

**SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK**

**Laundry Online (On-Laundry)**

untuk:

Umum

Dipersiapkan oleh:

Gabriel Almayda Sudarisman (1301170398)

Muhammad Alkakfi Khuzaimy Abdullah (1301174048)

Rafie Afif Andika (1301173341)

Yulian Andyka Cahya Fauzan (1301173357)

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| **Program Studi S1**  **Teknik**  **Informatika**  **-**  **Fakultas**  **Informatika** | **Nomor Dokumen** | | **Halaman** |
| ***SKPL-xxx*** | | **35** |
| **Revisi** | *<nomor revisi>* | *Tgl: <isi tanggal>* |

**Daftar Perubahan**

|  |  |
| --- | --- |
| **Revisi** | **Deskripsi** |
| **A** |  |
| **B** |  |
| **C** |  |
| **D** |  |
| **E** |  |
| **F** |  |
| **G** |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | - | A | B | C | D | E | F | G |
| TGL |  |  |  |  |  |  |  |  |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

**Daftar Halaman Perubahan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Halaman** | **Revisi** | **Halaman** | **Revisi** |
|  |  |  |  |

**Daftar Isi**

[1. Pendahuluan 7](#_Toc6814391)

[1.1 Tujuan Penulisan Dokumen 7](#_Toc6814392)

[1.2 Ruang Lingkup / Cakupan Dokumen 7](#_Toc6814393)

[1.3 Definisi, Singkatan, dan Akronim 7](#_Toc6814394)

[1.4 Referensi 8](#_Toc6814395)

[2. Deskripsi Rinci Perangkat Lunak 8](#_Toc6814396)

[2.1 *Statement of Objective* Perangkat Lunak 8](#_Toc6814397)

[2.2 Perspektif dan Fungsi Perangkat Lunak 9](#_Toc6814398)

[2.3 Profil dan Karakteristik Penggunaan 9](#_Toc6814399)

[2.4 Lingkungan Operasi 10](#_Toc6814400)

[2.5 Batasan Perangkat Lunak / Sistem 10](#_Toc6814401)

[2.6 Asumsi dan Dependensi 11](#_Toc6814402)

[3. Deskripsi Rinci Perangkat Lunak 12](#_Toc6814403)

[3.1 Deskripsi Kebutuhan 12](#_Toc6814404)

[3.1.1 *Kebutuhan Fungsional* 12](#_Toc6814405)

[3.1.2 *Kebutuhan Non-Fungsional* 14](#_Toc6814406)

[3.2 Pemodelan Analisis 15](#_Toc6814407)

[3.2.1 *Usecase Diagram* 15](#_Toc6814408)

[3.3 Antarmuka Pengguna 32](#_Toc6814409)

[3.3.1 LOGIN 33](#_Toc6814410)

[3.3.2 REGISTRASI 33](#_Toc6814411)

[3.3.3 HOMEPAGE 34](#_Toc6814412)

[3.3.4 ORDER 35](#_Toc6814413)

[3.3.5 VIEW PEMESANAN 36](#_Toc6814414)

[3.3.6 PEMBAYARAN 37](#_Toc6814415)

[3.3.7 INPUT DATA LAUNDRY 38](#_Toc6814416)

[3.3.8 VIEW DATA LAUNDRY 38](#_Toc6814417)

[3.3.9 DATA *CUSTOMER* 39](#_Toc6814418)

[3.3.10 DATA PEMASUKAN 39](#_Toc6814419)

[3.3.11 EDIT PROFILE 40](#_Toc6814420)

[3.4 Antarmuka Perangkat Keras 40](#_Toc6814421)

[3.5 Antarmuka Perangkat Lunak 41](#_Toc6814422)

[3.6 Antarmuka Komunikasi 41](#_Toc6814423)

[Lampiran A: Daftar Kata-Kata Sukar 41](#_Toc6814424)

[Lampiran B: Analysis Models 42](#_Toc6814425)

[4.1 Flowchart 43](#_Toc6814426)

[4.2 ER Diagram 44](#_Toc6814427)

[4.3 Skema Relasi 45](#_Toc6814428)

# 1. Pendahuluan

## 1.1 Tujuan Penulisan Dokumen

Dokumen SKPL ini berisi Spesifikasi Kebutuhan Perangkat Lunak untuk Laundry Online (On-Laundry).

Adapun tujuan dari dibuatnya dokumen ini ialah sebagai berikut:

1. Untuk memberikan penjelasan lengkap dan gambaran umum mengenai kebutuhan perangkat lunak yang akan dibangun,
2. Dokumen ini akan digunakan sebagai acuan rancangan perangkat lunak yang akan dijadikan bahan evaluasi oleh pengembang perangkat lunak sistem informasi laundry online pada saat pengembangan.

Dengan adanya dokumen ini, diharapkan pengembangan perangkat lunak Laundry Online (On-Laundry) menjadi lebih terarah

## 1.2 Ruang Lingkup / Cakupan Dokumen

Perangkat lunak yang akan dikembangkan berisi suatu sistem informasi laundry online berbasis *website* untuk melakukan pemesanan layanan laundry secara *online*. Perangkat lunak ini menghubungkan antara *customer* yang malas antar jemput laundry atau bahkan tidak mengetahui letak tempat laundry dengan sebuah tempat laundry yang bisa mengurus barang *customer*, hanya dengan memesannya dimanapun *customer* berada. Tujuannya untuk menghemat waktu dan mempermudah *customer* karena tidak perlu repot datang ke tempat laundry.

## 1.3 Definisi, Singkatan, dan Akronim

1. SKPL

Spesifikasi Kebutuhan Perangkat Lunak adalah kondisi atau kemampuan yang harus dimilliki untuk memenuhi apa yang diisyaratkan oleh perangkat lunak atau diinginkan oleh pemakai.

1. ERD

Entity Relationship Diagram adalah salah satu model yang digunakan untuk mendesain *database* dengan tujuan menggambarkan data yang berelasi pada sebuah *database*.

1. DPPL

Dokumen Deskripsi Perancangan Perangkat Lunak merupakan dokumen deskripsi dari perancangan perangkat lunak yang akan dikembangkan.

## 1.4 Referensi

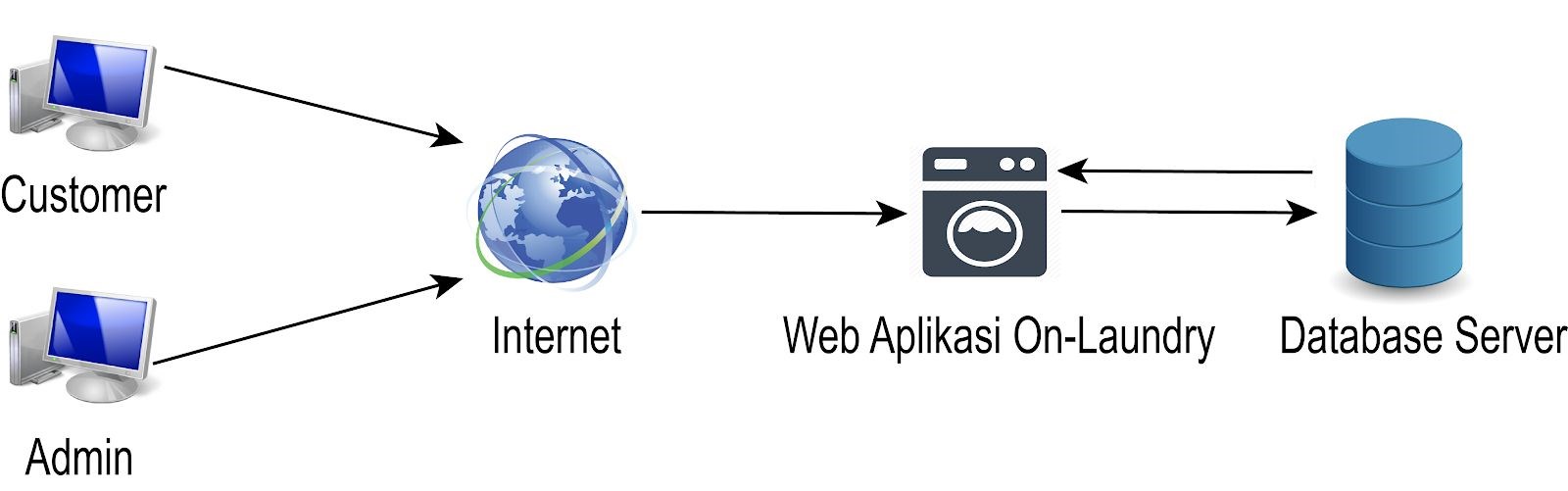
* Template SKPL yang diberikan oleh dosen
* Budi Jatmiko, dkk. *Spesifikasi Kebutuhan Perangkat Lunak Aplikasi Komunitas Auditor Sistem Informasi (ISAC) Berbasis Web*​. Program Studi Manajemen Informatika Universitas Telkom. November 2017.
* https://www.wikipedia.org/

# 2. Deskripsi Rinci Perangkat Lunak

## 2.1 *Statement of Objective* Perangkat Lunak

On-Laundry adalah Perangkat lunak berbasis *website* yang berfungsi sebagai sarana transaksi jasa Laundry yang dilakukan secara *online* antara pelanggan dan juga pegawai. Produk ini merupakan bisa dikatakan produk yang sengaja kami buat khusus untuk daerah Bojongsoang, produk ini kami bangun dengan tujuan untuk memudahkan pelanggan dan para pengusaha laundry untuk melakukan transaksi jasa secara lebih modern, cepat dan praktis yang di khususkan untuk daerah Bojongsoang, secara produk ini akan memberi keuntungan mulai dari berbagai pihak mulai dari pengusaha yang akan di mudahkan dengan mendapatkan pelanggan yang banyak dan juga untuk pelanggan yang tidak perlu repot-repot untuk datang ke tempat laundry.

On-Laundry dapat di akses dengan aplikasi *browser* apa saja dengan berbagai media mulai dari smartphone hingga komputer yang terhubung dengan internet, On-Laundry hanya dapat diakses oleh pengguna dengan menggunakan *username* dan *password* yang terlebih dahulu di daftarkan.



*Admin* akan mengelola data pelanggan, data usaha laundry, data pesanan, data transaksi, dan data pembayaran pada sistem. Sementara, pelanggan hanya bisa memesan atau melakukan pesan antar jemput cucian pada wilayah Bojongsoang saja. *Admin* dapat melakukan semua pengelolaan data antara pelanggan dan pegawai dan melakukan verifikasi pembayaran yang akan dilanjutkan kepada pegawai. Proses transaksi pada aplikasi ini dapat menerima pembayaran tunai dan non-tunai, yaitu berupa pembayaran yang dilakukan pelanggan langsung ke tempat laundry dengan menggunakan uang cash dan bisa juga menggunakan transfer bank. Berikut *statement of objective* dari perangkat lunak On-Laundry:

●Menjadi perantara pelaksanaan transaksi jasa Laundry antara pelanggan dan pengusaha laundry ●Membantu Pelanggan untuk mencari laundry yang sesuai dengan kebutuhan yang sesuai.

