

BAB III

PERANCANGAN SISTEM UJIAN *ONLINE*

Pada bagian ini akan dibahas perancangan sistem ujian *online*, konsep ketergantungan serta normalisasi basis data relasional.

3.1 PERSIAPAN KEBUTUHAN SISTEM

Didalam pengembangan sistem ujian *online* ini, agar menjadi sistem yang baik harus ditunjang dengan teknologi yang memadai baik dari perangkat keras maupun perangkat lunak.

3.1.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang akan dijelaskan meliputi perangkat keras yang dibutuhkan pada saat pembuatan program dan pada saat program diimplementasikan pada objek penelitian. Spesifikasi komputer yang digunakan adalah sebagai berikut:

Tabel 3.1 Kebutuhan Perangkat Keras

Kebutuhan	Spesifikasi
Model	AXIO PICO CJW
Processor	Intel Atom
System type	32-bit Operating System
RAM	2048MB

3.1.2 Kebutuhan Perangkat Lunak

Berikut ini diuraikan kebutuhan perangkat lunak yang dibutuhkan pada saat pembuatan aplikasi. Beberapa *software* yang digunakan adalah :

Tabel 3.2 Kebutuhan Perangkat Lunak

No.	Software	Kebutuhan
1	Windows 7 Ultimate	Sistem Operasi
2	PHP	Bahasa Pemrograman
3	MySQL	Database Server
4	XAMPP	Database Engine
5	Apache	Web Server
6	Google Chrome	Web Browser

3.1.3 Kebutuhan Pengguna

Sistem ujian *online* yang akan dibangun dari kebutuhan informasi dapat dikelompokkan dalam 2 tingkatan yang akan mengakses aplikasi yaitu administrator (dalam hal ini adalah dosen), dan mahasiswa.

a. Administrator

Administrator merupakan tingkatan tertinggi yang dapat mengakses seluruh bagian dari database. Mempunyai hak penuh untuk verifikasi data *user* dan memelihara isi dari database. Administrator juga diberikan hak untuk menambah, mengubah maupun menghapus data atau informasi yang ada baik data mahasiswa dan data yang berkaitan dengan soal ujian.

b. Mahasiswa

Mahasiswa sendiri hanya dapat mengakses fitur-fitur layanan sistem berupa: mengerjakan soal ujian dan mengubah *password* default nya.

3.2 ANALISIS MASALAH

Dengan berkembangnya internet membawa perubahan yang signifikan terhadap semua bidang termasuk salah satunya dalam bidang pendidikan. Dengan internet maka dunia pendidikan tidak hanya terbatas pada ruang dan wilayah tertentu saja, namun sudah meluas ke semua wilayah yang mana pengaksesannya bisa dilakukan secara langsung/*online*.

Kegiatan manual di dalam pelatihan yang selama ini berjalan dilihat kurang terlalu efektif (terutama dalam hal ujian) karena banyak memakan biaya, waktu, tempat dan personil juga merupakan salah satu alasan. Biaya dalam hal banyaknya dana yang dikeluarkan untuk menyediakan dokumen-dokumen ujian seperti fotokopi materi, kuesioner. Waktu dalam hal lamanya tahap-tahap pengerjaan dan pelaporan hasil ujian.

Selain masalah biaya dan waktu yang telah disebutkan di atas masih ada masalah lain, yaitu pengarsipan yang memerlukan tempat yang besar serta personil yang melakukan pengarsipan tersebut. Untuk itu dirancanglah sebuah sistem untuk mengatasi ini, dalam hal ini sistem untuk ujian *online*.

Dalam pembuatan sistem ujian *online* ini terlebih dahulu diperlukan perancangan basis data, karena sistem basis data merupakan komponen dari sistem informasi. Namun dalam pembuatan basis data ini, dalam proses nya banyak terjadi redundansi data dan duplikasi data. Oleh karena itu, dirancanglah basis data untuk masalah ini, dalam hal ini basis data dengan konsep matematika atau biasa dikenal basis data relasional.

Melihat permasalahan yang telah dijelaskan di atas, maka alternatif pemecahan masalah yang ditawarkan adalah dengan menggunakan jasa teknologi jaringan komputer intranet (yang dapat dikembangkan ke internet) yaitu melalui sistem yang *online* sehingga proses ujian yang dahulu terlalu rumit bisa disederhanakan. Sistem ini dijalankan dengan menggunakan jaringan yang *online* dan bahasa pemrograman *web* yaitu PHP dilihat dari pemrograman PHP yang *open source* sehingga tidak memerlukan pembelian lisensi dan kemudahan ketersediaan contoh kode-kode PHP serta banyaknya variasi yang bisa digunakan yang sesuai dengan pengetahuan dasar penulis.

Dan tidak lupa juga dari sisi keamanan sistem yang cukup baik diatasi oleh pemrograman PHP yang bersifat *Client-Server*. Sistem ini juga menggunakan program basis data MySQL yang sangat cocok dengan bahasa pemrograman PHP yang dipilih penulis. Di sini penulis mencoba untuk membantu membuat program untuk sistem tersebut walaupun masih banyak kekurangan, namun sistem ini sudah

dapat digunakan untuk memecahkan masalah atas permasalahan di atas. Peserta cukup mengetikkan data-data dirinya melalui sistem ujian ini, setelah itu peserta tersebut langsung bisa mengikuti ujian melalui komputer. Selain itu peserta bisa juga langsung mengetahui hasil dari ujian yang mereka ikuti.

Untuk membuat sistem ujian *online* ini, basis data yang baik sangat diperlukan untuk kelancaran dari sistem ini sendiri. Ketika redundansi dan dupikasi data sudah minimal, basis data dalam sistem ujian *online* ini dapat dikatakan baik. Untuk itu diperlukan suatu proses normalisasi yang bertujuan untuk meminimalkan redundansi dan duplikasi data tersebut.

3.3 PERANCANGAN SISTEM

Perancangan sistem basis data terdiri dari tiga tahap, yaitu perancangan *view user*, struktur tabel, rancang skema konseptual seperti di bawah ini:

3.3.1 *View User*

Hal pertama dalam perancangan sistem adalah bagaimana mengimplementasikan kebutuhan setiap *user* dalam sebuah gambaran, dapat disebut skema eksternal (*view user*). Gambaran di sini berupa menu-menu apa saja yang dapat dilihat oleh user, seperti dibawah ini:

a. Administrator

Administrator sebagai tingkatan tertinggi dalam sistem ujian *online* ini mempunyai hak untuk melihat keseluruhan sistem, berupa beberapa menu diantaranya:

1. Menu rancangan profil administrator

Dalam menu rancangan profil administrator, administrator dapat melihat data diri dari administrator itu sendiri berupa ID USER, NAMA ADMIN, ALAMAT, dan AGAMA.

