

IF2240 - Basis Data

Milestone 3 Basis Data

Pemodelan Entity-Relationship

Disusun untuk Memenuhi Tugas Mata Kuliah Basis Data
pada Semester II (dua) Tahun Akademik 2023



Kelompok 5

13519203	R. B. Wishnumurti
13521053	Athif Nirwasito
13521077	Husnia Munzayana
13521089	Kenneth Ezekiel Supranton
13521143	Raynard Tanadi

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung
2023

Daftar Isi

Daftar Isi.....	2
Bab 1 : Deskripsi Sistem.....	3
Bab 2 : ER Model.....	5
Bab 3 : Relational Database Design.....	6
3.1. Skema basis data awal sebelum normalisasi.....	6
3.2. Functional Dependencies.....	6
3.3. Identifikasi bentuk normal dari setiap relasi.....	8
3.4. Langkah-langkah normalisasi.....	10
3.5. Skema basis data hasil normalisasi.....	12
Bab 4 : Business Rule.....	14
4.1. Query-query untuk menerapkan business rule pada basis data.....	14
1. View statistik semua seleksi.....	14
2. Dynamic Update.....	15
3. Constraint.....	18
4.2. Query-query untuk menunjukkan penerapan business rule pada basis data.....	18
1. View statistik semua seleksi.....	18
2. Dynamic Update.....	18
3. Constraint.....	18
Bab 5 : Kesimpulan.....	19
Referensi.....	20

Bab 1 : Deskripsi Sistem

LPDP (Lembaga Pengelola Dana Pendidikan) adalah sebuah lembaga pemerintah yang bertanggung jawab atas pengelolaan dana pendidikan untuk mahasiswa Indonesia yang ingin melanjutkan studi S2 atau S3 di dalam negeri maupun luar negeri. Kepala LPDP ingin membuat sistem basis data yang memudahkan pengelolaan data pendaftar seleksi beasiswa LPDP. Sebagai mahasiswa informatika, Anda diminta untuk membuat desain basis data untuk sistem pendaftaran seleksi LPDP.

Setiap tahunnya, LPDP membuka pendaftaran bagi mahasiswa yang ingin mendapatkan beasiswa untuk melanjutkan studi S2 atau S3. Ketika melakukan pendaftaran, mahasiswa diwajibkan untuk melengkapi berbagai macam informasi pribadi seperti nama lengkap, jenis kelamin, tempat dan tanggal lahir, alamat rumah, nomor telepon, dan alamat email. Selain itu, mahasiswa juga harus melampirkan berbagai macam dokumen seperti ijazah S1, transkrip nilai, dan surat rekomendasi dari dosen atau supervisor penelitian.

Setiap mahasiswa yang mendaftar akan mengikuti tahapan seleksi yang terdiri dari beberapa tahap, seperti seleksi administrasi, seleksi wawancara, seleksi psikotes, dan seleksi kesehatan. Setiap tahap seleksi memiliki tanggal pelaksanaan yang berbeda-beda dan berbeda pula untuk seluruh mahasiswa, kecuali seleksi administrasi yang hanya dilakukan sekali di awal periode dan sama untuk seluruh mahasiswa. Tidak semua mahasiswa akan mengikuti seluruh rangkaian seleksi, kecuali tahap seleksi administrasi. Maka, perlu dicatat passing grade untuk setiap tahap seleksi. Untuk tahap seleksi administrasi, mahasiswa yang lulus adalah mahasiswa yang dokumennya lengkap.

Setelah pendaftaran administrasi dinyatakan lengkap, calon penerima beasiswa akan mengikuti tahap seleksi wawancara, psikotes, dan kesehatan. Untuk tahap seleksi wawancara, setiap calon penerima beasiswa akan diwawancara dengan jenis tertentu (online/offline) dengan sesi wawancara tertentu. Untuk setiap jenis wawancara dan sesi wawancara, terdapat panitia yang berperan sebagai pewawancara. Pewawancara LPDP memiliki jabatan berbeda-beda yang perlu dicatat di sistem.

Pada tahap seleksi psikotes, setiap calon penerima beasiswa harus mengikuti tes psikotes. Tes psikotes dilaksanakan di sesi dan ruangan tertentu, tetapi seluruh sesi dan ruangan dilaksanakan di gedung yang sama. Untuk setiap sesi dan ruangan, terdapat seorang pengawas yang memiliki divisi yang berbeda-beda.

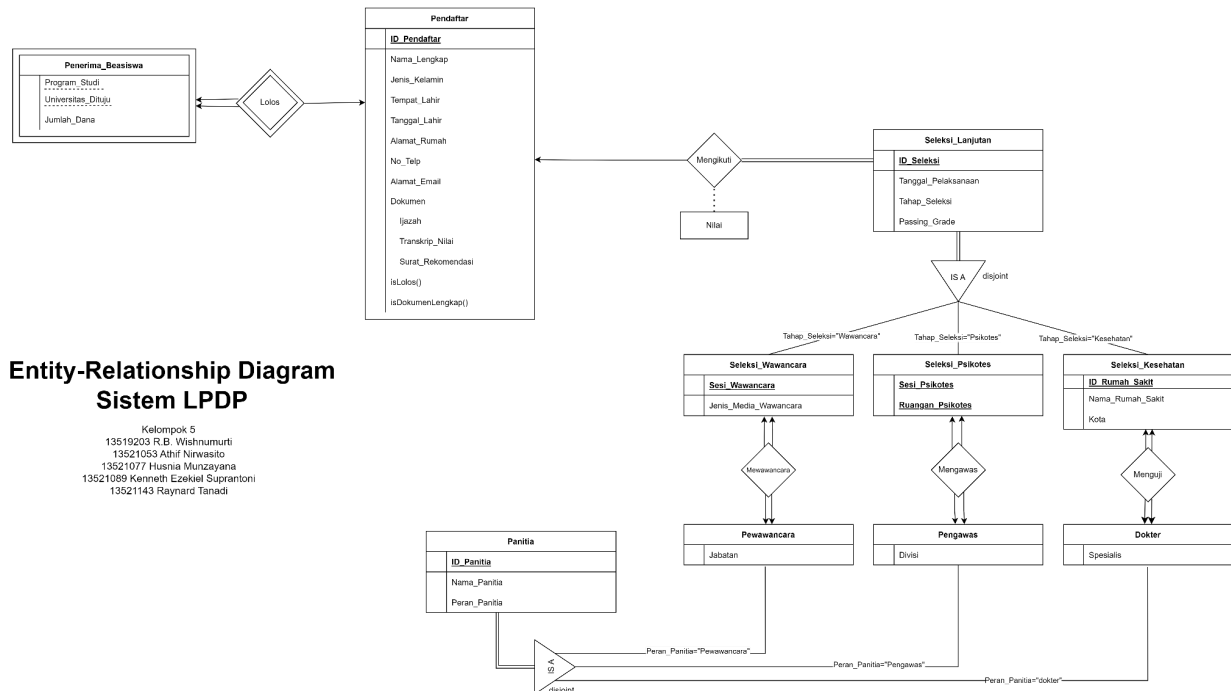
Pada tahap seleksi kesehatan, setiap calon penerima beasiswa harus melakukan pemeriksaan kesehatan di rumah sakit tertentu yang berada di kota tertentu. Untuk setiap rumah sakit, terdapat panitia yang merupakan dokter dengan spesialis tertentu. Nilai yang didapat dari ketiga jenis tes tersebut akan dicatat untuk dievaluasi apakah mahasiswa yang bersangkutan lulus sebuah tahap seleksi. Setiap panitia, baik itu pewawancara, dokter, maupun pengawas perlu dicatat nama lengkapnya.

