

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pusaka

Dalam penelitian ini digunakan beberapa sumber pustaka sebagai acuan dan pedoman dalam membangun aplikasi. Pustaka yang digunakan ditinjau dari segi objek, *tools*, serta teknologi yang digunakan.

Tabel 1 Tinjauan Pustaka

No.	Penulis	<i>Tools</i>	Objek	Teknologi
1.	Muhammad Syahrul Anwar (2019)	Android studio (Kotlin)	Aplikasi Pembelajaran Tokoh Wayang Kulit Berbasis Android	Material Design
2.	Tintus Pramuli (2014)	Eclipse (Java)	Aplikasi Pembelajaran Ilmu Tajwid Berbasis Android	-
3.	Rifqi Darmawan (2019)	Flutter (Dart)	Implementasi <i>Push Notification</i>	-
4.	Khusnul Al Mukaromah (2017)	Android Studio (Java dan XML)	Penggunaan Material Design Untuk Pengembangan Antarmuka Aplikasi	Material Design

			Berbasis Android	
5.	Usulan	Flutter (Dart)	Aplikasi Pembelajaran Ilmu Tajwid Berbasis Android	<i>Material Design</i>

Penelitian yang dilakukan oleh Muhammad Syahrul Anwar (2019) yaitu membuat aplikasi pembelajaran tokoh wayang kulit berbasis Android. Dalam penelitian ini menggunakan android studio dengan bahasa pemrograman Kotlin, aplikasi ini mengimplementasikan material design sesuai dengan panduan/*guideline* dengan tujuan agar memiliki tampilan yang baik dari segi UI/UX.

Penelitian yang dilakukan oleh Tintus Pramuli (2014) yaitu aplikasi pembelajaran ilmu tajwid berbasis Android. Pada penelitian ini aplikasi dibangun dengan menggunakan eclipse dan bahasa pemrograman Java, penelitian ini bertujuan untuk menghasilkan aplikasi pembelajaran ilmu tajwid berbasis android dengan fitur evaluasi dengan 10 soal pilihan ganda.

Penelitian yang dilakukan oleh Rifqi Darmawan (2019) yaitu implementasi *push notification* untuk aplikasi pengingat kegiatan organisasi. Pada penelitian ini menggunakan SDK (*Software Development Kit*) Flutter dengan menggunakan bahasa pemrograman Dart. Penelitian ini membahas tentang implementasi dari *push notification* dengan menggunakan Flutter agar dapat tampil di perangkat *smartphone*.

Penelitian yang dilakukan oleh Khusnul Al Mukaromah (2017) yaitu penggunaan *material design* untuk pengembangan antarmuka aplikasi Android.

Pada penelitian ini menggunakan Android Studio dengan bahasa pemrograman Java dan XML untuk membuat aplikasi info bioskop. Tujuan dari penelitian ini adalah menganalisis kepuasan pengguna terhadap antarmuka Material Design melalui aplikasi info bioskop menggunakan model kano.

## 2.2 Material Design

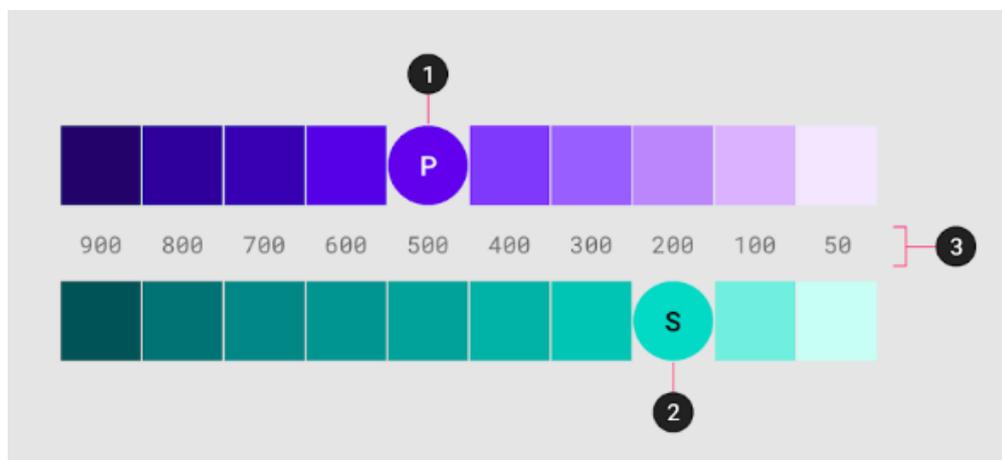
Material Design adalah bahasa visual yang dikembangkan oleh Google untuk mendesain tampilan antarmuka (*user interface*) aplikasi *mobile* atau *web* yang mensintesis prinsip-prinsip klasik dari desain yang baik dengan inovasi teknologi dan sains. Material terinspirasi oleh dunia fisik dan teksturnya, termasuk bagaimana mereka memantulkan cahaya dan membentuk bayangan. Dalam membuat tampilan antarmuka, Google memperhatikan unsur-unsur dasar dari desain cetak yang nyata mulai dari desain tipografi, grid, ruang, skala, warna, dan penggunaan visual.

