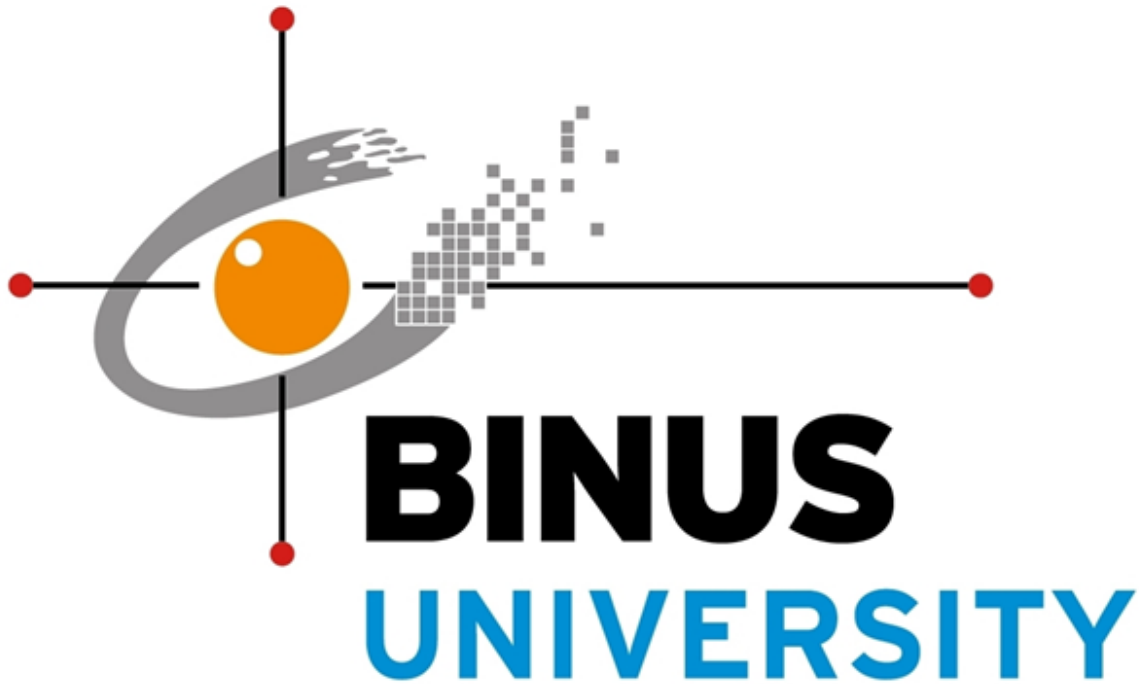


Project Database Design



LB01

Kelompok 4

Anggota Kelompok:

2301863521 - Adrian Kristanto
2301887830 - Dewi Mutiara
2301919366 - Paula Khuangnata
2301922796 - Vitariani
2301923464 - Ahmad Naufal Farras
2301941271 - Kace Purnomo
2301941990 - Fauza Wahidira
2301947520 - Fachrurrozi Maulana
2301950710 - Muhammad Hizam Al Ibrahim

Binus University
Jakarta

Case : Universitas

Mission Statement

Dalam suatu universitas tentunya memiliki berbagai macam data yang harus dikelola. Mulai dari data mahasiswa, nilai-nilai mahasiswa, transaksi pembayaran, dan lain sebagainya. Seluruh data tersebut harus dikelola dengan benar agar dapat terhindar dari kesalahan pendataan yang dapat merugikan pihak-pihak yang terkait. Karena terdapat banyaknya data yang harus ditangani, kami memutuskan untuk membuat database sebagai sarana penampungan data yang akan dilakukan oleh sebuah universitas. Dengan adanya database ini, maka data-data tidak akan terjadi anomali data maupun redundansi data sehingga tidak akan terjadi kesalahan dalam pendataan maupun pada saat melakukan transaksi. Database ini juga akan memudahkan para staff, baik para petinggi maupun karyawan biasa untuk mengakses beberapa data yang diizinkan untuk melihat data mahasiswa, dosen maupun transaksi yang pernah dilakukan, juga memudahkan para mahasiswa dan dosen untuk mengecek data-data pribadi, jadwal kelas, maupun pembayaran yang telah dilakukan.

Mission Objectives

- Melakukan penambahan, pembaharuan, dan penghapusan data pada mahasiswa.
- Melakukan penambahan, pembaharuan, dan penghapusan data pada dosen.
- Melakukan penambahan, pembaharuan, dan penghapusan data pada staff.
- Melakukan penambahan, pembaharuan, dan penghapusan data pada transaksi.
- Melakukan penambahan, pembaharuan, dan penghapusan data pada jadwal.
- Melakukan penambahan, pembaharuan, dan penghapusan data pada enrollment.

- Melakukan pencarian data pada mahasiswa.
- Melakukan pencarian data pada dosen.
- Melakukan pencarian data pada data staff.
- Melakukan pencarian data pada transaksi.
- Melakukan pencarian data pada jadwal.
- Melakukan pencarian data enrollment.

- Melacak status data pada transaksi.
- Melacak status data pada jadwal.
- Melacak status data enrollment.

- Memberi laporan kepada mahasiswa.
- Memberi laporan kepada dosen.
- Memberi laporan kepada staff.
- Memberi laporan dalam transaksi.
- Memberi laporan kepada jadwal.

- Memberi laporan dalam enrollment.

Particular task

Membantu pendataan mahasiswa dan dosen.

- Database akan menyimpan data pribadi mahasiswa, dosen dan staff yang bekerja.

Mendata keperluan yang dibutuhkan mahasiswa.

- Keperluan seperti kelayakan mengikuti ujian dan kelayakan mendapatkan kelas.

Membantu pendataan transaksi mahasiswa menjadi lebih mudah.

- Mengidentifikasi pendataan pembayaran berdasarkan jenis kelas yang diambil.
- Menampilkan data transaksi mahasiswa.

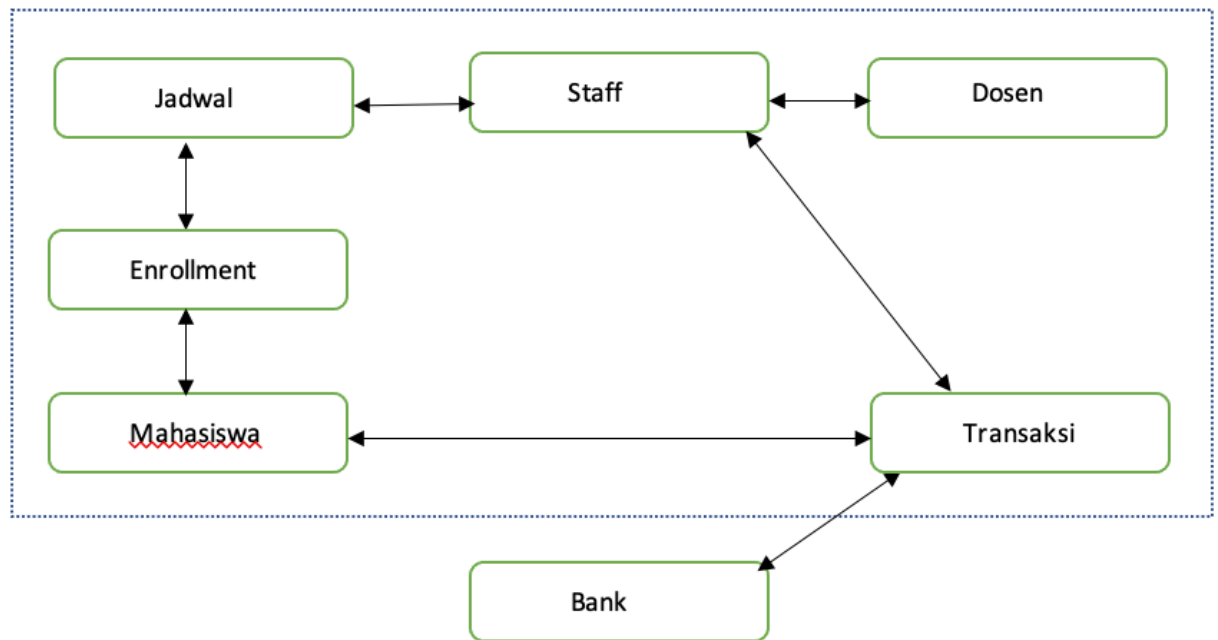
Untuk mengelola data mahasiswa dan dosen.

- Untuk melakukan update data mahasiswa dan dosen.
- Melakukan pencarian data mahasiswa dan dosen.
- Mengetahui apakah mahasiswa sudah melakukan transaksi pembayaran atau belum.

Scope :

- Staff
- Dosen
- Mahasiswa
- Jadwal: Mata Kuliah, Ruang kelas, Sesi, Fakultas, Lokasi
- Transaksi
- Mitra kerja (Bank)

Boundaries :



Cross-reference :

	Rektor	Wakil Rektor	Dekan	Wakil Dekan	Kepala Bagian	Staff	Dosen	Mahasiswa
Staff	x	x	x	x	x			
Dosen	x	x	x	x	x	x		x
Mahasiswa	x	x	x	x	x	x	x	x
Income					x	x	x	x
Outcome					x	x		x
Mitra Kerja					x	x		
Jadwal	x	x	x	x	x	x	x	x

Teknik fact finding yang kami pilih dalam pembuatan database ini untuk mengumpulkan fakta-fakta yang dibutuhkan agar dapat menunjang kebutuhan perusahaan adalah berupa examining documentation, interview, dan melakukan observasi.

Pertanyaan :

Top Level

1. Apa visi dan misi dari proyek tersebut?
2. Bagaimana tujuan/rencana dari proyek ini?
3. Support apa saja yang bisa diberikan?
4. Berapa lama tenggat waktu pengerjaan proyek?
5. Seberapa banyak budget/dana yang ada?
6. Seberapa jauh akses yang bisa diberikan?
7. Kebijakan apa saja yang harus diterapkan?

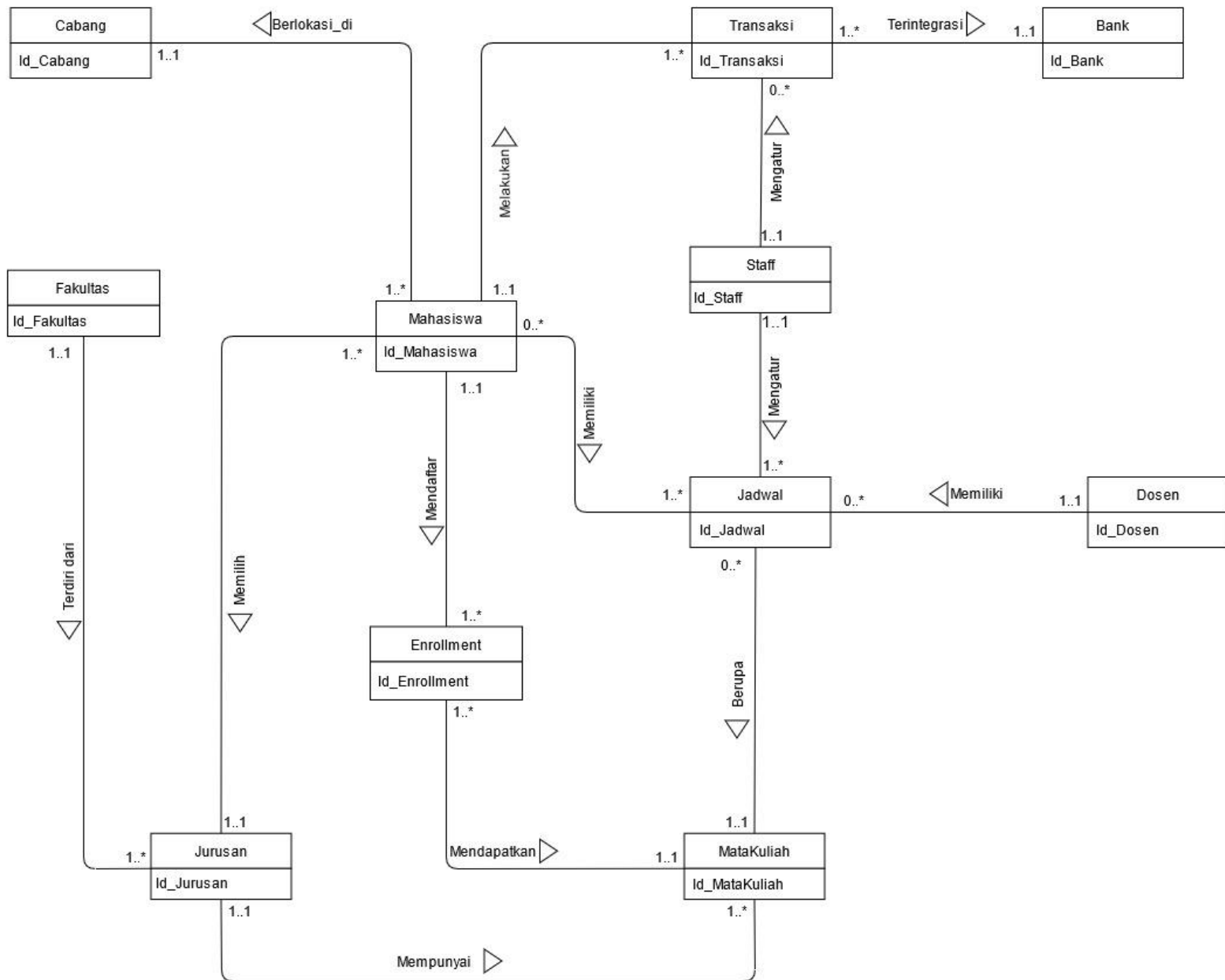
Middle Level

1. Apa deskripsi pekerjaan anda?
2. Apa jenis tugas yang anda kerjakan sehari-hari?
3. Data apa yang anda perlukan?
4. Jenis laporan seperti apa yang Anda gunakan?
5. Hal-hal apa yang harus Anda awasi?
6. Layanan apa yang disediakan oleh perusahaan Anda untuk customer?

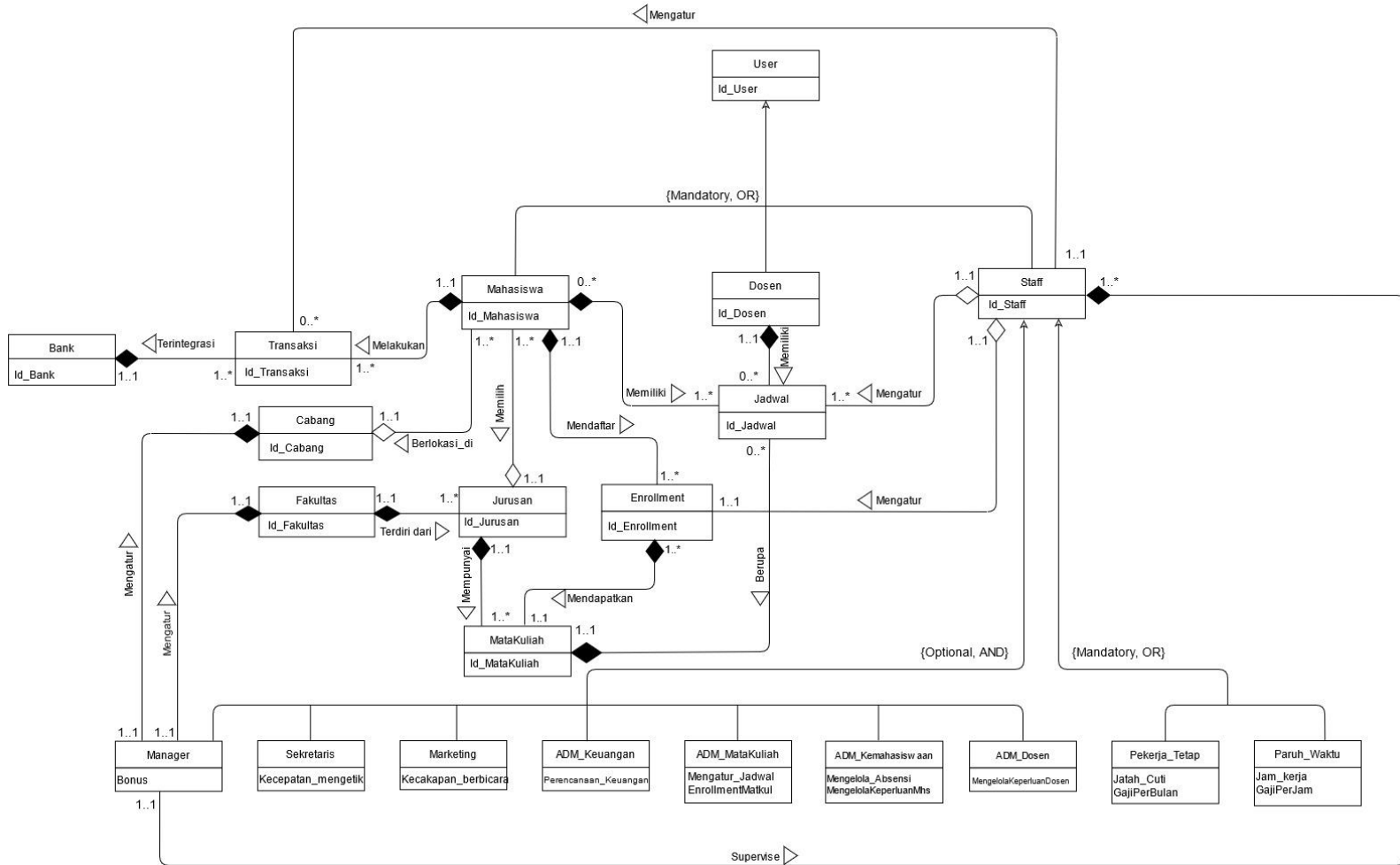
Low Level

1. Menurut anda, apakah sistem pembayaran yang sedang berjalan sudah nyaman untuk dipakai?
2. Bagaimana skill yang kalian dapatkan selama bekerja disini?
3. Apa kesulitan yang dihadapi saat mengoperasikan database yang sedang berjalan?
4. Apa kontribusi yang akan anda berikan dalam project pembuatan database ini?
5. Apa yang akan anda lakukan ketika menemukan kesulitan dalam mengoperasikan sistem yang sedang berjalan?

