

SKRIPSI

IMPLEMENTASI *DASHBOARD* UNTUK VISUALISASI DATA PENERIMAAN MAHASISWA BARU UNIVERSITAS JANABADRA MENGGUNAKAN *METABASE*



Disusun oleh:
Tan Anugrah Ramadhani
18330119

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS JANABADRA
YOGYAKARTA
2022

**LEMBAR PENGESAHAN
SKRIPSI**

**IMPLEMENTASI DASHBOARD UNTUK VISUALISASI DATA PENERIMAAN
MAHASISWA BARU UNIVERSITAS JANABADRA
MENGGUNAKAN METABASE**

Dipersiapkan dan disusun oleh:
Tan Anugrah Ramadhani
18330119

Telah dipertahankan di depan dewan pengaji
Pada tanggal: 08 Februari 2022

Susunan Dewan Pengaji

Pembimbing Utama

Jemmy Edwin B, S.Kom., M.Eng

Pembimbing Pendamping

Yumarlin MZ, S.Kom., M.Pd., M.Kom

Anggota Dewan Pengaji Lain

Ryan Ari Setyawan, S.Kom., M.Eng

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Komputer
Tanggal: 23 Februari 2022

Ketua Jurusan Informatika

Ryan Ari Setyawan, S.Kom., M.Eng.

NPP. 113Q50488

Mengetahui,
Dekan Fakultas Teknik
Universitas Janabadra

Titiek Widwasari, S.T., M.T.

NIP. 19901030

PERSEMBAHAN

Skripsi ini saya persembahkan kepada:

Allah SWT yang telah melimpahkan rahmat dan karunia-Nya. Sehingga tugas akhir dengan judul “Implementasi *Dashboard* untuk Visualisasi Data Penerimaan Mahasiswa Baru Universitas Janabadra Menggunakan *Metabase*” dapat terselesaikan.

Solawat serta salam semoga selalu tercurah limpahkan kepada Nabi Muhammad SAW.

Kepada diri saya sendiri yang telah berjuang menyelesaikan skripsi, terimakasih karena sudah bekerja keras selama ini.

Kepada kedua orang tua saya bapak dan ibu tercinta dengan penuh kesabaran dan pengorbanannya selalu memberikan dorongan, bantuan material maupun nonmaterial agar penulis dapat menyelesaikan studi.

Ucapan terimakasih yang sebesar-besarnya kepada Bapak Jemmy Edwin B, S,Kom, M.Eng selaku dosen pembimbing utama, Ibu Yumarlin MZ, S.kom., M.Pd, M.Kom selaku dosen wali dan dosen pendamping yang selalu membantu, memberikan masukan, dan memberikan semangat kepada mahasiswanya. Juga tak lupa kepada seluruh dosen Program Studi Informatika Universitas Janabadra yang telah memberikan banyak sekali ilmunya.

Serta untuk teman-teman terimakasih karena sudah banyak membantu dan memotivasi saya.

Semoga kita semua selalu dalam lindungan ALLAH SWT

MOTTO

“Jangan pernah kehilangan harapan. Badai membuat orang lebih kuat
dan tidak pernah bertahan selamanya”

~Roy T. Bennett

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga Skripsi yang berjudul **“Implementasi Dashboard untuk Visualisasi Data Penerimaan Mahasiswa Baru Universitas Janabadra Menggunakan Metabase”**, dapat diselesaikan dengan baik dan sesuai waktu yang telah ditentukan tanpa suatu halangan apapun. Tujuan penulisan skripsi ini adalah untuk memenuhi syarat dalam mendapatkan gelar Strata 1 (S1) di Perguruan Tinggi.

