*Software Requirements Specification*

for

<POS Order Bakso Pentol>

Version 1.0 approved

Prepared by

<Nim 1 - Nama>

<Nim 2 - Nama>

<Nim 3 - Nama>

<Nim 4 - Nama>

<date created>

Table of Contents

1. Pendahuluan 1

1.1 Tujuan Penulisan Dokumen 1

1.2 Audien yang Dituju dan Pembaca yang Disarankan 1

1.3 Batasan Produk 1

1.4 Definisi dan Istilah 1

1.5 Refrensi 1

2. Deskripsi Keseluruhan 2

2.1 Deskripsi Produk 2

2.2 Fungsi Produk 2

2.3 Penggolongan Karakterik Pengguna 2

2.4 Lingkungan Operasi 2

2.5 Batasan Desain dan Implementasi 2

2.6 Dokumentasi Pengguna 3

3. Kebutuhan Antarmuka Eksternal 3

3.1 User Interfaces 3

3.2 Hardware Interface 3

3.3 Software Interface 3

3.4 Communication Interface 3

4. Functional Requirement 4

4.1 Use Case Diagram 4

4.2 Nama Use Case 1 **Error! Bookmark not defined.**

4.3 Nama Use Case 2 **Error! Bookmark not defined.**

4.4 Class Diagram 8

5. Non Functional Requirements 10

Revision History

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

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini bertujuan untuk merinci kebutuhan perangkat lunak aplikasi POS Order Bakso berbasis mobile, yang dikembangkan menggunakan Flutter dan Firebase. Dokumen ini menjadi acuan utama dalam proses pengembangan, pengujian, dan pemeliharaan aplikasi.

## Audien yang Dituju dan Pembaca yang Disarankan

Dokumen ini ditujukan untuk:

* Pengembang aplikasi
* Manajer proyek
* Penguji perangkat lunak
* Pengguna akhir (admin dan user)
* Stakeholder terkait lainnya

## Batasan Produk

Aplikasi POS Order Bakso adalah aplikasi mobile untuk memudahkan proses pemesanan bakso oleh admin / cashier. Admin memiliki hak akses untuk mengelola data produk dan pesanan.

## Definisi dan Istilah

* **POS** : Point of Sale, Sistem kasir digital untuk transaksi penjualan.
* **Admin** : Pengguna dengan hak akses penuh, dapat mengelola produk, pesanan, user dan juga melakukan pemesanan.
* **Flutter** : Framwork open-source untuk pengembangan aplikasi mobile.
* **Firebase** : Platform Backend-as-a-service (BaaS) yang digunakan untuk autentikasi, database, dan penyimpanan data aplikasi.
* SRS : *Software Requirements Specification*, atau

Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

* IEEE : *Institute of Electrical and Electronics Engineering*

Standar internasional untuk pengembangan dan perancangan produk.

## Refrensi

* IEEE SRS Standard.
* Dokumen desain UI/UX aplikasi.
* Panduan integrasi pembayaran digital.
* Dokumentasi Flutter.
* Dokumentasi Firebase.

# Deskripsi Keseluruhan

## Deskripsi Produk

Aplikasi POS Order Bakso adalah aplikasi mobile yang memfasislitasi proses pemesanan bakso secara digital, dengan fitur untuk mengelola data produk dan pesanan. Semua data tersimpan dan tersinkronisasi secara real-time menggunakan Firebase.

## Fungsi Produk

* Registrasi dan login admin.
* Pemesanan bakso (order).
* Manajemen pesanan (lihat, update status).
* Riwayat pesanan.
* Logout.

## Penggolongan Karakterik Pengguna

* Mencatat pesanan pelanggan.
* Memproses pembayaran (tunai/digital).
* Mengelola menu bakso pentol.
* Menampilkan laporan penjualan harian/bulanan.
* Mencetak struk transaksi.

Tabel 1 Karakteristik Pengguna

| **Kategori Pengguna** | **Tugas** | **Hak Akses ke aplikasi** | **Kemampuan yang harus dimiliki** |
| --- | --- | --- | --- |
| Admin | Mengelola pesanan, menu, kategori, stok dan laporan. | Insert, Update, Delete Data | Manupulasi Data Transaksi Penjualan dan menu |

## Lingkungan Operasi

* Platform : Mobile (android/ios).
* Framework : Flutter.
* Backend : Firebase (Authentication, Firestore Database, Cloud Messaging).
* Perangkat : Smartphone, tablet, laptop/pc.

