**BAB II**

**LANDASAN TEORI**

* 1. **Teori Antrian**

Antrian yang panjang sering kali kita temukan di bank saat nasabah mengantri di teller untuk melakukan transaksi, di klinik saat pasien mengantri untuk mendapatkan pelayanan, di airport saat para calon penumpang melakukan check-in, di super market saat para pembeli antri untuk melakukan pembayaran, di tempat cuci mobil saat mobil antri untuk dicuci dan masih banyak contoh lainnya. Hal ini dapat menyebabkan konsumen berhenti untuk mengantri atau bahkan dapat meninggalkan sistem sehingga dapat mengakibatkan kehilangan konsumen atau kerugian bagi perusahaan.

Teori tentang antrian diketemukan dan dikembangkan oleh A. K. Erlang, seorang insinyur dari Denmark yang bekerja pada perusahaan telepon di Kopenhagen pada tahun 1910. Erlang melakukan eksperimen tentang fluktuasi permintaan fasilitas telepon yang berhubungan dengan automatic dialing equipment, yaitu peralatan penyambungan telepon secara otomatis. Dalam waktu – waktu yang sibuk operator sangat kewalahan untuk melayani para penelepon secepatnya, sehingga para penelepon harus antri menunggu giliran, mungkin cukup lama. Persoalan aslinya Erlang hanya memperlakukan perhitungan keterlambatan (delay) dari seorang operator, kemudian pada tahun 1917 penelitian dilanjutkan untuk menghitung kesibukan beberapa operator. Dalam periode ini Erlang menerbitkan bukunya yang terkenal berjudul Solution of some problems in the theory of probabilities of significance in Automatic Telephone Exhange. Baru setelah perang dunia kedua, hasil penelitian Erlang diperluas penggunaannya antara lain dalam teori antrian (Supranto, 1987). Menurut Siagian (1987), antrian ialah suatu garis tunggu dari nasabah (satuan) yang memerlukan layanan dari satu atau lebih pelayan (fasilitas layanan). Richard Universitas Sumatera Utara Bronson (1982), proses antrian (queueing process) adalah suatu proses yang berhubungan dengan kedatangan seseorang pelanggan pada suatu fasilitas pelayanan, kemudian menunggu dalam suatu baris (antrian) jika semua pelayannya sibuk, dan akhirnya meninggalkan fasilitas tersebut. Sebuah sistem antrian adalah suatu himpunan pelanggan, pelayan dan suatu aturan yang mengatur kedatangan pada pelanggan dan pemroses masalahnya.

* 1. **Sistem Antrian**

Sistem antrian terdiri satu atau lebih pelayanan yang mana penyedian pelayanan tersebut digunakan untuk melayani bermacam-macam jenis antar kedatangan pelanggan. pelanggan yang datang jika mendapati keadaan pelayanan sedang sibuk maka pelanggan tersebut akan bergabung dalam antrian dalam satu baris dan yang terdekat siap untuk masuk pada pelayanan berikutnya. perlakuan yang seperti inilah yang disebut sistem antrian. sedangkan jika ketika masuk pelanggan mendapat kondisi pelayanan sedang kosong maka pelanggan tersebut dapat langsung untuk dilayani dan tidak perlu menunggu untuk antri.

Menurut Lelono Djati proporsi yang ada dalam discrete-event simulation sangat besar untuk suatu pengamatan karena meliputi model sistem antrian yang kita temui secara nyata atau justru lebih sedikit komponen dari sistem yang disimulasikan jika dengan menggunakan sistem antrian yang sebenarnya.

