

LAPORAN PROJECT

Java Database Connectivity (JDBC)

Dibuat untuk memenuhi salah satu syarat tugas Ujian Akhir Semester mata kuliah Pemrograman Berorientasi Object

Dosen Pengampu:

Shinta Ayuningtias, M.Kom



Disusun Oleh:

Yuni Lestari (20220040006)

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK, KOMPUTER DAN DESAIN
UNIVERSITAS NUSA PUTRA
SUKABUMI**

2024

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan project yang berjudul "Java Database Connectivity (JDBC)" ini. Laporan Project ini disusun sebagai salah satu tugas dalam rangkaian Ujian Akhir Semester mata kuliah Pemrograman Berorientasi Object.

Perkembangan teknologi informasi yang sangat pesat telah membawa perubahan signifikan dalam berbagai aspek kehidupan manusia. Salah satu komponen penting dalam teknologi informasi adalah kemampuan untuk mengakses dan mengelola data secara efisien dan efektif. Di sinilah peran Java Database Connectivity (JDBC) menjadi sangat vital. JDBC merupakan sebuah API (Application Programming Interface) yang disediakan oleh Java untuk memungkinkan aplikasi Java berinteraksi dengan basis data. Dengan menggunakan JDBC, pengembang dapat melakukan berbagai operasi basis data seperti menghubungkan, mengambil, menyisipkan, memperbarui, dan menghapus data dari berbagai jenis basis data yang ada.

Laporan Project ini bertujuan untuk memberikan penjelasan secara keseluruhan mengenai tugas ataupun project yang penulis buat untuk memenuhi salah satu syarat tugas Ujian Akhir Semester mata kuliah Pemrograman Berorientasi Object.

Penulis menyadari bahwa dalam penyusunan laporan ini masih terdapat kekurangan dan keterbatasan. Oleh karena itu, penulis sangat mengharapkan saran dan kritik yang membangun dari para pembaca demi perbaikan dan penyempurnaan laporan ini di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan kontribusinya dalam penyelesaian laporan ini.

Sukabumi, 25 Juni 2024

Penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, pengelolaan data menjadi salah satu aspek kritis dalam dunia teknologi informasi. Basis data (database) berperan penting sebagai tempat penyimpanan informasi yang digunakan dalam berbagai aplikasi. Dalam konteks pengembangan aplikasi berbasis Java, Java Database Connectivity (JDBC) memegang peranan vital sebagai teknologi untuk menghubungkan aplikasi Java dengan berbagai jenis basis data.

Perkembangan aplikasi Java yang semakin kompleks dan kebutuhan akan integrasi dengan basis data yang lebih fleksibel mendorong pengembang untuk memahami dan menggunakan JDBC secara efektif. Melalui JDBC, aplikasi Java dapat melakukan berbagai operasi terhadap basis data seperti pengambilan data, penyisipan, pembaruan, dan penghapusan, sehingga memungkinkan aplikasi untuk mengelola informasi dengan lebih efisien.

1.2 Tujuan

Laporan Project ini bertujuan untuk memberikan pemahaman yang mendalam mengenai Java Database Connectivity (JDBC) dalam konteks pengembangan aplikasi Java. Adapun tujuan khusus dari Laporan ini adalah sebagai berikut:

1. Memahami Konsep Dasar JDBC:

Menjelaskan konsep dasar tentang JDBC, termasuk cara kerja dan pentingnya dalam pengembangan aplikasi berbasis Java.

2. Menjelaskan Arsitektur JDBC:

Menguraikan komponen-komponen utama dalam arsitektur JDBC, seperti driver JDBC, API JDBC, dan JDBC URL.

3. Menggambarkan Penggunaan JDBC dalam Praktek:

Memberikan contoh penerapan JDBC dalam situasi nyata untuk berbagai operasi basis data seperti pengambilan data, penyisipan, pembaruan, dan penghapusan.

