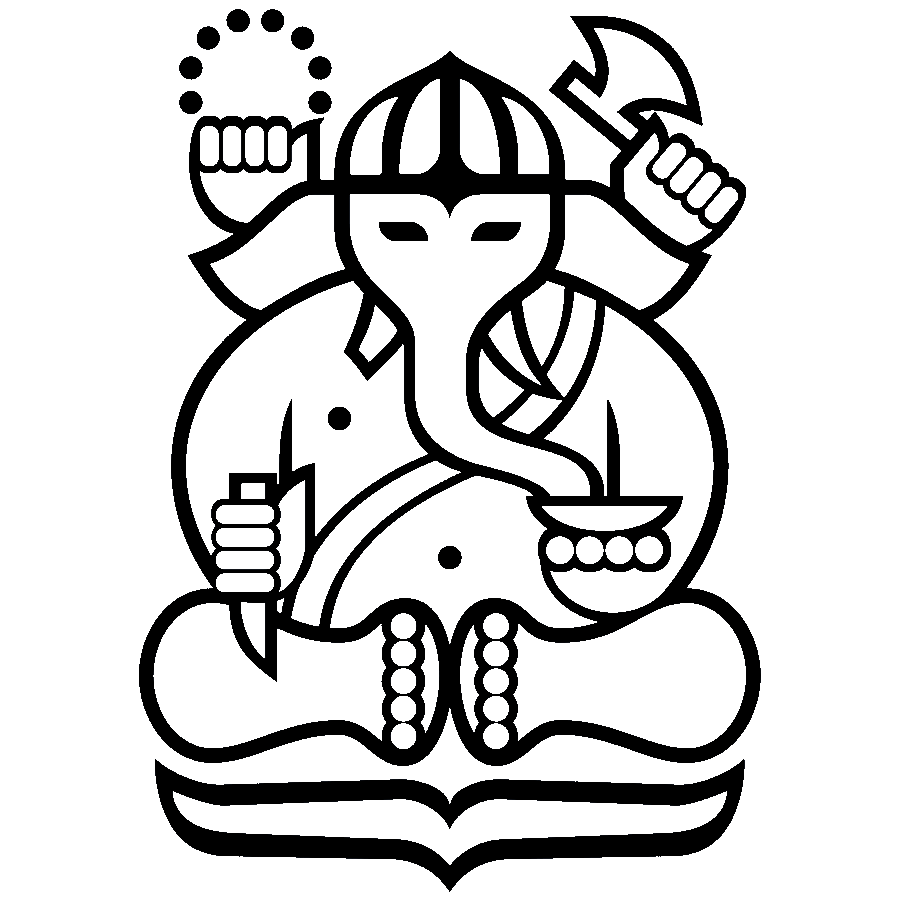
**MILESTONE 1 IF2240 BASIS DATA**

**SEMESTER II TAHUN 2022/2023**

**Pemodelan Entity-Relationship**

****

Disusun oleh :

13516055 - Nathaniel Evan Gunawan

13521057 - Hosea Nathanael Abetnego

13521105 - Haidar Hamda

13521127 - Marcel Ryan Antony

13521153 - Made Debby Almadea Putri

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2023**

# **DAFTAR ISI**

[**DAFTAR ISI 2**](#_d867cmubl5up)

[**DAFTAR GAMBAR 4**](#_3pop6s9u3y2g)

[**DAFTAR TABEL 5**](#_is453dgnu9f2)

[**Deskripsi Sistem 6**](#_ha68w79rzpou)

[**Entity Relationship Model 7**](#_drtmlnwyjev3)

[2.1. Diagram (Revisi) 7](#_mj4h7cdu8ulq)

[2.2. Deskripsi (Revisi) 7](#_upsszgw2if6n)

[2.2.1. Entitas User 7](#_fi91vgjnslmy)

[2.2.2. Entitas Pertanyaan 7](#_47t6mvi8dbrs)

[**Pemetaan Model ER menjadi Relasional 9**](#_spoxle1y9zep)

[3.1. Langkah Pemetaan 9](#_xdke32xyj6pm)

[3.1.1. Entity Latihan Soal 9](#_k0nekwp3clla)

[3.1.2. Entity Pertanyaan 9](#_jub9c8atkqrh)

[3.1.3. Entity Modul 9](#_a5q1l6712q6j)