Klasifikasi menurut Hillier dan Lieberman adalah sebagai berikut :

* + - 1. Sistem pelayanan komersial merupakan aplikasi yang sangat luas dari modelmodel antrian, seperti restoran, kafetaria, toko-toko, salon, butik, supermarket, dan sebagainya.
      2. Sistem pelayanan bisnis-industri mencakup sistem produksi, sistem material, handling, sistem pergudangan, dan sistem-sistem informasi komputer.
      3. Sistem pelayanan transportasi.
      4. Sistem pelayanan sosial merupakan sistem-sistem pelayanan yang dikelola oleh kantor-kantor dan perusahaan-perusahan lokal maupun nasional, seperti kantor registrasi SIM dan STNK, kantor pos, rumah sakit, puskesmas, dan lain-lain (Subagyo, 2000).

Dalam sistem antrian terdapat beberapa komponen dasar proses antrian antara lain adalah:

1. Kedatangan.

Setiap masalah antrian melibatkan kedatangan, misalnya orang, mobil, panggilan telepon untuk dilayani, dan lain-lain. Unsur ini sering dinamakan proses input. Proses input meliputi sumber kedatangan atau biasa dinamakan calling population, dan cara terjadinya kedatangan yang umumnya merupakan variabel acak. Karakteristik dari populasi yang akan dilayani dapat dilihat menurut ukurannya, pola kedatangan, serta perilaku dari populasi yang akan dilayani. Menurut ukurannya, populasi yang dilayani bisa terbatas (finite) dan tidak terbatas (infinite). pola kedatangan bisa teratur, dapat pula bersifat acak atau random. Menurut Levin, dkk (2002), variabel acak adalah suatu variabel yang nilainya bisa berapa saja sebagai hasil dari percobaan acak. Variabel acak dapat berupa diskrit atau kontinu. Bila variabel acak hanya dimungkinkan memiliki beberapa nilai saja, maka ia merupakan variabel acak diskrit. Sebaliknya bila nilainya dimungkinkan bervariasi pada rentang tertentu, ia dikenal sebagai variabel acak kontinu.

1. Pelayanan.

Pelayanan atau mekanisme pelayanan dapat terdiri dari satu atau lebih pelayan, atau satu atau lebih fasilitas pelayanan. Tiap-tiap fasilitas pelayanan kadangkadang disebut sebagai saluran (channel) (Schroeder,1997). Contohnya, jalan tol dapat memiliki beberapa pintu tol. Mekanisme pelayanan dapat hanya terdiri dari satu pelayan dalam satu fasilitas pelayanan yang ditemui pada loket seperti pada penjualan tiket di gedung bioskop.

Dalam mekanisme pelayanan ini ada 3 aspek yang harus diperhatikan yaitu :

* Tersedianya pelayanan

Mekanisme pelayanan tidak selalu tersedia untuk setiap saat. Misalnya dalam pertunjukan bioskop, loket penjualan karcis hanya dibuka pada waktu tertentu antara satu pertunjukan dengan pertunjukan berikutnya, sehingga saat loket ditutup mekanisme pelayanan terrhenti dan petugas beristirahat.

* Kapasitas pelayanan

Kapasitas dari mekanisme pelayanan diukur berdasarkan jumlah pelanggan yang tidak dapat dilayani secara bersama-sama. Kapasitas pelayan yang tidak selalu sama untuk setiap saat, ada yang tetap, tapi ada juga yang berubah-ubah. Karena itu, fasilitas pelayanan dapat memiliki satu atau lebih saluran. Fasilitas yang mempunyai satu saluran disebut saluran tunggal atau sistem pelayanan tunggal dan fasilitas yang mempunyai lebih dari satu saluran disebut saluran ganda atau pelayanan ganda.

* Lama pelayanan

Lama pelayanan adalah waktu yang dibutuhkan untuk melayani seseorang langganan atau satu satuan. Ini harus dinyatakan secara pasti. Oleh karena itu, waktu pelayanan boleh tetap dari waktu ke waktu untuk semua langgannan atau boleh juga berupa variabel acak. Umumnya dan untuk keperluan analisis, waktu pelayanan dianggap sebagai varriabel acak yang terpancar secara bebas dan sama tidak tergantung pada waktu pertibaan.

