[ISG2E4] Pemrograman Berorientasi Objek

Minggu 12
Konsep JAVA - DBMS

Overview

- Mahasiswa mampu menjelaskan kegunaan library JDBC
- Mahasiswa mampu menggunakan library JDBC
- Mahasiswa mampu mengintegrasikan aplikasi tugas besar dengan database Oracle menggunakan JDBC;
- Mahasiswa mampu mempresentasikan dan memberikan laporan hasil final aplikasi tugas besar

Java Database

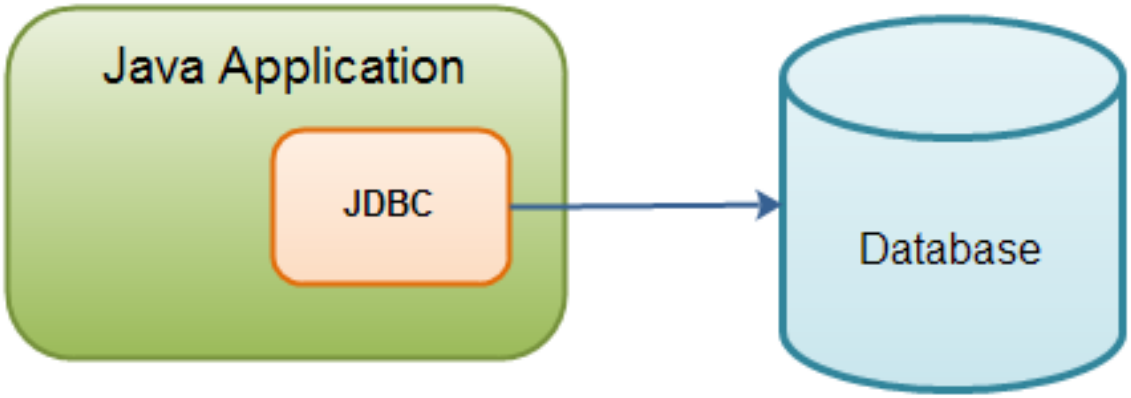
1. Pengantar Database
2. Pengantar SQL
3. Koneksi Aplikasi Java ke Database
4. Studi Kasus Aplikasi Java Database

Apa itu JDBC ?

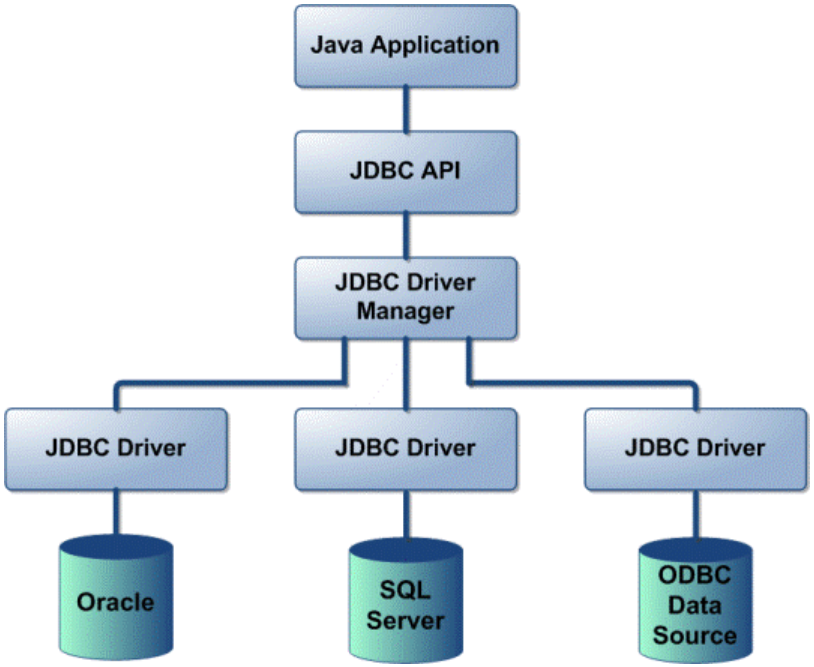
JDBC (Java database connectivity) adalah java library yang memungkinkan program Java untuk mengakses sistem database manajemen yang telah dibuat.

JDBC menyediakan methods untuk melakukan query dan modifikasi data pada RDBMS seperti Oracle, SQL Server, MySQL, dll menggunakan Driver Manager. JDBC mirip dengan ODBC (Open Database Connectivity), hanya saja JDBC spesifik digunakan untuk program Java. ODBC bersifat language independent

Ilustrasi Penggunaan JDBC



Ilustrasi Lain



10 Arsitektur JDBC

1. Database connections
2. SQL Statements
3. Result set
4. Database metadata
5. Prepared Statement
6. Binary Large Objects (BLOBs)
7. Character Large Objects (CLOBs)
8. Callable statements
9. Database Driver
10. Driver Manager

4 Komponen JDBC

1. JDBC API
2. JDBC Driver Manager
3. JDBC Test Suite
4. JDBC-ODBC Bridge

1. JDBC API

JDBC-API menyediakan fasilitas untuk mengakses database relasional dari program berbahasa Java. Melalui komponen ini user dapat melakukan proses query dan perubahan data dalam database. JDBC-API memiliki package utama yang tersedia pada java.sql dan javax.sql.

2. JDBC Driver Manager

Merupakan komponen kelas utama yang mendefinisikan object yang mengkoneksikan aplikasi Java ke JDBC driver. Komponen ini berfungsi untuk mengatur beberapa tipe JDBC database driver yang berbeda. JDBC Driver Manager memilih driver mana yang sesuai untuk koneksi ke suatu database.

3. JDBC Test Suite

Komponen ketiga ini memastikan JDBC driver dapat menjalankan program user dan sangat berguna dalam melakukan testing sebuah aplikasi yang menggunakan teknologi JDBC.

JDBC-ODBC Bridge

Telkom University

Adalah database driver yang menggunakan ODBC driver untuk koneksi ke database serta memiliki fungsi utama untuk translasi dari JDBC method calls ke ODBC function calls dan mengimplementasikan JDBC untuk semua driver yang didukung oleh ODBC. Komponen ini dapat diimplementasikan dalam package sun.jdbc.odbc.

Langkah Dasar Penggunaan JDBC di Java

1. Establish a **connection**
2. Create JDBC **Statements**
3. Execute **SQL** Statements
4. GET **ResultSet**
5. **Close** connections

PENGANTAR DATABASE

Introduction to Database

- Database system is a computer based record keeping system
- It is a system whose overall purpose is to record and maintain information that is deemed important to the organization
- Database is collection of stored operational data which can be used and shared by different applications and users of any organization

Why Database

- Database system provides the organization with centralized control of its operational data, which is one of its most valuable assets
- This is totally opposite of the situation that is happening in many organizations, where typically each application has its own private files (flat file). This makes the operational data widely dispersed and difficult to control

Advantage of Centralized Database

- Redundancy can be reduced
- Inconsistency can be avoided
- Data can be shared
- Standards can be enforced
- Security restrictions can be applied
- Integrity can be maintained
- Conflicting requirements can be balanced

Disadvantage of Database Systems

- Database is more vulnerable to destruction thru:
  - machine malfunction
  - personal error
  - Deliberate human tampering
- Cost: the cost of required hardware, DB development, and DB maintenance is high
- Complexity: Due to its complexity, the user should understand it well enough to use it efficiently and effectively

Database Models - Product - Vendor

MODEL	PRODUCT	VENDOR
1. Relational	DB2	IBMSQL/DS
	Ingress	Relational Tech.
	Oracle	Oracle corp
	Access	Microsoft
	PostgreSQL	
	MySQL	
2. Network	DMS100	Unysis
	IDMS	Cullinet
3. Heirarchical	IMS	IBM
	System 2000	Intel
4. Object oriented	Starburst	IBM
	Gemstone	
	Orion	

Relational Database

Telkom University

- Relational database is a collection of tables
- Formally a table is called a relation
- Database is a structure that can hold information about tables, rows, and columns

Relational Database

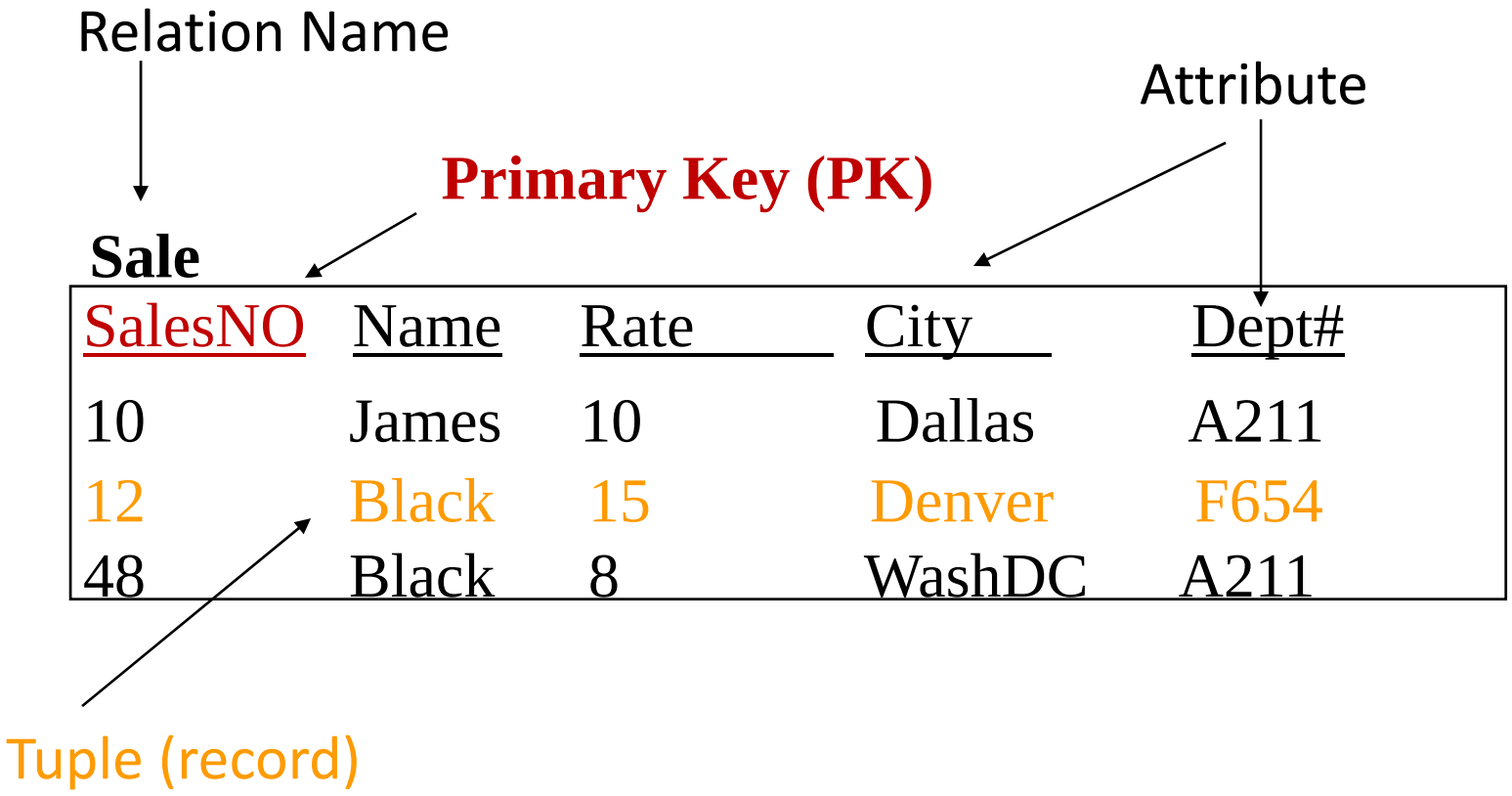
Relational Model	Relational DBMS	Traditional File System
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field
Primary Key (PK)	Primary Key (PK)	Search Key
Relationship (FK)	Relationship (FK)	Not Used

