**SKPL-02**

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

SuryaMenerangi.id

untuk:

Muhammad Tamiramin Hayat Suhendar

Dipersiapkan oleh:

Kelompok 02

Gede Prasidha Bhawarnawa 13520004

Bryan Bernigen 13520034

Hansel Valentino Tanoto 13520046

Aditya Prawira Nugroho 13520049

Program Studi Teknik Informatika

STEI - ITB

Jl. Ganesha 10, Bandung 40132

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Program Studi Teknik Informatika / Sistem dan Teknologi Informasi**  **STEI – ITB** | Nomor Dokumen | | Halaman |
| *SKPL-02* | | *1/23* |
| Revisi | *-* | 23 Maret 2022 |

DAFTAR PERUBAHAN

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

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

Daftar Halaman Perubahan

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

Daftar Isi

[1 Pendahuluan 7](#_Toc98943221)

[1.1 Tujuan Penulisan Dokumen 7](#_Toc98943222)

[1.2 Lingkup Masalah 7](#_Toc98943223)

[1.3 Definisi, Istilah dan Singkatan 7](#_Toc98943224)

[1.4 Aturan Penomoran 7](#_Toc98943225)

[1.5 Referensi 7](#_Toc98943226)

[1.6 Deskripsi umum Dokumen (Ikhtisar) 8](#_Toc98943227)

[2 Deskripsi Umum Perangkat Lunak 9](#_Toc98943228)

[2.1 Deskripsi Umum Sistem 9](#_Toc98943229)

[2.2 Karakteristik Pengguna 9](#_Toc98943230)

[2.3 Batasan 9](#_Toc98943231)

[2.4 Lingkungan Operasi 9](#_Toc98943232)

[3 Deskripsi Kebutuhan 11](#_Toc98943233)

[3.1 Kebutuhan Fungsional 11](#_Toc98943234)

[3.2 Kebutuhan Non Fungsional 11](#_Toc98943235)

[3.3 Model Use Case 12](#_Toc98943236)

[3.3.1 Diagram *Use* *Case* 12](#_Toc98943237)

[3.3.2 Definisi Aktor 12](#_Toc98943238)

[3.3.3 Definisi *Use* *Case* 13](#_Toc98943239)

[3.3.4 Skenario *Use* *Case* 13](#_Toc98943240)

[4 Model Analisis 20](#_Toc98943241)

[4.1 Identifikasi Kelas 20](#_Toc98943242)

[4.2 Diagram Kelas 20](#_Toc98943243)

[5 Kerunutan (*Traceability*) 23](#_Toc98943244)

[5.1 Kebutuhan Fungsional vs Use Case 23](#_Toc98943245)

[5.2 Use Case vs Kelas Terkait 23](#_Toc98943246)

**Daftar Tabel**

[Tabel 1.1 Definisi, istilah, dan singkatan 7](#_Toc98943247)

[Tabel 1.2 Aturan penomoran 7](#_Toc98943248)

[Tabel 2.1 Karakteristik pengguna 9](#_Toc98943249)

[Tabel 3.1 Kebutuhan fungsional 11](#_Toc98943250)

[Tabel 3.2 Kebutuhan non fungsional 11](#_Toc98943251)

[Tabel 3.3 Definisi aktor 12](#_Toc98943252)

[Tabel 3.4 Definisi use case 13](#_Toc98943253)

[Tabel 3.5 Skenario use case register 13](#_Toc98943254)

[Tabel 3.6 Skenario use case sign in 14](#_Toc98943255)

[Tabel 3.7 Skenario use case memesan layanan 15](#_Toc98943256)

[Tabel 3.8 Skenario use case validasi pemesanan 15](#_Toc98943257)

[Tabel 3.9 Skenario use case cek status pesanan 16](#_Toc98943258)

[Tabel 3.10 Skenario use case mengontak teknisi 16](#_Toc98943259)

[Tabel 3.11 Skenario use case membaca artikel 16](#_Toc98943260)

[Tabel 3.12 Skenario use case melakukan isi saldo 17](#_Toc98943261)

[Tabel 3.13 Skenario use case balancing data pengguna 17](#_Toc98943262)

[Tabel 3.14 Skenario use case konfirmasi pesanan 17](#_Toc98943263)

[Tabel 3.15 Skenario use case mengontak customer 18](#_Toc98943264)

[Tabel 3.16 Skenario use case melakukan cashout saldo 18](#_Toc98943265)

[Tabel 4.1 Identifikasi kelas 20](#_Toc98943266)

[Tabel 4.2 Diagram kelas 20](#_Toc98943267)

[Tabel 5.1 Kebutuhan fungsional vs use case 23](#_Toc98943268)

[Tabel 5.2 Use case vs kelas terkait 23](#_Toc98943269)

**Daftar Gambar**

[Gambar 3.1 Diagram use case 12](#_Toc98943270)

[Gambar 4.1 Diagram kelas 20](#_Toc98943271)

# Pendahuluan

## Tujuan Penulisan Dokumen

Adapun tujuan dari penulisan dokumen SKPL ini adalah sebagai gambaran kasar mengenai perangkat lunak yang akan dibuat sehingga memudahkan *developer* dalam proses pengembangan perangkat lunak tersebut. Dokumen SKPL ini juga bertujuan untuk memberikan gambaran kepada klien mengenai perangkat lunak dan menjadi panduan bagi *developer* dalam mengembangkan perangkat lunak.