## Batasan Desain dan Implementasi

* Bahasa pemrograman : Dart (Flutter).
* Backend menggunakan Firebase.
* Harus mendukung autentikasi.
* Tidak mendukung mode offline.
* Standar keaman Firebase (Firestore rules, Auth).

## Dokumentasi Pengguna

* User manual (panduan pengguna).
* Online help (FAQ di aplikasi).
* Video tutorial (opsional).

# Kebutuhan Antarmuka Eksternal

## User Interfaces

* Tampilan login/register.
* Dashboard utama (list produk).
* Halaman pemesanan (order form).
* Riwayat pesanan.
* Halaman kelola produk, pesanan, user.

## Hardware Interface

* Smartphone Android/iOS.
* Koneksi internet.

## Software Interface

* Flutter SDK.
* Firebase SDK (Auth, Firestore, Cloud Messaging).

## Communication Interface

* Protokol komunikasi HTTPS (REST API Firebase).
* Notifikasi push via Firebase Cloud Messaging.

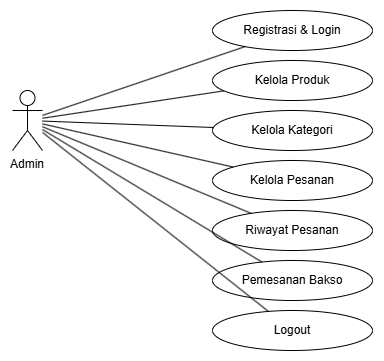
# Functional Requirement

| **ID** | **Kebutuhan Fungsional** | **Penjelasan** |
| --- | --- | --- |
| FR1 | Registrasi & Login | Admin dapat mendaftar dan login. |
| FR2 | Pemesanan Bakso | Admin dapat membuat order. |
| FR3 | Kelola Kategori | Admin dapat tambah, edit, hapus kategori. |
| FR4 | Kelola Produk | Admin dapat tambah, edit, hapus produk. |
| FR5 | Kelola Pesanan | Admin dapat melihat dan update status pesanan. |
| FR6 | Riwayat Pesanan | Admin dapat melihat riwayat pesanan. |
| FR7 | Logout | User & admin dapat logout. |

## Use Case Diagram

Gambaran use case diagram untuk sistem **POS Order Bakso Pentol** ditampilkan seperti di bawah ini. Diagram ini menunjukkan interaksi antara **aktor** (Admin) dengan **fitur-fitur sistem** berdasarkan kebutuhan fungsional:

* Admin dapat melakukan proses **Registrasi & Login**
* Admin dapat **mengelola produk, kategori, dan pesanan**
* Admin dapat melihat **riwayat pesanan**
* Admin dapat melakukan **pemesanan bakso**
* Admin dapat melakukan **logout** dari sistem

**Diagram lengkap ditunjukkan pada gambar berikut:

*(Gambar use case)*

## Registrasi & Login

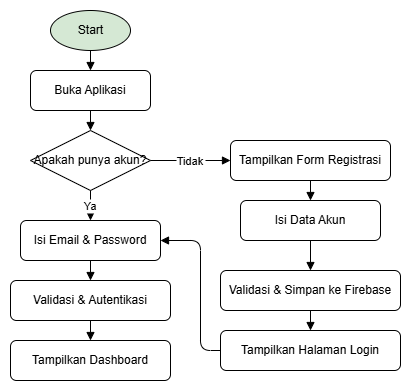
4.2.1 Deskripsi Use Case

Use case ini menjelaskan proses saat Admin melakukan registrasi akun baru atau login ke dalam sistem untuk mendapatkan akses penuh terhadap fitur-fitur aplikasi POS Bakso Pentol.

