

BAB IV

HASIL DAN PEMBAHASAN

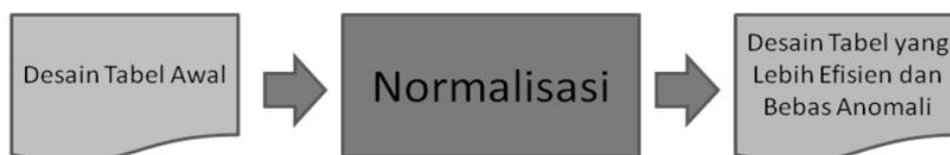
4.1 Normalisasi

Setelah tabel ER Diagram dibuat maka tahap selanjutnya tabel tabel tersebut diperiksa validasinya dan diperbaiki (dinormalisasikan). Tujuan dilakukan normalisasi adalah

- menghilangkan rangkap data sehingga berdampak kepada pemborosan memori.
- untuk menghindari *anomali* (ketidak konsistenan data).

Pada dasarnya normalisasi dilakukan untuk memperbaiki desain tabel yang kurang baik sehingga penyimpanan data menjadi lebih efisien dan bebas anomali data.

Untuk memperjelas pemahaman tentang proses normalisasi, perhatikan diagram berikut:



Gambar 4. 1 Diagram normalisasi

adapun tabel yang di normalisasikan berdasarkan desain awal tabel yaitu, tabel tempat wisata dan trip.

4.1.1 Normalisasi Tabel Tempat Wisata

Berikut adalah desain tabel awal

id_tempa	nama	id_daerah	nama_daerah	id_jenis	nama_jenis	gambar	koordinat	budget	durasi	informasi_umum
1	parangtritis	3	bantul	2	wisata alam	gambar 12,gambar 23	7607863	100000	3 jam	wisata pantai
2	malioboro	1	jogja kota	1	wisata belar	gambar 13,gambar 24	7607860	1000000	4 jam	wisata belanja

Gambar 4. 2 Desain Awal Tabel Tempat Wisata

Berikut adalah bentuk tidak normal dari tabel tempat wisata:

id_tempa	nama	id_daerah	nama_daerah	id_jenis	nama jenis	gambar	koordinat	budget	durasi	informasi_umum
1	parangtritis	3	bantul	2	wisata alam	gambar 12,gambar 23	7607863	100000	3 jam	wisata pantai
2	malioboro	1	jogja kota	1	wisata belar	gambar 13,gambar 24	7607860	1000000	4 jam	wisata belanja

Gambar 4. 3 Tabel Tempat Wisata Tidak Normal

Pada kolom gambar ditunjukkan pada Gambar 4.3 masih mempunyai data ganda

maka perlu di normalisasikan mendaji 1NF ditunjukkan pada Gambar 4.4

id_tempa	nama	id_daerah	nama_daerah	id_jenis	nama jenis	gambar	koordinat	budget	durasi	informasi_umum
1	parangtritis	3	bantul	2	wisata alam	gambar 12	7607863	100000	3 jam	wisata pantai
1	parangtritis	3	bantul	2	wisata alam	gambar 23	7607863	100000	3 jam	wisata pantai
2	malioboro	1	jogja kota	1	wisata belar	gambar 13	7607860	1000000	4 jam	wisata belanja
2	malioboro	1	jogja kota	1	wisata belar	gambar 24	7607860	1000000	4 jam	wisata belanja

Gambar 4. 4 Tabel Tempat Wisata Bentuk 1NF

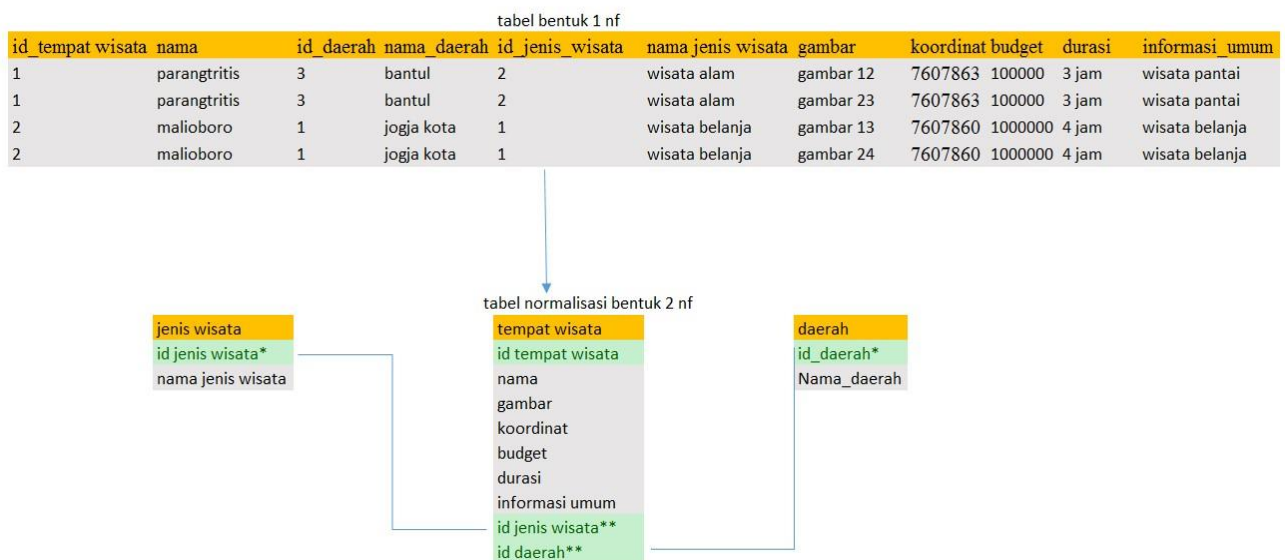
Pada kolom gambar di tunjukan pada Gambar 4.4 terdapat ketergantungan