## Lingkup Masalah

Aplikasi ini bernama SuryaMenerangi.id. Dari sisi pengguna, aplikasi ini dibangun dengan tujuan membantu pengguna yang ingin melakukan instalasi pembangkit listrik tenaga listrik (PLTS). Aplikasi juga dapat berkomunikasi langsung dengan teknisi serta mencari lebih banyak informasi mengenai PLTS. Dari sisi teknisi PLTS, aplikasi menjadi penghubung antara teknisi dengan pengguna. Teknisi juga dapat menerima atau menolak pesanan yang masuk, serta dapat menerima pembayaran atas pekerjaan yang telah dilakukan.

## Definisi, Istilah dan Singkatan

Tabel 1.1 Definisi, istilah, dan singkatan

|  |  |  |
| --- | --- | --- |
| **ID** | **Definisi / Istilah / Singkatan** | **Deskripsi** |
| DIS-01 | SuryaMenerangi.id | Nama aplikasi pemesanan layanan pemasangan, perbaikan, dan pencabutan panel surya yang akan diembangkan |
| DIS-02 | P/L | Singkatan dari perangkat lunak / *software* |
| DIS-03 | Use Case | Sebuah skenario yang mungkin terjadi ketika P/L digunakan oleh para aktor |
| DIS-04 | Aktor | Generalisasi untuk tipe-tipe pengguna yang akan menggunakan P/L |
| DIS-05 | Diagram Kelas | Diagram yang menggambarkan hubungan antar kelas pada P/L beserta atribut dan *method* yang terkandung di setiap kelas |

## Aturan Penomoran

Tabel 1.2 Aturan penomoran

|  |  |  |
| --- | --- | --- |
| **No.** | **Cara penomoran** | **Definisi** |
| 1 | DIS-XX | XX adalah nomor urutan dari definisi, istilah, dan singkatan |
| 2 | FXX | XX adalah nomor urutan dari kebutuhan fungsional |
| 3 | NFXX | XX adalah nomor urutan dari kebutuhan non fungsional |
| 4 | UCXX | XX adalah nomor urutan dari *use* *case* |
| 5 | IKXX | XX adalah nomor urutan dari identifikasi kelas |

## Referensi

Mulyanto, A. Widyani, Y. Rusmawati, Y. (2022). *Penulisan SKPL*. Institut Teknologi Bandung, IF2250

Booch, Grady. (2006). *Object Oriented Analysis & Design with Application 3rd Edition*. Pearson Education India.

Lethbridge, Timothy Christian, and Robert Laganiere. (2005). *Object-Oriented Software Engineering*. New York: McGraw-Hill.

Mulyanto, A. Widyani, Y. Rusmawati, Y. (2022). *Pemodelan dengan Kelas*. Institut Teknologi Bandung, IF2250

Rosa, A. S. Shalahuddin, M. (2018). *Rekayasa Perangkat Lunak*. Penerbit Informatika.

## Deskripsi umum Dokumen (Ikhtisar)

Dokumen SKPL ini terdiri dari 5 bab. Bab 1 Pendahuluan membahas mengenai tujuan penulisan, lingkup masalah yang dibahas, definisi, istilah, dan singkatan yang digunakan, aturan penomoran, referensi yang dipakai, serta deskripsi/ikhtisar dokumen SKPL. Bab 2 Deskripsi Umum Perangkat Lunak membahas mengenai deskripsi sistem secara umum, karakteristik pengguna, batasan perangkat lunak, dan lingkungan pengoperasian perangkat lunak. Bab 3 Kebutuhan Perangkat Lunak mengenai kebutuhan fungsional, kebutuhan nonfungsional, dan model *use* *case*. Pada subbab model *use case* ini dipaparkan diagram *use case*, definisi aktor, definisi *use case*, dan skenario *use case*. Bab 4 Model Analisis membahas mengenai pengidentifikasian kelas dan diagram kelas. Bab 5 Kerunutan (*traceability*) membahas mengenai hubungan suatu kebutuhan fungsional dengan *use case*. Bab ini juga membahas bagaimana *use case* di translasikan menjadi sebuah kelas.

# Deskripsi Umum Perangkat Lunak

## Deskripsi Umum Sistem

Sistem yang ingin didesain adalah sistem perangkat lunak pemesanan pembangkit listrik tenaga surya (PLTS). Sistem dirancang dengan tujuan menyambungkan pelanggan sebagai *customer* dengan teknisi. Sistem ini didesain agar *customer* dapat melakukan *login* dan *register* di aplikasi, lalu mengajukan pesanan instalasi paket pembangkit listrik untuk dipasang di kediaman masing-masing. Bila ada satu atau lain hal, *customer* dapat mengajukan pemesanan paket reparasi atau pencopotan pembangkit listrik yang sudah ada. Setelah pesanan terkonfirmasi, *customer* dapat melakukan pembayaran internal di dalam aplikasi dan dapat juga berkomunikasi secara langsung dengan teknisi. Selain itu, *customer* juga dapat melihat informasi mengenai PLTS, seperti cara kerja dan kelebihannya.

Di lain sisi, teknisi juga dapat melakukan *Login* dan *Register* di aplikasi, lalu melakukan tinjauan terhadap semua pesanan yang telah dikonfirmasi oleh *customer* di aplikasi. Teknisi dapat menerima maupun menolak pesanan yang telah masuk. Bila pesanan diterima, maka teknisi dapat menerima pembayaran setelah selesai mengerjakan pesanan dan dapat berkomunikasi di aplikasi secara langsung dengan klien/*customer*.