Dalam menyusun Skripsi ini, penulis mendapat banyak bantuan, dukungan dan bimbingan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Dr. Ir. Edy Sriyono,M.T., selaku Rektor Universitas Janabadra
2. Titiek Widyasari, S.T.,M.T., selaku Dekan Fakultas Teknik, Universitas Janabadra.
3. Ryan Ari Setyawan, S.Kom.,M.Eng., selaku Ketua Program Studi Informatika.
4. Jemmy Edwin B, S.Kom.,M.Eng., selaku Dosen Pembimbing Utama yang telah membimbing penulisan selama mengerjakan skripsi ini
5. Yumarlin MZ, S.Kom., M.Pd.,M.Kom selaku Dosen wali dan Dosen Pembimbing Pendamping yang telah memberikan dukungan dan pengarahan dalam menyelesaikan skripsi ini.
6. Keluarga, saudara dan teman yang telah memberikan dukungan yang kuat kepada penulis.

Penulis menyadari bahwa di dalam penyusunan Skripsi ini terdapat banyak kekurangan dan kesalahan. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun, sebagai bahan perbaikan dimasa yang akan datang untuk menjadi lebih baik lagi. Semoga laporan Skripsi ini dapat memberi manfaat bagi pembaca umumnya dan bagi penulis khususnya.

Yogyakarta, 14 Februari 2022

Tan Anugrah Ramadhani

NIM 18330119

ABSTRAK

Sistem pelaporan yang tidak terintegrasi dengan baik dapat menghambat monitoring pencapaian target yang diinginkan. Adanya *dashboard* visualisasi yang merupakan alat bantu pembuatan laporan PMB yang dapat digunakan sebagai informasi pendukung dalam rangka pengambilan keputusan oleh struktural Universitas Janabadra. Penelitian ini berjudul ‘‘Implementasi *Dashboard* untuk Visualisasi Data Penerimaan Mahasiswa Baru Universitas Janabadra menggunakan *Metabase*’’.

Metodologi penelitian menggunakan *business intelligent roadmap* dengan 6 tahapan yaitu *justification, planning, business case, design, construction dan deployment* yang menjadi acuan dalam perancangan serta pembangunan data *warehouse* dan *dashboard* visualisasi penerimaan mahasiswa baru Universitas Janabadra. Proses membangun data *warehouse*, menggunakan *airflow* untuk integrasi data dari data *source* menuju *data lake* dan *python script* untuk proses *ETL* data menjadi data *warehouse* dan pembuatan *dashboard* menggunakan *metabase*.

Dari data yang diperoleh, *Dashboard* penerimaan mahasiswa baru dapat dibuat berjumlah empat halaman yaitu, *dashboard* calon mahasiswa, *dashboard* profil calon mahasiswa, *dashboard* mahasiswa registrasi dan *dashboard* profil mahasiswa registrasi. Pada *dashboard* calon mahasiswa dapat dilihat total calon mahasiswa yang mendaftar program PMB Universitas Janabadra dari tahun akademik 2017/2018 yang berjumlah 963 terus mengalami peningkatan hingga tahun akademik 2019/2020 menjadi 1.261 orang dan selanjutnya mengalami penurunan sejak pandemi covid-19 pada tahun akademik 2020/2021 dan terus mengalami penurunan hingga titik terendah dalam 5 tahun terakhir yakni tahun akademik 2021/2022 berjumlah 653 orang.

Pada *dashboard* mahasiswa registrasi dapat dilihat total mahasiswa yang melakukan registrasi pada program PMB Universitas Janabadra dari tahun akademik 2018/2019 berjumlah 694 orang dan terus mengalami peningkatan hingga tahun akademik 2019/2020 menjadi 1.011 orang dan selanjutnya mengalami penurunan sejak pandemi covid-19 pada tahun akademik 2020/2021 berjumlah 813 orang dan total mahasiswa registrasi PMB terus mengalami penurunan hingga titik terendah dalam 4 tahun terakhir yakni tahun akademik 2021/2022 berjumlah 508 orang. Dari *dashboard* calon mahasiswa dan mahasiswa registrasi mayoritas calon mahasiswa dan mahasiswa yang melakukan registrasi berasal dari Provinsi DIY diikuti dengan Provinsi Jawa Tengah dan fakultas yang paling banyak diminati mahasiswa adalah Fakultas Hukum dan yang paling sedikit peminat adalah Fakultas Pertanian.