ID USER : 1103934 NAMA ADMIN : Imam Maliki ALAMAT : Bandung AGAMA : Islam
--

Gambar 3.1 Tampilan menu rancangan profil admin

2. Menu rancangan tambah mahasiswa

Dalam menu rancangan tambah mahasiswa, administrator diberikan hak untuk menambahkan mahasiswa dengan memasukkan beberapa data diri mahasiswa berupa ID USER, NAMA MAHASISWA, ALAMAT, dan AGAMA.

ID USER : 1205452 NAMA MAHASISWA : Ayyuna Zulfa Zahidah ALAMAT : Banten AGAMA : Islam
--

Gambar 3.2 Tampilan menu rancangan tambah mahasiswa

3. Menu rancangan tambah ujian

Dalam menu rancangan tambah ujian, administrator dapat menambahkan ujian yang akan diujikan, dengan detail ID MATA KULIAH, NAMA MATA KULIAH, NAMA UJIAN, TANGGAL, DAN WAKTU.

ID MATA KULIAH : 1 NAMA MATA KULIAH : Kalkulus Diferensial NAMA UJIAN : UTS TANGGAL : 2015-08-26 WAKTU : 100 Menit
--

Gambar 3.3 Tampilan menu rancangan tambah ujian

4. Menu rancangan tambah soal

Dalam menu rancangan tambah soal, administrator dapat menambahkan soal yang akan diujikan berdasarkan data dari nama mata kuliah dan nama ujian. Pada menu tambah soal terdapat empat data yang dapat diinputkan administrator, yaitu ID UJIAN, ID SOAL, ISI SOAL, JAWABAN, dan STATUS.

ID UJIAN : 01
ID SOAL : 1
ISI SOAL : log 1000
JAWABAN : 3
STATUS : Benar

Gambar 3.4 Tampilan menu rancangan tambah soal

5. Menu rancangan lihat nilai

Dalam menu rancangan lihat nilai, administrator dapat melihat data nilai mahasiswa dengan detail ID USER, NAMA MAHASISWA, NAMA MATA KULIAH, NAMA UJIAN, NILAI.

ID USER : 1205452
NAMA MAHASISWA : Ayyuna Zulfa Zahidah
NAMA MATA KULIAH : Kalkulus Diferensial
NAMA UJIAN : UTS
NILAI : 90

Gambar 3.5 Tampilan menu rancangan lihat nilai

b. Mahasiswa

Dalam sistem ujian *online* ini, mahasiswa hanya diberikan beberapa hak tertentu saja. Seperti di bawah ini:

1. Menu rancangan profil mahasiswa

Dalam menu rancangan profil mahasiswa, mahasiswa diberikan hak untuk melihat data diri dari mahasiswa itu sendiri berupa ID USER, NAMA MAHASISWA, ALAMAT, dan AGAMA.

ID USER : 1205452 NAMA MAHASISWA : Ayyuna Zulfa Zahidah ALAMAT : Banten AGAMA : Islam
--

Gambar 3.6 Tampilan menu rancangan profil mahasiswa

2. Menu rancangan pilih ujian

Dalam menu rancangan pilih ujian, mahasiswa dapat memilih NAMA MATA KULIAH dan NAMA UJIAN apa yang akan diikuti dengan rincian sebagai berikut:

NAMA MATA KULIAH : Kalkulus Diferensial NAMA UJIAN : UTS TANGGAL : 2015-08-26 WAKTU : 100 Menit
--

Gambar 3.7 Tampilan menu rancangan pilih ujian

3. Menu rancangan kerjakan ujian

NAMA MAHASISWA : Ayyuna Zulfa Zahidah NAMA MATA KULIAH : Kalkulus Diferensial NAMA UJIAN : UTS WAKTU : 100 Menit ID SOAL : 1 ISI SOAL : log 1000 JAWABAN : 3 STATUS : Benar NILAI : 10
--

Gambar 3.8 Tampilan menu rancangan kerjakan ujian

3.3.2 Struktur Tabel

Sebelum implementasi basis data dapat dilakukan (dengan menggunakan salah satu perangkat lunak DBMS), maka terlebih dahulu kita harus memiliki contoh data yang akan digunakan. Contoh data ini harus dianalisa untuk mengetahui karakteristik masing-masing tabel data. Pengetahuan tentang karakteristik ini penting, karena hal itu akan dijadikan dasar untuk penetapan struktur masing-masing tabel. Di dalam penerapan basis data, penetapan struktur tersebut harus dilakukan sebelum penyimpanan datanya dapat dilakukan.

Berikut ini merupakan gambaran dari struktur tabel-tabel pada ujian *online* ini yang disusun berdasarkan *view user* di atas:

- a. Tabel Administrator, dengan isi sebagai berikut:

Tabel 3.3 Struktur Data pada Tabel Administrator

Nama Kolom	Tipe	Lebar	Untuk Menampung Data
ID_USER	Character	20	ID admin
NAMA_ADMIN	Character	50	Nama admin
ALAMAT	Character	80	Alamat admin
AGAMA	Character	30	Agama admin

Berdasarkan data pada tabel di atas dapat kita ketahui beberapa karakteristik tabel, seperti berikut ini:

- Kolom pertama berisi data angka tetapi tidak menunjukkan suatu jumlah (disebut alfa numerik) dengan lebar (banyaknya karakter) tetap, yaitu 20 karakter (digit).
- Kolom kedua berisi data karakter (*string* atau *text*) dengan lebar maksimum 50 karakter. Angka 50 ini merupakan perkiraan bahwa untuk nama admin tidak ada yang melebihi 50 karakter.

- Kolom ketiga berisi data karakter (*string* atau *text*) dengan lebar maksimum 80 karakter. Angka ini lebih banyak ketimbang lebar kolom yang kedua, karena umumnya data alamat lebih panjang dibandingkan data nama.
 - Kolom keempat berisi karakter (*string* atau *text*) dengan lebar maksimum 30 karakter.
- b. Tabel mahasiswa, dengan isi sebagai berikut:

Tabel 3.4 Struktur Data pada Tabel Mahasiswa

Nama Kolom	Tipe	Lebar	Untuk Menampung Data
ID_USER	Character	20	ID mahasiswa
NAMA_MAHASISWA	Character	50	Nama mahasiswa
ALAMAT	character	80	Alamat mahasiswa
AGAMA	character	30	Agama mahasiswa

Berdasarkan data pada tabel di atas dapat kita ketahui beberapa karakteristik tabel, seperti berikut ini:

- Kolom pertama berisi data angka tetapi tidak menunjukkan suatu jumlah (disebut alfa numerik) dengan lebar (banyaknya karakter) tetap, yaitu 20 karakter (digit).
- Kolom kedua berisi data karakter (*string* atau *text*) dengan lebar maksimum 50 karakter. Angka 50 ini merupakan perkiraan bahwa untuk nama mahasiswa tidak ada yang melebihi 50 karakter.
- Kolom ketiga berisi data karakter (*string* atau *text*) dengan lebar maksimum 80 karakter. Angka ini lebih banyak ketimbang lebar kolom yang kedua, karena umumnya data alamat lebih panjang dibandingkan data nama.
- Kolom keempat berisi karakter (*string* atau *text*) dengan lebar maksimum 30 karakter.

c. Tabel Ujian, dengan isi sebagai berikut:

Tabel 3.5 Struktur Data pada Tabel Ujian

Nama Kolom	Tipe	Lebar	Untuk Menampung Data
ID_MK	Integer	3	ID mata kuliah
NAMA_MK	Character	50	Nama mata kuliah
ID_UJIAN	Integer	10	ID ujian
NAMA_UJIAN	Character	50	Nama ujian
TANGGAL	Date/time		Tanggal ujian
WAKTU	Integer	3	Durasi ujian (dalam menit)

Berdasarkan data pada tabel di atas dapat kita ketahui beberapa karakteristik tabel, seperti berikut ini:

- Kolom pertama berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat.
- Kolom kedua berisi data karakter (*string* atau *text*) dengan lebar maksimum 50 karakter. Angka 50 ini merupakan perkiraan bahwa untuk nama mata kuliah tidak ada yang melebihi 50 karakter.
- Kolom ketiga berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat.
- Kolom keempat berisi data karakter (*string* atau *text*) dengan lebar maksimum 50 karakter. Angka 50 ini merupakan perkiraan bahwa untuk nama mata ujian tidak ada yang melebihi 50 karakter.
- Kolom kelima berisi data penanggalan ujian berlangsung.
- Kolom keenam berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat.

d. Tabel Soal, dengan isi sebagai berikut:

Tabel 3.6 Struktur Data pada Tabel Soal

Nama Kolom	Tipe	Lebar	Untuk Menampung Data
ID_UJIAN	integer	10	ID ujian
ID_SOAL	integer	3	ID soal
ISI_SOAL	Text	200	ISI SOAL
ID_JAWABAN	integer	10	ID jawaban
JAWABAN	Text	40	Pilihan jawaban
STATUS	tinyint	1	Keterangan jawaban benar atau salah

Berdasarkan data pada tabel di atas dapat kita ketahui beberapa karakteristik tabel, seperti berikut ini:

- Kolom pertama berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat
- Kolom kedua berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat.
- Kolom ketiga berisi data *text* dengan lebar maksimum 200 karakter.
- Kolom keempat berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat.
- Kolom kelima berisi data *text* dengan lebar maksimum 40 karakter.
- Kolom keenam berisi data angka yang menunjukkan jawaban benar atau salah.

e. Tabel Nilai, dengan isi sebagai berikut:

Tabel 3.7 Struktur Data pada Tabel Nilai

Nama Kolom	Tipe	Lebar	Untuk Menampung Data
ID_NILAI	integer	10	ID NILAI
ID_USER	character	20	ID mahasiswa
NAMA_MAHASISWA	character	50	Nama mahasiswa
ID_UJIAN	integer	10	ID ujian
NAMA_UJIAN	character	50	Nama ujian
NILAI	integer	3	Nilai ujian

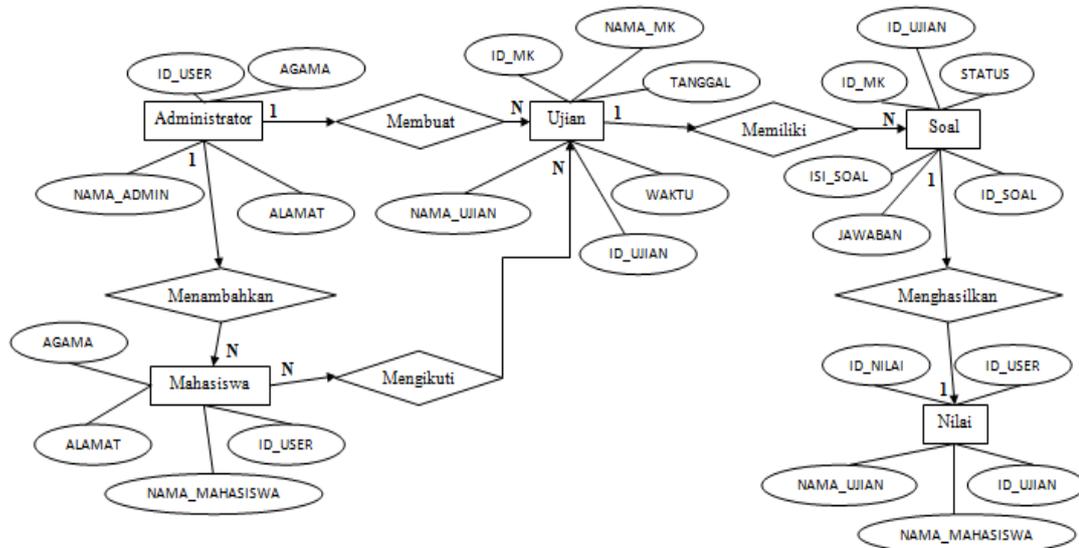
Berdasarkan data pada tabel di atas dapat kita ketahui beberapa karakteristik tabel, seperti berikut ini:

- Kolom pertama berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat.
- Kolom kedua berisi data angka tetapi tidak menunjukkan suatu jumlah (disebut alfa numerik) dengan lebar (banyaknya karakter) tetap, yaitu 20 karakter (digit).
- Kolom ketiga berisi data karakter (*string* atau *text*) dengan lebar maksimum 50 karakter. Angka 50 ini merupakan perkiraan bahwa untuk nama mahasiswa tidak ada yang melebihi 50 karakter.
- Kolom keempat berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat.
- Kolom kelima berisi data karakter (*string* atau *text*) dengan lebar maksimum 50 karakter. Angka 50 ini merupakan perkiraan bahwa untuk nama mata ujian tidak ada yang melebihi 50 karakter.

- Kolom keenam berisi data angka yang menunjukkan jumlah yang jenisnya bilangan bulat.

3.3.3 Rancang skema konseptual (view logis)

Dari struktur tabel di atas, langkah selanjutnya adalah membuat suatu skema konseptual atau *view logis* untuk sistem ujian online. Seperti di bawah ini:



Gambar 3.9 Skema Konseptual Sistem Ujian Online

Pada gambar 3.9, merupakan relasi antar lima tabel yang menjelaskan bahwa:

1. Seorang admin dapat menambahkan banyak mahasiswa.
2. Seorang admin dapat menambahkan banyak ujian.
3. Seorang mahasiswa dapat mengikuti banyak ujian, begitu juga sebaliknya. Satu ujian dapat diikuti oleh banyak mahasiswa.
4. Satu ujian memiliki banyak soal.
5. Dan satu soal memiliki satu nilai.

