|  |  |  |
| --- | --- | --- |
| C:\Users\Admin\Pictures\JATA KPM.png123456  **BAHAGIAN PENDIDIKAN TEKNIK DAN VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN #PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)*** | | |
| **KOD DAN NAMA PROGRAM** | IT-010-3:2016 APPLICATION DEVELOPMENT | |
| **TAHAP DAN SEMESTER** | 3 (SEMESTER 1) | |
| **KOD DAN TAJUK KURSUS** | KPD1043 PRINCIPLES AND PRACTICE IN DATABASE | |
| **NO.DAN TAJUK KOMPETENSI** | K1 INTRODUCTION TO LATEST DATABASE PROGRAMMING  K2 STRUCTURE DATA MODELLING AND DATABASE DESIGN  **K3 CREATE AND MANAGE TABLE USING DATA DEFINITION LANGUAGE**  K4 COMMIT PROTOTYPE SOURCE CODE | |
| **NO. KOD KSKV** | KPD 1043 / KP(3/4) | Muka Surat : 1 Drp : 13 |
| **NO. KOD NOSS** | IT-010-3:2016-C01/ P(3/4) |

**TAJUK :**

**3.0 CREATE AND MANAGE TABLE USING DATA DEFINITION LANGUAGE**

**TUJUAN:**

Diakhir sesi pembelajaran, pelajar akan dapat:

1. Keperluan perkakasan dan perisian pemasangan SQL
2. Memasang *Development Stack* untuk *Structured Query Language* (SQL) menggunakan WAMP
3. Pengenalan kepada *Database administration*
   1. *Database access*
   2. *Data manipulation*
   3. *Database maintenance*
4. Mengwujudkan Kamus Data
5. Membina jadual menggunakan SQL
6. Menggunakan Data Manipulating Language dalam SQL

**PENERANGAN/**INFORMATION**:**

1. **KEPERLUAN PERKAKASAN DAN PERISIAN PEMASANGAN SQL**

Dalam topik ini, kita akan mempelajaran bagaimana cara untuk memasang dan kongfigurasi pangkalan data menggunakan platform Windows.

Sebelum pemasangan Perisian Pangkalan Data, berikut adalah antara perkara yang perlu diambil kira

1. Keperluan perisian sedia ada (pre-installation softaware)
2. Keperluan perkakasan sedia ada (hardware requirement)

Bagaimana untuk mendapatkan maklumat mengenai perisian dan perkakasan sedia ada? Anda perlu membaca manual perisian pangkalan data yang bakal anda pasang.

Langkah-langkah sebelum memasang perisian:

1. Membaca nota keluaran (release notes) dari pembekal.

Contoh release notes dari MySQL:

<http://dev.mysql.com/doc/mysql-monitor/3.0/en/index.html>

1. Mengkaji maklumat pelesenan : Walaupun media pemasangan dalam pek media yang mengandungi banyak komponen perisian pangkalan data, anda dibenarkan untuk menggunakan hanya komponen yang mana anda telah membeli lesen.
2. Merancang pemasangan
3. Melengkapkan keperluan pra-pemasangan : keperluan perisian dan perkakasan sedia ada
4. Melengkapkan pemasangan dengan konfigurasi akhir
5. Memulakan penggunaan perisian pangkalan data

Perlu di ingatkan bahawa setiap **keperluan perisian pangkalan data ada berbeza**. Merancang spesifikasi adalah penting bagi memastikan pemasangan perisian sistem pangkalan data berjalan dengan baik.

Sangat penting untuk pemilihan sistem pangkalan data yang boleh **menampung keperluan pangkalan data yang betu**l.

**KEPERLUAN PEMASANGAN PERISIAN PANGKALAN DATA**

Berikut adalah contoh perisian pangkalan data dan ciri-ciri yang boleh didapati:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ciri (feature)** | **Oracle** | **MySQL** | **Microsoft SQL** |
| **Antaramuka**  **(*user interface*)** | GUI, SQL | SQL | GUI, SQL |
| **Bahasa Sokongan** | Banyak termasuk C,C#,C++,Java, Ruby dan Objective C | Banyak termasuk C,C#,C++,D,Java, Ruby dan Objective C | Java,Ruby,Python, VB.Net dan PHP |
| **Sistem Pengoperasian** | Windows, Linux,Solaris,HP-UX, OS X,z/OS | Windows,Linux,OS X, Solaris | Windows |
| **Perlesenan** | Hak milik/proprietary | Sistem Terbuka/Open Source | Hak milik/Proprietary |

**MySQL memerlukan pemasangan (installation) pra-syarat**:

1. Visual Studio Tools for Office 2012 Runtime
2. Microsoft .NET Framework 4 Client Profile
3. Microsoft Excel 2007 or higher
4. Microsoft Visual C++ 2010 32-bit Runtime

Visual Studio Tools Runtime:-

Merupakan satu alat pembanguna yang terdapat dalam bentuk Visual Studio *add-in* (template projek) dan *runtime* bagi membolehkan versi Microsoft Office 2003 dan kemudiannya menggunakan aplikasi Office .NET Framework Common Language Runtime (CLR) berfungsi dengan baik.

.NET Framework:-

Merupakan sebuah komponen kerangka kerja bagi perisian tertentu untuk digunakan atau dijalankan pada sistem operasi Microsoft Windows. Ia berfungsi untuk menterjemah bahasa pemprograman (*programming language*) yang akan digunakan oleh perisian tersebut.