fungsional antara kolom id daerah dengan nama daerah dan id jenis dengan nama

jenis, sehingga dapat tabel tempat wisata data diubah menjadi 2nf, dengan cara

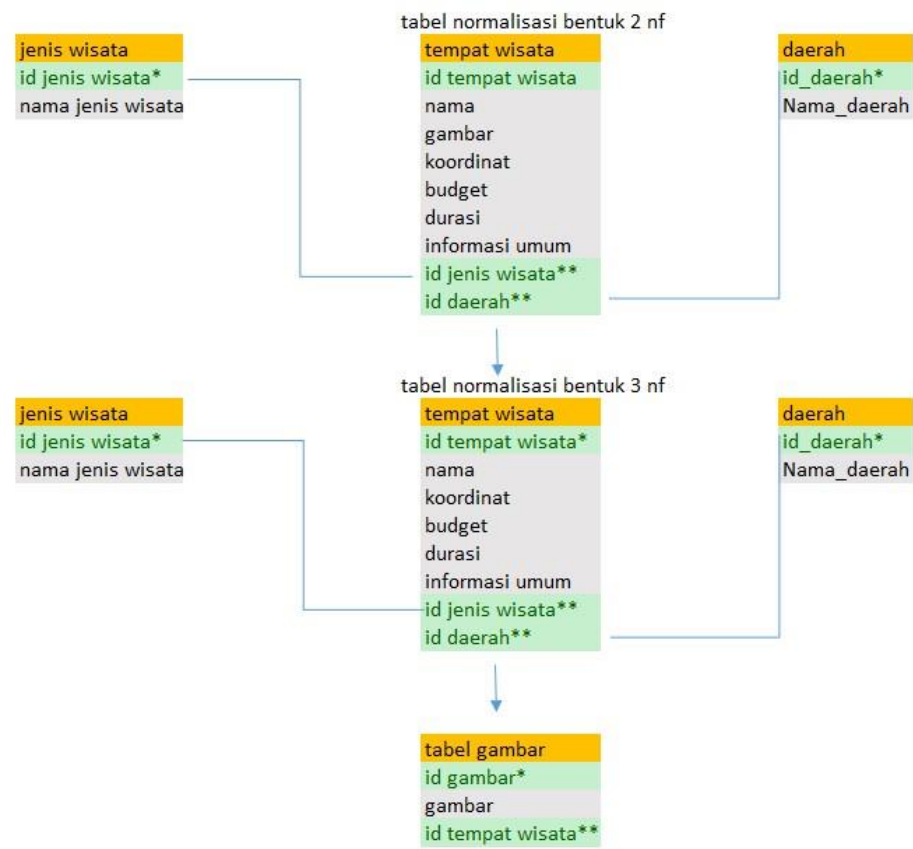
memisahkan kolom yang memiliki ketergantungan fungsional dapat dilihat pada

Gambar 4.5 Tempat Wisata Bentuk 2NF.



Gambar 4. 5 Tempat Wisata Bentuk 2NF

Pada gambar 4.4 masih terdapat ketergantungan *transitif* antara kolom id tempat wisata, nama dengan gambar, dan terdapat anomaly dimana kolom gambar mempunyai 4 data yang berbeda sedangkan tempat wisata hanya ada 2 data yang berbeda, maka perlu dilakukan normalisasi 3NF. ditunjukkan pada Gambar 4.6.



Gambar 4. 6 Tempat Wisata Bentuk 3NF

4.1.2 Normalisasi Tabel Trip

Berikut desain rancangan tabel trip

id	id trip	session start	status	judul	tanggal	id member	nama	tlp	alamat	...
1	1	1	1	jalan jalan 1	22/09/2016	1	wisnu	987655	gidean	...

user	id tem	jenis	starting	end point	posisi	jam	durasi	budget
wisr	23,12		1	1	1,2	7	2,4	80000 dan 10000

Gambar 4. 7 Desain Awal Tabel Trip.

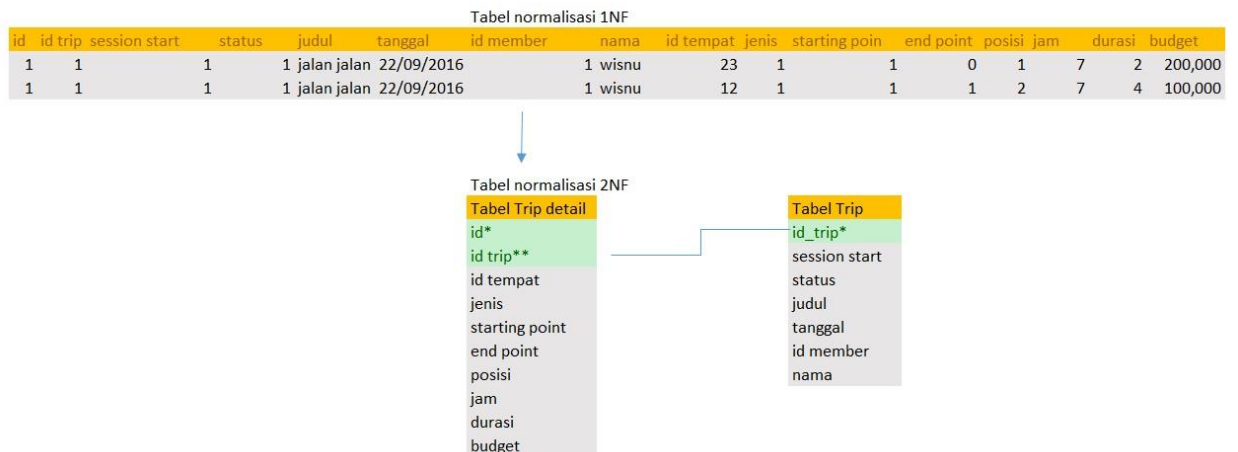
Pada Gambar 4.7 Tabel masih dalam bentuk tidak normal, ada terdapat data bernilai ganda. Sehingga dapat dinormalisasikan ke bentuk 1NF ditunjukkan pada Gambar 4.8.