## Karakteristik Pengguna

Tabel 2.1 Karakteristik pengguna

| **Kategori Pengguna** | **Tugas** | **Hak Akses ke Aplikasi** |
| --- | --- | --- |
| *Customer* | Membuat akun (*register*), masuk (*login*) ke dalam akun, memesan layanan, melakukan percakapan (*chat*) dengan teknisi, melihat informasi PLTS, melakukan *top-up* saldo. | *Customer* dapat melakukan *login* dan *register* di aplikasi, melakukan pemesanan untuk pemasangan dan perbaikan kepada teknisi, bertanya / berkomunikasi secara langsung kepada teknisi yang dipilih, dapat memesan pencabutan layanan, melihat informasi keuntungan dalam memakai PLTS, serta melakukan *top-up* saldo untuk membayar layanan. *Customer* juga dapat melihat paket-paket layanan / pemesanan yang ditawarkan di dalam aplikasi. |
| Teknisi | Membuat akun (*register*), masuk (*login*) ke dalam akun, mengonfirmasi layanan, melakukan percakapan (*chat*) dengan pengguna, melakukan *cashout* saldo. | Teknisi dapat melakukan *login* dan *register* di aplikasi, melihat pemesanan yang telah dikonfirmasi oleh *customer*, menerima atau menolak pemesanan, berkomunikasi secara langsung dengan *customer*, serta melakukan *cashout* saldo. |

## Batasan

Adapun batasan-batasan dari perangkat lunak ini adalah perangkat lunak harus dapat dijalankan di *platform* *desktop* dengan *operating* *system* Windows 10. Antarmuka perangkat lunak harus ditampilkan dalam bahasa Indonesia. Penyimpanan data dilakukan dengan menggunakan MySQL.

## Lingkungan Operasi

Perangkat lunak / *software* pemesanan PLTS yang akan dikembangkan memiliki spesifikasi / lingkungan implementasi sebagai berikut.

*Client* : Perangkat *customer* dan teknisi berupa komputer *desktop*

*Operating* *System* : Windows 10

DBMS : MySQL / Oracle / PostgreSQL

*Development* *Tools* : Visual Studio Code

*Filing* *System* : *Alphanumeric*

Bahasa Pemrograman : Python, .Net

# Deskripsi Kebutuhan

## Kebutuhan Fungsional

Tabel 3.1 Kebutuhan fungsional

| **ID** | **Kebutuhan** | **Penjelasan** |
| --- | --- | --- |
| F01 | P/L dapat menerima masukan berupa pemesanan layanan dari *customer* | P/L dapat menerima masukan dari *customer* berupa berupa jenis layanan (perbaikan/pemasangan/pencabutan), jenis paket PLTS, alamat pemasangan, waktu pemasangan, dan konfirmasi pesanan. |
| F02 | P/L dapat menerima masukan berupa konfirmasi dari teknisi | P/L dapat menerima masukan dari teknisi berupa konfirmasi pesanan (menerima/menolak layanan). |
| F03 | P/L memiliki fitur *register* | P/L dapat mendaftarkan akun baru dan akan menyimpan informasi akun tersebut ke dalam penyimpanan. Informasi akun yang dapat dimasukkan ke dalam P/L berupa *email*, *password*, nama, nomor telepon, alamat). |
| F04 | P/L memiliki fitur *login* | P/L dapat memberikan akses terhadap *customer* maupun teknisi yang melakukan *login* ke dalam sistem dengan data (*email* dan *password*) yang tepat. |
| F05 | P/L dapat menangani pembayaran layanan | P/L dapat menjadi perantara transaksi keuangan atas layanan yang dipesan antara *customer* dan teknisi. |
| F06 | P/L memiliki fitur *top-up* | P/L dapat menerima masukan nominal uang yang ingin ditambahkan ke saldo *customer*. |
| F07 | P/L memiliki fitur *cashout* / penarikan uang | P/L juga dapat menerima permintaan penarikan dana dari akun teknisi. |
| F08 | P/L mengakomodasi *chat* antara *customer* dengan teknisi | P/L dapat mengakomodasi percakapan secara daring antara *customer* dan teknisi mengenai detail layanan. |
| F09 | P/L dapat menampilkan pilihan layanan yang tersedia | P/L dapat menampilkan pilihan layanan dan paket yang bisa dipilih oleh *customer* beserta daftar harganya. |
| F10 | P/L dapat menampilkan informasi mengenai PLTS | P/L dapat menampilkan artikel/informasi terkait PLTS, seperti informasi umum dan kelebihan pemakaiannya. |

