

## SURVEY KEPUASAN MAHASISWA BERBASIS ANDROID BAGIAN DATABASE (STUDI KASUS : XYZ)

### ANDROID BASED STUDENT SATISFACTION SURVEY DATABASE SECTION (CASE : XYZ)

Rakha Luthfi Fauzani<sup>1</sup>, Meta Kallista<sup>2</sup>, Ashri Dinimaharawati<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

[rakhalut@student.telkomuniversity.ac.id](mailto:rakhalut@student.telkomuniversity.ac.id), [metakallista@telkomuniversity.ac.id](mailto:metakallista@telkomuniversity.ac.id),

[ashridini@telkomuniversity.ac.id](mailto:ashridini@telkomuniversity.ac.id)

---

#### Abstrak

Survey Kepuasan Mahasiswa Berbasis Android Bagian Database (Studi Kasus: XYZ) merupakan penelitian yang dilakukan untuk solusi bagi Prodi S1 Matematika XYZ dalam mendata kepuasan mahasiswa prodi matematika XYZ. Proses pengumpulan data survey masih menggunakan google form sebagai media utama dalam mengumpulkan data dan harus diolah secara manual dalam hitungan excel yang cukup memakan waktu lama dalam proses pengumpulan data maupun olah data. Peranan Komputer merupakan salah satu yang terpenting dalam era digital seperti saat ini. Sistem *database* merupakan suatu sistem informasi yang dapat membantu proses pengelolaan data seperti input data, edit data, dan pencarian data sehingga dapat memudahkan pengguna untuk menemukan informasi yang diinginkan. Penggunaan MySQL sebagai *database* yang berbasis SQL menjadi salah satu pilihan untuk perancangan aplikasi survey kepuasan mahasiswa. Perpaduan MySQL yang bertindak sebagai tempat menyimpan data dari proses sistem yang dijalankan oleh *backend* menggunakan visual studio code dan tampilan untuk pengguna yang dijalankan oleh *frontend* menggunakan flutter menjadi satu kesatuan yang menjadikan aplikasi survey kepuasan mahasiswa berbasis android ini berjalan dengan baik. Hasil yang didapatkan dalam pengujian aplikasi ini menggunakan normalisasi dan integritas data dari beberapa *test case*.

Kata kunci : Aplikasi, MySQL, *database*, survey kepuasan, mahasiswa, *Android*

---

#### Abstract

*Android-Based Student Satisfaction Survey Database Section (Case Study: XYZ) is a research conducted for a solution for XYZ Mathematics S1 Study Program in recording the satisfaction of mathematics study program students at XYZ. The survey data collection process still uses google form as the main media in collecting data and must be processed manually in excel which is quite time consuming in the process of data collection and data processing. The role of computers is one of the most important in today's digital era. Database system is an information system that can help the data management process such as data input, data editing, and data search so that it can facilitate users to find the desired information. The use of MySQL as a SQL-based database is one of the choices for designing student satisfaction survey applications. The combination of MySQL which acts as a place to store data from system processes run by the backend using visual studio code and the display for users run by the frontend using flutter is a unity that makes this android-based student satisfaction survey application run well. The results obtained in testing this application using normalization and data integrity from several test cases.*

*Keywords: Application, MySQL, database, survey satisfaction, student, Android*

---

#### 1. Pendahuluan

Teknologi dapat disebut sebagai salah satu poros perkembangan dunia dikarenakan teknologi berada pada setiap aspek kehidupan manusia seperti karir, pendidikan, kesehatan, dan juga produksi pangan.

Salah satu metode penelitian untuk mengumpulkan dan mempelajari berbagai informasi dari informan yang berbeda disebut dengan survei yang berguna untuk membuat rencana, kesimpulan dan keputusan. Kemajuan teknologi pada era industri 4.0 ini semakin pesat, sehingga kebutuhan akan informasi yang mudah diakses dan akurat sangat dibutuhkan.

Kumpulan informasi yang sistematis tersimpan dalam komputer disebut dengan *database*. Jadi secara konsep basis data atau *database* adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tatacara yang tertentu untuk membentuk data baru atau informasi[1].