4.2.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Admin membuka aplikasi | 1. Sistem menampilkan halaman login |
| 2. Admin memilih "Register" (jika belum punya akun) | 2. Sistem menampilkan form registrasi |
| 3. Admin mengisi dan mengirim data | 3. Sistem memvalidasi dan menyimpan data akun |
| |  | | --- | | 4. Admin login dengan email & password |  |  | | --- | |  | | |  | | --- | | 4. Sistem memverifikasi dan mengarahkan ke dashboard | |

4.2.4 Activity Diagram



*(activity diagram login & register)*

## Kelola Produk

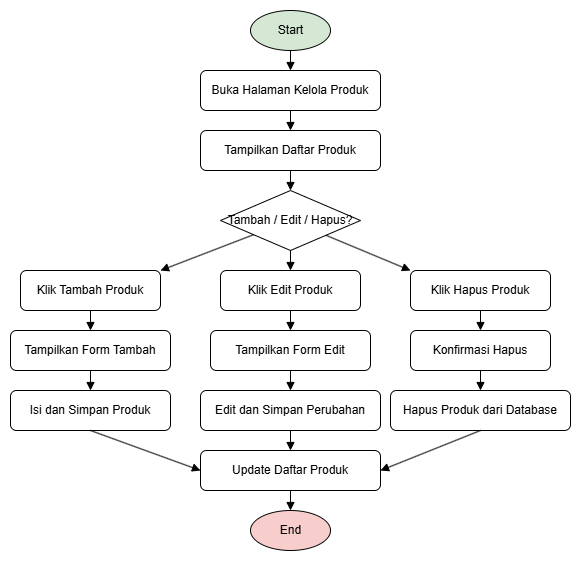
4.3.1 Deskripsi Use Case

Use case ini menggambarkan proses ketika admin mengelola daftar produk bakso yang tersedia di aplikasi. Admin dapat menambahkan, mengubah, dan menghapus data produk sesuai kebutuhan operasional warung bakso.

Fitur ini penting untuk menjaga informasi menu tetap relevan, lengkap, dan terbarui, seperti nama produk, harga, kategori, dan stok.

4.3.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Admin membuka halaman kelola produk | 1. Sistem menampilkan daftar produk yang sudah tersedia |
| 2. Admin menekan tombol "Tambah Produk" | 2. Sistem menampilkan form tambah produk baru |
| 3. Admin mengisi form dan menyimpan | 3. Sistem memvalidasi dan menyimpan data ke database |
| |  | | --- | | 4. Admin mengedit produk |  |  | | --- | |  | | |  | | --- | | 4. Sistem menampilkan form edit dengan data terisi | |
| 5. Admin menyimpan perubahan | 5. Sistem memperbarui data produk |
| 6. Admin menghapus produk tertentu | 6. Sistem menghapus data dari database dan update tampilan |

4.3.4 Activity Diagram

*(activity diagram kelola produk)*

## Kelola Kategori

4.4.1 Deskripsi Use Case

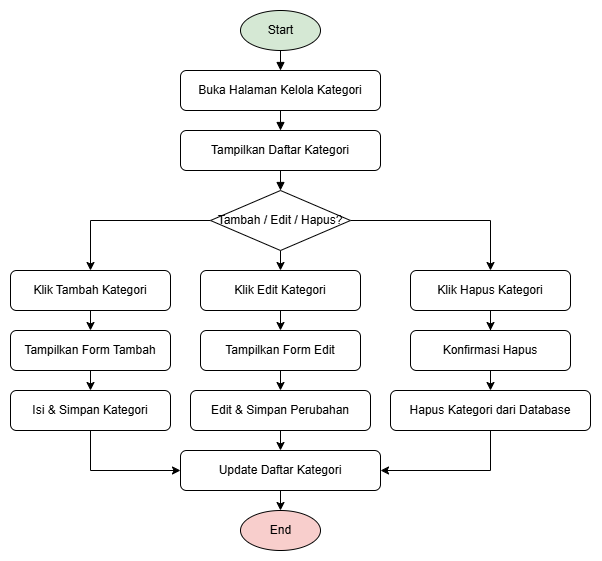
Use case ini menjelaskan proses saat admin mengelola daftar **kategori produk bakso** seperti Bakso Pentol, Bakso Tahu, Bakso Pangsit, dll. Admin dapat menambahkan kategori baru, mengedit nama kategori yang ada, dan menghapus kategori yang tidak digunakan.

Fitur ini mendukung pengelompokan produk agar lebih mudah ditemukan dan dikelola, terutama jika jumlah produk sangat banyak.

