

Laporan Tugas Besar Teknologi Sistem Terintegrasi  
Kelompok 14

Feby Eliana Tengry - 18217030  
Muhammad Yanza Hattari - 18217043

1. Latar Belakang

Berdasarkan data dari *Worldometers*, jumlah populasi manusia saat ini telah mencapai 7,7 miliar jiwa dan *UN Department of Economic and Social Affairs* memprediksi bahwa populasi manusia akan mencapai 9,8 miliar jiwa pada tahun 2050. Peningkatan penduduk yang besar ini tentu akan berdampak pada peningkatan kebutuhan masyarakat yang salah satunya adalah pangan. Untuk dapat memenuhi permintaan pasar yang besar, penyedia pangan harus mampu menghasilkan hasil pangan dengan produktivitas yang tinggi.

Dalam sektor agrikultur, cuaca memegang peran yang sangat penting dalam kesuksesan produksi. Cuaca dapat mempengaruhi pertumbuhan tanaman, hasil panen, kemunculan hama, kebutuhan air dan pupuk, serta berbagai aktivitas yang dilakukan selama bercocok tanam. Dalam kata lain, agrikultur di lahan terbuka sangat bergantung pada cuaca. Namun sayangnya, perubahan cuaca adalah perihai yang diluar kuasa manusia. Yang dapat para petani lakukan adalah beradaptasi dan melakukan pencegahan untuk melindungi tanamannya agar dapat memperoleh tingkat keberhasilan yang tinggi dalam produksi.

Untuk itu kami ingin turut berkontribusi dalam memenuhi *Sustainable Development Growth* kedua yaitu mengakhiri kelaparan, mencapai ketahanan pangan dan nutrisi yang lebih baik dan mendukung pertanian berkelanjutan dengan membantu menyediakan teknologi bagi para petani untuk dapat memonitor cuaca agar dapat mengambil tindakan terbaik untuk meningkatkan produktivitas panen.

2. Rumusan Masalah

- 1) Bagaimana aplikasi yang dibangun dapat membantu petani dalam melakukan pemonitoran cuaca?
- 2) Seberapa baik dampak dari penggunaan aplikasi terhadap produktivitas panen?

