

SKPL-SC

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

SMART CANTEEN

untuk:

<nama pelanggan>

Dipersiapkan oleh:

Ahmed Lasca R (1301180267)

Athalla Rizky Arsyian (1301184451)

Fajar Widhi Ardiyanto (1301180333)

M. Alif Naufal Yasin (1301184321)


Novita (1301184101)

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

	Program Studi S1 Informatika - Fakultas Informatika	Nomor Dokumen		Halaman
		SKPL-xxx		< isi halaman >
		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								
Diperiks a oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

--	--	--	--

Daftar Isi

Daftar Perubahan	2
Daftar Halaman Perubahan	3
Daftar Isi	4
1. Pendahuluan	5
1.1. Tujuan Penulisan Dokumen	5
1.2. Ruang Lingkup / Cakupan Dokumen	5
1.3. Definisi, Singkatan, dan Akronim	5
1.4. Referensi	5
2. Deskripsi Global Perangkat Lunak	6
2.1. Statement of Objective Perangkat Lunak	6
2.2. Perspektif dan Fungsi Perangkat Lunak	6
2.3. Profil dan Karakteristik Pengguna	6
2.4. Arsitektur Lingkungan Operasi Perangkat Lunak (digambarkan)	6
2.5. Kebutuhan Perangkat Keras	6
2.6. Asumsi dan Batasan Perangkat Lunak	7
3. Deskripsi Rinci Perangkat Lunak	8
3.1. Deskripsi Kebutuhan	8
3.1.1. Kebutuhan Fungsional	8
3.1.2. Kebutuhan Non-Fungsional	8
3.2. Pemodelan Analisis	8
3.2.1. Usecase Diagram	8
3.3. Class Diagram	9

1. Pendahuluan

1.1. Tujuan Penulisan Dokumen

Spesifikasi Kebutuhan Perangkat Lunak (SKPL) merupakan dokumentasi atau catatan yang menjelaskan secara terperinci tentang spesifikasi kebutuhan perangkat lunak yang akan dikembangkan. Dokumen ini dibuat untuk dijadikan sebagai acuan dalam proses pengembangan perangkat lunak agar sesuai dengan apa yang dibutuhkan pengguna.

1.2. Ruang Lingkup / Cakupan Dokumen

SmartCanteen merupakan aplikasi untuk mempermudah penjual di kantin dalam menemukan posisi pelanggannya, mempermudah pelanggan mencari meja kosong yang dirancang dengan *server* Node.js dengan MySQL atau Firebase sebagai *database*-nya. Adapun fitur-fitur yang menjadi fokus dalam pengembangan aplikasi ini adalah *scan barcode* dari sisi pelanggan untuk *men-generate* nomor tempat duduk dan *layouting* kantin oleh admin.

1.3. Definisi, Singkatan, dan Akronim

- *Developer*, adalah orang yang merancang dan mengembangkan perangkat lunak.
- *Project Manager*, adalah orang yang memimpin proses pengembangan perangkat lunak.
- *User*, adalah orang yang menggunakan perangkat lunak tersebut.
- *Scan*, adalah proses memindai suatu objek yang bertujuan mengidentifikasi suatu data atau informasi tertentu,
- *Barcode*, adalah kode batang yang memrepresentasikan suatu data dalam bentuk visual atau gambar tertentu.

1.4. Referensi

Referensi yang digunakan untuk membuat dokumentasi perancangan perangkat lunak ini antara lain:

- Bayu Hendradjaya. Panduan Penulisan Kebutuhan Perangkat Lunak (SKPL). Jurusan Teknik Informatika ITB.
- *IEEE Guide for Software Requirements Specifications.*
- *IEEE Recommended Practice for Software Requirements Specifications.*
- Panduan Penggunaan dan Pengisian Spesifikasi Perangkat Lunak (SKPL), Jurusan Informatika, Universitas Telkom.

Prodi S1 Teknik Informatika - Universitas Telkom	SKPL-xxx	Halaman 6 dari 19
Dokumen ini dan informasi yang ada di dalamnya adalah milik Prodi S1 Teknik Informatika-Universitas Telkom dan bersifat rahasia. Dilarang untuk mereproduksi dokumen ini tanpa diketahui oleh Program Studi S1 Teknik Informatika, Universitas Telkom		

- Panduan Pengisian Spesifikasi Perangkat Lunak (SKPL) Berorientasi Proses, Jurusan Informatika, Universitas Telkom.
- *Software Requirements Specification* – IBM
- STAF IF. GL01, *Spesifikasi Kebutuhan Perangkat Lunak*. Jurusan Teknik Informatika ITB.