3.4 ILUSTRASI SISTEM

Untuk mengilustrasikan model basis data lebih nyata, berikut ini dengan memisalkan data-data, dibuat tabel-tabel yang berasal dari struktur data dari tabel di atas, seperti di bawah ini:

3.4.1 Data Administrator

Berdasarkan struktur data dari tabel administrator (**Tabel 3.3**) di atas dibuatlah basis data yang lebih nyata seperti di bawah ini:

Tabel 3.8 Data Administrator

id_user	nama_admin	alamat	agama
0000001	Imam Maliki	Lampung	Islam
0000002	Andi Haerul	Makasar	Islam
0000003	Rita Yekti	Lampung	Islam
0000004	Yohannes Surya	Bandung	Kristen

3.4.2 Data Mahasiswa

Berdasarkan struktur data dari tabel mahasiswa (**Tabel 3.4**) di atas dibuatlah basis data yang lebih nyata seperti di bawah ini:

Tabel 3.9 Data Mahasiswa

id_user	nama_mahasiswa	alamat	agama
1102060	Asep Egi Kurniawan	Bekasi	Islam
1102486	Rakhmat Nurul Hakim	Bekasi	Islam
1105136	Muhammad Rifqy Agustian	Bangka	Islam
1102379	Naro Cahya	Medan	Budha
1100354	Asep Egi Kurniawan	Medan	Islam

3.4.3 Data Ujian

Berdasarkan struktur data dari tabel ujian (**Tabel 3.5**) di atas dibuatlah basis data yang lebih nyata seperti di bawah ini:

Tabel 3.10 Data Ujian

id_mk	nama_mk	id_ujian	nama_ujian	tanggal	waktu (menit)
--------------	----------------	-----------------	-------------------	----------------	--------------------------

001	Kalkulus I	01	UTS	16042015	100
002	Matematika Dasar	02	UTS	18042015	100
002	Matematika Dasar	03	UAS	20062015	120

3.4.4 Data Soal

Berdasarkan struktur data dari tabel Soal (**Tabel 3.6**) di atas dibuatlah basis data yang lebih nyata seperti di bawah ini:

Tabel 3.11 Data Soal

id_ujian	id_soal	isi_soal	id_jawaban	jawaban	Status
01	01_1	log 1000	1	3	1
01	01_1	log 1000	2	1	0
01	01_2	sin 45°	3	$\frac{1}{2}\sqrt{2}$	1
01	01_2	sin 45°	4	$\frac{1}{2}$	0
02	02_1	log 0	5	1	1

pada atribut *id_soal*, misalkan 01_1. “01” mengartikan id soal, dan “1” mengartikan nomor soal. Pada atribut *status* angka “1” mengartikan bahwa jawaban benar dan “0” berarti salah.

3.4.5 Data Nilai

Berdasarkan struktur data dari tabel Nilai (**Tabel 3.7**) di atas dibuatlah basis data yang lebih nyata seperti di bawah ini:

Tabel 3.12 Data Nilai

id_nilai	id_user	nama_mahasiswa	id_ujian	nama_ujian	Nilai
1	1102060	Asep Egi Kurniawan	01	UTS	80
2	1102379	Naro Cahya	01	UTS	75

3	1102486	Rakhmat Nurul	02	UTS	85
4	1102060	Asep Egi Kurniawan	02	UTS	90

3.5 ATRIBUT, KEY, dan KETERGANTUNGAN FUNGSIONAL

Sebelum pengujian data dengan normalisasi, terlebih dahulu kita tentukan **Atribut, Key, dan Ketergantungan Fungsional** untuk tiap-tiap tabel yang telah terbentuk di atas, seperti di bawah ini:

3.5.1 Atribut

a. Untuk Tabel Data Administrator

Terdapat empat buah atribut, dengan rincian sebagai berikut:

1. Atribut ke-1 = $dom(A_1) = id_user$
2. Atribut ke-2 = $dom(A_2) = nama_admin$
3. Atribut ke-3 = $dom(A_3) = alamat$
4. Atribut ke-4 = $dom(A_4) = agama$

b. Untuk Tabel Data Mahasiswa

Terdapat empat buah atribut, dengan rincian sebagai berikut:

1. Atribut ke-1 = $dom(A_1) = id_user$
2. Atribut ke-2 = $dom(A_2) = nama_mahasiswa$
3. Atribut ke-3 = $dom(A_3) = alamat$
4. Atribut ke-4 = $dom(A_4) = agama$

c. Untuk Tabel Data Ujian

Terdapat enam buah atribut, dengan rincian sebagai berikut:

1. Atribut ke-1 = $dom(A_1) = id_mk$
2. Atribut ke-2 = $dom(A_2) = nama_mk$
3. Atribut ke-3 = $dom(A_3) = id_ujian$
4. Atribut ke-4 = $dom(A_4) = nama_ujian$
5. Atribut ke-5 = $dom(A_5) = tanggal$

6. Atribut ke-6 = $dom(A_6) = waktu$

d. Untuk Tabel Data Soal

Terdapat enam buah atribut, dengan rincian sebagai berikut:

1. Atribut ke-1 = $dom(A_1) = id_ujian$
2. Atribut ke-2 = $dom(A_2) = id_soal$
3. Atribut ke-3 = $dom(A_3) = isi_soal$
4. Atribut ke-4 = $dom(A_4) = id_jawaban$
5. Atribut ke-5 = $dom(A_5) = jawaban$
6. Atribut ke-6 = $dom(A_6) = status$

e. Untuk Tabel Data Nilai

Terdapat enam buah atribut, dengan rincian sebagai berikut:

1. Atribut ke-1 = $dom(A_1) = id_nilai$
2. Atribut ke-2 = $dom(A_2) = id_user$
3. Atribut ke-3 = $dom(A_3) = nama_mahasiswa$
4. Atribut ke-4 = $dom(A_4) = id_ujian$
5. Atribut ke-5 = $dom(A_5) = nama_ujian$
6. Atribut ke-6 = $dom(A_6) = nilai$

3.5.2 Key

a. Untuk Tabel Data Administrator

Terdapat beberapa *key* pada tabel data administrator, dengan rincian sebagai berikut:

1. $K = (id_user, nama_admin, alamat, agama) \subseteq r (id_user, nama_admin, alamat, agama)$
2. $K = (id_user, nama_admin, alamat) \subseteq r (id_user, nama_admin, alamat, agama)$
3. $K = (id_user, nama_admin) \subseteq r (id_user, nama_admin, alamat, agama)$
4. $K = (id_user) \subseteq r (id_user, nama_admin, alamat, agama)$
5. $K = (nama_admin) \subseteq r (id_user, nama_admin, alamat, agama)$

Kelima *key* diatas merupakan *Super-key* karena untuk sembarang tupel μ_j dan μ_k dengan $j, k = 1, 2, \dots, n$ dan $j \neq k, \mu_j(K) \neq \mu_k(K)$. Untuk (*alamat*), dan (*agama*)

tidak dipilih menjadi *Super-key* karena ada kecenderungan memiliki nilai yang sama untuk entitas yang berbeda.

