**SKPL**-02

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

Aplikasi Hemat.com

untuk:

Membantu Pengguna Mengatur Keuangannya

Dipersiapkan oleh:

Farhan Alfariqi/1301161770

Shella Gustiana/1301164693

Rima Rizita I/1301160781

Refka M. Furqon/ 1301162748

Muhammad Derian Akbar/1301164722

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-02* | | 29 |
| Revisi |  | *04 Oktober 2018* |

# Daftar Perubahan

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A | 04 Oktober 2018, Memperbaiki Use Case dan melengkapi data SKPL yang masih belum tepat. |
| B | 22 Oktober 2018, Memperbaiki Use Case, Class Diagram dan Entity Relationship Diagram. |
| C |  |
| D | . |
| E |  |
| F |  |
| G |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX | - | A | B | C | D | E | F | G |
| TGL | 04/10/18 |  |  |  |  |  |  |  |
| Ditulis oleh | Kelompok 2 |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

# Daftar Halaman Perubahan

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

# Daftar Isi

[Daftar Perubahan 1](#_Toc529175009)

[Daftar Halaman Perubahan 2](#_Toc529175010)

[Daftar Isi 3](#_Toc529175011)

[1. Pendahuluan 4](#_Toc529175012)

[1.1 Tujuan Penulisan Dokumen 4](#_Toc529175013)

[1.2 Konvensi Dokumen 4](#_Toc529175014)

[1.3 Cakupan Produk 4](#_Toc529175015)

[1.4 Referensi 4](#_Toc529175016)

[2. Overall Description 5](#_Toc529175017)

[2.1 Perspektif Produk 5](#_Toc529175018)

[2.2 Fungsi Produk 5](#_Toc529175019)

[2.3 Kelas dan Karakteristik Pengguna 5](#_Toc529175020)

[2.4 Lingkungan Operasi 6](#_Toc529175021)

[2.5 Batasan Perancangan dan Implementasi 6](#_Toc529175022)

[2.6 Dokumentasi Pengguna 6](#_Toc529175023)

[2.7 Asumsi dan Dependensi 10](#_Toc529175024)

[3. Requirements Antarmuka Eksternal 11](#_Toc529175025)

[3.1 Antarmuka Pengguna 11](#_Toc529175026)

[3.2 Antarmuka Perangkat Keras 11](#_Toc529175027)

[3.3 Antarmuka Perangkat Lunak 11](#_Toc529175028)

[3.4 Antarmuka Komunikasi 11](#_Toc529175029)

[4. Fitur Sistem (Use Cases) 12](#_Toc529175030)

[4.1 Use Case 1 13](#_Toc529175031)

[4.2 Use Case 2 14](#_Toc529175032)

[4.3 Use Case 3 15](#_Toc529175033)

[4.4 Use Case 4 16](#_Toc529175034)

[4.5 Use Case 5 18](#_Toc529175035)

[4.6 Use Case 6 19](#_Toc529175036)

[4.7 Use Case 7 21](#_Toc529175037)

[4.8 Use Case 8 22](#_Toc529175038)

[4.9 Use Case 9 23](#_Toc529175039)

[4.10 Use Case 10 24](#_Toc529175040)

[4.11 Use Case 11 26](#_Toc529175041)

[4.12 Use Case 12 27](#_Toc529175042)

[4.13 Use Case 13 28](#_Toc529175043)

[4.14 Use Case 14 29](#_Toc529175044)

[5. Requirements Nonfungsional Lainnya 31](#_Toc529175045)

[5.1 Requirements Performa 31](#_Toc529175046)

[5.2 Requirements Keselamatan 31](#_Toc529175047)

[5.3 Requirements Keamanan 31](#_Toc529175048)

[5.4 Atribut Kualitas Perangkat Lunak 31](#_Toc529175049)

[6. Requirements Lain 32](#_Toc529175050)

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini ditunjukkan untuk memudahkan para pengguna dalam mengelola keuangan. Dokumen spesifikasi kebutuhan perangkat lunak (SKPL) merupakan spesifikasi kebutuhan perangkat lunak yang akan dikembangkan.

## Konvensi Dokumen

Konvensi dokumen dalam Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini berisi beberapa *requirement* yang dibutuhkan dalam perangkat lunak yang akan kami buat. Requirement yang **dicetak tebal** dibawah ini merupakan *requirement* utama dari perangkat lunak yang akan kami buat dan yang tidak dicetak tebal merupakan *requirement* turunannya, diantaranya adalah:

1. **Member dapat menginput budget yang diinginkan**
2. **Member dapat melihat budget yang sudah di inputkan**
3. Member dapat menambahkan barang impian yang diinginkan oleh member
4. Member dapat menginputkan dan melihat pemasukan
5. Member dapat menginputkan dan melihat pengeluaran

## Cakupan Produk

Hemat.com merupakan perangkat lunak yang memudahkan para user untuk mengelola keuangan user itu sendiri, berorientasi objek untuk monitoring atau memantau setiap transaksi yang telah dilaksanakan agar setiap uang yang masuk dan keluar menjadi lebih rapi dan padu sehingga tidak terjadi lagi kekurangan yang tidak diinginkan dan agar para user bisa memiliki barang impian sesuai dengan yang user inginkan. Sasaran membuat aplikasi ini sendiri lebih kepada pengguna yang sudah cukup umur dan yang sudah berpenghasilan.

## Referensi

Dokumen-dokumen yang digunakan sebagai referensi dalam pembuatan SKPL ini adalah

sebagai berikut:

### Template SKPL Fakultas Informatika, Telkom University.

1. Arie, A. (2014, September). *academia.edu.* Retrieved from academia.edu: http://www.academia.edu/11355568/Contoh\_SKPL\_SPESIFIKASI\_KEBUTUHAN\_PERANGKAT\_LUNAK\_

# Overall Description

## Perspektif Produk

Produk yang akan kami buat merupakan pengembangan dari sistem yang sudah ada, dikarenakan keinginan kami untuk mengembangkan sistem yang sudah ada untuk mengatasi masalah keuangan para user.

## Fungsi Produk

Berikut rangkuman utama dari produk kami :

### User melakukan sign-up apabila belum daftar sebelumnya, setelah itu login.

### User memilih menu utama yang diinginkan pada tampilan awal.

### Terdapat beberapa menu utama pada tampilan awal, diantaranya “ Mengelola Budget, Mengelola Barang Impian, dan Mengelola Hutang ”

### Pada menu Mengelola Budget terdapat pilihan “ Menginput Budget, Menginput Data Pengeluaran, Menginput Data Pemasukan, dan Melihat Data Budget ”yang akan diinput oleh user.

### Pada menu Mengelola Barang Impian terdapat pilihan “ Menginput Barang Impian, dan Mengubah Barang Impian ” yang akan diinput oleh user.

### Pada menu Mengelola Hutang terdapat pilihan “ Menginput Data Hutang, dan Melihat Data Hutang ” yang akan diinput oleh user.

## Kelas dan Karakteristik Pengguna

Pengguna pada aplikasi “Hemat.com” ditujukan kepada semua user yang memiliki masalah dalam mengatur keuangan mereka. Aplikasi tersebut dapat digunakan oleh semua jenis user, baik user yang memiliki segala pengetahuan tentang perangkat lunak maupun user yang memiliki sedikit pengetahuan tentang perangkat lunak.

## Lingkungan Operasi

* Kebutuhan perangkat keras dan lunak untuk user :

1. Sistem informasi “Hemat.com” hanya dapat dijalankan pada sistem operasi Android dan IOS dengan minimal ram 1GB, internal memory minimal 128 MB.
2. Jenis perangkat lunak yang digunakan minimal versi Marshmallow untuk Android dan versi 10.1.1 untuk OS IOS
3. Menggunakan Handphone, dan Tablet.

* Kebutuhan perangkat keras dan lunak untuk server :

1. Membuat aplikasi “Hemat.com” menggunakan Android Studio dengan spesifikasi ram 1GB.
2. Menggunakan laptop dengan spesifikasi minimal Processor Intel Core i3 ram 4GB.
3. Perangkat lunak dengan spesifikasi minimal Windows 8.1 untuk Mac Intel Core i5 dual-core 1.6 GHz ram 4GB.

## Batasan Perancangan dan Implementasi

Pengembangan sistem informasi “Hemat.com” ini memiliki beberapa keterbatasan sebagai berikut :

### Sistem informasi “Hemat.com” hanya dapat dijalankan pada sistem operasi Android dan IOS.

### Sistem informasi “Hemat.com” merupakan perangkat lunak berorientasi objek.

### Sistem informasi “Hemat.com” hanya bisa digunakan apabila saat online.

### Sistem informasi “Hemat.com” hanya bisa digunakan 1 akun saja.

## Dokumentasi Pengguna

### Dokumentasi Login.

1. *User* membuka aplikasi.
2. *User* memilih menu *Login*.
3. *User* memasukan *ussername* dan *password* pada form login.
4. Kemudian *user* mengklik button “Submit”, apabila user mengklik button "Submit” tanpa mengisi form maka keluar peringatan.

### Dokumentasi Registrasi.

1. User membuka aplikasi.
2. Akan ditampilkan menu Login, kemudian user memilih menu Sign-Up.
3. *User* memasukan *field* sesuai data diri.
4. Kemudian *user* mengklik *button* “Submit”, apabila *user* mengklik button "Submit” tanpa mengisi form maka keluar peringatan.

### Dokumentasi Input Pengeluaran.

1. Ketika user mengklik fitur Mengelola Budget kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Input Pengeluaran.
3. Ketika *user* sudah menginput data pengeluaran pada form kemudian mengklik button "Input", apabila user mengklik button "Input” tanpa mengisi form maka keluar peringatan.
4. Setelah form terisi dan mengklik button "Save" maka akan timbul message "Apakah anda yakin ingin mengubah data?".
5. Kemudian user mengklik button "Yes" pada message yang timbul maka data pengeluaran tersebut tersimpan ke database.

### Dokumentasi Input Pemasukan.

1. Ketika user mengklik fitur Mengelola Budget kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Input Pemasukan.
3. Ketika user sudah menginput data pemasukan pada form kemudian mengklik button "Input", apabila user mengklik button "Input” tanpa mengisi form maka keluar peringatan.
4. Setelah form terisi dan mengklik button "Save" maka akan timbul message "Apakah anda yakin ingin mengubah data?".
5. Kemudian user mengklik button "Yes" pada message yang timbul maka data pemasukan tersebut tersimpan ke database.

### Dokumentasi Melihat Budget.

1. Ketika user mengklik fitur Mengelola Budget kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Melihat Budget.
3. Kemudian akan ditampilkan budget dari user.
4. Setelah selesai melihat budget, user mengklik button “Done” pada message yang timbul, maka user akan kembali pada menu Mengelola Budget.

### Dokumentasi Input Barang Impian.

1. Ketika user mengklik fitur Mengelola Barang Impian kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Input Barang Impian.
3. Ketika user sudah menginput barang impian pada form kemudian mengklik button "Input", apabila user mengklik button "Input” tanpa mengisi form maka keluar peringatan.
4. Setelah form terisi dan mengklik button "Save" maka akan timbul message "Apakah anda yakin ingin menginput data?".
5. Kemudian user mengklik button "Yes" pada message yang timbul maka data barang impian tersebut tersimpan ke database.

### Dokumentasi Ubah Barang Impian.

1. Ketika user mengklik fitur Mengelola Barang Impian kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Ubah Barang Impian.
3. Ketika user sudah mengubah barang impian pada form kemudian mengklik button "Input", apabila user mengklik button "Input” tanpa mengisi form maka keluar peringatan.
4. Setelah form terisi dan mengklik button "Save" maka akan timbul message "Apakah anda yakin ingin mengubah data?".
5. Kemudian user mengklik button "Yes" pada message yang timbul maka data barang impian tersebut tersimpan ke database.

### Dokumentasi Hapus Barang Impian.

1. Ketika user mengklik fitur Mengelola Barang Impian kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Hapus Barang Impian.
3. Ketika user sudah menghapus barang impian pada form kemudian mengklik button "Input", apabila user mengklik button "Input” tanpa mengisi form maka keluar peringatan.
4. Setelah form terisi dan mengklik button "Save" maka akan timbul message "Apakah anda yakin ingin menghapus data?".
5. Kemudian user mengklik button "Yes" pada message yang timbul maka data barang impian tersebut tersimpan ke database.

### Dokumentasi Melihat Barang Impian.

1. Ketika user mengklik fitur Mengelola Barang Impian kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Melihat Barang Impian.
3. Kemudian akan ditampilkan barang impian dari user.
4. Setelah selesai melihat barang impian, user mengklik button “Done” pada message yang timbul, maka user akan kembali pada menu Mengelola Barang Impian.

### Dokumentasi Input Hutang.

1. Ketika user mengklik fitur Mengelola Hutang kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Input Hutang.
3. Ketika user sudah menginput hutang pada form kemudian mengklik button "Input", apabila user mengklik button "Input” tanpa mengisi form maka keluar peringatan.
4. Setelah form terisi dan mengklik button "Save" maka akan timbul message "Apakah anda yakin ingin menginput data?".
5. Kemudian user mengklik button "Yes" pada message yang timbul maka data hutang tersebut tersimpan ke database.

### Dokumentasi Ubah Hutang.

1. Ketika user mengklik fitur Mengelola Hutang kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Ubah Hutang.
3. Ketika user sudah mengubah hutang pada form kemudian mengklik button "Input", apabila user mengklik button "Input” tanpa mengisi form maka keluar peringatan.
4. Setelah form terisi dan mengklik button "Save" maka akan timbul message "Apakah anda yakin ingin mengubah data?".
5. Kemudian user mengklik button "Yes" pada message yang timbul maka data hutang tersebut tersimpan ke database.

### Dokumentasi Hapus Hutang.

1. Ketika user mengklik fitur Mengelola Hutang kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Hapus Hutang.
3. Ketika user sudah menghapus hutang impian pada form kemudian mengklik button "Input", apabila user mengklik button "Input” tanpa mengisi form maka keluar peringatan.
4. Setelah form terisi dan mengklik button "Save" maka akan timbul message "Apakah anda yakin ingin menghapus data?".
5. Kemudian user mengklik button "Yes" pada message yang timbul maka data hutang tersebut tersimpan ke database.

### Dokumentasi Melihat Hutang.

1. Ketika user mengklik fitur Mengelola Hutang kemudian akan ditampilkan beberapa menu pilihan.
2. User memilih menu Melihat Hutang.
3. Kemudian akan ditampilkan hutang dari user.
4. Setelah selesai melihat hutang, user mengklik button “Done” pada message yang timbul, maka user akan kembali pada menu Mengelola Hutang.

## Asumsi dan Dependensi

Asumsi :

* + - 1. User memiliki hak akses penuh dalam penggunaan aplikasi “Hemat.com”.
      2. Aplikasi ini dapat dikategorikan sebagai Mobile Store yang dapat dioperasikan kapanpun dan dimanapun.

Kebergantungan :

Sistem informasi ini hanya bisa diakses apabila ada koneksi internet.

Sistem informasi ini hanya dapat dijalankan pada sistem operasi android.

User harus melakukan login untuk menjalankan semua sistem informasi “Hemat.com”.

# Requirements Antarmuka Eksternal

## Antarmuka Pengguna

Pada saat pengguna memasuki aplikasi ini, pengguna akan melihat tampilan utama berupa menu-menu yang merupakan fungsionalitas dari perangkat lunak ini berupa select menu,input nominal, edit nominal, hapus nominal, select barang impian, select hutang. Pengguna hanya perlu tap menu yang ada dilayar kemudian bisa menjalankan fungsionalitas pada aplikasi ini. Ketika kesulitan dalam menggunakan aplikasi ini ada fitur bantuan di laman home pojok kanan bawah, disitu terdapat cara penggunaan aplikasi ini.

