**LAPORAN TUGAS**

**Pemrograman Berbasis Java**

**Database Wilayah Indonesia Berdasarkan Master File Desa Badan Pusat Statistik (MFD BPS) Indonesia**



**Disusun oleh:**

Dwi Putra Sudaryanto

12111075

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS MERCUBUANA YOGYAKARTA**

# **DAFTAR ISI**

[**DAFTAR ISI** 2](#_Toc438075765)

[**BAB I PENDAHULUAN** 4](#_Toc438075766)

[**1.1.** **Latar Belakang** 4](#_Toc438075767)

[**1.2.** **Alasan Pemilihan Masalah** 4](#_Toc438075768)

[**BAB II DATABASE** 5](#_Toc438075769)

[**2.1.** **Rancangan Tabel** 5](#_Toc438075770)

[2.1.1. Tabel 5](#_Toc438075771)

[2.1.2. Trigger 5](#_Toc438075772)

[2.1.3. Store Prosedure 7](#_Toc438075773)

[2.1.4. View 8](#_Toc438075774)

[**2.2.** **Relasi Database** 8](#_Toc438075775)

[**BAB III JDBC** 10](#_Toc438075776)

[**3.1.** **Provinsi** 10](#_Toc438075777)

[3.1.1. Insert 10](#_Toc438075778)

[3.1.2. Search 10](#_Toc438075779)

[3.1.3. Update 11](#_Toc438075780)

[3.1.4. Delete 11](#_Toc438075781)

[3.1.5. View 12](#_Toc438075782)

[**3.2.** **Kabupaten / Kota** 12](#_Toc438075783)

[3.2.1. Insert 12](#_Toc438075784)

[3.2.2. Search 13](#_Toc438075785)

[3.2.3. Update 13](#_Toc438075786)

[3.2.4. Delete 14](#_Toc438075787)

[3.2.5. View 14](#_Toc438075788)

[**3.3.** **Kecamatan** 15](#_Toc438075789)

[3.3.1. Insert 15](#_Toc438075790)

[3.3.2. Search 15](#_Toc438075791)

[3.3.3. Update 16](#_Toc438075792)

[3.3.4. Delete 16](#_Toc438075793)

[3.3.5. View 17](#_Toc438075794)

[**3.4.** **Desa / Kelurahan** 17](#_Toc438075795)

[3.4.1. Insert 17](#_Toc438075796)

[3.4.2. Search 18](#_Toc438075797)

[3.4.3. Update 18](#_Toc438075798)

[3.4.4. Delete 19](#_Toc438075799)

[3.4.5. View 19](#_Toc438075800)

[**BAB IV MENU** 21](#_Toc438075801)

[**4.1.** **Menu** 21](#_Toc438075802)

[**4.2.** **Integrasi Menu** 22](#_Toc438075803)

[4.2.1. Installer 22](#_Toc438075804)

[4.2.2. Provinsi 22](#_Toc438075805)

[4.2.3. Kebupaten, Kota 22](#_Toc438075806)

[4.2.4. Kecamatan 23](#_Toc438075807)

[4.2.5. Kelurahan, Desa 24](#_Toc438075808)

[4.2.6. About 25](#_Toc438075809)

[**BAB V PENUTUP** 27](#_Toc438075810)

[**5.1.** **Kesimpulan** 27](#_Toc438075811)

[**5.2.** **Kesulitan** 27](#_Toc438075812)

# **BAB I PENDAHULUAN**

* 1. **Latar Belakang**

Dokumen ini merupakan diskripsi dari aplikasi Database Wilayah Indonesia Berdasarkan Master File Desa Badan Pusat Statistik (MFD BPS) Indonesia yang dikembangkan menggunakan bahasa pemograman berbasis java. Dokumen ini dibuat untuk memenuhi tugas mata kuliah Pemograman Berbasis Java.

* 1. **Alasan Pemilihan Masalah**

Master File Desa Badan Pusat Statistik (MFD BPS) Indonesia merupakan acuan dasar yang digunakan oleh pengembang aplikasi baik website, mobile maupun desktop jika ingin mengembangkan sebuah aplikasi yang memiliki basis data daerah Indonesia.

Aplikasi ini dikembangkan untuk mempermudah siapapun baik personal maupun organisasi untuk mengelola database wilayah Indonesia menggunakan konsep MFD online untuk selanjutnya dapat dikembangkan kembali.

# **BAB II DATABASE**

1. **Rancangan Tabel**
2. Tabel

|  |  |
| --- | --- |
| **Tebel** | **Colom** |
| ommu\_core\_zone\_province | `province\_id` smallint(5), `publish` tinyint(1), `province` varchar(64), `mfdonline` char(2), `creation\_date`, `modified\_date` |
| ommu\_core\_zone\_city | `city\_id` int(11), `publish` tinyint(1), `province\_id` smallint(5), `city` varchar(64) , `mfdonline` char(4), `creation\_date`, `modified\_date` |
| ommu\_core\_zone\_districts | `district\_id` int(11), `publish` tinyint(1), `city\_id` int(11), `district\_name` varchar(64), `mfdonline` char(7), `creation\_date`, `modified\_date` |
| ommu\_core\_zone\_village | `village\_id`, `publish`, `district\_id`, `village\_name`, `zipcode`, `mfdonline`, `creation\_date`, `modified\_date` |

