**SISTEM INFORMASI UMKM**

**MENGGUNAKAN *CODEIGNITER***

**(Studi Kasus:** **Kantor Dinas Perdagangan dan Perindustrian Kabupaten Bandung)**

**KERJA PRAKTIK**

**Oleh**

**NAMA : DENI PURWANTO NPM : 41155050210017**

**NAMA : SIFA MARYAM RAHMAN NPM : 41155050210031**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS LANGLANGBUANA**

**2024**

LEMBAR PENGESAHAN

**SISTEM INFORMASI UMKM**

**MENGGUNAKAN *CODEIGNITER***

**(Studi Kasus: Kantor Dinas Perdagangan dan Perindustrian Kabupaten Bandung)**

**Oleh**

**NAMA : DENI PURWANTO NPM : 41155050210017**

**NAMA : SIFA MARYAM RAHMAN NPM : 41155050210031**

Menyetujui

Pembimbing Prodi Pembimbing Lapangan

(Iwan Abadi, S.Kom., M.M., M.T.) (Fahmi Asnia Maulida, S.M.B)

NIDN: 0416046903 NIP: 199011062020122001

Mengetahui

Ketua Program Studi

Teknik Informatika

(Kusmaya, S.Kom., M.T.)

NIDN: 0423096501

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadirat Allah Subhanahu wa ta’ala, yang telah memberikan rahmat, taufik serta hidayah-Nya kepada penulis sehingga pembuatan laporan kerja praktik ini dapat terselesaikan.

Dalam penyusunan ini penulis mengambil judul: “SISTEM INFORMASI UMKM MENGGUNAKAN CODEIGNITER (Studi Kasus: Kantor Dinas Perdagangan dan Perindustrian Kabupaten Bandung).” Laporan kerja praktik ini berguna agar mahasiswa dapat memahami dan menerapkan ilmu pengetahuan yang telah diperoleh dibangku kuliah dan dapat menghadapi masalah-masalah yang kemungkinan akan muncul, baik di lapangan atau di dunia kerja.

Dalam penyusunan laporan ini tidak lupa juga penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan dukungan dan dorongan dalam melakukan kerja praktik ini baik secara moril maupun materiil, diantaranya:

1. Dr. Sally Octaviana Sari, S.T., M.T. selaku Dekan Fakultas Teknik Universitas Langlangbuana.
2. Kusmaya, S.Kom., M.T. selaku ketua program studi Informatika Universitas Langlangbuana.
3. Iwan Abadi, S.Kom., M.M., M.T. selaku pembimbing akademis di Universitas Langlangbuana.
4. Fahmi Asnia Maulida, S.M.B selaku pembimbing lapangan di Dinas Perdagangan dan Perindustrian Kabupaten Bandung.
5. Seluruh Staff Dosen program studi Informatika Universitas Langlangbuana yang banyak memberikan pengetahuan dan pengajaran pada penyusunan.
6. Semua Staff dan Karyawan Universitas Langlangbuana.
7. Orang tua dan keluarga yang telah memberikan doa dan dukungannya.
8. Seluruh pihak yang telah mendukung sehingga penulis dapat menyelesaikan laporan ini.

Penulis menyadari sepenuhnya bahwa penyusunan laporan ini masih jauh dari sempurna dan terdapat kekurangan-kekurangan yang harus diperbaiki. Penulis meminta masukan berupa pandangan serta kritik yang bersifat membangun semua pihak. Semoga Allah S.W.T membalas kebaikan semua pihak yang telah banyak membantu dalam penyusunan laporan ini. Amiin.

Bandung, 18 Agustus 2024

Penulis

RINGKASAN

Aplikasi Pengelolaan Data UMKM bernama "SISTEM INFORMASI UMKM MENGGUNAKAN *CODEIGNITER* (Studi Kasus: Kantor Dinas Perdagangan dan Perindustrian Kabupaten Bandung)" merupakan suatu sistem yang digunakan untuk membantu pengelola dalam mengelola data UMKM. Aplikasi ini memiliki fitur kelola data master seperti User, Data UMKM, Pemilik UMKM, Promosi, dan Platform. Pembuatan aplikasi ini dimulai dari pengumpulan seluruh data yang dibutuhkan dengan melakukan observasi, literasi karya ilmiah, dan wawancara. Pengembangan perangkat lunak menggunakan metode Waterfall, perancangan model aplikasi menggunakan Unified Modeling Language (UML), dan diimplementasikan menggunakan *Framework CodeIgniter dan Perl Hypertext Preprocessor (PHP)* sebagai bahasa pemrogramannya serta *MySQL* sebagai sistem manajemen basis datanya. Setelah aplikasi selesai dibuat, pengujian perangkat lunak yang digunakan adalah metode *Black Box Testing.*

Kata Kunci: *CodeIgniter*, UMKM, *MySQL.*

DAFTAR ISI

[LEMBAR PENGESAHAN ii](#_Toc176415884)

[KATA PENGANTAR iii](#_Toc176415885)

[RINGKASAN v](#_Toc176415886)

[DAFTAR ISI vi](#_Toc176415887)

[DAFTAR GAMBAR viii](#_Toc176415888)

[DAFTAR TABEL x](#_Toc176415889)

[DAFTAR SINGKATAN DAN ISTILAH xi](#_Toc176415890)

[DAFTAR SIMBOL xii](#_Toc176415891)

[DAFTAR LAMPIRAN xiii](#_Toc176415892)

[BAB I PENDAHULUAN I-1](#_Toc176415893)

[I.1 Latar Belakang I-1](#_Toc176415894)

[I.2 Maksud dan Tujuan I-2](#_Toc176415895)

[I.3 Ruang Lingkup I-2](#_Toc176415896)

[I.4 Keluaran Kerja Praktik I-2](#_Toc176415897)

[I.5 Lokasi dan Waktu Pelaksanaan I-3](#_Toc176415898)

[I.6 Sistematika Penulisan I-3](#_Toc176415899)

[BAB II KAJIAN PUSTAKA II-5](#_Toc176415900)

[II.1 Pengertian Sistem II-5](#_Toc176415901)

[II.2 Pengertian Informasi II-5](#_Toc176415902)

[II.3 Sistem Informasi II-5](#_Toc176415903)

[II.4 Perangkat Lunak II-7](#_Toc176415904)

[II.5 Rekayasa Perangkat Lunak II-8](#_Toc176415905)

[II.6 UMKM II-9](#_Toc176415906)

[II.7 *SDLC* II-10](#_Toc176415907)

[II.8 *Model Waterfall* II-11](#_Toc176415908)

[II.9 *Website* II-12](#_Toc176415909)

[II.10 *Xampp* II-12](#_Toc176415910)

[II.11 *MySQL* II-13](#_Toc176415911)

[II.12 *Personal Hypertext Preprosessor (PHP)* II-13](#_Toc176415912)

[II.13 *Framework* II-14](#_Toc176415913)

[II.14 *CodeIgniter* II-15](#_Toc176415914)

[II.15 *JavaScript* II-15](#_Toc176415915)

[BAB III Proses Bisnis dan Arsitektur Sistem III-17](#_Toc176415916)

[III.1 Proses Bisnis III-17](#_Toc176415917)

[III.1.1 Proses Bisnis Sistem Berjalan III-17](#_Toc176415918)

[III.1.2 Proses Bisnis Usulan III-17](#_Toc176415919)

[III.2 Arsitektur Sistem III-19](#_Toc176415920)

[III.2.1 Arsitektur Sistem Usulan III-19](#_Toc176415921)

[BAB IV HASIL DAN PEMBAHASAN IV-1](#_Toc176415922)

[IV.1 Daftar Kebutuhan Fungsional dan Non Fungsional IV-1](#_Toc176415923)

[IV.1.1 Daftar Kebutuhan Fungsional IV-1](#_Toc176415924)

[IV.1.2 Daftar Kebutuhanm Non Fungsional IV-3](#_Toc176415925)

[IV.2 Design System IV-4](#_Toc176415926)

[IV.2.1 Diagram Use case IV-4](#_Toc176415927)

[IV.2.2 Skenario Use Case IV-5](#_Toc176415928)

[IV.2.3 *Class Diagram* IV-17](#_Toc176415929)

[IV.2.4 *Sequence Diagram* IV-18](#_Toc176415930)

[IV.3 Fitur-fitur Aplikasi IV-36](#_Toc176415931)

[IV.3.1 Implementasi *Interface* IV-36](#_Toc176415932)

[IV.4 Pseudocode Fitur-fitur Aplikasi IV-48](#_Toc176415933)

[IV.4.1 Pseudocode Create User IV-48](#_Toc176415934)

[IV.4.2 Pseudocode Edit User IV-51](#_Toc176415935)

[IV.4.3 Pseudocode Delete User IV-53](#_Toc176415936)

[IV.4.4 Pseudocode Create Data UMKM IV-53](#_Toc176415937)

[IV.4.5 Pseudocode Edit Data UMKM IV-54](#_Toc176415938)

[IV.4.6 Pseudocode Delete Data UMKM IV-55](#_Toc176415939)

[IV.4.7 Pseudocode Create Promosi IV-56](#_Toc176415940)

[IV.4.8 Pseudocode Update Promosi IV-58](#_Toc176415941)

[IV.4.9 Pseudocode Delete Promosi IV-60](#_Toc176415942)

[IV.5 Testing IV-60](#_Toc176415943)

[IV.5.1 Rencana Pengujian IV-60](#_Toc176415944)

[IV.5.2 Hasil Pengujian IV-64](#_Toc176415945)

[BAB V SIMPULAN DAN SARAN V-1](#_Toc176415946)

[V.1 Simpulan V-1](#_Toc176415947)

[V.2 Saran V-1](#_Toc176415948)

[DAFTAR PUSTAKA xii](#_Toc176415949)

[LAMPIRAN A: STRUKTUR ORGANISASI xiv](#_Toc176415950)

[LAMPIRAN B: HASILINTERVIEW xv](#_Toc176415951)

[LAMPIRAN C: KODE PROGRAM xvi](#_Toc176415952)

DAFTAR GAMBAR

[Gambar II. 1 Model Pengembangan Perangkat Lunak Waterfall II-11](#_Toc176114692)

[Gambar III. 1 Gambar Proses Bisnis Sistem Berjalan III-17](#_Toc176115025)

[Gambar III. 2 Gambar Proses Bisnis Sistem Usulan III-18](#_Toc176115026)

[Gambar III. 3 Gambar Arsitektur Sistem Usulan III-19](#_Toc176115027)

[Gambar IV. 1 Use Case Diagram IV-5](#_Toc176118038)

[Gambar IV. 2 Class Diagram IV-18](#_Toc176118039)

[Gambar IV. 3 Sequence Diagram Login IV-19](#_Toc176118040)

[Gambar IV. 4 Sequence Diagram Memasukan Data UMKM IV-21](#_Toc176118041)

[Gambar IV. 5 Sequence Diagram Melihat Data UMKM IV-22](#_Toc176118042)

[Gambar IV. 6 Sequence Diagram Mengubah Data UMKM IV-23](#_Toc176118043)

[Gambar IV. 7 Sequence Diagram Menghapus Data UMKM IV-24](#_Toc176118044)

[Gambar IV. 8 Sequence Diagram Memasukkan Pemilik UMKM IV-25](#_Toc176118045)

[Gambar IV. 9 Sequence Diagram Melihat Pemilik UMKM IV-26](#_Toc176118046)

[Gambar IV. 10 Sequence Diagram Mengubah Pemilik UMKM IV-27](#_Toc176118047)

[Gambar IV. 11 Sequence Diagram Menghapus Pemilik UMKM IV-28](#_Toc176118048)

[Gambar IV. 12 Sequence Diagram Memasukkan Platform IV-29](#_Toc176118049)

[Gambar IV. 13 Sequence Diagram Melihat Platform IV-30](#_Toc176118050)

[Gambar IV. 14 Sequence Diagram Mengubah Platform IV-31](#_Toc176118051)

[Gambar IV. 15 Sequence Diagram Menghapus Platform IV-32](#_Toc176118052)

[Gambar IV. 16 Sequence Diagram Memasukkan Prmosi IV-33](#_Toc176118053)

[Gambar IV. 17 Sequence Diagram Melihat Promosi IV-34](#_Toc176118054)

[Gambar IV. 18 Sequence Diagram Mengubah Promosi IV-35](#_Toc176118055)

[Gambar IV. 19 Sequence Diagram Menghapus Promosi IV-36](#_Toc176118056)

[Gambar IV. 20 Tampilan Home IV-37](#_Toc176118057)

[Gambar IV. 21 Tampilan Kategori IV-37](#_Toc176118058)

[Gambar IV. 22 Produk UMKM IV-38](#_Toc176118059)

[Gambar IV. 23 Tampilan Search UMKM IV-38](#_Toc176118060)

[Gambar IV. 24 Tampilan Login IV-39](#_Toc176118061)

[Gambar IV. 25 Tampilan Sign Up IV-39](#_Toc176118062)

[Gambar IV. 26 Tampilan Logout IV-40](#_Toc176118063)

[Gambar IV. 27 Tampilan Dashboard Admin IV-40](#_Toc176118064)

[Gambar IV. 28 Tampilan Menambahkan Pemilik UMKM IV-41](#_Toc176118065)

[Gambar IV. 29 Tampilan Menghapus Pemilik UMKM IV-41](#_Toc176118066)

[Gambar IV. 30 Tampilan Melihat Pemilik UMKM IV-42](#_Toc176118067)

[Gambar IV. 31 Tampilan Edit Pemilik UMKM IV-42](#_Toc176118068)

[Gambar IV. 32 Tampilan Memasukan UMKM IV-43](#_Toc176118069)

[Gambar IV. 33 Tampilan Melihat UMKM IV-43](#_Toc176118070)

[Gambar IV. 34 Tampilan Mengubah UMKM IV-44](#_Toc176118071)

[Gambar IV. 35 Tampilan Menghapus UMKM IV-44](#_Toc176118072)

[Gambar IV. 36 Tampilan Memasukkan Promosi IV-45](#_Toc176118073)

[Gambar IV. 37 Tampilan Mengubah Promosi IV-45](#_Toc176118074)

[Gambar IV. 38 Tampilan Melihat Promosi IV-46](#_Toc176118075)

[Gambar IV. 39 Tampilan Menghapus Promosi IV-46](#_Toc176118076)

[Gambar IV. 40 Tampilan Memasukkan Platfotm IV-47](#_Toc176118077)

[Gambar IV. 41 Tampilan Melihat Platform IV-47](#_Toc176118078)

[Gambar IV. 42 Tampilan Mengubah Platform IV-48](#_Toc176118079)

[Gambar IV. 43 Tampilan Menghapus Platform IV-48](#_Toc176118080)

DAFTAR TABEL

[Tabel II. 1 Jenis-Jenis Usaha II-9](#_Toc180489834)

[Tabel VI. 1 Table Kebutuhan Fungsional IV-1](#_Toc176121641)

[Tabel VI. 2 Table Kebutuhan Non-Fungsional Hardware IV-3](#_Toc176121642)

[Tabel VI. 3 Table Kebutuhan Non-Fungsional Software IV-4](#_Toc176121643)

[Tabel VI. 4 Deskripsi Pendefinisian Aktor Use Case IV-6](#_Toc176121644)

[Tabel VI. 5 Skenario Use Case Login IV-6](#_Toc176121645)

[Tabel VI. 6 Skenario Use Case Logout IV-7](#_Toc176121646)

[Tabel VI. 7 Skenario Use Case Memeriksa Statun Login IV-8](#_Toc176121647)

[Tabel VI. 8 Skenario Use Case memasukan pemilik UMKM IV-8](#_Toc176121648)

[Tabel VI. 9 Skenario Use Case Mengubah Pemilik UMKM IV-9](#_Toc176121649)

[Tabel VI. 10 Skenario Use Case Menghapus Pemilik UMKM IV-11](#_Toc176121650)

[Tabel VI. 11 Skenario Use Case Memasukkan Data UMKM IV-13](#_Toc176121651)

[Tabel VI. 12 Skenario Use Case Mengubah Data UMKM IV-14](#_Toc176121652)

[Tabel VI. 13 Skenario Use Case Menghapus Data UMKM IV-15](#_Toc176121653)

[Tabel VI. 14 Pengujian Sistem IV-61](#_Toc176121654)

[Tabel VI. 15 Hasil Pengujian Login IV-64](#_Toc176121655)

[Tabel VI. 16 Hasil Pengujian Kelola UMKM IV-65](#_Toc176121656)

[Tabel VI. 17 Hasil Pengujian Kelola Pemilik UMKM IV-65](#_Toc176121657)

[Tabel VI. 18 Hasil Pengujian Kelola Promosi IV-66](#_Toc176121658)

[Tabel VI. 19 Hasil Pengujian Kelola Platform IV-67](#_Toc176121659)

DAFTAR SINGKATAN DAN ISTILAH

|  |  |
| --- | --- |
| **Singkatan** | **Kepanjangan** |
| UMKM | Usaha Mikro Kecil dan Menengah |
| CI | *Codeigniter* |
| *SDLC* | *Software Development Life Cycle* |
| *Xampp* | *X (cross platform), A (Apache), M (MySQL/MariaDB), P (PHP), dan P (Perl)* |
| *MySQL* | *My Structured Query Language* |
| *PHP* | *Personal Hypertext Preprosessor* |
| *DBMS* | *Database Management System* |
| *RAM* | *Random Access Memory* |
| *VGA* | *Video Graphics Adapter* |

DAFTAR SIMBOL

1. **SImbol-simbol Dalam Diagram Usecase**

|  |  |
| --- | --- |
| **Simbol** | **Arti** |
|  | Aktor, pelaku dari sebuah usecase |
|  | Usecase, sekumpulan aksi yang membentuk suatu kesatuan lojik untuk menyelesaikan suatu kebutuhan fungsional |
|  | Asosiasi, menunjukkan siapa aktor yang bertanggung jawab atas suatu usecase |

**Sumber: Booch, 2005**

1. **Simbol-simbol Dalam Diagram Sekuens**

|  |  |
| --- | --- |
| **­** | **Arti** |
| IMG_256 | Kelas/Objek Interface/Boundary, kelas/objek yang merupakan penghubung antara aktor dengan sistem, atau menghubungkan sistem dengan sistem lain yang terkait (API) |
| IMG_256 | Kelas/Objek Controller, kelas//objek yang mengandung lojik dari sistem/usecase, library, atau kumpulan fungsi-fungsi dan prosedur yang digunakan dalam sistem. |
| IMG_256 | Kelas/Objek Etitas, kelas yang merepresentasikan data yang diolah di dalam sistem |

**Sumber: Booch, 2005**

DAFTAR LAMPIRAN

Lampiran A: Diagram Usecase Sistem Pembanding di Kementerian PUPR 105

Lampiran B: Alur pengambilan Data GIS Terpusat menurut Peraturan Kementrian PUPR no 15 Tahun 2017 107

# PENDAHULUAN

## Latar Belakang

Usaha Mikro, Kecil dan Menengah (UMKM) memiliki peran dan posisi yang strategis dalam perekonomian di Indonesia. UMKM berperan dalam menciptakan lapangan pekerjaan dan menggerakan roda perekonomian Indonesia serta mendistribusikan hasil-hasil pembangunan ekonomi.

Kantor Dinas Perdagangan dan Perindustrian (Disperindag) di Kabupaten Bandung merupakan salah satu unsur pelaksana Pemerintah Daerah dibidang perdagangan dan perindustrian. Salah satu sasaran dari tugas dan fungsi Disperindag adalah pembinaan Usaha Mikro Kecil dan Menengah (UMKM). Pembinaan ini dilakukan agar UMKM dapat berkembang dan semakin banyak lagi sektor usaha yang dilakukan oleh masyarakat, hal ini akan meningkatkan pertumbuhan perekonomian daerah bahkan perekonomian nasional.

Untuk mengetahui pertumbuhan UMKM maka dilakukannya pendataan. Saat ini Disperindag menghadapi tantangan dalam pengelolaan data Usaha Mikro Kecil dan Menengah (UMKM). Proses pendataan UMKM selama ini dilakukan secara manual, dimulai dari pengisian formulir kertas oleh petugas lapangan hingga penginputan data kedalam format digital, seperti excel. Metode tersebut rentan terhadap kesalahan dan memakan waktu yang cukup lama. Selain itu, terdapat kendala dalam aksesibilitas data dan kolaborasi antar petugas, karena informasi tersimpan dalam format terpisah. Resiko kehilangan data juga menjadi perhatian, mengingat dokumen kertas rentan terhadap kerusakan dan kehilangan.

Sehubung dengan latar belakang diatas, dalam melaksanakan pengelolaan data tersebut diperlukannya suatu perangkat lunak (software) yang berisi sistem informasi pengelolaan data Usaha Mikro Kecil dan Menengah (UMKM). Yang diharapkan tugas pokok dan fungsi dari sistem informasi ini dapat mengurangi kehilangan informasi, meningkatkan aksesibilitas, meningkatkan efisiensi dan keakuratan data. Berdasarkan latar belakang tersebut, peneliti tertarik untuk mengambil judul tentang: **“SISTEM INFORMASI UMKM MENGGUNAKAN *CODEIGNITER* (Studi Kasus : KANTOR DINAS PERDAGANGAN DAN PERINDUSTRIAN KABUPATEN BANDUNG)”.**

## Maksud dan Tujuan

Berdasarkan latar belakang diatas, peneliti merumuskan maksud dan tujuan sebagai berikut :

1. Membangun sistem informasi pengelolaan data UMKM berbasis web dapat mengurangi kehilangan informasi, meningkatkan aksesibilitas, dan keakuratan data.
2. Membangun sistem informasi pengelolaan data UMKM berbasis web dapat menyediakan pelayanan servis yang lebih baik, efektif dan efisien.
3. Membantu memudahkan Disperindag melakukan pendataan UMKM.

## Ruang Lingkup

Aplikasi sistem informasi pengelolaan data UMKM berbasis web menggunakan *Codeigniter (CI)* mempunyai ruang lingkup sebagai berikut :

1. Sistem informasi ini memiliki fitur-fitur pengelolaan data UMKM, pengelolaan data pemilik UMKM, dan pengelola data user meliputi create, read, update dan delete yang disesuaikan dengan kebutuhan aplikasi.
2. Sistem informasi ini juga memiliki fitur seperti management hak akses dimana itu untuk mengatur hak akses pengguna terhadap pengelolaan data (create, read, update, dan delete).
3. Sistem informasi ini menyimpan data dalam basis data.
4. Sistem informasi ini memiliki fitur untuk validasi data UMKM oleh admin.
5. Sistem informasi ini memiliki fitur monthly report jumlah UMKM yang terdaftar.
6. Sistem informasi ini memiliki fitur search untuk mempermudah pengguna menemukan data yang dicari.

## Luaran Kerja Praktik

Adapun luaran kerja praktik yang dihasilkan sebagai berikut:

1. Sistem Informasi UMKM berbasis web menggunakan *Codeigniter (CI).*
2. Laporan Kerja Praktik.

## Lokasi dan Waktu Pelaksanaan

Adapun Lokasi dan waktu pelaksanaan sebagai berikut:

Tempat : Kantor Dinas Perdagangan Dan Perindustrian Kabupaten Bandung

Alamat : Jl. Raya Soreang Km.17 Soreang Kabupaten Bandung, Jawa Barat, 40912.

Waktu Pelaksanaan : 3 Bulan

## Sistematika Penulisan

Sistematika penulisan proposal kerja praktik ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan.

Sistematika penulisan proposal kerja praktik ini sebagai berikut:

**BAB I PENDAHULUAN**

Pada bab ini meliputi latar belakang, maksud dan tujuan, ruang lingkup , luaran kerja praktik, lokasi dan waktu pelaksanaan dan sistematika pemulisan.

**BAB II KAJIAN PUSTAKA**

Pada bab ini membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian mengenai pemahaman yang diambil dari berbagai sumber seperti jurnal, buku, maupun artikel yang berkaitan dengan permasalahan.

**BAB III PROSES BISNIS DAN ARSITEKTUR SISTEM**

Pada bab ini digunakan untuk mendefinisikan kebutuhan pemakai mulai dari proses bisnis sistem berjalan, proses bisnis usulan, arsitektur sistem.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini digunakan untuk mendefinisikan kebutuhan pemakai mulai dari proses bisnis sistem berjalan, proses bisnis usulan, arsitektur sistem.

**BAB V SIMPULAN DAN SASARAN**

Pada bab ini berisi kesimpulan dan saran yang berkaitan dengan analisa dan kinerja sistem berdasarkan pada bab-bab sebelumnya. Pada bab ini juga berisi pokok bahasan yang telah diuraikan untuk kepentingan penulis dan kepentingan proses pengembangan selanjutnya.

# KAJIAN PUSTAKA

## Pengertian Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gamabaran, jika sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem (Kadir, 2014).

Sistem adalah sekumpulan komponen atau jaringan kerja dari prosedur-prosedur yang saling berkaitan dan saling bekerja sama membentuk suatu jaringan kerja untuk mencapai sasaran atau tujuan tertentu (Muslihudin, 2016).

Sistem merupakan suatu kumpulan dari komponen-komponen yang membentuk satu kesatuan (Tyoso, 2016).

## Pengertian Informasi

Menurut (Darmawan, 2013) informasi merupakan hasil dari pengolahan data, akan tetapi tidak semua hasil dari pengelolaan tersebut bisa menjadi informasi, hasil pengelolaan data yang tidak memberikan makna atau arti serta tidak bermanfaat bagi seseorang bukanlah merupakan informasi bagi orang tersebut.

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimannya (Prof. Dr. Jogiyanto HM, 2017).

Informasi merupakan data yang sudah diolah yang ditunjukan untuk seseorang, organisasi ataupun siapa saja yang membutuhkan. Informasi akan menjadi berguna apabila objek yang menerima informasi membutuhkan informasi tersebut (Mulyani, 2016).

## Sistem Informasi

Sesungguhnya yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer *(Computer Based Information System atau CBIS).*

Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer, walaupun dalam kenyataannya komputer merupakan bagian yang penting. Di buku ini, yang dimaksudkan dengan sistem iformasi adalah sistem informasi berbasis komputer.Ada beragam definisi sistem informasi, yaitu:

1. Alter, sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
2. Bodnar dan Hopwoo, sistem informasi adalah kumpulan perangkat keras dan peragkat lunak yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna.
3. Gelinas, Oram dan Wiggins, sistem informasi adalah suatu system buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis computer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.
4. Hall, Sistem informasi adalah sebuah rangkaian prosedur formal, dimana data dikelompokkan, diproses menjadi informasi dan di distribusikan kepada para pemakai.
5. Turban, McLean dan Wetherbe, Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuam yang spesifik.
6. Wilkinson, Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia dan komputer) untuk mengubah masukan (input) menjadi keluaran (informasi) guna mencapai sasaran-sasaran perusahaan.

Berdasarkan berbagai definisi tersebut, dapat disimpulkan bahwa sistem informasi mencangkup sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Kadir, 2014).

## Perangkat Lunak

(M.Shalahuddin., 2019) Perangkat Lunak *(Software)* adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan *(user manual)* . sebuah program komputer yang belum terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak *(software).* Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak. Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin tercapai.

Karakter perangkat lunak adalah sebagai berikut:

* Perangkat lunak dibangun dengan rekayasa *(software engineering)* bukan diproduksi secara manufaktur atau pabrikan.
* Perangkat lunak tidak pernah usang *(ware out)*  karena kecacatan dalam perangkat lunak dapat diperbaiki.
* Barang produksi pabrikan biasanya komponen barunya akan terus diprosuduksi, sedangkan perangkat lunak biasanya terus diperbaiki seiring bertambahnya kebutuhan.

Aplikasi perangkat lunak adalah sebagai berikut:

* Perangkat lunak sistem *(system software)*

Adalah kumpulan program dalam hal ini program yang satu ditulis untuk memenuhi kebutuhan program lainnya.

* Perangkat lunak waktu nyata *(real-time software)*

Merupakan perangkat lunak yang memonitor, menganalisis, mengontrol sesuatu secara waktu nyata *(real-time).* Reaksi yang dibutuhkan pada perangkat lunak harus langsung menghasilkan respon yang diinginkan

* Perangkat lunak bisnis *(business software)*

Merupakan perangkat lunak pengelola informasi bisnis (seperti akuntansi, penjualan, pembayaran, penyimpanan *(inventory)*)*.*

* Perangkat lunak untuk keperluan rekayasa dan keilmuan *(engineering and scientific software)*

Merupakan perangkat lunak yang diimplementasikan algoritma yang terkait dengan keilmuan ataupun perangkat lunak yang membantu keilmuan, misalkan perangkat lunak dibidang astronomi, dibidang matematika dan lain sebagainya.

* Perangkat lunak tambahan untuk membantu mengerjakan suatu fungsi dari perangkat lunak yang lainnya *(embendded software)*

Merupakan perangkat lunak untuk mencetak dokumen ditambahkan agar perangkat lunak yang memerlukan dapat mencetak laporan. Maka perangkat lunak untuk mencetak dokumen ini disebut *embedded software.*

* Perangkat lunak komputer personal *(personal computer software)*

Merupakan perangkat lunak untuk PC misalnya perangkat lunak pemroses teks, pemroses grafik dan lain sebagainya.

* Perangkat lunak berbasis web *(web based software)*

Merupakan perangkat lunak yang dapat diakses dengan menggunakan browser.