Ia bertindak seperti jambatan perhubungan antara perisian dengan sistem operasi Windows. Bagi Windows 7, enjin .NET Framework secara asasnya sudah di pasangan dengan .NET 3.5 dan je atas. Bagi yang menggunakan Windows XP, anda perlu memasangan secara manual kerana ia tidak wujud dalam pakej pemasangan.

**KEPERLUAN SPESIFIKASI PERKAKASAN**

* Perkakasan IT(hardware) diperlukan untuk menjalankan pangkalan data
* Keperluan minima bergantung kepada jenis penggunaan dan senibina sistem DB tersebut
* Juga bergantung kepada keperluan organisasi dan DBMS yang digunakan
* Umumnya memerlukan ruang memori dan storan yang besar serta unit pemprosesan yang tinggi.

Bagi perisian MySQL, ia tidak memerlukan perkakasan yang rumit dan secara umumnya kita hanya memerlukan perkakasan yang minimum sahaja (komputer yang boleh berfungsi dengan baik).

Berikut adalah spesifikasi perkakasan bagi 3 jenis sistem pangkalan data yang popular:-

1. ORACLE

Berikut adalah keperluan minimum bagi Oracle:-

|  |  |
| --- | --- |
| **KEPERLUAN** | **NILAI** |
| Ingatan fizikal  (*Physical memory*) (RAM) | 1 Gb minimum  On Windows Vista, the minimum requirement is 512 MB |
| Ingatan Maya  (*Virtual memory*) | Double the amount of RAM |
| Kapasiti Disk (*Disk space*) | Total: 4.76 Gb |
| Video adapter | 256 colors |
| Processor | 550 MHz minimum  On Windows Vista, the minimum requirement is 800 MHz |

1. MySQL

Keperluan minimum bagi MySQL:-

|  |  |
| --- | --- |
| **KEPERLUAN** | **NILAI** |
| Ingatan fizikal  (*Physical memory*) (RAM) | 128 MB RAM |
| Ingatan Maya  (*Virtual memory*) | Double the amount of RAM |
| Kapasiti Disk (*Disk space*) | Basic Installation Type total: 2.04 GB  Advanced Installation Types total: 1.94 GB |
| Video adapter | 256 colors |
| Processor | Pentium V processor |

Anda boleh menggunakan MySQL Calculator yang terdapat di atas talian untuk menilai keperluan minimum perkakasan yang diperlukan:-<http://www.mysqlcalculator.com/>

Nota: MySQL boleh dipasang (installed) pada platform sekurang-kurangnya 32 MB. Walaubagaimanapun, untuk mendapat capaian yang baik memerlukan sekurang-kurangnya 128MB memory.

1. Microsoft SQL (MS SQL)

Keperluan minimum bagi Microsoft SQL (MS SQL):-

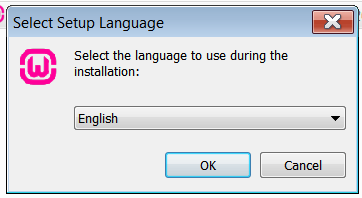
|  |  |
| --- | --- |
| **KEPERLUAN** | **NILAI** |
| Processor | Intel® or compatiblePentium 166 MHz or higher. |
| Ingatan fizikal  (*Physical memory*) (RAM) | Enterprise Edition and Enterprise Evaluation Edition: 64 MB minimum, 128 MB or more recommended Standard Edition and Developer Edition: 64 MB minimum  Personal Edition and Desktop Engine (MSDE 2000):   * 128 MB minimum on Windows XP |
| Kapasiti Disk (*Disk space*) | * SQL Server v – 270MB (penuh), 250MB (tipikal), 95MB(minimum) * Servis OLAP- 50MB * English Query -12MB |
| **Monitor** | VGA or higher resolution800x600 or higher resolution required for the SQL Server graphical tools |