## 2.2 Perspektif dan Fungsi Perangkat Lunak

Fungsi dari sistem aplikasi ini adalah untuk menghubungkan antara *customer* yang ingin mencuci pakaian tetapi sibuk untuk mencuci pakaiannya dan *staff* akan menjemput pakaian *customer* yang akan dicuci. Aplikasi ini juga memberikan informasi terkait pakaian yang sedang dalam masa penjemputan, pencucian, maupun pengembalian, dan aplikasi memberikan informasi juga terkait dengan berat pakaian *customer* dan rincian biaya dari perhitungan berat pakaian *customer*. *Staff* berperan sebagai *admin* dan menjemput atau pengembalian pakaian *customer* dalam aplikasi on-laundry.

## 2.3 Profil dan Karakteristik Penggunaan

|  |  |  |  |
| --- | --- | --- | --- |
| Pemakai Sistem | Hak Akses | Tugas | Syarat |
| *Admin* | Mengelola data *customer*, data laundry, data transaksi. Melihat data *customer* yang melakukan transaksi | Mengelola data, memberikan  verifikasi terhadap kegiatan laundry. | *Admin* harus *login* untuk mengelola data laundry. |
| Mamang Deliv | Melihat data pesanan pelanggan. | Mengantarkan cucian kepada pelanggan. | Mamang Deliv akan mengantarkan cucian dalam cakupan Bojongsoang saja.    Cucian akan diantar  apabila Pembayaran  telah selesai dilakukan. |
| *Customer* | Menginputkan data diri, data pesanan, dan transaksi pembayaran. | Melakukan input data diri sebagai syarat melakukan transaksi jasa laundry online. dan melakukan konfirmasi apabila cucian telah diantar ke tempat. | Pelanggan harus membuat akun dan login terlebih dahulu.  Jika ingin melakukan pemesanan.  Pelanggan yang memesan harus pada daerah Bojongsoang. |

## 2.4 Lingkungan Operasi

Perangkat lunak ini akan beroperasi pada lingkungan komputer, dan juga *mobile*, melalui aplikasi web browser yang sudah terhubung dengan internet, dalam artian lain dapat diakses menggunakan *platform* PC, MAC, dan Smartphone. Sistem operasi yang *compatible* digunakan adalah Microsoft Windows XP/Vista/7/8/10.

## 2.5 Batasan Perangkat Lunak / Sistem

Perangkat lunak pada sisi server ​ yang dibutuhkan untuk On-Laundry adalah:​

* Sistem operasi : Microsoft Windows XP/Vista/7/8/10
* *Web server* : Personal Web Server (PWS)
* *Scripting language* : Hypertext Preprocessor (PHP)
* DBMS : MySQL

Perangkat lunak pada sisi client yang dibutuhkan untuk On-Laundry adalah jika menggunakan perangkat ​*mobile*​:

* Sistem operasi : Android/ IOS/Windows 10 for mobile
* *Web browser* : Chrome, Mozila Firefox, Opera Mini, UC Browser

Perangkat lunak pada sisi client yang dibutuhkan untuk on-Laundry adalah jika menggunakan perangkat komputer :

* Sistem operasi : Microsoft Windows XP/Vista/7/8/10/MacOS
* *Web browser* : Internet Explorer, Chrome, Mozila Firefox, Opera mini

## 2.6 Asumsi dan Dependensi

Berikut ini adalah ketentuan khusus pada kondisi tertentu:

* *Admin* mengecek proses dari sistem yang melakukan transaksi pembayaran, jika *customer* belum dapat bertransaksi, maka pakaian akan di-*keep* dahulu menunggu transaksi oleh *customer*;
* Sistem juga akan menentukan yang dapat dilakukan penjemputan adalah berdomisili >10 km dari tempat laundry.

# 3. Deskripsi Rinci Perangkat Lunak

## 3.1 Deskripsi Kebutuhan

### 3.1.1 *Kebutuhan Fungsional*

|  |  |  |  |
| --- | --- | --- | --- |
| NO | KODE  KEBUTUHAN | FUNGSI | DESKRIPSI |
| 1. | FR-01 | LOGIN | Sistem menerima inputan *username* dan password dari *customer* |
| 2. | FR-02 | REGISTRASI AKUN | Sistem menerima data registrasi berupa Nama, *email*, *username*, *password* untuk *customer* yang tidak memiliki akun |
| 3. | FR-03 | INPUT  PEMESANAN | Sistem menerima inputan berupa nama, no telp, alamat *customer*, jenis layanan, jenis parfum untuk dikirimkan kepada *admin* |
| 4. | FR-04 | INPUT DATA LAUNDRY | Sistem menerima inputan dari *admin* berupa jenis layanan |
|  |  |  | laundry, dan jenis parfum yang nantinya dapat dipilih oleh *customer* |
| 5. | FR-05 | PEMBAYARAN | Sistem dapat menampilkan tagihan pembayaran dan menerima metode pembayaran yang dipilih oleh *customer* |
| 6. | FR-06 | VERIFIKASI PEMBAYARAN | Sistem menerima masukan data konfirmasi pembayaran dari *customer* dan akan  diverifikasi oleh  *admin* |
| 7. | FR-07 | BUKTI  PEMBAYARAN | Sistem menerima hasil pembayaran dari *customer* dan dapat memberikan keluaran data total transaksi kepada *customer* |

### 3.1.2 *Kebutuhan Non-Fungsional*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Quality | Kode Kebutuhan | Deskripsi |
| 1. | *Operational Time* | NFR-01 | On-Laundry dapat menerima pesan antar laundry setiap hari, penjemputan laundry setiap hari kecuali hari minggu, dengan jam operasional 06:00-22:00 |
| 2. | *Service Guarantee* | NFR-02 | On-Laundry memberikan jaminan pelayanan jika terdapat kekurangan atau kesalahan pelayanan, *user* dapat *complain* kedalam sistem, dan uang akan dikembalikan |
| 3. | *Free Use* | NFR-03 | On-Laundry dapat diakses di *browser* apapun, dengan perangkat apapun yang dapat browsing |

## 3.2 Pemodelan Analisis

### 3.2.1 *Usecase Diagram*

A close up of a map

Description generated with very high confidence

#### 3.2.1.1 Usecase Scenario #1

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Login | |
| Deskripsi | Fungsi ini digunakan untuk masuk kedalam sistem | |
| Pre-Kondisi | *User* ingin menggunakan aplikasi dan belum masuk ke sistem | |
| Post-Kondisi | *User* sudah masuk ke sistem dan dapat menggunakan fitur yang ada | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
|  | 1. Menampilkan halaman awal sebelum masuk kedalam aplikasi |
| 2. Mengklik tab “Login” |  |
|  | 3. Menampilkan form login |
| 4. Menginputkan *user*name,  dan password, lalu mengklik “Login” |  |
|  | 5. Jika *user*name, dan password yang diinputkan itu valid maka lanjut ke tahap 6, namun jika tidak, maka akan menampilkan notifikasi “Anda belum terdaftar, Silahkan daftar terlebih dahulu, lalu login kembali” |
|  |  | 6. Menampilkan halaman awal atau menu aplikasi |

#### 3.2.1.2 Usecase Scenario #2

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Registrasi | |
| Deskripsi | Fungsi ini digunakan untuk *user* yang belum memiliki akun atau ingin mendaftarkan akun baru | |
| Pre-Kondisi | *User* ingin memiliki akun dan belum melakukan registrasi | |
| Post-Kondisi | *User* sudah melakukan registrasi dan sudah bisa menggunakan akunnya di dalam aplikasi | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
|  | 1 Menampilkan halaman awal sebelum masuk kedalam aplikasi |
| 2. Mengklik tab “Daftar” |  |
|  |  | 3. Menampilkan form pendaftaran |
| 4. Menginputkan Nama,  Email, No. Handphone,  Alamat, *User*name, dan  Password lalu mengklik  “Daftar” |  |
|  | 5. Menampilkan notifikasi “Selamat! Akun anda telah terdaftar, silahkan login”, lanjut ke tahap 6 |
|  | 6. Menampilkan form login |

#### 3.2.1.3 Usecase Scenario #3

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Input Pesanan | |
| Deskripsi | Fungsi ini digunakan oleh *user* untuk menginputkan pesanan ke dalam sistem | |
| Pre-Kondisi | *User* ingin memesan layanan laundry dan belum melakukan pemesanan | |
| Post-Kondisi | *User* telah melakukan pemesanan dan pesanan akan diteruskan kepada *admin* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Login sebagai *user* |  |
| 2. Membuka menu pemesanan |  |
|  | 3. Menampilkan menu pemesanan |
| 4. Mengklik sub menu  “Pesan” |  |
|  | 5. Menampilkan form input pesanan |
| 6. Menginputkan data pesanan, seperti jenis layanan, dan parfum, kemudian memilih apakah menggunakan alamat rumah atau alamat berbeda, jika alamat yang berbeda maka menginputkan data |  |
|  | alamat, lalu mengklik tombol “pesan” |  |
|  | 7. Sistem menyimpan data pesanan |

#### 3.2.1.4 Usecase Scenario #4

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Edit Pesanan | |
| Deskripsi | Fungsi ini digunakan oleh *user* untuk mengubah atau memperbarui pesanan nya sebelum waktu yang ditentukan | |
| Pre-Kondisi | *User* ingin memperbarui pesanannya dan *user* belum mengedit pesanan | |
| Post-Kondisi | *User* sudah mengedit pesanan dan pembaruan pesanan akan diberitahukan kepada *admin* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Login sebagai *user* |  |
| 2. Membuka menu pemesanan |  |
|  | 3. Menampilkan menu pemesanan |
| 4. Mengklik sub menu  “Lihat pesanan” |  |
|  | 5. Jika *user* belum melakukan pemesanan apapun maka tampil notifikasi “Anda belum membuat pesanan”, jika sudah maka lanjut ke tahap 6 |
|  | 6. Menampilkan list pesanan yang telah dibuat |
| 7. Mengklik salah satu pesanan yang mau diubah |  |
|  | 8. Menampilkan data pesanan yang tadi diklik oleh *user*, dan juga tombol “edit” dan “batal” |
| 9. Mengklik “edit” |  |
|  |  | 10. Jika masih masuk dalam rentang waktu diperbolehkannya perubahan pesanan, maka lanjut ke tahap 11, namun jika tidak maka akan muncul notifikasi “Maaf, anda sudah tidak bisa mengubah pesanan, karena sedang diproses”, lalu kembali ke menu  sebelumnya |
|  | 11. Menampilkan informasi pesanan yang mau diubah dan form pengeditan pesanan |
| 12. Menginputkan ulang semua data pesanan yang diinginkan seperti pada proses input pesanan |  |
|  |  | 13. Sistem menyimpan data pesanan yang baru dan memberitahukannya kepada *admin* |