* Perangkat lunak berintelijensia buatan *(artificial intelligence software)*

Merupakan perangkat lunak yang menggunakan algoritma tertentu untuk mengelola data sehingga seakan-akan memiliki intelijensia seiring bertambahnya data yang diproses.

## Rekayasa Perangkat Lunak

(M.Shalahuddin., 2019) Rekayasa Perangkat Lunak *(Software Engineering)* merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.

Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan *(Customer).* Adapun ilmu komputer lebih fokus pada teori dan konsep dasar perangkat komputer. Rekayasa perangkat lunak lebih fokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria berikut:

* Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan *(maintainability).*
* Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi *(dependability* dan *robust).*
* Efisien dari segi sumber daya dan penggunaan.
* Kemampuan untuk dipakai sesuai dengan kebutuhan *(usability).*

## UMKM

Sesuai dengan Undang-Undang Nomor 20 Tahun 2008 tentang Usaha Mikro,Kecil dan Menengah (UMKM):

1. Usaha Mikro adalah usaha produktif milik orang perorangan dan/atau badan usaha perorangan yang memenuhi kriteria Usaha Mikro sebagaimana diatur dalam Undang-Undang ini.
2. Usaha Kecil adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh orang perorangan atau badan usaha yang bukan merupakan anak perusahaan atau bukan cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dari usaha menengah atau usaha besar yang memenuhi kriteria Usaha Kecil sebagaimana dimaksud dalam Undang-Undang ini.
3. Usaha Menengah adalah usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh badan usaha yang bukan merupakan anak perusahaan atau cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dengan usaha kecil atau usaha besar dengan jumlah kekayaan bersih atau hasil penjualan tahunan sebagaimana diatur dalam Undang-Undang ini.
4. Kriteria UMKM berdasarkan jumlah asset dan omzet

Jenis-jenis usaha akan dijelaskan pada Tabel II.1

Tabel II. 1 Jenis-Jenis Usaha

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Jenis Usaha** | **Asset** | **Omzet** |
| 1 | Usaha Mikro | Maximal 50 Juta | Maximal 300 Juta |
| 2 | Usaha Kecil | >50 Juta – 500 Juta | >300 Juta – 2,3 Miliyar |
| 3 | Usaha Menengah | >500 Juta – 10 Miliyar | >2,5 Miliyar-50 Miliyar |

Berdasarkan UU.no 20 tahun 2008 di atas jelas menunjukan perbedaan yang cukup besar baik dari segi asset ataupun omzet antara usaha mikro dengan kecil dan usaha kecil dengan menengah. Namun yang jelas secara keseluruhan UMKM berperan dalam pembangunan perekonomian nasional, hal ini sesuai juga dengan uu.no 20 tahun 2008 bab II pasal yang berbunyi:

“usaha mikro kecil dan menengah bertujuan menumbuhkan dan mengembangkan usahanya dalam rangka membangun perekonomian nasional berdasarkan demokrasi ekonomi yang berkeadilan”.

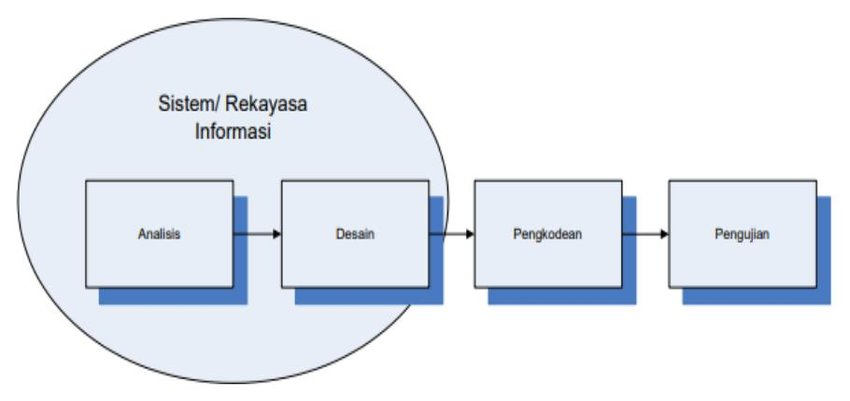
Pengembangan UMKM perlu mendapatkan perhatian yang besar baik dari pemerintah maupun masyarakat agar dapat berkembang lebih kompetitif bersama pelaku ekonomi lainnya. Kebijakan pemerintah ke depan perlu diupayakan lebih kondusif bagi tumbuh dan berkembangnya UMKM. Pemerintah perlu meningkatkan perannya dalam memberdayakan UMKM disamping mengembangkan kemitraan usaha yang saling menguntungkan antara pengusaha besar dengan pengusaha kecil, dan meningkatkan kualitas Sumber Daya Manusianya.

## *SDLC*

(M.Shalahuddin., 2019) *SDLC (Software Development Life Cycle)* sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metedelogi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *based pratice* atau cara-cara yang sudah teruji baik). Seperti halnya proses metamorfosis pada kupu-kupu, untuk menjadi kupu-kupu yang indah maka dibutuhkan beberapa tahap untuk dilalui, sama halnya dengan membuat perangkat lunak, memiliki daur tahapan yang dilalui agar menghasilkan perangkat lunak yang berkualitas.

## *Model Waterfall*

(M.Shalahuddin., 2019) model *SDLC* air terjun *(waterfall)* sering juga disebut model sekuensial linier *(sequential linier)* atau alur hidup klasik *(classic life cycle).* Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis desain, pengodean, pengujian, dan tahap pendukung *(support).* Adapun skema model *waterfall* dapat dilihat pada Gambar II.1.



Gambar II. 1 Model Pengembangan Perangkat Lunak Waterfall

Berikut penjelasan dari gambar diatas:

1. Analisis kebutuhan perangkat lunak

Menganalisa permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan seperti data UMKM, Pemilik UMKM, User, Platform dan Promosi. Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak seperti apa yang dibutuhkan oleh *user.*

1. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk proses bisnis, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

1. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program computer sesuai dengan desain yang telah dibuat pada tahap desain.

1. Pengujian

Pengujian focus pada perangkat lunak dari segi logik dan fungsionalitas dan juga memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan *(error)* dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Kemudian pemeliharaan sistem secara berkala, perbaikan sistem dan evaluasi sistem.

## *Website*

Menurut (Abdulloh, 2015) *Website* atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet.

## *Xampp*

Menurut (Hariyanto, 2012) *Xampp* adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah *server*. Kegunaan *Xampp* ini untuk membuat jaringan *local* sendiri dalam artian kita dapat membuat *website* secara *offline* untuk masa coba-coba di komputer sendiri. Jadi fungsi dari *Xampp server* itu sendiri merupakan *server website* kita untuk cara memakainya. Disebut *server* karena dalam hal ini komputer yang akan kita pakai harus memberikan pelayanan untuk mengakseskan web, untuk itu komputer kita harus menjadi *server.*

Dapat disimpulkan *xampp* adalah aplikasi *tools* untuk menyediakan paket lunak yang berisi konfigurasi *Web Server, Apache, PHP, MySQL* untuk membantu kita dalam proses pembuatan aplikasi web yang menyatu menjadi satu sehingga memudahkan kita dalam membuat program web.

Menurut (Nugroho, 2013) *Xampp* adalah gabungan program web lengkap yang dapat digunakan untuk berlatih bahasa pemrograman web, khususnya *PHP* dan *MySQL.*

Menurut (Buana, 2014) *XAMPP* adalah perangkat lunak *open source* yang didapatkan secara gratis dan bisa digunakan di seluruh operasi yang berbeda seperti *windows, linux, solaris, dan mac.*

## *MySQL*

*My Structured Query Language (MySQL)* merupakan salah satu aplikasi *Button Management System (DBMS)* yang mampu menjadi penerima dan pengirim suatu data dengan begitu cepat, adanya penggunaan perintah *SQL* dalam fitur *multiuser. MySQL* sendiri bersifat *open source* sehingga dapat dimanfaatkan secara gratis. *Pemrograman PHP* juga sangat mendukung adanya *button MySQL.*

Menurut (A.S, 2014) *My Structured Query Language (SQL)* adalah salah satu bahasa yang dimanfaatkan untuk mengatur suatu data pada *Relational Button Management System (RDBMS).*

Menurut (Raharjo, 2014) mendefinisikan bahwa *MySQL* merupakan *RDBMS (server button)* yang dapat mengelola *button* dengan sangat cepat, dapat menghimpun data dalam jumlah sangat banyak, dan dapat diakses oleh banyak *user*.

## *Personal Hypertext Preprosessor (PHP)*

Menurut (Abdulloh, 2015) *PHP* singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programing*, yaitu bahasa pemrograman yang diproses di sisi *server.* Fungsi utama *PHP* dalam membangun *website* adalah untuk melakukan pengolahan data pada *database.* Data *website* akan dimasukkan ke database, diedit, dihapus, dan ditampilkan pada website yang diatur oleh *PHP..*

*PHP* berasal dari kata *Hypertext Preprocessor,* yaitu bahas pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan *HTML.*

Menurut (Kusumo, 2016) *Personal Hypertext Preprosessor (PHP)* adalah bahasa yang dirancang secara khusus untuk penggunaan pada *Web. PHP* adalah *tool* untuk pembuatan halaman web dinamis. Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page (Situs Personal).* *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *FI (Form Interpreted),* yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari web. Saat ini *PHP* adalah singkatan dari *Hypertext PreProcessor,* sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri yaitu *Hypertext PreProcessor.*

## *Framework*

Menurut Budi Raharjo (2015) mendefinisikan *Web Application Framework* *(WAF),* atau biasa disebut dengan *web framework* adalah suatu gabungan *Code* berupa pustaka (*library*) dan alat (*tool*) yang dipadupadankan sesuai aturan yang ada menjadi satu kerangka kerja (*framework*) guna mempermudah dan mempercepat kinerja pengembangan aplikasi *web*. Proses pengembangan *web* itu sendiri dapat dikerjakan dengan berbagai macam bahasa pemrograman; seperti *PHP, Python*, *Ruby*, *Java* dan lainnya.

* + - 1. *Framework* *web* untuk *PHP:*

1. *CodeIgniter*
2. *YII*
3. *Zend Framework*
4. *Laravel*
5. *Symfony*
6. *dll*
   * + 1. *Framework* web untuk *Python:*
7. *Django*
8. *CherryPy*
9. *Flask*
10. *Pyramid*
11. *dll*
    * + 1. *Framework web* untuk *Ruby:*
12. *Ruby on Rails*
13. *Sinatra*
14. *Ramaze*
15. *Padrino*
16. *dll*

Sebagian besar dari *framework* yang ada di atas menggunakan pola desain atau kerangka *Model-View-Controller (MVC),* yang dipisahkan antara kode untuk penggunaan proses bisnis dengan bagian kode untuk penggunaan tampilan.

## *CodeIgniter*

Code *Igniter* (*CI*) dapat didefinisikan berupa *framework PHP* yang sederhana dan mudah digunakan dalam membuat aplikasi berbasis *web* yang dibuat sesuai acuan *Model*, *View* dan *Controller (MVC).*

Adapun pendapat (Daqiqil, 2011), *CodeIgniter* merupakan sebuah *framework* PHP yang dapat mempemudah pengembang untuk membangun aplikasi *web*. MVC merupakan sebuah pakem/teknik pemograman yang memisahkan antara *bisnis* *logic* (alur pikir), *data* *logic* (penyimpanan data) dan *presentation* *logic* (antarmuka aplikasi) atau dapat disederhanakan menjadi suatu pemisah antara desain, data, dan proses.

Sedangkan menurut (Raharjo B. , 2015) *Codeigniter* adalah *framework* *web* untuk bahasa pemrograman *PHP,* yang dibuat oleh Rick Ellis pada tahun 2006, Rick Ellis sendiri adalah seorang penemu dan pendiri EllisLab. Dapat disimpulkan bahwa *Codeigniter* merupakan *framework* *PHP,* sederhana, mudah digunakan dan cukup handal untuk membuat aplikasi berbasis *web* yang dibuat sesuai acuan komponen yang berupa *Model*, *View*, dan *Controller* *(MVC).*

## *JavaScript*

Secara umum JavaScript adalah sebuah bahasa pemrograman tingkat tinggi yang diinterpretasikan secara dinamis. JavaScript digunakan untuk memberikan interaktivitas pada halaman web, seperti menangani peristiwa pengguna, memanipulasi DOM (Document Object Model), dan berkomunikasi dengan server melalui teknologi seperti Ajax.

Menurut (Nurcahyono, 2013:1-2) menjelaskan untuk mempelajari pemrograman JavaScript, ada dua aplikasi yang diperlukan yaitu:

Teks Editor digunakan untuk menuliskan kode-kode JavaScript. Teks Editor yang dapat digunakan antara lain notepad, editplus, dreamweaver.

Web Browser digunakan untuk menampilkan halaman web. Web browser yang digunakan harus mendukung JavaScript. Browser yang dapat digunakan adalah internet explorer, firefox dan lainnya.

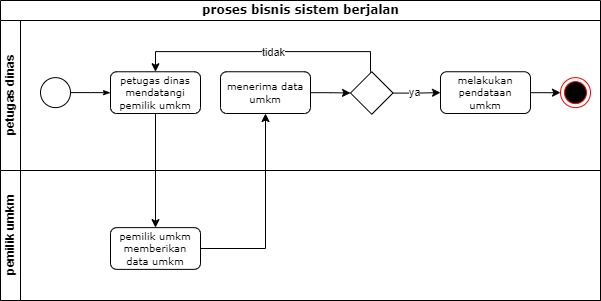
Dari definisi diatas penulis menyimpulkan bahwa Javascript merupakan bahasa pemrograman web yang dapat bekerja dan dijalankan dengan menggunakan web browser.

# Proses Bisnis dan Arsitektur Sistem

## Proses Bisnis

### Proses Bisnis Sistem Berjalan

Proses bisnis sistem berjalan dapat dijelaskan pada Gambar III.1

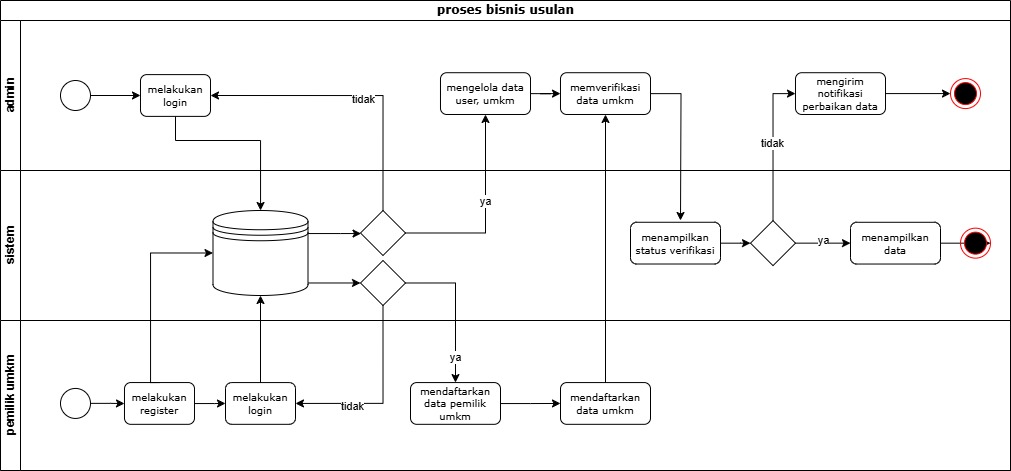


Gambar III. 1 Gambar Proses Bisnis Sistem Berjalan

Berdasarkan gambar diatas, proses bisnis sistem berjalan dimulai ketika petugas dinas mendatangi pemilik UMKM secara langsung untuk mengumpulkan informasi terkait. Pemilik UMKM memberikan data UMKM mereka kepada petugas dinas, yang meliputi profil usaha, produk yang dihasilkan, serta kebutuhan yang mungkin dimiliki oleh UMKM tersebut. Data yang diterima kemudian diverifikasi oleh petugas dinas untuk memastikan kebenaran dan keakuratan informasi. Setelah diverifikasi, petugas dinas mencatat informasi yang diberikan oleh pemilik UMKM dan memasukkannya ke dalam format yang sesuai, seperti file Excel.

### Proses Bisnis Usulan

Aplikasi yang sedang dirancang adalah aplikasi Sistem Informasi UMKM berbasis Website yang bisa diakses oleh komputer maupun smartphone yang menjadi keuntungan baru dari metode sistem sebelumnya, dengan mobilitas yang diberikan, menjadikan proses yang dilakukan sangat mudah dan data UMKM pun menjadi terorganisir dengan baik. Proses bisnis usulan dapat dijelaskan pada Gambar III.2



Gambar III. 2 Gambar Proses Bisnis Sistem Usulan

Berdasarkan gambar diatas, proses bisnis usulan sistem informasi ini dimulai dengan admin melakukan login menggunakan username dan password yang telah ditentukan. Setelah berhasil login, admin memiliki akses untuk mengelola data user, UMKM (Usaha Mikro Kecil dan Menengah). Salah satu fitur yang dimiliki admin adalah verifikasi, yang memungkinkan untuk memverifikasi data UMKM yang masuk. Jika data UMKM memenuhi kriteria yang telah ditetapkan, admin akan menyetujuinya. Pemilik UMKM kemudian dapat melakukan registrasi dalam aplikasi dan login menggunakan kredensial yang diberikan. Setelah login, mereka dapat menambahkan data UMKM mereka. Jika data UMKM tersebut telah diverifikasi oleh admin, Sistem akan menampilkan data UMKM yang terdaftar. Dengan demikian, sistem ini menyediakan platform yang terorganisir dan terverifikasi untuk administrasi data UMKM tersebut.

## Arsitektur Sistem

### Arsitektur Sistem Usulan

Arsitektur sistem usulan dapat dijelaskan pada Gambar III.3



Gambar III. 3 Gambar Arsitektur Sistem Usulan

Berdasarkan gambar arsitektur sistem usulan berbasis web ini, terdapat dua aktor utama yaitu admin dan pemilik UMKM, admin dan pemilik umkm mengakses sistem melalui web browser yang terhubung ke sebuah server. Ketika admin atau pemilik UMKM mengirimkan permintaan melalui browser, server menerima permintaan tersebut, memprosesnya sesuai dengan permintaan yang diajukan, dan mengirimkan respons kembali ke klien. Server juga berperan dalam menjaga keamanan dan mengatur hak akses, memastikan bahwa setiap aktor hanya memiliki akses ke fitur dan data yang sesuai dengan peran mereka. Semua data yang terkait dengan sistem, seperti informasi pengguna, data UMKM, dan produk, disimpan dalam sebuah database yang terhubung dengan server. Dengan demikian, melalui interaksi antara klien, server, dan database, sistem dapat beroperasi secara efisien untuk memenuhi kebutuhan admin dan pemilik UMKM dalam mengelola data dan operasional UMKM.

# HASIL DAN PEMBAHASAN

## Daftar Kebutuhan Fungsional dan Non Fungsional

### Daftar Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi tentang proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan fungsional juga

berisikan tentang informasi-informasi apa saja yang harus ada dan dihasilkan oleh

sistem.

Pada table IV.1 Berikut ini adalah kebutuhan fungsional dari perancangan aplikasi yang akan dibuat adalah sebagai berikut.

Tabel VI. 1 Table Kebutuhan Fungsional

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Nomor *Requirement* | Nama *Requirement* | Keterangan | Pengguna |
| 1 | Req-01 | Login | Digunakan untuk proses masuk kedalam sistem informasi UMKM dengan memasukan *username* dan *password* oleh admin/ Pemilik UMKM untuk mengakses sesuai hak akses. | Admin & Pemilik UMKM |
| 2 | Req-02 | Kelola Data Users | Digunakan untuk memasukkan, melihat, mengubah dan menghapus data Users. | Admin & Pemilik UMKM |
| 3 | Req-03 | Kelola Data UMKM | Digunakan untuk memasukkan, melihat, mengubah dan menghapus data UMKM. | Admin |
| 4 | Req-04 | Kelola Pemilik UMKM | Digunakan untuk memasukkan, melihat, mengubah dan menghapus data Pemilik UMKM. | Admin |
| 5 | Req-05 | Kelola Data Promosi | Digunakan untuk memasukkan, melihat, mengubah dan menghapus data Promosi. | Admin & Pemilik UMKM |
| 6 | Req-06 | Kelola Data Platform | Digunakan untuk memasukkan, melihat, mengubah dan menghapus data Platform. | Admin & Pemilik UMKM |
| 7 | Req-07 | Menambahkan Data UMKM | Digunakan untuk memasukkan data UMKM | Admin & Pemilik UMKM |
| 8 | Req-08 | Menambahkan Data Pemilik UMKM | Digunakan untuk memasukkan data pemilik UMKM | Admin & Pemilik UMKM |
| 9 | Req-09 | Melihat Data UMKM | Digunakan untuk memasukkan data UMKM | Admin & Pemilik UMKM |
| 10 | Req-10 | Melihat Data Pemilik UMKM | Digunakan untuk melihat data pemilik UMKM | Admin & Pemilik UMKM |
| 11 | Req-11 | Logout | Digunakan untuk mengakhiri proses dari sistem | Admin & Pemilik UMKM |

### Daftar Kebutuhanm Non Fungsional

Analisa kebutuhan non-fungsional merupakan analisa yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi non-fungsional juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan mulai dari sistem dibangun sampai diimplementasikan.

Pada analisis kebutuhan non-fungsional dijelaskan analisis kebutuhan perangkat keras, perangkat lunak, dan juga analisis pengguna.

1. Spesifikasi Perangkat Keras *(Hardware)*

Spesifikasi Perangkat Keras *(hardware)* computer yang digunakan dalam Pembangunan sistem informasi UMKM akan dijelaskan pada table IV.2.

Tabel VI. 2 Table Kebutuhan Non-Fungsional Hardware

|  |  |  |
| --- | --- | --- |
| No | Perangkat Keras | Spesifikasi |
| 1. | Processor | * Intel Core i5-Gen10 * AMD Ryzen 7-5000U |
| 2 | RAM | * 20 GB * 16 GB |
| 3 | Harddisk | * SSD 512 GB * SSD 1 TB |
| 4 | VGA | * nvidia geforce * Integrated AMD * Radeon Graphics |

1. Spesifikasi Perangkat Lunak *(Software)*

Spesifikasi Perangkat Lunak *(software)* computer yang digunakan dalam Pembangunan sistem informasi UMKM akan dijelaskan pada table IV.3.

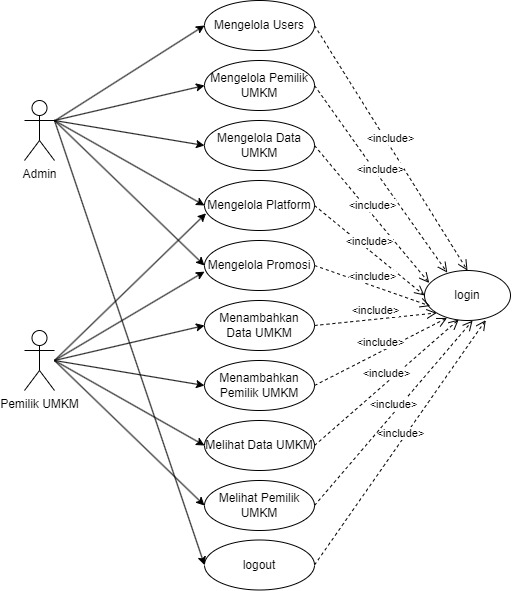
Tabel VI. 3 Table Kebutuhan Non-Fungsional Software

|  |  |  |
| --- | --- | --- |
| No | Perangkat Lunak | Spesifikasi |
| 1. | Sistem Operasi | * Wndows 11 (64bit) * Wndows 10 (64bit) |
| 2 | Bahasa Pemograman | * PHP * JavaScript |
| 3 | DBMS | * MySql |
| 4 | Web Server | * Apache |
| 5 | Interface | * CI |

## Design System

### Diagram Use case

Gambar VI.1 merupakan *Use Case Diagram* pada sistem informasi UMKM.



Gambar IV. 1 Use Case Diagram

Berikut adalah deskripsi pendefinisian aktor pada sistem informasi UMKM. Akan dijelaskan pada table IV.4.

Tabel VI. 4 Deskripsi Pendefinisian Aktor Use Case

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1. | Admin/ Petugas Disperidag | Orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan user, pemilik UMKM, dan pengelolaan data UMKM |
| 2. | Pemilik UMKM/ Pengunjung | Orang yang diperbolehkan memasukan data pemilik UMKM, memasukan data UMKM. Sedangkan pengunjung hanya memiliki akses melihat data UMKM tanpa memiliki hak akses untuk memasukan data UMKM, data Pemilik UMKM. |

### Skenario Use Case

Berikut adalah Skenario Use Case jalannya masing-masing use case yang telah didefinisikan sebelumnya.

1. Skenario Use Case Login

Deskripsi skenario Use Case Login akan dijelaskan pada table IV.5.

Tabel VI. 5 Skenario Use Case Login

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
| 1. Memasukan *username* dan *password* |  |
|  | 1. Memeriksa valid tidaknya data masukan dengan memeriksa table users |
|  | 1. Masuk ke aplikasi pengelolaan data UMKM, pemilik UMKM, dan Promosi |
| Skenario Alternatif | |
| 1. Memasukan *username* dan *password* |  |
|  | 1. Memeriksa valid tidaknya data masukan dengan memeriksa table users |
|  | 1. Menampilkan pesan login tidak valid |
| 1. Memasukan usernamedan *password* yang valid |  |
|  | 1. Memeriksa valid tidaknya data masukan dengan memeriksa table users |
|  | 1. Masuk ke aplikasi pengelolaan data UMKM, pemilik UMKM, dan Promosi |

1. Skenario Use Case Logout

Deskripsi skenario Use Case Logout akan dijelaskan pada table IV.6.

Tabel VI. 6 Skenario Use Case Logout

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
| 1. Memilih menu logout |  |
|  | 1. Melakukan logout |

1. Skenario Use Case memeriksa status Login

Deskripsi skenario Use Case memeriksa status Login akan dijelaskan pada table IV.7.

Tabel VI. 7 Skenario Use Case Memeriksa Statun Login

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
|  | 1. Memeriksa ke variable session sebagai penanda *login* apakah admin sudah *login* |
|  | 1. Mengembalikan status *login,* sudah login atau belum |

1. Memasukan pemilik UMKM

Deskripsi skenario Use Case memasukan pemilik UMKM akan dijelaskan pada table IV.8.

Tabel VI. 8 Skenario Use Case memasukan pemilik UMKM

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
|  | 1. Memeriksa status login |
| 1. Memasukan data pemilik UMKM |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menyimpan data pemilik UMKM ke basis data |
|  | 1. Menampilkan pesan sukses disimpan |
| Skenario Alternatif | |
|  | 1. Memeriksa status login |
| 1. Memasukan data pemilik UMKM |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menampilkan pesan bahwa data masukan tidak *valid* |
| 1. Memperbaiki data masukan yang tidak *valid* |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menyimpan data pemilik UMKM ke basis data |
|  | 1. Menampilkan pesan sukses disimpan |

1. Mengubah pemilik UMKM

Deskripsi skenario Use Case mengubah pemilik UMKM akan dijelaskan pada table IV.9.

Tabel VI. 9 Skenario Use Case Mengubah Pemilik UMKM

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
|  | 1. Memeriksa status login |
| 1. Memasukan kata kunci pada fitur pencarian |  |
|  | 1. Mencari data pemilik UMKM yang akan diubah |
|  | 1. Menampilkan data pemilik UMKM yang dicari |
| 1. Memilih data pemilik UMKM yang akan diubah |  |
|  | 1. Menampilkan semua kolom data pemilik UMKM yang akan diubah |
| 1. Pengubah data Pemilik UMKM |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menyimpan data yang telah diubah ke basis data |
|  | 1. Menampilkan pesan bahwa data berhasil diubah |
| Skenario Alternatif | |
|  | 1. Memeriksa status login |
| 1. Memasukan kata kunci pada fitur pencarian |  |
|  | 1. Mencari data pemilik UMKM yang akan diubah |
|  | 1. Menampilkan data pemilik UMKM yang dicari |
| 1. Memilih data pemilik UMKM yang akan diubah |  |
|  | 1. Menampilkan semua kolom data pemilik UMKM yang akan diubah |
| 1. Pengubah data Pemilik UMKM |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menampilkan bahwa data pemilik UMKM tidak valid |
| 1. Memperbaiki data masukan yang diubah dan tidak *valid* |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menyimpan data yang telah diubah ke basis data |
|  | 1. Menampilkan pesan bahwa data berhasil diubah |

1. Menghapus pemilik UMKM

Deskripsi skenario Use Case menghapus pemilik UMKM akan dijelaskan pada table IV.10.

Tabel VI. 10 Skenario Use Case Menghapus Pemilik UMKM

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
|  | 1. Memeriksa status login |
| 1. Memasukan kata kunci pada fitur pencarian |  |
|  | 1. Mencari data pemilik UMKM yang akan dihapus |
|  | 1. Menampilkan data pemilik UMKM yang dicari |
| 1. Memilih data pemilik UMKM yang akan dihapus |  |
|  | 1. Menampilkan semua kolom data pemilik UMKM yang akan dihapus |
| 1. Mengklik data setuju data dihapus |  |
|  | 1. Menghapus data pemilik UMKM dari basis data |
|  | 1. Menampilkan pesan bahwa data berhasil dihapus |
| Skenario Alternatif | |
|  | 1. Memeriksa status login |
| 1. Memasukan kata kunci pada fitur pencarian |  |
|  | 1. Mencari data pemilik UMKM yang akan dihapus |
|  | 1. Menampilkan data pemilik UMKM yang dicari |
| 1. Memilih data pemilik UMKM yang akan dihapus |  |
|  | 1. Menampilkan semua kolom data pemilik UMKM yang akan dihapus |
| 1. Mengklik pilihan data tidak setuju data dihapus |  |
|  | 1. Kembali ke form pencarian pemilik UMKM |

1. Memasukan data UMKM

Deskripsi skenario Use Case memasukan data UMKM akan dijelaskan pada table IV.11.

Tabel VI. 11 Skenario Use Case Memasukkan Data UMKM

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
|  | 1. Memeriksa status login |
| 1. Memasukan data UMKM |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menyimpan data UMKM ke basis data |
|  | 1. Menampilkan pesan sukses disimpan |
| Skenario Alternatif | |
|  | 1. Memeriksa status login |
| 1. Memasukan data UMKM |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menampilkan pesan bahwa data masukan tidak *valid* |
| 1. Memperbaiki data masukan yang tidak *valid* |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menyimpan data UMKM ke basis data |
|  | 1. Menampilkan pesan sukses disimpan |

1. Mengubah data UMKM

Deskripsi skenario Use Case mengubah data UMKM akan dijelaskan pada table IV.12.

Tabel VI. 12 Skenario Use Case Mengubah Data UMKM

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
|  | 1. Memeriksa status login |
| 1. Memasukan kata kunci pada fitur pencarian |  |
|  | 1. Mencari data UMKM yang akan diubah |
|  | 1. Menampilkan data UMKM yang dicari |
| 1. Memilih data UMKM yang akan diubah |  |
|  | 1. Menampilkan semua kolom data UMKM yang akan diubah |
| 1. Pengubah data UMKM |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menyimpan data yang telah diubah ke basis data |
|  | 1. Menampilkan pesan bahwa data berhasil diubah |
| Skenario Alternatif | |
|  | 1. Memeriksa status login |
| 1. Memasukan kata kunci pada fitur pencarian |  |
|  | 1. Mencari data UMKM yang akan diubah |
|  | 1. Menampilkan data UMKM yang dicari |
| 1. Memilih data UMKM yang akan diubah |  |
|  | 1. Menampilkan semua kolom data UMKM yang akan diubah |
| 1. Pengubah data UMKM |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menampilkan bahwa data UMKM tidak valid |
| 1. Memperbaiki data masukan yang diubah dan tidak *valid* |  |
|  | 1. Memeriksa *valid* tidaknya data masukan |
|  | 1. Menyimpan data yang telah diubah ke basis data |
|  | 1. Menampilkan pesan bahwa data berhasil diubah |

1. Menghapus data UMKM

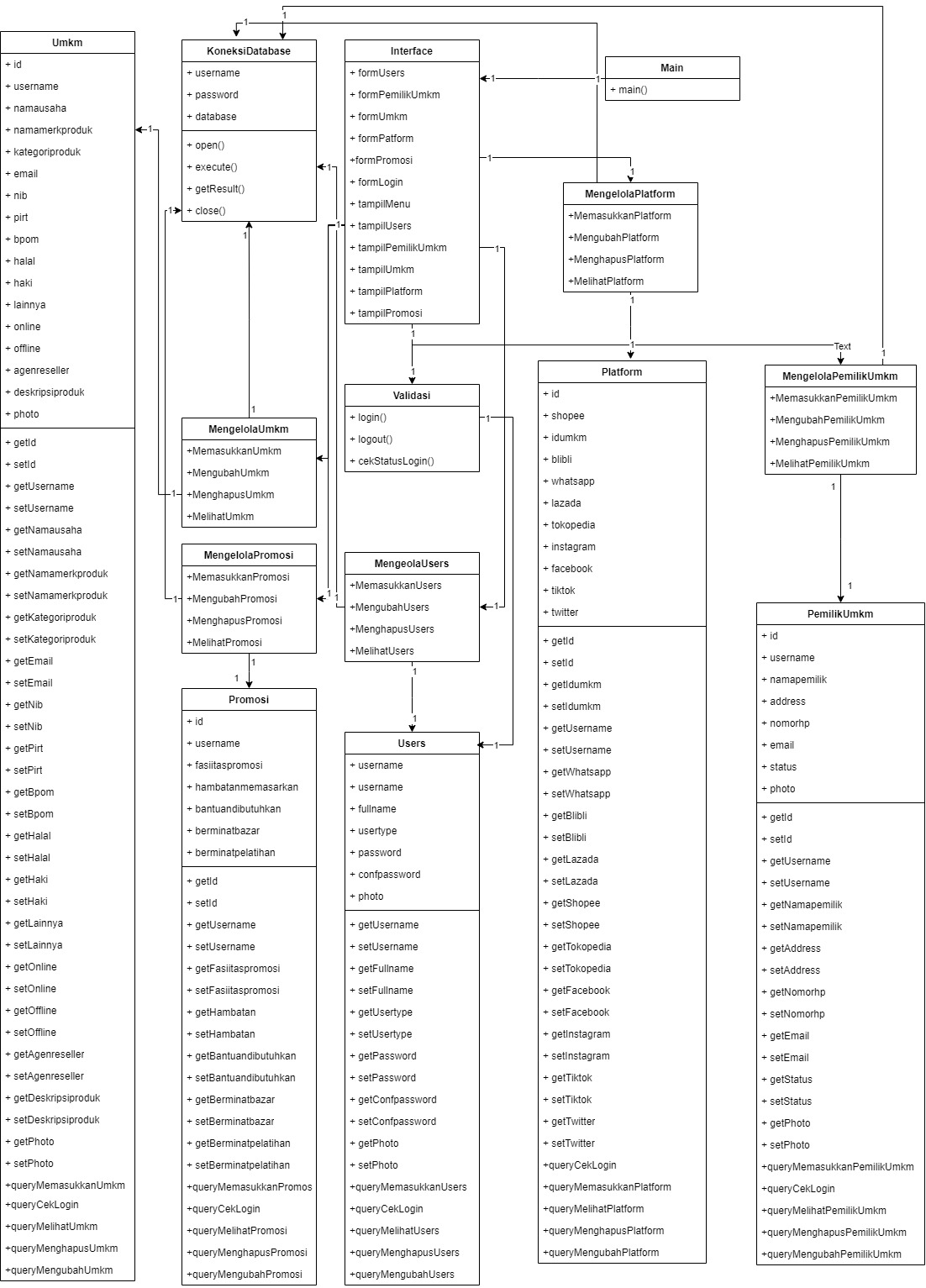
Deskripsi skenario Use Case menghapus data UMKM akan dijelaskan pada table IV.13.

Tabel VI. 13 Skenario Use Case Menghapus Data UMKM

|  |  |
| --- | --- |
| Aksi Aktor | Reaksi Sistem |
| Skenario Normal | |
|  | 1. Memeriksa status login |
| 1. Memasukan kata kunci pada fitur pencarian |  |
|  | 1. Mencari data UMKM yang akan dihapus |
|  | 1. Menampilkan data UMKM yang dicari |
| 1. Memilih data UMKM yang akan dihapus |  |
|  | 1. Menampilkan semua kolom data UMKM yang akan dihapus |
| 1. Mengklik data setuju data dihapus |  |
|  | 1. Menghapus data UMKM dari basis data |
|  | 1. Menampilkan pesan bahwa data berhasil dihapus |
| Skenario Alternatif | |
|  | 1. Memeriksa status login |
| 1. Memasukan kata kunci pada fitur pencarian |  |
|  | 1. Mencari data UMKM yang akan dihapus |
|  | 1. Menampilkan data UMKM yang dicari |
| 1. Memilih data UMKM yang akan dihapus |  |
|  | 1. Menampilkan semua kolom data UMKM yang akan dihapus |
| 1. Mengklik pilihan data tidak setuju data dihapus |  |
|  | 1. Kembali ke form pencarian data UMKM |

### *Class Diagram*

Diagram Class ialah alat untuk membantu menggambarkan serta memvisualisasikan struktur kelas yang terdapat dalam suatu sistem. Pada gambar IV.2. berikut adalah *Class Diagram* sistem informasi UMKM.

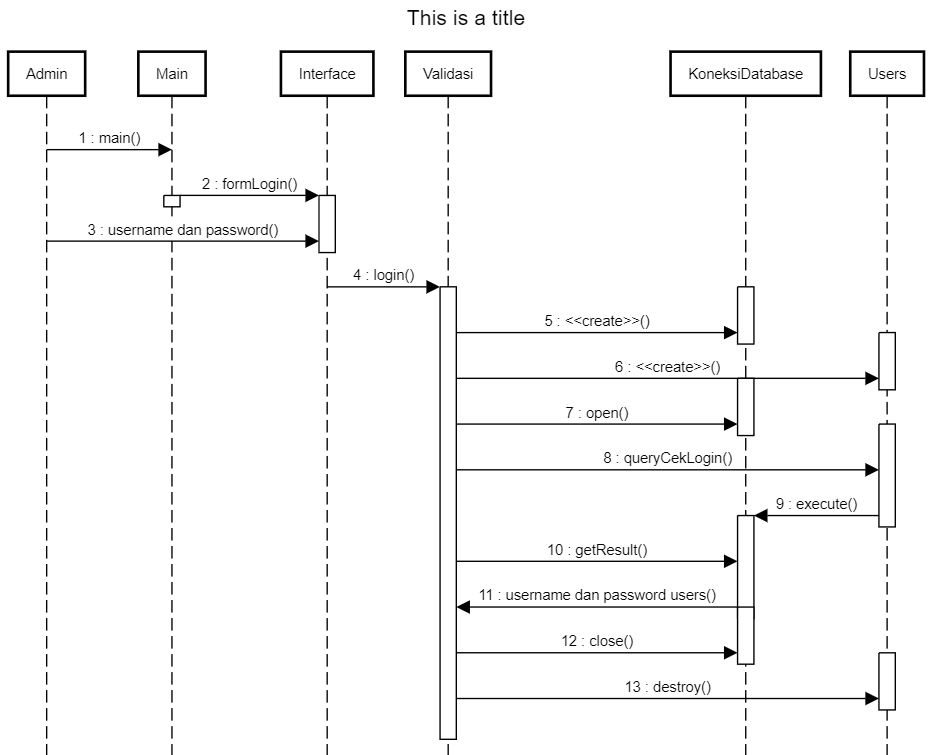


Gambar IV. 2 Class Diagram

### *Sequence Diagram*

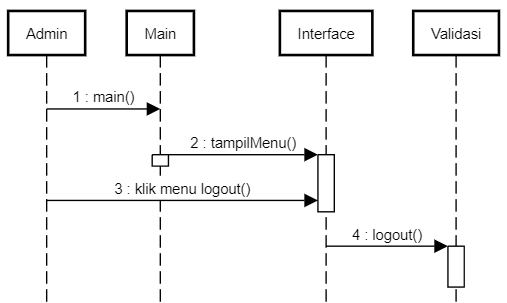
Diagram Sequence aplikasi dari sebuah system informasi UMKM akan dijelaskan pada gambar dibawah ini.

1. *Sequence diagram login*



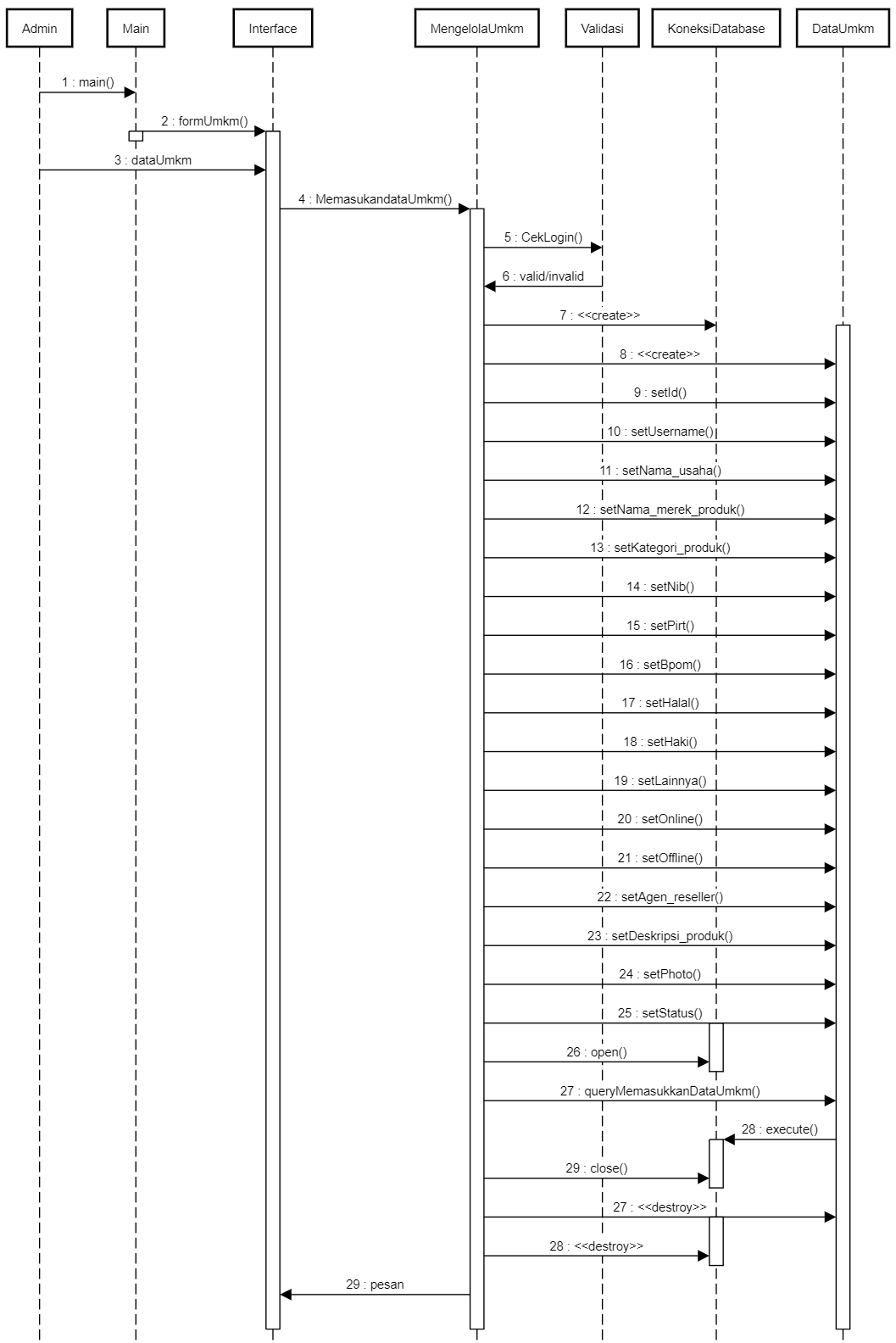
Gambar IV. 3 Sequence Diagram Login

1. *Sequence* *diagram logout*



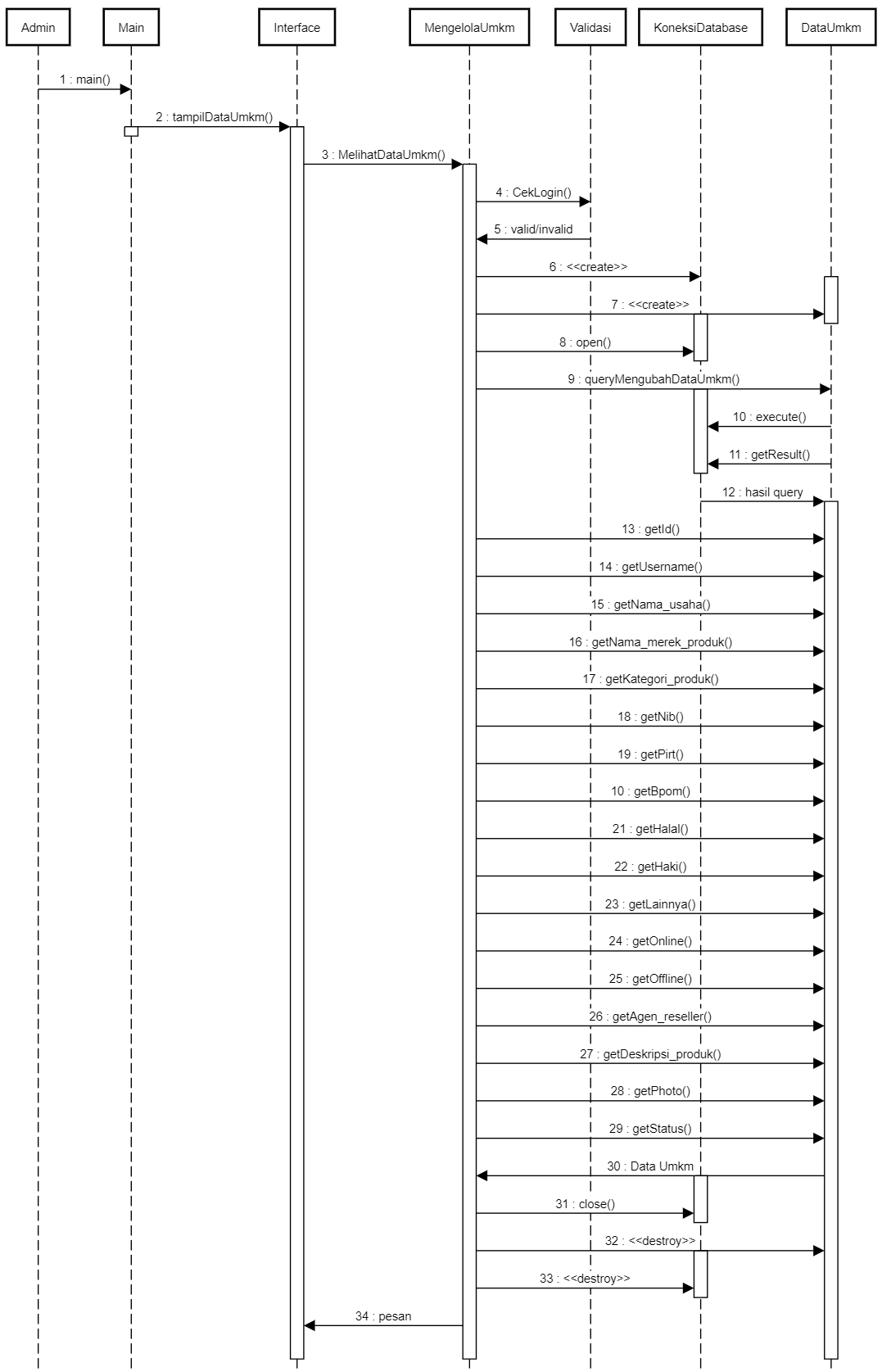
Gambar VI. 1 Sequence Diagram Logout

1. *Sequence* *Diagram* Memasukkan Data UMKM



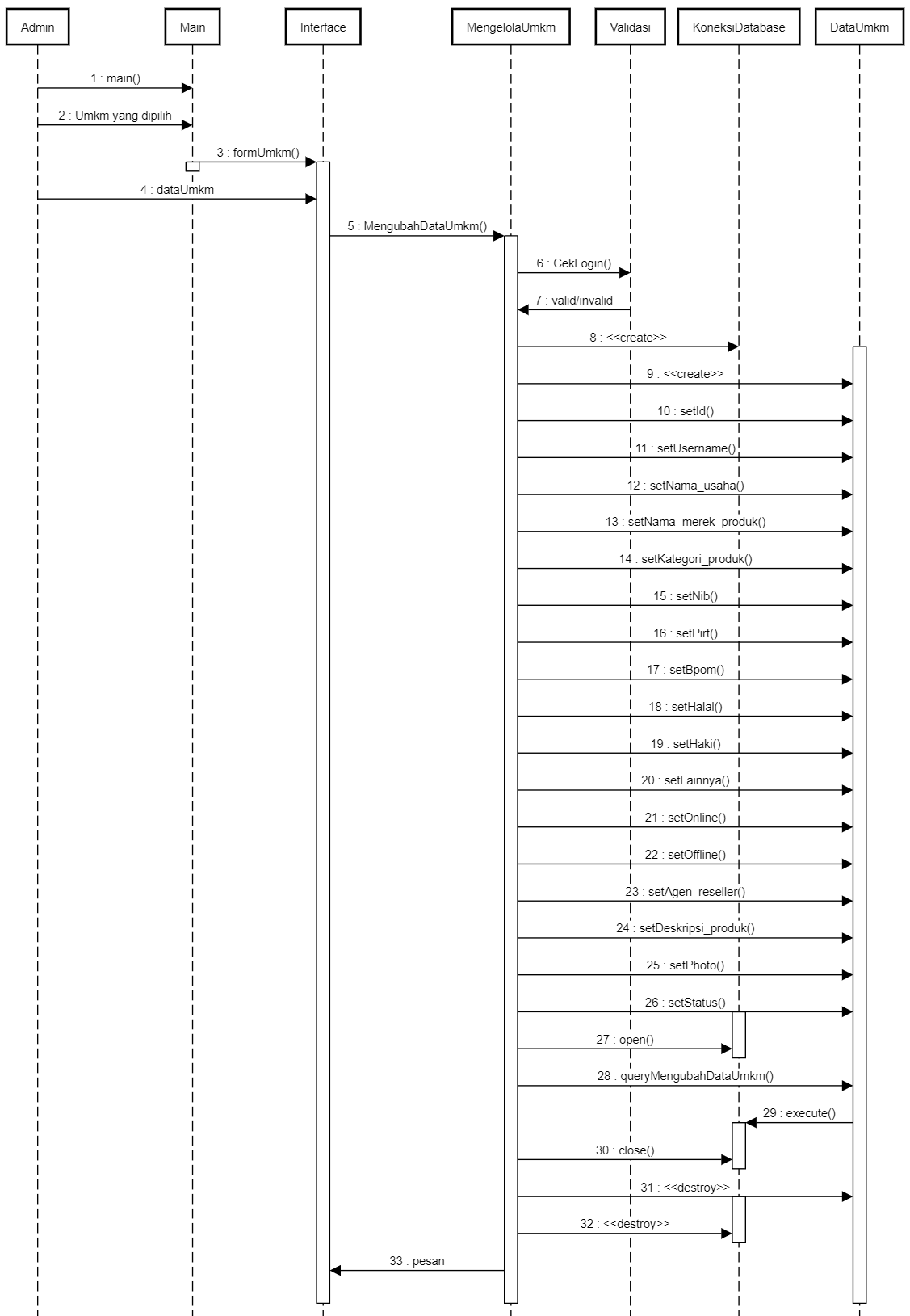
Gambar IV. 4 Sequence Diagram Memasukan Data UMKM

1. *Sequence* *Diagram* Melihat Data UMKM



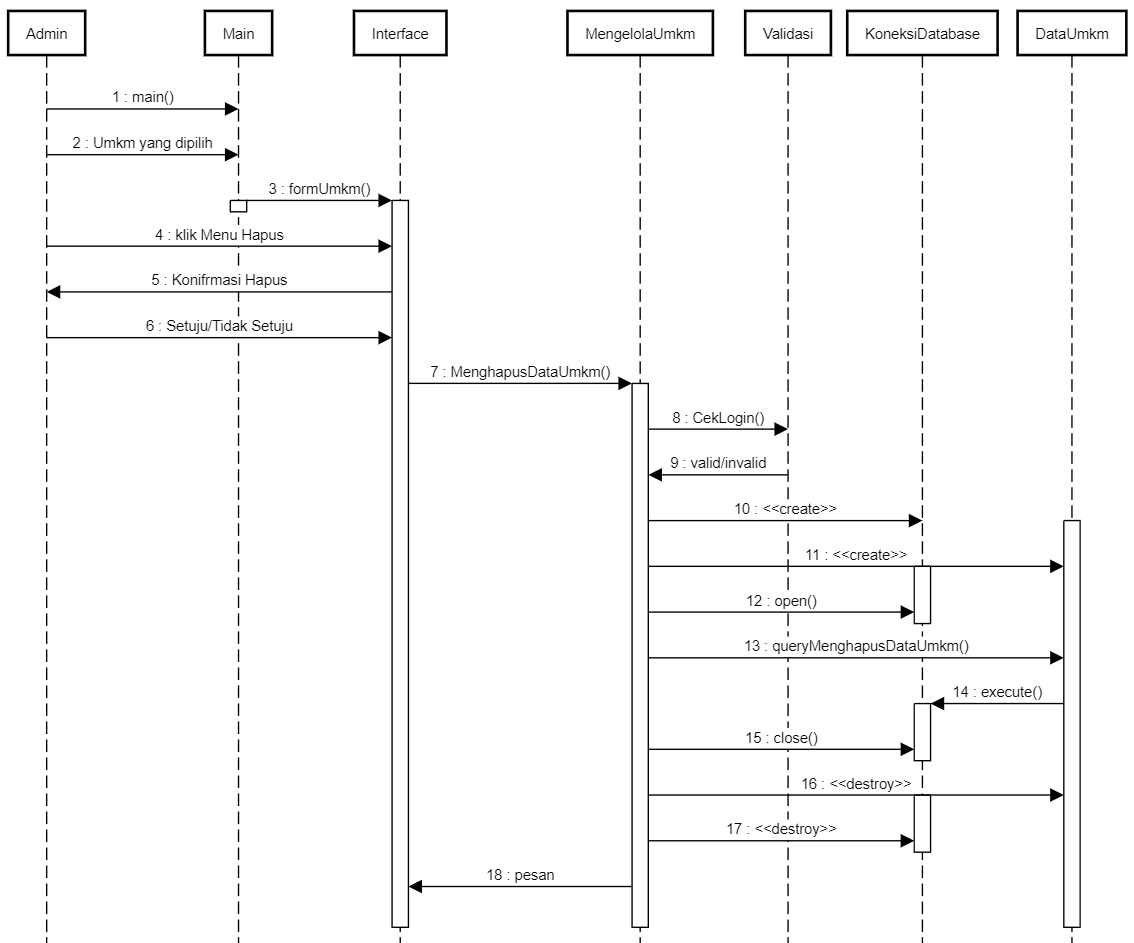
Gambar IV. 5 Sequence Diagram Melihat Data UMKM

1. *Sequence* *Diagram* Mengubah Data UMKM



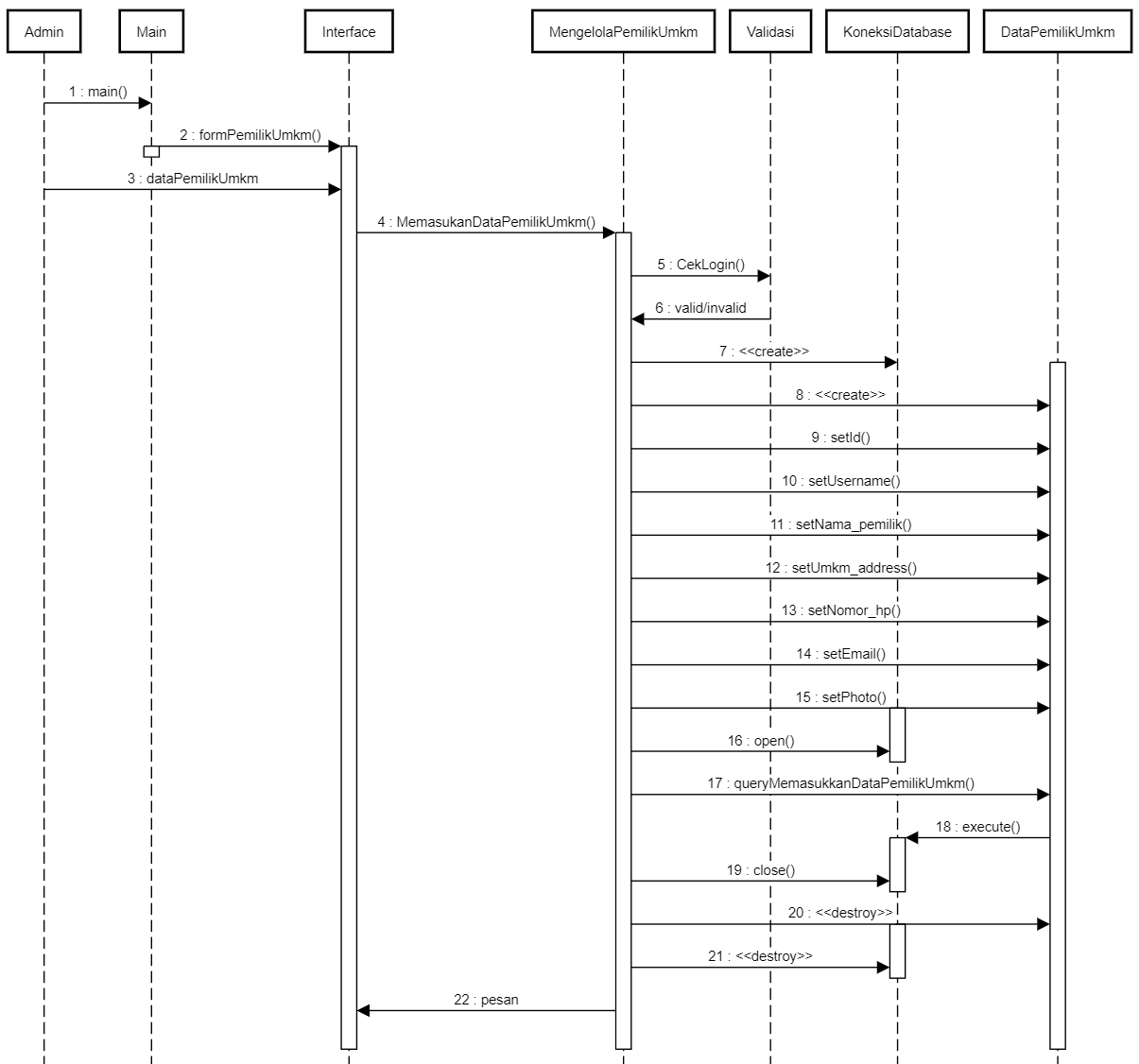
Gambar IV. 6 Sequence Diagram Mengubah Data UMKM