Sebelum adanya Material design, para pengembang yang berfokus pada *User Interface* (UI) dan *User Experience* (UX) lebih mengutamakan bentuk tiga dimensi dimana banyak *gradient* yang memiliki kesan “Timbul” dari layar. Tetapi dalam beberapa kasus, design seperti itu sudah terlihat jadul dan usang, sehingga tidak populer lagi dan mulai ditinggalkan, baik yang menutamakan UI maupun UX. Material Design sendiri cenderung menggunakan warna-warna solid dan tanpa *gradient*, dan hanya menggunakan efek pencahayaan sederhana namun tetap terkesan simpel dan elegan. Material Design sangat membantu dalam membuat aplikasi terkait dengan UI/UX. Berikut adalah beberapa komponen dari Material Design :

### 2.2.1 Warna

Warna merupakan salah satu bagian terpenting dalam hal tampilan antarmuka karena itulah pihak Google mencoba menampilkan warna-warna yang menarik namun tidak membuat mata lelah dan enak untuk dipandang. Google menyajikan 500 warna utama dalam aplikasi dan warna lain sebagai aksentuasi.

Material Design menyajikan warna *primary color* (palet utama) dan warna sekunder. *Primary color* (palet utama) harus menjadi warna yang paling banyak digunakan di seluruh layar dan komponen. Sedangkan warna sekunder digunakan untuk menunjukkan tindakan atau informasi terkait. Warna sekunder dapat menjadi variasi gelap atau lebih terang dari warna primer, seperti yang ditunjukkan pada Gambar 2.1. (<https://material.io/design/color/the-color-system.html>)

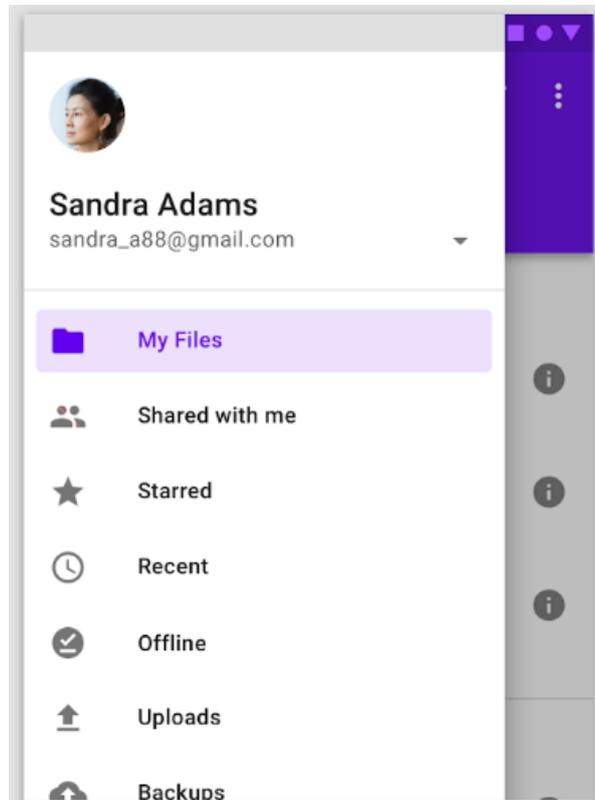


Gambar 2.1. Warna primer dan sekunder

### 2.2.2 Navigation Drawer

Navigation Drawer adalah panel yang menampilkan opsi navigasi aplikasi di tepi kiri layar. Pada saat tidak dibutuhkan, panel tersebut akan tersembunyi. Panel dapat ditampilkan saat pengguna melakukan *swipe* dari tepi kiri layar atau menyentuh ikon Navigation Drawer di App Bar, contoh dari Navigation Drawer

dapat dilihat pada Gambar 2.2. ( <https://material.io/components/navigation-drawer>)



Gambar 2.2. Navigation Drawer

### 2.2.3 Card

Elemen ini berfungsi sebagai *frame layout* untuk membungkus *layout* yang ada di dalamnya dengan tampilan seperti kartu. Card memiliki ciri khas diantaranya yaitu adanya *rounded corner* dan *shadow* untuk efek *elevation*. Elemen seperti teks dan gambar, harus ditempatkan pada mereka dengan cara yang secara jelas menunjukkan hierarki.

Card dapat diidentifikasi sebagai unit tunggal yang dapat berdiri sendiri, tanpa bergantung pada elemen sekitarnya untuk konteksnya. Card tidak dapat

digabung dengan Card lain, atau dibagi menjadi beberapa Card, penggunaan dari Card dapat dilihat pada Gambar 2.3. (<https://material.io/components/cards>)



Gambar 2.3. Card

#### 2.2.4 Tab Bar

Tab mengatur dan memungkinkan navigasi antar grup konten yang terkait dan pada tingkat hierarki yang sama. Tab dalam satu set, setiap tab harus berisi konten yang berbeda dari tab lain dalam satu set. Misalnya, tab dapat menyajikan bagian berita yang berbeda, genre musik yang berbeda, atau tema dokumen yang berbeda.

Tab bersifat *Scaleable* karena tab dapat menggulir secara horizontal, UI dapat memiliki tab sebanyak yang diperlukan. Tab mengatur konten ke dalam kategori untuk membantu pengguna dengan mudah menemukan berbagai jenis informasi, tab ditampilkan di samping satu sama lain sebagai rekan, dalam kategori yang sama pentingnya seperti yang terlihat pada Gambar 2.4.

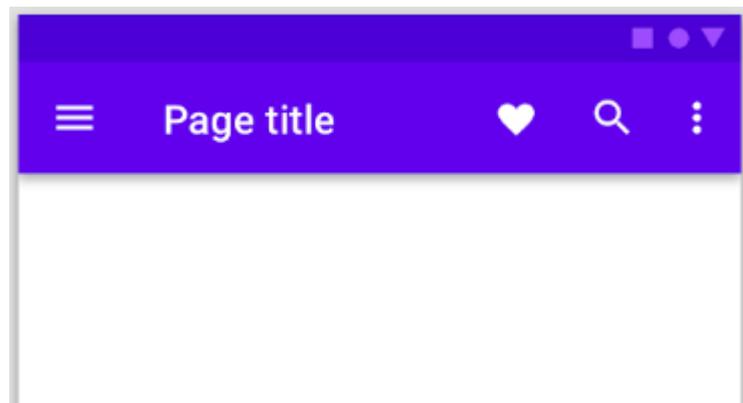
(<https://material.io/components/tabs>)



Gambar 2.4. Tab Bar

### 2.2.5 App Bar

App Bar merupakan *bar* aplikasi teratas menampilkan informasi dan tindakan yang berkaitan dengan layar saat ini. *Bar* ini menyediakan konten terkait dengan branding, judul layar, navigasi, dan tindakan. App Bar bersifat *Persistent* yang muncul di bagian atas setiap layar dalam suatu aplikasi, dan dapat menghilang saat gulir. Konsistensi *bar* aplikasi teratas memiliki posisi dan konten yang konsisten untuk meningkatkan keakraban seperti yang terlihat pada Gambar 2.5 (<https://material.io/components/app-bars-top>)

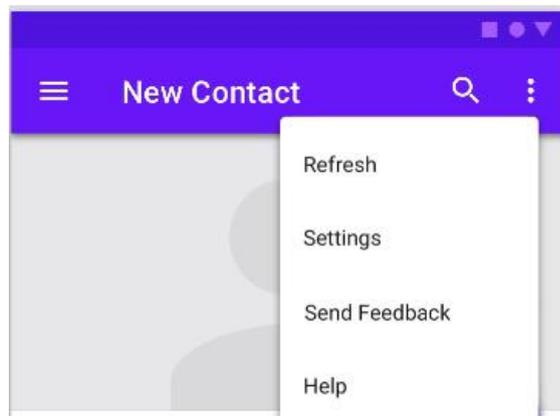


Gambar 2.5. App Bar

### 2.2.6 Menu

Menu menampilkan daftar pilihan pada permukaan sementara. Mereka muncul ketika pengguna berinteraksi dengan tombol, aksi, atau kontrol lainnya. Menu harus mudah dibuka, ditutup, dan berinteraksi dengan pengguna, konten menu harus sesuai dengan kebutuhan pengguna dan item menu harus mudah dipindai. Menu memungkinkan pengguna membuat pilihan dari banyak opsi, mereka kurang menonjol dan memakan ruang lebih sedikit daripada kontrol seleksi, seperti seperangkat Radio Button.

Menu harus diposisikan relatif ke tepi layar, menu biasanya muncul di sebelah (atau di depan) elemen yang lainnya. Jika mereka berada di posisi yang harus dipotong oleh tepi layar, menu dapat muncul di sebelah kiri, kanan, atau di atas elemen yang menghasilkannya, seperti yang terlihat pada Gambar 2.6. (<https://material.io/components/menus>)

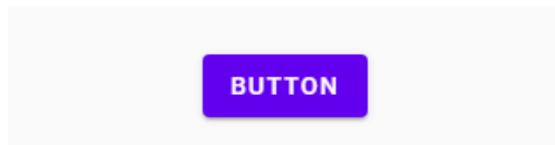


Gambar 2.6. Menu

### 2.2.7 Button

Button mengkomunikasikan tindakan yang dapat dilakukan pengguna. Mereka biasanya ditempatkan di seluruh UI, di tempat-tempat seperti Dialog, Modal Windows, Form, Card.

Button harus menunjukkan bahwa mereka dapat memicu tindakan dan mudah ditemukan di antara elemen lain, termasuk Button lainnya, tindakan dan status Button harus jelas, contoh dari penggunaan Button dapat dilihat pada Gambar 2.7. (<https://material.io/components/buttons>)



Gambar 2.7. Button

### 2.2.8 Floating Action Button

Floating Action Button (FAB) melakukan tindakan utama, atau paling umum, pada layar. Itu muncul di depan semua konten layar, biasanya sebagai bentuk lingkaran dengan ikon di tengahnya. FAB datang dalam tiga jenis: *regular*, *mini*, dan *extended*. FAB biasanya melakukan tindakan konstruktif (seperti membuat, berbagi, atau menjelajahi), sebuah FAB harus relevan dengan layar yang muncul, penggunaan FAB *regular* dapat dilihat pada Gambar 2.8. (<https://material.io/components/buttons-floating-action-button>)



Gambar 2.8. Floating Action Button

### 2.2.9 Divider

Divider memisahkan konten menjadi grup yang jelas, Divider harus terlihat dalam tata letak, tetapi tidak terlalu mencolok. Divider hanya boleh digunakan jika elemen tidak dapat dipisahkan menggunakan ruang putih, gunakan Divider untuk membuat pengelompokan dan memisahkan item, seperti yang terlihat pada Gambar 2.9. (<https://material.io/components/dividers>)



Gambar 2.9. Divider

### 2.2.10 Text Field

Text Field memungkinkan pengguna untuk memasukkan teks ke UI, Text Field biasanya muncul dalam *form* dan *dialog*. Text Field harus menonjol dan menunjukkan bahwa pengguna dapat memasukkan informasi, status Text Field harus dibedakan secara jelas satu sama lain. Text Field harus memudahkan untuk memahami informasi yang diminta dan untuk mengatasi kesalahan, Text Field terdiri dari dua jenis:

1. Filled Text Fields
2. Outlined Text Fields

Kedua jenis bidang teks menggunakan Container untuk memberikan kemampuan interaksi yang jelas, seperti yang terlihat pada Gambar 2.10. (<https://material.io/components/text-fields>)



Gambar 2.10. Text Field

### 2.3 Flutter

Flutter merupakan sebuah SDK untuk pengembangan aplikasi mobile yang dikembangkan oleh Google untuk membangun aplikasi yang memiliki kinerja tinggi serta dapat dipublikasi ke *platform* Android dan iOS dari *codebase* tunggal, Aplikasi yang dibuat dengan Flutter dapat di-build ke Android dan iOS, sedangkan Android Studio hanya bisa di-build ke Android saja. Flutter dapat dengan mudah dipelajari karena menggunakan bahasa pemrograman Dart yang pastinya terasa familiar jika sudah terbiasa menggunakan bahasa pemrograman Java atau Javascript. Selain itu Flutter juga menyertakan mesin *render* sendiri untuk menampilkan *widget*-nya, hal ini menguntungkan *developer* yang ingin memiliki tampilan UI unik yang konsisten pada semua perangkat karena tidak bergantung pada *widget* bawaan OEM.(CrossTechno, 2019)

Dalam Flutter tersedia banyak *package* yang dapat di-*import* kedalam *project* Flutter diantaranya, *package* audioplayer.dart yang dapat digunakan untuk memutar file *audio*, *package* folding\_cell.dart yang berfungsi untuk membuat animasi dengan efek melipat kartu, agar aplikasi bisa mengirim *feedback* dapat menggunakan *package* flutter\_email\_sender.dart. Flutter secara *default* menyediakan *package* material.dart yang dapat digunakan untuk membuat *widget* dengan *style* Material Design, *package* material.dart sudah tersedia dalam Flutter sehingga tidak perlu untuk meng-*install plugin* terlebih dahulu, berbeda dengan *package-package* yang tidak tersedia secara *default* dalam Flutter maka harus meng-*install plugin* terlebih dahulu dalam *pubspec.yaml* agar dapat di *import* dalam bentuk *package* dan dapat digunakan dalam penulisan *code*.

### 2.3.1 Keunggulan Flutter

Berikut adalah beberapa kelebihan yang dimiliki oleh Flutter

1. Animasi *Super Smooth*, jika tidak ada elemen yang berlebihan maka mobile apps yang dikembangkan dengan menggunakan Flutter dapat berjalan secara konstan 60-120 fps. Karena menggunakan sistem render miliknya sendiri.
2. *Integrated development environment (IDE)* terkenal, Dapat dikembangkan dengan IDE seperti Android Studio, IntelliJ IDEA serta VSCode.
3. Mudah dipelajari, untuk yang lebih familiar dengan bahasa Java, Objective-C ataupun Swift Flutter akan mudah dipahami
4. Hot Reload, build mobile apps untuk debug lebih cepat. (Crocodic.com, 2018)

### 2.3.2 Kekurangan Flutter

Berikut adalah beberapa kekurangan yang dimiliki oleh Flutter

1. File (*Android application package*) APK lebih besar, konsekuensi dari konstannya performa mobile apps yang dikembangkan dari Flutter berimbas pada ukuran file yang dihasilkan, karena Flutter menggunakan Widgetnya sendiri.
2. Logika animasi lebih sulit, selain File yang dihasilkan lebih besar, performa tinggi dari mobile apps yang dihasilkan juga berimbas pada saat penulisan logika pembuatan mobile apps.
5. Environment pengembang masih dalam perkembangan, karena Flutter tergolong masih baru maka jika terjadi bug atau kesalahan pada proses pengembangan, akan sulit untuk mencari solusinya. (Crocodic.com, 2018)

## 2.4 Firebase

Firebase adalah platform pengembangan aplikasi seluler dan web yang dikembangkan oleh Firebase, Inc. pada 2011, kemudian diakuisisi oleh Google pada 2014. Firebase memiliki produk utama, yaitu menyediakan database realtime dan backend sebagai layanan (*Backend as a Service*). Layanan ini menyediakan pengembang aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di cloud Firebase ini. Firebase menyediakan library untuk berbagai client platform yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Objective-C dan Node aplikasi Js dan dapat juga disebut sebagai layanan DbaaS (*Database as a Service*) dengan konsep realtime. Firebase digunakan untuk mempermudah dalam penambahan fitur-fitur yang akan dibangun oleh developer. (<https://firebase.google.com/docs/database>)

Pada aplikasi yang akan dibuat akan menggunakan Cloud Firestore dari Firebase sebagai *database* untuk soal latihan beserta jawabannya, Cloud Firestore merupakan *realtime database* maka dari itu jika data yang ada dalam Cloud Firestore diubah maka soal dan jawaban yang ada dalam aplikasi juga akan berubah dalam waktu yang sama (*realtime*). Untuk dapat menghubungkan Firebase dengan aplikasi yang dibangun menggunakan Flutter diperlukan *google-service plugin*, dan *FlutterFire plugin* dengan menggunakan *plugin* aplikasi Flutter dapat terhubung secara *realtime* dengan *database* Cloud Firestore.

## 2.5 Tajwid

Secara etimologi, kata tajwid merupakan bentuk mashdar dari kata jawwada yang berarti memperbaiki atau memperindah, sedangkan menurut terminologi,

tajwid adalah “Mengucapkan setiap huruf dari tempat keluarnya serta memberikan ḥaq dan mustahq dari sifat-sifatnya”. Menurut al-Suyuṭi, tajwid adalah hiasan bacaan, yaitu memberikan kepada setiap huruf hak-haknya dan urutan-urutannya serta mengembalikan setiap huruf kepada makhraj dan asalnya, melunakkan pengucapan dengan keadaan yang sempurna, tanpa berlebih-lebihan dan memaksakan diri. Oleh karena itu, ilmu tajwid adalah ilmu yang mempelajari tentang pemenuhan ḥaq dan mustahq huruf meliputi tempat keluar huruf (*makhrāj*) dan sifatsifatnya. (Ahmad Hanifuddin Ishaq dan Ruston Nawawi, 2017)

Yang dimaksud dengan hak huruf adalah sifat asli yang selalu bersama dengan huruf tersebut, seperti *al-jahr*, *isti'la*, *istifal* dan lain sebagainya. sedangkan mustahak huruf adalah sifat yang nampak sewaktu-waktu, seperti *tafkhim*, *tarqiq*, *ikhfa* dan lain sebagainya. Maka dapat dikatakan ilmu *tajwid* adalah pengetahuan tentang kaidah serta cara-cara membaca Al-Qur'an dengan mengeluarkan huruf dari makhrojnya serta memberi hak dan mustahaknya. (Abu Ezra Laili Al-Fadhli, 2017)

### 2.5.1 Hukum Bacaan *Nun Mati* dan *Tanwin* (نْ / نً)

*Nun mati* atau *tanwin* (نْ / نً) jika bertemu dengan huruf *hijaiyah* hukum bacaannya ada 5 macam, yaitu:

#### 1. *Izhar halqi*

Yaitu membaca *nun mati* atau *tanwin* dengan jelas. Apabila ada *nun mati* atau *tanwin* (نْ / نً) bertemu dengan salah satu huruf halqi (ا ح خ ع غ ه) maka dibaca jelas.

Contoh: فَصَلِّ لِرَبِّكَ وَأَنحِرْ

Faṣalli lirabbika wan-ḥar

## 2. *Idgham Bighunnah*

Yaitu membaca *nun* mati atau *tanwin* (  $\text{ـ} / \text{ـ}$  ) dengan memasukkan atau meleburkan kedalam huruf sesudahnya dengan disertai dengung, Apabila ada *nun* mati atau *tanwin* (  $\text{ـ} / \text{ـ}$  ) bertemu dengan salah satu huruf ( ن م و ي ) maka dibaca dengan dengung.

Contoh: الَّذِي جَمَعَ مَالًا وَعَدَّدَهُ

Allazī jama'a mālaw wa 'addadah

## 3. *Idgham Bilaghunnah*

Yaitu membaca *nun* mati atau *tanwin* (  $\text{ـ} / \text{ـ}$  ) dengan meleburkan kedalam huruf sesudahnya tanpa disertai dengung, Apabila ada *nun* mati atau *tanwin* (  $\text{ـ} / \text{ـ}$  ) bertemu dengan salah satu huruf ( ر dan ل ) maka dibaca tanpa dengung.

Contoh: فَهُوَ فِي عَيْشَةٍ رَّاضِيَةٍ

Fahuwa fii 'iisyatir(n) raadhiyat(in)

## 4. *Iqlab*

Yaitu membaca *nun* mati atau *tanwin* (  $\text{ـ} / \text{ـ}$  ) yang bertemu dengan huruf ( ب ), dengan menyuarakan atau merubah bunyi (  $\text{ـ}$  ) menjadi suara (  $\text{ـ}$  ), dengan merapatkan dua bibir serta mendengung.

Contoh: سَمِيعٌ بِصِيرٍ

Samii'un bashiirun

## 5. *Ikhfa' haqiqi*

Yaitu membaca *nun* mati atau *tanwin* (  $\text{ـ} / \text{ـ}$  ) dengan samar-samar, apabila ada *nun* mati atau *tanwin* (  $\text{ـ} / \text{ـ}$  ) bertemu dengan salah satu

huruf (ك ق ف ظ ط ض ص ش س ذ د ج ث ت), maka dibacanya samar-samar dengan mendengung.

Contoh: مِنْ شَرِّ مَا خُلِقَ

Min syarri mā khalaq

### 2.5.2 Hukum Bacaan *Mim Mati* (مْ)

*Mim* mati (مْ) bila bertemu dengan huruf *hijaiyah*, hukum bacaannya ada 3 macam, yaitu:

#### 1. *Ikhfa Syafawi*

Apabila *mim* mati (مْ) bertemu dengan *ba* (ب), maka cara membacanya harus dibunyikan samar-samar di bibir dan didengungkan.

Contoh: تَرْمِيهِمْ بِحِجَارَةٍ

Tarmīhim biḥijāratin

#### 2. *Idgham Mimi*

Apabila *mim* mati (مْ) bertemu dengan *mim* (م), maka cara membacanya adalah seperti menyuarakan *mim* rangkap atau ditasyidkan dan wajib dibaca dengung.

Contoh: إِنْ كُنْتُمْ مُؤْمِنِينَ

Ing kuntum m'miniin

#### 3. *Izhar Syafawi*

Apabila *mim* mati (مْ) bertemu dengan salah satu huruf *hijaiyah* selain huruf *mim* (م) dan *ba* (ب), maka cara membacanya dengan jelas di bibir dan mulut tertutup.

Contoh: عَلَيْهِمْ حَافِظِينَ

‘Alaihim haafidziin

### 2.5.3 Qalqalah

Menurut istilah *qalqalah* adalah bunyi huruf yang memantul bila ia mati atau dimatikan. Huruf *qalqalah* terdiri atas lima huruf (ق ج ب ط ق) berikut adalah 2 macam jenis *qalqalah* :

#### 1. *Qalqalah kubra*

Yaitu huruf *qalqalah* yang berbaris hidup, dimatikan karena *waqaf* (cirinya ada di akhir kalimat). inilah *qalqalah* yang paling utama, cara membacanya dijelaskan *qalqalahnya*.

Contoh: وَالسَّمَاءِ وَالطَّارِقِ

Was-samā`i waṭ-ṭāriq

#### 2. *Qalqalah Sugra*

Yaitu huruf *qalqalah* yang berbaris mati, tetapi tidak *waqaf* padanya (cirinya ada di tengah kalimat), caranya membacanya tidak terlalu jelas *qalqalahnya*.

Contoh: وَقَوْمِ إِبْرَاهِيمَ

Waqaumū ibrahīim