

Analisis dan Perancangan Basis Data pada Aplikasi Tekos

1st Gilang Gumelar
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
[gilanggumelar@student.telkomuni-
versity.ac.id](mailto:gilanggumelar@student.telkomuni-
versity.ac.id)

2nd Mira Kania Sabariah
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
mirakania@telkomuniversity.ac.id

3rd Imanuddin Hasbi
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
imanhasbi@telkomuniversity.ac.id

Abstrak - Aplikasi Tekos adalah suatu aplikasi yang memudahkan pengguna dalam mencari informasi kos, kontrakan atau layanan di sekitar wilayah Telkom University. Aplikasi ini menyimpan banyak data, seperti data pengguna, data kos, data kontrakan dan data layanan. Oleh karena itu, diperlukan suatu sistem pemodelan basis data yang terstruktur dengan baik dan efisien dalam menyimpan dan mengelola data. Analisis dan perancangan basis data pada aplikasi Tekos dilakukan menggunakan metode Database Life Cycle (DBLC) yang memiliki tahap mulai dari perencanaan, analisis, desain, implementasi, dan pemeliharaan. Selain itu, dilakukan verifikasi basis data dan pengujian unit basis data dengan skenario yang mewakili penyimpanan, pembaruan, pengambilan dan penghapusan data. Hasil pengujian yang dilakukan menunjukkan tingkat keberhasilan dari pengujian unit sebesar 100%. Tingkat keberhasilan ini mengindikasikan tingkat kualitas dan keandalan unit-unit yang diuji dalam kode perangkat lunak sehingga dapat disimpulkan basis data yang ada sudah sesuai dengan pemodelan yang telah dibuat dan basis data yang dibuat sudah menerapkan pembatasan dan validasi yang telah ditentukan sesuai dengan kebutuhan yang tertera pada dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL).

Kata kunci : Tekos, basis data, DBLC.

I. PENDAHULUAN

A. Latar Belakang

Aplikasi Tekos adalah suatu aplikasi yang memudahkan pengguna dalam mencari informasi kos, kontrakan atau layanan di sekitar wilayah Telkom University. Aplikasi ini menyimpan banyak data, seperti data pengguna, data kos, data kontrakan dan data layanan. Untuk dapat memproses data dan informasi dengan efektif, dibutuhkan sebuah basis data yang terstruktur dengan baik [1]. Oleh karena itu, pada tugas akhir ini dilakukan analisis dan perancangan basis data pada aplikasi Tekos dengan menggunakan metode *Database Life Cycle* (DBLC) yang terdiri dari tahap perencanaan, analisis, desain, implementasi, dan pemeliharaan [2]–[4]. Pada tugas akhir ini dilakukan verifikasi basis data untuk memastikan bahwa *database* sesuai dengan

pemodelan analisis, oleh karena itu dilakukan verifikasi kesesuaian, validasi skema basis data dan konfirmasi hubungan antar tabel [5]. Selain itu dilakukan pengujian unit basis data guna mengetahui apakah basis data yang dibuat dapat berjalan sesuai kebutuhan yang ada [6], [7]. Pada perancangan basis data ini digunakan SKPL (Spesifikasi Kebutuhan Perangkat Lunak) pada bit.ly/SKPLTekos digunakan sebagai acuan dalam analisis dan perancangan basis data pada aplikasi. Dalam dokumen SKPL dijabarkan mengenai kebutuhan basis data, dimana basis data yang dibuat harus dapat menjawab kebutuhan data yang dimodelkan dalam bentuk model konseptual yang telah ditetapkan.

B. Topik dan Batasannya

Topik dari tugas akhir ini adalah analisis dan perancangan basis data pada aplikasi Tekos dengan menggunakan metode DBLC dan pengujian unit. Adapun batasan dari tugas akhir ini adalah analisis dan perancangan basis data hanya mencakup data pengguna, data kos, kontrakan, dan nearby pada aplikasi Tekos. Selain itu dilakukan verifikasi basis data dan pengujian unit basis data guna mengetahui apakah basis data yang dibuat dapat berjalan sesuai kebutuhan yang ada. Dengan adanya batasan tersebut, diharapkan tugas akhir ini dapat fokus pada analisis dan perancangan basis data serta pengujian antarmuka pengguna pada aplikasi Tekos dengan menggunakan metode yang tepat.

C. Tujuan

Berdasarkan rumusan masalah, maka didapatkan tujuan dari perancangan basis data aplikasi TEKOST yaitu menganalisis dan merancang sistem basis data pada aplikasi TeKos menggunakan metode DBLC sehingga menghasilkan suatu media penyimpanan terstruktur dan sesuai dengan kebutuhan pada SKPL. Selain itu, melakukan validasi dari model basis data yang sudah dirancang.