Kata kunci: *airflow, business intelligence roadmap, data warehouse, dashboard, ETL, Metabase*.

PERNYATAAN

Yang bertanda tangan dibawah ini penulis :

Nama : Tan Anugrah Ramadhani
Nomor Induk Mahasiswa : 18330119
Program Studi : Informatika
Fakultas : Teknik

Dengan ini menyatakan bahwa Skripsi yang berjudul :

**"IMPLEMENTASI DASHBOARD UNTUK VISUALISASI DATA
PENERIMAAN MAHASISWA BARU UNIVERSITAS JANABADRA
MENGGUNAKAN METABASE"**

Adalah benar hasil penelitian dan karya penulis dan belum pernah diajukan sebagai karya ilmiah, sebagian atau seluruhnya. Apabila kemudian hari terbukti bahwa penelitian penulis adalah hasil penelitian pihak lain, penulis bersedia menerima sanksi yang akan dibebankan kepada penulis. Demikian pernyataan ini penulis buat dalam keadaan sehat dan tanpa tekanan pihak manapun.

Yogyakarta, 14 Februari 2022



Tan Anugrah Ramadhani
NIM 18330119

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
PERSEMBERAHAN	iii
MOTTO	iv
KATA PENGANTAR	v
ABSTRAK	vi
PERNYATAAN	Error! Bookmark not defined.
DAFTAR ISI.....	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian.....	4
1.7 Sistematika Penyusunan	6
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	8
2.1 Tinjauan Pustaka	8
2.2 Landasan Teori.....	15
2.3 Perangkat Lunak.....	23
BAB III ANALIS DAN PERANCANGAN	33
3.1 Analisis Sistem.....	33

3.2 Perancangan Sistem.....	34
BAB IV HASIL DAN PEMBAHASAN	111
4.1 Implementasi Dashboard PMB Universitas Janabadra.....	111
4.2 <i>Deployment</i>	115
4.3 Validasi Data	130
4.4 Analisis <i>Dashboard</i>	131
BAB V KESIMPULAN DAN SARAN.....	134
5.1 Kesimpulan	134
5.2 Saran	135
DAFTAR PUSTAKA	136
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 Kekurangan Tinjauan Pustaka	8
Tabel 2.2 Kelebihan Tinjauan Pustaka.....	11
Tabel 3.1 <i>Business Intelligent Roadmap</i> PMB Universitas Janabadra	34
Tabel 3.2 Tabel Informasi yang akan ditampilkan.....	36
Tabel 3.3 Tabel mhs_cln	38
Tabel 3.4 Tabel m_tahun.....	40
Tabel 3.5 Tabel adm_provinsi.....	40
Tabel 3.6 Tabel mhs_regis	41
Tabel 3.7 Tabel m_fakultas.....	43
Tabel 3.8 Tabel m_prodi	44
Tabel 3.9 Tabel m_agama	44
Tabel 3.10 Tabel m_kewarganegaraan	45
Tabel 3.11 Konfigurasi Instance	49
Tabel 3. 12 Tabel dim_waktu.....	66
Tabel 3.13 Tabel fakta calon mahasiswa	67
Tabel 3.14 Tabel fakta mahasiswa	75

