**DOKUMENTASI UNTUK DATABASE JURUSAN**

Created By: Rizal Fathur Rahman

Dokumen ini berisi tentang *database* yang bernama ‘jurusan’ yang dibuat dengan tujuan eksplorasi struktur *database relational* atau RDBMS. Struktur ini dibuat dengan sederhana hanya untuk tujuan pemahaman terhadap struktur RDBMS serta jenis-jenis relasi antar objek *database* atau biasa disebut dengan hubungan relasi antar *table.* Dapat bersifat *one-to-many, many-to-many,* atau *one-to-one.* Penjelasan mengenai jenis relasi akan dijabarkan bersamaan dengan deskripsi dari database yang akan dibahas pada poin-poin selanjutnya. Dan kasus dari pembuatan *database* ini sedikit-banyaknya disesuaikan dengan kondisi lapangan yang ada pada jurusan namun dibuat lebih sederhana. Simulasi *Database* dilakukan menggunakan bantuan *tool* DBeaver serta MySQL.

1. Deskripsi *Database*

Database dibuat dengan kasus sebagai berikut:

Suatu jurusan di dalam kampus sedang melakukan digitalisasi *management system* untuk dosen, mahasiswa, dan civitas lainnya. Dalam hal ini hubungan terhadap eksternal diabaikan terlebih dahulu agar pembahasan dapat difokuskan kepada *database* internal jurusan. Dengan tambahan entitas *table* ujian, nilai mahasiswa,UKT, daftar mahasiswa beserta mata kuliahnya, serta gaji untuk melihat daftar dosen dan civitas lainnya beserta gajinya.

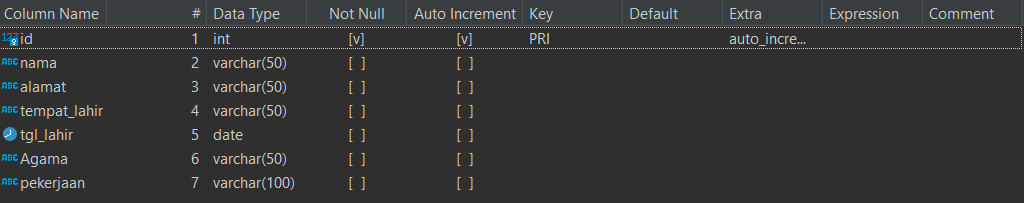
Syarat :

* Satu dosen dan civitas hanya boleh memiliki satu nilai gaji.
* Ujian dapat dilakukan beberapa kali sesuai kategori (UTS, UAS)
* Mahasiswa memiliki mata kuliah lebih dari satu
* Setiap mahasiswa tercatat memiliki satu nilai UKT tidak boleh lebih
* Satu dosen mengajar satu atau lebih mata kuliah.

1. Entitas atau Objek *Database*

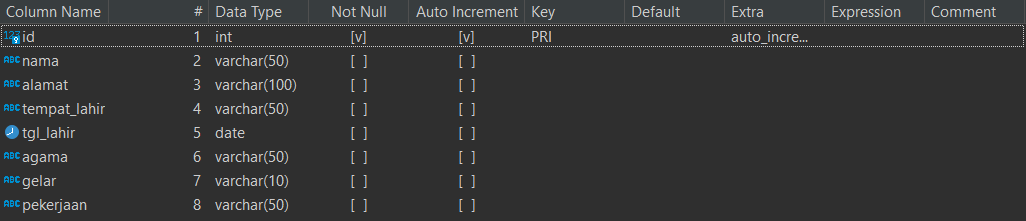
Entitas atau entity di dalam [***database***](https://midteknologi.com/blog/database) adalah benda,orang,tempat,unit,objek atau hal lainnya yang mempresentasikan data, dan tersebut akan disimpan ke dalam pangkal data. Dalam hal ini entitas yang dimaksud merupakan sekumpulan *table* yang ada pada *database* jurusan. Dan di dalam entitas memiliki sekumpulan *attribute* yang apabila dilihat dari *table* itu merupakan sekumpulan *column.* Berikut merupakan entitas serta *attribute* yang dimiliki:

1. Civitas



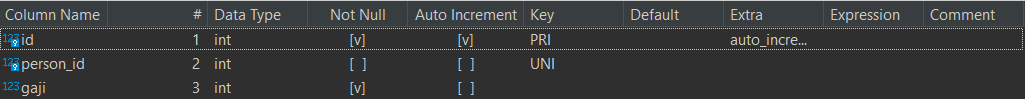
*Table* Civitas memiliki tujuh *attribute* yaitu id (int - primary key), nama (varchar), alamat (varchar), tempat\_lahir (varchar), tgl\_lahir (date), agama (varchar), pekerjaan (varchar). Data yang ada pada civitas ini merupakan data pekerja selain dosen.