Entity Relationship



Enhanced Entity Relationship



Normalisasi

PT. BANK PAULA SEJAHTERA
RESI BUKTI BAYAR



=====

KODE TRANSAKSI : KT-098878
TANGGAL TRANSAKSI : 12 Maret 2020
NAMA : Fauza Wahidira
NIM : 2301941990
ID PRODI - NAMA PRODI : SOCS - Computer Science
JENIS PEMBAYARAN : Autodebet
TAHUN AJARAN : 2020, Even Semester

=====

No	Nomor Pembayaran	Jenis Pembayaran	Biaya (Rp)
1	NP-001	Fixed Tuition Fee	2.500.000
2	NP-002	Fixed Tuition Fee	2.500.000
3	NP-003	Lab Fee	1.500.000
4	NP-005	Variable Tuition Fee	1.000.000
5	NP-006	Textbook Fee	500.000
6	NP-007	Variable Tuition Fee	1.000.000
Total Biaya			9.000.000

TRANSAKSI SUKSES, RESI INI ADALAH BUKTI PEMBAYARAN YANG SAH.

**Dokumen ini dibuat secara otomatis oleh sistem komputer sebagai bukti pembayaran.

Staff

SF001 - Arul

UNF

Transaksi

{Kode_Transaksi, Tanggal_Transaksi, Nama_Mahasiswa, NIM_Mahasiswa, Id_Prodi, Nama_Prodi, Jenis_Transaksi, TahunAjaran_Mahasiswa, Semester_Mahasiswa, Nomor_Pembayaran, Jenis_Pembayaran, Biaya, Nama_Staff, Id_Staff, Total_Biaya_Transaksi}

1NF

Transaksi

{Kode_Transaksi(PK), Tanggal_Transaksi, Nama_Mahasiswa, NIM_Mahasiswa, Id_Prodi, Nama_Prodi, Jenis_Transaksi, TahunAjaran_Mahasiswa, Semester_Mahasiswa, Nama_Staff, Id_Staff}

Detail_Transaksi

{Kode_Transaksi(PK)(FK), Nomor_Pembayaran(PK), Jenis_Pembayaran, Biaya}

2NF

Transaksi

{Kode_Transaksi(PK), Tanggal_Transaksi, Nama_Mahasiswa, NIM_Mahasiswa, Id_Prodi, Nama_Prodi, Jenis_Transaksi, TahunAjaran_Mahasiswa, Semester_Mahasiswa, Nama_Staff, Id_Staff}

Detail_Transaksi

{Kode_Transaksi(PK)(FK), Nomor_Pembayaran(PK)(FK), Biaya}

Pembayaran

{Nomor_Pembayaran(PK), Jenis_Pembayaran}

3NF

Transaksi

{Kode_Transaksi(PK), Tanggal_Transaksi, Jenis_Transaksi, NIM_Mahasiswa(FK), Id_Staff(FK)}

Detail_Transaksi

{Kode_Transaksi(PK)(FK), Nomor_Pembayaran(PK)(FK), Biaya}

Pembayaran

{Nomor_Pembayaran(PK), Jenis_Pembayaran}

Mahasiswa

{NIM_Mahasiswa(PK), Nama_Mahasiswa, Id_Prodi(FK), TahunAjaran_Mahasiswa, Semester_Mahasiswa}

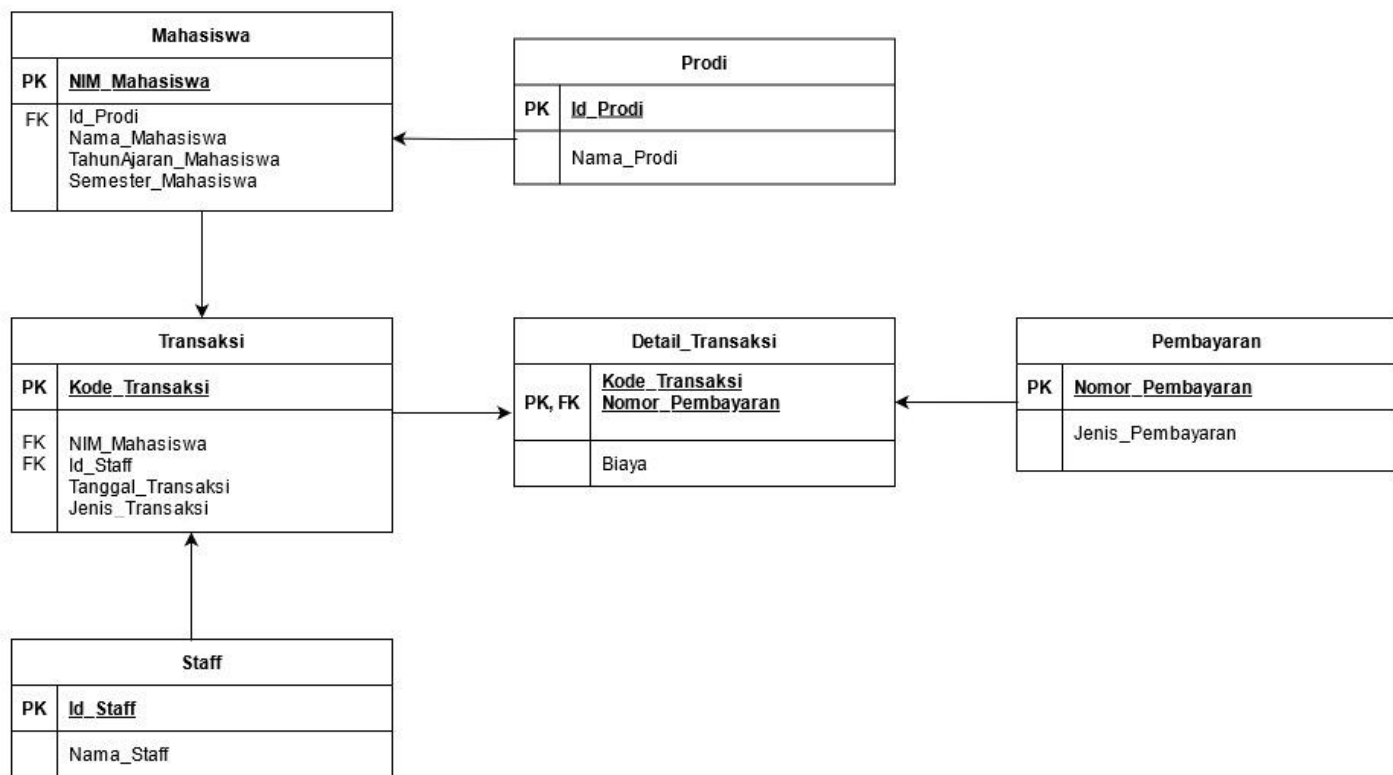
Prodi

{Id_Prodi(PK), Nama_Prodi}

Staff

{Id_Staff(PK), Nama_Staff}

ERD



Conceptual Database Design

- **Step 1.1 Identify entity types**

(Adrian Kristanto, Muhammad Hizam Al Ibrahim, Fauza Wahidira, Kace Purnomo, Ahmad Naufal Farras)

Entity name	Description	Aliases	Occurrence
Mahasiswa	Mahasiswa yang berada di universitas binus	Mahasiswa	Setiap mahasiswa mempunyai jadwal masing-masing dan telah terdaftar di salah satu jurusan fakultas. Siswa dapat melakukan banyak pendaftaran mata kuliah dan akan terdaftar pada mata kuliah tersebut.
Staff	Staff yang dipekerjakan oleh Universitas Binus	Pegawai	Setiap staff mengelola lebih dari satu transaksi yang dilakukan oleh mahasiswa dan setiap staff dapat memiliki lebih dari satu posisi jabatan. Setiap staff juga bisa mengatur jadwal untuk mahasiswa dan dosen.
Fakultas	Divisi yang terdiri dari beberapa jurusan di Universitas Binus	Fakultas	Setiap Fakultas memiliki lebih dari 1 jurusan
Transaksi	Pencatatan semua transaksi yang dilakukan oleh mahasiswa.	Transaksi	Semua catatan dan riwayat transaksi keuangan akan disimpan pada divisi keuangan.
Cabang	Sub bagian dari cabang pusat..	Cabang	Setiap mahasiswa hanya bisa berada di satu cabang

Bank	Bank yang bekerja sama untuk transaksi pembayaran mahasiswa	Bank	Setiap bank berperan sebagai perantara untuk transaksi yang dilakukan oleh mahasiswa dan banyak transaksi yang dikelola oleh staff.
Jurusan	Bagian dari sebuah fakultas untuk mengembangkan suatu bidang studi	Jurusan	Setiap jurusan memiliki lebih dari satu mahasiswa dan terdiri dari banyak mata kuliah.
Enrollment	Aktivitas atau tindakan untuk mendaftar atau didaftarkan sebuah mata kuliah	Enrollment	Setiap mahasiswa bisa melakukan lebih dari satu pendaftaran pada banyak mata kuliah
Jadwal	Pembagian waktu yang berhubungan dengan pembelajaran di Universitas Binus	Jadwal	Setiap mata kuliah mempunyai banyak jadwal untuk mahasiswa dan dosen berdasarkan rencana pengaturan perintah kerja.
Dosen	Dosen yang mengajar di Universitas Binus	Pengajar	Setiap dosen memiliki lebih dari satu jadwal mengajar
MataKuliah	Mata kuliah yang tersedia di Universitas Binus	Mata Kuliah	Setiap mata kuliah hanya memiliki satu pendaftaran dan terdiri dari banyak jadwal dari berbagai jurusan.

- **Step 1.2 Identify relationship types**

(Adrian Kristanto, Muhammad Hizam Al Ibrahim, Fauza Wahidira, Kace Purnomo, Ahmad Naufal Farras)

Entity name	Multiplicity	Relationship	Multiplicity	Entity name
Mahasiswa	0..*	Memiliki	1..*	Jadwal
	1..1	Melakukan	1..*	Transaksi
	1..*	Berlokasi_d	1..1	Cabang
	1..*	Memilih	1..1	Jurusan
	1..1	Mendaftar	1..*	Enrollment
Staff	1..1	Mengatur	0..*	Transaksi
	1..1	Dimiliki_oleh	1..*	PosisiJabatan
	1..1	Mengatur	1..*	Jadwal
Fakultas	1..1	Terdiri dari	1..*	Jurusan
Transaksi	1..*	Terintegrasi	1..1	Bank
	1..*	Melakukan	1..1	Mahasiswa
	0..*	Mengatur	1..1	Staff
Cabang	1..1	Berlokasi_di	1..*	Mahasiswa
Bank	1..1	Terintegrasi	1..*	Transaksi
Jurusan	1..1	Memilih	1..*	Mahasiswa
	1..*	Terdiri dari	1..1	Fakultas
	1..1	Mempunyai	1..*	MataKuliah
Enrollment	1..*	Mendaftar	1..1	Mahasiswa

	1..1	Mendapatkan	1..*	MataKuliah
Jadwal	0..*	Berupa	1..1	MataKuliah
Dosen	1..1	Memiliki	0,,*	Jadwal
Mata Kuliah	1..1	Berupa	0..*	Jadwal
	1..*	Mendapatkan	1..1	Enrollment
	1..*	Mempunyai	1..1	Jurusan

- **Step 1.3 Identify and associate attributes with entity or relationship types**
(Paula Khuangnata, Dewi Mutiara, Vitariani)

Entity Name	Attributes	Description	Data Type & Length	Nulls	Multi-valued
Mahasiswa	NIM,	Kode unik mahasiswa	10 variable character,	no	no
	Nama NamaDepan	Nama depan mahasiswa	255 variable character,	no	no
	NamaBlkng,	Nama belakang mahasiswa	255 variable character,	no	no
	TempatLahir,	tempat lahir mahasiswa	20 variable character,	no	no
	TanggalLahir,	tanggal lahir mahasiswa	date,	no	no
	Gender,	Jenis kelamin mahasiswa	6 variable character,	no	no
	Alamat,	Alamat tempat tinggal mahasiswa	255 variable character,	no	no
	GPA	Nilai IPK	3 angka desimal dengan 2 angka di belakang koma		
Staff	IDStaff,	Kode unik staff	10 variable character,	no	no
	NamaStaff, NamaDepan	Nama depan staff	255 variable	no	no

	NamaBlkng, PosisiStaff, GenderStaff, DOB, Alamat Email	Nama belakang staff Jenis pekerjaan staff Jenis kelamin staff Tanggal lahir staff Alamat tempat tinggal staff Alamat email staff	character, 255 variable character, 255 variable character, 6 variable character, date 255 variable character 50 variable character	no no yes no no	no yes no no no
Fakultas	IDFakultas, NamaFakultas	Kode unik fakultas Nama fakultas	6 variable character, 20 variable character	no no	no no
Transaksi	KodeTransaksi, Biaya	Kode unik transaksi Biaya transaksi yang perlu dibayar	12 variable character, integer	no no	no no
Cabang	KodeCabang, NamaCabang, AlamatCabang	Kode unik cabang universitas Nama cabang universitas Alamat cabang universitas	10 variable character, 255 variable character, 255 variable character	no no no	no no no
Bank	KodeBank, NamaBank	Kode unik bank terdaftar Nama bank terdaftar	7 variable character, 255 variable character	no no	no no
Jurusan	KodeJurusan, NamaJurusan	Kode unik jurusan Nama jurusan	4 variable character, 255 variable character	no no	no no
Enrollment	IDEnrollment, TanggalEnrollme nt	Kode unik transaksi enrollment Tanggal dilakukan enrollment	10 variable character, date	no no	no no
Jadwal	KodeJadwal, Waktu, Sesi, Lokasi	Kode unik jadwal Waktu jadwal sesi jadwal Tempat jadwal	8 variable character, date, integer, 20 variable character	no no no no	no no yes yes

Dosen	IDDosen,	Kode unik dosen	8 variable character,	no	no
	NamaDosen,	Nama depan dosen	255 variable character,	no	no
	NamaDepan	Nama belakang dosen	255 variable character,	no	no
	NamaBlkng,	Jenis kelamin dosen	6 variable character,	no	no
	GenderDosen,	Alamat tempat tinggal dosen	255 variable character,	yes	no
	AlamatDosen,	Tanggal lahir dosen	date	no	no
	DOB	Alamat email dosen	50 variable character	no	no
	Email				
Mata kuliah	KodeMataKuliah,	Kode unik mata kuliah	8 variable character,	no	no
	NamaMataKuliah	Nama mata kuliah	225 variable character	no	no

- **Step 1.4 Determine attribute domains**
(Paula Khuangnata, Dewi Mutiara, Vitariani)