1. Antrian

Timbulnya antrian terutama tergantung dari sifat kedatangan dan proses pelayanan. Jika tak ada antrian berarti terdapat pelayan yang menganggur atau kelebihan fasilitas pelayanan (Mulyono, 1991).

* 1. **Disiplin Antrian**

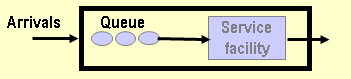
Menurut Thomas J. Kakiay disiplin antrian adalah aturan di mana para pelanggan dilayani, atau disiplin pelayanan (service discipline) yang memuat urutan (order) para pelanggan menerima layanan.

Ada 4 bentuk bentuk disiplin antrian menurut urutan kedatangan antara lain adalah :

* + - 1. First Come First Served (FCFS) atau First In First Out (FIFO), di mana pelanggan yang terlebih dahulu datang akan dilayani terlebih dahulu. Misalnya, antrian pada loket pembelian tiket bioskop, antrian pada loket pembelian tiket kereta api.
      2. Last Come First Served (LCFS) atau Last In First Out (LIFO), di mana pelanggan yang datang paling akhir akan dilayani terlebih dahulu. Misalnya, sistem antrian pada elevator untuk lanti yang sama, sistem bongkar muat barang dalam truk, pasien dalam kondisi kritis, walaupun dia datang paling akhir tetapi dia akan dilayani terlebih dahulu.
      3. Service In Random Order (SIRO) atau Random Selection for Service (RSS), di mana panggilan didasarkan pada peluang secara random, jadi tidak menjadi permasalahan siapa yang lebih dahulu datang. Misalnya, pada arisan di mana penarikan berdasarkan nomor undian.
      4. Priority Service (PS), di mana prioritas pelayanan diberikan kepada pelanggan yang mempunyai prioritas lebih tinggi dibandingkan dengan pelanggan yang mempunyai prioritas yang lebih rendah, meskipun mungkin yang dahulu tiba di garis tunggu adalah yang terakhir datang. Hal ini mungkin disebabkan oleh beberapa hal, misalnya seseorang yang memiliki penyakit yang lebih berat dibandingkan orang lain pada suatu tempat praktek dokter, hubungan kekerabatan pelayan dan pelanggan potensial akan dilayani terlebih dahulu.
  1. **Struktur Antrian**

Ada 4 model struktur antrian dasar yang umum terjadi dalam seluruh sistem antrian :

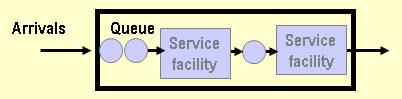
* + - 1. Single Channel – Single Phase



Gambar 2.1 Single Channel – Single Phase.

Single Channel berarti hanya ada satu jalur yang memasuki sistem pelayanan atau ada satu fasilitas pelayanan. Single Phase berarti hanya ada satu fasilitas pelayanan. Contohnya adalah sebuah rumah sakit yang hanya mempunyai satu loket pelayananan pendaftaran dengan jalur satu antrian, supermarket yang hanya memiliki satu kasir sebagai tempat pembayaran, dan lain-lain.

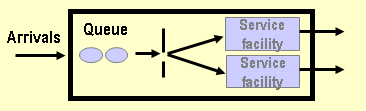
* + - 1. Single Channel – Multi Phase



Gambar 2.2 Single Channel – Multi Phase.

Sistem antrian jalur tunggal dengan tahapan berganda ini atau menunjukkan ada dua atau lebih pelayanan yang dilaksanakan secara berurutan. Sebagai contoh adalah : pencucian mobil, tukang cat mobil, dan sebagainya.

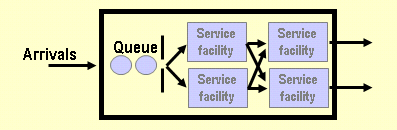
* + - 1. Multi Channel – Single Phase



Gambar 2.3 Multi Channel – Single Phase.