4.4.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Admin membuka halaman kelola kategori | 1. Sistem menampilkan daftar kategori yang ada |
| 2. Admin menekan tombol "Tambah Kategori" | 2. Sistem menampilkan form tambah kategori |
| 3. Admin mengisi dan menyimpan | 3. Sistem memvalidasi & menyimpan kategori baru |
| |  | | --- | | 4. Admin mengedit kategori |  |  | | --- | |  | | |  | | --- | | 4. Sistem menampilkan form edit dengan data yang diubah | |
| 5. Admin menyimpan perubahan | 5. Sistem memperbarui kategori di database |
| 6. Admin menghapus kategori | 6. Sistem meminta konfirmasi, lalu menghapus dan memperbarui daftar |

4.4.4 Activity Diagram

*(activity diagram kelola kategori)*

## Kelola Pesanan

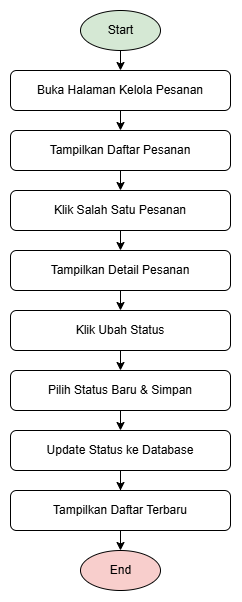
4.5.1 Deskripsi Use Case

Use case ini menggambarkan proses ketika Admin mengelola daftar pesanan bakso yang masuk. Admin dapat melihat semua pesanan yang dibuat sebelumnya, memperbarui status pesanan (misalnya: diproses, selesai), dan melakukan tindakan bila diperlukan terhadap pesanan tersebut.

Fitur ini berperan penting dalam memastikan alur pemrosesan transaksi berjalan lancar dan pesanan pelanggan dilayani tepat waktu.

4.5.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Admin membuka halaman kelola pesanan | 1. Sistem menampilkan daftar semua pesanan |
| 2. Admin memilih salah satu pesanan | 2. Sistem menampilkan detail pesanan |
| 3. Admin menekan tombol "Ubah Status" | 3. Sistem menampilkan form perubahan status |
| |  | | --- | | 4. Admin memilih status baru |  |  | | --- | |  | | |  | | --- | | 4. Sistem menyimpan perubahan dan menampilkan notifikasi | |
| 5. Admin kembali ke daftar pesanan | 5. Sistem menampilkan daftar yang telah diperbarui |

4.5.4 Activity Diagram

*(activity diagram kelola pesanan)*

## Riwayat Pesanan

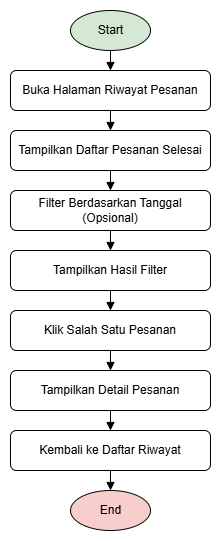
4.6.1 Deskripsi Use Case

Use case ini menjelaskan proses ketika admin ingin melihat daftar pesanan yang telah selesai atau memiliki status final (selesai, dibatalkan, dsb). Riwayat ini dapat membantu untuk memantau transaksi sebelumnya, mengevaluasi kinerja penjualan, atau mencetak laporan bulanan.

Fitur ini umumnya bersifat hanya read-only tanpa interaksi edit atau hapus.

4.6.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Admin membuka halaman riwayat pesanan | 1. Sistem menampilkan semua pesanan yang telah selesai |
| 2. Admin memilih rentang waktu tertentu | 2. Sistem memfilter dan menampilkan pesanan sesuai rentang |
| 3. Admin mengklik salah satu pesanan | 3. Sistem menampilkan detail pesanan yang dipilih |
| |  | | --- | | 4. Admin kembali ke daftar |  |  | | --- | |  | | |  | | --- | | 4. Sistem tetap menampilkan daftar tanpa reload | |