#### 3.2.1.5 Usecase Scenario #5

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Delete Pesanan | |
| Deskripsi | Fungsi ini digunakan oleh *user* untuk membatalkan pesanan-pesanan yang telah dibuat dalam rentang waktu yang ditentukan | |
| Pre-Kondisi | *User* ingin membatalkan pesanan dan belum melakukan delete pesanan | |
| Post-Kondisi | *User* sudah melakukan delete pesanan dan pembatalan pesanan akan dikirimkan ke *admin* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Login sebagai *user* |  |
|  | 2. Membuka menu pemesanan |  |
|  |  | 3. Menampilkan menu pemesanan |
|  | 4. Mengklik sub menu  “Lihat pesanan” |  |
|  |  | 5. Jika *user* belum melakukan pemesanan apapun maka tampil notifikasi “Anda belum membuat pesanan”, jika sudah maka lanjut ke tahap 6 |
|  |  | 6. Menampilkan list pesanan yang telah dibuat |
|  | 7. Mengklik salah satu pesanan yang mau dibatalkan |  |
|  | 8. Menampilkan data pesanan yang tadi diklik oleh *user*, dan juga tombol “edit” dan “batal” |
| 9. Mengklik “batal” |  |
|  |  | 10. Jika masih masuk dalam rentang waktu diperbolehkannya |
|  | pembatalan, maka lanjut ke tahap 11, namun jika tidak maka akan muncul notifikasi “Maaf, anda tidak bisa membatalkan pesanan karena driver sudah dekat”, lalu kembali ke menu sebelumnya |
|  | 11. Menampilkan confirm box “Apakah anda yakin membatalkan pesanan?” |
| 12. Mengklik “Iya” |  |
|  | 13. Sistem menghapus pesanan yang dipilih, dan |
|  | memberitahukannya kepada *admin* |

#### 3.2.1.6 Usecase Scenario #6

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Input Pembayaran | |
| Deskripsi | Fungsi ini digunakan oleh *user* untuk memilih metode pembayaran yang akan digunakan dalam membayar tagihan | |
| Pre-Kondisi | *User* ingin membayar tagihan dan belum menentukan metode pembayaran | |
| Post-Kondisi | *User* sudah menentukan metode pembayaran dan data pembayaran diteruskan ke *admin* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Login sebagai *user* |  |
|  | 2. Jika pesanan baru dibuat, maka lanjut ke tahap 3, namun jika sudah masuk waktu dimana pesanan harus segera dibayarkan, maka akan langsung ditampilkan tahap 8 yang mana pesanan itu harus segera dibayarkan |
| 3. Membuka menu pembayaran |  |
|  | 4. Menampilkan menu pembayaran |
| 5. Mengklik sub menu  “bayar pesanan” |  |
|  | 6. Menampilkan list pesanan yang belum dibayar |
|  | 7. Mengklik salah satu pesanan yang ingin dibayarkan |  |
|  |  | 8. Menampilkan informasi pesanan, biaya tagihan, dan juga metode pembayaran |
|  | 9. Mengklik salah satu metode pembayaran yang ingin digunakan |  |
|  |  | 10. Menampilkan informasi cara pembayaran dengan metode pembayaran yang diklik oleh *customer*, disertai dengan tagihan dan tenggat waktu pembayaran |
|  |  | 11. Jika *user* telah membayar, maka sistem akan otomatis mengirimkan hasil pembayaran kepada *admin* dan mencetak bukti pembayaran/struk langsung dikirimkan ke email *user* |

#### 3.2.1.7 Usecase Scenario #7

|  |  |  |
| --- | --- | --- |
| Nama Use Case | View Proses Laundry | |
| Deskripsi | Fungsi ini digunakan oleh *user* untuk melihat atau tracking proses yang dialami oleh laundry an nya | |
| Pre-Kondisi | *User* ingin mengetahui laundry sedang dalam tahap atau proses apa dan *user* belum melihat proses laundry | |
| Post-Kondisi | *User* sudah melihat proses laundry dan mengetahui proses berjalan laundry miliknya | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Login sebagai *user* |  |
| 2. Membuka tab view proses pesanan |  |
|  |  | 3. Menampilkan halaman view proses pesanan, yang berisi 4 tahap yang sedang dijalani oleh laundry |

#### 3.2.1.8 Usecase Scenario #8

|  |  |  |
| --- | --- | --- |
| Nama Use Case | View Pesanan | |
| Deskripsi | Fungsi ini digunakan oleh Mamang Deliv untuk mengetahui bahwa adanya pesanan yang harus dijemput dan diantarkan | |
| Pre-Kondisi | *User* ingin mengetahui apakah ada orderan dan belum melihat pesanan | |
| Post-Kondisi | *User* sudah melihat pesanan dan mengirimkan konfirmasi pengambilan kepada *customer* dan *admin* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Login sebagai mamang deliv |  |
| 2.Membuka tab menu view pesanan | 3.Menampilkan tabmenu view pesanan |
| 4.Membuka lihat penjemputan pakaian dari *customer* | 5.Menampilkan view pesanan *customer* |
| 6. Konfirmasi view pesanan | 7. Eksekuksi konfirmasi kepada *customer* bahwa mamang deliv sudah terkonfirmasi melakukan penjemputan pakaian |
|  |  |
|  |  |

#### 3.2.1.9 Usecase Scenario #9

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Input Data Laundry | |
| Deskripsi | Fungsi ini digunakan oleh *admin* untuk memasukan data-data layanan laundry yang mau disediakan | |
| Pre-Kondisi | *Admin* ingin menambahkan data laundry yang baru dan belum menginputkan data laundry | |
| Post-Kondisi | *Admin* sudah menginputkan data laundry dan data laundry yang baru telah tersimpan di *database* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Login sebagai *admin* |  |
| 2. Membuka menu data laundry |  |
|  | 3. Menampilkan menu data laundry |
|  | 4. Mengklik sub menu  “Tambahkan” |  |
|  | 5. Menampilkan form input penambahan data laundry |
| 6. Menginputkan data laundry, seperti jenis layanan, dan parfum, lalu mengklik tombol “tambah” |  |
|  |  | 7. Sistem menyimpan data laundry |

#### 3.2.1.10 Usecase Scenario #10

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Edit Data Laundry | |
| Deskripsi | Fungsi ini digunakan oleh *admin* untuk memperbarui data laundry yang telah ada dikarenakan suatu hal | |
| Pre-Kondisi | *Admin* ingin mengubah data laundry dan belum mengedit data laundry | |
| Post-Kondisi | *Admin* telah mengedit data laundry, dan data laundry yang baru akan disimpan menggantikan data laundry yang lama di *database* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Login sebagai *admin* |  |
| 2.Membuka tab edit data laundry | 3.Menampilkan tab edit data laundry |
| 4. *Admin* memilih data laundry yang akan di edit |  |
| 5. Mengklik sub menu “Edit” |  |
| 6. Jika pengeditan data sudah selesai, maka *admin* dapat mengklik sub menu “save” | 7.Eksekusi data dan  disimpan ke *database* |
|  | 8.Return “Berhasil di save” |

#### 3.2.1.11 Usecase Scenario #11

|  |  |  |
| --- | --- | --- |
| Nama Use Case | Delete Data Laundry | |
| Deskripsi | Fungsi ini digunakan oleh *admin* untuk menghapus satu atau beberapa data laundry yang tersimpan dalam *database* | |
| Pre-Kondisi | *Admin* ingin menghapus data laundry dan belum men-delete data laundry | |
| Post-Kondisi | *Admin* telah men-delete data laundry dan data laundry yang didelete oleh *admin* akan hilang di *database* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Login sebagai *admin* |  |
| 2.Membuka tab delete data laundry | 3.Menampilkan tab delete data laundry |
| 4. Aktor memilih data yang akan di delete |  |
| 5. Mengklik sub menu “Delete” | 6.Eksekusi data dan data telah hilang dari *database* |
|  | 7. Return “Berhasil terhapus” |

##### 3.2.1.12 Usecase Scenario #12

|  |  |  |
| --- | --- | --- |
| Nama Use Case | View Data Laundry | |
| Deskripsi | Fungsi ini digunakan oleh *Admin* untuk melihat list data laundry yang berada dalam *database* | |
| Pre-Kondisi | *Admin* ingin melihat list data laundry dan belum melakukan view data laundry | |
| Post-Kondisi | *Admin* telah melakukan view data laundry dan list data laundry ditampilkan untuk dilihat oleh *admin* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1.Login sebagai *admin* |  |
| 2.Buka Tab view data laundry |  |
|  | 3.Menampilkan Tab view data Laundry |
| 4.Mengklik sub menu “View Data laundry” | 5.Eksekusi data dan melihat list dari *database* |
|  |  | 6.Menampilkan List Data  Laundry |
| 7.Jika sudah melihat, dapat mengklik sub menu “Exit view” |  |

##### 3.2.1.13 Usecase Scenario #13

|  |  |  |
| --- | --- | --- |
| Nama Use Case | View Hasil Pembayaran | |
| Deskripsi | Fungsi ini digunakan oleh *admin* untuk melihat invoice / pembayaran tagihan yang masuk dari *customer* untuk dikonfirmasi | |
| Pre-Kondisi | *Admin* ingin melihat invoice dan belum melakukan view hasil pembayaran | |
| Post-Kondisi | *Admin* sudah melakukan view hasil pembayaran dan *admin* dapat mengonfirmasi pembayaran yang telah dilakukan oleh *customer* | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Login sebagai *admin* |  |
|  | 2. Jika ada tagihan masuk yang telah dibayarkan oleh *customer*, maka sistem langsung menampilkan halaman yang berisi invoice, jika belum maka lanjut tahap 3 |
| 3. Membuka tab pemasukan pembayaran |  |
|  | 4. Menampilkan halaman yang berisi list tagihan dan nama *customer*nya |
| 5. Memilih salah satu *customer* dan mengecek maupun memberi konfirmasi |  |
|  | 6. Menampilkan tahap 2 |
| 7. Mengklik konfirmasi lunas |  |

##### 3.2.1.14 Usecase Scenario #14

|  |  |  |
| --- | --- | --- |
| Nama Use Case | View Data *Customer* | |
| Deskripsi | Fungsi ini digunakan oleh *Admin* untuk melihat data-data *customer* yang telah terdaftar atau memiliki akun | |
| Pre-Kondisi | *Admin* ingin melihat data akun *customer* dan belum melakukan view data *customer* | |
| Post-Kondisi | *Admin* sudah melakukan view data *customer* dan | |
| Skenario Utama |  | |
|  | Aktor | Sistem |
| 1. Login sebagai *admin* |  |
| 2. Buka tab data *customer* |  |
|  | 3. Menampilkan list semua *customer* yang telah terdaftar atau memiliki akun |
| 4. Mengklik salah satu nama *customer* yang ingin dilihat lebih detail data nya |  |
|  | 5. Menampilkan data *customer* yang dipilih *admin* secara lengkap |