Entity Name	Attributes	Description	Domain	Domain Description	Data Type & Length	Nulls	Multi-valued
Mahasiswa	NIM,	Kode unik mahasiswa	MMxxyyzzaa	MM - sebagai kode mahasiswa xx - tahun lulus yy - tanggal lahir zz - bulan lahir aa - mahasiswa ke berapa dengan tanggal dan bulan lahir yang sama	10 variable character,	no	no
	Nama	Nama depan mahasiswa			255 variable character,	no	no
	NamaDepan						
	NamaBlkng,						
	TempatLahir,	Nama belakang mahasiswa	Male / Female	Memilih antara Male / Female	255 variable character,	no	no
	TanggalLahir,	tempat lahir mahasiswa					
	Gender,	tanggal lahir mahasiswa					
	Alamat,	Jenis kelamin mahasiswa	Jl. xxxxx	Alamat harus diawali dengan huruf "Jl."	6 variable character,	no	no
		Alamat tempat tinggal mahasiswa					
	GPA	Nilai IPK			255 variable character,	no	yes
					3 angka desimal	no	no

	Email	Alamat email mahasiswa	xxxxxx@gmail.com	Harus diakhiri dengan @gmail.com	dengan 2 angka di belakang koma 50 variable character	no	no
Staff	IDStaff,	Kode unik staff	STxyyyzzaa	ST - sebagai kode staff xx - tanggal masuk staff yy - bulan masuk staff zz - kode divisi aa - staff ke berapa pada bulan masuk yang sama	10 variable character,	no	no
	NamaStaff, NamaDepan	Nama depan staff			255 variable character,	no	no
	NamaBlkng,	Nama belakang staff			255 variable character,	no	no
	PosisiStaff,	Jenis pekerjaan staff			255 variable character,	no	yes
	GenderStaff,	Jenis kelamin staff	Male / Female	Memilih antara Male / Female	6 variable character, date	no	no
	DOB, Alamat	Tanggal lahir staff Alamat tempat tinggal staff	Jl. xxxxx	Alamat harus diawali dengan huruf "Jl."	255 variable character	yes no	no yes
	Email	Alamat email staff	xxxxxx@gmail.com	Harus diakhiri dengan @gmail.com	50 variable character	no	no
Fakultas	IDFakultas,	Kode unik fakultas	FKyyxx	FK - sebagai kode fakultas yy - sebagai kode wilayah fakultas xx - nomor fakultas	6 variable character,	no	no
	NamaFakultas	Nama fakultas			20 variable character	no	no
Transaksi	KodeTransaksi,	Kode unik transaksi	xyyyzzddaaaa	xx - tanggal transaksi yy - bulan transaksi zz - tahun	12 variable character,	no	no

	Biaya	Biaya transaksi yang perlu dibayar		transaksi dd - hari apa aaaa - transaksi ke berapa	integer	no	no
Cabang	KodeCabang,	Kode unik cabang universitas	CBxx	CB - sebagai kode cabang xx - kode wilayah cabang	10 variable character,	no	no
	NamaCabang,	Nama cabang universitas			255 variable character,	no	no
	AlamatCabang	Alamat cabang universitas	Jl. xxxxx	Alamat harus diawali dengan huruf “Jl.”	255 variable character	no	no
Bank	KodeBank,	Kode unik bank terdaftar	BNKxxxx	BNK - sebagai kode bank xxxx - kode bank contoh BCA memiliki kode bank 6094	7 variable character,	no	no
	NamaBank	Nama bank terdaftar			255 variable character,	no	no
Jurusan	KodeJurusan,	Kode unik jurusan	AAAA	AAAA - singkatan dari jurusan	4 variable character,	no	no
	NamaJurusan	Nama jurusan			255 variable character	no	no
Enrollment	IDEnrollment,	Kode unik transaksi enrollment	EMAAAAxxxx	EM- singkatan untuk enrollment aaaa - char untuk singkatan jurusan contoh BISS untuk jurusan bisnis xxxx - kode matakuliah	10 variable character,	no	no
	TanggalEnrollment	Tanggal dilakukan enrollment			date	no	no
Jadwal	KodeJadwal,	Kode unik jadwal	AAAxxxx	aaaa - char untuk singkatan jurusan contoh BISS untuk jurusan bisnis xxxx - kode	8 variable character,	no	no

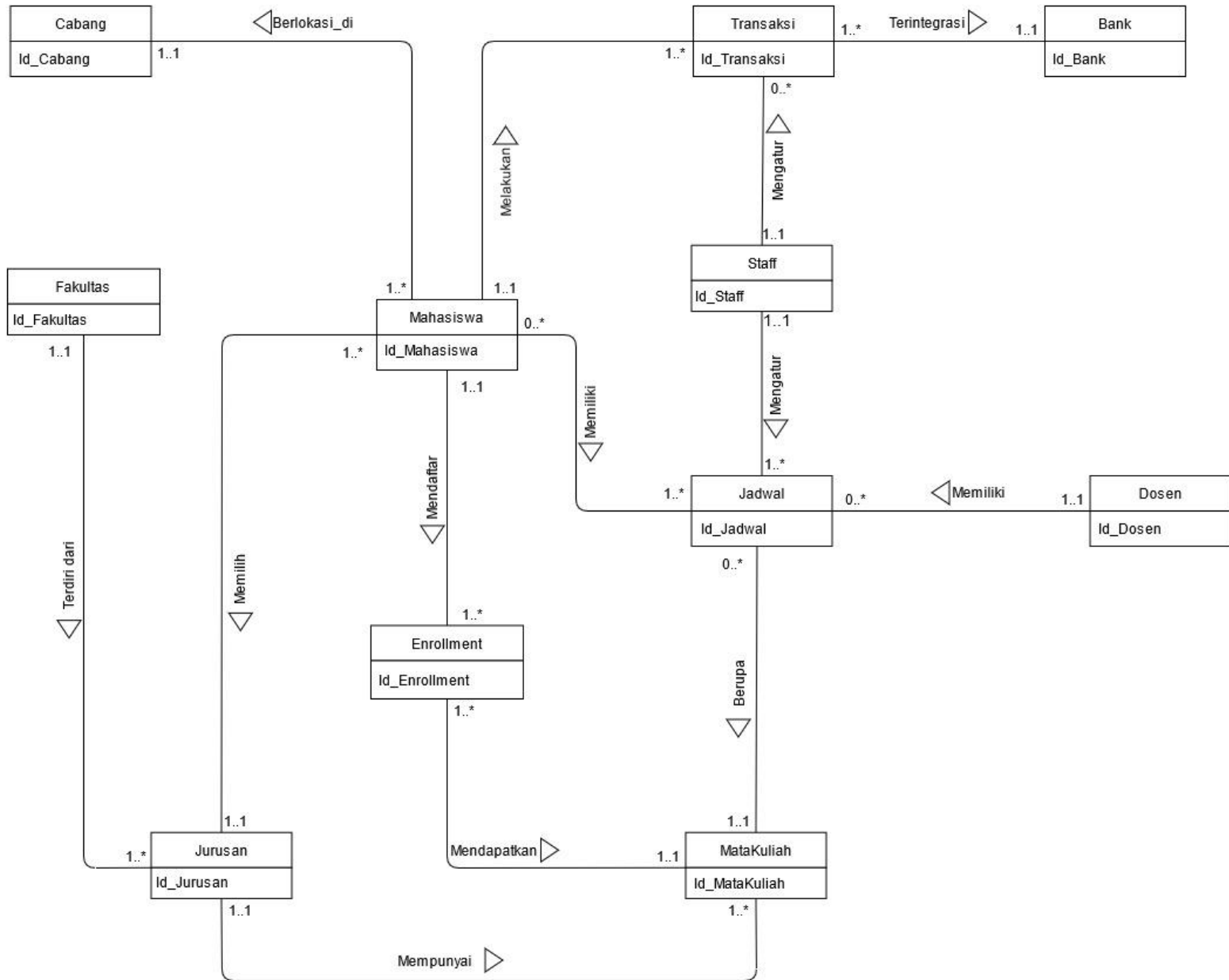
	Waktu, Sesi, Lokasi	Waktu jadwal sesi jadwal Tempat jadwal		matakuliah	date, integer, 20 variable character	no no no	no yes yes
Dosen	IDDosen, NamaDosen, NamaDepan NamaBlkng, GenderDosen, AlamatDosen, DOB Email	Kode unik dosen Nama depan dosen Nama belakang dosen Jenis kelamin dosen Alamat tempat tinggal dosen Tanggal lahir dosen Alamat email dosen	DDxxaaaa Female/Male Jl. xxxxx xxxxxxx@gmail. com	DD - sebagai kode staff xx - kode fakultas aaaa - tahun masuk dosen Memilih antara Male / Female Alamat harus diawali dengan huruf "Jl." Harus diakhiri dengan @gmail.com	8 variable character, 255 variable character, 255 variable character, 6 variable character, 255 variable character, date 50 variable character	no no no no yes no no	no no no no yes no no
Mata kuliah	KodeMataKulia h, NamaMataKulia h	Kode unik mata kuliah Nama mata kuliah	MKyyyxxx	MK - sebagai kode mata kuliah yyy - kode prodi xxx - kode mata kuliah	8 variable character, 225 variable character	no no	no no

- **Step 1.5 Determine candidate, primary, and alternate key attributes**
(Paula Khuangnata)

Entity Name	Candidate Key	Primary Key	Alternate Key
Mahasiswa	NIM, Email	NIM	Email
Staff	IDStaff, Email	IDStaff	Email
Fakultas	IDFakultas, NamaFakultas	IDFakultas	NamaFakultas

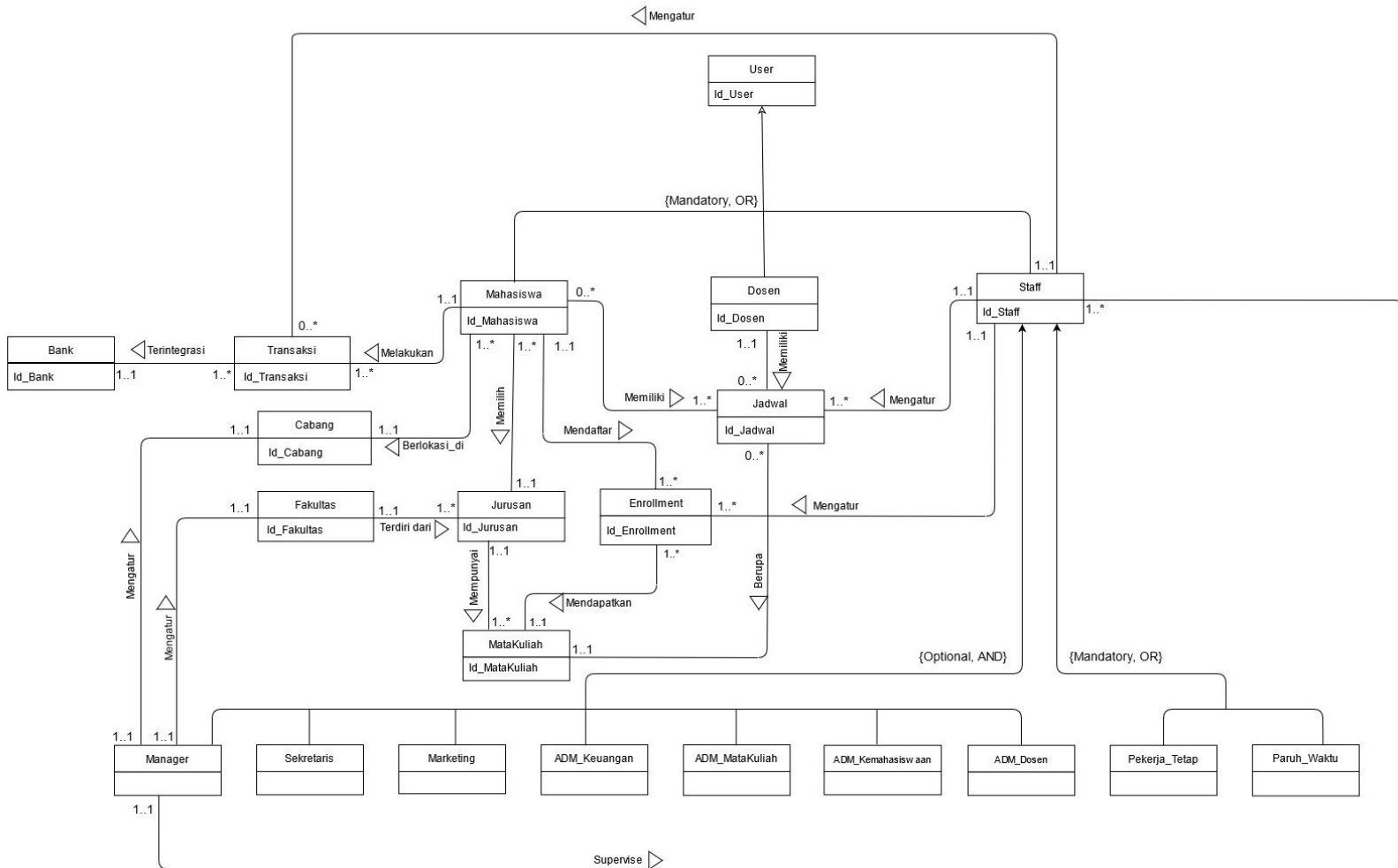
Transaksi	KodeTransaksi	KodeTransaksi	-
Cabang	KodeCabang, NamaCabang	KodeCabang	NamaCabang
Bank	KodeBank, NamaBank	KodeBank	NamaBank
Jurusan	KodeJurusan, NamaJurusan	KodeJurusan	NamaJurusan
Enrollment	IDEnrollment	IDEnrollment	-
Jadwal	KodeJadwal, Lokasi	KodeJadwal	Lokasi
Dosen	IDDosen, Email	IDDosen	Email
Mata Kuliah	KodeMataKuliah NamaMataKuliah	KodeMataKuliah	NamaMataKuliah

(Adrian Kristanto, Paula Khuangnata, Vitariani, Ahmad Naufal Farras, Fauza Wahidira, Muhammad Hizam Al Ibrahim, Dewi Mutiara, Kace Purnomo)



- **Step 1.6 Consider use of enhanced modeling concepts (optional step)**
(Adrian Kristanto, Paula Khuangnata, Vitariani, Ahmad Naufal Farras, Fauza Wahidira, Muhammad Hizam Al Ibrahim, Dewi Mutiara, Kace Purnomo)

- **Step 1.6 Consider use of enhanced modeling concepts (optional step)**
(Adrian Kristanto, Paula Khuangnata, Vitariani, Ahmad Naufal Farras, Fauza Wahidira, Muhammad Hizam Al Ibrahim, Dewi Mutiara, Kace Purnomo)

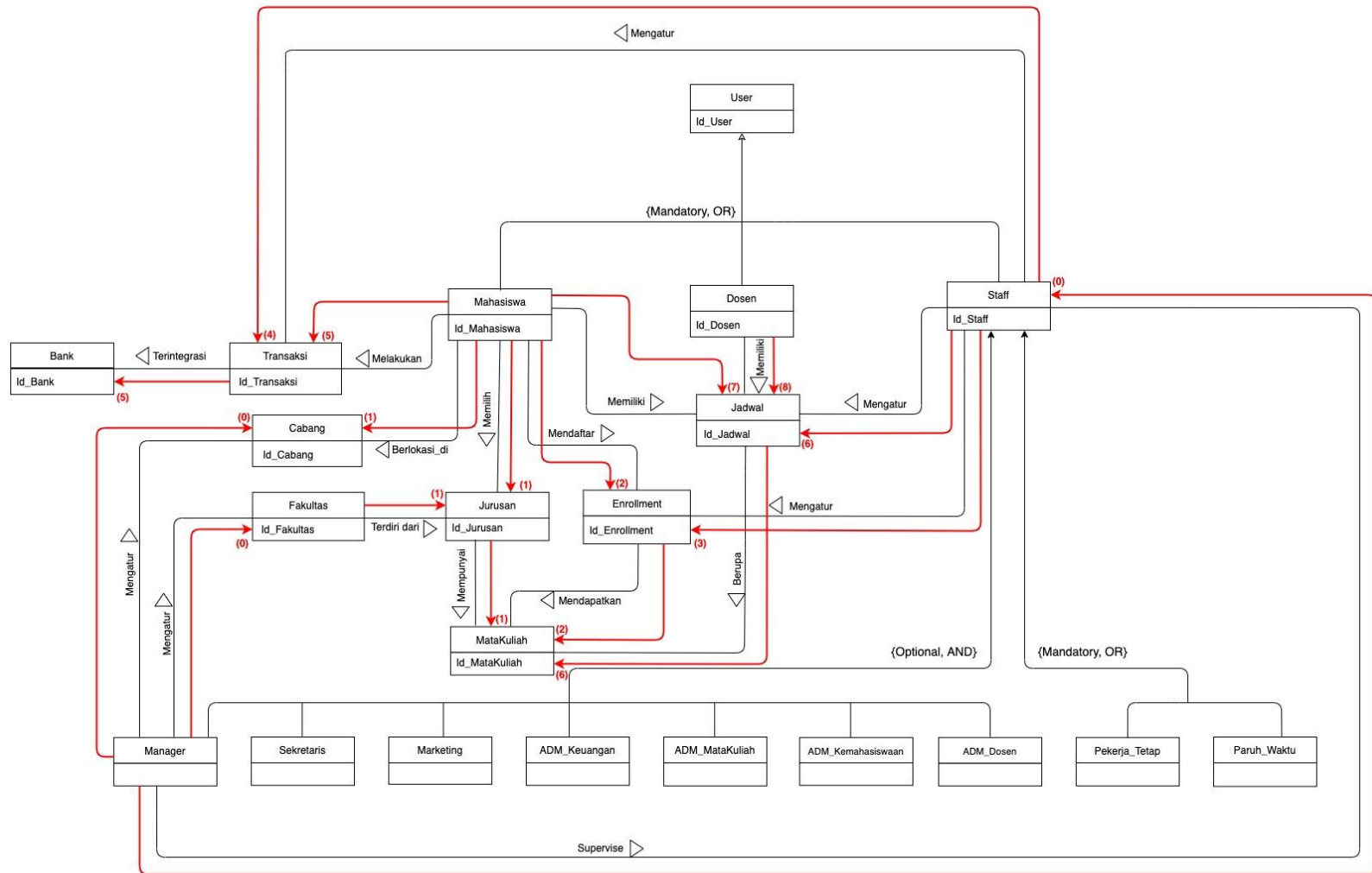


- **Step 1.7 Check model for redundancy**

(Paula Khuangnata)

Tidak ada redundansi karena setiap entity dan relationship mempunyai fungsi masing-masing.

