

Perbandingan Pemrograman Swift dan Java pada Bujang Kurir

Pandhe Prakarsa^{a1}, M. Azhar Irwansyah^{a2}, Helen Sastypratiwi^{a3}

*^aProgram Studi Sarjana Informatika Fakultas Teknik Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak 78124*

¹pandheprakarsa@gmail.com

²irwansyah.azhar@gmail.com

³helensastypratiwi@informatics.untan.ac.id

Abstrak

Android dan iOS adalah dua platform yang paling mendominasi pangsa pasar perangkat mobile di dunia ini. Semua perusahaan dan Start-up berlomba lomba untuk merancang dan membangun aplikasi untuk kedua platform ini. Akan tetapi belum ada penelitian yang mengangkat topik untuk mengetahui bagaimana perbedaan merancang dan membangun aplikasi menggunakan Java android dan Swift iOS serta bagaimana perilaku aplikasi tersebut terhadap perangkat mobile. Untuk mengatasi masalah ini, dilaksanakan penelitian yang melakukan analisis fungsional pada bahasa pemrograman Java Mobile pada Android dan Swift pada iOS, serta analisis performa CPU dan memori menggunakan pengukuran yang telah disediakan oleh masing masing IDE bahasa pemrograman tersebut. Berdasarkan hasil penelitian ini, pembangunan aplikasi iOS menggunakan perancangan aplikasi Android dapat dilakukan, dan dapat dilakukan lebih cepat karena struktur bahasa pemrograman Swift iOS yang lebih ringkas jika dibandingkan dengan Java Android dan tidak menghasilkan tambahan beban CPU dan Memori sebesar yang terjadi pada Java Android setiap tambahan aktivitas yang dilakukan.

Kata kunci: Bujang Kurir, Android, iOS, Java Mobile, Swift, Analisis, Fungsional, Performa, CPU, Memory, IDE

Comparison of Swift and Java Programming in Bujang Kurir

Abstract

Android and iOS are the two platforms that most dominate the market share of mobile devices in the world. All companies and start-ups are competing to design and build applications for both platforms. However, there has been no research that raised the topic to find out how the differences in designing and building applications using Java Android and iOS Swift and how the application behaves towards mobile devices. In overcoming this problem, a study was carried out which performed a functional analysis of the Java Mobile programming language on Android and Swift on iOS, as well as a review of CPU and memory performance using measurements provided by each IDE of the programming language. Based on the results of this study, iOS application development using Android application design can be done faster. Because the structure of the iOS Swift programming language is more concise when compared to Java Android and does not produce additional CPU and Memory load as much as happens on Java Android, each other activities carried out.

Keywords: Bujang Kurir, Android, iOS, Java Mobile, Swift, Analysis, Functional, Performance, CPU, Memory, IDE

I. PENDAHULUAN

Aplikasi berbasis mobile saat ini sudah semakin diminati oleh para pegiat usaha Start-up terutama di bidang jasa termasuk pada perusahaan Bujang Kurir. Dengan pangsa pasar telepon pintar yang pada saat di dominasi oleh sistem operasi Android dan IOS dimana pada tahun 2015 Google sebagai pemilik lisensi Android menguasai 47,7% market share distribusi aplikasi dan Apple sebagai pemilik lisensi iOS menyusul dengan market share 45,4% [1]. Perusahaan yang ingin menawarkan jasa mereka melalui aplikasi berbasis mobile, harus melakukan development terhadap dua platform tersebut, dimana jika tidak dilakukan berpotensi kehilangan lebih dari 40% bangsa pasar.