1. Trigger

|  |  |
| --- | --- |
| **Name** | **Deskripsi** |
| coreBeforeInsertCity | DECLARE province\_id\_tr SMALLINT;    /\*CALL getOmmuZoneProvinceIdWithCityMfdonline(NEW.mfdonline, province\_id\_tr);\*/  SELECT `province\_id` INTO province\_id\_tr FROM `ommu\_core\_zone\_province` WHERE `mfdonline`=LEFT(NEW.mfdonline,2);  IF (province\_id\_tr IS NOT NULL) THEN  SET NEW.province\_id = province\_id\_tr;  END IF;    SET NEW.city = TRIM(NEW.city); |
| coreBeforeUpdateCity | DECLARE province\_id\_tr SMALLINT;    IF (NEW.mfdonline <> OLD.mfdonline) THEN  /\*CALL getOmmuZoneProvinceIdWithCityMfdonline(NEW.mfdonline, province\_id\_tr);\*/  SELECT `province\_id` INTO province\_id\_tr FROM `ommu\_core\_zone\_province` WHERE `mfdonline`=LEFT(NEW.mfdonline,2);  IF (province\_id\_tr IS NOT NULL) THEN  SET NEW.province\_id = province\_id\_tr;  END IF;  END IF;  SET NEW.city = TRIM(NEW.city);  SET NEW.modified\_date = NOW(); |
| coreBeforeInsertDistricts | DECLARE `city\_id\_tr` INT;    /\*CALL getOmmuZoneCityIdWithDistrictMfdonline(NEW.mfdonline, city\_id\_tr);\*/  SELECT `city\_id` INTO city\_id\_tr FROM `ommu\_core\_zone\_city` WHERE `mfdonline`=LEFT(NEW.mfdonline,4);  IF (city\_id\_tr IS NOT NULL) THEN  SET NEW.city\_id = city\_id\_tr;  END IF;    SET NEW.district\_name = TRIM(NEW.district\_name); |
| coreBeforeUpdateDistricts | DECLARE `city\_id\_tr` INT;    IF (NEW.mfdonline <> OLD.mfdonline) THEN  /\*CALL getOmmuZoneCityIdWithDistrictMfdonline(NEW.mfdonline, city\_id\_tr);\*/  SELECT `city\_id` INTO city\_id\_tr FROM `ommu\_core\_zone\_city` WHERE `mfdonline`=LEFT(NEW.mfdonline,4);  IF (city\_id\_tr IS NOT NULL) THEN  SET NEW.city\_id = city\_id\_tr;  END IF;  END IF;  SET NEW.district\_name = TRIM(NEW.district\_name);  SET NEW.modified\_date = NOW(); |
| coreBeforeInsertProvince | SET NEW.province = TRIM(NEW.province); |
| coreBeforeUpdateProvince | SET NEW.province = TRIM(NEW.province);  SET NEW.modified\_date = NOW(); |
| coreBeforeInsertVillage | DECLARE `district\_id\_tr` INT;    /\*CALL getOmmuZoneCityIdWithDistrictMfdonline(NEW.mfdonline, city\_id\_tr);\*/  SELECT `district\_id` INTO district\_id\_tr FROM `ommu\_core\_zone\_districts` WHERE `mfdonline`=LEFT(NEW.mfdonline,7);  IF (district\_id\_tr IS NOT NULL) THEN  SET NEW.district\_id = district\_id\_tr;  END IF;    SET NEW.village\_name = TRIM(NEW.village\_name);  SET NEW.zipcode = TRIM(NEW.zipcode); |
| coreBeforeUpdateVillage | DECLARE `district\_id\_tr` INT;    IF (NEW.mfdonline <> OLD.mfdonline) THEN  /\*CALL getOmmuZoneCityIdWithDistrictMfdonline(NEW.mfdonline, city\_id\_tr);\*/  SELECT `district\_id` INTO district\_id\_tr FROM `ommu\_core\_zone\_districts` WHERE `mfdonline`=LEFT(NEW.mfdonline,7);  IF (district\_id\_tr IS NOT NULL) THEN  SET NEW.district\_id = district\_id\_tr;  END IF;  END IF;  SET NEW.village\_name = TRIM(NEW.village\_name);  SET NEW.zipcode = TRIM(NEW.zipcode);  SET NEW.modified\_date = NOW(); |

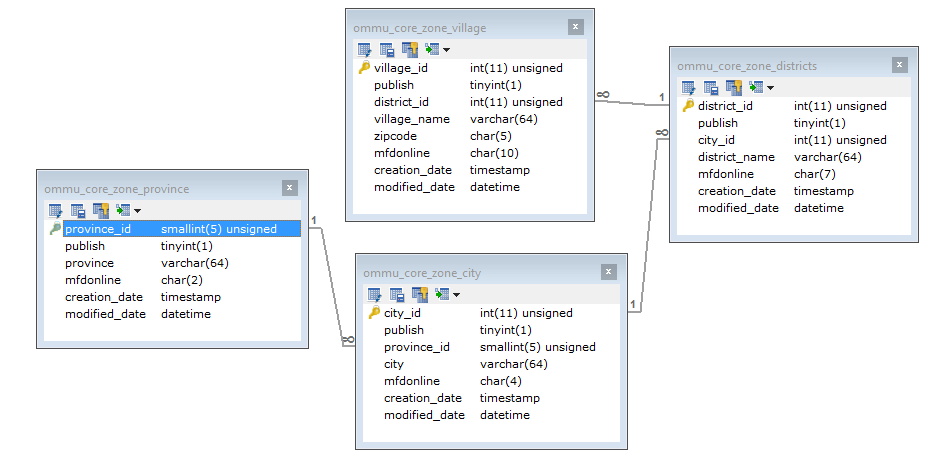
1. Store Prosedure

|  |
| --- |
| **Prosedure** |
| /\*!50003 CREATE DEFINER=`root`@`localhost` PROCEDURE `getOmmuZoneCityIdWithDistrictMfdonline`(IN `mfdonline\_sp` CHAR, OUT `city\_id\_sp` INT)  BEGIN  SELECT `city\_id` INTO city\_id\_sp FROM `ommu\_core\_zone\_city` WHERE `mfdonline`=LEFT(mfdonline\_sp,4);  END \*/$$  DELIMITER ; |
| /\*!50003 CREATE DEFINER=`root`@`localhost` PROCEDURE `getOmmuZoneCountryId`(IN `province\_id\_sp` SMALLINT, OUT `country\_id\_sp` SMALLINT)  BEGIN  SELECT `country\_id` INTO country\_id\_sp FROM `ommu\_core\_zone\_province` WHERE `province\_id`=province\_id\_sp;  END \*/$$  DELIMITER ; |
| /\*!50003 CREATE DEFINER=`root`@`localhost` PROCEDURE `getOmmuZoneProvinceId`(in `city\_id\_sp` INT, OUT `province\_id\_sp` SMALLINT)  BEGIN  SELECT `province\_id` INTO province\_id\_sp FROM `ommu\_core\_zone\_city` WHERE `city\_id`=city\_id\_sp;  END \*/$$  DELIMITER ; |
| /\*!50003 CREATE DEFINER=`root`@`localhost` PROCEDURE `getOmmuZoneProvinceIdWithCityMfdonline`(IN `mfdonline\_sp` CHAR, OUT `province\_id\_sp` SMALLINT)  BEGIN  SELECT `province\_id` INTO province\_id\_sp FROM `ommu\_core\_zone\_province` WHERE `mfdonline`=LEFT(mfdonline\_sp,2);  /\*  CALL getOmmuZoneProvinceIdWithCityMfdonline(NEW.mfdonline, province\_id\_tr);  SELECT `province\_id` INTO province\_id\_tr FROM `ommu\_core\_zone\_province` WHERE `mfdonline`=LEFT(NEW.mfdonline,2);\*/  END \*/$$  DELIMITER ; |