4.6.4 Activity Diagram

*(activity diagram riwayat pesanan)*

## Pemesanan Bakso

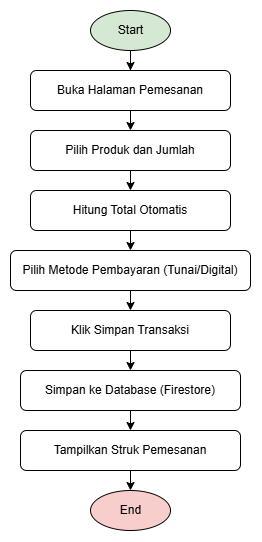
4.7.1 Deskripsi Use Case

Use case ini menjelaskan proses ketika admin melakukan transaksi **pemesanan bakso** melalui aplikasi. Admin dapat memilih produk, menentukan jumlah, menghitung total harga, memilih metode pembayaran (tunai/digital), dan menyimpan transaksi.

Fitur ini menjadi inti dari sistem POS dan harus berjalan cepat dan akurat untuk kebutuhan operasional.

4.7.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Admin membuka halaman pemesanan | 1. Sistem menampilkan form pemesanan dan daftar produk |
| 2. Admin memilih produk dan jumlah | 2. Sistem menghitung subtotal dan total harga |
| 3. Admin memilih metode pembayaran | 3. Sistem menampilkan metode tunai / digital |
| |  | | --- | | 4. Admin menyimpan pesanan |  |  | | --- | |  | | |  | | --- | | 4. Sistem menyimpan ke database dan menampilkan notifikasi | |
| 5. Admin mencetak atau melihat struk | 5. Sistem menampilkan struk atau ringkasan transaksi |

4.7.4 Activity Diagram

*(activity diagram pemesanan bakso)*

## Logout

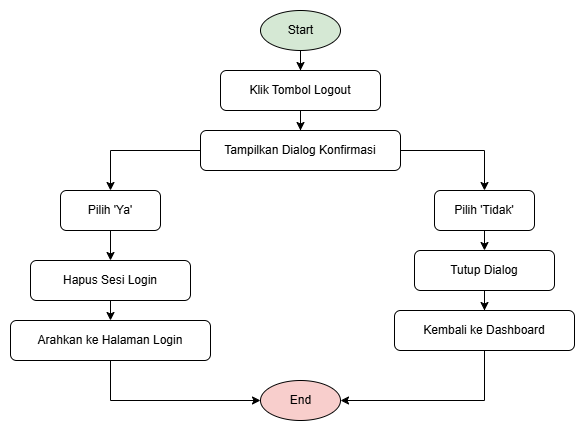
4.8.1 Deskripsi Use Case

Use case ini menggambarkan proses saat **Admin keluar (logout)** dari aplikasi setelah selesai menggunakan sistem. Fitur logout bertujuan menjaga keamanan data dan memastikan tidak ada pihak lain yang mengakses sistem tanpa izin.

Meskipun prosesnya sederhana, fitur ini krusial dalam sistem berbasis autentikasi seperti POS ini.