4. Mendorong Pengembangan Keterampilan Praktis:

Memberikan pembaca pemahaman yang cukup untuk dapat mengimplementasikan JDBC dalam proyek-proyek pengembangan perangkat lunak dengan cara yang efisien dan efektif.

BAB II

PEMBAHASAN

2.1 Konsep Dasar JDBC

Java Database Connectivity (JDBC) adalah sebuah API (Application Programming Interface) yang disediakan oleh Java untuk memungkinkan aplikasi Java berinteraksi dengan basis data secara konsisten dan platform-independent. JDBC memberikan kemampuan bagi pengembang untuk melakukan berbagai operasi terhadap basis data seperti pengambilan data, penyisipan, pembaruan, dan penghapusan data.

- **Cara Kerja JDBC:**

- **Driver JDBC:** JDBC menggunakan konsep driver untuk menghubungkan aplikasi Java dengan berbagai jenis basis data. Driver ini bertindak sebagai perantara antara aplikasi Java dan sistem basis data yang sebenarnya.
- **JDBC API:** JDBC menyediakan sekumpulan kelas dan antarmuka yang memungkinkan aplikasi Java untuk mengirim perintah SQL (Structured Query Language) ke basis data dan mengelola hasilnya.
- **JDBC URL:** Untuk membuat koneksi ke basis data, JDBC menggunakan JDBC URL yang berisi informasi tentang jenis driver, lokasi basis data, dan parameter koneksi lainnya.

- **Pentingnya JDBC dalam Pengembangan Aplikasi Java:**

- **Konsistensi Platform:** JDBC memungkinkan aplikasi Java untuk terhubung dengan berbagai jenis basis data seperti MySQL, PostgreSQL, Oracle, dan lain-lain tanpa perlu mengubah kode aplikasi secara drastis.
- **Fleksibilitas:** Dengan JDBC, pengembang dapat menggunakan bahasa SQL standar untuk berinteraksi dengan basis data, memungkinkan implementasi operasi basis data yang konsisten di berbagai platform.
- **Integrasi dengan Java:** JDBC terintegrasi secara native dengan platform Java, sehingga memudahkan pengembang Java untuk mengelola data dalam aplikasi mereka tanpa harus mengandalkan teknologi atau bahasa pemrograman tambahan.
- **Skalabilitas dan Performa:** Dengan JDBC, pengembang dapat mengoptimalkan kinerja aplikasi dengan melakukan koneksi dan operasi basis data secara efisien, serta mengelola transaksi dengan baik.

Dengan demikian, JDBC menjadi komponen integral dalam pengembangan aplikasi berbasis Java yang membutuhkan akses ke basis data. Memahami dan menguasai JDBC adalah kunci untuk membangun aplikasi yang handal, efisien, dan dapat diandalkan dalam mengelola informasi dan data.

2.2 Arsitektur JDBC

Dalam arsitektur JDBC, terdapat beberapa komponen utama yang penting untuk dipahami, yaitu driver JDBC, API JDBC, dan JDBC URL. Mari kita uraikan masing-masing komponen tersebut:

1. Driver JDBC:

- **Definisi:** Driver JDBC adalah komponen yang memungkinkan aplikasi Java untuk berkomunikasi dengan berbagai jenis basis data. Setiap jenis basis data memiliki driver JDBC yang spesifik.
- **Tipe Driver:** Terdapat empat jenis driver JDBC:
 - **JDBC-ODBC Bridge Driver:** Menghubungkan aplikasi Java ke basis data menggunakan ODBC (Open Database Connectivity), tetapi kini sudah tidak disarankan karena ketergantungan pada ODBC.
 - **Native-API Driver (JDBC-Native Driver):** Memanfaatkan API spesifik dari basis data tertentu, biasanya ditulis dalam bahasa C atau C++. Performa driver ini sangat bergantung pada basis data yang digunakan.
 - **Network Protocol Driver:** Berkomunikasi dengan basis data melalui protokol jaringan spesifik. Biasanya terdapat subkategori seperti thin client driver (tanpa memerlukan instalasi tambahan pada klien) dan fat client driver (memerlukan instalasi tambahan pada klien).
 - **Thin Driver (JDBC Type 4):** Merupakan driver yang paling disukai saat ini karena tidak memerlukan instalasi tambahan pada klien dan langsung berkomunikasi melalui protokol basis data yang didukung.
- **Penggunaan:** Pengembang memilih driver JDBC yang sesuai dengan jenis basis data yang digunakan dalam aplikasi mereka untuk memfasilitasi koneksi dan operasi basis data.