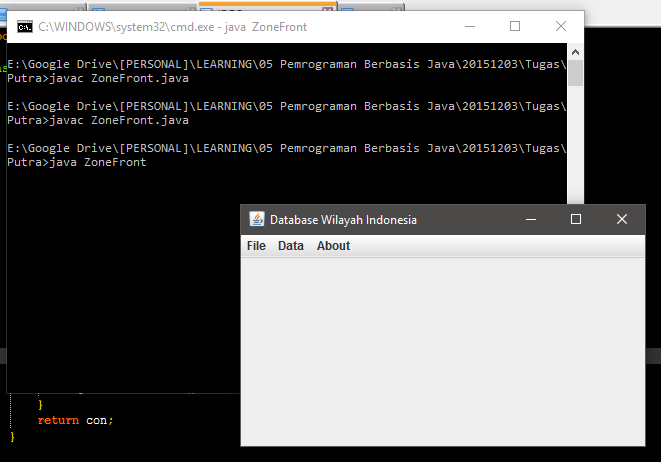
1. View

|  |  |
| --- | --- |
| **Name and View** | **Deskripsi** |
| `\_view\_core\_zone\_city`(  `city\_id` int(11) unsigned ,  `city` varchar(64) ,  `province` varchar(64)  ) | select `a`.`city\_id` AS `city\_id`, `a`.`city` AS `city`, `b`.`province` AS `province` from (`ommu\_core\_zone\_city` `a` left join `ommu\_core\_zone\_province` `b` on((`a`.`province\_id` = `b`.`province\_id`))) |
| `\_view\_core\_zone\_districts`(  `district\_id` int(11) unsigned ,  `district\_name` varchar(64) ,  `city` varchar(64) ,  `province` varchar(64)  ) | select `a`.`district\_id` AS `district\_id`, `a`.`district\_name` AS `district\_name`, `b`.`city` AS `city`, `b`.`province` AS `province` from (`ommu\_core\_zone\_districts` `a` left join `\_view\_core\_zone\_city` `b` on((`a`.`city\_id` = `b`.`city\_id`))) |
| `\_view\_core\_zone\_village`(  `village\_id` int(11) unsigned ,  `village\_name` varchar(64) ,  `district\_name` varchar(64) ,  `city` varchar(64) ,  `province` varchar(64)  ) | select `a`.`village\_id` AS `village\_id`, `a`.`village\_name` AS `village\_name`, `b`.`district\_name` AS `district\_name`, `b`.`city` AS `city`, `b`.`province` AS `province` from (`ommu\_core\_zone\_village` `a` left join `\_view\_core\_zone\_districts` `b` on((`a`.`district\_id` = `b`.`district\_id`))) |