II. STUDI TERKAIT

A. Basis Data

Basis data merupakan suatu kumpulan data yang saling berhubungan dan tersimpan pada *hardware* komputer dan digunakan oleh *software* untuk memanipulasinya [11]. Data merupakan dasar penting suatu proses dalam pengambilan keputusan. Data juga merupakan suatu fakta mengenai benda, orang, *entity*, kejadian atau yang sejenisnya [4]. Basis data adalah kumpulan data yang dihubungkan satu sama lainnya sehingga data dapat ditampilkan, dimanipulasi dan dicari dengan cepat dan tepat. [12].

B. Database Life Cycle (DBLC)

Database Life Cycle (DBLC) adalah suatu metode yang memperlihatkan siklus hidup dari perancangan basis data. Proses utama untuk mendesain *database* terdiri dari tiga tahapan, yaitu perancangan basis data konseptual, perancangan logikal, dan terakhir perancangan fisik [15] :

1. *Database Planning* (Perencanaan Basis Data) yang merupakan tahap yang menentukan tujuan dari dibuatnya database dan apa yang akan dilakukan oleh database.
2. *System Definition* (Pendefinisian Sistem) merupakan tahapan di mana akan dilakukan identifikasi sudut pandang pengguna pada database sesuai dengan peran pengguna.
3. *Requirement Collection and Analysis* (Analisis dan Pengumpulan Kebutuhan) merupakan tahapan untuk menganalisis dan mengumpulkan kebutuhan dalam membangun basis data.

Selanjutnya merupakan tahap perancangan basis data yang terbagi menjadi 3 tahap yaitu:

- a. *Conceptual Database Design* adalah tahap untuk menentukan pengguna mana saja yang akan terlibat di dalam sistem, selain itu ditentukan juga apa saja input dan output yang mungkin diperlukan dalam basis data.
- b. *Logical Database Design* adalah tahap perancangan basis data yang terdiri dari penentuan data object dan field
- c. *Phisycal Design* merupakan tahap perubahan desain logical database ke dalam Phisycal Design ke dalam media penyimpanan menggunakan Database Management System (DBMS).

Setelah semua tahapan desain basis data dilakukan, selanjutnya merupakan tahapan implementasi yang terdiri dari implementasi basis data dan tabel. Tahap implementasi adalah tahap realisasi basis data ke dalam pemrograman menggunakan MongoDB.

C. Verifikasi Basis Data

Verifikasi basis data adalah proses untuk memastikan bahwa data yang disimpan dalam basis data sesuai dengan standar dan persyaratan yang ditetapkan. Tujuan utama dari verifikasi basis data adalah untuk memverifikasi apakah data yang ada

dalam database benar, konsisten, lengkap, dan sesuai dengan harapan. Ini adalah langkah penting dalam memastikan kualitas data dan keandalan data dari sistem yang bergantung pada basis data. Selain itu, proses verifikasi dapat mengurangi risiko pembangunan atas kualitas produk yang rendah [5].

D. Unit Testing.

Unit Testing merupakan Pengujian otomatis yang digunakan untuk memvalidasi komponen terkecil (*unit*) dari kode, berlangsung secara cepat dan dalam lingkungan yang terpisah. Sebuah *unit test* memiliki struktur yang membentuknya[7]. Pengujian unit basis data adalah jenis pengujian yang fokus pada pengujian komponen-komponen yang berkaitan dengan interaksi dan pengoperasian basis data (*database*). Tujuannya adalah untuk memastikan bahwa operasi yang berkaitan dengan penyimpanan, pengambilan, pembaruan, dan penghapusan data dari database berjalan dengan baik sesuai dengan yang kebutuhan yang ada.

III. SISTEM YANG DIBANGUN

A. Perencanaan Basis Data

Melalui wawancara yang melibatkan pemilik kos dan pencari kos, menunjukkan bahwa basis data yang telah dirancang harus mampu menampung informasi yang mencakup data pengguna, kos, kontrakan dan nearby.

B. Pendefinisian Sistem

Desain basis data pada aplikasi Tekos Dibuat untuk menyimpan informasi terkait aktivitas layanan yang diberikan dan kebutuhan informasi bagi pengguna aplikasi. Terdapat beberapa pihak yang berperan dalam sistem pada aplikasi tekos. Peran-peran ini dapat diidentifikasi dalam Tabel 3.1.

TABEL 3. 1
Pendefinisian Pengguna Berdasarkan Kebutuhan

No	Aktor	Aktivitas
1.	Pencari	Melakukan pencarian informasi
2.	Pemilik	Melakukan pengelolaan kos atau kontrakan
3.	Admin	Melakukan verifikasi data kos atau kontrakan, melakukan pengelolaan nearby

C. Analisa dan Pengumpulan Kebutuhan

Hasil dari analisis kebutuhan yang diperoleh melalui observasi dan wawancara langsung dengan pihak terkait adalah sebagai berikut:

1. Analisis Kebutuhan Informasi Pencari kos memiliki kesulitan dalam mencari informasi seperti:
 - a. Mendapatkan informasi secara detail mengenai kos mulai dari lokasi, harga sewa, kondisi dan fasilitas kos.
 - b. Mendapatkan informasi mengenai pemilik kos seperti nama, alamat, kontak, dan email.

- c. Mendapatkan informasi mengenai kondisi daerah sekitar seperti tempat makan, layanan dan lainnya.
- 2. Analisis Kebutuhan Teknologi
 Pada aplikasi tekos akan digunakan teknologi NoSQL karena teknologi ini dapat menjadi pilihan yang tepat bagi aplikasi yang membutuhkan performa tinggi dan fleksibilitas dan mengelola data seperti aplikasi Tekos.

D. Desain Basis Data Konseptual

Pada fase ini, tipe entitas diidentifikasi berdasarkan daftar aktivitas pengguna yang merujuk pada kata benda yang ada di dalamnya. Hasil identifikasi tipe entitas dapat ditemukan dalam Tabel 3.2.

TABEL 3. 2
Identifikasi Tipe Entitas

No	Entitas	Deskripsi	Kegiatan
1	Pemilik	Entitas yang berisikan data pemilik	Semua pemilik yang telah mendaftar
2	Pencari	Entitas yang berisikan data pencari	Semua pengguna sebagai pencari yang sudah terdaftar
3	Admin	Entitas yang berisikan data admin tekos	Semua admin sebagai pengelola aplikasi
4	Kos	Entitas yang berisikan informasi terkait kos	Semua kos yang didaftarkan oleh pemilik kos
5	Kontrakan	Entitas yang berisikan informasi terkait kontrakan	Semua kontrakan yang didaftarkan oleh pemilik kontrakan
6	Nearby	Entitas yang berisikan informasi terkait nearby (layanan)	Semua layanan yang didaftarkan oleh pemilik layanan
7	Token	Entitas yang berisikan token reset password dan verifikasi akun	Semua token yang auto generate Ketika adanya permintaan reset password atau verifikasi akun

Langkah berikutnya adalah mengidentifikasi tipe-tipe relasi dengan tujuan untuk menetapkan hubungan penting antara entitas yang telah diidentifikasi sebelumnya dari proses identifikasi entitas. Berikut ini adalah daftar entitas beserta penjelasan mengenai relasinya, yang dijabarkan dalam Tabel 3.3.

TABEL 3. 3
Identifikasi Tipe Relasional

Nama Entitas	Tipe Relasi	Nama Entitas	Deskripsi
Pemilik	1..N	Kos	Pemilik dapat memiliki lebih dari satu kos.
	1..N	Kontrakan	Pemilik dapat memiliki lebih dari satu kontrakan.
	1..N	Link	Pemilik dapat memiliki lebih dari satu link yang digunakan untuk melakukan pemulihan akun.

Pencari	1..N	Kos	Pencari dapat melihat lebih dari satu kos.
	1..N	Kontrakan	Pencari dapat melihat lebih dari satu kontrakan.
	1..N	Nearby	Pencari dapat melihat lebih dari satu nearby.
	1..N	Link	Pencari dapat memiliki lebih dari satu link yang digunakan untuk melakukan pemulihan akun.
Admin	1..N	Kos	Admin dapat melihat lebih dari satu kos.
	1..N	Kontrakan	Admin dapat melihat lebih dari satu kontrakan.
	1..N	Nearby	Admin dapat menambahkan lebih dari satu nearby.
	1..N	Link	Admin dapat memiliki lebih dari satu link yang digunakan untuk melakukan pemulihan akun.

E. Desain Logikal Basis Data

Pada tahap ini, perancangan basis data logikal dimulai dengan mengidentifikasi atribut dan kunci kandidat yang akan digunakan untuk setiap entitas. Hasil dari proses identifikasi atribut dan kandidat kunci kandidat dapat di lihat dalam Tabel 3.4.

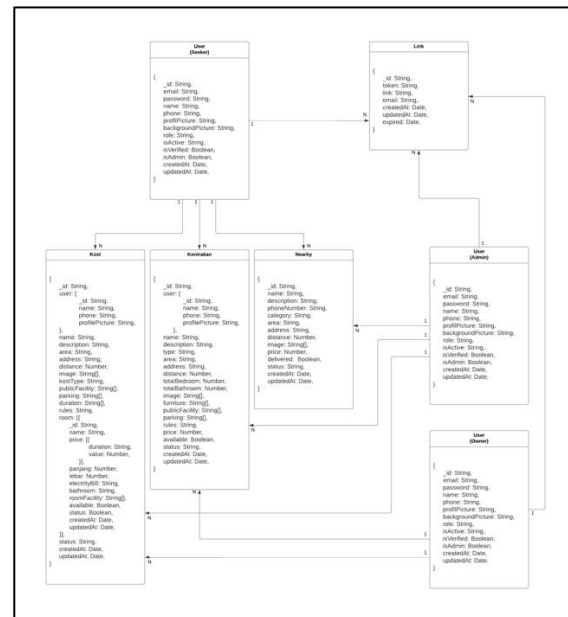
TABEL 3. 4
Identifikasi Atribut dan Kandidat Key

No	Nama Entitas	Atribut
1	Pemilik	_Id (primary) email password name Phone profilePicture backgroundPicture role isActive isVerified isAdmin createDate updateAt
2	Pencari	_Id (primary) email password name Phone profilePicture backgroundPicture role isActive isVerified isAdmin createDate updateAt
3	Admin	_Id (primary) email password name Phone profilePicture backgroundPicture role isActive isVerified

		isAdmin createDate updateAt
4	Kos	_Id (primary) User { _id Name Phone profilePicture } Name description area address Distance image kostType publicFacility parking duration Rules Room [{ _id name price: ({ duration value }) }] panjang lebar electricityBill bathroom roomFacility available status createdAt updatedAt } status createDate updateAt
5	Kontrakan	_Id (primary) User { _id Name Phone profilePictures } Name description type area address distance totalBedroom totalBathroom image furniture publicFacility parking duration rules Price Available status createDate updateAt
6	Nearby	_Id (primary) Name Description phoneNumber

		category Area address Distance image Price status createdAt updateAt
7	Token	Id (primary) token link email expired createdAt updateAt

Selanjutnya dilakukan pembuatan diagram relasi entitas tabel. Pada diagram terdapat tiga jenis user yaitu Admin, Owner dan Seeker. Setiap user dapat memiliki token yang akan dipakai untuk melakukan reset password dan verifikasi akun. Untuk user seeker hanya bisa melakukan pencarian dan melihat detail dari kos, kontrakan dan nearby. Sedangkan untuk user owner dapat melakukan penambahan data kos, kontakn, atau nearby. Dan untuk user admin melakukan verifikasi kos, kontrakan dan melakukan manipulasi data nearby yang dapat dilihat pada Gambar 3.1.



GAMBAR 3. 1
Diagram Hubungan Entitas Konseptual

Selanjutnya dilakukan pemetaan persyaratan dari SKPL ke dalam komponen konkret dalam sistem. Pada tahap ini dijabarkan setiap persyaratan dan bagaimana persyaratan tersebut diimplementasikan dalam pemodelan basis data. Pemetaan tersebut dapat dilihat pada bit.ly/PemetaanSKPL.

Desain Fisik Basis Data

Pada langkah ini, rancangan basis data logis diimplementasikan menjadi bentuk fisik. Hasil dari

transformasi entitas dan atribut menjadi tabel dan kolom dapat ditemukan dalam Tabel 3.5 sampai 3.9.

TABEL 3.5
Kamus Data Tabel User

Name	Data type	Constrain
_Id (primary)	String	Required, Min: 24, Max: 24
email	String	Require: "@", "."
password	String	Require: Char, Num, Symbol
name	String	Required, Min:2, Max:100
Phone	String	Min: 11, Max: 16
profilePicture	String	Max: 25
backgroundPicture	String	Max: 25
role	String	Min: 2, Max: 16
isActive	Boolean	
isVerified	Boolean	
isAdmin	Boolean	
createDate	Date	
updateAt	Date	

TABEL 3.6
Kamus Data Tabel Kots

Name	Data type	Constrain
_Id (primary)	String	Required, Min: 24, Max: 24
User { _id Name Phone profilePicture }	String String String String	Required, Min: 24, Max: 24 Min: 11, Max: 100 Min: 11, Max: 16 Max: 25
Name	String	Required, Min: 2, Max: 100
description	String	Min: 10, Max: 255
area	String	Min: 10, Max: 255
address	String	Min: 10, Max: 255
Distance	Number	
image	String[]	
kostType	String	
publicFacility	String[]	
parking	String[]	
duration	String[]	
Rules	String	
Room { _id name price: ({ duration value }) panjang lebar electrityBill bathroom roomFacility available status createdAt updatedAt }	String String String Number	
status	String	Min: 10, Max: 25
createDate	Date	
updateAt	Date	