Google dan Apple telah menciptakan bahasa pemrograman yang sudah dioptimalkan untuk produk

mobile mereka. Google mengembangkan Bahasa Java Mobile serta XML, dan Apple mengembangkan Swift. Kedua platform tersebut telah meningkatkan performa pada bahasa pemrograman high level, dan kode program yang sensitif dengan performa perangkat dibungkus dengan pustaka pihak ketiga [2]. Akan tetapi pada saat ini masih belum ada penelitian yang melakukan perbandingan struktur dan penulisan bahasa pemrograman Java Mobile dan Swift iOS. Sedangkan untuk melakukan proses penambahan dukungan platform native pada suatu usaha Start-up, para programmer harus mengetahui perbedaan dan persamaan bahasa pemrograman yang akan dikembangkan dan yang sudah dikembangkan agar kedua bahasa pemrograman native tersebut dapat berjalan bersamaan.

Penelitian terdahulu yang dilakukan untuk mengetahui perbandingan antara pemrograman Android dan IOS adalah “Code Smells in iOS Apps: How do they compare to Android” Sarra Habchi [3]. Penelitian ini melakukan perbandingan code smells atau kode yang berpotensi menyebabkan masalah pada Android dan iOS. Penelitian ini menyebutkan bahwa pemrograman Android lebih memiliki banyak code smell dibandingkan IOS. Akan tetapi penelitian tersebut menyimpulkan bahwa code smell tersebut lebih disebabkan oleh platform dan bukan dari bahasa pemrograman yang digunakan. Terdapat juga penelitian yang merancang dan berhasil membangun aplikasi iOS [4]. Akan tetapi tidak ada perbandingan dengan platform Android.

Pada saat ini, Bujang Kurir menggunakan aplikasi Android berbasis Java Mobile untuk melayani konsumen. Diperlukan aplikasi berbasis IOS untuk melengkapi segmen pasar di Pontianak. Penelitian ini akan melakukan Analisa terhadap bahasa pemrograman yang digunakan untuk membangun aplikasi Bujang Kurir berbasis IOS, dan membandingkan aplikasi Android yang sudah ada dengan aplikasi IOS yang akan dibangun.

Berdasarkan dari latar belakang permasalahan yang telah dideskripsikan sebelumnya maka dianggap perlu dilakukan suatu penelitian terhadap proses development aplikasi berbasis IOS menggunakan Xcode dan bahasa pemrograman swift dan dibandingkan dengan aplikasi Android yang telah digunakan oleh Bujang kurir sebelumnya.

II. METODOLOGI

A. Metodologi Penelitian

Metodologi penelitian adalah cara yang dilakukan untuk mendapatkan suatu hasil penelitian yang dilakukan secara sistematis. Penelitian ini diawali dengan melakukan analisa terhadap penelitian sebelumnya yang berkaitan dengan Android, IOS, tools yang digunakan untuk membuat aplikasi Android dan iOS, dan komparasi terhadap kedua platform tersebut.

Dari hasil studi literatur, penelitian dilanjutkan dengan mengumpulkan data yang tersedia. Baik data pelanggan dan gerai pada bujang kurir, maupun perancangan aplikasi sebelumnya.

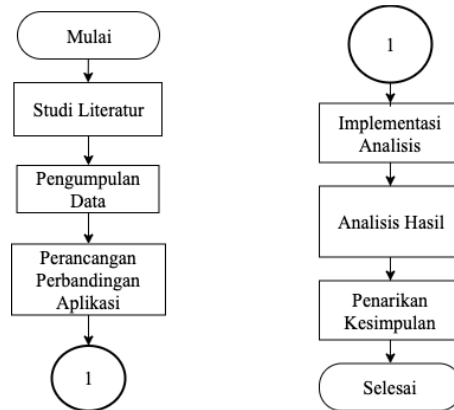
Dari hasil studi literatur dan pengumpulan data, kemudian penelitian dilanjutkan dengan merancang analisis yang dapat dilakukan untuk mencari perbedaan dari perbedaan dari perancangan dan pembangunan aplikasi berbasis Java Android dan Swift iOS.

Pada tahap implementasi analisis, berdasarkan hasil studi literatur, data yang tersedia, dan perancangan analisis yang telah dilakukan sebelumnya, penelitian dilanjutkan dengan melakukan analisis berdasarkan urutan yang telah dirancang sebelumnya.

