

# APLIKASI *FREE JOB FOR CARRER* BERBASIS *ANDROID*

Khairunnisyah<sup>1</sup>, Randi Naswiro<sup>2</sup>

<sup>1,2</sup> Informatika, Fakultas Teknik Universitas Muhammadiyah Bengkulu  
Jl. Bali Po. Box, 118 Kota Bengkulu 38119 INDONESIA  
(telp : 0736-22765; fax : 0736-26161)

<sup>1</sup> anis\_zhu@yahoo.co.id

<sup>2</sup>randi@yahoo.com

*Abstrak* : Banyak orang-orang yang menjadi pengangguran dikarenakan kurangnya lapangan pekerjaan dan kurangnya bekal ilmu yang mereka miliki. Permasalahan pengangguran ini sekarang sudah sangat mengawatirkan sehingga pemerintah harus terus berupaya untuk mengatasi permasalahan ini. Untuk membantu pemerintah dalam mengurangi jumlah pengangguran maka penulis ingin membangun sebuah aplikasi yang bisa digunakan untuk mengecek lowongan pekerjaan yang sedang buka. Agar aplikasi dapat diakses dengan mudah kapan saja dan dimana saja, maka penulis akan menggunkan salah satu teknologi yang sedang berkembang saat ini, yaitu *smartphone Android*. Dalam aplikasi ini juga nantinya akan diberikan penjelasan mengenai hak dan kewajiban dari jabatan yang dilamar. Sehingga pelamar sudah mengerti mengenai jabatan yang dilamar dan nantinya diharapkan bisa berguna saat tes yang akan dilakukan. Dengan mengetahui jenis pekerjaan yang akan dilamar maka *user* juga bisa merencarakan jenjang karir yang akan didapat melalui jabatan tersebut.

*Kata kunci* : aplikasi *Android*, *smartphone*, aplikasi lowongan kerja

***Abstract: Unemployment is a condition where a person does not have a job to make ends meet. In developing countries unemployment is a problem that still shackle including our country Indonesia. Many people who become unemployed due to lack of jobs and lack of provision of knowledge they have. Problems of unemployment is now very worrying that the government should continue its efforts to overcome this problem. To assist the government in reducing the number of unemployed, the authors want to build an application that can be used to check the jobs that are open. So that applications can be easily accessed anytime and anywhere, the author will use the one technology that is being developed at this time, the Android smartphone. In this application will also be given an explanation of the rights and obligations of the proposed position. So that the applicant has understood the proposed position and is expected to be useful during the test to be performed. By knowing the type of work that will be spoken, the user also can merencarakan career path will be obtained through the post.***

***Keywords: Android, smartphone, job searching application.***

## I. PENDAHULUAN

Pengangguran adalah sebuah kondisi dimana seseorang tidak memiliki pekerjaan untuk memenuhi kebutuhan hidup. Di negara berkembang pengangguran adalah masalah yang masih membelenggu. Termasuk Negara kita Indonesia. Banyak orang-orang yang menjadi pengangguran dikarenakan kurangnya lapangan pekerjaan dan kurangnya bekal ilmu yang mereka miliki. Minimnya ilmu yang mereka miliki membuat sumber daya manusia yang ada menjadi kurang berkualitas, sedangkan perusahaan-perusahaan besar membutuhkan sumber daya manusia yang berkualitas tentunya. Masalah pengangguran tersebut tidak bisa kita anggap sepele karena dapat membawa dampak negatif yang sangat beragam.