Relational Database

1. Primary Key (PK): An attribute which can uniquely identify each record (tuple) of a relation (table)

2. Foreign Key (FK): An attribute which is a regular attribute in one table but a primary key in another table

Example of a Relational Database



Example of a Relational Database

Customer

CustID	Name	Balance	City	SaleNo
132	Black	2000.00	Dallas	10
135	Tom	129.89	Denver	12
198	Tom	(132.90)	Dallas	10

SalesNO is PK in Sales table

Sales

SalesNO	Name	Rate	City	Dept#
10	James	10	Dallas	A211
12	Black	15	Denver	F654
48	Black	8	WashDC	A211

Example of a Relational Database

Customer

CustID	Name	Balance	City	SaleNo
132	Black	2000.00	Dallas	10
135	Tom	129.89	Denver	12
198	Tom	(132.90)	Dallas	10

SalesNO is PK in Sales table and FK in Customer table

Sales

SalesNO	Name	Rate	City	Dept#
10	James	10	Dallas	A211
12	Black	15	Denver	F654
48	Black	8	WashDC	A211

Order

ONO	DATE	CustID	SalesNO
102	11/2/94	132	10
199	2/15/95	135	12
92	10/4/94	102	53

OrderLine

ONO	**Oline#**	Part#	Qty	Part#
102	1	12.00	10	EX454
102	2	129.89	1	DE012
199	1	32.90	3	DC810

Customer

CustID	Name	Balance	City	SaleNo
132	Black	2000.00	Dallas	10
135	Tom	129.89	Denver	12
198	Tom	(132.90)	Dallas	10

Sales

SalesNO	Name	Rate	City	Dept#
10	James	10	Dallas	A211
12	Black	15	Denver	F654
48	Black	8	WashDC	A211

Functionality of a DBMS

- The programmer sees SQL, which has two components:
  1. Data Definition Language (DDL)
  2. Data Manipulation Language (DML)

- Behind the scenes the DBMS has:
  1. Query engine
  2. Query optimizer
  3. Storage management
  4. Transaction Management (concurrency, recovery)

How the Programmer Sees the DBMS

1. Start with DDL to *create tables*:

```
CREATE TABLE Students (
        Name CHAR(30)
        SSN CHAR(9) PRIMARY KEY NOT NULL,
        Category CHAR(20)
)  . . .
```

2. Continue w

```
INSERT INTO Students
VALUES('Charles', '123456789', 'undergraduate')
. . . .
```

Transactions

- Enroll “Mary Johnson” in “CSE444”:

```
BEGIN TRANSACTION;

INSERT INTO Takes
   SELECT Students.SSN, Courses.CID
   FROM Students, Courses
   WHERE Students.name = ‘Mary Johnson’ and
               Courses.name = ‘CSE444’

-- More updates here....

IF everything-went-OK
     THEN COMMIT;
ELSE ROLLBACK
```

If system crashes, the transaction is still either committed or aborted

Transactions

- A transaction = sequence of statements that either all succeed, or all fail
- Transactions have the ACID properties:
  1. A = atomicity (a transaction should be done or undone completely)
  2. C = consistency (a transaction should transform a system from one consistent state to another consistent state)
  3. I = isolation (each transaction should happen independently of other transactions)
  4. D = durability (completed transactions should remain permanent)

Queries

- Find all courses that “Mary” takes

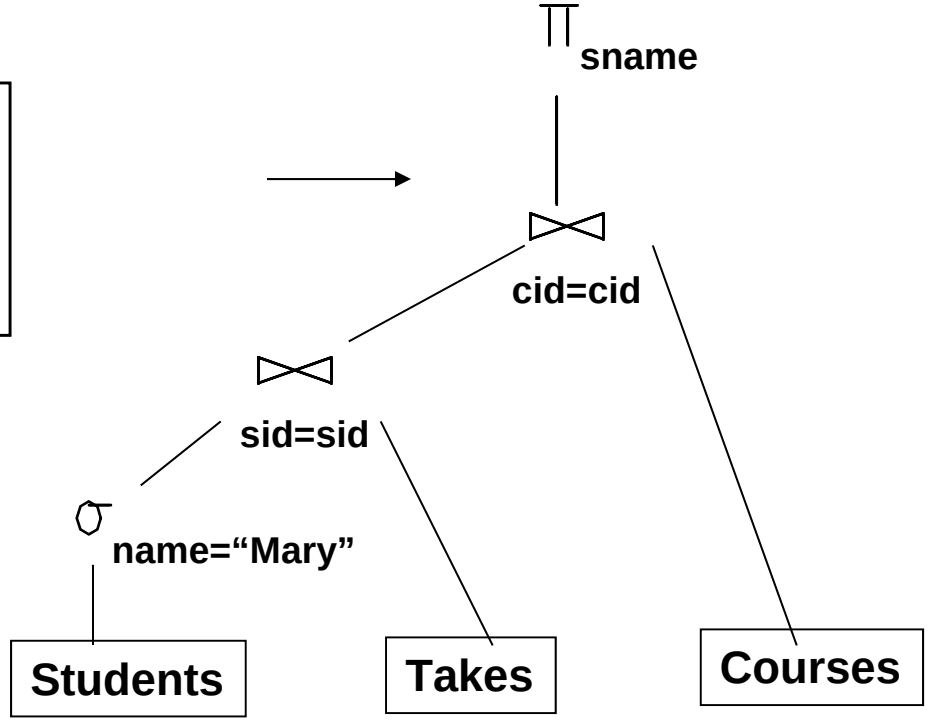
```
SELECT  C.name
FROM    Students S, Takes T, Courses C
WHERE   S.name=“Mary” and
        S.ssn = T.ssn and T.cid = C.cid
```

- What happens behind the scene ?
  - Query processor figures out how to answer the query efficiently.

Queries, Behind the Scene

Declarative SQL query → **Imperative query execution plan:**

```
SELECT  C.name
FROM Students S, Takes T, Courses C
WHERE S.name="Mary" and
        S.ssn = T.ssn and T.cid = C.cid
```



The **optimizer** chooses the best execution plan for a query

Pengantar SQL

SQL Introduction

Telkom University

- Standard language for querying and manipulating data
- SQL = Structured Query Language
- Many standards out there:
  - ANSI SQL
  - SQL92 (a.k.a. SQL2)
  - SQL99 (a.k.a. SQL3)
  - Vendors support various subsets of these
  - What we discuss is common to all of them

SQL

Telkom University

- Data Definition Language (DDL)
  - Create/alter/delete tables and their attributes
- Data Manipulation Language (DML)
  - Query one or more tables
  - Insert/delete/modify tuples in tables
- Transact-SQL
  - Idea: package a sequence of SQL statements → server

Data Types in SQL

Telkom University

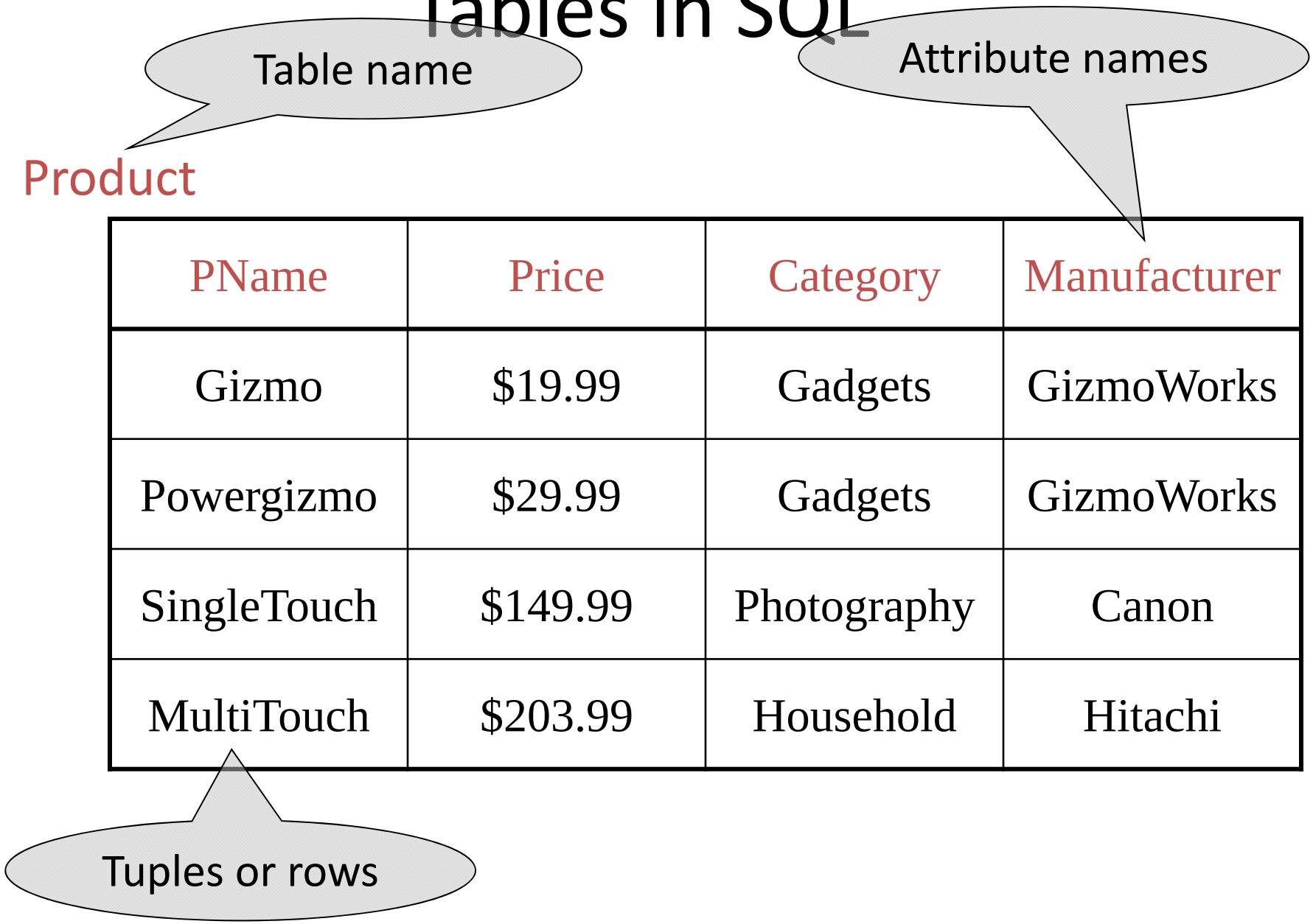
- Characters:
  - CHAR(20) -- fixed length
  - VARCHAR(40) -- variable length
- Numbers:
  - BIGINT, INT, SMALLINT, TINYINT
  - REAL, FLOAT -- differ in precision
  - MONEY
- Times and dates:
  - DATE
  - DATETIME -- SQL Server
- Others... All are simple

SQL Data Type vs Java Data

Telkom University

SQL Data Type	Java Data Type
INTEGER or INT	int
REAL	float
DOUBLE	double
DECIMAL(m, n)	Fixed-point decimal numbers with m total digits and n digits after the decimal point; similar to BigDecimal.
BOOLEAN	Boolean
VARCHAR(n)	Variable-length String of length up to n
CHARACTER(n) or CHAR(n)	Fixed-length String of length n

Tables in SQL



Tables Explained

Telkom University

- A tuple = a record
  - Restriction: all attributes are of atomic type
- A table = a set of tuples
  - Like a list…
  - …but it is unorderd: no **first()**, no **next()**, no **last()**.
- No nested tables, only flat tables are allowed!