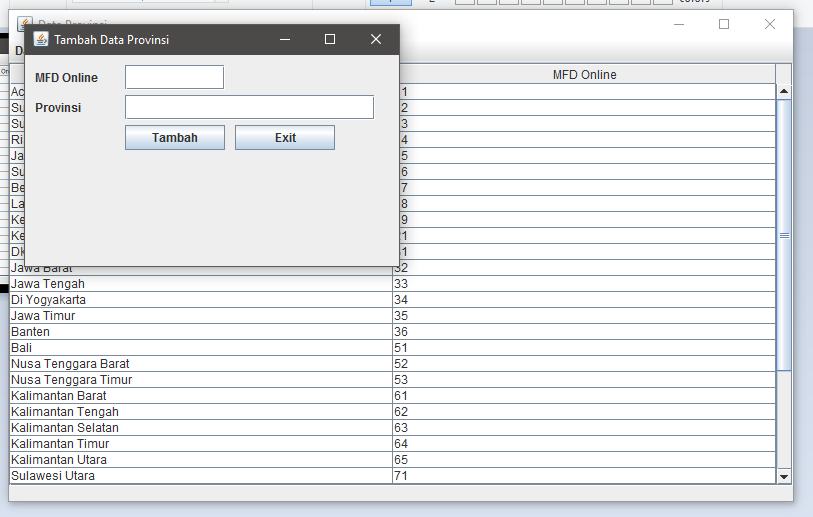
1. **Relasi Database**



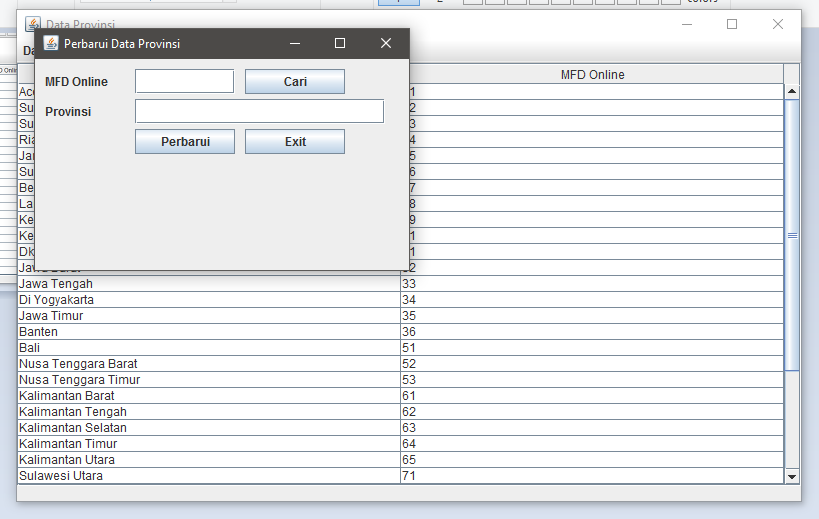
# **BAB III JDBC**



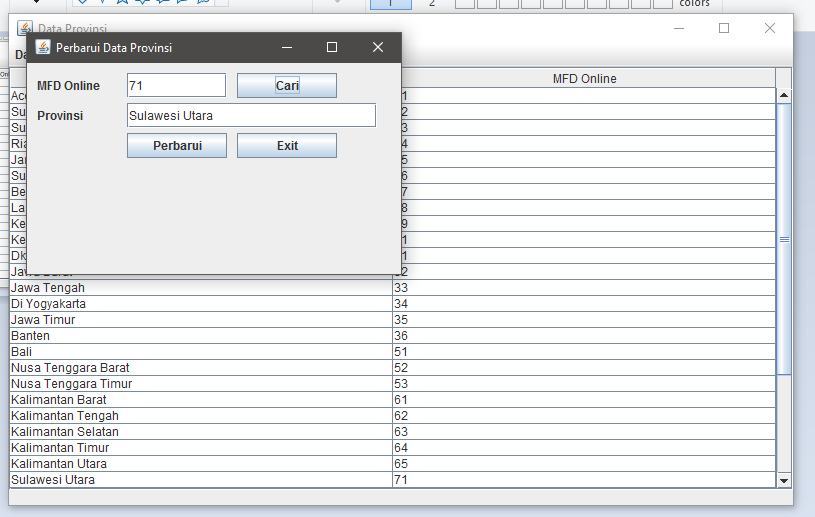
1. **Provinsi**
2. Insert



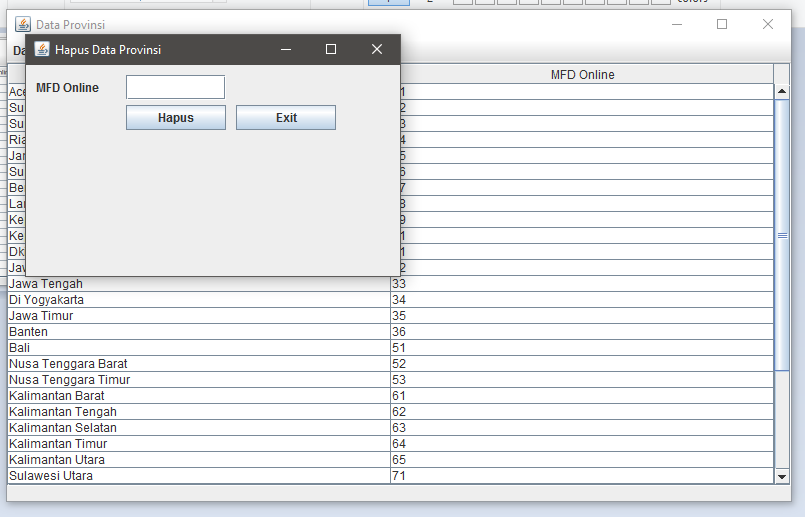
1. Search



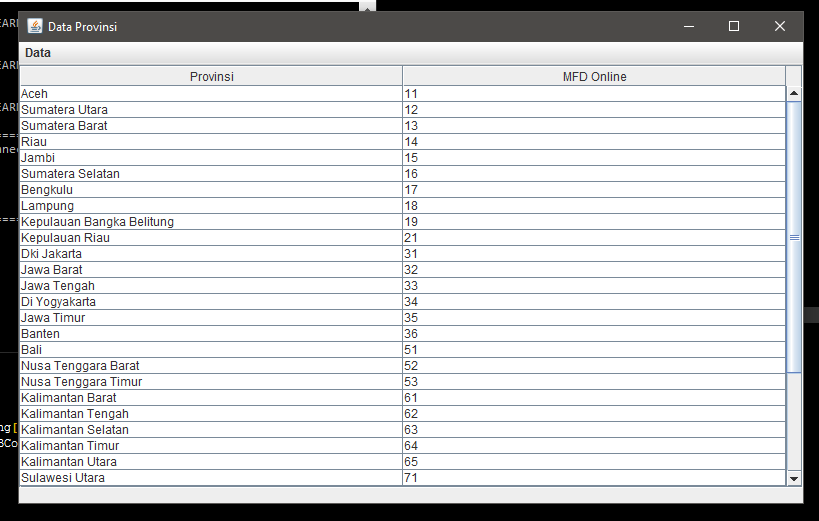
1. Update



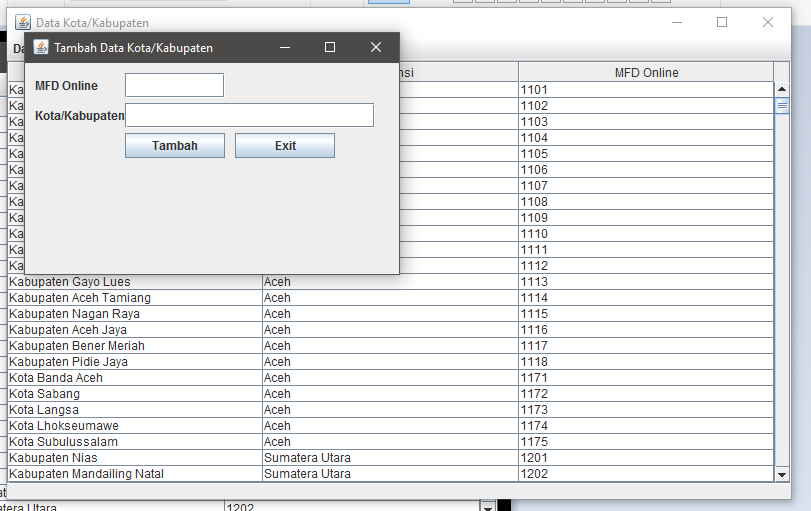
1. Delete



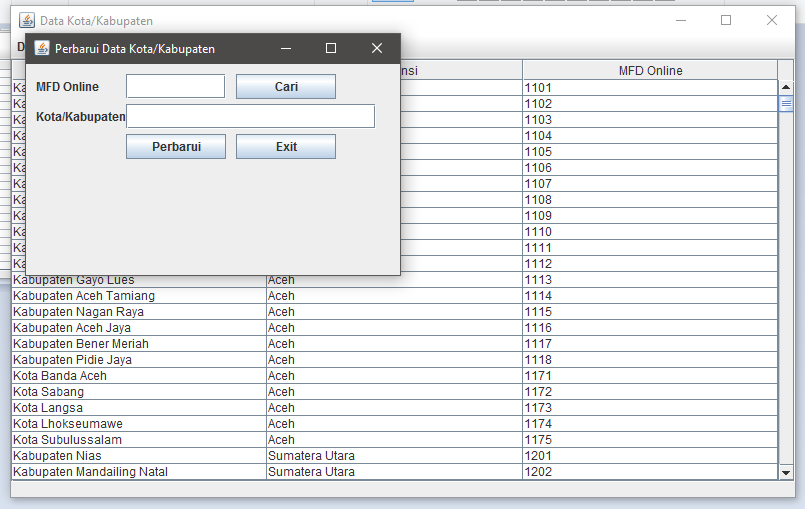
1. View