- **Step 1.8 Validate conceptual model against user transactions**
(Adrian Kristanto, Paula Khuangnata, Vitariani)



Logical Database Design

2.1 Derive Relations for Logical Data Models

2.1.1 Strong Entity Types

Strong Entity	Attribute
Cabang	(Primary Key) KodeCabang, NamaCabang, AlamatCabang
Fakultas	(Primary Key) IDFakultas, NamaFakultas
Bank	(Primary Key) KodeBank, NamaBank
Dosen	(Primary Key) IDDosen, NamaDosen, NamaDepan, NamaBlkng, GenderDosen, AlamatDosen, DOB, Email
Staff	(Primary Key) IDStaff, NamaStaff, NamaDepan, NamaBlkng, PosisiStaff, GenderStaff, DOB, Alamat, Email

2.1.2 Weak Entity Types

Weak Entity	Attribute
Mahasiswa	(Primary Key) NIM, (Foreign Key) KodeCabang reference to Cabang(KodeCabang), (Foreign Key) KodeJadwal reference to Jadwal(KodeJadwal), (Foreign Key) KodeJurusan reference to Jurusan(KodeJurusan), Nama

	NamaDepan NamaBlkng, TempatLahir, TanggalLahir, Gender, Alamat, GPA Email
Enrollment	(Primary Key) IDEnrollment, (Foreign Key) Id_Mahasiswa reference to Mahasiswa(NIM), (Foreign Key) KodeMataKuliah reference to MataKuliah (KodeMataKuliah), TanggalEnrollment
Jadwal	(Primary Key) KodeJadwal, (Foreign Key) Id_Staff reference to Staff(IDStaff), (Foreign Key) Id_Dosen reference to Dosen(Id_Dosen), (Foreign Key) Id_MataKuliah reference to MataKuliah(Id_MataKuliah), Waktu, Sesi, Lokasi
Transaksi	(Primary Key) KodeTransaksi, (Foreign Key) Id_Staff reference to Staff(IDStaff), (Foreign Key) ID_Mahasiswa reference to Mahasiswa(NIM), (Foreign Key) KodeBank references Bank(KodeBank), Biaya
Mata Kuliah	(Primary Key) KodeMataKuliah, (Foreign Key) IDJurusan reference to Jurusan(KodeJurusan), NamaMataKuliah
Jurusan	(Primary Key) KodeJurusan, (Foreign Key) IDFakultas reference to Fakultas (IDFakultas), NamaJurusan

2.1.3 One to Many (1:*) Binary Relationship Types

Entity	Attribute	Relationship
--------	-----------	--------------

Fakultas (Parent)	(Primary Key) IDFakultas, (Alternate Key) NamaFakultas	Terdiri dari
Jurusan (Child)	(Primary Key) KodeJurusan, (Foreign Key) IDFakultas reference to Fakultas (IDFakultas), (Alternate Key) NamaJurusan	
Jurusan (Parent)	(Primary Key) KodeJurusan, NamaJurusan	Mempunyai
MataKuliah (Child)	(Primary Key) KodeMataKuliah, (Foreign Key) ID_Matkul reference to Jurusan(KodeJurusan), (Alternate Key) NamaMataKuliah	
MataKuliah (Parent)	(Primary Key) KodeMataKuliah, (Alternate Key) NamaMataKuliah	Mendapatkan
Enrollment (Child)	(Primary Key) IDEnrollment, (Foreign Key) KodeMataKuliah reference to MataKuliah (KodeMataKuliah) TanggalEnrollment	
Mahasiswa (Parent)	(Primary Key) NIM, Nama NamaDepan NamaBlkng, TempatLahir, TanggalLahir, Gender, Alamat, GPA (Alternate Key) Email	Mendaftar
Enrollment (Child)	(Primary Key) IDEnrollment, (Foreign Key) Id_Mahasiswa reference to Mahasiswa(NIM), TanggalEnrollment	
Cabang (Parent)	(Primary Key) KodeCabang, (Alternate Key) NamaCabang, AlamatCabang	Berlokasi_di
Mahasiswa (Child)	(Primary Key) NIM, (Foreign Key) KodeCabang reference to Cabang(KodeCabang), Nama NamaDepan,	

	NamaBlkng, TempatLahir, TanggalLahir, Gender, Alamat, GPA (Alternate Key) Email	
Mahasiswa (Parent)	(Primary Key) NIM, Nama NamaDepan NamaBlkng, TempatLahir, TanggalLahir, Gender, Alamat, GPA (Alternate Key) Email	Melakukan
Transaksi (Child)	(Primary Key) KodeTransaksi, (Foreign Key) ID_Mahasiswa reference to Mahasiswa(NIM), Biaya	
Bank (Parent)	(Primary Key) KodeBank, (Alternate Key) NamaBank	Terintegrasi
Transaksi (Child)	(Primary Key) KodeTransaksi, (Foreign Key) Id_Staff reference to Staff(IDStaff), (Foreign Key) ID_Mahasiswa reference to Mahasiswa(NIM), Biaya (Foreign Key) KodeBank reference to Bank(KodeBank)	
MataKuliah (Parent)	(Primary Key) KodeMataKuliah, (Alternate Key) NamaMataKuliah	Berupa
Jadwal (Child)	(Primary Key) KodeJadwal, (Foreign Key) Id_MataKuliah reference to MataKuliah(Id_MataKuliah), Waktu, Sesi, (Alternate Key) Lokasi	
Dosen (Parent)	(Primary Key) IDDosen, NamaDosen, NamaDepan,	Memiliki

Jadwal (Child)	NamaBlkng, GenderDosen, AlamatDosen, DOB, (Alternate Key) Email (Primary Key) KodeJadwal, (Foreign Key) Id_Dosen reference to Dosen(Id_Dosen), Waktu, Sesi, (Alternate Key) Lokasi	
Staff (Parent)	(Primary Key) IDStaff, NamaStaff, NamaDepan, NamaBlkng, PosisiStaff, GenderStaff, DOB, Alamat, (Alternate Key) Email	Mengatur
Transaksi (Child)	(Primary Key) KodeTransaksi, (Foreign Key) Id_Staff reference to Staff(IDStaff), Biaya	
Staff (Parent)	(Primary Key) IDStaff, NamaStaff, NamaDepan, NamaBlkng, PosisiStaff, GenderStaff, DOB, Alamat, (Alternate Key) Email	Mengatur
Jadwal (Child)	(Primary Key) KodeJadwal, (Foreign Key) Id_Staff reference to Staff(IDStaff), Waktu, Sesi, (Alternate Key) Lokasi	
Jurusan (Parent)	(Primary Key) KodeJurusan, (Alternate Key) NamaJurusan	Memilih
Mahasiswa (Child)	(Primary Key) NIM, (Foreign Key) KodeJurusan	

	reference to Jurusan (KodeJurusan), Nama NamaDepan, NamaBlkng, TempatLahir, TanggalLahir, Gender, Alamat, GPA, (Alternate Key) Email	
--	---	--

2.1.4 One to One Binary Relationship Mandatory Participation on One Side

Entity	Atribut	Relationship
Manager (Parent) Fakultas (Child)	(Primary Key) Id_Manager, Tugas (Primary Key) Id_Fakultas, (Foreign key) KodeManager references Manager(Id_Manager), Nama Fakultas	Mengatur
Manager (Parent) Cabang (Child)	(Primary Key) Id_Manager, Tugas (Primary Key) KodeCabang, (Foreign key) KodeManager references Manager(Id_Manager), NamaCabang, AlamatCabang	Mengatur

2.1.5 One to One Recursive Relationship

Pada data model kami tidak memiliki one to one Relationship.

2.1.6 Superclass/Subclass Relationship Types

Option	Relationship
--------	--------------

<p>Mandatory, Disjoint User = Superclass Mahasiswa, Dosen, Staff =Subclass</p>	<p>Dosen :</p> <ul style="list-style-type: none"> - Id_User - NamaDosen <ul style="list-style-type: none"> NamaDepan NamaBlkng - GenderDosen - AlamatDosen - DOB - Email <p>Primary Key : Id_User</p> <p>Staff :</p> <ul style="list-style-type: none"> - Id_User - NamaStaff - NamaDepan - NamaBlkng - PosisiStaff - GenderStaff - DOB - Alamat - Email <p>Primary Key : Id_User</p> <p>Mahasiswa :</p> <ul style="list-style-type: none"> - Id_User - Nama - NamaDepan - NamaBlkng - TempatLahir - TanggalLahir - Gender - Alamat - GPA - Email <p>Primary Key : Id_User</p>
<p>Mandatory, Disjoint Staff = Superclass PekerjaTetap, ParuhWaktu =Subclass</p>	<p>Pekerja_Tetap :</p> <ul style="list-style-type: none"> - Id_Staff - Bonus - Gaji_Pokok - Jatah_Cuti <p>Primary Key : Id_Staff</p> <p>Paruh_Waktu :</p> <ul style="list-style-type: none"> - Id_Staff - GajiPerJam

	Primary Key : Id_Staff
Optional, Nondisjoint Staff = Superclass Manager, Sekretaris, Marketing, ADM_Keuangan, ADM_MataKuliah, ADM_Kemahasiswaan, ADM_Dosen = Subclass	Staff : <ul style="list-style-type: none"> - Id_Staff - NamaStaff - GenderStaff - DOB - Alamat - Email Primary Key : Id_Staff StaffDetail : <ul style="list-style-type: none"> - Id_Staff - PosisiStaff - Gaji - NamaDepan - NamaBlkng - TanggungJawab Primary Key : Id_Staff Foreign Key : Id_Staff references Staff(Id_Staff)

2.1.7 Many to Many (*:*) Binary Relationship Types

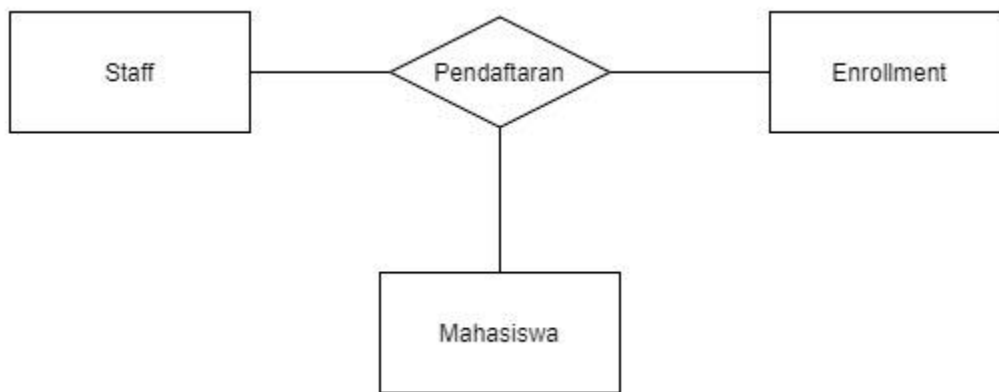
Mahasiswa - Jadwal

Mengambil (Id_Mahasiswa, Id_Jadwal, comment)

Entity	Atribut	Relationship
Mahasiswa (Parent)	(Primary Key) NIM, Nama NamaDepan, NamaBlkng, TempatLahir, TanggalLahir, Gender, Alamat, GPA, (Alternate Key) Email	Request
Mengambill (Child)	(Primary Key, Foreign Key) Id_Mahasiswa references Mahasiswa(NIM),	

	Komentar	
Jadwal (Parent)	(Primary Key) KodeJadwal, Waktu, Sesi, (Alternate Key) Lokasi	Berupa
Mengambil (Child)	(Primary Key, Foreign key) ID_Jadwal references Jadwal(ID_Jadwal), Komentar	

2.1.8 Complex Relationship Types



Staff mendaftarkan mahasiswa untuk enrollment

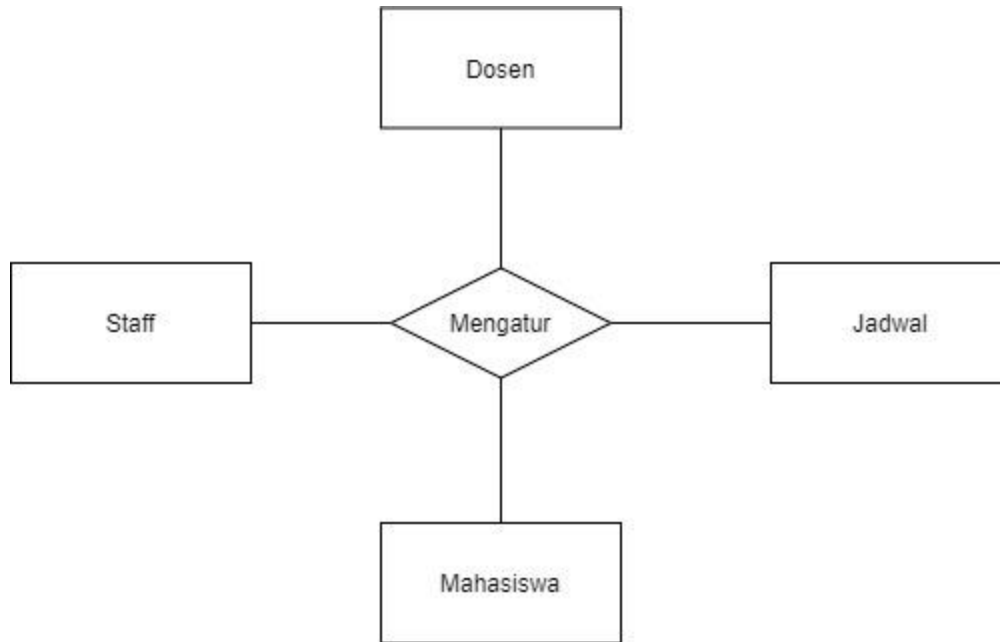
Pendaftaran(Id_Mahasiswa, Id_Enrollment, Id_Staff, tanggal)

Primary Key : Id_Enrollment

Foreign Key : Id_Mahasiswa **references** Mahasiswa(Id_Mahasiswa)

Foreign Key : Id_Enrollment **references** Enrollment(Id_Enrollment)

Foreign Key : Id_Staff **references** Staff(Id_Staff)