1. **MEMASANG *DEVELOPMENT STACK* UNTUK *STRUCTURED QUERY LANGUAGE* (SQL)**

1. Pilih fail Installer yang mengandungi WAMP server yang sesuai

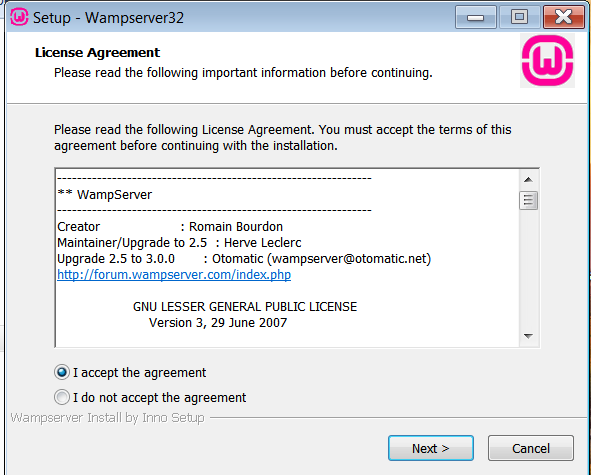
I:\Wampserver Installation steps\1.PNG

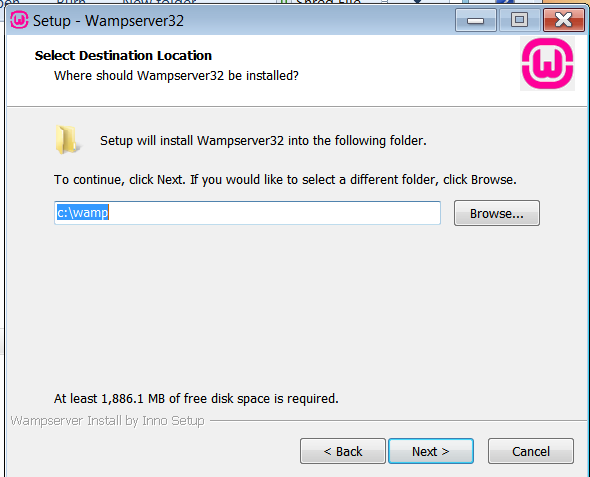
2. Pilih bahasa yang anda kehendaki



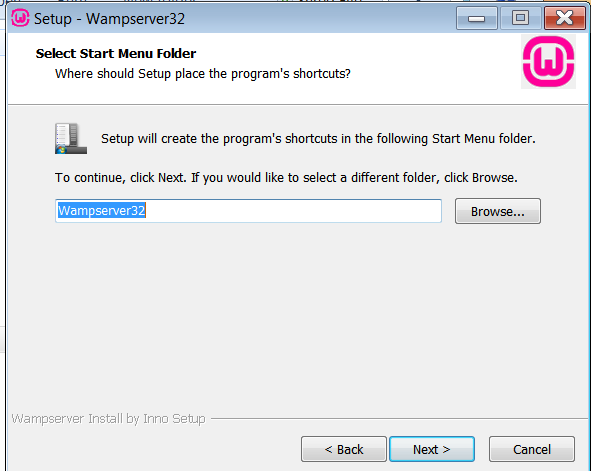
3.klik “I accept the agreement” >”

4. Baca informasi yang tertera dan klik “Next >”

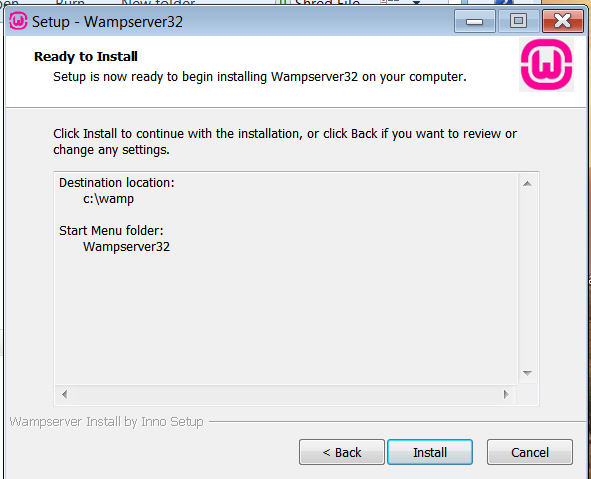


5. Pilih fail untuk destinasi WAMP dan klik “Next >”

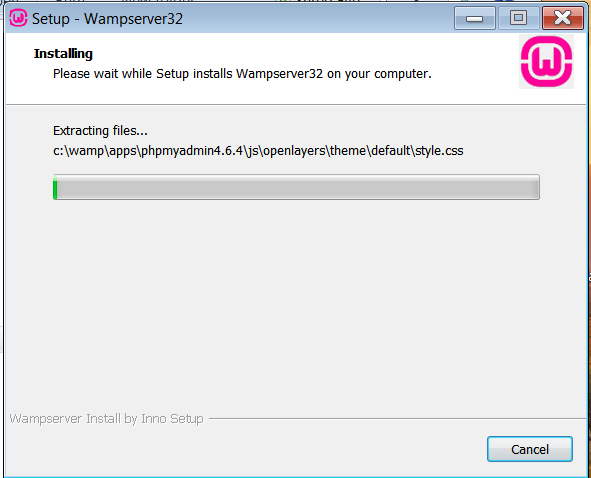
6) Pilih fail untuk destinasi WAMP dan klik “Next >”



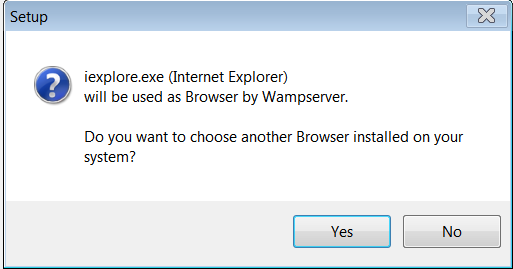
7. Pastikan maklumat fail betul dan klik “Install”



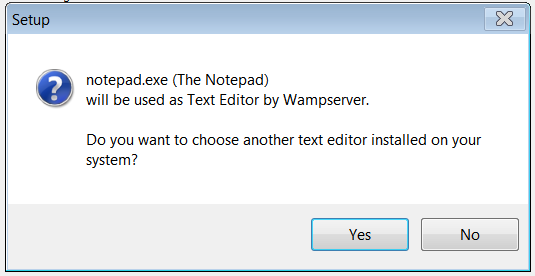
8. WAMPserver32 sedang di pasang pada komputer