TABEL 3.7
Kamus Data Tabel Kontrakan

Name	Data type	Constrain
_Id (primary)	String	Required, Min: 24, Max: 24
User { _id Name Phone profilePictures }	String String String String	Required, Min: 24, Max: 24 Min: 11, Max: 100 Min: 11, Max: 16 Max: 25
Name	String	Required, Min: 2, Max: 100
description	String	Min: 10, Max: 255
type	String	Min: 2, Max: 25
area	String	Min: 10, Max: 255
address	String	Min: 10, Max: 255
distance	Number	
totalBedroom	Number	
totalBathroom	Number	
image	String[]	
furniture	String[]	
publicFacility	String[]	
parking	String[]	
duration	String[]	
rules	String	
Price	number	
Available	Boolean	
status	String	Min: 10, Max: 25
createDate	Date	
updateAt	Date	

TABEL 3.8
Kamus Data Tabel Nearby

Name	Data type	Constrain
_Id (primary)	String	Required, Min: 24, Max: 24
Name	String	Required, Min: 2, Max: 100
Description	String	Min: 10, Max: 255
phoneNumber	String	Min: 11, Max: 16
category	String	Min: 2, Max: 50
Area	String	Min: 10, Max: 255
address	String	Min: 10, Max: 255
Distance	Number	
image	String[]	
Price	number	
Delivered	Boolean	
status	String	Min: 10, Max: 25
createdAt	Date	
updateAt	Date	

TABEL 3.9
Kamus Data Tabel Link

Name	Data type	Constrain
_id	String	Required, Min: 24, Max: 24
token	String	Min: 40, Max: 40
link	String	Min: 10, Max: 100
email	String	Min: 10, Max: 100
expired	Date	
createdAt	Date	
updateAt	Date	

Verifikasi Basis Data

Berdasarkan NFR-10 pada dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak), basis data yang dibuat harus sesuai dengan pemodelan yang telah dibuat seperti nama tabel, atribut dan tipe data. Oleh karena itu dilakukan pengecekan antara basis

data dengan pemodelan yang telah dibuat. Hal ini terdapat pada Tabel 3.10 – 3.11.

TABEL 3. 10
Verifikasi Nama Tabel

Tabel	Nama Tabel Pada Basis Data	Sesuai
Users	Users	Ya
Kosts	Kosts	Ya
Kontrakan	Kontrakan	Ya
Nearby	Nearby	Ya
Link	Link	Ya

Pada hasil verifikasi nama tabel yang telah dilakukan, seluruh tabel yang ada pada basis data telah sesuai dengan pemodelan yang telah dibuat.

TABEL 3. 11
Verifikasi Atribut dan Tipe Data

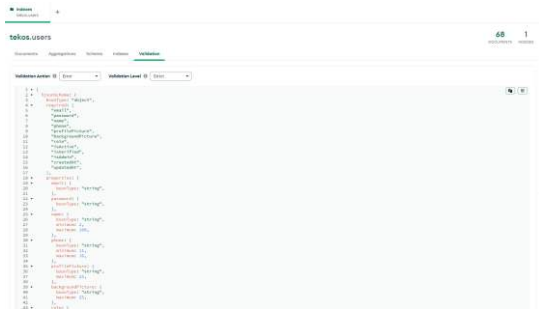
Nama Tabel	Atribut	Tipe	Sesuai
User	Id (primary)	String	Ya
	Email	String	Ya
	Password	String	Ya
	Name	String	Ya
	Phone	String	Ya
	profilePicture	String	Ya
	backgroundPicture	String	Ya
	Role	String	Ya
	isActive	Boolean	Ya
	isVerified	Boolean	Ya
	isAdmin	Boolean	Ya
	createDate	Date	Tidak
	updateAt	Date	Tidak
	Kosts	Id (primary)	String
User { id Name Phone profilePicture }		String String String String	Ya
Name		String	Ya
description		String	Ya
Area		String	Ya
address		String	Ya
Distance		Number	Ya
Image		String[]	Ya
kostType		String	Ya
publicFacility		String[]	Ya
parking		String[]	Ya
duration		String[]	Ya
Rules		String	Ya
Room [{ id name price: ({ duration value }) panjang lebar electricityBill bathroom roomFacility available status createdAt		String String String Number Number String String String[] Boolean Boolean	Ya

