Perancangan Database Relasi

Yang di bahas pada bab ini :

- **Perancangan Database Konseptual**
- **Perancangan Database Logis**
- **Entity Relationship Diagram**
- **Normalisasi**
- **Perancangan Database Fisik**

Pada bagian ini anda akan belajar tentang prinsip dasar dalam merancang **database relasi**, karena jenis database ini yang paling banyak dipakai saat ini. Perancangan database model hirarki dan jaringan sudah mulai ditinggalkan, karena ketidakmampuannya dalam mengakomodasi berbagai persoalan dalam suatu sistem database.

Merancang database merupakan hal yang sangat penting, karena disini anda akan menentukan entity, attribut, relasinya dan konsep lainnya dalam suatu sistem database, sehingga hasil rancangan tersebut memenuhi kebutuhan anda akan informasi untuk saat ini dan masa yang akan datang.

Ada tiga langkah dalam merancang database, yaitu :

1. Perancangan Database Konseptual (***Conceptual Database Design***)
2. Perancangan Database Logik (***Logical Database Design***)
3. Perancangan database Fisik (***Physical Database Design***)

Agar memudahkan dalam memahami konsep perancangan database, dalam panduan ini anda akan langsung mempraktekkan, bagaimana membangun **sistem informasi akademik**, karena proses kegiatan akademik sudah terbiasa dengan kegiatan anda sehari-hari dalam dunia akademik.

Sebelum merancang suatu database anda harus mengetahui, apakah akan merancang database baru atau merancang pengembangan database dari sistem yang sedang berjalan. Jika yang dirancang adalah pengembangan dari database yang telah berjalan, maka anda harus memahami konsep yang berlaku pada sistem yang lama dan berusaha merancang database baru agar tetap mengakomodasi sistem yang lama.

Proses pengembangan merancang database lebit sulit daripada merancang database yang benar-benar baru.

Kasus :

Perancangan Sistem Informasi Akademik
Universitas " TEKNOLOGI PRATAMA" Padang

Universitas Teknologi Pratama merupakan salah satu Universitas di kota Padang yang telah berdiri sejak pertengahan tahun 2000 yang lalu. Dalam perkembangannya, dari beberapa fakultas yang ada, ternyata Fakultas Ilmu Komputer menunjukkan perkembangan yang begitu pesat, hal ini terlihat dari semakin meningkatnya jumlah mahasiswa yang mendaftar setiap tahunnya.

Melihat kondisi seperti itu, **Dekan Fakultas Ilmu Komputer Universitas Teknologi Pratama, Prof. Dr. Muhammad Daffa** menginstruksikan kepada Pembantu Dekan I, agar sekiranya dapat merancang suatu Sistem Database untuk mengelola sistem informasi akademik yang terpadu. Karena selama ini, proses registrasi dan kegiatan akademik masih banyak dilakukan secara manual, sehingga sering mengalami tumpang tindih kerja, pemborosan serta banyaknya duplikasi data, yang paling sulit adalah lambatnya mendapatkan informasi akademik seorang mahasiswa, karena bagian kemahasiswaan harus mencek kembali satu persatu arsip mahasiswa tersebut. Akibatnya proses pengambilan keputusan juga berjalan lambat.

Dalam instruksinya tersebut, Bapak Dekan mengharapkan setidaknya dari hasil rancangan sistem informasi tersebut, dapat melakukan hal-hal sebagai berikut :

1. Dapat membuat Informasi detil tentang data mahasiswa
2. Dapat membuat Informasi detil tentang matakuliah
3. Dapat membuat Informasi detil tentang staf pengajar
4. Dapat mencetak absensi secara otomatis untuk setiap matakuliah, sesuai dengan mahasiswa yang mengambil matakuliah tersebut.
5. Dapat menginformasikan, setiap matakuliah berapa mahasiswa yang mengambilnya ?
6. Dapat mencetak data nilai mahasiswa, baik Kartu Hasil Studi maupun Transkrip Akademik.
7. Dapat memberikan informasi siapa dosen pembimbing seorang mahasiswa, dan atau berapa mahasiswa yang dibimbing oleh seorang dosen ?.
8. Dapat memberikan informasi jadwal kuliah.
9. Dapat menentukan berapa lama seorang menempuh kuliah di Fakultas Ilmu Komputer.
10. Dapat menentukan berapa lama seorang mahasiswa membuat skripsinya ?
11. Dan informasi akademik lainnya.

Tujuan utamanya adalah terkomputerisasinya seluruh data akademik, dan mampu mengefisiensikan penggunaan kertas, penggunaan SDM yang terlibat proses akademik, dan yang paling penting adalah **cepat, tepat dan akurat** dalam menyajikan informasi yang diinginkan.

3.1 Perancangan Database Konseptual

Perancangan secara konsep merupakan langkah pertama dalam merancang database. Sesuai dengan namanya, pada tahap ini anda hanya menentukan konsep-konsep yang berlaku dalam sistem database yang akan di bangun.

Dalam tahap ini, setidaknya anda harus mengetahui :

1. Prosedur kerja secara keseluruhan yang berlaku pada sistem yang sedang berjalan.
2. Informasi (output) apa yang diinginkan dari database ?
3. Apa saja kelemahan-kelemahan dari sistem yang sedang berjalan ?
4. Pengembangan sistem di masa yang akan datang.
5. Bagaimana tingkat keamanan data saat ini ?
6. Siapa saja yang terlibat dalam sistem yang sedang berjalan.
7. Apa saja input yang di perlukan ?

Seorang perancang database harus paham benar terhadap sistem yang sedang berjalan dan harus mengetahui sistem yang bekerja pada database yang akan di bangun serta output apa yang diharapkan. Jika anda akan merancang **Sistem Informasi Akademik**, maka anda harus mengetahui bagaimana prosedur kerja dalam dunia akademik secara keseluruhan, siapa-siapa saja yang terlibat didalamnya, apa saja peraturan-peraturan yang berlaku, apakah sebuah ruang kuliah dibatasi siswanya ?, apakah mahasiswa dengan IPK < 2.00 boleh lulus atau tidak ?, bagaimana prosedur tranformasi dari nilai angka ke nilai huruf ?, apakah semua karyawan boleh mengakses database tersebut ?, dan lain sebagainya.

Pemahaman seorang perancang database terhadap sistem yang akan di bangun sangat menentukan baik atau tidaknya hasil rancangan databasenya.

3.2 Perancangan Database Logik

Perancangan database logik merupakan tahapan untuk memetakan proses perancangan konseptual kedalam model database yang akan digunakan, apakah model data hirarki, jaringan atau relasi.Perancangan database secara logik ini tidak tergantung pada **DBMS** yang digunakan, sehingga tahap perancangan ini disebut juga **pemetaan model data**.