2. Deskripsi Global Perangkat Lunak

2.1. Statement of Objective Perangkat Lunak

SmartCanteen merupakan aplikasi yang mempermudah para pengguna yang terlibat dalam sistem transaksi di kantin universitas atau sekolah. Dengan adanya aplikasi ini, diharapkan agar para pelanggan lebih mudah dalam mencari meja kosong untuk ditempati dan tidak harus mengantri dalam memesan makanan. Penjual pun akan dimudahkan dalam hal mencari lokasi meja pemesan dan manajemen data transaksi.

2.2. Perspektif dan Fungsi Perangkat Lunak

Sistem ini melibatkan beberapa entitas, seperti pelanggan, penjual, dan admin. Kemudian, ada fitur *history* untuk melihat riwayat transaksi yang sudah pernah dilakukan.

- Fungsi *Scan Barcode* (FR-01)

Fungsi dari sisi **pelanggan** yang memungkinkan pelanggan menempati meja kosong dan mempermudah penjual menemukan lokasinya.

- Fungsi Lihat Menu (FR-02)

Fungsi dari sisi **pelanggan** dan **admin** sehingga pelanggan dapat melihat menu makanan yang tersedia di kantin tersebut.

- Fungsi Pesan Makanan (FR-03)

Fungsi dari sisi **pelanggan** untuk memilih makanan yang ingin dipesan dan mengonfirmasi pesanan.

- Fungsi Pembayaran (FR-04)

Fungsi dari sisi **pelanggan** untuk melakukan pembayaran via *financial technology*, seperti OVO, GoPay, LinkAja, dll.

- Fungsi Input Menu Makanan (FR-05)

Fungsi dari sisi **penjual** untuk meng-*input* menu makanan.

- Fungsi Menerima Pesanan (FR-06)

Fungsi dari sisi **penjual** untuk menerima maupun menolak pesanan pelanggan (dengan alasan tertentu).

- Fungsi Lihat Riwayat Transaksi (FR-07)

Fungsi dari sisi **pembeli**, **penjual** dan **admin** untuk melihat riwayat transaksi yang terjadi.

- Fungsi *Layouting* (FR-08)

Fungsi dari sisi **admin** untuk meng-*input* posisi tempat duduk secara nyata dari sebuah kantin.

- Fungsi Tambah Penjual (FR-09)

Fungsi dari sisi **admin** untuk menambah gerai yang akan terlibat dalam sistem.

- Fungsi *Login* (FR-10)
- Fungsi *Register* (FR-11)

2.3. Profil dan Karakteristik Pengguna

gagan	san makanan dan melakukan pembayaran	
l	dia makanan dan mengonfirmasi pesanan	
	najemen segala proses interaksi, mengatur tata letak kantin, menambahkan data gerai makanan	

2.4. Lingkungan Operasi

Perangkat lunak dari sisi *server* yang dibutuhkan **SmartCanteen**, antara lain:

- Sistem operasi: Microsoft Windows 8/8.1/10, Linux, MacOS
- Web server: Niagahoster, Heroku, Netlify
- *Scripting language*: HTML, CSS, Javascript
- *Database*: MySQL, Firebase

Perangkat lunak dari sisi *client* yang dibutuhkan **SmartCanteen**, antara lain:

- Sistem operasi: MacOS, Solaris, Linux, Android
- *Web browser*: Microsoft Edge, Opera, Safari, Google Chrome, Mozilla Firefox.

2.5. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk mengakses **SmartCanteen**, antara lain:

- *Smartphone*

Digunakan pembeli untuk mengakses *website*.

Spesifikasi: *smartphone* yang mendukung akses internet dan pengambilan gambar menggunakan kamera.

- Komputer/laptop

Digunakan penjual dan admin untuk mengakses dasbor aplikasi.

Spesifikasi: komputer/laptop yang mendukung akses internet dan *web browser*.

- *Server* dan basis data

Digunakan untuk menyimpan data-data pendukung aplikasi dan segala proses transaksi yang terjadi di dalamnya.

Spesifikasi: *server* yang mendukung akses internet dan siaga selama aplikasi sedang digunakan oleh pengguna.

2.6. Asumsi dan Batasan Perangkat Lunak

Dalam proses interaksi dengan aplikasi dibutuhkan akses internet yang stabil karena sistem saling terintegrasi satu dengan yang lainnya, baik dari sisi pelanggan, penjual, maupun admin. Selain itu, penggunaan aplikasi ini hanya terbatas di kantin tertentu yang sudah meng-*apply* sistem aplikasi **SmartCanteen**.