[3.1.4. Entity User 9](#_9ov7h6jq15jt)

[3.1.5. Entity Materi 9](#_wcn17lz11m54)

[3.1.6. Entity Diktat Pembelajaran 10](#_tmsxkzap8ofb)

[3.1.7. Entity Video Pembelajaran 10](#_ju6axhpv5ayi)

[3.1.8. Entity Mata Kuliah 10](#_81dzgukkjsz)

[3.1.9. Entity Jurusan 10](#_iv2xyx6bx4kq)

[3.1.10. Entity Fakultas 10](#_1w79t0z9ojkq)

[3.1.11. Relationship antara Latihan Soal dan Pertanyaan 10](#_w6iitzw35rnf)

[3.1.12. Relationship antara Latihan Soal dan Modul 10](#_xrxuvagf75bh)

[3.1.13. Relationship antara User dan Latihan Soal 11](#_j24p1cyfvba2)

[3.1.14. Relationship antara User dan Materi 11](#_jpxu373ciuda)

[3.1.15. Relationship antara User dan Mata Kuliah 11](#_2wyvz4syt3gp)

[3.1.16. Relationship antara Mata Kuliah dan Modul 11](#_7hug7aiaiudo)

[3.1.17. Relationship antara Modul dan Materi 11](#_xi0a5eb33h6c)

[3.1.18. Relationship antara Mata Kuliah dan Jurusan 11](#_q6hj9lrovkug)

[3.1.19. Relationship antara Jurusan dan Fakultas 11](#_io79zf5f66nh)

[3.2. Model Relasional yang Dihasilkan 12](#_jbb8kurnptml)

[**Implementasi dan Pengujian 13**](#_ddyi3c3qko97)

[4.1. Insight dan Query dari Basis Data 13](#_i2hggehqqnip)

[**REFERENSI 17**](#_je0bi0nz56on)

[**LAMPIRAN 18**](#_b0hi3ph9l85)

[1. Deskripsi Wawancara 18](#_fjjogvaz6es9)

[2. Hasil Wawancara 18](#_f7qwj0ymth9)

# **DAFTAR GAMBAR**

# 

[Gambar 2.1.1 Entity Relationship Diagram](#_b80zhkvrpmho) 7

Gambar 3.2.1 Model Relasional12

[Gambar Dokumentasi](#_v5rq7tr6n5i3) 18

# 

# 

# 

# **DAFTAR TABEL**

# 

Tabel Insight *Database* yang dihasilkan 13

[Tabel Deskripsi Wawancara](#_epk6l6rl462g) 17

[Tabel Hasil Wawancara](#_wlmwdq66fnqv) 17

# 

**BAB 1**

# **Deskripsi Sistem**

OpenCourseWare (OCW) adalah istilah platform pembelajaran daring yang dibuat oleh sebuah perguruan tinggi untuk menyediakan materi perkuliahan dari perguruan tinggi tersebut secara gratis. Materi perkuliahan dapat dibuat dalam bentuk slide perkuliahan, referensi tambahan, video pembelajaran, serta latihan soal. Materi ini dikelompokkan berdasar mata kuliah atau jurusan tertentu. Sekelompok mahasiswa ITB sedang mencoba untuk membuat website ITB OpenCourseWare dan meminta anda mendesain basis data untuk website ini. ITB OpenCourseWare adalah website yang menyediakan bahan ajar berupa slide perkuliahan, video, serta latihan soal dari kegiatan perkuliahan ITB untuk dapat diakses oleh berbagai kalangan.

Untuk mengakses materi, pengguna perlu melakukan pendaftaran terlebih dahulu untuk mengakses materi dan mengerjakan latihan soal. Untuk mendaftar, pengguna perlu mengisi nama lengkap, username, email, dan password. Kemudian pengguna diperlukan untuk memverifikasi email tersebut.

Setiap mata kuliah berada di bawah satu jurusan dan memiliki kode tertentu yang unik, dan tiap jurusan berada di bawah satu fakultas dengan singkatan yang unik untuk tiap fakultasnya. Setiap mata kuliah akan memiliki modul pembelajaran yang terurut mulai dari modul pertama. Modul pembelajaran tersebut akan berisi materi yang dapat dipelajari oleh pengguna. Untuk menyajikan materi, website ini akan memanfaatkan video pembelajaran dengan durasi tertentu dan berkas materi bacaan seperti slide dengan jumlah halaman tertentu yang akan diunggah pada website ini. Tiap materi dapat berisi beberapa media pembelajaran.