3. Solusi

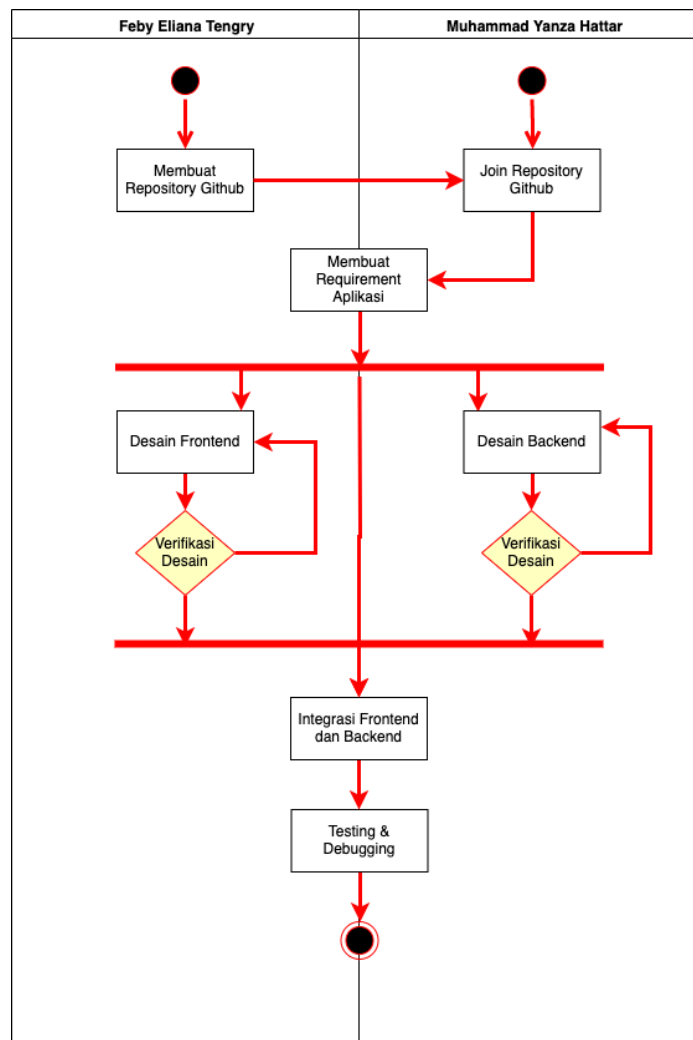
3.1. Solusi yang Ditawarkan

- Aplikasi dapat memberikan data cuaca saat ini
- Aplikasi dapat menampilkan data perkiraan cuaca

3.2. Perencanaan Pengerjaan Aplikasi

### 3.2.1. Rencana Pengerjaan

Berikut ini adalah rencana pengerjaan aplikasi berikut (sesuai proposal pengajuan proyek)



### 3.2.2. Timeline

Berikut ini *timeline* pengerjaan aplikasi (sesuai proposal pengajuan proyek)

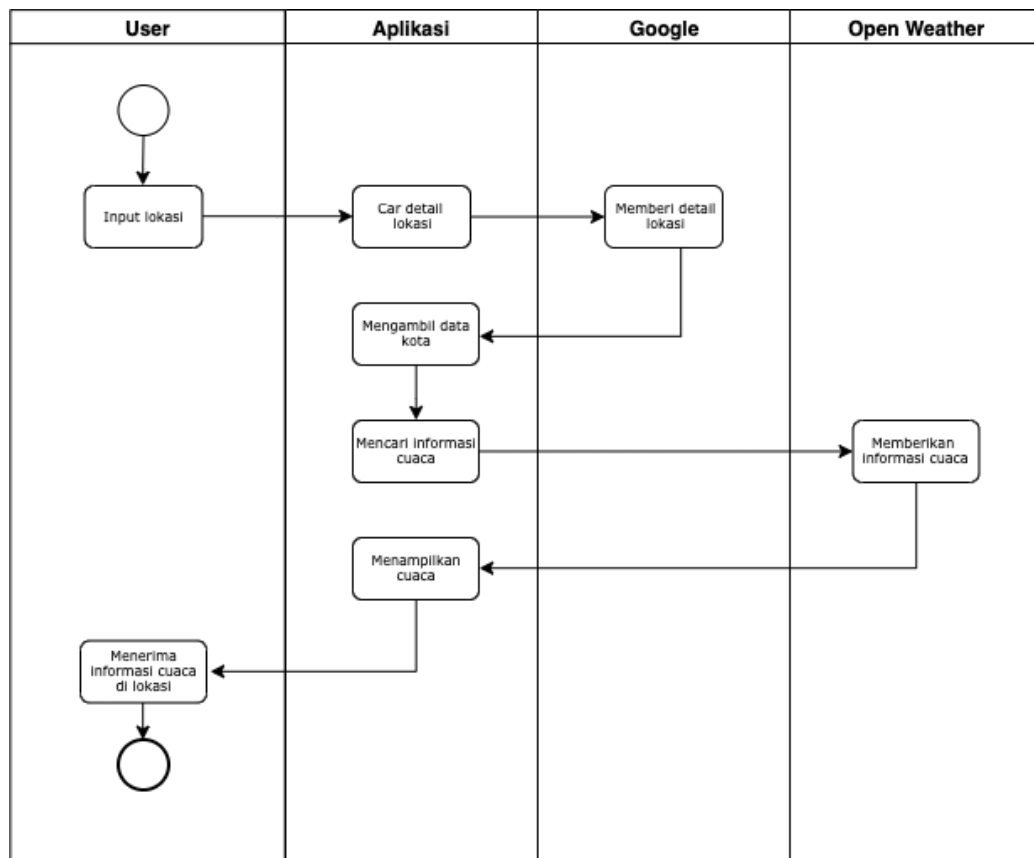
Milestone	Tanggal
Dokumen desain aplikasi	12 November 2019
Membuat <i>repository</i> Github	13 November 2019
Join <i>repository</i> Github	13 November 2019
Mengumpulkan <i>requirement</i> untuk aplikasi	15 November 2019