Setelah analisis bahasa pemrograman dan pembangunan aplikasi iOS telah dilaksanakan, penelitian akan dilanjutkan dengan melakukan analisis terhadap hasil dari perbandingan bahasa pemrograman tersebut. Hasil dari analisis tersebut akan menghasilkan kesimpulan tentang perbedaan dari kedua bahasa pemrograman tersebut, sehingga dapat disimpulkan apakah hasil dari analisis

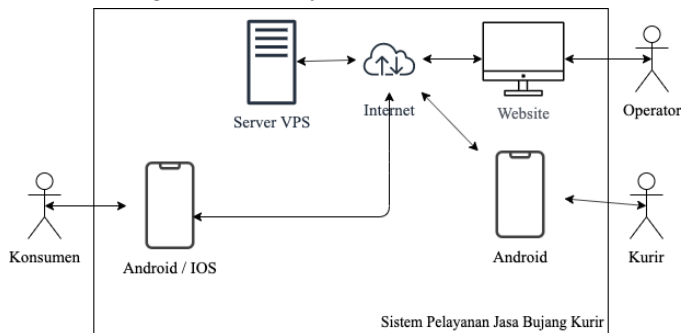
perbandingan bahasa pemrograman tersebut sesuai dengan hasil dari implementasi pembuatan kode program pada aplikasi Bujang Kurir atau tidak.

Pada tahap penarikan kesimpulan, sudah dapat ditarik kesimpulan terhadap perbandingan perbedaan bahasa pemrograman Swift iOS dan Java. Diagram dari langkah penelitian ini dapat dilihat pada gambar 1 sebagai berikut.



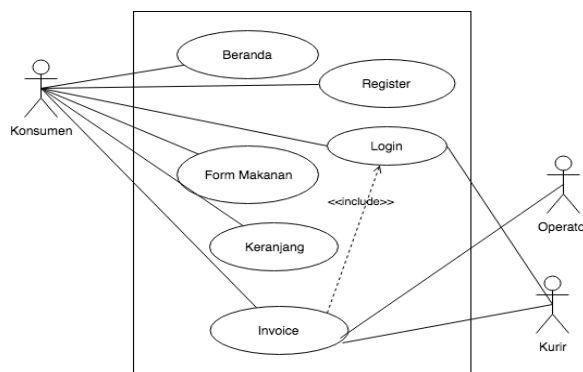
Gambar 1. Metodologi penelitian

A. Perancangan Sistem Berjalan



Gambar 2. Diagram arsitektur sistem

Pengguna jasa Bujang Kurir menggunakan aplikasi berbasis Android dan IOS. Aplikasi tersebut mengakses database pada server dengan melakukan JSON request. Seluruh data yang dikirim oleh aplikasi pengguna tersimpan pada penyimpanan data yang terdapat pada server. Data orderan, pelanggan, serta gerai dan outlet diakses oleh operator menggunakan aplikasi berbasis website. Jika terdapat orderan baru yang dikirimkan oleh aplikasi pelanggan, administrator akan mendapatkan notifikasi.



Gambar 3. Diagram use case

Pada Gambar 3 diatas, menunjukkan pada saat ini sistem orderan bujang kurir memiliki tiga aktor yang berperan, yaitu konsumen, operator, dan kurir. Proses membuka beranda aplikasi, mengisi form kanan, dan melihat keranjang belanjaan dilakukan oleh konsumen. Ketiga proses tersebut tidak memerlukan login. Proses invoice atau proses menyelesaikan transaksi dilakukan oleh konsumen dan diproses oleh operator dan kurir.