DAFTAR GAMBAR

Gambar 1.1 Business Intelligence Roadmap (L.T. Moss, and S. Atre,2003)	5
Gambar 2.1 Alur Proses Bussines Intellegence (AltexSoft Inc,2019).....	15
Gambar 2.2 Grafik batang (Ibeng Parta,2021).....	22
Gambar 2.3 Grafik Garis (Ibeng Parta,2021).....	22
Gambar 2.4 Grafik Lingkaran (Ibeng Parta,2021)	23
Gambar 2.5 Arsitektur <i>Docker</i> (Docker,2021)	23
Gambar 2.6 Tampilan Antarmuka <i>Google Cloud Platform</i>	26
Gambar 2. 7 Tampilan <i>Software Datagrip</i>	28
Gambar 2.8 Tampilan antarmuka <i>Airbyte</i>	28
Gambar 2.9 Tampilan antarmuka Visual Studio Code	30
Gambar 2.10 Tampilan antarmuka <i>metabase</i>	32
Gambar 3.1 <i>Database PMB</i>	36
Gambar 3.2 <i>Star schema</i> calon mahasiswa	45
Gambar 3.3 <i>Star schema</i> mahasiswa.....	46
Gambar 3.4 <i>Create Instance</i>	47
Gambar 3.5 Nama <i>Instance</i>	47
Gambar 3.6 <i>Machine Configuration</i>	48
Gambar 3.7 <i>Public Image</i>	48
Gambar 3.8 <i>Identity</i> dan <i>API access</i>	49
Gambar 3.9 <i>Set RSA Key</i>	50
Gambar 3.10 <i>Input Key RSA</i> komputer ke dalam <i>Metadata</i>	50
Gambar 3.11 <i>Install Extension Remote Development</i>	50
Gambar 3.12 <i>SSH Config</i>	51
Gambar 3.13 <i>Connect to host</i>	51
Gambar 3.14 <i>Container Docker</i> pada <i>Instance-db</i>	53
Gambar 3.15 <i>Container Docker</i> pada <i>Instance-dwh</i>	54
Gambar 3.16 Container pada Instace-etl	57
Gambar 3.17 Docker Container Pada Instace -metabase	59
Gambar 3.18 <i>Setup Metabase</i>	60
Gambar 3.19 Koneksi <i>data warehouse postgresql</i>	60

Gambar 3.20 Alur data pembuatan data <i>warehouse</i>	61
Gambar 3.21 <i>Import Data source</i>	61
Gambar 3.22 <i>Data Source PMB</i>	62
Gambar 3.23 <i>Set data source</i>	62
Gambar 3.24 <i>Set data destination</i>	63
Gambar 3.25 Connection Setting	64
Gambar 3.26 <i>Connection</i> data pada <i>Airbyte</i>	64
Gambar 3.27 <i>Data Lake PMB</i>	65
Gambar 3.28 Susunan Modul <i>Script Python</i>	66
Gambar 3.29 <i>Run Main.py</i> Fakta Calon Mahasiswa.....	74
Gambar 3.30 Run Main.py Fakta Mahasiswa.....	81
Gambar 3.31 Tampilan tabel fakta calon mahasiswa.....	82
Gambar 3.32 Tampilan tabel fakta mahasiswa	82
Gambar 3.33 Create Question	83
Gambar 3.34 Total calon mahasiswa PMB.....	84
Gambar 3.35 Calon Mahasiswa Berdasarkan Program Studi dan Kelas	85
Gambar 3.36 Calon Mahasiswa Berdasarkan Tahun Akademik dan Semester	86
Gambar 3.37 Jumlah pendaftar berdasarkan tahun akademik tiap Prodi.....	87
Gambar 3. 38 Tampilan jumlah pendaftar berdasarkan semester	88
Gambar 3.39 Tampilan Jumlah pendaftar berdasarkan kelas	89
Gambar 3.40 Tampilan jumlah calon mahasiswa berdasarkan jenis kelamin	90
Gambar 3.41 Jumlah calon mahasiswa berdasarkan status kewarganegaraan.....	91
Gambar 3.42 Jumlah calon mahasiswa berdasarkan agama yang dianut.....	92
Gambar 3.43 Jumlah calon mahasiswa berdasarkan pekerjaan orang tua	93
Gambar 3.44 Jumlah mahasiswa registrasi	93
Gambar 3.45 Jumlah mahasiswa registrasi berdasarkan program studi dan kelas	94
Gambar 3.46 Jumlah mahasiswa registrasi berdasarkan tahun akademik	95
Gambar 3.47 Tampilan jumlah mahasiswa registrasi berdasarkan kelas.....	96
Gambar 3.48 Tampilan jumlah mahasiswa registrasi berdasarkan jalur PMB	97
Gambar 3.49 Tampilan jumlah mahasiswa registrasi berdasarkan alih jenjang ...	98
Gambar 3.50 Tampilan jumlah mahasiswa registrasi berdasarkan jenis kelamin	99
Gambar 3.51 Jumlah mahasiswa registrasi status kewarganegaraan.....	100
Gambar 3.52 Jumlah mahasiswa registrasi berdasarkan jurusan SLTA	101