2. API JDBC:

- **Definisi:** API JDBC terdiri dari serangkaian kelas dan antarmuka yang disediakan oleh Java untuk memungkinkan aplikasi Java mengakses dan mengelola basis data menggunakan bahasa SQL.
- **Kelas-kelas Utama:**
 - **Connection:** Membentuk koneksi fisik antara aplikasi Java dan basis data.
 - **Statement:** Mengirim perintah SQL ke basis data dan menerima hasilnya.
 - **ResultSet:** Menyimpan hasil dari query SQL yang dieksekusi pada basis data.
 - **PreparedStatement dan CallableStatement:** Jenis-jenis Statement yang dioptimalkan untuk pengiriman query berulang (PreparedStatement) atau untuk memanggil prosedur tersimpan di basis data (CallableStatement).

- **Penggunaan:** Pengembang menggunakan kelas-kelas dan antarmuka dari JDBC API untuk membangun aplikasi Java yang dapat berinteraksi dengan basis data, termasuk mengirim query, mengelola transaksi, dan memanipulasi data.

3. JDBC URL:

- **Definisi:** JDBC URL (Uniform Resource Locator) adalah string yang digunakan untuk menentukan lokasi dan parameter koneksi ke basis data.
- **Struktur Umum:** JDBC URL biasanya memiliki format umum yang mencakup informasi berikut:
 - **JDBC Protocol:** Menentukan protokol yang digunakan untuk koneksi (misalnya, jdbc:mysql:// untuk MySQL atau jdbc:oracle:thin:@ untuk Oracle).
 - **Host dan Port:** Menunjukkan lokasi basis data yang diakses, termasuk alamat host dan nomor port jika diperlukan.
 - **Database Name:** Nama basis data yang akan diakses.
 - **Parameter Koneksi Tambahan:** Opsional, seperti username, password, dan parameter koneksi lainnya.
- **Contoh:** jdbc:mysql://localhost:3306/mydatabase