(id_user) dan $(nama_admin)$ merupakan *Candidate-key*, asalkan pada $(nama_admin)$ dapat dijamin bahwa tidak ada nilai yang sama untuk kolom ini. Dengan didasari kriteria pemilihan *Primary-Key*, (id_user) dipilih menjadi *Primary-Key*.

b. Untuk Tabel Data Mahasiswa

Terdapat beberapa *key* pada tabel data mahasiswa, dengan rincian sebagai berikut:

1. $K = (id_user, nama_mahasiswa, alamat, agama) \subseteq r (id_user, nama_mahasiswa, alamat, agama)$
2. $K = (id_user, nama_mahasiswa, alamat) \subseteq r (id_user, nama_mahasiswa, alamat, agama)$
3. $K = (id_user, nama_mahasiswa) \subseteq r (id_user, nama_mahasiswa, alamat, agama)$
4. $K = (id_user) \subseteq r (id_user, nama_mahasiswa, alamat, agama)$
5. $K = (nama_mahasiswa) \subseteq r (id_user, nama_mahasiswa, alamat, agama)$

Kelima *key* diatas merupakan *Super-key* karena untuk sembarang tupel μ_j dan μ_k dengan $j, k = 1, 2, \dots, n$ dan $j \neq k, \mu_j(K) \neq \mu_k(K)$. Untuk $(alamat)$, dan $(agama)$ tidak dipilih menjadi *Super-key* karena ada kecenderungan memiliki nilai yang sama untuk entitas yang berbeda.

(id_user) dan $(nama_mahasiswa)$ merupakan *Candidate-key*, asalkan pada $(nama_mahasiswa)$ dapat dijamin bahwa tidak ada nilai yang sama untuk kolom ini. Dengan didasari kriteria pemilihan *Primary-Key*, (id_user) dipilih menjadi *Primary-Key*.

c. Untuk Tabel Data Ujian

Terdapat beberapa *key* pada tabel data ujian, dengan rincian sebagai berikut:

1. $K = (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu) \subseteq r (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu)$
2. $K = (id_mk, nama_mk, id_ujian, nama_ujian, tanggal) \subseteq r (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu)$
3. $K = (id_mk, nama_mk, id_ujian, nama_ujian) \subseteq r (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu)$
4. $K = (id_mk, nama_mk, nama_ujian) \subseteq r (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu)$
5. $K = (id_mk, nama_mk) \subseteq r (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu)$
6. $K = (id_ujian, nama_ujian) \subseteq r (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu)$
7. $K = (id_mk) \subseteq r (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu)$
8. $K = (id_ujian) \subseteq r (id_mk, nama_mk, id_ujian, nama_ujian, tanggal, waktu)$

Kedelapan *key* diatas merupakan *Super-key* karena untuk sembarang tupel μ_j dan μ_k dengan $j, k = 1, 2, \dots, n$ dan $j \neq k, \mu_j(K) \neq \mu_k(K)$. Untuk $(nama_mk)$, $(nama_ujian)$, $(tanggal)$ dan $(waktu)$ tidak dipilih menjadi *Super-key* karena ada kecenderungan memiliki nilai yang sama untuk entitas yang berbeda.

(id_mk) dan (id_ujian) merupakan *Candidate-key*, dan dengan didasari kriteria pemilihan *Primary-Key*, (id_mk) dan (id_ujian) dipilih menjadi *Primary-Key*.

d. Untuk Tabel Data Soal

Terdapat beberapa *key* pada tabel data soal, dengan rincian sebagai berikut:

1. $K = (id_ujian, id_soal, isi_soal, id_jawaban, jawaban, status) \subseteq r (id_ujian, id_soal, isi_soal, id_jawaban, jawaban, status)$
2. $K = (id_ujian, id_soal, isi_soal, id_jawaban, jawaban) \subseteq r (id_ujian, id_soal, isi_soal, jawaban, id_jawaban, status)$
3. $K = (id_ujian, id_soal, isi_soal) \subseteq r (id_ujian, id_soal, isi_soal, id_jawaban, jawaban, status)$
4. $K = (id_soal, isi_soal) \subseteq r (id_ujian, id_soal, isi_soal, id_jawaban, jawaban, status)$

5. $K = (id_jawaban, jawaban, status) \subseteq r (id_ujian, id_soal, isi_soal, id_jawaban, jawaban, status)$
6. $K = (id_soal) \subseteq r (id_ujian, id_soal, isi_soal, id_jawaban, jawaban, status)$
7. $K = (id_jawaban) \subseteq r (id_ujian, id_soal, isi_soal, id_jawaban, jawaban, status)$

Ketujuh *key* diatas merupakan *Super-key* karena untuk sembarang tupel μ_j dan μ_k dengan $j, k = 1, 2, \dots, n$ dan $j \neq k, \mu_j(K) \neq \mu_k(K)$.

(id_soal) dan $(id_jawaban)$ merupakan *Candidate-key* dan dengan didasari kriteria pemilihan *Primary-Key* (id_soal) dan $(id_jawaban)$ dipilih menjadi *Primary-Key*.

e. Untuk Tabel Data Nilai

Terdapat beberapa *key* pada tabel data nilai, dengan rincian sebagai berikut:

1. $K = (id_nilai, id_user, nama_mahasiswa, id_ujian, nama_ujian, nilai) \subseteq r (id_nilai, id_user, nama_mahasiswa, id_ujian, nama_ujian, nilai)$
2. $K = (id_nilai, id_user, nama_mahasiswa, id_ujian, nila) \subseteq r (id_nilai, id_user, nama_mahasiswa, id_ujian, nama_ujian, nilai)$
3. $K = (id_nilai, id_user, id_ujian, nilai) \subseteq r (id_nilai, id_user, nama_mahasiswa, id_ujian, nama_ujian, nilai)$
4. $K = (id_nilai) \subseteq r (id_nilai, id_user, nama_mahasiswa, id_ujian, nama_ujian, nilai)$

Keempat *key* diatas merupakan *Super-key* karena untuk sembarang tupel μ_j dan μ_k dengan $j, k = 1, 2, \dots, n$ dan $j \neq k, \mu_j(K) \neq \mu_k(K)$. Untuk (id_user) , (id_ujian) $(nama_mahasiswa)$, $(nama_ujian)$ dan $(nilai)$ tidak dipilih menjadi *Super-key* karena ada kecenderungan memiliki nilai yang sama untuk entitas yang berbeda.