Gambar 3.53 Jumlah mahasiswa registrasi berdasarkan provinsi.....	102
Gambar 3.54 Jumlah mahasiswa registrasi berdasarkan Pendidikan orang tua..	103
Gambar 3.55 Jumlah mahasiswa registrasi berdasarkan pendapatan orang tua..	104
Gambar 3.56 Tampilan jumlah mahasiswa berdasarkan pekerjaan orang tua	105
Gambar 3.57 Tampilan jumlah Pendapatan berdasarkan tanggal bayar	105
Gambar 3.58 Menambahkan Filter Pada <i>Question</i>	106
Gambar 3.59 <i>New Dashboard</i>	107
Gambar 3.60 <i>Add Question</i>	108
Gambar 3.61 <i>Add Filter</i>	109
Gambar 3.62 <i>Set Filter</i>	109
Gambar 3.63 Tes Fungsi Filter.....	109
Gambar 4.1 <i>Dashboard</i> informasi calon mahasiswa PMB	112
Gambar 4.2 <i>Dashboard</i> profil calon mahasiswa PMB Universitas Janabadra ...	113
Gambar 4.3 <i>Dashboard</i> mahasiswa registrasi PMB Universitas Janabadra	114
Gambar 4.4 <i>Dashboard</i> profil mahasiswa registrasi PMB	115
Gambar 4.5 Tampilan visualisasi jumlah calon mahasiswa PMB	115
Gambar 4.6 Jumlah pendaftar berdasarkan program studi dan kelas.....	116
Gambar 4.7 Jumlah pendaftar PMB berdasarkan tahun akademik	117
Gambar 4.8 Jumlah pendaftar berdasarkan tahun akademik tiap Prodi.....	117
Gambar 4.9 Jumlah pendaftar berdasarkan semester.....	118
Gambar 4.10 Jumlah pendaftar berdasarkan kelas.....	119
Gambar 4.11 Tampilan jumlah calon mahasiswa berdasarkan jenis kelamin	119
Gambar 4.12 Calon mahasiswa berdasarkan status kewarganegaraan	120
Gambar 4.13 Jumlah calon mahasiswa berdasarkan agama	120
Gambar 4.14 Jumlah calon mahasiswa berdasarkan pekerjaan orang tua	121
Gambar 4.15 Jumlah mahasiswa registrasi	121
Gambar 4.16 Mahasiswa registrasi berdasarkan program studi dan kelas.....	122
Gambar 4.17 Jumlah mahasiswa registrasi berdasarkan tahun akademik	123
Gambar 4.18 Tampilan jumlah mahasiswa registrasi berdasarkan kelas.....	123
Gambar 4.19 Jumlah mahasiswa registrasi berdasarkan jalur PMB	124
Gambar 4.20 Jumlah mahasiswa registrasi berdasarkan alih jenjang	125
Gambar 4.21 Jumlah mahasiswa registrasi berdasarkan jenis kelamin	125
Gambar 4.22 Jumlah mahasiswa registrasi status kewarganegaraan	126