Untuk membantu pemerintah dalam mengurangi jumlah pengangguran maka penulis ingin membangun sebuah aplikasi yang bisa digunakan untuk mengecek lowongan pekerjaan yang sedang buka. Agar aplikasi dapat diakses dengan mudah kapan saja dan dimana saja, maka penulis akan menggunakan salah satu teknologi yang sedang berkembang saat ini, yaitu *smartphone Android*. *Smartphone Android* merupakan ponsel pintar yang saat ini banyak beredar di Indonesia. Dalam aplikasi ini juga nantinya akan diberikan penjelasan mengenai hak dan kewajiban dari jabatan yang dilamar. Sehingga pelamar sudah mengerti mengenai jabatan yang dilamar dan nantinya diharapkan bisa berguna saat tes yang akan dilakukan. Dengan mengetahui jenis pekerjaan yang akan dilamar maka *user* juga bisa merencanakan jenjang karir yang akan didapat melalui jabatan tersebut. Berdasarkan latar belakang diatas, maka peneliti tertarik untuk membuat sebuah “Membuat Aplikasi *Free Job For Career* Berbasis *Android*” dengan harapan sistem tersebut dapat membantu *user* mengidentifikasi pekerjaan yang sesuai dengan kemampuan.

## II. LANDASAN TEORI

### A. Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output* [1]. Program aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer supaya dapat melakukan pekerjaan sesuai dengan keinginan *programmer* atau *user*. Atau definisi lain aplikasi merupakan kumpulan dari prosedur-prosedur yang digunakan untuk mengolah data menjadi informasi. Misalnya

penjumlahan, klasifikasi, rotasi, koreksi geometri, *query*, *overlay*, *buffer*, *jointable* dan sebagainya.

### B. Android

*Android* sebagai sebuah sistem, adalah sistem operasi berbasis Java yang berjalan pada kernel 2.6 *Linux*. Aplikasi *Android* yang dikembangkan menggunakan Java dan mudah menyesuaikan ke *platform* baru. *Android* merupakan satu kumpulan lengkap perangkat lunak yang dapat berupa sistem operasi, *middleware*, dan aplikasi kunci perangkat *mobile*. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari *boot loader*, *device driver*, dan fungsi-fungsi pustaka, hingga perangkat lunak API (*Application Programming Interface*), termasuk aplikasi SDK (*Software Development Kit*). Jadi, sebenarnya *Android* bukanlah satu perangkat tertentu, melainkan sebuah *platform* yang dapat digunakan dan diadaptasikan untuk mendukung berbagai konfigurasi perangkat keras. Walaupun kelas utama perangkat yang didukung oleh *Android* adalah telepon *mobile*, tetapi sekarang ini juga digunakan pada *electronic book readers*, *netbooks*, *tablet*, dan *set-top boxes* (STB) [2].

#### a. Arsitektur Android

Arsitektur *Android* menunjukkan komponen-komponen utama yang terdapat pada sistem operasi *Android*. Berikut ini arsitektur *Android* :

- *Linux Kernel*

*Android* dibangun di atas kernel *Linux* 2.6. Namun secara keseluruhan *Android* bukanlah *Linux*, karena dalam *Android* tidak terdapat paket standar yang dimiliki oleh *Linux* lainnya. *Linux* merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada *Android* hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan *driver*. Kernel *Linux* menyediakan *driver*

layar, kamera, *keypad*, *WiFi*, *Flash Memory*, *audio*, dan *IPC (Interprocess Communication)* untuk mengatur aplikasi dan lubang keamanan.

- *Libraries*

*Android* menyertakan *libraries C/C++* yang digunakan oleh berbagai komponen dari sistem *Android*. Kemampuan ini disediakan kepada Developer aplikasi melalui *Framework Aplikasi Android*. Beberapa inti *libraries* tercantum dibawah ini:

1. *System C Library*

Variasi dari implementasi *BSD*-berasal pelaksana sistem standar *C library (libc)*, sesuai untuk perangkat *embedded* berbasis *Linux*.

2. *Media Libraries*

*PacketVideo* berdasarkan *OpenCORE*; *library* mendukung pemutaran rekaman dan populer banyak *format audio* dan *video*, serta file gambar, termasuk *MPEG4*, *H.264*, *MP3*, *AAC*, *AMR*, *JPG*, dan *PNG*.

3. *Surface Manager*

Mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi.

4. *LibWebCore*

Mesin *web modern* yang *powerfull* yang baik browser *Android embedded web*.