3.2.2 ***Class Diagram***

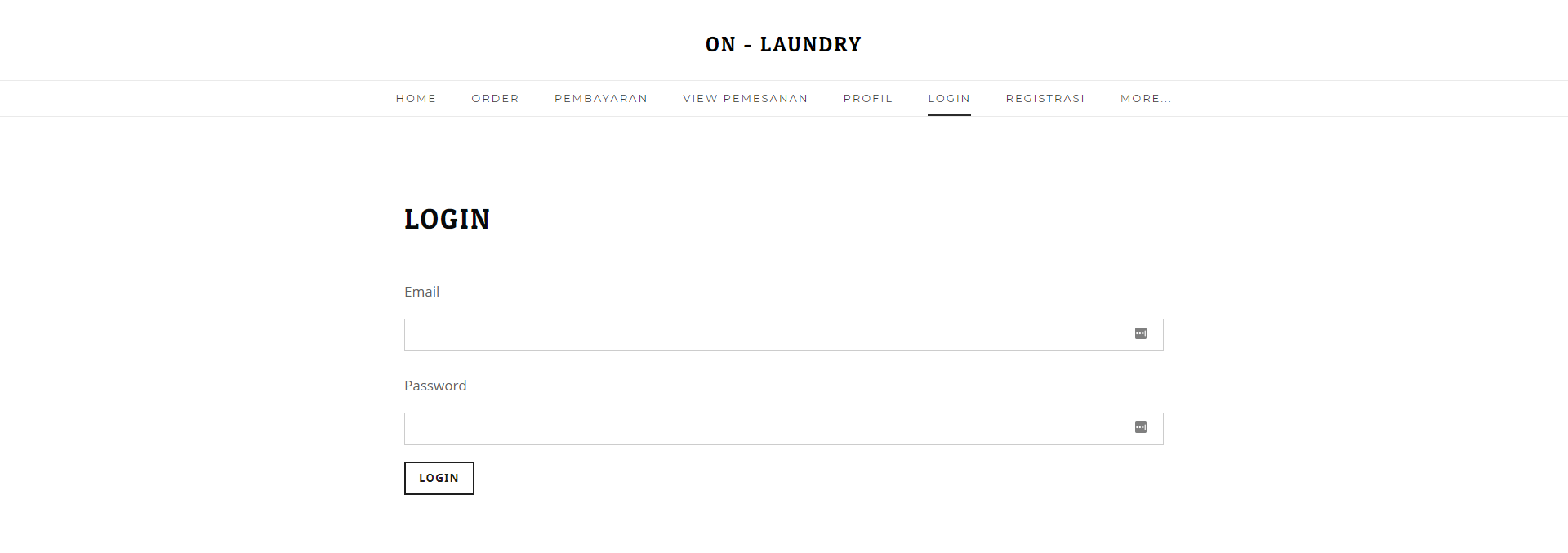
A screenshot of a cell phone

Description generated with high confidence

## 3.3 Antarmuka Pengguna

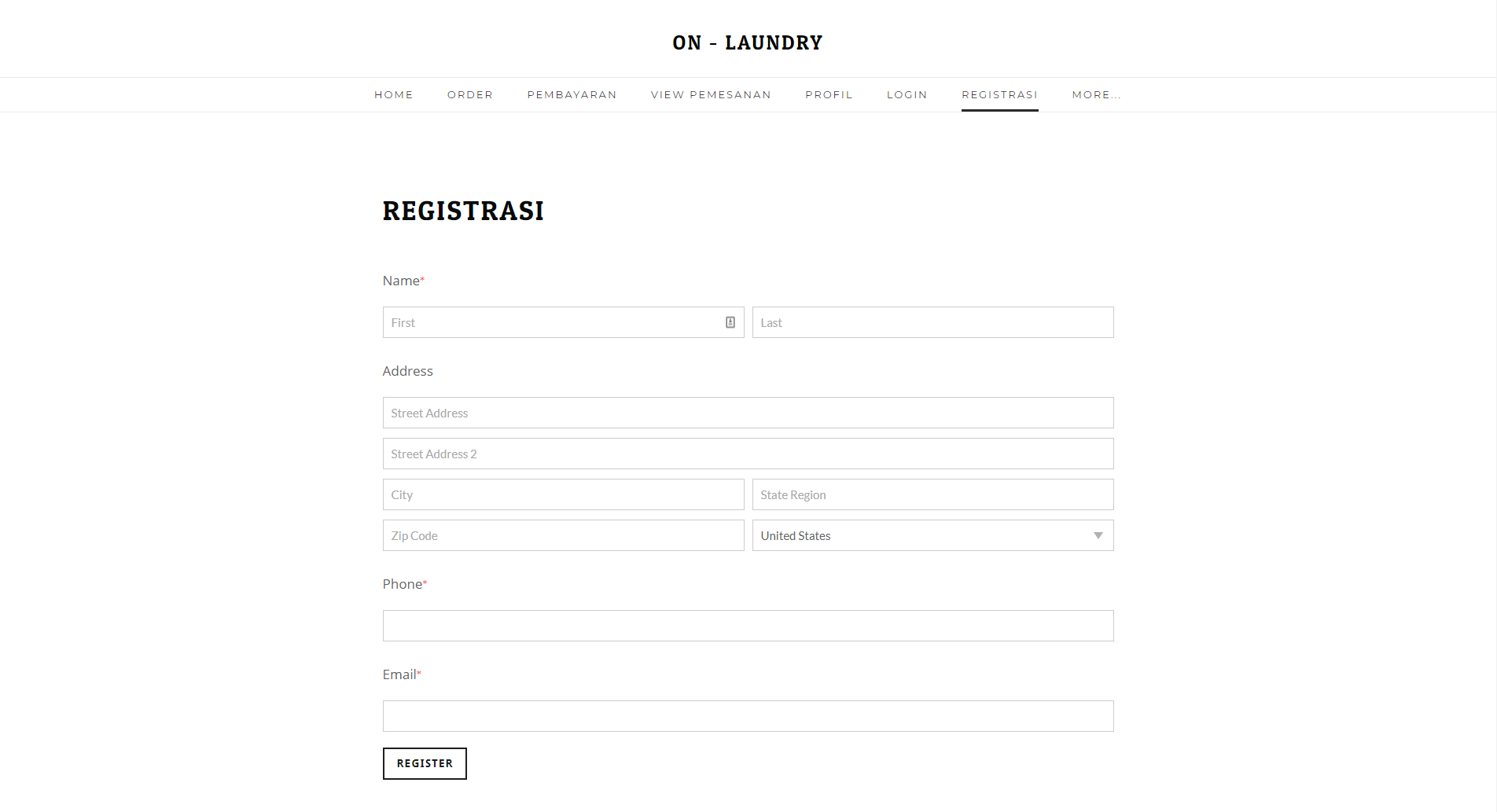
Pengguna perangkat lunak akan berinteraksi melalui antarmuka grafis berbasis web yang ditampilkan pada halaman *web browser*. Sistem menerima masukan dari pengguna melalui perintah yang dilakukan dengan mengklik dan mengetik pada perangkat komputer dan arahan layar pada perangkat mobile. Hasil keluaran dari proses sistem akan ditampilkan pada layar perangkat yang digunakan.

### 3.3.1 LOGIN



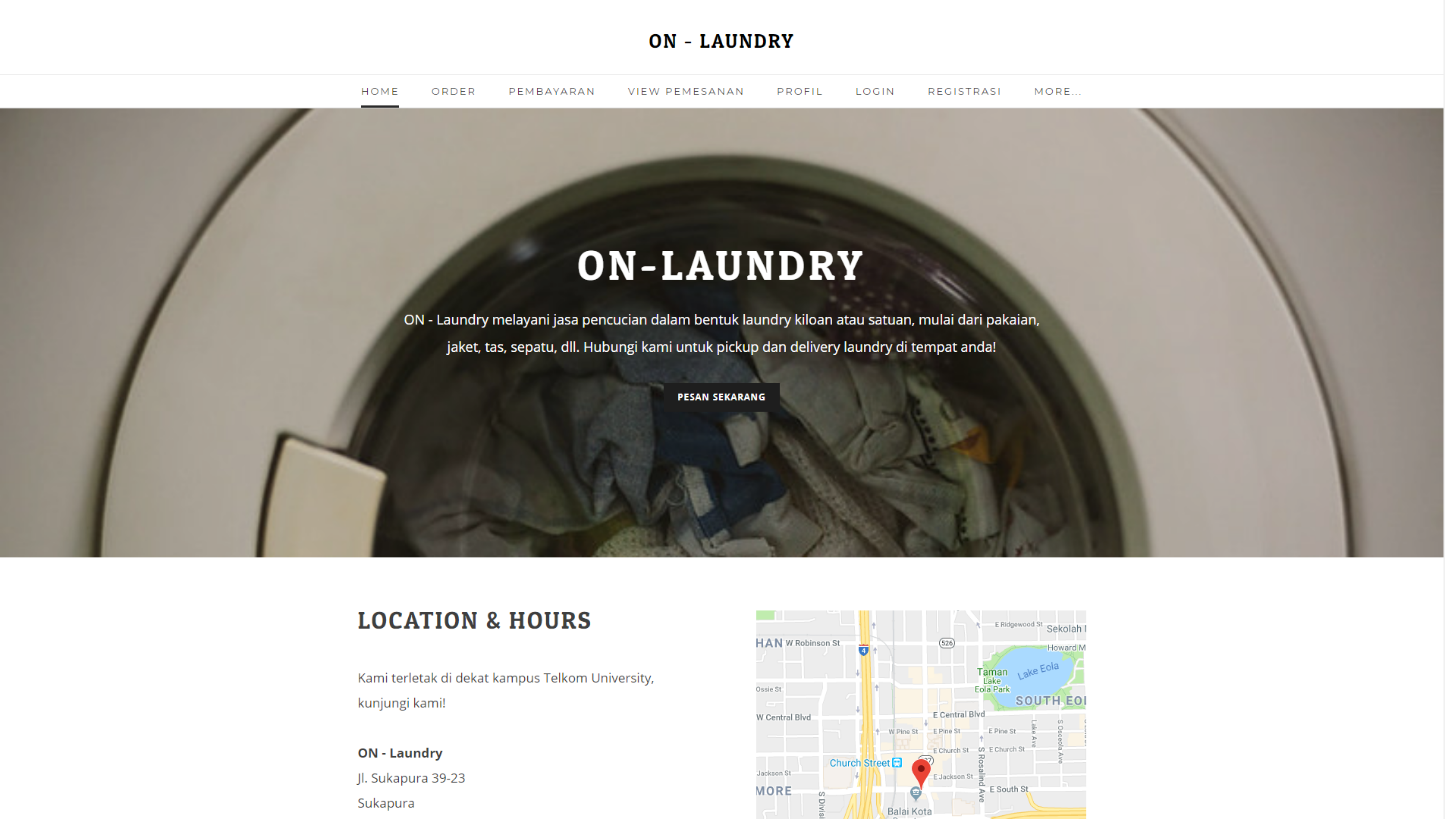
Pengguna dapat masuk ke aplikasi sebagai *user* atau *admin*. Hanya yang sudah terdaftar akun nya yang dapat melakukan login.