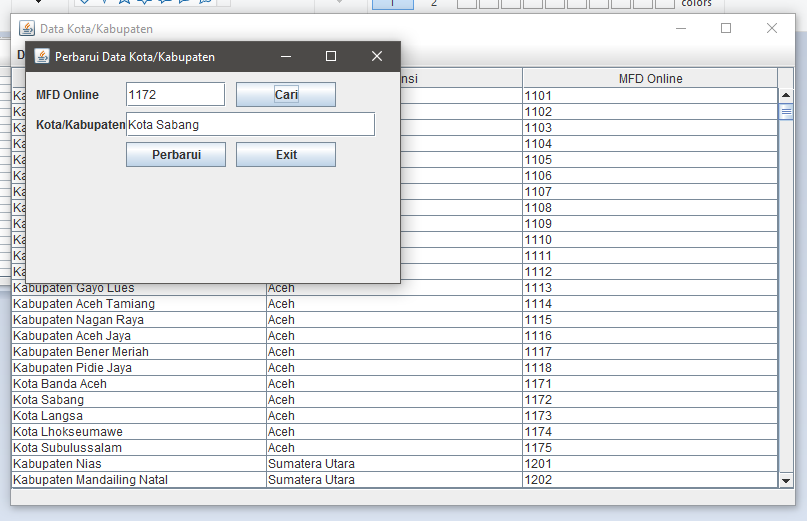
1. **Kabupaten / Kota**
2. Insert



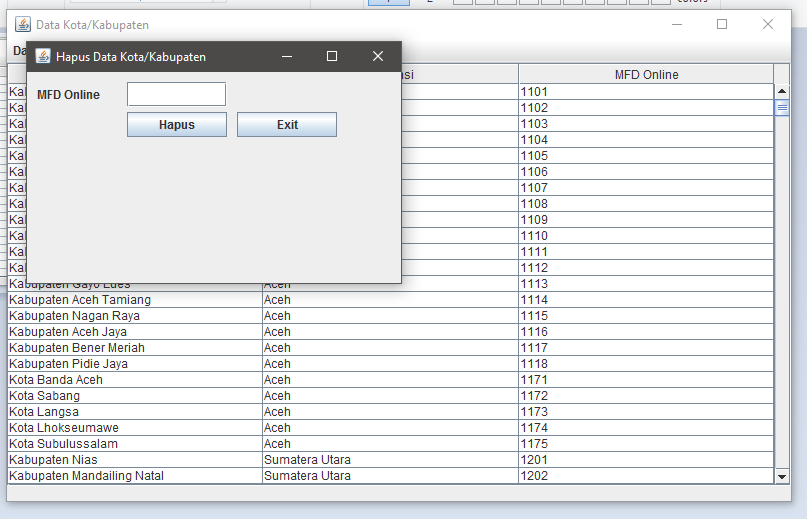
1. Search



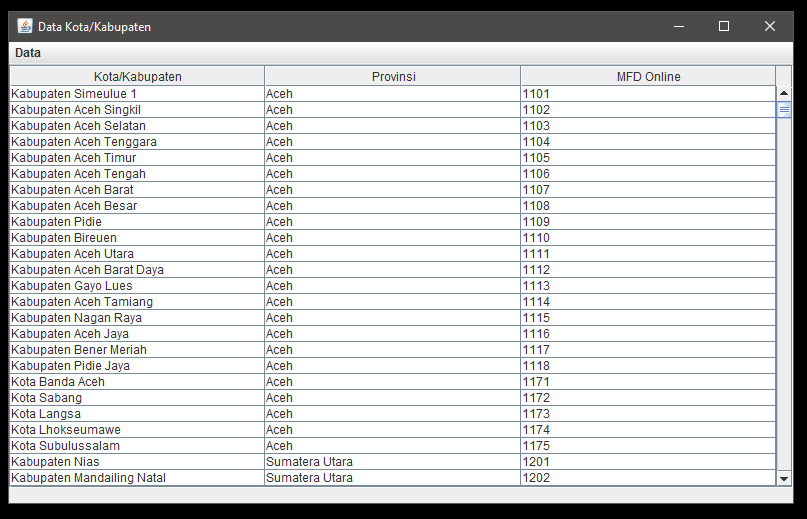
1. Update



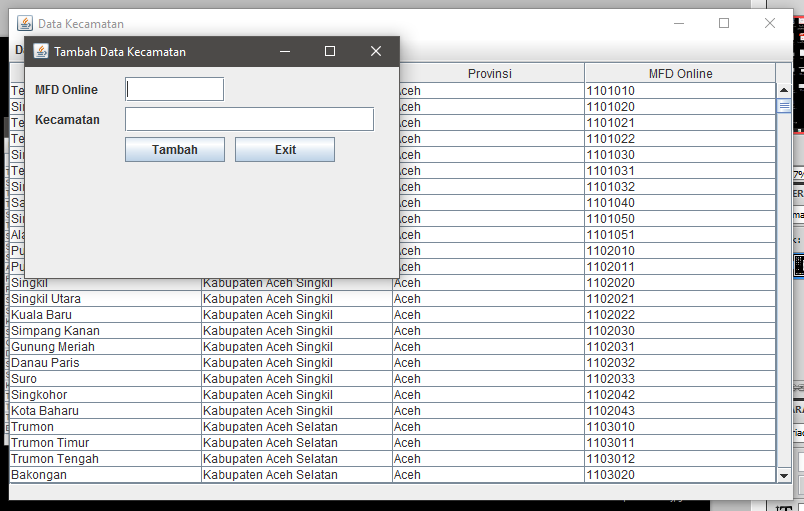
1. Delete



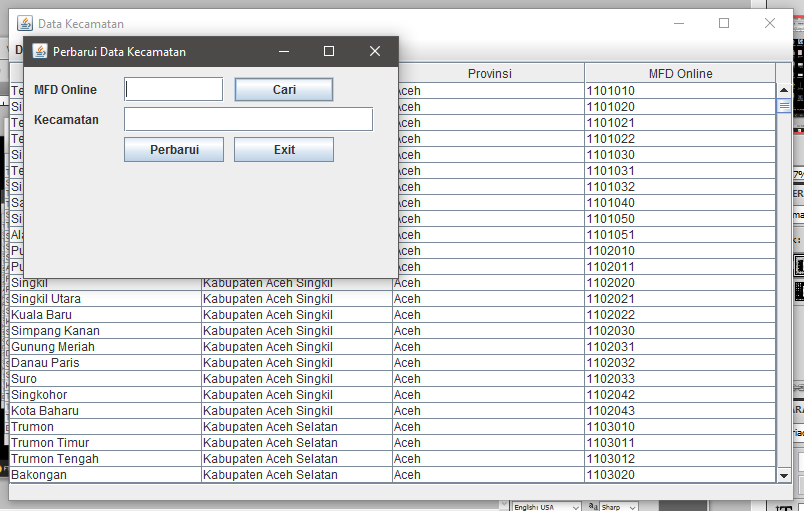
1. View



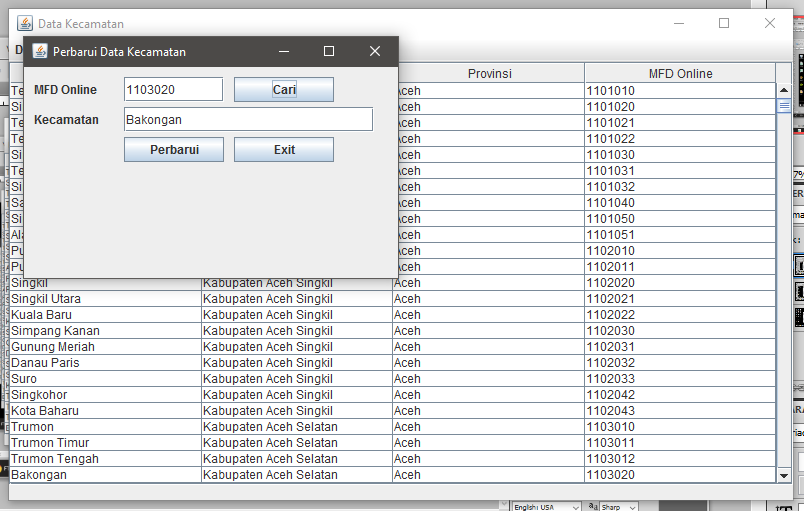
1. **Kecamatan**
2. Insert



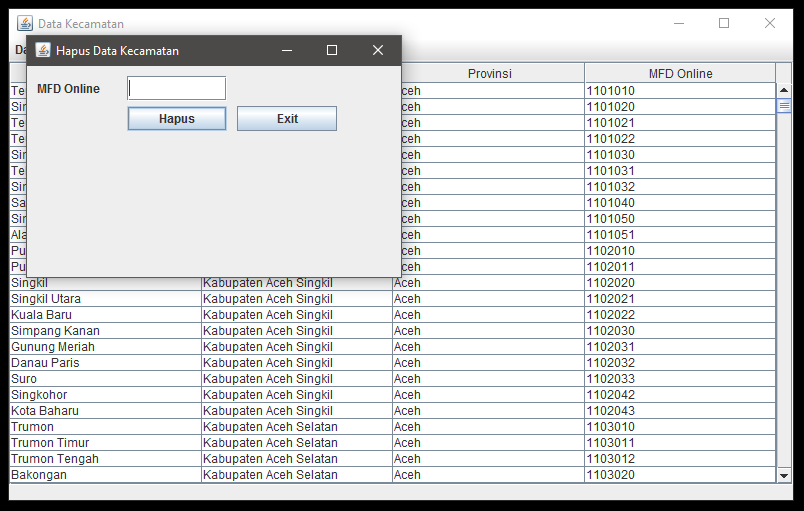
1. Search