5. *3D Libraries*

Sebuah pelaksana berdasarkan *OpenGL ES 1.0 API*; perpustakaan baik menggunakan perangkat keras akselerasi 3D (apabila tersedia) atau yang disertakan, sangat optimal *3D software rasterizer*.

6. *FreeType*

7. *Bitmap* dan vektor *font rendering*.

8. *SQLite*

Mesin *database* yang kuat dan ringan, dan penghubung tersedia untuk semua aplikasi.

- b. *Android Runtime*

*Android* terdiri dari satu *set core libraries* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core libraries* bahasa pemrograman *Java*. Setiap aplikasi menjalankan prosesnya sendiri dalam *Android*, dengan masing-masing instan dari mesin *virtual Dalvik (Dalvik VM)*. *Dalvik* dirancang agar perangkat dapat menjalankan *multiple VMs* secara efisien. Mesin *Virtual Dalvik* mengeksekusi *file* dalam *Dalvik executable (.dex)*, sebuah format yang dioptimalkan untuk memori yang kecil. *Dalvik VM* berbasis, berjalan, dan dikompilasi oleh *compiler* bahasa *Java* yang telah ditransformasikan ke dalam *.dex format* oleh *tool "dx"* yang telah disertakan. *Dalvik VM* bergantung pada kernel *Linux* untuk berfungsi seperti *threading* dan manajemen memori tingkat rendahnya.

- c. *Framework Aplikasi*

Pengembang memiliki akses penuh menuju *API framework* yang sama dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali dengan mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini (sesuai dengan batasan keamanan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna. Semua aplikasi merupakan rangkaian set layanan dan sistem, termasuk:

- *Views* yang kaya dan *extensible* yang dapat digunakan untuk membangun aplikasi, termasuk *list*, *grids*, kotak teks, tombol, dan bahkan sebuah *embedded web*.
- *Content Provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak telepon) atau membagi data tersebut.

- *Resource Manager*, yang menyediakan akses ke kode non-sumber lokal seperti *string*, gambar, dan tata letak *file*.
- *Notifications Manager* yang memungkinkan semua aplikasi menampilkan *alert* yang bisa dikostumisasi di dalam status bar.
- *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack*.

d. *Application*

*Android* telah menyertakan aplikasi inti seperti email *client*, *SMS*, kalender, peta, *browser*, kontak, dan lainnya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman *Java*. Pada layer inilah *developer* atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada *Android* semua aplikasi baik aplikasi inti maupun aplikasi pihak ketiga berjalan pada layer aplikasi dengan menggunakan *library API* yang sama. Ini berarti semua aplikasi yang dibuat untuk *Android* akan memiliki akses yang setara dalam mengakses seluruh kemampuan handset, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan platform *Android* ini, *Programmer* atau *Developer* secara penuh akan bisa mengkostumisasi perangkat *Android*nya [3].

e. *Hardware dan Software*

1. *Hardware*

Perangkat keras yang digunakan dalam penelitian ini adalah satu unit *Notebook aspire 4736* dengan spesifikasi sebagai berikut :

- *Processor* : Intel ® Core™2 duo processor,
- *Memory* : Ram 4 GB,
- *Harddisk* : HDD 320 GB,
- *Monitor* : 14"
- *Handphone* : *Smartphone* Lenovo a369i

2. *Software*

Perangkat lunak yang digunakan dalam penelitian ini adalah sistem operasi *Windows 7*. Perangkat lunak tambahan seperti software pengolah kata untuk laporan menggunakan :

- *Eclipse*
- *Android SDK*

### III. METODOLOGI PENELITIAN

#### A. *Metode Pengumpulan Data*

Untuk memperoleh data, penulis menggunakan metode-metode pengumpulan data sebagai berikut:

1. *Metode Observasi*

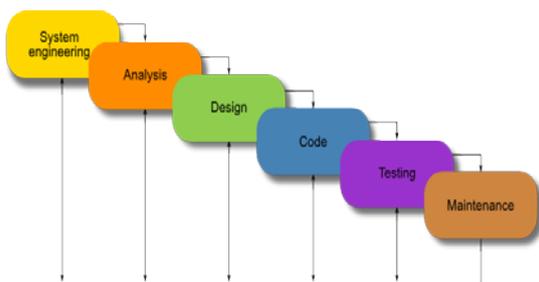
Merupakan metode pengumpulan data dengan melakukan penulisan dan pengamatan terhadap permasalahan yang dihadapi penulis dalam mencari data.

2. *Metode Studi Pustaka*

Merupakan metode pengumpulan data dengan cara mencari informasi baik didapat pada perpustakaan, internet.

#### B. *Pengembangan Sistem*

Pengembangan sistem didefinisikan sebagai aktivitas untuk menghasilkan sistem informasi berbasis komputer untuk menyelesaikan persoalan (*problem*) organisasi atau memanfaatkan kesempatan (*opportunities*) yang timbul. Model air terjun (*waterfall*). Mengambil kegiatan dasar seperti spesifikasi, pengembangan, validasi, dan evolusi dan merepresentasikannya sebagai fase seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian dan seterusnya. *Waterfall* Biasa juga disebut siklus hidup perangkat lunak.



Gambar 1. Model *Waterfall*

Keterangan Menurut gambar diatas alur dari Model *Waterfall* sebagai berikut:

1. Rekayasa perangkat lunak (*system enginerring*), melakukan pengumpulan data dan penetapan kebutuhan semua elemen system .
2. *Requirements analysis* , melakukan analisis terhadap permasalahan yang dihadapi dan menetapkan kebutuhan perangkat lunak, fungsi *performsi* dan *interfacing*.
3. *Design*, menetapkan domain informasi untuk perangkat lunak, fungsi dan *interfacing*.
4. *Coding* (implementasi), pengkodean yang mengimplementasikan hasil desain ke dalam kode atau bahasa yang dimengerti oleh mesin komputer dengan menggunakan bahasa pemrograman tertentu.
5. *Testing* (pengujian) , kegiatan untuk melakukan pengetesan program yang sudah dibuat apakah udah benar atau belum di uji dengan cara manual.jika *testing* sudah benar maka program boleh digunakan.
6. *Maintenance* (perawatan), menangani perangkat lunak yang sudah selesai supaya dapat berjalan lancar dan terhindar dari gangguan-gangguan yang dapat menyebabkan kerusakan.

C. *Perancangan Aplikasi*

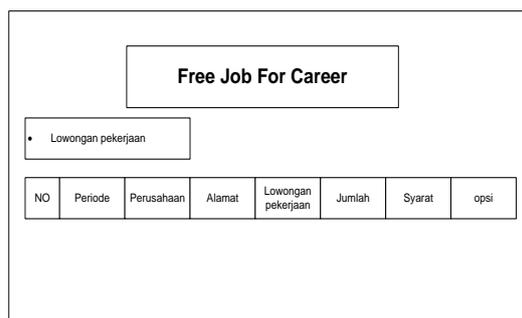
Berikut ini adalah rancangan tampilan sementara dari Aplikasi *Free Job For Career* yang akan dibuat:

1. Desain Tampilan Depan



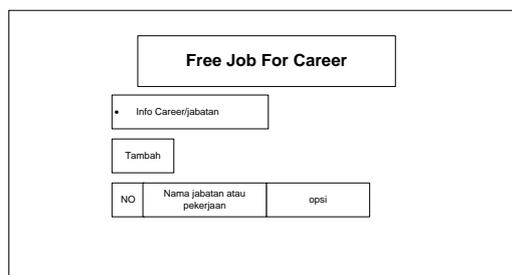
Gambar 2. Desain Tampilan Depan

2. Desain Menu Index lowongan



Gambar 3. Desain *Index* Lowongan

3. Desain Menu *Index Career*



Gambar 4. Desain *Index Career*

IV. HASIL DAN PEMBAHASAN

A. *Admin*

Data-data lowongan senantiasa akan diupdate oleh operator aplikasi atau *admin* yang bertugas. Pada sub bab ini akan membahas aplikasi yang digunakan pada *platform* komputer.



Gambar 5. Halaman *Home*

Halaman *home* adalah halaman yang pertama kali dilihat *admin* ketika mengakses aplikasi. Halaman ini berisi dua *icon* menu yang berfungsi untuk menampilkan halaman lowongan dan jabatan.

### 1. Halaman *Index* Lowongan

Halaman *index* lowongan merupakan halaman utama dari kategori lowongan. Di halaman inilah *admin* menginputkan data lowongan pekerjaan. Tampilan dari halaman ini bisa dilihat pada gambar dibawah ini :

No.	Periode	Perusahaan	Alamat	Pekerjaan	Jumlah	Syarat	Opsi
1	2015-01-22 sid 2015-03-30	PT.JDA Indonesia	Jakarta Selatan	Staff Administrasi	2	1. High school graduate or equivalent 2. D1 and above are welcome to apply 3. Maximum age 25 years old 4. Computer literate 5. Diligent and honest 6. Detail oriented	Baca Edit Hapus
2	2015-01-22 sid 2015-01-29	UMB	Jalan Bali	Dosen	5	1. pendidikan minimal S2 2. muslim	Baca Edit Hapus
3	2015-01-25 sid 2015-01-27	Gudang Garam		admin	2	1. Pendidikan minimal s1 2. Berpengalaman 3. Mampu Mengoperasikan Komputer	Baca Edit Hapus

Gambar 6. Halaman *Index* Lowongan

### 2. Halaman Tambah Data

Halaman tambah data merupakan halaman yang digunakan untuk menginputkan data lowongan pekerjaan ke dalam *database*. Halaman ini berisi *form input* data lowongan, data-data tersebut antara lain mulai, berakhir, perusahaan, alamat, jenis, jumlah dan syarat. Tampilan dari form input data terlihat seperti gambar berikut ini :

Gambar 7. Halaman Input Data Lowongan

Setelah *form* diisi maka *user* harus mengklik tombol simpan agar data inputan tersimpan ke dalam *database*. Jika *admin* lupa atau sengaja mengosongkan salah satu *input text* maka akan tampil pemberitahuan seperti pada gambar dibawah ini :

Gambar 8. Validasi Halaman *Input* Data

### 3. Halaman Baca

Halaman ini hanya menampilkan data yang dipilih untuk dilihat secara detail pada halaman *index* lowongan. Halaman ini akan tampil jika *user* mengklik menu baca di kolom opsi pada tabel daftar di halaman *index*. Tampilan dari halaman baca akan terlihat seperti gambar dibawah ini:

Gambar 9. Halaman baca

### 4. Halaman Edit

Jika *user* mengklik menu edit yang terdapat di halaman *index*, maka akan tampil halaman edit data. Halaman edit digunakan untuk mengubah data lowongan yang telah tersimpan di dalam *database*. Halaman ini mirip dengan halaman *input* data, akan tetapi pada halaman ini input teksnya telah terisi sesuai data yang dipilih pada halaman sebelumnya.

Gambar 10. Halaman edit data

### 5. Menu Hapus

Menu hapus yang terdapat di halaman *index* memiliki fungsi untuk menghapus data yang dipilih dari dalam database secara permanen. Data yang telah terhapus tidak bisa dikembalikan lagi kecuali dengan cara *diinput* ulang, memberikan sebuah peringatan sebelum *user* menggunakan menu ini. Ketika *user* mengklik menu hapus, maka akan tampil halaman konfirmasi seperti yang terlihat pada gambar dibawah ini :

Gambar 11. Konfirmasi hapus data

Jika *user* memilih untuk mengklik tombol “Ya” maka data yang dipilih tadi akan terhapus, sedangkan jika yang diklik tombol “Tidak” maka *user* akan dikembalikan ke halaman *index*.

### 6. Halaman *index* Jabatan

Selain halaman lowongan, ada juga halaman jabatan yang digunakan untuk menginput data mengenai jabatan dari suatu perusahaan beserta definisinya sehingga *user* yang ingin melamar

pekerjaan bisa mengerti mengenai jenis pekerjaan apa yang mereka lamar. Halaman *index* jabatan ini sama dengan halaman *index* lowongan, yang membedakannya hanya tabel daftar yang ditampilkan. Jika halaman *index* lowongan menampilkan tabel lowongan maka halaman *index* jabatan menampilkan tabel jabatan dari dalam *database*. Untuk lebih jelasnya halaman *index* jabatan ini bisa dilihat seperti gambar dibawah ini :

No.	Nama	Ops
1	Marketing	Baca Edit Hapus
2	Staff Administrasi	Baca Edit Hapus
3	General Affair/Bagian Umum	Baca Edit Hapus
4	Kepala Cabang	Baca Edit Hapus
5	Supervisor	Baca Edit Hapus
6	Manager	Baca Edit Hapus
7	General Manager	Baca Edit Hapus

Gambar 12. Halaman *Index* Jabatan

### 7. Halaman *input* data jabatan

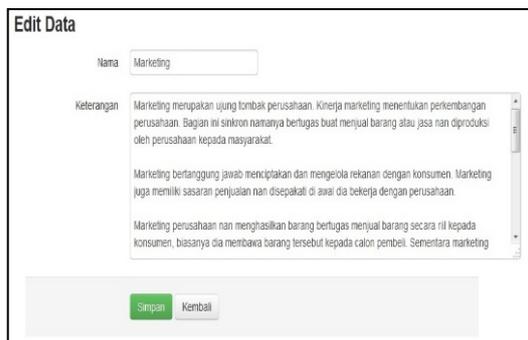
Halaman ini digunakan untuk menambah data jabatan yang nantinya akan dilihat oleh *user*. Konsep dari halaman ini sama dengan halaman *input* data lowongan, tetapi data yang diisi berbeda. Pada *form* ini terdapat dua *input* teks berupa nama dan keterangan.

Gambar 13. Halaman *Input* Data Jabatan

### 8. Halaman edit data jabatan

Halaman ini digunakan untuk mengubah data jabatan yang telah tersimpan ke *database*. *User* bisa menampilkan halaman ini dengan mengklik tombol menu edit yang terdapat di halaman *index*

jabatan. Konsep dari halaman ini sama dengan halaman edit data lowongan diatas.



Gambar 14. Halaman Edit Data Jabatan

lebih jelasnya bisa dilihat pada gambar dibawah ini :



Gambar 16. Halaman Utama

### 9. Halaman baca data jabatan

Halaman ini digunakan oleh *user* untuk melihat secara detail data jabatan yang dipilih. Untuk menampilkan halaman ini *user* harus mengklik tombol menu baca yang terdapat di halaman index jabatan. Halaman ini juga nantinya akan di lihat *user* melalui *smartphone Android* mereka.



Gambar 15. Halaman Baca Jabatan

### 10. User

*User* disini merupakan tampilan yang akan dilihat *user* melalui *smartphone* mereka. Halaman yang pertama kali akan dilihat adalah halaman utama yang memiliki dua *icon menu*, yaitu menu *job* dan menu karir. *Icon menu job* digunakan untuk menampilkan halaman lowongan kerja sedangkan *icon menu karir* digunakan untuk menampilkan halaman informasi jabatan. Untuk

Jika *user* mengklik *icon job* maka akan tampil halaman seperti berikut ini :

No.	Periode	Perusahaan	Alamat	Pekerjaan	Jumlah	Opsi
1	2015-01-22 s/d 2015-03-30	PT JDA Indonesia	Jakarta Selatan	Staff Administrasi	2	<a href="#">Baca</a>
2	2015-01-22 s/d 2015-	UMB	Jalan Bali	Dosen	5	<a href="#">Baca</a>