Tabel normalisasi 1NF

id	id trip	session start	status	judul	tanggal	id member	nama	id tempat	jenis	starting poin	end point	posisi	jam	durasi	budget
1	1	1	1	1 jalan jalan	22/09/2016	1	wisnu	23	1	1	0	1	7	2	200,000
1	1	1	1	1 jalan jalan	22/09/2016	1	wisnu	12	1	1	1	2	7	4	100,000

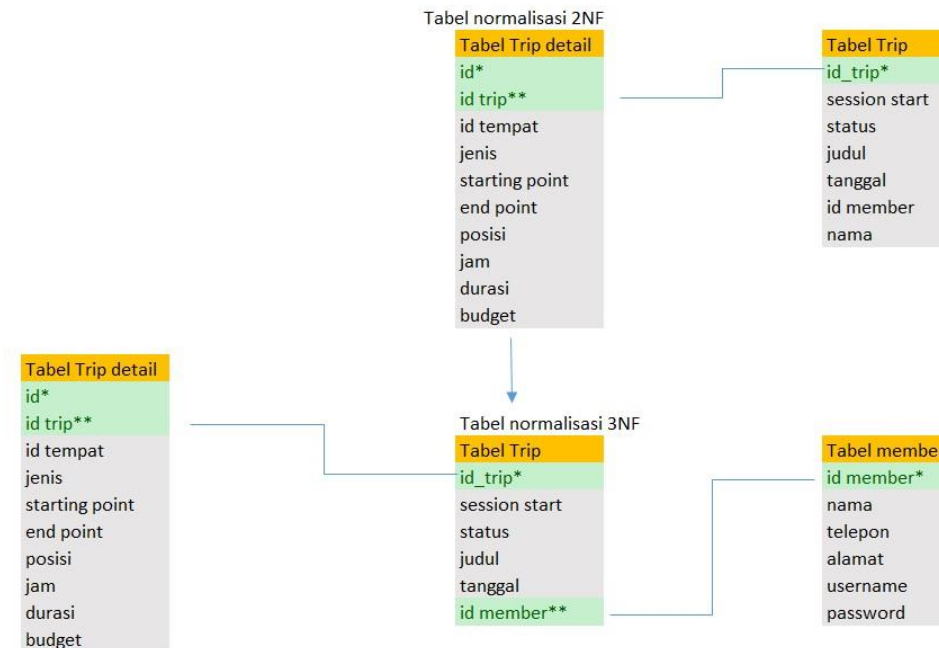
Gambar 4. 8 Tabel Trip Bentuk 1NF

Dari Gambar 4.8 Tabel Trip Bentuk 1NF trip banyak terdapat ketergantungan fungsional seperti pada kolom id trip, session start, judul, tanggal, id member dan nama. oleh sebab itu perlu dilakukan normalisasi dengan menghilangkan ketergantungan



Gambar 4. 9 Trip Bentuk 2NF

pada Gambar 4.9. Pada tabel trip masih mempunyai ketergantungan fungsional transitif antara kolom id trip, id member dan nama. Setelah dinormalisasikan ketergantungan fungsi tansitif dihilangkan dan tabel trip menjadi 3NF ditunjukkan pada Gambar 4.10.



Gambar 4. 10 Trip Bentuk 3NF

4.2 Tabel

Setelah melakukan proses normalisasi maka terbentuk beberapa tabel, tabel-tabel tersebut berupa tabel *trip*, tabel *trip detail*, tabel admin, tabel daerah, tabel gambar wisata, tabel hotel, tabel jenis wisata, tabel *member*, tabel tempat wisata, tabel tempat wisata like, tabel testimoni, tabel *transport*.

4.2.1 Tabel *Trip*

Tabel perjalanan adalah tabel yang menunjukkan data perjalanan. Tabel perjalanan dapat dilihat pada gambar 4.11

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	<i>id</i> 🔑	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	<i>session_start</i>	int(1)			Tidak	Tidak ada	
3	<i>status</i>	int(1)			Tidak	Tidak ada	
4	<i>judul</i>	varchar(40)			Tidak	Tidak ada	
5	<i>tanggal</i>	date			Tidak	Tidak ada	
6	<i>id_member</i> 🗝️	int(11)			Tidak	Tidak ada	

Gambar 4. 11 Tabel Perjalanan

Penjelasan pada gambar 4.11 tabel perjalanan

- 1) *Id* adalah kolom yang menunjukkan *id* perjalanan, *id* perjalanan bersifat *auto increment* dan *primary key*.
- 2) *Session start* adalah kolom yang menunjukkan *starting point*. *Session start* bertipe INT, jika *Session start* 1 maka *starting point* sudah di tambahkan.
- 3) *Status* adalah kolom yang menunjukkan *status*. *Session end* bertipe INT, jika *status* 1 berarti data perjalanan sudah di simpan.
- 4) Judul adalah kolom yang menunjukkan judul perjalanan bertipe varchar.
- 5) Tanggal adalah kolom yang menunjukkan tanggal perjalanan bertipe date.
- 6) *Id member* adalah kolom yang menunjukkan *id member* dari tabel *member*, dan merupakan foreign Key.

4.2.2 Tabel *Trip* detail

Tabel detail perjalanan adalah tabel yang menunjukkan data detail perjalanan .

Tabel detail perjalanan dapat dilihat pada gambar 4.12

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id 🔑	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	id_trip 🔑	int(11)			Tidak	Tidak ada	
3	id_tempat	int(11)			Tidak	Tidak ada	
4	jenis	int(1)			Tidak	Tidak ada	
5	starting_point	int(1)			Tidak	Tidak ada	
6	end_point	int(1)			Tidak	Tidak ada	
7	posisi	int(2)			Tidak	Tidak ada	
8	jam	int(2)			Tidak	Tidak ada	
9	durasi	int(2)			Tidak	Tidak ada	
10	budget	double			Tidak	Tidak ada	

Gambar 4. 12 Tabel Detail Perjalanan

Penjelasan pada Gambar 4.12

- 1) *Id* adalah kolom yang menunjukkan *id* detail perjalanan, *id* detail perjalanan bersifat *auto increment* dan *primary key*, bertipe INT.
- 2) *Id trip* adalah kolom yang menunjukkan *Id trip* dari tabel perjalanan, dan *Foreign Key* bertipe INT.
- 3) *Id tempat* adalah kolom yang menunjukkan *id* tempat wisata yang dikunjungi termasuk *id* hotel, dan bertipe INT
- 4) *Jenis* adalah kolom yang menunjukkan jenis wisata, jenis wisata bertipe INT , yang mana jika jenis wisata sama dengan 1 maka itu berarti hotel, dan jika 0 berarti wisata.
- 5) *Starting point* adalah kolom yang menunjukkan *starting point*. *Starting point* bertipe INT, jika *starting point* 1 maka *starting point* sudah di tambahkan.
- 6) *End point* adalah kolom yang menunjukkan *end point*. *End point* bertipe INT, jika *end point* 1 maka *end point* sudah ditambahkan.
- 7) *Posisi* adalah kolom yang menunjukkan nomor urutan tempat wisata, dan bertipe INT
- 8) *Jam* adalah kolom yang menunjukkan jam perjalanan detail, dan bertipe INT
- 9) *Durasi* adalah kolom yang menunjukkan durasi perjalanan detail, dan bertipe INT
- 10) *Budget* adalah kolom yang menunjukkan biaya perjalanan detail, dan bertipe DOUBLE.