Finalisasi <i>requirement</i>	18 November 2019
Membuat desain <i>frontend</i> dan <i>backend</i>	25 November 2019
Verifikasi desain <i>frontend</i> dan <i>backend</i>	2 Desember 2019
Integrasi <i>frontend</i> dan <i>backend</i>	6 Desember 2019
<i>Testing</i> dan <i>debugging</i>	10 Desember 2019

### 3.3. Desain Aplikasi

#### 3.3.1. Cara Kerja Aplikasi

Berikut ini adalah cara kerja aplikasi.

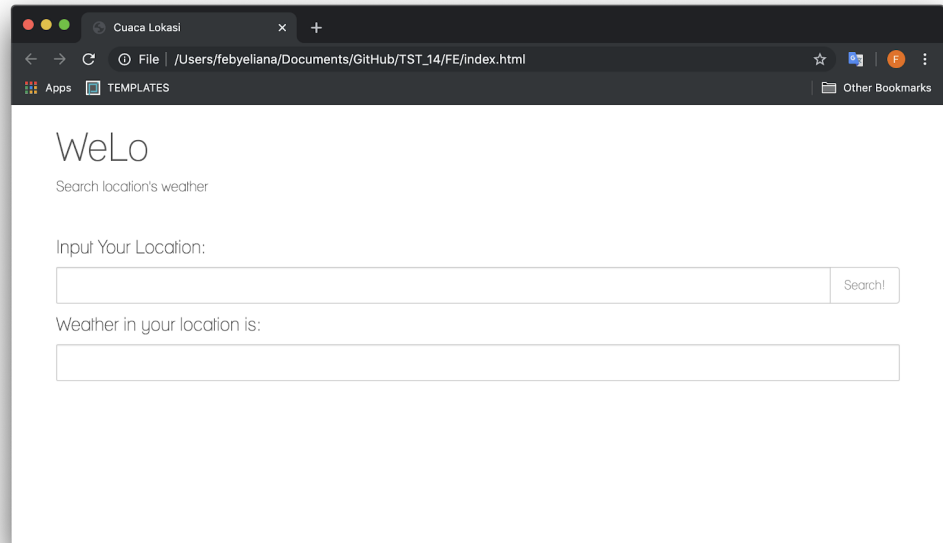


Untuk membangun aplikasi ini digunakan 2 API yaitu **API Get Place Detail** dari Muhammad Yanza Hattari (NIM 18217043) dan **API Search Weather** dari Dwi Nova Wijayanti (NIM 18217048). Pada aplikasi pengguna akan membuat input berupa nama lokasi. Data yang diterima akan menjadi *query* untuk pencarian detail lokasi pada API Get Place Detail. Dengan memproses *query* dari input dan *key* yang sudah disediakan *backend*, API Get Place Detail akan menghasilkan detail lokasi berupa data bujur dan lintang dari lokasi. Data detail lokasi yang dihasilkan dari API Get Place Detail akan

menjadi query untuk pencarian data cuaca pada API Search Weather. API Search Weather akan mengembalikan data cuaca saat ini dan perkiraan cuaca dari lokasi yang dicari pengguna. Data ini selanjutnya akan ditampilkan pada layar pengguna.