Tables Explained

- The schema of a table is the table name and its attributes:

  Product(PName, Price, Category, Manfacturer)

- A key is an attribute whose values are unique; we underline a key

  Product(PName, Price, Category, Manfacturer)

SQL Query

Basic form: (plus many many more bells and whistles)

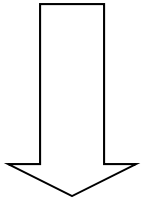
```
SELECT  attributes
FROM    relations (possibly multiple, joined)
WHERE   conditions (selections)
```

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

```
SELECT  *
FROM    Product
WHERE   category='Gadgets'
```



PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks

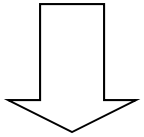
"selection"

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

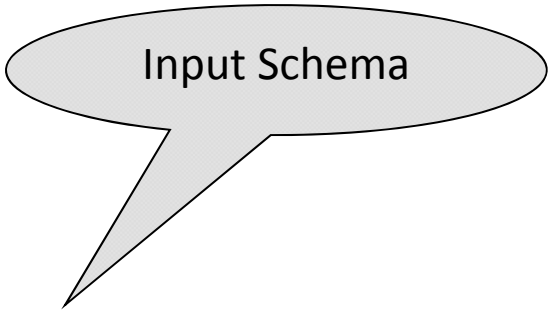
```
SELECT   PName, Price, Manufacturer
FROM     Product
WHERE    Price > 100
```



"selection" and "projection"

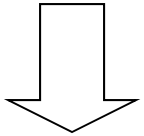
PName	Price	Manufacturer
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

A Notation for SQL Queries

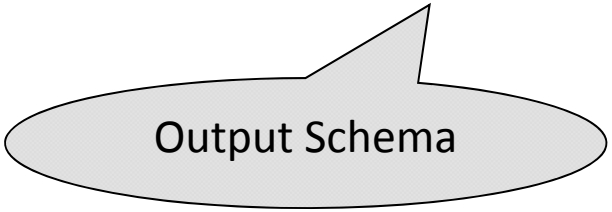


Product(PName, Price, Category, Manfacturer)

```
SELECT   PName, Price, Manufacturer
FROM     Product
WHERE    Price > 100
```



Answer(PName, Price, Manfacturer)



Selections

Telkom University

What goes in the WHERE clause:

- x = y, x < y, x <= y, etc
  - For number, they have the usual meanings
  - For CHAR and VARCHAR: lexicographic ordering
    - Expected conversion between CHAR and VARCHAR
  - For dates and times, what you expect...
- Pattern matching on strings: s LIKE p

The **LIKE** operator

- s **LIKE** p: pattern matching on strings
- p may contain two special symbols:
  - % = any sequence of characters
  - _ = any single character

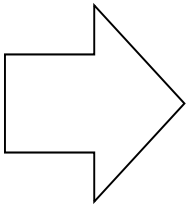
Product(Name, Price, Category, Manufacturer)

Find all products whose name mentions ‘gizmo’:

```
SELECT  *
FROM    Products
WHERE   PName LIKE ‘%gizmo%’
```

Eliminating Duplicates

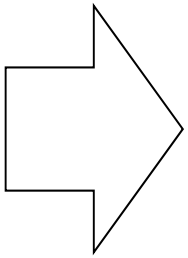
```
SELECT   category
FROM     Product
```



Category
Gadgets
Gadgets
Photography
Household

Compare to:

```
SELECT   DISTINCT category
FROM     Product
```



Category
Gadgets
Photography
Household

Ordering the Results

```
SELECT   pname, price, manufacturer
FROM     Product
WHERE    category='gizmo' AND price > 50
ORDER BY  price, pname
```

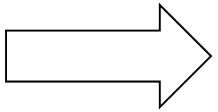
Ordering is ascending, unless you specify the DESC keyword.

Ties are broken by the second attribute on the ORDER BY list, etc.

Ordering the Results

```
SELECT   Category
FROM     Product
ORDER BY PName
```

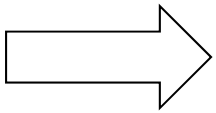
PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi



?

Ordering the Results

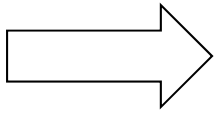
```
SELECT   DISTINCT category
FROM     Product
ORDER BY category
```



Category
Gadgets
Household
Photography

Compare to:

```
SELECT   DISTINCT category
FROM     Product
ORDER BY PName
```



?

Joins in SQL

- Connect two or more tables:

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

What is the Connection between them ?

Joins


```
SELECT   PName, Price
FROM     Product, Company
WHERE    Manufacturer=CName AND Country='Japan'
         AND Price <= 200
```

Joins in SQL

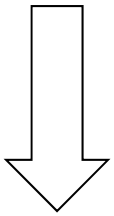
Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   PName, Price
FROM     Product, Company
WHERE    Manufacturer=CName AND Country='Japan'
         AND Price <= 200
```



PName	Price
SingleTouch	$149.99

Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

Find all countries that manufacture some product in the 'Gadgets' category.

```
SELECT   Country
FROM     Product, Company
WHERE    Manufacturer=CName AND Category='Gadgets'
```

Joins in SQL

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   Country
FROM     Product, Company
WHERE    Manufacturer=CName AND Category='Gadgets'
```

What is the problem ? What's the solution ?

Country
??
??

Joins

Product (pname, price, category, manufacturer)
Purchase (buyer, seller, store, product)
Person(persname, phoneNumber, city)

Find names of people living in Seattle that bought some product in the 'Gadgets' category, and the names of the stores they bought such product from

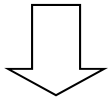
```
SELECT   DISTINCT persname, store
FROM     Person, Purchase, Product
WHERE    persname=buyer AND product = pname AND
         city='Seattle'  AND category='Gadgets'
```

Disambiguating Attributes

- Sometimes two relations have the same attribute:
  Person(pname, address, worksfor)
  Company(cname, address)

```
SELECT   DISTINCT pname, address
FROM      Person, Company
WHERE   worksfor = cname
```

Which address ?



```
SELECT   DISTINCT Person.pname, Company.address
FROM      Person, Company
WHERE   Person.worksfor = Company.cname
```

Tuple Variables

Product (pname, price, category, manufacturer)
Purchase (buyer, seller, store, product)
Person(persname, phoneNumber, city)

Find all stores that sold at least one product that the store 'BestBuy' also sold:

```
SELECT DISTINCT  x.store
FROM     Purchase AS x, Purchase AS y
WHERE   x.product = y.product AND y.store = 'BestBuy'
```

Answer (store)

Tuple Variables

General rule:
tuple variables introduced automatically by the system:

Product (name, price, category, manufacturer)

```
SELECT  name
FROM    Product
WHERE  price > 100
```

Becomes:

```
SELECT Product.name
FROM   Product AS Product
WHERE Product.price > 100
```

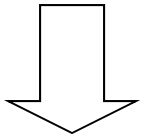
Doesn't work when Product occurs more than once:
In that case the user needs to define variables explicitely.

Renaming Columns

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

```
SELECT   Pname AS prodName, Price AS askPrice
FROM     Product
WHERE    Price > 100
```