4.2.3 Tabel Admin

Tabel admin adalah tabel yang menunjukkan data admin. Tabel admin dapat dilihat pada gambar 4.13

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id_auth 	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	nama	varchar(30)			Tidak	Tidak ada	
3	hp	varchar(20)			Tidak	Tidak ada	
4	alamat	text			Tidak	Tidak ada	
5	username	varchar(20)			Tidak	Tidak ada	
6	password	varchar(40)			Tidak	Tidak ada	
7	level	int(1)			Tidak	Tidak ada	
8	is_delete	int(1)			Tidak	Tidak ada	

Penjelasan pada gambar 4.13

1. *Id auth* adalah kolom yang menunjukkan *ID* admin, bersifat *auto increment* dan *Primary key*, dan bertipe INT
2. Nama adalah kolom yang menunjukkan nama admin, dan bertipe varchar.
3. Hp adalah kolom yang menunjukkan nomor hp admin, dan bertipe varchar.
4. Alamat adalah kolom yang menunjukkan alamat admin, dan bertipe text.
5. *Username* adalah kolom yang menunjukkan *username* admin, dan bertipe varchar.
6. *Password* adalah kolom yang menunjukkan *password* admin, dan bertipe varchar.
7. *Level* adalah kolom yang menentukan apakah admin atau bukan. Kolom ini bersifat INT, jika angka 0 maka menunjukkan admin, dan jika selain 0 berarti bukan admin dan tidak bisa login.

8. *Is delete* adalah kolom yang menentukan data yang sudah dihapus, atau belum.

Is delete berupa INT, jika *is delete* menunjukkan angka 1 maka data sudah dihapus di website admin. Tetapi masih ada di *database*

4.2.4 Tabel Daerah

Tabel daerah adalah tabel yang menunjukkan data daerah. Tabel daerah dapat dilihat pada gambar 4.14

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id 	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	nama	varchar(30)			Tidak	Tidak ada	
3	is_delete	int(1)			Tidak	Tidak ada	


Gambar 4. 14 Tabel Daerah

Penjelasan pada gambar 4.14

1. *Id* adalah kolom yang menunjukkan *Id* daerah, *Id* daerah bersifat *auto increment* dan *primary key*, dan bertipe INT.
2. Nama adalah kolom yang menunjukkan nama daerah, dan bertipe varchar.
3. *Is delete* adalah kolom yang menentukan data yang sudah dihapus, atau belum. *Is delete* berupa INT, jika *is delete* menunjukkan angka 1 maka data sudah dihapus di website admin. Tetapi masih ada di *database*.

4.2.5 Tabel Jenis wisata

Tabel Jenis wiata adalah tabel yang menunjukkan data admin. Tabel jenis wisata dapat dilihat pada gambar 4.15

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 	int(11)			No	None	AUTO_INCREMENT
2	nama	varchar(30)			No	None	
3	is_delete	int(1)			No	None	

Gambar 4. 15 Tabel Jenis Wisata

Penjelasan pada gambar 4.15

1. *Id* adalah kolom yang menunjukkan *Id* jenis wisata, *Id* jenis wisata bersifat *auto increment* dan *primary key*, dan bertipe INT.
2. Nama adalah kolom yang menunjukkan nama jenis wisata, dan bertipe varchar.
3. *Is delete* adalah kolom yang menentukan data yang sudah dihapus, atau belum. *Is delete* berupa INT, jika *is delete* menunjukkan angka 1 maka data sudah dihapus di website admin. Tetapi masih ada di *database*.

4.2.6 Tabel Wisata

Tabel wisata adalah tabel yang menunjukkan data wisata. Tabel wisata dapat dilihat pada gambar 4.16

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id 🔑	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	nama	varchar(50)			Tidak	Tidak ada	
3	id_daerah 🔑	int(11)			Tidak	Tidak ada	
4	id_jenis_wisata 🔑	int(11)			Tidak	Tidak ada	
5	koordinat	text			Tidak	Tidak ada	
6	estimasi_budget	double			Tidak	Tidak ada	
7	estimasi_durasi	varchar(5)			Tidak	Tidak ada	
8	informasi_umum	text			Tidak	Tidak ada	
9	is_delete	int(1)			Tidak	Tidak ada	

Gambar 4. 16 Tabel Wisata

Penjelasan pada gambar 4.16

1. *Id* adalah kolom yang menunjukkan *id* wisata, *id* wisata bersifat *auto increment* dan *primary key*, dan bertipe INT.
2. Nama adalah kolom yang menunjukkan nama wisata, dan bertipe varchar.

3. *Id daerah* adalah kolom yang berhubungan dengan tabel daerah. *Id daerah* menunjukkan nama daerah, yang berdasarkan *Id daerah* yang tersimpan di tabel wisata, merupakan *Foreign Key*, dan bertipe INT.
4. *Id jenis wisata* adalah kolom yang berhubungan dengan tabel jenis wisata. *Id jenis wisata* menunjukkan nama jenis wisata, yang berdasarkan *Id jenis wisata* yang tersimpan di tabel wisata, merupakan *Foreign Key*, dan bertipe INT.
5. *Koordinat* adalah kolom yang menunjukkan titik koordinat wisata dan bertipe text.
6. *Estimasi budget* adalah kolom yang menunjukkan perkiraan biaya wisata dan bertipe *double*.
7. *Estimasi durasi* adalah kolom yang menunjukkan perkiraan waktu wisata dan bertipe *varchar*.
8. Informasi umum adalah kolom yang menunjukkan informasi umum wisata dan bertipe text.
9. *Is delete* adalah kolom yang menentukan data yang sudah dihapus, atau belum. *Is delete* berupa INT, jika *is delete* menunjukkan angka 1 maka data sudah dihapus di website admin. Tetapi masih ada di *database*.