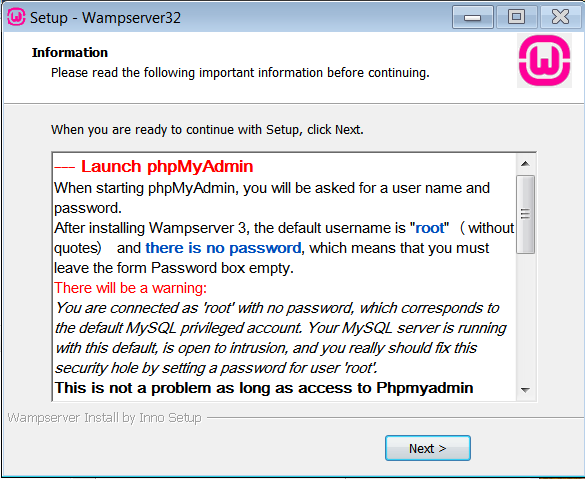
9.Jika anda setuju , klik “Yes “ untuk langkah seterusnya



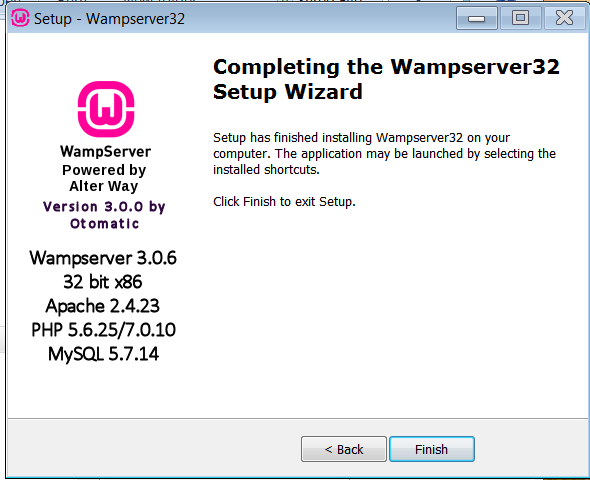
10. Jika anda setuju , klik “Yes “ untuk langkah seterusnya



11. Baca informasi yang tertera dan klik “Next >”



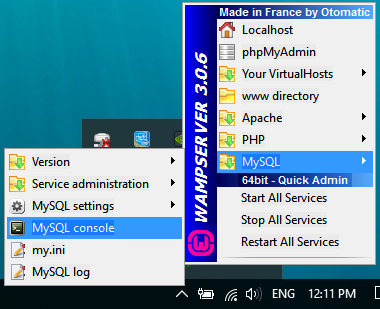
12)setelah muat turun telah selesai , klik “Finish” untuk keluar dari setup



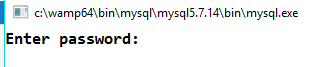
13) Klik untuk mengaktifkan WAMPserver



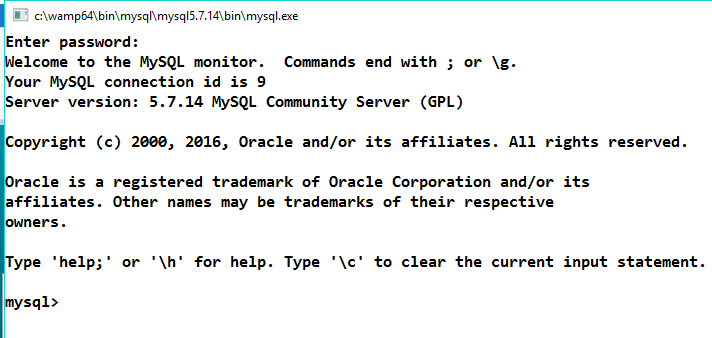
14) Pilih MYSQL console



15) tekan enter



16) MYSQL sedia digunakan



1. **PENGENALAN KEPADA *DATABASE ADMINISTRATION***

Pentadbiran pangkalan data (*database administration*) merujuk kepada keseluruhan rangkaian aktiviti yang dilakukan oleh pentadbir pangkalan data (*database administrator*) untuk memastikan pangkalan data sentiasa tersedia seperti yang diperlukan. Tugas dan peranan yang berkaitan dengan lain adalah keselamatan pangkalan data, pemantauan pangkalan data dan penyelesaian masalah, dan perancangan untuk pertumbuhan masa hadapan.Pentadbiran pangkalan data (*database administration*)mempunyai fungsi penting dalam mana-mana organisasi yang bergantung kepada satu atau lebih pangkalan data.

Pentadbir pangkalan data/ *database administrator* (DBA) mempunyai peranan khusus dalam jabatan IT untuk organisasi besar. Walau bagaimanapun, banyak syarikat yang lebih kecil yang tidak mampu membayar DBA sepenuh masa biasanya mengalih keluar atau mengikat peranan kepada vendor khusus, atau menggabungkan peranan dengan yang lain di jabatan ICT supaya kedua-duanya dilakukan oleh satu orang.