Penulis berangkat dari permasalahan di kampus XYZ yang masih menggunakan google form sebagai sarana survey yang dinilai kurang efektif untuk jangka waktu yang panjang maka dari itu dibuatlah aplikasi SURVEY KEPUASAN MAHASISWA BERBASIS ANDROID BAGIAN DATABASE.

## 2. Dasar Teori

### 2.1 Survey Kepuasan

Survey menjadi metode efektif dalam mengumpulkan data dikarenakan keunggulan yang dimiliki oleh metode ini yaitu mudah untuk diakses oleh mahasiswa yang dapat digunakan kapanpun dan dimanapun karena bersifat online dari perangkat android yang dimiliki oleh mahasiswa. Keuntungan dalam penggunaan metode survey adalah mengelola survei relatif mudah, data yang diperoleh dapat dipercaya, jawabannya terbatas pada alternatif yang telah ditentukan, dan pemberian kode, analisis dan interpretasi data juga relatif mudah[2].

### 2.2 Database

Database adalah kumpulan fakta dan angka yang saling berkaitan dan dapat diproses untuk menghasilkan informasi. Database merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut [2]. Database yang dipakai meliputi:

#### 1. MYSQL

MySQL merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user.

#### 2. PHPmyAdmin

PHPMyAdmin adalah sebuah aplikasi perangkat lunak yang ditulis secara open source. Dalam bahasa pemrograman PHP yang digunakan untuk manajemen database MySQL melalui jaringan area lokal atau jaringan Internet.

#### 3. Entity Relationship Diagram

Entity Relationship Diagram atau ERD adalah sebuah diagram struktural yang digunakan untuk merancang sebuah database. Berikut adalah beberapa komponen utama dalam ERD menurut antara lain [3]:

##### a. Entitas

Entitas adalah objek yang bisa berupa orang, benda, ataupun keterangan atau kondisi yang memiliki syarat menjadi entitas yaitu memiliki atribut, kunci primer dan relasi[3]

##### b. Atribut

Atribut merupakan informasi atau keterangan tentang keunikan dari sebuah entitas berupa kunci utama atau *primary key* sehingga memungkinkan suatu objek dapat menjadi sebuah entitas[3].

##### c. Relasi

Relasi atau hubungan yang merupakan jembatan bagi setiap entitas yang memiliki kunci utama atau *primary key* terhubung dengan kunci asing atau *foreign key* sehingga memungkinkan entitas untuk terkoneksi[4]. Dalam relasi dapat dibedakan untuk beberapa jenis hubungan atau bisa disebut dengan kardinalitas seperti[4] :

- *one-to-one*

Hubungan atau relasi yang menunjukkan satu entitas dengan satu entitas yang lain[4].

- *one-to-many*

Hubungan atau relasi yang menunjukkan satu entitas dengan banyak entitas yang lain[4].

- *many-to-many*

Hubungan atau relasi yang menunjukkan banyak entitas dengan banyak entitas lainnya[4].

### 2.3 Business Rule

Business rules merupakan aturan yang berhubungan dengan interaksi dan suatu kejadian dari sistem yang akan dibuat[5]. Business rule terdiri taksonomi yang disederhanakan mencakup enam kategori yaitu[6]:

a. Fakta adalah pernyataan yang menjelaskan entitas dan hubungan dalam model data[6].

b. Batasan (constraints) atau adalah pernyataan yang membatasi tindakan yang boleh dilakukan oleh penggunaanya[6].

c. Pemicu aksi (Action enablers) adalah aturan yang memicu beberapa aktivitas jika suatu kondisi adalah benar[6].

d. Inferensi (inferences) adalah aturan yang digunakan dalam pembuatan fakta baru dari fakta yang ada[6].

e. Komputasi adalah suatu algoritma yang dapat mengubah nilai numerik yang sudah ada menjadi nilai baru[6].

f. Ketentuan (terms) berupa singkatan, frasa dan kata yang memiliki nilai[6].

### 2.4 Normalisasi Data

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan. Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sebagai berikut :

a. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (Lossless-Join Decomposition). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis[12].