(id_nilai) merupakan *Candidate-key*, dan dengan didasari kriteria pemilihan *Primary-Key*, (id_nilai) dipilih menjadi *Primary-Key*.

3.5.3 Ketergantungan Fungsional

Dengan melihat perancangan sistem di atas, hal selanjutnya adalah mengasumsikan suatu *FD*. Diantaranya adalah:

a. Untuk Tabel Data Administrator

Didapat beberapa *FD*, yaitu:

$$\left. \begin{array}{l} id_user \rightarrow nama_admin \\ id_user \rightarrow alamat \\ id_user \rightarrow agama \end{array} \right\} \begin{array}{l} \text{berdasarkan aturan inferensi aksioma 5,} \\ \text{diperoleh:} \\ id_user \rightarrow nama_admin \text{ alamat agama} \end{array}$$

$id_user \rightarrow nama_admin$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Administrator berlaku:

$$\mu_1(id_user) = \mu_2(id_user) \Rightarrow \mu_1(nama_admin) = \mu_2(nama_admin)$$

$id_user \rightarrow alamat$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Administrator berlaku:

$$\mu_1(id_user) = \mu_2(id_user) \Rightarrow \mu_1(alamat) = \mu_2(alamat)$$

$id_user \rightarrow agama$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Administrator berlaku:

$$\mu_1(id_user) = \mu_2(id_user) \Rightarrow \mu_1(agama) = \mu_2(agama)$$

b. Untuk Tabel Data Mahasiswa

Didapat beberapa *FD*, yaitu:

$$\left. \begin{array}{l} id_user \rightarrow nama_mahasiswa \\ id_user \rightarrow alamat \\ id_user \rightarrow agama \end{array} \right\} \begin{array}{l} \text{berdasarkan aturan inferensi aksioma 5,} \\ \text{diperoleh:} \\ id_user \rightarrow nama_mahasiswa \text{ alamat} \\ \text{agama} \end{array}$$

$id_user \rightarrow nama_mahasiswa$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Mahasiswa berlaku:

$$\mu_1(id_user) = \mu_2(id_user) \Rightarrow \mu_1(nama_mahasiswa) = \mu_2(nama_mahasiswa)$$

$id_user \rightarrow alamat$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Mahasiswa berlaku:

$$\mu_1(id_user) = \mu_2(id_user) \Rightarrow \mu_1(alamat) = \mu_2(alamat)$$

$id_user \rightarrow agama$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Mahasiswa berlaku:

$$\mu_1(id_user) = \mu_2(id_user) \Rightarrow \mu_1(agama) = \mu_2(agama)$$

c. Untuk Tabel Data Ujian

Didapat beberapa *FD*, yaitu:

$$id_mk \rightarrow nama_mk$$

$$id_ujian \rightarrow id_mk$$

$$id_ujian \rightarrow nama_ujian$$

$$id_ujian \rightarrow tanggal$$

$$id_ujian \rightarrow waktu$$

berdasarkan aturan inferensi

aksioma 5, diperoleh:

$$id_ujian \rightarrow nama_ujian id_mk$$

$$tanggal waktu$$

$id_mk \rightarrow nama_mk$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Ujian berlaku:

$$\mu_1(id_mk) = \mu_2(id_mk) \Rightarrow \mu_1(nama_mk) = \mu_2(nama_mk)$$

$id_ujian \rightarrow id_mk$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Ujian berlaku:

$$\mu_1(id_ujian) = \mu_2(id_ujian) \Rightarrow \mu_1(id_mk) = \mu_2(id_mk)$$

$id_ujian \rightarrow nama_ujian$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Ujian berlaku:

$$\mu_1(id_ujian) = \mu_2(id_ujian) \Rightarrow \mu_1(nama_ujian) = \mu_2(nama_ujian)$$

$id_ujian \rightarrow tanggal$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Ujian berlaku:

$$\mu_1(id_ujian) = \mu_2(id_ujian) \Rightarrow \mu_1(tanggal) = \mu_2(tanggal)$$

$id_ujian \rightarrow waktu$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Ujian berlaku:

$$\mu_1(id_ujian) = \mu_2(id_ujian) \Rightarrow \mu_1(waktu) = \mu_2(waktu)$$

d. Untuk Tabel Data Soal

Didapat beberapa *FD*, yaitu:

$$\left. \begin{array}{l} id_soal \rightarrow isi_soal \\ id_soal \rightarrow id_ujian \end{array} \right\} \begin{array}{l} \text{berdasarkan aturan inferensi aksioma 5, diperoleh:} \\ id_soal \rightarrow isi_soal \ id_ujian \end{array}$$

$$\left. \begin{array}{l} id_jawaban \rightarrow jawaban \\ id_jawaban \rightarrow status \\ id_jawaban \rightarrow id_soal \end{array} \right\} \begin{array}{l} \text{berdasarkan aturan inferensi aksioma 5, diperoleh:} \\ id_jawaban \rightarrow id_soal \ jawaban \ status \end{array}$$

$id_soal \rightarrow isi_soal$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Soal berlaku:

$$\mu_1(id_soal) = \mu_2(id_soal) \Rightarrow \mu_1(isi_soal) = \mu_2(isi_soal)$$

$id_soal \rightarrow id_ujian$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Soal berlaku:

$$\mu_1(id_soal) = \mu_2(id_soal) \Rightarrow \mu_1(id_ujian) = \mu_2(id_ujian)$$

$id_jawaban \rightarrow jawaban$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Soal berlaku:

$$\mu_1(id_jawaban) = \mu_2(id_jawaban) \Rightarrow \mu_1(jawaban) = \mu_2(jawaban)$$

$id_jawaban \rightarrow status$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Soal berlaku:

$$\mu_1(id_jawaban) = \mu_2(id_jawaban) \Rightarrow \mu_1(status) = \mu_2(status)$$