Peranan utama pentadbiran pangkalan data adalah untuk memastikan masa maksimum untuk pangkalan data supaya sentiasa tersedia apabila diperlukan. Ini biasanya melibatkan pemantauan dan pemecahan berkala proaktif. Ini seterusnya melibatkan beberapa kemahiran teknikal di bahagian DBA. Selain pengetahuan mendalam tentang pangkalan data yang dipersoalkan, DBA juga perlu pengetahuan dan mungkin latihan di platform (enjin pangkalan data dan sistem operasi) di mana pangkalan data dijalankan. Sebahagian tugas pentadbiran pangkalan data (*database administrator*) adalah seperti berikut :

* Keselamatan Pangkalan Data: Memastikan bahawa hanya pengguna yang dibenarkan mempunyai akses kepada pangkalan data dan memperkuatnya terhadap sebarang akses luaran dan tanpa izin.
* *Tuning* Pangkalan Data: *Tweaking* mana-mana beberapa parameter untuk mengoptimumkan prestasi, seperti peruntukan memori pelayan (*server*), pemecahan fail dan penggunaan cakera .
* Sandaran dan Pemulihan: Ini adalah peranan DBA untuk memastikan bahawa pangkalan data mempunyai prosedur sandaran dan pemulihan yang mencukupi untuk mendapatkan kembali dari kehilangan data yang tidak sengaja atau sengaja.
* Menghasilkan Laporan dari *Queries*: DBAs sering dipanggil untuk menghasilkan laporan dengan menulis *queries*, yang kemudian dijalankan terhadap pangkalan data.

1. ***Database Access***

Akses data merujuk kepada keupayaan pengguna untuk mengakses atau mengambil data yang disimpan dalam pangkalan data atau repositori lain. Pengguna yang mempunyai akses data boleh menyimpan, mengambil, memindahkan atau memanipulasi data yang disimpan, yang boleh disimpan pada pelbagai cakera keras dan peranti luaran.

1. ***Data Manipulation***

Manipulasi data adalah proses menukar data dalam usaha untuk memudahkan membaca atau menjadi lebih teratur. Sebagai contoh, log data boleh diatur mengikut urutan abjad, menjadikan entri individu lebih mudah dicari. Manipulasi data sering digunakan pada log pelayan web untuk membolehkan pemilik laman web untuk melihat halaman paling popular serta sumber trafiknya.

Pengguna dalam bidang Perakaunan atau bidang lain yang bekerja dengan nombor seringkali memanipulasi data untuk memikirkan kos produk, trend jualan, kewajiban cukai yang berpotensi, atau barang dagangan yang dijual setiap minggu atau bulan. Penganalisis pasaran saham kerap menggunakan manipulasi data untuk meramalkan trend dalam pasaran saham dan bagaimana saham boleh dilakukan dalam masa terdekat.

Komputer juga boleh menggunakan manipulasi data untuk memaparkan maklumat kepada pengguna dengan cara yang lebih bermakna, berdasarkan kod dalam program perisian, laman web, atau pemformatan data yang ditakrifkan oleh pengguna.

1. ***Database Maintenance***

Penyelenggaraan pangkalan data adalah satu aktiviti yang direka untuk memastikan pangkalan data dijalankan dengan lancar. Sejumlah sistem yang berbeza boleh digunakan untuk membina dan menyelenggarakan pangkalan data, dengan satu contoh yang popular adalah MYSQL. Penyelenggaraan pangkalan data biasanya dilakukan oleh orang-orang yang selesa dan biasa dengan sistem pangkalan data dan spesifik pangkalan data tertentu.

1. **KAMUS DATA**

Kamus data merupakan dokumen untuk mengumpul, mendokumen dan menguruskan semua keterangan mengenai elemen yang digunakan dalam memodelkan sistem yang meliputi proses, storan data, entiti luaran dan sebagainya.

Maklumat tersebut perlu disimpan bagi memudahkan aktiviti penyelenggaraan dan penggunaan semula.

Maklumat bagi setiap elemen adalah berbeza-beza mengikut kesesuaian.

Contohnya,bagi storan data, maklumat yang perlu disimpan ialah keterangan mengenai storan data tersebut, struktur data yang disimpan dan jenis data yang disimpan; bagi entiti luaran pula, maklumat yng perlu disimpan hanyalah keterangan mengenai entiti luaran tersebut. Rajah berikut menunjukkan contoh kamus data bagi entiti luaran pelajaran dan storan data rekod pelajar.

|  |  |  |
| --- | --- | --- |
| **Label** | **Nama Pelajar** | |
| **Jenis** | **Elemen Data** | |
| **Keterangan** | **Data bagi rekod peribadi pelajar Kolej Vokasional Shah Alam** | |
| **Struktur data** | **Jenis data** | **Aksara** |
| **Panjang data** | **30** |
| **Sumber** | **Borang permohonan asrama** |
| **Nilai lalai** | **Tiada** |

**Contoh Kamus Data Pelajar bagi atribut Nama Pelajar**

|  |  |
| --- | --- |
| **Label** | **Pelajar** |
| **Jenis** | **Entiti Luaran** |
| **Keterangan** | **Pelajar KV Shah Alam yang memohon untuk menginap di asrama** |

**Contoh Kamus Data bagi Jadual Pelajar**

1. **MEMBINA JADUAL MENGGUNAKAN SQL**

**Password WAMP: *root123***

* + 1. **PENGENALAN KEPADA JENIS-JENIS DATA**

Apabila mencipta jadual, anda perlu memilih jenis data untuk setiap lajur yang akan dibentuk dalam jadual tersebut. Jenis data yang dipilih menentukan jenis data yang boleh disimpan dalam medan tersebut.

Pemilihan jenis data yang betul adalah sangat penting kerana pemilihannya memberikan kesan kepada prestasi keseluruhan pangkalan data. Jenis data juga menentukan jumlah ruang storan yang diperlukan.

Jadual 3.5.1 menerangkan jenis data dalam MySQL

|  |  |
| --- | --- |
| **Jenis Data** | **Keterangan** |
| CHAR | String yang **tetap** antara 0 – 255 patah perkataan.  Char digunakan untuk lajur yang mengandungi data yang tetap dan seragam. |
| VARCHAR | String yang **boleh diubah** antara 0 – 255 patah perkataan.  VARCHAR digunakan dalam lajur yang mengandungi data yang tidak seragam. |
| TINYTEXT | String dengan panjang maksimum 255 patah perkataan |
| TEXT | String dengan maksimum 65,535 patah perkataan |
| INT | Nombor dalam julat -2,147,483,648 hingga 2,147,483,648 atau  Positif 0 hingga 4,294,967,275 |
| FLOAT | Nombor kecil yang mempunyai titik perpuluhan |
| DOUBLE | Nombor besar yang mempunyai titik perpuluhan |
| DATE | Tarikh dengan format YYYY-MM-DD |
| DATETIME | Tarikh dan waktu dengan format YYYY-MM-DD/HH:MM:SS |
| TIME | Waktu dengan format HH:MM:SS |

* + 1. **PENYATAAN CREATE**

Penyataan CREATE digunakan untuk membina pangkalan data dan jadual. Selepas membina pangkalan data, arahan USE mesti digunakan untuk memilih pangkalan data yang hendak digunakan sebelum membina jadual. Arahan SHOW pula digunakan untuk melihat semula senarai pangkalan data atau jadual yang telah diwujudkan. Berikut adalah penerangan dan sintaks menggunakan penyataan CREATE, USE dan SHOW.

**CREATE DATABASE**

Penyataan CREATE DATABASE digunakan untuk membina pangkalan data.

**Sintaks:**

**CREATE DATABASE** nama\_pangkalan\_data;

**Contoh**:

Berikut adalah penyataan SQL untuk membina pangkalan data yang dinamakan sebagai “kolej\_cyberpoint” :



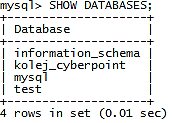
**SHOW DATABASES**

Penyataan SHOW DATABASES digunakan untuk memaparkan pangkalan data yang telah dibina.

**Sintaks:**

**SHOW DATABASES;**

**Contoh:**



**USE**

Penyataan USE digunakan untuk memilih pangkalan data yang hendak digunakan.

**Sintaks:**

**USE** nama\_pangkalan\_data;

**Contoh:**



**Penyataan CREATE TABLE**

Penyataan CREATE TABLE digunakan untuk :

* Membina jadual
* Menakrifkan jadual
* Menakrifkan lajur / *column*

**Sintaks:**

**CREATE TABLE** nama\_jadual (

nama\_medan1 jenis\_data (saiz) **PRIMARY KEY**,

nama\_medan2 jenis\_data (saiz) **NOT NULL**,

nama\_medan3 jenis\_data (saiz),

…………..

);

Dalam pembinaan jadual, jenis dan saiz data yang sesuai perlu diambilkira. Sebagai contoh data berjenis varchar, number, float, year dan sebagainya.

SQL membenarkan untuk mendefinisikan had atau sekatan (*constraints*)pada lajur dalam jadual. Jika pengguna dibenarkan untuk menyimpan data dalam lajur yang tidak mengikut had, ralat akan muncul. Sebagai contoh medan yang ingin ditentukan sebagai *Primary Key, Auto Increment, Unique, Null dan Not Null.*

**Contoh** :

Berikut adalah penyataan SQL untuk membina jadual “pelajar” berdasarkan spesifikasi di bawah:

Sintaks:

CREATE TABLE pelajar

( id\_pelajar int NOT NULL AUTO\_INCREMENT,

nama\_pelajar varchar(50) NOT NULL,

email\_pelajar varchar(20) NOT NULL,

alamat\_pelajar varchar(100),

PRIMARY KEY (id\_pelajar) );

|  |  |  |
| --- | --- | --- |
| **Nama lajur (*column*)** | **Jenis data ( saiz data )** | ***Constraint*** |
| id\_pelajar | int | Auto Increment  Primary Key |
| nama\_pelajar | varchar(50) | Not Null |
| email\_ pelajar | varchar(20) | Not Null |
| alamat\_ pelajar | varchar(100) |  |

Berdasarkan contoh di atas, medan id\_pelajar ditetapkan sebagai data yang akan mengalami peningkatan nombor atau *Auto Increment*. Medan ini juga memestikan pengguna mengisi data dan tidak dibenarkan ruang dibiarkan kosong atau *Not Null.* Medan ini juga ditetapkan sebagai kekunci utama atau *Primary Key* bagi jadual ‘pelajar’.

Medan nama\_pelajar merupakan medan yang akan menerima data berjenis gabungan teks, simbol dan nombor. Medan ini akan menerima sehingga 50 karekter sahaja.

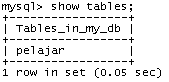
**Penyataan SHOW TABLES;**

Penyataan SHOW TABLES digunakan untuk memaparkan jadual-jadual yang telah dibina. Di dalam pangkalan data yang telah dipilih tadi.