1. Dosen



*Table* Dosen memiliki delapan *attribute* yaitu id (int – primary key), nama (varchar), alamat (varchar), tempat\_lahir (varchar), tgl\_lahir (date), agama (varchar), gelar (varchar), pekerjaan (varchar). Seperti namanya, *table* ini berisi data karyawan atau pekerja dengan status dosen pengajar. Selain pekerja yang berstatus dosen maka akan masuk pada *table* civitas.

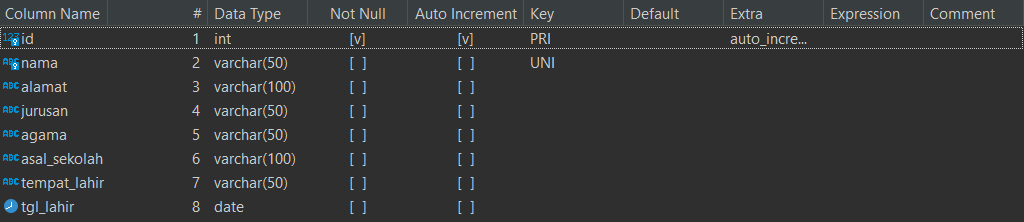
1. Daftar Gaji

**

Pada *table* daftar\_gaji merupakan pencatatan gaji karyawan yang ada pada jurusan.

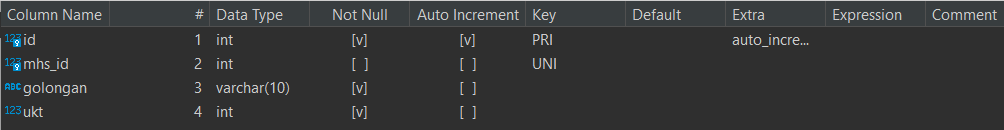
Daftar\_gaji memiliki *attribute* id (int – primary key), person\_id (int – unique - foreign key references dosen(id) AND civitas(id)) Yang berarti data yang ada pada *attribute* person\_id diambil dari civitas dan dosen serta bersifat *unique* yang berarti tidak akan mungkin terjadi duplikat data atau bisa dikatakan bahwa setiap *record* pada *table* civitas atau dosen hanya boleh tepat memiliki satu nilai gaji. Hal ini merupakan implementasi dari hubungan relasi *one-to-one*.

1. Mahasiswa



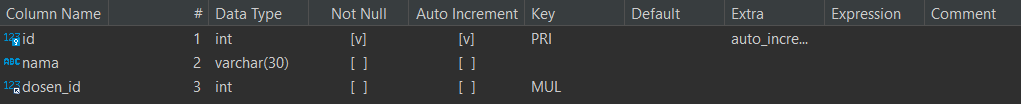
Mahasiswa atau mhs merupakan entitas yang memiliki delapan *attribute* yaitu: id (int – primary key), nama (varchar - unique) sebetulnya pada kasus ini *unique* yang ada pada *attribute* nama bersifat opsional karena tidak berpengaruh terhadap apapun. Namun pada kenyataannya setiap *attribute* data dari *entitas* tunggal yang berarti tidak mengandung foreign key, sebaiknya dibuat unik agar tidak ada duplikasi data sehingga *database* bisa lebih rapi, jurusan (varchar), agama (varchar), asal\_sekolah (varchar), tempat\_lahir (varchar), tgl\_lahir (date).