## Antarmuka Perangkat Keras

Jenis perangkat keras yang bisa menggunakan aplikasi ini semua jenis smartphone android dan IOS. Spesifikasi yang dibutuhkan minimal ram 1GB, internal memory minimal 128 MB. Perangkat keras server yang membuat aplikasi ini menggunakan laptop dengan spesifikasi minimal Processor Intel Core i3 ram 4GB

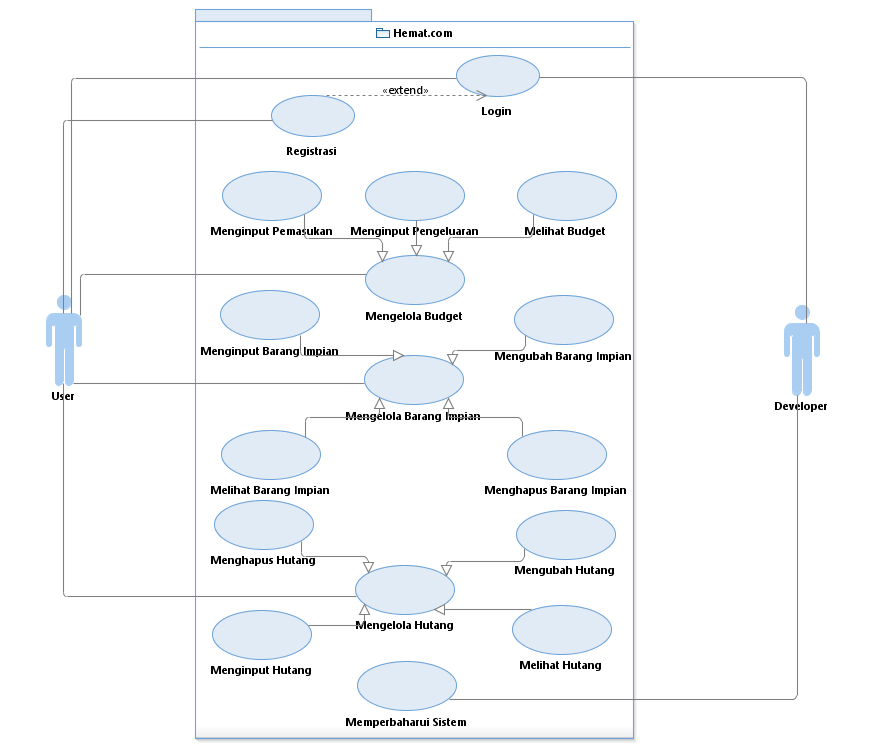
## Antarmuka Perangkat Lunak

Jenis perangkat lunak dengan minimal versi marshmallow untuk Android dan versi 10.1.1 untuk iOS. Jenis perangkat lunak server dengan spesifikasi minimal Windows 8.1 untuk Mac Intel Core i5 dual-core 1,6 GHz RAM 4GB, dibuat dengan Android Studio dengan spesifikasi RAM 1GB.

## Antarmuka Komunikasi

Aplikasi yang digunakan akan di hubungkan dengan internet,karena aplikasi ini ada update data yang dilakukan oleh developer, tetapi jika tidak ada internet tetap bisa menggunakan aplikasi ini untuk mengatur kebutuhan kita hanya saja tidak terupdate.

# Fitur Sistem (Use Cases)



## Use Case 1

### Nama Use Case:

### Login

### Tujuan

### Untuk masuk kedalam aplikasi dan agar dapat menggunakan semua fitur di dalam aplikasi secara optimal

### Input

*User* telah melakukan proses registrasi terlebih dahulu

### Output:

*User* berhasil masuk kedalam aplikasi

### Skenario Utama

Agar *User* dapat mengakses semua fitur didalam aplikasi

### Prakondisi

### User telah memiliki data login berupa username dan password

### Langkah-langkah:

#### User membuka aplikasi

#### Sistem akan menampilkan form login yang perlu diisi oleh User

#### User mengisi data username dan password yang sesuai

#### User mengklik tombol “Submit”

#### Sistem akan memvalidasi inputan user

#### Jika username dan password sesuai, maka sistem menampilkan halaman utama aplikasi

### Pascakondisi

*User* berhasil masuk kedalam halaman utama aplikasi, dan siap untuk mengakses semua fitur yang telah disediakan.

### Skenario eksepsional 1

* Alt-step 4 :

Jika *User* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop*-*up* “Anda belum mengisi data *Login*” dan kembali ke *step* 3.

## Use Case 2

### Nama Use Case:

### Registrasi

### Tujuan

### Untuk dapat menggunakan aplikasi, maka User diwajibkan untuk mendaftar agar mendapatkan username dan password yang nantinya akan digunakan dalam tahap Login.

### Input

*User* telah membuka aplikasi

### Output:

*User* berhasil mendapatkan akun yang berisikan *username* dan *password*

### Skenario Utama

Agar *User* dapat mengakses semua fitur didalam aplikasi

### Prakondisi

### User telah menyiapkan data diri berupa nama, alamat, tanggal lahir beserta informasi seperti nomor telpon dan sebagainya

### Langkah-langkah:

#### User membuka aplikasi

#### Sistem akan menampilkan form login

#### User mengklik tombol “Sign-Up”

#### Sistem akan menampilkan form Sign-Up

#### User menginputkan isi field sesuai dengan data diri, dan lalu mengklik tombol “Submit”

#### Jika data diri sesuai, maka sistem akan beralih menampilkan halaman Login kembali.

### Pascakondisi

*User* berhasil masuk kedalam halaman *login*, dan telah memiliki *username* dan *password* yang akan digunakan untuk *login* kedalam aplikasi

### Skenario eksepsional 1

* Alt-step 5 :

Jika *User* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop*-*up* “Anda belum mengisi data diri” dan kembali ke *step* 3.

## Use Case 3

### Nama Use Case:

### Melihat Budget

### Tujuan

### Untuk melihat semua aliran data budget yang masuk, baik data pemasukan maupun pengeluaran.

### Input

*User* telah melakukan *login*

### Output:

Data *Budget* berhasil ditampilkan

### Skenario Utama

Menampilkan data pemasukan dan pengeluaran dari keseluruhan data yang tersimpan pada *Database*.