### 3.3.2. Tampilan

Berikut desain dari aplikasi ini



## 3.4. Hasil Pengembangan Aplikasi

### 3.4.1. Dokumentasi Pengembangan Aplikasi

#### 1. Dokumentasi program

Kode program dapat diakses menggunakan link berikut

[https://github.com/febyeliana/TST\\_14](https://github.com/febyeliana/TST_14)

#### 2. Dokumentasi API

Berikut ini dokumentasi API yang digunakan dari aplikasi ini

Method	Parameter query	Link Endpoint	Deskripsi	Contoh Penggunaan
GET	key, query	http://3.210.119.72:5001/weather	Endpoint yang menghubungkan informasi suatu tempat dari google API (berupa <b>latitude</b> dan <b>longitude</b> ) dengan menjadi masukan dari Open Weather API untuk mendapatkan informasi cuaca  <b>Masukan:</b>	http://3.210.119.72:5001/weather?key=YOUR_API_KEY&query=Institut Teknologi Bandung

			<p>Key : Google API Key Query : nama tempat yang ingin dicari</p> <p><b>Keluaran:</b></p> <ul style="list-style-type: none"> <li>- Koordinat,</li> <li>- Kode negara,</li> <li>- Nama kota,</li> <li>- Jumlah populasi,</li> <li>- <i>Sunrise</i>,</li> <li>- <i>Sunset</i>,</li> <li>- <i>Timezone</i>,</li> <li>- Suhu,</li> <li>- Tekanan udara,</li> <li>- Kelembapan</li> <li>- Perkiraan hujan</li> <li>- Deskripsi cuaca</li> <li>- Kecepatan angin</li> </ul>	
GET	key, query	http://3.210.119.72:5001/forecast	<p>Endpoint yang menghubungkan informasi suatu tempat dari google API (berupa <b>latitude</b> dan <b>longitude</b>) dengan menjadi masukan dari Open Weather API untuk mendapatkan informasi <i>forecast</i></p> <p><b>Masukan:</b> Key : Google API Key Query : nama tempat yang ingin dicari</p> <p><b>Keluaran:</b></p> <ul style="list-style-type: none"> <li>- Kode negara,</li> <li>- Nama kota,</li> <li>- Jumlah populasi,</li> <li>- <i>Sunrise</i>,</li> <li>- <i>Sunset</i>,</li> <li>- <i>Timezone</i>,</li> <li>- List perkiraan cuaca yang berisi: <ul style="list-style-type: none"> <li>- Jam dan tanggal,</li> <li>- Suhu,</li> <li>- Tekanan udara,</li> </ul> </li> </ul>	http://3.210.119.72:5001/forecast?key=YOUR_API_KEY&query=Institut Teknologi Bandung

			<ul style="list-style-type: none"> <li>- Kelembapan</li> <li>- Perkiraan hujan</li> <li>- Deskripsi cuaca</li> <li>- Kecepatan angin</li> </ul>	
GET	lat, lon	http://3.210.119.72:5000/weather	<p>Endpoint untuk mengakses Open Weather API dengan masukan <i>latitude</i> dan <i>longitude</i> dari lokasi untuk memperoleh informasi cuaca</p> <p><b>Masukan:</b>  Lat: <i>latitude coordinate</i>  Lon: <i>longitude coordinate</i></p> <p><b>Keluaran:</b></p> <ul style="list-style-type: none"> <li>- Koordinat,</li> <li>- Kode negara,</li> <li>- Nama kota,</li> <li>- Jumlah populasi,</li> <li>- <i>Sunrise</i>,</li> <li>- <i>Sunset</i>,</li> <li>- <i>Timezone</i>,</li> <li>- Suhu,</li> <li>- Tekanan udara,</li> <li>- Kelembapan</li> <li>- Perkiraan hujan</li> <li>- Deskripsi cuaca</li> <li>- Kecepatan angin</li> </ul>	http://3.210.119.72:5000/weather?lat=-6.89148&lon=107.6106591
GET	lat,lon	http://3.210.119.72:5000/forecast	<p>Endpoint untuk mengakses Open Weather API dengan masukan <i>latitude</i> dan <i>longitude</i> dari lokasi untuk memperoleh informasi <i>forecast</i></p> <p><b>Masukan:</b>  Lat: <i>latitude coordinate</i>  Lon: <i>longitude coordinate</i></p> <p><b>Keluaran:</b></p> <ul style="list-style-type: none"> <li>- Kode negara,</li> </ul>	http://3.210.119.72:5000/forecast?lat=-6.89148&lon=107.6106591