Gambar 4.23 Jumlah mahasiswa registrasi berdasarkan jurusan SLTA	127
Gambar 4.24 Jumlah mahasiswa registrasi berdasarkan provinsi.....	127
Gambar 4.25 Jumlah mahasiswa registrasi berdasarkan Pendidikan orang tua..	128
Gambar 4.26 Jumlah mahasiswa registrasi berdasarkan pendapatan orang tua..	129
Gambar 4.27 Jumlah mahasiswa berdasarkan pekerjaan orang tua.....	129
Gambar 4.28 Jumlah Pendapatan berdasarkan tanggal bayar.....	130

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pendidikan adalah proses pengubahan sikap dan tata laku seseorang atau kelompok orang dalam usaha mendewasakan manusia melalui upaya pengajaran dan pelatihan. Berdasarkan UU No. 2 Tahun 1985 yang berbunyi bahwa tujuan pendidikan yaitu mencerdaskan kehidupan bangsa dan mengembangkan manusia yang seutuhnya yaitu yang beriman dan bertakwa kepada Tuhan Yang Maha Esa dan berbudi pekerti luhur, memiliki pengetahuan dan keterampilan, kesehatan jasmani dan rohani, kepribadian yang mantap dan mandiri serta rasa tanggung jawab kemasyarakatan bangsa (Fikry Muhammad,2021).

Persaingan di dalam dunia Pendidikan sekarang semakin kompetitif, tidak hanya bagi perguruan tinggi negeri tetapi juga perguruan tinggi swasta. Para mahasiswa yang tidak diterima di perguruan tinggi negeri akan berlomba-lomba mencari universitas swasta terbaik dalam melanjutkan pendidikannya. Tujuan perguruan tinggi swasta dalam membantu pemerintah dibidang pendidikan yaitu menghasilkan generasi muda yang berkarakter dan berkualitas. Untuk mendukung hal tersebut, terutama dalam bidang teknologi informasi, diperlukan suatu sistem pengolahan jumlah mahasiswa baru (Yohanna & Rumapea, 2020).

Menurut (Bororing, 2021) Universitas Janabadra (UJB) merupakan salah satu universitas tertua di Yogyakarta. Setiap tahunnya hampir 700 hingga 1000 calon mahasiswa yang mendaftar untuk menjadi mahasiswa di Universitas Janabadra. Sistem penerimaan mahasiswa baru yang ada dan digunakan, masih berupa Sistem Informasi manajemen yang mencatat data-data transaksional secara sederhana dalam bentuk teks dan angka–angka. Informasi yang dihasilkan tersebut belum dapat dijadikan sebagai alat penunjang untuk mendukung keputusan karena masih dalam bentuk data-data tabel yang berisi jumlah calon mahasiswa berdasarkan status penerimaan yaitu pendaftar, diterima dan registrasi. Pelaporan masih dibuat melalui *interface* yang sebatas teks atau angka-angka biasa sehingga masih sulit untuk dimengerti dan diambil kesimpulannya. Sistem pelaporan yang kurang baik

dan kurang menarik bisa menghambat dalam monitoring pencapaian target yang diinginkan.

Business Intelligence adalah seperangkat teori, metodologi, proses, arsitektur, dan teknologi yang mengubah data mentah menjadi informasi yang bermakna dan berguna untuk tujuan bisnis. *Business Intelligence* dapat menangani sejumlah besar informasi untuk membantu mengidentifikasi dan mengembangkan peluang baru (Akbar, Rasyiddah, Anrisya, Fritania Julyazti, & Syaputri, 2018).

Business Intelligence merupakan sebuah sistem dasar bagi hampir seluruh kondisi yang melibatkan pembuatan keputusan bisnis dan formulasi strategi dalam pengembangannya. Menurut (Prasetya & Meme Susilowati, 2020) *Business Intelligence* memungkinkan sebuah Perguruan Tinggi lebih percaya diri dalam menghadapi berbagai Tantangan bisnis, yakni penurunan keuntungan, merosotnya pangsa pasar dan ancaman dari pesaing. *Business intelligent* bukan suatu solusi untuk menyelesaikan tantangan tersebut tetapi dapat mengidentifikasi permasalahan yang muncul dan mempersiapkan solusi penyelesaiannya.