### Prakondisi

### User harus sudah memiliki akun.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “View”

#### User mengklik tombol submenu “Budget”

#### Sistem akan menampilkan informasi budget, baik data pemasukan maupun pengeluaran melalui suatu laporan visual

### Pascakondisi

Sistem berhasil menampilkan data budget dalam suatu laporan visual

### Skenario eksepsional 1

* Alt-step 1 :

Jika *user* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop-up* “Anda belum menginputkan data *Login*” dan kembali ke *step* 1.

## Use Case 4

### Nama Use Case:

### Menginput Pengeluaran.

### Tujuan

### Untuk memasukkan jumlah budget yang dikeluarkan.

### Input

*User* menginputkan jumlah *budget* yang sesuai dengan pengeluaran.

### Output:

*Budget* berhasil di-inputkan ke dalam *Database*.

### Skenario Utama

Menginputkan jumlah *budget* pengeluaran yang sesuai, agar data *budget* tersebut tersimpan di *Database*.

### Prakondisi

### User harus sudah memiliki akun dan sudah pernah menginputkan budget awal.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User menginputkan budget di kolom yang tersedia pada menu pengeluaran

#### User mengklik tombol “Submit”

#### Sistem akan menampilkan informasi pengeluaran melalui laporan

#### Budget yang telah di-input akan tersimpan pada Database

### Pascakondisi

*User* berhasil menginput *budget* pengeluaran dan data *budget* pengeluaran tersimpan pada *Database.*

### Skenario eksepsional 1

* Alt-step 4 :

Jika *user* mengklik tombol “*Reset*” maka data *budget* yang telah diinput akan di-*reset*.

* Alt-step 5 :

Jika *user* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop-up* “Anda belum menginputkan *budget*” dan kembali ke *step* 4.

## Use Case 5

### Nama Use Case:

### Menginput Pemasukan.

### Tujuan

### Untuk memasukkan jumlah budget yang masuk.

### Input

*User* menginputkan jumlah *budget* yang sesuai dengan pemasukan.

### Output:

*Budget* berhasil di-inputkan ke dalam *Database*.

### Skenario Utama

Menginputkan jumlah *budget* pemasukan yang sesuai, agar data *budget* tersebut tersimpan di *Database*.

### Prakondisi

### User harus sudah memiliki akun dan sudah pernah menginputkan budget awal.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User menginputkan budget di kolom yang tersedia pada menu pemasukan

#### User mengklik tombol “Submit”

#### Sistem akan menampilkan informasi pemasukan melalui laporan

#### Budget yang telah diinput akan tersimpan pada Database

### Pascakondisi

*User* berhasil menginput *budget* pemasukan dan data *budget* pemasukan tersimpan pada *Database.*

### Skenario eksepsional 1

* Alt-step 4 :

Jika *user* mengklik tombol “*Reset*” maka data *budget* yang telah diinput akan di-*reset*.

* Alt-step 5 :

Jika *user* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop*-*up* “anda belum memilih menginputkan *budget*” dan kembali ke *step* 4.

## Use Case 6

### Nama Use Case:

### Menginput Barang Impian.

### Tujuan

### Untuk melihat perkiraan berapa uang yang diperlukan perharinya, perminggu dll. yang berfungsi sebagai perencanaan jangka panjang.

### Input

*User* menginputkan nama barang, harga beserta *deadline* barang yang diinginkan.

### Output:

Data barang impian berhasil di-inputkan ke dalam *Database*.

### Skenario Utama

Menginputkan barang impian yang diinginkan, lalu oleh sistem akan dikalkulasikan jumlah uang yang harus disimpan tiap waktunya sesuai batas *deadline* yang telah diinputkan sebelumnya.

### Prakondisi

### User harus sudah menginputkan budget awal.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “Barang Impian”

#### Sistem akan menampilkan form inputan barang impian

#### User diharuskan mengisi nama, harga, beserta deadline (batas waktu) dari Barang Impian

#### User mengklik tombol “Submit”, maka data barang impian akan tersimpan di Database

### Pascakondisi

*User* berhasil menginput data barang impian, dan sistem akan menampilkan informasi rinci mengenai barang impian, dari nama, harga, serta batas waktunya.

### Skenario eksepsional 1

* Alt-step 5 :

Jika *User* mengklik tombol “*Reset*” maka data barang impian yang telah di-*input* akan di reset.

* Alt-step 7 :

Jika *User* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop*-*up* “Anda belum menginputkan barang impian” dan kembali ke *step* 6.

## Use Case 7

### Nama Use Case:

### Mengubah Barang Impian.

### Tujuan

### Untuk mengubah data barang impian yang telah diinputkan pada Database, serta hasil dari pengubahan tersebut akan tersimpan di dalam Database.

### Input

*User* mengubah nama barang, harga beserta deadline barang yang diinginkan.

### Output:

Hasil pengubahan data barang impian akan tersimpan pada *Database*.

### Skenario Utama

Mengubah data barang impian mulai dari nama barang, harga beserta *deadline* dari barang impian tersebut. Sehingga nantinya, data hasil pengubahan tersebut akan tersimpan pada *Database*.

### Prakondisi

### User harus sudah menginputkan data barang impian.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “Barang Impian”

#### User memilih dan mengklik barang impian yang ingin diubah.

#### User mengubah nama, harga, beserta deadline (batas waktu) dari Barang Impian

#### User mengklik tombol “Submit”, maka data barang impian akan tersimpan di Database

### Pascakondisi

*User* berhasil mengubah data barang impian, serta data barang impian tersebut akan tersimpan pada *Database*.