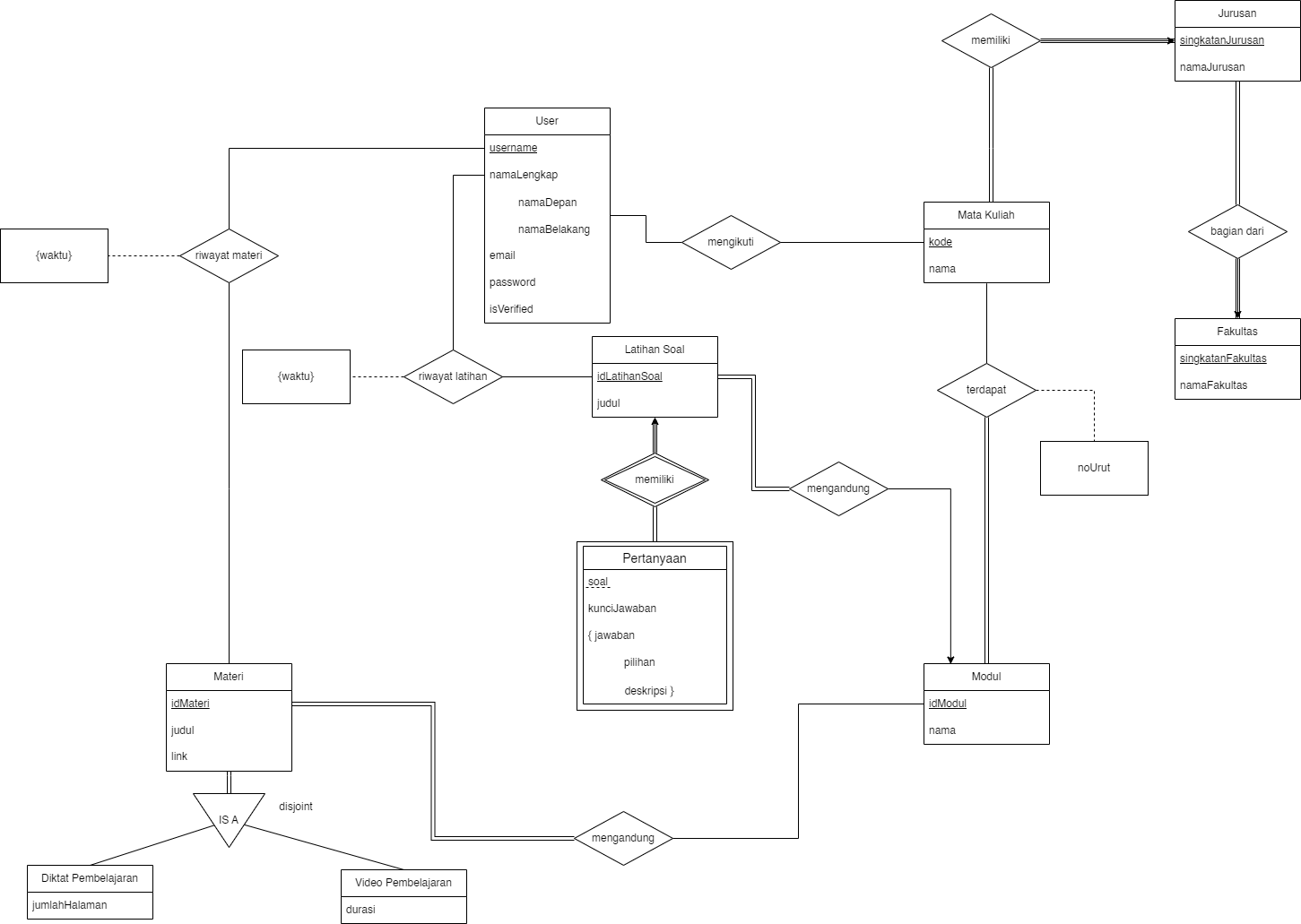
Selain penyajian materi, terdapat pula latihan soal yang dibuat dalam bentuk pilihan ganda dan dapat dikerjakan berkali-kali oleh pengguna. Hanya ada satu jawaban benar di antara pilihan lainnya. Latihan soal tidak harus selalu ada di tiap materi.