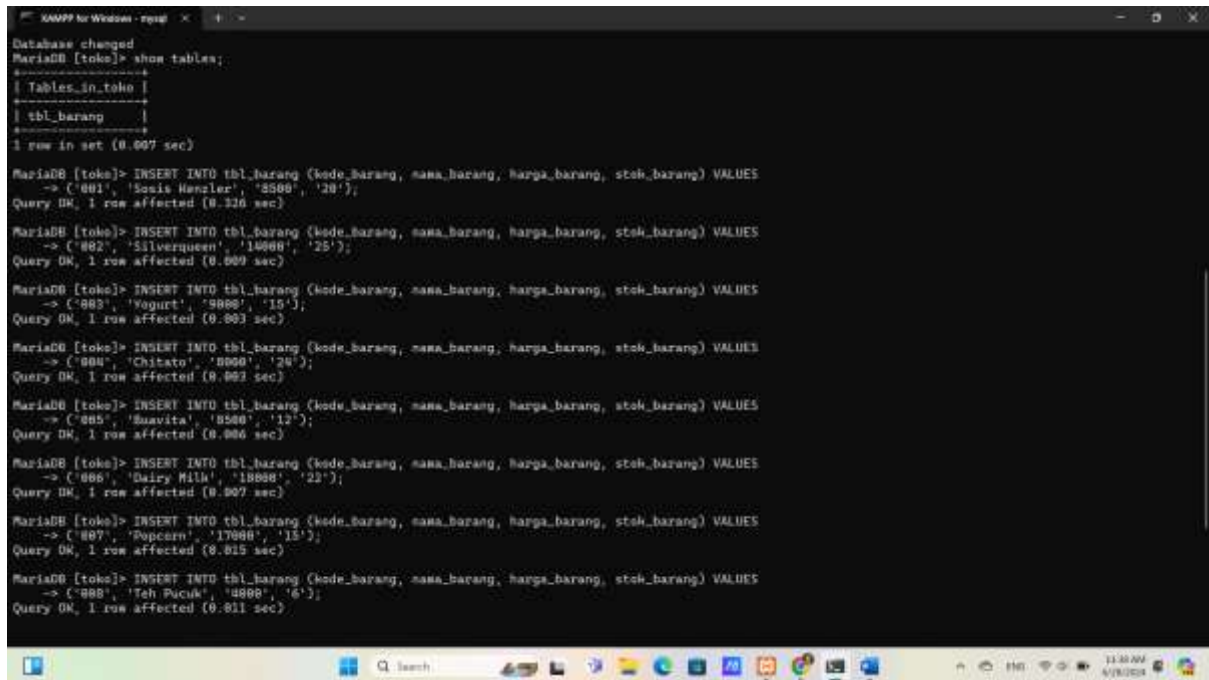
Dengan memahami komponen-komponen utama dalam arsitektur JDBC, pengembang dapat secara efektif mengelola koneksi dan operasi basis data dalam aplikasi Java mereka dengan memilih driver yang sesuai, menggunakan API JDBC untuk interaksi dengan basis data, dan menentukan JDBC URL untuk membuat koneksi yang diperlukan.

BAB III

CARA KERJA

3.1 Membuat Database dan table

Dengan nama database “toko”, dan nama table “tbl_barang”



```
Database changed
MariaDB [toko]> show tables;
+-----+
| Tables_in_toko |
+-----+
| tbl_barang      |
+-----+
1 row in set (0.007 sec)

MariaDB [toko]> INSERT INTO tbl_barang (kode_barang, nama_barang, harga_barang, stok_barang) VALUES
-> ('001', 'Sosis Kenzler', '8500', '20');
Query OK, 1 row affected (0.326 sec)

MariaDB [toko]> INSERT INTO tbl_barang (kode_barang, nama_barang, harga_barang, stok_barang) VALUES
-> ('002', 'Silverqueen', '14000', '25');
Query OK, 1 row affected (0.009 sec)

MariaDB [toko]> INSERT INTO tbl_barang (kode_barang, nama_barang, harga_barang, stok_barang) VALUES