4.2.7 Tabel Hotel

Tabel hotel adalah tabel yang menunjukkan data hotel. Tabel hotel dapat dilihat pada gambar 4.17

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id 	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	nama	varchar(50)			Tidak	Tidak ada	
3	koordinat	text			Tidak	Tidak ada	
4	alamat	text			Tidak	Tidak ada	
5	bintang	int(1)			Tidak	Tidak ada	
6	website	varchar(80)			Tidak	Tidak ada	
7	gambar	varchar(50)			Tidak	Tidak ada	
8	harga	double			Tidak	Tidak ada	
9	informasi	text			Tidak	Tidak ada	
10	is_delete	int(1)			Tidak	Tidak ada	
11	id_daerah 	int(11)			Tidak	Tidak ada	

Gambar 4. 17 Tabel Hotel

Penjelasan pada gambar 4.17 tabel hotel


1. *Id* adalah kolom yang menunjukkan *id* hotel, *id* hotel bersifat *auto increment* dan *primary key* dan bertipe INT.
2. Nama adalah kolom yang menunjukkan nama hotel dan bertipe varchar.
3. *Koordinat* adalah kolom yang menunjukkan titik koordinat hotel dan bertipe text.
4. Alamat adalah kolom yang menunjukkan alamat hotel, dan bertipe text.
5. Bintang adalah kolom yang menunjukkan bintang hotel, dan bertipe INT.
6. Website adalah kolom yang menunjukkan website hotel, dan bertipe varchar.
7. Gambar adalah kolom menunjukkan gambar hotel, dan bertipe varchar.
8. Harga adalah kolom menunjukkan harga hotel, dan bertipe double.
9. Informasi adalah kolom menunjukkan informasi hotel, dan bertipe text.

10. *Is delete* adalah kolom yang menentukan data yang sudah dihapus, atau belum. *Is delete* berupa INT, jika *is delete* menunjukkan angka 1 maka data sudah dihapus di *website* admin. Tetapi masih ada di *database*.
11. *Id daerah* adalah kolom yang berhubungan dengan tabel daerah. *Id daerah* menunjukkan nama daerah, yang berdasarkan *Id daerah* yang tersimpan di tabel hotel, merupakan *foreign Key* dan bertipe int.

4.2.8 Tabel Transportasi

Tabel Transportasi adalah tabel yang menunjukkan data Transportasi. Tabel

Transportasi dapat dilihat pada gambar 4.18

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 	int(11)			No	None	AUTO_INCREMENT
2	nama	varchar(50)			No	None	
3	jenis_usaha	varchar(50)			No	None	
4	alamat	text			No	None	
5	tlp	varchar(20)			No	None	
6	gambar	varchar(50)			No	None	
7	is_delete	int(1)			No	None	

Gambar 4. 18 Tabel Transportasi

Penjelasan pada gambar 4.18

- 1) *Id* adalah kolom yang menunjukkan *id* transportasi, *id* transportasi bersifat *auto increment* dan *primary key*.
- 2) Nama adalah kolom yang menunjukkan nama transportasi.
- 3) Jenis usaha adalah kolom yang menunjukkan jenis usaha transportasi.
- 4) Alamat adalah kolom yang menunjukkan alamat transportasi.
- 5) Tlp adalah kolom yang menunjukkan nomor telepon transportasi.
- 6) Gambar adalah kolom yang menunjukkan gambar transportasi.



7) *Is delete* adalah kolom yang menentukan data yang sudah dihapus, atau belum.

Is delete berupa INT, jika *is delete* menunjukkan angka 1 maka data sudah dihapus di website admin. Tetapi masih ada di *database*.

4.2.9 Tabel Gambar wisata

Tabel Gambar wisata adalah tabel yang menunjukkan data Gambar wisata. Tabel

Gambar wisata dapat dilihat pada gambar 4.19

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id 	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	id_tempat_wisata 	int(11)			Tidak	Tidak ada	
3	gambar	varchar(30)			Tidak	Tidak ada	



Gambar 4. 19 Tabel Gambar Wisata

Penjelasan pada gambar 4.19

- 1) *Id* adalah kolom yang menunjukkan *id* gambar wisata, *id* gambar wisata bersifat *auto increment* dan *primary key*, dan bertipe INT.
- 2) *Id* tempat wisata adalah kolom yang menampilkan *Id* tempat wisata pada tabel wisata, merupakan *foreign Key* dan bertipe INT.
- 3) *Gambar* adalah kolom yang menunjukkan gambar wisata, dan bertipe varchar.

4.2.10 Tabel Wisata Like

Tabel wisata like adalah tabel yang menunjukkan data wisata like. Tabel wisata like dapat dilihat pada gambar 4.20 tabel wisata like

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra
1	id 	int(11)			Tidak	Tidak ada	AUTO_INCREMENT
2	id_tempat_wisata 	int(11)			Tidak	Tidak ada	
3	id_member 	int(11)			Tidak	Tidak ada	


Gambar 4. 20 Tabel Wisata Like

Penjelasan pada gambar 4.20 tabel wisata like

- 1) *Id* adalah kolom yang menunjukkan *id* wisata like, *id* wisata bersifat *auto increment* dan *primary key*, dan bertipe INT.
- 2) *Id* tempat wisata adalah kolom yang menampilkan Id tempat wisata pada tabel wisata, merupakan *foreign Key* dan bertipe INT.
- 3) *Id Member* adalah kolom yang menampilkan id *member* yang terdapat pada tabel *member*, merupakan *foreign Key* dan bertipe INT.

4.2.11 Tabel Member

Tabel *Member* adalah tabel yang menunjukkan data *Member*. Tabel *Member* dapat dilihat pada gambar 4.21 tabel *Member*

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id 	int(11)			No	None	AUTO_INCREMENT
2	nama	varchar(40)			No	None	
3	tlp	varchar(20)			No	None	
4	alamat	text			No	None	
5	username	varchar(30)			No	None	
6	password	varchar(32)			No	None	
7	email	varchar(50)			No	None	

Gambar 4. 21 Tabel Member

Penjelasan pada gambar 4.21 tabel *Member*

- 1) *Id* adalah kolom yang menunjukkan *id member*, *id member* bersifat *auto increment* dan *primary key*.
- 2) Nama adalah kolom yang menunjukkan nama *member*
- 3) Tlp adalah kolom yang menunjukkan nomor telepon *member*
- 4) Alamat adalah kolom yang menunjukkan alamat *member*
- 5) *Username* ada kolom yang menunjukkan *Username member*
- 6) *Password* adalah kolom yang menunjukkan *password member*
- 7) *Email* adalah kolom yang menunjukkan *email member*

4.2.12 Tabel Testimoni

Tabel Testimoni adalah tabel yang menunjukkan data Testimoni. Tabel Testimoni dapat dilihat pada gambar 4.22 tabel Testimoni