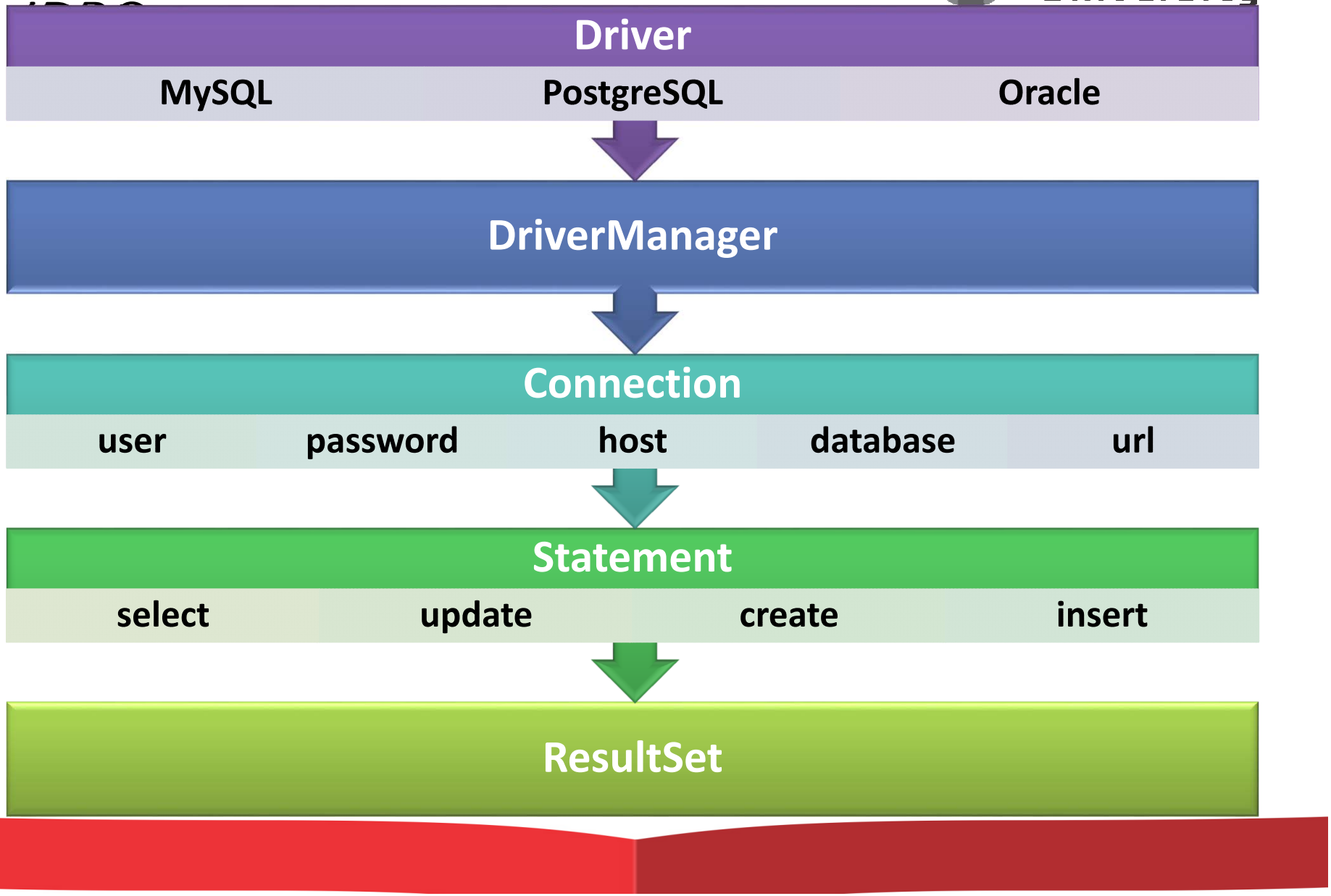


Query with renaming

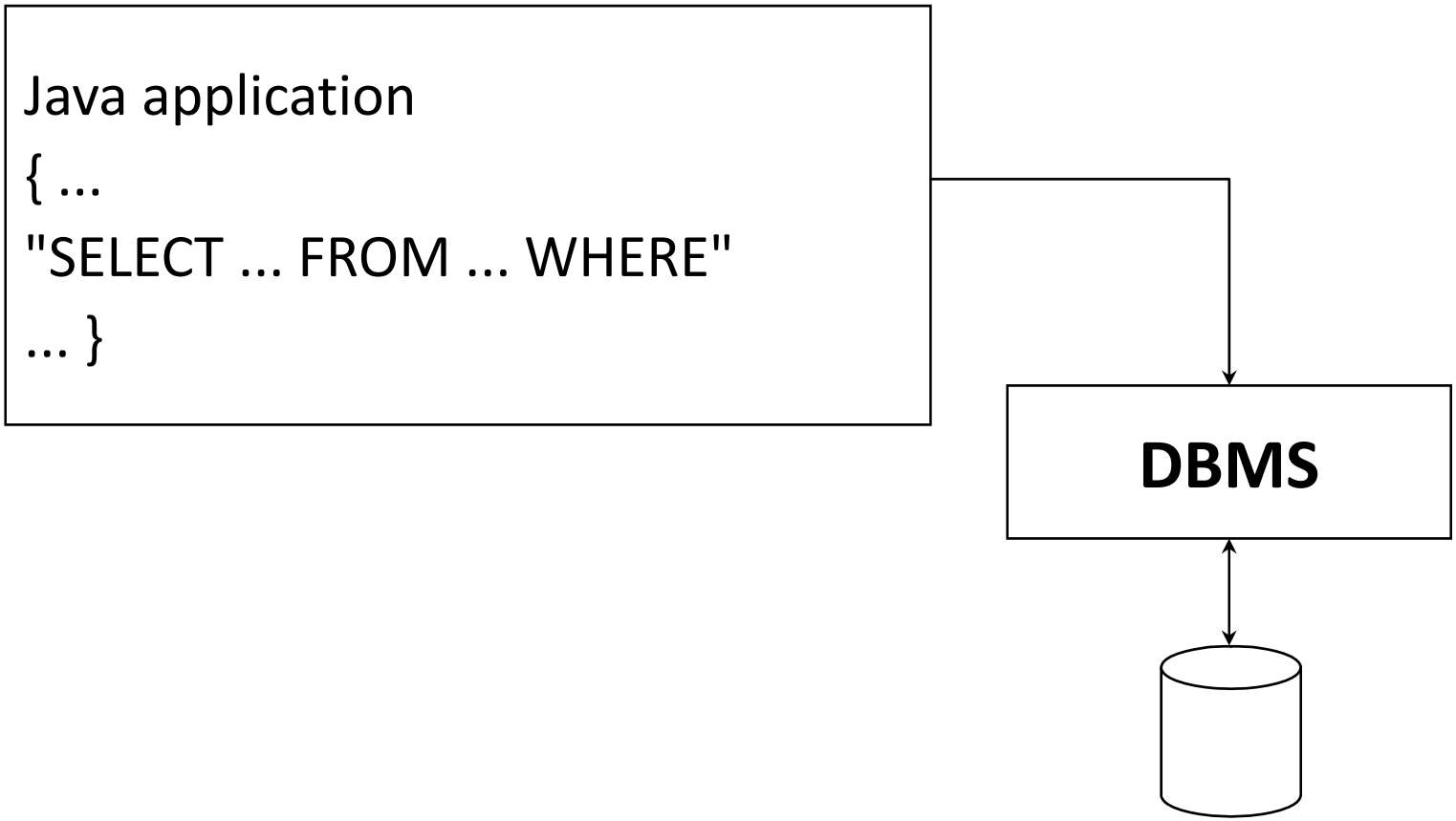
prodName	askPrice
SingleTouch	$149.99
MultiTouch	$203.99

Koneksi Aplikasi Java ke Database

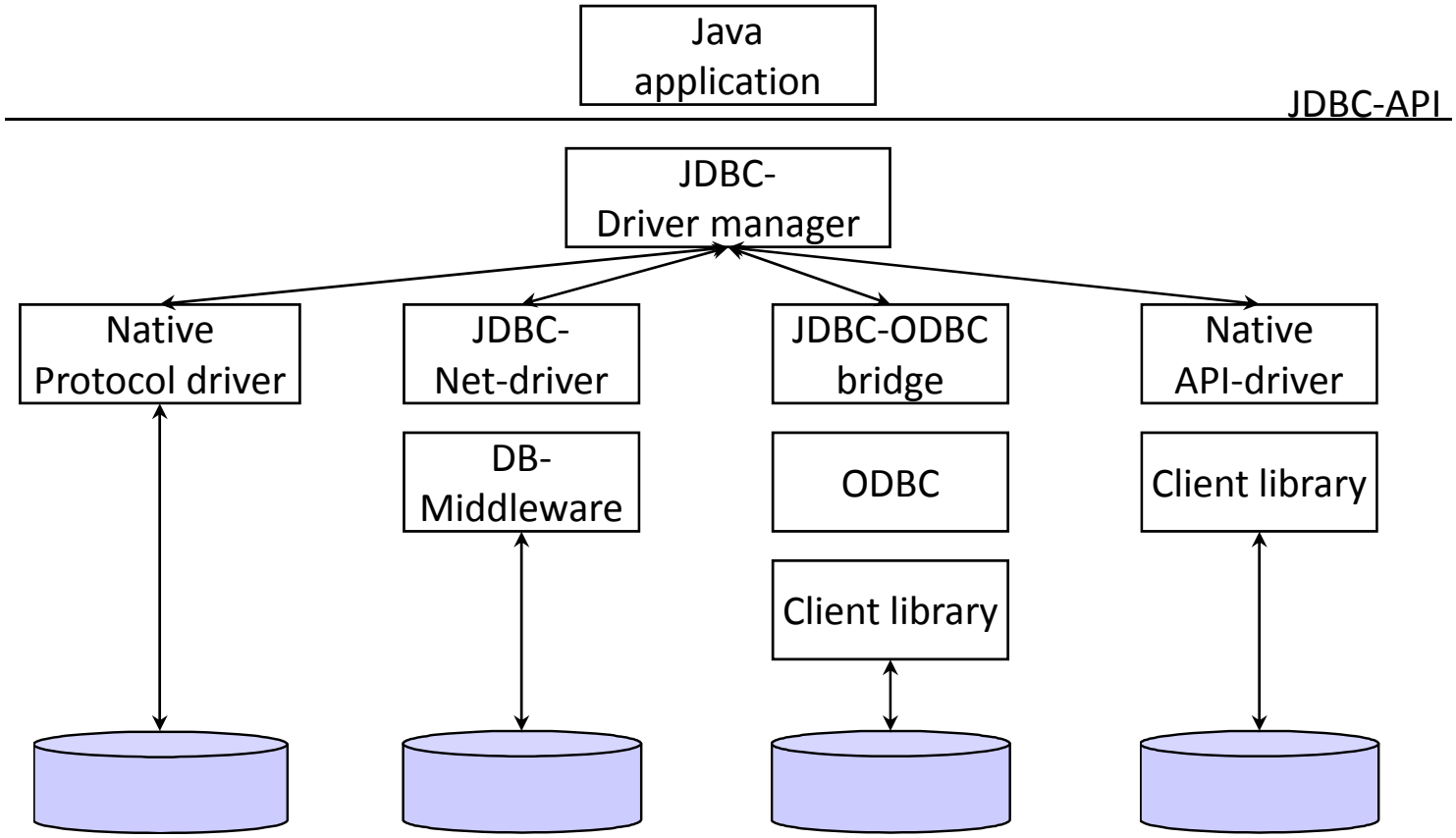
Tahapan Akses Database dengan



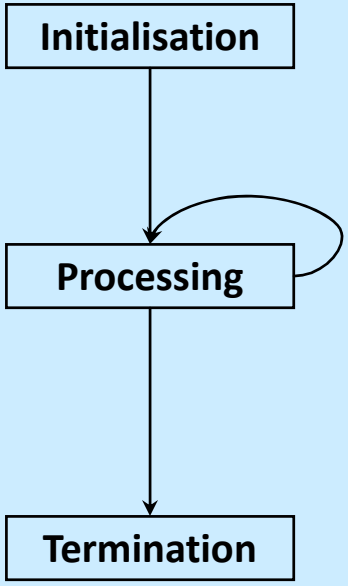
JDBC (Java DB Connectivity)



JDBC Drivers

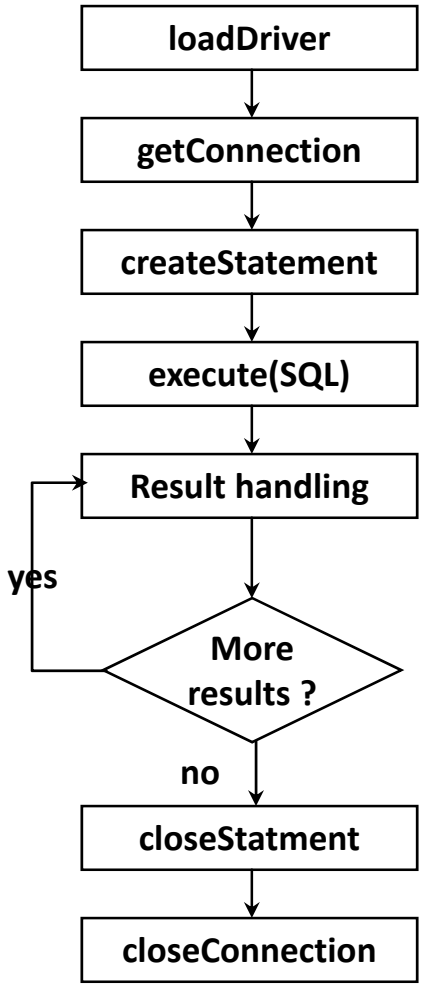


Running a JDBC Application



Phase	Task	Relevant java.sql classes
Initialisation	Load driver Create connection	DriverManager Connection
Processing	Generate SQL statements Process result data	Statement ResultSet etc.
Termination	Terminate connection Release data structures	Connection Statement etc.

A Simple JDBC application




```
import java.sql.*;
public class jdbctest {
  public static void main(String args[]){
      try{
       Class.forName("org.postgresql.Driver");
       Connection con = DriverManager.getConnection
          ("jdbc:postgresql://lsir-cis-pc8:5401/pcmdb", "user", "passwd");
       Statement stmt = con.createStatement();
       ResultSet rs = stmt.executeQuery
          ("select name, number from pcmtable where number < 2");
       while(rs.next())
                System.out.println(rs.getString(1) + " (" + rs.getInt(2) + ")");
       stmt.close()
       con.close();
      } catch(Exception e){
       System.err.println(e);
}
}}
```

Loading of Driver

Telkom University

- Creates an instance of the driver
- Registers driver in the driver manager
- Explicit loading

```
String l_driver = "org.postgresql.Driver";
Class.forName(l_driver);
```

- Several drivers can be loaded and registered

Implicit Driver Loading

Telkom University

- Setting system property: jdbc.drivers
  - A colon-separated list of driver classnames
- Can be set when starting the application

```
java -Djdbc.drivers=org.postgresql.Driver application
```

- Can also be set from within the Java application

```
Properties prp = System.getProperties();
prp.put("jdbc.drivers"
      "com.mimer.jdbc.Driver:org.postgresql.Driver");
System.setProperties(prp);
```

- The DriverManager class attempts to load all the classes specified in jdbc.drivers when the DriverManager class is initialized

Addressing Database

- A connection is a session with one database
- Databases are addressed using a URL of the form "jdbc:<subprotocol>:<subname>"
- Examples

  jdbc:postgresql:database

  jdbc:postgresql://host/database

  jdbc:postgresql://host:port/database
- Defaults: host=localhost, port=5432

Connecting to Database

Telkom University

- Connection is established
  Connection con =
  DriverManager.getConnection(URL,USERID,PWD);
- Connection properties (class Properties)
- Close the connection
  con.close();

Simple SQL Statements

Telkom University

- Statement object for invocation

```
stmt = conn.createStatement();
ResultSet rset= stmt.executeQuery(
        "SELECT address,script,type FROM worklist");
```

