



JURNAL TEKNIK INFORMATIKA

Halaman Jurnal: <http://jurnal.stmik-dci.ac.id/index.php/jutekin/>

Halaman LPPM STMIK DCI : <http://lppm.stmik-dci.ac.id/>



IMPLEMENTASI TEKNOLOGI *CROSSPLATFORM* DALAM PENENTUAN PROGRAM STUDI MENGGUNAKAN METODE *WEIGHTED PRODUCT*

Bambang Mohammad Azhari

Prodi Teknik Informatika, STMIK DCI

Email : mohammad.azhari12@gmail.com

ABSTRAK

Terdapat banyak masalah didunia salah satunya pada bidang bimbingan dan konsultasi. Disebagian tempat konsultasi pemilihan program studi bagi lulusan siswa SMA/SMK/Sederajat yang akan melanjutkan ke jenjang pendidikan lebih tinggi masih dirasa belum terkomputasi, konsultasi tersebut masih berupa *face-to-face* dalam artian masih membutuhkan sumber daya manusia dalam menentukan program studi yang cocok untuk siswa tersebut. Dan semakin berkembangnya teknologi ini menghasilkan beberapa macam teknologi salah satunya *Crossplatform*. Penulis berinisiatif dalam membangun aplikasi prediksi program studi yang penulis harapkan bisa mempermudah serta mempercepat penentuan program studi bagi siswa yang akan melanjutkan ke jenjang pendidikan yang lebih tinggi.

Kata Kunci: *Crossplatform, Weighted Product, React Native, Prediksi Program Studi*

I. PENDAHULUAN

Dizaman Teknologi ini, Teknologi berkembang secara pesat dan simultan. Dengan berkembangnya Zaman dan teknologi tersebut, keingingan manusia untuk menciptakan teknologi yang bersifat mempermudah menjadi lebih tinggi, Contoh kecilnya adalah dengan adanya Gadget yang saat ini banyak digunakan oleh Mayoritas masyarakat Terutama Gadget (Smartphone) Berbasis Android dan iOS. Disebagian tempat konsultasi pemilihan program studi bagi lulusan siswa SMA/SMK/Sederajat Sebagai bahan perbandingan. Jika seorang staff membutuhkan waktu $\pm 5-$

yang akan melanjutkan ke jenjang pendidikan lebih tinggi masih dirasa belum terkomputasi, konsultasi tersebut masih berupa *face-to-face* dalam artian masih membutuhkan sumber daya manusia dalam menentukan program studi yang cocok untuk siswa tersebut. Selain daripada itu ada masalah lain yang dihadapi yakni staff bagian konseling akan terasa membutuhkan sumberdaya lebih jika banyak siswa yang konsultasi dikarenakan manualnya dalam penentuan Program Studi.

10 menit untuk menentukan Program Studi siswa maka untuk 20 siswa ± 100

menit untuk menyelesaikan tugasnya sebagai konsultan jika dikerjakan secara serial. Berbeda jika kita menggunakan Aplikasi untuk setup awal data membutuhkan waktu ± 5 menit dan input data ± 1menit. Jadi jika kita memakai aplikasi Akan menghemat sebanyak ± 4-9 menit / bisa diasumsikan efektif waktu sebanyak ± 80%. Meskipun begitu aplikasi ini hanya berdasarkan Nilai (Belum *include* yang lain, seperti psikologi dan sebagainya)

II. LANDASAN TEORI

II.a. Implementasi

Implementasi adalah Bentuk aksi nyata dalam menjalankan rencana yang telah dirancang dengan Matang sebelumnya. Implementasi juga bisa dikatakan sebagai kegiatan yang dilakukan untuk menguji data dan menerapkan sistem yang diperoleh dari kegiatan seleksi. Selain daripada itu, Menurut Abdul Khadir mengatakan bahwa “Implementasi merupakan salah satu unsur pertahapan dari keseluruhan pembangunan sistem komputerisasi, dan unsur yang harus dipertimbangkan dalam pembangunan sistem komputerisasi yaitu masalah perangkat lunak (software), karena perangkat lunak yang digunakan haruslah sesuai dengan masalah yang akan diselesaikan, disamping masalah perangkat keras (hardware) itu sendiri”. (Andi, 2003).

II.b. *Weighted Product*

Metode *Weighted Product* (WP) adalah salah satu metode penyelesaian pada sistem pendukung keputusan. Metode ini mengevaluasi beberapa alternatif terhadap

sekumpulan atribut atau kriteria, dimana setiap atribut saling tidak bergantung satu dengan yang lainnya. Menurut Yoon, Metode *Weighted Product* menggunakan teknik perkalian untuk menghubungkan rating atribut, dimana rating tiap atribut harus dipangkatkan terlebih dahulu dengan bobot atribut yang bersangkutan. Terbagi menjadi beberapa langkah dalam menyelesaikan permasalahan menggunakan *Weighted Product*, diantaranya :

1. Menentukan Bobot Kriteria

Kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu $C[i]$ dan sifat dari masing-masing kriteria.

2. Menentukan Rating kecocokan dari setiap alternatif.

Menentukan Rating kecocokan setiap alternatif pada setiap kriteria, dan membuat matriks keputusan.

3. Melakukan Normalisasi Bobot.

Bobot Ternormalisasi = Bobot setiap kriterian / penjumlahan semua bobot kriteria. Sehingga hasilnya ketika dijumlahkan (\sum) = 1

$$\sum_{i=1}^n w_i = 1$$

Gambar 2.1. (Rumus 1)

Dimana:

n : Banyaknya bobot kriteria

I : Index bobot kriteria

w : Bobot kriteria

4. Menentukan Vektor S

Dengan cara mengalikan seluruh kriteria bagi sebuah alternatif dengan bobot sebagai pangkat positif untuk kriteria Linear Positif dan bobot

berfungsi sebagai pangkat negatif pada kriteria Linear Negatif. Note : dalam penelitian ini hanya membutuhkan Linear Positif Rumus untuk menghitung nilai preferensi untuk alternatif A_i , diberikan sebagai berikut (Lihat Gambar 1.2.):

$$S_i = \prod_{g=1}^n x_{ig}^{w_i}; g = 1, 2, 3 \dots m$$

Gambar 2.2. (Rumus 2)

Dimana :

S: menyatakan preferensi alternatif yang dianalogikan sebagai vektor.

x : menyatakan nilai kriteria.

w : menyatakan bobot kriteria.

i : menyatakan alternatif.

g : menyatakan kriteria.

n : menyatakan banyaknya kriteria

5. Menentukan Nilai Vektor V (Preferensi Alternatif)

Yaitu nilai yang akan digunakan untuk pe-rangking-an.

Nilai preferensi relatif dari setiap alternatif dapat dihitung dengan rumus (Gambar 2.3.):

$$V_i = \frac{\prod_{g=1}^n x_{ig}^{w_g}}{\prod_{g=1}^n (x_g)^{w_g}}; g = 1, 2, 3 \dots m$$

Gambar 2.3. (Rumus 3)

Dimana :

V : menyatakan preferensi alternatif yang dianalogikan V

x : menyatakan nilai kriteria

w : menyatakan bobot kriteria

i : menyatakan alternatif

g : menyatakan kriteria

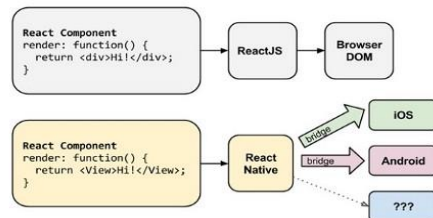
n : menyatakan banyaknya kriteria

6. Mengurutkan data berdasarkan rangking

Mengurutkan nilai yang telah didapatkan (Bisa langsung menjadi persentasi)

II.c. Sistem Bridging

Untuk penjelasan singkatnya bisa dilihat pada dibawah berikut (gambar 2.4).



Gambar 2.4. (Bridging)

II.d. NodeJS

Node.js adalah environment runtime, crossplatform JavaScript yang mengeksekusi kode JavaScript di luar Browser. Node.js memungkinkan pengembang menggunakan JavaScript untuk menulis script sisi di sisi Server. Menjalankan script sisi server menghasilkan konten halaman Web dinamis sebelum halaman dikirim ke browser web pengguna. Oleh karena itu, Node.js mewakili paradigma "Javascript Everywhere", menyatukan pengembangan aplikasi web di sekitar bahasa pemrograman tunggal (Client dan Server) daripada bahasa yang berbeda untuk scipt Server dan sisi Client. Node.js Pertama kali Rilis pada Tanggal 29 Mei 2009, dan rilis yang stabil terdapat pada versi 13.8.0 pada tanggal 6 Februari 2020. Node.js dibangun dengan bahasa (C,C++, dan Javascript). V8 adalah mesin eksekusi JavaScript yang awalnya dibuat untuk Google Chrome. Lisensi code tersebut kemudian diperbarui ke opensource oleh Google pada tahun 2008. Ditulis dalam C++, V8 mengkompilasi kode

sumber JavaScript ke kode mesin asli saat runtime.

II.e. ExpressJS

Express.js adalah satu web framework paling populer di dunia Node.js. Dokumentasinya yang lengkap dan penggunaannya yang cukup mudah, dapat membuat kita mengembangkan berbagai produk seperti aplikasi web ataupun RESTful API. Express.js juga dapat digunakan menjadi pijakan untuk membangun web framework yang lebih kompleks seperti, Sails.js, MERN (MongoDB, Express.js, Angular.js, Node.js) dan MERN (MongoDB, Express.js, React.js, Node.js).

Express.js dibuat oleh TJ Holowaychuk dan sekarang dikelola oleh komunitas. Beberapa keunggulan yang dimiliki oleh Express.js antara lain :

1. Dukungan pembuatan Middleware Dukungan terhadap berbagai HTTP Verb seperti POST, GET, PUT, DELETE, dan method option lainnya.
2. Sudah terpasang template engine Jade
3. Manajemen file statik seperti CSS dan Javascript
4. Sangat leluasa untuk dikostumisasi

II.f. React Native

React Native adalah sebuah framework javascript yang dikembangkan oleh facebook yang memungkinkan kita membuat aplikasi mobile Android atau iOS secara simultan. Di zaman sekarang ini sudah banyak framework javascript yang bisa kita gunakan untuk membuat aplikasi Android atau iOS, akan tetapi untuk React Native ini berbeda dengan

framework javascript lainnya. Dengan React Native ini kita tidak membuat aplikasi Hybrid / Crossplatform yang dimana aplikasi ini berjalan di javascript runtime akan tetapi kita akan membuat Real Aplikasi dimana untuk Android akan di compile di java dan untuk iOS akan di compile di Objective-C.

III. ANALISIS MASALAH

III.a. Evaluasi Sistem

Keunggulan dari sistem konvensional dalam menentukan Program Studi adalah sebagai berikut :

1. Tingkat akurasi dari pemilihan Program Studi siswa lebih tinggi.
2. Sinkronasi dengan penilaian Non Akademis (Akhlak dan Keterampilan).

Kelemahan dari sistem konvensional dalam menentukan Program Studi adalah sebagai berikut :

1. Waktu dalam penentuan Prediksi Program Studi relatif Lama
2. Tingkat manajemen Siswa relatif Sulit, seperti Sorting, Filtering, Searching.

Kesempatan dari sistem konvensional dalam menentukan Program Studi adalah sebagai berikut :

Jika data sudah terkumpul banyak (Lebih dari 500 data, digunakan untuk data latihan), dapat dikembangkan metode penelitian menggunakan yang lebih “Memudahkan” seperti Backpropagation sehingga bobot yang digunakan otomatis.

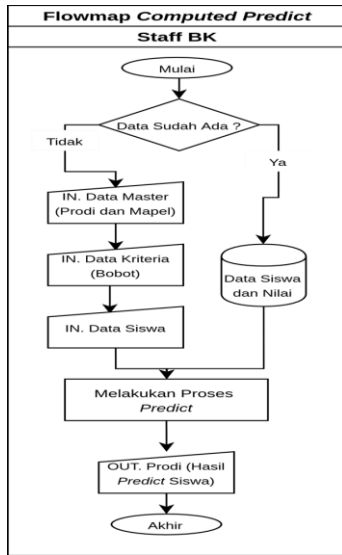
Tantangan dari sistem konvensional dalam menentukan Program Studi adalah sebagai berikut :

Perlunya pelaku khusus dalam penilaian yang bersifat “Terperinci”.

Sehingga penilai bisa menilai dengan tepat pada setiap siswa sesuai kemampuan IQ/EQ/SQ-nya.

IV. PERANCANGAN SISTEM

IV.a. Rancangan Flowmap

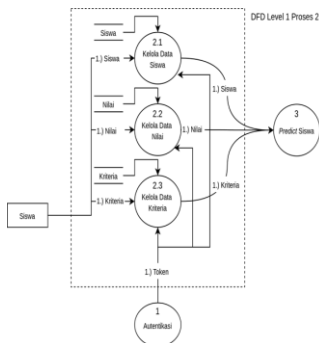


Gambar 4.1. (Flowmap)

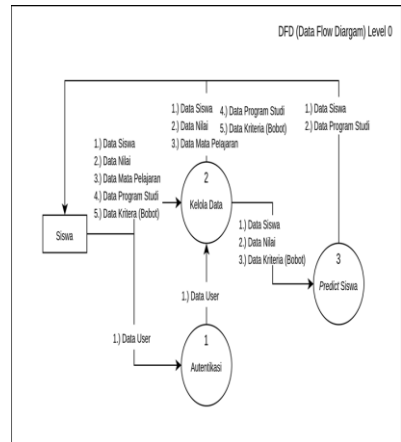
IV.b. DFD Level 0

Metodologi yang digunakan adalah DFO (Data Flow Oriented). Dimana terdapat tahapan tahapan yang harus dipenuhi :

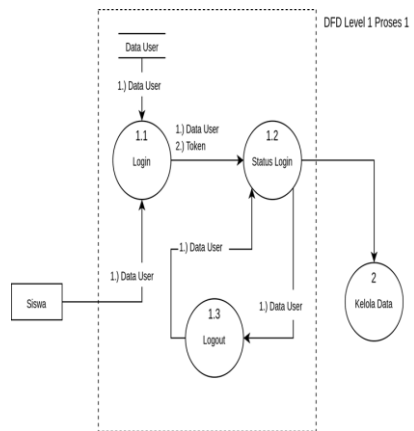
1. Penggambaran Diagram Konteks
2. Penggambaran DFD dan Turunannya
3. Data Dictionary
4. PSpec (*Process Specification*)



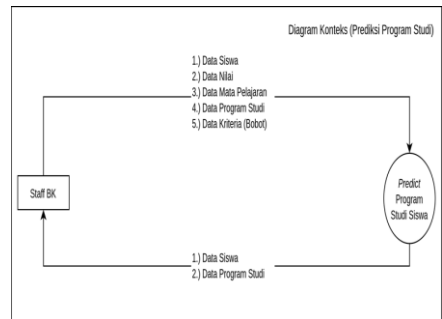
Gambar 4.2. (Diagram Konteks)



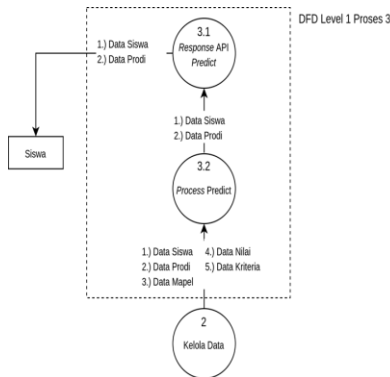
Gambar 4.2. (DFD Level 0)



Gambar 4.3. (DFD Level 1 Untuk Proses 1)



Gambar 4.4. (DFD Level 1 Untuk Proses 2)



Gambar 4.5. (DFD Level 1 Untuk Proses 3)

Kamus Data :

Kamus data disini menjelaskan tentang Prototypenya dari data yang mengalir dari DFD. Berikut Kamus Data yang mengalir dari DFD tersebut. :

- a. Data Sekolah = {NPSN, Nama}
- b. Data Siswa = {NISN, Nama, Jenis_Kelamin, Alamat, ID_Sekolah, ID_Program_Studi}
- c. Data Program_Studi = {ID, Nama}
- d. Data Mata_Pelajaran = {ID, Nama}
- e. Data Nilai = {ID, NISN, ID_Mata_Pelajaran, Nilai}
- f. Data Kriteria = {ID, ID_Program_Studi, ID_Mata_Pelajaran, Bobot}

Process Spesification :

Process 1.1. Login (IN : Obj.User; OUT : Obj.User, Obj.Token)

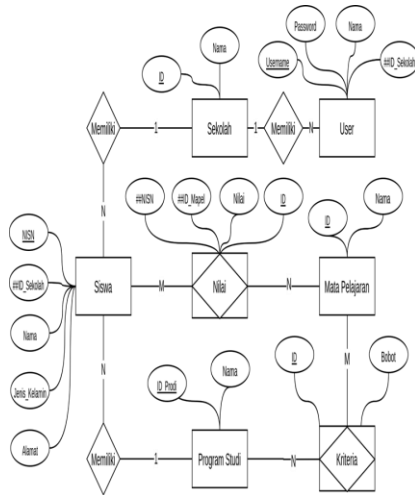
Process 1.2. Status_Login (IN : Obj.User, Obj.Token ; OUT : Obj.User, Obj.Token)

Process 3.2. Process_Predict (IN : Obj.Siswa, Obj.Prodi, Obj.Mapel, Obj.Nilai, Obj.Kriteria; OUT : Obj.Siswa, Obj.Prodi)

Process 3.1. Response_API (IN : Obj.Siswa, Obj.Prodi, Obj.Mapel, Obj.Kriteria, Obj.Nilai; OUT :

Obj.Siswa, Obj.Prodi, Obj.Mapel, Obj.Kriteria, Obj.Nilai)

IV.c. ERD



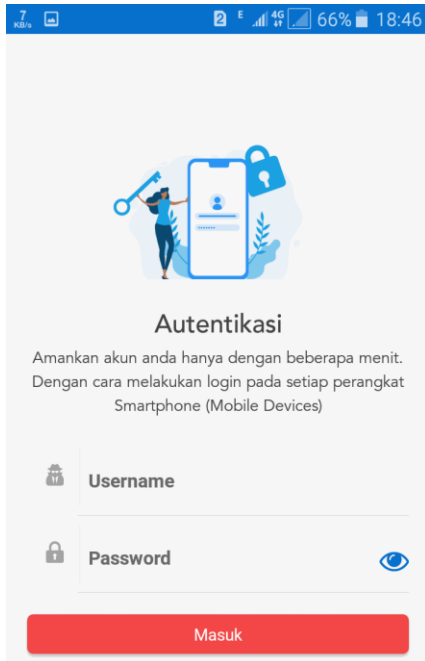
Gambar 4.6. (ERD)

V. IMPLEMENTASI

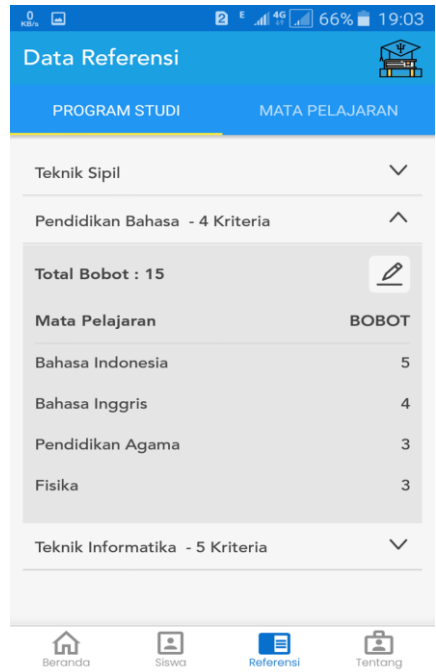
Bahasa dan Aplikasi yang digunakan adalah Javascript serta Visual Studio Code IDE merupakan salah satu perangkat lunak environment. Pengembangan aplikasi juga tentunya sesuai judul, menggunakan teknologi *crossplatform* (*React Native*).



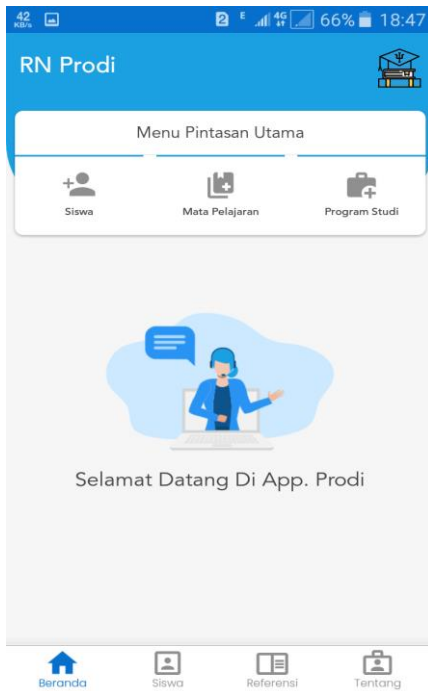
Gambar 5.1. (Landing 1)



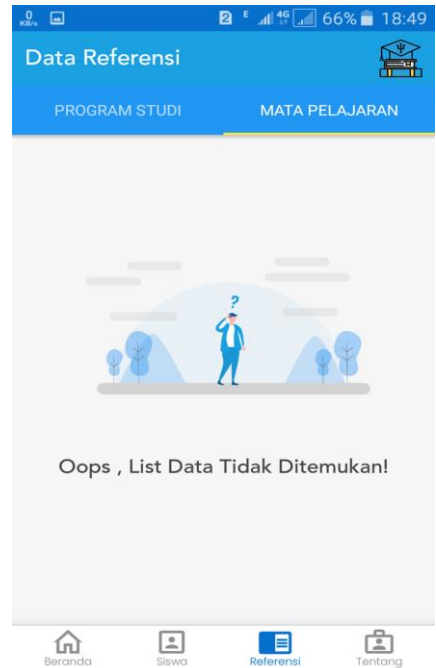
Gambar 5.2. (Login)



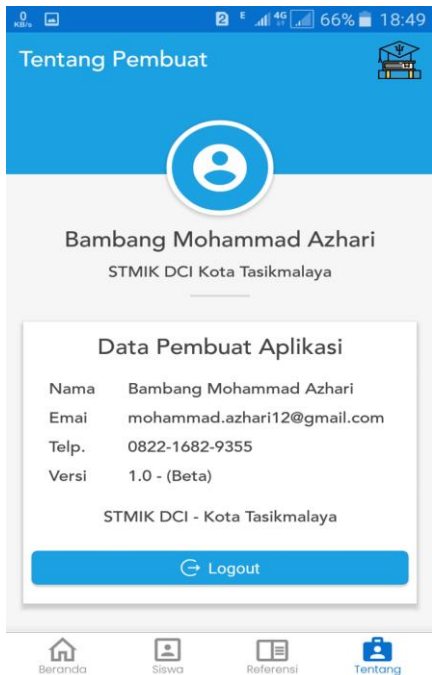
Gambar 5.4. (List Data Ref.)



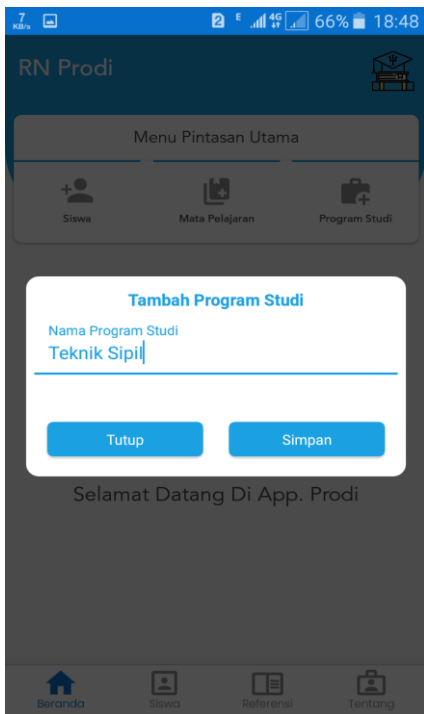
Gambar 5.3. (Dashboard)



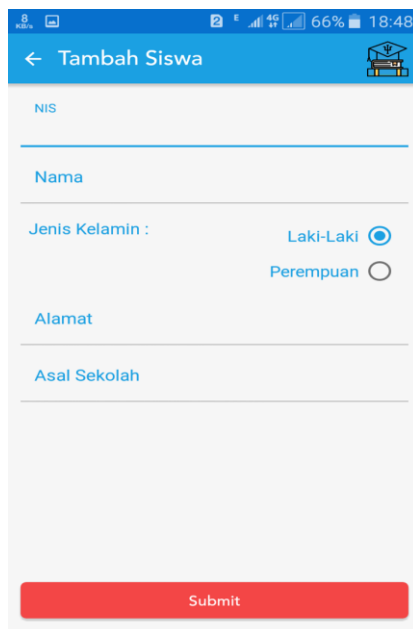
Gambar 5.5. (List Data Ref. Kosong)



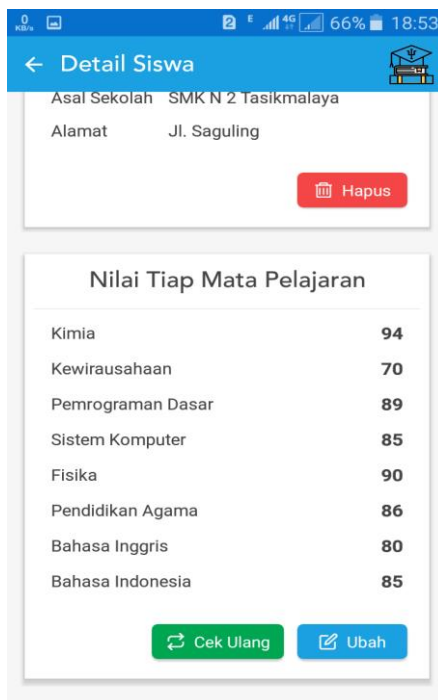
Gambar 5.6. (Tentang Pembuat App.)



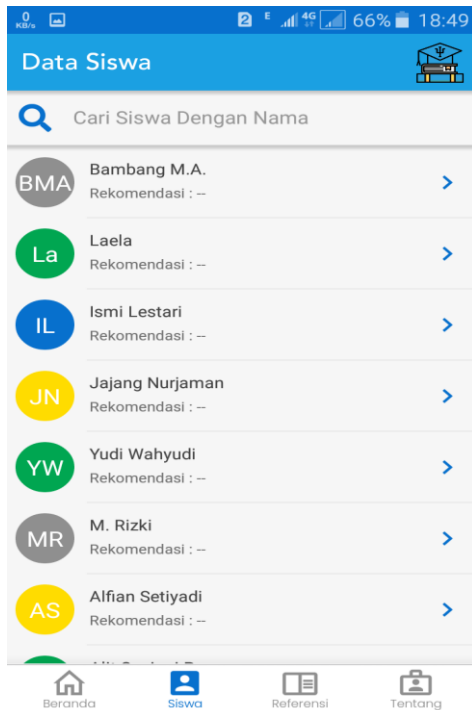
Gambar 5.7. (Tambah Data Ref.)



Gambar 5.8. (Tambah Siswa)



Gambar 5.9. (Detail Siswa)



Gambar 5.10. (List Data Siswa)

VI. KESIMPULAN

Berdasarkan uraian diatas berikut yang dapat penulis kemukakan untuk diambil kesimpulan sebagai berikut :

- a. Perbedaan antara prediksi / penentuan program studi bagi siswa, untuk efektifitas sangatlah berubah secara drastis (Hanya saja membutuhkan setup data untuk inialisasi awal, baik Siswa, Prodi, Mapel, maupun kriteria) yang awalnya ± 10 menit jadi bisa $<$ dari 2 menit. Dengan kata lain bisa menghemat $\pm 80\%$. Dan untuk efisiensi yang tadinya dikerjakan 2–3 Orang, sekarang bisa dikerjakan hanya dengan 1 Orang. Tetapi dengan adanya aplikasi ini, meskipun bisa

mempercepat dan mengurangi sumberdaya, ada hal yang harus diperhatikan yaitu Kurangnya tingkat akurasi dan presisi (Dikarenakan tidak semua variable diambil, Seperti Akhlak, Keterampilan, dsb.

- b. Dengan adanya smartphone kesempatan pengembang aplikasi semakin besar (Dimana dilihat banyaknya pengguna Smartphone khususnya berbasis android sangat banyak).

DAFTAR PUSTAKA

Abdul Kadir. 2003. *Pengenalan Sistem Informasi*. ANDI Yogyakarta, Yogyakarta. Bank Indonesia.

Android.com. (2018).

(<https://developer.android.com/studio/intro/?hl=id>)

Mengenal Android Studio. Diakses pada tanggal 3 Januari 2020.

Hariningsih, SP. (2005). *Teknologi Informasi*. Yogyakarta, Graha Ilmu.

Hariyanto, Bambang. (2018). *Dasar Informatika & Ilmu Komputer*. Yogyakarta, Graha Ilmu.

Haryanto, Bambang (2018). *Struktur Data, Membuat Program yang Elegan dan Effisien*.

Bandung, Informatika Bandung.

- Haryanto, Bambang (2010). Esensi-
Esensi Bahasa Pemrograman
Java.
Bandung, Informatika Bandung.
- Indrajani. (2014). Pengantar Sistem
Basis Data , Case Study All In
One.
Jakarta, PT. Elex Media
Komputindo.
- Kusumaningati, Imam FR. (2012).
NGANDROID. Hidup menjadi
mudah dan
menyenangkan dengan Android.
Jakarta, PT. Elex Media
Komputindo.
- Nugroho, Adi. (2004). Pemrograman
Berorientasi Objek.
Bandung, Informatika Bandung.
- React Native Developers (2020). React
Native Learn once, write
anywhere.
(<https://reactnative.dev/>)
Diakses pada tanggal 3 Januari
2020.