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	nama	varchar(40)			No	None	
3	email	varchar(50)			No	None	
4	waktu	datetime			No	None	
5	pesan	text			No	None	

Gambar 4. 22 Tabel Testimoni

Penjelasan pada gambar 4.22 tabel Testimoni

- 1) *Id* adalah kolom yang menunjukkan *id testimoni*, *id testimoni* bersifat *auto increment* dan *primary key*.
- 2) Nama adalah kolom yang menunjukkan nama *responden testimoni*
- 3) *Email* adalah kolom yang menunjukkan *email responden testimoni*

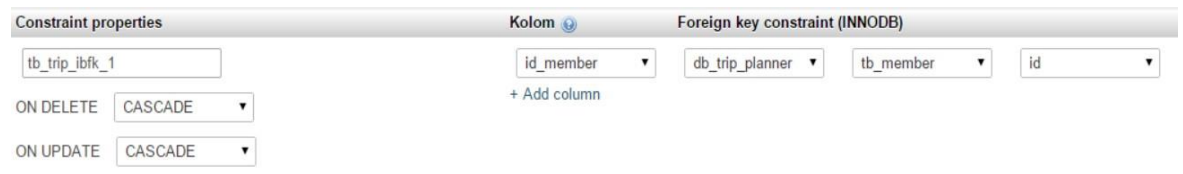
- 4) Waktu adalah kolom yang menunjukkan waktu testimoni
- 5) Pesan adalah kolom yang menunjukkan isi pesan testimony

4.3 Foreign Key Constrains

Foreign Key Constrains adalah *constrains* suatu kolom yang ada pada suatu tabel, dimana kolom tersebut juga dimiliki oleh tabel yang lain sebagai suatu Primary key. Adapun beberapa tabel yang memiliki *Foreign Key Constrains*, seperti tabel *trip*, *trip detail*, gambar wisata, hotel, tempat wisata, tempat wisata *like*.

4.3.1 Tabel Trip

Constrains tabel *trip* dapat dilihat pada gambar 4.23



Gambar 4. 23 Constrains pada tabel *trip*

Penjelasan pada gambar 4.23

Tabel *trip* mempunyai *constrains on delete cascade* dan terhubung dengan tabel *member* yang artinya jika kolom tabel *member* dihapus, maka kolom tabel *trip* akan ikut juga terhapus. Tabel *trip* mempunyai *constrains on update cascade* dan terhubung dengan tabel *member* yang artinya jika kolom tabel *member* diubah, maka kolom tabel *trip* akan ikut juga berubah.

4.3.2 Tabel *Trip Detail*

Constrains tabel *trip* dapat dilihat pada gambar 4.24



Gambar 4. 24 Constrains pada tabel *trip detail*

Penjelasan pada gambar 4.24

Tabel *trip detail* mempunyai *constrains on delete cascade* dan terhubung dengan tabel *trip* yang artinya jika kolom tabel *trip* dihapus, maka kolom tabel *trip detail* akan ikut juga terhapus. Tabel *trip detail* mempunyai *constrains on update cascade* dan terhubung dengan tabel *trip* yang artinya jika kolom tabel *trip* diubah, maka kolom tabel *trip detail* akan ikut juga berubah.

4.3.3 Tabel Gambar Wisata

Constrains tabel *trip* dapat dilihat pada gambar 4.25



Gambar 4. 25 Constrains pada tabel gambar wisata

Penjelasan pada gambar 4.25

Tabel gambar wisata mempunyai *constrains on delete cascade* dan terhubung dengan tabel tempat wisata yang artinya jika kolom tabel tempat wisata dihapus, maka kolom tabel gambar wisata akan ikut juga terhapus. Tabel gambar wisata mempunyai *constrains on update cascade* dan terhubung dengan tabel

tempat wisata yang artinya jika kolom tabel tempat wisata diubah, maka kolom tabel gambar wisata akan ikut juga berubah.

4.3.4 Tabel Hotel

Constrains tabel *trip* dapat dilihat pada gambar 4.26

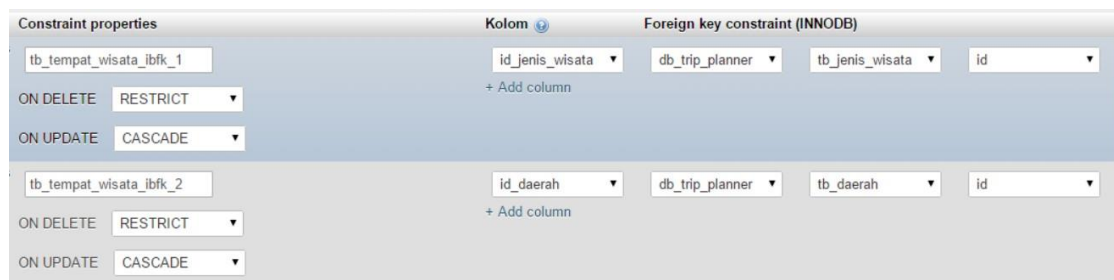


Gambar 4. 26 Constrains pada tabel hotel

Penjelasan pada gambar 4.26

Tabel hotel mempunyai *constrains on delete restrict* dan terhubung dengan tabel daerah yang artinya kolom tabel daerah tidak dapat dihapus, sebelum menghapus kolom tabel hotel terlebih dahulu. Tabel hotel mempunyai *constrains on update cascade* dan terhubung dengan tabel daerah yang artinya jika kolom tabel daerah diubah, maka kolom hotel akan ikut juga berubah.