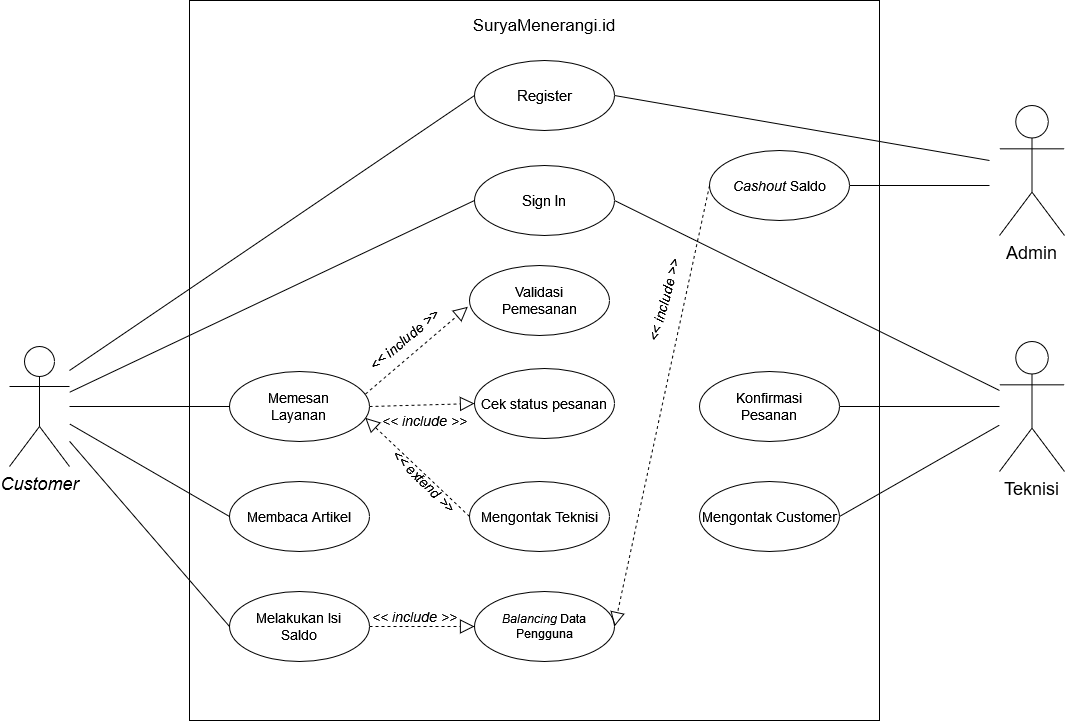
## Kebutuhan Non Fungsional

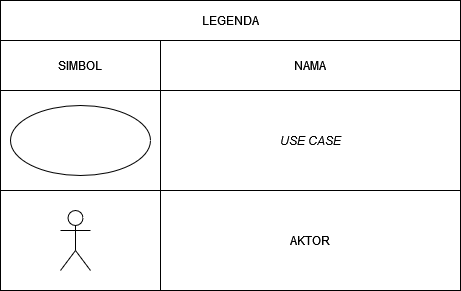
Tabel 3.2 Kebutuhan non fungsional

| **ID** | **Parameter** | **Kebutuhan** |
| --- | --- | --- |
| NF01 | *Availability* | P/L tersedia di *desktop* dan beroperasi 7 hari perminggu, 24 jam per hari dengan waktu *down* maksimal 2 jam tiap 2 minggu. |
| NF02 | *Portability* | P/L menggunakan Windows-OS. |
| NF03 | *Memory* | P/L melakukan penyimpanan dalam bentuk *file* csv atau sql. |
| NF04 | *Security* | P/L melakukan *hashing* pada *password* pengguna. |
| NF05 | Bahasa komunikasi | P/L menggunakan bahasa Indonesia pada antarmukanya. |

## Model Use Case

### Diagram *Use* *Case*





Gambar 3.1 Diagram use case

### Definisi Aktor

Tabel 3.3 Definisi aktor

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| DA01 | *Customer* | Aktor dengan *role* ini berhak untuk melakukan *login* atau *register* serta memesan layanan, yang kemudian bisa di-*extend* melakukan percakapan dengan teknisi. Selain itu, aktor ini bisa membaca artikel dan melakukan isi saldo. |
| DA02 | Teknisi | Aktor dengan *role* ini berhak untuk melakukan *login,* mengonfirmasi pesanan, dan melakukan percakapan dengan *customer*. |
| DA03 | Admin | Aktor dengan *role* ini berhak untuk mendaftarkan teknisi ke dalam sistem dan melakukan penarikan saldo untuk diserahkan ke teknisi. |

### Definisi *Use* *Case*

Tabel 3.4 Definisi use case

|  |  |  |
| --- | --- | --- |
| **No** | ***Use Case*** | **Deskripsi** |
| UC01 | *Register* | *Customer* mendaftarkan akun untuk masuk ke dalam P/L. Admin mendaftarkan teknisi ke dalam sistem sehingga dapat dipilih oleh *customer* |
| UC02 | *Sign In* | *Customer* atau teknisi masuk ke dalam P/L. |
| UC03 | Memesan Layanan | *Customer* melakukan pemesanan layanan, bisa berupa pemasangan, pencabutan, ataupun perbaikan. |
| UC04 | Validasi Pemesanan | Sistem melakukan validasi pesanan untuk mengecek apakah pesanan *customer* valid. |
| UC05 | Cek Status Pesanan | *Customer* melakukan konfirmasi pesanan selesai, kemudian sistem mengecek apakah pesanan sudah dikonfirmasi selesai oleh *customer*. |
| UC06 | Mengontak Teknisi | *Customer* melakukan percakapan dengan teknisi. |
| UC07 | Membaca Artikel | Sistem menampilkan artikel yang dapat dibaca oleh *customer.* |
| UC08 | Melakukan Isi Saldo | *Customer* melakukan isi saldo ke dalam P/L. |
| UC09 | *Balancing* Data Pengguna | Sistem memastikan penambahan atau pengurangan saldo dari *customer* atau teknisi berjalan dengan lancar dan jumlah yang ditambah ataupun dikurangi sesuai dengan yang di-*request* oleh *customer* ataupun admin. |
| UC10 | Konfirmasi pesanan | Teknisi melakukan konfirmasi atau menolak pesanan *customer* yang memilih teknisi tersebut untuk mengerjakan pesanannya. |
| UC11 | Mengontak *Customer* | Teknisi melakukan percakapan dengan *customer* yang sedang ia layani |
| UC12 | *Cashout* saldo | Admin mengurangi saldo teknisi sebanyak jumlah yang di-*request* oleh teknisi. Teknisi hanya dapat melakukan *request* penarikan saldo di kantor suryaMenerangi.id terdekat. |

### Skenario *Use* *Case*

Nama *Use* *Case*: *Register*

Skenario:

Tabel 3.5 Skenario use case register

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal 1** |  |
| 1. *Customer* memilih tombol Daftar (*Register*) |  |
|  | 2. Sistem menampilkan *form* *input* nama lengkap, nomor telepon, tanggal lahir, *email*, *password*, dan alamat |
| 3. *Customer* mengetikkan nama lengkap, nomor telepon, tanggal lahir, *email*, *password*, dan alamat (opsional) |  |
|  | 4. Sistem menerima dan memverifikasi *input* dari *customer* |
|  | 5. Sistem menampilkan laman Masuk (*Sign In*) |
| **Skenario Normal 2** | |
| 1. Admin menambahkan data diri teknisi (nama lengkap, nomor telepon, tanggal lahir, *email*, dan *password*) ke *database* |  |
|  | 2. Sistem meng-*update* *database* data teknisi |
| **Skenario Alternatif 1** | |
| 1. *Customer* memilih tombol Daftar (*Register*) |  |
|  | 2. Sistem menampilkan *form* *input* nama lengkap, nomor telepon, tanggal lahir, *email*, *password*, dan alamat |
| 3. *Customer* memasukkan data sesuai dengan yang diminta pada laman pendaftaran |  |
|  | 4. Sistem memverifikasi *input* customer dan menemukan data *email* atau nomor telepon yang di-*input* *customer* sudah terdaftar di *database* |
|  | 5. Sistem menampilkan pesan kesalahan “*Email* atau nomor telepon anda sudah terdaftar” dan sistem akan menampilkan kembali laman Daftar (*Register*) |
| **Skenario Alternatif 2** | |
| 1. Admin menambahkan data diri teknisi (nama lengkap, nomor telepon, tanggal lahir, *email*, dan *password*) ke *database* |  |
|  | 2. Sistem menampilkan pesan kesalahan “Data teknisi sudah terdaftar di *database*” |

Nama *Use* *Case*: *Sign In*

Skenario:

Tabel 3.6 Skenario use case sign in

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal 1** |  |
| 1. *Customer* memilih tombol Masuk (*Sign* *In*) |  |
|  | 2. Sistem menampilkan *form* *input* *email* dan *password* |
| 3. *Customer* memasukkan *email* dan *password* |  |
|  | 4. Sistem mencocokkan *email* dan *password* yang dimasukkan oleh *customer* dengan *database* *customer*. |
|  | 5. Sistem menemukan *email* dan *password* yang cocok pada *database customer* sehingga sistem memberikan *customer* tersebut akses laman utama |
| **Skenario Normal 2** | |
| 1. Teknisi memilih tombol Masuk (*Sign* *In*) |  |
|  | 2. Sistem menampilkan *form* *input* *email* dan *password* |
| 3. Teknisi memasukkan *email* dan *password* |  |
|  | 4. Sistem mencocokkan *email* dan *password* yang dimasukkan oleh pengguna dengan *database* teknisi. |
|  | 5. Sistem menemukan *email* dan *password* yang cocok pada *database* teknisi sehingga sistem memberikan teknisitersebut akses laman utama |
| **Skenario Alternatif 1** | |
| 1. *Customer* memilih tombol Masuk (*Sign* *In*) |  |
|  | 2. Sistem menampilkan *form* *input* *email* dan *password* |
| 3. *Customer* memasukkan *email* dan *password* |  |
|  | 4. Sistem mencocokkan *email* dan *password* yang dimasukkan oleh *customer* dengan *database* *customer*. |
|  | 5. Sistem tidak menemukan *email* dan *password* yang cocok pada *database* *customer* sehingga sistem tidak memberikan *customer* akses ke laman utama |
| **Skenario Alternatif 2** | |
| 1. Teknisi memilih tombol Masuk (*Sign* *In*) |  |
|  | 2. Sistem menampilkan *form* *input* *email* dan *password* |
| 3. Teknisi memasukkan *email* dan *password* |  |
|  | 4. Sistem mencocokkan *email* dan *password* yang dimasukkan oleh teknisi dengan *database* teknisi. |
|  | 5. Sistem tidak menemukan *email* dan *password* yang cocok pada *database* teknisi sehingga sistem tidak memberikan teknisi akses ke laman utama |

Nama *Use* *Case*: Memesan Layanan

Skenario:

Tabel 3.7 Skenario use case memesan layanan

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
| 1. *Customer* menekan tombol Pesan Layanan |  |
| 2. *Customer* memasukkan data pesanan yang dipilih |  |
|  | 3. Sistem melakukan verifikasi pemesanan dengan mengecek saldo *customer*. |
|  | 4. Sistem meneruskan pesanan ke teknisi. |
| 5. *Customer* melakukan konfirmasi pesanan selesai. |  |
| **Skenario Alternatif** | |
| 1. |  |
|  | 2. |

Nama *Use* *Case*: Validasi Pemesanan

Skenario:

Tabel 3.8 Skenario use case validasi pemesanan

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
| 1. *Customer* memilih tombol Konfirmasi pesanan pada laman pemesanan |  |
|  | 2. Sistem melakukan validasi terhadap saldo *customer* saat ini |
|  | 3. Sistem mendapati bahwa saldo *customer* cukup untuk layanan ini sehingga saldo *customer* dikurangi sementara dan pesanan *customer* diteruskan ke teknisi. |
| **Skenario Alternatif** | |
| 1. *Customer* memilih tombol Konfirmasi pesanan pada laman pemesanan |  |
|  | 2. Sistem melakukan validasi terhadap saldo *customer* saat ini |
|  | 3. Sistem mendapati bahwa saldo *customer* tidak cukup untuk layanan ini sehingga sistem meminta *customer* untuk melakukan pengisian saldo. |

Nama *Use* *Case*: Cek Status Pesanan

Skenario:

Tabel 3.9 Skenario use case cek status pesanan

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
| 1. *Customer* memilih tombol Cek Pesanan |  |
|  | 2. *Customer* memiliki pesanan aktif (status pesanan bukan selesai) sehingga sistem menampilkan status pesanan tersebut (menunggu konfirmasi teknisi/ dikonfirmasi teknisi/ ditolak teknisi/ menunggu konfirmasi selesai) |
| **Skenario Alternatif** | |
| 1. *Customer* memilih tombol Cek Pesanan |  |
|  | 2. *Customer* tidak memiliki pesanan aktif sehingga sistem hanya memberi tahu *customer* bahwa tidak ada pesanan yang sedang berlangsung |

Nama *Use* *Case*: Mengontak Teknisi

Skenario:

Tabel 3.10 Skenario use case mengontak teknisi

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
| 1. *Customer* memilih tombol *Chat* dengan teknisi |  |
|  | 2. Sistem mengecek apakah ada pesanan yang berstatus aktif oleh *customer* tersebut |
|  | 3. Sistem menemukan pesanan aktif oleh *customer* tersebut sehingga sistem menampilkan laman *chatting* dengan teknisi |
| 4. *Customer* melakukan *chatting* dengan teknisi yang bersangkutan |  |
|  | 5. Pesan dari *customer* diteruskan ke teknisi dan pesan dari teksnisi ditampilkan ke *customer*. |
| **Skenario Alternatif** | |
| 1. *Customer* memilih tombol *Chat* dengan teknisi |  |
|  | 2. Sistem mengecek apakah ada pesanan yang berstatus aktif oleh *customer* tersebut |
|  | 3. Sistem tidak menemukan pesanan aktif oleh *customer* tersebut sehingga sistem tidak menampilkan laman *chatting* dengan teknisi. |

Nama *Use* *Case*: Membaca Artikel

Skenario:

Tabel 3.11 Skenario use case membaca artikel

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
| 1. *Customer* memilih menu Baca Artikel |  |
|  | 2. Sistem menampilkan daftar artikel yang tersedia |
| 3. *Customer* memilih salah satu artikel yang akan dibaca |  |
|  | 4. Sistem menampilkan artikel yang dipilih *customer* |
| 5. *Customer* menekan tombol Kembali ke menu membaca artikel |  |
|  | 6. Sistem menampilkan kembali daftar artikel yang tersedia |
| 7. *Customer* menekan tombol Kembali ke menu utama |  |
|  | 8. Sistem menampilkan menu utama kembali |

Nama *Use* *Case*: Melakukan Isi Saldo

Skenario:

Tabel 3.12 Skenario use case melakukan isi saldo

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
| 1. *Customer* memilih tombol Isi Saldo |  |
|  | 2. Sistem menampilkan *form* *input* nominal saldo yang ingin ditambahkan |
| 3. *Customer* memasukkan nominal yang ingin ditambahkan ke saldonya |  |
|  | 4. Sistem memvalidasi nominal yang dimasukkan *customer* |
|  | 5. Sistem menambahkan nominal uang ke saldo akun *customer* dan menampilkan notifikasi pada layar *customer* bahwa penambahan saldo berhasil |

Nama *Use* *Case*: *Balancing* Data Pengguna

Skenario:

Tabel 3.13 Skenario use case balancing data pengguna

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
|  | 1. Sistem meng-*update database* Data\_Customer dengan mengurangi nilai saldo *customer* sebesar biaya layanan. |
|  | 2. Sistem meng-*update database* Data\_Teknisi dengan menambahkan nilai saldo teknisi sebesar biaya layanan. |

Nama *Use* *Case*: Konfirmasi Pesanan

Skenario:

Tabel 3.14 Skenario use case konfirmasi pesanan

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
| 1. Teknisi memilih tombol Konfirmasi Pesanan |  |
|  | 2. Sistem menampilkan *form* berisi pilihan untuk menolak atau menerima pesanan |
| 3. Teknisi memasukkan pilihannya |  |
|  | 4. Sistem memverifikasi pesanan dan menampilkan pesan “Pesanan berhasil diterima” |
|  | 5. Sistem mengubah status pemesanan di *database* Order\_Log |
| **Skenario Alternatif** | |
| 1. Teknisi memilih tombol Konfirmasi Pesanan |  |
|  | 2. Sistem menampilkan *form* berisi pilihan untuk menolak atau menerima pesanan |
| 3. Teknisi memasukkan pilihannya |  |
|  | 4. Sistem memverifikasi pesanan dan menampilkan pesan “Pesanan berhasil ditolak” |
|  | 5. Sistem mengubah status pemesanan di *database* Order\_Log |

Nama *Use* *Case*: Mengontak *Customer*

Skenario:

Tabel 3.15 Skenario use case mengontak customer

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal 1** |  |
| 1. Teknisi memilih menu Percakapan |  |
|  | 2. Sistem menampilkan data pesanan yang masuk dan meminta konfirmasi teknisi |
| 3. Teknisi mengonfirmasi pesanan yang masuk |  |
|  | 4. Sistem menampilkan laman percakapan dengan *customer* yang pesanannya sudah terkonfirmasi tadi |
| 5. Teknisi mengetik pesan pada *field* percakapan |  |
| 6. Teknisi menekan tombol Kirim |  |
|  | 7. Sistem meneruskan pesan ke akun pengguna yang bersesuaian |
| **Skenario Normal 2** | |
| 1. Teknisi memilih menu Percakapan |  |
|  | 2. Sistem menampilkan laman percakapan dengan *customer* yang pesanannya sudah terkonfirmasi sebelumnya |
| 3. Teknisi mengetik pesan pada *field* percakapan |  |
| 4. Teknisi menekan tombol Kirim |  |
|  | 5. Sistem meneruskan pesan ke akun pengguna yang bersesuaian |
| **Skenario Alternatif** | |
| 1. Teknisi memilih menu Percakapan |  |
|  | 2. Sistem menampilkan laman percakapan namun tidak ada pesanan yang masuk sehingga tidak ada percakapan dengan satupun *customer* |
| 3. Teknisi menekan tombol Kembali |  |
|  | 5. Sistem menampilkan menu utama kembali |

Nama *Use* *Case*: *Cashout* Saldo

Skenario:

Tabel 3.16 Skenario use case melakukan cashout saldo

|  |  |
| --- | --- |
| **Aksi Aktor** | **Reaksi Sistem** |
| **Skenario Normal** |  |
| 1. Admin me-*request* sistem untuk mengurangi saldo teknisi tertentu sebanyak sebuah nominal |  |
|  | 2. Sistem mengecek jumlah saldo teknisi yang di-*request* oleh admin |
|  | 3. Sistem mendapati bahwa jumlah yang ingin dikurangi lebih kecil dibanding saldo yang dimiliki oleh teknisi tersebut sehingga sistem mengurangi saldo teknisi tersebut sebanyak nominal yang dimasukkan oleh admin. |
| **Skenario Alternatif** | |
| 1. Admin me-*request* sistem untuk mengurangi saldo teknisi tertentu sebayak sebuah nominal |  |
|  | 2. Sistem mengecek jumlah saldo teknisi yang di-*request* oleh admin |
|  | 3. Sistem mendapati bahwa jumlah yang ingin dikurangi lebih besar dibanding saldo yang dimiliki oleh teknisi tersebut sehingga sistem memberikan pesan kesalahan “Jumlah yang ingin ditarik *invalid*” |

# Model Analisis

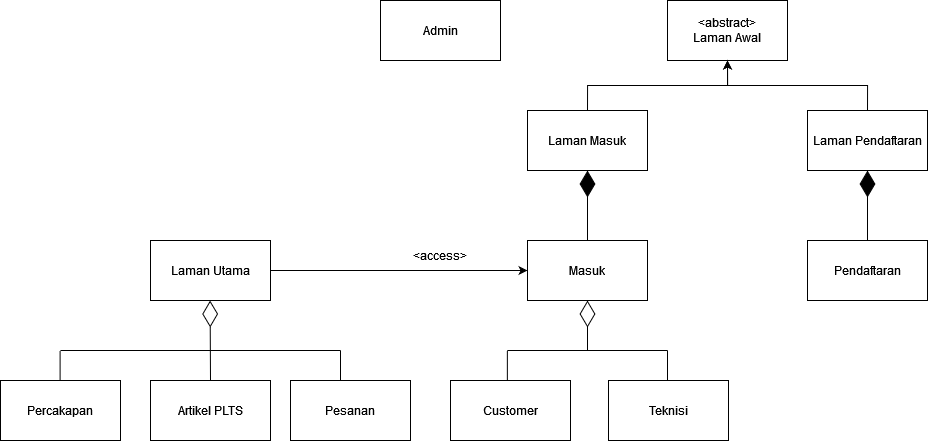
## Identifikasi Kelas

Tabel 4.1 Identifikasi kelas

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Kelas** | **Jenis (*Boundary*, *Controller*, *Entity*)** |
| IK01 | Customer | *Entity* |
| IK02 | Teknisi | *Entity* |
| IK03 | Admin | *Entity* |
| IK04 | Pesanan | *Entity* |
| IK05 | Pendaftaran | *Boundary* |
| IK06 | Percakapan | *Entity* |
| IK07 | Artikel PLTS | *Boundary* |
| IK08 | Laman Utama | *Controller* |
| IK09 | Masuk | *Controller* |
| IK10 | Laman Masuk | *Boundary* |
| IK11 | Laman Awal | *Controller* |
| IK12 | Pendaftaran | *Boundary* |