Berikut langkah-langkah dalam merancang database logik :

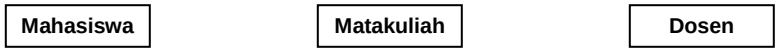
3.2.1 Mendefinisikan Entity Yang Dibutuhkan

Entiti adalah sesuatu yang mudah diidentifikasi dengan mudah dari suatu sistem database, bisa berupa objek, orang, tempat, kejadian atau konsep yang informasinya akan disimpan. Hal-hal yang terlibat dalam suatu sistem database dapat dijadikan entity. Dari sekian banyak kemungkinan entity yang ada, anda harus memilah-milah entity mana saja yang sesuai dan mampu mengakomodasi

kebutuhan sistem yang akan dirancang. Misalnya dalam proses merancang **Sistem Informasi Akademik**, ada banyak kemungkinan yang bisa di jadikan entity, Misalnya entity mahasiswa, matakuliah, dosen, fakultas, jurusan, lokal dan lain sebagainya.

Maka secara sederhana, anda dapat menentukan tiga entity utama yang terlibat dalam proses kegiatan akademik, yaitu :

1. **Entity Mahasiswa**, berfungsi untuk menyimpan data mahasiswa.
2. **Entity Dosen**, berfungsi untuk menyimpan data dosen, dan
3. **Entity Matakuliah**, untuk menyimpan data matakuliah.



Gambar 3.1 : Entity pada sistem informasi akademik

Proses menentukan entity ini memang agak sulit tapi akan menjadi mudah apabila sering latihan terhadap berbagai macam kasus database.

3.2.2 Menentukan Attribut Setiap Entity Beserta Kuncinya

Setelah menentukan entity-entity yang terlibat pada sistem database yang dirancang, langkah berikutnya adalah menentukan attribut yang melekat pada entity tersebut. Attribut adalah ciri khas yang melekat pada suatu entity dan menunjukkan item sejenis. Sama halnya dalam menentukan entity, dalam menentukan attribut ini juga banyak kemungkinan, maka anda harus memilah-memilah attribut apa saja yang diperlukan oleh sistem database yang dirancangan. Berikut beberapa entity yang mungkin pada entity mahasiswa :

- Nobp
- Nama
- Tempat Lahir
- Tanggal Lahir
- Fakultas
- Jurusan
- Agama
- Jenis Kelamin
- Tanggal Masuk
- Nama Pembimbing Akademik
- Asal SMU
- Alamat

- Nama Orang Tua
- Pendidikan Orang Tua
- Pekerjaan Orang Tua
- Alamat Orang Tua
- Dan seterusnya.

Anda boleh saja menggunakan semua kemungkinan attribut tersebut, selama attribut-attribut tersebut dibutuhkan dalam sistem database.

Berikutnya adalah menentukan **attribut kunci** (***key***) dari entity. Kunci ini bersifat unik, sehingga antara satu tuple dengan tuple yang lainnya tidak boleh sama, disebut juga **primary key**. Namun perlu juga diketahui bahwa tidak semua kunci bersifat unik, tergantung kepada keberadaan attribut tersebut pada suatu entity. Sebuah **kunci** dapat saja berupa satu attribut, bisa juga terdiri dari beberapa attribut. Kunci ini akan digunakan nantinya dalam relasi antar entity.

Lihat kembali pembahasan tentang jenis – jenis kunci pada bab II : Memahami Konsep Database.

Dalam entity mahasiswa, attribut **nobp** dapat dijadikan sebagai **kunci**, karena antara satu mahasiswa dengan mahasiswa lainnya tidak akan ada mempunyai nobp yang sama (**unik**). Beda halnya, kalau attribut **nama** anda jadikan kunci, maka bisa saja ada dua mahasiswa atau lebih yang mempunyai nama sama, artinya attribut nama tidak unik. **Kalau dalam suatu entity tidak ada attribut yang bersifat unik, maka anda boleh menambahkan satu buah attribut lagi, sebagai kunci yang bersifat unik**. Misalnya dalam entity matakuliah, terdapat attribut sebagai berikut :

- Nama matakuliah
- Sks
- Semester
- Pra syarat
- Sifat matakuliah, wajib ambil atau hanya matakuliah pilihan.
- Dan sebagainya.

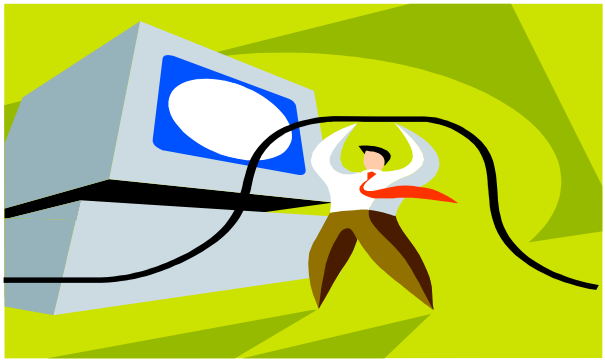
Terlihat pada daftar attribut diatas, tidak ada attribut yang bersifat unik, maka anda tambahkan satu buat attribut lagi dengan nama **kode matakuliah**, attribut ini yang aka dijadikan kunci, karena masing matakuliah mempunya kode yang berbeda.

Dalam entity dosen, banyak juga attribut yang dapat anda tentukan, seperti :

- NIP
- Nama Dosen
- Pendidikan dosen (S.1, S.1, S.3)
- Status Perkawinan
- Jenis kelamin
- Nama Istri
- Jumlah Anak
- Alamat
- Bidang Keahlian
- Jurusan
- Tanggal diangkat
- Dan sebagainya

Pada prinsipnya, semakin banyak attribut dari suatu entity yang ditentukan, semakin banyak pula informasi detil yang diperoleh terhadap entity tersebut, tentunya hal ini juga berimbas kepada semakin besarnya kapasitas daya tampung dalam media penyimpanan database. **Yang penting, gunakanlah entity dan attribut yang diperlukan saja**.

Usahakan dalam pemberian nama entity dan attribut menggambarkan isi dari entity atau attribut tersebut. Misalnya : entity proses belajar mengajar, bisa di beri nama entity belajar. Sehingga memudahkan anda dalam mengolah datanya nanti.



Berikut daftar attribut dan kunci dari entity **Mahasiswa, matakuliah dan dosen** yang digunakan dalam perancangan **Sistem Informasi Akademik Fakultas Ekonomi Universitas Helga Jaya Padang** :

Entity		Attribute
Mahasiswa	1	**NoBP**
	2	Nama Mahasiswa
	3	Tempat Lahir mahasiswa
	4	Tanggal Lahir Mahasiswa
	5	Agama Mahasiswa
	6	Jenis kelamin Mahasiswa
	7	Pembimbing Akademik
	8	Alamat
Matakuliah	1	**Kode Matakuliah**
	2	Nama Matakuliah
	3	Sks
	4	Semester
	5	Pilihan
Dosen	1	**NIP**
	2	Nama Dosen
	3	Jurusan
	4	Bidang Keahlian
	5.	Alamat
Keterangan : attribut yang di cetak tebal dan bergaris bawah merupakan **kunci utama (*Primery Key*).**		

Tabel 3.1 : Daftar attribut beserta kuncinya.

