

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Pada kesempatan ini, praktikan berkesempatan untuk mengikuti program kerja profesi di Divisi *System and Technology* PT. Indeksstat Konsultan Indonesia. Tugas utama praktikan adalah mengumpulkan data, mengolah data, dan memvisualisasikan data terkait hasil Pemilihan Kepala Daerah (Pilkada) Gubernur tahun 2020 dari Sembilan Provinsi di Indonesia. Proses pengumpulan data dilakukan menggunakan teknik *web scraping* untuk mengekstrak data dari sumber resmi pilkada2020.kpu.go.id. Data yang diambil mencakup informasi jumlah suara, persentase perolehan suara, serta data geografis berdasarkan Tempat Pemungutan Suara (TPS).

Setelah proses pengumpulan data, praktikan melakukan tahap data *cleaning* dan pengolahan data untuk memastikan data yang digunakan baik dan dapat diolah lebih lanjut. Tahapan ini meliputi pembersihan data dari duplikasi, pengisian data yang hilang, serta standarisasi format data agar sesuai dengan kebutuhan analisis. Dari data yang sudah bersih dan siap digunakan diintegrasikan ke dalam platform *Google Looker Studio* untuk dibuatkan *dashboard* yang interaktif dan informatif. Praktikan bertanggung jawab untuk mendesain dan menyusun tampilan visual dari *dashboard* ini, sehingga pengguna dapat dengan mudah mengakses dan memahami hasil Pilkada Gubernur dari Sembilan Provinsi di Indonesia secara terstruktur. Hasil yang divisualisasi mencakup grafik perolehan suara dan rangkuman statistik dari tiap calon gubernur.

Melalui proyek ini, praktikan tidak hanya mengembangkan keterampilan teknis seperti pemrograman menggunakan Python untuk *scraping data*, serta penggunaan *Looker Studio* untuk membuat visualisasi data, tetapi juga memperoleh pemahaman mendalam tentang proses pemilu dan cara terbaik untuk menyajikan informasi secara efektif kepada publik dan pemangku kepentingan.

3.2 Pelaksanaan Kerja

Pada tanggal 16 Februari 2024, praktikan memulai kerja profesi yang berlangsung hingga 16 Juni 2024 dengan penempatan di divisi *System and Technology* PT. Indekstat Konsultan Indonesia. Selama periode tersebut, praktikan berperan dalam pengembangan *dashboard* hasil Pilkada Gubernur Tahun 2020 dari Sembilan provinsi di Indonesia dengan menggunakan teknik *web scraping* dan memanfaatkan platform *Looker Studio* sebagai laman visualisasi hasil data pilkada Gubernur tahun 2020. Tugas praktikan mencakup pengumpulan data melalui web scraping menggunakan Bahasa pemrograman Python dan penyajian data dalam bentuk visualisasi interaktif yang dikembangkan dengan Looker Studio.

Sebagai bagian dari divisi *System and Technology*, praktikan berkolaborasi erat dengan tim untuk memastikan infrastruktur teknis yang mendukung proses pengolahan data berjalan dengan baik. Praktikan bertanggung jawab dalam melakukan scraping data hasil Pilkada dari situs resmi pilkada2020.kpu.go.id, serta mengelola data tersebut untuk diproses lebih lanjut. Visualisasi data yang dihasilkan kemudian disajikan dalam *dashboard* yang memudahkan pengguna dalam membaca informasi penting terkait hasil pilkada di Indonesia tahun 2020.

Selama periode kerja profesi ini, praktikan tidak hanya mengembangkan kemampuan teknisnya dalam *scraping* dan visualisasi data, tetapi juga mengasah keterampilan kolaborasi dan komunikasi dengan anggota tim. Praktikan terlibat dalam diskusi rutin untuk memastikan bahwa *dashboard* yang dibuat memenuhi standar yang diharapkan oleh PT Indekstat Konsultan Indonesia dan relevan dengan kebutuhan pengguna.

3.2.1 Pengumpulan Data

Pengumpulan data merupakan salah satu tahapan penting dalam proses analisis data, terutama untuk data yang bersumber dari internet. Salah satu metode yang digunakan adalah *web scraping*, yaitu proses pengambilan data secara otomatis dari situs web. *Web scraping* memungkinkan pengumpulan data dalam jumlah besar dari berbagai sumber yang tersedia secara *online*,

yang kemudian diolah menjadi format yang lebih mudah dianalisis dalam bentuk tabel (Rizquina & Ratnasari, 2023).

Dalam proyek pengembangan *dashboard* hasil Pilkada Gubernur tahun 2020 di Indonesia, praktikan menggunakan metode *web scraping* dengan menggunakan Bahasa pemrograman Python Selenium. Selenium adalah sebuah pustaka (*library*) yang digunakan dalam bahasa pemrograman, terutama Python dan Java, untuk otomatisasi browser. Selenium memungkinkan pengguna untuk mengontrol dan memprogram browser untuk melakukan tugas-tugas otomatis, seperti mengunjungi halaman web, mengisi formulir, mengklik tautan, atau melakukan pengujian aplikasi web.

Selenium sering digunakan dalam proses pengujian perangkat lunak, *web scraping*, dan tugas lain yang melibatkan interaksi dengan elemen-elemen situs web. Dengan kata lain, Selenium berfungsi seperti manusia yang mengoperasikan browser, sehingga memungkinkan pengguna untuk memproses situs web yang membutuhkan interaksi lebih lanjut seperti login, klik tombol, atau *scrolling*.

Dengan menggunakan *Library* Selenium memungkinkan praktikan untuk melakukan *scraping* pada situs yang memerlukan rendering dinamis dari JavaScript, mengingat data hasil Pilkada seringkali disajikan melalui elemen-elemen dinamis yang tidak dapat diakses dengan metode *scraping* statis. Dengan Selenium, praktikan dapat secara otomatis mengumpulkan data yang relevan dari situs resmi pemilu dan menyimpannya dalam format yang lebih mudah diolah untuk analisis lebih lanjut.

Berikut adalah penjelasan terkait kode Python yang digunakan dalam proses *scraping* data. Penjelasan akan dibagi menjadi 3 point utama yaitu *Import Libraries*, *Functions*, dan *Main Program*. Dimana pada bagian ***Import Libraries*** akan menjelaskan *library* apa saja yang digunakan dalam kode program *scraping* ini.

Bagian **Functions** akan membahas fungsi-fungsi yang terdapat dalam program dimana setiap fungsi memiliki peran spesifik dalam keseluruhan proses scraping dan bagian **Main Program** membahas logika utama dari program.

3.2.1.1 Import Libraries

Berikut merupakan bagian Import Libraries yang digunakan pada program web scraping.

```
import pandas as pd
import os
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
import csv
import time
```

- a. `import` digunakan untuk mengimpor modul atau pustaka eksternal ke dalam kode Python. `import` memungkinkan menambahkan pustaka yang dibutuhkan agar fungsionalitas tambahan bisa digunakan dengan meng-import library yang ada.
- b. `pandas` merupakan pustaka Python yang digunakan untuk analisis data. Pustaka ini digunakan untuk memanipulasi dan menganalisis data dalam bentuk DataFrame, yang mempermudah pengelolaan data seperti tabel. Melalui penggunaan `pandas` memungkinkan praktikan untuk membaca file Excel, mengelola data dalam bentuk tabel, serta memanipulasi dan menganalisis data dalam berbagai format (seperti CSV, Excel, dll.).
- c. `as` digunakan untuk memberi alias atau nama pendek bagi pustaka yang diimpor. Dalam hal ini, `pandas` diimpor dengan alias `pd`, sehingga semua fungsi pandas dapat diakses dengan `pd` untuk mempersingkat penulisan.
Contoh: Alih-alih menulis `pandas.read_excel()`, bisa ditulis dengan `pd.read_excel()`.

- d. **os** adalah pustaka Python standar yang menyediakan fungsi untuk berinteraksi dengan sistem operasi, seperti menghapus file, memeriksa path file, atau mengelola file. Pada kode ini, **os** digunakan untuk menghapus file CSV lama sebelum menulis data baru ke file yang sama.
- e. **from** adalah bagian dari sintaks Python yang digunakan dalam pernyataan impor modul atau pustaka. **from** digunakan untuk mengimpor bagian tertentu dari modul dengan memfokuskan import hanya pada komponen yang dibutuhkan. Jadi Ketika menggunakan **from**, Praktikan bisa mengambil fungsi, kelas, atau komponen tertentu dari modul atau pustaka tanpa mengimpor semuanya, sehingga dapat menghemat memori dan mempercepat proses.
- f. **selenium.webdriver** merupakan gabungan selenium dan webdriver dimana **selenium** adalah pustaka Python yang digunakan untuk mengotomatisasi browser dan **webdriver** menyediakan driver yang dapat berinteraksi langsung dengan browser. Dalam kode ini **webdriver** digunakan untuk mengendalikan browser Chrome secara otomatis, termasuk membuka halaman web dan mengambil data dari halaman tersebut.
- g. **Options** merupakan kelas dalam Selenium yang memungkinkan untuk mengatur berbagai opsi untuk WebDriver, seperti menjalankan browser dalam mode "*headless*" (tanpa antarmuka grafis). Dalam kode ini, Options digunakan untuk menyiapkan pengaturan khusus untuk menjalankan browser Chrome.
- h. **By** merupakan kelas dalam Selenium yang menyediakan cara untuk menemukan elemen HTML pada halaman web berdasarkan atribut seperti ID, class, tag, atau selector CSS. Contoh penggunaannya adalah `By.CSS_SELECTOR` dimana pada kode ini diperuntukkan untuk menemukan elemen HTML dengan class tertentu (misalnya, `table.table-bordered`).

- i. **csv** merupakan modul Python standar yang memungkinkan penulisan dan pembacaan data dalam format CSV (*Comma Separated Values*). Dalam kode ini, **csv** digunakan untuk menulis data hasil *scraping* ke dalam file CSV agar dapat digunakan atau dianalisis lebih lanjut.
- j. **time** merupakan pustaka Python standar yang menyediakan fungsi untuk mengelola waktu, seperti menambahkan jeda waktu dalam program. Dalam kode ini, *time.sleep()* digunakan untuk memberi jeda agar halaman web bisa termuat sepenuhnya sebelum data diambil.