4.3.5 Tabel Tempat Wisata



Gambar 4. 27 Constrains pada tabel tempat wisata

Constrains tabel *trip* dapat dilihat pada gambar 4.27

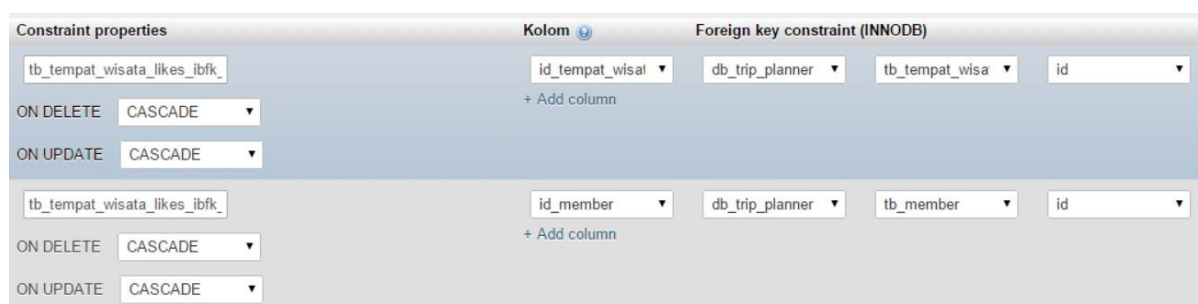
Penjelasan pada gambar 4.27

Tabel tempat wisata mempunyai *constrains on delete restrict* dan terhubung dengan tabel jenis wisata yang artinya kolom tabel jenis wisata tidak dapat dihapus, sebelum menghapus kolom tabel tempat wisata terlebih dahulu. Tabel tempat wisata mempunyai *constrains on update cascade* dan terhubung dengan tabel jenis wisata yang artinya jika kolom tabel jenis wisata diubah, maka kolom tempat wisata akan ikut juga berubah.

Tabel tempat wisata mempunyai *constrains on delete restrict* dan terhubung dengan tabel daerah yang artinya kolom tabel daerah tidak dapat dihapus, sebelum menghapus kolom tabel tempat wisata terlebih dahulu. Tabel tempat wisata mempunyai *constrains on update cascade* dan terhubung dengan tabel daerah yang artinya jika kolom tabel daerah diubah, maka kolom tempat wisata akan ikut juga berubah.

4.3.6 Tabel Wisata Like

Constrains tabel *trip* dapat dilihat pada gambar 4.28



Gambar 4. 28 Constrains pada tabel wisata like

Penjelasan pada gambar 4.28

Tabel wisata *like* mempunyai *constrains on delete cascade* dan terhubung dengan tabel tempat wisata yang artinya jika kolom tabel tempat wisata dihapus, maka kolom tabel wisata *like* akan ikut juga terhapus. Tabel wisata *like* mempunyai *constrains on update cascade* dan terhubung dengan tabel tempat wisata yang artinya jika kolom tabel tempat wisata diubah, maka kolom tabel wisata *like* akan ikut juga berubah.

Tabel wisata *like* mempunyai *constrains on delete cascade* dan terhubung dengan tabel *member* yang artinya jika kolom tabel *member* dihapus, maka kolom tabel wisata *like* akan ikut juga terhapus. Tabel wisata *like* mempunyai *constrains on update cascade* dan terhubung dengan tabel *member* yang artinya jika kolom tabel *member* diubah, maka kolom tabel wisata *like* akan ikut juga berubah.

4.4 Procedure

Stored procedure adalah salah satu objek routine yang tersimpan pada *database* MySQL dan dapat digunakan untuk menggantikan berbagai kumpulan perintah yang sering kita gunakan, dalam *database* ini terdapat 2 perintah *procedure* yaitu *procedure* hapus tempat wisata dan *procedure* hapus hotel.

4.4.1 Procedure Hapus Tempat

Constrains tabel *trip* dapat dilihat pada gambar 4.29

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE
  `hapuswisata`(IN `_idtempat` INT(11))
2 BEGIN
3     DELETE FROM tb_trip_detail WHERE id_tempat = _idtempat
  AND jenis = 0;
4     DELETE FROM tb_tempat_wisata WHERE id = _idtempat;
5 END
```

Gambar 4. 29 procedure hapus wisata

Penjelasan pada gambar 4.29

Untuk menghapus tempat wisata diperlukan procedure hapus wisata, yang dimana perintah tersebut berfungsi untuk menghapus kolom tabel trip detail dahulu, baru menghapus tabel tempat wisata.

4.4.2 Procedure Hapus Hotel

Constrains tabel *trip* dapat dilihat pada gambar 4.30

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `hapushotel` (IN
  `_idtempat` INT(11))
2 BEGIN
3   DELETE FROM tb_trip_detail WHERE id_tempat = _idtempat
  AND jenis = 1;
4   DELETE FROM tb_hotel WHERE id = _idtempat;
5 END
```