	updatedAt	String	String
	}}]		
	Status	String	Ya
	createDate	Date	Tidak
	updateAt	Date	Tidak
Kontrakan	Id (primary)	String	Ya
	User { _id Name Phone profilePictures	String String String String	Ya
	Name	String	Ya
	description	String	Ya
	Type	String	Ya
	Area	String	Ya
	Address	String	Ya
	Distance	Number	Ya
	totalBedroom	Number	Ya
	totalBathroom	Number	Ya
	Image	String[]	Ya
	Furniture	String[]	Ya
	publicFacility	String[]	Ya
	Parking	String[]	Ya
	Duration	String[]	Ya
	Rules	String	Ya
	Price	number	Ya
	Available	Boolean	Ya
	Status	String	Ya
	createDate	Date	Tidak
	updateAt	Date	Tidak
Nearby	Id (primary)	String	Ya
	Name	String	Ya
	Description	String	Ya
	phoneNumber	String	Ya
	Category	String	Ya
	Area	String	Ya
	Address	String	Ya
	Distance	Number	Ya
	Image	String[]	Ya
	Price	number	Ya
	Delivered	Boolean	Ya
	Status	String	Ya
	createdAt	Date	Ya
	updateAt	Date	Ya
Link	id	String	Ya
	token	String	Ya
	link	String	Ya
	email	String	Ya
	expired	Date	Ya
	createdAt	Date	Tidak
	updateAt	Date	Tidak

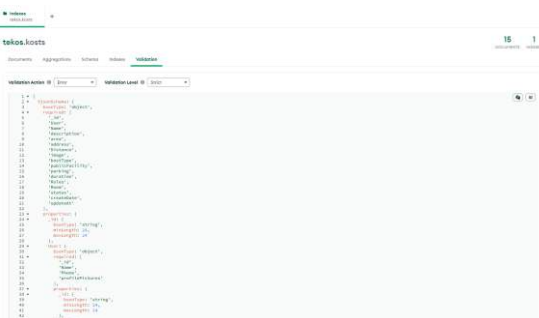
Pada hasil verifikasi atribut dan tipe data yang telah dilakukan, menunjukkan bahwa tabel users memiliki 13 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada tabel kosts memiliki 17 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada tabel kontrakan memiliki 21 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada tabel nearby memiliki 14 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada tabel link memiliki 7 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada keseluruhan tabel, terdapat 10 atribut yang memiliki tipe data tidak sesuai dimana atribut createdAt dan

updatedAt harusnya memiliki tipe data Date, pada basis data yang dibuat kedua atribut tersebut memiliki tipe data String.

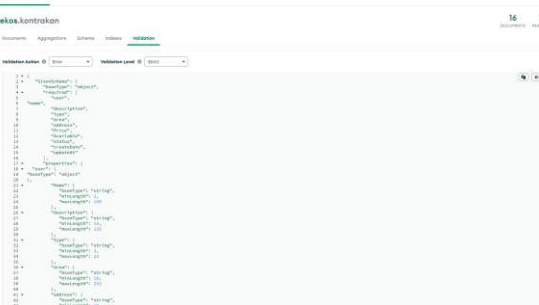
Selanjutnya dilakukan pengecekan pembatasan dan validasi. Berdasarkan NFR-11 pada dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak), basis data yang dibuat harus menerapkan pembatasan dan validasi data yang telah ditentukan. Oleh karena itu dilakukan pengecekan pada basis data dengan melihat pembatasan dan validasi pada MongoDB Compass. Hasil dari pengecekan tersebut terdapat pada Gambar 3.2 – 3.6.



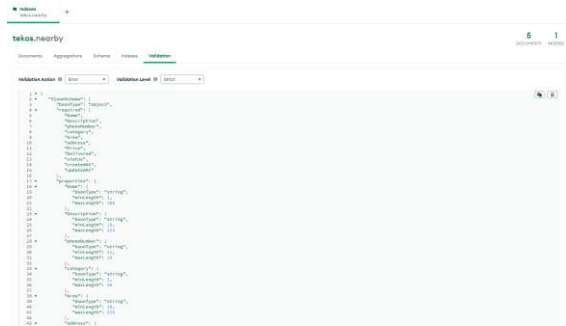
GAMBAR 3.2 Penerapan Pembatasan dan Validasi pada Tabel Users



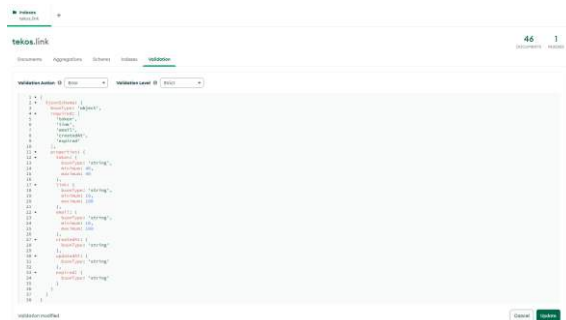
GAMBAR 3.3. Penerapan Pembatasan dan Validasi pada Tabel Kostos