**Sintaks:**

**SHOW TABLES**;

**Contoh**:



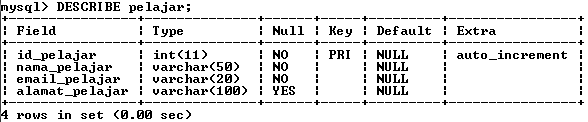
**Penyataan DESCRIBE**

Penyataan DESCRIBE digunakan untuk memaparkan medan-medan yang telah didefinisikan dalam jadual.

**Sintaks:**

**DESCRIBE** nama\_jadual;

**Contoh**:



1. **PENYATAAN ALTER**

Penyataan ALTER digunakan untuk menambah, menghapuskan atau mengubahsuai lajur (column) dalam jadual yang telah dibina.

* + - 1. **PENYATAAN ALTER TABLE – ADD COLUMN**

Penyataan ALTER TABLE ADD COLUMN digunakan untuk menambah lajur dalam jadual yang sedia ada.

**Sintaks:**

**ALTER TABLE** nama\_jadual **ADD** nama\_lajur jenis\_data;

**Contoh:**

Berikut adalah penyataan SQL untuk menambah lajur “nomatrik\_p” di dalam jadual “pelajar”

ALTER TABLE pelajar ADD nomatrik\_p int ;

* + - 1. **PENYATAAN ALTER TABLE – MODIFY COLUMN**

Penyataan ALTER TABLE - MODIFY COLUMN digunakan untuk mengubahsuai jenis dan saiz data bagi lajur(column) dalam jadual yang sedia ada.

**Sintaks:**

ALTER TABLE nama\_jadual MODIFY nama\_lajur

definisi\_baharu;

**Contoh:**

Berikut adalah penyataan SQL untuk mengubahsuai saiz data bagi lajur “nomatrik\_p” di dalam jadual “pelajar” :

ALTER TABLE pelajar MODIFY nomatrik\_p varchar(15) ;

* + - 1. **PENYATAAN ALTER TABLE – DROP COLUMN**

Penyataan ALTER TABLE - DROP COLUMN digunakan untuk menghapuskan lajur dalam jadual.

**Sintaks:**

ALTER TABLE nama\_jadual DROP nama\_lajur;

**Contoh:**

Berikut adalah penyataan SQL untuk menghapuskan lajur “email\_pelajar” di dalam jadual “pelajar” :

ALTER TABLE pelajar DROP email\_pelajar ;

**PENYATAAN DROP**

Penyataan DROP digunakan untuk menyingkir atau menghapuskan jadual dan pangkalan data yang sedia ada.

**PENYATAAN DROP TABLE**

Penyataan DROP TABLE digunakan untuk menghapuskan jadual yang sedia ada.

**Sintaks:**

DROP TABLE nama\_jadual;

**PENYATAAN DROP DATABASE**

Penyataan DROP DATABASE digunakan untuk menghapuskan pangkalan data yang sedia ada.

**Sintaks:**

DROP DATABASE nama\_pangkalan\_data;

1. **MEMANIPULASI DATA**

Data Manipulating Language atau Bahasa Pengolahan Data (DML) berfungsi untuk memanipulasi data dalam pangkalan data.

Setelah struktur jadual dicipta, anda boleh memasukkan data ke dalam jadual tersebut.

Selain itu, anda juga boleh menghapuskan dan mengubah suai data yang telah dimasukkan.

**Penyataan INSERT INTO**

Digunakan untuk memasukkan data ke dalam jadual

**INSERT INTO** nama\_jadual **VALUES** (nilai1,nilai2, nilai3,…,nilaiN);

Jika memerlukan hanya beberapa nilai dimasukkan kedalam struktur yang berkenaan, boleh menggunakan sintaks:

**INSERT INTO** nama\_jadual (medan1,medan2,medan3,…)

**VALUES** (nilai1,nilai2, nilai3,…,nilaiN);

**Penyataam DELETE**

Digunakan untuk memadam data dalam jadual. Untuk memadam rekod tertentu, penyatan WHERE diperlukan.

**DELETE FROM** nama\_jadual

**WHERE** nama\_medan = nilai;

**Penyataan UPDATE**

Untuk mengemaskini rekod yang sedia ada dalam jadual. Penyataan SET menunjukkan data baharu yang hendak dimasukkan manakala WHERE digunakan untuk mengemaskini rekod tertentu sahaja.

**UPDATE nama\_jadual**

**SET nama\_medam = nilai\_baharu**

**WHERE nama\_medan = nilai\_tertentu;**

1. **MENDAPATKAN SEMULA DATA**

Bagi mencapai rekod dalam jadual arahan SELECT akan digunakan.

**Penyataan SELECT**

Pernyataan ini terdiri daripada tiga pernyataan asas

|  |  |
| --- | --- |
| **Pernyataan** | **Keterangan** |
| SELECT | Senarai nama medan yang ingin dipaparkan |
| FROM | Nama jadual yang hendak dicapai |
| WHERE | Menentukan rekod tertentu yang ingin dipaparkan. Tanpa penyataan ini, semua rekod akan dipaparkan |

Penyataan SELECT dan FROM perlu dinyatakan apabila mencapai data manakala pernyataan WHERE adalah untuk pilihan.

Penyataan WHERE hanya diperlukan jika anda ingin mencapai data yang tertentu. Sekiranya terdapat lebih daripada satu syarat, pernyataan AND boleh digunakan.

