	<i>Delete</i> tidak berhasil dilakukan, terjadi <i>error</i> pada saat perintah <i>delete</i> dieksekusi. Data pada tabel <i>room</i> tidak terhapus.	Valid
7	Menguji hubungan antar tabel dengan menghapus salah satu data dari <i>user</i> dimana data tersebut masih terhubung atau masih digunakan pada salah satu data tabel <i>booking</i>	Terjadi <i>error</i> ketika mencoba <i>delete</i> data dari <i>user</i> , karena data tersebut masih digunakan atau masih memiliki hubungan dengan data yang ada pada tabel <i>booking</i> .	<i>Delete</i> tidak berhasil dilakukan, terjadi <i>error</i> pada saat perintah <i>delete</i> dieksekusi. Data pada tabel <i>user</i> tidak terhapus.	Valid

Peran CV. Avatar Solution dalam penelitian ini membantu penulis dalam pelaksanaan tahap analisis kebutuhan sistem *database* pada Aplikasi Peminjaman Ruangan. Pembimbing lapangan dari CV. Avatar Solution memberikan arahan dengan melaksanakan sesi diskusi untuk mengetahui kebutuhan dari sistem yaitu membantu penulis untuk menentukan banyaknya entitas yang digunakan, atribut utama yang harus ada pada setiap entitas dan membantu merevisi rancangan konseptual dari *database*. Pihak CV. Avatar Solution juga membantu penulis pada tahap perancangan *database level* fisik dengan memberikan masukan berupa penambahan 3 atribut yang sebaiknya ada pada masing-masing tabel entitas.

4. KESIMPULAN

Perancangan dan pembuatan *database* untuk Aplikasi Peminjaman Ruangan telah berhasil dilaksanakan sesuai dengan kebutuhan dari aplikasi. Hasil migrasi *database* telah sesuai dengan rancangan *database* yang telah dibuat sebelumnya melalui 3 tahap. *Schema database* dalam bentuk *file* perintah *query* telah dapat digunakan oleh tim *developer* untuk membangun sistem dari Aplikasi Peminjaman Ruangan.

UCAPAN TERIMA KASIH

Ucapan terima kasih penulis sampaikan kepada Komisi PKL Program Studi Informatika Universitas Udayana beserta staf, Dekan Fakultas MIPA Universitas Udayana beserta jajarannya, serta, staf dan pembimbing lapangan CV. Avatar Solutions karena telah memberikan pengalaman baru dan juga kesempatan kepada penulis untuk dapat melaksanakan program Praktek Kerja Lapangan yang diselenggarakan oleh Komisi PKL Program Studi Informatika Universitas Udayana.

DAFTAR PUSTAKA

Andrianto, S., & Wijoyo, H. (2020). Rancang Bangun Sistem Informasi Siswa Berbasis Web di Sekolah Minggu Buddha Vihara Dharmaloka Pekanbaru. TIN : Terapan Informatika Nusantara, 1(2).

Azura, A., & Wildian, W. (2018). Rancang Bangun Sistem Absensi Mahasiswa Menggunakan Sensor RFID dengan Database MySQL XAMPP dan Interface Visual Basic. Jurnal Fisika Unand, 7(2). <https://doi.org/10.25077/jfu.7.2.186-193.2018>

- Elizabeth, T., & Tinaliah, T. (2018). Rancang Bangun Aplikasi Penjualan dan Stok Barang Material (Studi Kasus: Toko Bangunan XYZ). *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 4(2). <https://doi.org/10.35957/jatisi.v4i2.100>
- Husni, M., Jatmiko, N. P., & Prasetyo, A. (2005). Rancang Bangun Perangkat Lunak Manajemen Database Sql Server Berbasis Web. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 4(1). <https://doi.org/10.12962/j24068535.v4i1.a244>
- Jadid, A. (2017). Rancang Bangun Sistem Absensi Perkuliahan Auto Id Berbasis Rfid Yang Terintegrasi Dengan Database Berbasis Web. *Karya Ilmiah Teknik Elektro*, 2(2).
- Jessica, Christina. (2021). "Software Development Life Cycle (SDLC): Arti, Cara Kerja, Penerapan, dan Manfaatnya," [Online]. Tersedia: <https://glints.com/id/lowongan/sdlc-software-development-life-cycle/>. [Diakses pada 29 Oktober 2022]
- Kurnianti, A., Angguningtyas, & Isnanda, R.G. (2017). Perancangan Database Pada Sistem Aasesmen Dan Pemetaan Hasil Aasesmen Berbasis Tag Sebagai Pembantu Penyusunan Strategi Pembelajaran. *Jurnal Ilmiah Semesta Teknika*, 20(2), 106-115.
- Nur, H. (2019). Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan. *Generation Journal*, 3(1). <https://doi.org/10.29407/gj.v3i1.12642>
- Usada, E., Yuniarsyah, Y., & Rifani, N. (2012). Rancang Bangun Sistem Informasi Jadwal Perkuliahan Berbasis Jquery Mobile Dengan Menggunakan PHP Dan MySQL. *JURNAL INFOTEL - Informatika Telekomunikasi Elektronika*, 4(2). <https://doi.org/10.20895/infotel.v4i2.107>
- Wijaya, T.A.dkk. (2021). Perancangan Desain Basis Data Sistem Informasi Geografis Tanah Penduduk Dengan Menerapkan Model Data Relasional(Studi Kasus : Desa Tumbang Mantuhe Kabupaten Gunung Mas Provinsi Kalimantan Tengah). *Jurnal Teknologi Informasi: Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika*, 15(1), 72-81. <https://doi.org/10.47111/jti.v15i1.1867>

Halaman ini sengaja dikosongkan