Website juga mencatat riwayat pengguna berupa tanggal dalam membaca materi ataupun mengerjakan latihan soal.

**BAB 2**

# **Entity Relationship Model**

## 2.1. Diagram (Revisi)



Gambar 2.1.1 Entity Relationship Diagram

## 2.2. Deskripsi (Revisi)

### 2.2.1. Entitas User

Atribut “namaLengkap” pada entitas user direvisi menjadi atribut komposit yang terdiri dari namaDepan dan namaBelakang. Hal ini karena pemisahan nama lengkap menjadi nama depan dan nama belakang juga memungkinkan aplikasi atau sistem untuk menuliskan nama pengguna dengan format tertentu

### 2.2.2. Entitas Pertanyaan

Atribut “listJawaban” pada entitas pertanyaan direvisi menjadi atribut “jawaban” dan bersifat komposit multivalued yang terdiri dari pilihan dan deskripsi. Revisi ini dilakukan karena mempertimbangkan kunciJawaban yang disimpan hanya berupa pilihan seperti A, B, C, atau D. Dalam ERD sebelumnya, “listJawaban” bersifat atomic multivalued seperti "A. ER Diagram". Data seperti ini akan menyulitkan sistem dalam menambahkan jawaban karena perlu menambahkan pilihannya juga. Dengan adanya atribut "jawaban" yang terdiri dari pilihan dan deskripsi, kita dapat lebih mudah mengelola dan mencocokkan kunci jawaban dengan jawaban yang telah diberikan oleh pengguna. Revisi ini akan memudahkan sistem dalam mengelola data jawaban dan mencocokkan kunci jawaban dengan jawaban yang telah diberikan oleh pengguna.

**BAB 3**

# **Pemetaan Model ER menjadi Relasional**

## Langkah Pemetaan

### Entity Latihan Soal

Entitas Latihan Soal merupakan *strong entity* sehingga relasi Latihan Soal memiliki atribut yang sama dengan entitas-nya. *Primary key* dari relasi Latihan Soal sama dengan entitas Latihan Soal yaitu atribut idLatihanSoal. Latihan Soal memiliki *relationship* *many-to-one* dengan entitas Modul sehingga pada relasi Latihan Soal juga menyimpan *foreign key* ke relasi Modul yaitu idModul karena satu Latihan Soal hanya terkait ke satu Modul.

### **Entity Pertanyaan**

Latihan Soal memiliki relasi dengan *weak entity* yaitu Pertanyaan. Karena Pertanyaan merupakan *weak entity* yang *dependent* dengan *entity* Latihan Soal, maka Pertanyaan dibuat menjadi sebuah relasi dengan tambahan *primary key* yang merupakan *foreign key* yaitu idLatihanSoal yang mengacu pada entitas Latihan Soal. Pada pertanyaan, terdapat atribut *multivalue* yaitu jawaban, sehingga diperlukan sebuah relasi yang merepresentasikan jawaban tersebut. Pada relasi jawaban, terdapat atribut idLatihanSoal, soal, dan pilihan sebagai *primary key* dengan idLatihanSoal dan soal mengacu pada idLatihanSoal dan soal pada relasi Pertanyaan dilengkapi dengan atribut deskripsi yang menjelaskan *detail* dari pilihan tersebut.

### Entity Modul

Entitas Modul merupakan *strong entity* sehingga relasi Modul memiliki atribut yang sama dengan entitas-nya. *Primary key* dari relasi Modul sama dengan entitas Modul yaitu atribut idModul.

### Entity User

Entitas User merupakan *strong entity* sehingga relasi User memiliki atribut yang sama dengan entitas-nya. *Primary key* dari relasi User sama dengan entitas User yaitu atribut username. Langkah selanjutnya adalah mengubah atribut komposit menjadi atribut tunggal. Entitas User memiliki atribut namaLengkap yang komposit sehingga atribut tersebut pada relasi diubah menjadi dua atribut baru, yaitu namaDepan dan namaBelakang.

### Entity Materi

Entitas Materi merupakan generalisasi dari dua buah entitas Diktat Pembelajaran dan Video Pembelajaran. Kita memilih metode pertama dalam mengubah hubungan ISA menjadi model relasi, yaitu tetap mempertahankan atribut-atribut dari induk sehingga relasi Materi memiliki atribut yang sama dengan entitas-nya. *Primary key* dari relasi Materi sama dengan entitas Materi yaitu atribut idMateri. Entitas Materi hanya memiliki *relationship* yang bersifat *many-to-many* ke entitas lain sehingga relasi tidak menyimpan *foreign key* ke relasi lain.