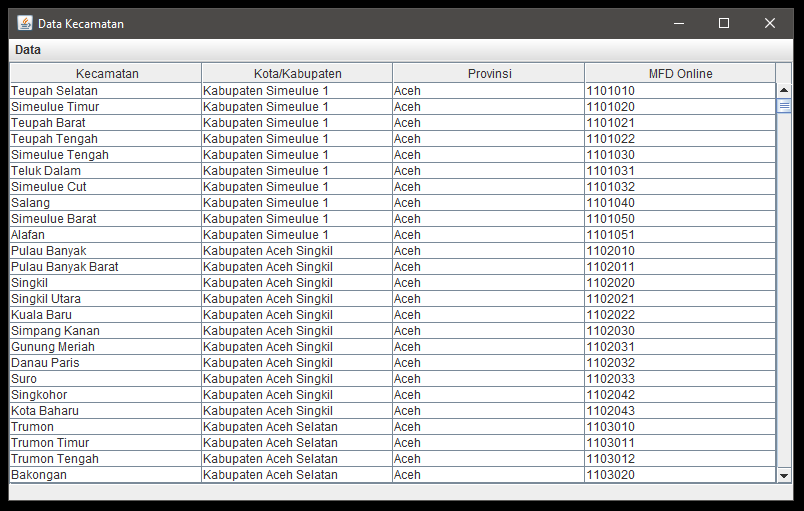
1. Update



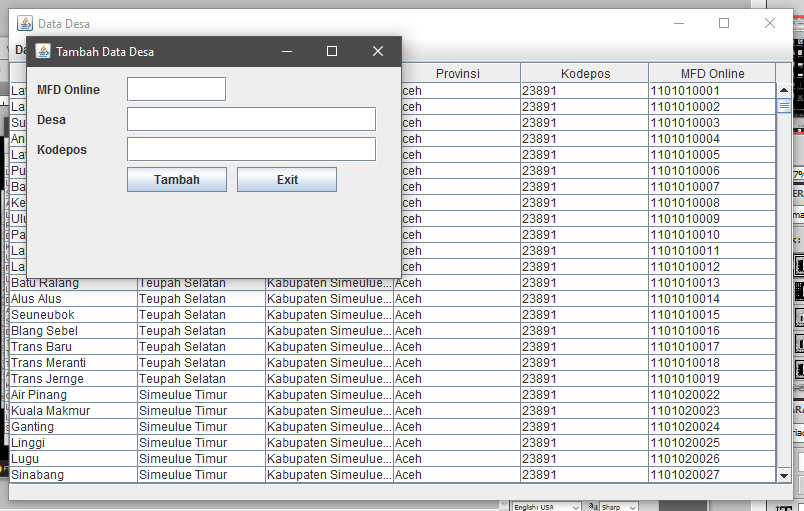
1. Delete



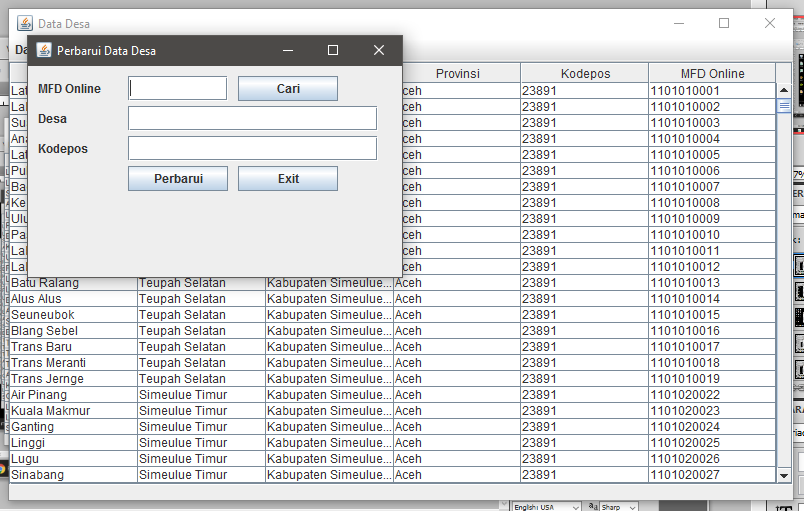
1. View



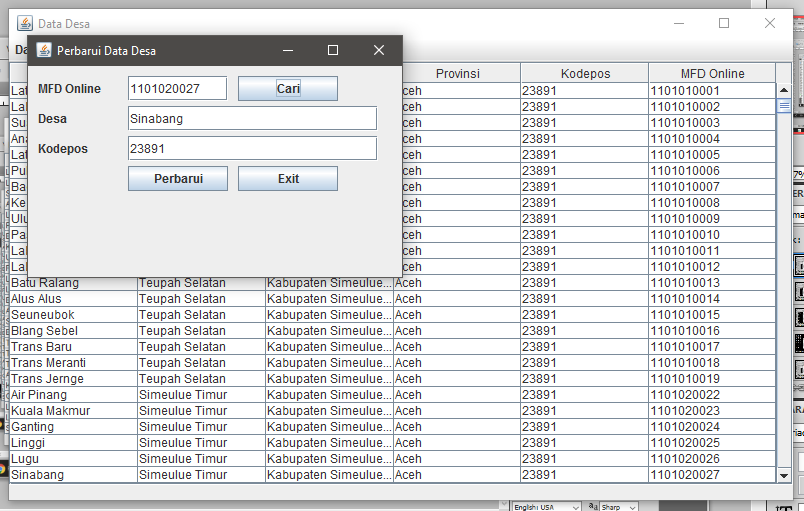
1. **Desa / Kelurahan**
2. Insert



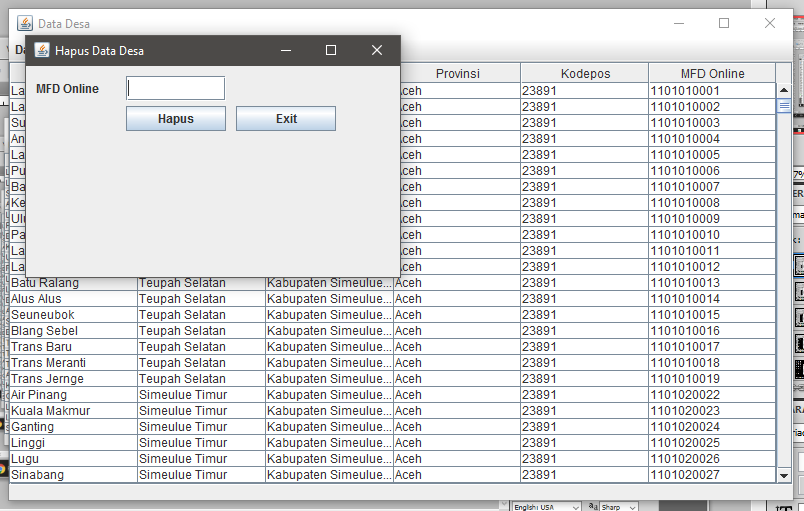
1. Search



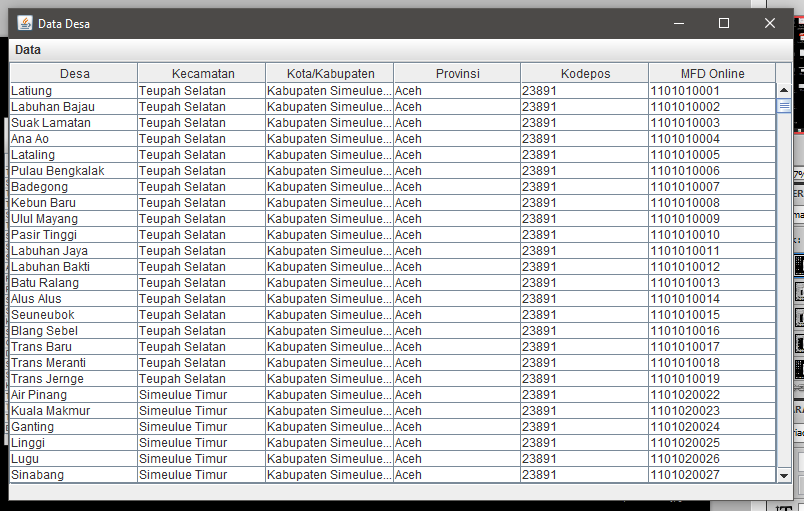
1. Update



1. Delete



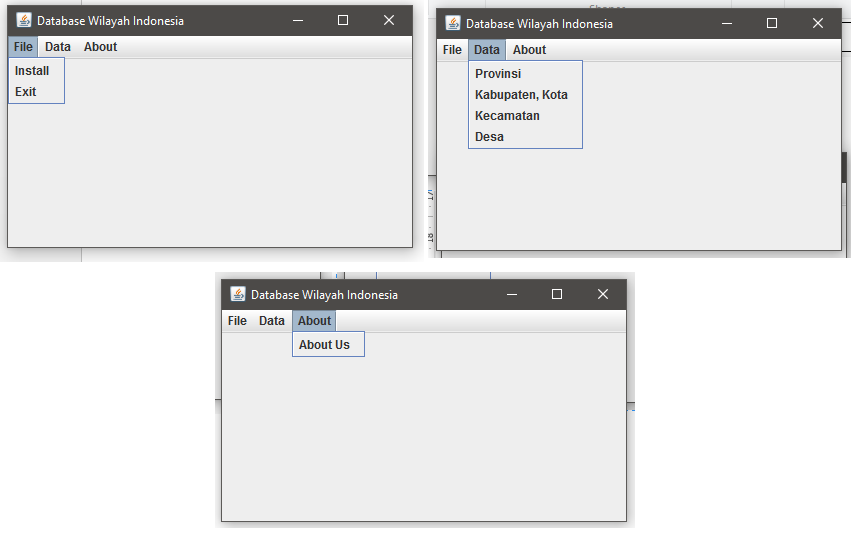
1. View



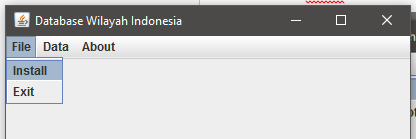
# **BAB IV MENU**

1. **Menu**

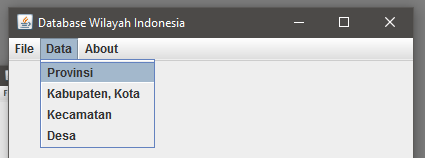
|  |  |  |
| --- | --- | --- |
| **Mainmenu** | **Submenu** | **Content Menu** |
| File | Install | - |
| Exit | - |
| Data | Provinsi | Tambah |
| Perbarui |
| Hapus |
| Kabupaten, Kota | Tambah |
| Perbarui |
| Hapus |
| Kecamatan | Tambah |
| Perbarui |
| Hapus |
| Kelurahan | Tambah |
| Perbarui |
| Hapus |
| About | About Us | - |