1. *Sequence* *Diagram* Menghapus Data UMKM



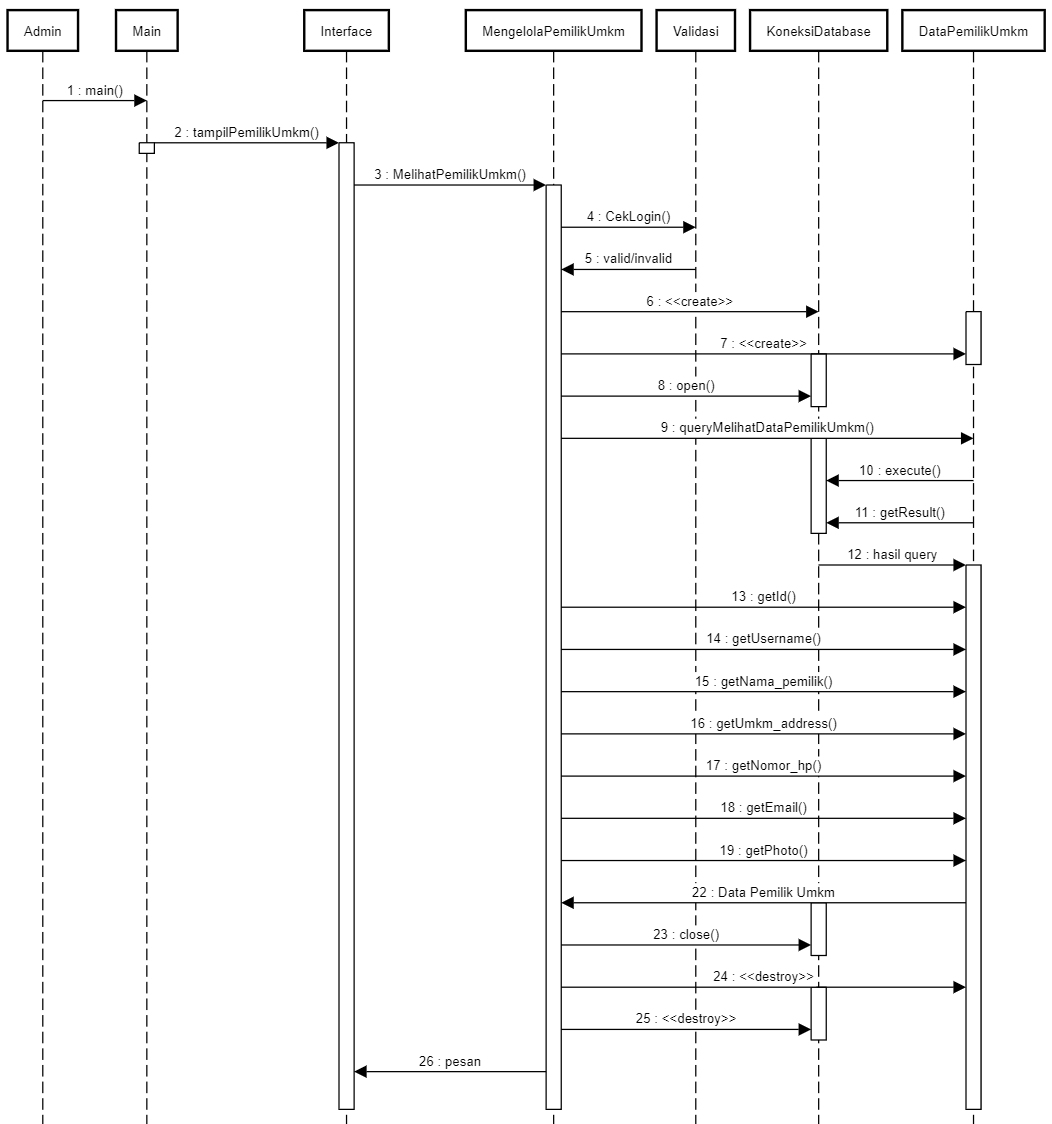
Gambar IV. 7 Sequence Diagram Menghapus Data UMKM

1. *Sequence* *Diagram* Memasukkan Pemilik UMKM



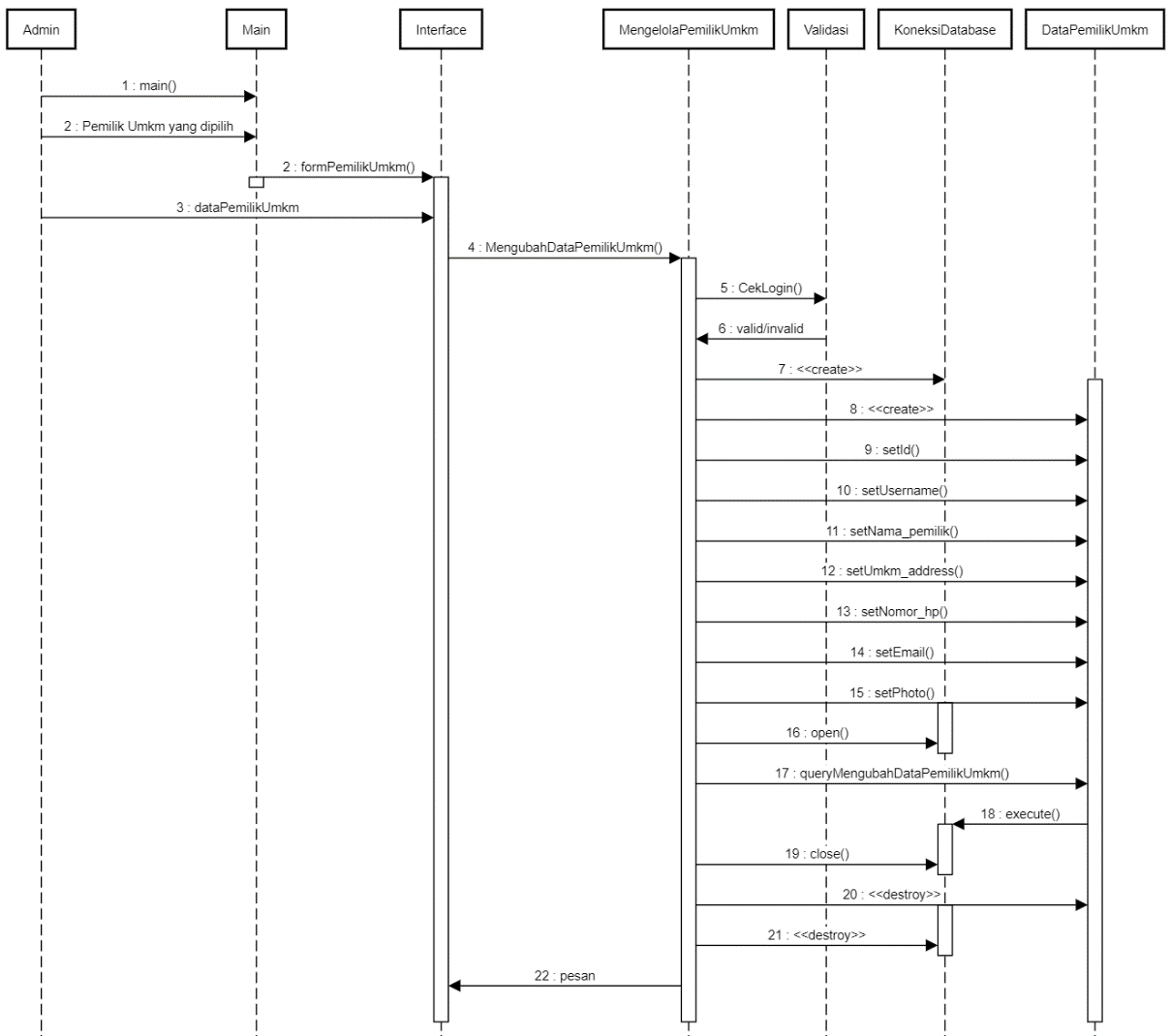
Gambar IV. 8 Sequence Diagram Memasukkan Pemilik UMKM

1. *Sequence* *Diagram* Melihat Pemilik UMKM



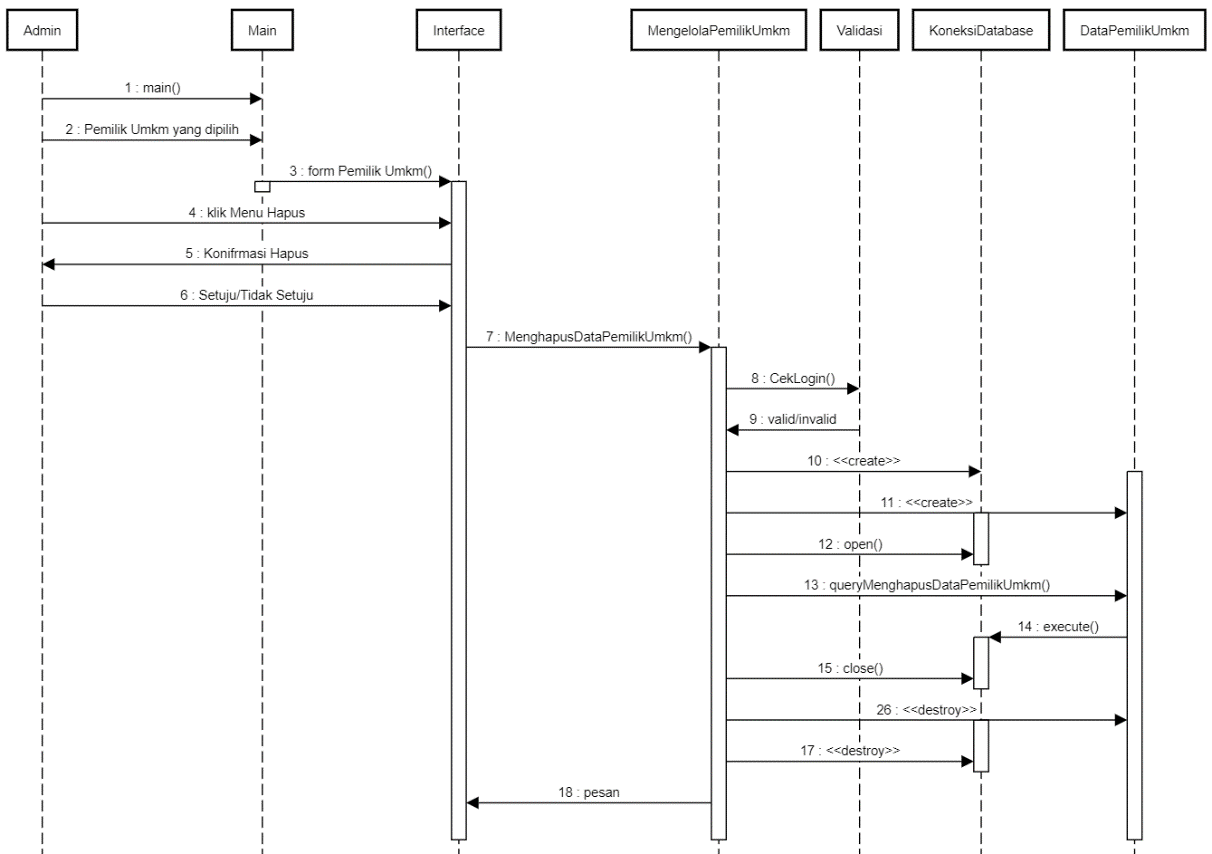
Gambar IV. 9 Sequence Diagram Melihat Pemilik UMKM

1. *Sequence* *Diagram* Mengubah Pemilik UMKM



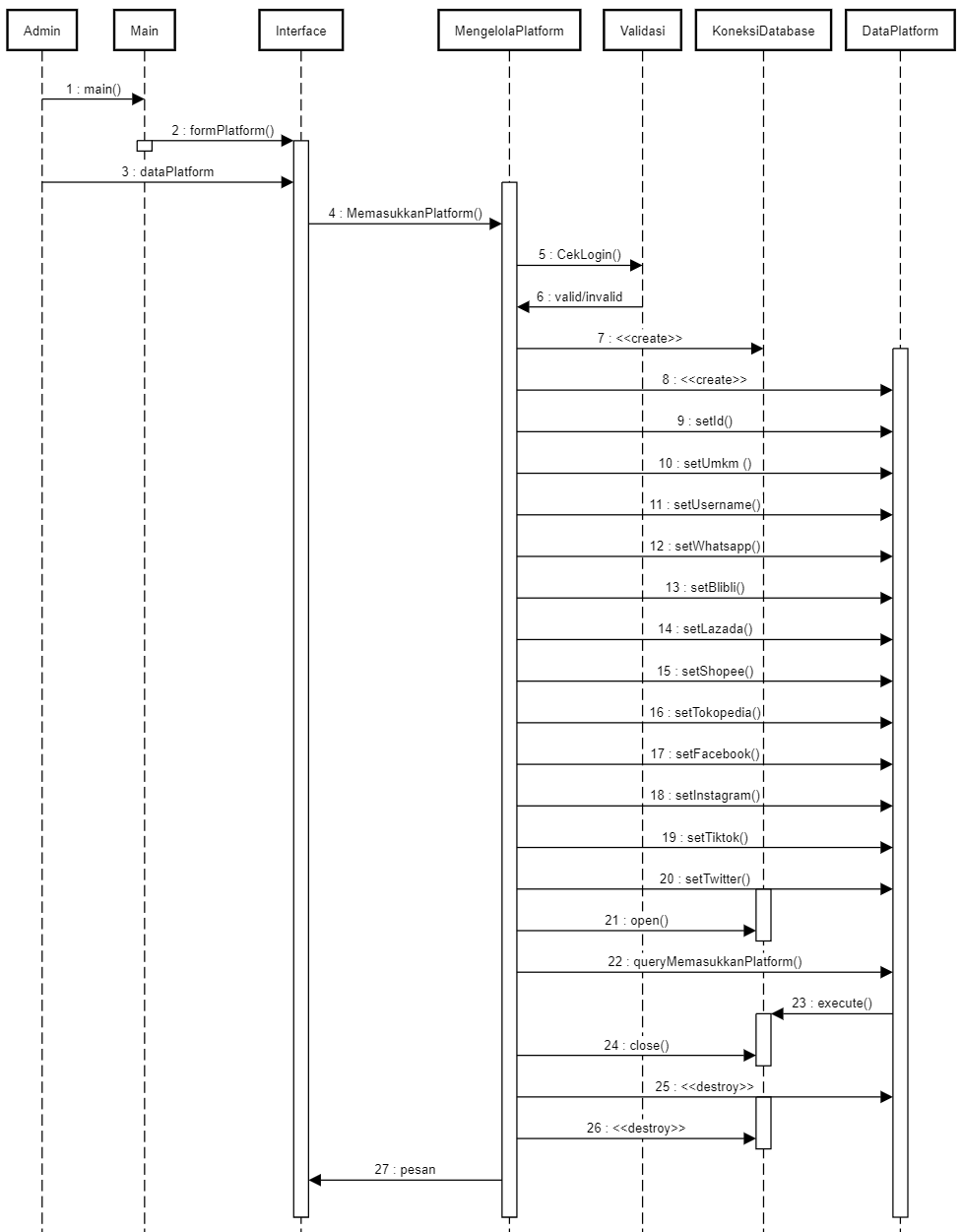
Gambar IV. 10 Sequence Diagram Mengubah Pemilik UMKM

1. *Sequence* *Diagram* Menghapus Pemilik UMKM



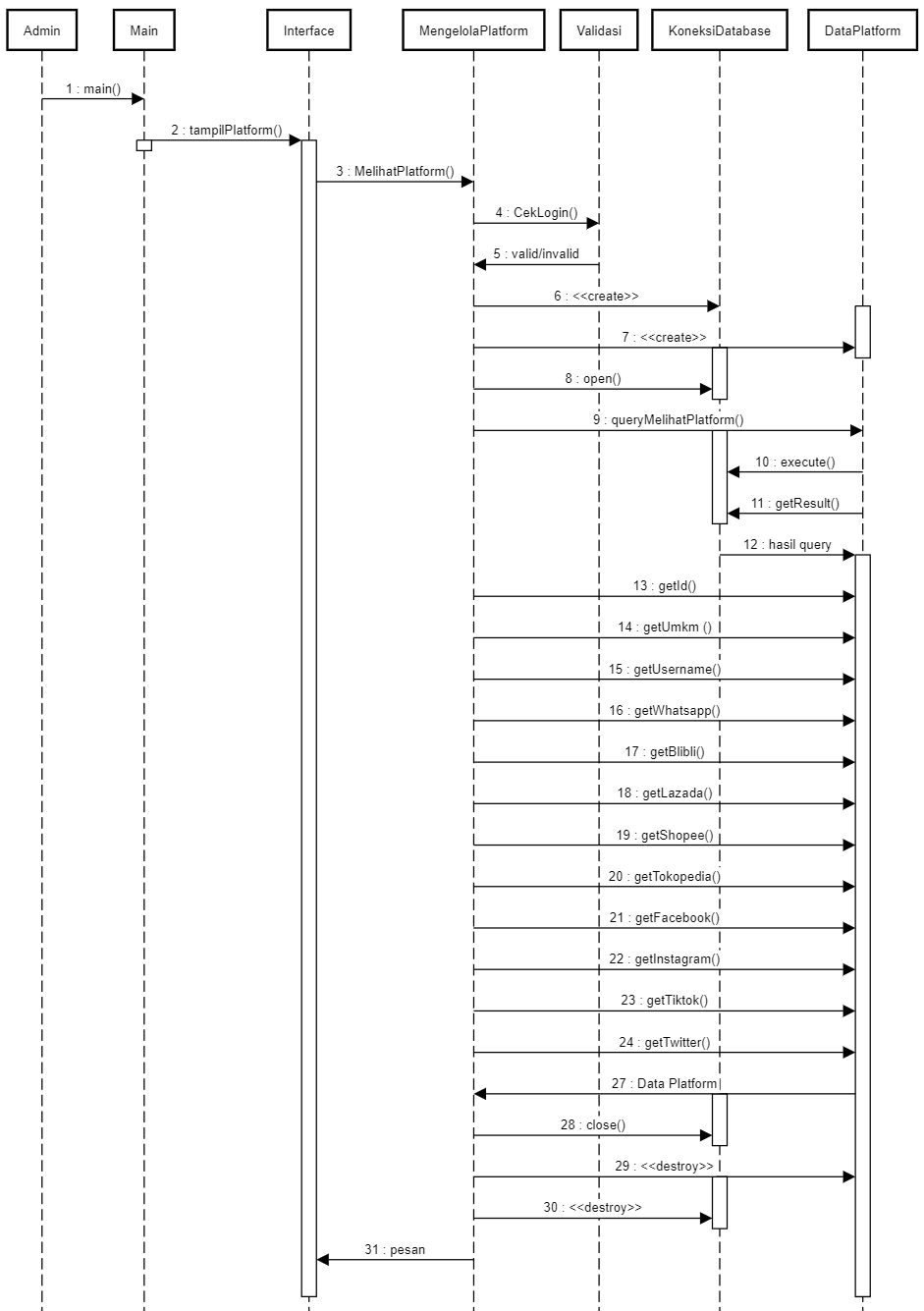
Gambar IV. 11 Sequence Diagram Menghapus Pemilik UMKM

1. *Sequence* *Diagram* Memasukkan Platform



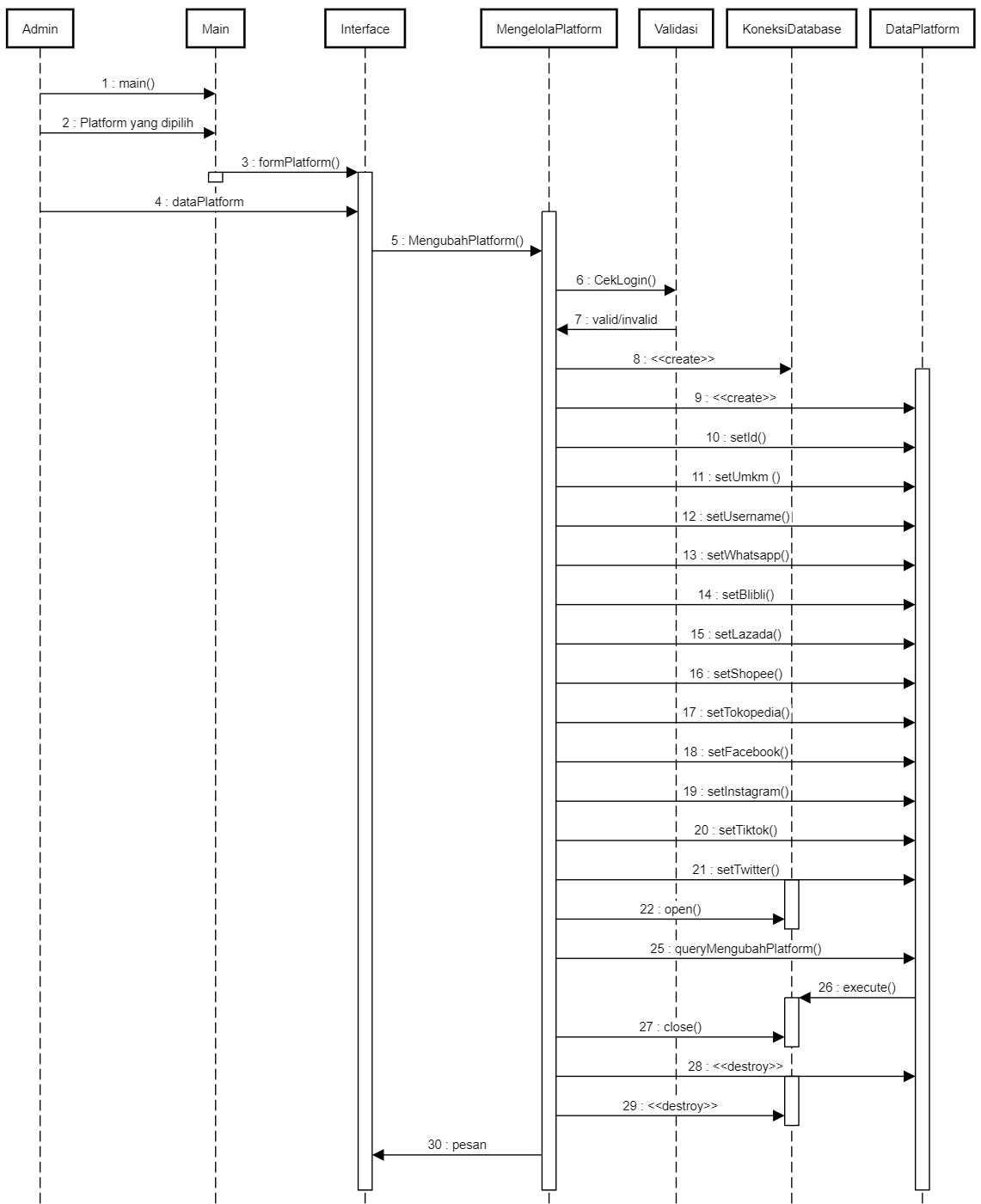
Gambar IV. 12 Sequence Diagram Memasukkan Platform

1. *Sequence* *Diagram* Melihat Platform



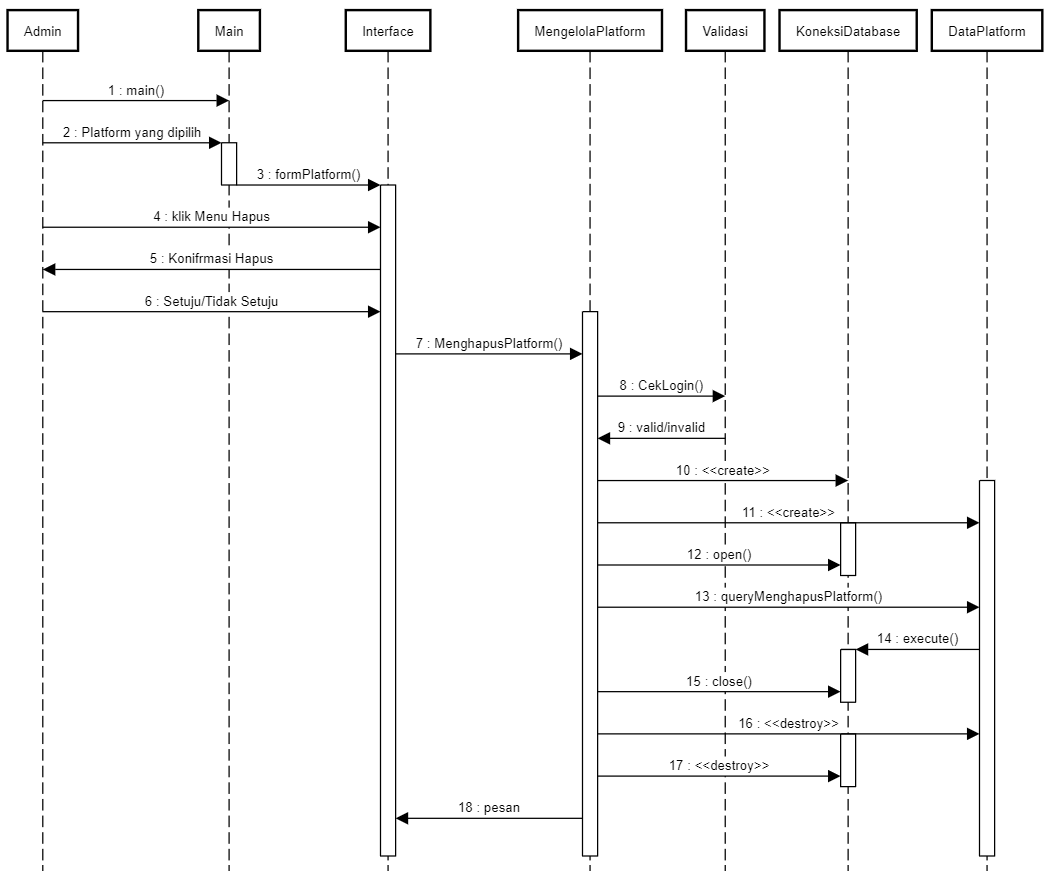
Gambar IV. 13 Sequence Diagram Melihat Platform

1. *Sequence* *Diagram* Mengubah Platform



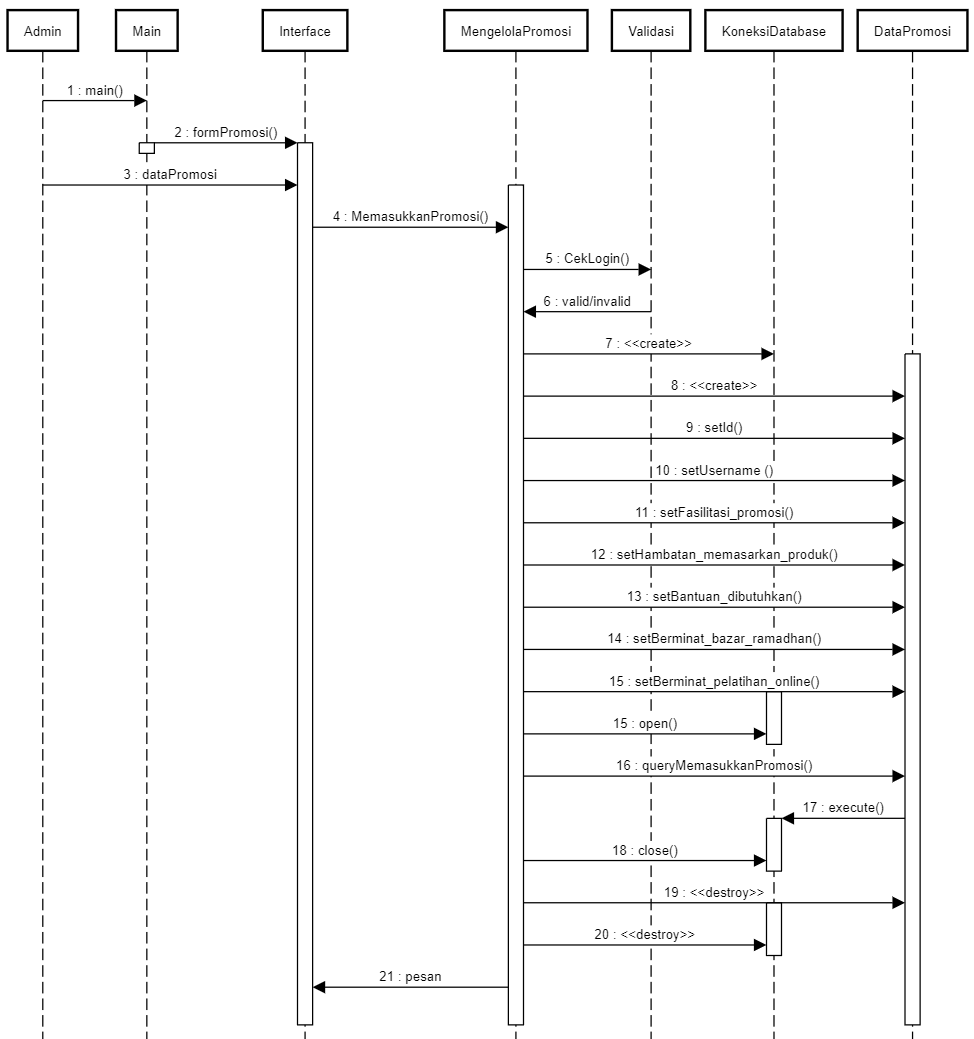
Gambar IV. 14 Sequence Diagram Mengubah Platform