1. UKT Mahasiswa



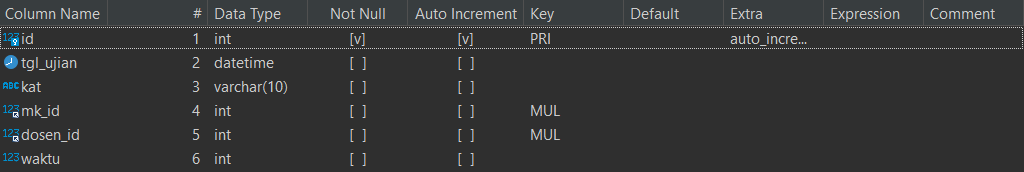
UKT Mahasiswa atau ukt\_mhs merupakan entitas yang mengandung data nilai ukt mahasiswa yang harus dibayarkan menurut id mahasiswa yang terdaftar. Entitas ini bersifat konstan atau tidak berubah sehingga satu mahasiswa hanya memiliki satu nilai ukt. Artinya ini hanyalah *table* pencatatan dari ukt yang harus dibayarkan dan tagihan per-semester tidak masuk ke dalam *table* ini. *Attribute* yang yaitu: id (int – primary key), mhs\_id (unique - foreign key references mhs(id)), golongan (varchar) dan ukt (int).

1. Mata Kuliah



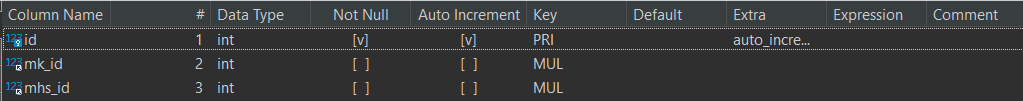
Entitas Mata Kuliah atau mk ini berisikan daftar mata kuliah yang tersedia di kampus beserta dosen pengajarnya dan *attribute* nya yaitu: id (int), nama (varchar), dosen\_id (int foreign key references dosen(id)). Karena tidak ada persyaratan satu dosen hanya boleh mengajar satu mata kuliah maka fk\_dosen\_id tidak dibuat unik.

1. Ujian



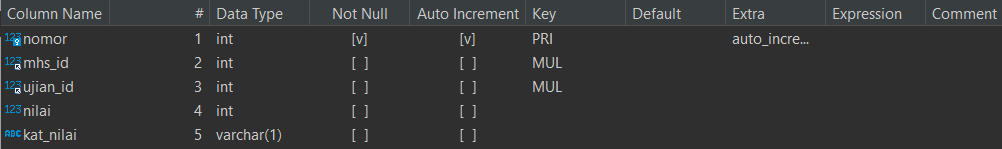
Entitas Ujian umumnya berisi tentang waktu, tanggal, dan mata kuliah yang akan diujikan. *Attribute* nya yaitu: id (int), tgl\_ujian (datetime) dibuat menjadi datetime karena membutuhkan waktu yang lebih akurat selain tanggal melainkan jam ujian dimulai, kat (varchar) apakah kategorinya uts, uas, ataupun lainnya akan dicatat pada *attribute* ini, mk\_id (int foreign key references mk(id)), dosen\_id (int foreign key references dosen(id)), waktu (int) lama dari waktu ujian akan dicatat pada *attribute* ini.

1. *Table* Relasi Mahasiswa dengan Mata Kuliah



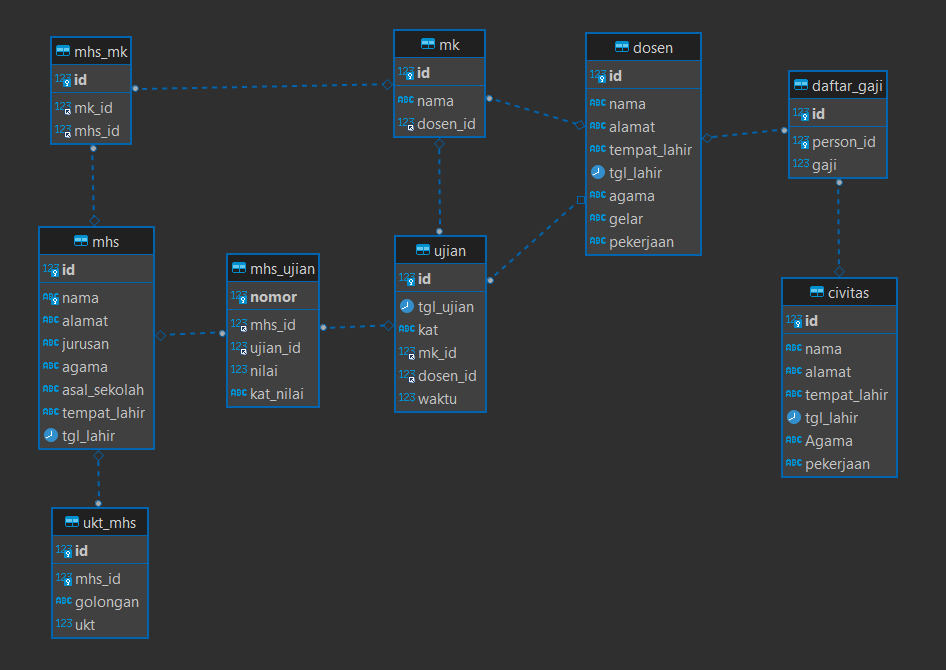
Entitas yang dinamakan mhs\_mk ini merupakan perwujudan relasi dari mahasiswa terhadap mata kuliah yang diambilnya. *Attribute* nya yaitu: id (int primary key), mk\_id (int foreign key references mk(id)), mhs\_id(int foreign key references mhs(id)).