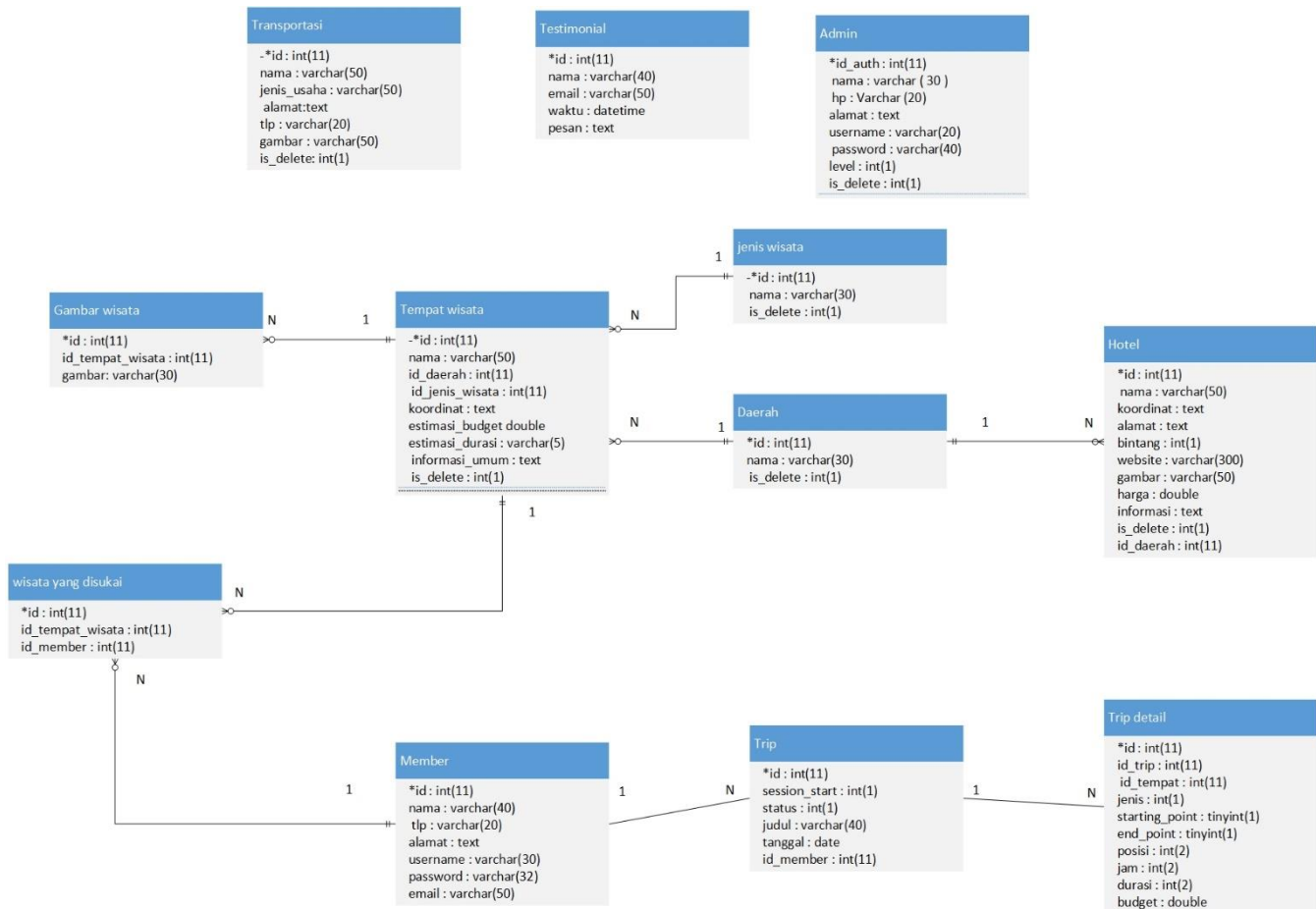
Gambar 4. 30 procedure hapus hotel

Penjelasan pada gambar 4.30

Untuk menghapus tempat wisata diperlukan procedure hapus hotel, yang dimana perintah tersebut berfungsi untuk menghapus kolom tabel trip detail dahulu, baru menghapus tabel hotel.

4.5 RAT (Rancangan Antar Tabel)

Setelah diperoleh tabel tabel hasil normalisasi maka di bentuklah rancangan antar tabel di tunjukan pada gambar 4.31.



Gambar 4. 31 RAT tabel

4.6 Pengujian

4.6.1 Metode Pengujian

Pengujian perangkat lunak dilakukan untuk memperoleh informasi serta mengevaluasi kualitas dari produk atau layanan yang sedang diuji. Tujuan pengujian dalam pengembangan database adalah untuk mengetahui apakah database

yang diuji dapat memenuhi kebutuhan *admin* dan *member* dengan mendasari pada rancangan dan pengembangan perangkat lunak.

Metode pengujian yang dipakai dalam pengembangan *database* adalah *anomaly testing*. Pengujian *anomaly testing* dilakukan melalui page admin. *Anomaly testing* berfungsi untuk mengetahui apakah proses basis data yang *memberikan* efek samping yang tidak diharapkan (misalnya menyebabkan ketidakonsistenan data atau membuat suatu data menjadi hilang ketika data dihapus)

4.6.2 Pengujian Anomali

Pengujian *anomaly* di lakukan terhadap tabel tabel memiliki relasi, misalnya tabel daerah , berhubungan dengan tabel tempat wisata ,berhubungan dengan tabel hotel, berhubungan dengan tabel jenis wisata.pengujian di lakukan sebanyak 5 kali *insert anomaly, delete anomaly dan update anomaly.*

Insert Anomali

Insert anomaly dilakukan pada tabel daerah karena tabel daerah merupakan bagian dari tabel tempat wisata dan hotel. Dapat dilihat pada tabel 4.1 pengujian insert anomali.

Tabel 4. 1 pengujian *insert* anomali

Tabel daerah	Tabel yang berhubungan	
	Tabel tempat wisata	Tabel hotel
<i>Insert pertama</i>	Tidak ada perubahan	Tidak ada perubahan
<i>Insert kedua</i>	Tidak ada perubahan	Tidak ada perubahan
<i>Insert ketiga</i>	Tidak ada perubahan	Tidak ada perubahan

Tabel 4. 2 pengujian *insert* anomali lanjutan

<i>Insert</i> keempat	Tidak ada perubahan	Tidak ada perubahan
<i>Insert</i> kelima	Tidak ada perubahan	Tidak ada perubahan

Setelah 5 kali di lakukan *insert* data sebanyak 5 kali pada tabel daerah , tidak ada perubahan yang terjadi di tabel tempat wisata dan hotel .maka tidak ada terjadi *anomali* pada pemasukan data

Update Anomali

Update anomali dilakukan pada tabel daerah karena tabel daerah merupakan bagian dari tabel tempat wisata dan hotel.

Tabel 4. 3 pengujian *update* anomali

Tabel daerah	Tabel yang berhubungan	
	Tabel tempat wisata	Tabel hotel
<i>Update</i> pertama	ada perubahan	ada perubahan
<i>Update</i> kedua	ada perubahan	ada perubahan
<i>Update</i> ketiga	ada perubahan	ada perubahan
<i>Update</i> keempat	ada perubahan	ada perubahan
<i>Update</i> kelima	ada perubahan	ada perubahan

Pada Tabel 4.2 pengujian *update anomali* dilakukan *update* sebanyak 5 kali tabel yang memiliki hubungan dan memiliki data yang sama akan berubah mengikuti data yang di *update*, menandakan bahwa tidak terjadi *anomali* saat data di *update*.

Delete Anomali

Delete anomali dilakukan pada tabel daerah karena tabel daerah merupakan bagian dari tabel tempat wisata dan hotel

Tabel 4. 4 pengujian *delete* anomali

Tabel daerah	Tabel yang berhubungan	
	Tabel tempat wisata	Tabel hotel
Tidak dapat dihapus	Tidak ada perubahan	Tidak ada perubahan
Tidak dapat dihapus	Tidak ada perubahan	Tidak ada perubahan
Tidak dapat dihapus	Tidak ada perubahan	Tidak ada perubahan
Tidak dapat dihapus	Tidak ada perubahan	Tidak ada perubahan
Tidak dapat dihapus	Tidak ada perubahan	Tidak ada perubahan

Pada Tabel 4.3 dilakukan *delete* sebanyak 5 kali, dan data tabel daerah tidak dapat dihapus, maka dari itu tidak ada perubahan yang terjadi di tabel tempat wisata dan hotel walaupun memiliki data yang sama . maka tidak terjadi anomali saat data dihapus.