- ResultSet object for result processing

Studi Kasus Aplikasi Database

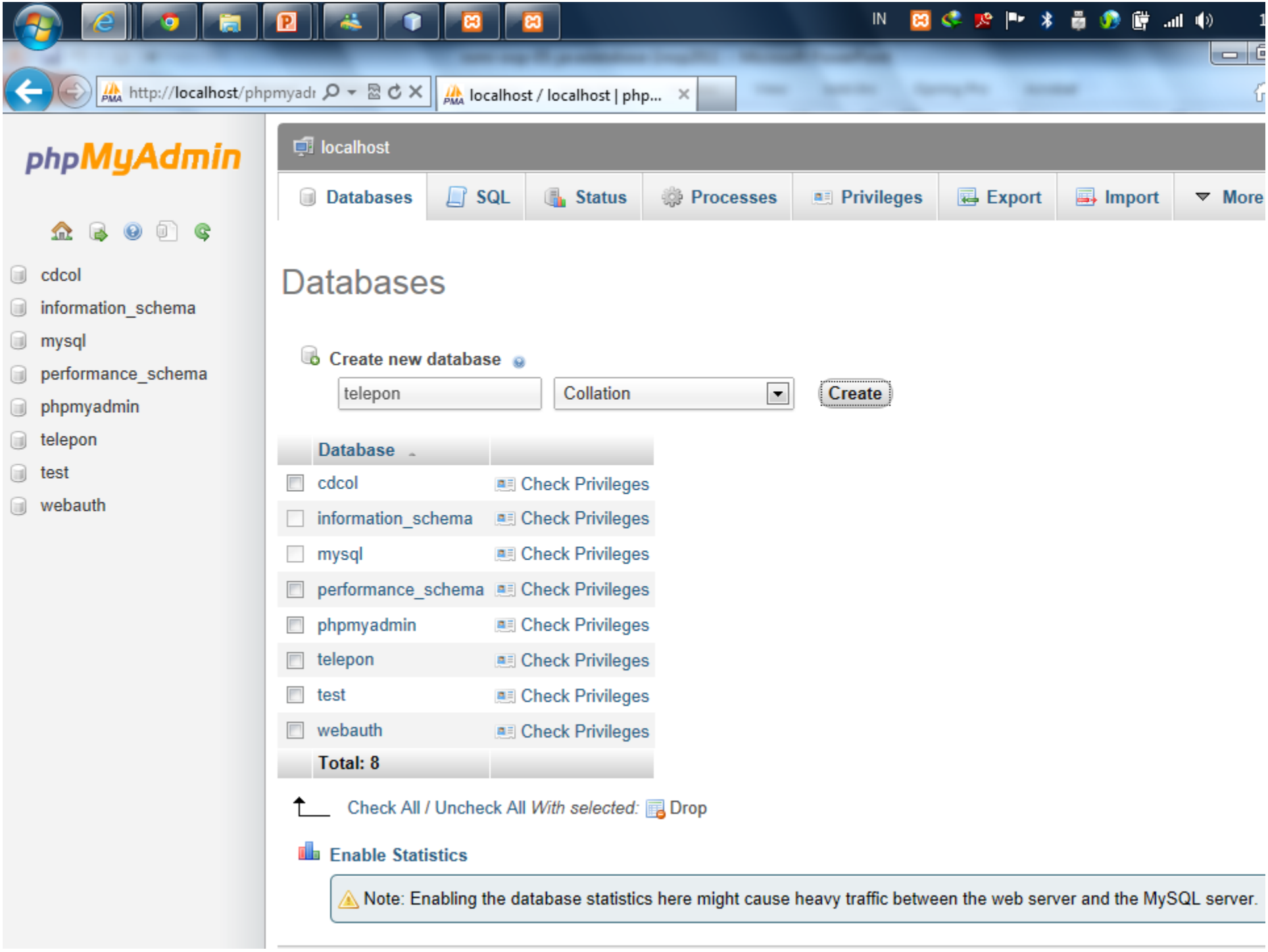
Aplikasi Database

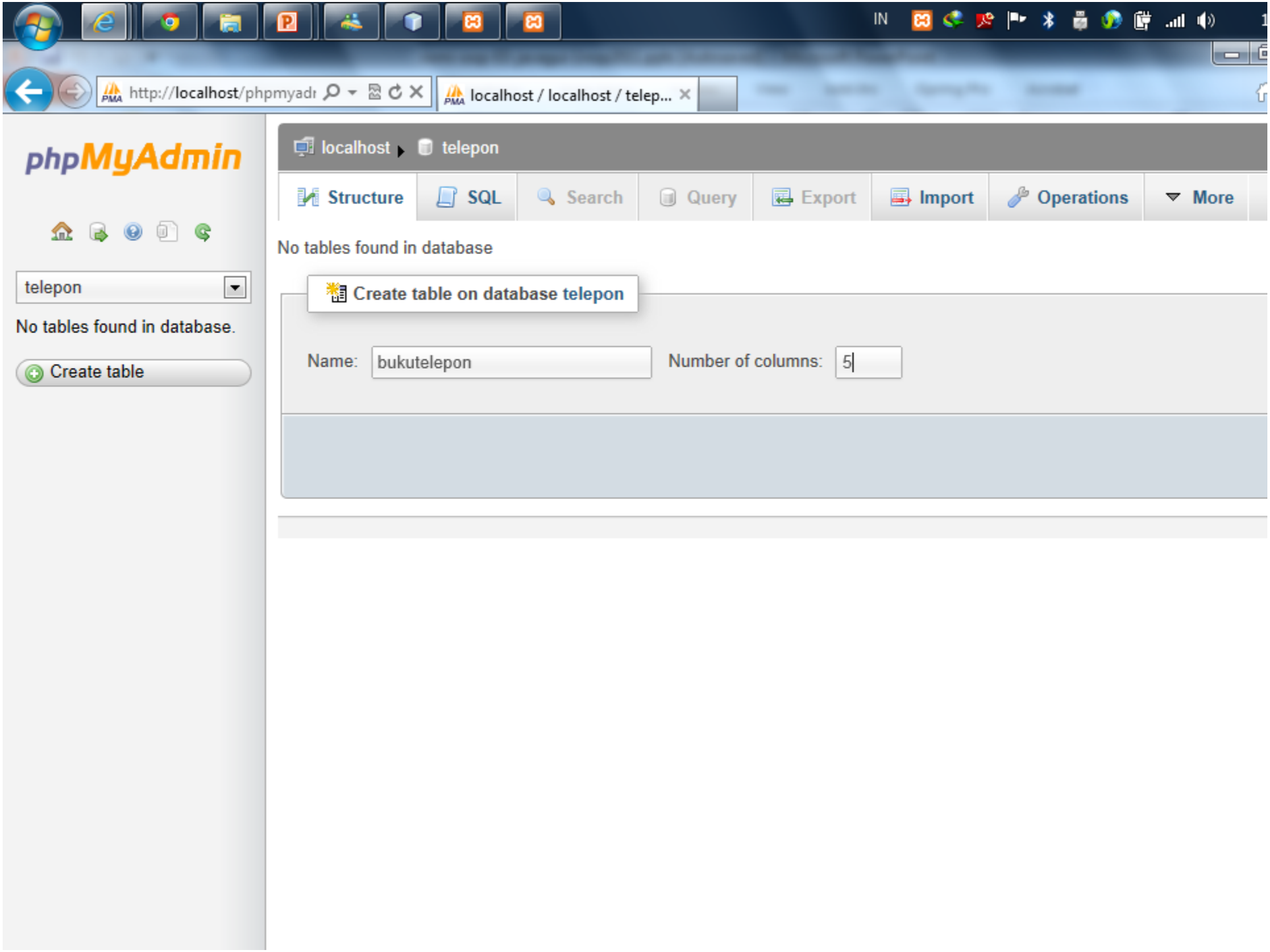
1. Aplikasi Telepon
2. Aplikasi Guru
3. Aplikasi Bank
4. Aplikasi Penjualan Barang

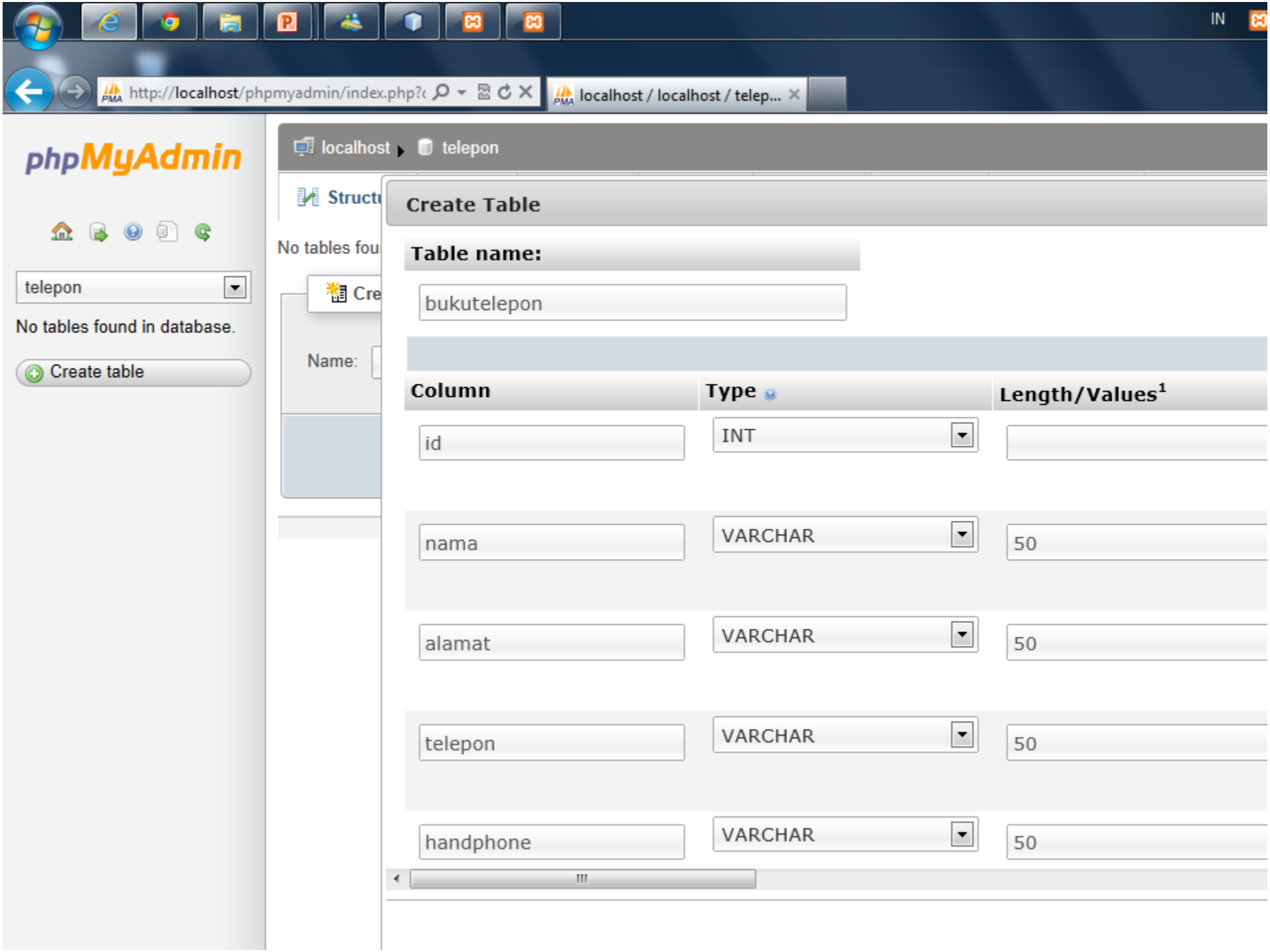
Aplikasi Telepon

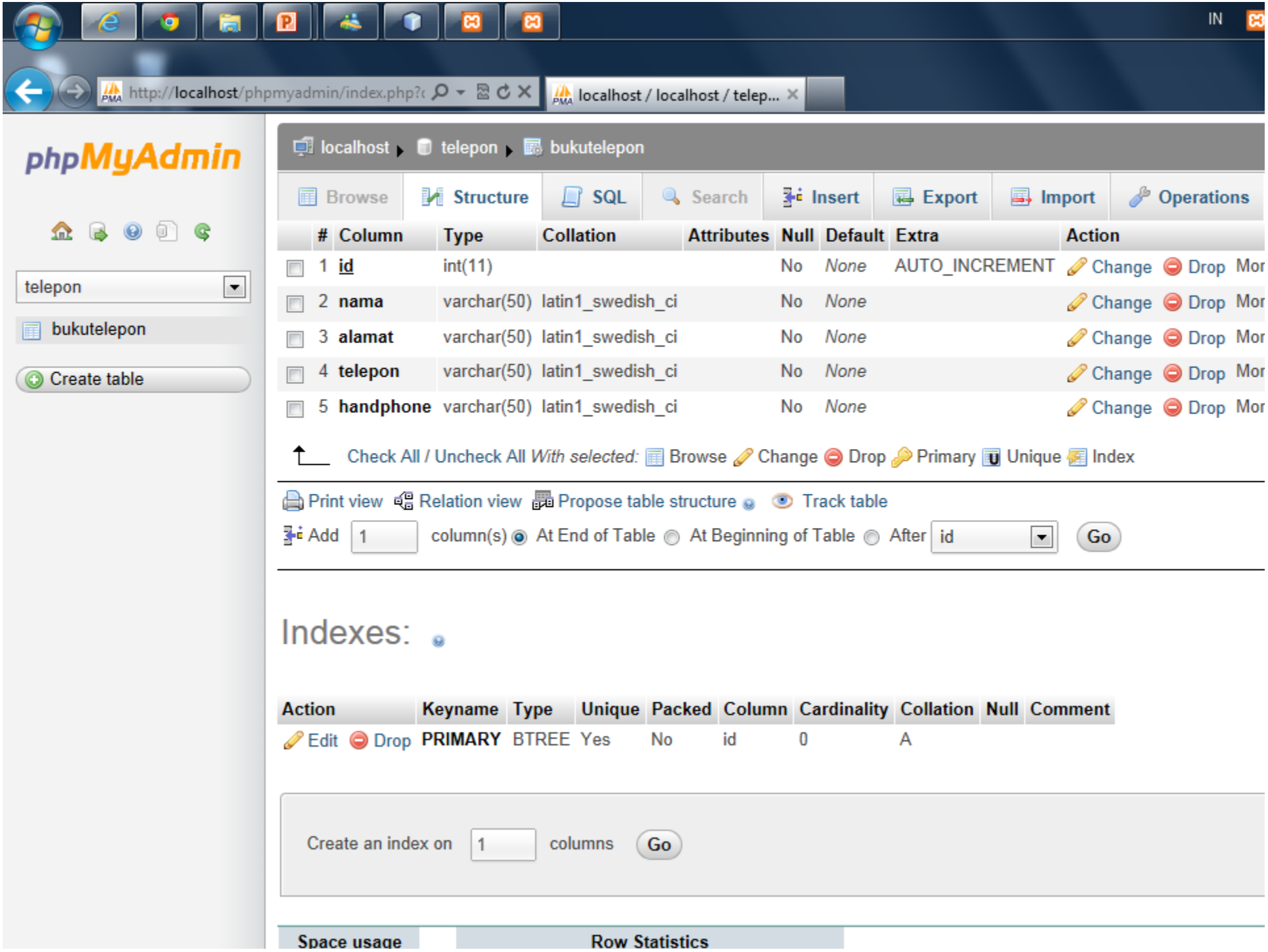
Aplikasi Telepon

1. Ekstrak xampplite dan jalankan xampp_start.exe untuk mengaktifkan Apache dan MySQL
2. Buka browser, arahkan url ke http://localhost dan klik link ke phpMyAdmin
3. Buat database telepon
4. Buat satu table bukutelepon, yang berisi field dengan id sebagai primary key (PK):
    1. id — integer (auto increment)
    2. nama — varchar(20)
    3. alamat — varchar(50)
    4. telepon — varchar(20)
    5. handphone — varchar(20)


http://localhost/phpmyadr
localhost / localhost | php...
phpMyAdmin
cdcol
information_schema
mysql
performance_schema
phpmyadmin
telepon
test
webauth
localhost
Databases
SQL
Status
Processes
Privileges
Export
Import
More
Databases
Create new database
telepon
Collation
Create
Database
cdcol Check Privileges
information_schema Check Privileges
mysql Check Privileges
performance_schema Check Privileges
phpmyadmin Check Privileges
telepon Check Privileges
test Check Privileges
webauth Check Privileges
Total: 8
Check All / Uncheck All With selected: Drop
Enable Statistics
Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.


http://localhost/phpmyadr
localhost / localhost / telep...
phpMyAdmin
localhost
telepon
Structure
SQL
Search
Query
Export
Import
Operations
More
No tables found in database
telepon
No tables found in database.
Create table
Create table on database telepon
Name:
bukutelepon
Number of columns:
5


phpMyAdmin
http://localhost/phpmyadmin/index.php?
localhost / localhost / telep...
localhost
telepon
telepon
No tables found in database.
Create table
No tables fou
Name:
Create Table
Table name:
bukutelepon
Column
Type
Length/Values[1]
id
INT
nama
VARCHAR
50
alamat
VARCHAR
50
telepon
VARCHAR
50
handphone
VARCHAR
50

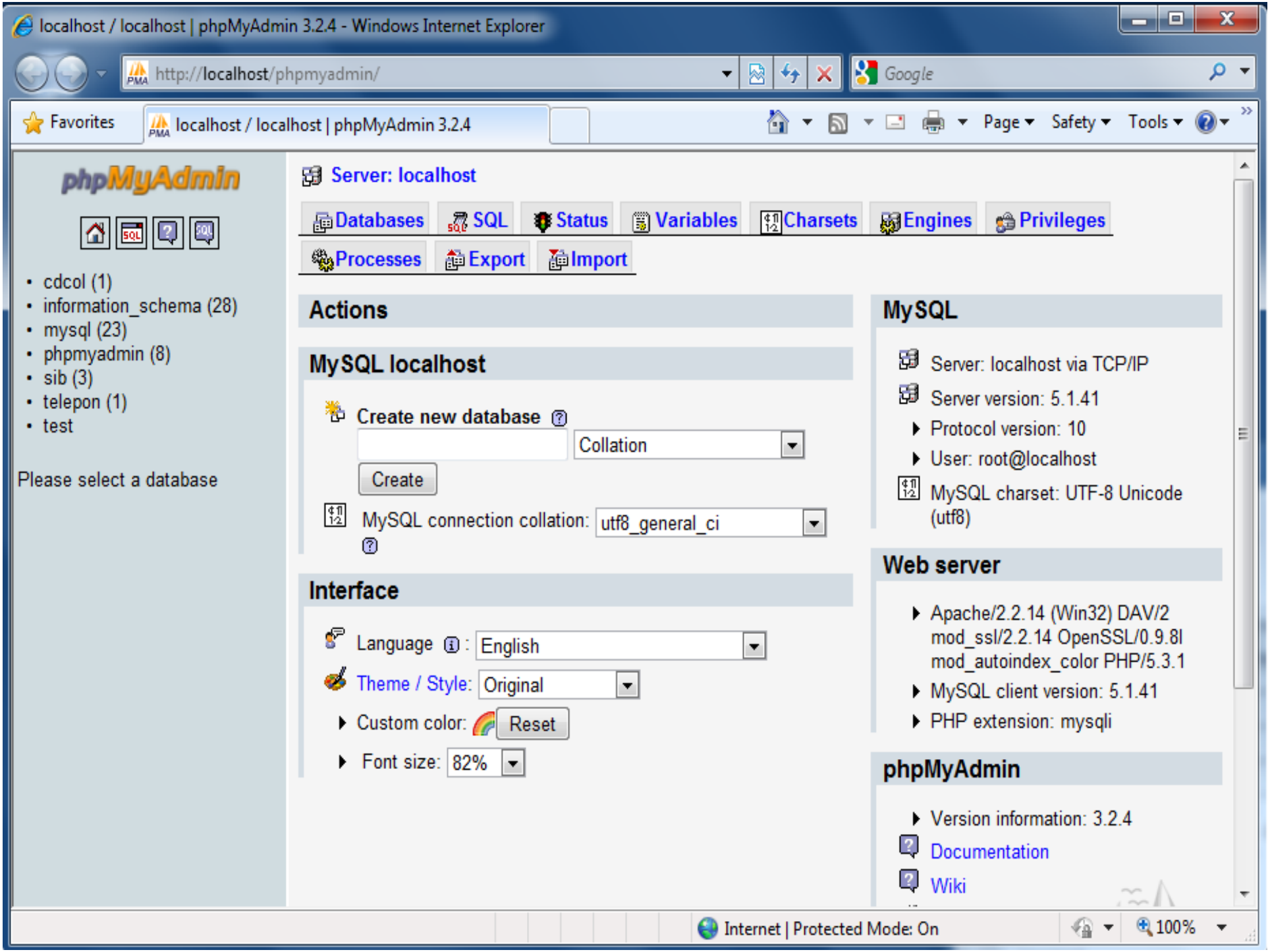

phpMyAdmin
localhost / localhost / telep...
localhost ▸ telepon ▸ bukutelepon
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
telepon
bukutelepon
Create table
Column Type Collation Attributes Null Default Extra Action
1 id int(11) No None AUTO_INCREMENT Change Drop Mor
2 nama varchar(50) latin1_swedish_ci No None Change Drop Mor
3 alamat varchar(50) latin1_swedish_ci No None Change Drop Mor
4 telepon varchar(50) latin1_swedish_ci No None Change Drop Mor
5 handphone varchar(50) latin1_swedish_ci No None Change Drop Mor
Check All / Uncheck All With selected: Browse Change Drop Primary Unique Index
Print view Relation view Propose table structure Track table
Add 1 column(s) At End of Table At Beginning of Table After id Go
Indexes:
Action Keyname Type Unique Packed Column Cardinality Collation Null Comment
Edit Drop PRIMARY BTREE Yes No id 0 A
Create an index on 1 columns Go
Space usage
Row Statistics

Autoincrement di PostgreSQL

```
CREATE SEQUENCE bukutelepon_id_seq;

ALTER TABLE bukutelepon
   ALTER COLUMN id
      SET DEFAULT NEXTVAL('bukutelepon_id_seq');

UPDATE bukutelepon
   SET id = NEXTVAL('bukutelepon_id_seq');
```