Staff Mengatur Jadwal untuk Dosen dan Mahasiswa

Mengatur(Id_Mahasiswa, Id_Dosen, Id_Jadwal, Id_Staff, tanggal)

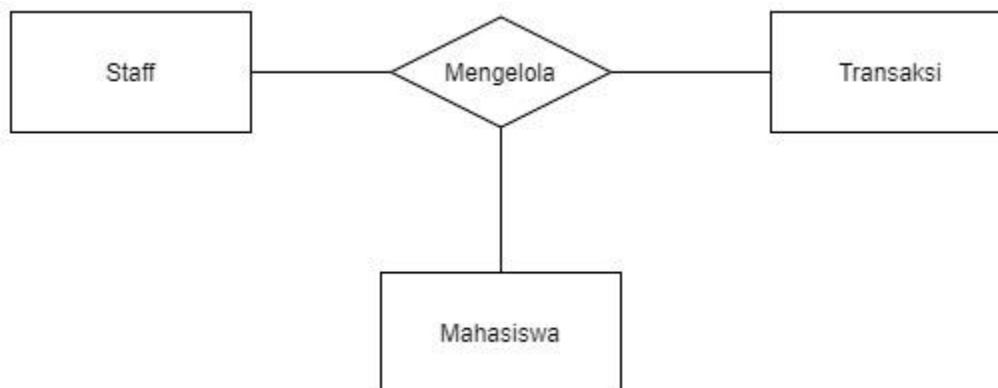
Primary Key : Id_Mahasiswa, Id_Jadwal

Foreign Key : Id_Mahasiswa **references** Mahasiswa(Id_Mahasiswa)

Foreign Key : Id_Dosen **references** Dosen(Id_Dosen)

Foreign Key : Id_Staff **references** Staff(Id_Staff)

Foreign Key : Id_Jadwal **references** Jadwal(Id_Jadwal)



Staff mengelola Transaksi yang dilakukan Mahasiswa

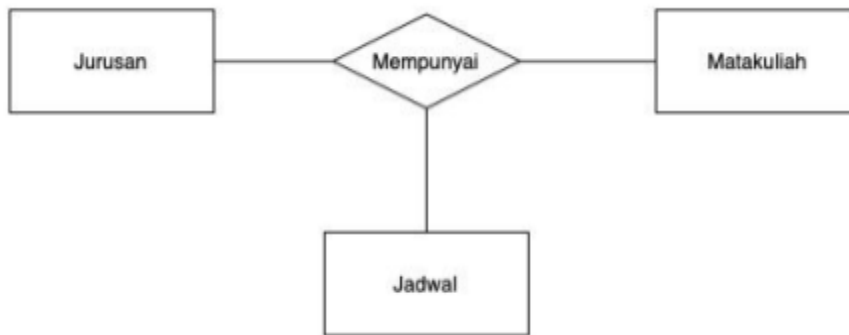
Mengelola(Id_Staff, Id_Transaksi, Id_Mahasiswa, tanggal)

Primary Key : Id_Staff, Id_Transaksi

Foreign Key : Id_Mahasiswa **references** Mahasiswa(Id_Mahasiswa)

Foreign Key : Id_Transaksi **references** Transaksi(Id_Transaksi)

Foreign Key : Id_Staff **references** Staff(Id_Staff)



Jurusan mempunyai Matakuliah yang berupa Jadwal

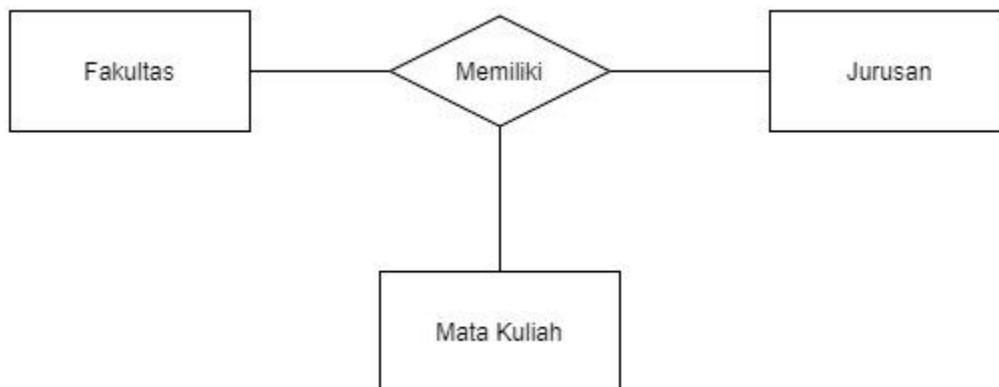
Mempunyai(KodeJurusan, KodeMataKuliah, KodeJadwal)

Primary key : KodeJurusan, KodeMataKuliah, KodeJadwal

Foreign key : KodeJadwal **references** Jadwal(Id_Jadwal)

Foreign key : KodeMataKuliah **references** Matakuliah(Id_MataKuliah)

Foreign key : KodeJurusan **references** Jurusan(Id_Jurusan)



Fakultas mempunyai jurusan, jurusan memiliki mata kuliah.

Memiliki (Id_Fakultas, KodeJurusan, KodeMataKuliah)

Primary Key : Id_Fakultas, KodeJurusan, KodeMataKuliah

Foreign Key : KodeMataKuliah **references** MataKuliah(Id_MataKuliah)

Foreign Key : KodeJurusan **references** Jurusan(Id_Jurusan)

Foreign Key : Id_Fakultas **references** Fakultas(Id_Fakultas)

2.1.9 Multi-valued Attributes

Entity	Detail Entity
MahasiswaDetailAddress	MahasiswaDetailAddress :

	<ul style="list-style-type: none"> - (Primary Key) Id_Address - (Foreign Key) Id_Mahasiswa references Mahasiswa(Id_Mahasiswa) - Address
DosenAddress	MahasiswaDetailAddress : <ul style="list-style-type: none"> - (Primary Key) Id_Address - (Foreign Key) Id_Dosen references Dosen(Id_Dosen) - Address
StaffAddress	StaffAddress : <ul style="list-style-type: none"> - (Primary Key) Id_Address - (Foreign Key) Id_Staff references Staff(Id_Staff) - Address
PosisiStaff	PosisiStaff : <ul style="list-style-type: none"> - (Primary Key)(Foreign Key) Id_Staff references Staff(Id_Staff) - (Primary Key)(Foreign Key) ID references Posisi(ID_Posisi) Posisi: <ul style="list-style-type: none"> - (Primary Key) ID_Posisi - DetailPosisi

2.2 Validate Relations Using Normalization

PT. BANK PAULA SEJAHTERA
RESI BUKTI BAYAR



=====

KODE TRANSAKSI : KT-098878
TANGGAL TRANSAKSI : 12 Maret 2020
NAMA : Fauza Wahidira
NIM : 2301941990
ID PRODI - NAMA PRODI : SOCS - Computer Science
JENIS TRANSAKSI : Autodebet
TAHUN AJARAN : 2020, Even Semester

=====

No	Nomor Pembayaran	Jenis Pembayaran	Biaya (Rp)
1	NP-001	Fixed Tuition Fee	2.500.000
2	NP-002	Fixed Tuition Fee	2.500.000
3	NP-003	Lab Fee	1.500.000
4	NP-005	Variable Tuition Fee	1.000.000
5	NP-006	Textbook Fee	500.000
6	NP-007	Variable Tuition Fee	1.000.000
Total Biaya			9.000.000

TRANSAKSI SUKSES, RESI INI ADALAH BUKTI PEMBAYARAN YANG SAH.

**Dokumen ini dibuat secara otomatis oleh sistem komputer sebagai bukti pembayaran.

Staff

SF001 - Arul

UNF
Transaksi

{Kode_Transaksi, Tanggal_Transaksi, Nama_Mahasiswa, NIM_Mahasiswa, Id_Prodi, Nama_Prodi, Jenis_Transaksi, TahunAjaran_Mahasiswa, Semester_Mahasiswa, Nomor_Pembayaran, Jenis_Pembayaran, Biaya, Nama_Staff, Id_Staff, Total_Biaya_Transaksi}

1NF

Transaksi

{Kode_Transaksi(PK), Tanggal_Transaksi, Nama_Mahasiswa, NIM_Mahasiswa, Id_Prodi, Nama_Prodi, Jenis_Transaksi, TahunAjaran_Mahasiswa, Semester_Mahasiswa, Nama_Staff, Id_Staff}

Detail_Transaksi

{Kode_Transaksi(PK)(FK), Nomor_Pembayaran(PK), Jenis_Pembayaran, Biaya}

2NF

Transaksi

{Kode_Transaksi(PK), Tanggal_Transaksi, Nama_Mahasiswa, NIM_Mahasiswa, Id_Prodi, Nama_Prodi, Jenis_Transaksi, TahunAjaran_Mahasiswa, Semester_Mahasiswa, Nama_Staff, Id_Staff}

Detail_Transaksi

{Kode_Transaksi(PK)(FK), Nomor_Pembayaran(PK)(FK), Biaya}

Pembayaran

{Nomor_Pembayaran(PK), Jenis_Pembayaran}

3NF

Transaksi

{Kode_Transaksi(PK), Tanggal_Transaksi, Jenis_Transaksi, NIM_Mahasiswa(FK), Id_Staff(FK)}

Detail_Transaksi

{Kode_Transaksi(PK)(FK), Nomor_Pembayaran(PK)(FK), Biaya}

Pembayaran

{Nomor_Pembayaran(PK), Jenis_Pembayaran}

Mahasiswa

{NIM_Mahasiswa(PK), Nama_Mahasiswa, Id_Prodi(FK), TahunAjaran_Mahasiswa, Semester_Mahasiswa}

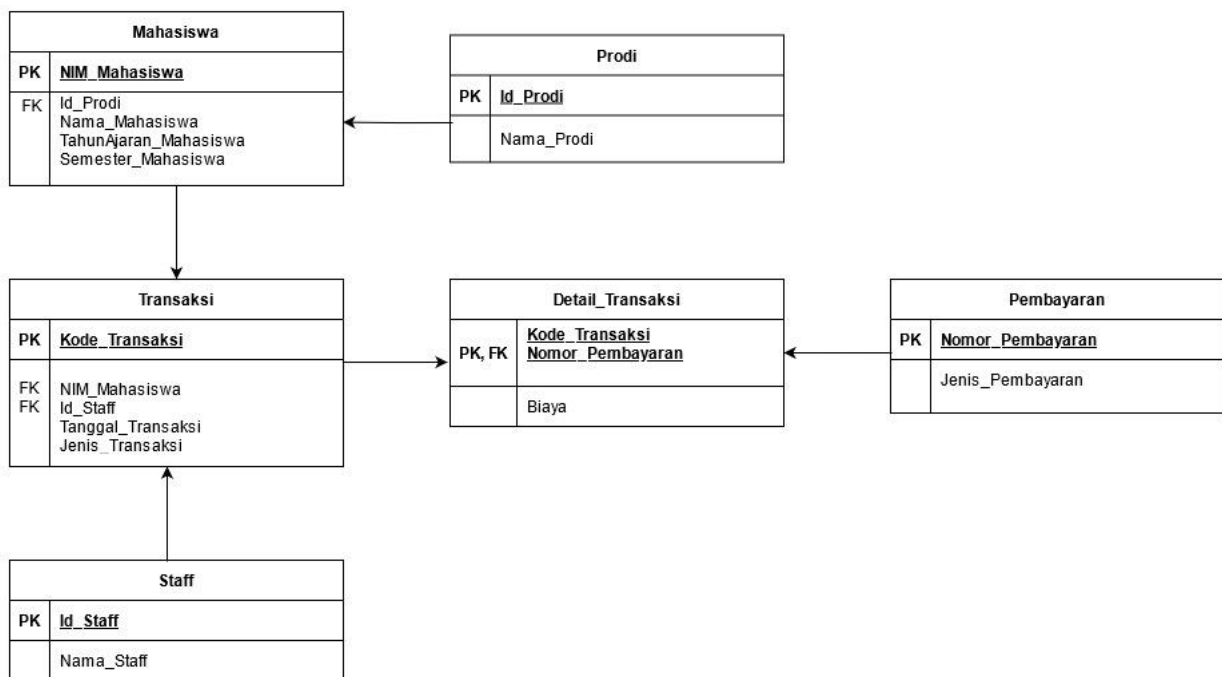
Prodi

{Id_Prodi(PK), Nama_Prodi}

Staff

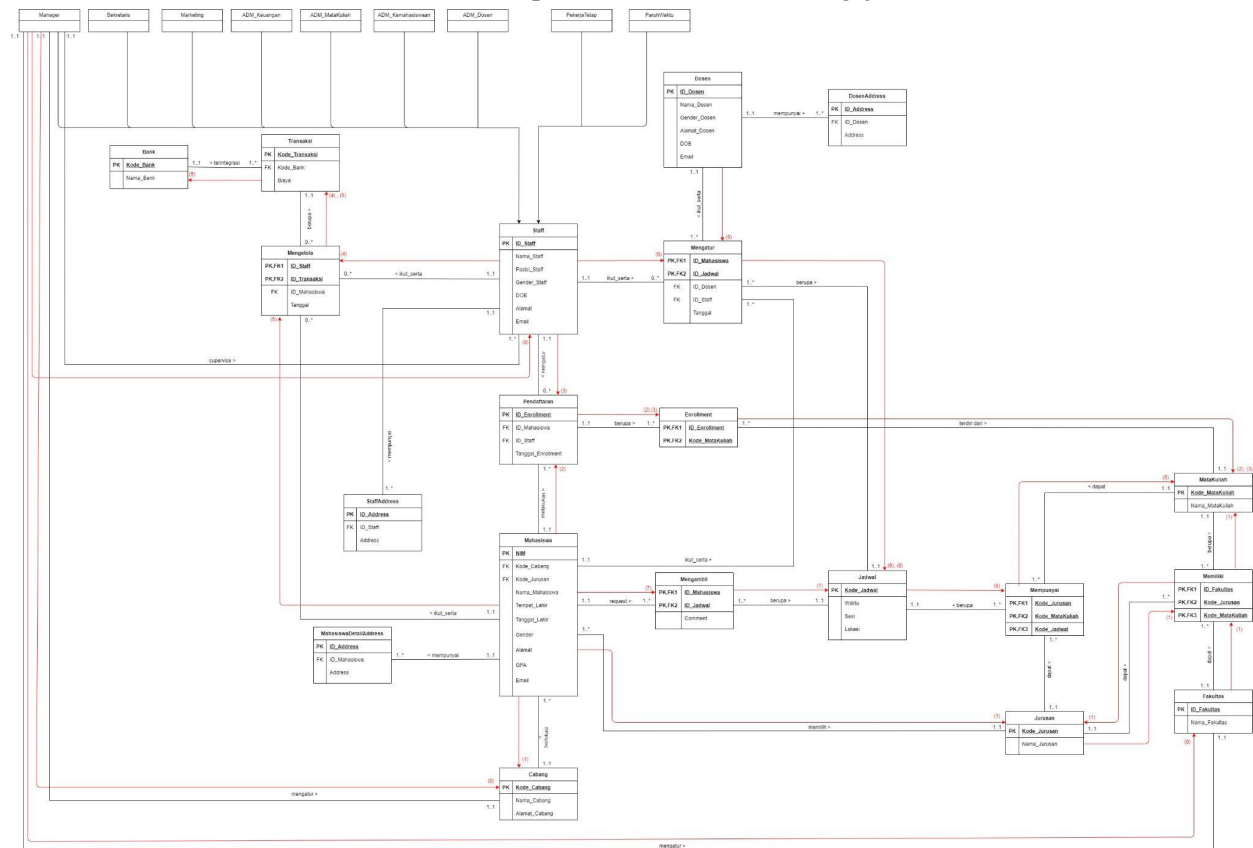
{Id_Staff(PK), Nama_Staff}

ERD



2.3 Validate Relations Against User Transactions

Sebelum mahasiswa melakukan aktivitas transaksi, **Manager** mengatur **Staff** dan **Cabang** dan juga **Manager** mengatur divisi-divisi staff pada **Cabang** maupun **Fakultas** yang tersedia. Saat pertama kali **Mahasiswa** masuk ke sebuah universitas, **Mahasiswa** terlebih dahulu memilih **Fakultas** yang terdiri dari berbagai **Jurusan** yang mempunyai **Mata Kuliah** dan berlokasi di **Cabang** yang dipilih. Kemudian, **Mahasiswa** melakukan pendaftaran **Enrollment** untuk mendapatkan **Mata Kuliah** yang selanjutnya pendaftaran tersebut diatur oleh **Staff**. Selanjutnya **Staff** mengurus **Transaksi** mahasiswa yang kemudian **Mahasiswa** akan melakukan **Transaksi** yang terintegrasi oleh **Bank** tertentu. **Staff** kemudian mengurus **Jadwal** mahasiswa yang berupa jadwal **Mata Kuliah**. Setelah itu, **Mahasiswa** mendapatkan **Jadwal** mata kuliah mereka. **Dosen** kemudian mendapatkan **Jadwal** untuk mengajar.



