BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang membandingkan Kotlin dengan Java pernah dilakukan oleh Mateus dan Martinez (2018). Pada penelitian ini dilakukan perbandingan mengenai pengaruh Koltin terhadap aplikasi Android. Hasil dari penelitian ini menunjukkan bahwa dampak langsung dari penggunaan bahasa pemrograman Kotlin pada aplikasi Android adalah, aplikasi menghasilkan peningkatan kualitas setidaknya sebesar 50%.

Kemudian Schwermer (2018) pernah melakukan penelitian tentang Kotlin dengan Java mengenai kinerja aplikasi Android menggunakan metode *benchmarking*. Pada penelitian ini ditemukan bahwa *compiler* AOT menghasilkan *bytecode* DEX yang lebih ringkas untuk Java dibandingkan dengan Kotlin, dimana *bytecode* Kotlin berisi instruksi sebesar 3.125% – 119% lebih banyak daripada Java pada implementasi 3 dari 4 *benchmark* yang digunakan. Selanjutnya Java juga lebih sedikit menggunakan *unique instructions* dibandingkan dengan Kotlin untuk 3 dari 4 *benchmark*, dengan perbedaan antara 3.226% - 63.55%. Tren ini tetap sama ketika melihat urutan *bytecode*, bahwa Java menunjukkan sedikit variasi dalam *bytecode* yang dihasilkan. Selanjutnya, karena Kotlin menunjukkan jumlah variasi *bytecode* yang lebih besar daripada Java, ini dapat mengakibatkan sistem *runtime* kurang optimal untuk urutan *bytecode* yang diproduksi oleh Kotlin, sehingga mengarah berdampak negatif pada runtime.

Kemudian Sibarani dkk. (2018) telah melakukan analisis tentang performa aplikasi Android pada bahasa pemrograman Java dengan Kotlin dengan objek penelitian aplikasi pencari rumah sakit. Hasil dari penelitian ini adalah bahasa pemrograman Kotlin memiliki performa yang lebih baik dibandingkan bahasa pemrograman Java untuk aplikasi berbasis android.

Aji (2018) telah melakukan analisis perbandingan dengan objek yang dibandingkan adalah *framework* Yii dan Codeigniter, perbandingan dilakukan pada bagian CRUD antar *framework*. Hasil dari penelitian ini adalah kecepatan akses halaman menunjukkan bahwa *framework* Codeigniter lebih cepat dibandingkan dengan *framework* Yii.

Kemudian Hellbrück juga telah melakukan pendekatan Data Mining untuk membandingkan Java dengan Kotlin. Pada penelitian ini dilakukan analisis pada beberapa proyek aplikasi yang diambil dari GitHub kemudian diterapkan beberapa metode statistik. Hasil dari penelitian ini menunjukkan bahwa perbedaan substansial pada Kotlin dan Java dalam terjadinya praktik buruk tidak dapat dikonfirmasi.

Perbedaan antara penelitian sebelumnya dengan penelitian yang sedang dilakukan yaitu dari objek yang diteliti dan parameter yang dibandingkan, pada penelitian ini dilakukan perbandingan mengenai proyek aplikasi dengan parameter jumlah baris kode, banyak kelas, dan waktu kompilasi, dan perbandingan pada aplikasi dengan ukuran aplikasi, penggunaan CPU, penggunaan memori, dan performa *render*, dan perbandingan pada akses data dari *web server*.

Rangkuman penelitian yang pernah dilakukan dan akan dilakukan mengenai perbandingan antara bahasa pemrograman Kotlin dengan Java seperti pada tabel 2.1.

Tabel 2.1 Data Penelitian Mengenai Bahasa Pemrograman Kotlin dan Java

Penulis	Topik	Teknologi	Objek	Hasil
Bruno Gois Mateus, Matias Martinez (2018)	Uji kualitas aplikasi Android	Analisis APK	925 aplikasi Android open source	Grafik kualitas aplikasi
Patrik Schwermer (2018)	Evaluasi kinerja Java dan Kotlin	Bencmarking	Aplikasi Android	Metrics benchmark
Niko Sumanda Sibarani, Ghifari Munawar, Bambang Wisnuadhi (2018)	Performa aplikasi Android	Snapdragon Profiler	Aplikasi pencari rumah sakit	Rekapitulasi pengukuran performa
Muhammad Darma Aji (2017)	Perbandingan framework	Analisis perbandingan	Pendaftaran siswa baru	Hasil analisis perbandingan *framework*
Simon Hellbrück (2019)	Perbandingan dengan Data Mining	Data Mining	Proyek Java dan Kotlin	Hasil analisis proyek Java dan Kotlin
Katon Aditya Putra (2019)	Komparasi bahasa pemrograman	Komparasi	2 Aplikasi Android akses data *web server*	Hasil perbandingan proyek, aplikasi, dan penggunaan data.

2.2 Dasar Teori

2.2.1 Android

Android adalah sebuah sistem operasi *mobile* yang berbasiskan pada versi modifikasi dari Linux. Pertama kali sistem operasi ini dikembangkan oleh

perusahaan Android.Inc. Nama perusahaan inilah yang pada akhirnya digunakan sebagai nama proyek sistem operasi *mobile* tersebut, yaitu sistem operasi Android.

Pada tahun 2005, sebagai bagian dari strategi untuk memasuki pasar *mobile*, Google membeli Android dan mengambil alih proses pengembangannya sekaligus Team developer Android. Google menginginkan Android untuk menjadi sistem operasi *Open Source* dan gratis, kebanyakan *code* Android dirilis di bawah lisensi Open Source Apache yang berarti setiap orang bebas untuk menggunakan dan mengunduh *source code* Android secara penuh.

Terlebih lagi para vendor bebas untuk mengubah sekaligus membuat penyesuaian untuk Android. Di samping itu, perusahaan dapat secara bebas untuk membuat perbedaan dari produk vendor lainnya. Model pengembangan yang sederhana membuat Android sangat atraktif dan hal tersebutlah yang membuat para vendor tertarik untuk mencoba sistem operasi Android.

2.2.2 Java

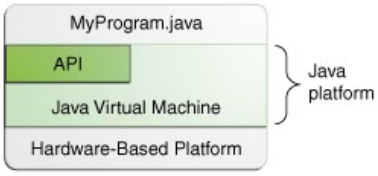
Bahasa Java merupakan bahasa pemrograman yang berorientasi objek yang diciptakan oleh James Gosling dan beberapa insinyur lainnya di Sun Microsystems. Java dikembangkan pertama sekali pada tahun 1991 sebagai bagian dari Green Project. Pada awalnya, Java dirancang untuk menggantikan bahasa C++ dan dikenal dengan nama Oak.

Platform Java berbeda dengan kebanyakan platform yang lain. Dalam platform Java, platform perangkat lunak berjalan di atas platform berbasis

perangkat keras. Kebanyakan platform yang lain merupakan kombinasi antara perangkat keras dan sistem operasi. Platform Java memiliki dua komponen

1. Java Virtual Machine (JVM)
2. Java Application Programming Interface (Java API)

Java API merupakan kumpulan komponen perangkat lunak yang siap buat yang menyediakan berbagai fasilitas, seperti GUI *widget*. Java API dikelompokkan dalam paket (*package*) komponen-komponen yang berkaitan.



Gambar 2.1 Platform Java (sumber: Oracle,2018)

Dari gambar 2.2 di atas terlihat, dalam setiap program Java berjalan di atas platform Java. Platform Java mengisolasi program Java dengan perangkat keras, sehingga program Java tidak bergantung dengan perangkat keras yang digunakan (*hardware independent*).

Java memiliki beberapa keunggulan bila dibandingkan dengan bahasa pemrograman lainnya, yaitu:

1. Java bersifat sederhana dan relatif mudah

   Java dimodelkan sebagian dari bahasa C++, namun dengan memperbaiki beberapa karakteristik C++, seperti mengurangi kompleksitas beberapa fitur, penambahan fungsionalitas, serta

penghilangan beberapa aspek pemicu ketidakstabilan sistem pada C++ Sebagai contoh, Java menggantikan konsep pewarisan lebih dari satu (multiple inheritance) dengan interface, menghilangkan konsep pointer yang sering membingungkan, otomatisasi sistem alokasi memory, dan sebagainya. Ini membuat Java menjadi relatif sederhana dan mudah untuk dipelajari dibandingkan bahasa pemrograman lainnya.

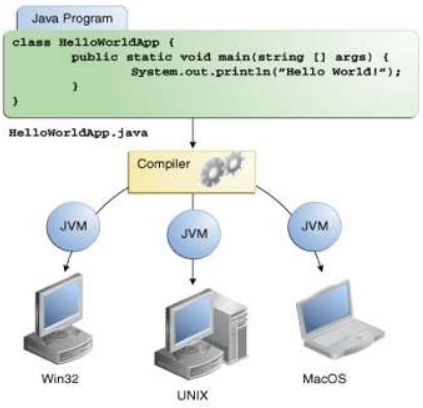
2. Java berorientasi pada objek (Object Oriented)

   Java adalah bahasa pemrograman yang berorientasi objek (OOP), bukan seperti Pascal, Basic, atau C yang berbasis prosedural. Dalam memecahkan masalah, Java membagi program menjadi objek-objek, kemudian memodelkan sifat dan tingkah laku masing-masing. Selanjutnya, Java menentukan dan mengatur interaksi antara objek yang satu dengan lainnya.

3. Java bersifat terdistribusi

   Pada dekade awal perkembangan PC (Personal Computer), komputer hanya bersifat sebagai workstation tunggal, tidak terhubung satu sama lain. Saat ini, sistem komputerisasi cenderung terdistribusi, mulai dari workstation client, e-mail server, database server, web server, proxy server, dan sebagainya.

4. Java bersifat Multiplatform



Gambar 2.2 Multiplatform Java (Sumber: Oracle,2018)

Dewasa ini kita mengenal banyak platform Operating System, mulai dari Windows, Apple, berbagai varian UNIX dan Linux, dan sebagainya. Pada umumnya, program yang dibuat dan dikompilasi di suatu platform hanya bisa dijalankan di platform tersebut. Java bersifat multiplatform, yakni dapat diterjemahkan oleh Java Interpreter pada berbagai sistem operasi.

5. Java bersifat MultiThread

   Thread adalah proses yang dapat dikerjakan oleh program dalam suatu waktu. Java bersifat Multithreaded, artinya dapat mengerjakan beberapa proses dalam waktu yang hampir bersamaan.

Developer menggunakan edisi platform Java yang dibuat khusus oleh Google (lihat https://developer.android.com/index.html) untuk membuat aplikasi Android yang berjalan di perangkat yang mendukung Android. Edisi ini dikenal sebagai *Android platform*(Friesen, 2013).

2.2.3 Kotlin

Kotlin adalah bahasa pemrograman yang diketik secara statis yang menargetkan Java virtual machine (JVM), Android, JavaScript, dan Native. Kotlin dikembangkan oleh JetBrains, proyek Kotlin dimulai pada tahun 2010 dan merupakan Bahasa pemrograman *open source*. Kotlin versi 1.0 secara resmi dirilis pada bulan Februari 2016. Dalam mengembangkan aplikasi Android, Kotlin menawarkan beberapa fitur yaitu:

1. **Kompatibilitas**: Kotlin sepenuhnya kompatibel dengan JDK 6, sehingga Kotlin dapat berjalan pada perangkat Android versi lama. Kotlin juga sepenuhnya didukung di Android Studio dan kompatibel dengan Android build system.
2. **Performa**: Sebuah aplikasi Kotlin dapat berjalan secepat Java, karena memiliki struktur bytecode yang sangat mirip. Dengan dukungan Kotlin untuk fungsi inline, kode menggunakan lambda sering berjalan lebih cepat daripada kode yang sama yang ditulis di Java.
3. **Interoperabilitas**: Kotlin 100% dapat beroperasi dengan Java, memungkinkan *programmer* untuk menggunakan semua *Android Library* dalam aplikasi, termasuk pengolahan anotasi, sehingga *databinding* dan *Dagger* dapat bekerja.
4. ***Footprint***: Kotlin memiliki *runtime library* yang sangat padat/tersusun rapat, yang ke depannya dapat dikurangi melalui penggunaan *ProGuard*. Dalam aplikasi nyata, *runtime* pada Kotlin hanya

menambahkan beberapa ratus metode dan kurang dari 100K untuk ukuran dari file .apk.

5. **Waktu kompilasi**: Kotlin mendukung kompilasi Inkremental yang efisien, sehingga sementara ada beberapa tambahan *overhead* untuk *clean build*, pembangunan bertahap ini biasanya sama cepat atau lebih cepat daripada dengan Java.
6. **Learning Curve**: untuk pengembang Java, memulai menggunakan Kotlin sangat mudah. *Automated Java to Kotlin converter* yang terdapat dalam plugin Kotlin membatu pengembang Java pada langkah pertama. Terdapat juga *Kotlin Koans* yang menawarkan panduan dalam penggunaan Bahasa pemrograman ini.

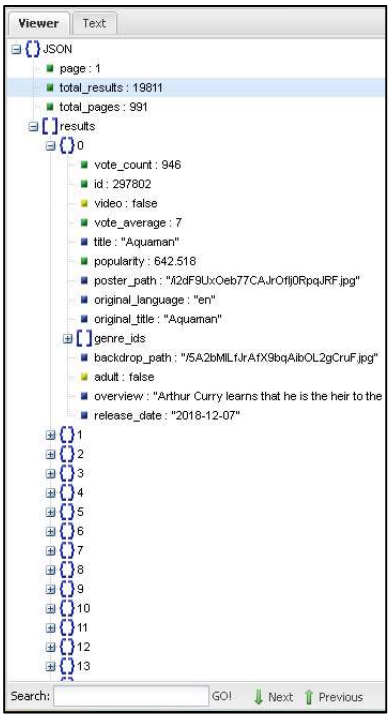
2.2.4 *Material design*

Material design adalah bahasa visual yang dikembangkan oleh Google pada tahun 2014. Pada *material design* terdapat *CardView*, *CardView* merupakan *FrameLayout* dengan latar belakang sudut tumpul dan bayangan.

2.2.5 The Movie Database (TMDb)

TMDb adalah *database online* yang menyediakan informasi berkaitan dengan film dan program TV, informasi yang disediakan antara lain daftar film dan TV populer, informasi detail seperti tanggal rilis, sutradara, bintang film, aktor, sinopsis, foto, video trailer, dll.

TMDb menyediakan layanan API sehingga developer aplikasi dapat menambahkan informasi mengenai Film dan program TV pada aplikasi mereka. Sejak tahun 2008, TMDb telah digunakan oleh lebih dari 200,000 *developer* dan perusahaan. Berikut contoh respons JSON yang didapat dari API TMDb.



Gambar 2.3 Respon JSON dari API TMDb

2.2.6 Dumpsys

Dumpsys adalah alat Android yang berjalan di perangkat dan memberikan informasi menarik tentang status layanan sistem. Dengan memberikan perintah *gfxinfo* pada *dumpsys*, maka akan memberikan *output* pada *logcat* tentang informasi

performa yang berkaitan dengan *frame* dari animasi yang terjadi selama fase perekaman.

Contoh perintah *gfxinfo* pada *dumpsys* dengan framestat:

```
> adb shell dumpsys gfxinfo <PACKAGE_NAME> framestats
```

Perintah tersebut akan menghasilkan beberapa *output*, di antaranya FRAME_COMPLETED dan INTENDED_VSYNC. Jika kedua data tersebut diselisihkan maka didapat total waktu yang dihabiskan untuk memproses *frame* (*rendering*).

2.2.7 Android Profiler

Android profiler adalah alat pada Android Studio yang menyediakan data *real time* aplikasi seperti penggunaan CPU, memori, jaringan, dan baterai. Android profiler membantu *developer* dalam mengukur kinerja aplikasi.

2.2.8 Standar Deviasi

Standar deviasi (SD) merupakan ukuran variabilitas dalam sampel data. Semakin tinggi nilai SD maka semakin tinggi variabilitas dalam sebuah sampel. SD dapat dihitung dengan mengakarkan nilai varian seperti pada gambar berikut.

$$\text{Variance} = \frac{\sum_{i=1}^{n} (\bar{x} - x_i)^2}{n-1}, \text{Standard Deviation (SD)} = \sqrt{\frac{\sum_{i=1}^{n} (\bar{x} - x_i)^2}{n-1}}$$

Gambar 2.4 Rumus Standar Deviasi (Sumber: Lee, D.K, dkk, 2015)

Keterangan:

Variance = Tingkat keragaman dalam data

$\sum_{i=1}^{n}$ = Jumlah nilai ke i sampai n

x_i = Nilai x ke-i

n = Ukuran sampel

$\bar{x}$ = Nilai rata-rata