3.2.3 Menentukan Relasi Antar Entity Beserta Kunci Tamunya.

Setelah menentukan entity dan attribut beserta kuncinya, maka selanjutnya adalah menentukan relasi antar entity. Bisa saja antara satu entity dengan entity yang lainnya tidak saling berhubungan, tapi entity tersebut berhubungan dengan entity yang satu lagi. **Jika antara satu entity dengan entity yang lain saling berhubungan, maka hubungan tersebut dinyatakan sebagai entity baru, dan**

harus ditentukan pula attribut dan field kuncinya. Entity hasil relasi pasti mempunyai **kunci tamu *(foreign key)***. Kunci tamu adalah attribute yang berfungsi sebagai kunci pada entity yang lain, tapi digunakan juga sebagai kunci pada entity hasil relasi, maka keberadaan attribut tersebut pada entity hasil relasi di sebut **kunci tamu**.

Untuk menentukan relasi antar tabel, dapat dilakukan dengan merelasikan secara satu per satu (*one by one*). Dalam merancang sistem informasi akademik, anda telah mendapatkan 3 buah entity, yaitu :

1. Mahasiswa
2. Matakuliah
3. Dosen

Maka anda dapat melakukan relasi :

1. **Entity Dosen dengan Entity Matakuliah**
2. **Entity Mahasiswa dengan Entity Dosen**
3. **Entity Mahasiswa dengan Entity Matakuliah**

Untuk lebih jelasnya, penggambaran relasi ini, anda gunakan Diagram E-R.

1. Relasi Antara Entity Dosen Dengan Entity Matakuliah.

Antara entity dosen dengan entity matakuliah, terdapat relasi dalam bentuk jadwal kuliah. Perhatikan gambar berikut :

Gambar 3.2 : Relasi antara entity dosen dengan matakuliah

Terlihat pada gambar diatas, antara entity dosen dengan matakuliah terdapat relasi yaitu jadwal kuliah. Maka **jadwal kuliah merupakan entity baru** pada sistem informasi akademik yang dirancang, sebagai akibat dari relasi antara entity dosen dengan matakuliah.

Selanjutnya adalah menentukan attribut yang melekat pada entity baru hasil relasi (**entity jadwal**), berikut attribut yang mungkin :

- **NIP**
- **Kode matakuliah**
- Hari
- Jam
- Lokal

Attribut NIP dan Kode Matakuliah pada entity jadwal disebut **kunci tamu (*foreign key*)**, karena attribut tersebut merupakan kunci utama (***primary key***) pada entity dosen dan matakuliah. Boleh saja anda menambahkan attribut baru sebagai kunci utama pada entity jadwal ini, seperti **kode jadwal** dan lain sebagainya, kalau seandainya keberadaan kunci tamu tidak dapat mewakili entity jadwal.
Berikut diagram E-R dengan notasi lain untuk menggambarkan secara lengkap jenis hubungan ini :

Kamus data :

- **Dosen** = (NIP, Nama Dosen, jurusan, bidang keahlian, alamat)
- **Matakuliah** = (Kode Matakuliah, nama, sks, semester, pilihan)
- **Jadwal** = (NIP, Kode Matakuliah, hari, jam, lokal)

Gambar 3.3 : Relasi antara entity dosen dengan matakuliah lengkap

2. Relasi Antara Entity Mahasiswa Dengan Entity Dosen.

Relasi atau hubungan antara dosen dengan mahasiswa dapat berupa hubungan dalam hal bimbingan skripsi, bimbingan akademik atau dalam proses belajar mengajar. Biasanya dalam perguruan tinggi, pembimbing akademik sudah ditentukan sejak terdaftar sebagai mahasiswa, sedangkan proses belajar mengajar sudah terangkum pada entity jadwal diatas. Maka hubungan yang mungkin antara dosen dengan mahasiswa adalah dalam hal bimbingan skripsi (**skripsi**). Perhatikan gambar berikut :

Gambar 3.4 : Relasi antara entity mahasiswa dengan dosen

Terlihat pada gambar diatas, antara entity mahasiswa dengan dosen terbentuk relasi dalam hal bimbingan skripsi (**skripsi)**. Maka **bimbingan skripsi merupakan entity baru** pada sistem informasi akademik yang dirancang, sebagai akibat dari relasi antara entity mahasiwa dengan dosen.

Berikutnya menentukan attribut yang melekat pada entity baru hasil relasi (**entity skripsi**), attribut yang mungkin :

- **Nobp**
- **NIP**
- Judul Skripsi
- Tanggal Mulai bimbingan skripsi
- Tanggal Selesai

Attribut **Nobp** dan **NIP** pada entity skripsi disebut **kunci tamu (*foreign key*)**, karena attribut tersebut merupakan kunci utama (***primary key***) pada entity mahasiswa dan dosen. Berikut diagram E-R dengan menggunakan kamus data untuk menggambarkan secara lengkap jenis hubungan ini :

Kamus data :

- **Mahasiswa** = (Nobp, Nama, tempat lahir, tanggal lahir,agama, jenis kelamin, PA, alamat)
- **Dosen** = (NIP, Nama Dosen, jurusan, bidang keahlian, alamat)
- **Skripsi** = (Nobp, NIP, Judul Skripsi, Tanggal Mulai, Tanggal Selesai)

Gambar 3.5 : Relasi antara entity mahasiswa dengan dosen lengkap

3. Relasi Antara Entity Mahasiswa Dengan Entity Matakuliah.

Relasi antara mahasiswa dengan matakuliah dapat berupa hubungan dalam hal pengisian kartu rencana studi, kartu hasil studi dan transkrip. Relasi yang terbentuk anda beri nama KRS misalnya. Perhatikan gambar berikut :

Gambar 3.6 : Relasi antara entity mahasiswa dengan matakuliah

KRS sebagai entity baru pada sistem informasi akademik yang dirancang, sebagai akibat dari relasi antara entity mahasiswa dengan matakuliah.