### 3.3.2 REGISTRASI



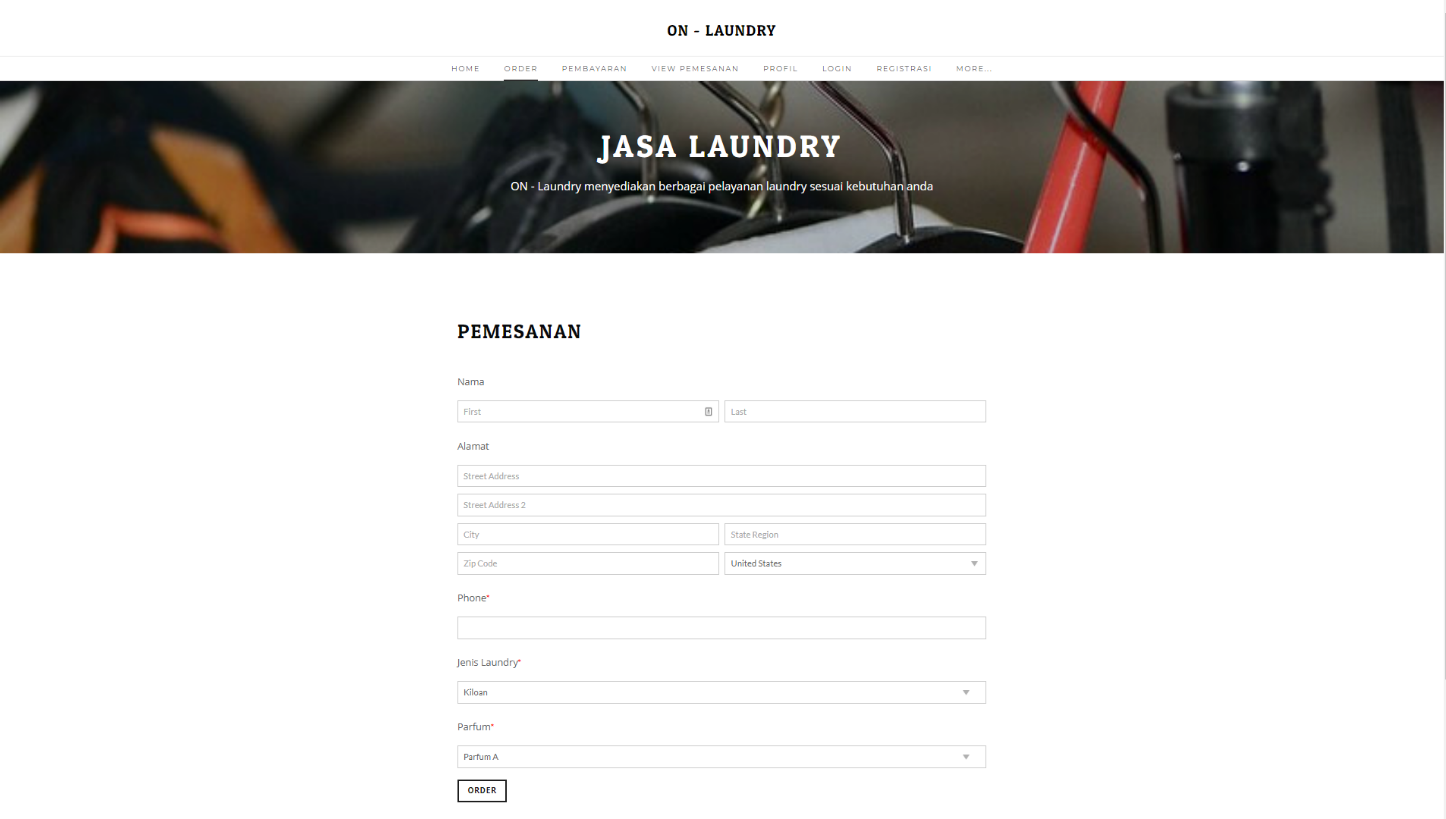
Pengguna dapat melakukan registrasi untuk mendaftarkan diri nya, kemudian mendapatkan akun sehingga dapat melakukan login.

### 3.3.3 HOMEPAGE



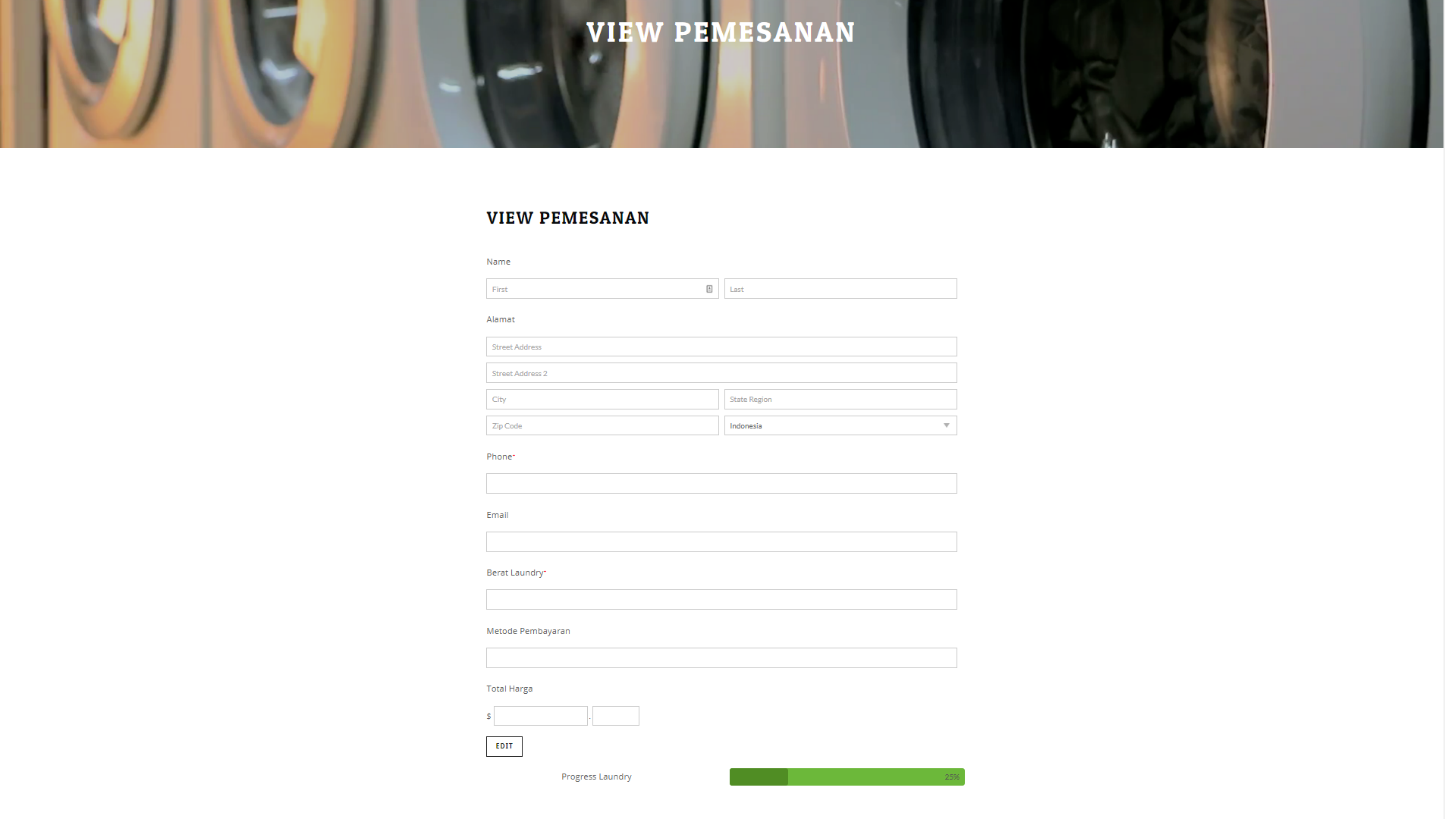
Menu yang ada pada laman *homepage* tidak semua nya dapat diakses, tergantung dari *role* pengguna ketika melakukan login, jika sebagai *user*, maka menu-menu *admin* tidak dapat diakses, begitupun untuk *admin*.

### 3.3.4 ORDER



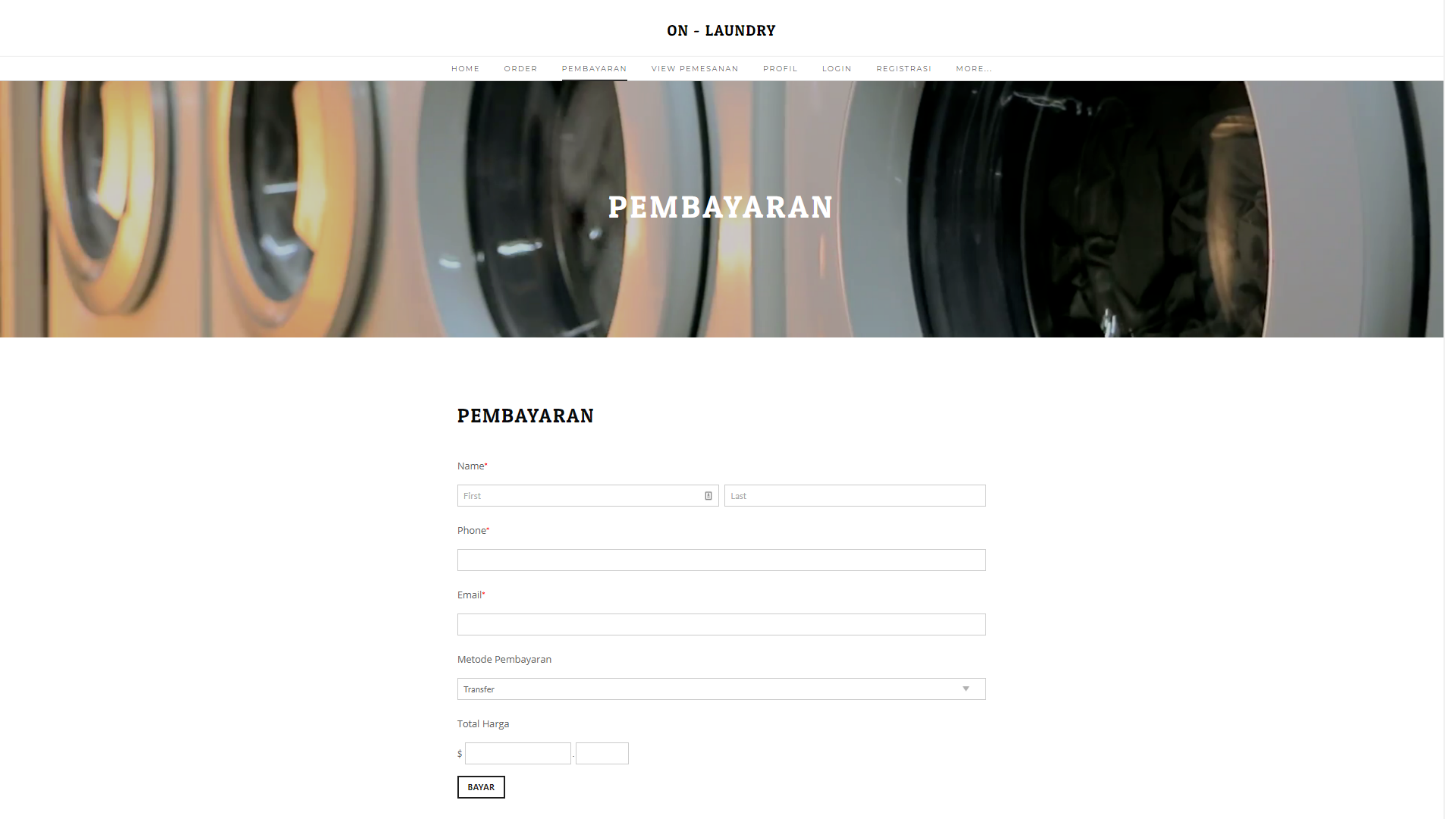
*User* yang ingin melakukan pemesanan layanan laundry dapat mengisi form ini.

