

APLIKASI PERENCANA BELANJA DENGAN PENGURUTAN SKALA PRIORITAS BERBASIS ANDROID

Barry Nuqoba¹⁾, Faried Effendy²⁾

^{1,2)} Departemen Matematika, Fakultas Sains dan Teknologi Universitas Airlangga Surabaya
Email : barry.nu@gmail.com¹⁾, faried_effendy99@yahoo.com²⁾

ABSTRAK

Aplikasi Perencana Belanja merupakan aplikasi untuk perencanaan kegiatan belanja dengan membuat daftar belanja beserta daftar barang yang akan dibeli secara digital, sehingga Pengguna dapat menghemat waktu tanpa menulis lagi daftar beserta barangnya menggunakan kertas (*paperless*) karena penulisan menggunakan kertas sudah tidak lagi efisien. Aplikasi daftar belanja juga mampu melakukan perhitungan total belanja dari daftar barang dalam satu daftar belanja sehingga Pengguna dapat langsung mengetahui berapa banyak total belanja yang sedang dilakukan. Pengguna juga dapat mengurutkan daftar barang yang ada pada daftar belanja dengan berbagai kriteria.

Dalam penelitian pembangunan aplikasi ini dapat dianalisa dan dikembangkan menjadi beberapa fungsionalitas meliputi aplikasi mampu membuat daftar belanja beserta daftar barang yang akan dilaksanakan, pada daftar barang aplikasi mampu mengurutkan data dengan berbagai kriteria menggunakan algoritma pengurutan *selection sort*, mampu mengoptimalkan daftar belanja ketika melebihi anggaran, mampu mengelompokkan barang berdasarkan kategori yang dibuat, mampu membuat laporan sebagai tolak ukur aktifitas belanja Pengguna secara periodik.

Penelitian juga menghasilkan hitungan nilai validitas dari data yang diambil dari koresponden. Adapun hasil didapatkan prosentase kevalidan 70% sesuai dengan kebutuhan user dengan korelasi *Pearson* menggunakan tabel *r*, dan adapun hasil validasi dengan *Kappa (K) Cohen* diperoleh kesimpulan bahwa nilai $K=0.444$ masuk dalam kategori sedang.

Kata Kunci: *Perencana, Belanja, Android.*

PENDAHULUAN

Belanja merupakan bagian yang melekat pada diri manusia dalam kehidupan sehari-hari. Daftar belanja dapat diwujudkan dalam beberapa bentuk. Orang-orang kadang menampilkannya sebagai daftar tertulis sebelum berangkat belanja (*pre-planned*), atau dengan hanya mengingat saja tanpa tersusun terlebih dahulu (*impromptu*).

Perlu diperhatikan bahwa dengan mengingat, kecenderungan manusia untuk melupakan hal-hal yang ringan, selain itu orang juga bisa membeli barang dengan jumlah yang lebih dari perkiraan yang mengakibatkan pembengkakan anggaran (boros). Dan jika menggunakan daftar tertulis, terkadang orang lupa untuk mencatat harga dan jumlah barang yang dibeli. Daftar tertulis memiliki kekurangan yang dapat diatasi oleh daftar belanja digital (Malhotra, 2012).

Dalam era yang didukung oleh perangkat *mobile*, penggunaan daftar belanja secara tertulis sudah tidak lagi efisien untuk digunakan, oleh sebab itu dikembangkanlah aplikasi daftar belanja yang bertujuan untuk lebih mampu memajemen waktu berbelanja dan mempermudah orang untuk berbelanja dengan bijak memilih barang

yang diperlukan sehingga terhindar dari pemborosan anggaran (Praditya, 2013).

Setelah itu masalah muncul ketika seseorang sudah mendapatkan barang yang akan mereka beli, tanpa disadari sebelum menghitung ulang pengeluaran dan mereka hendak membayar ke kasir, akibatnya pengeluaran dari barang yang diambil dan anggaran yang disiapkan tidak sinkron (sama) maka yang terjadi adalah antrian kasir menjadi lama dan memanjang, tindakan inilah yang seharusnya dilakukan oleh Pelaku yaitu mengembalikan barang yang sudah diambil kembali ke tempat semula namun tidak demikian, orang-orang terkadang mengembalikan barang tersebut ke tempat yang tidak semestinya.

Yang diangkat dalam penelitian ini sebagai solusi untuk memajemen daftar belanja adalah dengan memberikan prioritas kepada setiap jenis barang yang diambil/dibeli. Dengan memberikan kategori prioritas diharapkan nanti Pembeli dapat mengira mana barang yang penting dan kurang penting, sehingga dapat mengurangi adanya proses pengembalian barang yang tidak semestinya. Prioritas barang ini nantinya akan dapat diurutkan berdasarkan prioritas tertinggi/terendah.

Adapun dalam manajemen pengurutan prioritas digunakan algoritma pengurutan (*sorting algorithm*). Daftar belanja akan diurutkan dalam urutan tertentu baik *ascending* maupun *descending*. Dari daftar belanja yang akan dibentuk setiap propertinya mempunyai tipe data yang berbeda, tipe data inilah yang nanti akan diurutkan (contoh: *string*, *integer*) (Beniwal & Grover, 2013). Algoritma *selection sort* digunakan karena data yang ada tidak terlalu besar dan *selection sort* merupakan salah satu algoritma pengurutan yang efisien untuk ukuran data yang ringan dengan kompleksitas yang rendah, kestabilan yang baik dan penggunaan memori yang ringan (Sareen, 2013).

LANDASAN TEORI

Algoritma Pengurutan

Algoritma pengurutan merupakan algoritma yang menempatkan unsur-unsur dari sebuah daftar dalam urutan tertentu. Secara umum algoritma pengurutan melakukan dua operasi seperti membandingkan dua elemen dan menetapkan nilai (Saniman & Fathoni, 2010). Permasalahan pengurutan (*sorting problem*) secara formal didefinisikan sebagai berikut. Input adalah suatu urutan dari n bilangan. Sedangkan output adalah suatu permutasi atau penyusunan kembali dari input sedemikian rupa sehingga pada tata urutan *ascending* (dari nilai kecil ke besar) atau pada tata urutan *descending* (dari nilai besar ke kecil). (Saptadi & Sari, 2012)

Selection Sort

Selection sort adalah algoritma pengurutan yang sangat mudah dipahami dan sangat berguna ketika berhadapan dengan sejumlah kecil data. (Jadoon, Solehria, Rehman, & Jan, 2011). Meskipun banyak melakukan perbandingan, tetapi dapat meminimalisir proses pertukaran (*swapping*). (Jadoon, Solehria, & Qayum, 2011).

Dalam penelitian ini, data yang digunakan berukuran relatif kecil. Beberapa algoritma pengurutan yang cocok digunakan untuk data berukuran kecil adalah *bubble sort*, *insertion sort*, *selection sort*. Dari ketiga algoritma tersebut yang paling stabil adalah *insertion sort*, namun proses pertukaran data yang terjadi dapat diminimalisir oleh algoritma *selection sort* (Tjaru, 2009).

Di dalam sebuah kasus pengurutan, sumberdaya yang digunakan untuk melakukan *swapping* itu sangat tinggi, *selection sort* memiliki sifat meminimalkan jumlah *swaps* (pertukaran data) (Yahya, 2014). Adapun pseudo code daripada *selection sort* akan dijabarkan di bawah (Khairullah M., 2013):

1. for $i=n-1$ to 0
2. $indexMax = 0$
3. for $j=1$ to i
 - if ($X[j] > X[indexMax]$) then
 - set $indexMax=j$
 - End if

4. Terbesar = $X[indexMax]$
5. $X[indexMax] = X[i]$
6. $X[i] =$ Terbesar

Adapun langkah kerja yang dapat dilihat dari algoritma *selection sort*, mengatur elemen *array* dalam posisi terurut dengan terlebih dahulu mencari nilai terkecil dari *array*. Tergantung pada bagaimana *array* disusun, kemudian tukar data (*swapping*) dengan elemen *array* yang ada di posisi yang benar. (Sutopo, 2011).

Di bawah ini adalah contoh proses pengurutan menggunakan algoritma *selection sort*. Berikut adalah angka-angka yang akan diurutkan secara *ascending* (dari kecil ke besar).

32, 4, 1, 12, 19

Proses pertama adalah dengan membandingkan angka 32 ke seluruh angka yang ada, dan tukarkan (*swap*) dengan angka terendah. Dalam proses ini, perulangan pertama adalah menukar angka 32 dengan 1, dimana angka 1 adalah angka terendah. Hasilnya dapat dilihat seperti di bawah.

1, 4, 32, 12, 19

Proses kedua dilakukan dengan membandingkan angka 4 sebagai elemen ke-2. Namun tidak ada elemen setelah angka 4 yang lebih kecil, jadi angka 4 akan tetap pada posisinya dan hasilnya tetap seperti berikut.

1, 4, 32, 12, 19

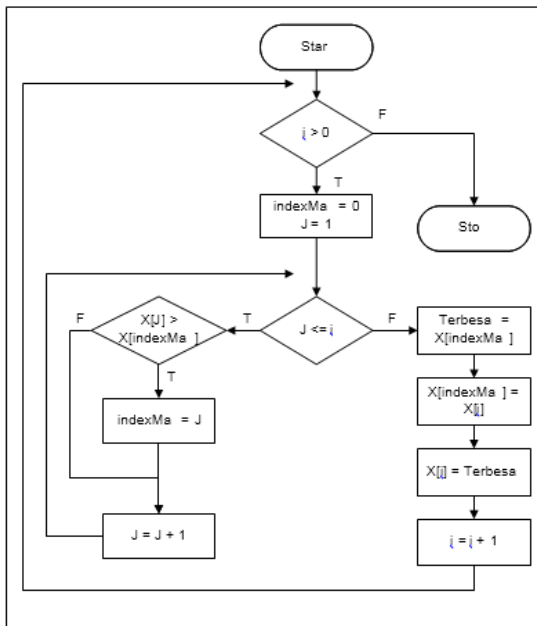
Proses ketiga membandingkan elemen ke-3 yaitu angka 32. Bersebelahan dengan angka 12 sebagai angka terkecil yang belum terurut, jadi proses berikutnya menukar 32 dengan 12. Hasilnya dapat dilihat di bawah.

1, 4, 12, 32, 19

Proses berikutnya adalah proses ke-4, dan angka 19 adalah angka terendah dan lebih kecil dari 32, maka dilakukan penukaran elemen dan akan menjadi pertukaran terakhir dalam proses pengurutan. Hasil pengurutan akan terlihat seperti berikut

1, 4, 12, 19, 32

Pseudocode algoritma *selection sort* di atas dapat digambarkan dengan lebih jelas menggunakan diagram alur kerja (*flowchart diagram*) sebagaimana yang tampak pada gambar 1 berikut.

Gambar 1 Flowchart proses *selection sort*

HASIL DAN PEMBAHASAN

Gambaran Umum Sistem

Hasil penelitian yang didapatkan adalah diimplementasikan sebuah aplikasi perencana belanja dengan pengurutan skala prioritas berbasis android yang memiliki beberapa fitur utama yaitu menyusun daftar rencana belanja, membuat skala prioritas barang belanjaan, menetapkan anggaran belanja. Pengurutan skala prioritas menggunakan algoritma sederhana yaitu *selection sort* karena jumlah data belanja relatif sedikit.

Berikut adalah tampilan awal serta beberapa tampilan utama dari aplikasi.



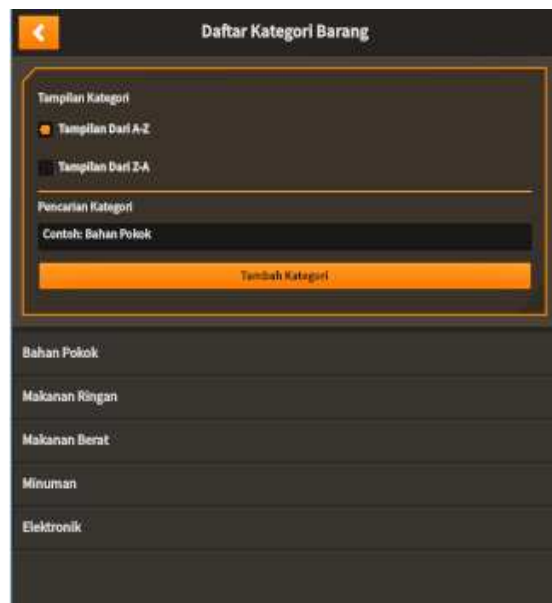
Gambar 2. Tampilan Awal

Pada tampilan awal sebagaimana yang terlihat pada Gambar 2, terdapat tiga menu utama yaitu menu Daftar Belanja untuk membuat daftar belanja dan daftar barang yang akan dibeli pada tiap daftar belanja. Selanjutnya adalah menu Kategori untuk menambahkan kategori barang, serta menu Laporan untuk membuat laporan pengeluaran belanja per-bulan.



Gambar 3. Daftar Belanja

Setelah masuk ke menu daftar belanja maka muncul semua daftar belanja yang sudah pernah dibuat sebagaimana terlihat pada Gambar 3. Pengguna dapat melihat detail belanja, melihat skala prioritas barang, melihat anggaran belanja, serta menambahkan daftar belanja baru melalui halaman ini.



Gambar 4. Menu Kategori

Setelah masuk ke menu Kategori maka muncul semua kategori barang yang sudah pernah dibuat sebagaimana terlihat pada Gambar 4. Pengguna dapat melihat semua kategori yang ada, mengurutkan kategori, mencari kategori, serta menambahkan kategori baru melalui halaman ini.

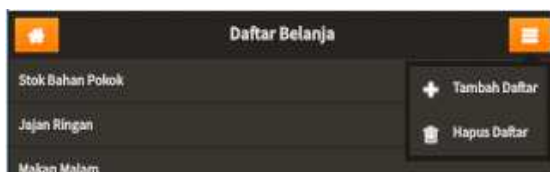


Gambar 5. Menu Laporan

Setelah masuk ke menu Laporan maka muncul semua laporan yang sudah pernah dibuat sebagaimana terlihat pada Gambar 5. Pengguna dapat melihat semua laporan berdasarkan periode tertentu, harian, maupun berdasar kategori barang tertentu. Selain itu pengguna juga dapat menambahkan laporan baru melalui halaman ini.

Pengujian

Pengujian dilakukan terhadap fitur utama aplikasi yaitu membuat perencanaan daftar belanja. Langkah awal untuk melakukan penambahan daftar belanja adalah dengan masuk ke menu daftar belanja yang ada pada halaman utama. Setelah memasuki halaman daftar belanja sebagaimana yang terlihat pada Gambar 3, pengguna menekan sub menu Tambah Daftar pada navigasi kanan *header* dan selanjutnya akan tampil opsi Tambah Daftar dan Hapus Daftar sebagaimana terlihat pada gambar 6.



Gambar 6. Menu Laporan

Setelah memilih opsi Tambah Daftar maka akan muncul tampilan untuk menambah daftar belanja baru sebagaimana terlihat pada Gambar 7 berikut.



Gambar 7. Halaman Tambah Daftar Belanja

Setelah mengisi semua data belanja dan pengguna menekan tombol Selesai maka daftar belanja baru tersimpan ke Database. Selanjutnya tampilan akan kembali ke halaman Daftar Belanja yang menampilkan semua daftar belanja yang sudah ada, termasuk daftar belanja yang baru saja dibuat. Gambar 8 berikut menunjukkan bahwa daftar belanja yang dibuat telah berhasil disimpan.



Gambar 8. Daftar Belanja Berhasil Tersimpan

KESIMPULAN

Aplikasi perencanaan belanja dapat digunakan untuk merencanakan kegiatan belanja dengan baik. Adanya fitur skala prioritas membuat barang dengan skala prioritas yang lebih tinggi diutamakan untuk dibeli terlebih dahulu apabila terjadi kenaikan harga barang. Belanja menjadi lebih terkontrol dan sesuai kebutuhan dengan adanya aplikasi perencanaan belanja. Selain itu, fitur Laporan juga memudahkan pengguna dalam mengevaluasi belanja yang telah dilakukan, serta merencanakan belanja ke depan.

DAFTAR PUSTAKA

- [1]. Beniwal, S., & Grover, D. (2013). Comparison Of Various Sorting Algorithms: A review. *International Journal of Emerging Research in Management & Technology*, 1-2.
- [2]. Jadoon, S., Solehria, S. F., & Qayum, M. (2011). Optimized Selection Sort Algorithm is faster than Insertion Sort Algorithm: a Comparative Study. *International Journal of Electrical & Computer Sciences IJECS-IJENS*, 2.
- [3]. Jadoon, S., Solehria, S. F., Rehman, P. U., & Jan, P. (2011). Design and Analysis of Optimized Selection Sort Algorithm. *International Journal of Electric & Computer Sciences IJECS-IJENS*, 1-3.
- [4]. Khairullah, M. (2013). Enhancing Worst Sorting Algorithms. *International Journal of Advanced Science and Technology*, 3.
- [5]. Malhotra, B. (2012). Analysis of Shopping List Apps for Android and iPhone. 3-4.
- [6]. Praditya, P. L. (2013, 6 3). Aplikasi Perencanaan Belanja Berbasis Android. *Naskah Publikasi*. Yogyakarta, Jawa Tengah, Indonesia: STMIK AMIKOM Yogyakarta.
- [7]. Saniman, & Fathoni, M. (2010). Konsep Sorting dalam Pemrograman. *SAINTIKOM*, 1.
- [8]. Saptadi, A. H., & Sari, D. W. (2012). Analisis Algoritma Insertion Sort, Merge Sort Dan Implementasinya Dalam Bahasa Pemrograman C++. *Infotel*, 1-2.
- [9]. Sareen, P. (2013). Comparison of Sorting Algorithms (On the Basis of Average Case). *International Journal of Advanced Research in Computer Science and Software Engineering*, 1-3.
- [10]. Sutopo, H. (2011). Selection Sorting Algorithm Visualization Using Flash. *The International Journal of Multimedia & Its Applications (IJMA)*, 3.
- [11]. Tjaru, S. N. (2009). Kompleksitas Algoritma Pengurutan Selection Sort dan Insertion Sort. *IF2091 STRATEGI ALGORITMIK*, 1-5.