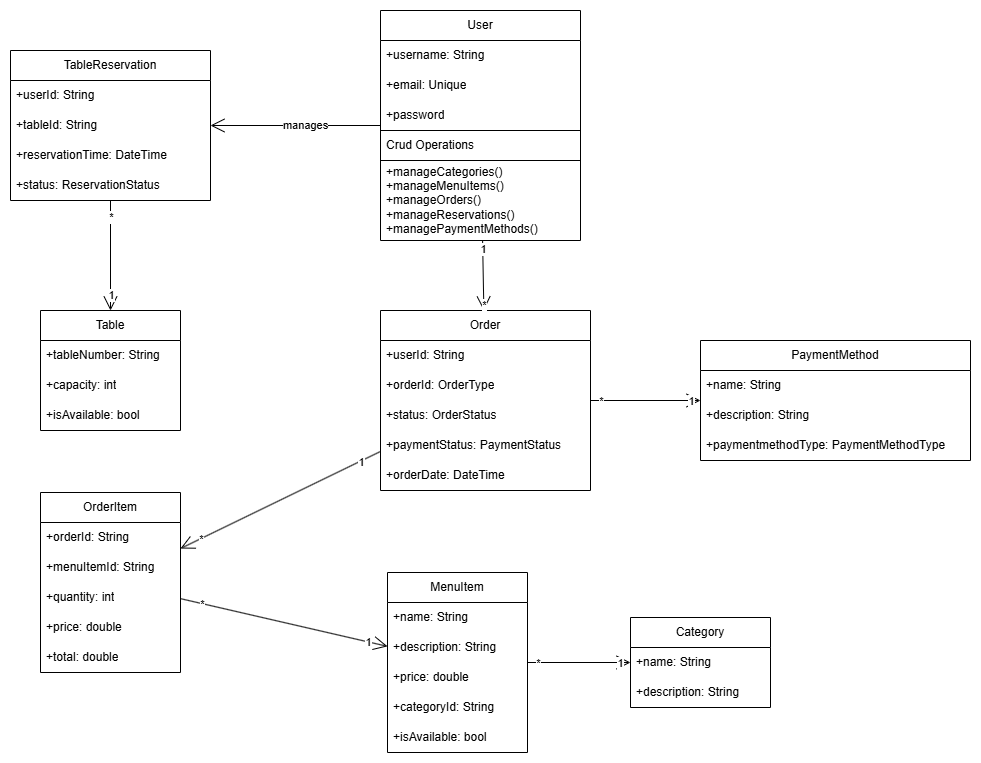
4.8.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Admin menekan tombol "Logout" | 1. Sistem menampilkan dialog konfirmasi logout |
| 2. Admin menekan "Ya" | 2. Sistem menghapus sesi login dan mengarahkan ke halaman login |
| 3. Admin menekan "Tidak" | 3. Sistem menutup dialog dan tetap berada di dashboard |

4.8.4 Activity Diagram

*(activity diagram logout)*

## Class Diagram

<*identifikasi kelas yang terkait dan hubungannya pada sistem yang dikembangkan*>

# Non Functional Requirements

| **ID** | **Kebutuhan** |
| --- | --- |
| NFR1 | Aplikasi harus tersedia 24/7 (Availability) |
| NFR2 | Waktu respon aplikasi < 3 detik untuk setiap aksi utama |
| NFR3 | Aplikasi harus mudah digunakan oleh user dan admin (Ergonomy) |
| NFR4 | Data harus terenkripsi saat transfer dan penyimpanan (Security) |
| NFR5 | Aplikasi dapat berjalan di Android & iOS (Portability) |
| NFR6 | Penggunaan memori optimal, tidak melebihi 200MB (Memory) |
| NFR7 | Notifikasi harus diterima dalam waktu < 5 detik (Response Time) |

| **ID** | **Parameter** | **Kebutuhan** |
| --- | --- | --- |
|  | Availability |  |
|  | Reliability |  |
|  | Ergonomy |  |
|  | Portability |  |
|  | Memory |  |
|  | Response time |  |
|  | Safety | N/A |
|  | Security |  |
|  |  |  |
|  | Others 1: Bahasa komunikasi | Misalnya : semua tanya jawab harus dalam bahasa Indonesia |
|  |  | Setiap layar harus mengandung logo PT Pos Indonesia |
|  |  |  |

Catatan :

*Availability : ketersediaan aplikasi, misalnya harus terus menerus beroperasi 7 hari perminggu, 24 jam per haritanpa gagal*

*Reliability : keandalan, misalnya tidak pernah boleh gagal(atau kegagalan yang ditolerir adalah …%) sehingga harus dipikirkan fault tolerant architecture. Biasanya hanya perlu untuk Critical Application yang jika gagal akan berakibat fatal.*

*Ergonomy : kenyamanan pakai bagi pengguna*

*Portability : kemudahan untuk dibawa dan dioperasikan ke mesin/sistem operasi/platform yang lain*

*Memory : jika perhitungan kapasitas memori internal kritis (misalnya untuk SW yang harus dijadikan CHIPS dan ukurannya harus kecil*

*Response time : Batasan waktu yang harus dipenuhi. Sangat penting untuk aplikasi Real Time. Contoh: “Aaplikasi harus mampu menampilkan hasil dalam 4 detik”, atau “ATM harus menarik kembali kartu yang tidak diambil dalam waktu 3 menit”*

*Safety: yang menyangkut keselamatan manusia, misalnya untuk SW yang dipakai pada sistem kontrol di pabrik*

*Security : aspek keamanan yang harus dipenuhi*