			<ul style="list-style-type: none"> <li>- Nama kota,</li> <li>- Jumlah populasi,</li> <li>- <i>Sunrise</i>,</li> <li>- <i>Sunset</i>,</li> <li>- <i>Timezone</i>,</li> <li>- List perkiraan cuaca yang berisi: <ul style="list-style-type: none"> <li>- Jam dan tanggal,</li> <li>- Suhu,</li> <li>- Tekanan udara,</li> <li>- Kelembapan</li> <li>- Perkiraan hujan</li> <li>- Deskripsi cuaca</li> <li>- Kecepatan angin</li> </ul> </li> </ul>	
GET	Key, query	http://3.210.119.72:1234/search	<p>Endpoint untuk memperoleh informasi detail suatu tempat dari Google API</p> <p><b>Masukan:</b>  Key: Google API Key  Query: Nama tempat yang ingin dicari</p> <p><b>Keluaran:</b></p> <ul style="list-style-type: none"> <li>- Alamat</li> <li>- Geometry <ul style="list-style-type: none"> <li>- <b>Latitude</b></li> <li>- <b>Longitude</b></li> </ul> </li> <li>- Icon</li> <li>- Id</li> <li>- Name</li> <li>- Phone number</li> <li>- List of Photos <ul style="list-style-type: none"> <li>- Height</li> <li>- References</li> <li>- Html</li> </ul> </li> <li>- Place_id</li> <li>- Rating</li> <li>- List of reviews</li> <li>- Google map URL</li> <li>- Website</li> <li>- status</li> </ul>	.210.119.72:1234/search?key=YOUR_API_KEY&query=Institut Teknologi Bandung

### 3. Dokumentasi Log Kerja

Dokumentasi pembagian kerja dan log kerja dapat dilihat pada bagian log kerja dan trello yang dapat diakses pada link berikut

<https://trello.com/b/ijnwQNK6/18217043-tst>

#### 3.4.2. Penggunaan *Versioning Control Software*

Untuk kebutuhan *versioning control* pada aplikasi yang dikembangkan, kami menggunakan GitHub. Berikut ini adalah link untuk mengakses GitHub dari kelompok 14: [https://github.com/febyeliana/TST\\_14](https://github.com/febyeliana/TST_14)

Pada Github dapat dilihat riwayat *commit* yang dilakukan oleh setiap anggota kelompok. Berikut adalah cara yang kami gunakan untuk melakukan *versioning control* dengan menggunakan GitHub:

1. Mengecek apakah terdapat *update* terbaru yang belum ada di *repository* lokal. Langkah ini dilakukan dengan melihat *commit* pada GitHub maupun komunikasi secara langsung.
2. Melakukan *pull* dari branch master ke repository lokal dengan menggunakan *command*:  
*git pull*
3. Menambahkan file yang mengalami perubahan dengan menggunakan *command*:
  - a. Untuk menambahkan semua file:  
*git add .*
  - b. Untuk menambahkan file tertentu:  
*git add namaFile.py*
4. Melakukan *commit* dan menambahkan pesan commit sebagai deskripsi singkat mengenai perubahan yang dilakukan dengan menggunakan *command*:  
*git commit -m "isi pesan commit"*
5. Melakukan *push* untuk menambahkan perubahan pada *repository* lokal ke *branch master* dengan *command*:  
*Git push origin master*

#### 3.4.3. Perubahan Perencanaan Dalam Eksekusi Kerja

Berikut ini adalah beberapa hal dalam proposal pengajuan proyek yang mengalami perubahan dalam pengerjaannya:

1. *Deployment* aplikasi menggunakan *Virtual Machine*

Pada proposal pengajuan proyek, aplikasi akan di-*deploy* menggunakan docker. Namun, karena keterbatasan spesifikasi *hardware* pengembang, *deployment* aplikasi diubah menjadi menggunakan *Virtual Machine* dan di-*deploy* ke server AWS.