1. *Sequence* *Diagram* Menghapus Platform



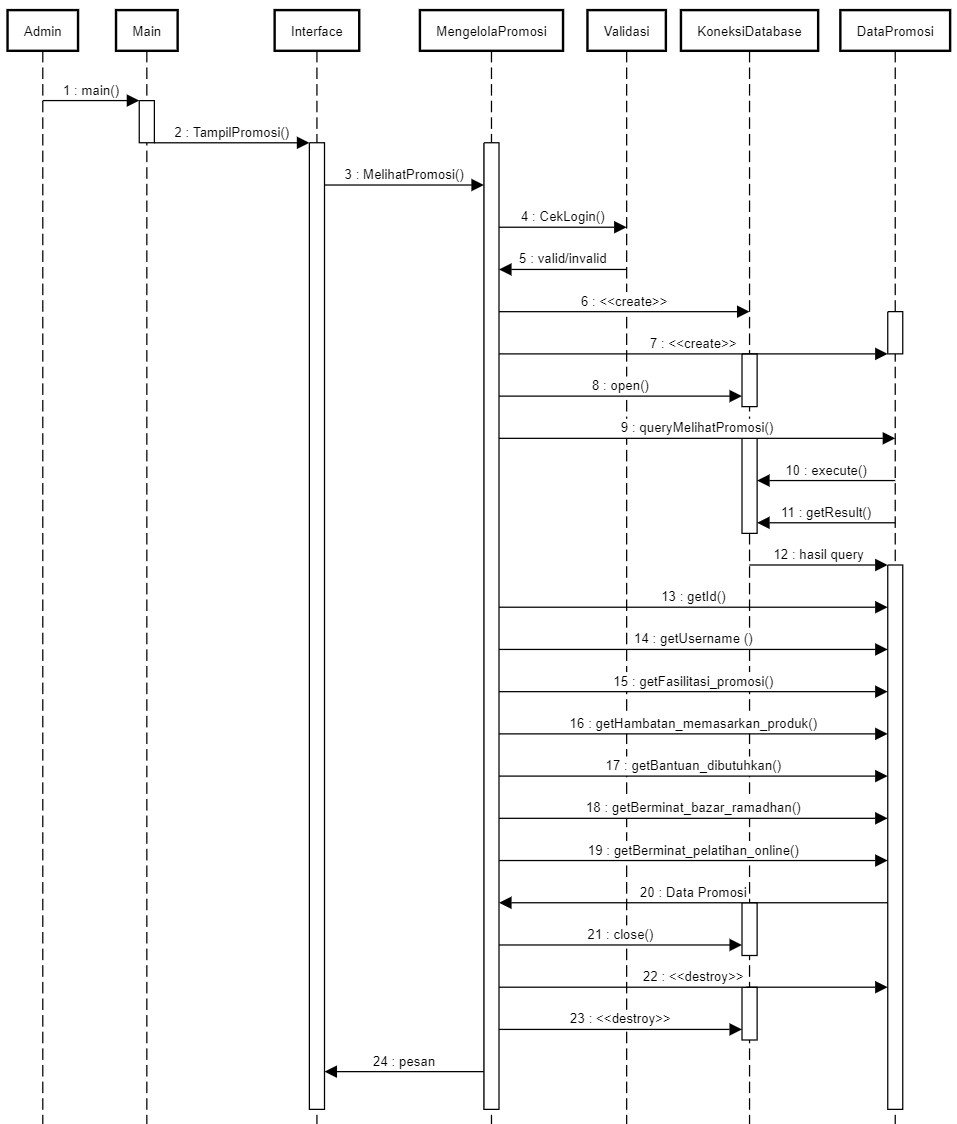
Gambar IV. 15 Sequence Diagram Menghapus Platform

1. *Sequence* *Diagram* Memasukkan Promosi



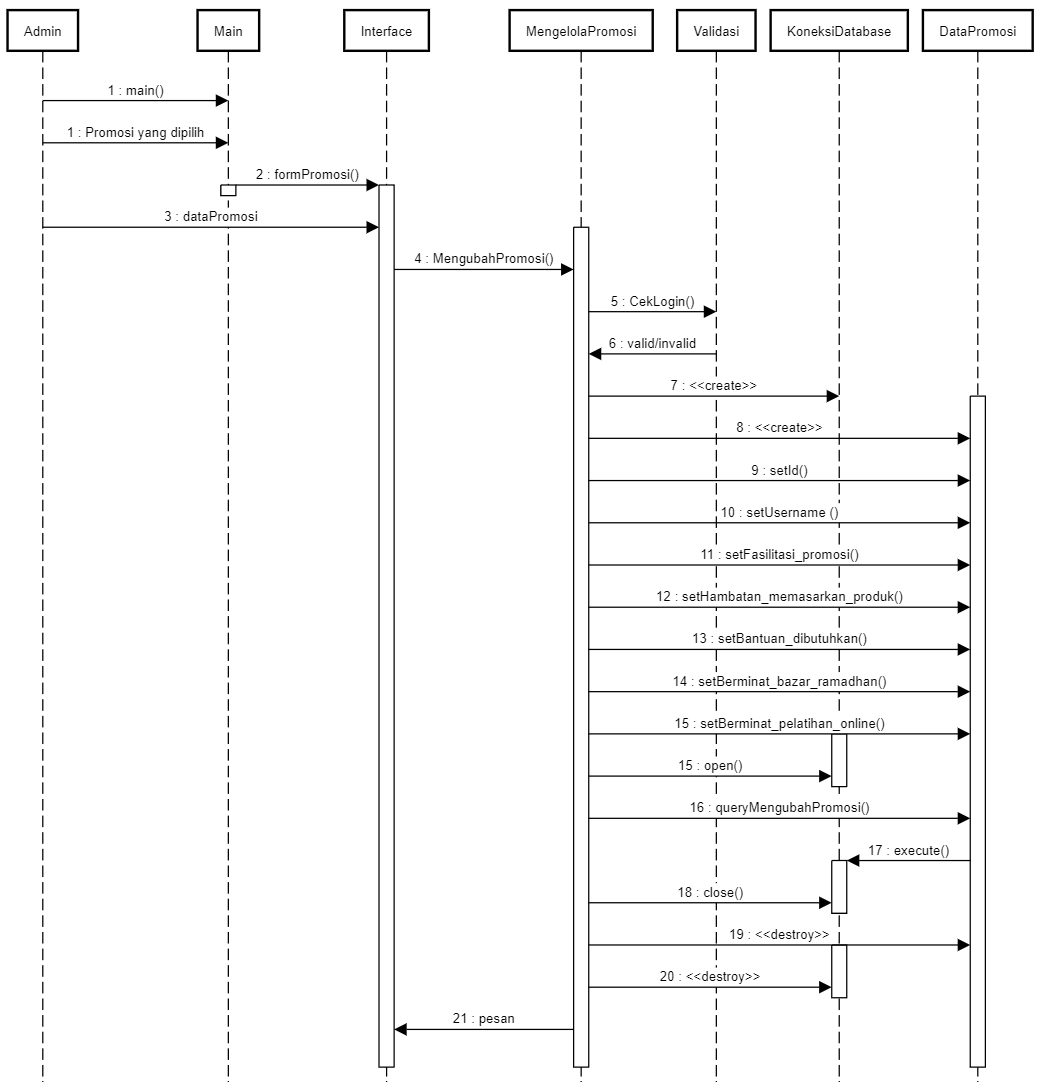
Gambar IV. 16 Sequence Diagram Memasukkan Prmosi

1. *Sequence* *Diagram* Melihat Promosi

**

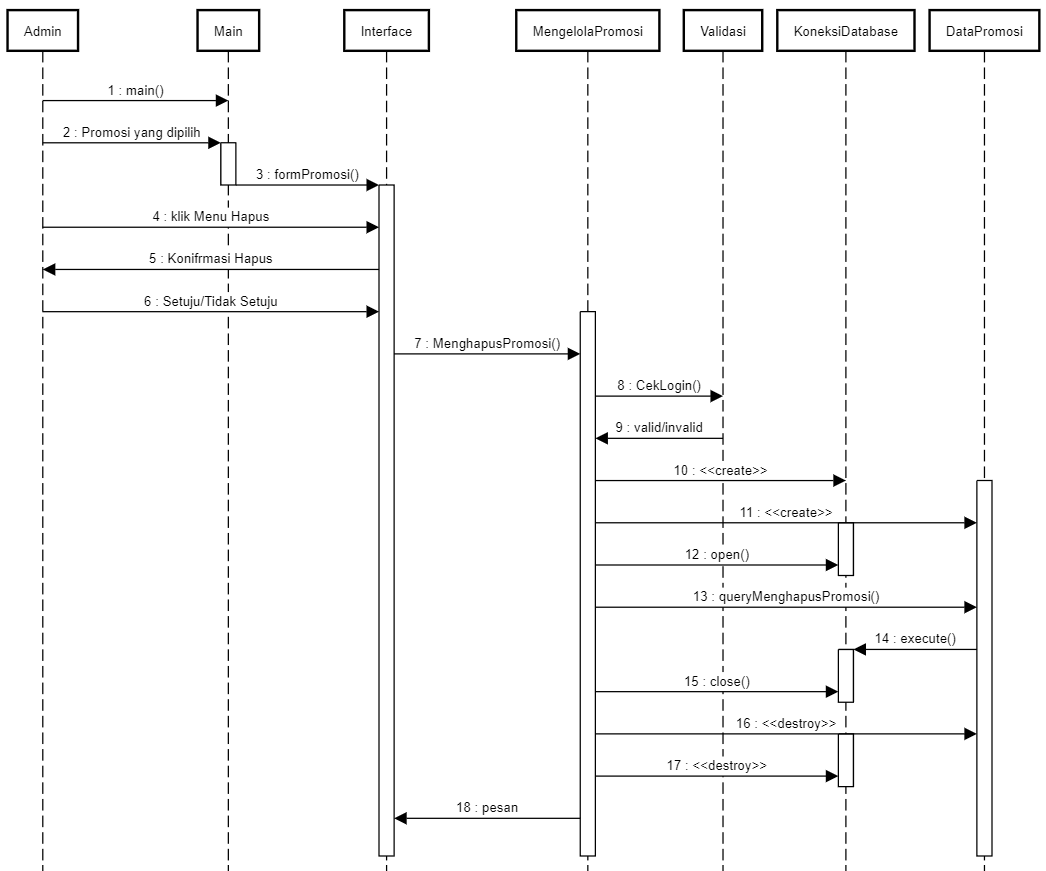
Gambar IV. 17 Sequence Diagram Melihat Promosi

1. *Sequence* *Diagram* Mengubah Promosi



Gambar IV. 18 Sequence Diagram Mengubah Promosi

1. *Sequence* *Diagram* Menghapus Promosi



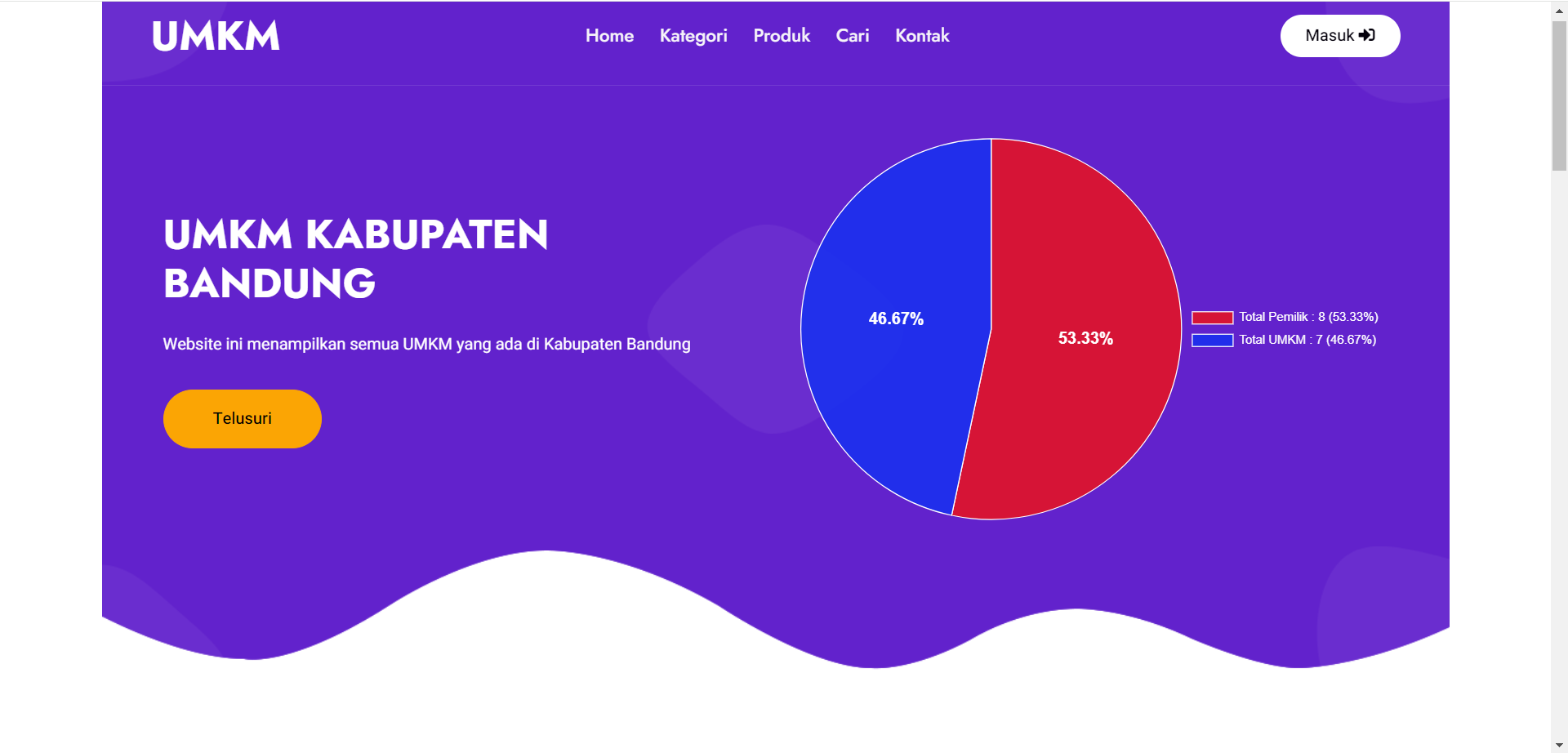
Gambar IV. 19 Sequence Diagram Menghapus Promosi

## Fitur-fitur Aplikasi

### Implementasi *Interface*

Implementasi antarmuka sistem Informasi UMKM akan dijelaskan pada gambar-gambar dibawah ini:

1. Tampilan *Home*



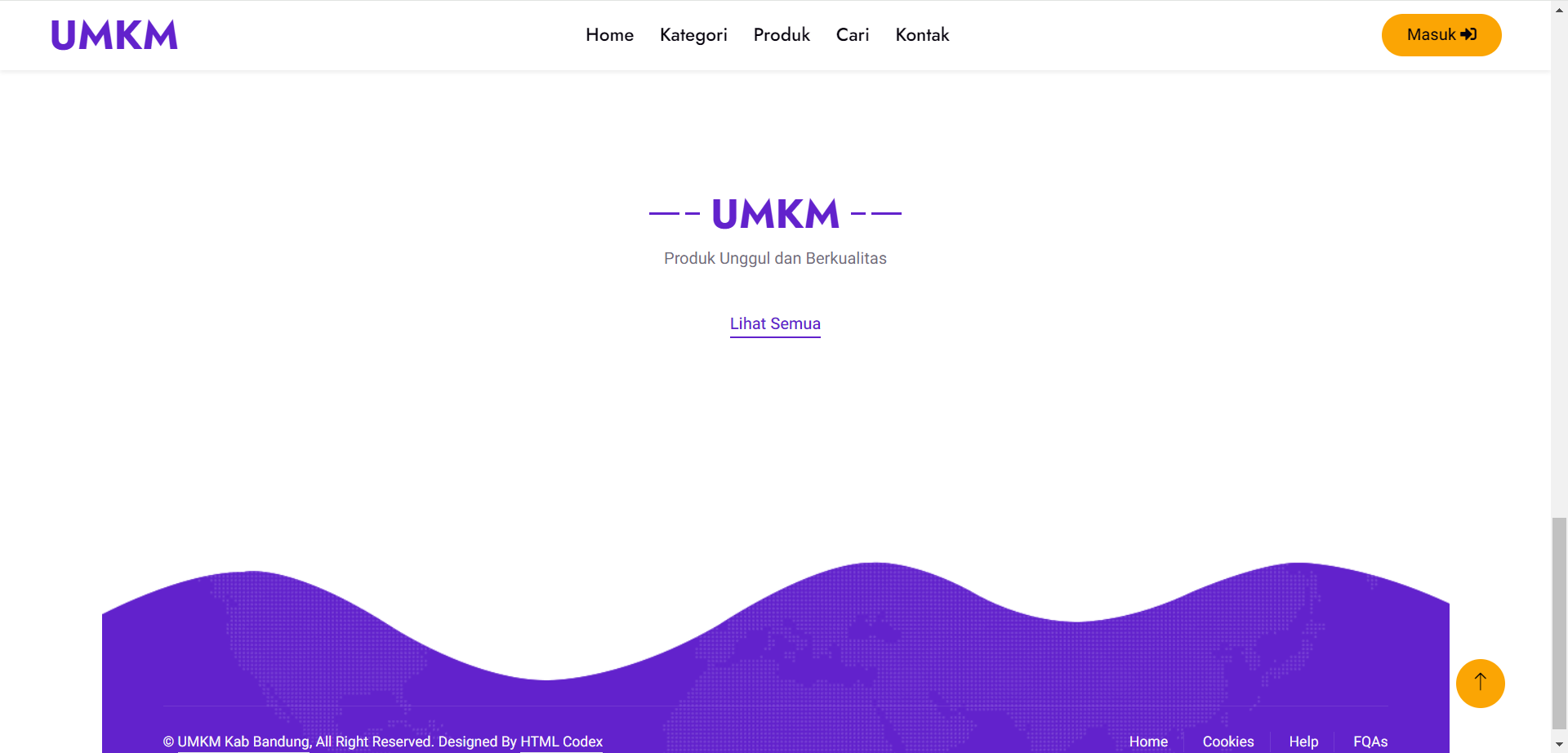
Gambar IV. 20 Tampilan Home

1. Tampilan Kategori



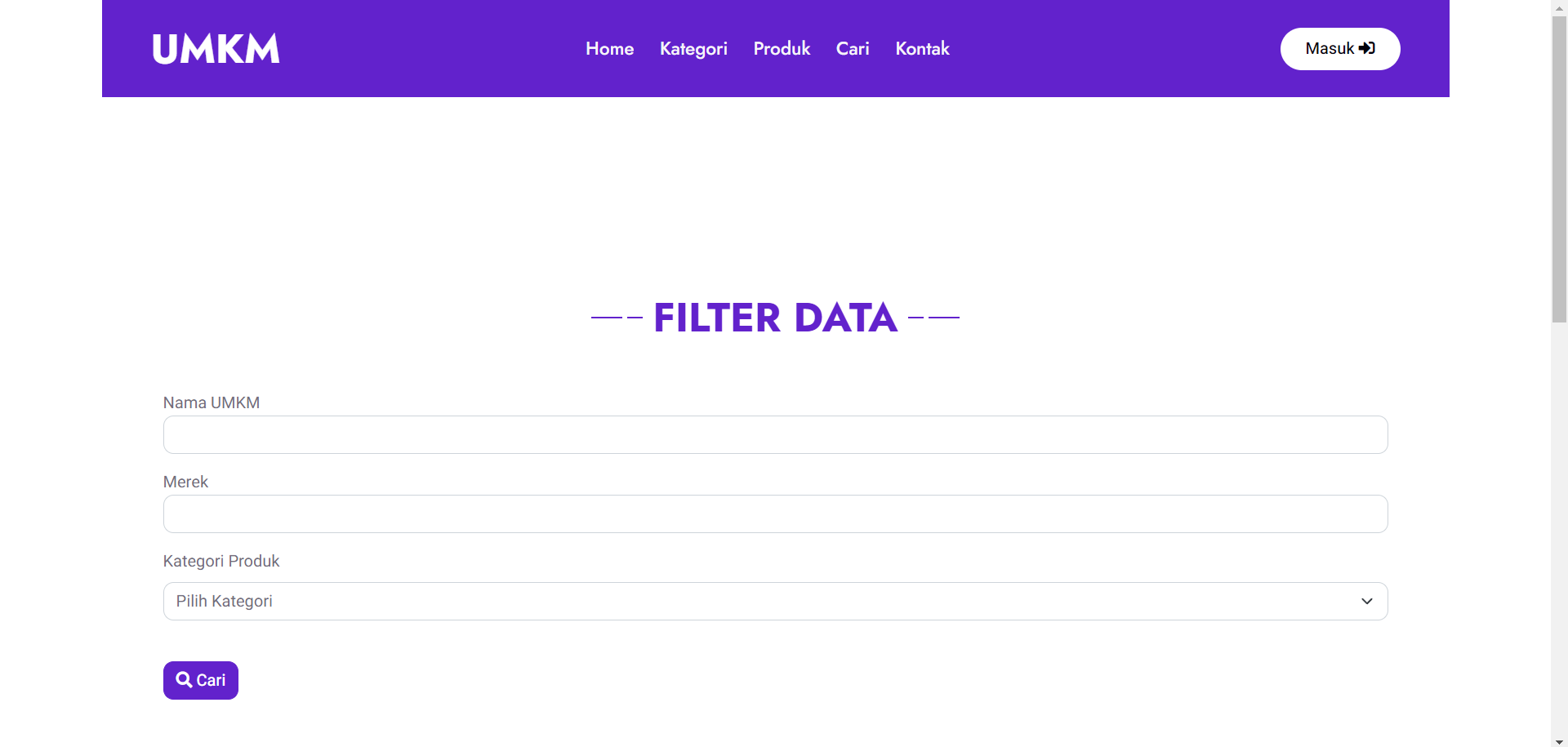
Gambar IV. 21 Tampilan Kategori

1. Tampilan Produk Umkm



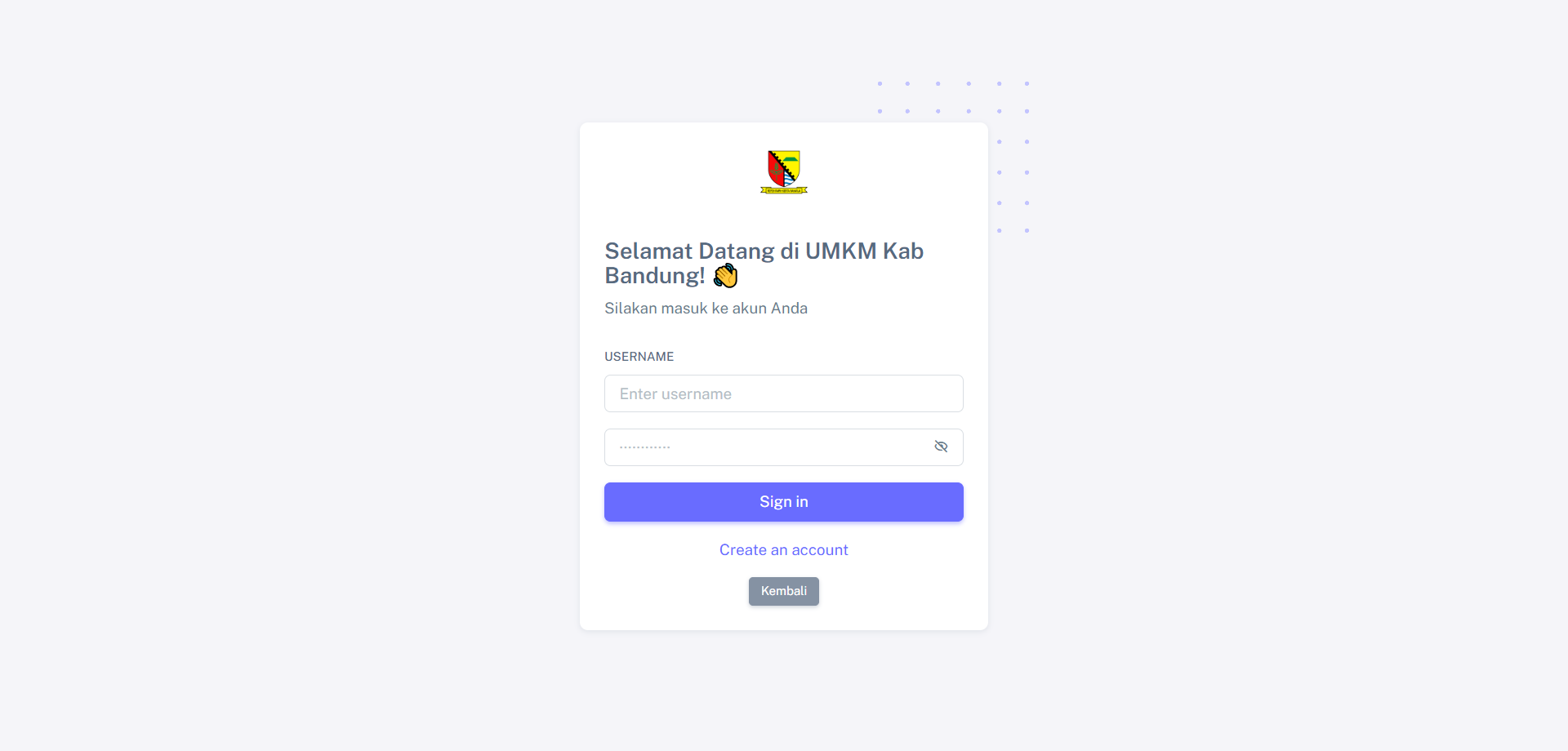
Gambar IV. 22 Produk UMKM

1. Tampilan Search UMKM



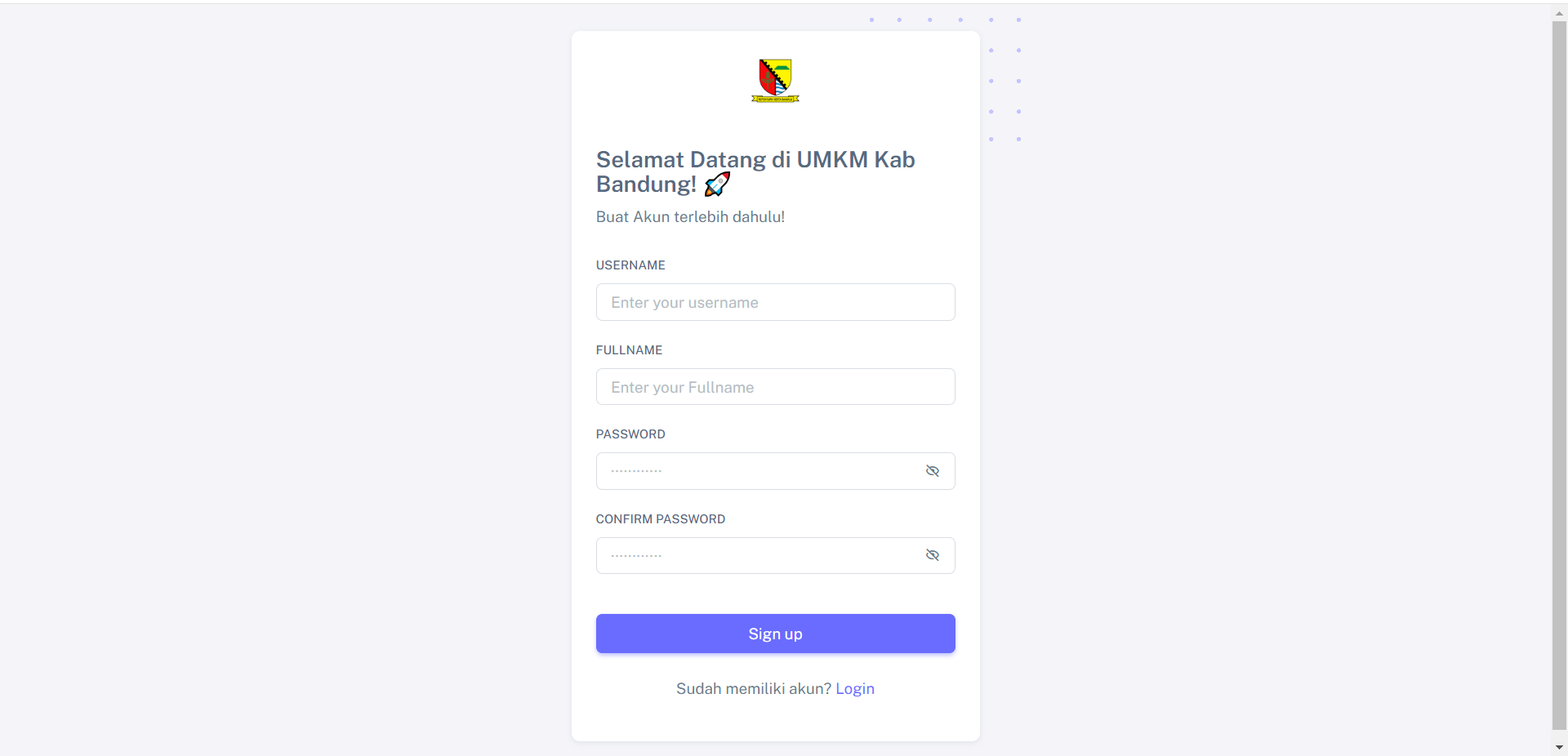
Gambar IV. 23 Tampilan Search UMKM

1. Tampilan Login



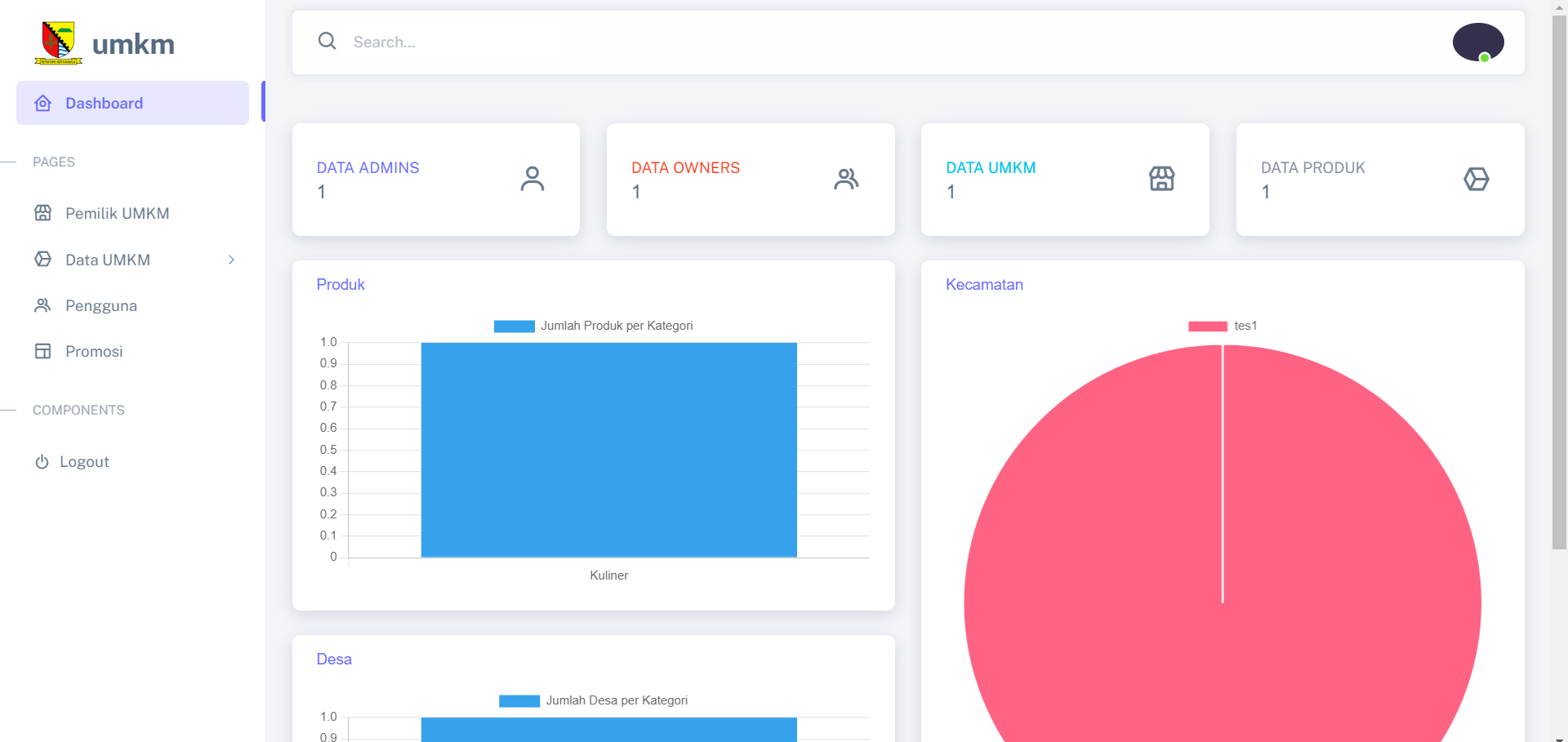
Gambar IV. 24 Tampilan Login

1. Tampilan *Sign Up*



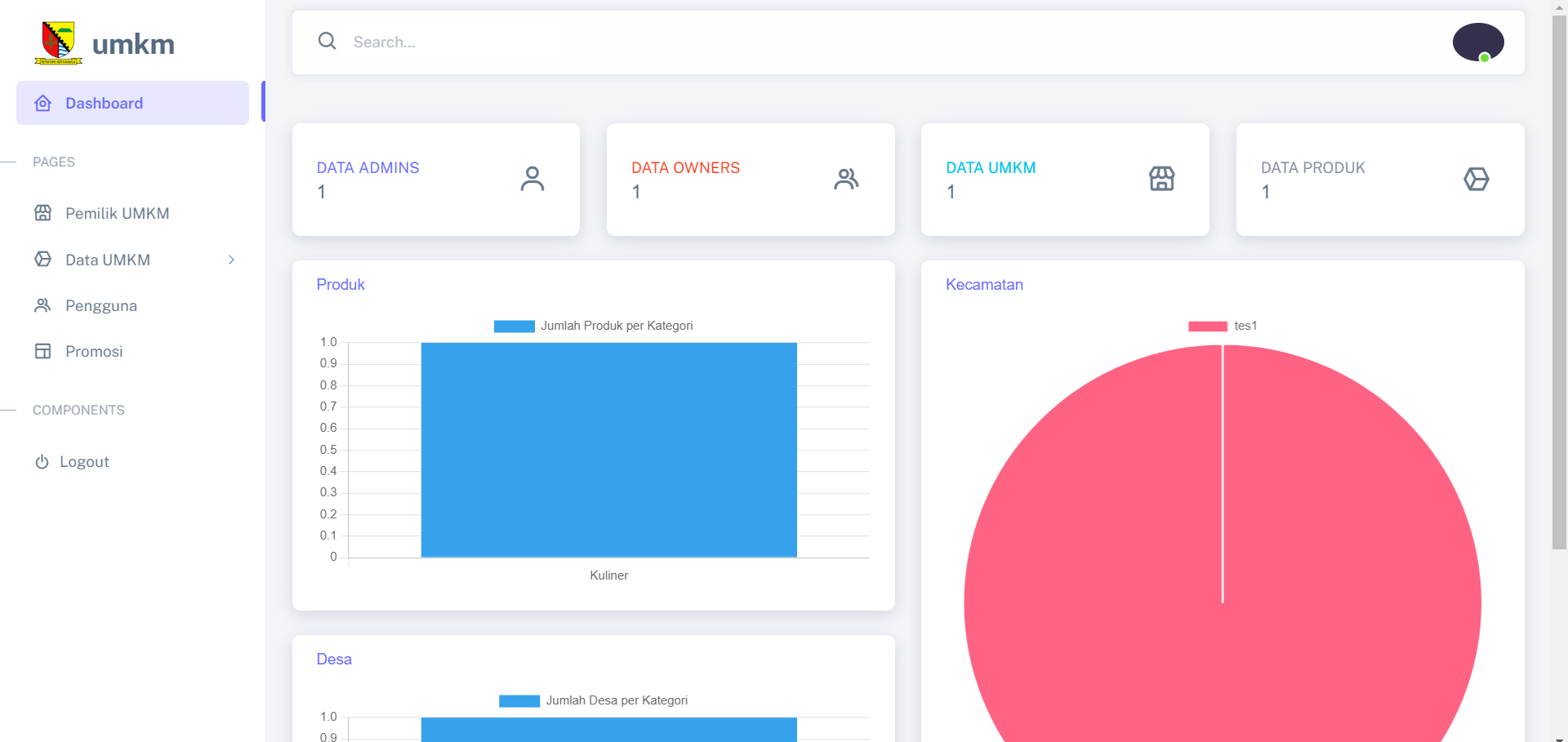
Gambar IV. 25 Tampilan Sign Up

1. Tampilan Logout



Gambar IV. 26 Tampilan Logout

1. Tampilan *Dashboard* Admin



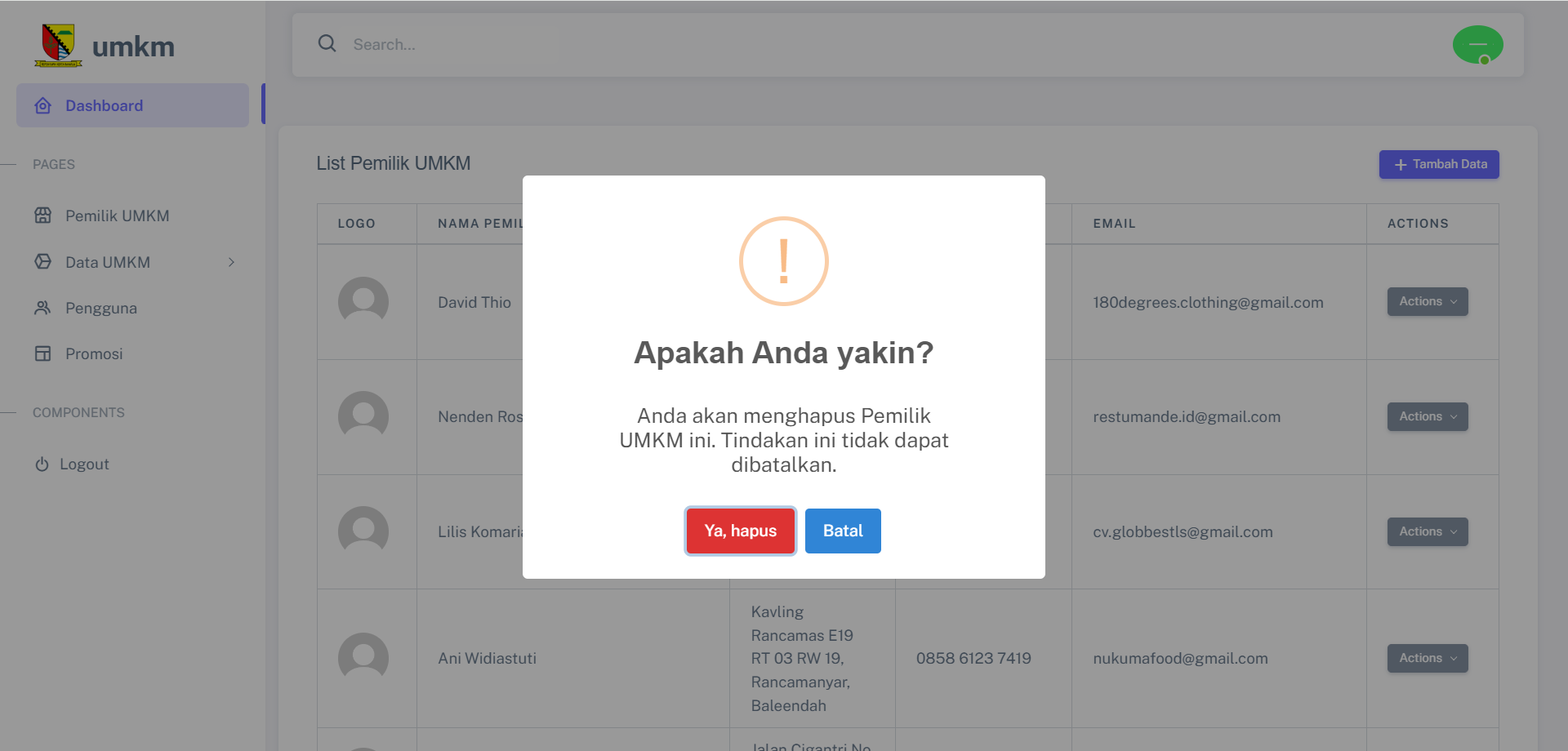
Gambar IV. 27 Tampilan Dashboard Admin

1. Tampilan Menambahkan Pemilik UMKM



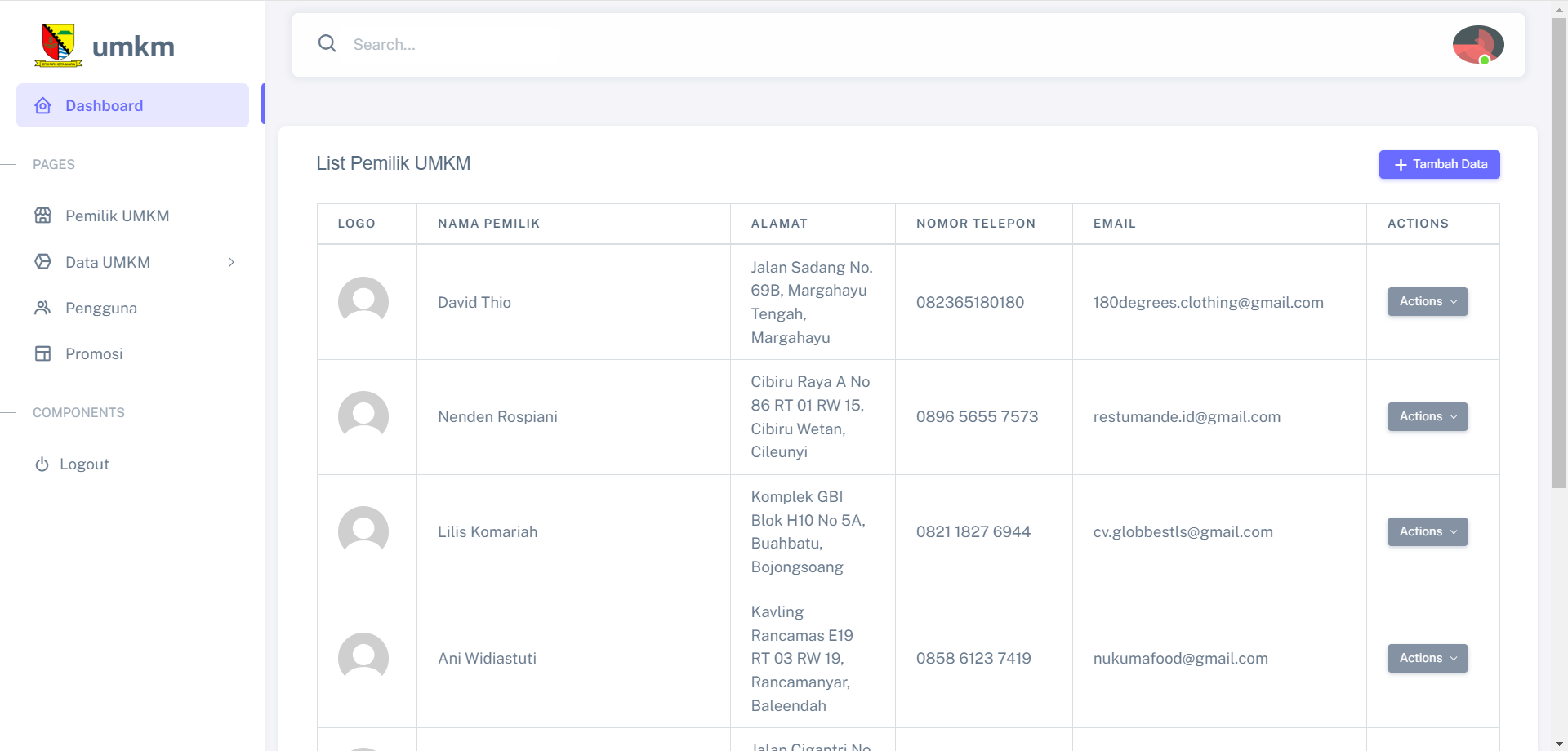
Gambar IV. 28 Tampilan Menambahkan Pemilik UMKM

1. Tampilan Menghapus Pemilik UMKM



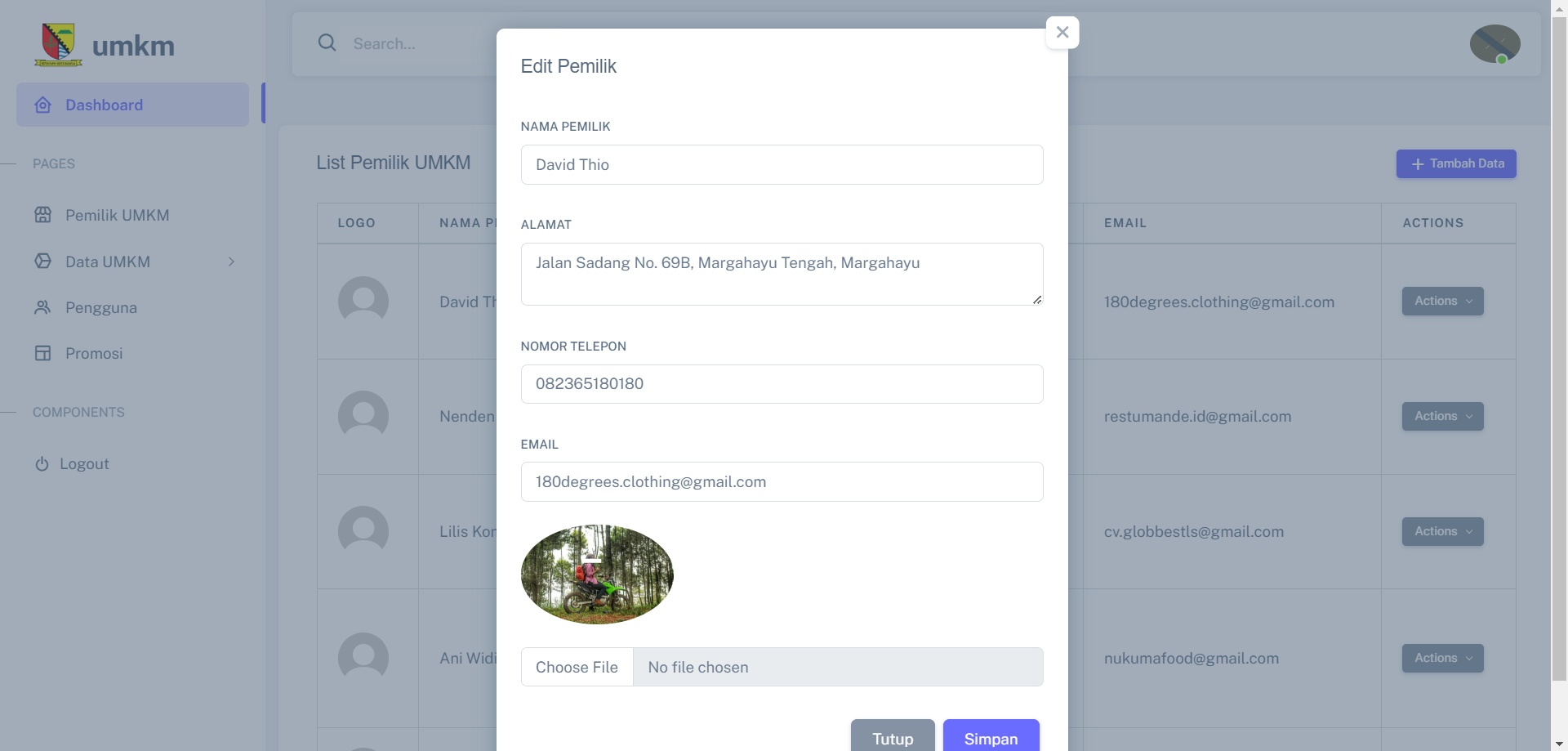
Gambar IV. 29 Tampilan Menghapus Pemilik UMKM

1. Tampilan Melihat Pemilik UMKM



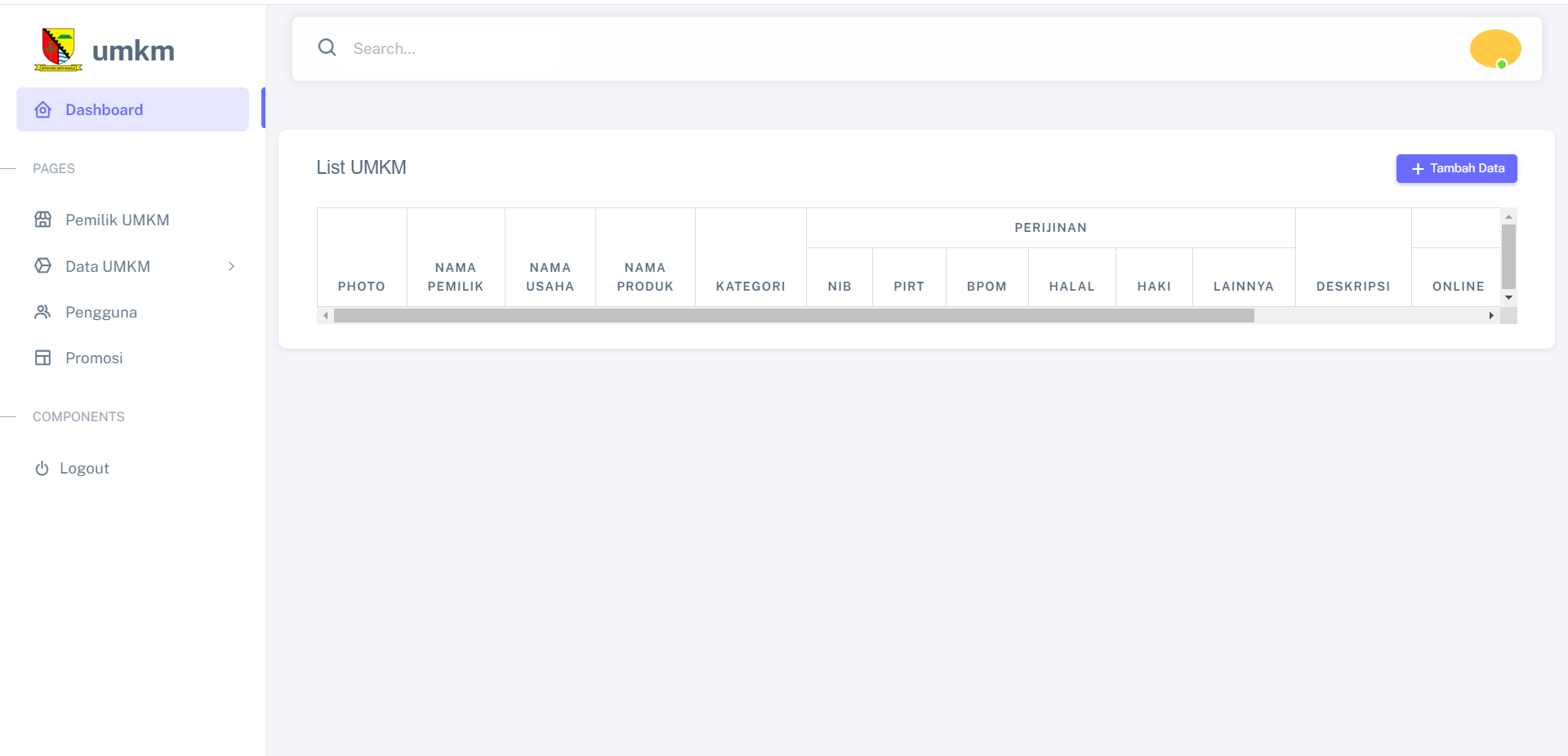
Gambar IV. 30 Tampilan Melihat Pemilik UMKM

1. Tampilan Edit Pemilik UMKM



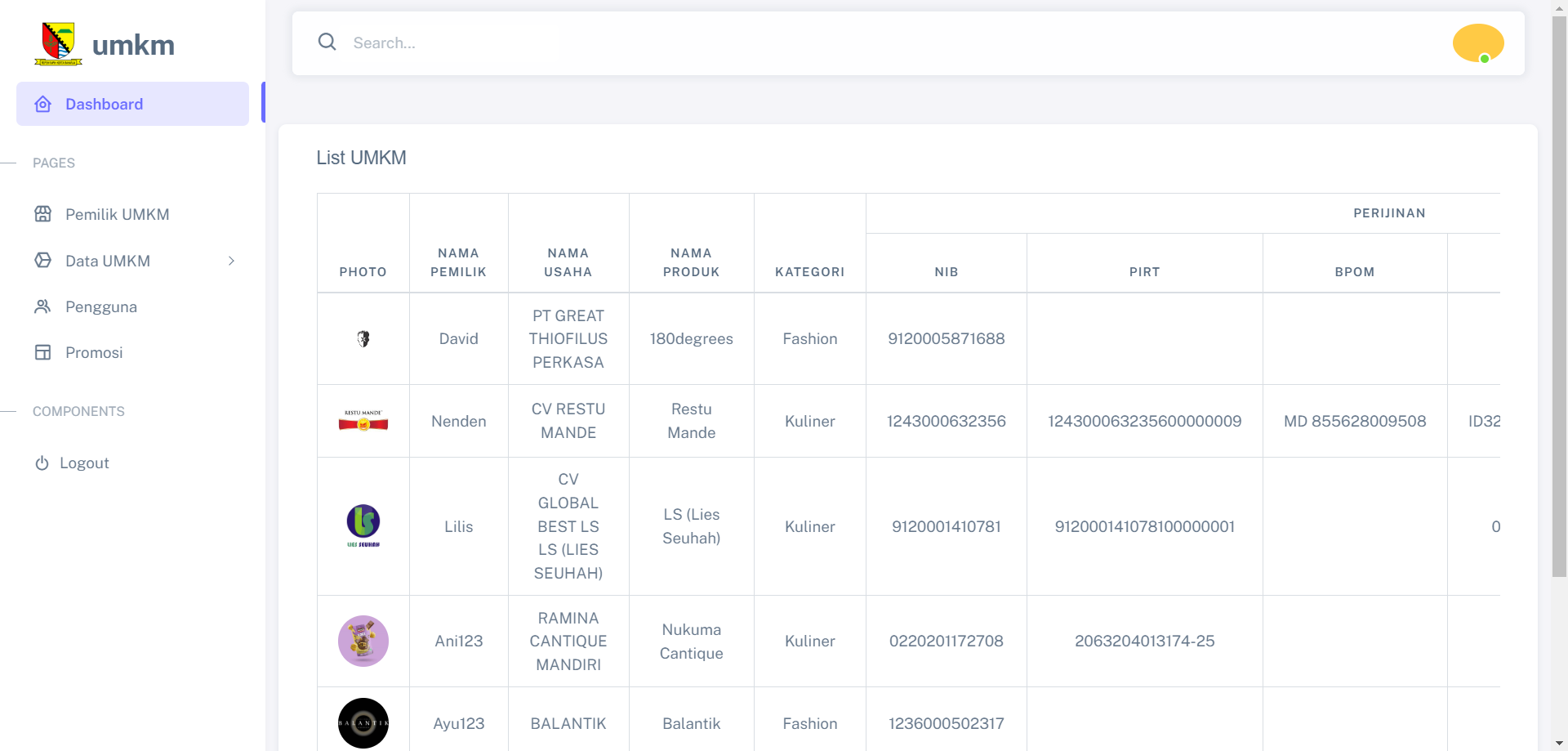
Gambar IV. 31 Tampilan Edit Pemilik UMKM