### Entity Diktat Pembelajaran

Entitas Diktat Pembelajaran merupakan spesialisasi dari entitas Materi. Pemetaan hubungan ISA menggunakan metode pertama sehingga relasi Diktat Pembelajaran memiliki *primary key* yang merupakan *foreign key* ke relasi induk, yaitu relasi Materi. Atribut lainnya dari relasi Diktat Pembelajaran sama dengan atribut pada entitas Diktat Pembelajaran.

### Entity Video Pembelajaran

Entitas Video Pembelajaran merupakan spesialisasi dari entitas Materi. Pemetaan hubungan ISA menggunakan metode pertama sehingga relasi Video Pembelajaran memiliki *primary key* yang merupakan *foreign key* ke relasi induk, yaitu relasi Materi. Atribut lainnya dari relasi Video Pembelajaran sama dengan atribut pada entitas Video Pembelajaran.

### Entity Mata Kuliah

Entitas Mata Kuliah merupakan *strong entity* sehingga relasi MataKuliah memiliki atribut yang sama dengan entitas-nya dengan tambahan atribut singkatanJurusan yang menjadi *foreign key* pada relasi ini dan juga merupakan *primary key* pada entitas Jurusan karena entitas Mata Kuliah dan entitas Jurusan terhubung secara *many-to-one*. *Primary key* dari relasi Jurusan sama dengan entitas Jurusan yaitu singkatanJurusan.

### Entity Jurusan

Entitas Jurusan merupakan *strong entity* sehingga relasi Jurusan memiliki atribut yang sama dengan dengan entitas-nya dengan tambahan atribut singkatanFakultas yang menjadi *foreign key* pada relasi ini dan juga merupakan *primary key* pada entitas Fakultas karena entitas Jurusan dan entitas Fakultas terhubung secara *many-to-one*. *Primary key* dari relasi Jurusan sama dengan entitas Jurusan yaitu singkatanJurusan.

### Entity Fakultas

Entitas Fakultas merupakan *strong entity* sehingga relasi Fakultas memiliki atribut yang sama dengan entitas-nya. *Primary key* pada relasi Fakultas sama dengan entitas Fakultas, yaitu singkatanFakultas.

### Relationship antara Latihan Soal dan Pertanyaan

Relationship antara Latihan Soal dan Pertanyaan, yaitu relationship Memiliki bersifat *one-to-many* sehingga relationship ini tidak dipetakan menjadi sebuah relasi.

### Relationship antara Latihan Soal dan Modul

Relationship antara Latihan Soal dan Modul, yaitu relationship Mengandung bersifat *many-to-one* sehingga relationship ini tidak dipetakan menjadi sebuah relasi.

### Relationship antara User dan Latihan Soal

Relationship antara User dan Latihan Soal, yaitu relationship Riwayat Latihan bersifat *many-to-many* sehingga relationship ini dipetakkan menjadi sebuah relasi. Relasi Riwayat Latihan memiliki *primary key* yang komposit dan merupakan *foreign key* ke relasi User dan Latihan Soal. Relationship ini memiliki atribut tambahan waktu yang bersifat *multivalued* sehingga pada relasi ini juga ditambahkan *primary key* lain berupa waktu. Alasan waktu disimpan di dalam relasi Riwayat Latihan Soal langsung dan tidak dibuat relasi baru karena hanya memiliki 1 buah atribut tambahan sehingga jika dibuat relasi baru, relasi Riwayat Latihan Soal hanya menyimpan *primary key*

### Relationship antara User dan Materi

Relationship antara User dan Materi, yaitu relationship Riwayat Materi bersifat *many-to-many* sehingga relationship ini dipetakkan menjadi sebuah relasi. Relasi Riwayat Materi memiliki *primary key* yang komposit dan merupakan *foreign key* ke relasi User dan Materi. Relationship ini memiliki atribut tambahan waktu yang bersifat *multivalued* sehingga pada relasi ini juga ditambahkan *primary key* lain berupa waktu. Alasan waktu dibuat menjadi *primary key* dibandingkan membuat suatu relasi tambahan karena jika dijadikan suatu relasi baru, relasi Riwayat Materi hanya memuat *primary key*

### Relationship antara User dan Mata Kuliah

Relationship antara User dan Mata Kuliah, yaitu relationship Mengikuti bersifat *many-to-many* sehingga relationship ini dipetakan menjadi sebuah relasi. Relasi pesertaKuliah memiliki *primary key* dan juga merupakan *foreign key* ke relasi User dan MataKuliah.