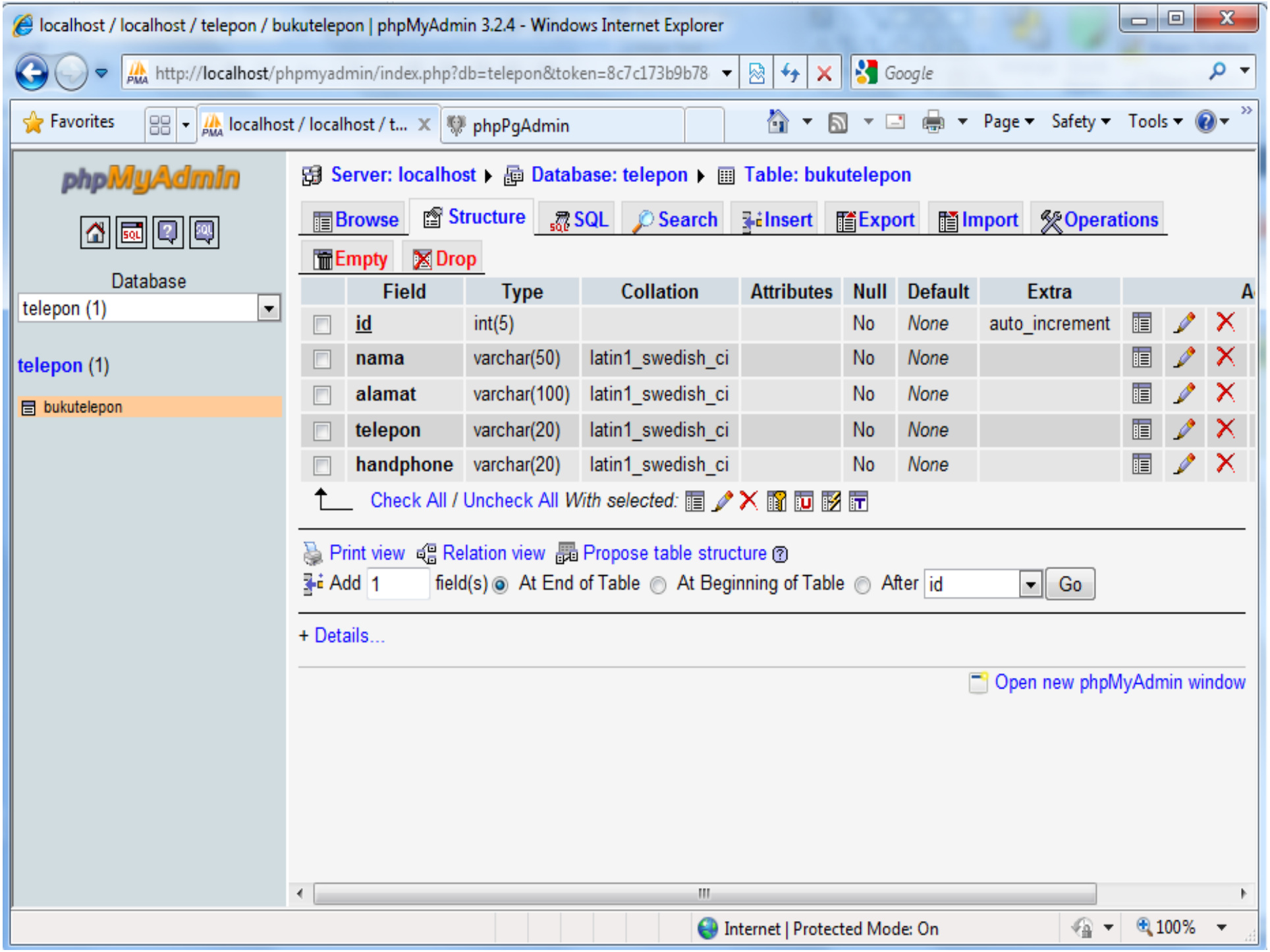

localhost / localhost | phpMyAdmin 3.2.4 - Windows Internet Explorer
http://localhost/phpmyadmin/
Google
Favorites
localhost / localhost | phpMyAdmin 3.2.4
Page
Safety
Tools
phpMyAdmin
cdcol (1)
information_schema (28)
mysql (23)
phpmyadmin (8)
sib (3)
telepon (1)
test
Please select a database
Server: localhost
Databases
SQL
Status
Variables
Charsets
Engines
Privileges
Processes
Export
Import
Actions
MySQL localhost
Create new database
Collation
Create
MySQL connection collation:
utf8_general_ci
Interface
Language :
English
Theme / Style:
Original
Custom color:
Reset
Font size:
82%
MySQL
Server: localhost via TCP/IP
Server version: 5.1.41
Protocol version: 10
User: root@localhost
MySQL charset: UTF-8 Unicode (utf8)
Web server
Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1
MySQL client version: 5.1.41
PHP extension: mysqli
phpMyAdmin
Version information: 3.2.4
Documentation
Wiki
Internet | Protected Mode: On
100%

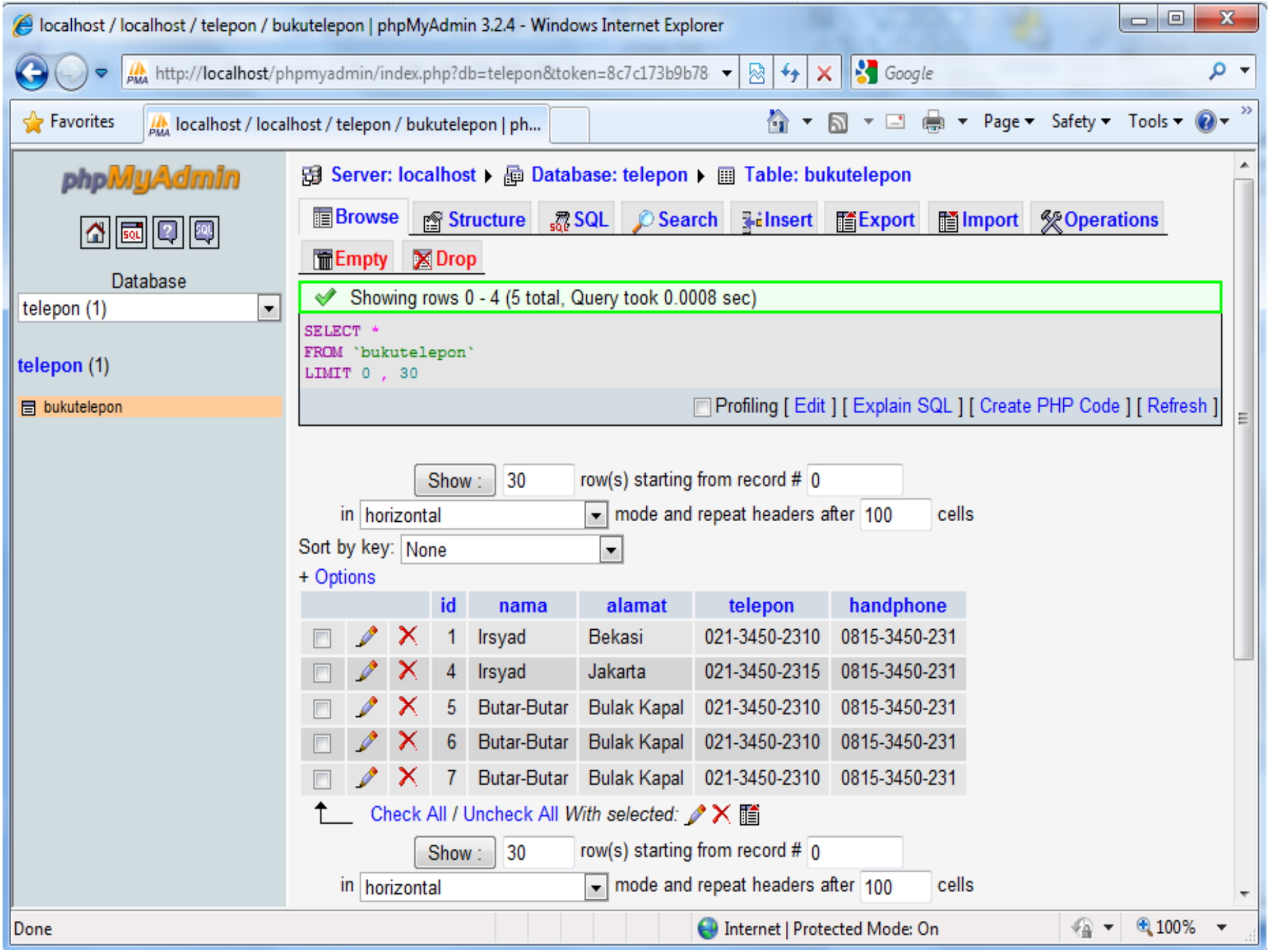
Server: localhost ▸ Database: telepo ▸ Table: bukutelepon

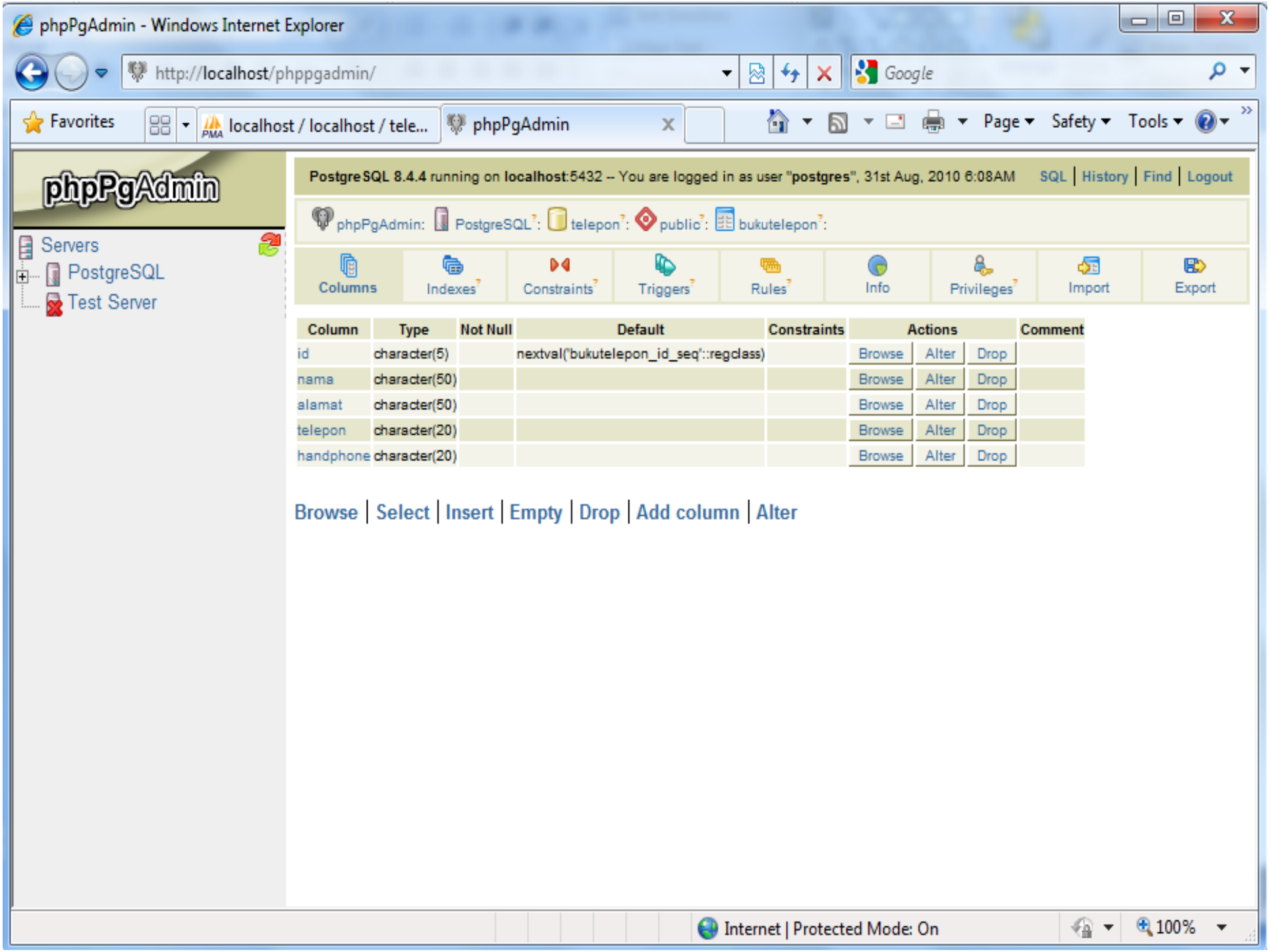
Field	Type	Length/Values[1]	Default[2]
id	INT		None
nama	VARCHAR	20	None
alamat	VARCHAR	50	None
telepon	VARCHAR	20	None
handphone	VARCHAR	20	None