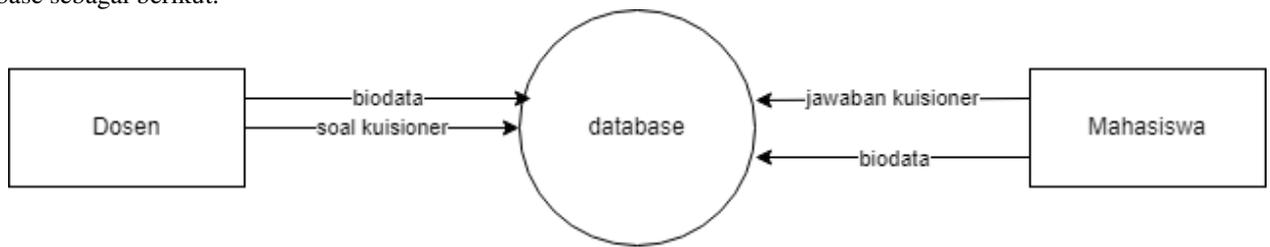
b. Terpeliharanya ketergantungan fungsional pada saat perubahan data (Dependency Preservation)[12].

c. Tidak melanggar Boyce-Codd Normal Form (BCNF)[12].

### 3. Pembahasan

#### 3.1. Gambaran Umum Database

Gambaran umum yang dibuat menggunakan context diagram yang merupakan konsep awal dari pembuatan database sebagai berikut.



Gambar 1 Context Diagram

#### 3.2 Business Rule

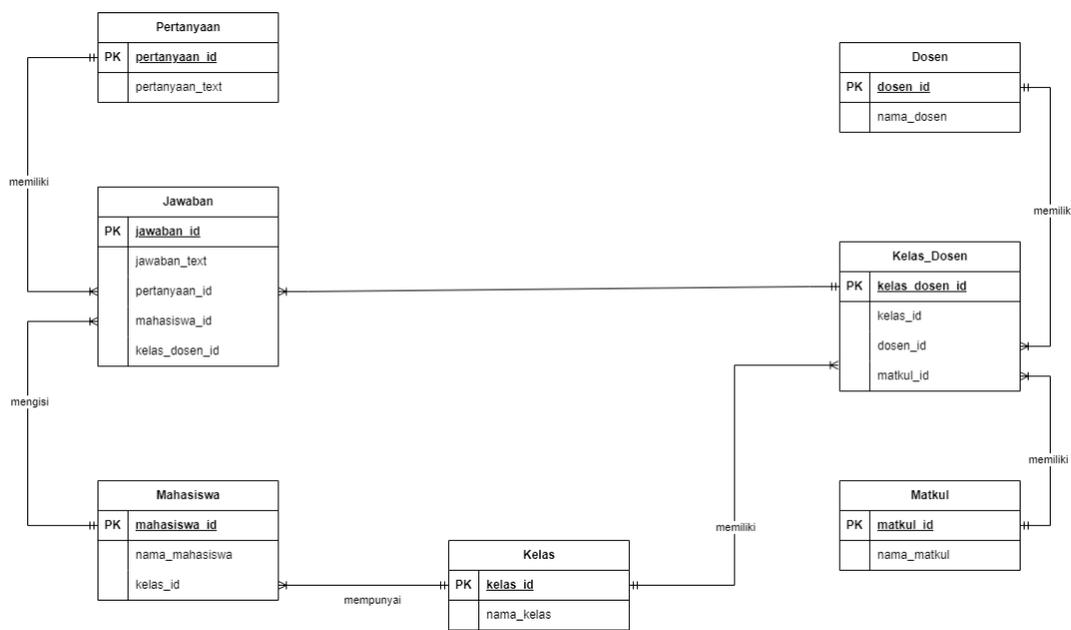
Business rule disini adalah peraturan yang dimiliki oleh setiap entitas yaitu admin, dosen dan mahasiswa. Sebelum membuat business rule diperlukan penjelasan dengan berupa matriks entitas terhadap fungsi yaitu sebagai berikut :

