

TIPE DATA

Numerik

- Integer
- Decimal
- Float, Double

String

- Char, varchar
- Binary, Varbinary
- Text
- Blob

Date

- Date
- Time
- Datetime
- Timestamp
- Year

Enum

- Enum
- Set

TIPE DATA NUMERIK

Integer



- Tinyint
- Smallint
- Mediumint
- Integer (Int)
- Bigint

Decimal



- Decimal (4,1)
- Decimal (6,2)
- Decimal (3,2)
- Decimal (8,2)

Float dan Double



- Float
- Double

Integer

Tipe Data	Jangkauan SIGNED	Jangkauan UNSIGNED	Ukuran
TINYINT	-128 to 127	0 to 255	1 byte
SMALLINT	-32,768 to 32,767	0 to 65,535	2 bytes
MEDIUMINT	-8,388,608 to 8,388,607	0 to 16,777,215	3 bytes
INT	-2,147,483,648 to 2,147,483,647	0 to 4,294,967,295	4 bytes
BIGINT	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	0 to 18,446,744,073,709,551,615	8 bytes

INT [(M)] [UNSIGNED] [ZEROFILL]

↓
jumlah digit yang disediakan
untuk menampilkan data

↓
mengorbankan nilai negatif, untuk mendapatkan
jangkauan nilai positif yang lebih tinggi

↓
berhubungan dengan nilai **M**, yang akan mengisi
angka 0 pada bagian kiri nilai yang diinput

```
mysql> CREATE TABLE contoh_int (mini TINYINT, kecil SMALLINT UNSIGNED,  
-> sedang MEDIUMINT(4) ZEROFILL, biasa INT(4) UNSIGNED,  
-> besar BIGINT(6) UNSIGNED ZEROFILL);  
Query OK, 0 rows affected (0.35 sec)
```

```
mysql> DESC contoh_int;
```

Field	Type	Null	Key	Default	Extra
mini	tinyint(4)	YES		NULL	
kecil	smallint(5) unsigned	YES		NULL	
sedang	mediumint(4) unsigned zerofill	YES		NULL	
biasa	int(4) unsigned	YES		NULL	
besar	bigint(6) unsigned zerofill	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql>
```

```
mysql> INSERT INTO contoh_int values ((122), (122), (122),  
-> (122), (122));
```

```
Query OK, 1 row affected (0.19 sec)
```

```
mysql> SELECT * FROM contoh_int;
```

mini	kecil	sedang	biasa	besar
122	122	0122	122	000122

```
1 row in set (0.00 sec)
```

```
mysql> _
```

Decimal

Deklarasi	Jangkauan
DECIMAL (4,1)	-999,9 to 999,9
DECIMAL (6,2)	-9999,99 to 9999,99
DECIMAL (3,2)	-9,99 to 9,99
DECIMAL (8,2)	-999999,99 to 999999,99

DECIMAL

[(M [, D])]

[UNSIGNED]

[ZEROFILL]

total jumlah digit keseluruhan
(maks. 65)

jumlah digit dibekang koma (pecahan)
(maks 30)

membuang nilai negatif dari **DECIMAL**,
tanpa memperbesar jangkauannya

mengisi angka 0 pada bagian kiri angka

Syarat $M > D$
Default $M=10; D=0$

```
mysql> CREATE TABLE contoh_dec (satuan DECIMAL(3,2), puluhan DECIMAL(4,2),
-> ribuan DECIMAL(5,2), normal DECIMAL, cantik DECIMAL(8,2) ZEROFILL);
Query OK, 0 rows affected (0.38 sec)
```

```
mysql> DESCRIBE contoh_dec;
```

Field	Type	Null	Key	Default	Extra
satuan	decimal(3,2)	YES		NULL	
puluhan	decimal(4,2)	YES		NULL	
ribuan	decimal(5,2)	YES		NULL	
normal	decimal(10,0)	YES		NULL	
cantik	decimal(8,2) unsigned zerofill	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> _
```

```
mysql> INSERT INTO contoh_dec values ((1.3), (55.32),  
-> (523.77), (7832.3), (150000.45));
```

```
Query OK, 1 row affected, 1 warning (0.11 sec)
```

```
mysql> SELECT * FROM contoh_dec;
```

satuan	puluhan	ribuan	normal	cantik
1.30	55.32	523.77	7832	150000.45

```
1 row in set (0.00 sec)
```

```
mysql>
```