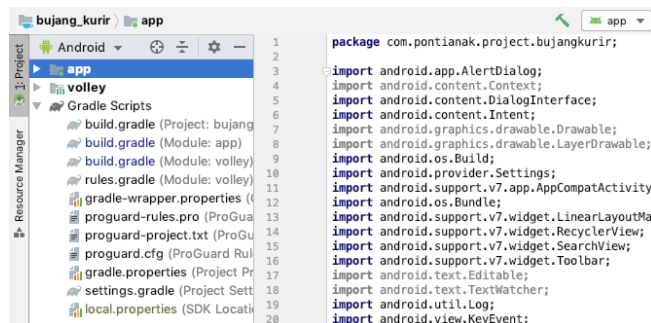
B. Perancangan Analisis Fungsional



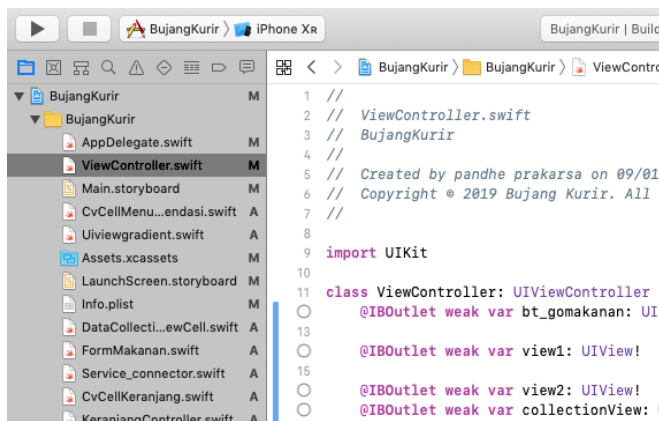
Gambar 4. Proses analisis fungsional

Tahapan pertama dari proses analisis adalah membuka dan memilih projek dan file yang akan di proses. Projek yang dibuka pada penelitian ini adalah project aplikasi Bujang Kurir. Pada project ini terdapat baris perintah serta source yang diperlukan.

Tahapan selanjutnya adalah memproses panjang dan jumlah baris perintah. Proses ini dilakukan dengan mengamati banyaknya perintah yang diperlukan untuk mendeklarasikan suatu perintah pada Java Mobile dan Swift iOS dimana perintah tersebut dituliskan pada IDE Android Studio dan Xcode seperti yang terlihat pada gambar 4 dan 5 sebagai berikut.



Gambar 5. Proses menghitung jumlah baris perintah pada Java Mobile



Gambar 6. Proses menghitung jumlah baris perintah pada Swift iOS

Terlihat pada gambar 5 dan 6, proses menghitung panjang dan jumlah baris menggunakan editor yang telah disediakan pada Android Studio untuk Java mobile dan Xcode untuk Swift. Banyaknya baris perintah dapat dilihat pada penomoran yang terletak tepat pada sisi kiri perintah.

Tahapan selanjutnya adalah mencatat jumlah baris dan panjang baris. Pada setelah panjang baris dan jumlah baris didapatkan, kemudian dilakukan analisis struktur penulisannya. Kode program akan dipotong untuk menghilangkan kode program yang tidak berkaitan dengan penulisan perintah yang sedang dianalisis dan baris kolom kosong yang biasanya dibuat programmer untuk memudahkan proses development, sehingga proses perhitungan panjang baris perintah mendapatkan hasil yang efektif. Seperti yang terlihat pada gambar 7 sebagai berikut.

```

1 public class Keranjang {
2     public static final String TABLE_NAME = "keranjang";
3
4     public static final String COLUMN_ID = "id";
5     public static final String COLUMN_STATUS_RESTO = "status_resto";
6     public static final String COLUMN_NAMA_RESTO = "nama_resto";
7     public static final String COLUMN_ITEM_ORDER = "item_order";
8     public static final String COLUMN_HARGA_ITEM = "harga_item";
9     public static final String COLUMN_POSRI_ITEM = "porsi_item";
10    public static final String COLUMN_TARIF = "tarif";
11    public static final String COLUMN_ALAMAT_DETAIL = "alamat_detail";
12    public static final String COLUMN_NAMA_NOHP = "id_item";
13    public static final String COLUMN_ID_ITEM = "id_item";
14    public static final String COLUMN_LAT = "lat";
15    public static final String COLUMN_LNG = "lng";
  