### Skenario eksepsional 1

* Alt-step 7 :

Jika *User* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop*-*up* “Anda belum menginputkan barang impian” dan kembali ke *step* 6.

## Use Case 8

### Nama Use Case:

### Melihat Barang Impian

### Tujuan

### Untuk mengetahui seluruh daftar barang impian yang terdapat didalam Database.

### Input

*User* telah menginputkan data barang impian, yang terdiri dari nama, deskripsi, beserta nominal hutang.

### Output:

Menampilkan data barang impian yang tersimpan pada *Database.*

### Skenario Utama

Jika *User* sudah pernah menginputkan data barang impian, maka nanti akan muncul daftar seluruh barang impian dalam bentuk tabel.

### Prakondisi

### User telah menginputkan data barang impian.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “View”

#### User memilih dan mengklik submenu “Barang Impian”

#### Sistem akan menampilkan seluruh barang impian yang pernah disimpan didalam Database.

### Pascakondisi

*User* dapat melihat keseluruhan data barang impian dalam bentuk tabel.

### Skenario eksepsional 1

* Alt-step 2 :

Jika autentifikasi *User* gagal, maka akan kembali ke step 1.

## Use Case 9

### Nama Use Case:

### Menghapus Barang Impian

### Tujuan

### Untuk menghapus data dari daftar barang impian yang terdapat didalam Database.

### Input

*User* telah menghapus data barang impian, yang terdiri dari nama, deskripsi, beserta nominal

### Output:

Menghapus data barang impian yang diinginkan, lalu menampilkan data barang impian yang tersisa pada *Database.*

### Skenario Utama

Jika *User* sudah pernah menginputkan data barang impian yang diinginkan lalu menghapusnya, maka nanti akan muncul daftar selain barang impian yang dihapus dalam bentuk tabel.

### Prakondisi

### User telah menginputkan data barang impian.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “Barang Impian”

#### User memilih dan mengklik “Delete” pada data yang ingin dihapus

#### Sistem akan menampilkan seluruh barang impian yang tersisa didalam Database.

### Pascakondisi

*User* dapat menghapus data, serta melihat data barang impian lainnya yang tersisa dalam bentuk tabel.

### Skenario eksepsional 1

* Alt-step 2 :

Jika autentifikasi *User* gagal, maka akan kembali ke step 1.

## Use Case 10

### Nama Use Case:

### Menginput Hutang

### Tujuan

### Untuk menginputkan data hutang serta data input tersebut akan dikalkulasikan pada data budget.

### Input

*User* menginputkan data hutang, yang terdiri dari nama, deskripsi, beserta nominal utang.

### Output:

Data hutang berhasil di-inputkan ke dalam *Database*.

### Skenario Utama

Menginputkan data hutang, lalu sistem akan mengkalkulasikan semua *budget* hingga menampilkan total *budget* yang sebenarnya.

### Prakondisi

### User telah memiliki akun.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “Hutang”

#### Sistem akan menampilkan form inputan data hutang

#### User diharuskan mengisi nama, deskripsi, berserta nominal dari data hutang

#### User mengklik tombol “Submit”, maka hutang akan tersimpan di Database

### Pascakondisi

*User* berhasil menginput data hutang dan sistem akan menampilkan keseluruhan total *budget* yang dimiliki oleh *user* berdasarkan data *budget* awal, pemasukan dan pengeluaran beserta data hutang.

### Skenario eksepsional 1

* Alt-step 6 :

Jika *user* mengklik tombol “*Reset*” maka data hutang yang telah di-*input* akan di-*reset*.

* Alt-step 7 :

Jika *user* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop*-*up* “Anda belum menginputkan data hutang” dan kembali ke *step* 6.

## Use Case 11

### Nama Use Case:

### Melihat Hutang

### Tujuan

### Untuk mengetahui total budget yang dimiliki dari hasil perhitungan budget awal ditambah hasil data hutang.

### Input

*User* telah menginputkan data hutang, yang terdiri dari nama, deskripsi, beserta nominal hutang.

### Output:

Menampilkan data hutang beserta total *budget* yang tersimpan pada *Database.*

### Skenario Utama

Jika *User* sudah pernah Menginputkan data hutang, maka sistem akan menampilkan dan mengkalkulasikan semua *budget* hingga menampilkan total *budget* yang sebenarnya. Namun, jika *User* belum memiliki data *budget*, maka yang ditampilkan hanya uang dari hasil hutang-piutang.

### Prakondisi

### User telah menginputkan data hutang

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “View”

#### User memilih dan mengklik submenu “Hutang”

#### Sistem akan menampilkan seluruh total budget yang dimiliki User, yangmana total budget tersebut merupakan gabungan budget awal dan data hutang

### Pascakondisi

*User* dapat melihat keseluruhan data total *budget* dalam bentuk tabel dan grafik.

### Skenario eksepsional 1

* Alt-step 2 :

Jika autentifikasi *User* gagal, maka akan kembali ke step 1.

## Use Case 12

### Nama Use Case:

### Menghapus Hutang

### Tujuan

### Untuk menghapus data hutang dari daftar data hutang yang terdapat didalam Database.

### Input

*User* telah menghapus data hutang, yang terdiri dari nama, deskripsi, beserta nominal hutang.

### Output:

Menghapus data hutang yang diinginkan, lalu menampilkan data hutang yang tersisa pada *Database.*

### Skenario Utama

Jika *User* sudah pernah menginputkan data hutang yang diinginkan lalu menghapusnya, maka nanti akan muncul daftar selain data hutang yang dihapus dalam bentuk tabel.