1. *Table* hasil ujian Mahasiswa



Entitas yang dinamakan mhs\_ujian merupakan *table* yang berisi data hasil ujian mahasiswa dengan *attribute*: nomor (int primary key), mhs\_id (int foreign key mhs(id)), ujian\_id (int foreign key ujian(id)), nilai (int), kat\_nilai (varchar).

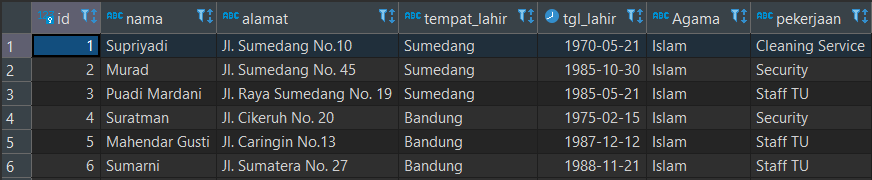
Berikut merupakan struktur *database* secara keseluruhan.



1. *Record*

Berikut merupakan *record* atau data pada setiap entitas:

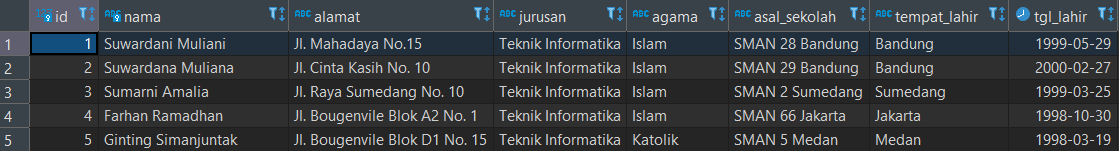
1. Civitas



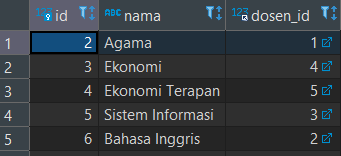
1. Dosen



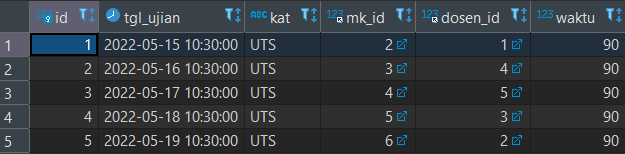
1. Mhs



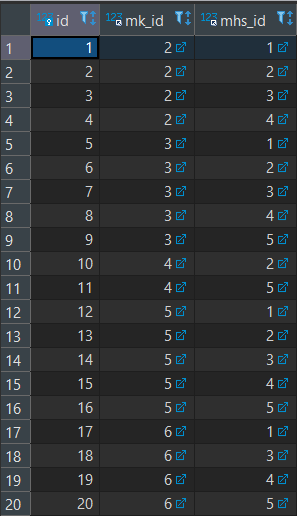
1. Mata Kuliah



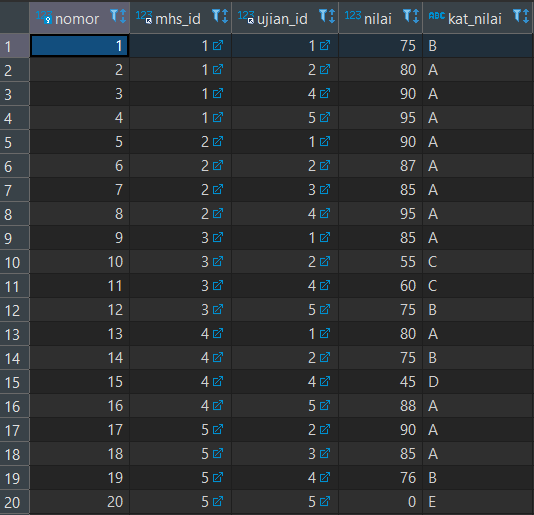
1. Ujian



1. Mhs\_mk



1. Mhs\_ujian



1. *Query*

*Query* di sini merupakan beberapa contoh pengambilan data menggunakan perintah SELECT sesuai dengan kebutuhan.