```

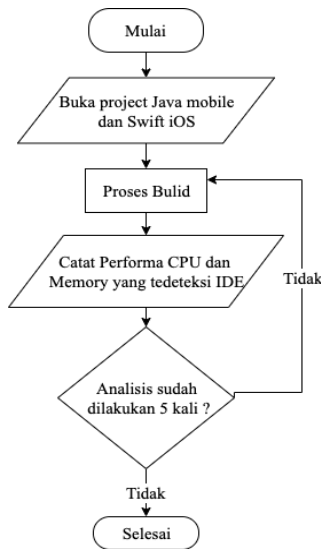
Gambar 7. Kode program yang dihitung jumlah barisnya dalam analisis.

Pada gambar 7 terlihat perintah tersebut memiliki 15 baris, akan tetapi karena terdapat 1 baris kolom kosong pada baris 3, maka secara efektif perintah tersebut memiliki 14 baris perintah. Perintah ini akan dibandingkan dengan perintah yang dibutuhkan untuk mendeklarasikan hal yang sama pada Swift iOS.

Tahapan terakhir dari analisis fungsional adalah mencatat kesimpulan sementara dari masing masing perintah yang akan dianalisis.

C. Perancangan Pengujian Performa

Pengujian pada penelitian ini menggunakan pengujian black box performance testing. Berikut skenario pengujian performa yang akan dilakukan untuk mendapatkan hasil analisis perbedaan performa aplikasi IOS dan Android.



Gambar 8. Proses Analisis Performa

Pada gambar 8 terlihat analisis performa CPU dan Memory dimulai dengan membuka proyek Android dan Xcode. Kemudian dilanjutkan dengan proses build, atau membangun hasil aplikasi Langkah selanjutnya adalah mencatat performa yang terlihat pada IDE Android Studio dan Xcode, pada Android Studio performa CPU dan Memory terlihat pada fitur *profile*

Data hasil pengujian diperoleh dengan menjalankan aplikasi dan mengamati resource monitor yang sudah disediakan pada Android Studio dan Xcode. Pengamatan akan diulangi lima kali untuk memastikan hasil yang diperoleh adalah hasil yang konsisten.

III. HASIL DAN PEMBAHASAN

A. Implementasi Hasil Analisis

Implementasi hasil analisis adalah analisis yang dilakukan untuk mengetahui perbedaan dari pemrograman Android dan pemrograman IOS, sehingga dapat ditarik kesimpulan dari keuntungan dan kerugian dari penggunaan kedua bahasa pemrograman tersebut. Hasil analisis fungsional dapat dilihat pada table I sebagai berikut.

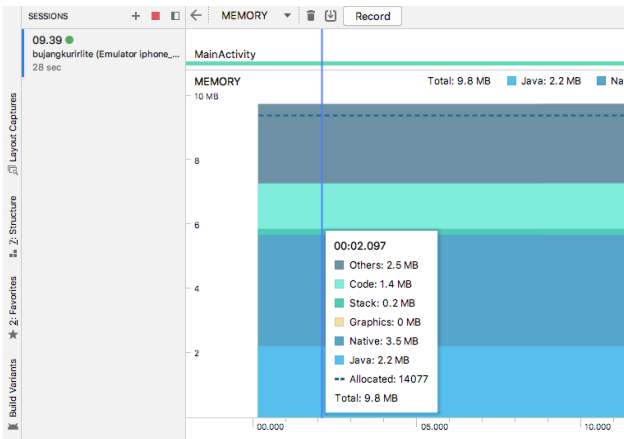
TABEL I
ANALISIS FUNGSIONAL

Aspek	Bahasa pemrograman		Kesimpulan
	Java Android	Swift iOS	
Kelas	Java Android	Swift iOS	Java Android dan Swift memiliki struktur kelas yang sama.
Variabel & Konstanta	Menggunakan deklarasi tipe variabel dan nama variabel	Tidak harus deklarasi tipe variabel	Sama sama memiliki variabel dan konstanta, tapi deklarasi pada Swift lebih ringkas.
Kelas	Menggunakan kelas perpanjangan untuk deklarasi Activity	Menggunakan kelas perpanjangan untuk deklarasi <i>ViewController</i>	Java Android dan Swift memiliki struktur kelas yang sama.

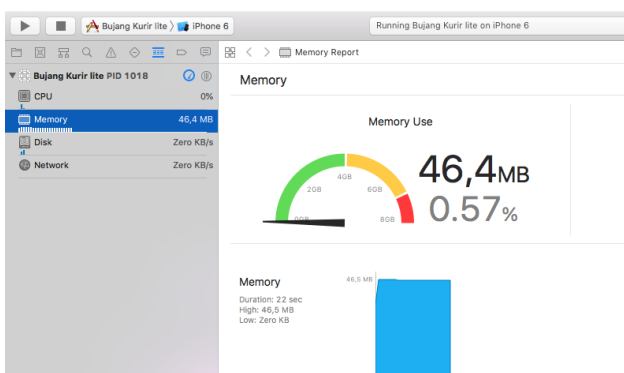
Aspek	Bahasa pemrograman		Kesimpulan
	Java Android	Swift iOS	
Controller	Java Android	Swift iOS	Controller pada Java Android dan Swift memiliki struktur yang sama
Metode	Menggunakan deklarasi tipe pengembalian	Tidak memerlukan deklarasi tipe pengembalian	Swift lebih ringkas.
Objek	Menggunakan deklarasi objek pada Java untuk deklarasi objek yang didapatkan dari server	Secara default, Menggunakan bantuan class <i>decoder</i> untuk deklarasi objek dari server	Swift lebih ringkas.
Model	Menggunakan tipe deklarasi standar Java	Menggunakan deklarasi variabel <i>inference</i>	Swift lebih ringkas
View	Menggunakan GUI editor dan XML editor	Menggunakan GUI editor	Perancangan view pada Android dapat diterapkan pada iOS
Beranda	Memerlukan 35 baris perintah untuk menampilkan gerai, dan 10 baris perintah untuk kategori	Memerlukan 21 baris perintah untuk menampilkan gerai, dan 10 baris perintah untuk kategori	Swift lebih ringkas
Form makanan	Memerlukan 34 baris perintah untuk menampilkan makanan, dan 8 baris perintah untuk menambah keranjang	Memerlukan 20 baris perintah untuk menampilkan makanan, dan 9 baris perintah untuk menambah keranjang	Swift lebih ringkas
Keranjang	Memerlukan 16 baris perintah untuk menampilkan item keranjang	Memerlukan 17 baris perintah untuk menampilkan item keranjang	Java Mobile lebih ringkas

B. Pengujian Performa

Hasil pengujian performa menunjukkan penggunaan memori pada aplikasi Bujang Kurir pada iOS lebih besar [9] dibandingkan versi Android. Akan tetapi aplikasi iOS menggunakan memori lebih banyak karena proyek awal dari iOS dapat dipastikan langsung mengalokasikan memori aplikasi lebih besar dibandingkan Android. Seperti yang terlihat pada gambar 9 dan 10 sebagai berikut.