### 3.3.5 VIEW PEMESANAN



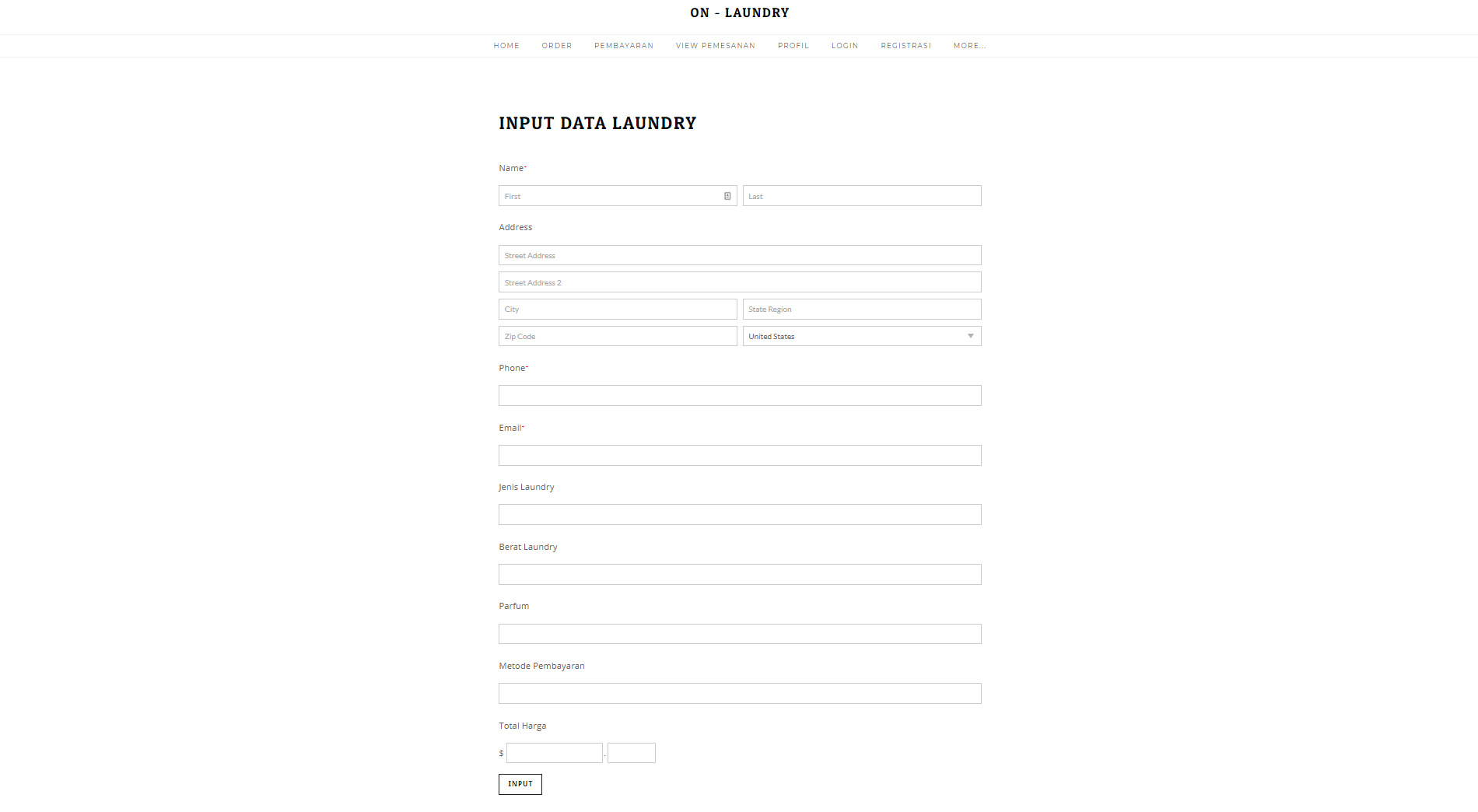
*User* yang ingin melihat proses berjalan pada laundry dapat membuka laman ini. Pada laman ini, *user* juga dapat mengedit data pesanan laundry yang baru saja dilakukan.

### 3.3.6 PEMBAYARAN



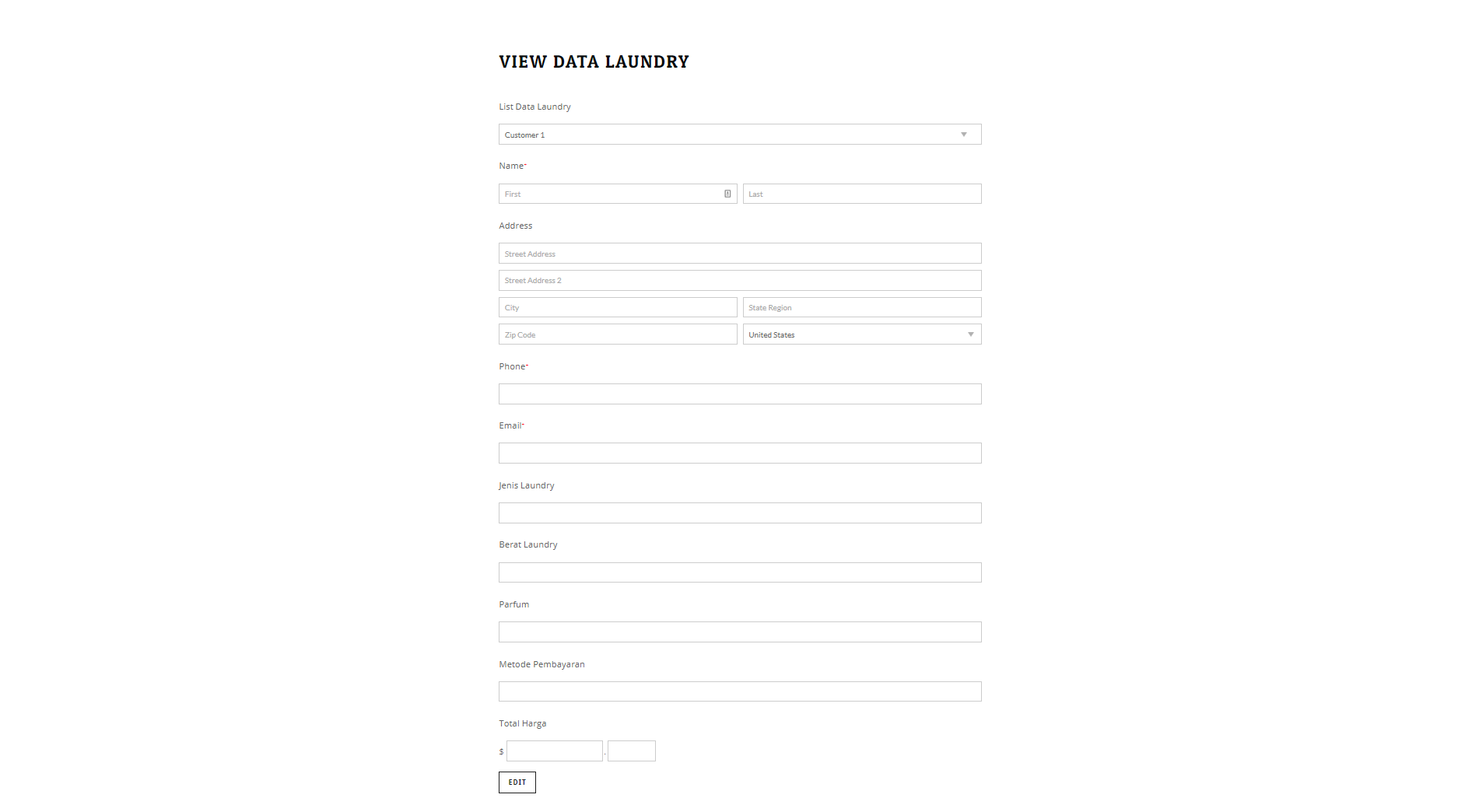
*User* yang sudah melakukan pemesanan, diharuskan nanti nya mengisi form pembayaran ini.