3.2.1.2 Functions

1. Scrape_Table_Data

Berikut merupakan gambar yang menjelaskan implementasi kode Python untuk melakukan *scraping* data tabel dari sebuah halaman web menggunakan Selenium.

```
def scrape_table_data(url, excel_data):
    chrome_options = Options()
    driver = webdriver.Chrome(options=chrome_options)

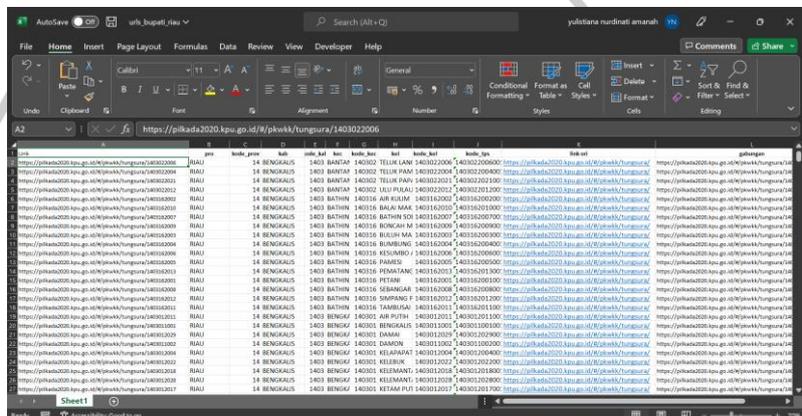
    try:
        driver.get(url)
        time.sleep(1)
        table_element = driver.find_element(By.CSS_SELECTOR, 'table.table-bordered')
        table_data = []
        for row in table_element.find_elements(By.TAG_NAME, 'tr'):
            try:
                row_data = [cell.text for cell in row.find_elements(By.TAG_NAME, 'td')]
                if row_data:
                    tps_info = row_data[0]
                    link = row.find_element(By.TAG_NAME, 'a').get_attribute('href')
                    tps_number, percentage = tps_info.split('\n')
                    excel_row = excel_data[excel_data['Link'] == url]
                    pro = excel_row['pro'].iloc[0] if 'pro' in excel_row.columns else ''
                    kab = excel_row['kab'].iloc[0] if 'kab' in excel_row.columns else ''
                    kec = excel_row['kec'].iloc[0] if 'kec' in excel_row.columns else ''
                    kel = excel_row['kel'].iloc[0] if 'kel' in excel_row.columns else ''
                    new_row = [link, f"{tps_number}", {percentage}', pro, kab, kec, kel ] + row_data[1:]
                    table_data.append(new_row)
            except Exception as e:
                print(f"Error scraping row: {e}")
                continue
        return table_data

    except Exception as e:
        print(f"An error occurred: {e}")
        return None

    finally:
        driver.quit()
```

Gambar 3.1 Kode Program Function Scrape_Table Data

- a. `def scrape_table_data(url, excel_data)` baris ini mendefinisikan (`def`) sebuah fungsi bernama `scrape_table_data`. Fungsi ini mengambil dua argumen, yaitu `url` dan `excel_data`.
- b. Argumen `url` merupakan string yang merepresentasikan *link* web yang akan diakses oleh Selenium untuk melakukan scraping. *Link* ini berisi data yang akan diekstraksi oleh Selenium, seperti tabel hasil Pilkada atau data lain yang relevan.
- c. Argumen `excel_data` adalah data yang berasal dari file Excel yang telah dibaca sebelumnya menggunakan `pandas`. Ini biasanya berupa `DataFrame` yang digunakan untuk membantu melengkapi atau memperkaya data yang diambil dari halaman web. Misalnya, data tambahan seperti informasi provinsi atau kabupaten bisa berasal dari file Excel ini.
- d. `scrape_table_data` merupakan fungsi utama dimana pada function ini berfungsi untuk mengambil data dari halaman web yang diakses menggunakan URL yang diberikan, dan menggabungkannya dengan informasi dari file Excel jika diperlukan. Dalam hal ini, URL telah di kumpulkan dalam satu file excel di mana salah satu kolomnya berisikan list URL yang akan *discraping* seperti pada **gambar 3.2**.



Gambar 3.2 File Excel Url

e. `chrome_options = Options()` baris ini mendefinisikan sebuah objek `chrome_options` yang merupakan instance dari kelas `Options()` di Selenium. Kelas `Options()` memungkinkan untuk mengonfigurasi berbagai pengaturan browser Chrome yang akan dijalankan oleh Selenium. Dengan menggunakan objek `chrome_options`, praktikan dapat menyesuaikan cara Chrome beroperasi selama proses scraping.

f. `driver=webdriver.Chrome(options=chrome_option)`
Baris ini digunakan untuk menginisialisasi Chrome WebDriver yang akan menjalankan browser Chrome secara otomatis. WebDriver adalah elemen kunci dalam Selenium yang memungkinkan otomatisasi browser. `webdriver.Chrome(options=chrome_options)`

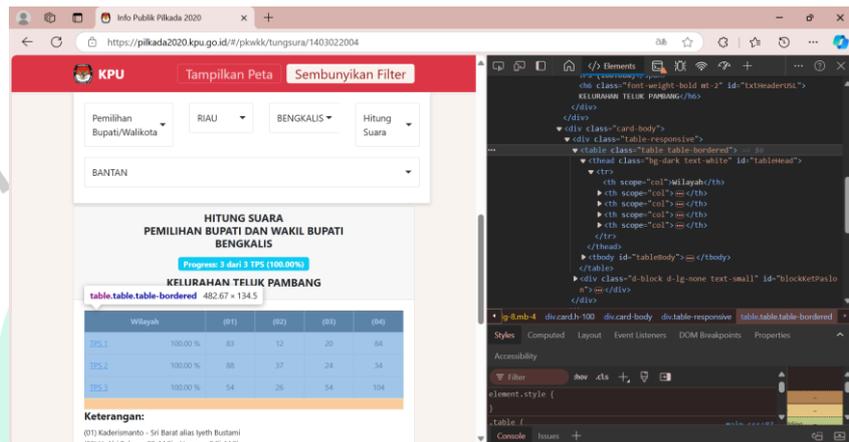
memulai instance baru dari Chrome, yang dikendalikan oleh Selenium. Selenium akan menggunakan instance ini untuk membuka halaman web, menavigasi elemen-elemen HTML, dan melakukan berbagai tugas otomatisasi dalam hal ini adalah mengambil data dari halaman.

g. Blok `try-except` pertama digunakan untuk menangani kemungkinan terjadinya exception atau kesalahan selama program berjalan. Jika terjadi kesalahan dalam blok `try` utama, program akan melompat ke blok `except` untuk menangani kesalahan tersebut, tanpa menghentikan eksekusi keseluruhan program. Kesalahan yang mungkin terjadi pada blok `try-except` pertama ini adalah Jika ada kesalahan saat navigasi halaman `driver.get(url)` atau masalah dalam mengakses elemen utama `'table.table-bordered'`, blok ini akan menangkapnya. Penerapannya adalah jika terjadi kesalahan maka akan langsung melompat pada blok `except` dengan output yang akan dikeluarkan `print(f"An error occurred: {e}")` dimana `print` sebagai perintah untuk menampilkan ke

halaman output array " `An error occurred:` dan `{e}` menampilkan jenis kesalahan apa yang terjadi. Bisa jadi karena kesalahan jaringan, kesalahan web, dll. `{e}` ini sendiri diambil dari perpendekan exception yang telah diubah penamaannya dengan `as`.

- h. Fungsi `driver.get(url)` digunakan untuk membuka halaman web dengan URL yang diberikan menggunakan browser otomatis yang dikendalikan oleh Selenium. `driver.get()` memberitahu WebDriver untuk mengakses URL yang ditentukan dan memuat halaman web tersebut. URL ini berisi data yang akan diambil melalui proses scraping. Halaman yang dibuka akan sepenuhnya dimuat dalam jendela browser sebelum instruksi selanjutnya dieksekusi.
- i. `time.sleep(1)` adalah fungsi dari pustaka `time` yang memberikan jeda eksekusi program selama 1 detik. Fungsi `time.sleep()` digunakan untuk memberi waktu agar halaman web dapat dimuat sepenuhnya sebelum Selenium mulai mencari elemen HTML di halaman tersebut. Waktu jeda (*delay*) ini sangat penting, terutama untuk situs yang memuat data secara dinamis yang mungkin memerlukan waktu beberapa detik untuk menampilkan semua elemen.
- j. Pada baris `table_element=driver.find_element(By.CSS_SELECTOR, 'table.table-bordered')`, `driver.find_element()` adalah fungsi Selenium yang digunakan untuk menemukan elemen di halaman web. Di sini, `By.CSS_SELECTOR` digunakan untuk mencari elemen berdasarkan CSS selector. `'table.table-bordered'` adalah selector yang merujuk pada elemen `<table>` dengan class `table-bordered`. Fungsi ini mencari tabel HTML di halaman web yang memiliki class `table-bordered`. Dalam konteks scraping ini, elemen tabel berisi data yang

akan diambil. Selenium mencari elemen ini dengan menggunakan CSS selector, yang merupakan cara umum untuk menargetkan elemen HTML dalam dokumen. Berikut merupakan contoh penggunaan CSS Selector pada laman Website Pilkada2020.kpu.go.id pada **gambar 3.3**.



Gambar 3.3 CSS Selector Web Pilkada dengan Inspect

- k. `table_data = []` merupakan inisialisasi dari list kosong bernama `table_data`. List ini digunakan untuk menyimpan data yang akan diambil dari setiap baris dalam tabel. Setelah elemen-elemen tabel ditemukan dan diproses, hasilnya akan dimasukkan ke dalam list `table_data`. Pada akhir proses scraping, `table_data` akan berisi seluruh data yang diambil dari tabel.
- l. Pada baris `for row in table_element.find_elements(By.TAG_NAME, 'tr')`, `table_element.find_elements(By.TAG_NAME, 'tr')` adalah instruksi Selenium yang digunakan untuk menemukan semua elemen `<tr>` (table rows atau baris tabel) di dalam elemen tabel yang telah ditemukan (`table_element`). `for row in ...` adalah `loop` yang akan melakukan iterasi melalui setiap baris tabel tersebut. Dimana Loop berjalan melalui setiap baris (`<tr>`) dalam tabel dan mengambil data dari setiap baris.
- m. Blok `try-except` kedua digunakan untuk menangani kemungkinan terjadinya exception atau kesalahan selama

program berjalan dengan kesalahan yang mungkin terjadi pada blok ini adalah masalah pada baris tertentu, seperti elemen yang hilang atau error atribut, blok ini akan menangkapnya, memungkinkan program untuk melewati baris tersebut dan melanjutkan ke baris berikutnya tanpa menghentikan proses pengambilan data.

- n. Pada baris `row_data = [cell.text for cell in row.find_elements(By.TAG_NAME, 'td')]` merupakan list comprehension yang mengekstrak teks dari setiap elemen `<td>` (table data) dalam baris tabel (`<tr>`). Kode ini mencari semua elemen `<td>` dalam baris tersebut, lalu menyimpan teks dari setiap elemen ke dalam list `row_data`.
- o. Kondisi `if row_data` : digunakan untuk memeriksa apakah `row_data` berisi data. Jika `row_data` tidak kosong (artinya ada data yang diambil dari baris tabel), blok kode di bawahnya akan dieksekusi. Batasannya adalah Hanya memproses baris yang berisi data, untuk menghindari kesalahan atau upaya pengolahan pada baris kosong.
- p. `tps_info = row_data[0]` digunakan untuk mengambil elemen pertama dari list `row_data`, yang biasanya berisi informasi TPS (Tempat Pemungutan Suara) atau informasi penting lainnya yang terdapat pada kolom pertama tabel.
- q. Pada baris `link= row.find_element(By.TAG_NAME, 'a').get_attribute('href')` Ini mencari elemen `<a>` (*hyperlink*) dalam baris tabel, lalu mengambil atribut href dari elemen tersebut, yang berisi URL tautan. Kode ini digunakan untuk mendapatkan URL terkait dengan TPS yang sedang diproses, yang mungkin berisi informasi lebih lanjut mengenai TPS tersebut. Jika elemen `<a>` tidak ada, akan muncul kesalahan, yang kemudian akan ditangkap oleh blok `except`.
- r. `tps_number, percentage = tps_info.split('\n')` baris ini digunakan untuk membagi teks yang terkandung

dalam `tps_info` berdasarkan karakter newline (`\n`). Biasanya, informasi TPS akan muncul dalam format seperti "TPS 1\n100%", yang artinya baris ini memisahkan nomor TPS dan persentase hasil.

- s. `excel_row = excel_data[excel_data['Link'] == url]` baris ini digunakan untuk mencari baris di DataFrame `excel_data` dimana kolom 'Link' sesuai dengan URL yang sedang diproses (`url`). Melalui kode ini memungkinkan pengambilan data tambahan dari file Excel dengan kolom Link. Jika URL tersebut ditemukan dalam file Excel, data terkait akan digunakan. **Gambar 3.4** menunjukkan salah satu file excel yang akan di scraping berisikan data URL.

Link	pro	kode_prov	kab	kode_kab	kec	kode_kec	kel	kode_kel	kode_kel
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403022006	RIAU	14	BENGKALIS	1403	BANTAN	140302	TELUK LAM	1403022006	1403022006
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403022004	RIAU	14	BENGKALIS	1403	BANTAN	140302	TELUK PAM	1403022004	1403022004
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403022021	RIAU	14	BENGKALIS	1403	BANTAN	140302	TELUK PAP	1403022021	1403022021
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403022012	RIAU	14	BENGKALIS	1403	BANTAN	140302	ULU PULAU	1403022012	1403022012
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162002	RIAU	14	BENGKALIS	1403	BATHIN	140316	AIR KULIM	1403162002	1403162002
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162010	RIAU	14	BENGKALIS	1403	BATHIN	140316	BALAI MAK	1403162010	1403162010
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162007	RIAU	14	BENGKALIS	1403	BATHIN	140316	BATHIN SOE	1403162007	1403162007
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162009	RIAU	14	BENGKALIS	1403	BATHIN	140316	BONCAH M	1403162009	1403162009
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162003	RIAU	14	BENGKALIS	1403	BATHIN	140316	BULUH MAI	1403162003	1403162003
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162004	RIAU	14	BENGKALIS	1403	BATHIN	140316	BUMBUNG	1403162004	1403162004
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162006	RIAU	14	BENGKALIS	1403	BATHIN	140316	KESUMBO	1403162006	1403162006
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162005	RIAU	14	BENGKALIS	1403	BATHIN	140316	PAMESI	1403162005	1403162005
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162013	RIAU	14	BENGKALIS	1403	BATHIN	140316	PEMATANG	1403162013	1403162013
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162001	RIAU	14	BENGKALIS	1403	BATHIN	140316	PETANI	1403162001	1403162001
https://pilkada2020.kpu.go.id/#/pkwkk/tungsur/1403162008	RIAU	14	BENGKALIS	1403	BATHIN	140316	SEBANGAR	1403162008	1403162008

Gambar 3.4 Data URL Riau

- t. `pro = excel_row['pro'].iloc[0] if 'pro' in excel_row.columns else ''` pada baris ini digunakan untuk mengecek apakah kolom 'pro' (provinsi) ada di DataFrame `excel_row`. Jika kolom tersebut ada, baris pertama `.iloc[0]` dari kolom 'pro' akan diambil; jika tidak ada, string kosong akan digunakan sebagai default. Proses serupa dilakukan untuk kolom lainnya 'kab' (kabupaten), 'kec' (kecamatan), dan 'kel' (kelurahan)
- u. `new_row = [link, f'"{tps_number}"', {percentage}', pro, kab, kec, kel] + row_data[1:]` pada baris ini

digunakan untuk membangun list `new_row`, yang berisi beberapa elemen yaitu `link`, `tps_number`, `percentage`, `pro`, `kab`, `kec`, `kel`. Pada bagian `row_data[1:]` maksudnya adalah sisa data dari baris tabel, dimulai dari elemen kedua (setelah `tps_info`). Fungsi dari baris ini adalah untuk menyimpan semua informasi yang diambil dari halaman web dan file Excel dalam satu baris untuk dimasukkan ke dalam list `table_data`.

v. `table_data.append(new_row)` pada baris ini terdapat fungsi `append` yang biasa digunakan digunakan untuk menambahkan `new_row`, berisikan semua data yang diambil dari halaman web dan file Excel, ke dalam list `table_data`. Fungsi dari baris kode ini adalah mengumpulkan semua data dari setiap baris tabel dalam bentuk list untuk diolah atau disimpan ke file CSV di tahap selanjutnya.

w. `return table_data` digunakan setelah semua baris tabel diolah dan datanya dikumpulkan, list `table_data` yang berisi semua baris data yang diambil dari halaman web akan dikembalikan oleh fungsi. Fungsi dari `return` ini adalah untuk mengembalikan hasil scraping, yaitu data dari semua baris tabel yang telah diproses dan digabungkan dengan data tambahan dari file Excel.

2. Save_Table_Data_to_CSV

Berikut merupakan gambar yang menjelaskan implementasi fungsi Python untuk menyimpan data tabel ke dalam file CSV.

```
def save_table_data_to_csv(table_data, csv_file):
    if table_data:
        try:
            with open(csv_file, 'a', newline='', encoding='utf-8') as file:
                writer = csv.writer(file)
                writer.writerows(table_data)
                print(f"Data dari tabel telah disimpan ke '{csv_file}'.")
        except Exception as e:
            print(f"Error occurred while saving CSV file: {e}")
    else:
        print("Tidak ada data yang berhasil diambil.")
```

Gambar 3.5 Kode Program Function Save_Table_Data_to_CSV

- a. `def save_table_data_to_csv (table_data, csv_file)` : Fungsi ini didefinisikan dengan nama `save_table_data_to_csv` dan menerima dua argument yaitu `table_data` dan `csv_file`. Baris kode ini bertugas untuk menyimpan data yang telah di-scrape ke dalam file CSV dengan nama yang diberikan (`csv_file`). Fungsi juga menangani kesalahan jika ada masalah saat menulis ke file.
- b. `if table_data`: Kondisi ini memeriksa apakah variabel `table_data` tidak kosong. Jika ada data di dalam `table_data`, maka blok kode selanjutnya akan dijalankan. Jika `table_data` kosong atau bernilai None, maka program akan melewati blok try dan langsung mencetak pesan bahwa tidak ada data yang berhasil diambil. Ini merupakan bentuk validasi awal untuk memastikan bahwa ada data yang diambil dari hasil scraping sebelum menulisnya ke file CSV. Jika `table_data` kosong, maka tidak ada gunanya membuka file CSV dan menulis data.
- c. Blok `try-except` ini digunakan untuk mencoba mengeksekusi operasi yang berpotensi menyebabkan kesalahan misalnya saat menulis ke file CSV. Jika ada kesalahan dalam proses penulisan, program akan

menangkap kesalahan tersebut di blok `except` tanpa menghentikan eksekusi program secara keseluruhan. Kegunaan dari `try-except` ini adalah untuk menangani potensi kesalahan saat membuka file dan menulis data ke CSV, misalnya jika file tidak dapat dibuka atau terjadi masalah penulisan.

- d. `with open(csv_file, 'a', newline='', encoding='utf-8') as file:` Pada baris kode ini membuka file CSV dengan nama yang diberikan di argumen `csv_file`. Mode `'a'` digunakan untuk append, artinya jika file sudah ada, data akan ditambahkan di akhir file tanpa menghapus data yang sudah ada. Jika file belum ada, Python akan membuat file baru. `newline=''` Ini menghilangkan baris kosong tambahan yang mungkin ditambahkan saat menulis data ke CSV (terutama di Windows). `encoding='utf-8'` digunakan untuk memastikan bahwa semua karakter (termasuk karakter non-ASCII) dapat ditulis ke file tanpa masalah encoding.
- e. `writer = csv.writer(file)` pada baris ini membuat objek `csv.writer` yang memungkinkan kita menulis data ke file CSV dalam format CSV standar.
- f. `writer.writerows(table_data)` Pada baris ini menulis data yang ada di dalam `table_data` ke file CSV. `writerows()` adalah fungsi yang digunakan untuk menulis beberapa baris data ke dalam file CSV. Jika `table_data` berisi beberapa baris data, semua baris akan ditulis ke dalam file CSV secara berurutan.
- g. `print(f"Data dari tabel telah disimpan ke '{csv_file}'.")` Jika proses penulisan data ke file CSV berhasil, baris ini akan mencetak pesan konfirmasi ke layar, yang memberi tahu bahwa data telah berhasil disimpan ke file dengan nama yang diberikan.

h. `else: print("Tidak ada data yang berhasil diambil.")` Blok ini dijalankan jika kondisi `if table_data` tidak terpenuhi, artinya `table_data` kosong atau tidak ada data yang tersedia untuk disimpan.

3.2.1.3 Program Utama

Berikut merupakan gambar yang menjelaskan bagian utama program Python untuk proses scraping data dan penyimpanannya ke file CSV

```
if __name__ == "__main__":
    excel_file = "urls_bupati_riau.xlsx"
    csv_file = "data_bupati_riau2.csv"

    try:
        df = pd.read_excel(excel_file)
        if 'Link' in df.columns:
            urls = df['Link'].tolist()

            try:
                os.remove(csv_file)
            except OSError:
                pass

            for url in urls:
                print(f"Scraping from URL: {url}")
                table_data = scrape_table_data(url, df)
                save_table_data_to_csv(table_data, csv_file)

        else:
            print("Kolom 'URL' tidak ditemukan dalam dataframe.")
    except Exception as e:
        print(f"Error reading Excel file: {e}")
```

Gambar 3.6 Kode Program Logika Utama Program Scraping Web Pilkada 2020

a. Pada baris `if __name__ == "__main__":` adalah cara standar dalam bahasa pemrograman Python untuk memastikan bahwa kode di dalam blok tersebut hanya akan dieksekusi jika file Python ini dijalankan sebagai skrip utama. Baris ini merupakan bagian penting karena memungkinkan file Python ini untuk digunakan sebagai modul oleh file Python lain, tanpa langsung menjalankan blok kode tersebut. Dengan kata lain, fungsi ini memberikan fleksibilitas sehingga file Python dapat

digunakan baik sebagai program utama maupun sebagai pustaka yang diimpor oleh skrip lain.

- b. `excel_file = "urls_bupati_riau.xlsx"` dan `csv_file = "data_bupati_riau2.csv"` Merupakan bentuk variable dimana Variabel `excel_file` menyimpan nama file Excel yang diharapkan berisi daftar URL yang akan di-scrape. Pada kasus ini, file tersebut bernama "urls_bupati_riau.xlsx". File ini diasumsikan berisi satu kolom dengan nama 'Link' dimana setiap barisnya adalah URL yang akan digunakan dalam proses scraping. Lalu Variabel `csv_file` menyimpan nama file CSV tempat hasil scraping akan disimpan, yaitu "data_bupati_riau2.csv". Format CSV (Comma Separated Values) dipilih karena umumnya digunakan untuk menyimpan data dalam format tabel yang dapat dengan mudah dibuka dan diproses dengan berbagai alat, seperti Excel atau program pemrosesan data lainnya.
- c. Blok `try-except` ini digunakan untuk mengeksekusi sekumpulan kode yang mungkin menyebabkan kesalahan selama eksekusi program. Dalam hal ini, kode di dalam blok `try` melibatkan pembacaan file Excel, penghapusan file CSV, dan proses scraping dari URL. Jika ada kesalahan yang terjadi di salah satu langkah tersebut, blok `except` yang mengikuti akan menangkap kesalahan tersebut.
- d. Pada baris `df = pd.read_excel(excel_file)` menggunakan pustaka `pandas` untuk membaca file Excel yang disimpan dalam variabel `excel_file`. Fungsi `pd.read_excel()` digunakan untuk membaca file Excel dan mengkonversinya menjadi dataframe, yaitu struktur data seperti tabel yang digunakan di `pandas`. Nama file yang dibaca ditentukan oleh `excel_file`, dalam hal ini adalah "urls_bupati_riau.xlsx". Jika file tidak ditemukan

atau file rusak, ini akan memunculkan pengecualian (error) yang akan ditangkap oleh blok `except`.

- e. Pada syntax `if 'Link' in df.columns:` digunakan untuk memeriksa apakah dataframe yang dihasilkan dari file Excel memiliki kolom bernama `'Link'`. Kolom ini diharapkan berisi daftar URL yang akan digunakan untuk scraping. Jika kolom tersebut ada dalam dataframe (df), program akan melanjutkan eksekusi. Namun, jika kolom `'Link'` tidak ditemukan, eksekusi akan melewati bagian ini dan menuju ke blok `else`, dimana pesan kesalahan akan ditampilkan.
- f. `urls = df['Link'].tolist()` Cara kerja baris ini adalah jika kolom `'Link'` ditemukan, baris ini akan mengkonversi kolom tersebut ke dalam bentuk daftar Python menggunakan metode `.tolist()`. Ini karena kolom di pandas pada dasarnya adalah sebuah objek Series, dan `.tolist()` mengubahnya menjadi daftar biasa yang dapat diiterasi. Daftar `urls` kemudian akan berisi URL-URL yang nantinya digunakan dalam proses scraping.
- g. Blok `try-except` kedua ini mencoba menghapus file CSV yang sudah ada sebelumnya menggunakan fungsi `os.remove()`. Nama file CSV disimpan dalam variabel `csv_file`. Penghapusan file dilakukan untuk memastikan bahwa data hasil scraping baru tidak dicampur dengan data lama yang mungkin sudah ada di file CSV sebelumnya.
- h. Baris `for url in urls` memulai *loop* untuk mengiterasi setiap `url` di dalam daftar `urls`. Daftar `urls` dihasilkan dari kolom `'Link'` dalam dataframe. Pada setiap iterasi, sebuah URL akan diproses untuk scraping.
- i. Pada setiap iterasi baris `print(f"Scraping from URL: {url}")`, URL yang sedang diproses akan ditampilkan ke terminal menggunakan fungsi `print()`. `f"Scraping from`

`URL: {}`" digunakan untuk menampilkan nilai dari variabel `url` di dalam string. Ini memberikan informasi kepada pengguna tentang URL mana yang sedang di-scrape, sehingga proses scraping dapat dimonitor secara langsung.

- j. Pada baris `table_data = scrape_table_data(url, df)`, diasumsikan ada sebuah fungsi bernama `scrape_table_data()` yang bertugas untuk mengekstraksi data dari halaman web di URL yang diberikan. Fungsi ini menerima dua argument yaitu `url` sebagai URL dari halaman web yang akan di-scrape. Dan `df` sebagai Dataframe yang dihasilkan dari file Excel, yang mungkin diperlukan untuk memproses atau membandingkan data selama scraping. Hasil scraping (data dari tabel di halaman web) kemudian disimpan dalam variabel `table_data`. Fungsi `scrape_table_data()` ini sendiri tidak didefinisikan di dalam kode yang diberikan, tetapi bisa jadi menggunakan pustaka Selenium untuk mengambil data dari halaman HTML.
- k. Setelah data diambil dari halaman web, baris `save_table_data_to_csv(table_data, csv_file)` memanggil fungsi `save_table_data_to_csv()` untuk menyimpan hasil scraping ke dalam file CSV. Fungsi ini menerima dua argument yaitu `table_data` (data yang dihasilkan dari proses scraping) dan `csv_file` (nama file CSV tempat data akan disimpan). Fungsi `save_table_data_to_csv()` bertugas menulis data yang sudah di-scrape ke dalam file CSV yang disebutkan (`csv_file`). Kemungkinan fungsi ini melakukan penambahan data (append) ke file CSV, sehingga data dari setiap URL disimpan ke dalam file yang sama tanpa menimpa data sebelumnya.

I. Blok `else` ini dijalankan jika kondisi `if 'Link' in df.columns:` gagal, artinya kolom 'Link' tidak ditemukan dalam dataframe yang dihasilkan dari file Excel. Jika kondisi ini terjadi, program menampilkan pesan "Kolom 'URL' tidak ditemukan dalam dataframe." ke konsol untuk memberitahu pengguna bahwa file Excel yang dibaca tidak memiliki kolom yang sesuai. Hal ini membantu pengguna untuk memahami bahwa ada masalah dengan struktur file Excel yang digunakan.

3.2.2 Pengolahan Data

Pengolahan data ini merupakan langkah lanjutan dari proses scraping yang dilakukan untuk memperoleh data Pilkada 2020 dari situs pilkada2020.kpu.go.id. Data hasil scraping ini masih terpisah berdasarkan provinsi, sehingga perlu dilakukan pengolahan agar data dapat disajikan dalam bentuk yang lebih terstruktur dan siap untuk dianalisis. Proses ini sangat penting mengingat data yang dihasilkan dari scraping biasanya mentah dan memerlukan penyempurnaan sebelum dapat digunakan untuk tahap berikutnya, yaitu visualisasi di Looker Studio. Oleh karena itu, pengolahan data mencakup berbagai tahapan penting yang dirancang untuk memastikan integritas dan keakuratan data. Berikut merupakan hasil dari Scraping data ditunjukkan pada **gambar 3.7**.

Row	Cell A
1	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 2"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,6,172,98
2	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 3"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,11,136,115
3	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 4"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,5,131,148
4	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 5"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,4,217,140
5	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 6"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,8,157,70
6	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 7"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,6,150,121
7	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 8"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,4,132,80
8	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 9"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,15,107,148
9	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 10"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,4,38,51
10	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112006 , ""TPS 11"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BACEM,1,120,41
11	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112014 , ""TPS 1"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGREJO,3,99,124
12	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112014 , ""TPS 2"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGREJO,1,113,172
13	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112014 , ""TPS 3"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGREJO,1,134,74
14	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112014 , ""TPS 4"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGREJO,2,140,77
15	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112005 , ""TPS 2"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGSARI,6,102,148
16	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112005 , ""TPS 3"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGSARI,3,161,138
17	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112005 , ""TPS 4"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGSARI,3,106,90
18	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112005 , ""TPS 5"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGSARI,9,154,47
19	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112005 , ""TPS 6"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGSARI,3,120,127
20	https://pilkada2020.kpu.go.id/#/pkwkk/tungsuru/3316112005 , ""TPS 7"" ,100.00 %,"JAWA TENGAH,BLORA,BANIAREJO,BALONGSARI,8,98,57

Gambar 3.7 Data Hasil Scraping Sebelum Diolah

Proses pengolahan data dimulai dari merapikan data. Seperti pada **Gambar 3.7** dimana pada gambar tersebut merupakan hasil dari *scraping* dan hasil data dari *scraping* masih belum rapih dan masih belum memiliki *head table*, sehingga praktikkan bertugas untuk membuat *head table* dengan melihat pada website asli dengan mengambil data secara manual. Data yang diambil merupakan nama calon gubernur yang ada.

Setelah data hasil *scraping* dari situs Pilkada 2020 diimpor dan setiap kolom serta baris dipastikan terisi, langkah selanjutnya adalah melakukan transformasi data. Transformasi ini mencakup perubahan struktur data, dimana sebelumnya satu baris data mewakili satu Tempat Pemungutan Suara (TPS), dipecah menjadi beberapa baris, dengan setiap baris merepresentasikan satu pasangan calon gubernur. Hal ini membuat struktur data menjadi lebih kompleks karena satu TPS akan berisi beberapa baris yang berbeda, tergantung pada jumlah pasangan calon yang berkompetisi di wilayah tersebut.

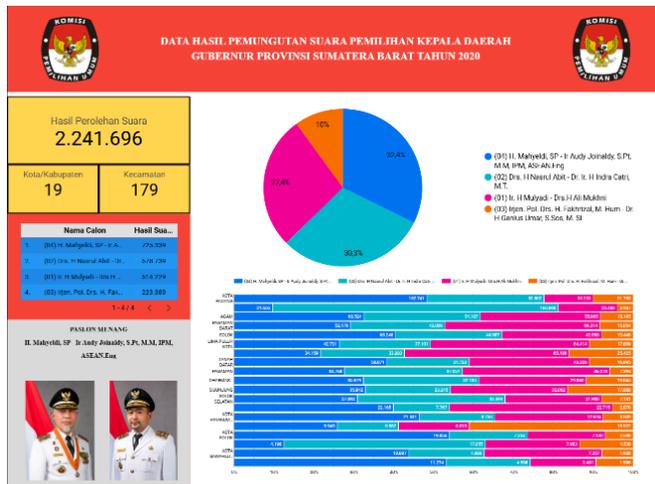
Proses perubahan ini bertujuan agar data dapat dianalisis dengan lebih mendetail dan diolah untuk visualisasi yang lebih fleksibel, terutama dalam platform Looker Studio. Dengan merinci setiap TPS berdasarkan pasangan calon, data yang sebelumnya hanya menampilkan agregat total suara di satu baris, kini dapat menunjukkan suara yang diperoleh oleh setiap pasangan calon di setiap TPS. Hal ini memungkinkan analisis yang lebih granular, misalnya, untuk melihat distribusi suara berdasarkan TPS atau tren suara untuk pasangan calon tertentu di suatu wilayah. Perubahan struktur ini juga membantu dalam menyajikan data dalam berbagai format visual seperti diagram batang dan grafik persentase. Berikut merupakan salah satu hasil dari perapihan data sesuai dengan format untuk dijadikan file input Looker Studio ditunjukkan pada **gambar 3.8**.

persentase kemenangan masing-masing calon, serta diagram batang yang memperlihatkan hasil perolehan suara setiap calon di tiap wilayah. Selain itu, terdapat foto pasangan pemenang Pemilihan Kepala Daerah Provinsi tahun 2020.

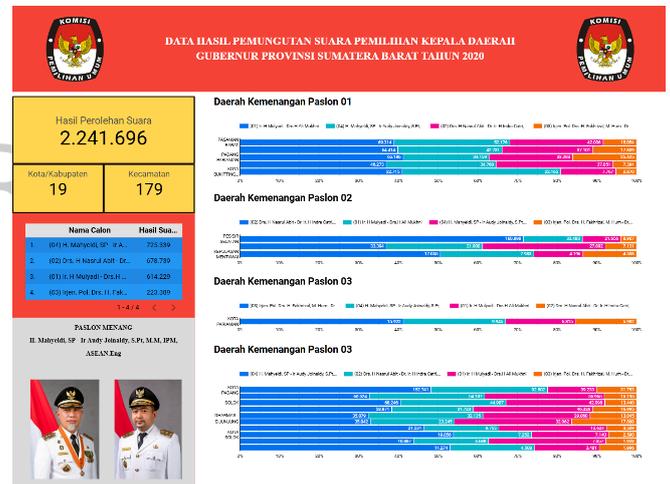
Proses visualisasi data ini dibuat menggunakan platform *Looker Studio*. *Looker Studio* atau dulu akrab dengan *Google Data Studio* merupakan salah satu *tools* untuk membuat *digital dashboard* bersifat *open access* dan *open source* yang disediakan oleh Google (Jariyah et al., 2022). Melalui *platform* ini, praktikkan dapat membuat visualisasi data yang interaktif dan dinamis. Keunggulan dari *Looker Studio* memiliki kemampuan untuk mengintegrasikan berbagai jenis data khususnya data excel dalam jumlah besar. Dengan visualisasi ini, hasil pemilihan gubernur 2020 dapat disajikan dengan cara yang lebih menarik, informatif, dan mudah dipahami oleh audiens, baik dari kalangan pengambil kebijakan, maupun masyarakat umum. Berikut penjelasan hasil visualisasi data hasil pemungutan suara pemilihan kepala daerah di Sembilan provinsi di Indonesia tahun 2020.

3.2.3.1 Sumatera Barat

Berikut merupakan hasil visualisasi hasil pemungutan suara gubernur tahun 2020 di provinsi Sumatera Barat.



Gambar 3.9 Dashboard Hasil Tungsura Provinsi Sumatera Barat



Gambar 3.10 Dashboard Hasil Tungsura Provinsi Sumatera Barat Slide 2

Visualisasi data hasil pemilihan kepala daerah (Pilkada) untuk Gubernur Sumatera Barat ini menampilkan beberapa elemen untuk memberikan gambaran menyeluruh mengenai distribusi suara. Dalam hasil visualisasi di atas dapat dijelaskan bahwa:

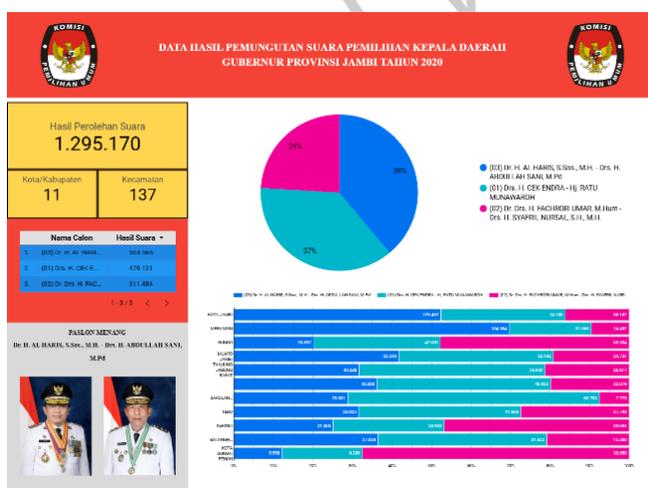
1. Total suara pada bagian kiri atas kotak berwarna kuning, terdapat informasi mengenai total perolehan suara, yaitu **2.241.696** suara, serta jumlah wilayah yang terlibat dalam pemilihan, yaitu **19** kota/kabupaten dengan total **179** kecamatan.
2. Di bawah total suara terdapat rincian masing-masing pasangan calon yang menunjukkan hasil perolehan suara yang didapat dari tiap pasangan calon yang ada diurutkan dari pasangan calon dengan perolehan suara terbesar. Pada hasil visualisasi Sumatera Barat menunjukkan pasangan calon nomor (04) memperoleh total suara sebanyak 725.339, pasangan calon nomor urut (02) memperoleh total suara sebanyak 678.739, pasangan calon nomor urut (01) memperoleh total suara sebanyak 614.229, dan pasangan calon nomor urut (03) memperoleh total suara sebanyak 223.389.

3. Diagram lingkaran (*pie chart*) menunjukkan persentase suara yang diperoleh oleh masing-masing pasangan calon. Dalam hasil visualisasi Sumatera Barat ini, terlihat bahwa pasangan calon nomor (04) mendapatkan persentase tertinggi sebesar 32,4%, dilanjut dengan pasangan calon nomor (02) dengan perolehan terbesar kedua sebesar 30,3%, lalu di peringkat ketiga terlihat pasangan calon nomor (01) dengan perolehan suara sebesar 27,4% dan di posisi terakhir terlihat pasangan calon (03) dengan hasil perolehan suara sebesar 10%.
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) Ir. H. Mulyadi – Drs. H. Ali Mukhni. Pasangan calon nomor urut (02) Drs. H. Nasrul Abit – Dr. Ir. H. Indra Catri, M. T. Pasangan calon nomor urut (03) Irjen. Pol. Drs. H. Fakhrizal, M.Hum – Dr. H. Genius Umar, S.Sos., M.Si. Pasangan calon nomor urut (04) H. Mahyeldi, SP – Ir. Audy Joinaldy, S.Pt., M.M., IPM, ASEAN.Eng
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu H. Mahyeldi, SP bersama wakilnya Ir. Audy Joinaldy, yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) memenangkan wilayah Pasaman Barat, Lima Puluh Kota, Padang Pariaman, Pasaman, dan Kota Bukittinggi. Pasangan calon nomor (02) memenangkan daerah Pesisir Selatan, Solok Selatan, dan Kepulauan Mentawai. Pasangan calon nomor (03) mendominasi Kota Pariaman sebagai satu-satunya daerah

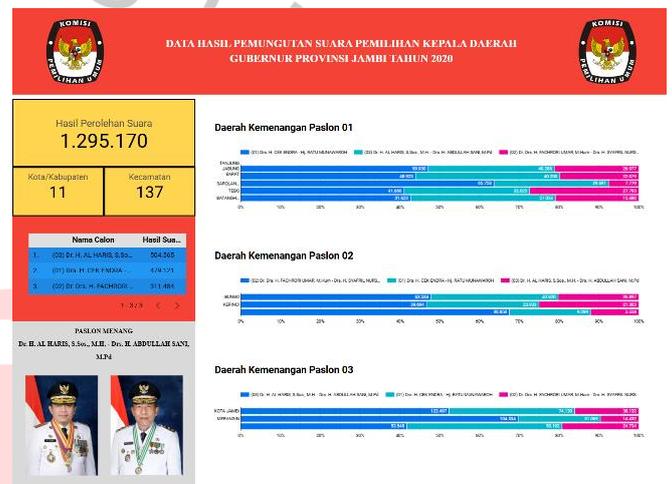
kemenangan. Pasangan calon nomor (04) berhasil menang di Kota Padang, Agam, Solok, Tanah Datar, Dharmasraya, Sijunjung, Kota Payakumbuh, Kota Solok, Kota Sawahlunto, dan Kota Padang Panjang.

3.2.3.2 Jambi

Berikut merupakan hasil visualisasi hasil pemungutan suara gubernur tahun 2020 di provinsi Jambi.



Gambar 3.11 Dashboard Hasil Tungsura Provinsi Jambi



Gambar 3.12 Dashboard Hasil Tungsura Provinsi Jambi Slide 2

Visualisasi data hasil pemilihan kepala daerah (Pilkada) Gubernur Jambi ini menampilkan beberapa elemen untuk memberikan gambaran menyeluruh mengenai distribusi suara. Dalam hasil visualisasi di atas dapat dijelaskan bahwa:

1. Total suara pada bagian kiri atas Kotak berwarna kuning, terdapat informasi mengenai total perolehan suara, yaitu **1.295.170** suara, serta jumlah wilayah yang terlibat dalam pemilihan, yaitu **11** kota/kabupaten dengan total **137** kecamatan.
2. Dibawah total suara terdapat rincian masing-masing pasangan calon yang menunjukkan hasil perolehan suara yang didapat dari tiap pasangan calon yang ada diurutkan dari pasangan calon dengan perolehan suara terbesar. Pada hasil visualisasi Jambi menunjukkan pasangan calon nomor (03) memperoleh

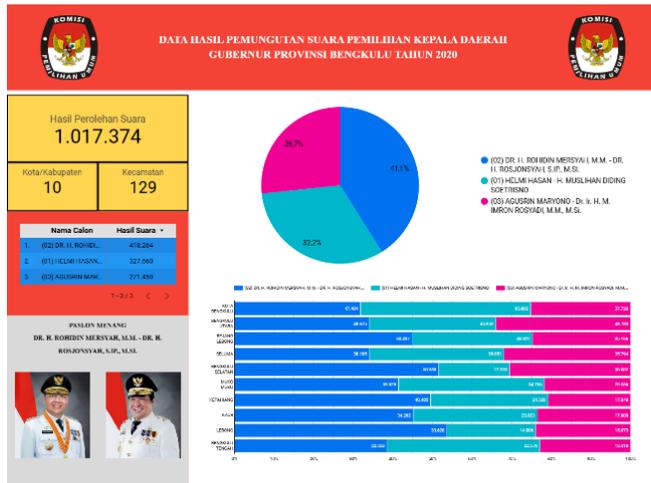
total suara sebanyak 504.565, pasangan calon nomor urut (01) memperoleh total suara sebanyak 479.121, dan pasangan calon nomor urut (02) memperoleh total suara sebanyak 311.484.

3. Diagram lingkaran (pie chart) menunjukkan persentase suara yang diperoleh oleh masing-masing pasangan calon. Dalam hasil visualisasi Jambi ini, terlihat bahwa pasangan calon nomor (03) mendapatkan persentase tertinggi sebesar 39%, dilanjut dengan pasangan calon nomor (01) dengan perolehan terbesar kedua sebesar 37%, dan di posisi terakhir terlihat pasangan calon (02) dengan hasil perolehan suara sebesar 24%.
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) Drs. H. Cek Endra – Hj. Ratu Munawaroh. Pasangan calon nomor urut (02) Dr. Drs. H. Fachrori Umar, M.Hum – Drs. H. Syafril Nursal, S.H., M.H. Pasangan calon nomor urut (03) Dr. H. Al Haris, S.Sos., M.H. – Drs. H. Abdullah Sani, M.Pd.
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu Dr. H. Al Haris, S.Sos., M.H. beserta wakilnya Drs. H. Abdullah Sani, M.Pd., yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) memenangkan wilayah Tanjung Jabung Barat, Tanjung Jabung Timur, Sarolangun, Tebo, dan Batanghari. Pasangan calon nomor (02) memenangkan daerah Bungo, Kerinci, dan Kota Sungai

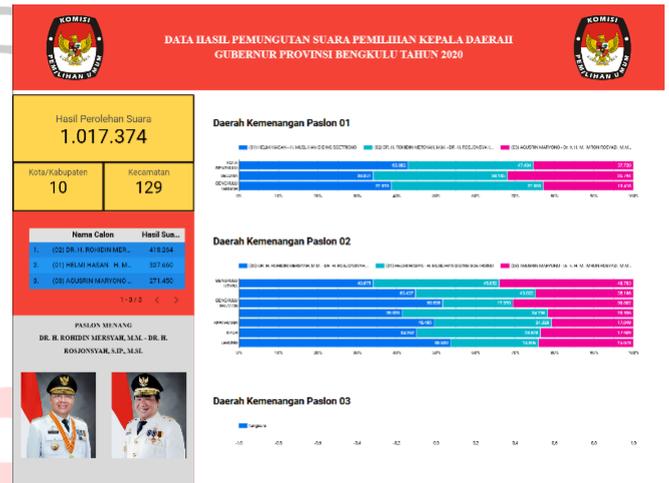
Penuh. Pasangan calon nomor (03) berhasil menang di Kota Jambi, Merangin, dan Muaro Jambi.

3.2.3.3 Bengkulu

Berikut merupakan hasil visualisasi hasil pemungutan suara gubernur tahun 2020 di provinsi Bengkulu.



Gambar 3.13 Dashboard Hasil Tungsura Provinsi Bengkulu



Gambar 3.14 Dashboard Hasil Tungsura Provinsi Bengkulu Slide 2

Visualisasi data hasil pemilihan kepala daerah (Pilkada) Gubernur Bengkulu ini menampilkan beberapa elemen untuk memberikan gambaran menyeluruh mengenai distribusi suara. Dalam hasil visualisasi di atas dapat dijelaskan bahwa:

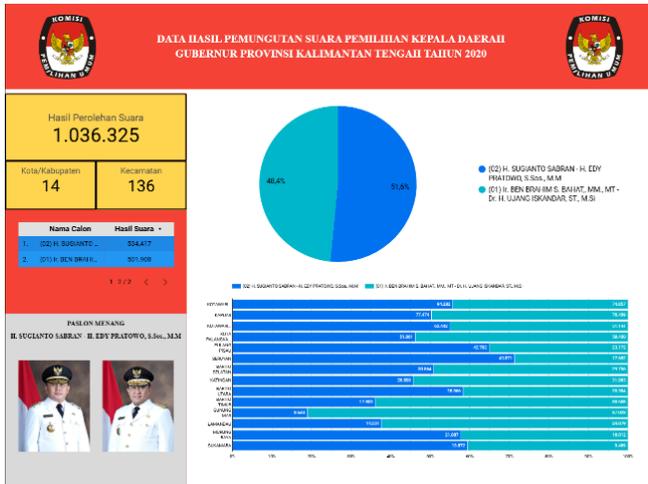
1. Total suara pada bagian kiri atas Kotak berwarna kuning, terdapat informasi mengenai total perolehan suara, yaitu **1.017.374** suara, serta jumlah wilayah yang terlibat dalam pemilihan, yaitu **10** kota/kabupaten dengan total **129** kecamatan.
2. Dibawah total suara terdapat rincian masing-masing pasangan calon yang menunjukkan hasil perolehan suara yang didapat dari tiap pasangan calon yang ada diurutkan dari pasangan calon dengan perolehan suara terbesar. Pada hasil visualisasi Bengkulu menunjukkan pasangan calon nomor (02) memperoleh total suara sebanyak 418.264, pasangan calon nomor urut (01) memperoleh total suara sebanyak 327.660,

dan pasangan calon nomor urut (03) memperoleh total suara sebanyak 271.450.

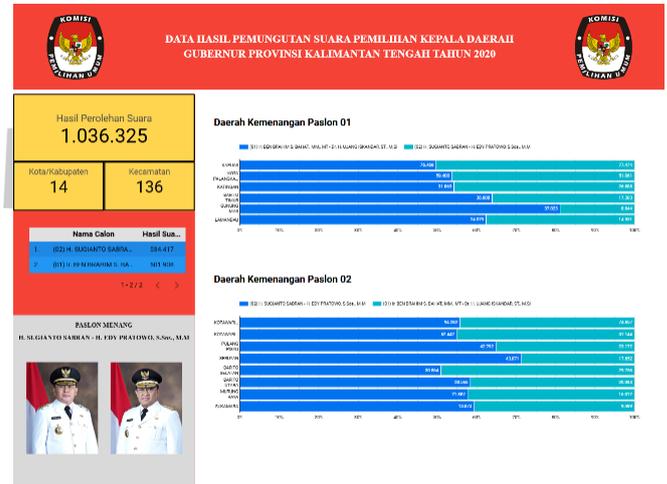
3. Diagram lingkaran (pie chart) menunjukkan persentase suara yang diperoleh oleh masing-masing pasangan calon. Dalam hasil visualisasi Bengkulu ini, terlihat bahwa pasangan calon nomor (02) mendapatkan persentase tertinggi sebesar 41,1%, dilanjut dengan pasangan calon nomor (01) dengan perolehan terbesar kedua sebesar 32,2%, dan di posisi terakhir terlihat pasangan calon (03) dengan hasil perolehan suara sebesar 26,7%.
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) Helmi Hasan – H. Muslihan Diding Soetrisno. Pasangan calon nomor urut (02) Dr. H. Rohidin Mersyah, M.M. – Dr. H. Rosjonsyah, S.IP., M.SI. Pasangan calon nomor urut (03) Agusrin Maryono - Dr. Ir. H. M. Imron Rosyadi, M.M., M.Si.
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu Dr. H. Rohidin Mersyah, M.M. beserta wakilnya Dr. H. Rosjonsyah, S.IP., M.SI. yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) memenangkan wilayah Bengkulu, Seluma, dan Bengkulu Tengah. Pasangan calon nomor (02) memenangkan daerah Bengkulu Utara, Rejang Lebong, Bengkulu Selatan, Muko Muko, Kepahiang, Kaur, dan Lebong. Sementara itu pasangan calon nomor (03) tidak memiliki daerah kemenangannya.

3.2.3.4 Kalimantan Tengah

Berikut merupakan hasil visualisasi hasil pemungutan suara gubernur tahun 2020 di provinsi Kalimantan Tengah.



Gambar 3.15 Dashboard Hasil Tungsura Provinsi Kalimantan Tengah



Gambar 3.16 Dashboard Hasil Tungsura Provinsi Kalimantan Tengah Slide 2

Visualisasi data hasil pemilihan kepala daerah (Pilkada) Gubernur Kalimantan Tengah ini menampilkan beberapa elemen untuk memberikan gambaran menyeluruh mengenai distribusi suara. Dalam hasil visualisasi di atas dapat dijelaskan bahwa:

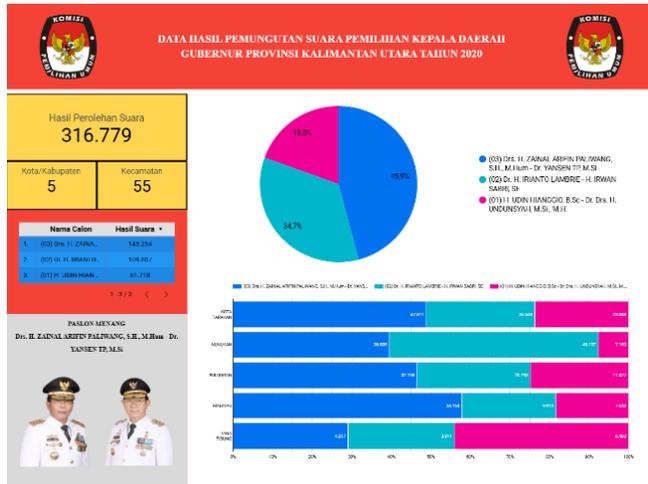
1. Total suara pada bagian kiri atas Kotak berwarna kuning, terdapat informasi mengenai total perolehan suara, yaitu **1.036.325** suara, serta jumlah wilayah yang terlibat dalam pemilihan, yaitu **14** kota/kabupaten dengan total **136** kecamatan.
2. Dibawah total suara terdapat rincian masing-masing pasangan calon yang menunjukkan hasil perolehan suara yang didapat dari tiap pasangan calon yang ada diurutkan dari pasangan calon dengan perolehan suara terbesar. Pada hasil visualisasi Kalimantan Tengah menunjukkan pasangan calon nomor (02) memperoleh total suara sebanyak 534.417 dan pasangan calon nomor urut (01) memperoleh total suara sebanyak 501.908
3. Diagram lingkaran (pie chart) menunjukkan persentase suara yang diperoleh oleh masing-masing pasangan calon. Dalam

hasil visualisasi Kalimantan Tengah ini, terlihat bahwa pasangan calon nomor (02) mendapatkan persentase tertinggi sebesar 51,6%, dilanjut dengan pasangan calon nomor (01) dengan perolehan terbesar kedua sebesar 48,4%

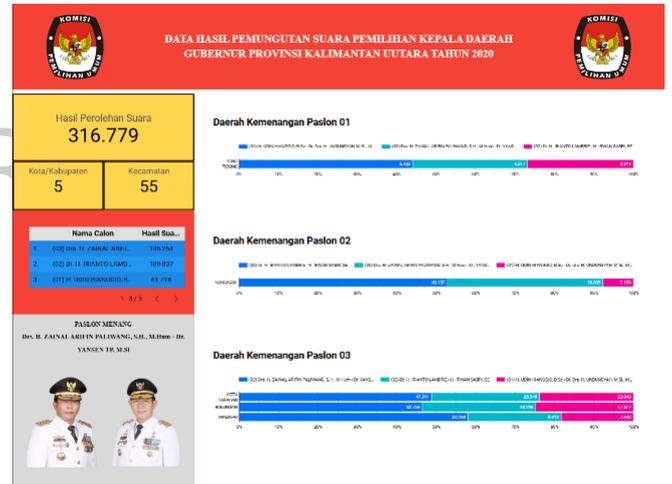
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) Ir. Ben Brahim S. Bahat, M.M., M.T - Dr. H. Ujang Iskandar, S.T., M.Si. Pasangan calon nomor urut (02) H. Sugianto Sabran - H. Edy Pratowo, S.Sos., M.M.
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu H. Sugianto Sabran beserta wakilnya H. Edy Pratowo, S.Sos., M.M. yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) memenangkan wilayah Kapuas, Kota Palangkaraya, Katingan, Barito Timur, Gunung Mas, dan Lamandau. Pasangan calon nomor (02) memenangkan wilayah Kota Waringin, Kota Waringin Barat, Pulang Pisau, Seruyan, Barito Selatan, Barito Utara, Murung Raya, dan Sukamara.

3.2.3.5 Kalimantan Utara

Berikut merupakan hasil visualisasi hasil pemungutan suara gubernur tahun 2020 di provinsi Kalimantan Utara.



Gambar 3.17 Dashboard Hasil Tungsura Provinsi Kalimantan Utara



Gambar 3.18 Dashboard Hasil Tungsura Provinsi Kalimantan Utara Slide 2

Visualisasi data hasil pemilihan kepala daerah (Pilkada) Gubernur Kalimantan Utara ini menampilkan beberapa elemen untuk memberikan gambaran menyeluruh mengenai distribusi suara. Dalam hasil visualisasi di atas dapat dijelaskan bahwa:

1. Total suara pada bagian kiri atas Kotak berwarna kuning, terdapat informasi mengenai total perolehan suara, yaitu **316.779** suara, serta jumlah wilayah yang terlibat dalam pemilihan, yaitu **5** kota/kabupaten dengan total **55** kecamatan.
2. Dibawah total suara terdapat rincian masing-masing pasangan calon yang menunjukkan hasil perolehan suara yang didapat dari tiap pasangan calon yang ada diurutkan dari pasangan calon dengan perolehan suara terbesar. Pada hasil visualisasi Kalimantan Utara menunjukkan pasangan calon nomor (03) memperoleh total suara sebanyak 145.254, pasangan calon nomor urut (02) memperoleh total suara sebanyak 109.807, dan pasangan calon nomor urut (01) memperoleh total suara sebanyak 61.718.
3. Diagram lingkaran (pie chart) menunjukkan persentase suara yang diperoleh oleh masing-masing pasangan calon. Dalam

hasil visualisasi Kalimantan Utara ini, terlihat bahwa pasangan calon nomor (03) mendapatkan persentase tertinggi sebesar 45,9%, dilanjut dengan pasangan calon nomor (02) dengan perolehan terbesar kedua sebesar 34,7%, dan di posisi terakhir terlihat pasangan calon (01) dengan hasil perolehan suara sebesar 19,5%.

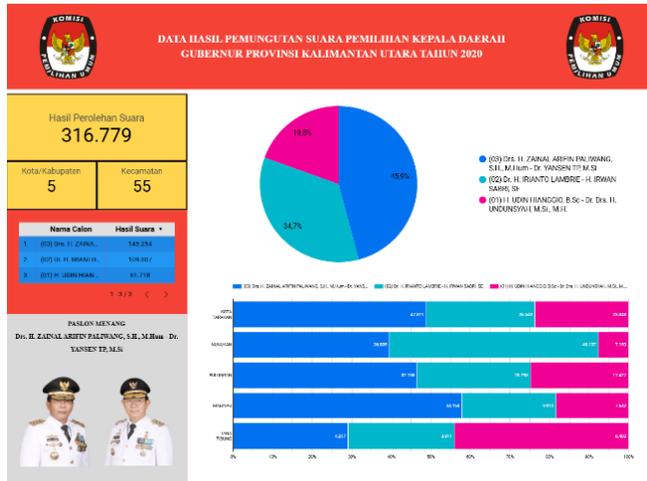
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) H. Udin Hianggio, B.Sc. – Dr. Drs. H. Undunyah, M.Si., M.H. Pasangan calon nomor urut (02) Dr. H. Irianto Lambrie – H. Irwan Sabri, S.E. Pasangan calon nomor urut (03) Drs. H. Zainal Arifin Paliwang, S.H., M.Hum. – Dr. Yansen TP, M.Si.
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu Drs. H. Zainal Arifin Paliwang, S.H., M.Hum. beserta wakilnya Dr. Yansen TP, M.Si. yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) memenangkan wilayah Tana Tidung. Pasangan calon nomor (02) memenangkan wilayah Nunukan. Pasangan calon nomor (03) berhasil menang di Kota Tarakan, Bulungan, dan Malinau.

hasil visualisasi Kalimantan Selatan ini, terlihat bahwa pasangan calon nomor (02) mendapatkan persentase tertinggi sebesar 50,6%, dilanjut dengan pasangan calon nomor (01) dengan perolehan terbesar kedua sebesar 49,4%

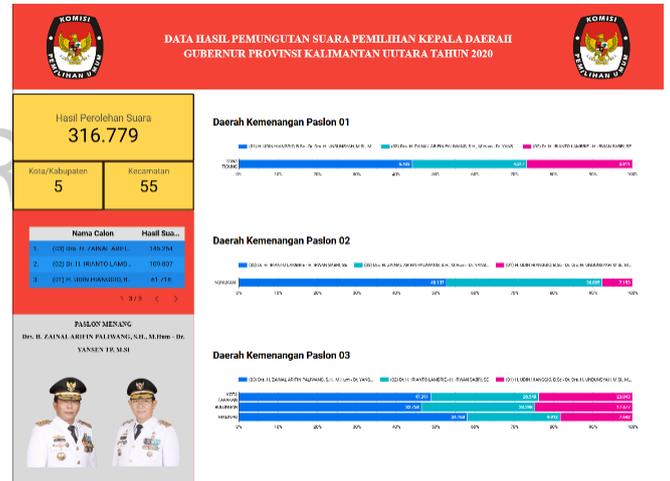
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) H. Sahbirin noor, S.Sos., MH – H. Muhidin. Pasangan calon nomor urut (02) Prof H Denny Indrayana, S.H., LL.M., Ph.D. - Drs H Difriadi.
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu Prof H Denny Indrayana, S.H., LL.M., Ph.D. beserta wakilnya Drs H Difriadi yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) memenangkan wilayah Banjar, Tanah Bumbu, Barito Kuala, Tapin, dan Balangan. Pasangan calon nomor (02) memenangkan wilayah Kota Banjarmasin, Kota Baru, Hulu Sungai Tengah, Kota Banjar Baru, Tanah Laut, Tabalong, Hulu Sungai Utara, dan Hulu Sungai Selatan.

3.2.3.7 Kepulauan Riau

Berikut merupakan hasil visualisasi hasil pemungutan suara gubernur tahun 2020 di provinsi Kepulauan Riau.



Gambar 3.21 Dashboard Hasil Tungsura Provinsi Kepulauan Riau



Gambar 3.22 Dashboard Hasil Tungsura Provinsi Kepulauan Riau Slide 2

Visualisasi data hasil pemilihan kepala daerah (Pilkada) Gubernur Kepulauan Riau ini menampilkan beberapa elemen untuk memberikan gambaran menyeluruh mengenai distribusi suara. Dalam hasil visualisasi di atas dapat dijelaskan bahwa:

1. Total suara pada bagian kiri atas Kotak berwarna kuning, terdapat informasi mengenai total perolehan suara, yaitu **761.083** suara, serta jumlah wilayah yang terlibat dalam pemilihan, yaitu **7** kota/kabupaten dengan total **76** kecamatan.
2. Dibawah total suara terdapat rincian masing-masing pasangan calon yang menunjukkan hasil perolehan suara yang didapat dari tiap pasangan calon yang ada diurutkan dari pasangan calon dengan perolehan suara terbesar. Pada hasil visualisasi Kepulauan Riau menunjukkan pasangan calon nomor (03) memperoleh total suara sebanyak 304.597, pasangan calon nomor urut (02) memperoleh total suara sebanyak 274.393, dan pasangan calon nomor urut (01) memperoleh total suara sebanyak 182.093.
3. Diagram lingkaran (pie chart) menunjukkan persentase suara yang diperoleh oleh masing-masing pasangan calon. Dalam

hasil visualisasi Kepulauan Riau ini, terlihat bahwa pasangan calon nomor (03) mendapatkan persentase tertinggi sebesar 40%, dilanjut dengan pasangan calon nomor (02) dengan perolehan terbesar kedua sebesar 36,1%, dan di posisi terakhir terlihat pasangan calon (01) dengan hasil perolehan suara sebesar 23,9%.

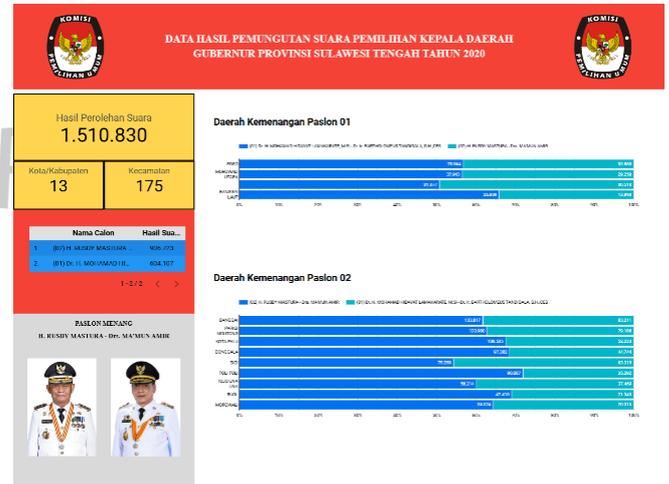
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) Dr. H. M. Soerya Respationo, S.H., M.H - Iman Sutiawan, S.E. Pasangan calon nomor urut (02) H. Isdianto, S.Sos., M.M - Suryani, S.E. Pasangan calon nomor urut (03) Ansar Ahmad, S.E., M.M – Marlin Agustina.
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu Ansar Ahmad, S.E., M.M beserta wakilnya Marlin Agustina yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) tidak memiliki daerah kemenangan. Pasangan calon nomor (02) memenangkan wilayah Kota Batam dan Karimun. Pasangan calon nomor (03) berhasil menang di Kota Tanjung Pinang, Bintan, Lingga, Natuna, dan Kepulauan Anambas.

3.2.3.8 Sulawesi Tengah

Berikut merupakan hasil visualisasi hasil pemungutan suara gubernur tahun 2020 di provinsi Sulawesi Tengah.



Gambar 3.23 Dashboard Hasil Tungsura Provinsi Sulawesi Tengah



Gambar 3.24 Dashboard Hasil Tungsura Provinsi Sulawesi Tengah Slide 2

Visualisasi data hasil pemilihan kepala daerah (Pilkada) Gubernur Sulawesi Tengah ini menampilkan beberapa elemen untuk memberikan gambaran menyeluruh mengenai distribusi suara. Dalam hasil visualisasi di atas dapat dijelaskan bahwa:

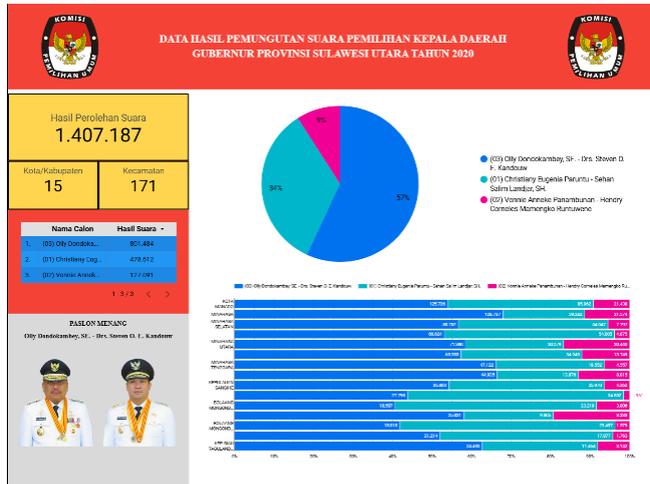
1. Total suara pada bagian kiri atas Kotak berwarna kuning, terdapat informasi mengenai total perolehan suara, yaitu **1.510.830** suara, serta jumlah wilayah yang terlibat dalam pemilihan, yaitu **13** kota/kabupaten dengan total **175** kecamatan.
2. Dibawah total suara terdapat rincian masing-masing pasangan calon yang menunjukkan hasil perolehan suara yang didapat dari tiap pasangan calon yang ada diurutkan dari pasangan calon dengan perolehan suara terbesar. Pada hasil visualisasi Sulawesi Tengah menunjukkan pasangan calon nomor (02) memperoleh total suara sebanyak 906.723 dan pasangan calon nomor urut (01) memperoleh total suara sebanyak 604.107.
3. Diagram lingkaran (pie chart) menunjukkan persentase suara yang diperoleh oleh masing-masing pasangan calon. Dalam

hasil visualisasi Sulawesi Tengah ini, terlihat bahwa pasangan calon nomor (02) mendapatkan persentase tertinggi sebesar 60%, dilanjut dengan pasangan calon nomor (01) dengan perolehan terbesar kedua sebesar 40%

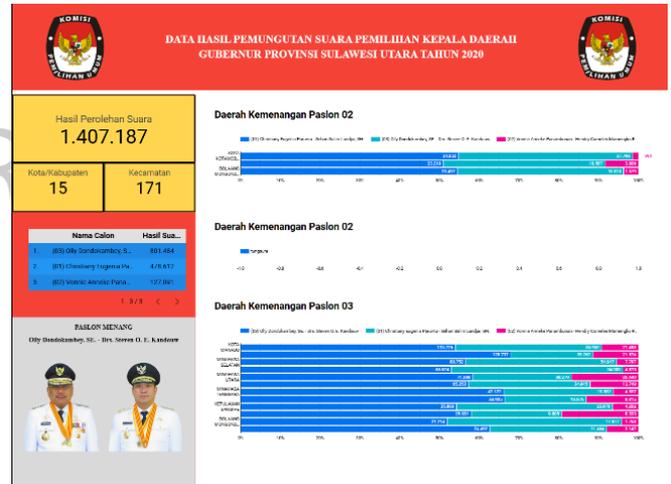
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) Dr. H. Mohamad Hidayat Lamakarate, M.Si - Dr. Ir. Bartholomeus Tandigala, S.H., CES. Pasangan calon nomor urut (02) H. Rusdy Mastura - Drs. Ma'mun Amir.
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu H. Rusdy Mastura beserta wakilnya Drs. Ma'mun Amir yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) memenangkan wilayah Poso, Morowali Utara, Banggai Kepulauan, dan Banggai Laut. Pasangan calon nomor (02) memenangkan wilayah Banggai, Parigi Moutong, Kota Palu, Donggala, Sigi, Toli Toli, Tojo Una Una, Buol, dan Morowali.