Berikut attribut yang mungkin pada entity KRS :

- **Nobp**
- **Kode Matakuliah**
- Nilai
- Tanggal lulus

Attribut **Nobp** dan **Kode Matakuliah** pada entity KRS disebut **kunci tamu (*foreign key*)**, karena attribut tersebut merupakan kunci utama (***primary key***) pada entity mahasiswa danmatakuliah. Berikut diagram E-R dengan menggunakan kamus data untuk menggambarkan secara lengkap jenis hubungan ini :

Kamus data :

- **Mahasiswa** = (Nobp, Nama, tempat lahir, tanggal lahir,agama, jenis kelamin, PA, alamat)
- **Matakuliah** = (Kode Matakuliah, nama, sks, semester, pilihan)
- **KRS** = (Nobp, Kode Matakuliah, Nilai, Tanggal Lulus)

Gambar 3.7 : Relasi antara entity mahasiswa dengan matakuliah Lengkap

Proses menentukan hubungan antar entity sangat menentukan kualitas sistem database yang dirancang, karena semua informasi yang diharapkan dari database sangat tergantung kepada relasi yang terbentuk.

Pada tahap ini anda telah melakukan proses relasi antar entity pada sistem informasi akademik yang anda rancang, dan menghasilkan tiga buah entity baru yaitu : **Entity Jadwal, Entity Skripsi** dan **Entity KRS**. Entity baru hasil relasi ini disebut file transaksi, karena salah satu attribut pada entity tersebut merupakan attribut kunci pada entity lainnya. Seperti pada entity jadwal, terdapat attribut **NIP** dan **Kode Matakuliah**, kedua attribut tersebut merupakan attribut kunci pada entity Dosen (**NIP**) dan entity Matakuliah (**Kode Matakuliah**)

Berikut attribut dan kunci dari entity **Jadwal, Skripsi** dan **KRS** yang digunakan dalam perancangan **sistem informasi akademik** :

Entity		Attribute
Jadwal	1	**NIP**
	2	**Kode Matakuliah**
	3	Hari
	4	Jam
	5	Lokal

Skripsi	1	**NoBP**
	2	**NIP**
	3	Tanggal Mulai
	4	Tanggal Selesai
KRS	1	**NoBP**
	2	**Kode Matakuliah**
	3	Nilai
	4	Tanggal Lulus
Keterangan : attribut yang di cetak tebal dan bergaris bawah merupakan **kunci tamu (*Foreign Key*).**		

Tabel 3.2 : Daftar attribut entity hasil relasi beserta kuncinya.

3.2.4 Menentukan Derajat Relasi

Sebagaimana telah dijelaskan pada bab sebelumnya, bahwa derajat relasi menunjukkan jumlah maksimum record suatu entity ber-relasi dengan record pada entity yang lainnya. Derajat relasi yang mungkin terjadi antara satu entity dengan entity lainnya adalah **satu ke satu**, **satu ke banyak atau sebaliknya**, atau **banyak ke banyak.**
Berikut derajat relasi antara entity-entity yang telah terbentuk dalam perancangan sistem informasi akademik :

1. Derajat Relasi Entity Dosen Dengan Entity Matakuliah.

Derajat relasi entity dosen dengan entity matakuliah dapat anda tentukan dengan memahami bahwa seorang dosen dapat mengampu **lebih dari satu** matakuliah dan matakuliah biasanya hanya diajar oleh **satu** dosen. Dari kenyataan tersebut, dapatlah anda simpulkan bahwa derajat relasi antara entity dosen dengan matakuliah adalah **satu ke banyak**. Perhatikan gambar berikut :

Gambar 3.8 : Derajat relasi antara entity dosen dengan matakuliah

2. Derajat Relasi Antara Entity Mahasiswa Dengan Entity Dosen.

Dalam hal bimbingan skripsi, dapat anda ketahui bahwa pembimbing seorang mahasiswa dapat mempunyai **lebih dari satu** pembimbing, begitu juga seorang pembimbing dapat mempunyai **lebih dari satu** mahasiswa bimbingnnya. Derajat relasi yang terbentuk adalah **Banyak ke banyak**. Dapat digambarkan sebagai berikut :

Gambar 3.9 : Derajat relasi antara entity mahasiswa dengan dosen

3. Derajat Relasi Antara Entity Mahasiswa Dengan Entity Matakuliah.

Anda mengetahui bahwa seorang mahasiswa boleh mengambil **lebih dari satu** matakuliah dan satu matakuliah boleh di ambil **lebih dari satu** mahasiswa. Sehingga derajat relasi yang terbentuk adalah **banyak ke banyak**. Perhatikan gambar berikut :

Gambar 3.10 : Derajat relasi antara entity mahasiswa dengan matakuliah

Derajat relasi ini akan sangat diperlukan pada saat anda ingin mengimplementasikannya ke dalam **Database Management System.**

3.2.5 Normalisasi

Normalisasi adalah suatu proses yang bertujuan untuk menciptakan struktur-struktur entity yang dapat mengurangi redundansi data dan meningkatkan stabilitas database. Ada dua fungsi normalisasi, yaitu :

1. Dapat digunakan sebagai metodologi dalam menciptakan desain database.,
2. Dapat digunakan sebagai verifikasi terhadap hasil desain database yang telah dibuat, baik menggunakan E-R Model atau menggunakan model relasi, seperti yang anda buat diatas atau dari model yang lain.

3.2.6 Bentuk-bentuk Normalisasi

Aturan-aturan normalisasi dinyatakan dalam istilah **bentuk normal**. Bentuk normal adalah suatu aturan yang dikenakan pada entity-entity dalam database dan harus dipenuhi oleh entity-entity tersebut sehingga tercapai normalisasi. Suatu entity

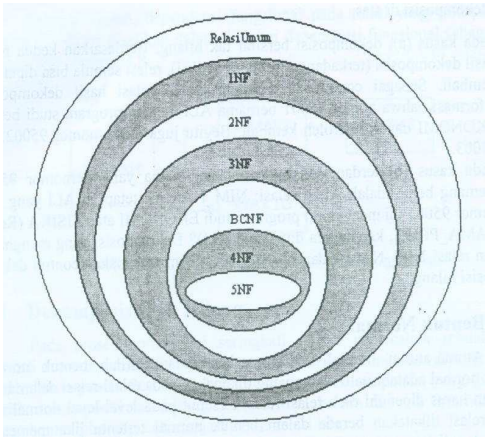
dikatakan dalam bentuk normal apabila entity tersebut memenuhi aturan pada bentuk normal tersebut.

Proses normalisasi dilakukan secara bertingkat. Pada tingkat ketiga (*Third Normal Form*, 3NF) sebenarnya telah dapat menghasilkan suatu rancangan database yang baik.