GAMBAR 3.4. Penerapan Pembatasan dan Validasi pada Tabel Kontrakan



GAMBAR 3.5. Penerapan Pembatasan dan Validasi pada Tabel Nearby

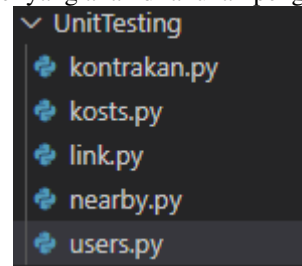


GAMBAR 3.6. Penerapan Pembatasan dan Validasi pada Tabel Link

F. Unit Testing

Pengujian unit menggunakan library dengan nama unittest dari python. Pada pengujian ini diperlukan tahapan konfigurasi terhadap basis data. Tahapan konfigurasi dilakukan dengan cara sebagai berikut:

- a) Membuat sebuah file dengan ekstensi “.py” untuk setiap tabel yang akan dilakukan pengujian.



GAMBAR 3.7. Pembuatan file .py untuk Unit Testing

- b) Mengimport library unittest, MongoClient dari pymongo dan datetime. Library-library tersebut digunakan untuk pengujian unit, koneksi ke database, dan pengecekan atribut bertipe date pada database.

```

1 import unittest
2 from pymongo import MongoClient
3 from datetime import datetime
4
    
```

GAMBAR 3.8 Import Library Yang Dibutuhkan

c) Menuliskan kode konfigurasi untuk terhubung ke database.

```
class TestUserInsert(unittest.TestCase):
    @classmethod
    def setUpClass(cls):
        cls.mongo_url = "mongodb://tekosAdmin:12
        cls.client = MongoClient(cls.mongo_url)
        cls.db = cls.client.tekos
        cls.collection = cls.db.users
```

GAMBAR 3.9
Kode Konfigurasi Database

d) Menuliskan kode konfigurasi data, dan testcase yang dibutuhkan untuk setiap tabel.

```
def setUp(self):
    self.test_data = {
        "email": "gilangg@gmail.com",
        "password": "bf83b76e7bda4ab84e",
        "name": "Gilang",
        "phone": "087823837566",
        "profilPicture": "profile-default.jpg",
        "backgroundPicture": "background-default.jpg",
        "role": "seeker",
        "isActive": True,
        "isVerified": False,
        "isAdmin": "admin",
        "createdAt": "2023-07-20 14:06:57",
        "updatedAt": "2023-08-12 00:17:03"
    }

    insert_result = self.collection.insert_one(self.test_data)
    self.inserted_id = insert_result.inserted_id

def tearDown(self):
    self.collection.delete_one({"_id": self.inserted_id})

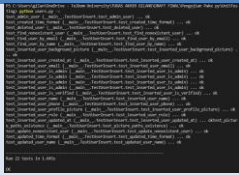

def test_inserted_user_exists(self):
    inserted_user = self.collection.find_one({"_id": self.inserted_id})
    self.assertIsNotNone(inserted_user)

def test_inserted_user_email(self):
    inserted_user = self.collection.find_one({"_id": self.inserted_id})
    self.assertEqual(inserted_user["email"], self.test_data["email"])
```

GAMBAR 3.10
Cuplikan Kode Konfigurasi Data dan Test Case

Berikut ini hasil dari unit testing database pada semua tabel yang ada. Pengujian dilakukan dengan beberapa skenario yang mewakili prosedur penyimpanan, pembaruan, pengambilan, dan penghapusan data.

TABEL 3.12
Penguian Unit

Tabel	Hasil	Catatan
Users		Dari 22 testcase yang diuji, semua testcase menunjukkan keberhasilan 100%
Kosts		Dari 14 testcase yang diuji, semua testcase menunjukkan keberhasilan 100%

Kontrakan		keberhasilan 100% Dari 18 testcase yang diuji, semua testcase menunjukkan keberhasilan 100%
Nearby		Dari 18 testcase yang diuji, semua testcase menunjukkan keberhasilan 100%
Link		Dari 12 testcase yang diuji, semua testcase menunjukkan keberhasilan 100%

Dari total 84 testcase yang melibatkan operasi-operasi penyimpanan, pengambilan, pembaruan, dan penghapusan data pada setiap tabel dengan rincian 22 testcase pada tabel users, 14 testcase pada tabel kosts, 18 testcase pada tabel kontrakan, 18 testcase pada tabel nearby, dan 12 testcase pada tabel link. Dapat dilihat tingkat keberhasilan dalam pengujian unit basis data ini adalah 100%, hal ini menunjukkan bahwa database berjalan dengan lancar sesuai harapan.