1. Tampilan Memasukkan UMKM



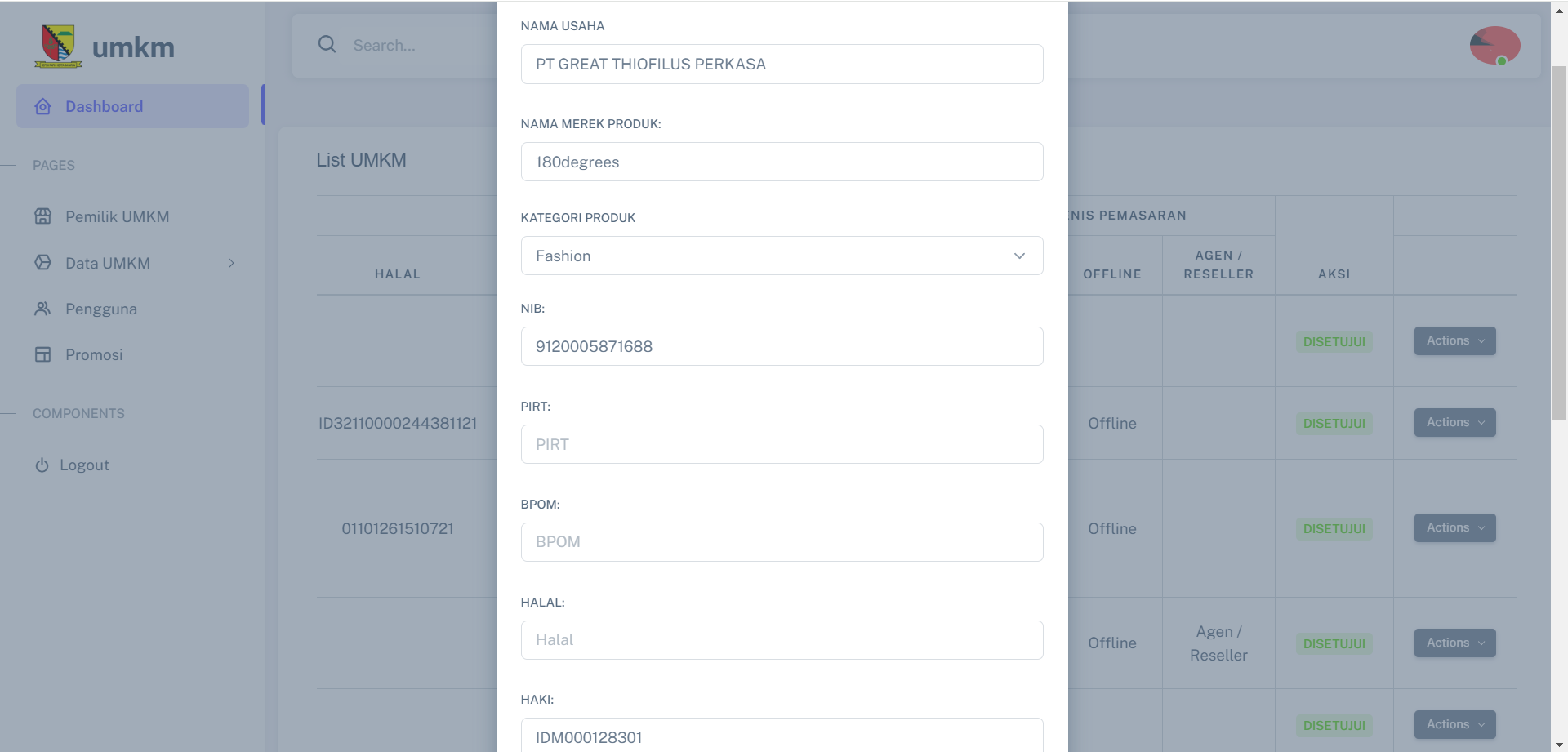
Gambar IV. 32 Tampilan Memasukan UMKM

1. Tampilan Melihat UMKM



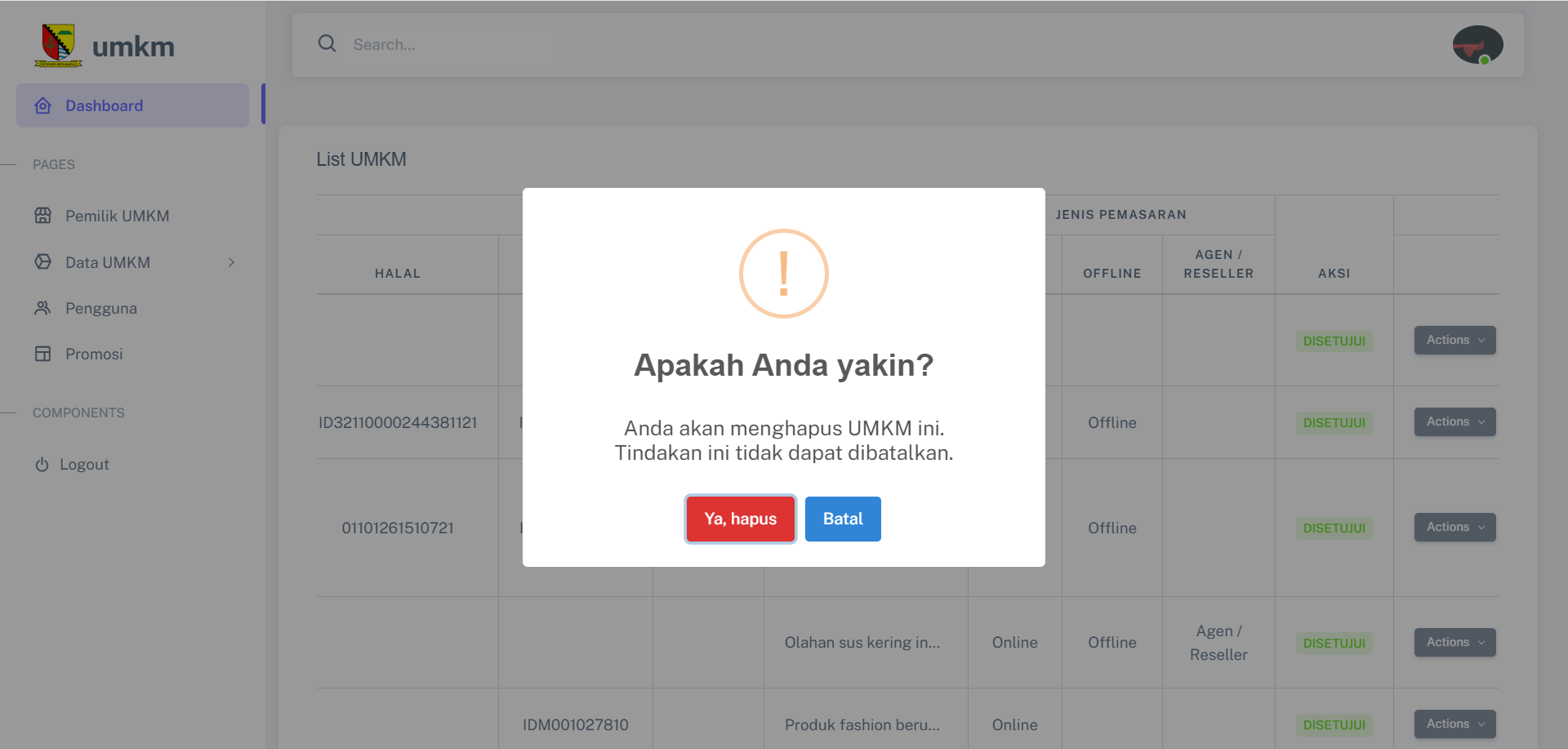
Gambar IV. 33 Tampilan Melihat UMKM

1. Tampilan Mengubah UMKM



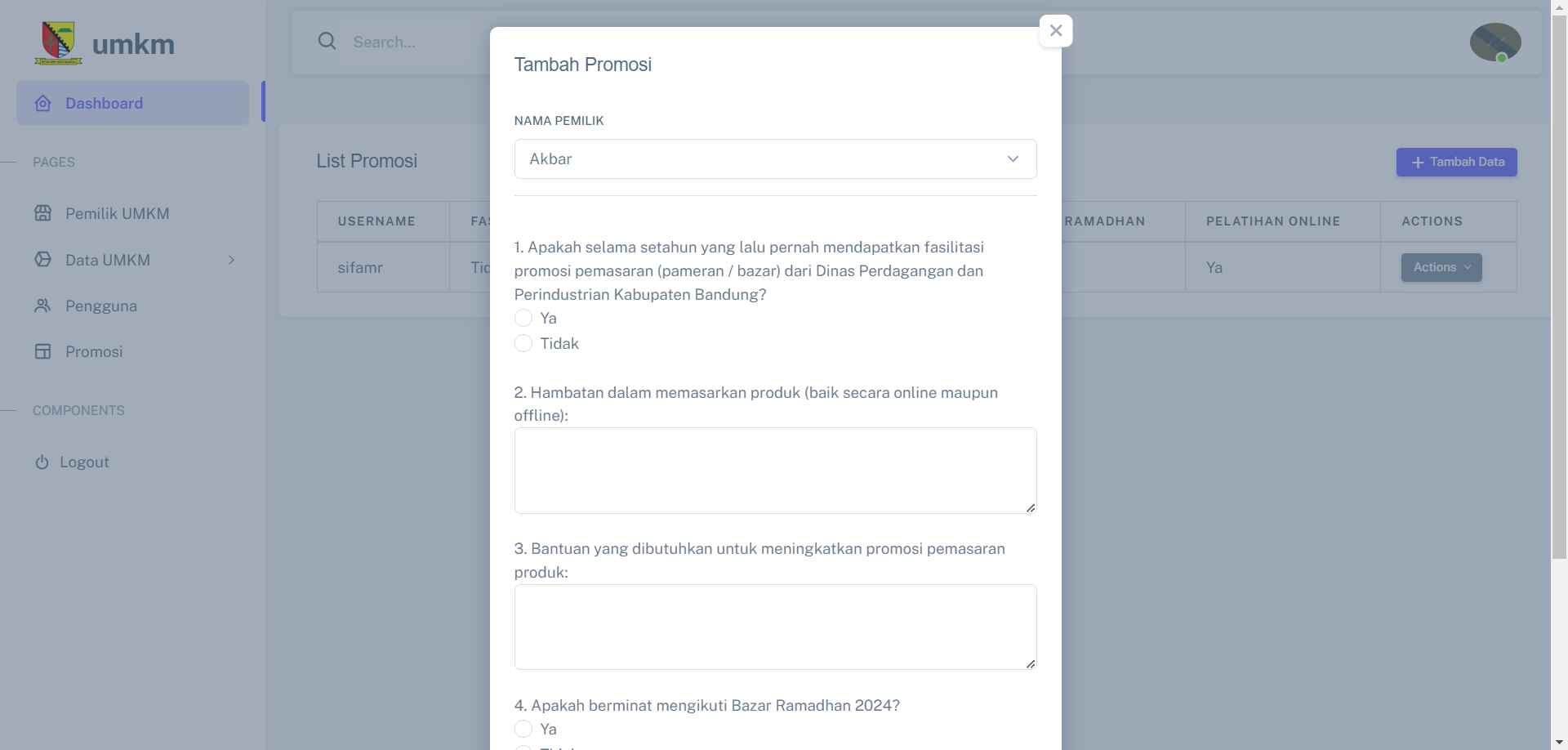
Gambar IV. 34 Tampilan Mengubah UMKM

1. Tampilan Menghapus UMKM



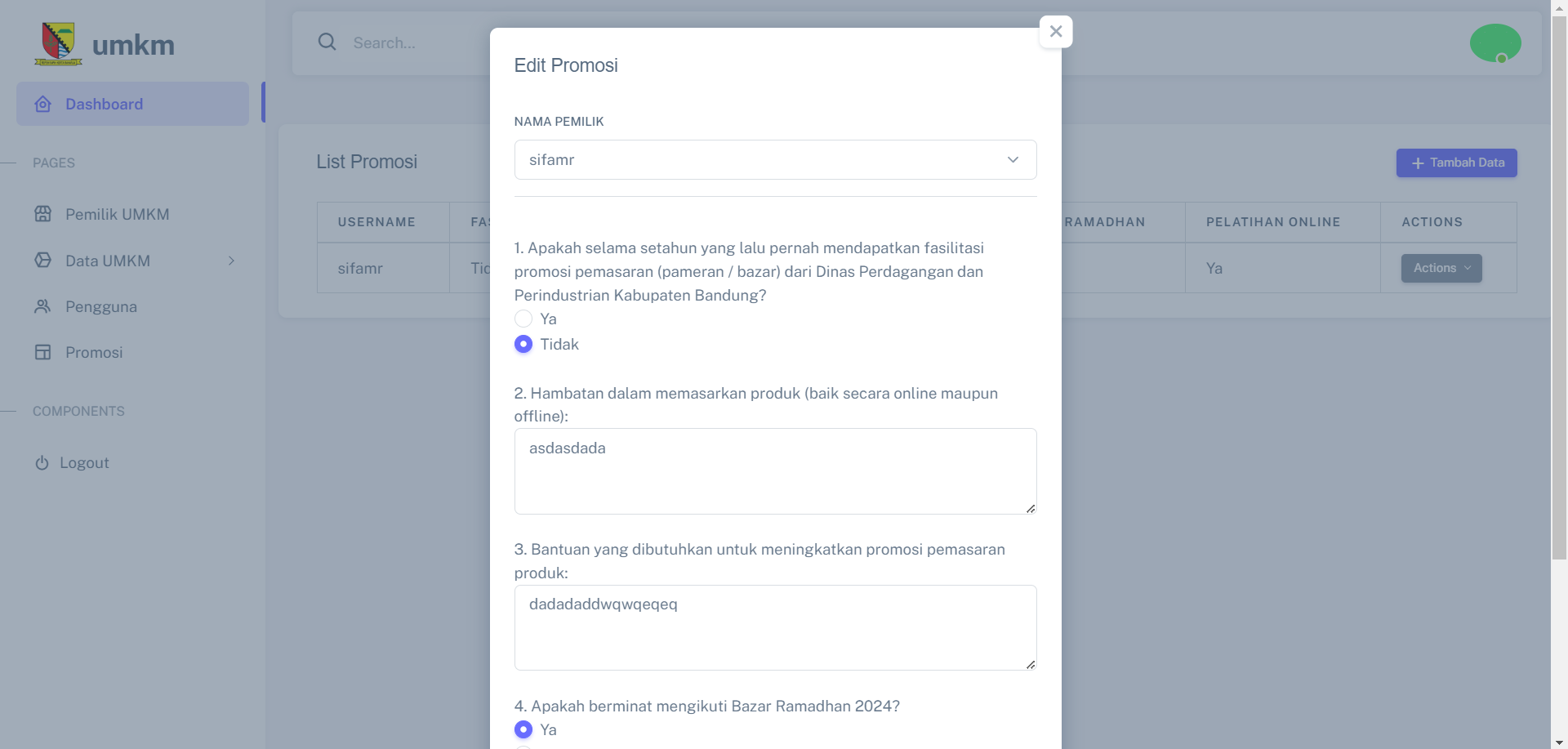
Gambar IV. 35 Tampilan Menghapus UMKM

1. Tampilan Memasukkan Promosi



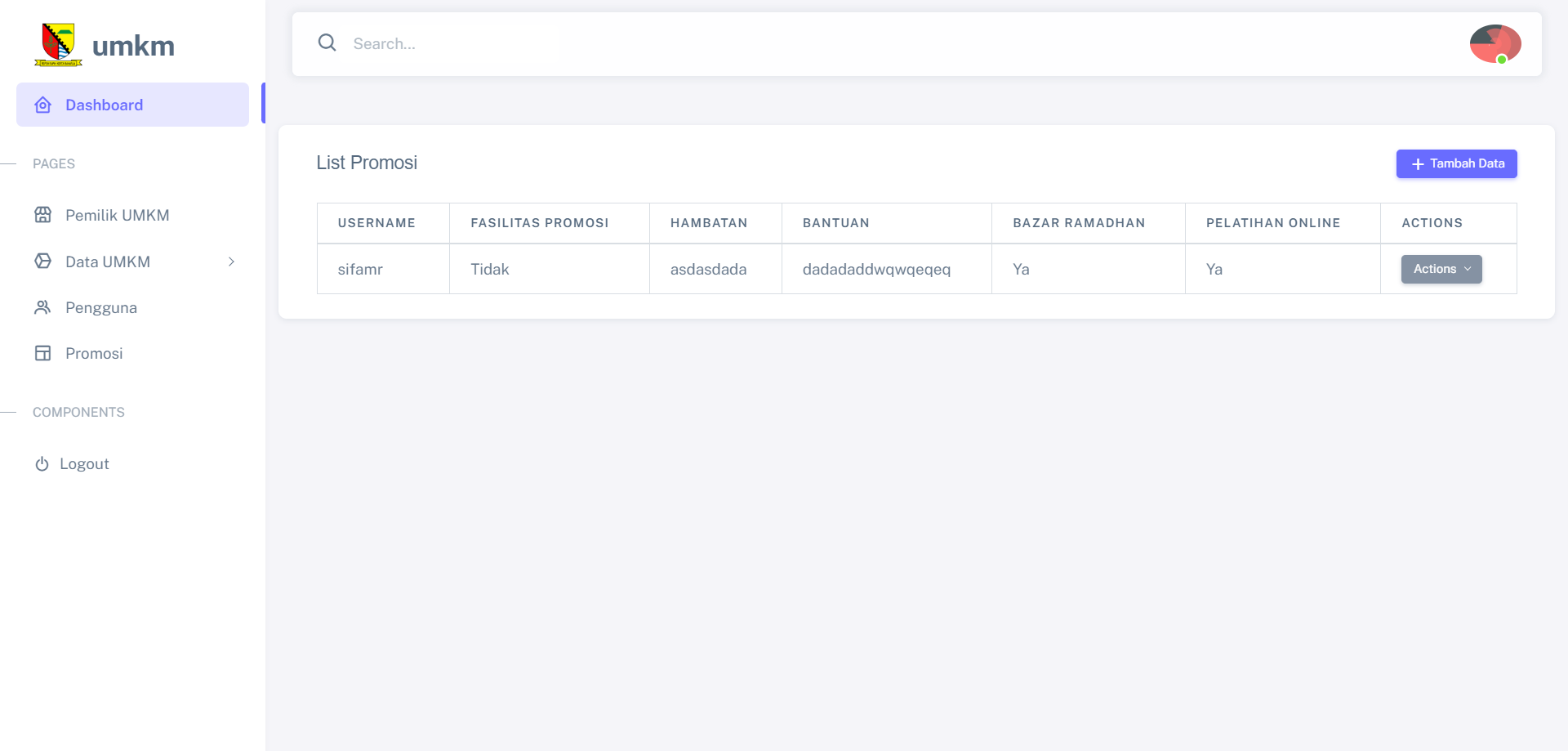
Gambar IV. 36 Tampilan Memasukkan Promosi

1. Tampilan Mengubah Promosi



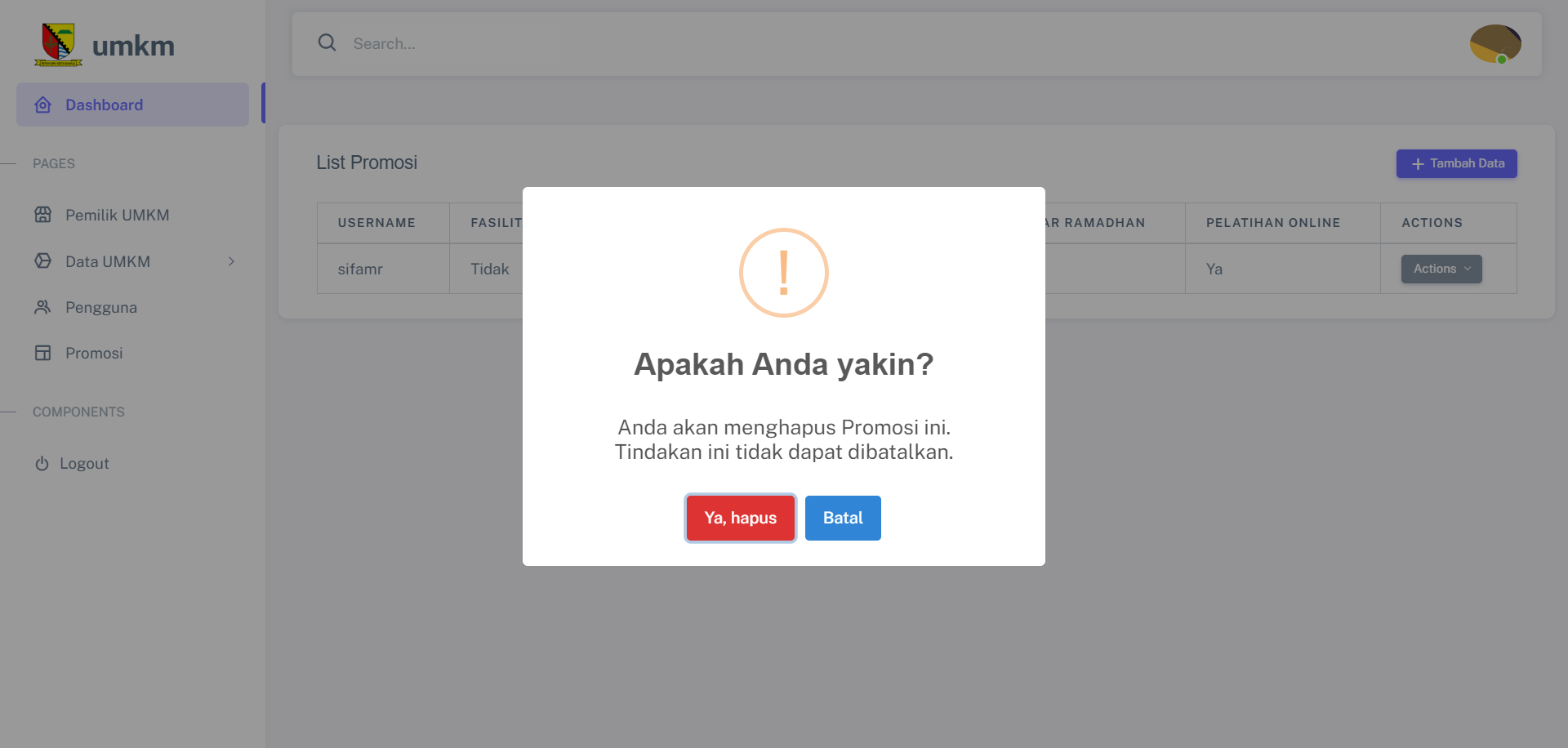
Gambar IV. 37 Tampilan Mengubah Promosi

1. Tampilan Melihat Promosi



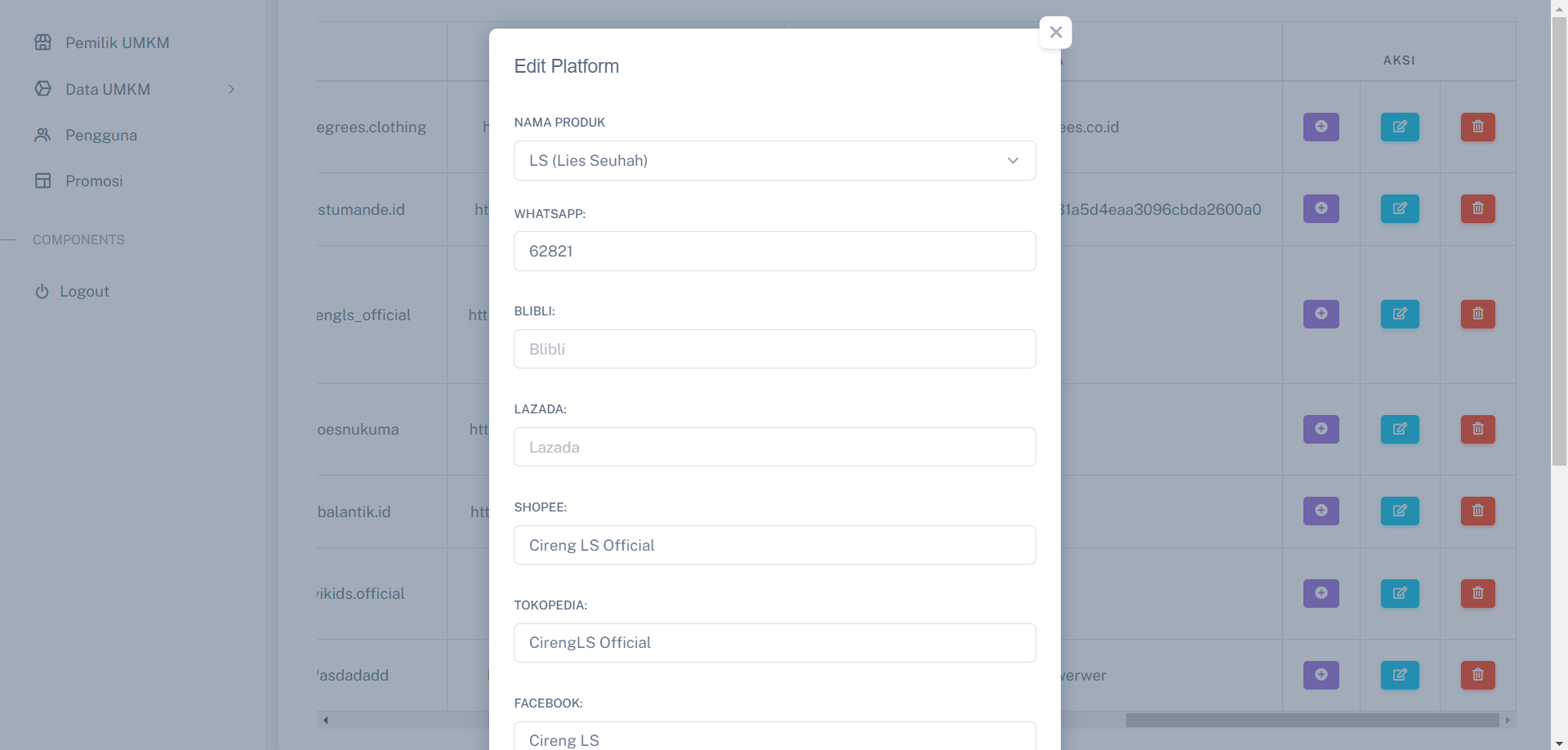
Gambar IV. 38 Tampilan Melihat Promosi

1. Tampilan Menghapus Promosi



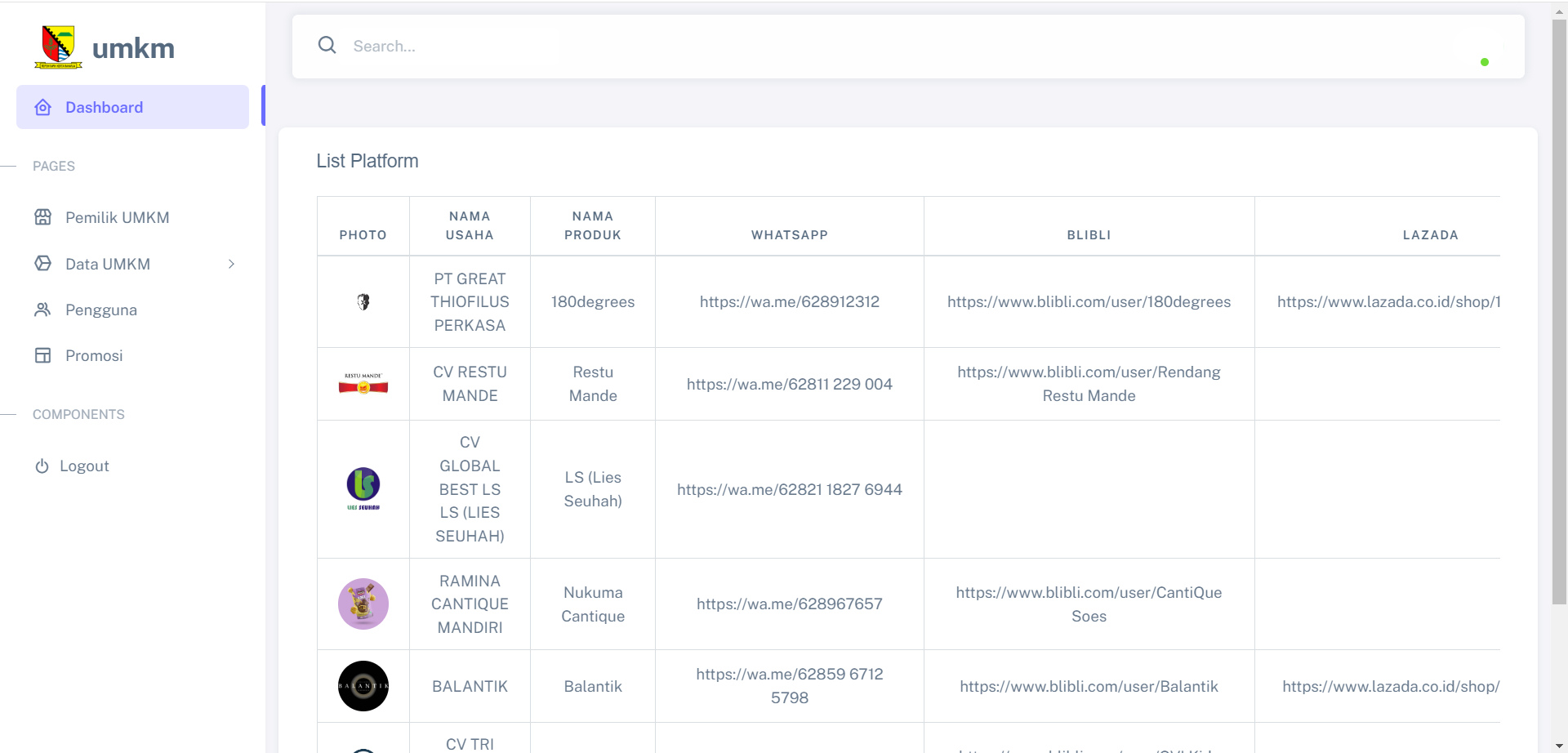
Gambar IV. 39 Tampilan Menghapus Promosi