Berikut ini adalah *dependencies requirement* yang diterapkan untuk VM::

requests==2.20.0

Flask==1.1.1

flask\_cors==3.0.8

## 2. *Timeline* pengerjaan aplikasi

*Timeline* pengerjaan aplikasi menjadi 2 minggu lebih cepat dibandingkan *timeline* pada proposal pengajuan proyek karena perubahan *requirement*.

### 3.4.4. Masalah selama Proyek dan *Lesson Learned*

Berikut ini adalah berbagai masalah yang terjadi selama proyek dan *lesson learned* yang dapat diambil dari masalah tersebut.

#### 1. Perubahan *timeline* proyek

Pada pertengahan pengerjaan proyek ini, terdapat perubahan jadwal pengumpulan menjadi 2 minggu lebih cepat dibandingkan waktu pengumpulan yang telah ditetapkan sebelumnya. Akibatnya kami sebagai *developer* harus melakukan diskusi kembali terkait *timeline* proyek dan juga menetapkan target baru agar proyek dapat diselesaikan sebelum tenggat waktu.

#### ***Lesson learned:***

Dari proyek ini kami belajar bahwa proyek merupakan suatu hal yang sangat fleksibel dan sangat mungkin terjadi perubahan dalam keberjalanannya. Karena hal tersebut, seorang *developer* harus siap terhadap perubahan yang akan terjadi dan dapat beradaptasi dengan cepat terhadap perubahan agar pekerjaan dapat diselesaikan tepat waktu dan sesuai dengan harapan pemberi proyek.

#### 2. Keterbatasan spesifikasi *hardware* pengembang

Terdapat perubahan perencanaan dalam eksekusi kerja, yaitu perubahan *deployment* aplikasi yang pada saat perencanaan proyek akan menggunakan docker, tetapi saat eksekusi menjadi *virtual machine*. Hal ini terjadi karena adanya keterbatasan spesifikasi *hardware* dari *developer* sehingga Docker tidak *compatible* dengan laptop pengembang.

Permasalahan ini diselesaikan melalui diskusi anggota kelompok, dan diputuskan bahwa *deployment* akan menggunakan *virtual machine*. *Virtual machine* merupakan hal yang baru bagi kami, sehingga kami membutuhkan waktu untuk mempelajari cara

penggunaan *virtual machine* dan melakukan terhadap penggunaannya.

***Lesson learned:***

Seorang *developer* harus dapat mengidentifikasi berbagai keterbatasan yang dimiliki dalam mengembangkan suatu sistem dengan baik, agar saat pengembangan telah dilakukan tidak terjadi masalah-masalah teknis yang dapat menghambat pengembangan.

Selain itu, *developer* harus dapat menguasai berbagai ilmu terkait pengembangan sistem agar saat terjadi suatu masalah dalam melakukan pengembangan, *developer* dapat mencari alternatif lain untuk menyelesaikan masalah tersebut.

3. *Subscribe* Google API

Dalam melakukan perencanaan dan pembuatan desain API, kami tidak mempertimbangkan penggunaan Google API yang membutuhkan *key* dari akun berbayar/premium. Hal ini dikarenakan pada awalnya, Google API dapat dipanggil menggunakan *key* dari akun gratis walaupun terdapat keterbatasan kuota penggunaan. Namun, pengembangan sistem menjadi terhambat dikarenakan batasan kuota ini. Masalah ini dapat diselesaikan dengan membuat akun *google developer* premium.

***Lesson learned:***

*Developer* harus dapat mengidentifikasi berbagai risiko yang mungkin muncul dalam pengembangan sistem dan juga dapat melakukan mitigasi risiko dalam menghadapi risiko yang mungkin terjadi.