Setelah tahap seleksi selesai dilaksanakan, mahasiswa yang lulus seluruh tahap seleksi akan mendapatkan informasi mengenai pemberian beasiswa LPDP melalui email atau telepon. Selain itu, LPDP juga akan mencatat informasi mengenai profil mahasiswa yang mendapatkan beasiswa, seperti program studi yang diambil, universitas yang dituju, dan jumlah dana yang diberikan.

Bab 2 : ER Model

Berikut adalah ER Model yang dibuat berdasarkan deskripsi sistem, dengan satu revisi:

Gambar 3.1 ER Model



Terdapat beberapa asumsi yang digunakan dalam ER Model di atas, yaitu:

1. Setiap pendaftar memiliki satu alamat rumah, nomor telepon, serta alamat *email*.
2. Setiap panitia pasti memiliki satu peran, baik sebagai pewawancara, pengawas, maupun dokter.
3. Setiap waktu dan ruang seleksi pasti diawasi oleh seorang panitia.
4. Setiap pewawancara, pengawas, serta dokter pasti mengambil peran pada satu sesi seleksi.
5. Setiap pendaftar, dokter, serta rumah sakit teridentifikasi secara unik berdasarkan ID.
6. Setiap penerima mahasiswa hanya dapat diterima pada satu program studi serta universitas dan mendapat satu besaran dana.

Revisi yang terdapat pada ER Model di atas adalah pada bagian entitas seleksi_lanjutan, dimana karena halangan implementasi, PK dijadikan ID_Seleksi

Bab 3 : Relational Database Design

3.1. Skema basis data awal sebelum normalisasi.

Penerima_Beasiswa	= { <u>ID_Pendaftar</u> , <u>Program Studi</u> , <u>Universitas Dituju</u> , Jumlah_Dana}
Pendaftar	= { <u>ID_Pendaftar</u> , Nama_Lengkap, Jenis_Kelamin, Tempat_Lahir, Tanggal_Lahir, Alamat_Rumah, No_Telp, Alamat_EMail, Ijazah, Transkrip_Nilai, Surat_Rekomendasi}
Seleksi_Lanjutan	= { <u>ID_Pendaftar</u> , <u>ID Seleksi</u> , Nilai, Passing_Grade}
Seleksi_Wawancara	= { <u>ID Seleksi</u> , <u>Sesi Wawancara</u> , Tanggal_Pelaksanaan, Jenis_Media_Wawancara, ID_Pewawancara, Nilai}
Seleksi_Psikotest	= { <u>ID Seleksi</u> , <u>Sesi Psikotest</u> , <u>Ruangan Psikotest</u> , Tanggal_Pelaksanaan, ID_Pengawas, Nilai}
Seleksi_Kesehatan	= { <u>ID Seleksi</u> , <u>ID Rumah Sakit</u> , Tanggal_Pelaksanaan, ID_Dokter, Nama_Rumah_Sakit, Kota, Nilai}
Panitia	= { <u>ID Panitia</u> , Nama_Panitia}
Pewawancara	= { <u>ID Panitia</u> , Jabatan}
Pengawas	= { <u>ID Panitia</u> , Divisi}
Dokter	= { <u>ID Panitia</u> , Spesialis}

3.2. Functional Dependencies

Functional Dependencies yang diidentifikasi, serta penjelasan mengapa FD tersebut berlaku.

1. Penerima Beasiswa

FD1: ID_Pendaftar, Program_Studi, Universitas_Dituju → Jumlah_Dana

Alasan: ID_Pendaftar dengan pilihan Program_Studi dan Universitas_Dituju yang dipilihnya hanya akan diberikan satu beasiswa dengan Jumlah_Dana tertentu.

2. Pendaftar

FD1: ID_Pendaftar → Nama_Lengkap, Jenis_Kelamin, Tempat_Lahir, Tanggal_Lahir,
Alamat_Rumah, Ijazah, Surat_Rekomendasi, Transkrip_Nilai

Alasan: Identitas seorang pendaftar pasti unik, maka satu ID_Pendaftar pasti dapat dan hanya menentukan satu Nama_Lengkap, Jenis_Kelamin, Tempat_Lahir, Tanggal_Lahir, Alamat_Rumah, Ijazah, Surat_Rekomendasi, dan Transkrip_Nilai.

3. Seleksi_Lanjutan

FD1: ID_Pendaftar, ID_Seleksi → Nilai

Laporan Milestone 3

Tugas Besar IF2240

K01 - Kelompok 5

Page 6

Alasan: ID_Pendaftar untuk seleksi tertentu disimpan sebuah nilai dari hasil seleksi.

FD2: ID_Seleksi->Passing_Grade

Alasan: ID_Seleksi menentukan jenis seleksi antara seleksi wawancara, seleksi psikotest, atau seleksi kesehatan. Oleh karena itu, ID_Seleksi dapat menentukan passing_grade dari seleksi tersebut.

FD3: ID_Seleksi->Nilai

Alasan: Setiap ID_Seleksi terdapat sebuah nilai yang menyimpan hasil dari seleksi tersebut.

4. Seleksi_Wawancara

FD1: ID_Seleksi->Tanggal_Pelaksanaan

Alasan: ID_Seleksi dapat digunakan untuk menentukan kapan suatu seleksi untuk orang tertentu akan dilaksanakan.

FD2: ID_Seleksi, Sesi_Wawancara → Jenis_Media_Wawancara, ID_Pewawancara

Alasan: Untuk seseorang yang mengikuti sesi tertentu pada seleksi wawancara, akan diwawancarai menggunakan satu jenis media wawancara oleh seorang pewawancara dan didapatkan suatu nilai dari wawancara tersebut.

FD3: ID_Seleksi → Nilai

Alasan: Setiap ID_Seleksi terdapat sebuah nilai yang menyimpan hasil dari seleksi tersebut.

5. Seleksi_Psikotest

FD1: ID_Seleksi → Tanggal_Pelaksanaan

Alasan: ID_Seleksi dapat digunakan untuk menentukan kapan suatu seleksi untuk orang tertentu akan dilaksanakan.

FD2: Sesi_Psikotest, Ruangan_Psikotest → ID_Pengawas

Alasan: Untuk setiap sesi dan ruangan, terdapat seorang pengawas yang memiliki divisi yang berbeda-beda.

FD3: ID_Seleksi → Nilai

Alasan: Setiap ID_Seleksi terdapat sebuah nilai yang menyimpan hasil dari seleksi tersebut.

6. Seleksi_Kesehatan

FD1: ID_Rumah_Sakit -> Nama_Rumah_Sakit, Kota

Alasan: Setiap ID_Rumah_Sakit mereferensikan rumah sakit yang unik sehingga ID_Rumah_Sakit -> Nama_Rumah_Sakit. Sebuah rumah sakit pasti terletak di sebuah kota sehingga ID_Rumah_Sakit -> Kota

FD2: ID_Seleksi → ID_Dokter, Tanggal_Pelaksanaan

Alasan: Setiap orang akan diperiksa oleh dokter tertentu pada tanggal tertentu.

FD3: ID_Seleksi → Nilai

Alasan: Setiap ID_Seleksi terdapat sebuah nilai yang menyimpan hasil dari seleksi tersebut.

7. Pewawancara

FD1: ID_Pewawancara -> Jabatan

Alasan: Asumsi setiap pewawancara hanya bisa memegang satu jabatan.

8. Pengawas

FD1: ID_Pengawas -> Divisi

Alasan: Asumsi setiap pengawas hanya bisa berasal dari satu divisi.

9. Dokter

FD1: ID_Dokter -> Spesialis

Alasan: Dokter umumnya memiliki satu spesialis. Jika ada dokter yang mempunyai dua spesialis maka kedua spesialis digabung dalam satu entri.