Sintaks:

**SELECT** nama\_medan

**FROM** nama jadual

**WHERE** syarat1

**AND** syarat2

**AND** syarat3

**….;**

Penyataan GROUP BY

Digunakan untuk menggabungkan beberapa rekod dalam jadual kepada kumpulan kecil

Sintaks:

**SELECT** nama\_medan

**FROM** nama\_jadual

**GROUP BY** nama\_medan**;**

Pernyataan ORDER BY

Setelah dimasukkan data ke dalam jadual, anda boleh memanggil data dalam bentuk susunan menaik atau menurun dengan perkataan ORDER BY.

Pada asasnya, ia akan mengisih data mengikut susunan menaik. Jika anda ingin mengisih mengikut susuanan menurun, anda perlu menambah perkataan DESC

Sintaks:

**SELECT** nama\_medan

**FROM** nama\_jadual

**ORDER BY** nama\_medan**;**

**SOALAN**

1. Penyataan DROP digunakan untuk menyingkir atau menghapuskan ……………………. dan ……………………………………….yang sedia ada.

(2 markah)

1. Jawab soalan di bawah dengan merujuk jadual Pesakit dalam pangkalan data Hospital\_Bahagia.

|  |  |  |
| --- | --- | --- |
| **Nama lajur (*column*)** | **Jenis data ( saiz data )** | ***Constraint*** |
| No\_KP | Int(14) | Auto Increment  Primary Key |
| Nama\_Pesakit | varchar(50) | Not Null |
| Wad\_Pesakit | varchar(20) | Not Null |
| Nama\_Penyakit | varchar(100) |  |

Jadual Pesakit

1. Tuliskan arahan SQL untuk menghapuskan jadual Pesakit.

……………………………………………………………………………………….

1. markah)
2. Tuliskan arahan SQL untuk menghapuskan pangkalan data tersebut.

……………………………………………………………………………………….

(1 markah)

References :

1. <https://www.w3schools.com/sql/sql_intro.asp>
2. Buku Teks Sains Komputer Tingkatan 5, KPM, 2017

**SOALAN**

1. Nyatakan tujuan penyataan ALTER.

………………………………………………………………………………………………

(1 markah)

1. Jawab soalan di bawah dengan merujuk jadual Pesakit.

|  |  |  |
| --- | --- | --- |
| **Nama lajur (*column*)** | **Jenis data ( saiz data )** | ***Constraint*** |
| No\_KP | Int(14) | Auto Increment  Primary Key |
| Nama\_Pesakit | varchar(50) | Not Null |
| Wad\_Pesakit | varchar(20) | Not Null |
| Nama\_Penyakit | varchar(100) |  |

Jadual Pesakit

1. Mengubah suai nama lajur No\_KP kepada No\_Kad\_Pengenalan.

……………………………………………………………………………………….

1. Mengubah suai dengan menambah lajur ‘Alamat’.

……………………………………………………………………………………….

1. Mengubah suai dengan membuang lajur ‘Nama\_Penyakit’.

……………………………………………………………………………………….

1. Mengubah suai dengan menukar jenis data Wad\_Pesakit kepada varchar(10)

……………………………………………………………………………………….

(4 markah)

References :

1. <https://www.w3schools.com/sql/sql_intro.asp>

**SOALAN**

1. Nyatakan tujuan penyataan CREATE.

………………………………………………………………………………………………

(1 markah)

1. Tuliskan sintaks bagi membina pangkalan data dengan nama ‘KV DSAZI’

………………………………………………………………………………………………

(1 markah)

1. Tuliskan sintaks bagi membina jadual dengan merujuk *data library* di bawah.

|  |  |  |
| --- | --- | --- |
| **Nama lajur (*column*)** | **Jenis data ( saiz data )** | ***Constraint*** |
| No\_KP | Int(14) | Auto Increment  Primary Key |
| Nama\_Pesakit | varchar(50) | Not Null |
| Wad\_Pesakit | varchar(20) | Not Null |
| Nama\_Penyakit | varchar(100) |  |

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

(6 markah)

1. Tuliskan sintaks bagi memaparkan semula senarai pangkalan data.

………………………………………………………………………………………………

(1 markah)

1. Tuliskan sintaks bagi memaparkan jadual yang telah dibina.

………………………………………………………………………………………………

(1 markah)

References :

1. <https://www.w3schools.com/sql/sql_intro.asp>

**SOALAN/***QUESTION* **:**

1. Apakah rasionalnya kewujudan pangkalan data
2. Berikan lima contoh aplikasi pangkalan data digunakan
3. Lukiskan gambar rajah konsep pangkalan data
4. Bezakan ketiga-tiga paras dalam seni bina tiga paras ANSI-SPARC menggunakan gambar rajah
5. Nyatakan ciri-ciri Entiti Relationship Diagram (ERD)

**RUJUKAN/***REFERENCE* **:**

1. Cosmopoint Institut of information technology , 2001, data analysis & database design, Penerbitan cosmopoint
2. Abdullah Embong, 2000, sistem pangkalan data konsep asas, rekabentuk dan pelaksanaan, tradisi ilmu sdb.bhd
3. Modu Lakulu, Aslina Saad, 2007, Sistem Pangkalan Data, Universiti Pendidikan Sultan Idris