### Relationship antara Mata Kuliah dan Modul

Relationship antara Modul dan Materi, yaitu relationship Terdapat bersifat *many-to-many* sehingga relationship ini dipetakan menjadi sebuah relasi. Relasi MatkulModul memiliki *primary key* dan juga sekaligus *foreign key* ke relasi Modul dan Materi dengan tambahan atribut noUrut.

### Relationship antara Modul dan Materi

Relationship antara Modul dan Materi, yaitu relationship Mengandung bersifat *many-to-many* sehingga relationship ini dipetakan menjadi sebuah relasi. Relasi MateriModul memiliki *primary key* dan juga sekaligus *foreign key* ke relasi Modul dan Materi.

### Relationship antara Mata Kuliah dan Jurusan

Relationship antara Mata Kuliah dan Jurusan, yaitu relationship Memiliki bersifat *one-to-many* sehingga relationship ini tidak dipetakan menjadi sebuah relasi.

### Relationship antara Jurusan dan Fakultas

Relationship antara Jurusan dan Fakultas, yaitu relationship Bagian Dari bersifat *one-to-many* sehingga relationship ini tidak dipetakan menjadi sebuah relasi.

## 3.2. Model Relasional yang Dihasilkan

## 

Gambar 3.2.1 Model Relasional

**BAB 4**

# **Implementasi dan Pengujian**

## 4.1. Insight dan Query dari Basis Data

###### **Tabel Insight Database yang dihasilkan**

| **1** | **INSIGHT**  Menampilkan 5 username pengguna yang paling sering mengakses materi dan latihan soal. Tampilkan juga berapa kali pengguna mengakses materi dan latihan soal sebagai poinAktivitas |
| --- | --- |
| **QUERY**  SELECT username, COUNT(username) AS poinAktivitas  FROM (SELECT username FROM riwayatlatihansoal  UNION ALL  SELECT username FROM riwayatmateri) AS riwayatuser  GROUP BY username  ORDER BY poinAktivitas DESC  LIMIT 5; |
| **SCREENSHOT** |
| **2** | **INSIGHT**  Menampilkan judul materi yang diujikan dengan latihan soal pada suatu modul dengan judul latihannya |
| **QUERY**  SELECT mat.judul, latsol.judul  FROM materi as mat, materimodul as matmod, modul as m, latihansoal as latsol  WHERE mat.idMateri = matmod.idMateri AND matmod.idModul = m.idModul AND latsol.idModul = m.idModul; |
| **SCREENSHOT** |
| **3** | **INSIGHT**  Menampilkan username yang pernah mengerjakan latihan soal yang terdapat pada suatu modul yang berasal dari mata kuliah ‘BIOL5536’ |
| **QUERY**  Select username  From user  Where username in (  select username  From user natural inner join riwayatlatihansoal natural inner join latihansoal natural inner join modul natural inner join modulmatakuliah natural inner join matakuliah  Where kodeMataKuliah like 'BIOL5536'); |
| **SCREENSHOT** |
| **4** | **INSIGHT**  Menampilkan username user yang mengambil mata kuliah yang berasal dari jurusan NURS dan user pernah mengakses materi dan latihan soal setidaknya sekali. Dan juga menampilkan kodeMataKuliah |
| **QUERY**  SELECT username, kodeMataKuliah  FROM user  NATURAL JOIN pesertakuliah  NATURAL JOIN matakuliah  WHERE singkatanJurusan = 'NURS' AND username IN (SELECT username FROM riwayatmateri  INTERSECT  SELECT username FROM riwayatlatihansoal); |
| **SCREENSHOT** |
| **5** | **INSIGHT**  Menampilkan 5 nama mata kuliah yang memiliki konten materi paling banyak, dihitung dari total durasi video pembelajaran ditambah jumlah halaman diktat pembelajaran |
| **QUERY**  WITH kontenmateri AS ((SELECT videopembelajaran.idMateri, COALESCE(diktatpembelajaran.jumlahHalaman, 0) + COALESCE(videopembelajaran.durasi, 0) AS totalmateri FROM videopembelajaran LEFT OUTER JOIN diktatpembelajaran ON videopembelajaran.idMateri = diktatpembelajaran.idMateri) UNION (SELECT diktatpembelajaran.idMateri, COALESCE(diktatpembelajaran.jumlahHalaman, 0) + COALESCE(videopembelajaran.durasi, 0) AS totalmateri FROM videopembelajaran RIGHT OUTER JOIN diktatpembelajaran ON videopembelajaran.idMateri = diktatpembelajaran.idMateri)) SELECT kodeMataKuliah, namaMataKuliah, SUM(totalmateri) AS totalmaterimatkul FROM matakuliah NATURAL JOIN modulmatakuliah NATURAL JOIN materimodul NATURAL JOIN kontenmateri GROUP BY kodeMataKuliah ORDER BY totalmaterimatkul DESC LIMIT 5; |
| **SCREENSHOT** |