3. Deskripsi Rinci Perangkat Lunak

3.1. Deskripsi Kebutuhan

3.1.1. Kebutuhan Fungsional

No.	Kode Kebutuhan	Fungsi	Deskripsi
1.	FR-01	<i>Scan Barcode</i>	Nomor meja akan ter- <i>generate</i> secara otomatis dan masuk ke sistem.
2.	FR-02	Lihat Menu	Pelanggan dan admin dapat melihat menu makanan berdasarkan gerai yang dipilihnya.
3.	FR-03	Pesan Makanan	Pelanggan memesan makanan melalui aplikasi.
4.	FR-04	Pembayaran	Dilakukan oleh pelanggan melalui <i>payment gateway</i> .

5.	FR-05	Input Menu Makanan	Penjual meng- <i>input</i> menu makanan yang tersedia di gerainya.
6.	FR-06	Menerima Pesanan	Penjual dapat menerima maupun menolak pesanan masuk.
7.	FR-07	Lihat Riwayat Transaksi	Penjual dan pembeli dapat melihat data transaksi terbatas pada transaksi yang dilakukannya, sedangkan admin dapat melihat data transaksi secara keseluruhan yang terjadi di kantin.
8.	FR-08	<i>Layouting</i>	Admin meng- <i>input</i> tata letak nyata meja kantin.
9.	FR-09	Tambah Penjual	Admin menambahkan gerai ke dalam sistem.
10.	FR-10	<i>Login</i>	Hanya bisa dilakukan oleh pelanggan.
11.	FR-11	<i>Register</i>	Hanya bisa dilakukan oleh pelanggan.