-> ('003', 'Yogurt', '9000', '15');
Query OK, 1 row affected (0.003 sec)

MariaDB [toko]> INSERT INTO tbl_barang (kode_barang, nama_barang, harga_barang, stok_barang) VALUES
-> ('004', 'Chitato', '8000', '24');
Query OK, 1 row affected (0.002 sec)

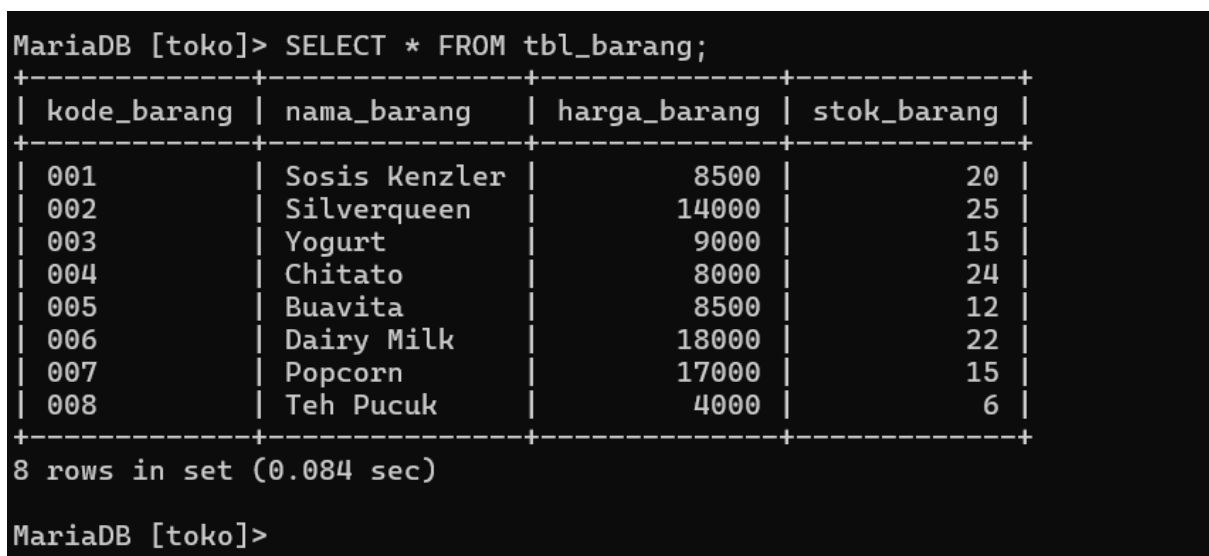
MariaDB [toko]> INSERT INTO tbl_barang (kode_barang, nama_barang, harga_barang, stok_barang) VALUES
-> ('005', 'Buavita', '8500', '12');
Query OK, 1 row affected (0.006 sec)

MariaDB [toko]> INSERT INTO tbl_barang (kode_barang, nama_barang, harga_barang, stok_barang) VALUES
-> ('006', 'Dairy Milk', '18000', '22');
Query OK, 1 row affected (0.007 sec)

MariaDB [toko]> INSERT INTO tbl_barang (kode_barang, nama_barang, harga_barang, stok_barang) VALUES
-> ('007', 'Popcorn', '17000', '15');
Query OK, 1 row affected (0.015 sec)

MariaDB [toko]> INSERT INTO tbl_barang (kode_barang, nama_barang, harga_barang, stok_barang) VALUES
-> ('008', 'Teh Pucuk', '4000', '6');
Query OK, 1 row affected (0.011 sec)
```

3.2 Menambahkan Value pada table “tbl_barang”



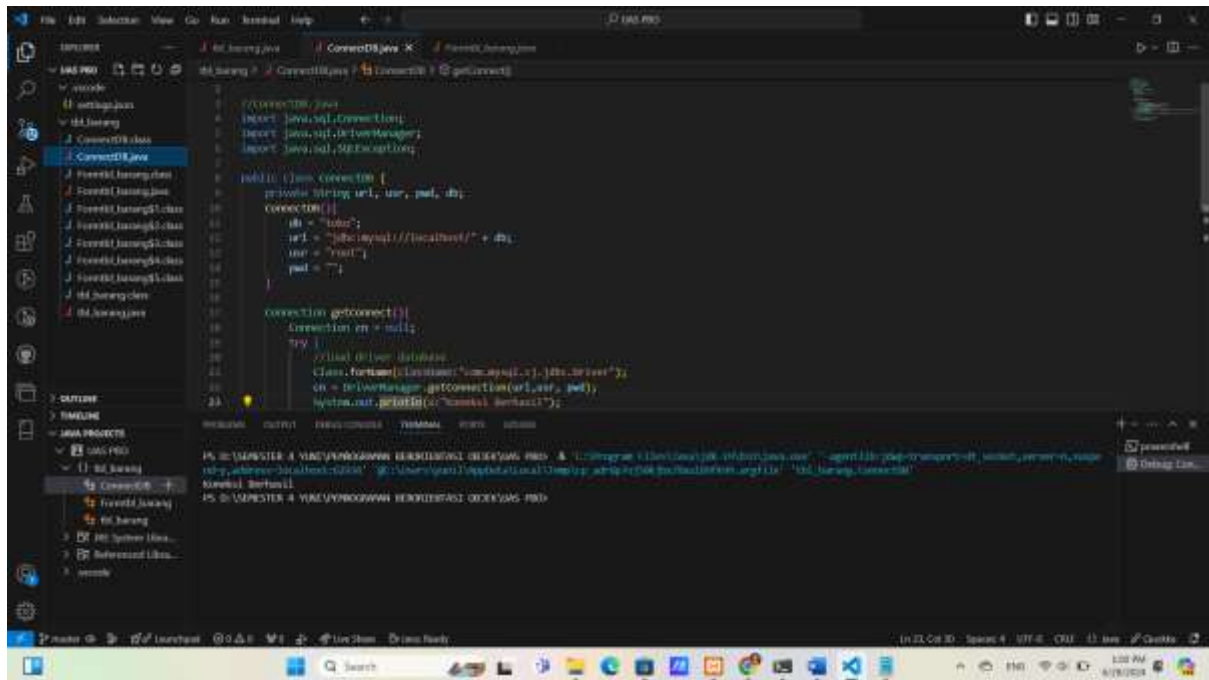
```
MariaDB [toko]> SELECT * FROM tbl_barang;
+-----+-----+-----+-----+
| kode_barang | nama_barang | harga_barang | stok_barang |
+-----+-----+-----+-----+
| 001         | Sosis Kenzler | 8500         | 20          |
| 002         | Silverqueen  | 14000        | 25          |
| 003         | Yogurt       | 9000         | 15          |
| 004         | Chitato      | 8000         | 24          |
| 005         | Buavita      | 8500         | 12          |
| 006         | Dairy Milk   | 18000        | 22          |
| 007         | Popcorn      | 17000        | 15          |
| 008         | Teh Pucuk    | 4000         | 6           |
+-----+-----+-----+-----+
8 rows in set (0.084 sec)

MariaDB [toko]>
```

3.3 Membuat New Folder/Project Java di dalam src (didalam folder tbl_barang)

- ConnectDB.java
- Formtbl_barang.java
- tbl_barang.java


Menambahkan Connector untuk menghubungkan ConnectDB



BAB IV

HASIL

4.1 Output dari File Formtbl_barang.java

 Data Barang

Kode Barang

Nama Barang

Harga Barang

Stok Barang

Baru


Simpan

Ubah

Hapus

Kode Barang	Nama Barang	Harga Barang	Stok Barang
001	Sosis Kenzler	8500	20
002	Silverqueen	14000	25
003	Yogurt	9000	15
004	Chitato	8000	24
005	Buavita	8500	12
006	Dairy Milk	18000	22
007	Popcorn	17000	15
008	Teh Pucuk	4000	6

4.2 Menambahkan Data Baru

 Data Barang

Kode Barang

Nama Barang

Harga Barang

Stok Barang

Baru

Simpan

Ubah

Hapus

Kode Barang	Nama Barang	Harga Barang	Stok Barang
001	Sosis Kenzler	8500	20
002	Silverqueen	14000	25
003	Yogurt	9000	15
004	Chitato	8000	24
005	Buavita	8500	12
006	Dairy Milk	18000	22
007	Popcorn	17000	15
008	Teh Pucuk	4000	6

Data Barang

Kode Barang: 009
 Nama Barang: Teh Celup
 Harga Barang: 5000
 Stok Barang: 13

Baru
 Simpan
 Ubah
 Hapus

Kode Barang	Nama Barang	Harga Barang	Stok Barang
001	Sosis Kenzler	8500	20
002	Silverqueen	14000	25
003	Yogurt	9000	15
004	Chitato	8000	24
005	Buavita	8500	12
006	Dairy Milk	18000	22
007	Popcorn	17000	15
008	Teh Pucuk	4000	6