Gambar 17. Halaman Daftar Lowongan

Jika posisi *smartphone* tegak lurus maka tampilannya akan menjadi seperti ini :

Lowongan	
Nomor	1
Lowongan	2015-01-22 s/d 2015-03-30
PT JDA Indonesia	
Jakarta Selatan	
Staff Administrasi	
2	
<a href="#">Baca</a>	
Nomor	2
Lowongan	2015-01-22 s/d 2015-01-29
UMB	
Jalan Bali	
Dosen	
5	
<a href="#">Baca</a>	

Gambar 18. Halaman Daftar Lowongan Posisi

Halaman daftar lowongan ini sama dengan halaman daftar lowongan yang telah dijelaskan sebelumnya, akan tetapi daftar halaman ini tidak

memiliki menu edit dan hapus. Seperti yang terlihat pada gambar dibawah ini :



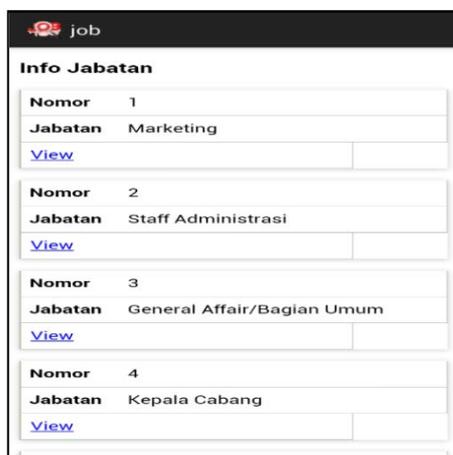
Gambar 19. Halaman Baca Lowongan

Ketiga halaman diatas merupakan halaman yang tampil jika *user* memilih *icon job* dan jika *user* tersebut memilih *icon* karir maka akan tampil halaman daftar karir seperti gambar berikut ini :

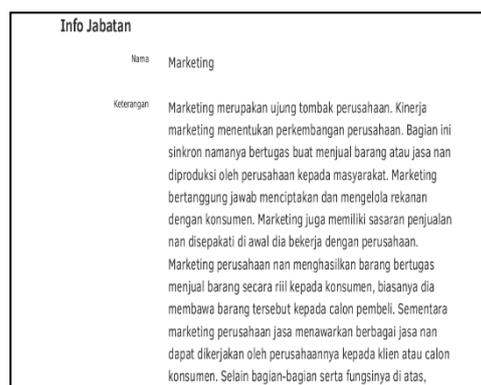


Gambar 20. Halaman Daftar Karir Posisi

Sama dengan sebelumnya, jika posisi *smartphone* tegak atau potrait maka yang akan tampil adalah halaman seperti berikut ini :



Gambar 21. Halaman daftar karir posisi potrait



Gambar 22. Halaman Baca Info

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Aplikasi ini mudah dijalankan dan memiliki efektifitas yang tinggi bagi *user*nya. *User* akan sangat terbantu dalam mencari lowongan pekerjaan yang sesuai dengan kemampuan dan ijazahnya serta mengenali pekerjaan yang akan dilamar.

### B. Saran

Untuk pengembangan lebih lanjut dari aplikasi *Android* ini, ada beberapa saran yang dapat diberikan, antara lain :

1. Aplikasi ini dapat dikembangkan lagi ketingkat *user* yang lebih tinggi lagi, misalnya untuk lowongan pekerjaan perusahaan besar.
2. Dalam hal pembuktian pengisian data pelamar dapat dicetak atau di print.

## REFERENSI

- [1] Jogiyanto, HM. *Pengertian Aplikasi dan Perkembangan Aplikasi*. Yogyakarta: Andi Offset. 2001.
- [2] Safaat, H Nazruddin. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung : Informatika. 2011.
- [3] Supardi, Ir. Yuniar. 2012. *Sistem Operasi Andal Android*. Jakarta : Elex Media