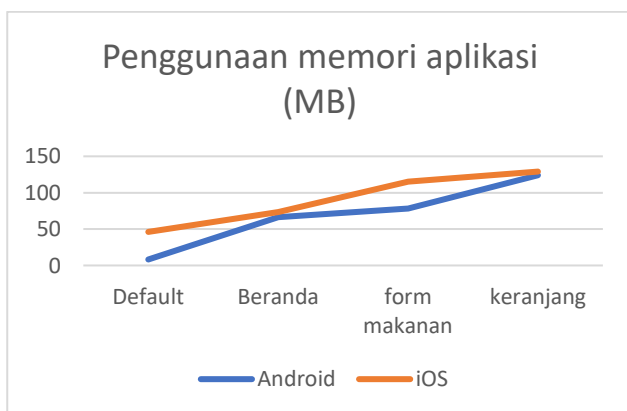


Gambar. 9 Performa memori aplikasi Android



Gambar. 10 Performa memori aplikasi iOS

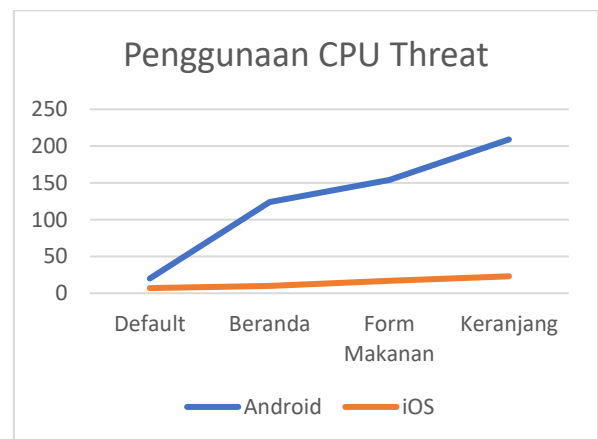
Pada gambar 11 dan 12 terlihat penggunaan memori pada aplikasi awal projek Android menggunakan memori lebih sedikit yaitu 9.8MB dibandingkan dengan aplikasi iOS yaitu 46,4MB. Akan tetapi jika seorang programmer memulai proses pengembangan, penggunaan memori aplikasi akan meningkat drastis. Berdasarkan hasil pengamatan dengan data yang paling besar, penggunaan memori pada Android meningkat sebesar 1412 % dari permulaan proyek sampai fitur terakhir yang diuji. Sedangkan penggunaan memori pada iOS meningkat 179,8 %. Hal ini didukung dengan hasil pengamatan jika kedua tabel tersebut ditampilkan dalam bentuk grafik seperti berikut.



Gambar. 11 Performa memori

Berdasarkan Gambar 11, dapat disimpulkan manajemen memori iOS lebih baik jika dibandingkan dengan Android, karena selama proses pembangunan aplikasi menggunakan swift penambahan kode di masing masing aktivitas tidak menyebabkan lonjakan penggunaan memori pada perangkat. Dibandingkan pemrograman pada swift, pemrograman java pada android sering menyebabkan lonjakan penggunaan memori pada masing masing aktivitas, bahkan pada akhir pengujian, penggunaan memori pada android semakin mendekati penggunaan memori pada iOS yang lebih besar penggunaan memorinya di *project default*.

Pada pengujian performa cpu terlihat penggunaan cpu pada aplikasi Bujang Kurir versi Android lebih tinggi dibandingkan dengan aplikasi iOS. Tidak berbeda dengan pengujian memori, penggunaan sumber daya cpu pada aplikasi android akan meningkat pada saat programmer memulai proses pembangunan aplikasi menjadi 1061 % pada saat pengujian berakhir. Sedangkan pada aplikasi iOS meningkat hanya 228 %, seperti yang dapat dilihat pada Gambar 12 sebagai berikut.



Gambar. 12 Performa memori

Pada Gambar 12 diatas terlihat bahwa penggunaan sumber daya CPU pada aplikasi android lebih banyak jika dibandingkan dengan aplikasi iOS, dan selisih antara keduanya cenderung semakin menjauh.

IV. KESIMPULAN

Setelah melakukan analisis fungsional serta melakukan pengujian black box performance testing pada bahasa pemrograman swift pada iOS dan java pada Android maka dapat ditarik kesimpulan Dari sisi kecepatan proses development bahasa pemrograman swift dapat lebih cepat karena struktur pemrograman serupa dengan Java Mobile dan penulisan baris perintah yang lebih ringkas, sesuai dengan hasil analisis fungsional.

Berdasarkan hasil dari pengujian black box performance testing, aplikasi Bujang Kurir berbasis iOS menunjukkan manajemen memori dan threat yang lebih baik dibandingkan aplikasi Bujang Kurir berbasis Android, sehingga proses pembangunan aplikasi Swift akan lebih mudah karena tidak perlu khawatir dengan lonjakan penggunaan sumber daya perangkat keras seperti yang terjadi pada Android.