Info Proses
 Data Berhasil Disimpan
 OK

008	Teh Pucuk	4000	6
009	Teh Celup	5000	13

4.3 Mengubah Data

Data Barang

Kode Barang: 009
 Nama Barang: Teh Sosro
 Harga Barang: 7000
 Stok Barang: 12

Baru
 Simpan
 Ubah
 Hapus

Kode Barang	Nama Barang	Harga Barang	Stok Barang
002	Silverqueen	14000	25
003	Yogurt	9000	15
004	Chitato	8000	24
005	Buavita	8500	12
006	Dairy Milk	18000	22
007	Popcorn	17000	15
008	Teh Pucuk	4000	6
009	Teh Celup	5000	13

Data Barang

Kode Barang: 009
 Nama Barang: Teh Sosro
 Harga Barang: 7000
 Stok Barang: 12

Baru
 Simpan
 Ubah
 Hapus

Kode Barang	Nama Barang	Harga Barang	Stok Barang
002	Silverqueen	14000	25
003	Yogurt	9000	15
004	Chitato	8000	24
005	Buavita	8500	12
006	Dairy Milk	18000	22
007	Popcorn	17000	15
008	Teh Pucuk	4000	6
009	Teh Celup	5000	13

Info Proses
 Data Berhasil Diubah
 OK

008	Teh Pucuk	4000	6
009	Teh Sosro	7000	12

4.4 Menghapus Data

Data Barang

Kode Barang: 001
 Nama Barang: Sosis Kenzler
 Harga Barang: 8500
 Stok Barang: 20

Baru
 Simpan
 Ubah
 Hapus

Kode Barang	Nama Barang	Harga Barang	Stok Barang
001	Sosis Kenzler	8500	20
002	Silverqueen	14000	25
003	Yogurt	9000	15
004	Chitato	8000	24
005	Buavita	8500	12
006	Dairy Milk	18000	22
007	Popcorn	17000	15
008	Teh Pucuk	4000	6

Data Barang

Kode Barang

001

Baru

Nama Barang

Sosis KENZLER

Simpan

Harga Barang

8500

Ubah

Stok Barang

20

Hapus

Kode Barang	Nama Barang	Harga Barang	Stok Barang
001	Sosis KENZLER	8500	20
002	Silverqueen	14000	25
003	Yogurt	9000	15
004	Chitato	8000	24
005	Buavita	8500	12
006	Dairy Milk	18000	22
007	Popcorn	17000	15
008	Teh Pucuk	4000	6

Konfirmasi

Yakin data akan dihapus?

OK

Cancel

Data Barang

Kode Barang

Baru

Nama Barang

Simpan

Harga Barang

Ubah

Stok Barang

Hapus

Kode Barang	Nama Barang	Harga Barang	Stok Barang
002	Silverqueen	14000	25
003	Yogurt	9000	15
004	Chitato	8000	24
005	Buavita	8500	12
006	Dairy Milk	18000	22
007	Popcorn	17000	15
008	Teh Pucuk	4000	6
009	Teh Sosro	7000	12

BAB V

PENUTUP

5.1 Kesimpulan

Java Database Connectivity (JDBC) adalah teknologi kunci dalam pengembangan aplikasi Java yang memungkinkan koneksi dan interaksi dengan berbagai jenis basis data. Dengan memanfaatkan driver JDBC, API JDBC, dan JDBC URL, pengembang dapat mengelola data secara efisien dan konsisten di seluruh platform, menjadikan JDBC fondasi yang vital dalam integrasi aplikasi dengan sistem basis data.

5.2 Saran

Penulis menyadari bahwa penulis masih sangat jauh sekali dari kata-kata sempurna, untuk kedepannya penulis akan lebih jelas dan lebih fokus lagi dalam menerangkan penjelasan mengenai laporan diatas dengan sumber sumber yang lebih lengkap dan lebih banyak lagi, dan tentunya bisa untuk dipertanggung jawabkan. Untuk saran yang akan ibu berikan kepada penulis, bisa berupa kritikan-kritikan dan saran-saran penulis guna untuk menyimpulkan kepada kesimpulan dari pembahasan laporan yang sudah dijelaskan di dalam laporan.