Tabel 1 Analisa Kebutuhan

Kode Rule	Rule
R:1	seorang Mahasiswa memiliki banyak Dosen
R:2	seorang Dosen memiliki banyak mahasiswa
R:3	seorang Mahasiswa mengisi banyak Pertanyaan
R:4	satu Pertanyaan diisi oleh banyak Mahasiswa
R:5	seorang Dosen memiliki banyak Pertanyaan
R:6	satu Pertanyaan dimiliki oleh banyak Dosen
R:7	seorang Dosen mengampu banyak Matkul
R:8	satu Matkul diampu oleh satu atau lebih Dosen
R:9	satu Dosen memiliki satu atau lebih Kelas
R:10	satu Kelas memiliki banyak Dosen
R:11	satu Matkul memiliki banyak Kelas
R:12	satu Kelas memiliki banyak Matkul
R:13	satu Mahasiswa memiliki satu Kelas
R:14	satu Kelas memiliki banyak Mahasiswa

#### 3.3 ERD

Entity Relationship Diagram (ERD) Mendesain dengan basis data ini hubungan detail antara objek entitas dan atributnya, Dibentuk dengan penggunaan yang terstruktur dan berbeda dengan beberapa notasi dan simbol.



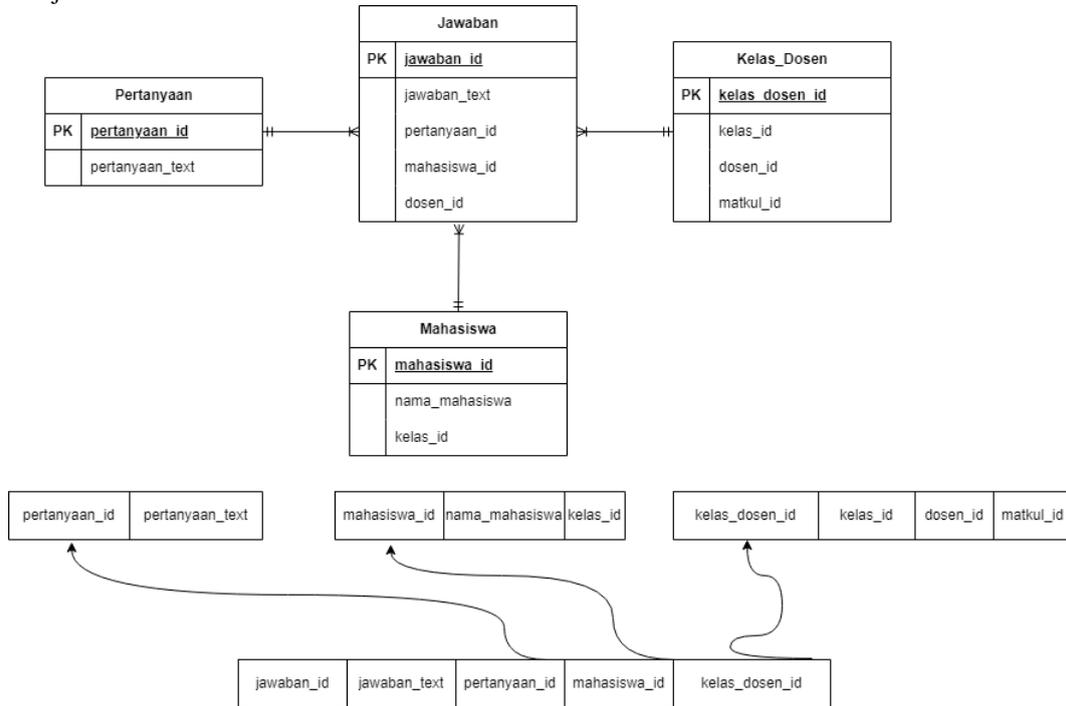
Gambar 2 ERD

#### 4. Pengujian dan Analisis

##### 4.1 Implementasi Tabel Relasi

Berikut ini adalah bahasa SQL yang berguna untuk menunjukkan relasi dari satu entitas ke entitas lainnya sebagai berikut:

###### 1. Relasi jawaban



Gambar 3 Gambar Relasi Jawaban

Pada tabel jawaban memiliki relasi antara tabel pertanyaan, dosen, dan mahasiswa. Tabel jawaban dapat diimplementasikan menggunakan bahasa query dibawah ini.