*Query* untuk mengambil data pada *table* ujian di-*combine* dengan mk dan dosen:

**select** ujian.id **as** 'No.', ujian.tgl\_ujian **as** 'Tanggal Ujian', ujian.kat, mk.nama **as** 'Mata kuliah', **concat\_ws**(' ', dosen.gelar, dosen.nama) **as** 'Pengajar', **concat\_ws**(' ', **convert**(ujian.waktu, **char**), 'menit') **as** 'Waktu Ujian' **from** ujian **inner** **join** mk **on** ujian.mk\_id = mk.id **inner** **join** dosen **on** ujian.dosen\_id = dosen.id;

Output:



*Query* untuk mengambil data hasil ujian:

**select** mhs\_ujian.nomor **as** 'No.', ujian.tgl\_ujian **as** 'Tanggal Ujian', **concat\_ws**(' ', **convert**(ujian.waktu, **char**), 'menit') **as** 'Waktu Ujian', mk.nama **as** 'Mata Kuliah', **concat\_ws**(',',dosen.nama, dosen.gelar) **as** 'Dosen', mhs.nama **as** 'Mahasiswa', mhs\_ujian.nilai, mhs\_ujian.kat\_nilai **as** 'Angka Mutu',

**case**

**when** mhs\_ujian.kat\_nilai = 'A' **then** 'SANGAT BAIK'

**when** mhs\_ujian.kat\_nilai = 'B' **then** 'BAIK'

**when** mhs\_ujian.kat\_nilai = 'C' **then** 'CUKUP'

**when** mhs\_ujian.kat\_nilai = 'D' **then** 'BURUK'

**else** 'SANGAT BURUK'

**end** **as** 'Kategori Penilaian' **from** mhs\_ujian

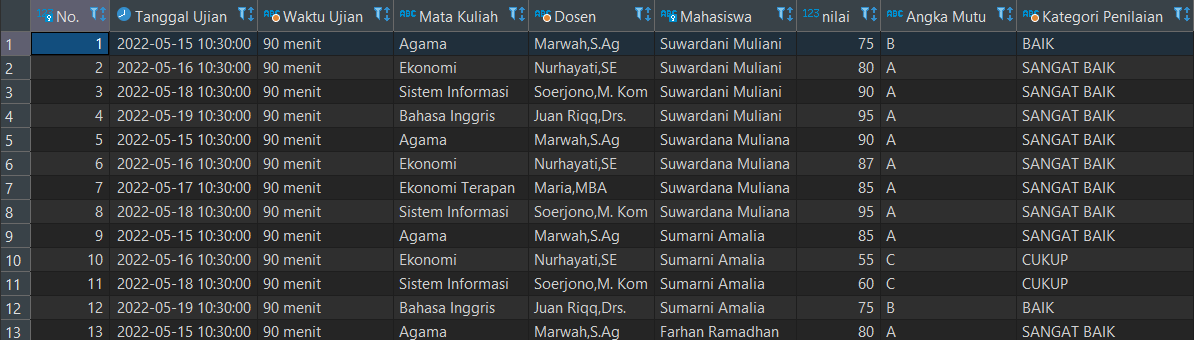
**inner** **join** mhs **on** mhs\_ujian.mhs\_id = mhs.id

**inner** **join** ujian **on** mhs\_ujian.ujian\_id = ujian.id

**inner** **join** mk **on** mk.id = ujian.mk\_id

**inner** **join** dosen **on** dosen.id = mk.dosen\_id **order** **by** mhs\_ujian.nomor;

Output:



*Query* untuk mengambil data untuk analisis mahasiswa yang mengulang serta yang lulus pada periode ujian tersebut:

**select** mhs.nama, **avg**(mhs\_ujian.nilai) **as** nilai\_total,

**case**

**when** **avg**(nilai) > 75 **then** 'LULUS'

**else** 'MENGULANG'

**end** **as** 'KATEGORI PENILAIAN'

**from** mhs\_ujian **inner** **join** mhs **on** mhs\_ujian.mhs\_id = mhs.id **group** **by** mhs.id;

Output:

