



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00201852868, 6 November 2018

Pencipta

Nama : **Fatmasari, Yesi Novaria Kunang, , dkk**

Alamat : **Jalan Sutan Syahril Lr.Masjid No.1610 Rt.016 Rw.004 Kel.5 Ilir
Kec.IT II, Pslembang, Sumatera Selatan, 30115**

Kewarganegaraan : **Indonesia**

Pemegang Hak Cipta

Nama : **Fatmasari, Yesi Novaria Kunang, , dkk**

Alamat : **Jalan Sutan Syahril Lr.Masjid No.1610 Rt.016 Rw.004 Kel.5 Ilir
Kec.IT II, Palembang, Sumatera Selatan, 30115**

Kewarganegaraan : **Indonesia**

Jenis Ciptaan : **Laporan Penelitian**

Judul Ciptaan : **PENGEMBANGAN DATA WAREHOUSE CUACA
SUMATERA SELATAN**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : **5 November 2018, di Palembang**

Jangka waktu perlindungan : **Berlaku selama hidup Pencipta dan terus berlangsung selama 70 (tujuh puluh) tahun setelah Pencipta meninggal dunia, terhitung mulai tanggal 1 Januari tahun berikutnya.**

Nomor pencatatan : **000124304**

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Fatmasari	Jalan Sutan Syahril Lr.Masjid No.1610 Rt.016 Rw.004 Kel.5 Ilir Kec.IT II
2	Yesi Novaria Kunang	Jalan Kijang Mas Blok E-12 No.4431 Rt.041 Rw.011 Kel.Demang Lebar Daun
3	Susan Dian Purnamasari	JL. BANTEN LRG. KARYA JASA II NO. 549 RT 09 RW 09 PLAJU

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Fatmasari	Jalan Sutan Syahril Lr.Masjid No.1610 Rt.016 Rw.004 Kel.5 Ilir Kec.IT II
2	Yesi Novaria Kunang	Jalan Kijang Mas Blok E-12 No.4431 Rt.041 Rw.011 Kel.Demang Lebar Daun
3	susan Dian Purnamasari	JL. BANTEN LRG. KARYA JASA II NO. 549 RT 09 RW 09 PLAJU



PENGEMBANGAN DATA WAREHOUSE CUACA SUMATERA SELATAN

Fatmasari¹, Yesi Novaria Kunang², Susan Dian Purnamasari³

^{1,2,3}Fakultas Ilmu Komputer, Universitas Bina Darma

Email : fatmasari@binadarma.ac.id¹, yesinovariakunang@binadarma.ac.id²,
susandian@binadarma.ac.id³

Abstrak. Tujuan penelitian ini adalah mengembangkan datawarehouse atau pangkalan data cuaca kota Palembang sekitarnya dengan menggunakan teknik web scraping. Untuk itu pada penelitian ini akan memanfaatkan teknologi web scraping atau web scraping yaitu teknik untuk mengambil isi sebuah halaman web secara spesifik dengan Pentaho Kettle, untuk mengumpulkan data cuaca di stasiun cuaca di kota Palembang dan sekitarnya. Di kota Palembang dan sekitarnya memiliki beberapa titik pengukuran cuaca yang dimiliki oleh beberapa instansi yang berbeda seperti BMKG, Angkasa Pura, Lapan dan lainnya akan tetapi untuk mendapatkan data cuaca terbaru secara detail dalam perioda tertentu terkendala proses birokrasi ke masing-masing instansi yang berbeda. Data yang dikumpulkan dengan teknik web scraping ini nantinya akan membentuk pangkalan data atau data warehouse yang bisa dimanfaatkan untuk penelitian lebih lanjut untuk data mining perkiraan cuaca kota Palembang dan sekitarnya, yang kemudian ke depannya bisa dikembangkan aplikasi cuaca berbasis Sistem Informasi Geografis.

Kata Kunci : *scraping, Web scraping, data cuaca, Pentaho Kettle*

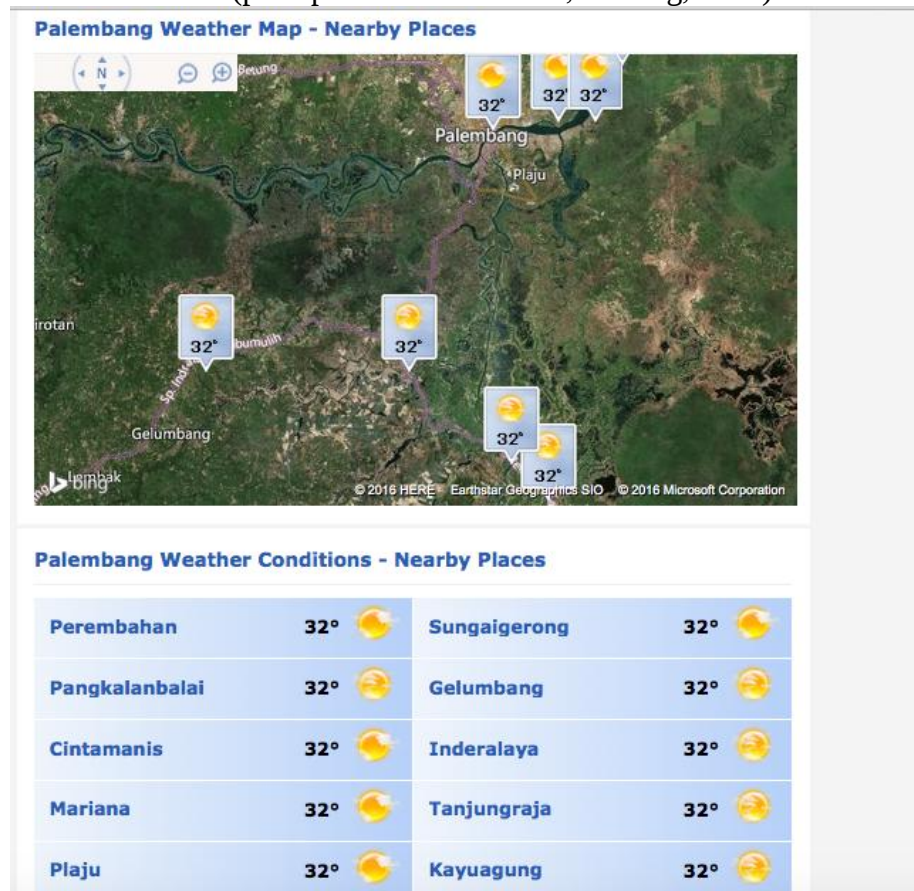
1. Pendahuluan

Seiring dengan fenomena cuaca ekstrim di tahun 2015 dan kejadian bencana alam yang diakibatkan oleh faktor cuaca sering terjadi, seperti peristiwa banjir, kemarau panjang yang menyebabkan gagal panen dan lain-lain maka diperlukan tersedianya sistem yang bisa memantau, menyediakan dan memberikan informasi prakiraan cuaca yang sesuai dengan kebutuhan masyarakat. Disisi lain masyarakat saat ini memerlukan informasi cuaca yang tepat dan cepat diterima oleh masyarakat. Hal tersebut dikarenakan cuaca merupakan faktor yang dapat menentukan sesuatu menjadi baik atau tidak, atau sesuatu bisa dijalankan atau tidak. Contohnya pada dunia industri pada sektor transportasi, kegiatan transportasi akan lebih cepat sampai pada tujuannya apabila kondisi jalan kering dan tidak turun hujan atau berbagai macam kondisi cuaca lainnya yang belum dapat diprediksi sebelumnya.

Untuk memperkirakan cuaca tidaklah mudah karena cuaca selalu berubah, selain itu perlu dipelajari sifat atmosfer atau dinamika atmosfer. Pendekatan-pendekatan dalam membuat prakiraan cuaca sangat tergantung dari data hasil pengamatan dan prosedur serta metode prakiraan cuaca yang digunakan. Untuk memperkuat analisis prakiraan cuaca maka diperlukan data pengukuran tidak hanya di satu titik saja tetapi di beberapa titik untuk melihat pergerakan atmosfer, pergerakan awan, angin, dan lain-lain sehingga hasil prakiraan cuaca tersebut lebih valid.

Di kota Palembang dan sekitarnya memiliki beberapa titik pengukuran cuaca yang dimiliki oleh beberapa instansi yang berbeda seperti BMKG, Angkasa Pura, Lapan dan lainnya (lihat gambar 1), akan tetapi untuk mendapatkan data cuaca terbaru secara detail dalam perioda tertentu sangat sulit dikarenakan proses birokrasi ke masing-

masing instansi yang berbeda. Di sisi lain beberapa lokasi stasiun cuaca itu datanya bisa diakses di beberapa situs seperti <http://www.accuweather.com>, <http://www.timeanddate.com>, <http://www.bmkg.go.id> dan data yang didapat dari AWS Universitas Bina Darma (pada penelitian terdahulu, Kunang, 2015).



Gambar 1. Lokasi Beberapa Stasiun Weather di situs www.accuweather.com

Untuk itu pada penelitian ini akan memanfaatkan teknologi *screen scraping* atau webscraping yaitu teknik untuk mengambil isi sebuah halaman web secara spesifik, misalnya link gambar, isi berita atau sejenisnya untuk mengumpulkan data cuaca di stasiun cuaca di Palembang dan sekitarnya. Data yang dikumpulkan dengan teknik *screen scraping* ini nantinya akan membentuk pangkalan data atau datawarehouse yang bisa dimanfaatkan untuk penelitian lebih lanjut untuk *data mining* perkiraan cuaca kota Palembang dan sekitarnya.

2. Metode Penelitian

Penelitian ini merupakan penelitian tindakan (*action research*) yang langsung diimplementasikan pada situasi aktual yang ada dengan mengeksplorasi beberapa data di website cuaca. Pemilihan metoda ini dikarenakan *action research* bersifat praktis dan bisa langsung diimplemetasikan. Kerangka kerja yang diikuti untuk mendapatkan pemecahan masalah adalah sebagai berikut :

1. Melakukan diagnosa (*Diagnosing*).
Pada tahapan ini dilakukan identifikasi masalah-masalah pokok yang ada. Mencari beberapa website yang menyediakan data cuaca di kota Palembang.
2. Membuat rencana tindakan (*Action Planning*)

Pada tahapan ini kita memahami pokok masalah dengan melakukan studi pustaka yang berhubungan dengan teknologi Sistem Informasi monitoring cuaca, serta menyusun rencana tindakan yang tepat untuk menyelesaikan masalah yang ada.

3. Melakukan tindakan (*Action Taking*)

Pada tahapan ini kita mengimplementasikan rencana tindakan dengan harapan dapat menyelesaikan masalah. Pada tahapan ini akan dibuat perancangan (desain job dan transform) *web scraping* dengan tools Pentaho Kettle dari beberapa website yang menyediakan informasi cuaca.

4. Melakukan evaluasi (*Evaluating*)

Pada tahapan ini kita evaluasi hasil dari hasil implementasi. Setelah dirancang dan dibuat hasil job dan transformasi menarik data dengan *web scraping*. Di evaluasi hasil data yang ditarik dan disimpan di database.

5. Pembelajaran (*Learning*)

Pada tahap ini kita melakukan *review* tahapan-tahapan yang telah berakhir dan mempelajari kriteria dalam prinsip pembelajaran. Dari hasil evaluasi bisa didapatkan kelebihan dan kekurangan *prototype* pangkalan data cuaca yang dikembangkan.

3. Hasil dan Pembahasan

1. Aplikasi Web Scraping Data Cuaca Sumsel

Web scraping (atau dikenal juga dengan istilah web harvesting atau web data extraction) merupakan teknik yang digunakan perangkat lunak komputer untuk mengekstrak informasi dari website. Pada pengembangan aplikasi Web Scarping Data Cuaca ini menggunakan HTML Parsing dalam bahasa pemrograman python, dengan memanfaatkan library BeautifulSoup library.

Proses Instalasi

Pada pengembangan aplikasi web scraping cuaca ini digunakan [Beautiful Soup 4](#) dan library [Requests](#) pada Python 3.6 pada komputer Windows dengan platform Anaconda.

Proses instalasi BeautifulSoup dan Requests dijalankan dengan [pip](#):

```
$ pip install requests
```

```
$ pip install beautifulsoup4
```

Tentang Beautiful Soup

Beautiful soup berdasarkan situs developernya

<https://www.crummy.com/software/BeautifulSoup/>. Dijelaskan pada situsnya Beautiful Soup yang dikembangkan sejak 2004 membantu programmer untuk mempermudah pekerjaan scraping dengan mendapatkan dan mengumpulkan data dari web yang diakses tanpa mengganggu dan merusak web tersebut.

Fitur yang dimiliki Beautiful Soup:

Beautiful Soup menyediakan metode sederhana dengan Python untuk melakukan proses navigasi, mencari, dan memodifikasi pohon parse: atau dengan kata lain beautiful soup merupakan toolkit untuk membedah sebuah dokumen halaman web dan mengambil data yang dibutuhkan, tanpa banyak menulis kode aplikasi. Beautiful Soup secara otomatis untuk merubah dokumen incoming ke dalam Unicode dan outgoing dokumen dalam format UTF-8. Kecuali dokumen tidak menentukan format

encodingnya, dan BeautifulSoup tidak bisa mendeteksinya. Maka programmer hanya perlu menentukan original encoding. BeautifulSoup termasuk top popular Python parsers seperti lxml and html5lib, yang memungkinkan kita bisa mencoba berbagai strategi parsing berbeda atau mengevaluasi kecepatan untuk fleksibilitas

Proses Ekstraksi URL Website

Adapun situs yang dicrawling adalah:

<https://www.worldweatheronline.com>

Adapun script crawling bisa dilihat sebagai berikut:

Berikut ini merupakan header dari program yang memanggil library yang dibutuhkan termasuk library utama yaitu requests dan BeautifulSoup

```
# web scraping data csv file python21`
```

```
from requests import get
```

```
from bs4 import BeautifulSoup
```

```
import csv
```

```
import os
```

```
import datetime
```

```
import pytz
```

Di bawah ini adalah untuk mengambil waktu saat melakukan scraping berdasarkan waktu yang dijadwalkan di Server setiap jam. Perintah berikut akan membaca waktu server kemudian mengkonversinya ke Waktu Indonesia Barat.

```
# inialisasi jam untuk convert utc london to jakarta
```

```
# inialisasi waktu untuk hari ini
```

```
time = datetime.datetime.now()
```

```
now_utc = datetime.datetime.utcnow()
```

```
local_tz = pytz.timezone('Asia/Jakarta')
```

```
now_utc = pytz.utc.localize(now_utc)
```

```
local_time = now_utc.astimezone(local_tz)
```

```
waktu_now = local_time.strftime('%H')
```

```
date_now = local_time.strftime('%d/%m/%y:%H:%M:%S')
```

Class berikut menyimpan library fungsi yang dibutuhkan untuk proses web scraping

```
class Wathers():
```

```
    """container for a base web scraping at here"""
```

Library berikut untuk penamaan file hasil yang disimpan berdasarkan nama situs website, kota

```
    def __init__(self, kota):
```

```
        self.situs = []
```

```
        self.citys = kota
```

```
        self.path_of_directory = os.path.dirname(os.path.realpath(__file__))
```

```
        self.name_file = []
```

library untuk membuka link <https://www.worldweatheronline.com> berdasarkan nama kota di sumatra selatan yang sudah ditentukan di variabel citys

```
    def mainparse(self):
```

```
        # lakukan parsing data from
```

```
        for i in range(len(self.citys)) :
```

```
            self.situs.append('https://www.worldweatheronline.com/{ }-weather/south-sumatra/id.aspx?day=0&tp=1#hourly'.format(self.citys[i]))
```

```
            # membuat nama file berdasarkan kota kota
```

```

        self.name_file.append('worldweatheronline-{}-south-
sumatra.csv'.format(self.citys[i]))
        # get data from situs situs yang sudah di parsing
        self.get_data()
memeriksa apakah data dari website bisa diambil
    def get_data(self):
        # mulai parsing data disini base main program disini
        response = []
        for i in range(len(self.situs)) :
            response.append(get(self.situs[i]))
        # pastikan apakah bisa di get atau tidak
        for x in range(len(response)) :
            if (response[x].status_code != 200) :
                print('kesalahan ketika request dari
{}'.format(self.citys[x]))
            else :
                # print('data dari kota {} sumatera selatan bisa di
ambil'.format(self.citys[x]))
                base_of_content = self.content(response[x])
                self.select(base_of_content, self.citys[x],
self.name_file[x])

```

Proses pengambilan data dengan library BeautifulSoup

```

    def content(self, res) :
        # content of html document here
        return BeautifulSoup(res.content, 'html.parser')
    def select(self, base_of_content, city, name_file) :
        # melakukan konfersi disini pasing data
        base_content =
base_of_content.select('.weather_tb.tb_without_img.tb_time_vertical.tb_time_long
div.tb_content .tb_row')
        # for i in base_content :
        for i in range(len(base_content)) :
            # pengkondisian dilakukan dengan menggunakan jam sekarang
yang disesuaikan dengan jam yang ada di situs web
            kondisi = base_content[i].select('.tb_cont_item')[0].get_text()
            if(kondisi[:2] == waktu_now):
                # jika jam dan waktu sama dengan jam sekarang maka
ambil data tersebut
                # print('{} sama dengan waktu sekarang untuk lokasi {}'.
format(kondisi[:2], city))
                hour =
base_content[i].select('.tb_cont_item')[0].get_text()
                weather_Forecast =
base_content[i].select('.tb_cont_item')[1].select('img.tb_item_img')[0].attrs
                temp =
base_content[i].select('.tb_cont_item')[2].get_text()
                feels =
base_content[i].select('.tb_cont_item')[3].get_text()
                rain = base_content[i].select('.tb_cont_item')[4].get_text()

```

```

        cloud =
base_content[i].select('.tb_cont_item')[5].get_text()
        wind =
base_content[i].select('.tb_cont_item')[6].get_text()
        precipit =
base_content[i].select('.tb_cont_item')[7].get_text()
        gust = base_content[i].select('.tb_cont_item')[8].get_text()
        wind_dir =
base_content[i].select('.tb_cont_item')[9].get_text()
        humidity =
base_content[i].select('.tb_cont_item')[10].get_text()
        pressure =
base_content[i].select('.tb_cont_item')[11].get_text()

```

```

# pengisian data
data_select_tag = {
    'Time' : hour,
    'Date' : date_now,
    'Weather_Forecast' : weather_Forecast['title'],
    'Temperatur' : temp[:2] + ' C',
    'Feels' : feels[:2] + ' C',
    'Rain' : rain,
    'Cloud' : cloud,
    'Wind' : wind,
    'Precipt' : precipit,
    'Gust' : gust,
    'Wind_dir' : wind_dir,
    'Humidity' : humidity,
    'Pressure' : pressure,
}

```

```

# check data for store data in csv file
if(self.check_of_file(name_file)):
    # jika data sudah ada di directory maka
perbolehkan penambahan data ke dalam csv file
    self.update_data(name_file, data_select_tag)
    break
else :
    # buat file setelah directory berhasil dibuat karena
file tidak ada
    self.create_csv(name_file, data_select_tag)
    break

```

```

# pengecekan file untuk penambahan data
def check_of_file(self, name_file):
    # check directory terlebih dahulu jika kosong buat folder jika tidak
langsung cek file
    if(not
os.path.isdir(os.path.join(self.path_of_directory,'data_worldweatheronline'))):

```

```

        # jika tidak ada maka buat folder dan return false
        print('make a folder data_worldweatheronline'.center(30, '*'))
        os.mkdir(os.path.join(self.path_of_directory,
'data_worldweatheronline'))
        return False
    else :
        # check file directory sudah dibuat
        file_of_path = os.path.join(self.path_of_directory,
'data_worldweatheronline', name_file)
        if(os.path.isfile(file_of_path)) :
            # jika ada file di direcotoy data_weaters maka return true
            return True
        else :
            return False
    def create_csv(self, name_of_file, data_file) :
        # awal pendefinisian path file
        file_of_path = os.path.join(self.path_of_directory,
'data_worldweatheronline', name_of_file)
        with open(file_of_path, 'w') as file:
            dataHeader = [i for i in data_file.keys()]
            writer = csv.DictWriter(file, fieldnames= dataHeader)
            writer.writeheader()
            writer.writerow(data_file)
            print('success to write file{}'.format(name_of_file))
    def update_data(self, name_file, data_file) :
        file_of_path = os.path.join(self.path_of_directory,
'data_worldweatheronline', name_file)
        # adding data at file at here
        with open(file_of_path, 'a') as file :
            writer = csv.writer(file)
            writer.writerow([data_file[i] for i in data_file])
        print('succeess append file {}'.format(name_file))
# main file disini
if __name__ == '__main__':
    # kota kota lainnya yang akan di ambil datanya
    citys = ['Baturaja', 'Indralaya', 'Pageralam', 'Palembang', 'Prabumulih',
'Lubuklinggau', 'Talang-Selapan', 'Lahat']
    mulai = Wathers(citys)
    mulai.mainparse()

```

Hasil Ekstraksi URL Website

Hasil ekstrasi data dari situs: <https://www.worldweatheronline.com>

Enter new file name	New empty file	New
worldweatheronline-Baturaja-south-sumatra.csv		2018-07-11 04:31 22.4 KB
worldweatheronline-Indralaya-south-sumatra.csv		2018-07-11 04:31 22.3 KB
worldweatheronline-Lahat-south-sumatra.csv		2018-07-11 04:31 22.2 KB
worldweatheronline-Lubuklinggau-south-sumatra.csv		2018-07-11 04:31 22.7 KB
worldweatheronline-Pageralam-south-sumatra.csv		2018-07-11 04:31 22.2 KB
worldweatheronline-Palembang-south-sumatra.csv		2018-07-11 04:31 22.2 KB
worldweatheronline-Prabumulih-south-sumatra.csv		2018-07-11 04:31 21.7 KB
worldweatheronline-Talang-Selapan-south-sumatra.csv		2018-07-11 04:31 22.5 KB

Contoh file hasil crawling yang dilakukan tiap jam bisa dilihat seperti berikut:

A1	Gust,Precipt,Humidity,Time,Wind,Weather_Forecast,Rain,Pressure,Feels,Temperatur,Cloud,Date,Wind_dir														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Gust,Precipt,Humidity,Time,Wind,Weather_Forecast,Rain,Pressure,Feels,Temperatur,Cloud,Date,Wind_dir														
2	5 mph,4 mph,60%,13:00,75%,Partly cloudy,0.1 mm,1010 mb,35 C,31 C,43%,14/06/18:13:07,SE														
3	5 mph,4 mph,56%,14:00,83%,Patchy rain possible,0.2 mm,1009 mb,36 C,32 C,64%,14/06/18:14:31:10,ESE														
4	5 mph,4 mph,59%,15:00,60%,Partly cloudy,1.3 mm,1009 mb,35 C,31 C,46%,14/06/18:15:31:04,E														
5	5 mph,3 mph,58%,16:00,41%,Light rain shower,0.6 mm,1009 mb,36 C,31 C,23%,14/06/18:16:31:11,E														
6	4 mph,3 mph,57%,17:00,21%,Partly cloudy,0.0 mm,1008 mb,36 C,32 C,0%,14/06/18:17:31:26,E														
7	4 mph,3 mph,67%,18:00,39%,Patchy rain possible,0.2 mm,1009 mb,34 C,30 C,31%,14/06/18:18:31:25,E														
8	4 mph,2 mph,76%,19:00,56%,Partly cloudy,0.5 mm,1009 mb,32 C,29 C,63%,14/06/18:19:31:49,E														
9	4 mph,2 mph,85%,20:00,73%,Patchy rain possible,0.7 mm,1010 mb,30 C,28 C,94%,14/06/18:20:31:57,E														
10	6 mph,3 mph,97%,21:00,83%,Patchy rain possible,3.0 mm,1011 mb,31 C,28 C,98%,14/06/18:21:31:45,ESE														
11	5 mph,2 mph,97%,22:00,79%,Moderate or heavy rain shower,2.1 mm,1011 mb,30 C,27 C,97%,14/06/18:22:31:41,SE														
12	4 mph,2 mph,97%,23:00,75%,Patchy rain possible,1.1 mm,1012 mb,29 C,26 C,97%,14/06/18:23:31:41,SE														
13	4 mph,2 mph,98%,00:00,79%,Patchy rain possible,1.1 mm,1012 mb,28 C,25 C,97%,15/06/18:00:31:43,SE														
14	4 mph,2 mph,98%,01:00,83%,Patchy rain possible,1.2 mm,1012 mb,27 C,24 C,96%,15/06/18:01:31:42,SSE														
15	4 mph,2 mph,98%,02:00,87%,Patchy rain possible,1.2 mm,1012 mb,26 C,23 C,96%,15/06/18:02:31:42,S														
16	4 mph,2 mph,98%,03:00,87%,Fog,1.5 mm,1012 mb,26 C,23 C,65%,15/06/18:03:31:08,SSE														
17	3 mph,2 mph,98%,04:00,93%,Patchy rain possible,0.8 mm,1012 mb,25 C,23 C,33%,15/06/18:04:31:07,SSE														
18	3 mph,2 mph,98%,05:00,100%,Fog,0.0 mm,1012 mb,25 C,23 C,0%,15/06/18:05:31:06,SSE														

Proses Ekstraksi dari Website timeanddate dan situs weather.com

Sebagai perbandingan dilakukan proses collecting data untuk website timeanddate.com dan weather.com. Tapi kedua situs tersebut hanya mengkolek data kota Palembang (Stasiun Cuaca Sultan Mahmud badarudibn II)

<https://www.timeanddate.com/weather/indonesia/palembang/climate>

<https://weather.com/weather/hourbyhour/l/IDXX0036:1>:

```

from requests import get
from bs4 import BeautifulSoup
import csv
import os
import datetime
import pytz
# inisialisasi jam untuk convert waktu utc server london to jakarta
time = datetime.datetime.now()
now_utc = datetime.datetime.utcnow()
local_tz = pytz.timezone('Asia/Jakarta')
now_utc = pytz.utc.localize(now_utc)
local_time = now_utc.astimezone(local_tz)
waktu_now = local_time.strftime('%H')

```

```

date_now = time.strftime("%y-%m-%d:%H:%M:%S")
class Wathers():
    """container for a base web scraptig at here"""
    def __init__(self,url, situs):
        self.url = url
        self.situs = situs
        self.path_of_directry = os.path.dirname(os.path.realpath(__file__))
        self.nameOfFile = ['output-timeanddate-{}-{}.csv'.format(time.month,
time.year),
                            'output-foreca-{}-{}.csv'.format(time.month, time.year),
                            'output-weather-{}-{}.csv'.format(time.month, time.year)]
    def get_data(self):
        # mulai parsing data disini
        response = get(self.url)
        if(response.status_code != 200 ):
            print('error when requests from {}'.format(self.url))
        else :
            self.content(response)
    def content(self, res) :
        # content of html document here
        base_of_content = BeautifulSoup(res.content, 'html.parser')
        self.select(base_of_content)
    def select(self, base_of_content) :
        # data pertama disini
        if(self.situs == 'timeanddate.com') :
            fells = base_of_content.select('#qlook p')
            # suhu hari jam ini
            suhu = base_of_content.select('div.h2')[0].get_text()
            deskripsi_cuaca = fells[0].get_text()
            # data feels, forecast dan wind
            fells_des = fells[1].select('br')[0].previous_sibling[11:]
            forecast = fells[1].select('br')[1].previous_sibling[11:]
            wind = fells[1].select('br span')[0].previous_sibling
            # wind_from = fells[1].select('span')[0].next_sibling
            if(fells[1].select('span')[0].next_sibling) :
                wind = wind + fells[1].select('br span')[0].next_sibling
            # data sebelah kanan
            visibility = base_of_content.select("#qfacts p
span")[5].next_sibling
            pressure = base_of_content.select("#qfacts p
span")[6].next_sibling
            humidity = base_of_content.select("#qfacts p
span")[7].next_sibling
            dew_point = base_of_content.select("#qfacts p
span")[8].next_sibling
            hour = base_of_content.select('#qfacts p span')[3].get_text()

            # lakukan penyimpanan
            data_weathers = {

```

```

        'Hour' : hour[16:],
        'Date' : date_now,
        'Suhu' : suhu[:2] + ' F' ,
        'Deskripsi_cuaca' : deskripsi_cuaca[:-1],
        'Fells_des' : fells_des[:3] + 'F',
        'Forecast' : forecast[:6] + ' F',
        'Wind' : wind[5:],
        'Visibility' : visibility[1:2] + ' mi',
        'Pressure' : pressure[1:6] + ' Hg',
        'Humidity' : humidity,
        'Dew_point' : dew_point[:3] + ' F'
    }
    # check data for store data in csv file
    if(self.check_of_file(self.nameOfFile[0])):
        # jika data sudah ada di directory maka perbolehkan
        penambahan data ke dalam csv file
        self.update_data(self.nameOfFile[0], data_weathers)
    else :
        # lakukan apa pun disini jika tidak ada file
        self.create_csv(self.nameOfFile[0], data_weathers)
    # data ketiga disini
    elif(self.situs == 'weather.com') :
        #container base untuk tabel weater per jam
        base_container = base_of_content.find('table')
        # data of weathers
        hour = base_of_content.select('.observation-
timestamp')[0].get_text()
        desc = base_container.select('tbody .hidden-cell-
sm.description')[0].get_text()
        temp = base_container.select('tbody .temp')[0].get_text()
        feels = base_container.select('tbody .feels')[0].get_text()
        precip = base_container.select('tbody .precip')[0].get_text()
        humidity = base_container.select('tbody .humidity')[0].get_text()
        wind = base_container.select('tbody .wind')[0].get_text()
        # proses make open data
        data_weathers = {
            'Hour' : hour,
            'Date' : date_now,
            'Temperatur' : temp[:2] + ' F',
            'Feels' : feels[:2] + ' F',
            'Precipt' : precip,
            'Humidity' : humidity,
            'Wind' : wind,
            'Description' : desc
        }
        # check data for store data in csv file
        if(self.check_of_file(self.nameOfFile[2])):
            # jika data sudah ada di directory maka perbolehkan
            penambahan data ke dalam csv file

```

```

        self.update_data(self.nameOfFile[2], data_weathers)
    else :
        # lakukan apa pun qdisisni jika tidak ada file
        self.create_csv(self.nameOfFile[2], data_weathers)
    else :
        print('data situs tidak tercantum')
# pengecekan file untuk penambahan data
def check_of_file(self, name_file):
    # check directory terlebih dahulu jika kosong buat folder jika tidak
    langsung cek file
    if(not os.path.isdir(os.path.join(self.path_of_directry, 'data_weathers'))):
        :
        # jika tidak ada maka buat folder dan return false
        print('make a folder data_weathers'.center(30, '*'))
        os.mkdir(os.path.join(self.path_of_directry, 'data_weathers'))
        return False
    else :
        # check file directory sudah dibuat
        file_of_path = os.path.join(self.path_of_directry,
'data_weathers', name_file)
        if(os.path.isfile(file_of_path)) :
            # jika ada file di directoy data_weathers maka return true
            return True
        else :
            return False
    def create_csv(self, name_of_file, data_file) :
        # awal pendefinisian path file
        file_of_path = os.path.join(self.path_of_directry, 'data_weathers',
name_of_file)

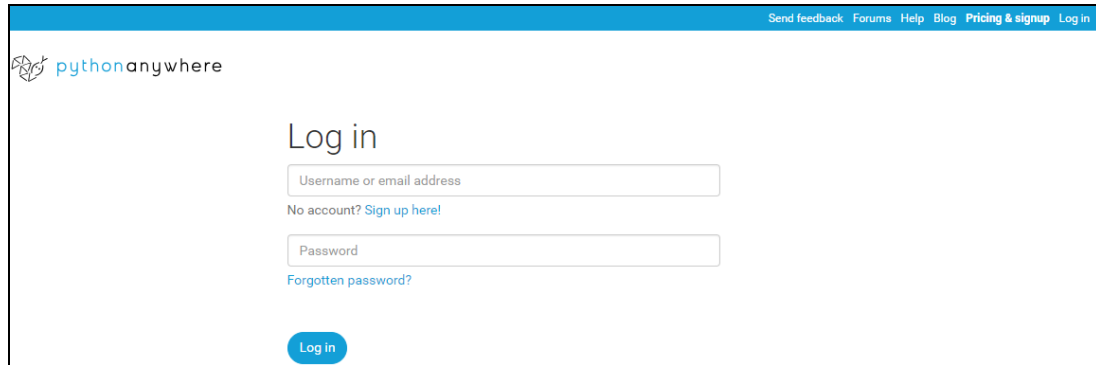
        with open(file_of_path, 'w') as file :
            dataHeader = [i for i in data_file.keys()]
            writer = csv.DictWriter(file, fieldnames= dataHeader)
            writer.writeheader()
            writer.writerow(data_file)
            print('success to write data to csv file from {}'.format(self.situs))
            print('check at {} file'.format(name_of_file))
    def update_data(self, name_file, data_file) :
        file_of_path = os.path.join(self.path_of_directry, 'data_weathers',
name_file)
        # adding data at file at here
        with open(file_of_path, 'a') as file :
            writer = csv.writer(file)
            writer.writerow([data_file[i] for i in data_file])
            print('succeess to append data to file from {}'.format(self.situs))
            print('check at {} file \n'.format(name_file))

```

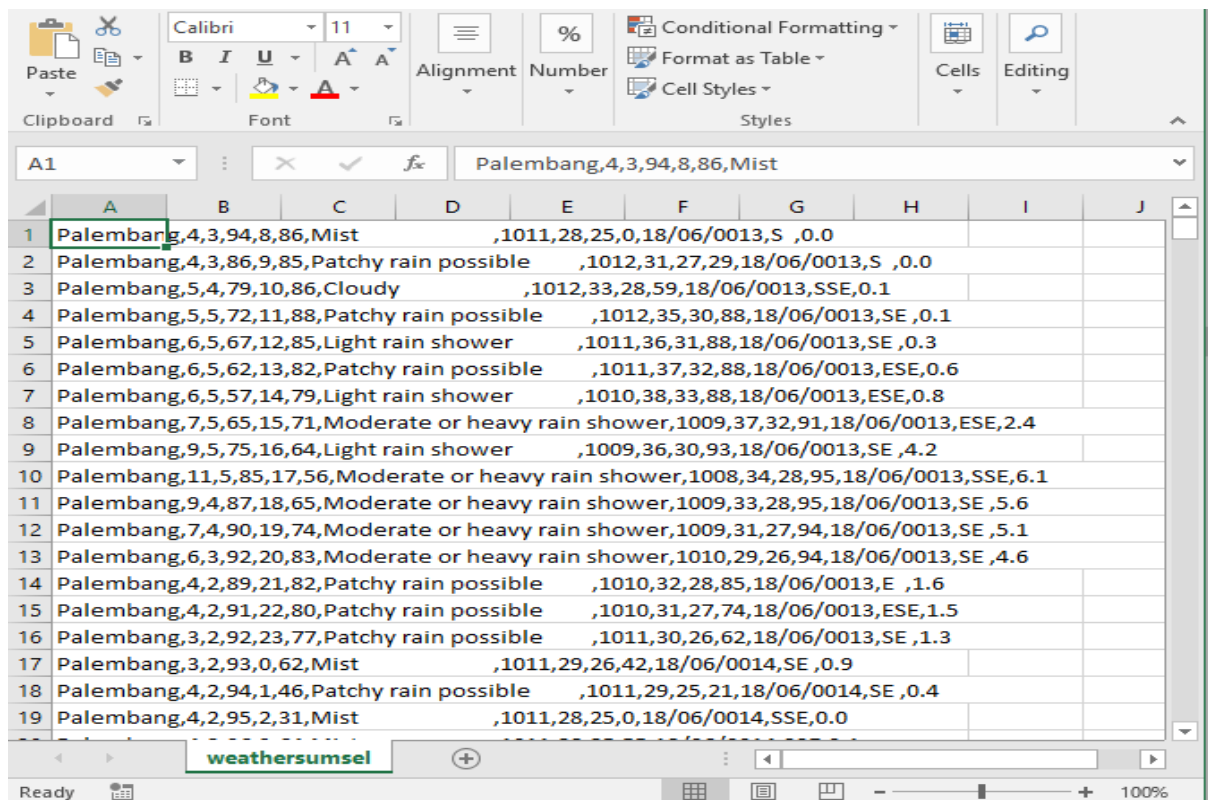

2. Proses Crawling Data di Pythonanywhere

Untuk menjalankan Script crawling data secara otomatis dan dilakukan berkala tiap jam, maka pada penelitian ini peneliti melakukan hosting script Python di server Pythonanywhere <https://www.pythonanywhere.com/>.

Pythonanywhere sendiri merupakan platform server dengan environment Python yang sudah diinstal di dalamnya. Dengan demikian kita bisa mengembangkan dan mennghosting website atau code dari browser tanpa melakukan instalasi dan manage server.



Setelah melakukan registrasi dan pembayaran untuk hosting, maka situs bisa diakses seperti berikut:



	A	B	C	D	E	F	G	H	I	J
1	Palembang,4,3,94,8,86,Mist				,1011,28,25,0,18/06/0013,S	,0.0				
2	Palembang,4,3,86,9,85,Patchy rain possible				,1012,31,27,29,18/06/0013,S	,0.0				
3	Palembang,5,4,79,10,86,Cloudy				,1012,33,28,59,18/06/0013,SSE,0.1					
4	Palembang,5,5,72,11,88,Patchy rain possible				,1012,35,30,88,18/06/0013,SE	,0.1				
5	Palembang,6,5,67,12,85,Light rain shower				,1011,36,31,88,18/06/0013,SE	,0.3				
6	Palembang,6,5,62,13,82,Patchy rain possible				,1011,37,32,88,18/06/0013,ESE,0.6					
7	Palembang,6,5,57,14,79,Light rain shower				,1010,38,33,88,18/06/0013,ESE,0.8					
8	Palembang,7,5,65,15,71,Moderate or heavy rain shower				,1009,37,32,91,18/06/0013,ESE,2.4					
9	Palembang,9,5,75,16,64,Light rain shower				,1009,36,30,93,18/06/0013,SE	,4.2				
10	Palembang,11,5,85,17,56,Moderate or heavy rain shower				,1008,34,28,95,18/06/0013,SSE,6.1					
11	Palembang,9,4,87,18,65,Moderate or heavy rain shower				,1009,33,28,95,18/06/0013,SE	,5.6				
12	Palembang,7,4,90,19,74,Moderate or heavy rain shower				,1009,31,27,94,18/06/0013,SE	,5.1				
13	Palembang,6,3,92,20,83,Moderate or heavy rain shower				,1010,29,26,94,18/06/0013,SE	,4.6				
14	Palembang,4,2,89,21,82,Patchy rain possible				,1010,32,28,85,18/06/0013,E	,1.6				
15	Palembang,4,2,91,22,80,Patchy rain possible				,1010,31,27,74,18/06/0013,ESE,1.5					
16	Palembang,3,2,92,23,77,Patchy rain possible				,1011,30,26,62,18/06/0013,SE	,1.3				
17	Palembang,3,2,93,0,62,Mist				,1011,29,26,42,18/06/0014,SE	,0.9				
18	Palembang,4,2,94,1,46,Patchy rain possible				,1011,29,25,21,18/06/0014,SE	,0.4				
19	Palembang,4,2,95,2,31,Mist				,1011,28,25,0,18/06/0014,SSE,0.0					

Output file data web scraping yang dihasilkan yang dibuat dalam bentuk file csv seperti pada gambar:

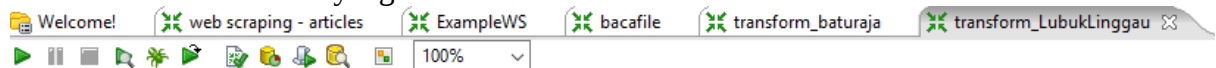
[illegible]

File data tersebut tidak bisa langsung digunakan karena terdapat satuan pada variabel data yang diambil. Selain itu juga perlu ada penyesuaian format dari data yang dibaca. Untuk itu proses dilakukan dengan menggunakan tools Pentaho.

Untuk melakukan transformasi data dari output hasil web scraping, perlu dilakukan proses pembersihan data (yang mengandung satuan) dan juga untuk menyatukan data dari beberapa kota yang dianalisis digunakan tools Pentaho.

A. FILE TRANSFORMASI DATA

Berikut file transformasi yang dibuat untuk mentransformasi data sesuai kebutuhan



Keterangan:

- a) Step pertama Baca file Kota (dalam gambar membaca file data kota Lubuk Linggau). Hasil dari step ini adalah membaca file csv kemudian merubah format data sesuai kebutuhan dan proses ini sekaligus membuang satuan dari variabel data.

CSV Input

Step name: Baca file kota Lubuk Linggau

Filename: E:\web scraping\data\world_wide\worldweatheronline-Lubuklinggau-south-sumatra.csv

Delimiter: ,

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Gust	Integer	#	15		\$	.	,	none
2	Precipt	Integer	#	15		\$	.	,	none
3	Humidity	Integer	#	15		\$	.	,	none
4	Time	Integer	#	15		\$	.	,	none
5	Wind	Integer	#	15		\$	.	,	none
6	Weather_Forecast	String		20		\$	.	,	none
7	Rain	String		6		\$	.	,	none
8	Pressure	Integer	#	15		\$	.	,	none
9	Feels	Integer	#	15		\$	.	,	none
10	Temperatur	Integer	#	15		\$	.	,	none
11	Cloud	Integer	#	15		\$	.	,	none
12	Date	Date	yyyy/MM/dd			\$	.	,	none
13	Wind_dir	String		3		\$	.	,	none

Output dari Step pertama bisa dilihat di gambar:

Execution Results

Execution History | Logging | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

#	Gust	Precipt	Humidity	Time	Wind	Weather_Forecast	Rain	Pressure	Feels	Temperatur	Cloud	Date	Wind_dir
1	3	2	79	8	16	Partly cloudy	0.0 mm	1012	29	26	0	0013/06/18	S
2	4	3	60	10	6	Partly cloudy	0.0 mm	1012	34	31	0	0013/06/18	E
3	4	3	56	11	8	Partly cloudy	0.0 mm	1011	35	31	0	0013/06/18	E
4	4	4	52	12	10	Partly cloudy	0.0 mm	1011	36	32	0	0013/06/18	E
5	4	4	47	13	12	Partly cloudy	0.0 mm	1010	37	33	0	0013/06/18	E
6	4	4	49	14	11	Partly cloudy	0.0 mm	1009	36	33	0	0013/06/18	ENE
7	4	3	50	15	10	Partly cloudy	0.0 mm	1009	36	33	0	0013/06/18	ENE
8	4	3	55	16	9	Partly cloudy	0.0 mm	1008	36	32	0	0013/06/18	NE
9	3	2	65	17	31	Patchy rain possible	0.1 mm	1009	34	31	25	0013/06/18	ENE
10	3	2	75	18	52	Partly cloudy	0.3 mm	1009	33	29	49	0013/06/18	E
11	2	1	86	19	74	Patchy rain possible	0.4 mm	1010	31	28	74	0013/06/18	ESE
12	3	2	86	20	73	Light rain shower	0.4 mm	1011	30	27	74	0013/06/18	SSW

Terlihat field rain masih belum sesuai dengan yang diinginkan (masih terdapat satuan mm), maka dilanjutkan pada step kedua.

- b) Step kedua: Fungsinya untuk merubah output field dari field Rain yang diberi nama field Rainy. Step ini akan mengambil 4 karakter dari 0 ke 3 (untuk menghilangkan karakter satuan)

String Cut

Step name: Strings cut Rain

The fields to cut:

#	In stream field	Out stream field	Cut from	Cut to
1	Rain	Rainy	0	3

Hasil outputnya seperti terlihat di bawah ini:

Execution Results

Execution History | Logging | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

#	Gust	Precip	Humidity	Time	Wind	Weather_Forecast	Rain	Pressure	Feels	Temperatur	Cloud	Date	Wind_dir	Rainy
1	3	2	79	8	16	Partly cloudy	0.0 mm	1012	29	26	0	0013/06/18	S	0.0
2	4	3	60	10	6	Partly cloudy	0.0 mm	1012	34	31	0	0013/06/18	E	0.0
3	4	3	56	11	8	Partly cloudy	0.0 mm	1011	35	31	0	0013/06/18	E	0.0
4	4	4	52	12	10	Partly cloudy	0.0 mm	1011	36	32	0	0013/06/18	E	0.0
5	4	4	47	13	12	Partly cloudy	0.0 mm	1010	37	33	0	0013/06/18	E	0.0
6	4	4	49	14	11	Partly cloudy	0.0 mm	1009	36	33	0	0013/06/18	ENE	0.0
7	4	3	50	15	10	Partly cloudy	0.0 mm	1009	36	33	0	0013/06/18	ENE	0.0
8	4	3	55	16	9	Partly cloudy	0.0 mm	1008	36	32	0	0013/06/18	NE	0.0
9	3	2	65	17	31	Patchy rain possible	0.1 mm	1009	34	31	25	0013/06/18	ENE	0.1
10	3	2	75	18	52	Partly cloudy	0.3 mm	1009	33	29	49	0013/06/18	E	0.3
11	2	1	86	19	74	Patchy rain possible	0.4 mm	1010	31	28	74	0013/06/18	ESE	0.4
12	3	2	86	20	73	Light rain shower	0.4 mm	1011	30	27	74	0013/06/18	SSE	0.4
13	3	2	79	21	12	Partly cloudy	0.0 mm	1011	32	29	0	0013/06/18	WSW	0.0
14	4	2	80	22	13	Partly cloudy	0.0 mm	1012	31	28	0	0013/06/18	SW	0.0
15	4	2	82	23	16	Partly cloudy	0.0 mm	1011	29	27	0	0013/06/18	SW	0.0
16	4	2	82	23	16	Partly cloudy	0.0 mm	1011	29	27	0	0013/06/18	SW	0.0

Terdapat field Rainy yang satuannya sudah tidak ada

c) Proses ke 3 Add Kota, digunakan untuk menambahkan Field Nama Kota

Add constant values

Step name:

Fields:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	Kota	String		30						Y

OK Cancel

Set empty string dipilih Yes, yang kemudian pada Step berikutnya dimasukan nama kota sebagai nilai default.

d) Step terakhir akan menyimpan hasil ke dalam file csv : weathersumsel.csv

Text file output

Step name:

File Content Fields

Filename: Browse...

Run this as command instead? ☐

Pass output to servlet ☐

Create Parent folder ☒

Do not create file at start ☐

Accept file name from field? ☐

File name field:

Extension:

Include stepnr in filename? ☐

Include partition nr in filename? ☐

Include date in filename? ☐

Include time in filename? ☐

Specify Date time format ☐

Date time format:

Show filename(s)...

Add filenames to result ☒

OK Cancel

Diseting Append untuk menambahkan data jika file sudah ada, dan digunakan separator ,

Text file output

Step name: cuaca_sumsel

File Content Fields

Append ☒

Separator: , Insert TAB

Enclosure: "

Force the enclosure around fields? ☐

Disable the enclosure fix? ☐

Header ☒

Footer ☐

Format: CR+LF terminated (Windows, DOS)

Compression: None

Encoding:

Right pad fields ☐

Fast data dump (no formatting) ☐

Split every ... rows: 0

Add Ending line of file:

OK Cancel

Berikut susunan field dari file csv hasil output.

Text file output

Step name: cuaca_sumsel

File Content Fields

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	N
1	Kota	String		30					none	L
2	Gust	Integer	#	15	0	\$	.	,	none	
3	Precipt	Integer	#	15	0	\$	.	,	none	
4	Humidity	Integer	#	15	0	\$	.	,	none	
5	Time	Number	#,.	15		\$	.	,	none	
6	Wind	Integer	#	15	0	\$	.	,	none	
7	Weather_Forecast	String		29					none	
8	Pressure	Integer	#	15	0	\$	.	,	none	
9	Feels	Integer	#	15	0	\$	.	,	none	
10	Temperatur	Integer	#	15	0	\$	.	,	none	
11	Cloud	Integer	#	15	0	\$	.	,	none	
12	Date	Date	dd/MM/yyyy						none	
13	Wind_dir	String		3					none	
14	Rainy	String	#	15					none	

< >

Get Fields Minimal width

OK Cancel

Setelah di running/dijalankan, maka hasilnya terlihat seperti berikut:

Execution Results

#	Gust	Precip	Humidity	Time	Wind	Weather_Forecast	Rain	Pressure	Feels	Temperatur	Cloud	Date	Wind_dir	Rainy	Kota
1	3	2	79	8	16	Partly cloudy	0.0 mm	1012	29	26	0	18/06/2013	S	0.0	
2	4	3	60	10	6	Partly cloudy	0.0 mm	1012	34	31	0	18/06/2013	E	0.0	
3	4	3	56	11	8	Partly cloudy	0.0 mm	1011	35	31	0	18/06/2013	E	0.0	
4	4	4	52	12	10	Partly cloudy	0.0 mm	1011	36	32	0	18/06/2013	E	0.0	
5	4	4	47	13	12	Partly cloudy	0.0 mm	1010	37	33	0	18/06/2013	E	0.0	
6	4	4	49	14	11	Partly cloudy	0.0 mm	1009	36	33	0	18/06/2013	ENE	0.0	
7	4	3	50	15	10	Partly cloudy	0.0 mm	1009	36	33	0	18/06/2013	ENE	0.0	
8	4	3	55	16	9	Partly cloudy	0.0 mm	1008	36	32	0	18/06/2013	NE	0.0	
9	3	2	65	17	31	Partly rain possible	0.1 mm	1009	34	31	25	18/06/2013	ENE	0.1	
10	3	2	75	18	52	Partly cloudy	0.3 mm	1009	32	29	49	18/06/2013	E	0.3	
11	2	1	86	19	74	Patchy rain possible	0.4 mm	1010	31	28	74	18/06/2013	ESE	0.4	
12	3	2	86	20	73	Light rain shower	0.4 mm	1011	30	27	74	18/06/2013	SSE	0.4	
13	3	2	79	21	12	Partly cloudy	0.0 mm	1011	32	29	0	18/06/2013	WSW	0.0	
14	4	2	80	22	13	Partly cloudy	0.0 mm	1012	31	28	0	18/06/2013	SW	0.0	
15	4	2	82	23	16	Partly cloudy	0.0 mm	1011	29	27	0	18/06/2013	SW	0.0	
16	4	2	83	24	16	Partly cloudy	0.0 mm	1011	27	24	0	18/06/2013	SW	0.0	

B. FILE JOB UNTUK MENJALANKAN SELURUH PROSES TRANSFORMASI KE -8 KOTA

Untuk menjalankan seluruh proses tranformasi dari k-8 kota dibuatlah 1 File Job yang akan menjalankan proses Transformasi secara berurut untuk ke- 8 kota

Execution results

Job / Job Entry	Comment	Result	Reason	Filename	Nr	Log date
Running Weather						
Job: Running Weather	Start of job execution		start			2018/07/11 11:22:32
START	Start of job execution		start			2018/07/11 11:22:32
START	Job execution finished	Success			0	2018/07/11 11:22:32
Transform Data Palembang	Start of job execution		Followed unconditional link	E:\web scraping\data\transform_P...	1	2018/07/11 11:22:32
Transform Data Palembang	Job execution finished	Success		E:\web scraping\data\transform_P...	1	2018/07/11 11:22:32
Transform Data Indralaya	Start of job execution		Followed link after success	E:\web scraping\data\transform_i...	2	2018/07/11 11:22:32
Transform Data Indralaya	Job execution finished	Success		E:\web scraping\data\transform_i...	2	2018/07/11 11:22:32
Transform Data Prabumulih	Start of job execution		Followed link after success	file:///E:/web scraping/data/transf...	3	2018/07/11 11:22:32
Transform Data Prabumulih	Job execution finished	Success		file:///E:/web scraping/data/transf...	3	2018/07/11 11:22:32
Transform Data Tulung Selape	Start of job execution		Followed link after success	E:\web scraping\data\transform_T...	4	2018/07/11 11:22:32
Transform Data Tulung Selape	Job execution finished	Success		E:\web scraping\data\transform_T...	4	2018/07/11 11:22:32
Transform Data Lahat	Start of job execution		Followed link after success	E:\web scraping\data\transform_la...		2018/07/11 11:22:32

Output file hasil semua tranformasi disimpan dalam file weathersumsel, seperti berikut:

	A	B	C	D	E	F	G	H	I	J
1	Palembang,4,3,94,8,86,Mist									
2	Palembang,4,3,86,9,85,Patchy rain possible									
3	Palembang,5,4,79,10,86,Cloudy									
4	Palembang,5,5,72,11,88,Patchy rain possible									
5	Palembang,6,5,67,12,85,Light rain shower									
6	Palembang,6,5,62,13,82,Patchy rain possible									
7	Palembang,6,5,57,14,79,Light rain shower									
8	Palembang,7,5,65,15,71,Moderate or heavy rain shower									
9	Palembang,9,5,75,16,64,Light rain shower									
10	Palembang,11,5,85,17,56,Moderate or heavy rain shower									
11	Palembang,9,4,87,18,65,Moderate or heavy rain shower									
12	Palembang,7,4,90,19,74,Moderate or heavy rain shower									
13	Palembang,6,3,92,20,83,Moderate or heavy rain shower									
14	Palembang,4,2,89,21,82,Patchy rain possible									
15	Palembang,4,2,91,22,80,Patchy rain possible									
16	Palembang,3,2,92,23,77,Patchy rain possible									
17	Palembang,3,2,93,0,62,Mist									
18	Palembang,4,2,94,1,46,Patchy rain possible									
19	Palembang,4,2,95,2,31,Mist									

4. Kesimpulan

1. Pada penelitian ini teknik *web scraping* berhasil dimanfaatkan untuk mengumpulkan data cuaca di beberapa kota di Sumatera Selatan. Proses *web scraping* yang dilakukan melakukan pengumpulan data dari beberapa *website* yang menyajikan data cuaca. Model *web scraping* menggunakan Bahasa pemrograman *Python* berhasil mengumpulkan data secara otomatis dan terus menerus setiap jam. Dan data yang dihasilkan sangat detail dan bisa dimanfaatkan lebih lanjut untuk keperluan *data analytics*.
2. Dari data yang dikumpulkan saat ini dihasilkan *dashboard* data statistik dari faktor-faktor cuaca secara berkala harian, mingguan, bulanan dan tahunan dari pangkalan data yang sudah dikembangkan.

Daftar pustaka

- Ahmat Josi, Leon Andretti Abdillah, Suryayusra, Jurnal Sistem Informasi (SISFO), Volume 5 no 2, September 2014, hlm 159-164, Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember, Surabaya, 2014.
- Bolin, M., Webber, M., Rha, P., Wilson, T. & Miller, R.C. Automation and customization of rendered web pages. Proceedings of the 18th annual ACM symposium on User interface software and

- technology, UIST '05, pp. 163-172. ACM, New York, NY, USA, 2005.
- Connolly ,T& Begg, Database Systems, A practical Approach To Design, Implementation, And Management. Fifth Edition. Harlow : Addison Wesley, 2010
- Eloisa Vargiu, Mirko Urru, Exploiting *web scraping* in a collaborative filtering- based approach to web advertising, Artificial Intelligence Research, Vol. 2, 2013
- ESRI, GIS for Climate Change, Redlands, CA., 2008.
- Folorunsho Olaiya, Adesesan Barnabas Adeyemo, *Application of Data Mining Techniques in Weather Prediction and Climate Change Studies*, .J. Information Engineering and Electronic Business,1, 51-59,Februari 2012
- Ian H. Witten, Eibe Frank & Mark A. Hall, Data mining practical machine learning tools and techniques, third edition, Morgan Kaufmann, Burlington, 2011
- J. Hirschey, "Symbiotic Relationships: Pragmatic Acceptance of Data Scraping," SSRN Electron. J., 2014.
- N. Marres and E. Weltevrede, "SCRAPING THE SOCIAL?: Issues in live social research," *J. Cult. Econ.*, vol. 6, no. 3, pp. 313–335, Aug. 2013.
- Rainardi, V.2008. Building A *Data warehouse*, With Examples in SQL Server. New York :Springer., 2008.
- Roldan, Maria, Pentaho 3.2 Data Integration Beginner's Guide, Packit Publishing, Birmingham, Mumbai, 2010.
- Schrenk, M. Webbots, spiders, and screen scrapers: a guide to developing Internet agents with PHP/CURL. No Starch Press, 2007.
- SubektiMujiasih, Pemanfaatan data mining untuk prakiraan cuaca, Jurnal meteorology dan geofisika volume 12 nomor 2, September 2011, 189-
- The Computer Advisor. Web site scraper the most effective tool for web data extraction, [Online] (Available at: <http://www.thecomputeradvisor.net/web-site-scraper-the-most-effective-tool-for-web-data-extraction/> [Diakses 25 Mei 2016]
- Travis M. Smith, Valliappa Lakshmanan, Utilizing google earth as a GIS platform for weather application, Januari 2006, [22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology](#)
- Yesi Novaria Kunang, Susan Dian Purnamasari, Pengembangan Sistem Informasi Monitoring Cuaca Kecamatan Seberang Ulu I Berbasis Web, Proposal Penelitian Pemula Dikti, Universitas Bina Darma, 2015.
- Y. Gorokhovich& G. Villarini, Application of GIS for processing and establishing the correlation between weather radar reflectivity and precipitation data, *Meteorol. Appl.* 12, 91–99, 2005
- Zakir, Achmad, Modul Praktis Analisa dan Prakiraan CuacaJakarta, Pusat Sistem Data dan Informasi Meteorologi, 2008.