### 3.3.7 INPUT DATA LAUNDRY



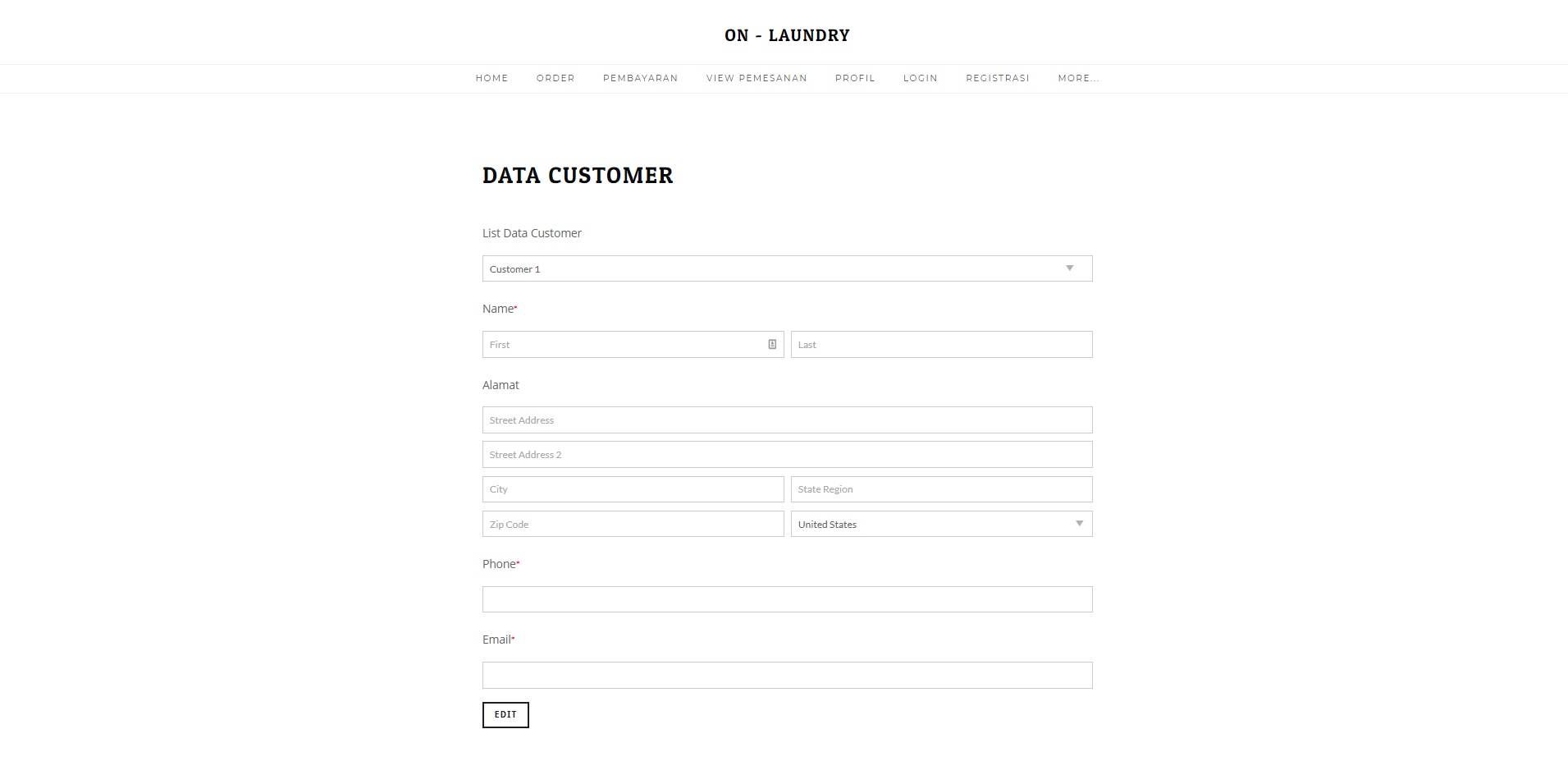
*Admin* mengisi form ini jika ingin memasukan data layanan laundry yang baru.

### 3.3.8 VIEW DATA LAUNDRY



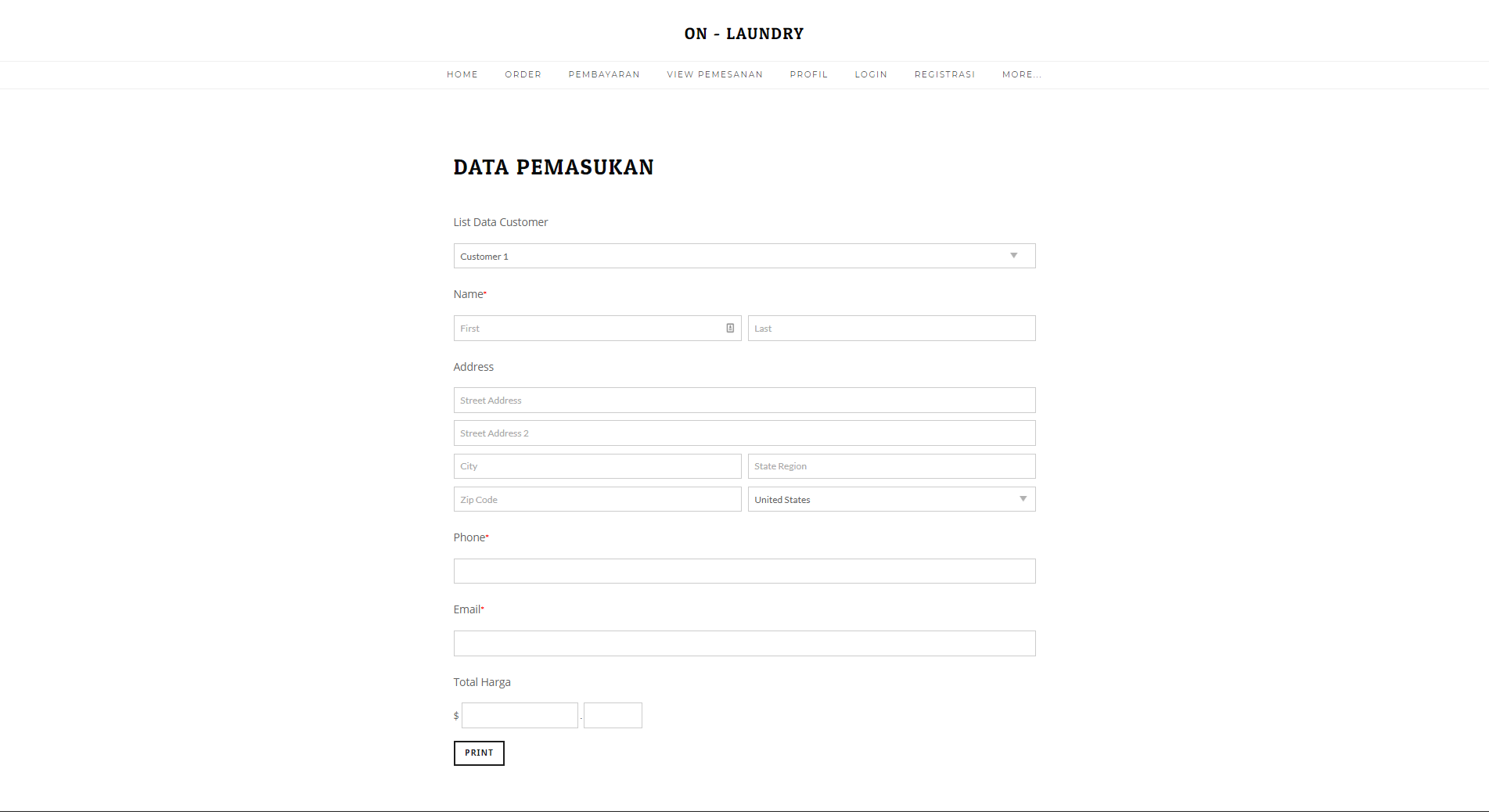
*Admin* dapat melihat dan mengubah data-data laundry yang telah tersimpan.

### 3.3.9 DATA *CUSTOMER*



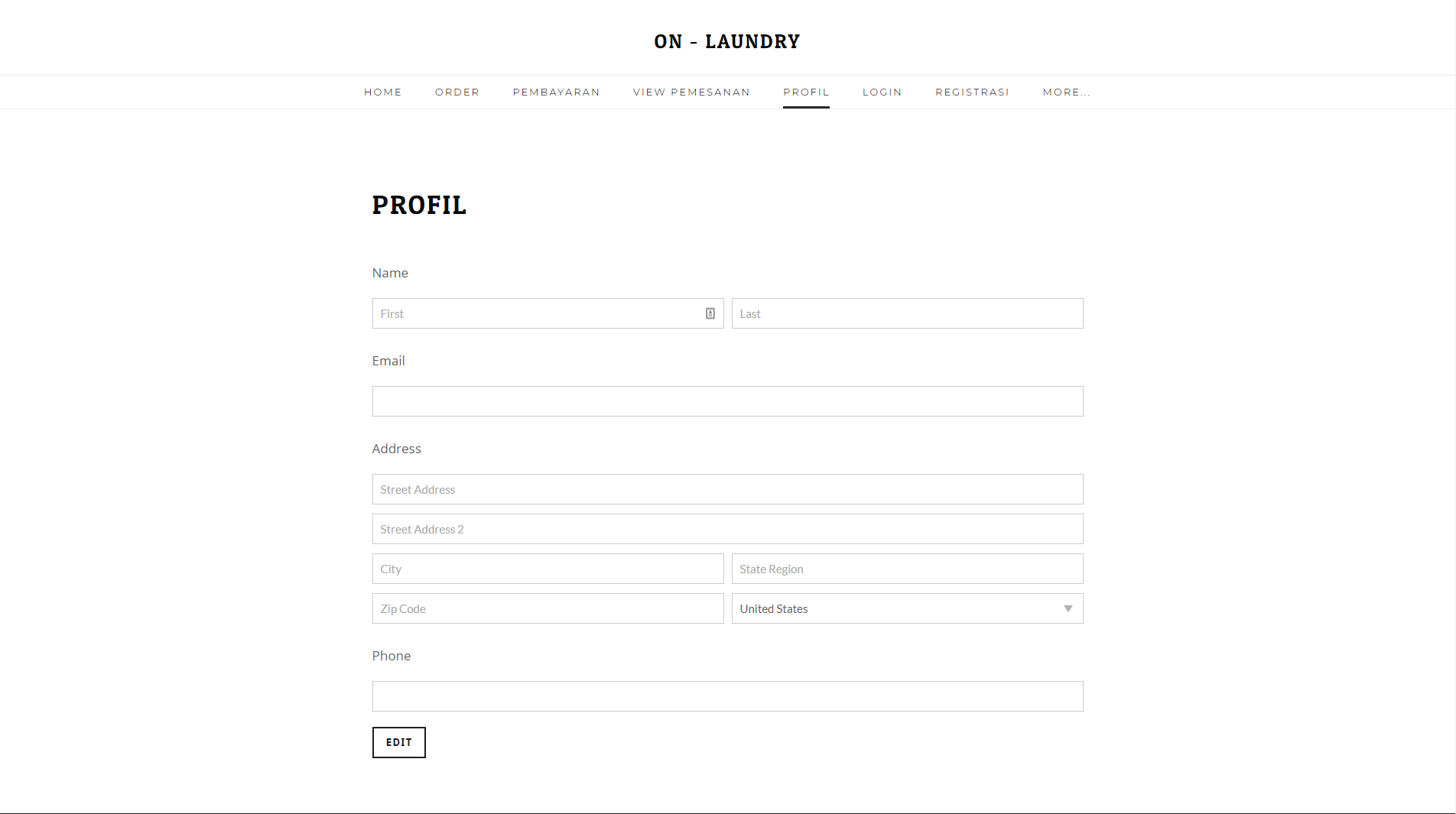
*Admin* dapat melihat dan mengubah data *customer* yang telah terdaftar.

### 3.3.10 DATA PEMASUKAN



*Admin* dapat mencetak laporan pemasukan dari pemesanan *customer*.