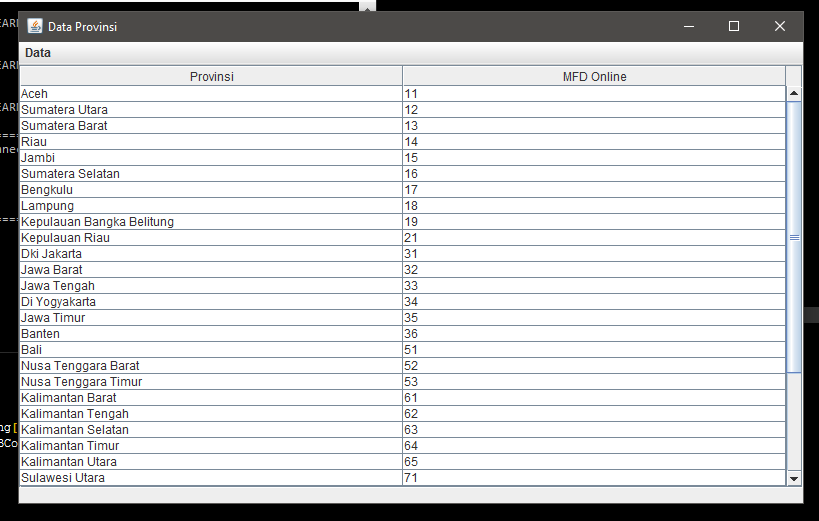


1. **Integrasi Menu**
2. Installer

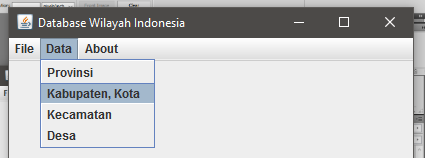


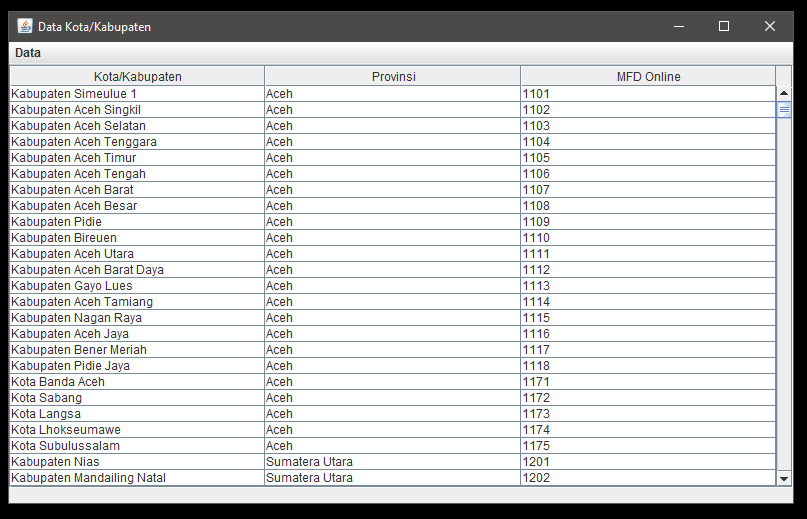
1. Provinsi



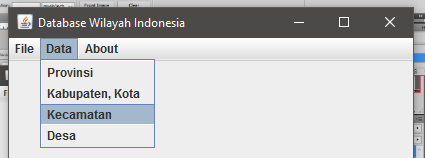


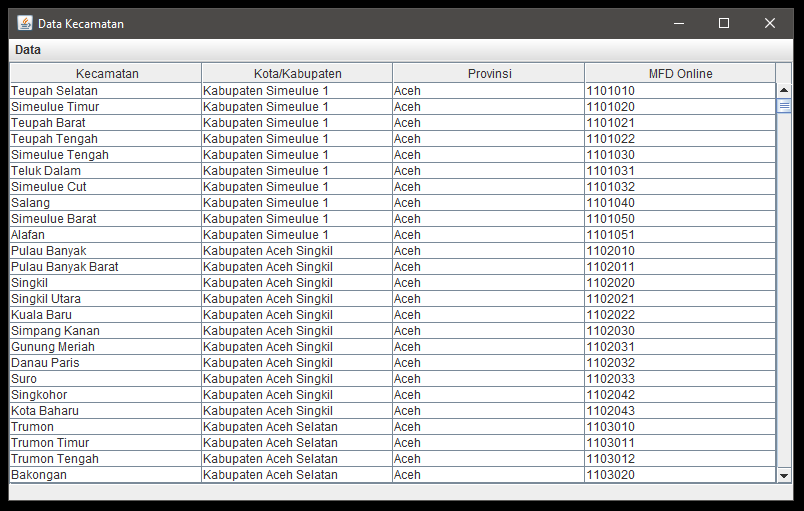
1. Kebupaten, Kota



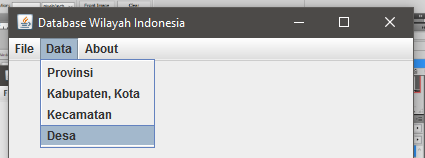


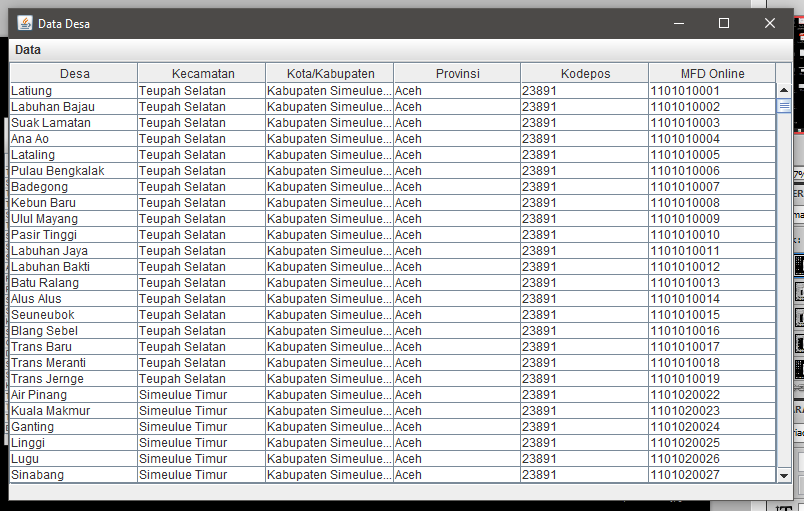
1. Kecamatan



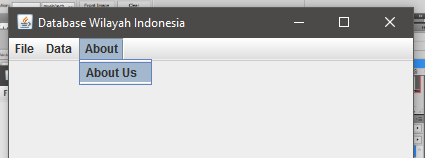


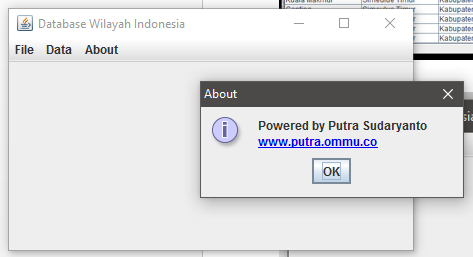
1. Kelurahan, Desa





1. About





# **BAB V PENUTUP**

1. **Kesimpulan**

Aplikasi yang dikembangkan dengan bahasa pemrograman Java tidah hanya bersifat statis, dengan bantuan JDBC kita mengembangkan sebuah aplikasi berbasis java yang bersifat dinamis dan terhubung dengan dengan basis data dengan menggunakan JDBC.

1. **Kesulitan**

Kesulitan yang saya alami dalam mengembangkan aplikasi berbasis java dengan JDBC, diantaranya:

1. Pada sub bab 4.2.1 untuk rencana awal saya ingin membuat sebuah aplikasi dengan installer, tetapi dikarenakan database (MySQL) yang saya buat menerapkan view, trigger dan store procedure untuk penerapan pada coding java terlalu sulit. Akhirnya saya mencoba untuk membuat installer dengan alternative lain yaitu mengambil source dari database yang sudah disediakan dan akhirnya sama, tidak berhasil juga.
2. Pada sub bab tentang view (select \* from) saya ingin membuat button diatas table untuk fungsi insert, update dan delete, dan akhirnya tidak bisa dan fungsi tersebut saya masukkan kedalam menu. Karena jika fungsi terdapat didalam menu harus ada event yang dilakukan (trigger) terlebih dahulu untuk menjalankan fungsi. Jika fungsi adalah sebuah button maka akan langsung terlihat oleh user.
3. Sebelum point 2 (dua) diatas saya ingin menerapkan fungsi update, delete langsung di row data, tetapi memang sangat sulit.
4. Penerapan validasi pada form terlalu ribet dan banyak (memakan waktu), saya mencoba menerapkan secara global (dipakai berulang) tapi masih gagal, akhirnya untuk validasi tidak saya gunakan pada aplikasi ini.
5. Dalan kesulitan-kesulitan lainnya.. ☺ ☺