10. Panitia

FD1: ID_Panitia->Nama_Panitia

Alasan: Setiap ID_Panitia dipegang oleh satu panitia dengan nama tertentu.

3.3. Identifikasi bentuk normal dari setiap relasi.

Penerima_Beasiswa

Candidate key : {ID_Pendaftar, Program_Studi, Universitas_Dituju}

Functional Dependency :

FD1 : ID_Pendaftar, Program_Studi, Universitas_Dituju → Jumlah_Dana

Bentuk Normal :

- Relasi Penerima_Beasiswa merupakan BCNF karena left-hand side FD1 merupakan superkey.

Pendaftar

Candidate key: {ID_Pendaftar}

Functional Dependency :

FD1 : ID_Pendaftar → Nama_Lengkap, Jenis_Kelamin, Tempat_Lahir, Tanggal_Lahir, Alamat_Rumah, Ijazah, Surat_Rekomendasi, Transkrip_Nilai

Bentuk Normal :

- Relasi Pendaftar merupakan BCNF karena left-hand side functional dependency pada relasi tersebut merupakan superkey.

Seleksi_Lanjutan

Candidate key: {ID_Pendaftar, ID_Seleksi}

Functional Dependency :

FD1 : ID_Pendaftar, ID_Seleksi → Nilai

FD2 : ID_Seleksi → Passing_Grade

FD3 : ID_Seleksi -> Nilai

- ID_Pendaftar extraneous pada FD1, sehingga ketiga FD tersebut dapat diubah menjadi

FD1 : ID_Seleksi → Nilai, Passing_Grade

Bentuk Normal :

- Relasi Seleksi_Lanjutan bukan BCNF karena pada left-hand side pada FD1 yaitu ID_Seleksi bukan superkey
- Relasi Seleksi_Lanjutan bukan 3NF karena $\beta - \alpha$ pada FD1 yaitu {Nilai, Passing_Grade} tidak ada di candidate key.
- Relasi Seleksi_Lanjutan bukan 2NF karena left-hand side FD1 yaitu {ID_Seleksi} partially

- dependent terhadap candidate key.
- Maka, bentuk normal relasi Seleksi_Lanjutan adalah 1NF.

Seleksi_Wawancara

Candidate key: {ID_Seleksi, Sesi_Wawancara}

Functional Dependency :

FD1 : ID_Seleksi → Tanggal_Pelaksanaan

FD2 : ID_Seleksi, Sesi_Wawancara → Jenis_Media_Wawancara, ID_Pewawancara

FD3 : ID_Seleksi → Nilai

- FD1 dan FD3 dapat diunion menjadi ID_Seleksi → Tanggal_Pelaksanaan, Nilai
- Maka *canonical cover* untuk Seleksi_Wawancara adalah
FD1 : ID_Seleksi → Tanggal_Pelaksanaan, Nilai
FD2 : ID_Seleksi, Sesi_Wawancara → Jenis_Media_Wawancara,
ID_Pewawancara

Bentuk Normal :

- Relasi Seleksi_Wawancara bukan BCNF karena pada left-hand side pada FD1 bukan superkey
- Relasi Seleksi_Wawancara bukan 3NF karena contohnya pada $\beta - \alpha$ pada FD1 yaitu {Tanggal_Pelaksanaan, Nilai} tidak ada di candidate key.
- Relasi Seleksi_Wawancara bukan 2NF karena left-hand side pada FD1 yaitu {ID_Seleksi} partially dependent dengan candidate key.
- Maka, bentuk normal relasi Seleksi_Wawancara adalah 1NF

Seleksi_Psikotest

Candidate key: {ID_Seleksi, Sesi_Psikotest, Ruangan_Psikotest}

Functional Dependency :

FD1 : ID_Seleksi → Tanggal_Pelaksanaan

FD2 : ID_Seleksi, Sesi_Psikotest, Ruangan_Psikotest → ID_Pengawas

FD3 : ID_Seleksi → Nilai

- FD1 dan FD3 dapat diunion menjadi ID_Seleksi → Tanggal_Pelaksanaan, Nilai
- Maka *canonical cover* untuk Seleksi_Psikotest adalah
FD1 : ID_Seleksi → Tanggal_Pelaksanaan, Nilai
FD2 : ID_Seleksi, Sesi_Psikotest, Ruangan_Psikotest → ID_Pengawas

Bentuk Normal :

- Relasi Seleksi_Psikotest bukan BCNF karena pada left-hand side pada FD1 bukan superkey
- Relasi Seleksi_Psikotest bukan 3NF karena pada FD1 yaitu {Tanggal_Pelaksanaan, Nilai} dan pada FD2 yaitu {ID_Pengawas} tidak ada di candidate key
- Relasi Seleksi_Psikotest bukan 2NF karena left-hand side FD1 yaitu {ID_Seleksi} dan left-hand side partially dependent dengan candidate key.
- Maka, bentuk normal relasi Seleksi_Psikotest adalah 1NF

Seleksi_Kesehatan

Candidate key: {ID_Seleksi, ID_Rumah_Sakit}

Functional Dependency :

FD1 : ID_Seleksi → Tanggal_Pelaksanaan, ID_Dokter

FD2 : ID_Rumah_Sakit → Nama_Rumah_Sakit, Kota

FD3 : ID_Seleksi → Nilai

- FD1 dan FD3 dapat diunion menjadi ID_Seleksi → Tanggal_Pelaksanaan, Nilai

Laporan Milestone 3

Tugas Besar IF2240

K01 - Kelompok 5

Page 9

- Maka *canonical cover* untuk Seleksi_Kesehatan adalah
FD1 : ID_Seleksi \rightarrow Tanggal_Pelaksanaan, Nilai, ID_Dokter
FD2 : ID_Rumah_Sakit \rightarrow Nama_Rumah_Sakit, Kota

Bentuk Normal :

- o Relasi Seleksi_Kesehatan bukan BCNF karena pada left-hand side pada FD1 dan FD2 bukan superkey
- o Relasi Seleksi_Kesehatan bukan 3NF karena contohnya $\beta - \alpha$ pada FD2 yaitu {Nama_Rumah_Sakit, Kota} tidak ada di candidate key.
- o Relasi Seleksi_Kesehatan bukan 2NF karena left-hand side FD2 yaitu {ID_Rumah_Sakit} serta left-hand side pada FD1 yaitu {ID_Seleksi} partially dependent dengan candidate key.
- o Maka, bentuk normal relasi Seleksi_Psikotest adalah 1NF.

Panitia

Candidate key: {ID_Panitia}

Functional Dependency :

FD1 : ID_Panitia \rightarrow Nama_Panitia

Bentuk Normal :

- o Relasi Panitia merupakan BCNF karena left-hand side pada FD1 yaitu {ID_Panitia} merupakan superkey.

Pewawancara

Candidate key: {ID_Panitia}

Functional Dependency :

FD1 : ID_Panitia \rightarrow Jabatan

Bentuk Normal :

- o Relasi Pewawancara merupakan BCNF karena left-hand side pada FD1 yaitu {ID_Panitia} merupakan superkey.