### Prakondisi

### User telah menginputkan data hutang.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “Hutang”

#### User memilih dan mengklik “Delete” pada data yang ingin dihapus

#### Sistem akan menampilkan seluruh data hutang yang tersisa didalam Database.

### Pascakondisi

*User* dapat menghapus data, serta melihat data hutang lainnya yang tersisa dalam bentuk tabel.

### Skenario eksepsional 1

* Alt-step 2 :

Jika autentifikasi *User* gagal, maka akan kembali ke step 1.

## Use Case 13

### Nama Use Case:

### Mengubah Hutang.

### Tujuan

### Untuk mengubah data hutang yang telah diinputkan pada Database, serta hasil dari pengubahan tersebut akan tersimpan di dalam Database.

### Input

*User* mengubah nama serta nominal dari data hutang yang diinginkan.

### Output:

Hasil pengubahan data hutang akan tersimpan pada *Database*.

### Skenario Utama

Mengubah data hutang mulai dari nama beserta nominal dari data hutang. Sehingga nantinya, data hasil pengubahan tersebut akan tersimpan pada *Database*.

### Prakondisi

### User harus sudah menginputkan data hutang.

### Langkah-langkah:

#### User melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun user

#### User berhasil memasuki aplikasi

#### User mengklik menu “Hutang”

#### User memilih dan mengklik data hutang yang ingin diubah.

#### User mengubah nama, serta nominal dari Data Hutang

#### User mengklik tombol “Submit”, maka data barang impian akan tersimpan di Database

### Pascakondisi

*User* berhasil mengubah data hutang, serta data hutang tersebut akan tersimpan pada *Database*.

### Skenario eksepsional 1

* Alt-step 7 :

Jika *User* mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop*-*up* “Anda belum menginputkan data hutang” dan kembali ke *step* 6.

## Use Case 14

### Nama Use Case:

### Memperbaharui Sistem

### Tujuan

### Untuk memperbaharui sistem yang ada, sehingga setelah diperbaharu, diharapkan adanya peningkatan pada performa maupun fitur dari aplikasi itu sendiri.

### Input

*Developer* melakukan *back-up* data dari *user*

### Output:

Hasil pembaharuan tersebut nantinya diharapkan dapat meningkatkan kemampuan maupun performa baik dari aplikasi maupun user itu sendiri.

### Skenario Utama

Memperbaharui sistem yang telah dan meningkatkan sistem tersebut agar lebih optimal.

### Prakondisi

### Developer telah merancang source code program yang ter-update.

### Langkah-langkah:

#### Developer melakukan login untuk memasuki aplikasi

#### Sistem akan mevalidasi akun developer

#### Developer berhasil memasuki aplikasi

#### Developer mengklik menu “Update”

#### Sistem akan memberhentikan keseluruhan fitur aplikasi dari segi user

#### Sistem berhasil diperbaharui

#### Sistem akan menjalankan kembali fitur yang telah diberhentikan sementara tadi.

### Pascakondisi

*Developer* berhasil memperbaharui sistem, sehingga sistem siap digunakan secara optimal

### Skenario eksepsional 1

* Alt-step 2 :

Jika Developer mengklik tombol “*Submit*” tanpa mengisi form maka muncul *pop*-*up* “Anda belum mengisi data *Login*” dan kembali ke *step* 1.

# Requirements Nonfungsional Lainnya

## Requirements Performa

Perhitungan pemasukan dan pegeluaran yang cepat dan tepat serta dokumentasi keuangan yang rapih dan teratur. Proses membuka perangkat lunak juga menjadi perhitungan, performa opening dan closing prangkat lunak juga menjadi pertimbangan *user* mau menggunakan perangkat lunak ini.

## Requirements Keselamatan

Setiap satu minggu sekali pada saat perangkat lunak aktif maka sistem secara otomatis akan menggandakan data, redudansi data ini diperuntukan menjaga catatan keuangan. Sehingga pada saat perangkat lunak error, data pada perangkat lunak ini lebih terjamin keselamatannya.

## Requirements Keamanan

Pada aplikasi pengguna bisa menggunakan secara offline sehingga tidak bisa diakses maupun diretas oleh pihak luar. Pada saat online pun database user akan terjaga dengan aman karena saat data pengguna akan diakses membutuhkan autentifikasi dalam login pengguna.

## Atribut Kualitas Perangkat Lunak

**Ketepatan** : Sistem memastikan bahwa dalam perhitungan setiap pengeluaran, pemasukan dan mengestimasi pembelian barang impian dengan harga yang diinputkan selalu menghasilkan perhitungan yang tepat.