3.2.3.9 Sulawesi Utara

Berikut merupakan hasil visualisasi hasil pemungutan suara gubernur tahun 2020 di provinsi Sulawesi Utara.



Gambar 3.25 Dashboard Hasil Tungsura Provinsi Sulawesi Utara



Gambar 3.26 Dashboard Hasil Tungsura Provinsi Sulawesi Utara Slide 2

Visualisasi data hasil pemilihan kepala daerah (Pilkada) Gubernur Sulawesi Utara ini menampilkan beberapa elemen untuk memberikan gambaran menyeluruh mengenai distribusi suara. Dalam hasil visualisasi di atas dapat dijelaskan bahwa:

1. Total suara pada bagian kiri atas Kotak berwarna kuning, terdapat informasi mengenai total perolehan suara, yaitu **1.407.187** suara, serta jumlah wilayah yang terlibat dalam pemilihan, yaitu **15** kota/kabupaten dengan total **171** kecamatan.
2. Dibawah total suara terdapat rincian masing-masing pasangan calon yang menunjukkan hasil perolehan suara yang didapat dari tiap pasangan calon yang ada diurutkan dari pasangan calon dengan perolehan suara terbesar. Pada hasil visualisasi Sulawesi Utara menunjukkan pasangan calon nomor (03) memperoleh total suara sebanyak 801.484, pasangan calon nomor urut (01) memperoleh total suara sebanyak 478.612, dan pasangan calon nomor urut (02) memperoleh total suara sebanyak 127.091.

3. Diagram lingkaran (pie chart) menunjukkan persentase suara yang diperoleh oleh masing-masing pasangan calon. Dalam hasil visualisasi Sulawesi Utara ini, terlihat bahwa pasangan calon nomor (03) mendapatkan persentase tertinggi sebesar 57%, dilanjut dengan pasangan calon nomor (01) dengan perolehan terbesar kedua sebesar 34%, dan di posisi terakhir terlihat pasangan calon (02) dengan hasil perolehan suara sebesar 9%.
4. Daftar kandidat di bawah diagram lingkaran dan disamping diagram lingkaran disebutkan melalui teks. Pasangan calon nomor urut (01) Christiany Eugenia Paruntu – Sehan Salim Landjar, S.H. Pasangan calon nomor urut (02) Vonnie Anneke Panambunan – Hendry Corneles Mamengko Runtuwen. Pasangan calon nomor urut (03) Olly Dondokambey, S.E. - Drs. Steven O. E. Kandouw.
5. Diagram batang horizontal di bagian bawah menunjukkan distribusi suara untuk masing-masing kandidat di setiap kabupaten/kota. Setiap warna dan panjang pada batang menunjukkan satu pasangan calon dan batang merepresentasikan proporsi suara di wilayah tersebut.
6. Foto kandidat terpilih di bagian bawah tabel kandidat yaitu Olly Dondokambey, S.E. beserta wakilnya Drs. Steven O. E. Kandouw yang mempermudah identifikasi terhadap pasangan calon yang mendapatkan suara tertinggi.
7. Pada slide 2 terdapat diagram batang yang menunjukkan daerah kemenangan masing-masing pasangan calon. Terlihat bahwa pasangan calon nomor (01) Memenangkan wilayah Kota Mobagu, Bolaang Mongondow Timur, dan Bolaang Mongondow Utara. Sementara itu, pasangan calon nomor (02) tidak memiliki daerah kemenangan. Pasangan calon nomor (03) berhasil menang di Kota Manado, Minahasa, Minahasa Selatan, Bolaang Mongondow, Minahasa Utara, Kota Bitung, Minahasa Tenggara, Kota Tomohon, Kepulauan Sangihe,

Kepulauan Talaud, Bolaang Mongondow Selatan, dan Kepulauan Siau Tagulandang Biaro.

3.3 Kendala Yang Dihadapi

Selama pelaksanaan kerja profesi dalam proses pengembangan dashboard hasil Pilkada Gubernur tahun 2020 di Sembilan provinsi Indonesia menggunakan teknik perolehan data *web scraping* dan visualisasi data di *Looker Studio* praktikan mengalami beberapa kendala:

1. Keterbatasan akses dan sumber data hasil Pilkada tidak selalu tersedia secara terbuka di situs resmi pilkada.kpu.go.id. Pada situs tersebut terdapat penerapan batasan akses serta protokol keamanan yang menyulitkan proses *scraping*.
2. Format data yang berbeda-beda di setiap provinsi atau daerah juga membuat pengumpulan data menjadi lebih rumit, memerlukan usaha tambahan untuk menyatukan data yang beragam. Selain itu, kualitas dan konsistensi data yang diperoleh sering kali tidak seragam atau mengandung *missing values* (nilai kosong) akibat perubahan halaman atau masalah teknis selama proses *scraping*. sehingga, data perlu melalui proses *data cleaning* yang cukup memakan waktu.
3. Integrasi data ke dalam *Looker Studio* juga menjadi salah satu tantangan teknis tersendiri. Data yang di-*scrape* perlu diproses agar sesuai dengan format yang diterima *Looker Studio*, yang sering kali membutuhkan langkah tambahan seperti pemformatan ulang atau penyesuaian tipe data.
4. Banyaknya data hingga mencapai ratusan ribu baris data membuat proses *scraping* memerlukan banyak waktu dan membutuhkan koneksi internet yang stabil. Setiap kali terjadi gangguan koneksi, proses *scraping* harus dicek ulang untuk memastikan data yang sudah diperoleh tidak terduplikasi dan data yang belum ter-*scrape* dapat dilanjutkan. Hal ini membutuhkan pemantauan secara berkala serta strategi untuk melanjutkan *scraping* dari data terakhir yang berhasil diambil, yang kadang sulit diterapkan tanpa otomatisasi yang canggih.

3.4 Cara Mengatasi Kendala

Untuk mengatasi kendala yang dihadapi dalam proses pengembangan dashboard, langkah yang dapat dilakukan antara lain:

1. Cara yang dilakukan dalam mengatasi keterbatasan akses dan sumber data pada situs pilkada.kpu.go.id, praktikan menambahkan interval waktu menggunakan library `time.sleep(5)`. Penambahan interval ini bertujuan agar scraping berjalan secara bertahap tanpa melampaui batas akses yang ditetapkan oleh situs dan tanpa mengganggu kinerja server.
2. Dalam mengatasi perbedaan format dan kualitas data yang tidak konsisten, dilakukan validasi data secara berkala selama proses scraping untuk memastikan konsistensi data yang diperoleh dari berbagai sumber. Validasi dilakukan dengan memisahkan data per provinsi agar proses pengecekan lebih mudah dan data yang diperiksa tidak terlalu menumpuk.
3. Dalam menyesuaikan format data agar dapat diintegrasikan dengan Looker Studio, praktikan sering mengalami proses trial and error karena data yang dihasilkan sering kali tidak sesuai dengan format yang dibutuhkan oleh Looker Studio dan gagal terbaca. Untuk mengatasi hal ini, praktikan melakukan pengecekan data secara mendalam, termasuk melakukan perubahan struktur data agar sesuai dengan kebutuhan Looker Studio. Selain itu, data juga dipisahkan per provinsi untuk mempermudah pengelolaan dan pengecekan, serta penginputan ke Looker Studio dilakukan secara bertahap per provinsi.
4. Cara menghadapi kendala dalam permasalahan jumlah data yang besar dan potensi gangguan koneksi selama proses scraping, praktikan menambahkan baris kode `print(f"Scraping from URL: {url}")` ke dalam skrip scraping. Baris ini berfungsi sebagai pemberitahuan bahwa link tersebut sedang di-scrape. Jika terjadi kegagalan dalam proses scraping, akan muncul pesan error `print(f"Error reading Excel file: {e}")`. Dengan cara ini, ketika terjadi error, praktikan dapat langsung menghapus data dari link yang sudah berhasil di-scrape dengan memeriksa output yang dihasilkan, sehingga proses scraping dapat dilanjutkan dengan lebih efisien dan terorganisir.

3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

Selama praktikan mengikuti kerja profesi di divisi *System and Technology* PT. Indekstat Konsultan Indonesia, banyak pembelajaran serta penerapan ilmu yang telah diperoleh selama perkuliahan. Pembelajaran yang berhasil diterapkan dalam konteks ini adalah pengetahuan yang telah dipelajari di perkuliahan. Berikut adalah bentuk penerapan mata kuliah yang telah diampu dalam aktivitas kerja profesi yang relevan dengan bidang Sistem dan Teknologi di PT. Indekstat Konsultan Indonesia:

1. *Business Intelligence*

Sesuai dengan salah satu capaian mata kuliah *Business Intelligence*, yaitu kemampuan untuk menjelaskan *Business Intelligence* dan pemanfaatan informasi sebagai aset dalam mendukung proses pengambilan keputusan, penerapan pembelajaran *Business Intelligence* dalam kerja profesi yang telah dilakukan, Sebagai bentuk penerapannya praktikan menggunakan Web Scraping untuk mengumpulkan data hasil Pilkada dari sumber online. Data yang dikumpulkan kemudian dianalisis dan disajikan dalam bentuk visualisasi yang interaktif menggunakan Looker Studio.

2. *Knowledge Management*

Sesuai dengan salah satu capaian mata kuliah *Knowledge Management*, yaitu kemampuan untuk memproses data menjadi informasi dengan menganalisis manajemen pengetahuan dan sistem pintar buatan, penerapan pembelajaran mata kuliah *Knowledge Management* dalam kerja profesi yang telah dilakukan adalah dengan menerapkan konsep *Knowledge Management* untuk mengubah data hasil Pilkada yang dikumpulkan melalui Web Scraping menjadi informasi yang berguna. Proses ini mencakup analisis terhadap data mentah, dimana praktikan belajar mengidentifikasi informasi penting dan relevan yang dapat mendukung pengambilan keputusan strategis. Selain itu, praktikan juga mampu menerapkan sistem terstruktur untuk mengelola data dan informasi, sehingga dapat memanfaatkan pengetahuan yang diperoleh dari hasil analisis guna meningkatkan efektivitas proses bisnis.

3. Komputer dan Masyarakat

Dalam pengembangan Dashboard Hasil Pilkada Gubernur Tahun 2020 di Indonesia Menggunakan Web Scraping dan Looker Studio, mahasiswa memahami pentingnya teknologi komputer dalam menyediakan akses informasi publik yang transparan dan akurat bagi masyarakat. Dimana hal ini sejalan dengan salah satu capaian pembelajaran mata kuliah komputer dan masyarakat. Proses ini melibatkan pemahaman tentang perkembangan komputer dan metode pengumpulan data otomatis, seperti *web scraping*. Praktikan menyadari tantangan menjaga keakuratan dan etika dalam pengelolaan data demi mencegah potensi kesalahpahaman di masyarakat.

Melalui pengalaman ini, praktikan tidak hanya berhasil menerapkan teori yang telah dipelajari selama perkuliahan, tetapi juga mengembangkan keterampilan praktis yang berguna dalam dunia kerja. Penerapan dari mata kuliah yang ada dalam proyek pembuatan dashboard Pilkada 2020 memberikan kontribusi terhadap analisis data dan pengambilan keputusan selama praktikan melaksanakan kerja profesi di PT. Indekstat Konsultan Indonesia.