Float dan Double

Tipe Data	Jangkauan	Ukuran
FLOAT	-3.402823466E+38 to 3.402823466E+38	4 bytes
DOUBLE	-1.7976931348623157E+308 to 1.7976931348623157E+308	8 bytes

FLOAT [(M, D)] [UNSIGNED] [ZEROFILL]

DOUBLE [(M, D)] [UNSIGNED] [ZEROFILL]



Maksimal nilai akan berbeda pada tiap komputer, tergantung kemampuan prosesor mengolah tipe data floating point

- ✓ **FLOAT** biasanya akan akurat untuk penyimpanan sekitar 7 angka decimal
- ✓ **DOUBLE** biasanya mencapai 15 angka desimal

```
mysql> CREATE TABLE contoh_float (satuan FLOAT(3,2), puluhan FLOAT(4,2),
-> ribuan FLOAT(5,2), positif DOUBLE ZEROFILL UNSIGNED,
-> cantik DOUBLE(8,2) ZEROFILL);
Query OK, 0 rows affected (0.30 sec)
```

```
mysql> DESCRIBE contoh_float;
```

Field	Type	Null	Key	Default	Extra
satuan	float(3,2)	YES		NULL	
puluhan	float(4,2)	YES		NULL	
ribuan	float(5,2)	YES		NULL	
positif	double unsigned zerofill	YES		NULL	
cantik	double(8,2) unsigned zerofill	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> _
```

```
mysql> INSERT INTO contoh_float values ((1.3), (55.32), (523.77),  
-> (7832.3), (150000.45));  
Query OK, 1 row affected (0.37 sec)
```

```
mysql> SELECT * FROM contoh_float;
```

satuan	puluhan	ribuan	positif	cantik
1.30	55.32	523.77	000000000000000000007832.3	150000.45

```
1 row in set (0.00 sec)
```

```
mysql>
```

TIPE DATA STRING

Char



Char

Varchar

Binary



Binary

Varbinary

Text



TINYTEXT

TEXT

MEDIUMTEXT

LONGTEXT

Blob



TINYBLOB

BLOB

MEDIUMBLOB

LONGBLOB

Char

CHAR [(M)]
VARCHAR [(M)]



Char → jumlah karakter tetap
Ex : nomor KTP, ID pelanggan dan NIM

Varchar → jumlah karakter yang berbeda-beda
Ex : nama, alamat dan kota

jumlah karakter yang akan dialokasikan

Nilai maksimal M

Char = 255 karakter

Varchar = 65,535 karakter

Default = 1

Data	CHAR(5)	Ukuran Penyimpanan	VARCHAR(5)	Ukuran Penyimpanan
' '	' '	5 byte	' '	1 byte
'du'	'du '	5 byte	'du'	3 byte
'dunia'	'dunia'	5 byte	'dunia'	6 byte
'duniaikom'	'dunia'	5 byte	'dunia'	6 byte

```
mysql> CREATE TABLE contoh_cha (cha CHAR(5),  
-> varcha VARCHAR(5));
```

```
Query OK, 0 rows affected (0.38 sec)
```

```
mysql> INSERT INTO contoh_cha values ('a ', 'a ');
```

```
Query OK, 1 row affected (0.13 sec)
```

```
mysql> INSERT INTO contoh_cha values ('dunia', 'dunia');
```

```
Query OK, 1 row affected (0.06 sec)
```

```
mysql> INSERT INTO contoh_cha values ('duniakampus', 'duniakampus');
```

```
ERROR 1406 (22001): Data too long for column 'cha' at row 1
```

```
mysql> _
```

Binary

BINARY [(M)]
VARBINARY [(M)]



Binary → jumlah karakter tetap
Ex : nomor KTP, ID pelanggan dan NIM

Varbinary → jumlah karakter yang berbeda-beda
Ex : nama, alamat dan kota

jumlah karakter yang akan dialokasikan

Nilai maksimal M

Binary = 255 karakter

Varbinary = 65,535 karakter

Default = 1