## Diagram Kelas

Berikut ini adalah gambar diagram kelas untuk perangkat lunak SuryaMenerangi.id:



Gambar 4.1 Diagram kelas

Sepesifikasi atribut dan *method* untuk setiap kelas pada diagram di atas adalah sebagai berikut:

Tabel 4.2 Diagram kelas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Kelas** | **Atribut/*Method*** | **Nama Atribut/*Method*** | **Keterangan** |
| 1 | Admin | *Method* | DaftarkanTeknisi(String nama, String tanggalLahir, integer nomorTelpon, string email, string password, integer saldo) | Mendaftarkan teknisi |
| 2 | Admin | *Method* | CashoutSaldo(Teknisi t, integer jumlah) | Melakukan *cashout* uang dari saldo perusahaan, kemudian diberikan ke teknisi. |
| 3 | Laman Awal | *Method* | TampilkanForm() | Menampilkan *form* |
| 4 | Laman Masuk | *Method* | TampilkanForm() | Menampilkan *form* masuk ke P/L |
| 5 | Laman Pendaftaran | *Method* | TampilkanForm() | Menampilkan *form* pendaftaran ke P/L |
| 6 | Masuk | Atribut | Email:string Password:string |  |
| 7 | Masuk | *Method* | Masuk(string email, string password) | Mengecek kombinasi *email* dan *password*. Jika benar, akan masuk ke laman utama. |
| 8 | Pendaftaran | Atribut | Email:string Password:string  Nama:string  Alamat:string  No.Telpon:string |  |
| 9 | Pendaftaran | *Method* | Daftar(string email, string password, string nama, string alamat, string noTelp) | Mendaftarkan akun baru sesuai atribut yang telah di-input oleh *customer* |
| 10 | Teknisi | Atribut | ID:integer  Nama:string  Rating:real |  |
| 11 | Customer | Atribut | ID:integer  Nama:string  Saldo:integer  Alamat:string |  |
| 12 | Customer | *Method* | IsiSaldo(Customer c, integer jumlahSaldo) | Mengisi saldo *customer* |
| 13 | Laman Utama | Atribut | NamaPengguna:string |  |
| 14 | Laman Utama | *Method* | TampilkanLamanUtama() | Menampilkan laman utama dan tombol-tombol menuju pemesanan, baca artikel, isi saldo, dan menampilkan nama dari pengguna |
| 15 | Percakapan | Atribut | IDPercakapan:integer  NamaPengirim:string  NamaPenerima:string  Pesan:string |  |
| 16 | Percakapan | *Method* | KirimPesan(String s) | Mengirim pesan yang diketik pengguna |
| 17 | Percakapan | *Method* | TampilkanPesan(Customer c, Teknisi t) | Menampilkan pesan yang dikirim oleh pengguna dan pesan yang masuk ke pengguna |
| 18 | Artikel PLTS | *Method* | TampilkanArtikel() | Menampilkan artikel yang tersedia ke *customer* |
| 19 | Pesanan | Atribut | PilihanPesanan:string  Biaya:integer  NamaCustomer:string  NamaTeknisi:string  AlamatCustomer:string  StatusPemesanan:string |  |
| 20 | Pesanan | *Method* | Pesan(string pilihan, int biaya, string namaCust, string namaTek, string alamat, string status) | Mengirimkan data pesanan ke *database* dan sistem melakukan validasi pesanan |
| 21 | Pesanan | *Method* | PotongSaldo(Customer c, integer jumlah) | Memotong saldo *customer* dengan syarat saldo *customer* mencukupi |
| 22 | Pesanan | *Method* | KonfimasiPesanan(Pesanan p) | Mengubah status pemesanan |

# Kerunutan (*Traceability*)

## Kebutuhan Fungsional vs Use Case

Tabel 5.1 Kebutuhan fungsional vs use case

|  |  |
| --- | --- |
| **ID Kebutuhan Fungsional** | **ID *Use* *Case* Terkait** |
| F01 | UC03, UC04, UC05 |
| F02 | UC05, UC10 |
| F03 | UC01 |
| F04 | UC02 |
| F05 | UC03, UC09 |
| F06 | UC08 |
| F07 | UC12 |
| F08 | UC06, UC11 |
| F09 | UC03 |
| F10 | UC07 |

## Use Case vs Kelas Terkait

Tabel 5.2 Use case vs kelas terkait

|  |  |
| --- | --- |
| **ID *Use* *Case*** | **Kelas Terkait** |
| UC01 | Laman Awal, Pendaftaran, Customer |
| UC02 | Laman Awal, Laman Masuk, Customer, Teknisi |
| UC03 | Laman Utama, Customer, Pesanan, |
| UC04 | Pesanan, Customer |
| UC05 | Laman Utama, Pesanan, Customer, Teknisi |
| UC06 | Laman Utama, Customer, Teknisi, Pesanan, Percakapan |
| UC07 | Laman Utama, Artikel PLTS |
| UC08 | Laman Utama, Customer |
| UC09 | Customer, Teknisi, Pesanan |
| UC10 | Pesanan, Teknisi, Pemesanan |
| UC11 | Laman Utama, Percakapan, Teknisi |
| UC12 | Admin, Teknisi |