$id_jawaban \rightarrow id_soal$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Soal berlaku:

$$\mu_1(id_jawaban) = \mu_2(id_jawaban) \Rightarrow \mu_1(id_soal) = \mu_2(id_soal)$$

e. Untuk Tabel Data Nilai

Didapat beberapa *FD*, yaitu:

$$\begin{array}{l}
 id_nilai \rightarrow id_user \\
 id_nilai \rightarrow nama_mahasiswa \\
 id_nilai \rightarrow id_ujian \\
 id_nilai \rightarrow nama_ujian \\
 id_nilai \rightarrow nilai \\
 id_user \rightarrow nama_mahasiswa \\
 id_ujian \rightarrow nama_ujian
 \end{array}
 \left. \vphantom{\begin{array}{l} id_nilai \rightarrow id_user \\ id_nilai \rightarrow nama_mahasiswa \\ id_nilai \rightarrow id_ujian \\ id_nilai \rightarrow nama_ujian \\ id_nilai \rightarrow nilai \\ id_user \rightarrow nama_mahasiswa \\ id_ujian \rightarrow nama_ujian \end{array}} \right\} \begin{array}{l} \text{berdasarkan aturan inferensi aksioma 5, diperoleh:} \\ id_nilai \rightarrow id_user \text{ nama_mahasiswa id_ujian} \\ nama_ujian \text{ nilai} \end{array}$$

$id_nilai \rightarrow id_user$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Nilai berlaku:

$$\mu_1(id_nilai) = \mu_2(id_nilai) \Rightarrow \mu_1(id_user) = \mu_2(id_user)$$

$id_nilai \rightarrow nama_user$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Nilai berlaku:

$$\mu_1(id_nilai) = \mu_2(id_nilai) \Rightarrow \mu_1(nama_user) = \mu_2(nama_user)$$

$id_nilai \rightarrow id_ujian$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Nilai berlaku:

$$\mu_1(id_nilai) = \mu_2(id_nilai) \Rightarrow \mu_1(id_ujian) = \mu_2(id_ujian)$$

$id_nilai \rightarrow nama_ujian$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Nilai berlaku:

$$\mu_1(id_nilai) = \mu_2(id_nilai) \Rightarrow \mu_1(nama_ujian) = \mu_2(nama_ujian)$$

$id_nilai \rightarrow nilai$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Nilai berlaku:

$$\mu_1(id_nilai) = \mu_2(id_nilai) \Rightarrow \mu_1(nilai) = \mu_2(nilai)$$

$id_user \rightarrow nama_mahasiswa$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Nilai berlaku:

$$\begin{aligned}\mu_1(id_user) = \mu_2(id_user) &\Rightarrow \mu_1(nama_mahasiswa) \\ &= \mu_2(nama_mahasiswa)\end{aligned}$$

$id_ujian \rightarrow nama_ujian$ adalah memenuhi konsep ketergantungan fungsional karena , untuk setiap $\mu_1, \mu_2 \in$ Tabel Data Nilai berlaku:

$$\mu_1(id_ujian) = \mu_2(id_ujian) \Rightarrow \mu_1(nama_ujian) = \mu_2(nama_ujian)$$

3.6 NORMALISASI

Setelah selesai menentukan Atribut, *Key*, dan Ketergantungan Fungsional, maka kita siap untuk melakukan pengujian apakah basis data yang dibuat di atas telah berada dalam keadaan baik atau normal. Selanjutnya, sebuah basis data dapat dikatakan baik atau normal apabila memenuhi kriteria berikut:

3.6.1 Dekomposisi Aman

Yang dapat dilakukan adalah dengan memilah berdasarkan *FD* minimum yang memuat seluruh atribut pada setiap tabel. Sebagaimana yang telah disebutkan di atas, ada tujuh *FD* untuk seluruh tabel. Yaitu:

$id_user \rightarrow nama_admin \text{ alamat agama}$

$id_user \rightarrow nama_mahasiswa \text{ alamat agama}$

$id_mk \rightarrow nama_mk$

$id_ujian \rightarrow nama_ujian \text{ id_mk tanggal waktu}$

$id_soal \rightarrow isi_soal \text{ id_ujian}$

$id_jawaban \rightarrow id_soal \text{ jawaban status}$

$id_nilai \rightarrow id_user \text{ nama_mahasiswa id_ujian nama_ujian nilai}$

Dengan berbekal ke-tujuh *FD* diatas, maka diperoleh tabel baru hasil dekomposisi tabel-tabel sebelumnya, yaitu:

1. Tabel **Administrator** dengan atribut *id_user*, *nama_admin*, *alamat*, dan *agama*
2. Tabel **Mahasiswa** dengan atribut *id_user*, *nama_mahasiswa*, *alamat*, dan *agama*
3. Tabel **Mata Kuliah** dengan atribut *id_mk* dan *nama_mk*
4. Tabel **Ujian** dengan atribut *id_ujian*, *nama_ujian*, *id_mk*, *tanggal*, dan *waktu*
5. Tabel **Soal** dengan atribut *id_soal*, *isi_soal*, *id_ujian*
6. Tabel **Jawaban** dengan atribut *id_jawaban*, *id_soal*, *jawaban*, dan *status*
7. Tabel **Nilai** dengan atribut *id_nilai*, *id_user*, *nama_mahasiswa*, *id_ujian*, *nama_ujian*, dan *nilai*

Tabel Mata Kuliah dan Tabel Ujian merupakan hasil dekomposisi dari Tabel Data Ujian. Proses dekomposisi nya diperlihatkan seperti di bawah ini:

Misalkan Tabel Data Ujian (Tabel 3.8) adalah r_1 . dapat didekomposisikan menjadi Tabel Mata Kuliah (r_2) dan Tabel Ujian (r_3) seperti berikut ini:

Tabel 3.13 Mata Kuliah

<i>id_mk</i>	<i>nama_mk</i>
001	Kalkulus I
002	Matematika Dasar

Tabel 3.14 Ujian

<i>id_ujian</i>	<i>nama_ujian</i>	<i>id_mk</i>	<i>Tanggal</i>	<i>waktu</i>
01	UTS	01	16042015	100
02	UTS	02	18042015	100
03	UAS	02	20062015	120

Jika kedua tabel di atas digabung kembali dengan menggunakan operasi join, maka Tabel Data Ujian akan diperoleh kembali. $r_1 = r_2 \otimes r_3$ Kita bisa mengatakan bahwa pemilahan tersebut merupakan **Dekomposisi Aman**.

Tabel Soal dan Tabel Jawaban merupakan hasil dekomposisi dari Tabel Data Soal. Proses dekomposisi nya diperlihatkan seperti di bawah ini:

Misalkan Tabel Data Soal (Tabel 3.9) adalah r_4 . dapat didekomposisikan menjadi Tabel Soal (r_5) dan Tabel Jawaban (r_6) seperti berikut ini:

Tabel 3.15 Soal

<i>id_soal</i>	<i>isi_soal</i>	<i>id_ujian</i>
01_1	log 1000	01
01_2	sin 45°	01
02_1	log 0	02

Tabel 3.16 Jawaban

<i>id_jawaban</i>	<i>id_soal</i>	<i>jawaban</i>	<i>Status</i>
1	01_1	3	1
2	01_1	1	0
3	01_2	$\frac{1}{2}\sqrt{2}$	1
4	01_2	$\frac{1}{2}$	0
5	02_1	1	1

Jika kedua tabel di atas digabung kembali dengan menggunakan operasi join, maka Tabel Data Ujian akan diperoleh kembali. $r_4 = r_5 \otimes r_6$ Kita bisa mengatakan bahwa pemilahan tersebut merupakan **Dekomposisi Aman**.