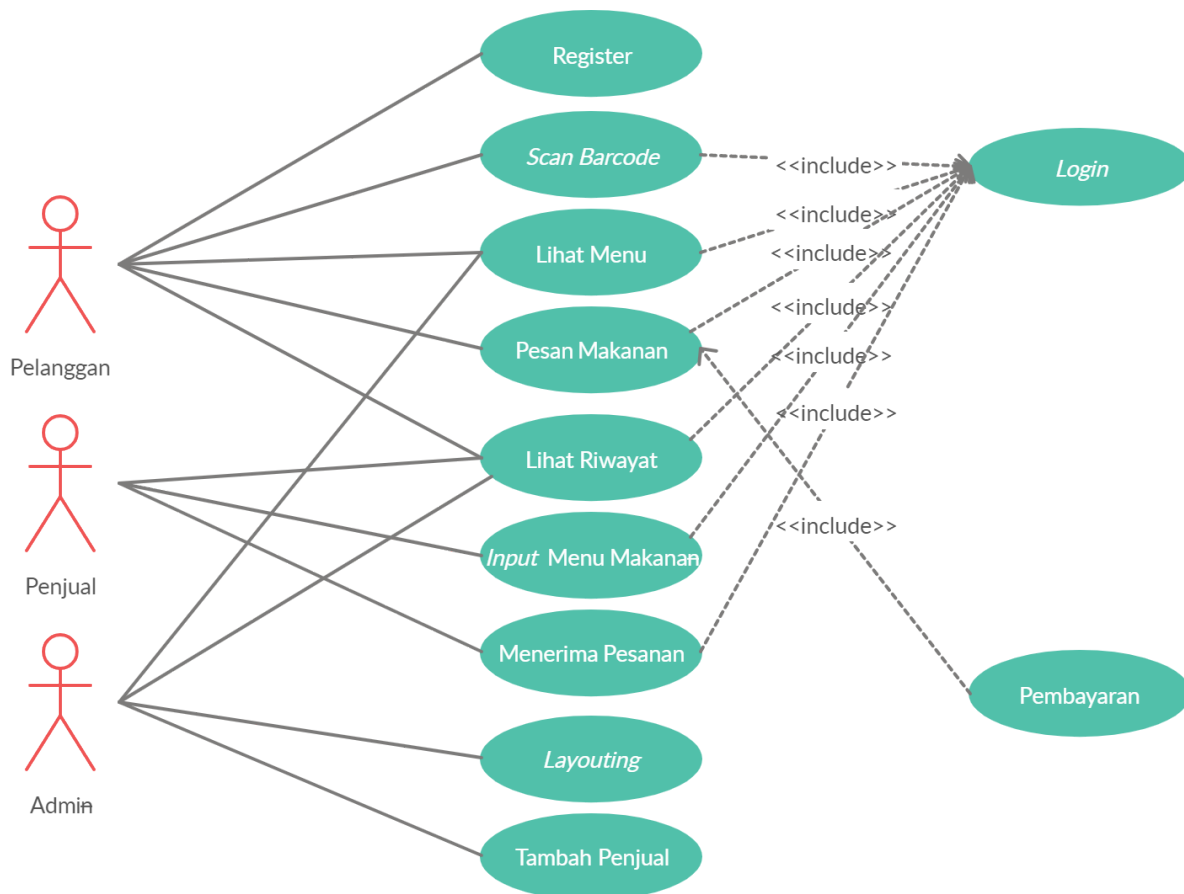
3.1.2. Kebutuhan Non-Fungsional

No.	Quality	Kode Kebutuhan	Deskripsi
1	<i>Portability</i>	NFR-01	Sebuah aplikasi berbasis <i>website</i> .

2	Usability	NFR-02	Memiliki tampilan antarmuka yang <i>user-friendly</i> .
3	Security	NFR-03	Setiap <i>user</i> memiliki akun masing-masing yang hanya dapat diakses oleh mereka sendiri.
4	Accessibility	NFR-04	Terbatas pada kantin yang sudah terintegrasi dengan sistem SmartCanteen .
5	Error-handling	NFR-05	Terdapat <i>error message</i> jika <i>input</i> -an tidak sesuai dengan sistem.

3.2 Pemodelan Analisis

3.2.1. Usecase Diagram



3.2.1.1. Usecase Skenario #1

Nama Use Case	<i>Scan Barcode</i>	
Deskripsi	Data-data pembeli mengenai nomor meja dan transaksi akan tersimpan dalam <i>database</i> .	
Pre-Kondisi	Pembeli mengunjungi kantin dan duduk di meja tertentu yang sudah diberi penomoran.	
Post-Kondisi	<i>Scan barcode</i> membawa pembeli ke suatu <i>website</i> dan pembeli akan mengisi data nama, memilih menu makanan, dan melakukan pembayaran.	
Skenario Utama	Aktor	Sistem
	1. <i>Scan barcode</i>	
		2. Menampilkan <i>website</i>
	3. Isi data	
	4. Menekan tombol <i>submit</i>	

3.2.1.2. Usecase Skenario #2

Nama Use Case	Lihat Menu Makanan	
Deskripsi	Menu makanan yang tersimpan di <i>database</i> akan ditampilkan kepada pelanggan.	
Pre-Kondisi	Pelanggan sudah melakukan <i>login</i> .	
Post-Kondisi	Pembeli telah menentukan makanan yang diinginkan dan siap memesan makanan tersebut.	
Skenario Utama	Aktor	Sistem
	1. Pilih menu “Membuat pesanan”.	
		2. Menampilkan daftar gerai di kantin.
	3. Memilih gerai yang diinginkan.	
		4. Menampilkan daftar makanan di gerai yang dipilih.

3.2.1.3. Usecase Skenario #3

Nama Use Case	Pesan Makanan
Deskripsi	Makanan dipesan dan akan disajikan.
Pre-Kondisi	Pembeli sudah memilih makanan yang diinginkan.

Post-Kondisi	Pembeli melakukan pembayaran.	
Skenario Utama	Aktor	Sistem
	1. Memilih makanan yang ingin dipesan.	
		2. Memasukkan daftar makanan ke <i>cart</i> .
	3. Klik “Pesan”.	
		4. Menampilkan menu pembayaran.

3.2.1.4. Usecase Skenario #4

Nama Use Case	Pembayaran	
Deskripsi	Pembeli melakukan pembayaran dan akan masuk ke <i>database</i> .	
Pre-Kondisi	Pembeli sudah memesan makanan melalui sistem.	
Post-Kondisi	Makanan dimasak dan akan disajikan ke pelanggan.	
Skenario Utama	Aktor	Sistem
	1. Setelah <i>checkout</i> , pilih tombol “Lakukan pembayaran”.	
		2. Menampilkan jendela pemilihan metode pembayaran.
	3. Pilih metode pembayaran.	
	4. Tekan tombol “Bayar”.	