Berikut tingkatan bentuk normal dalam proses normalisasi :

- Bentuk Normal Pertama (1NF)
- Bentuk Normal Kedua (2NF)
- Bentuk Normal Ketiga(3NF)
- Bentuk Normal Boyce-Codd (BCNF)
- Bentuk Normal Keempat (4NF)
- Bentuk Normal Kelima (5NF)

Perhatikan gambar berikut :



Gambar 3.11 : Bentuk-bentuk normalisasi

Terlihat pada gambar diatas, bahwa setiap level normalisasi bergantung pada level sebelumnya. Misalnya, bentuk normal kedua pasti telah memenuhi bentuk normal pertama, bentuk normal ketiga pasti telah memenuhi bentuk normal kedua, dan seterusnya.

- **Bentuk Normal Pertama (1NF)**

  Bentuk normal pertama dikenakan pada entity yang belum normal (*Unnormalized Form*). Bentuk tidak normal merupakan kumpulan data yang akan di rekam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja tersebut tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

  Berikut contoh entity dalam keadaan belum ternormalisasi :

Nama MHS	Matakuliah	SKS	Nilai
Elzar	S I M S I A	3 3	A C
Fikri	S I M S I A Database Management	3 3 2	C D B
Helga	Peny. Prog. Komp.	2	B
Helga	Peny. Prog. Komp.	2	A

Tabel 3.3 : Bentuk tidak normal

Ada dua kelemahan utama pada bentuk tidak normal diatas :

1. Terdapat attribut yang berulang (duplikat), yaitu attribut matakulia. Mahasiswa dengan nama Elzar mengambil 2 matakuliah, sementara Si Fikri mengambil 3 matakuliah dimana matakuliah yang mereka ambil ada yang sama.
2. Terdapat informasi yang meragukan, dimana ada dua baris memiliki matakuliah yang sama, tapi berbeda nilainya. Sebenernya kedua baris tersebut menunjukkan dua orang yang sama namanya tapi berbeda nilai.

Bentuk entity diatas harus di rubah menjadi bentuk normal pertama.

Aturan Bentuk Normal Pertama (1NF) :

Suatu entity dikatakan dalam bentuk normal pertama jika setiap attributnya bernilai tunggal untuk setip barisnya.

Entity diatas, setelah diubah kebentuk normal pertama sesuai dengan aturan diatas, dapat berupa sebagai berikut :

Nama MHS	Matakuliah	SKS	Nilai
Elzar	S I M	3	A
Elzar	S I A	3	C
Fikri	S I M	3	C
Fikri	S I A	3	D
Fikri	Database Management	2	B
Helga	Peny. Prog. Komp.	2	B
Helga	Peny. Prog. Komp.	2	A

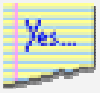
Dapat juga seperti berikut :

Nama MHS	MK 1	SKS	Nilai 1	MK 2	SKS	Nilai 2
Elzar	S I M	3	A	S I A	3	C
Fikri	S I M	3	C	S I A	3	D
Helga	Peny. Prog. Komp.	2	B			
Helga	Peny. Prog. Komp.	2	A			

Tabel 3.4 : Bentuk normal pertama

Terlihat pada entity diatas bahwa setiap attribut telah bernilai tunggal untuk setiap barisnya. Tapi redudansi dan adanya informasi yang meragukan masih belum teratasi.

- **Bentuk Normal Kedua (2NF)**



Aturan Bentuk Normal Kedua (2NF) :

Suatu entity dikatakan dalam bentuk normal pertama jika :

1. **Berada pada bentuk normal pertama.**
2. **Semua attribut bukan kunci memiliki ketergantungan fungsional (*Depedensi Fungsional*) dengan kunci utama (*primary key*)**

Ketergantungan fungsional adalah suatu attribut X mempunyai ketergantungan fungsional terhadap attribut Y apabila setiap nilai X berhubungan dengan sebuah nilai Y. Misalnya Attribut **Nama** pada entity Mahasiswa, mempunyai ketergantungan fungsional terhadap attribut **NoBP**, karena setiap **nama** mahasiswa harus mempunyai **NoBP**.

Pada tahap ini anda harus memilah-memilah dan membagi entity tersebut menjadi beberapa entity lainnya yang mempunyai kunci utama. Sehingga masing-masing attribut yang bukan kunci mempunyai ketergantungan fungsional dengan kunci utama tersebut.

Perhatikan tabel berikut setelah di ubah kedalam bentuk normal kedua :

Entity Mahasiswa

NoBP	**Nama Mahasiswa**
05955001	Elzar
05955002	Fikri
05955003	Helga
05955004	Helga

Entity Nilai

NoBP	**Kode Matakuliah**	**Matakuliah**	**SKS**	**Nilai**
05955001	EKP414	S I M	3	A
05955001	EKP415	S I A	3	C
05955002	EKP414	S I M	3	C
05955002	EKP415	S I A	3	D
05955002	EKP355	Database Management	2	B
05955003	EKP344	Peny. Prog. Komp.	2	B
05955003	EKP344	Peny. Prog. Komp.	2	A

Tabel 3.5 : Bentuk normal kedua

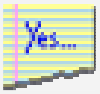
Pada gambar diatas terlihat, ada dua entity yang memiliki **kunci utama** (atribut yang bergaris bawah), sehingga attribut-attribut yang lainnya mempunyai ketergantungan fungsional terhadapnya.

- Attribut Nama mahasiswa mempunyai ketergantungan fungsional terhadap attribut **NOBP**.

- Attribut SKS dan Nilai mempunyai ketergantungan fungsional terhadap attribut **NoBP** dan **Kode Matakuliah.**

Ternyata rancangan entity baru diatas masih belum benar, khususnya pada entity nilai, karena : **adanya data yang berulang**, yaitu nama matakuliah dan sks, Kalau seandai salah mengentrikan data matakuliah dan sks, dapat mengakibatkan **data tidak konsisten** lagi.

- **Bentuk Normal Ketiga(3NF)**



Aturan Bentuk Normal Ketiga (3NF) :

Suatu entity dikatakan dalam bentuk normal pertama jika :

1. **Berada pada bentuk normal kedua.**
2. **Semua attribut bukan kunci tidak memiliki ketergantungan transitif (*Depedensi transitif*) dengan kunci utama (*primary key*)**

Ketergantungan Transitif terjadi pada entity yang menggunakan attribut gabungan sebagai kunci utama. Seperti pada entity nilai pada bentuk normal kedua diatas, yang menjadi kunci utama adalah **NoBP** dan **Kode Matakuliah.** Ketergantungan transitif terjadi bila :

a. Attribut X memiliki ketergantungan fungsional dengan attribut Y.
b. Attribut Z memiliki ketergantungan fungsional dengan attribut X.

Misalnya attribut **Kode Matakuliah** pada entiti nilai diatas, mempunyai ketergantungan fungsional dengan attribut **NoBP**, Attribut **Nama Matakuliah** mempunyai ketergantungan fungsional dengan attribut **Kode Matakuliah**.

Entity Nilai berikut merupakan contoh entity yang memenuhi normal kedua tapi tidak memenuhi bentuk normal ketiga, karena adanya ketergantungan transitif.