Selain itu, *developer* diharapkan berpikir untuk jangka panjang agar saat mengembangkan suatu sistem yang besar dan kompleks tidak terjadi berbagai masalah yang dapat mengurangi kualitas aplikasi/sistem yang dibangun.

4. Penggunaan API saat pengembangan *front end* sulit karena belum adanya dokumentasi API

Saat melakukan pengembangan *frontend*, terdapat masalah dimana pengembang *front end* merasa kesulitan untuk menggunakan API yang telah dibuat oleh pengembang *back end*. Hal ini menghambat pengembangan *front end* yang dapat berdampak mundurnya *timeline* proyek.

Masalah ini diselesaikan dengan pembuatan dokumentasi API sehingga pengembang *frontend* menjadi paham dengan API yang telah dibuat oleh pengembang *back end*.

***Lesson learned:***

Saat API telah selesai dibuat, *developer* harus segera membuat dokumentasi API agar API dapat segera dimengerti dan digunakan oleh orang lain yang membutuhkan sehingga keberjalanan proyek masih sesuai dengan *timeline* yang ditetapkan di awal.

5. Penggunaan *versioning* yang sempat *clash* antar *developer*

Pada saat awal penggunaan *versioning*, sempat terjadi *clash* antar *developer* dikarenakan waktu *push* yang hampir bersamaan antara anggota tim. Hal ini menyebabkan terhambatnya pengerjaan proyek karena kode program menjadi bercampur sehingga membutuhkan waktu untuk *restore* agar kode dapat berjalan dan sesuai dengan perubahan yang telah dilakukan kedua belah pihak.

Hal ini terjadi karena kurangnya komunikasi antar anggota kelompok dalam melakukan *push* ke repository github. Penyelesaian dari masalah ini adalah dengan mengkomunikasikan secara langsung dan melakukan penjadwalan terhadap *push*.

***Lesson learned:***

Dalam menjalani proyek, dibutuhkan komunikasi yang baik antar anggota tim sehingga kemungkinan *clash* (baik yang berakibat langsung pada program ataupun *clash* kepentingan) dapat dihindari.

3.4.5. Cara Penggunaan Aplikasi

1. Aplikasi ini dapat diakses menggunakan link berikut:  
<http://3.210.119.72:5005/>
2. Berikut ini adalah tampilan utama dari aplikasi ini

---

WeLo

Search location's weather

Input Your Location:

Search!

Weather in your location is:

Temperature:

Pressure:

Humidity:

Forecast

#	Date and Time	Weather	Temperature	Pressure	Humidity
---	---------------	---------	-------------	----------	----------

---

- 
3. Pengguna dapat melakukan pencarian cuaca dari suatu tempat / lokasi

Pengguna dapat melakukan pencarian nama suatu daerah, gedung, jalan, kota, provinsi dan negara.

---

WeLo

Search location's weather

Input Your Location:

Search!

Weather in your location is:

Temperature:

Pressure:

Humidity:

Forecast

#	Date and Time	Weather	Temperature	Pressure	Humidity
---	---------------	---------	-------------	----------	----------

---

- 
4. Setelah menekan tombol *search*, aplikasi akan memberikan data terkait *weather* dan *forecast* dari lokasi/tempat

Data yang akan dikeluarkan:

- Keterangan cuaca saat ini
- Temperatur (dalam celcius) saat ini
- Tekanan udara saat ini
- Kelembaban udara saat ini
- Forecast*
  - Tanggal dan jam
  - Keterangan cuaca
  - Temperatur udara
  - Tekanan udara
  - Kelembaban udara

WeLo

Search location's weather

Input Your Location:

Institut Teknologi Bandung

Search!