2.4 Check Integrity Constraints

Entity	Atribut
Cabang	Cabang (KodeCabang, NamaCabang,

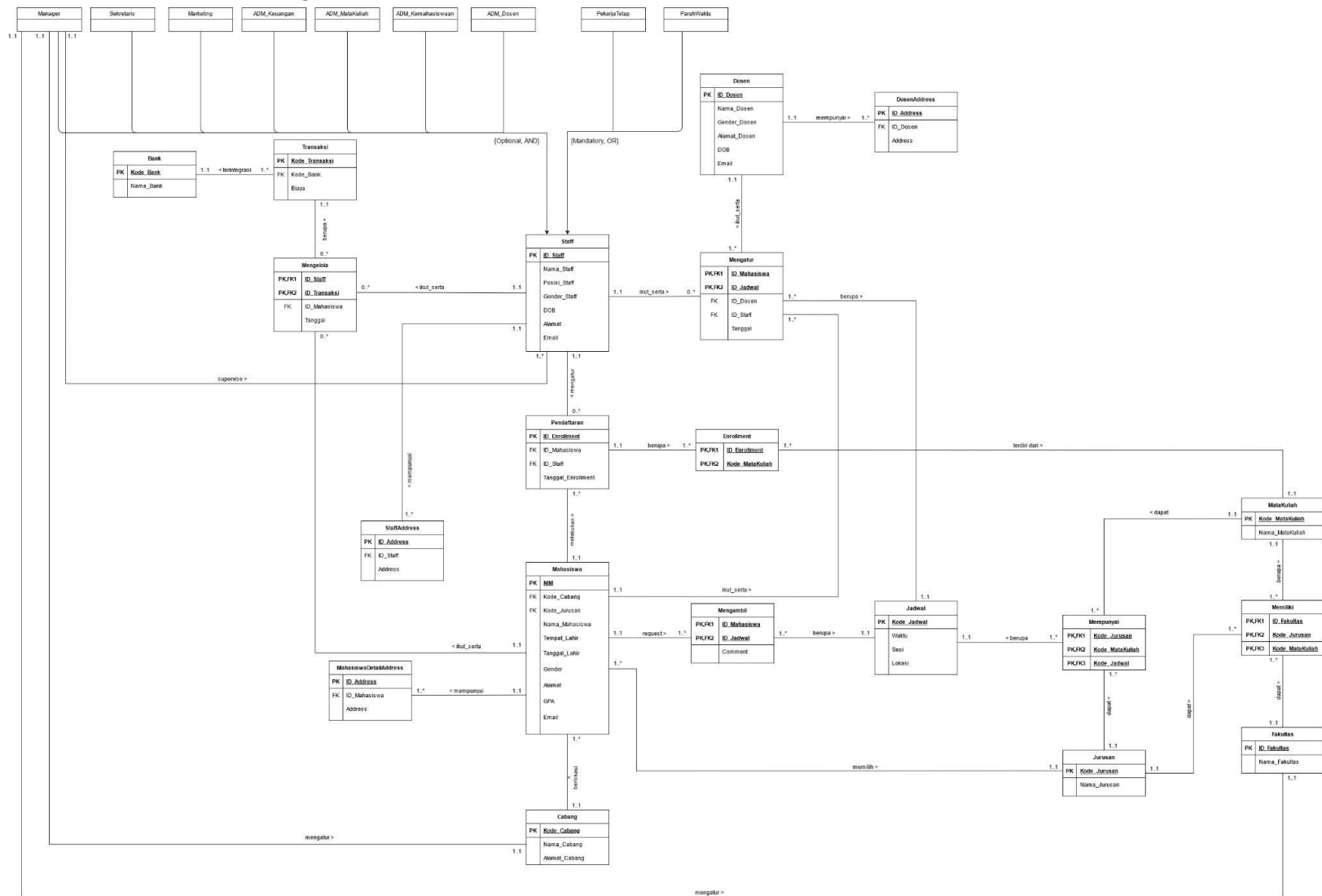
	AlamatCabang) Primary Key KodeCabang Alternate Key NamaCabang, AlamatCabang
Fakultas	Fakultas (IDFakultas, NamaFakultas) Primary Key IDFakultas Alternate Key NamaFakultas
Bank	Bank (KodeBank, NamaBank) Primary Key KodeBank Alternate Key NamaBank
Dosen	Dosen (NamaDosen, NamaDepan, NamaBlkng, GenderDosen, AlamatDosen, DOB, Email) Primary Key IDDosen Alternate Key Email
Staff	Staff (NamaStaff, NamaDepan, NamaBlkng, PosisiStaff, GenderStaff, DOB, Alamat, Email) Primary Key IDStaff, Alternate Key Email
Mahasiswa	Mahasiswa (NIM, KodeCabang, KodeJadwal, KodeJurusan, Nama, NamaDepan, NamaBlkng, TempatLahir, TanggalLahir, Gender, Alamat, GPA, Email) Primary Key NIM Foreign Key KodeCabang references Cabang(KodeCabang) ON UPDATE CASCADE ON DELETE SET NULL Foreign Key KodeJadwal references Jadwal(KodeJadwal) ON UPDATE CASCADE ON DELETE SET NULL Foreign Key KodeJurusan references Jurusan(KodeJurusan) ON UPDATE CASCADE ON DELETE SET NULL Alternate Key Email
Enrollment	Enrollment (IDEnrollment, Id_Mahasiswa, KodeMataKuliah, TanggalEnrollment) Primary Key IDEnrollment Foreign Key Id_Mahasiswa references Mahasiswa(NIM) ON UPDATE CASCADE ON DELETE SET NULL Foreign Key KodeMataKuliah references MataKuliah (KodeMataKuliah) ON UPDATE CASCADE ON DELETE SET NULL
Jadwal	Jadwal (KodeJadwal, Id_Staff, Id_Dosen, Id_MataKuliah, Waktu, Sesi, Lokasi)

	Primary Key KodeJadwal Foreign Key Id_Staff references Staff(IDStaff) ON UPDATE CASCADE ON DELETE SET NULL Foreign Key ID_Dosen references Dosen(Id_Dosen) ON UPDATE CASCADE ON DELETE SET NULL Foreign Key Id_MataKuliah references MataKuliah(ID_MataKuliah) ON UPDATE CASCADE ON DELETE SET NULL Alternate Key Sesi, Lokasi
Transaksi	Transaksi (Kode Transaksi, Id_Staff, ID_Mahasiswa, Biaya) Primary Key KodeTransaksi, Foreign Key Id_Staff references Staff(IDStaff) ON UPDATE CASCADE ON DELETE SET NULL Foreign Key ID_Mahasiswa references Mahasiswa(NIM) ON UPDATE CASCADE ON DELETE SET NULL Foreign Key KodeBank references Bank(KodeBank) ON UPDATE CASCADE ON DELETE SET NULL
Mata Kuliah	Mata Kuliah (KodeMataKuliah, ID_Matkul, NamaMataKuliah) Primary Key KodeMataKuliah Foreign Key ID_Matkul references Jurusan(KodeJurusan) ON UPDATE CASCADE ON DELETE SET NULL Alternate Key NamaMataKuliah
Jurusan	Jurusan (KodeJurusan, IDFakultas, NamaJurusan) Primary Key KodeJurusan Foreign Key IDFakultas references Fakultas (IDFakultas) ON UPDATE CASCADE ON DELETE SET NULL Alternate Key NamaJurusan

2.5 Review Logical Data Model With User

Logical data model yang telah dibuat sudah cukup bagus untuk merepresentasikan kebutuhan data pada sebuah universitas. Pada atribut yang terdapat dalam entitas sudah cukup mewakili inti dari entitas yang dibutuhkan pada Universitas serta penamaan entitas dan atribut yang konsisten sehingga mudah untuk dipahami. Alur data model cukup sesuai dengan syarat dan ketentuan Universitas.

2.6 Merge Logical Data Models Into Global Model (Optional)



Physical Design

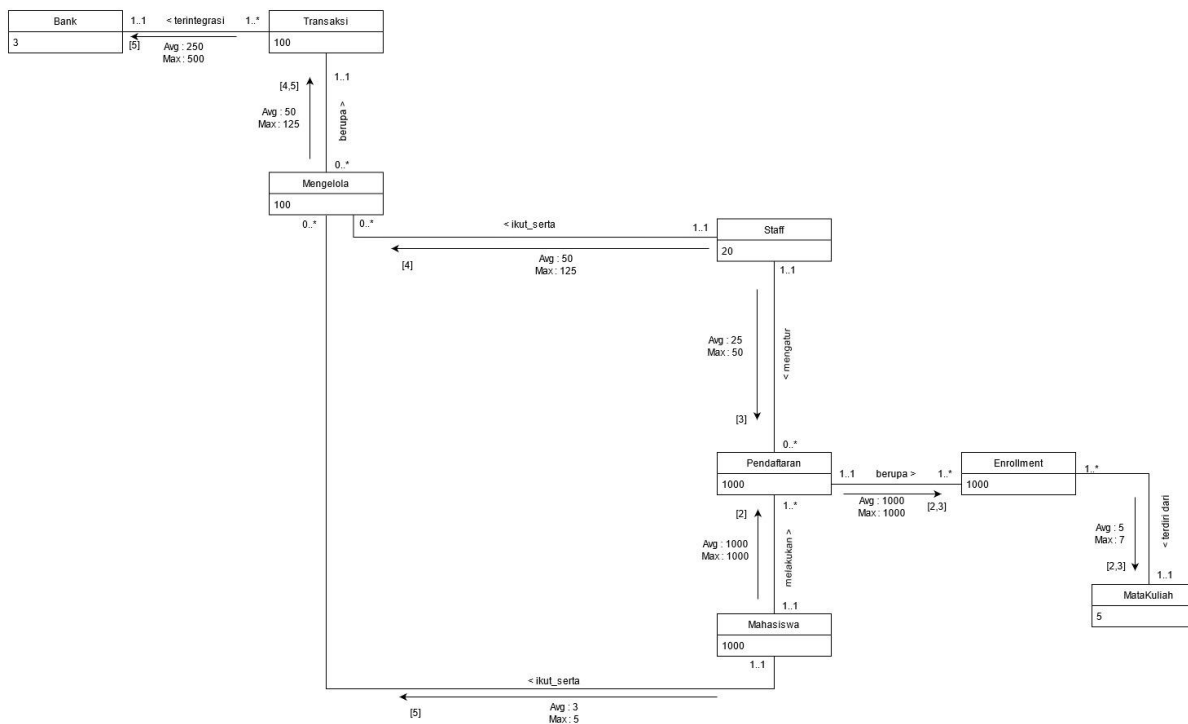
Cross-Referencing Transactions

Transaction/ Relation	[1]				[2]				[3]				[4]				[5]				[6]				[7]				[8]				[9]			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D				
Bank																	x	x	x																	
Transaksi													x	x	x		x	x	x																	
Staff									x	x	x	x		x	x	x					x	x	x	x												
Mahasiswa	x	x	x		x	x											x	x							x											
Cabang	x	x	x																																	
Fakultas	x	x	x																																	
Dosen																														x						
Enrollment					x	x	x	x	x	x	x	x																								
Jadwal																					x		x	x		x			x	x	x					
Mata Kuliah		x			x				x												x															
Jurusan	x	x																																		
Mengelola													x	x		x																				
Pendaftaran					x	x																														
StaffAddresses																																x		x	x	
Mahasiswa Detail Address																																x		x	x	
DosenAddress																																x		x	x	
Mengambil																									x											
Mengatur																					x		x	x					x							
Mempunyai																						x														
Memiliki		x																																		

- Pada tabel diatas, terdapat 9 transaksi dan relasi yang dilakukan antar entity seperti hubungan antara Manager untuk mengatur Cabang, Fakultas, dan Staff pada transaksi ke [0].

- Selanjutnya adalah hubungan antara Mahasiswa, Cabang, Jurusan, Memiliki, serta MataKuliah pada saat awal sebelum dilakukannya transaksi [1].
- Berikutnya, hubungan antara Mahasiswa, Pendaftaran, Enrollment, dan MataKuliah pada transaksi ke [2].
- Pada transaksi ke [3] merupakan hubungan antara Staff, Pendaftaran, Enrollment, dan MataKuliah.
- Transaksi ke [4] merupakan hubungan antara Staff, Mengelola, dan Transaksi.
- Pada transaksi ke [5] merupakan hubungan antara Mahasiswa, Mengelola, Transaksi, dan Bank.
- Transaksi ke [6] merupakan hubungan antara Staff, Mengatur, Jadwal, Mempunyai, dan MataKuliah.
- Transaksi ke [7] merupakan hubungan antara Mahasiswa, Mengambil, Jadwal.
- Transaksi ke [8] merupakan hubungan antara Dosen, Mengatur, dan Jadwal.
- Transaksi terakhir [9] merupakan hubungan antara Staff dengan StaffAddress, Dosen dengan DosenAddress, dan Mahasiswa dengan MahasiswaDetailAddress.

Relations Transaction Usage Map



Pada saat pendaftaran enrollment selama kurang lebih 1 minggu, terdapat 1000 mahasiswa dimana setiap mahasiswa melakukan setidaknya 1 pendaftaran. Lalu di universitas tersebut, terdapat 20 staff yang menangani pendaftaran enrollment yang dilakukan oleh mahasiswa. Setiap

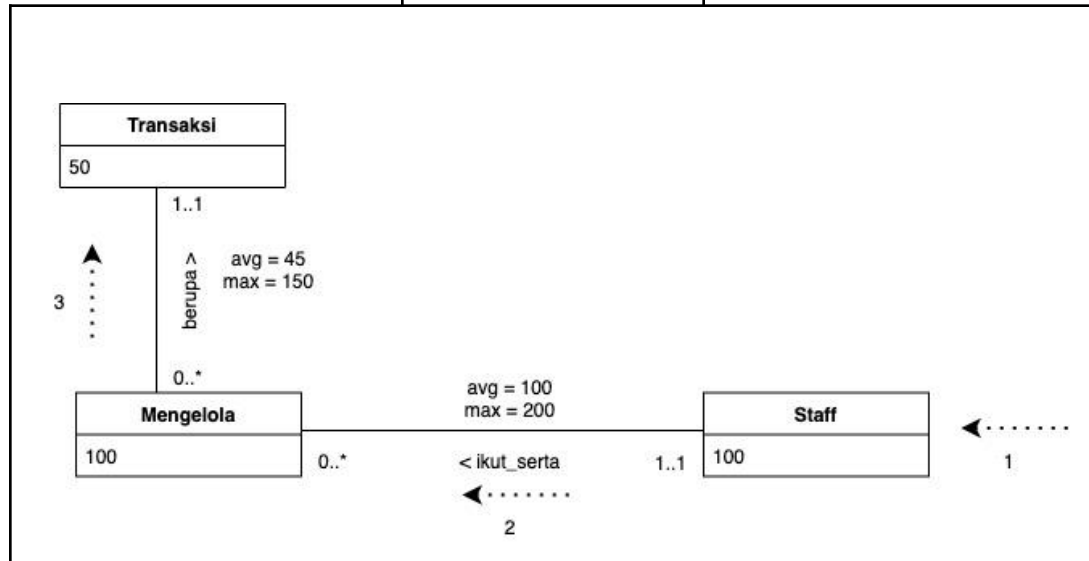
staff setidaknya menangani 25 pendaftaran dengan maksimal 50 pendaftaran. Untuk menyelesaikan pendaftaran enrollment, mahasiswa harus memenuhi biaya yang telah ditetapkan. Mahasiswa dapat melakukan maksimal 5 transaksi dengan rata-rata transaksi yang dilakukan sebanyak 3 transaksi yang nantinya akan dikelola oleh staff dimana setiap staff ikut serta mengelola transaksi dengan average 50 dan maksimal sebanyak 125 transaksi. Kemudian setiap transaksi nantinya akan terintegrasi dengan salah satu dari 3 bank dengan average 250 dan maksimal 500. Setelah transaksi telah berhasil dan dikonfirmasi oleh staff, nantinya staff akan melanjutkan pendaftaran enrollment mahasiswa dimana setiap enrollment yang dilakukan mahasiswa akan terdiri dari mata kuliah yang dipilih dengan average 5 dan maksimal sebanyak 7 mata kuliah.