NoBP	**Kode Matakuliah**	**Matakuliah**	**SKS**	**Nilai**
05955001	EKP414	S I M	3	A
05955001	EKP415	S I A	3	C
05955002	EKP414	S I M	3	C
05955002	EKP415	S I A	3	D
05955002	EKP355	Database Management	2	B
05955003	EKP344	Peny. Prog. Komp.	2	B
05955003	EKP344	Peny. Prog. Komp.	2	A

Tabel 3.6 : Ketergantungan transitif

Pada contoh diatas, kunci utama merupakan gabungan antara **NoBP** dan **Kode Matakuliah**. Attribut matakuliah, sks dan nilai mempunyai ketergantungan fungsional terhadap kunci utama tersebut. Namun perlu di perhatikan, bahwa jika **Kode Matakuliah** bernilai sama**, Nama Matakuliah** juga bernilai sama. Hal ini menandakan adanya suatu ketergantungan antara kedua attribut tersebut. Lalu manakah yang menjadi penentu ? Apakah **Kode Matakuliah** bergantung pada **Nama matakuliah** ? atau sebaliknya. Yang jadi penentu tentulah **Kode Matakuliah**, karena kode bersifat unik dan akan berbeda untuk setiap nama matakuliah. Oleh karena itu entity Nilai harus dibagi lagi agar memenuhi aturan bentuk normal ketiga.

Perhatikan tabel berikut setelah di ubah kedalam bentuk normal ketiga :

Entity Mahasiswa

NoBP	**Nama Mahasiswa**
05955001	Elzar
05955002	Fikri
05955003	Helga
05955004	Helga

Entity Nilai

NoBP	**Kode Matakuliah**	**Nilai**
05955001	EKP414	A
05955001	EKP415	C
05955002	EKP414	C
05955002	EKP415	D
05955002	EKP355	B
05955003	EKP344	B
05955003	EKP344	A

Entity Matakuliah

Kode Matakuliah	**Matakuliah**	**SKS**
EKP414	S I M	3
EKP415	S I A	3
EKP414	S I M	3
EKP415	S I A	3
EKP355	Database Management	2
EKP344	Peny. Prog. Komp.	2
EKP344	Peny. Prog. Komp.	2

Tabel 3.7 : Bentuk normal ketiga

Coba anda amati, apakah pada bentuk normal ketiga ini, masih ada redundansi dan ketidakkonsistenan data ?

Penerapan aturan normalisasi sampai dengan bentuk ketiga ini, sebenarnya telah memenuhi dalam menghasilkan desain database yang berkualitas baik. Namun demikian dari sejumlah literatur dapat pula dijumpai pembahasan tentang bentuk **normal keempat** (4NF) dan bentuk **normal kelima** (5NF) dan adapula bentuk normal **Boyce-Codd** sebagai perbaikan dari bentuk normal ketiga.

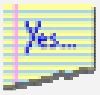
Ketiga bentuk normal yang disebut terakhir (Boyce-Codd, 4NF dan 5 NF), pembahasannya cukup kompleks, tetapi manfaatnya sendiri tidak begitu besar. Karena itu tidak terlalu di bahas pada buku ini. Berikut diberikan aturan umumnya saja :

- **Bentuk Normal Boyce-Codd (BCNF)**

Aturan Bentuk Normal Boyce-Codd (BCNF) :

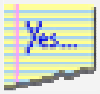
Suatu entity dikatakan dalam bentuk BCNF jika :
Semua kunci utama adalah kunci kandidat yang bersifat unik

- **Bentuk Normal Keempat (4NF)**



Bentuk normal keempat berhubungan dengan sifat ketergantungan banyak nilai (*Multivalued Dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional.

- **Bentuk Normal Kelima (5NF)**



Bentuk normal kelima berkenaan dengan ketergantungan relasi antar tabel (*Join Depedency*).

Hasil akhir dari perancangan database sistem informasi akademik secara logis adalah :

1. **Entity-entity utama, disebut juga file master**.

Entity		Attribute
Mahasiswa	1	**NoBP**
	2	Nama Mahasiswa
	3	Tempat Lahir mahasiswa
	4	Tanggal Lahir Mahasiswa
	5	Agama Mahasiswa
	6	Jenis kelamin Mahasiswa
	7	Pembimbing Akademik
	8	Alamat
Matakuliah	1	**Kode Matakuliah**
	2	Nama Matakuliah
	3	Sks
	4	Semester
	5	Pilihan
Dosen	1	**NIP**
	2	Nama Dosen
	3	Jurusan
	4	Bidang Keahlian
	5.	Alamat
Keterangan : attribut yang di cetak tebal dan bergaris bawah merupakan **kunci utama (*Primery Key*).**		

Tabel 3.8 : Entity utama

2. Entity-entity hasil relasi, disebut juga file transaksi

Entity		Attribute
Jadwal	1	**NIP**
	2	**Kode Matakuliah**
	3	Hari
	4	Jam
	5	Lokal
Skripsi	1	**NoBP**
	2	**NIP**
	3	Tanggal Mulai
	4	Tanggal Selesai
KRS	1	**NoBP**
	2	**Kode Matakuliah**
	3	Nilai
	4	Tanggal Lulus
Keterangan : attribut yang di cetak tebal dan bergaris bawah merupakan **kunci tamu (*Foreign Key*).**		

Tabel 3.9 : Tabel lengkap entity hasil relasi

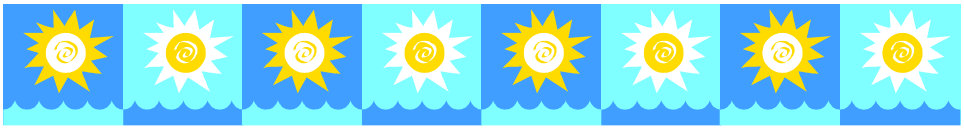
3.3 Perancangan Database Fisik

Perancangan database secara fisik merupakan tahapan untuk mengimplementasikan hasil perancangan database secara logis menjadi tersimpan secara fisik pada media penyimpanan eksternal sesuai dengan DBMS yang digunakan. Dapat disimpulkan bahwa proses perancangan fisik merupakan transformasi dari perancangan logis terhadap jenis DBMS yang digunakan sehingga dapat disimpan secara fisik pada media penyimpanan.

Dalam perancangan database ada beberapa istilah penting yang disamakan fungsi tapi berbeda penggunaannya, yaitu :

1. **Entity dan Tabel**
   Istilah Entity dan tabel mengandung maksud yang sama. Istilah Entity digunakan pada saat anda membicarakan konsep-konsep database, sedangkan istilah tabel merupakan implementasi dari entity ke dalam DBMS, seperti MS. Access.
2. **Tupple dan Record**
   Sama halnya dengan Entiy dan Tabel, istilah Tupple digunakan dalam membicarakan konsep database, sedangkan record digunakan dalam DBMS.
3. **Attribut dan Field**
   Istilah **attribut** digunakan dalam membahas konsep database, sedangkan **field** merupakan istilah yang digunakan oleh dbms, untuk menyebut attribut.

Dalam perancangan database secara logis, anda menggunakan istilah **entity, tupple** dan **record,** maka pada pembahasan database secara fisik, istilah-istilah tersebut digantikan kedudukannya oleh **tabel, record** dan **field** sesuai dengan DBMS yang digunakan.



3.3.1 Tranformasi Istilah Entity Menjadi Tabel

Pada pembahasan perancangan database secara logis, anda telah menghasilkan 6 buah entity. Dalam **DBMS** entity ini disebut dengan tabel, artinya sekarang anda mempunyai 6 buah tabel yaitu :

- Tabel Mahasiswa
- Tabel Matakuliah
- Tabel Dosen
- Tabel Jadwal
- Tabel Skripsi
- Tabel KRS

No.	Entity	Tabel
1	Entity Mahasiswa	Tabel Mahasiswa
2	Entity Matakuliah	Tabel Matakuliah
3	Entity Dosen	Tabel Dosen
4	Entity Jadwal	Tabel Jadwal
5	Entity Skripsi	Tabel Skripsi
6	Entity KRS	Tabel KRS

tabel 3.10 : Tranformasi entity menjadi tabel

Dalam pemberian nama tabel, sebaiknya mengacu kepada aturan-aturan yang telah ditentukan oleh masing-masing DBMS. Microsoft Access memberikan panduan sebagai berikut :

- Nama tabel maksimal panjangnya 64 karakter.
- Boleh terdiri dari kombinasi huruf, angka, spasi dan spesial karakter seperti #, %, ^, dll, kecuali karakter titik (.), tanda seru (!), karakter ` dan tanda kurung siku([ ]).
- Tidak boleh diawali dengan spasi.
- Tidak boleh menggunakan kode ascii, misalnya penggunaan kombinasi tombol alt-249.
- Nama sebaiknya menggambarkan isi dari tabel.

3.3.2 **Tranformasi Istilah Attribute Menjadi Field**

Lain halnya dengan tranformasi entity menjadi tabel, dimana anda hanya mengganti istilah saja, tapi dalam tranformasi attribute menjadi field, tidak saja sekedar mengganti penggunaan istilah, tapi anda akan melakukan tiga hal umum, yaitu :

- **Menyesuaikan nama field sesuai dengan aturan pada DBMS yang digunakan**
- **Menentukan tipe data dari field.**
- **Menentukan ukuran dari field.**

Berikut pembahasannya secara lengkap.

1. Menyesuaikan nama field sesuai dengan aturan pada DBMS yang digunakan

Perhatikan tabel mahasiswa berikut , hasil rancangan anda sebelumnya.

Tabel		Attribute
Mahasiswa	1	**NoBP**
	2	Nama Mahasiswa
	3	Tempat Lahir mahasiswa
	4	Tanggal Lahir Mahasiswa
	5	Agama Mahasiswa
	6	Jenis kelamin Mahasiswa
	7	Pembimbing Akademik
	8	Alamat

Terlihat nama-nama field dari tabel mahasiswa tersebut belum baik dan terlalu panjang, untuk membuat nama field, Microsoft Access memberikan panduan yang sama dengan pemberian nama pada tabel (Lihat panduan memberi nama tabel diatas). Nama field diatas dapat diubah menjadi sebagai berikut :

Tabel		Attribute	Field
Mahasiswa	1	**NoBP**	**NoBP**
	2	Nama Mahasiswa	Nama
	3	Tempat Lahir mahasiswa	Tempat Lahir
	4	Tanggal Lahir Mahasiswa	Tgl Lahir
	5	Agama Mahasiswa	Agama
	6	Jenis kelamin Mahasiswa	Jenis Kelamin
	7	Pembimbing Akademik	PA
	8	Alamat	Alamat

Tabel 3.11 : Contoh tranformasi nama attribute menjadi field

Dalam pemberian nama tabel maupun field, masing-masing DBMS mempunyai aturan tersendiri. Penamaan nama tabel dan field pada buku ini mengacu kepada DBMS MS. Access.

Usahakan nama tabel dan field sesingkat mungkin dan dapat dimengerti, karena nama yang panjang akan memakan space lebih besar lagi pada media penyimpanan.

Berikut daftar nama field baru untuk semua tabel yang telah anda rancang.

Tabel		Nama Field Lama	Nama Field Baru
Matakuliah	1	**Kode Matakuliah**	**Kode**
	2	Nama Matakuliah	Nama
	3	Sks	Sks
	4	Semester	Semester
	5	Pilihan	Pilihan
Dosen	1	**NIP**	**NIP**
	2	Nama Dosen	Nama
	3	Jurusan	Jurusan
	4	Bidang Keahlian	Spesialisasi
	5.	Alamat	Alamat
Jadwal	1	**NIP**	**NIP**
	2	**Kode Matakuliah**	**Kode**
	3	Hari	Hari
	4	Jam	Jam
	5	Lokal	Lokal
Skripsi	1	**NoBP**	**NoBP**
	2	**NIP**	**NIP**
	3	Tanggal Mulai	Tgl Mulai
	4	Tanggal Selesai	Tgl Selesai
KRS	1	**NoBP**	**NoBP**
	2	**Kode Matakuliah**	**Kode**
	3	Nilai	Nilai
	4	Tanggal Lulus	Tgl Lulus

Gambar 3.12 : Tabel lengkap tranformasi nama attribute menjadi field

2. Menentukan tipe data dari field.

Langkah selanjutnya adalah menentukan tipe data dari setiap field. Tipe data adalah jenis data yang akan diinputkan dan disimpan pada sebuah field. Berikut type-type data yang didukung oleh Microsoft Access :

a) **Text**, digunakan untuk menampung data berupa huruf (A – Z, a – z), karakter spesial (!, @. #. $, dll), serta Angka (0 – 9) yang tidak akan di proses secara matematika, seperti NoBP, Nomor Telp. Nomor Rumah. Bisa juga kombinasi dari huruf, angka dan karakter spesial, misalnya kode matakuliah. Daya tampungnya sampai dengan 255 karakter.

b) **Memo**, sama halnya dengan teks, tapi daya tampungnya sampai dengan 65.535 karakter.

c) **Number**, digunakan untuk jenis data angka (0 – 9) saja, yang dapat digunakan dalam proses matematika. Berdasarkan pada daya tampungnya, tipe data ini dibagi lagi atas :