Tabel 2 Relasi Jawaban dan Pertanyaan

Relasi	Data Definition Language
jawaban to pertanyaan	ALTER TABLE `jawaban` ADD CONSTRAINT `fk_jawaban_pertanyaan` FOREIGN KEY (`pertanyaan_id`) REFERENCES `pertanyaan` (`pertanyaan_id`) ON DELETE CASCADE ON UPDATE CASCADE;

Pembuatan relasi menggunakan bahasa query antara tabel jawaban dengan tabel pertanyaan memakai sql 'ALTER TABLE' untuk merelasikan tabel jawaban dengan memiliki foreign key `pertanyaan\_id` yang berasal dari tabel pertanyaan. Bila ada data yang terhapus atau diubah maka fitur cascade disini berfungsi untuk memperbaharunya.

Tabel 3 Relasi Jawaban dan Mahasiswa

Relasi	Data Definition Language
jawaban to mahasiswa	ALTER TABLE `jawaban` ADD CONSTRAINT `fk_jawaban_mahasiswa` FOREIGN KEY (`mahasiswa_id`) REFERENCES `mahasiswa` (`mahasiswa_id`) ON DELETE CASCADE ON UPDATE CASCADE;

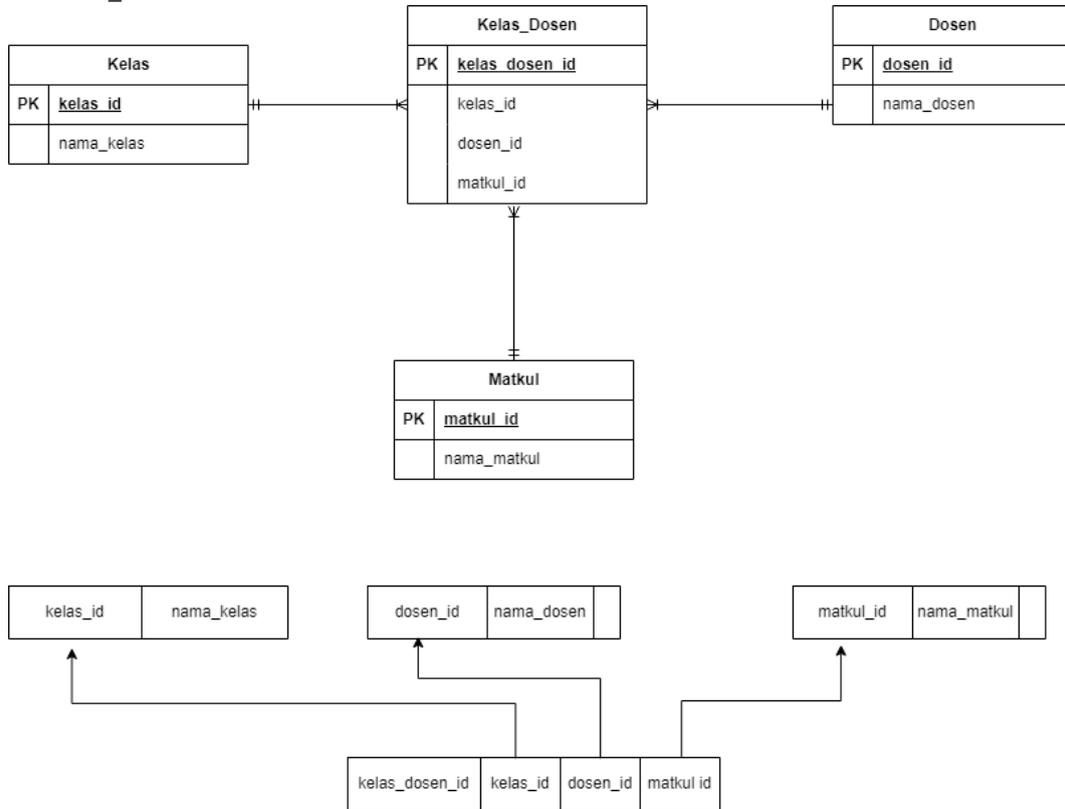
Pembuatan relasi menggunakan bahasa query antara tabel jawaban dengan tabel mahasiswa memakai sql 'ALTER TABLE' untuk merelasikan tabel jawaban dengan memiliki foreign key `mahasiswa\_id` yang berasal dari tabel mahasiswa. Bila ada data yang terhapus atau diubah maka fitur cascade disini berfungsi untuk memperbaharunya.