Berdasarkan analisis fungsional penggunaan bahasa pemrograman swift pada studi kasus Bujang Kurir dan hasil analisis performa aplikasi, implementasi aplikasi bujang kurir berbasis iOS menggunakan perancangan aplikasi Android yang telah berjalan sebelumnya dimungkinkan untuk dilakukan.

DAFTAR PUSTAKA

- [1] Ratningsih, "Forecasting Penjualan Rumah Dengan Menggunakan Metode Trend Moment Pada PT.Rumakita Prima Karsa," *Perspektif*, vol. 15, no. 1, 2017.
- [2] H. Tama, "Perumahan Pusat Kota dengan Konsep Efisiensi di Pontianak," vol. 4, no. 1, 2016.
- [3] M. Munandar, *Budgeting: Perencanaan Kerja, Pengkoordinasian Kerja, Pengawasan Kerja*. Yogyakarta: BPFE Universitas Gajahmada, 2001.
- [4] N. Rukhansah, M. A. Muslim, & A. Riza, "Peramalan Harga Emas Menggunakan Fuzzy Time Series Markov Chain Model," *KOMPUTAKI*, vol. 1, no. 1, 2016.
- [5] M. N. Saleh, M. A. Irwansyah, & H. H. Anra, "Implementasi Peramalan Menggunakan Fuzzy Time Series pada Aplikasi Helpdesk Inventaris Perangkat Teknologi Informasi," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 1, no. 2, 2017.
- [6] M. Y. Fathoni, "Implementasi Metode Fuzzy Time Series Cheng untuk prediksi Konsentrasi Gas NO2 Di Udara," *Jurnal Sistem Informasi Bisnis*, 2017.
- [7] I. Halimi & W. A. Kusuma, "Prediksi Indeks Harga Saham Gabungan (IHSG) Menggunakan Algoritma Neural Network," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 4, no. 1, 2018.
- [8] J. Lin, E. Keogh, S. Lonardi, & B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003.
- [9] J. R. Hwang, S. M. Chen, & C. H. Lee, "Handling forecasting problems using fuzzy time series," *Fuzzy Sets and Systems*, vol. 100, no. 1-3, 1998.
- [10] E. A. Nugroho, "Sistem Pengendali Lampu Lalulintas Berbasis Logika Fuzzy," *Jurnal SIMETRIS*, vol. 8, no. 1, 2017.
- [11] I. G. A. D. Saryanti, "Perancangan Sistem Klasifikasi Tingkat Hipertensi Manusia dengan Metode Logika Fuzzy," *Proceeding Seminar Nasional Sistem Informasi dan Teknologi Informasi*, vol. 1, no. 1, 2018.
- [12] M. C. C. Utomo, W. F. Mahmudy, & S. Anam, "Kombinasi Logika Fuzzy dan Jaringan Syaraf Tiruan untuk Prakiraan Curah Hujan Timeseries di Area Puspo – Jawa Timur," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 3, 2017.
- [13] P. C. L. Silva, H. J. Sadaei, R. Ballini, & F. G. Guimaraes, "Probabilistic Forecasting With Fuzzy Time Series," *IEEE Transactions on Fuzzy Systems*, 2019.
- [14] N. Nikentari, M. Bettiza, & H. Sastypratiwi, "Prediksi Kecepatan Angin Menggunakan Adaptive Neuro Fuzzy (ANFIS) dan Radial Basis Function Neural Network (RBFNN)," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 4, no. 1, 2018.
- [15] P. Singh, "A Brief Review of Modeling Approaches Based on Fuzzy Time Series," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 2, 2017.
- [16] W. Kurniadi, "Pendukung Keputusan Dalam Peramalan Penjualan Ayam Broiler Dengan Metode Trend Moment Dan Simple Moving Average Pada CV. Merdeka Adi Perkasa," *Jurnal Media Informatika Budidarma*, vol. 2, no. 3, 2018.