Software Requirements Specification

for

Penghitungan Pajak Tombak Bangsa

Version 1.0 approved

Prepared by

Tohap Maruli Pasaribu 1901525921

Reni Kartika 2101663281

I Gede Arya Krisnadi 2101668553

Naufal Hafiz 2101712390

Regy Mirza Arfiansyah 2101714080

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

Dokumen ini adalah dokumen yang dibuat guna menunjukkan secara spesifik tentang projek pembuatan aplikasi penghitung kalori yang akan dilakukan oleh developer dalam mengembangkan software.

## Document Conventions

Dokumen ini dibuat berdasarkan template IEEE untuk Dokumen System Requirement Specification

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Product Scope

Aplikasi ini merupakan aplikasi berbasis mobile application dengan tampilan yang user friendly. Aplikasi ini ditujukan untuk mereka yang ingin lebih memperhatikan pola makan dan juga olahraga, dan bagi mereka yang ingin mengetahui seberapa banyak kalori yang dikonsumsi setiap hari, dan kalori yang dibakar dari olahraga yang mereka lakukan. Dalam aplikasi ini, user akan diminta untuk mengisi data diri (seperti : Nama, Umur, Jenis kelamin, Berat badan, dan Tinggi badan). Kemudian user akan diarahkan kehalaman dimana mereka akan mengisi makanan apa saja yang telah mereka konsumsi, lalu akan dilakukan pengkalkulasian untuk menghitung total kalorinya, yang kemudian akan diarahkan ke halaman olahraga, user akan menginput lama waktu berlari, yang nantinya akan dikalkulasi untuk menghitung kalori yang terbakar dan akumulasi kalori dari awal hingga akhir.

## References

Ada beberapa jurnal yang dijadikan referensi untuk dipertimbangkan oleh developer, seperti jurnal yang ditulis oleh David Benton & Hayley A. Young dengan jurnal mereka yang berjudul Reducing calorie intake may not help you lose body weight yang diterbitkan pada tanggal 28 Juni 2017. Selain itu juga menggunakan jurnal yang berjudul Calories in, calories out and macronutrient intake : the hope, hype, and science of calories yang diterbitkan pada 29 November 2017 oleh Scott Howell. Sementara untuk info kalori pada tiap makanan, diambil pada <https://www.fatsecret.com/calories-nutrition/>.

# Overall Description

## Product Perspective

Aplikasi ini merupakan aplikasi berbasis mobile application yang akan digunakan untuk menghitung akumalasi kalori yang masuk dalam tubuh(makan) dan yang keluar dari tubuh(setelah berolahraga). Aplikasi ini tidak membutuhkan sambungan ke internet ataupun GPS. Dalam aplikasi inipun tidak dibutuhkan database, hanya dibutuhkan data dictionary yang ditunjukkan pada tabel dibawah.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Data Format | Field Size | Description | Example |
| NamaMakanMinum | string |  | 30 | Nama dari makanan/minuman | Nasi Putih |
| Jenis | string |  | 20 | Jenis (tipe) dari makanan/minuman | karbo |
| Qty | integer |  | 3 | Kuantitas dari makanan/minuman | 100 gram |
| Kalori | integer |  | 5 | Jumlah kalori dari makanan/minuman | 129 kalori |
| Thumbnail | string | .jpg | 50 | Keterangan gambar untuk setiap Makanan/Minuman | 20Mar2019153738.jpg |

## Product Functions

Aplikasi ini dibuat dengan tujuan memantau secara mudah, apakah makanan yang dikonsumsi sudah sebanding dengan olahraga yang sudah dilakukan, sehingga dapat membantu pengguna dalam menjaga pola makan dan olahraga secara teratur. Aplikasi ini diharapkan untuk dapat dijangkau oleh semua kalangan, yang dimana pada masa sekarang, masyarakat sudah jarang memperhatikan pola makan mereka.

## User Classes and Characteristics

Dalam aplikasi ini, hanya ada satu tipe user yang akan berinteraksi dengan sistem, yaitu user untuk aplikasi penghitung kalori, mereka dapat memasukkan data diri, memilih makanan apa saja yang sudah dikonsumsi beserta kuantitasnya, dan juga dapat memasukkan lama waktu berolahraga. Aplikasi ini tidak memerlukan admin dalam pelaksanaannya.

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.> MANUAL BOOK

## Assumptions and Dependencies

Asumsi yang ada pada aplikasi ini adalah hanya akan digunakan pada mobile phone yang memiliki performance yang memadai.

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

Aplikasi ini akan mengakses touch screen dari device yang dimiliki oleh pengguna untuk mengambil input.

## Software Interfaces

Aplikasi ini tidak terhubung dengan aplikasi apapun.

## Communications Interfaces

Aplikasi ini tidak memerlukan akses ke internet dalam penggunaannya.

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

Keamanan data pengguna sangat teramin, dikarenakan aplikasi tidak akan meminta data pribadi dari pengguna dan juga tidak akan meminta izin akses ke file pengguna.

## Security Requirements

Keamanan data pengguna sangat terjamin pada aplikasi ini, dikarenakan tidak akan ada history pengguna dalam aplikasi ini, dan aplikasi ini tidak akan meminta data pribadi dari pengguna. Data yang diisi oleh pengguna diawal tidak akan disimpan, oleh karena itu setiap masuk aplikasi, pengguna diharuskan untuk mengisi ulang data.

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

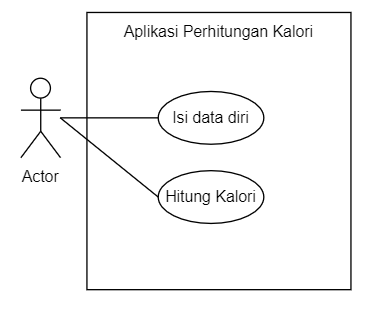
# Other Requirements

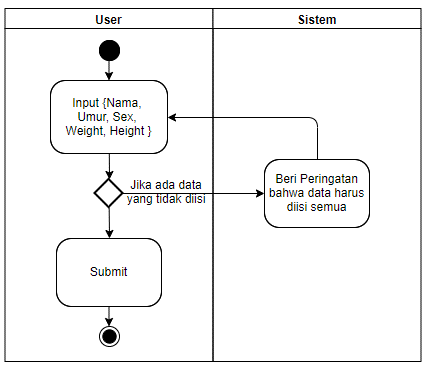
<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

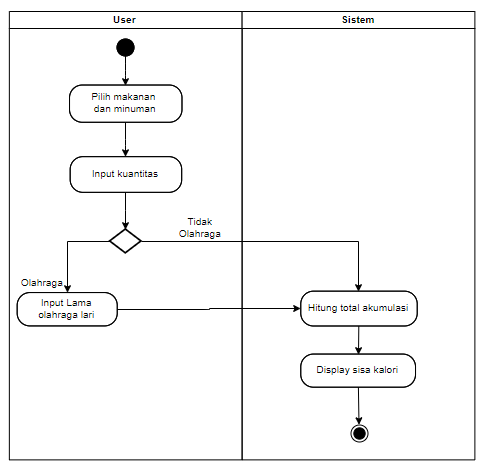
<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

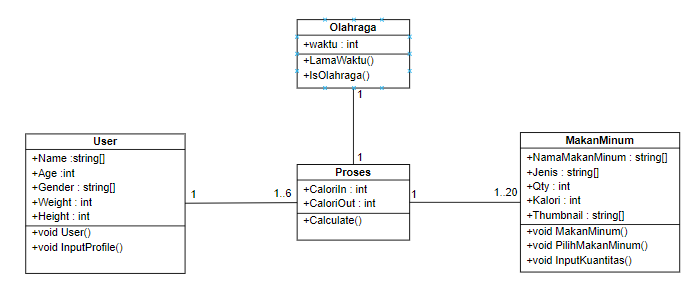
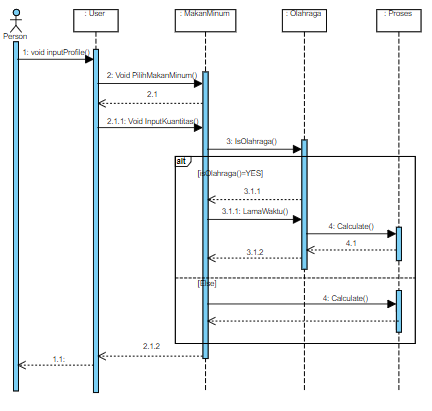
Appendix B: Analysis Models

* Use Case Diagram
* Activity Diagram

Untuk “Isi data diri”

Untuk “Hitung Kalori”



* Class Diagram
* Sequential Diagram

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>