Tabel 4 Relasi Jawaban dan Kelas Dosen

Relasi	Data Definition Language
jawaban to mahasiswa	ALTER TABLE `jawaban` ADD CONSTRAINT `fk_jawaban_kelas_dosen` FOREIGN KEY (`kelas_dosen_id`) REFERENCES `dosen` (`dosen_id`) ON DELETE CASCADE ON UPDATE CASCADE;

Pembuatan relasi menggunakan bahasa query antara tabel jawaban dengan tabel dosen memakai sql 'ALTER TABLE' untuk merelasikan tabel jawaban dengan memiliki foreign key 'kelas\_dosen\_id' yang berasal dari tabel dosen. Bila ada data yang terhapus atau diubah maka fitur cascade disini berfungsi untuk memperbaharainya.

2. Relasi kelas\_dosen



Gambar 4 Relasi Kelas Dosen

Pada tabel kelas\_dosen memiliki relasi antara tabel kelas, dosen dan matkul. Tabel kelas\_dosen dapat diimplementasikan menggunakan bahasa query dibawah ini.

Tabel 5 Tabel Relasi Kelas Dosen dan Kelas

Relasi	Data Definition Language
kelas_dosen to kelas	ALTER TABLE `kelas_dosen` ADD CONSTRAINT `fk_kelas_dosen_kelas_id` FOREIGN KEY (`kelas_id`) REFERENCES `kelas` (`kelas_id`) ON DELETE CASCADE ON UPDATE CASCADE;

Pembuatan relasi menggunakan bahasa query antara tabel kelas\_dosen dengan tabel kelas dengan memakai sql 'ALTER TABLE' untuk merelasikan tabel kelas\_dosen dengan tabel kelas yang memiliki foreign key 'kelas\_id' yang berasal dari tabel kelas. Bila ada data yang terhapus atau diubah maka fitur cascade disini berfungsi untuk memperbaharainya.

Tabel 6 Relasi Kelas Dosen dan Dosen

Relasi	Data Definition Language
kelas_dosen to dosen	ALTER TABLE `kelas_dosen` ADD CONSTRAINT `fk_kelas_dosen_dosen` FOREIGN KEY (`dosen_id`) REFERENCES `dosen` (`dosen_id`) ON DELETE CASCADE ON UPDATE CASCADE;