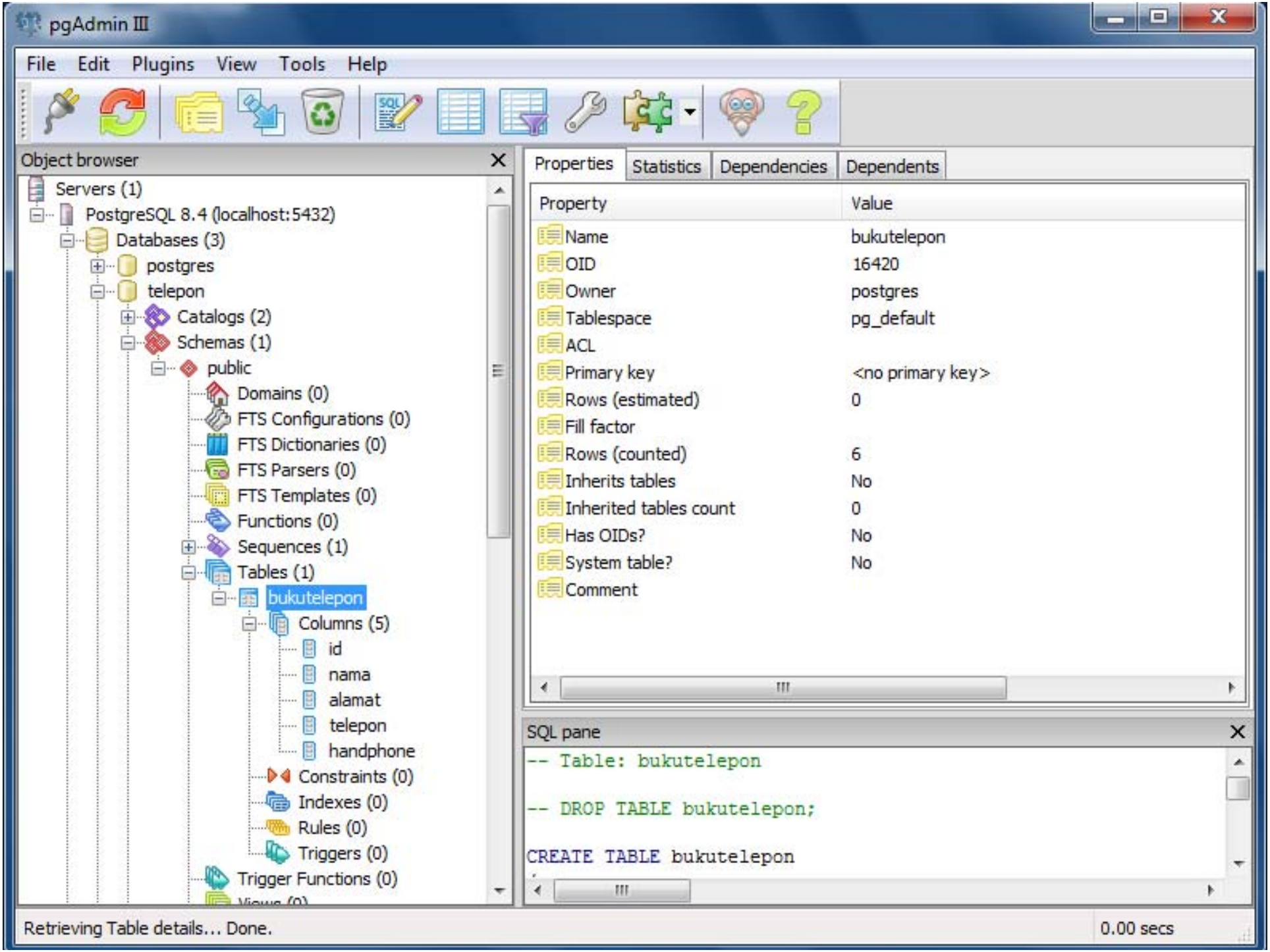
Table comments:	Storage Engine:	Collation:
	MyISAM	

PARTITION definition:


localhost / localhost / telepon / bukutelepon | phpMyAdmin 3.2.4 - Windows Internet Explorer
http://localhost/phpmyadmin/index.php?db=telepon&token=8c7c173b9b78
Google
Favorites
localhost / localhost / t...
phpPgAdmin
Page
Safety
Tools
phpMyAdmin
Database
telepon (1)
telepon (1)
bukutelepon
Server: localhost
Database: telepon
Table: bukutelepon
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Field
Type
Collation
Attributes
Null
Default
Extra
id
int(5)
No
None
auto_increment
nama
varchar(50)
latin1_swedish_ci
No
None
alamat
varchar(100)
latin1_swedish_ci
No
None
telepon
varchar(20)
latin1_swedish_ci
No
None
handphone
varchar(20)
latin1_swedish_ci
No
None
Check All / Uncheck All With selected:
Print view
Relation view
Propose table structure
Add 1 field(s) At End of Table At Beginning of Table After id Go
+ Details...
Open new phpMyAdmin window
Internet | Protected Mode: On
100%


localhost / localhost / telepon / bukutelepon | phpMyAdmin 3.2.4 - Windows Internet Explorer
http://localhost/phpmyadmin/index.php?db=telepon&token=8c7c173b9b78
Google
Favorites
localhost / localhost / telepon / bukutelepon | ph...
Page
Safety
Tools
phpMyAdmin
Database
telepon (1)
telepon (1)
bukutelepon
Server: localhost
Database: telepon
Table: bukutelepon
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Showing rows 0 - 4 (5 total, Query took 0.0008 sec)
SELECT *
FROM `bukutelepon`
LIMIT 0 , 30
Profiling [Edit] [Explain SQL] [Create PHP Code] [Refresh]
Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells
Sort by key: None
+ Options
id nama alamat telepon handphone
1 Irsyad Bekasi 021-3450-2310 0815-3450-231
4 Irsyad Jakarta 021-3450-2315 0815-3450-231
5 Butar-Butar Bulak Kapal 021-3450-2310 0815-3450-231
6 Butar-Butar Bulak Kapal 021-3450-2310 0815-3450-231
7 Butar-Butar Bulak Kapal 021-3450-2310 0815-3450-231
Check All / Uncheck All With selected:
Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells
Done
Internet | Protected Mode: On
100%


phpPgAdmin - Windows Internet Explorer
http://localhost/phppgadmin/
Google
Favorites
localhost / localhost / tele...
phpPgAdmin
Page
Safety
Tools
phpPgAdmin
Servers
PostgreSQL
Test Server
PostgreSQL 8.4.4 running on localhost:5432 -- You are logged in as user "postgres", 31st Aug, 2010 6:08AM
SQL | History | Find | Logout
phpPgAdmin: PostgreSQL: telepon: public: bukutelepon:
Columns
Indexes
Constraints
Triggers
Rules
Info
Privileges
Import
Export
Column
Type
Not Null
Default
Constraints
Actions
Comment
id
character(5)
nextval('bukutelepon_id_seq'::regclass)
nama
character(50)
alamat
character(50)
telepon
character(20)
handphone
character(20)
Browse
Alter
Drop
Browse | Select | Insert | Empty | Drop | Add column | Alter
Internet | Protected Mode: On
100%


pgAdmin III
File Edit Plugins View Tools Help
Object browser
Servers (1)
PostgreSQL 8.4 (localhost:5432)
Databases (3)
postgres
telepon
Catalogs (2)
Schemas (1)
public
Domains (0)
FTS Configurations (0)
FTS Dictionaries (0)
FTS Parsers (0)
FTS Templates (0)
Functions (0)
Sequences (1)
Tables (1)
bukutelepon
Columns (5)
id
nama
alamat
telepon
handphone
Constraints (0)
Indexes (0)
Rules (0)
Triggers (0)
Trigger Functions (0)
Properties
Statistics
Dependencies
Dependents
Property
Value
Name
bukutelepon
OID
16420
Owner
postgres
Tablespace
pg_default
ACL
Primary key
<no primary key>
Rows (estimated)
0
Fill factor
Rows (counted)
6
Inherits tables
No
Inherited tables count
0
Has OIDs?
No
System table?
No
Comment
SQL pane
-- Table: bukutelepon
-- DROP TABLE bukutelepon;
CREATE TABLE bukutelepon
Retrieving Table details... Done.
0.00 secs

Aplikasi Telepon

1. Extract dan copy folder 05 JAVA DATABASE di NetbeansProject anda
2. Di Netbeans buka file tersebut melalui Open project
3. Cek package db.mysql (versi text) dan dbgui.mysql (versi gui)
4. Program yang ada di kedua package tersebut akan mengakses dan melakukan query ke database telepon (table bukutelepon)

Cek Koneksi ke Database

```
String user="root";  String pswd ="";
String host="localhost";  String db="telepon";  String url="";
try {
    Class.forName("com.mysql.jdbc.Driver");
    url="jdbc:mysql://"+ host +"/"+  db +"?user=" + user +
"&password="+ pswd;
    Connection conn=DriverManager.getConnection(urlValue);
    System.out.println("koneksi sukses");
    conn.close();
} catch (SQLException e){
    System.out.println("koneksi gagal " + e.toString());
} catch(ClassNotFoundException e) {
    System.out.println("jdbc.Driver tidak ditemukan");
}
```

Cek Koneksi ke Database

```
String user="root";  String pswd ="";
String host="localhost";  String db="telepon";  String url="";
try {
    Class.forName("org.postgresql.Driver");
    url="jdbc:postgresql://"+ host +"/"+  db +"?user=" + user +
"&password="+ pswd;
    Connection conn=DriverManager.getConnection(urlValue);
    System.out.println("koneksi sukses");
    conn.close();
} catch (SQLException e){
    System.out.println("koneksi gagal " + e.toString());
} catch(ClassNotFoundException e) {
    System.out.println("jdbc.Driver tidak ditemukan");
}
```

Aplikasi Guru

Aplikasi Guru

1. Buat database Guru
2. Buat satu table dataguru, yang berisi field dengan nip sebagai primary key (PK). Field yang lain adalah seperti di bawah:

1.	nip	integer (auto increment)
2.	nama	varchar(30)
3.	status	varchar(20)
4.	institusi	varchar(30)
5.	kota	varchar(30)
6.	handphone	varchar(20)
7.	jeniskelamin	varchar(20)
8.	bidangstudi	varchar(30)

Tugas: Aplikasi Guru

3. Pahami program yang ada di package db.mysql
4. Buat 5 class java yang melakukan query ke database Guru:
   1. GuruConnection.java
   2. GuruInsert.java
   3. GuruRead.java
   4. GuruUpdate.java
   5. GuruDelete.java

Tugas: Aplikasi Guru

3. Pahami program yang ada di package dbgui.mysql
4. Buat 1 class MenuUtama dan 4 class java GUI yang melakukan query ke database Guru:
    1. GuruInsertUI.java
    2. GuruReadUI.java
    3. GuruUpdateUI.java
    4. GuruDeleteUI.java
    5. MenuUtama.java

Aplikasi Bank

Aplikasi Bank

1. Pahami dengan baik Case Study: A Bank Database yang terdapat pada buku Hortsmann (halaman 871)
2. Buat dua tabel database: BankCustomer dan Account
3. Buat dua class yang mendefinisikan dan mengoperasikan aplikasi Bank: Bank.java dan BankAccount.java
4. Buat satu class yang berisi method main yang mengeksekusi aplikasi bank

Aplikasi Penjualan Barang

Aplikasi Penjualan Barang (Quantum)

1. Ekstrak quantum.zip
2. Buat database sib di MySQL dan import sib.sql
3. Open project quantum
4. Lakukan pengecekan dan perbaikan error yang ada (klik kanan di project dan pilih Resolve Reference Problem)
5. Build dan jalankan program
6. Pelajari dengan baik source codenya

Tugas

- Kembangkan aplikasi java berbasis GUI yang mengakses database MySQL. Fitur utama dari aplikasi adalah kemampuan untuk CRUD (create, read (listing), update, delete) data dari database MySQL dan fitur transaksi serta reporting. Gunakan lebih dari satu table
- Pilih aplikasi dari list di bawah (digit terakhir NIM):
  1. Aplikasi Online Penjualan Buku
  2. Aplikasi Online Penjualan Handphone
  3. Aplikasi Online Pengelolaan KRS
  4. Aplikasi Online Penjualan Tiket Pesawat
  5. Aplikasi Online Penjualan Tiket Kereta
  6. Aplikasi Sirkulasi Perpustakaan
  7. Aplikasi Rental Mobil
  8. Aplikasi Penjualan Handphone
  9. Aplikasi Penjualan CD Musik
  0. Aplikasi Sewa PC
- Kirimkan file-file di bawah ke rahmatfauzi9013@gmail.com subject email SI4108-NAMAMAHASISWA-DATABASEJAVE
  - Source project netbeans dari aplikasi yang dibuat
  - Ekspor (dumped) database MySQL (*.sql)
- Deadline: 1 Minggu