1. Tampilan Memasukkan Platform



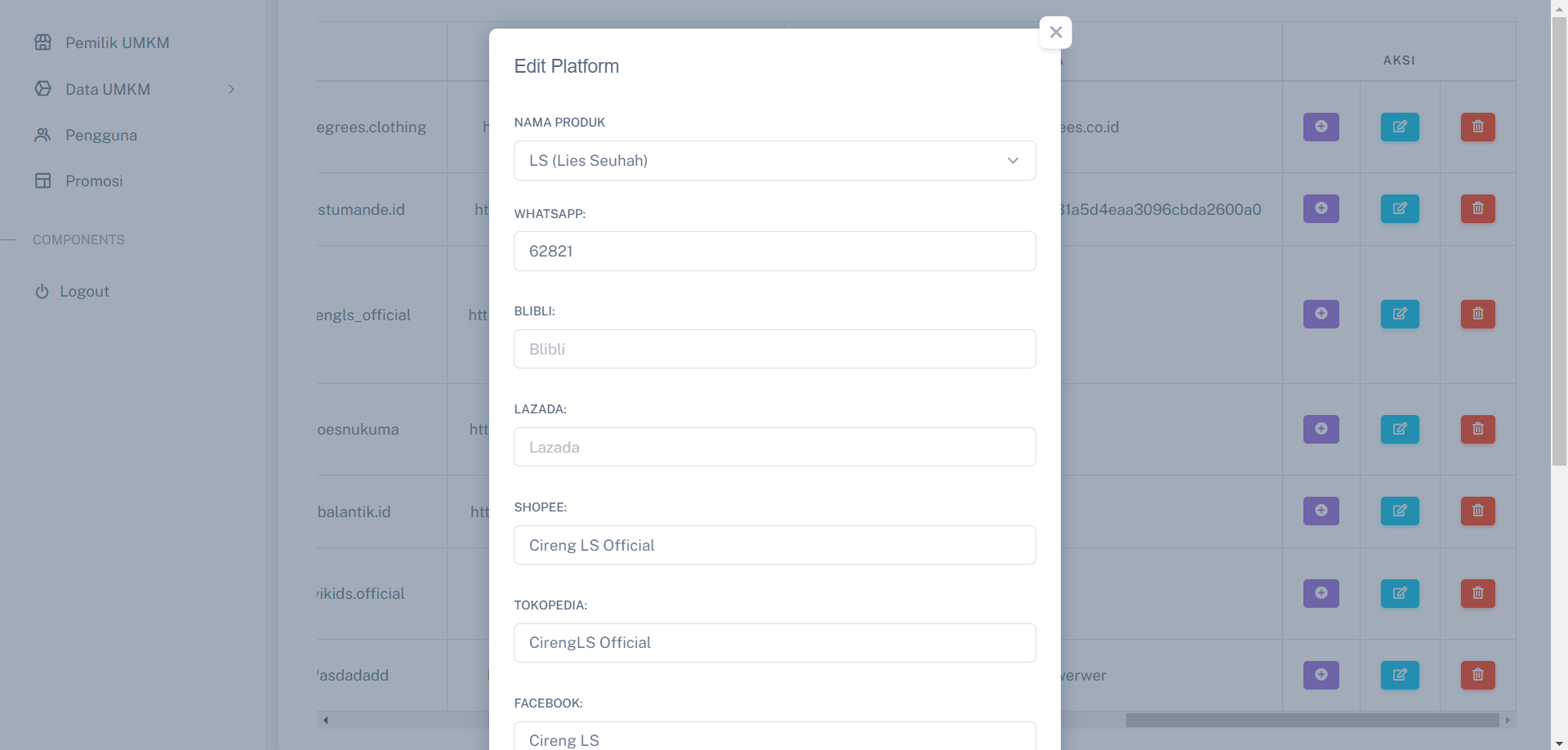
Gambar IV. 40 Tampilan Memasukkan Platfotm

1. Tampilan Melihat Platform



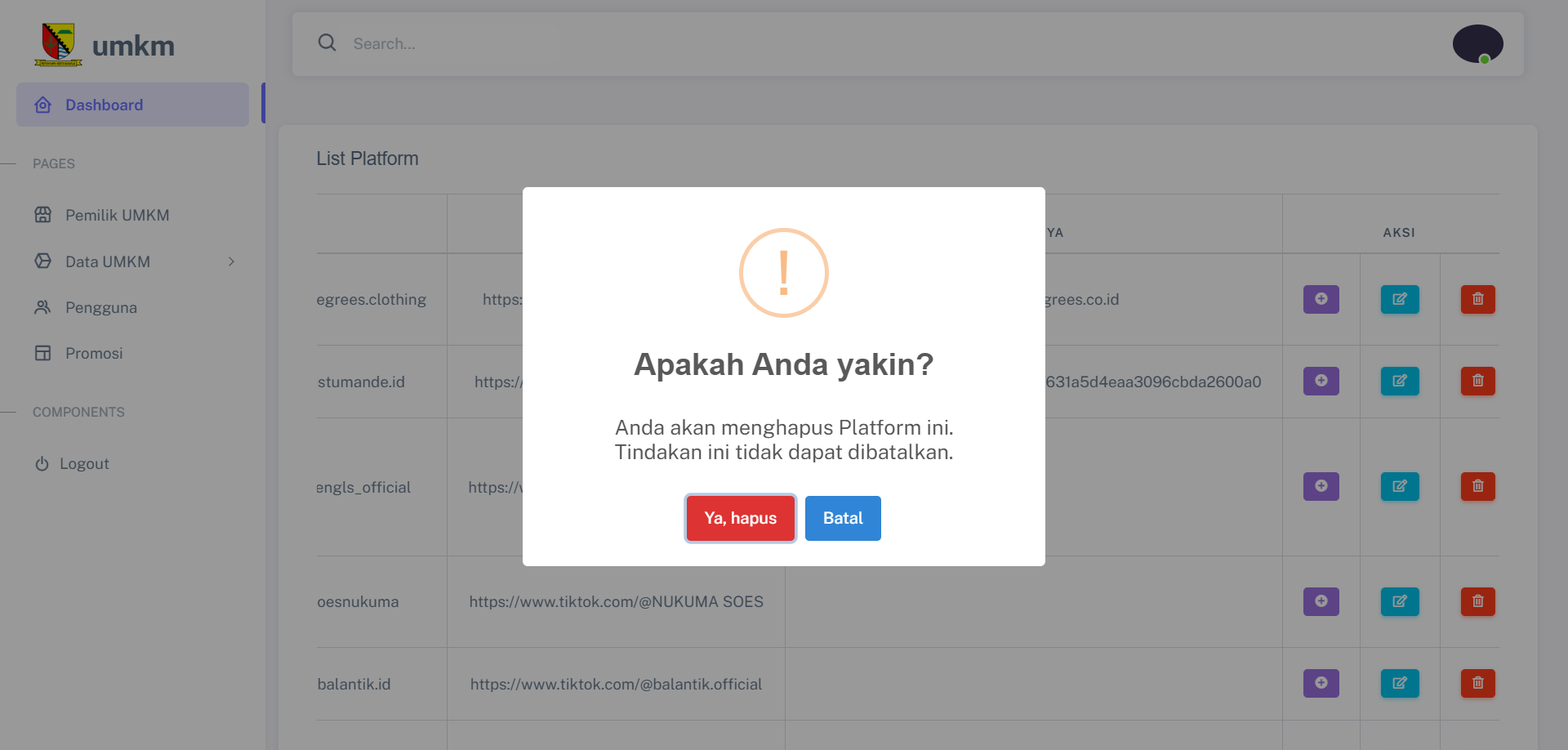
Gambar IV. 41 Tampilan Melihat Platform

1. Tampilan Mengubah Platform



Gambar IV. 42 Tampilan Mengubah Platform

1. Tampilan Menghapus Platform



Gambar IV. 43 Tampilan Menghapus Platform

## Pseudocode Fitur-fitur Aplikasi

### Pseudocode Create User

Judul: Program Create User

Deklarasi:

var username, fullname, password, confpassword, usertype: string;

var hashed\_password, hashed\_confpassword: string;

var data: array;

var existing\_user: array;

var result: boolean;

var response: array;

Implementasi

BEGIN

// Tangkap data dari formulir pendaftaran

username = form input 'username';

fullname = form input 'fullname';

password = form input 'password';

confpassword = form input 'confpassword';

usertype = form input 'usertype';

// Validasi panjang username, password, dan confpassword

IF length(username) < 5 OR length(password) < 5 OR length(confpassword) < 5 THEN

// Jika salah satu input kurang dari 5 karakter, tampilkan pesan kesalahan

response['message'] = "Username, password, and confpassword must be at least 5 characters long.";

RETURN json\_encode(response);

END IF

// Hash password

hashed\_password = password\_hash(password, PASSWORD\_DEFAULT);

hashed\_confpassword = password\_hash(confpassword, PASSWORD\_DEFAULT);

data = {

'username': username,

'fullname': fullname,

'password': hashed\_password,

'confpassword': hashed\_confpassword,

'usertype': usertype

};

// Validasi konfirmasi password

IF password ≠ confpassword THEN

// Jika konfirmasi password tidak sesuai, tampilkan pesan kesalahan

response['message'] = "Password confirmation doesn't match.";

RETURN json\_encode(response);

END IF

// Check if the username already exists

existing\_user = User\_model.get\_user\_by\_username(username);

IF existing\_user THEN

// Jika nama pengguna sudah ada, tampilkan pesan kesalahan

response['message'] = "Username already exists. Please choose a different username.";

RETURN json\_encode(response);

END IF

// Check if a file is uploaded

IF file 'photo' is uploaded THEN

// File is uploaded, include it in the data

config['upload\_path'] = './uploads/users/';

config['allowed\_types'] = 'gif|jpg|png';

config['file\_name'] = file name 'photo';

LOAD library 'upload' with config;

IF upload 'photo' successful THEN

data['photo'] = uploaded file name;

ELSE

// Jika gagal mengunggah gambar, tangani kesalahan

error = upload errors;

// Tampilkan pesan kesalahan jika perlu

END IF

ELSE

// No file uploaded, assign default image filename

data['photo'] = 'default.png';

END IF

// Simpan data pengguna ke dalam database menggunakan model

result = User\_model.save\_user(data); // Save user data using the model

IF result THEN

// Jika penyimpanan berhasil

response['message'] = 'Registration successful.';

ELSE

// Jika penyimpanan gagal

response['message'] = 'Registration failed.';

END IF

// Kirim respons JSON untuk notifikasi SweetAlert

RETURN json\_encode(response);

END

### Pseudocode Edit User

Judul: Program Edit User

Deklarasi:

var id\_user: integer;

var data: array;

var password, confpassword: string;

var updated: boolean;

var error: array;

var response: array;

Implementasi

BEGIN

// Tangkap data dari formulir pengeditan

data['username'] = form input 'username';

data['fullname'] = form input 'fullname';

data['usertype'] = form input 'usertype';

// Tangkap data password jika diubah

password = form input 'password';

confpassword = form input 'confpassword';

IF password is not empty AND password = confpassword THEN

// Hash password

data['password'] = password\_hash(password, PASSWORD\_DEFAULT);

ELSEIF password ≠ confpassword THEN

// Jika konfirmasi password tidak sesuai, tampilkan pesan kesalahan

response['message'] = "Password confirmation doesn't match.";

RETURN json\_encode(response);

END IF

// Check if a file is uploaded

IF file 'photo' is uploaded THEN

// File is uploaded, include it in the data

config['upload\_path'] = './uploads/users/';

config['allowed\_types'] = 'gif|jpg|png';

config['file\_name'] = file name 'photo';

LOAD library 'upload' with config;

IF upload 'photo' successful THEN

data['photo'] = uploaded file name;

ELSE

// Jika gagal mengunggah gambar, tangani kesalahan

error = upload errors;

// Tampilkan pesan kesalahan jika perlu

response['message'] = error['error'];

RETURN json\_encode(response);

END IF

END IF

// Update data pengguna ke dalam database menggunakan model

updated = User\_model.edit\_user(id\_user, data);

IF updated THEN

// Jika pembaruan berhasil

response['message'] = 'User updated successfully.';

ELSE

// Jika pembaruan gagal

response['message'] = 'Failed to update user.';

END IF

// Kirim respons JSON untuk notifikasi SweetAlert

RETURN json\_encode(response);

END

### Pseudocode Delete User

Judul: Program Delete User

Deklarasi:

var id\_user: integer;

var deleted: boolean;

var response: array;

Implementasi

BEGIN

// Panggil metode model untuk menghapus pengguna berdasarkan id\_user

deleted = User\_model.delete\_user(id\_user);

IF deleted THEN

// Jika penghapusan berhasil

response['message'] = 'User deleted successfully.';

ELSE

// Jika penghapusan gagal

response['message'] = 'Failed to delete user.';

END IF

// Kirim respons JSON untuk notifikasi SweetAlert

RETURN json\_encode(response);

END

### Pseudocode Create Data UMKM

Judul: Program Create Data UMKM

Deklarasi: var data, dataToInsert : array;

Implementasi

BEGIN

// Load Umkm\_model

LOAD Umkm\_model;

// Call the get\_umkms method from Umkm\_model to get the list of umkm\_name

data['umkms'] = Umkm\_model.get\_umkms();

// If the form is submitted

IF form is submitted THEN

// Prepare data to be inserted into the database

dataToInsert['nama\_pemilik'] = form input 'nama\_pemilik';

dataToInsert['name\_category'] = form input 'name\_category';

// Load Category\_model

LOAD Category\_model;

// Add the new category to the database

Category\_model.create\_kategori(dataToInsert);

// Redirect to the main category page

REDIRECT to 'category/kategori1';

ELSE

// If not, display the form to create a new category with the list of umkm\_name

LOAD VIEW 'umkm/kategori1' with data;

END IF

END

### Pseudocode Edit Data UMKM

Judul: Program Edit Data UMKM

Deklarasi:

var id: integer;

var data: array;

Implementasi

BEGIN

// Jika formulir dikirim

IF form is submitted THEN

// Menyiapkan data untuk dimasukkan ke dalam database

data['nama\_pemilik']= form input 'nama\_pemilik';

data['name\_category'] = form input 'name\_category';

// Memperbarui kategori di database

Category\_model.update\_kategori(id, data);

// Redirect ke halaman utama kategori

REDIRECT to 'category/index';

ELSE

// Mengambil detail kategori berdasarkan ID

data['category'] = Category\_model.get\_kategori\_by\_id(id);

// Menampilkan formulir edit dengan data kategori

LOAD VIEW 'umkm/kategori1/edit' with data;

END IF

END

### Pseudocode Delete Data UMKM

Judul: Program delete Data UMKM

Deklarasi:

var id: integer;

var result: boolean;

var response: array;

Implementasi

BEGIN

// Panggil method untuk menghapus kategori dari model

result = Category\_model.delete\_kategori(id);

// Jika penghapusan berhasil

IF result THEN

response['message'] = 'Category deleted successfully.';

ELSE

// Jika penghapusan gagal

response['message'] = 'Failed to delete category.';

END IF

// Mengembalikan respons dalam format JSON

RETURN json\_encode(response);

END

### Pseudocode Create Promosi

Judul: Program Create Promosi

Deklarasi:

var username, fasilitasi\_promosi, hambatan\_memasarkan\_produk, bantuan\_dibutuhkan, berminat\_bazar\_ramadhan, berminat\_pelatihan\_online : string;

var data: array;

var response: array;

var all\_users: array;

var usernames: array;

var user: object;

Implementasi

BEGIN

// Tangkap data dari formulir

username = form input 'username';

fasilitasi\_promosi = form input 'fasilitasi\_promosi';

hambatan\_memasarkan\_produk = form input 'hambatan\_memasarkan\_produk';

bantuan\_dibutuhkan = form input 'bantuan\_dibutuhkan';

berminat\_bazar\_ramadhan = form input 'berminat\_bazar\_ramadhan';

berminat\_pelatihan\_online = form input 'berminat\_pelatihan\_online';

// Masukkan data ke dalam array

data = {

'username': username,

'fasilitasi\_promosi': fasilitasi\_promosi,

'hambatan\_memasarkan\_produk': hambatan\_memasarkan\_produk,

'bantuan\_dibutuhkan': bantuan\_dibutuhkan,

'berminat\_bazar\_ramadhan': berminat\_bazar\_ramadhan,

'berminat\_pelatihan\_online': berminat\_pelatihan\_online

};

// Insert data promosi ke dalam database menggunakan model

IF Promosi\_model.insert\_promosi(data) THEN

response['message'] = 'Promosi added successfully.';

response['promosi'] = data;

ELSE

response['message'] = 'Failed to add Promosi.';

END IF

// Ambil semua pengguna dari database

all\_users = User\_model.get\_all\_users();

usernames = array();

// Masukkan username dari setiap pengguna ke dalam array

FOR EACH user IN all\_users DO

usernames.append(user.username);

END FOR

// Masukkan array usernames ke dalam response

response['usernames'] = usernames;

// Kirim respons dalam format JSON

RETURN json\_encode(response);

END

### Pseudocode Update Promosi

Judul: Program Update Promosi

Deklarasi:

var id: integer;

var username, fasilitasi\_promosi, hambatan\_memasarkan\_produk, bantuan\_dibutuhkan, berminat\_bazar\_ramadhan, berminat\_pelatihan\_online : string;

var data: array;

var response: array;

var all\_users: array;

var usernames: array;

var user: object;

Implementasi

BEGIN

// Retrieve data from POST request parameters

username = form input 'username';

fasilitasi\_promosi = form input 'fasilitasi\_promosi';

hambatan\_memasarkan\_produk = form input 'hambatan\_memasarkan\_produk';

bantuan\_dibutuhkan = form input 'bantuan\_dibutuhkan';

berminat\_bazar\_ramadhan = form input 'berminat\_bazar\_ramadhan';

berminat\_pelatihan\_online = form input 'berminat\_pelatihan\_online';

// Prevent fasilitasi\_promosi from NULL in DB

IF fasilitasi\_promosi IS NULL THEN

fasilitasi\_promosi = 'Tidak';

END IF

// Construct data array

data = {

'username': username,

'fasilitasi\_promosi': fasilitasi\_promosi,

'hambatan\_memasarkan\_produk': hambatan\_memasarkan\_produk,

'bantuan\_dibutuhkan': bantuan\_dibutuhkan,

'berminat\_bazar\_ramadhan': berminat\_bazar\_ramadhan,

'berminat\_pelatihan\_online': berminat\_pelatihan\_online

};

// Update promotional activity in the database

IF Promosi\_model.update\_promosi(id, data) THEN

response['message'] = 'Promosi updated successfully.';

response['promosi'] = data;

ELSE

response['message'] = 'Failed to update Promosi.';

END IF

// Get all usernames from User\_model and add to the response array

all\_users = User\_model.get\_all\_users();

usernames = array();

FOR EACH user IN all\_users DO

usernames.append(user.username);

END FOR

// Add usernames to response array

response['usernames'] = usernames;

// Encode the response array as JSON and echo it

RETURN json\_encode(response);

END

### Pseudocode Delete Promosi

Judul: Program Delete Promosi

Deklarasi:

Implementasi

## Testing

Pada tahap ini, penguji memastikan bahwa semua kebutuhan telah didokumentasikan dengan jelas dan benar-benar mencerminkan kebutuhan pengguna di Kantor Dinas Perdagangan dan Perindustrian Kabupaten Bandung. Pengujian ini melibatkan peninjauan dokumen spesifikasi kebutuhan untuk memeriksa kelengkapan dan konsistensi.

### Rencana Pengujian

Pada tahap ini hanya dilakukan pengujian dengan mengamati eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian bermaksud untuk memeriksa kesesuaian antara sistem dengan *requirement.* Pengujian ini menggunakan metode *black box testing* yang berfokus pada persyaratan fungsional perangkat lunak tanpa menguji design dan program. Penjelasan tentang pengujian akan dijelaskan pada table IV.14.

Tabel VI. 14 Pengujian Sistem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Requirement*** | **Deskripsi *Testing*** | **Skenario**  ***Testing*** | **Pengguna** |
| 1 | Login | Mengisi form login dengan benar | Mengisi *username* dan *password* yang valid | Admin dan Pemilik UMKM |
| *Form Login* belum diisi atau salah | Tidak mengisi *form login* atau mengisi dengan *username* dan *password* yang tidak valid |
| 2 | *Dashboard* | Melihat data UMKM | Mengakses dashboard | Admin dan Pemilik UMKM |
| 3 | Kelola data UMKM | Menambahkan, Mencari, Mengubah, dan Menghapus data UMKM | 1. Admin dan Pemilik UMKM menambahkan data UMKM kedalam basis data  2. Admin mencari data UMKM yang sudah dimasukkan dan disimpan kedalam basis data  3. Admin dapat mengubah data UMKM yang sudah tersimpan kedalam database | Admin dan Pemilik UMKM |
| 4 | Kelola data Pemilik UMKM | Menambahkan, Mencari, Mengubah, dan Menghapus data Pemilik UMKM | 1. Admin dan Pemilik UMKM menambahkan data pemilik UMKM kedalam basis data  2. Admin mencari data pemilik UMKM yang sudah dimasukkan dan disimpan kedalam basis data  3. Admin dapat mengubah data pemilik UMKM yang sudah tersimpan kedalam database | Admin dan Pemilik UMKM |
| 5 | Kelola data Promosi | Menambahkan, Mencari, Mengubah, dan Menghapus Promosi | 1. Admin dan Pemilik UMKM menambahkan Promosi kedalam basis data  2. Admin dan Pemilik UMKM mencari data promosi yang sudah dimasukkan dan disimpan kedalam basis data  3. Admin dan pemilik UMKM dapat mengubah promosi yang sudah tersimpan kedalam database | Admin dan Pemilik UMKM |
| 6 | Kelola data Platform | Menambahkan, Mencari, Mengubah, dan Menghapus Platform | 1. Admin dan Pemilik UMKM menambahkan Platform kedalam basis data  2. Admin dan Pemilik UMKM mencari data platform yang sudah dimasukkan dan disimpan kedalam basis data  3. Admin dan pemilik UMKM dapat mengubah platform yang sudah tersimpan kedalam database | Admin dan Pemilik UMKM |

### Hasil Pengujian

Tabel VI. 15 Hasil Pengujian Login

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| Mengisi *Username* dan *Password* secara valid | Jika data *Login* valid maka akan masuk kedalam menu utama | Berhasil |
| Tidak mengisi *form login* atau mengisi dengan *username* dan *password* yang tidak valid | Sistem menampilakn pesan kesalahan | Berhasil |

Tabel VI. 16 Hasil Pengujian Kelola UMKM

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| Admin dan Pemilik UMKM mengisi form UMKM | Jika data yang dibutuhkan sudah lengkap, sistem akan memproses data tersebut. | Berhasil |
| Data form tidak diisi dengan lengkap | Sistem akan menampilkan pesan kesalahan | Berhasil |
| Admin mencari data UMKM | Sistem akan menampilkan data yang dicari | Berhasil |
| Admin mengubah data UMKM | Jika data UMKM tersebut diubah maka otomatis data tersebut akan terubah di table dan di database | Berhasil |
| Admin menghapus data UMKM | Sistem akan menampilan *pop up massage* jika data UMKM dihapus maka otomatis data tersebut akan terhapus ditable dan database. | Berhasil |

Tabel VI. 17 Hasil Pengujian Kelola Pemilik UMKM

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| Admin dan Pemilik UMKM mengisi form pemilik UMKM | Jika data yang dibutuhkan sudah lengkap, sistem akan memproses data tersebut. | Berhasil |
| Data form tidak diisi dengan lengkap | Sistem akan menampilkan pesan kesalahan | Berhasil |
| Admin mencari data pemilik UMKM | Sistem akan menampilkan data yang dicari | Berhasil |
| Admin mengubah data pemilik UMKM | Jika data pemilik UMKM tersebut diubah maka otomatis data tersebut akan terubah di table dan di database | Berhasil |
| Admin menghapus data pemilik UMKM | Sistem akan menampilan *pop up massage* jika data pemilik UMKM dihapus maka otomatis data tersebut akan terhapus ditable dan database. | Berhasil |

Tabel VI. 18 Hasil Pengujian Kelola Promosi

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| Admin dan Pemilik UMKM mengisi form promosi | Jika data yang dibutuhkan sudah lengkap, sistem akan memproses data tersebut. | Berhasil |
| Data form tidak diisi dengan lengkap | Sistem akan menampilkan pesan kesalahan | Berhasil |
| Admin dan pemilik UMKM mencari data promosi | Sistem akan menampilkan data yang dicari | Berhasil |
| Admin mengubah data promosi | Jika data promosi tersebut diubah maka otomatis data tersebut akan terubah di table dan di database | Berhasil |
| Admin menghapus data promosi | Sistem akan menampilan *pop up massage* jika data promosi dihapus maka otomatis data tersebut akan terhapus ditable dan database. | Berhasil |

Tabel VI. 19 Hasil Pengujian Kelola Platform

|  |  |  |
| --- | --- | --- |
| **Kasus dan Hasil Uji** | | |
| **Skenario Pengujian** | **Hasil yang diharapkan** | **Kesimpulan** |
| Admin dan Pemilik UMKM mengisi form platform | Jika data yang dibutuhkan sudah lengkap, sistem akan memproses data tersebut. | Berhasil |
| Data form tidak diisi dengan lengkap | Sistem akan menampilkan pesan kesalahan | Berhasil |
| Admin dan pemilik UMKM mencari data platfrom | Sistem akan menampilkan data yang dicari | Berhasil |
| Admin mengubah data platform | Jika data platform tersebut diubah maka otomatis data tersebut akan terubah di table dan di database | Berhasil |
| Admin menghapus data platform | Sistem akan menampilan *pop up massage* jika data platform dihapus maka otomatis data tersebut akan terhapus ditable dan database. | Berhasil |

# SIMPULAN DAN SARAN

## Simpulan

Dari hasil perancangan dan pembuatan aplikasi laporan kerja praktik dengan judul Sistem Informasi UMKM Berbasis Codeigniter (Studi Kasus : Kantor Dinas Perdagangan dan Perindustrian Kabupaten Bandung). Maka dapat diambil kesimpulan sebagai berikut:

1. Pembangunan sistem informasi pengelolaan data UMKM berbasis web mampu mengurangi risiko kehilangan informasi sekaligus meningkatkan aksesibilitas dan keakuratan data.
2. Sistem ini juga berperan dalam menyediakan pelayanan publik yang lebih baik, efektif, dan efisien.
3. Selain itu, sistem informasi ini mempermudah Disperindag dalam melakukan pendataan, sehingga proses pengelolaan data menjadi lebih terstruktur dan terorganisir.

## Saran

Adapun saran terhadap pengguna sistem sebagai pengembangan sistem informasi ini antara lain sebagai berikut:

Desain antarmuka perlu ditingkatkan agar menjadi lebih baik dan lebih indah guna memberikan kenyaman terhadap pengguna.

Untuk peneliti selanjutnya sistem yang akan dikembangkan diharapkan dapat menambahkan fitur yang up to date kedalam aplikasi.

DAFTAR PUSTAKA

A.S, R. d. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek.* Bandung: Informatika.

Abdul, K. (2003). *Pengenalan Sistem Informasi.* Yogyakarta: CV.ANDI.

Abdulloh, R. (2015). *Web Programing is Easy.* Jakarta: PT Elex Media Komputindo. .

Buana. (2014). *Aplikasi XAMPP.* Jakarta: Andi.

Daqiqil, i. (2011). *Framework CodeIgniter : Sebuah Panduan dan Best Practice.* Pekanbaru.

Darmawan, D. &. (2013). *Sistem Informasi Manajemen.* Bandung: PT Remaja Rosdakarya.

Hariyanto, B. (2012). *Esensi-Esensi Bahasa Pemrograman Java : Revisi Keempat.* Bandung: Informatika.

Josi, A. (2017). PENERAPAN METODE PROTOTIPING DALAM PEMBANGUNAN WEBSITE DESA. *JTI, Vol 9 No.1*, 51-52.

Kadir, A. (2014). *Pengenalan Sistem Informasi Edisi Revisi.* Yogyakarta: CV.ANDI.

Kusumo, E. W. (2016). "Penerapan Metode Unified Modeling Language (UML) Berbasis Desktop Pada Sistem Pengolahan Kas Kecil Studi Kasus Pada PT Indo Mada Yasa Tanggerang". *J.Inform., vol. 5, no. 1, pp*, 24-36.

M.Shalahuddin., R. A. (2019). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek Edisi Revisi.* Bandung: Informatika.

Mulyani, S. (2016). *Metode Analisis dan Perangcangan Sistem.* Bandung: Abdi Sistematika.

Muslihudin, M. d. (2016). *"Analasis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML".* Yogyakarta: CV.ANDI OFFSET.

Nugroho. (2013). *Mengenal XAMPP Awal.* Yogyakarta: Mediakom.

Nugroho, B. (2008). *Membuat Sistem Informasi Penjualan Berbasis Web dengan Menggunakan PHP dan Mysql.* Yogyakarta: Gava Media.

Nurcahyono. (2013:1-2). *MySQL: Sistem Manajemen Basis Data Relasional.* Jakarta: Penerbit Informatika.

Prof. Dr. Jogiyanto HM, M. A. (2017). *MODEL KESUKSESAN SISTEM TEKNOLOGI INFORMASI.* CV. ANDI OFFSET.

Raharjo, B. (2015). *Belajar Otodidak Framework CodeIgniter.* Bandung: Informatika Bandung.

Raharjo, B. d. (2014). *Modul Pemrograman Web HTML, PHP, dan MySQL.* Bandung: Modula.

Tyoso, J. S. (2016). *Sistem Informasi Manajemen.* Deepublish.

LAMPIRAN A: STRUKTUR ORGANISASI

LAMPIRAN B: HASILINTERVIEW

LAMPIRAN C: KODE PROGRAM

class halaman\_awal {

}