1. **Byte**, mampu menampung nilai data, dalam rentang 0 – 255.
2. **Integer,** mampu menampung data angka (bilangan bulat), dalam rentang –32,768 sampai dengan 32,767.
3. **Long Integer**, mampu menampung data angka (bilangan bulat), dalam rentang –2,147,483,648 sampai dengan 2,147,483,647.
4. **Single**, mampu menampung data angka, dalam rentang –3.402823E38 sampai dengan –1.401298E–45 untuk nilai negatif dan 1.401298E–45 sampai dengan 3.402823E38 untuk nilai positif, baik bilangan pecahan maupun bilangan bulat.
5. **Doble**, mampu menampung data angka, dalam rentang –1.79769313486231E308 sampai dengan –4.94065645841247E–324 untuk nilai negatif dan 1.79769313486231E308 sampai dengan 4.94065645841247E–324 untuk nilai positif, baik bilangan pecahan maupun bilangan bulat.
6. **Decimal**, mampu menampung data angka dari -10^28 -1 sampai dengan 10^28 –1.

d) **Date/Time**, digunakan untuk menyimpan data tanggal dan waktu, ada banyak jenis tampilan tipe data ini, sesuai dengan setting tanggal dan waktu pada komputer.

e) **Currency**, digunakan untuk nilai data uang dan dapat diproses secara matematika.

f) **AutoNumber**, digunakan untuk menyimpan data angka yang bersifat unik yang otomotis bertambah satu setiap ada penambahan record. Biasanya tipe data ini digunakan oleh field kunci. Type data ini tidak bisa diupdate.

g) **Yes/No**, digunakan untuk menampung data yang terdiri dari dua nilai saja, yaitu yes/No, ya/tidak, On/Off, True/False,. Misalnya jenis kelamin, kalau tidak Laki-laki pasti perempuan, maka type datanya bisa diset **Yes/No**.

h) **OLE Object**, merupakan tipe data yang mendukung kolaborasi antara MS. Access dengan aplikasi office lainnya, seperti MS. Word, MS. Excel, dan lain-lain.

i) **Hyperlink**, berupa kombinasi text atau number, yang digunakan untuk berhubungan dengan link-link (URL) di internet.

j) **Lookup Wizard**, dengan tipe data ini, memungkin anda untuk menginputkan data yang terhubung dengan tabel lain. Biasanya menggunakan fasilitas combo box atau list box.

Berikut daftar tipe data dari keseluruhan tabel pada perancangan sistem informasi akademik :

Tabel		Nama Field	Type Data
Mahasiswa	1	**NoBP**	Text
	2	Nama	Text
	3	Tempat Lahir	Text
	4	Tgl Lahir	Date/Time
	5	Agama	Text
	6	Jenis Kelamin	Yes/No
	7	PA	Text
	8	Alamat	Text
Matakuliah	1	**Kode**	Text
	2	Nama	Text
	3	Sks	Number
	4	Semester	Text
	5	Pilihan	Yes/No
Dosen	1	**NIP**	Text
	2	Nama	Text
	3	Jurusan	Text
	4	Spesialisasi	Text
	5.	Alamat	Text

Jadwal	1	**NIP**	Text
	2	**Kode**	Text
	3	Hari	Text
	4	Jam	Date/Time
	5	Lokal	Text
Skripsi	1	**NoBP**	Text
	2	**NIP**	Text
	3	Tgl Mulai	Date/Time
	4	Tgl Selesai	Date/Time
KRS	1	**NoBP**	Text
	2	**Kode**	Text
	3	Nilai	Text
	4	Tgl Lulus	Date/Time

Tabel 3.13 : Tabel lengkap tipe data dari field

3. Menentukan ukuran dari field.

Pada tahap ini anda menentukan ukuran (*size*) dari masing-masing field yang telah ditentukan type datanya. Penentuan ukuran suatu field bertujuan untuk membatasi jumlah karakter yang diinputkan pada field tersebut sehingga sesuai dengan kebutuhan saja. Misalnya Field NoBP, pada setiap perguruan tinggi jumlah karakter yang membentuk NOBP adalah tetap, misalnya NOBP berupa 05955001, terlihat bahwa nobp tersebut mempunyai panjang 8 karakter, dan tidak ada nobp yang melebihi dari 8 karakter. Agar tidak terjadi kesalahan dan dapat menghemat ruang penyimpanan, maka ukuran dari NoBP diset menjadi 8. begitu juga untuk tipe data yang lainnya, anda harus dapat memperkirakan berapa panjang maksimal dari semua field.

Tipe data **text**, ukurannya ditentukan berupa angka, misalya 8, 20, 30, yang menyatakan panjang maksimunnya dalam karakter..
Type data **Number**, ukuran datanya ditentukan oleh jenis numeriknya, apakah berupa byte, integer, long integer, decimal, dan lain-lain.
Type data **Date/Time**, ukurannya di tentukan oleh jenis date/time, apakah berupa short date (23/11/05), atau long date (November 23, 2005), short time (17:34), long time (17:34:00), dan lain lain.

Berikut daftar ukuran dari field yang terdapat pada sistem informasi akademik yang anda rancang :

Tabel		Nama Field	Type Data	Ukuran
Mahasiswa	1	**NoBP**	Text	8
	2	Nama	Text	30
	3	Tempat Lahir	Text	30
	4	Tgl Lahir	Date/Time	Medium Date
	5	Agama	Text	20
	6	Jenis Kelamin	Yes/No	
	7	PA	Text	30
	8	Alamat	Text	50
Matakuliah	1	**Kode**	Text	6
	2	Nama	Text	30
	3	Sks	Number	Byte
	4	Semester	Text	4
	5	Pilihan	Yes/No	
Dosen	1	**NIP**	Text	9
	2	Nama	Text	30
	3	Jurusan	Text	20
	4	Spesialisasi	Text	30
	5.	Alamat	Text	50
Jadwal	1	**NIP**	Text	9
	2	**Kode**	Text	6
	3	Hari	Text	6
	4	Jam	Date/Time	Short Time
	5	Lokal	Text	5
Skripsi	1	**NoBP**	Text	8
	2	**NIP**	Text	9
	3	Tgl Mulai	Date/Time	Medium Date
	4	Tgl Selesai	Date/Time	Medium Date

KRS	1	**NoBP**	Text	8
	2	**Kode**	Text	6
	3	Nilai	Text	1
	4	Tgl Lulus	Date/Time	Medium Date

Tabel 3.14 : Tabel lengkap tipe data serta ukuran dari field

Tabel diatas merupakan hasil akhir dari proses perancangan database Sistem Informasi Akademik yang siap di implementasikan pada DBMS. Sebagaimana telah dijelaskan di awal, bahwa dalam prakteknya anda akan menggunakan **RDBMS Microsoft Access** Versi 2002.