Pengawas

Candidate key: {ID_Panitia}

Functional Dependency :

FD1 : ID_Panitia \rightarrow Divisi

Bentuk Normal :

- o Relasi Pengawas merupakan BCNF karena left-hand side pada FD1 yaitu {ID_Panitia} merupakan superkey

Dokter

Candidate key: {ID_Panitia}

Functional Dependency :

FD1: ID_Panitia \rightarrow Spesialis

Bentuk Normal :

- o Relasi Dokter merupakan BCNF karena left-hand side pada FD1, yaitu {ID_Panitia} merupakan superkey

3.4. Langkah-langkah normalisasi

Secara umum, normalisasi atau dekomposisi relasi ke bentuk BCNF dilakukan dengan langkah berikut:

- a. Pastikan functional dependency sudah minimum atau sudah dalam bentuk *canonical cover*
- b. Misal $\alpha \rightarrow \beta$ adalah functional dependency yang tidak trivial dan menyebabkan relasi R tidak BCNF.
- c. Dekomposisi relasi R menjadi :

$$R1 = (\alpha, \beta)$$

$$R2 = (R - \beta)$$
- d. Lakukan langkah a dan b secara berulang hingga relasi hasil dekomposisi bersifat BCNF

Dengan prosedur tersebut, dilakukan normalisasi terhadap relasi dengan bentuk 3NF, 2NF, maupun 1NF ke dalam bentuk relasi BCNF pada skema relasi penerimaan beasiswa LPDP dengan langkah berikut:

- a. Pada relasi Seleksi_Lanjutan = {ID_Pendaftar, ID_Seleksi, Passing_Grade, Nilai}, dengan FD sebagai berikut :

FD1 : $ID_Seleksi \rightarrow Nilai, Passing_Grade$
 Candidate Key = {ID_Pendaftar, ID_Seleksi}

 - o FD1 menyebabkan Seleksi_Lanjutan tidak BCNF, sehingga dilakukan dekomposisi menjadi :

Seleksi_Lanjutan = {ID_Pendaftar, ID_Seleksi}
 Nilai_Seleksi = {ID_Seleksi, Nilai, Passing_Grade}
- b. Pada relasi Seleksi_Wawancara = {ID_Seleksi, Sesi_Wawancara, Tanggal_Pelaksanaan, Jenis_Media_Wawancara, ID_Pewawancara, Nilai}, dengan FD sebagai berikut :

FD1 : $ID_Seleksi \rightarrow Tanggal_Pelaksanaan, Nilai$
 FD2 : $ID_Seleksi, Sesi_Wawancara \rightarrow Jenis_Media_Wawancara, ID_Pewawancara$
 Candidate Key = {ID_Seleksi, Sesi_Wawancara}

 - o FD1 menyebabkan Seleksi_Wawancara tidak BCNF, sehingga dilakukan dekomposisi menjadi :

Pelaksanaan_Seleksi = {ID_Seleksi, Tanggal_Pelaksanaan, Nilai}
 Seleksi_Wawancara = {ID_Seleksi, Sesi_Wawancara, Jenis_Media_Wawancara, ID_Pewawancara}
- c. Pada relasi Seleksi_Psikotest = {ID_Seleksi, Sesi_Psikotest, Ruangan_Psikotest, Tanggal_Pelaksanaan, ID_Pengawas}, dengan FD sebagai berikut :

FD1 : $ID_Seleksi \rightarrow Tanggal_Pelaksanaan, Nilai$
 FD2 : $ID_Seleksi, Sesi_Psikotest, Ruangan_Psikotest \rightarrow ID_Pengawas$
 Candidate Key = {ID_Seleksi, Sesi_Psikotest, Ruangan_Psikotest}

 - o FD1 menyebabkan Seleksi_Psikotest tidak BCNF, sehingga dilakukan dekomposisi menjadi :

Pelaksanaan_Seleksi = {ID_Seleksi, Tanggal_Pelaksanaan, Nilai}

Seleksi_Psikotest = {ID_Seleksi, Sesi_Psikotest, Ruangan_Psikotest,
ID_Pengawas}

- d. Pada relasi Seleksi_Kesehatan = {ID_Seleksi, ID_Rumah_Sakit,
Tanggal_Pelaksanaan, Nama_Rumah_Sakit, Kota, ID_Dokter, Nilai}, dengan FD
sebagai berikut :

FD1 : ID_Seleksi → Tanggal_Pelaksanaan, Nilai, ID_Dokter

FD2 : ID_Rumah_Sakit → Nama_Rumah_Sakit, Kota

Candidate Key = {ID_Seleksi, ID_Rumah_Sakit}

- o FD1 menyebabkan Seleksi_Kesehatan tidak BCNF, sehingga dilakukan dekomposisi menjadi :

Pelaksanaan_Seleksi_Kesehatan = {ID_Seleksi, Tanggal_Pelaksanaan, Nilai,
ID_Dokter}

Seleksi_Kesehatan = {ID_Seleksi, ID_Rumah_Sakit, Nama_Rumah_Sakit,
Kota}

- o FD2 menyebabkan Seleksi_Kesehatan tidak BCNF, sehingga dilakukan dekomposisi menjadi :

Pelaksanaan_Seleksi_Kesehatan = {ID_Seleksi, Tanggal_Pelaksanaan, Nilai,
ID_Dokter}

Seleksi_Kesehatan = {ID_Seleksi, ID_Rumah_Sakit}

Rumah_Sakit = {ID_Rumah_Sakit, Nama_Rumah_Sakit, Kota}

Normalisasi tersebut telah menghasilkan relasi yang BCNF, *lossless join*, dan *dependency preservation*.

3.5. Skema basis data hasil normalisasi

Dikarenakan sebuah *circular update* saat mengimplementasikan *business rule*, atribut nilai diturunkan dari relasi seleksi_lanjutan ke masing-masing seleksi seperti seleksi_wawancara, seleksi_psikotest, dan seleksi_kesehatan

Penerima_Beasiswa = {ID_Pendaftar, Program_Studi, Universitas_Dituju,
Jumlah_Dana}

Pendaftar = {ID_Pendaftar, Nama_Lengkap, Jenis_Kelamin,
Tempat_Lahir, Tanggal_Lahir, Alamat_Rumah,
No_Telp, Alamat_Email, Ijazah, Transkrip_Nilai,
Surat_Rekomendasi}

Seleksi_Lanjutan = {ID_Pendaftar, ID_Seleksi}

Nilai_Seleksi = {ID_Seleksi, Nilai, Passing_Grade}

Pelaksanaan_Seleksi = {ID_Seleksi, Tanggal_Pelaksanaan, Nilai}

Seleksi_Wawancara = {ID_Seleksi, Sesi_Wawancara,
Jenis_Media_Wawancara, ID_Pewawancara}

Seleksi_Psikotest = {ID_Seleksi, Sesi_Psikotest, Ruangan_Psikotest,
ID_Pengawas}

Seleksi_Kesehatan	= { <u>ID_Seleksi</u> , <u>ID_Rumah_Sakit</u> }
Pelaksanaan_Kesehatan	= { <u>ID_Seleksi</u> , Tanggal_Pelaksanaan, Nilai, ID_Dokter}
Rumah_Sakit	= { <u>ID_Rumah_Sakit</u> , Nama_Rumah_Sakit, Kota}
Panitia	= { <u>ID_Panitia</u> , Nama_Panitia}
Pewawancara	= { <u>ID_Panitia</u> , Jabatan}
Pengawas	= { <u>ID_Panitia</u> , Divisi}
Dokter	= { <u>ID_Panitia</u> , Spesialis}

Bab 4 : Business Rule

4.1. *Query-query* untuk menerapkan business rule pada basis data.