```
mysql> CREATE TABLE contoh_bin (bin BINARY(5),  
-> varbin VARBINARY(5));
```

```
Query OK, 0 rows affected (0.38 sec)
```

```
mysql> INSERT INTO contoh_bin values ('dunia','dunia');
```

```
Query OK, 1 row affected (0.17 sec)
```

```
mysql> SELECT * FROM contoh_bin;
```

```
+-----+-----+  
| bin   | varbin |  
+-----+-----+  
| dunia | dunia  |  
+-----+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> SELECT * FROM contoh_bin where bin='dunia';
```

```
+-----+-----+  
| bin   | varbin |  
+-----+-----+  
| dunia | dunia  |  
+-----+-----+
```

```
1 row in set (0.08 sec)
```

```
mysql> SELECT * FROM contoh_bin where bin='DUNIA';
```

```
Empty set (0.00 sec)
```

```
mysql> _
```

Char **VS** Binary

Tipe data binary akan **disimpan secara biner** (bit per bit), bukan secara karakter seperti char dan hal ini akan berefek pada **case-sensitif** data (perbedaan penggunaan huruf besar dan huruf kecil)

- **CHAR** → 'A' disimpan sebagai **karakter 'A'**, dimana 'A' sama dengan 'a'.
- **BINARY** → 'A' disimpan sebagai **bit** dari 'A', yaitu 65, tidak sama dengan 'a', dengan nilai biner 97

Text

Tipe Data	Ukuran Maksimum	Jumlah Karakter Maksimum
TINYTEXT	255 byte	255
TEXT	65.535 byte (64 KB)	6.5535
MEDIUMTEXT	16.777.215 byte (16MB)	16.777.215
LONGTEXT	4.294.967.295 (4GB)	4.294.967.295

Setiap tipe data text disimpan berdasarkan berapa data yang diinput, sama seperti **VARCHAR**, dimana jika kita mendefinisikan suatu kolom sebagai **LONGTEXT**, dan hanya berisi 100 karakter, ukuran penyimpanan yang digunakan sekitar 100 byte, bukan 4GB

```
mysql> CREATE TABLE contoh_text (tin TINYTEXT, tex TEXT,  
-> lon LONGTEXT);
```

```
Query OK, 0 rows affected (0.30 sec)
```

```
mysql> DESC contoh_text;
```

Field	Type	Null	Key	Default	Extra
tin	tinytext	YES		NULL	
tex	text	YES		NULL	
lon	longtext	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql>
```


Blob

Tipe Data	Ukuran Maksimum	Jumlah Karakter Maksimum
TINYBLOB	255 byte	255
BLOB	65.535 byte (64 KB)	65.535
MEDIUMBLOB	16.777.215 byte (16MB)	16.777.215
LOB	4.294.967.295 (4GB)	4.294.967.295

```
mysql> CREATE TABLE contoh_blob (tin TINYBLOB, blo BLOB,  
-> lon LONGBLOB);
```

```
Query OK, 0 rows affected (0.34 sec)
```

```
mysql> DESC contoh_blob;
```

Field	Type	Null	Key	Default	Extra
tin	tinyblob	YES		NULL	
blo	blob	YES		NULL	
lon	longblob	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> _
```

Text **VS** Blob

- Tipe data blob akan **disimpan secara biner** (bit per bit), bukan secara karakter seperti text dan hal ini akan berefek pada **case-sensitif** data (perbedaan penggunaan huruf besar dan huruf kecil)
- Karena sifatnya yang tersimpan secara binary, tipe data BLOB dapat digunakan untuk menyimpan data **multimedia**, seperti gambar dan musik kedalam tabel MySQL