Sistem Multi Channel – Single Phase terjadi di mana ada dua atau lebih fasilitas pelayanan dialiri oleh antrian tunggal. Contohnya adalah antrian pada sebuah bank dengan beberapa teller, pembelian tiket atau karcis yang dilayani oleh beberapa loket, pembayaran dengan beberapa kasir, dan lain-lain.

* + - 1. Multi Channel – Multi Phase



Gambar 2.4 Multi Channel – Multi Phase.

Sistem Multi Channel – Multi Phase ini menunjukkan bahwa setiap sistem mempunyai beberapa fasilitas pelayanan pada setiap tahap sehingga terdapat lebih dari satu pelanggan yang dapat dilayani pada waktu bersamaan. Contoh pada model ini adalah : pada pelayanan yang dibarikan kepada pasien di rumah sakit dimulai dari pendaftarran, diagnose, tindakan medis, samppai pembayaran, registrasi ulang mahasiswa baru pada sebuah universitas, dan lain-lain.

* 1. **Struktur Navigasi**

Struktur navigasi adalah alur dari buah program dan termasuk unsure terpenting dalam pembuatan aplikasi halaman *web*. Struktur navigasi ini merupakan perancangan hubungan dan rantai kerja dari beberapa lokasi yang berbeda dan dapat membantu mengorganisasikan seluruh elemen halaman web dan juga memudahkan untuk menganalisa keinteraktivan seluruh objek didalam halaman *web*.

Struktur navigasi digunakan untuk menggambarkan secara garis besar isi dari seluruh *website* dan menggambarkan bagaimana hubungan antara isi dari *website* tersebut. Dengan struktur navigasi, dapat terlihat semua susunan isi dari sebuat *website* secara menyeluruh.

Struktur navigasi memiliki beberapa jenis. Yaitu :

* + - 1. Navigasi Linier

Struktur ini menampilkan satu demi satu tampilan layar secara beruntun dan tidak adanya percabangan. Biasanya digunakan untuk presentasi karena kemudahan dalam hal penampilan data serta tidak membutuhkan banyak keinteraktifan di dalamnya. Tampilan yang ditampilkan adalah satu halaman sebelumnya atau satu halaman sesudahnya.

Gambar 2.5 Struktur Navigasi Bentuk Linier.

* + - 1. Navigasi Hirarki

Struktur ini biasa disebut juga struktur navigasi bercabang dan merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data atau gambar pada layar dengan kriteria tertentu.

Tampilan pada menu pertama disebut *master page*. Halaman tersebut memiliki halaman percabangan yang disebut dengan *slave page* (pendukung) dan jika dipilih menjadi halaman utama kedua.



Gambar 2.6 Struktur Navigasi Bentuk Hirarki.

* + - 1. Navigasi Non Linier

Struktur navigasi non linier (tidak berurutan) adalah pengembangan dari struktur navigasi linier, tetapi navigasi ini memperkenankan adanya percabangan. Pada struktur ini kedudukan semua *page* sama, sehingga tidak dikenal dengan adanya master atau *slave page.*



Gambar 2.7 Struktur Navigasi Bentuk Non Linier.

* + - 1. Navigasi Campuran

Merupakan gabungan dari struktur sebelumnya serta disebut juga dengan struktur navigasi bebas. Maksudnya adalah jika suatu tampilan membutuhkan percabangan maka dibuat percabangan. Struktur ini paling banyak digunakan dalam pembuatan aplikasi multimedia.

Pemakai dapat dengan bebas menelusuri program. Akan tetapi pada bagian tertentu gerakan dibatasi secara hirarki ataupun linier.



Gambar 2.8 Struktur Navigasi Bentuk Campuran

* 1. **Web Server**

Web server adalah sebuah software yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada klien yang dikenal dan biasanya kita kenal dengan nama web browser dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan pada umumnya akan berbentuk dokumen HTML. itulah pengertian web server sebenarnya. dalam bentuk sederhana web server akan mengirim data HTML kepada permintaan web Browser sehingga akan terlihat seperti pada umumnya yaitu sebuah tampilan website.