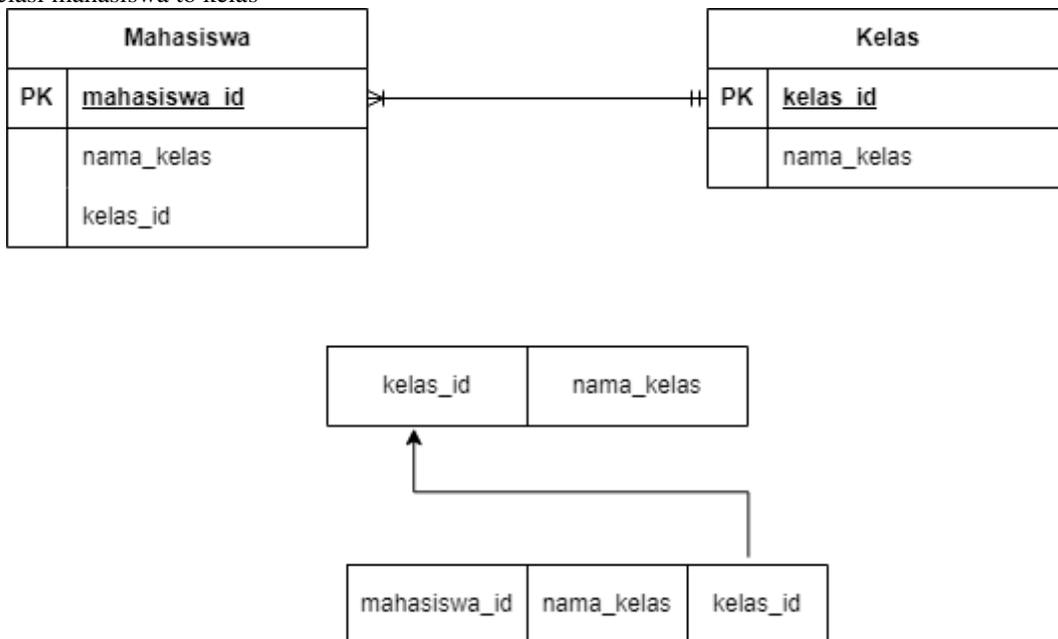
Pembuatan relasi menggunakan bahasa query antara tabel kelas\_dosen dengan tabel dosen dengan memakai sql 'ALTER TABLE' untuk merelasikan tabel kelas\_dosen dengan tabel dosen yang memiliki foreign key 'dosen\_id' yang berasal dari tabel dosen. Bila ada data yang terhapus atau diubah maka fitur cascade disini berfungsi untuk memperbaharainya.

Tabel 7 Relasi Kelas Dosen dan Matkul

Relasi	Data Definition Language
kelas_dosen to matkul	ALTER TABLE `kelas_dosen` ADD CONSTRAINT `fk_kelas_dosen_matkul_id` FOREIGN KEY (`matkul_id`) REFERENCES `matkul` (`matkul_id`) ON DELETE CASCADE ON UPDATE CASCADE;

Pembuatan relasi menggunakan bahasa query antara tabel kelas\_dosen dengan tabel matkul dengan memakai sql 'ALTER TABLE' untuk merelasikan tabel kelas\_dosen yang memiliki foreign key 'matkul\_id' yang berasal dari tabel matkul. Bila ada data yang terhapus atau diubah maka fitur cascade disini berfungsi untuk memperbaharainya.

3. Relasi mahasiswa to kelas



Gambar 5 Relasi Mahasiswa dan Kelas

Pada tabel mahasiswa memiliki relasi tabel kelas. Tabel hasil dapat diimplementasikan menggunakan bahasa query dibawah ini.

Tabel 8 Relasi Mahasiswa dan Kelas

Relasi	Data Definition Language
mahasiswa to kelas	ALTER TABLE `mahasiswa` ADD CONSTRAINT `fk_mahasiswa_kelas_id` FOREIGN KEY (`kelas_id`) REFERENCES `kelas` (`kelas_id`) ON DELETE CASCADE ON UPDATE CASCADE;

Pembuatan relasi menggunakan bahasa query antara tabel mahasiswa dengan tabel kelas dengan memakai sql 'ALTER TABLE' untuk merelasikan tabel mahasiswa yang memiliki foreign key 'kelas\_id' yang berasal dari tabel kelas. Bila ada data yang terhapus atau diubah maka fitur cascade disini berfungsi untuk memperbaharainya.

4.2 Pengujian

Tabel 9 Rekap Uji Primary Key

No	Skenario pengujian	Hasil yang diharapkan	Hasil Uji
1	Uji dilakukan dengan input duplikat primary key dosen	Muncul error karena nilai duplikat	Terpenuhi
2	Uji dilakukan dengan input duplikat primary key jawaban	Muncul error karena nilai duplikat	Terpenuhi
3	Uji dilakukan dengan input duplikat primary key kelas	Muncul error karena nilai duplikat	Terpenuhi
4	Uji dilakukan dengan input duplikat primary key kelas_dosen	Muncul error karena nilai duplikat	Terpenuhi
5	Uji dilakukan dengan input duplikat primary key mahasiswa	Muncul error karena nilai duplikat	Terpenuhi
6	Uji dilakukan dengan input duplikat primary key matkul	Muncul error karena nilai duplikat	Terpenuhi
7	Uji dilakukan dengan input duplikat primary key pertanyaan	Muncul error karena nilai duplikat	Terpenuhi