3.2.1.5. Usecase Skenario #5

Nama Use Case	Lihat Riwayat Transaksi	
Deskripsi	Menampilkan hasil transaksi.	
Pre-Kondisi	Penjual atau pembeli sudah melakukan <i>login</i> .	
Post-Kondisi	Riwayat transaksi ditampilkan.	
Skenario Utama	Aktor	Sistem
	1. Pilih menu “Riwayat transaksi”.	
		2. Menampilkan jendela yang berisi data transaksi.

3.2.1.6. Usecase Skenario #6

Nama Use Case	Input Menu Makanan	
Deskripsi	Menu makanan yang di- <i>input</i> akan masuk ke <i>database</i> .	
Pre-Kondisi	Daftar makanan belum ter- <i>input</i> ke sistem.	
Post-Kondisi	Menu makanan sudah ter- <i>input</i> dan dapat dilihat oleh pembeli.	
Skenario Utama	Aktor	Sistem
	1. Menekan tombol “Input menu makanan”.	
		2. Menampilkan jendela input menu makanan.
	3. Meng- <i>input</i> nama makanan, harga, dan deskripsi (opsional).	
	4. Menekan tombol “Simpan”.	

3.2.1.7. Usecase Skenario #7

Nama Use Case	Menerima Pesanan	
Deskripsi	Pesanan akan dimasukkan ke <i>database</i> , baik diterima maupun tidak.	
Pre-Kondisi	Pelanggan sudah menekan tombol “Pesan”.	
Post-Kondisi	Transaksi dimasukkan ke <i>database</i> (baik yang diterima maupun yang ditolak).	
Skenario Utama	Aktor	Sistem
		1. Memberikan notifikasi ketika ada pesanan masuk.
		2. Menampilkan jendela pesanan masuk.
	3. Menekan tombol “Terima pesanan” atau “Menolak pesanan”.	
		4. Menampilkan status pemesanan ke pelanggan.

3.2.1.8. Usecase Skenario #8

Nama Use Case	<i>Layouting</i>
----------------------	------------------

Deskripsi	Admin meng- <i>input layout</i> meja kantin, yang akan tersimpan ke dalam <i>database</i> .	
Pre-Kondisi	Belum ada <i>layout</i> kantin di dalam sistem	
Post-Kondisi	Sudah terdapat <i>layout</i> kantin di sistem	
Skenario Utama	Aktor	Sistem
	1. Memilih menu “ <i>Layouting</i> ”.	
		2. Menampilkan jendela untuk meng- <i>input layout</i> kantin.
	3. Meng- <i>input layout</i> sesuai keadaan kantin.	
	4. Menekan tombol “ <i>Simpan</i> ”.	
		5. <i>Merender layout</i> dan menyimpan ke <i>database</i> .

3.2.1.9. Usecase Skenario #9

Nama Use Case	Tambah Penjual	
Deskripsi	Penjual terdaftar akan dimasukkan ke <i>database</i> .	
Pre-Kondisi	Penjual belum terdaftar.	
Post-Kondisi	Penjual sudah terdaftar dan bisa melakukan <i>login</i> .	
Skenario Utama	Aktor	Sistem
	1. Pilih menu “Tambah penjual”.	
		2. Menampilkan <i>form</i> data penjual.
	3. Memasukkan data penjual.	
	4. Menekan tombol “Daftar”.	

3.2.1.10. Usecase Skenario #10

Nama Use Case	<i>Login</i>	
Deskripsi	Masuk ke dalam sistem berdasarkan akun terdaftar.	
Pre-Kondisi	Pembeli belum melakukan <i>login</i> .	
Post-Kondisi	Pembeli dapat memilih fitur yang diinginkan.	
Skenario Utama	Aktor	Sistem

	1. Pilih menu “Login”.	
		2. Menampilkan jendela yang berisi <i>text field</i> untuk memasukkan <i>username</i> dan <i>password</i> .
	3. Memasukkan <i>username</i> dan <i>password</i> .	
	4. Menekan tombol “Login”.	

3.2.1.11. Usecase Skenario #11

Nama Use Case	<i>Register</i>	
Deskripsi	Pelanggan harus <i>register</i> untuk bisa <i>login</i> , data <i>register</i> akan masuk ke <i>database</i> .	
Pre-Kondisi	Pelanggan belum memiliki akun.	
Post-Kondisi	Pelanggan telah memiliki akun.	
Skenario Utama	Aktor	Sistem
	1. Pilih menu “Register”.	
		2. Menampilkan jendela <i>form</i> yang akan diisi oleh pelanggan.
	3. Memasukkan data berdasarkan <i>form</i> yang ditampilkan sistem.	
	4. Menekan tombol “Register”.	

3.3 Class Diagram