1. View statistik semua seleksi

```
CREATE VIEW stat_admin AS
SELECT LolosAdmin
FROM (SELECT COUNT(*) AS LolosAdmin FROM seleksi_wawancara) AS Admin;

CREATE VIEW stat_wawancara AS
SELECT LolosWawancara, nilai_rata, nilai_min, nilai_max
FROM
    (SELECT COUNT(*) AS LolosWawancara FROM seleksi_psikotest) AS Wawancara,
    (SELECT AVG(Nilai) AS nilai_rata, MIN(Nilai) AS nilai_min, MAX(Nilai) as nilai_max
FROM seleksi_wawancara) AS Penilaian;

CREATE VIEW stat_psikotest AS
SELECT LolosPsikotest, nilai_rata, nilai_min, nilai_max
FROM
    (SELECT COUNT(*) AS LolosPsikotest FROM seleksi_kesehatan) AS Psikotest,
    (SELECT AVG(Nilai) AS nilai_rata, MIN(Nilai) AS nilai_min, MAX(Nilai) as nilai_max
FROM seleksi_psikotest) AS Penilaian;

CREATE VIEW stat_kesehatan AS
SELECT LolosKesehatan, nilai_rata, nilai_min, nilai_max
FROM
    (SELECT COUNT(*) AS LolosKesehatan FROM penerima_beasiswa) AS Kesehatan,
    (SELECT AVG(Nilai) AS nilai_rata, MIN(Nilai) AS nilai_min, MAX(Nilai) as nilai_max
FROM seleksi_kesehatan) AS Penilaian;
```

2. Dynamic Update

```
DELIMITER $$
CREATE TRIGGER update_passing_grade_insert_wawancara AFTER INSERT ON seleksi_wawancara
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_wawancara p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_wawancara);
```

```

END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER update_passing_grade_insert_psikotest AFTER INSERT ON seleksi_psikotest
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_psikotest p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_psikotest);
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER update_passing_grade_insert_kesehatan AFTER INSERT ON seleksi_kesehatan
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_kesehatan p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_kesehatan);
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER update_passing_grade_delete_wawancara AFTER DELETE ON seleksi_wawancara
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_wawancara p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_wawancara) OR ID_Seleksi = OLD.ID_Seleksi;
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER update_passing_grade_delete_psikotest AFTER DELETE ON seleksi_psikotest
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_psikotest p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_psikotest) OR ID_Seleksi = OLD.ID_Seleksi;
END $$
DELIMITER ;

```

```

DELIMITER $$
CREATE TRIGGER update_passing_grade_delete_kesehatan AFTER DELETE ON seleksi_kesehatan
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_kesehatan p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_kesehatan) OR ID_Seleksi = OLD.ID_Seleksi;
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER update_passing_grade_update_wawancara AFTER UPDATE ON seleksi_wawancara
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_wawancara p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_wawancara);

END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER update_passing_grade_update_psikotest AFTER UPDATE ON seleksi_psikotest
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_psikotest p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_psikotest);
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER update_passing_grade_update_kesehatan AFTER UPDATE ON seleksi_kesehatan
FOR EACH ROW
BEGIN
UPDATE seleksi_lanjutan SET passing_grade = (SELECT AVG(p.Nilai) FROM seleksi_kesehatan p
WHERE p.ID_Seleksi = ID_Seleksi) WHERE ID_Seleksi IN ( SELECT ID_Seleksi FROM
seleksi_kesehatan);
END $$
DELIMITER ;

```


3. Constraint

```
DELIMITER //
```

```
CREATE TRIGGER check_id_pendaftar
```

```
BEFORE INSERT ON seleksi_lanjutan
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE id_pendaftar_count INT;
```

```
    SELECT COUNT(*) INTO id_pendaftar_count FROM seleksi_lanjutan WHERE ID_Pendaftar =
```

```
NEW.ID_Pendaftar;
```

```
    IF id_pendaftar_count >= 3 THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Maximum limit of 3 reached for
```

```
ID_pendaftar';
```

```
    END IF;
```

```
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE TRIGGER check_urutan_seleksi
```

```
BEFORE INSERT ON seleksi_lanjutan
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE id_count_wawancara INT;
```

```
    DECLARE id_count_psikotes INT;
```

```
    SELECT COUNT(*) INTO id_count_wawancara FROM seleksi_lanjutan WHERE ID_Pendaftar =
```

```
NEW.ID_Pendaftar AND ID_Seleksi LIKE '%C%';
```

```
    SELECT COUNT(*) INTO id_count_psikotes FROM seleksi_lanjutan WHERE ID_Pendaftar =
```

```
NEW.ID_Pendaftar AND ID_Seleksi LIKE '%D%';
```

```
    IF NEW.ID_Seleksi LIKE '%D%' AND id_count_wawancara = 0 THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tidak bisa insert, seleksi psikotest
```

```
harus setelah seleksi wawancara';
```

```
    END IF;
```

```
    IF NEW.ID_Seleksi LIKE '%E%' AND (id_count_wawancara = 0 OR id_count_psikotes = 0)
```

```
THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tidak bisa melakukan insertion,
```

```
seleksi kesehatan harus setelah seleksi wawancara dan seleksi psikotest';
```

```
END IF;  
END //  
  
DELIMITER ;
```

4.2. Query-query untuk menunjukkan penerapan business rule pada basis data.

1. View statistik semua seleksi

```
MariaDB [sistemfakerfix]> select * from stat_admin;  
+-----+  
| LolosAdmin |  
+-----+  
|          300 |  
+-----+  
1 row in set (0.017 sec)
```

```
MariaDB [sistemfakerfix]> select * from stat_wawancara;  
+-----+-----+-----+-----+  
| LolosWawancara | nilai_rata | nilai_min | nilai_max |  
+-----+-----+-----+-----+  
|           238 |   87.3167 |         75 |        100 |  
+-----+-----+-----+-----+  
1 row in set (0.019 sec)
```

```
MariaDB [sistemfakerfix]> select * from stat_psikotest;  
+-----+-----+-----+-----+  
| LolosPsikotest | nilai_rata | nilai_min | nilai_max |  
+-----+-----+-----+-----+  
|           130 |   84.3235 |         70 |        100 |  
+-----+-----+-----+-----+  
1 row in set (0.015 sec)
```

```
MariaDB [sistemfakerfix]> select * from stat_kesehatan;
+-----+-----+-----+-----+
| LolosKesehatan | nilai_rata | nilai_min | nilai_max |
+-----+-----+-----+-----+
|           125 |   84.5308 |         70 |        100 |
+-----+-----+-----+-----+
1 row in set (0.013 sec)
```

2. Dynamic Update

Query dan Hasil	Keterangan
<pre>MariaDB [sistem_lpd] > select * from seleksi_lanjutan; +-----+-----+-----+ ID_Pendaftar ID_Seleksi Passing_Grade +-----+-----+-----+ A001041 C233746 80 A005605 C919219 80 A005605 D888494 80 A005754 C487924 80 A010348 C287108 80 A010348 D387131 80 A010348 E796971 80 A013139 C762857 80 +-----+-----+-----+ +-----+-----+-----+ A985265 D473382 80 A990514 C248305 80 A990514 D910019 80 A990514 E082945 80 A994194 C712830 80 A994194 D756507 80 A994194 E657985 80 A994871 C277746 80 A994871 D272183 80 A994871 E761170 80 A998426 C696018 80 A998426 D040378 80 A998753 C302116 80 A998753 D004033 80 A999775 C804902 80 A999775 D360612 80 +-----+-----+-----+ 679 rows in set (0.001 sec)</pre>	<p>State awal. ID_Seleksi yang diawali huruf C adalah seleksi_wawancara, D adalah seleksi_psikotest, dan E adalah seleksi_kesehatan.</p>

```
MariaDB [sistem_lpd]> INSERT INTO pendaftar (ID_Pendaftar, Nama_Lengkap, Jenis_Kelamin, Tempat_Lahir, Tanggal_Lahir, Alamat_Rumah, No_Telp, Alamat_Email, Ijazah, Transkrip_Nilai, Surat_Rekomendasi) VALUES ('A999999', 'Tina Jesi', 'Perempuan', 'Ternate', '1989-07-20', 'Gang Surapati No. 7 Tarakan, Papua Barat 08119', '+6263718069283', 'tina6emj@outlook.com', '266037-21-9/77212-092-2821', '3.64', '436108-36-9/33139-535-9981');
Query OK, 1 row affected (0.007 sec)
```

```
MariaDB [sistem_lpd]> INSERT INTO seleksi_lanjutan (ID_Pendaftar, ID_Seleksi, Passing_Grade) VALUES ('A999999', 'C999999', '80');
Query OK, 1 row affected (0.007 sec)
```

```
MariaDB [sistem_lpd]> INSERT INTO seleksi_lanjutan (ID_Pendaftar, ID_Seleksi, Passing_Grade) VALUES ('A999999', 'D999999', '80');
Query OK, 1 row affected (0.007 sec)
```

```
MariaDB [sistem_lpd]> INSERT INTO seleksi_lanjutan (ID_Pendaftar, ID_Seleksi, Passing_Grade) VALUES ('A999999', 'E999999', '80');
Query OK, 1 row affected (0.006 sec)
```

```
MariaDB [sistem_lpd]> INSERT INTO seleksi_wawancara (ID_Seleksi, Tanggal_Pelaksanaan, Sesi_Wawancara, Jenis_Media_Wawancara, ID_Pewawancara, NILAI) VALUES ('C999999', '2023-04-11', '3', 'Daring', 'B37464', '10000');
Query OK, 1 row affected (0.016 sec)
```

```
MariaDB [sistem_lpd]> select * from seleksi_lanjutan;
```