* 1. **Aplikasi Web**

Dalam rekayasa perangkat lunak, suatu aplikasi web (bahasa Inggris: web application atau sering disingkat webapp) adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti Internet atau intranet. Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web (seperti HTML, JavaScript, AJAX, Java, dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Aplikasi web menjadi populer karena kemudahan tersedianya aplikasi klien untuk mengaksesnya, penjelajah web, yang kadang disebut sebagai suatu thin client (klien tipis). Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya. Aplikasi web yang umum misalnya webmail, toko ritel daring, lelang daring, wiki, papan diskusi, weblog, serta MMORPG.

* 1. **AngularJS**

AngularJS merupakan framework javascript open source yang dirilis oleh google pada tahun 2009. Konsep dari AngularJS adalah meningkatkan fungsi dari HTML untuk membangun wepapp. Bayangkan awalnya HTML hanya digunakan untuk membuat halaman website statis dan kini bisa berfungsi untuk membuat wepapp dengan menggunakan AngularJS.

Tabel 2.1 Konsep AngularJs

|  |  |
| --- | --- |
| Konsep | Deskripsi |
| Template | HTML dengan markup tambahan |
| Directives | Menggabungkan HTML dengan atribut khusus dan elemen - elemen |
| Model | Data ditampilkan ke user dalam tampilan dan memungkinkan user berinteraksi |
| Scope | context dimana model disimpan agar controllers, directives dan expressions dapat mengaksesnya |
| Expressions | Variable dan function pada scope |
| Compiler | Parsing/mengubah template, instan, directives dan expressions |
| Filter | Format data yang akan di tampilkan ke user |
| View | Apa yang akan user lihat |
| Data Binding | Sync data model dengan view |
| Controller | Bisnis dan logic dibalik view |
| Dependency Injection | Membuat dan menghubungkan object dan function |
| Injector | Container dari Depedency Injection |
| Module | Sebuah container dari berbagai macam bagian aplikasi |
| Service | Bisni dan logic yang dapat digunakan kembali |

* 1. **Java**

Java adalah nama sebuah bahasa pemrograman yang sangat terkenal. Sebagai bahasa pemrograman, Java dapat digunakan untuk menulis program. Sebagaimana diketahui, program adalah kumpulan instruksi yang ditujukan unutk komputer. Melalui program, komputer dapat diatur agar melaksanakan tugas tertentu sesuai yang ditentukan oleh pemrogram (orang yang membuat program).

Versi awal Java pada tahun 1996 sudah merupakan versi release sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

* java.lang: Peruntukan kelas elemen-elemen dasar.
* java.io: Peruntukan kelas *input* dan *output*, termasuk penggunaan berkas.
* java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
* java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
* java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
* java.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

Dalam pemprograman Java perlu diketahui beberapa hal sebelum memulai mengerjakan program, berikut merupakan hal-hal yang perlu diketahui :

* **Case Sensitivity** berarti pengidentifikasi Hello dan hello akan memiliki arti yang berbeda pada Java.
* **Class Names**, Untuk semua nama class, huruf pertama harus kapital. Jika beberapa kata yang digunakan untuk membentuk nama class, huruf pertama setiap kata harus kapital.  
  Contoh class: MyFirstJavaClass
* **Method Names**, Semua nama Method harus diawali dengan huruf kecil. Jika beberapa kata yang digunakan untuk membentuk nama method, maka huruf pertama diawali dengan huruf kecil kemudian kata berikutnya diawali dengan huruf kapital.  
  Contoh public void myMethodName ()
* **Program File Name**, nama file program harus persis sesuai dengan nama class. Ketika menyimpan file, Anda harus menyimpannya dengan nama class (ingat Java adalah case sensitive) dan menambah ‘.java’ ke akhir nama ( jika nama file dan nama class tidak cocok, program Anda tidak dapat di compaile.  
  Contoh: Asumsikan ‘MyFirstJavaProgram’ adalah nama class, maka file tersebut harus disimpan dengan ‘MyFirstJavaProgram.java’.
* **public static void main(String args[])**, program Java memulai proses dari method main(),yang merupakan bagian wajib dari setiap program Java.
  1. **Maven**