# **REFERENSI**

1. <https://edunex.itb.ac.id/>
2. [Tim PengajarIF2140 Semester I TA 2020/2021. Database Design using E-R Model](https://cdn-edunex.itb.ac.id/48686-Database-Parallel-Class/143662-Week-06-SQL/68095-Design-Database-Using-ER-Model/1677186974842_IF2240---SemII_2223---m06-2---Design-using-E-R-Model---part-1.pdf)
3. [Tim PengajarIF2140 Semester I TA 2020/2021. Database Design using E-R Model (p.2)](https://cdn-edunex.itb.ac.id/storages/files/1677646554405_IF2240---SemII_2223---m07-1---Design-using-E-R-Model---part-2.pdf)
4. [Tim PengajarIF2140 Semester I TA 2020/2021. Database Design using E-R Model (p.3)](https://cdn-edunex.itb.ac.id/storages/files/1678768255399_IF2240---SemII_2223---m09-1---Design-using-E-R-Model---part-3.pdf)
5. [Tim PengajarIF2140 Semester I TA 2020/2021. Reducing E-R Diagrams to Relational Schemas](https://cdn-edunex.itb.ac.id/38016-Database-Parallel-Class/71649-Week-10-Reducing-ER-Model-to-Relational-Schema-Relational-Database-Design/39059-Reducing-ER-Model-To-Relational-Schema/1647916372288_IF2240---Reducing-ER-Diagram-to-Relational-Model-slide-Edunex.pdf)

# **LAMPIRAN**

Minutes of Meeting

## 1. Deskripsi Wawancara

###### Tabel Deskripsi Wawancara

| Nomor Kelompok | 1 |
| --- | --- |
| Anggota yang hadir saat wawancara | 1. Nathaniel Evan Gunawan (13516055) 2. Hosea Nathanael Abetnego (13521057) 3. Marcel Ryan Antony (13521127) 4. Haidar Hamda (13521105) 5. Made Debby Almadea Putri (13521153) |
| Tempat, tanggal | Google Meet, 8 April 2023 |
| Waktu | 13.00 |
| Nama Asisten | Alifia Rahmah |

## 2. Hasil Wawancara

###### Tabel Hasil Wawancara

| Pertanyaan | Jawaban |
| --- | --- |
| namaLengkap di user perlu dibuat jadi namaDepan dan namaBelakang atau tidak? | Lebih baik dipisah untuk memenuhi beberapa fungsional lain di sisi *frontend* |
| List jawaban di latihan soal itu termasuk jawaban salah saja atau semuanya? Apa perlu untuk dipisah? | Jawaban itu termasuk semuanya, kalau dipisah antara front end yang sulit untuk mengambil data 2 kali atau backend yang harus update karena akan redundant. Tapi ada cara lain yaitu kunci jawaban dibuat hanya indeks jawaban (a,b,c-nya saja) dan detail dari jawaban itu di simpan di atribut lain |
| Untuk pembuatan db-nya itu perlu berapa banyak data dan apa benar-benar harus data manual? | Sesuai spek, perlu ada minimal 100 record, tapi untuk data-data seperti jurusan dan fakultas tidak perlu sebanyak itu. Untuk data yang ada, random saja tidak masalah. |

###### Gambar Dokumentasi