IV. EVALUASI

Pada hasil verifikasi atribut dan tipe data yang telah dilakukan, menunjukkan bahwa tabel users memiliki 13 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada tabel kosts memiliki 17 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada tabel kontrakan memiliki 21 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada tabel nearby memiliki 14 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada tabel link memiliki 7 atribut dengan 2 atribut memiliki tipe data tidak sesuai. Pada keseluruhan tabel, terdapat 10 atribut yang memiliki tipe data tidak sesuai dimana atribut createdAt dan updatedAt harusnya memiliki tipe data Date, pada basis data yang dibuat kedua atribut tersebut memiliki tipe data String.

Pada pengujian unit, didapatkan total 84 testcase yang melibatkan operasi-operasi penyimpanan,

pengambilan, pembaruan, dan penghapusan data pada setiap tabel dengan rincian 22 testcase pada tabel users, 14 testcase pada tabel kosts, 18 testcase pada tabel kontrakan, 18 testcase pada tabel nearby, dan 12 testcase pada tabel link. Dapat dilihat tingkat keberhasilan dalam pengujian unit basis data ini adalah 100%, hal ini menunjukkan bahwa database dapat berjalan dengan benar sesuai harapan.

V. KESIMPULAN

Berdasarkan hasil verifikasi basis data dapat disimpulkan bahwa basis data yang dibuat menyimpan data pengguna, kost, kontrakan, dan layanan di sekitar Telkom University. Oleh karena itu, basis data yang ada sudah menerapkan pembatasan dan validasi yang telah ditentukan sesuai dengan NFR-10 dan NFR-11. Selain itu, hasil dari pengujian unit basis data menunjukkan tingkat keberhasilan 100% dimana unit-test tersebut berhasil melewati pengujian sesuai dengan harapan. Namun tetap ada kemungkinan celah pada operasi basis data yang dapat membuat proses penyimpanan, penyimpanan, pengambilan, pembaruan, dan penghapusan data tidak berjalan dengan semestinya.

REFERENSI

- [1] Setiyowati and S. Siswanti, *PERANCANGAN BASIS DATA & PENGENALAN SQL SERVER MANAGEMENT STUDIO*. Lembaga Penelitian dan Pengabdian Kepada Masyarakat Universitas Dian Nuswantoro Semarang, 2021.
- [2] I. R. I. Astutik and M. A. Rosid, *Basis Data*. Sidoarjo: UMSIDA PRESS, 2020.
- [3] M. Fikry, *Basis Data*. Lhokseumawe: UNIMAL PRES, 2019.
- [4] W. Tarigan, tulus P. Sihalo, Lismardiana, I. Sembiring, and H. Silalahi, *Perancangan Basis Data*. Purbalingga: CV. EUREKA MEDIA AKSARA, 2021.
- [5] A. Aulia Aziiza and A. Nur Fadhilah, "Analisis Metode Identifikasi dan Verifikasi Kebutuhan Non Fungsional," *Applied Technology and Computing Science Journal*, vol. 3, no. 1, 2020, doi: 10.33086/atcsj.v3i1.1623.
- [6] A. N. Hasibuan and T. Dirgahayu, "Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna." *MULTINETICS*, vol. 7, no. 1, 2021, doi: 10.32722/multinetics.v7i1.3525.
- [7] M. A. Rizkyana, Y. Yunanto, Y. Yoga, and S. R. Widiyanto, "Implementasi Unit Testing menggunakan metode Test-First Development," *MULTINETICS*, vol. 7, no. 1, 2021, doi: 10.32722/multinetics.v7i1.3525.
- [8] M. Muslihudin and Oktafianto, *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*. Yogyakarta: CV Andi Offset, 2016. Accessed: Nov. 20, 2022. [Online]. Available: <https://books.google.co.id/books?id=2SU3DgAAQBAJ&lpg=PP1&authuser=1&pg=PP1#v=onepage&q&f=false>
- [9] Tri Amri Wijaya, Constantin Menteng, Afis Julianto, Adi Surya, and Ema Utami, "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*, vol. 15, no. 1, 2021, doi: 10.47111/jti.v15i1.1867.
- [10] S. S. Wibagso and E. Lia, "Desain Model Database Layanan Panti Werdha dengan Menerapkan Metode Database Life Cycle," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 3, Dec. 2020, doi: 10.28932/jutisi.v6i3.3047.