3.6.2 Keterpeliharaan Ketergantungan

Merupakan kriteria kedua yang harus dicapai untuk mendapatkan tabel dan basis data yang baik. Ketika melakukan perubahan data, maka harus dapat dijamin agar perubahan tersebut tidak menghasilkan inkonsistensi data yang mengakibatkan *FD* yang sudah benar menjadi tidak terpenuhi.

Berdasarkan tabel hasil dekomposisi di atas, terdapat dua masalah untuk Tabel Nilai dengan atribut *id_nilai*, *id_user*, *nama_mahasiswa*, *id_ujian*, *nama_ujian*, dan *nilai* dengan dua buah *FD* yaitu: $id_user \rightarrow nama_mahasiswa$ dan $id_ujian \rightarrow nama_ujian$.

Jika pada Tabel Mahasiswa terjadi perubahan data pada atribut *alamat* atau *agama*, maka perubahan ini tidak perlu dijalarakan ke Tabel Nilai, karena memang

di Tabel Nilai atribut *alamat* atau *agama* tidak dilibatkan. Namun jika perubahan terjadi pada atribut *nama_mahasiswa* di Tabel Mahasiswa, maka perubahan harus dijalarkan ke Tabel Nilai, karena atribut ini juga digunakan di Tabel Nilai. Jika penjalaran perubahan ini hanya dilakukan pada satu baris data pertama di Tabel Nilai dengan nilai *id_user* yang sama dengan nilai *id_user* di Tabel Mahasiswa yang *nama_mahasiswa*-nya diubah tersebut, maka $FD\ id_user \rightarrow nama_mahasiswa$ menjadi tidak terpenuhi lagi. Hal ini karena peristiwa tersebut mengakibatkan adanya dua baris data di Tabel Nilai yang *id_user*-nya sama, tetapi *nama_mahasiswa*-nya berbeda. Jika begitu, maka penjalaran perubahan tersebut harus dilakukan ke seluruh baris data dengan nilai *id_user* yang sesuai. Tetapi masalahnya, penjalaran perubahan tersebut tidak efisien dan seharusnya dihindari.

Karena itu, solusi terbaik agar kriteria Keterpeliharaan Ketergantungan dapat terpenuhi adalah dengan meniadakan/melepaskan atribut *nama_mahasiswa* dari Tabel Nilai. Begitu juga untuk $FD\ id_ujian \rightarrow nama_ujian$, agar kriteria Keterpeliharaan Ketergantungan dapat terpenuhi, maka atribut *nama_ujian* harus ditiadakan/dilepaskan dari Tabel Nilai.

Sehingga diperoleh Tabel Nilai Baru yang berisi atribut *id_nilai*, *id_user*, *id_ujian*, dan *nilai*.

Tabel 3.17 Nilai Baru

id_nilai	id_user	id_ujian	Nilai
1	1102060	01	80
2	1102379	01	75
3	1102486	02	85
4	1102060	02	90

3.6.3 Boyce-Codd Normal Form (BCNF)

Kriteria berikutnya untuk mendapatkan tabel basis data yang baik adalah dengan menerapkan *Boyce-Codd Normal Form* (BCNF). BCNF terpenuhi jika untuk semua FD dengan notasi $X \rightarrow Y$, maka X harus merupakan *super-key* pada

tabel tersebut. Jika tidak demikian, maka tabel tersebut harus didekomposisikan sesuai *FD* yang ada, sedemikian sehingga *X* menjadi *super-key* dari tabel hasil dekomposisi.

a. Untuk Tabel Administrator

Berdasarkan Tabel 3.8, *FD* pada tabel tersebut adalah: $id_user \rightarrow nama_admin$ $alamat_agama$. Dengan demikian tabel ini telah memenuhi BCNF, karena menurut BCNF sisi kiri (*X*) dalam *FD* harus merupakan *super-key* dalam tabel. Dan benar bahwa $\{id_user\}$ merupakan *super-key* dari Tabel **Administrator** tersebut.

b. Untuk Tabel Mahasiswa

Berdasarkan Tabel 3.9, *FD* pada tabel tersebut adalah: $id_user \rightarrow nama_mahasiswa$ $alamat_agama$. Dengan demikian tabel ini telah memenuhi BCNF, karena menurut BCNF sisi kiri (*X*) dalam *FD* harus merupakan *super-key* dalam tabel. Dan benar bahwa $\{id_user\}$ merupakan *super-key* dari Tabel **Mahasiswa** tersebut.

c. Untuk Tabel Mata Kuliah

Berdasarkan Tabel 3.11, *FD* pada tabel tersebut adalah: $id_mk \rightarrow nama_mk$. Dengan demikian tabel ini telah memenuhi BCNF karena $\{id_mk\}$ merupakan *super-key* dari Tabel **Mata Kuliah** tersebut.

d. Untuk Tabel Ujian

Berdasarkan Tabel 3.12, *FD* pada tabel tersebut adalah: $id_ujian \rightarrow nama_ujian$ id_mk $tanggal_waktu$. Dengan demikian tabel ini telah memenuhi BCNF karena $\{id_ujian\}$ merupakan *super-key* dari Tabel **Ujian** tersebut.

e. Untuk Tabel Soal

Berdasarkan Tabel 3.13, *FD* pada tabel tersebut adalah: $id_soal, isi_soal, id_ujian$. Dengan demikian tabel ini telah memenuhi BCNF karena $\{id_soal\}$ merupakan *super-key* dari Tabel **Soal** tersebut.

f. Untuk Tabel Jawaban

Berdasarkan Tabel 3.14, *FD* pada tabel tersebut adalah: $id_jawaban \rightarrow id_soal$ $jawaban\ status$. Dengan demikian tabel ini telah memenuhi BCNF karena $\{id_jawaban\}$ merupakan *super-key* dari Tabel **Jawaban** tersebut.

g. Untuk Tabel Nilai

Berdasarkan Tabel 3.15, *FD* pada tabel tersebut adalah: $id_nilai \rightarrow id_user$ $id_ujian\ nilai$. Dengan demikian tabel ini telah memenuhi BCNF karena $\{id_nilai\}$ merupakan *super-key* dari Tabel **Nilai** tersebut.