### 3.3.11 EDIT PROFILE



Pengguna aplikasi dapat mengubah data diri nya pada laman ini.

## 3.4 Antarmuka Perangkat Keras

Kebutuhan minimum komputer yang dapat digunakan untuk menjalankan perangkat lunak:

*Processor*: 800 MHz / 1 GHz 32-bit;

RAM: 512 MB;

*Graphic card*: DirectX 9;

*Graphic memory*: 64 MB;

*Hard Disk Drive* (HDD): 20 GB;

*Operation System*: Windows XP

## 3.5 Antarmuka Perangkat Lunak

Untuk mengakses aplikasi Laundry Online On-Laundry dapat menggunakan berbagai browser (tidak membutuhkan browser khusus).

Perangkat lunak yang dibutuhkan: Visual Studio Code, XAMPP;

*Database* yang dibutuhkan: phpmyadmin;

*Web hosting*: Hostinger.

## 3.6 Antarmuka Komunikasi

Perangkat lunak ini membutuhkan perangkat komputer server dan client yang terhubung internet dengan TCP/IP sebagai antarmuka komunikasi

**4. Requirements Lain**

# Lampiran A: Daftar Kata-Kata Sukar

*Database*: Kumpulan [informasi](https://id.wikipedia.org/wiki/Informasi) yang disimpan di dalam [komputer](https://id.wikipedia.org/wiki/Komputer) secara sistematik sehingga dapat diperiksa menggunakan suatu [program komputer](https://id.wikipedia.org/wiki/Program_komputer) untuk memperoleh informasi dari basis data tersebut. Disebut juga basis data atau pangkalan data;

*Platform*: Kombinasi antara sebuah arsitektur [perangkat keras](https://id.wikipedia.org/wiki/Perangkat_keras) dengan sebuah [kerangka kerja](https://id.wikipedia.org/wiki/Kerangka_kerja) [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) (termasuk kerangka kerja [aplikasi](https://id.wikipedia.org/wiki/Aplikasi));

*Compatible*: Mampu bergerak dan bekerja dengan kesesuaian;

*PWS*: Sistem perangkat keras dan lunak yang dirancang untuk membuat dan mengelola server web pada komputer desktop;

*PHP*: Bahasa yang dapat ditanamkan atau disisipkan ke dalam [HTML](https://id.wikipedia.org/wiki/HTML);

*MYSQL*: Sebuah [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) sistem manajemen [basis data](https://id.wikipedia.org/wiki/Basis_data) [SQL](https://id.wikipedia.org/wiki/SQL) ([bahasa Inggris](https://id.wikipedia.org/wiki/Bahasa_Inggris): *database management system*) atau DBMS yang [multialur](https://id.wikipedia.org/w/index.php?title=Multialur&action=edit&redlink=1), [multipengguna](https://id.wikipedia.org/wiki/Multipengguna);

*TCP/IP*: Model jaringan komputer dan rangkaian protokol komunikasi yang digunakan di [internet](https://id.wikipedia.org/wiki/Internet) dan jaringan komputer yang mirip;

Visual Studio Code: Editor kode yang didefinisikan dan dioptimalkan untuk membangun dan men-debug aplikasi web dan cloud modern;

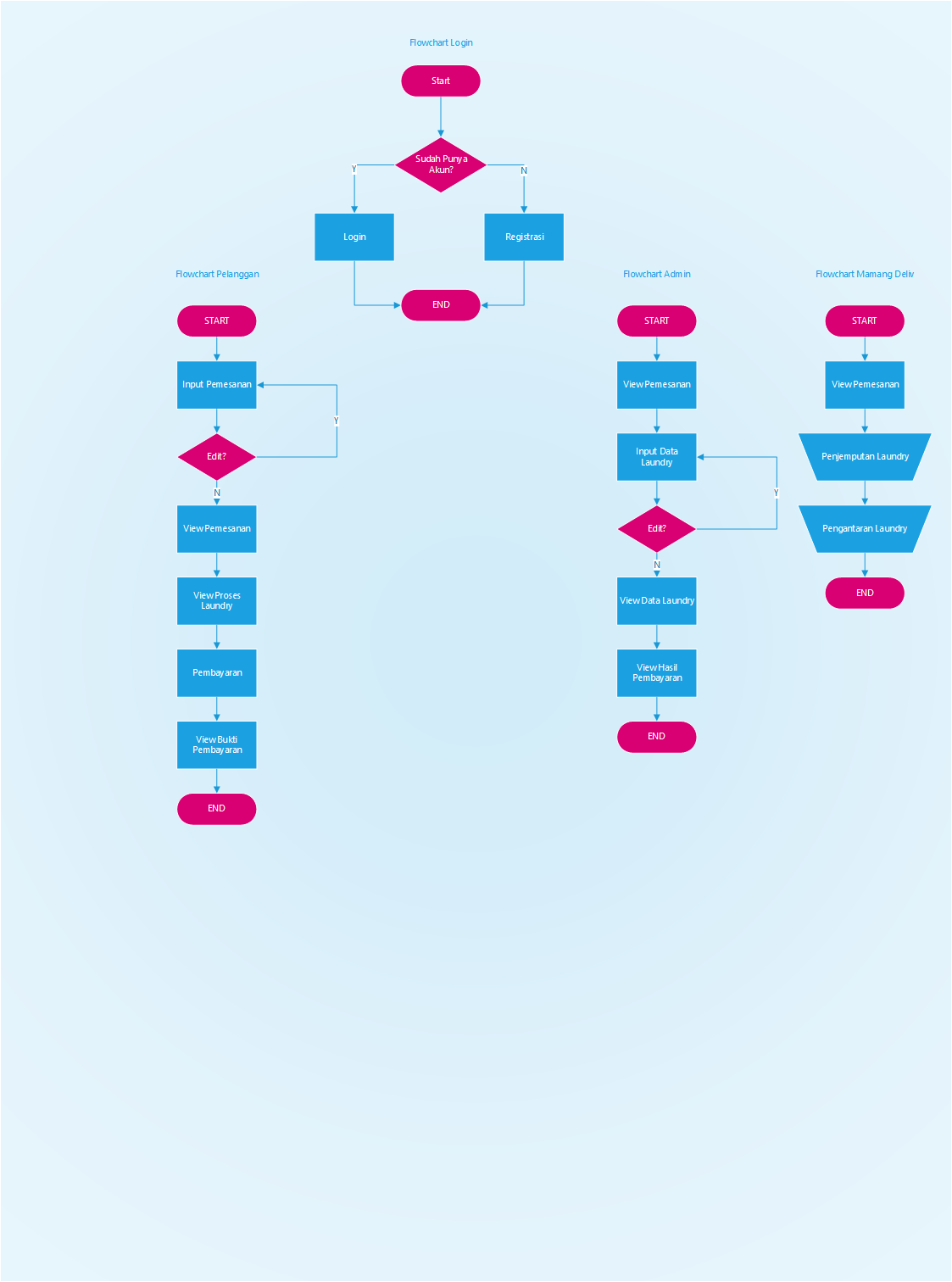
XAMPP: Paket tumpukan solusi server web lintas platform gratis dan sumber terbuka yang dikembangkan oleh Apache Friends, yang sebagian besar terdiri dari Server HTTP Apache, basis data MariaDB, dan juru bahasa untuk skrip yang ditulis dalam bahasa pemrograman PHP dan Perl;

Phpmyadmin: [Perangkat lunak bebas](https://id.wikipedia.org/wiki/Perangkat_lunak_bebas) yang ditulis dalam [bahasa pemrograman](https://id.wikipedia.org/wiki/Bahasa_pemrograman) [PHP](https://id.wikipedia.org/wiki/PHP) yang digunakan untuk menangani administrasi [MySQL](https://id.wikipedia.org/wiki/MySQL) melalui [website Jejaring Jagat Jembar](https://id.wikipedia.org/w/index.php?title=Jejaring_Jagat_Jembar&action=edit&redlink=1) ([*World Wide Web*](https://id.wikipedia.org/wiki/World_Wide_Web));

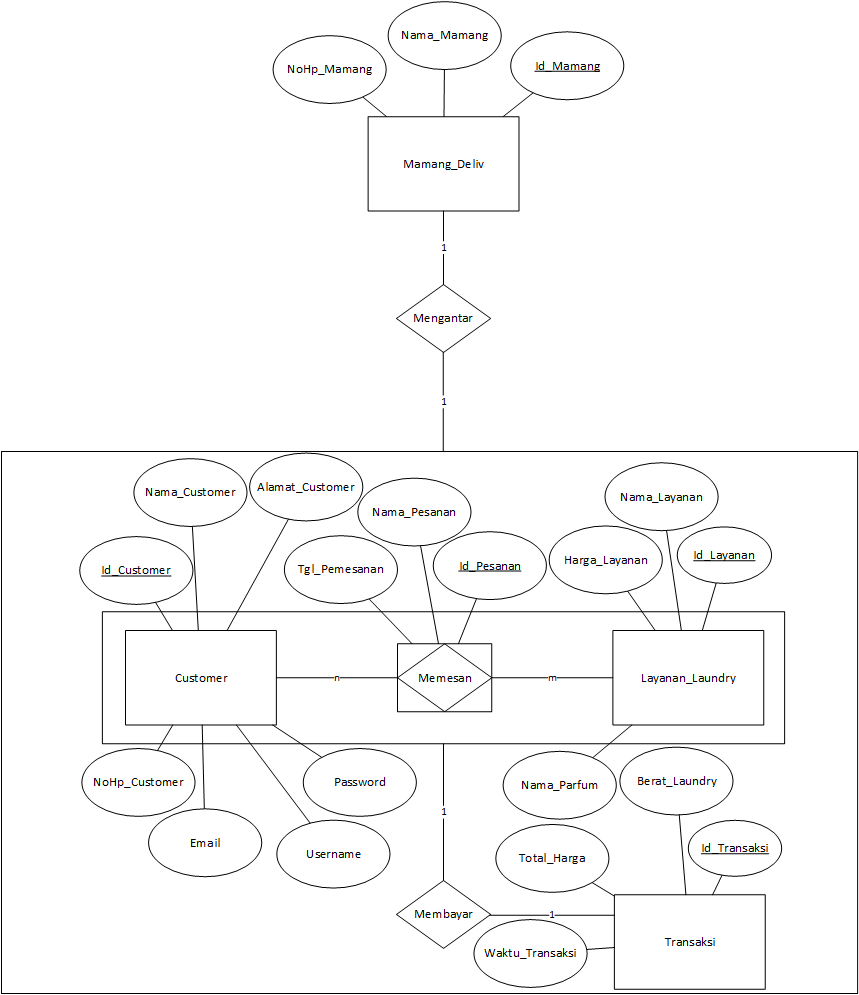
Hostinger: Penyedia hosting web milik karyawan dan pendaftar domain Internet.

# Lampiran B: Analysis Models

## 4.1 Flowchart



## 4.2 ER Diagram



## 4.3 Skema Relasi