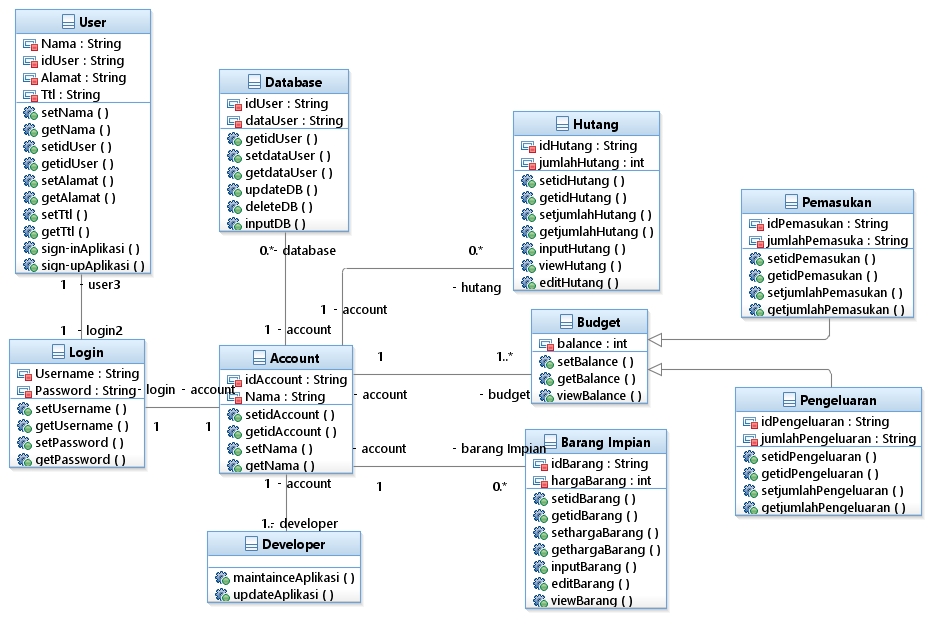
**Kecepetan** : Sistem memastikan bahwa perangkat lunak yang akan digunakan pengguna bersifat ringan dan tanpa delay. Sehingga pengguna yang memiliki spesifikasi smartphone yang minim, dapat menikmati perangkat lunak secara lancar.

# Requirements Lain

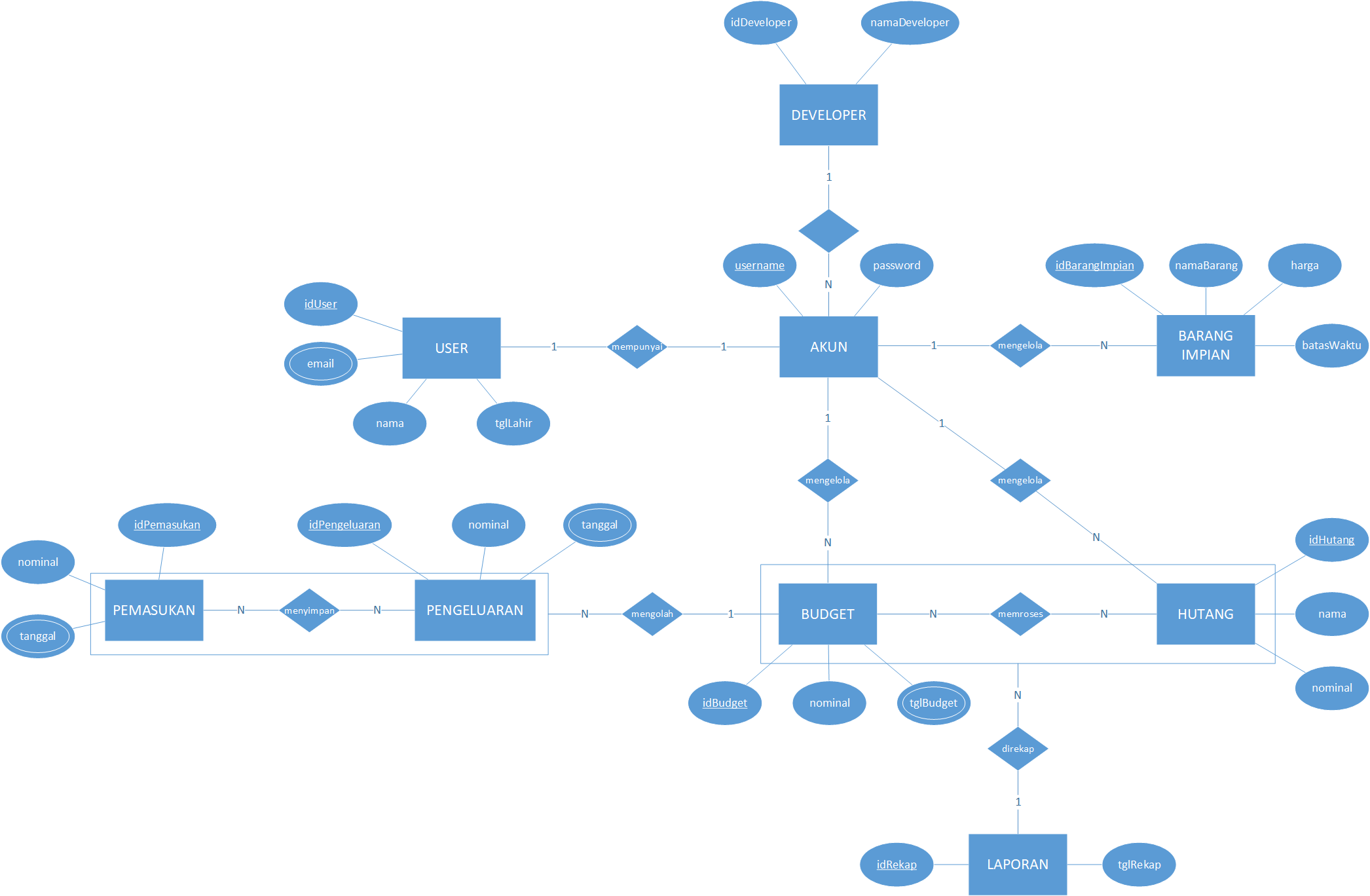
**Lampiran A: Daftar Kata-Kata Sukar**

|  |  |
| --- | --- |
| Istilah | Penjelasan |
| SKPL | Kepanjangan dari spesifikasi kebutuhan perangkat lunak, yang digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya |
| Use Case | Use Case merupakan merepresentasikan sebuah interaksi antara aktor (user atau sistem lainya) dengan sistem. |

**Lampiran B: Class Diagram**



**Lampiran C: ERD**

****