Transaction Analysis Form

Pada form ini, jumlah transaksi yang terjadi yaitu sebanyak 50 transaksi dalam satu jam dengan average transaksi 45 dan maksimal transaksi yang terjadi yaitu sebanyak 100 transaksi per jam yang mana masing-masing transaksinya dikelola oleh staff yang bekerja pada universitas tersebut. Terdapat sebanyak 100 staff yang mengelola transaksi yang terjadi dalam satu jam dan tiap staff dapat mengurus/mengelola 1 transaksi atau lebih banyak dengan average pengelolaan sebanyak 100 transaksi dalam satu jam dan maksimal sebanyak 200 jumlah transaksi yang dapat dikelola oleh staff per jam.

Transaction Analysis Form			
9 Juni 2021			
Transaction		Menampilkan staff dan jumlah transaksi yang dikelola tiap bulan	
Transaction Volume			
	Average	45	
	Peak	150	
SELECT DATEPART(MM, Tanggal) as 'Bulan', Id_Staff, NamaStaff, COUNT(Kode_Transaksi) as 'Jumlah Transaksi' FROM Staff s JOIN Mengelola m ON s.Id_Staff = m.Id_Staff JOIN Transaksi t ON m.ID_Transaksi = t.Kode_Transaksi GROUP BY DATEPART(MM, Tanggal), Id_Staff, NamaStaff ORDER BY DATEPART(MM, Tanggal)	Predicate:		-
	Join attributes:		s.Id_Staff = m.Id_Staff, m.ID_Transaksi = t.Kode_Transaksi
	Ordering attribut:		DATEPART(MM, Tanggal)

	Grouping attribut:	DATEPART(MM, Tanggal), Id_Staff, NamaStaff
	Built in function	DATEPART, COUNT
	Attribute updated:	-

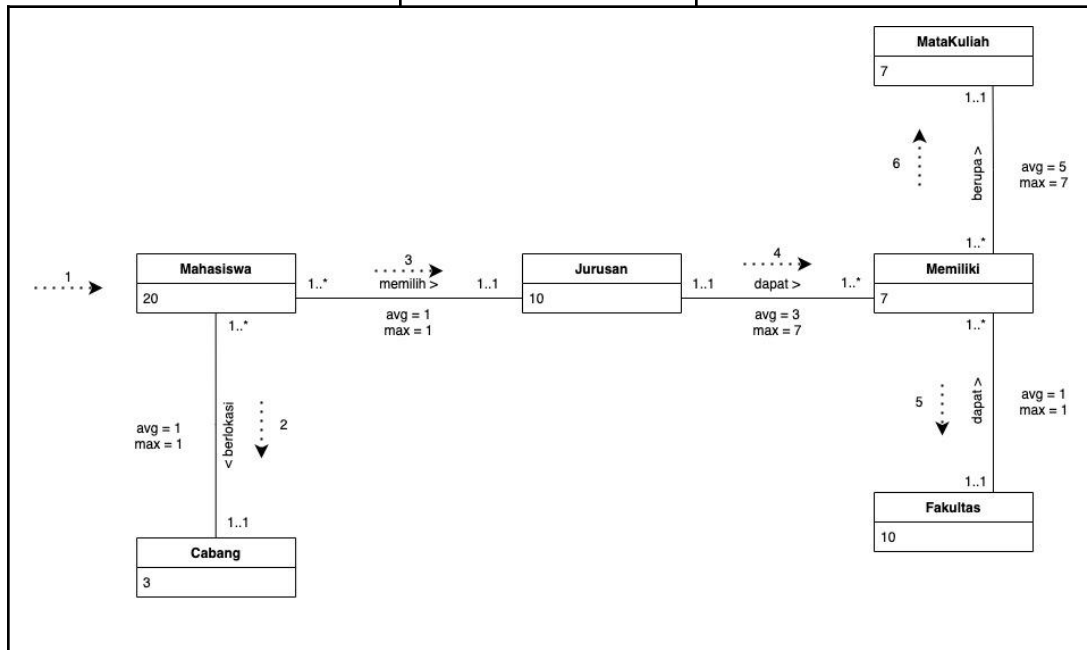


Access	Entity	Type of Access	No. of References		
			Per. Transaction	Avg Per Hour	Peak Per Hour
1	Staff	R	25	100	200
2	Mengelola	R	30	100	200
3	Transaksi	I, R, U	50	50	100
Total References			105	250	500

Pada form ini, terdapat 20 mahasiswa yang berlokasi di salah satu cabang dari universitas dimana terdapat 3 cabang dan setiap mahasiswa hanya dapat memilih lokasi cabang maksimal 1. Setiap mahasiswa juga dapat memilih jurusan dengan average 1 dan maksimal mahasiswa memilih jurusan yaitu sebanyak 1. Disini, terdapat 10 jurusan yang mana masing-masing jurusan dapat memiliki jumlah mata kuliah paling banyak yaitu 7 dengan average 3 mata kuliah. Setiap jurusan yang ada termasuk kedalam salah satu fakultas yang tersedia pada universitas.

Transaction Analysis Form	
9 Juni 2021	
Transaction	Melihat jumlah mahasiswa yang memiliki GPA lebih dari 3.00 berdasarkan cabang, jurusan, dan mata kuliah.

Transaction Volume		
	Average	109
	Peak	170
SELECT NamaCabang, COUNT(NIM) as 'Total Mahasiswa' FROM Mahasiswa m JOIN Cabang c ON c.Kode_Cabang = m.Kode_Cabang JOIN Jurusan j ON WHERE m.GPA > 3.00 GROUP BY c>NamaCabang, j.Kode_Jurusan, mk.Kode_MataKuliah ORDER BY COUNT(NIM) DESC	Predicate:	-
	Join attributes:	c.Kode_Cabang = m.Kode_Cabang, j.Kode_Jurusan = m.Kode_Jurusan, mk.Kode_Jurusan = j.Kode_Jurusan, matkul.Kode_MataKuliah = mk.Kode_MataKuliah, f.ID_Fakultas = mk.ID_Fakultas
	Ordering attribut:	COUNT(NIM) DESC
	Grouping attribut:	c>NamaCabang, j.Kode_Jurusan, mk.Kode_MataKuliah
	Built in function	COUNT
	Attribute updated:	-

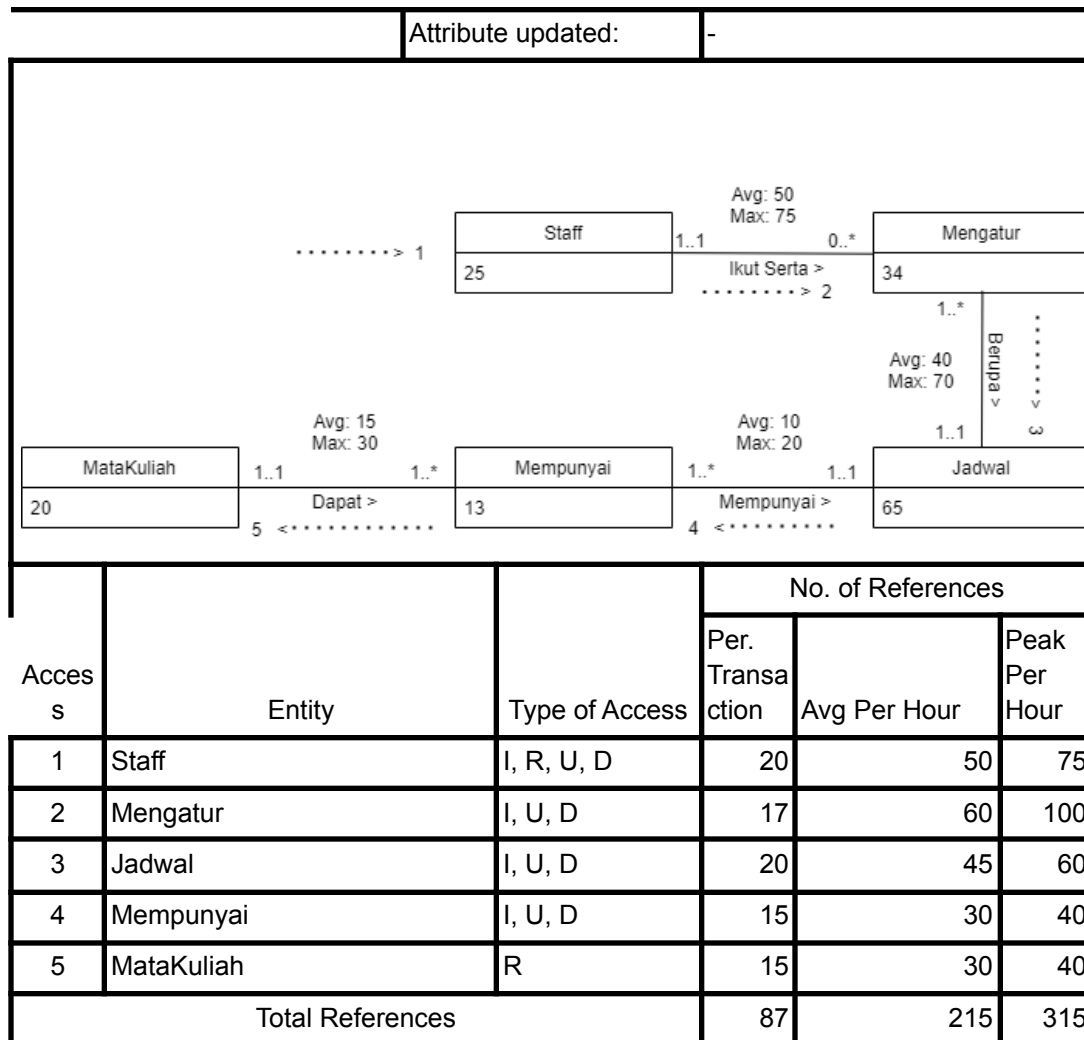


Access	Entity	Type of Access	No. of References		
			Per. Transaction	Avg Per Hour	Peak Per Hour

1	Mahasiswa	R, U	10	15	30
2	MataKuliah	R	7	18	25
3	Memiliki	R	7	16	30
4	Jurusan	R	10	30	35
5	Fakultas	R	10	20	30
6	Cabang	I, R, U	3	10	20
Total References			47	109	170

Pada form ini, jumlah jadwal yang diatur oleh staff adalah 20 jadwal per jam dengan rata-rata 50 dan maksimalnya adalah 75. Jadwal-jadwal tersebut diatur oleh 20 staff dengan rata-rata mengatur 40 jadwal per jam dan maksimalnya adalah 60 jadwal perjam. lalu setiap matakuliah dibagi kedalam jadwal dengan rata-rata 15 jadwal per jam dengan maksimal dibaginya 30 per jam. Setiap matakuliah pasti mempunyai jadwal dengan rata-rata 10 jadwal perjam dan maksimalnya 20 per jam.

Transaction Analysis Form		
9 Juni 2021		
Transaction	melihat jadwal dengan namaMataKuliah dan NamaJurusan yang diatur oleh Staff dengan kode 'ST28030125'	
Transaction Volume		
	Average	211
	Peak	305
select NamaMataKuliah, NamaJurusan from MataKuliah mk join Mempunyai mp on mk.Kode_MataKuliah = mp.Kode_MataKuliah join Jadwal j on mp. Kode_Jadwal = j.Kode_Jadwal join Mengatur mt on mt. ID_Jadwal = j.ID_Jadwal join Staff s on s.ID_Staff = mt.ID_Staff where ID_Staff = 'ST28030125' group by NamaMataKuliah, NamaJurusan	Predicate:	-
	Join attributes:	mk.Kode_MataKuliah = mp.Kode_MataKuliah, mp. Kode_Jadwal = j.Kode_Jadwal, mt. ID_Jadwal = j.ID_Jadwal, s.ID_Staff = mt.ID_Staff
	Ordering attribut:	-
	Grouping attribut:	NamaMataKuliah, NamaJurusan
	Built in function	-



File Organization

File organization yang akan dipilih adalah B+ Tree File Organization yang akan diimplementasikan pada SQL Server. B+ Tree File Organization merupakan salah satu metode lanjutan dari metode untuk mengakses data sekuensial dengan menggunakan indeks. Pada B+ Tree File Organization, strukturnya menyerupai B-Tree dimana setiap node menyimpan kumpulan record file. Dengan menggunakan konsep indeks kunci dimana primary key akan digunakan untuk mengurutkan file dan setiap primary key tersebut, nilai indeks yang dihasilkan akan dimapping dengan record yang ada.

Ada beberapa kelebihan yang dapat dimanfaatkan dari konsep B+ Tree File Organization yaitu :

- sistem pencarian menjadi lebih mudah karena setiap records hanya tersimpan pada node leaf dan terurut.
- mengunjungi tiap cabang tree menjadi lebih mudah dan cepat.

- ukuran dari B+ Tree tidak ada keharusan, sehingga B+ Tree lebih efisien jika ada penambahan record ataupun pengurangan record data.
- termasuk ke dalam struktur tree yang seimbang karena tiap query baik itu insert, update, ataupun delete tidak akan mempengaruhi kinerja dari B+ Tree.

Ada juga kekurangan dari konsep B+ Tree File Organization yaitu :

- tidak bisa digunakan dalam database yang bersifat statik.

Pada kasus kami, yaitu database universitas dimana akan menyimpan data mahasiswa, staff, dosen, enrollment, transaksi, dan data lainnya. Hal ini akan memudahkan untuk mengakses data-data yang ingin dicari hanya dengan menggunakan primary key dan kemudian pada pencarian di B+ Tree hanya akan mengunjungi jalur yang dimana tempat penyimpanan primary key/indeks tersebut berada. Hal ini tidak hanya membuat pencarian menjadi lebih efisien, tetapi juga penambahan atau pengurangan data pada B+ Tree juga dapat dilakukan secara fleksibel sehingga jika ada data mahasiswa yang bertambah ataupun berkurang, maka secara kinerja B+ Tree tidak akan terpengaruhi akibat operasi insert, update ataupun delete data yang sering dilakukan.

SQL Server akan menyimpan indeks-indeks(primary key) tersebut dalam format B+ Tree. Terdapat 2 jenis halaman pada SQL Server dimana halaman tipe pertama akan berisi node leaf (node paling akhir) dimana merupakan halaman data tunggal. Sedangkan halaman tipe kedua yang bukan merupakan node leaf akan berisi baris data sama seperti pada tipe pertama dan juga pointer untuk menuju ke halaman child berikutnya.

Indexing

Dalam indexing, kami memilih primary index karena setiap entity dan relasi yang ada telah memiliki atribut primary masing-masing. Kemudian, kami juga akan menambahkan beberapa secondary index untuk membuat key tambahan terhadap relasi dan entity yang paling sering digunakan serta atribut yang terlibat ke dalam beberapa operasi selection dan join seperti order by, group by, union, distinct, dan lain-lain.

Denormalization

7.1 Combining 1:1 relationships

Entity	Attributes
Manager 1:1 Fakultas	Fakultas: <ul style="list-style-type: none">- (PK) ID_Fakultas- Nama_Fakultas- ID_Manager- Nama_Manager
Manager 1:1 Cabang	Cabang: <ul style="list-style-type: none">- (PK) Kode_Cabang- Nama_Cabang- Alamat_Cabang- ID_Manager- Nama_Manager

7.2 Duplicating non-key attributes in 1:* relationships: Lookup Table

Entity	Attributes
Bank 1:* Transaksi	Transaksi: <ul style="list-style-type: none">- (PK) Kode_Transaksi- Kode_Bank references Bank (Kode_Bank)- Nama_Bank- Biaya
Manager 1:* Staff	Staff: <ul style="list-style-type: none">- (PK) ID_Staff- Nama_Staff- Posisi_Staff- Gender_Staff- DOB- Alamat- Email- Nama_Manager
Pendaftaran 1:* Enrollment	Enrollment: <ul style="list-style-type: none">- (PK) ID_Enrollment references Pendaftaran (ID_Enrollment)- (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah)

	<ul style="list-style-type: none"> - ID_Mahasiswa - ID_Staff - Tanggal_Enrollment
Dosen 1:* Mengatur	Mengatur: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - ID_Dosen references Dosen(ID_Dosen) - ID_Staff references Staff(ID_Staff) - Tanggal
Dosen 1:* DosenAddress	DosenAddress: <ul style="list-style-type: none"> - (PK) ID_Address references Dosen (ID_Address) - (PK) ID_Dosen references Dosen (ID_Dosen) - Address1 - Nama_Dosen - Gender_Dosen - DOB - Email
Jadwal 1:* Mengatur	Mengatur: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - (PK) Kode_Jadwal references Jadwal (Kode_Jadwal) - ID_Dosen references Dosen(ID_Dosen) - ID_Staff references Staff(ID_Staff) - Tanggal - Waktu - Sesi - Lokasi
Jadwal 1:* Mengambil	Mengambil: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - (PK) Kode_Jadwal references

	Jadwal (Kode_Jadwal) <ul style="list-style-type: none"> - Comment - Waktu - Sesi - Lokasi
Jadwal 1:* Mempunyai	Mempunyai: <ul style="list-style-type: none"> - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - (PK) Kode_Jadwal references Jadwal(Kode_Jadwal) - Waktu - Sesi - Lokasi
MataKuliah 1:* Enrollment	Enrollment: <ul style="list-style-type: none"> - (PK) ID_Enrollment references Pendaftaran (ID_Enrollment) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - Nama_MataKuliah
MataKuliah 1:* Memiliki	Memiliki: <ul style="list-style-type: none"> - (PK) ID_Fakultas references Fakultas (ID_Fakultas) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - Nama_MataKuliah
Matakuliah 1:* Mempunyai	Mempunyai: <ul style="list-style-type: none"> - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - (PK) Kode_Jadwal references Jadwal (Kode_Jadwal) - Nama_MataKuliah
Fakultas 1:* Memiliki	Memiliki: <ul style="list-style-type: none"> - (PK) ID_Fakultas references Fakultas (ID_Fakultas) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan)

	<ul style="list-style-type: none"> - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - Nama_Fakultas
Jurusan 1:* Memiliki	<p>Memiliki:</p> <ul style="list-style-type: none"> - (PK) ID_Fakultas references Fakultas (ID_Fakultas) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - Nama_Jurusan
Jurusan 1:* Mempunyai	<p>Mempunyai:</p> <ul style="list-style-type: none"> - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - (PK) Kode_Jadwal references Jadwal (ID_Jadwal) - Nama_Jurusan
Jurusan 1:* Mahasiswa	<p>Mahasiswa:</p> <ul style="list-style-type: none"> - (PK) NIM references Mahasiswa (NIM) - (PK) Kode_Cabang references Cabang (Kode_Cabang) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - Nama_Mahasiswa - Tempat_Lahir - Tanggal_Lahir - Gender - Alamat - GPA - Email - Nama_Jurusan
Cabang 1:* Mahasiswa	<p>Mahasiswa:</p> <ul style="list-style-type: none"> - (PK) NIM references Mahasiswa (NIM) - (PK) Kode_Cabang references Cabang (Kode_Cabang) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - Nama_Mahasiswa

	<ul style="list-style-type: none"> - Tempat_Lahir - Tanggal_Lahir - Gender - Alamat - GPA - Email - Nama_Cabang - Alamat_Cabang
Mahasiswa 1:* MahasiswaDetailAddress	<p>MahasiswaDetailAddress:</p> <ul style="list-style-type: none"> - (PK) ID_Address references Mahasiswa (ID_Address) - (PK) ID_Mahasiswa references Mahasiswa (NIM) - (PK) NIM references Mahasiswa (NIM) - (PK) Kode_Cabang references Cabang (Kode_Cabang) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - Nama_Mahasiswa - Tempat_Lahir - Tanggal_Lahir - Gender - GPA - Email - Address1
Mahasiswa 1:* Pendaftaran	<p>Pendaftaran:</p> <ul style="list-style-type: none"> - (PK) ID_Enrollment references Pendaftaran (ID_Enrollment) - (FK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - ID_Staff references Staff (ID_Staff) - Nama_Mahasiswa - Tempat_Lahir - Tanggal_Lahir - Gender - Alamat - GPA - Email
Mahasiswa 1:* Mengatur	<p>Mengatur:</p> <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal

	<p>(ID_Jadwal)</p> <ul style="list-style-type: none"> - ID_Dosen references Dosen(ID_Dosen) - ID_Staff references Staff(ID_Staff) - Tanggal - Nama_Mahasiswa - Tempat_Lahir - Tanggal_Lahir - Gender - Alamat - GPA - Email
Mahasiswa 1:* Mengambil	<p>Mengambil:</p> <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - Comment - Nama_Mahasiswa - Tempat_Lahir - Tanggal_Lahir - Gender - Alamat - GPA - Email
Staff 1:* StaffAddress	<p>StaffAddress:</p> <ul style="list-style-type: none"> - (PK) ID_Address - (FK) ID_Staff references Staff(ID_Staff) - Address - Nama_Staff - Posisi_Staff - Gender_Staff - DOB - Email
Staff 1:* Mengelola	<p>Mengelola:</p> <ul style="list-style-type: none"> - (PK) ID_Staff - (PK) ID_Transaksi - (FK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - Tanggal - Nama_Staff - Posisi_Staff

	<ul style="list-style-type: none"> - Gender_Staff - DOB - Alamat - Email
Staff 1:* Mengatur	Mengatur: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa - (PK) ID_Jadwal - (FK)ID_Dosen references Dosen(ID_Dosen) - (FK) ID_Staff references Staff(ID_Staff) - Tanggal - Nama_Staff - Posisi_Staff - Gender_Staff - DOB - Alamat - Email

7.3 Duplicating FK attributes in 1:* relationship to reduce joins

Entity	Attributes
Bank 1:* Transaksi	Transaksi: <ul style="list-style-type: none"> - (PK) Kode_Transaksi - Kode_Bank references Bank(Kode_Bank) - Biaya
Manager 1:* Staff	Staff: <ul style="list-style-type: none"> - (PK) ID_Staff - ID_Manager references Manager(ID_Manager) - Nama_Staff - Posisi_Staff - Gender_Staff - DOB - Alamat - Email
Pendaftaran 1:* Enrollment	Enrollment: <ul style="list-style-type: none"> - (PK) ID_Enrollment references Pendaftaran (ID_Enrollment) - (PK) Kode_MataKuliah references

	MataKuliah (Kode_MataKuliah)
Dosen 1:* Mengatur	Mengatur: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - ID_Dosen references Dosen(ID_Dosen) - ID_Staff references Staff(ID_Staff) - Tanggal
Dosen 1:* DosenAddress	DosenAddress: <ul style="list-style-type: none"> - (PK) ID_Address references Dosen (ID_Address) - (PK) ID_Dosen references Dosen (ID_Dosen) - Address1
Jadwal 1:* Mengatur	Mengatur: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - (PK) Kode_Jadwal references Jadwal (Kode_Jadwal) - ID_Dosen references Dosen(ID_Dosen) - ID_Staff references Staff(ID_Staff) - Tanggal
Jadwal 1:* Mengambil	Mengambil: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - (PK) Kode_Jadwal references Jadwal (Kode_Jadwal) - Waktu - Sesi - Lokasi
Jadwal 1:* Mempunyai	Mempunyai: <ul style="list-style-type: none"> - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah)

	<ul style="list-style-type: none"> - (PK) Kode_Jadwal references Jadwal (Kode_Jadwal) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - Waktu - Sesi - Lokasi
MataKuliah 1:* Enrollment	Enrollment: <ul style="list-style-type: none"> - (PK) ID_Enrollment references Pendaftaran (ID_Enrollment) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah)
MataKuliah 1:* Memiliki	Memiliki: <ul style="list-style-type: none"> - (PK) ID_Fakultas references Fakultas (ID_Fakultas) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah)
MataKuliah 1:* Mempunyai	Mempunyai: <ul style="list-style-type: none"> - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - (PK) ID_Jadwal references Jadwal (ID_Jadwal)
Fakultas 1:* Memiliki	Memiliki: <ul style="list-style-type: none"> - (PK) ID_Fakultas references Fakultas (ID_Fakultas) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah)
Jurusan 1:* Memiliki	Memiliki: <ul style="list-style-type: none"> - (PK) ID_Fakultas references Fakultas (ID_Fakultas) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah)
Jurusan 1:* Mempunyai	Mempunyai:

	<ul style="list-style-type: none"> - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - (PK) Kode_MataKuliah references MataKuliah (Kode_MataKuliah) - (PK) ID_Jadwal references Jadwal (ID_Jadwal)
Jurusan 1:* Mahasiswa	<p>Mahasiswa:</p> <ul style="list-style-type: none"> - (PK) NIM references Mahasiswa (NIM) - (PK) Kode_Cabang references Cabang (Kode_Cabang) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - Nama_Mahasiswa - Tempat_Lahir - Tanggal_Lahir - Gender - Alamat - GPA - Email
Cabang 1:* Mahasiswa	<p>Mahasiswa:</p> <ul style="list-style-type: none"> - (PK) NIM references Mahasiswa (NIM) - (PK) Kode_Cabang references Cabang (Kode_Cabang) - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan) - Nama_Mahasiswa - Tempat_Lahir - Tanggal_Lahir - Gender - Alamat - GPA - Email
Mahasiswa 1:* MahasiswaDetailAddress	<p>MahasiswaDetailAddress:</p> <ul style="list-style-type: none"> - (PK) ID_Address references Mahasiswa (ID_Address) - (PK) ID_Mahasiswa references Mahasiswa (NIM) - (PK) NIM references Mahasiswa (NIM) - (PK) Kode_Cabang references Cabang (Kode_Cabang)

	<ul style="list-style-type: none"> - (PK) Kode_Jurusan references Jurusan (Kode_Jurusan)
Mahasiswa 1:* Pendaftaran	Pendaftaran: <ul style="list-style-type: none"> - (PK) ID_Enrollment references Pendaftaran (ID_Enrollment) - (FK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - ID_Staff references Staff(ID_Staff) - (FK) Kode_Cabang references Cabang (Kode_Cabang) - (FK) Kode_Jurusan references Jurusan (Kode_Jurusan) - Tanggal_Enrollment
Mahasiswa 1:* Mengatur	Mengatur: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - ID_Dosen references Dosen(ID_Dosen) - ID_Staff references Staff(ID_Staff) - (FK) Kode_Cabang references Cabang (Kode_Cabang) - (FK) Kode_Jurusan references Jurusan (Kode_Jurusan) - Tanggal
Mahasiswa 1:* Mengambil	Mengambil: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - (PK) ID_Jadwal references Jadwal (ID_Jadwal) - (FK) Kode_Cabang references Cabang (Kode_Cabang) - (FK) Kode_Jurusan references Jurusan (Kode_Jurusan) - comment
Staff 1:* StaffAddress	StaffAddress: <ul style="list-style-type: none"> - (PK) ID_Address - (FK) ID_Staff references Staff(ID_Staff) - Address

Staff 1:* Mengelola	Mengelola: <ul style="list-style-type: none"> - (PK) ID_Staff - (PK) ID_Transaksi - (FK) ID_Mahasiswa references Mahasiswa (ID_Mahasiswa) - Tanggal
Staff 1:* Mengatur	Mengatur: <ul style="list-style-type: none"> - (PK) ID_Mahasiswa - (PK) ID_Jadwal - (FK) ID_Dosen references Dosen (ID_Dosen) - (FK) ID_Staff references Staff (ID_Staff) - Tanggal

7.4 Duplicating attributes in *:~ relationship to reduce joins

Pada EER logical kami, tidak memiliki entity dengan relationship *:~

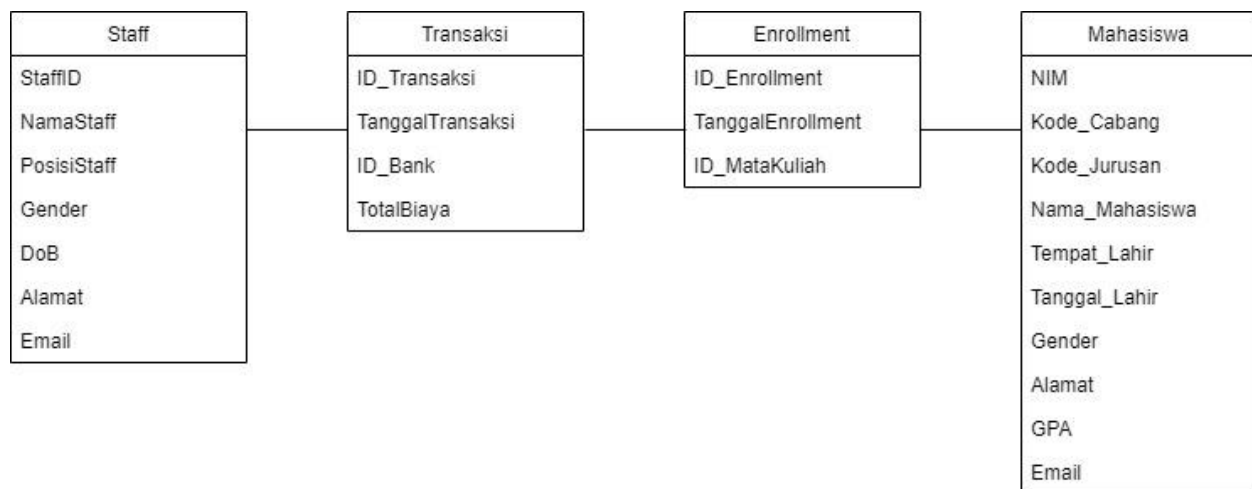
7.5 Introducing repeating groups

Entity	Attributes
Staff	<ul style="list-style-type: none"> - (PK) ID_Staff - Nama_Staff - Posisi_Staff1 - Posisi_Staff2 - Gender_Staff - DOB - Address1 - Address2 - Email
Mahasiswa	<ul style="list-style-type: none"> - (PK) NIM - Kode_Cabang references Cabang (Kode_Cabang) - Kode_Jurusan references Jurusan (Kode_Jurusan) - Nama_Mahasiswa - NamaDepan - NamaBlkg - Tempat_Lahir - Tanggal_Lahir - Gender

	<ul style="list-style-type: none"> - Alamat - GPA - Email - Address1 - Address2
Dosen	<ul style="list-style-type: none"> - (PK) ID_Dosen - Nama_Dosen NamaDepan NamaBlkg - Gender_Dosen - Alamat_Dosen - DOB - Email - Address1 - Address2

7.6 Creating extract tables

Untuk meminimalisir kerja dari sistem database utama, maka kami membuat entity tambahan agar sistem dapat bekerja lebih cepat saat melakukan kinerja transaksi yang paling sering digunakan. Entity yang kami tambahkan yaitu entity Staff, Mahasiswa, Enrollment dan Transaksi karena memiliki kinerja yang paling berat dari semua entity yang ada.



Kami menambahkan entity Staff karena Staff bertugas untuk mengatur semua aktivitas yang dilakukan mahasiswa, mulai dari transaksi sampai dengan enrollment. Selanjutnya kami menambahkan entity Mahasiswa karena transaksi ini membutuhkan mahasiswa dimana merupakan subyek dari case kami. Selanjutnya adalah Transaksi, kami menambahkan entity ini karena kami ingin meringankan beban kerja sistem saat melakukan transaksi, oleh karena itu

transaksi diperlukan. Tidak hanya itu, pada transaksi ini sering mengakses bank sehingga kami menambahkan ID_Bank agar menjadi lebih efektif. Terakhir adalah enrollment, entity ini kami tambahkan karena enrollment merupakan obyek dari transaksi ini. Terdapat juga penambahan ID_MataKuliah pada entity Enrollment karena Enrollment akan selalu mengakses ID_MataKuliah yang akan di enroll oleh mahasiswa.

7.7 Partitioning relations

Pada kasus kami tidak memerlukan partitioning karena tabel kami dapat digunakan secara efisien dan mudah karena tabel-tabel telah terpisah sekecil mungkin untuk menghindari adanya redundan. Namun pada data-data sering melakukan transaksi seperti pada saat mahasiswa melakukan enrollment, maka data-data tersebut akan dipecah secara horizontal partition dimana pada baris 1-10.000 akan masuk ke server A, dan 10.000 baris berikutnya akan masuk ke server B, dan seterusnya. Dengan menggunakan horizontal partition maka akan memudahkan pencarian data serta kinerja sistem diharapkan akan menjadi lebih cepat.