

## BAB 2 LANDASAN TEORI

Dalam bagian ini, akan dijelaskan teori-teori dasar yang menjadi dasar penelitian ini untuk memudahkan pembaca mengerti teori dasar dan terminologi yang banyak digunakan dalam penelitian ini.

### 2.1 Flutter

Flutter merupakan sebuah *framework cross-platform* yang digunakan untuk membuat aplikasi seluler dengan performa tinggi. Flutter diluncurkan pada tahun 2016 oleh Google [12]. Salah satu keunikan flutter dibandingkan dengan *framework cross-platform* lainnya adalah sistem arsitektur yang unik. Flutter menampilkan seluruh komponen yang dapat dilihat menggunakan *rendering engine* performa tinggi yang dimiliki oleh Flutter dibandingkan dengan menggunakan *web view* ataupun *OEM Widgets* seperti *framework cross-platform* lainnya. Dengan ini, Flutter memungkinkan pengembangan aplikasi dengan performa tinggi seperti aplikasi *native*. Dalam arsitektur Flutter, kode C/C++ dikompilasi dengan NDK pada Android dan LLVM pada iOS, dan kode Dart dikompilasi secara AOT (*Ahead Of Time*) [13].

### 2.2 Dart

Dart merupakan sebuah bahasa pemrograman yang didesain dengan prinsip kemudahan penggunaan, familiaritas terhadap mayoritas programmer, dan kemampuan skalabilitas. Dart adalah bahasa pemrograman yang sepenuhnya *object-oriented*. Dart dibuat untuk menyediakan layanan yang dibuat secara spesifik untuk memenuhi kebutuhan perangkat lunak dan perangkat keras di zaman sekarang [14]. Dart merupakan bahasa *Object Oriented Programming* (OOP) yang dibuat dan dikembangkan oleh Google, dan bahasa ini telah banyak digunakan dalam ekosistem perangkat lunak Google. Selain itu, Dart juga telah digunakan untuk membuat aplikasi web berskala besar [15].

### 2.3 Content-Based Filtering

Algoritma *Content Based Filtering* (CBF) merupakan salah satu algoritma rekomendasi yang paling sukses yang menggunakan korelasi antara konten sebagai dasar. CBF menggunakan informasi sebuah benda, yang direpresentasikan sebagai atribut yang akan dibandingkan dengan benda lainnya untuk mengkalkulasikan kesamaan antar benda [16]. Salah satu contoh implementasi *Content-Based Filtering* adalah sistem rekomendasi yang memberikan rekomendasi dengan membandingkan profil *user* dengan konten setiap dokumen yang ada di koleksi. Isi dalam sebuah dokumen dapat direpresentasikan dengan beberapa kata kunci yang akan merepresentasikan seorang *user* [17]. Salah satu cara untuk mengkalkulasikan kesamaan antar benda pada data tekstual adalah dengan *Cosine Similarity*.

### 2.4 Cosine Similarity

*Cosine Similarity* merupakan salah satu metrik yang banyak digunakan dalam mengukur derajat kesamaan antara dua vektor yang diukur dari sudut cosinus perkalian antara dua vektor yang dibandingkan [18]. Selain itu, metode ini berguna untuk mengukur kesamaan antara dua dokumen berdasarkan istilah yang sesuai dengan materi tersebut [19]. Berikut ini adalah rumus untuk *Cosine Similarity*:

$$\text{cosine\_similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (2.1)$$

Keterangan:

a : Vektor a

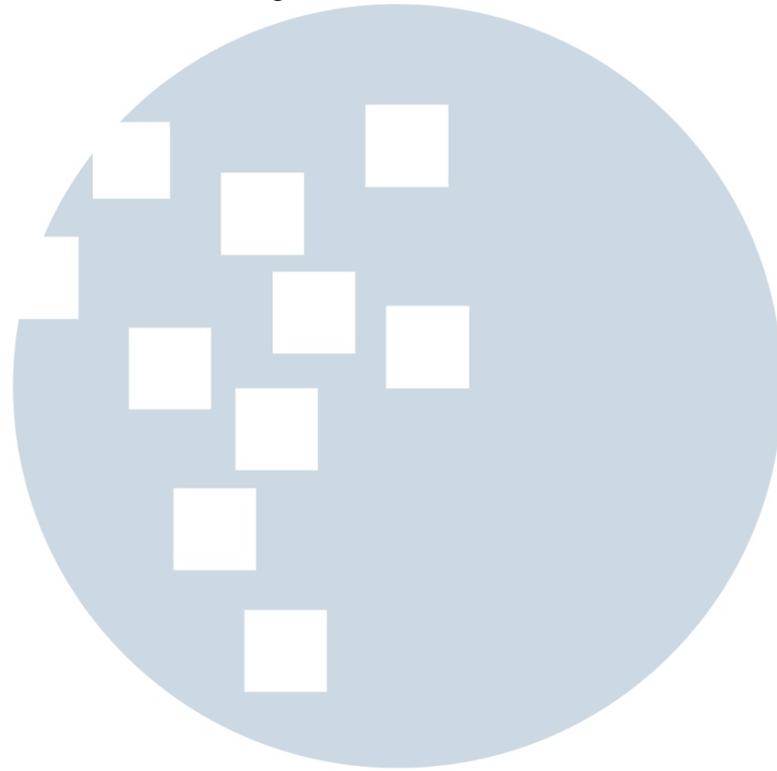
b : Vektor b

cosine\_similarity(a, b) : Nilai kemiripan antara vektor a dan b.

### 2.5 Technology Acceptance Model

Dengan meningkatnya kebutuhan teknologi di tahun 1970-an dan meningkatnya kegagalan pengadopsian sistem di berbagai organisasi, memprediksi kegunaan sistem menjadi sebuah bidang yang banyak diminati peneliti. Pada tahun 1985, Fred Davis mengusulkan *Technology Acceptance Model* (TAM) dengan basis dari *theory of reasoned action* (TRA) yang digunakan untuk menjelaskan perilaku seorang individu dalam menerima sesuatu [20]. *Technology Acceptance Model* mengusulkan bahwa kemudahan penggunaan dan kegunaan yang dirasakan

memprediksi penggunaan aplikasi yang menjadi faktor besar dalam memprediksi kegunaan sebuah sistem teknologi [21].



UMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA