

Aplikasi Metode Simple Additive Weighting (SAW) Dalam Pengembangan Sistem Pencarian Toko Batik Berbasis Android

Hafid Iqbalgis⁽¹⁾, Nurochman⁽²⁾

Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga
Jalan Laksda Adisucipto, Yogyakarta 55281

e-mail : hafidqbalgis@gmail.com⁽¹⁾, nurochman@uin-suka.ac.id⁽²⁾

Abstract

Information is an important thing at this time. One kind of information is a symbol of public information. Researchers conducted research to find a batik shop using the Simple Additive Weighting method by using various criteria. The research that will be conducted is to create a decision support system to choose a batik shop using the Simple Additive Weighting method. In its calculations, the criteria considered are distance, price, products, ease of transportation, comfort, cleanliness and safety of the store. The results of this study indicate that the system built is able to provide batik shop recommendations in accordance with the desired criteria.

Keywords : Android, Decision Support System, Simple Additive Weighting, Extreme Programming.

Abstrak

Informasi merupakan hal yang sangat diperlukan pada saat ini. Salah satu bentuk informasi adalah simbol informasi publik. Peneliti melakukan penelitian untuk mencari toko batik dengan menggunakan metode Simple Additive Weighting dengan menggunakan berbagai kriteria. Penelitian yang akan dilakukan adalah membuat suatu sistem pendukung keputusan untuk memilih toko batik dengan menggunakan metode Simple Additive Weighting. Dalam perhitungannya, kriteria yang dipertimbangkan adalah jarak, harga, produk, kemudahan transportasi, kenyamanan, kebersihan dan keamanan toko. Hasil penelitian ini menunjukkan bahwa sistem yang dibangun mampu memberikan rekomendasi toko batik sesuai dengan kriteria yang diinginkan.

Kata Kunci : Android, Sistem Pendukung Keputusan, Simple Additive Weighting, Extreme Programming.

1. PENDAHULUAN

Keberadaan lokasi toko yang begitu banyak membuat pelanggan terkadang merasa bingung untuk memilih toko yang sesuai dengan keinginan mereka. Hal ini menyebabkan pelanggan yang terdiri dari masyarakat yang tinggal disekitar kota Pekalongan maupun pelanggan dari luar Kota Pekalongan yang masih bingung mencari lokasi toko batik, membutuhkan rekomendasi untuk memutuskan mengunjungi lokasi toko batik yang sesuai dengan keinginan mereka. Untuk itu diperlukan suatu sistem pendukung pengambil keputusan yang dapat membantu pelanggan dalam memberikan rekomendasi lokasi toko batik dengan menerapkan metode SAW (Simple Additive Weighting). Metode ini menggunakan model perangsangan yang menampilkan lokasi toko batik yang telah direkomendasikan oleh sistem dengan menggunakan bobot kriteria yang telah ditentukan oleh sistem.

Sistem ini juga memanfaatkan integrasi Google Maps API untuk menampilkan lokasi toko batik. Diharapkan dengan adanya sistem ini dapat membantu pelanggan dalam memperoleh rekomendasi lokasi toko batik yang sesuai keinginan pelanggan.

Agar sistem ini lebih mudah diakses, maka sistem ini dibuat dengan berbasiskan sistem operasi android. Dengan berbasiskan sistem operasi android, maka sistem ini dapat diakses dari berbagai tempat dan tanpa batasan waktu selagi device pengguna memiliki akses internet.

2. METODE PENELITIAN

Metode *Agile* atau *Agile Software Development* adalah sekumpulan metodologi pengembangan perangkat lunak yang memerlukan adaptasi secara cepat dari pengembang terhadap perubahan dalam bentuk apapun. Dalam penelitian ini penulis menerapkan metode *Agile* karena sangat sesuai dengan pengembangan sistem nantinya yang membutuhkan proses adaptasi yang cepat dan fleksibel dengan perubahan-perubahan secara mendadak.

2.1. Extreme Programming

Adalah sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel. XP bukan hanya berfokus pada coding tetapi meliputi seluruh area pengembangan perangkat lunak. XP mengambil pendekatan 'ekstrim' dalam *iterative development*. XP merupakan sebuah gaya pengembangan perangkat lunak yang berfokus terhadap aplikasi yang sangat baik dari sisi teknik pemrograman, komunikasi yang jelas, dan kerjasama tim yang memungkinkan untuk mencapai hal-hal yang bahkan sebelumnya tidak akan terduga (Beck & Andres, 2004).

Extreme Programming terdiri dari aktifitas perencanaan (*planning*), desain (*design*), pengkodean (*coding*), dan pengujian (*testing*). Dan keterangan dari setiap tahapannya adalah sebagai berikut (Sutabri, 2012):

1. *Planning* Tahapan ini dimulai dengan membuat *user story* atau gambaran dari fitur serta fungsi dari sistem yang akan dibangun.
2. *Design* Aktifitas desain dalam pengembangan sistem bertujuan untuk merancang pola logika dalam sistem. Desain dibuat berdasarkan *user story* sebelumnya.
3. *Coding* Tahap pembuatan sistem berdasarkan desain yang telah dibuat. Tahap ini dapat dilakukan secara berulang-ulang (*refactoring*) apabila terdapat koreksi dari tahap berikutnya.
4. *Testing* Pada tahap pengujian sistem, setiap modul yang sedang dikembangkan akan terlebih dahulu mengalami pengujian. Apabila masih belum sesuai dengan permintaan maka akan dilakukan perbaikan pada bagian yang dikoreksi. Jika sudah sesuai dengan permintaan maka sistem sudah dapat diimplementasikan.

2.2. Unified Modeling Language (UML)

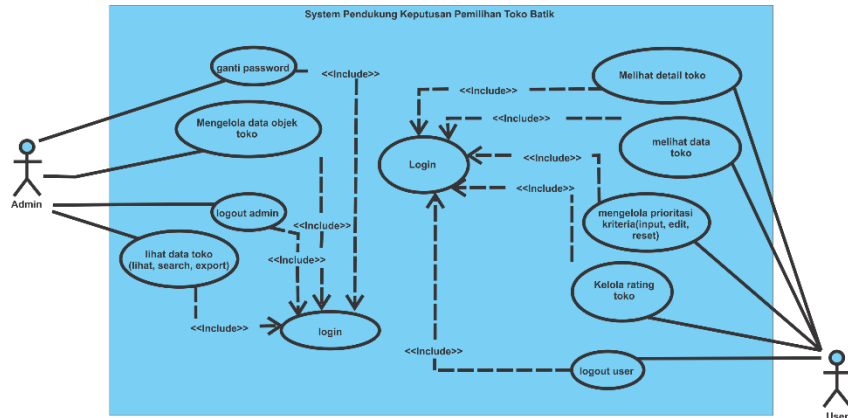
UML merupakan suatu pemodelan yang digunakan untuk menggambarkan sebuah sistem software yang terkait dengan obyek (Whitten, Bentley, & Dittman, 2007). Oleh sebab itu UML merupakan salah satu modeling yang tepat untuk perancangan pengembangan sistem yang berorientasi pada obyek.

3. HASIL DAN PEMBAHASAN

Penggunaan *framework Laravel* untuk menyimpan datanya dan *Android studio* untuk mobilennya, pada pengembangan sistem ini membuat penulis memilih permodelan UML untuk perancangannya.

3.1 Use Case

Use Case adalah pemodelan yang bertujuan untuk menggambarkan *behaviour* dari sistem. *Use Case* menggambarkan aktor yang nantinya akan ikut andil dalam sistem dan fungsi yang ada dalam sistem.



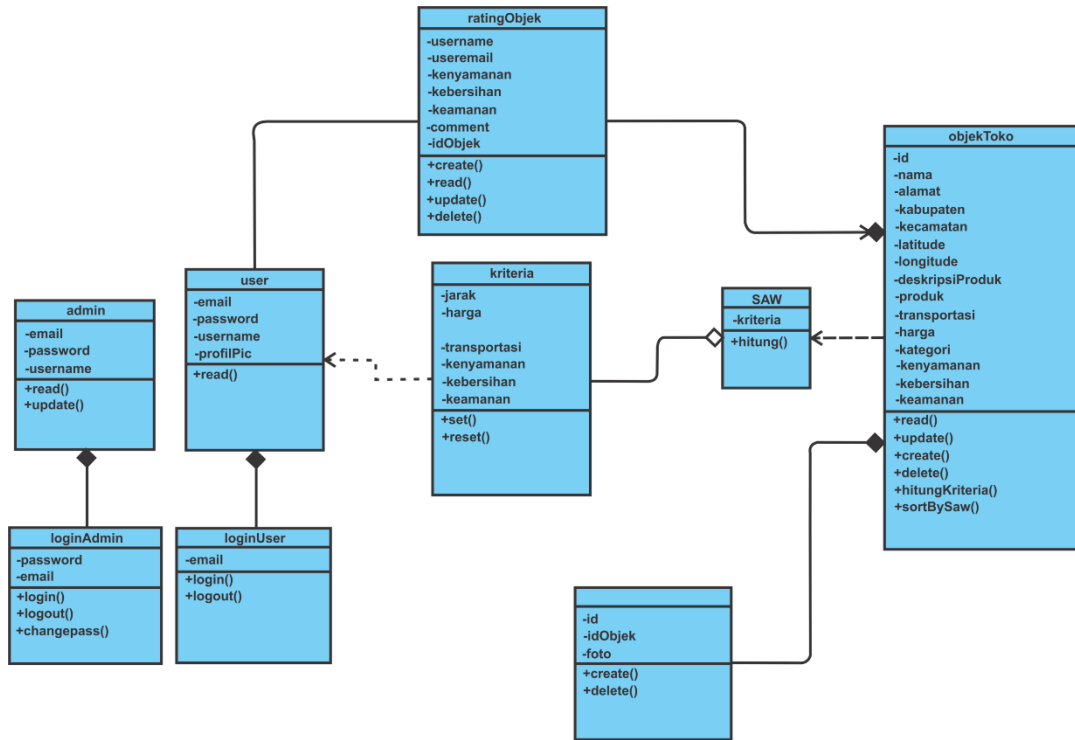
Gambar 1 Use Case Diagram

Table 1 Deskripsi dan tugas Aktor

No	Aktor	Hak Akses
1.	Admin	Aktor yang memiliki wewenang untuk mengelola data toko, melihat detail mengupdate, menghapus dan menamba toko
2.	User	Aktor yang memiliki wewenang untuk memberikan ulasan atau rating pada toko.

3.2 Class Diagram

Class Diagram adalah pemodelan untuk menggambarkan class-class yang akan digunakan untuk membangun sebuah sistem. Melalui class diagram nantinya programmer dapat mengembangkan sistem sesuai perencanaan yang sudah didokumentasikan.



Gambar 2 Class Diagram

3.3 Implementasi

Pada metode yang digunakan yaitu metode *extreme programming* terdapat 4 tahap dalam pengembangan, yaitu *planning*, *design*, *coding*, dan *testing*.

Langkah – langkah metode Simple Additive Weighting

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_j
2. Mengisi nilai setiap kriteria untuk setiap alternatif.
3. Membuat matriks keputusan berdasar kriteria, kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R .
4. Hasil akhir diperoleh dari setiap proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik sebagai solusi.

Contoh hitungan menggunakan Simple Additive Weighting

Semua bobot 20% atau 0.20

Alternatif	Kriteria						
	Transportasi	Produk	Kenyamanan	Kebersihan	Keamanan	Harga	Jarak
A1	4	5	3	3	3	4	5
A2	4	4	4	4	4	4	4
A3	5	1	1	2	2	5	4
A4	2	1	4	5	3	3	3
A5	4	3	2	4	5	2	1
A6	3	2	5	1	1	4	3

$$X = \begin{bmatrix} 4 & 5 & 3 & 3 & 3 & 4 & 5 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 1 & 1 & 2 & 2 & 5 & 4 \\ 2 & 1 & 4 & 5 & 3 & 3 & 3 \\ 4 & 3 & 2 & 4 & 5 & 2 & 1 \\ 3 & 2 & 5 & 1 & 1 & 4 & 3 \end{bmatrix}$$

Normalisasi kriteria transportasi, produk, kenyamanan, kebersihan, keamanan

$$\text{Rumus} : r_{ij} = \frac{X_{ij}}{\max X_{ij}}$$

• Alternatif 1 : transportasi

$$r_{11} = \left(\frac{4}{5}\right) = 0.8$$

$$r_{21} = \left(\frac{4}{5}\right) = 0.8$$

$$r_{31} = \left(\frac{5}{5}\right) = 1.00$$

$$r_{41} = \left(\frac{2}{5}\right) = 0.4$$

$$r_{51} = \left(\frac{4}{5}\right) = 0.8$$

$$r_{61} = \left(\frac{3}{5}\right) = 0.6$$

• Alternatif 2 : produk

$$r_{12} = \left(\frac{5}{5}\right) = 1.00$$

$$r_{22} = \left(\frac{4}{5}\right) = 0.8$$

$$r_{32} = \left(\frac{1}{5}\right) = 0.2$$

$$r_{42} = \left(\frac{1}{5}\right) = 0.2$$

$$r_{52} = \left(\frac{3}{5}\right) = 0.6$$

$$r_{62} = \left(\frac{2}{5}\right) = 0.4$$

• Alternatif 3 : kenyamanan

$$\bullet r_{13} = \left(\frac{3}{5}\right) = 0.6$$

$$\bullet r_{23} = \left(\frac{4}{5}\right) = 0.8$$

$$\bullet r_{33} = \left(\frac{1}{5}\right) = 0.2$$

$$\bullet r_{43} = \left(\frac{4}{5}\right) = 0.8$$

$$\bullet r_{53} = \left(\frac{2}{5}\right) = 0.4$$

$$\bullet r_{63} = \left(\frac{5}{5}\right) = 1.00$$

• Alternatif 4 : kebersihan

$$\bullet r_{14} = \left(\frac{3}{5}\right) = 0.6$$

$$\bullet r_{24} = \left(\frac{4}{5}\right) = 0.8$$

$$\bullet r_{34} = \left(\frac{2}{5}\right) = 0.4$$

$$\bullet r_{44} = \left(\frac{5}{5}\right) = 1.00$$

$$\bullet r_{54} = \left(\frac{4}{5}\right) = 0.8$$

$$\bullet r_{64} = \left(\frac{1}{5}\right) = 0.2$$

• Alternatif 5 : keamanan

$$\bullet r_{15} = \left(\frac{3}{5}\right) = 0.6$$

$$\bullet r_{25} = \left(\frac{4}{5}\right) = 0.8$$

$$\bullet r_{35} = \left(\frac{2}{5}\right) = 0.4$$

$$\bullet r_{45} = \left(\frac{3}{5}\right) = 0.6$$

$$\bullet r_{55} = \left(\frac{5}{5}\right) = 1.00$$

$$\bullet r_{65} = \left(\frac{1}{5}\right) = 0.2$$

Normalisasi kriteria Harga dan Jarak

$$\text{Rumus} : r_{ij} = \frac{\min X_{ij}}{X_{ij}}$$

• Alternatif 6 : harga

$$r_{16} = \left(\frac{2}{4}\right) = 0.5$$

$$r_{26} = \left(\frac{2}{4}\right) = 0.5$$

$$r_{36} = \left(\frac{2}{5}\right) = 0.4$$

$$r_{46} = \left(\frac{2}{3}\right) = 0.66$$

$$r_{56} = \left(\frac{2}{2}\right) = 1.00$$

$$r_{66} = \left(\frac{2}{4}\right) = 0.5$$

• Alternatif 7 : jarak

$$r_{17} = \left(\frac{1}{5}\right) = 0.2$$

$$r_{27} = \left(\frac{1}{4}\right) = 0.25$$

$$r_{37} = \left(\frac{1}{4}\right) = 0.25$$

$$r_{47} = \left(\frac{1}{3}\right) = 0.33$$

$$r_{57} = \left(\frac{1}{1}\right) = 1.00$$

$$r_{67} = \left(\frac{1}{3}\right) = 0.33$$

Matrik Ternormalisasi R

$$R = \begin{bmatrix} 0.8 & 1.00 & 0.6 & 0.6 & 0.6 & 0.5 & 0.2 \\ 0.8 & 0.8 & 0.8 & 0.8 & 0.8 & 0.5 & 0.25 \\ 1.00 & 0.2 & 0.2 & 0.4 & 0.4 & 0.4 & 0.25 \\ 0.4 & 0.2 & 0.8 & 1.00 & 0.6 & 0.66 & 0.33 \\ 0.8 & 0.6 & 0.4 & 0.8 & 1.00 & 1.00 & 1.00 \\ 0.6 & 0.4 & 1.00 & 0.2 & 0.2 & 0.5 & 0.33 \end{bmatrix}$$

Proses Perangkingan

$$V1 : (0.20)(0.8)+(0.20)(1.00)+(0.20)(0.6)+(0.20)(0.6)+(0.20)(0.6)+(0.20)(0.5)+(0.20)(0.2) = 0.86$$

$$V2 : (0.20)(0.8)+(0.20)(0.8)+(0.20)(0.8)+(0.20)(0.8)+(0.20)(0.8)+(0.20)(0.5)+(0.20)(0.25) = 0.95$$

$$V3 : (0.20)(1.00)+(0.20)(0.2)+(0.20)(0.2)+(0.20)(0.4)+(0.20)(0.4)+(0.20)(0.4)+(0.20)(0.25) = 0.57$$

$$V4 : (0.20)(0.4)+(0.20)(0.2)+(0.20)(0.8)+(0.20)(1.00)+(0.20)(0.6)+(0.20)(0.66)+(0.20)(0.33) = 0.798$$

$$V5 : (0.20)(0.8)+(0.20)(0.6)+(0.20)(0.4)+(0.20)(0.8)+(0.20)(1.00)+(0.20)(1.00)+(0.20)(1.00) = 1.12$$

$$V6 : (0.20)(0.6)+(0.20)(0.4)+(0.20)(1.00)+(0.20)(0.2)+(0.20)(0.2)+(0.20)(0.5)+(0.20)(0.33) = 0.646$$

$$V1 : 0,86 \quad \rightarrow 3$$

$$V2 : 0,95 \quad \rightarrow 2$$

$$V3 : 0,57 \quad \rightarrow 6$$

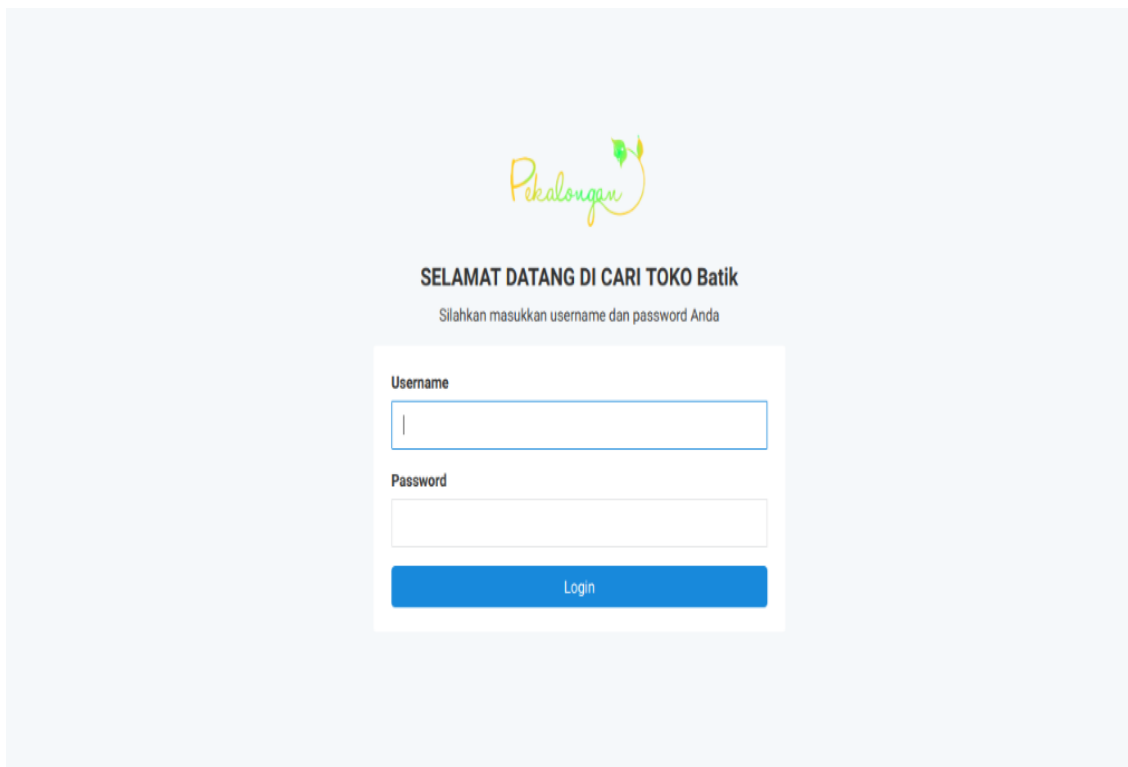
$$V4 : 0,798 \quad \rightarrow 4$$

$$V5 : 1,12 \quad \rightarrow 1$$

$$V6 : 0,646 \quad \rightarrow 5$$

```
//proses normalisasi matrix
for (index in allPlace.indices) {
    allPlace.get(index).setCriteria()
    normalisasiHarga = allPlace.get(index).criteriaModel.harga.toDouble().min()
    normalisasiJarak = allPlace.get(index).criteriaModel.jarak.toDouble().min()
    normalisasiTransportasi = allPlace.get(index).criteriaModel.transportasi.toDouble().max()
    normalisasiProduk = allPlace.get(index).criteriaModel.produk.toDouble().max()
    normalisasiKebersihan = allPlace.get(index).criteriaModel.kebersihan.toDouble().max()
    normalisasiKeamanan = allPlace.get(index).criteriaModel.keamanan.toDouble().max()
    normalisasiKenyamanan = allPlace.get(index).criteriaModel.kenyamanan.toDouble().max()
}
for (index in allPlace.indices) {
    val newJarak = allPlace.get(index).criteriaModel.jarak / normalisasiJarak
    allPlace.get(index).criteriaModel.jarak = newJarak.toDouble()
    allPlace.get(index).criteriaModel.harga = allPlace.get(index).criteriaModel.harga / normalisasiHarga
    allPlace.get(index).criteriaModel.transportasi = allPlace.get(index).criteriaModel.transportasi / normalisasiTransportasi
    allPlace.get(index).criteriaModel.produk = allPlace.get(index).criteriaModel.produk / normalisasiProduk
    allPlace.get(index).criteriaModel.keamanan = allPlace.get(index).criteriaModel.keamanan / normalisasiKeamanan
    allPlace.get(index).criteriaModel.kenyamanan = allPlace.get(index).criteriaModel.kenyamanan / normalisasiKenyamanan
    allPlace.get(index).criteriaModel.kebersihan = allPlace.get(index).criteriaModel.kebersihan / normalisasiKebersihan
}
```

```
//Perangkingan  
allPlace.get(index).criteriaModel.jarak += searchCriteria.jarak  
allPlace.get(index).criteriaModel.harga += searchCriteria.harga  
allPlace.get(index).criteriaModel.transportasi += searchCriteria.transportasi  
allPlace.get(index).criteriaModel.produk += searchCriteria.produk  
allPlace.get(index).criteriaModel.kenyamanan += searchCriteria.kenyamanan  
allPlace.get(index).criteriaModel.kebersihan += searchCriteria.kebersihan  
allPlace.get(index).criteriaModel.keamanan += searchCriteria.keamanan
```



Pekalongan

SELAMAT DATANG DI CARI TOKO Batik

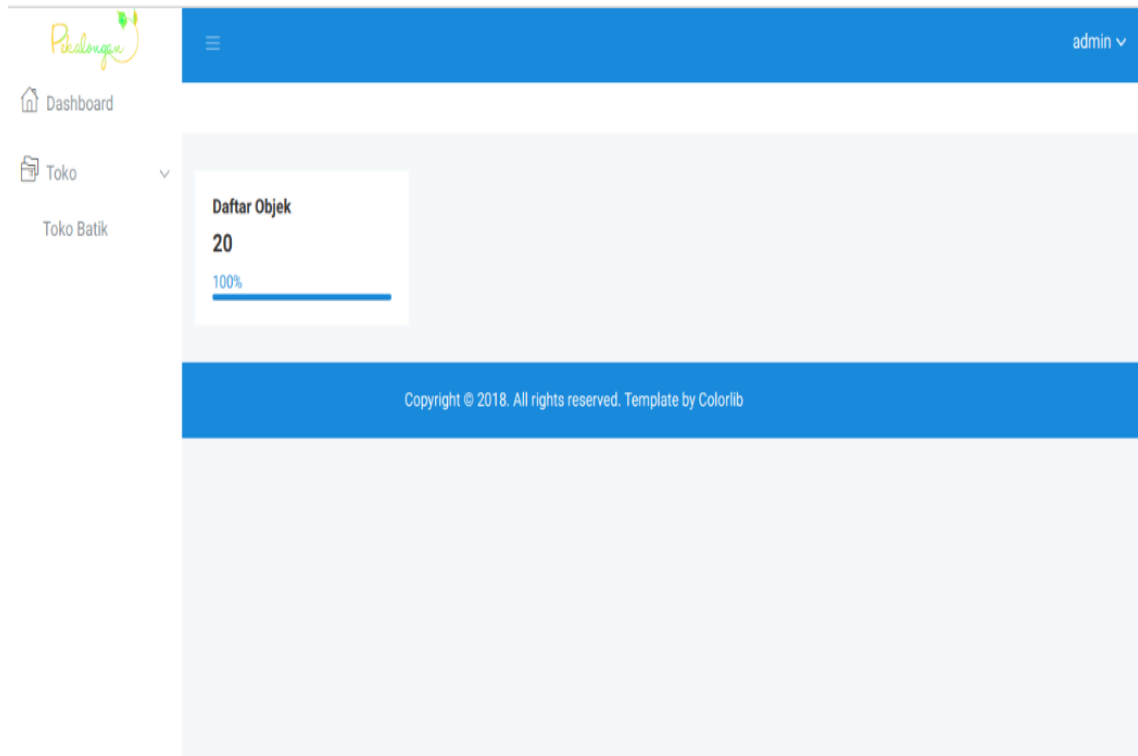
Silahkan masukkan username dan password Anda

Username

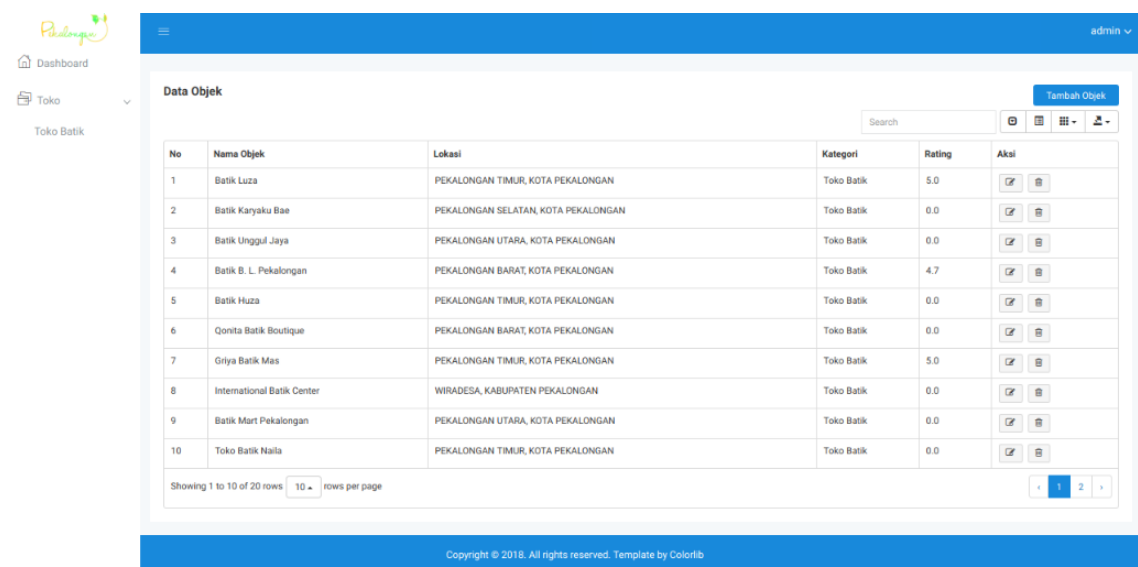
Password

Login

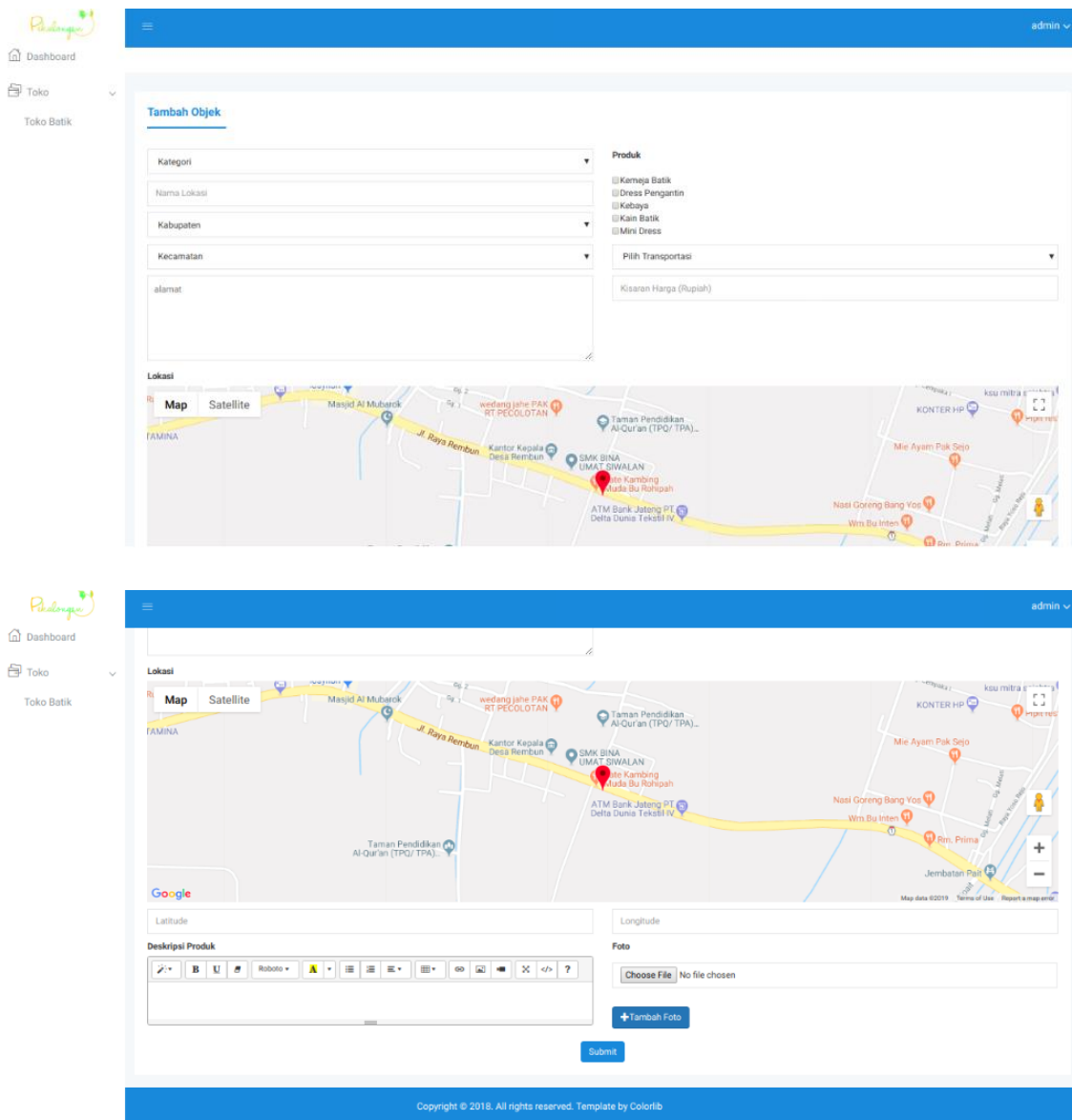
Gambar 3 Implementasi Login Admin



Gambar 4 Implementasi Home Admin

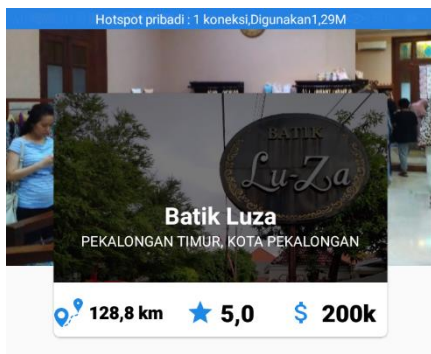
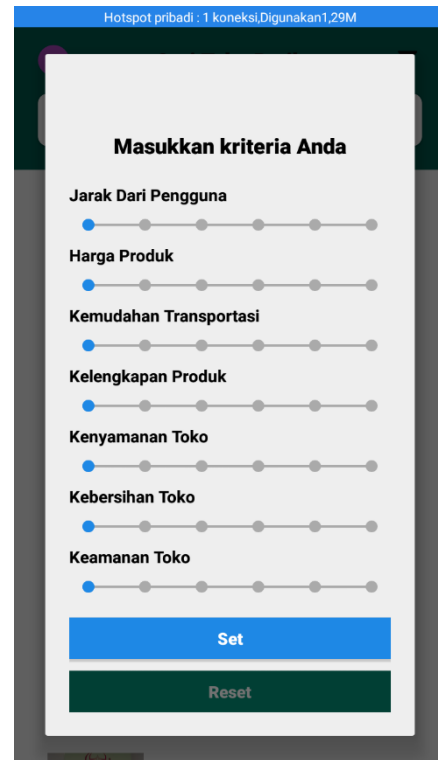
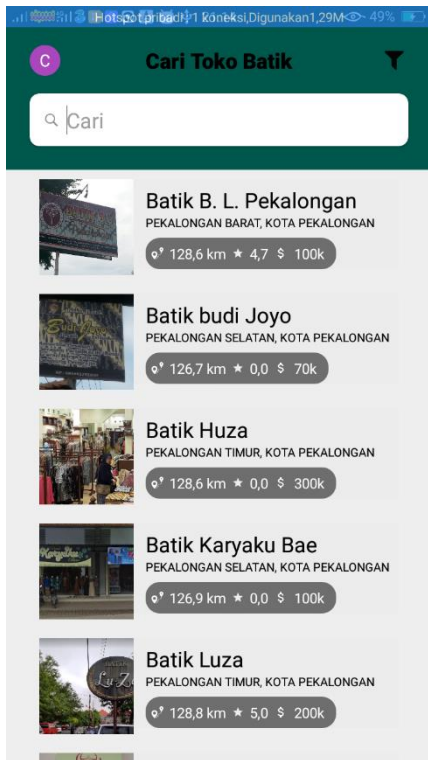


Gambar 5 Implementasi Data Toko



Gambar 6 Implementasi Tambah Toko

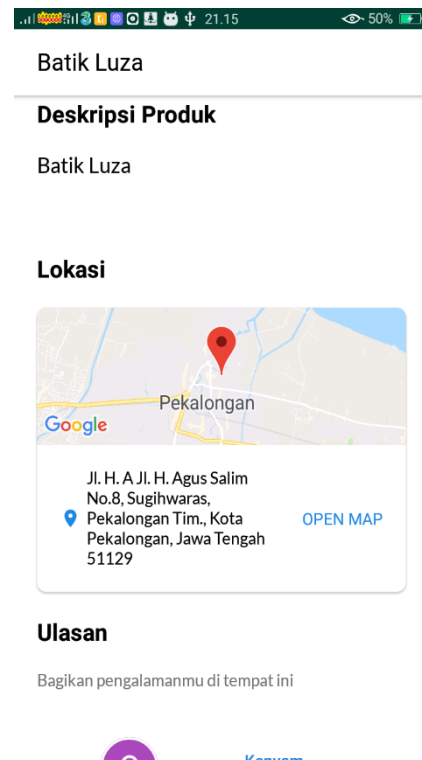
Implementasi User

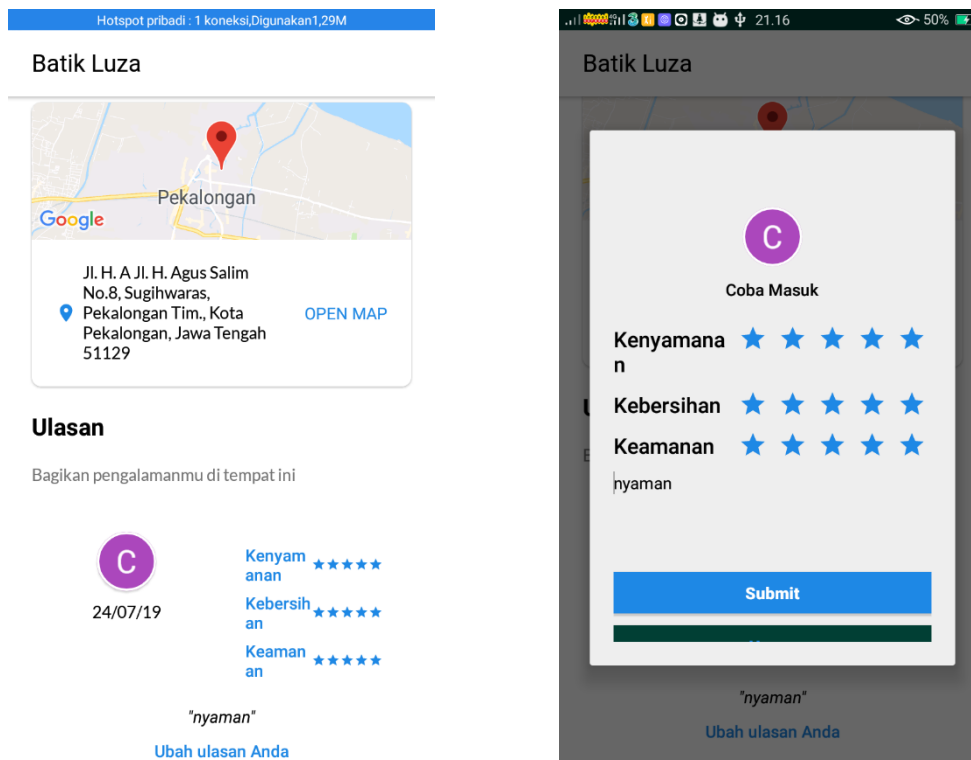


Galeri



Produk





3.4 PENGUJIAN

Dari pengujian yang melibatkan beberapa responden, yang terdiri pengunjung toko, melalui uji usability tentang efektifitas dan efisiensi sistem didapat hasil sebesar 52% setuju dan sangat setuju pada angka 40% serta sisa 8% pada nilai tidak setuju dari total responden untuk usability. Pada uji fungsionalitas sistem didapat hasil 100% berjalan dengan baik dari seluruh responden.

KESIMPULAN

Dalam penelitian ini, penggunaan metode Extreme Programming pada pengembangan sistem dan menggunakan metode Simple Additive Weighting sebagai pengolah data, menghasilkan dua tahap pengembangan. Pada setiap tahap penulis melibatkan pihak terkait (user) untuk mengetahui keberhasilan pengembangan sistem dalam menghasilkan sistem yang sesuai dengan kebutuhan dan tujuan awal penelitian. Pengembangan akan dinyatakan berhasil dan selesai apabila pengujian yang melibatkan user merasa puas dan tidak memberikan koreksi lagi terhadap sistem.

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Sistem yang dibangun berhasil menerapkan metode Simple Additive weighting untuk melakukan perancangan toko berbasis android.
2. Sistem Pendukung Keputusan Pemilihan Toko Batik telah berhasil dibangun, yang mampu memberikan user (pengunjung) rekomendasi toko batik dengan 7 kriteria yaitu harga, jarak lokasi dari pengguna, kemudahan transportasi, kelengkapan produk, kenyamanan toko, keamanan toko dan kebersihan toko.

DAFTAR PUSTAKA

- Bunafit, N. (2005). Database relasional dengan mysql. Andi, Yogyakarta.
- Fortina, D. R. (2012). Pembuatan aplikasi pemesan tiket bus pada po sinar dempo berbasis android.
- Hariyanto, B. (2004). Sistem manajemen basis data: Pemodelan, perancangan, dan terapannya. Informatika, Bandung.
- Hartini. (2013). Sistem pendukung keputusan pemilihan hotel di kota Palembang dengan metode simple additive weighting.
- Kadir, A. (2003). Konsep dan tuntunan praktis basis data. Andi, Yogyakarta.
- Kadir,A. (2008). Dasarpemrogramanwebdinamismenggunakanphp. AndiOffset, Yogyakarta.
- Kusumadewi, d., Sri. (2006). Fuzzy-attribute decision making(fuzzy madm). Graha Ilmu, Yogyakarta.
- Ladjamudin, A.-B. (2005). Analisis dan desain sistem informasi. Graha Ilmu, Yogyakarta.
- Rudianto, A. M. (2011). Pemrograman web dinamis menggunakan php dan mysql. C.V ANDI OFFSET. Yogyakarta.
- Sparague, R. H., & J, W. H. (1993). Decision support systems: Putting theory into practice.
- Turban, E. (2015). Decision support system and intelligent system 7th edition. Andi, Yogyakarta.
- Whitten, Jeffery, Bentley, Lonnie, Dittman, & Kevin. (2004). Metode desain dan analisis sistem. Terjemahan oleh Tim Penerjemah ANDI. 2004. ANDI:Yogyakarta.