ID_Pendaftar	ID_Seleksi	Passing_Grade
A001041	C233746	121
A005605	C919219	121
A005605	D888494	80
A005754	C487924	121
A010348	C287108	121
A010348	D387131	80
A010348	E796971	80
A013139	C762857	121
A013139	D276112	80
A013139	E852614	80

A990514	C248305	121
A990514	D910019	80
A990514	E082945	80
A994194	C712830	121
A994194	D756507	80
A994194	E657985	80
A994871	C277746	121
A994871	D272183	80
A994871	E761170	80
A998426	C696018	121
A998426	D040378	80
A998753	C302116	121
A998753	D004033	80
A999775	C804902	121
A999775	D360612	80
A999999	C999999	121
A999999	D999999	80
A999999	E999999	80

```
682 rows in set (0.001 sec)
```

```
MariaDB [sistem_lpd]> INSERT INTO seleksi_psikotest (ID_Seleksi, Tanggal_Pelaksanaan, Sesi_Psikotest, Ruang_Psikotest, ID_Pengawas, NILAI) VALUES ('D999999', '2023-04-27', '3', 'C', 'B93069', '1');
Query OK, 1 row affected (0.016 sec)
```

Insert Trigger dapat dilihat, misalkan pada seleksi_wawancara yang sebelumnya passing grade berada pada nilai 80, setelah di-insert satu value/row tambahan ke dalam tabel, passing_grade akan diupdate berdasarkan averagenya. Dapat dilihat juga passing_grade yang terupdate hanya yang berhuruf depan C yang menandakan hanya seleksi wawancara lah yang diupdate. Hal berikut juga diaplikasikan pada seleksi_psikotest dan seleksi_kesehatan seperti yang dapat dilihat pada gambar.

Laporan Milestone 3

Tugas Besar IF2240

K01 - Kelompok 5

Page 20

```

MariaDB [sistem_lpd]> select * from seleksi_lanjutan;
+-----+-----+-----+
| ID_Pendaftar | ID_Seleksi | Passing_Grade |
+-----+-----+-----+
| A001041      | C233746    | 121           |
| A005605      | C919219    | 121           |
| A005605      | D888494    | 84            |
| A005754      | C487924    | 121           |
| A010348      | C287108    | 121           |
| A010348      | D387131    | 84            |
| A010348      | E796971    | 80            |
| A013139      | C762857    | 121           |
| A013139      | D276112    | 84            |
| A013139      | E852614    | 80            |
| A016794      | C914263    | 121           |
| A016794      | D560431    | 84            |
+-----+-----+-----+
| A983104      | C959094    | 121           |
| A985265      | C290029    | 121           |
| A985265      | D473382    | 84            |
| A990514      | C248305    | 121           |
| A990514      | D910019    | 84            |
| A990514      | E082945    | 80            |
| A994194      | C712830    | 121           |
| A994194      | D756507    | 84            |
| A994194      | E657985    | 80            |
| A994871      | C277746    | 121           |
| A994871      | D272183    | 84            |
| A994871      | E761170    | 80            |
| A998426      | C696018    | 121           |
| A998426      | D040378    | 84            |
| A998753      | C302116    | 121           |
| A998753      | D004033    | 84            |
| A999775      | C804902    | 121           |
| A999775      | D360612    | 84            |
| A999999      | C999999    | 121           |
| A999999      | D999999    | 84            |
| A999999      | E999999    | 80            |
+-----+-----+-----+
682 rows in set (0.001 sec)

MariaDB [sistem_lpd]> INSERT INTO seleksi_kesehatan (ID_Seleksi,Tanggal_Pelaksanaan, ID_Rumah_Sakit, Kota, Nama_Rumah_Sakit, ID_Dokter, NILAI) VALUES ('E999999','2023-05-28','3578213','Surabaya','RSUD Bhakti Dharma Husada','B06154','50');
Query OK, 1 row affected (0.008 sec)

```

```
MariaDB [sistem_lpd] > select * from seleksi_lanjutan;
```

ID_Pendaftar	ID_Seleksi	Passing_Grade
A001041	C233746	121
A005605	C919219	121
A005605	D888494	84
A005754	C487924	121
A010348	C287108	121
A010348	D387131	84
A010348	E796971	87
A013139	C762857	121
A013139	D276112	84
A013139	E852614	87
A016794	C914263	121
A016794	D560431	84
A016794	E933904	87
A018445	C190598	121
A018445	D593286	84
A026026	C589648	121
A035489	C830344	121
A035489	D453917	84
A037110	C065839	121
A037110	D716351	84
A037110	E716351	87
A982467	C407218	121
A982467	D214952	84
A982467	E528866	87
A983104	C959094	121
A985265	C290029	121
A985265	D473382	84
A990514	C248305	121
A990514	D910019	84
A990514	E082945	87
A994194	C712830	121
A994194	D756507	84
A994194	E657985	87
A994871	C277746	121
A994871	D272183	84
A994871	E761170	87
A998426	C696018	121
A998426	D040378	84
A998753	C302116	121
A998753	D004033	84
A999775	C804902	121
A999775	D360612	84
A999999	C999999	121
A999999	D999999	84
A999999	E999999	87

```
682 rows in set (0.001 sec)
```

```
MariaDB [sistem_lpd] > UPDATE seleksi_wawancara SET Nilai = 1 WHERE ID_Seleksi = "C999999";
Query OK, 1 row affected (0.007 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Update trigger, dapat dilihat jika tabel seleksi_wawancara/seleksi_p sikotest/seleksi_kesehatan diupdate, maka passing_grade akan dihitung ulang untuk seleksi tersebut.

```
MariaDB [sistem_lpd] > select * from seleksi_lanjutan;
```

ID_Pendaftar	ID_Seleksi	Passing_Grade
A001041	C233746	88
A005605	C919219	88
A005605	D888494	84
A005754	C487924	88
A010348	C287108	88
A010348	D387131	84
A010348	E796971	87
A013139	C762857	88
A013139	D276112	84
A013139	E852614	87
A016794	C914263	88
A016794	D560431	84
A016794	E933904	87
A018445	C190598	88
A018445	D593286	84
A026026	C589648	88
A994871	C277746	88
A994871	D272183	84
A994871	E761170	87
A998426	C696018	88
A998426	D040378	84
A998753	C302116	88
A998753	D004033	84
A999775	C804902	88
A999775	D360612	84
A999999	C999999	88
A999999	D999999	84
A999999	E999999	87

```
682 rows in set (0.001 sec)
```

```
MariaDB [sistem_lpd] > UPDATE seleksi_psikotest SET Nilai = 500 WHERE ID_Seleksi = "D999999";
```

```
Query OK, 1 row affected (0.008 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [sistem_lpd] > select * from seleksi_lanjutan;
```

ID_Pendaftar	ID_Seleksi	Passing_Grade
A001041	C233746	88
A005605	C919219	88
A005605	D888494	86
A005754	C487924	88
A010348	C287108	88
A010348	D387131	86
A010348	E796971	87
A013139	C762857	88
A013139	D276112	86
A013139	E852614	87
A016794	C914263	88
A016794	D560431	86
A016794	E933904	87
A018445	C190598	88

A981186	D488145	86
A982467	C407218	88
A982467	D214952	86
A982467	E528866	87
A983104	C959094	88
A985265	C290029	88
A985265	D473382	86
A990514	C248305	88
A990514	D910019	86
A990514	E082945	87
A994194	C712830	88
A994194	D756507	86
A994194	E657985	87
A994871	C277746	88
A994871	D272183	86
A994871	E761170	87
A998426	C696018	88
A998426	D040378	86
A998753	C302116	88
A998753	D004033	86
A999775	C804902	88
A999775	D360612	86
A999999	C999999	88
A999999	D999999	86
A999999	E999999	87

682 rows in set (0.001 sec)

```
MariaDB [sistem_lpd]> UPDATE seleksi_kesehatan SET Nilai = 7000 WHERE ID_Se
leksi = "E999999";
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [sistem_lpd]> select * from seleksi_lanjutan;
```

ID_Pendaftar	ID_Seleksi	Passing_Grade
A001041	C233746	88
A005605	C919219	88
A005605	D888494	86
A005754	C487924	88
A010348	C287108	88
A010348	D387131	86
A010348	E796971	140
A013139	C762857	88
A013139	D276112	86
A013139	E852614	140
A016794	C914263	88
A016794	D560431	86


```

| A903104 | C999094 | 80 |
| A985265 | C290029 | 88 |
| A985265 | D473382 | 86 |
| A990514 | C248305 | 88 |
| A990514 | D910019 | 86 |
| A990514 | E082945 | 140 |
| A994194 | C712830 | 88 |
| A994194 | D756507 | 86 |
| A994194 | E657985 | 140 |
| A994871 | C277746 | 88 |
| A994871 | D272183 | 86 |
| A994871 | E761170 | 140 |
| A998426 | C696018 | 88 |
| A998426 | D040378 | 86 |
| A998753 | C302116 | 88 |
| A998753 | D004033 | 86 |
| A999775 | C804902 | 88 |
| A999775 | D360612 | 86 |
| A999999 | C999999 | 88 |
| A999999 | D999999 | 86 |
| A999999 | E999999 | 140 |
+-----+-----+-----+

```

682 rows in set (0.001 sec)

```

MariaDB [sistem_lpd]> DELETE FROM seleksi_wawancara WHERE ID_Seleksi = "C99
9999";
Query OK, 1 row affected (0.008 sec)

```

```

MariaDB [sistem_lpd]> select * from seleksi_lanjutan;

```

```

+-----+-----+-----+
| ID_Pendaftar | ID_Seleksi | Passing_Grade |
+-----+-----+-----+
| A001041 | C233746 | 88 |
| A005605 | C919219 | 88 |
| A005605 | D888494 | 86 |
| A005754 | C487924 | 88 |
| A010348 | C287108 | 88 |
| A010348 | D387131 | 86 |
| A010348 | E796971 | 140 |
| A013139 | C762857 | 88 |
| A013139 | D276112 | 86 |
| A013139 | E852614 | 140 |
| A016794 | C914263 | 88 |
| A016794 | D560431 | 86 |
| A016794 | E933904 | 140 |
| A018445 | C190598 | 88 |
| A018445 | D593286 | 86 |
|
| A994194 | E657985 | 140 |
| A994871 | C277746 | 88 |
| A994871 | D272183 | 86 |
| A994871 | E761170 | 140 |
| A998426 | C696018 | 88 |
| A998426 | D040378 | 86 |
| A998753 | C302116 | 88 |
| A998753 | D004033 | 86 |
| A999775 | C804902 | 88 |
| A999775 | D360612 | 86 |
| A999999 | C999999 | 88 |
| A999999 | D999999 | 86 |
| A999999 | E999999 | 140 |
+-----+-----+-----+

```

682 rows in set (0.001 sec)

Terakhir, delete trigger, passing_grade juga akan dihitung ulang untuk seleksi tersebut jika terdapat value/row yang di-delete dari seleksi_wawancara/seleksi_p sikotest/seleksi_kesehatan.

```
MariaDB [sistem_lpd] > DELETE FROM seleksi_psikotest WHERE ID_Seleksi = "D999999";
Query OK, 1 row affected (0.009 sec)
```

```
MariaDB [sistem_lpd] > select * from seleksi_lanjutan;
```

ID_Pendaftar	ID_Seleksi	Passing_Grade
A001041	C233746	88
A005605	C919219	88
A005605	D888494	84
A005754	C487924	88
A010348	C287108	88
A010348	D387131	84
A010348	E796971	140
A013139	C762857	88
A013139	D276112	84
A013139	E852614	140
A016794	C914263	88
A016794	D560431	84
A016794	E933904	140
A982467	E528866	140
A983104	C959094	88
A985265	C290029	88
A985265	D473382	84
A990514	C248305	88
A990514	D910019	84
A990514	E082945	140
A994194	C712830	88
A994194	D756507	84
A994194	E657985	140
A994871	C277746	88
A994871	D272183	84
A994871	E761170	140
A998426	C696018	88
A998426	D040378	84
A998753	C302116	88
A998753	D004033	84
A999775	C804902	88
A999775	D360612	84
A999999	C999999	88
A999999	D999999	84
A999999	E999999	140

```
682 rows in set (0.000 sec)
```

```
MariaDB [sistem_lpd] > DELETE FROM seleksi_kesehatan WHERE ID_Seleksi = "E999999";
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [sistem_lpd] > select * from seleksi_lanjutan;
```

ID_Pendaftar	ID_Seleksi	Passing_Grade
A001041	C233746	88
A005605	C919219	88
A005605	D888494	84
A005754	C487924	88
A010348	C287108	88
A010348	D387131	84
A010348	E796971	87
A013139	C762857	88
A013139	D276112	84
A013139	E852614	87
A016794	C914263	88
A016794	D560431	84
A016794	E933904	87
A018445	C190598	88
A018445	D593286	84
A026026	C589648	88
A035489	C830344	88
A035489	D453917	84
A037110	C065839	88
A037110	D716351	84
A982467	E528866	87
A983104	C959094	88
A985265	C290029	88
A985265	D473382	84
A990514	C248305	88
A990514	D910019	84
A990514	E082945	87
A994194	C712830	88
A994194	D756507	84
A994194	E657985	87
A994871	C277746	88
A994871	D272183	84
A994871	E761170	87
A998426	C696018	88
A998426	D040378	84
A998753	C302116	88
A998753	D004033	84
A999775	C804902	88
A999775	D360612	84
A999999	C999999	88
A999999	D999999	84
A999999	E999999	87