Maven adalah sebuah software dan tool yang komprehensif untuk project management. Konsep dasar dari Maven adalah Project Object Model (POM), dimana Maven dapat membangun, membuat laporan, dan dokumentasi dari sebuah project. Berikut isi POM :

* + - 1. Identifier project
* **groupId** : nama organisasi / perusahaan pembuat
* **artifactId** : nama modul / project
* **version** : versi project
  + - 1. Dependensi project, pembagian scope dependency:
* **compile** : digunakan (diimport) dalam main source dan akan di-include dalam produk akhir. Ini adalah pilihan default, kalau scope dikosongkan, artinya compile
* **runtime** : tidak digunakan dalam source code (tidak diimport), tapi disertakan dalam produk akhir. Contoh: library MySQL.
* **test** : digunakan (diimport) dalam test source, tapi tidak digunakan di main source. Tidak diinclude dalam produk akhir. Contoh: library JUnit.
* **provided** : digunakan (diimport) dalam main source, tapi tidak disertakan dalam produk akhir. Biasanya karena sudah disediakan oleh container tempat aplikasi dijalankan. Contoh: library javax.servlet.
  1. **Mysql Server**

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Database MySQL menyediakan banyak tipe data untuk digunakan pada saat membuat tabel pada database. Tipe data tersebut berkaitan dengan data yang akan kita simpan didalam tabel pada database. Secara garis besar, tipe data yang sering dipakai adalah tipe angka (INTEGER), texts (VARCHAR, TEXT) dan waktu (DATE, DATETIME, TIMESTAMP).

Menentukan tipe data yang tepat nantinya akan memudahkan dalam pengaturan tabel, seperti penggunaan tipe data DATE akan otomatis membuat format standar DATE misalnya '2014-11-13'. Tipe data sebaiknya ditentukan pada saat membuat tabel, karena jika diubah saat sudah ada isinya dikhawatirkan akan mengacaukan isi tabel yang berakibat pada kehilangan data.

Berikut ini beberapa tipe data yang didukung oleh database MySQL :

* + - 1. **Tipe Data Angka**

|  |  |
| --- | --- |
| **Tipe Data** | **Keterangan** |
| TINYINT(size) | Normal = -128 s/d 127, Unsigned = 0 s/d 255. Maksimal digit angka ditentukan pada parameter **size** didalam kurung. |
| SMALLINT(size) | Normal = -32768 s/d 32767, Unsigned = 0 s/d 65535. Maksimal digit angka ditentukan pada parameter **size** didalam kurung. |
| MEDIUMINT(size) | Normal = -8388608 s/d 8388607, Unsigned =0 s/d 16777215. Maksimal digit angka ditentukan pada parameter **size** didalam kurung. |
| INT(size) | Normal  = -2147483648 s/d 2147483647, Unsigned = 0 s/d 4294967295. Maksimal digit angka ditentukan pada parameter **size** didalam kurung. |
| BIGINT(size) | Normal = -9223372036854775808 s/d 9223372036854775807, Unsigned = 0 s/d 18446744073709551615. Maksimal digit angka ditentukan pada parameter **size** didalam kurung. |
| FLOAT(size,d) | Angka kecil dengan desimal. Maksimal digit angka ditentukan pada paramater **size**dan maksimal digit desimal ditentukan pada parameter **d**. |
| DOUBLE(size,d) | Angka besar dengan desimal. Maksimal digit angka ditentukan pada paramater **size** dan maksimal digit desimal ditentukan pada parameter **d**. |
| DECIMAL(size,d) | Tipe DOUBLE yang disimpan sebagai string, memungkinkan untuk poin desimal tetap. Maksimal digit angka ditentukan pada paramater **size** dan maksimal digit desimal ditentukan pada parameter **d**. |