Tabel 10 Rekap Uji Foreign Key

No	Skenario Pengujian	Hasil yang diharapkan	Hasil uji
1	Input nilai foreign key tidak sesuai pada jawaban.	Muncul pesan kesalahan.	Terpenuhi
2	Input nilai foreign key tidak sesuai pada kelas_dosen.	Muncul pesan kesalahan.	Terpenuhi
3	Input nilai foreign key tidak sesuai pada mahasiswa.	Muncul pesan kesalahan.	Terpenuhi

Tabel 11 Rekap Uji Referensial Integritas

No	Skenario Pengujian	Hasil yang diharapkan	Hasil uji
1	Bila ada input yang dihapus dari setiap foreign key akan berpengaruh pada tabel jawaban.	Input pada jawaban akan terhapus juga.	Terpenuhi
2	Bila ada input yang dihapus dari setiap foreign key akan berpengaruh pada tabel kelas_dosen.	Input pada kelas_dosen akan terhapus juga.	Terpenuhi
3	Bila ada input yang dihapus dari setiap foreign key akan berpengaruh pada tabel mahasiswa.	Input pada mahasiswa akan terhapus juga.	Terpenuhi

Tujuan uji integritas data adalah untuk memastikan bahwa setiap komponen yang ada dalam database berfungsi sesuai dengan harapan.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Hasil yang dapat disimpulkan dari pembangunan aplikasi survei kepuasan ini adalah sebagai berikut :

1. MySQL sebagai fitur database yang berbasis MySQL membuat penyimpanan menjadi lebih efisien.
2. Hasil yang didapatkan pada pengujian mendapatkan hasil yang sesuai dengan harapan.
3. Metode penelitian waterfall menjadikan rancangan aplikasi dan database cukup baik dan terstruktur.

### 5.2 Saran

Masih ada beberapa kekurangan yang menjadikan kelemahan pada perancangan dan pembuatan aplikasi survei kepuasan mahasiswa ini antara lain:

1. Data yang ada pada MySQL tidak termuat secara detail dikarenakan memakai data dummy sehingga beberapa data seperti salah satu nya persentase hasil survei tidak termuat.
2. Database ini hanya support pada localhost sehingga memerlukan hosting agar dapat terkoneksi dengan beberapa device.
3. Database yang dibuat masih dapat terbilang sangat sederhana sehingga untuk penelitian selanjutnya dapat dibuat lebih baik lagi.

### Daftar Pustaka:

- [1] A. Andaru, "PENGERTIAN DATABASE SECARA UMUM," 2018.
- [2] H. Malik, M. Nuh, and M. H. Fatoni, "Perancangan database Informasi Medis Untuk Sistem Prediksi Hipertensi," *Jurnal Teknik ITS*, vol. 9, no. 1, 2020.
- [3] A. Amijaya, F. Ferdinandus, M. Bayu, S. Tinggi Teknologi Cahaya Surya Kediri, and S. Tinggi Teknik Surabaya, "Sistem Pendukung Keputusan Pemilihan Handphone Dengan Metode Simple Additive Weighting Berbasis WEB," 2019.
- [4] K. ' Afiifah, Z. Fira Azzahra, A. D. Anggoro, D. Redaksi, R. Akhir, and D. Online, "Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database: Sebuah Literature Review," *JURNAL INTECH*, vol. 3, no. 1, pp. 8–11.
- [5] L. Kesmasari, C. Agus Sugianto, and P. Studi Teknik Informatika Politeknik TEDC Bandung, "APLIKASI MOBILE PENCARIAN JALUR ANGKUTAN UMUM DI KOTA CIMAHI BERBASIS ANDROID."
- [6] A. Kopp, D. Orlovskiy, and S. Orekhov, "An Approach and Software Prototype for Translation of Natural Language Business Rules into Database Structure."
- [7] S. Suryadi, "IMPLEMENTASI NORMALISASI DALAM PERANCANGAN DATABASE RELATIONAL," *Jurnal Teknik Informatika*, vol. 3, no. 2, 2019.