```
682 rows in set (0.001 sec)
```

Notes Tambahan

Trigger dibuat hanya untuk mengganti passing grade setiap jenis seleksi menjadi data yang dinamis. Passing grade akan berubah setiap terdapat insert atau delete atau update pada tabel seleksi wawancara atau seleksi psikotest atau seleksi kesehatan.

Passing grade yang diganti setelah melakukan suatu insert atau delete atau update pada

Laporan Milestone 3

Tugas Besar IF2240

K01 - Kelompok 5

Page 27

salah satu tabel yang disebutkan di atas hanyalah passing grade untuk seleksi yang dilakukan aksi insert/delete/update. Misalkan dilakukan insert pada seleksi wawancara, maka hanya passing grade pada seleksi wawancara yang akan diupdate karena seleksi lain tidak terdapat perubahan nilai.

Akan tetapi, trigger tidak dapat merubah status suatu pendaftar menjadi tidak lolos maupun lolos pada seleksi. Misalkan passing grade awal pada seleksi wawancara adalah 80, Budi memiliki nilai sebesar 81, tetapi terdapat insert pada seleksi wawancara yang menyebabkan passing grade menjadi 82. Maka Budi akan tetap dinyatakan lolos dan dapat mengikuti seleksi selanjutnya walaupun nilainya lebih rendah dari passing grade yang telah dirubah.

Hal ini sudah dicoba untuk di-handle dengan merubah trigger menjadi seperti di kanan. Akan tetapi, tidak dapat dilakukan karena jika misalkan pada kasus di atas tadi. Saat dilakukan insert pada seleksi wawancara, passing grade diubah menjadi 82 dan terdapat beberapa pendaftar yang menjadi tidak lolos seperti Budi. Maka akan dilakukan delete pada seleksi selanjutnya yaitu seleksi psikotest, tetapi aksi delete pada seleksi psikotest ini juga me-trigger trigger lain yaitu jika terdapat aksi delete pada seleksi psikotest yang tentunya trigger ini akan melakukan perhitungan ulang passing grade dan melakukan aksi delete kembali pada seleksi selanjutnya.

Masalahnya di sini adalah saat trigger kedua melakukan perhitungan ulang passing grade yang akan melakukan update pada tabel seleksi lanjutan di mana passing grade disimpan, padahal trigger sebelumnya juga melakukan suatu aksi pada tabel seleksi lanjutan. Hal ini tidak boleh dilakukan pada mariadb maupun mysql karena akan muncul error seperti berikut :

“ERROR 1442 (HY000): Can't update table 'seleksi_lanjutan' in stored function/trigger because it is already used by statement which invoked this stored function/trigger”

“MySQL does not allow a stored function or trigger to modify the same table that was used in the statement that called the function or trigger. This is a safety feature to prevent potential conflicts and ensure data integrity.”

Maka dari itu, kami tidak jadi melakukan dan mengaplikasikan hal tersebut pada trigger.

3. *Constraint*

Dari query yang diberikan dibawah, dihasilkan ID_Pendaftar yang tidak lolos tahap wawancara.

```
SELECT ID_Pendaftar
FROM seleksi_lanjutan NATURAL JOIN seleksi_wawancara
EXCEPT
SELECT ID_Pendaftar
FROM seleksi_lanjutan NATURAL JOIN seleksi_psikotest;
```

```
A005143, A027860, A035052, A038590, A112747, A131244, A153216, A171070, A173985,
```

```
A176635, A184863, A200815, A213017, A227766, A267999, A294809, A311382, A347808,
A370643, A384326, A406296, A411474, A418603, A460149, A473604, A483479, A487211,
A495916, A496983, A580016, A604040, A628992, A671773, A724920, A755347, A760323,
A763071, A780811, A782663, A805502, A810718, A815403, A840932, A896285, A955557,
A979809 (46 data)
```

Sehingga jika mencoba mengikutkan peserta pada seleksi kesehatan, akan seperti berikut:

```
INSERT INTO seleksi_lanjutan (ID_Pendaftar, ID_Seleksi, Passing_Grade) VALUES
('A005143', 'E999999', '80');
```

```
MariaDB [test_rules]> INSERT INTO seleksi_lanjutan (ID_Pendaftar, ID_Seleksi, Passing_Grade) VALUES ('A005143','E999999',
'80');
ERROR 1644 (45000): Tidak bisa melakukan insertion, seleksi kesehatan harus setelah seleksi wawancara dan seleksi psikot
est
```

Hal ini terjadi karena ID_Pendaftar 'A005143' tidak lolos seleksi wawancara, sehingga tidak bisa melakukan *insert* untuk ID_Pendaftar tersebut ke dalam seleksi kesehatan.

Bab 5 : Kesimpulan

5.1. Kesimpulan

Pada Tugas Besar IF2240 Basis Data ini telah diimplementasikan sebuah skema basis data dengan model relasional untuk memenuhi kebutuhan kasus sistem LPDP terkait pemberian beasiswa dan juga pendataan tes seleksi. Dalam implementasinya, dibuat juga sebuah E-R Diagram sebagai basis dari pengembangan skema, yang lalu dilanjutkan menjadi sebuah skema relasional yang diimplementasikan dalam MySQL. Dalam implementasi skema relasional, dibuat juga data-data pelengkap dalam basis data menggunakan sebuah *faker*. Terakhir, skema basis data yang sudah dikembangkan, dilakukan normalisasi dan juga diterapkan *business rule*.

Dengan demikian, penulis menyimpulkan bahwa melalui Tugas Besar IF2240 Basis Data ini, dapat dibuat sebuah skema relasional mengenai kasus LPDP terkait pemberian beasiswa yang mencakup tes seleksi dan penilaian dalam MySQL dan diaplikasikan normalisasi dan juga *business rule*, dengan skema yang dilandasi sebuah E-R Diagram.

5.2. Saran

Tugas Kecil IF2211 Strategi Algoritma Semester II Tahun 2022/2023 menjadi salah satu tugas yang memberikan pelajaran baru bagi penulis. Berdasarkan pengalaman penulis mengerjakan tugas ini, berikut merupakan saran untuk pembaca yang ingin melakukan atau mengerjakan hal yang serupa.

Keefektifan dalam kerjasama tim merupakan hal yang penting dalam mengerjakan tugas ini. Tugas ini sangat terbantu oleh pemakaian real-time collaboration app. Selain itu, pemakaian aplikasi pengelola version control seperti Github sangat disarankan agar memudahkan untuk mengelola pekerjaan secara asinkron.

Dibutuhkan sebuah list dari dependencies dan library yang akan digunakan. Tugas ini berbasis web dan sangat terbantu dengan pemakaian library, tetapi, library tersebut menjadi sebuah dependency yang kemudian perlu di-install untuk menjalankan web secara lokal.

Pemahaman Struktur Data, Algoritma A* dan Uniform Cost Search, dan Konsep Pemrograman Berorientasi Objek sangat penting dalam pengerjaan tugas. Dengan pemahaman yang memadai, penulis merasa sangat dimudahkan dalam perancangan sampai implementasi algoritma.

Pemahaman web development merupakan hal yang sangat membantu dalam pengerjaan tugas ini. Hal tersebut dikarenakan Bonus yang diberikan, yaitu penggunaan API sangat dimudahkan jika aplikasi dapat dibuat berbasis web.

Referensi

Silberschatz, Abraham, Korth, Henry F., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill Education.

Tim Pengajar IF2140/IF2240. (2020). *IF2140 – Pemodelan Basis Data / IF2240 – Basis Data Database Design using E-R Model*

Tim Pengajar IF2140/IF2240. (2020). *IF2140 – Pemodelan Basis Data / IF2240 – Basis Data Database Design using E-R Model (p.2)*

Tim Pengajar IF2140/IF2240. (2020). *IF2140 – Pemodelan Basis Data / IF2240 – Basis Data Database Design using E-R Model (p.3)*