Tabel 2.2 Tipe Data MySql

* + - 1. **Tipe Data String**

|  |  |
| --- | --- |
| **Tipe Data** | **Keterangan** |
| CHAR(size) | - Menangani tipe data string tetap/fixed (bisa berisi huruf, angka dan special karakter). - Panjang string ditentukan dengan nilai parameter size didalam kurung. - Kelebihan karakter akan dipotong sesuai panjang yang telah ditentukan. - Dapat menyimpan sampai maksimal 255 karakter. |
| VARCHAR(size) | - Menangani tipe data string variabel (bisa berisi huruf, angka dan special karakter). - Panjang string ditentukan dengan nilai parameter size didalam kurung. - Kelebihan karakter akan dipotong sesuai panjang yang telah ditentukan. - Dapat menyimpan sampai maksimal 255 karakter. - Jika dimasukkan nilai yang lebih dari 255 akan otomatis di konvert ke tipe TEXT |
| TINYTEXT | Menangani tipe data string dengan panjang maksimal 255 karakter. |
| TEXT | Menangani tipe data string dengan panjang maksimal 65.535 karakter. |
| BLOB | Untuk BLOBs (Binary Large OBjects), mampu menangani sampai 65,535 bytes data |
| MEDIUMTEXT | Menangani tipe data string dengan panjang maksimal 16,777,215 karakter |
| MEDIUMBLOB | Untuk BLOBs (Binary Large OBjects). mampu menangani sampai 16,777,215 bytes data |
| LONGTEXT | Menangani tipe data string dengan panjang maksimal 4,294,967,295 karakter |
| LONGBLOB | Untuk BLOBs (Binary Large OBjects). mampu menangani sampai 4,294,967,295 bytes data |
| ENUM(x,y,z,etc.) | Mengijinkan kita memasukkan beberapa nilai yang mungkin didalam sebuah daftar. Kita bisa membuat hingga 65535 nilai dalam daftar ENUM. Jika nilai yang dimasukkan yang tidak ada dalam daftar, maka nilai kosong akan disisipkan. Catatan: Nilai-nilai diurutkan berdasarkan urutan saat kita memasukkan mereka. Kita bisa memasukkan kemungkinan nilai dalam format ini: ENUM ('X', 'Y', 'Z') |
| SET | Mirip dengan ENUM, namun SET bisa berisi hingga 64 daftar item dan dapat menyimpan lebih dari satu pilihan. |

Tabel 2.3 Tipe Data MySql

* + - 1. **Tipe Data Date**

|  |  |
| --- | --- |
| **Tipe Data** | **Keterangan** |
| DATE() | - Sebuah tanggal. - Format: YYYY-MM-DD **Note:** Range yang disupport adalah dari '1000-01-01' sampai '9999-12-31' |
| DATETIME() | - Sebuah kombinasi tanggal dan waktu. - \*Format: YYYY-MM-DD HH:MI:SS **Note:** Range yang disupport adalah dari '1000-01-01 00:00:00' sampai '9999-12-31 23:59:59' |
| TIMESTAMP() | - Sebuah Timestamp - Nilai TIMESTAMP disimpan sebagai detik angka sejak masa Unix ('1970-01-01 00:00:00' UTC). - \*Format: YYYY-MM-DD HH:MI:SS **Note:** Range yang disupport adalah dari '1970-01-01 00:00:01' UTC sampai '2038-01-09 03:14:07' UTC |
| TIME() | - Sebuah waktu - Format: HH:MI:SS **Note:** Range yang disupport adalah dari '-838:59:59' sampai '838:59:59' |
| YEAR() | - Sebuah tahun dengan format dua digit atau empat digit. **Note:** Nilai yang diijinkan untuk format 4 digit adalah 1901 to 2155. Nilai yang diijinkan pada format 2 digit adalah 70 to 69, merupakan representasi dari tahun 1970 sampai 2069 |

Tabel 2.4 Tipe Data MySql