Weather in your location is:

light rain

Temperature:

27.12

Pressure:

1006

Humidity:

63

Forecast

#	Date and Time	Weather	Temperature	Pressure	Humidity
1	2019-12-01 09:00:00	light rain	27.12	1006	63
2	2019-12-01 12:00:00	light rain	23	1008	83
3	2019-12-01 15:00:00	light rain	22.25	1010	85

5. Jika nama tempat tidak dapat dikenali, aplikasi dapat memberikan notifikasi bahwa tidak ada hasil pencarian cuaca atas nama tempat tersebut

Apps Koneksi - Google... Template Dok Peng...

3.210.119.72:5005 says

There is no result from the query.

OK

WeLo

Search location's weather

Input Your Location:

dadadad

Search!

Weather in your location is:

light rain

Temperature:

27.12

Pressure:

1006

Humidity:

63

Forecast

#	Date and Time	Weather	Temperature	Pressure	Humidity
1	2019-12-01 09:00:00	light rain	27.12	1006	63
2	2019-12-01 12:00:00	light rain	23	1008	83
3	2019-12-01 15:00:00	light rain	22.25	1010	85

### 3.4.6. Kekurangan Aplikasi

1. Untuk melakukan pencarian cuaca, aplikasi bergantung kepada Google sebagai penyedia layanan untuk melakukan pencarian lokasi tempat dan Open Weather sebagai penyedia layanan untuk melakukan pencarian cuaca.
2. Aplikasi tidak dapat melakukan pencarian tempat dengan nama yang belum teridentifikasi oleh Google.
3. Dalam melakukan pencarian cuaca, Open Weather hanya dapat menyediakan informasi cuaca dengan ketelitian suatu kota, sehingga untuk pencarian suatu gedung atau daerah yang lebih kecil, Open Weather akan menyediakan informasi cuaca untuk kota lokasi gedung tersebut.

#### 3.4.7. *Future Works*

Terdapat berbagai pengembangan yang dapat dilakukan untuk kedepannya:

1. Aplikasi dapat diintegrasikan dengan IoT device sehingga aktivitas dalam kegiatan agrikultur dapat terotomasi dan disesuaikan dengan kondisi cuaca.
2. Menambahkan sumber perkiraan cuaca dari berbagai sumber lain dan melakukan agregasi data sehingga didapatkan data yang lebih akurat.
3. Meningkatkan ketelitian cakupan wilayah untuk menampilkan informasi cuaca.

#### 3.4.8. Log Kerja

##### 3.4.8.1. Feby Eliana Tengry /18217030

Tanggal	Tugas
14 November 2019	Melakukan perencanaan pengerjaan proyek dan pembagian tugas
18 November 2019	Membuat desain <i>front end</i>
19 November 2019	Verifikasi desain <i>front end</i> dan <i>endpoint backend</i> dengan kebutuhan
21 November 2019	Membuat <i>repository github</i>
22 November 2019	Melakukan pengembangan <i>front end</i>
27 November 2019	Melakukan <i>unit testing front end</i>
28 November 2019	Melakukan integrasi <i>back end</i> dan <i>front end</i>
29 November 2019	Revisi <i>front end</i> menambahkan tampilan data
29 November 2019	Revisi <i>front end</i> menambahkan <i>error handling</i>
30 November 2019	<i>System testing</i>
1 Desember 2019	Membuat laporan & dokumentasi

##### 3.4.8.2. Muhammad Yanza Hattari /18217043

Tanggal	Tugas
14 November 2019	Melakukan perencanaan pengerjaan proyek dan pembagian tugas

18 November 2019	Membuat desain API yang akan digunakan
19 November 2019	Verifikasi desain API dan <i>back end</i> apakah sesuai dengan kebutuhan
22 November 2019	Melakukan pengembangan <i>back end</i>
26 November 2019	Melakukan unit testing
28 November 2019	Membantu integrasi <i>back end</i> dan <i>front end</i>
29 November 2019	Revisi <i>back end</i> yang digunakan ( <i>error handling</i> )
30 November 2019	<i>System testing</i>
30 November 2019	Melakukan <i>deployment</i> ke server
1 Desember 2019	Membuat laporan & dokumentasi