Untuk menunjang penyediaan sistem informasi yang menarik, data-data yang ada dapat diolah dan direpresentasikan dengan menggunakan sistem *dashboard*. *Dashboard* merupakan alat untuk menyajikan informasi yang memberikan tampilan antarmuka dalam berbagai bentuk, baik berupa diagram, laporan, indikator visual maupun mekanisme *alert* yang dipadukan dengan sistem informasi yang dinamis dan relevan serta hasil yang disajikan mudah dipahami dan memberikan gambaran dengan jelas mengenai informasi yang ditampilkan (Yohanna & Rumapea, 2020).

Metabase adalah sebuah *software* atau *tool* yang sering digunakan untuk keperluan *Business Intelligence*. Menurut (Korocz, T. 2020), *Metabase* merupakan *tool open source* yang bisa memvisualisasikan data lewat grafik sehingga proses analisis lebih mudah. *Metabase* digunakan sebagai *tools* untuk mengolah data dan mendapatkan *insight* tentang bisnis yang dimiliki.

Berdasarkan latar belakang yang telah dijabarkan maka penulis mengambil judul skripsi yaitu Implementasi *Dashboard* untuk Visualisasi Data Penerimaan Mahasiswa Baru Universitas Janabadra Menggunakan *Metabase*. Penelitian ini

diharapkan dapat membantu pihak universitas untuk memonitoring penerimaan mahasiswa baru di Universitas Janabadra.

1.2 Rumusan Masalah

Berdasarkan Latar belakang yang diuraikan diatas, permasalahan yang dapat dirumuskan untuk penelitian ini adalah bagaimana merancang model *business intelligence* serta membangun *dashboard* visualisasi penerimaan mahasiswa baru Universitas Janabadra menggunakan *Metabase*?

1.3 Batasan Masalah

Di dalam penyelesaian tugas akhir ini terdapat beberapa hal yang menjadi batasan masalah yaitu:

1. Data yang digunakan merupakan data dari sistem penerimaan mahasiswa baru Universitas Janabadra.
2. Pada penelitian ini merancang dan membangun data *warehouse* dan desain *dashboard* penerimaan mahasiswa baru Universitas Janabadra.
3. Menampilkan data calon mahasiswa yang mendaftar pada tahun 2017-2021 dan mahasiswa registrasi tahun 2018-2019.
4. Hanya data mahasiswa registrasi yang menampilkan mahasiswa berdasarkan provinsi.
5. Menampilkan informasi dalam bentuk *dashboard* visualisasi penerimaan mahasiswa baru Universitas Janabadra menggunakan *metabase*.

1.4 Tujuan Penelitian

Di dalam penyelesaian tugas akhir ini terdapat beberapa hal yang menjadi tujuan dari penelitian ini yaitu:

1. Untuk merancang dan membangun sebuah data *warehouse* PMB menggunakan metodologi *business intelligent roadmap*.
2. Untuk merancang dan membangun sebuah *dashboard* visualisasi data PMB Universitas Janabadra menggunakan *software metabase*.
3. Untuk memberikan informasi secara grafik tentang informasi PMB.

4. Dapat digunakan sebagai informasi pendukung dalam rangka pengambilan keputusan oleh para struktural Universitas Janabadra dan solusi untuk peningkatan kegiatan Admisi dan promosi.
5. Dapat digunakan sebagai alat bantu untuk pembuatan laporan PMB.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini yaitu:

1. Memonitoring penerimaan mahasiswa baru dalam bentuk visual yang mudah dipahami.
2. Memberikan gambaran dengan jelas mengenai informasi yang ditampilkan untuk pengambilan keputusan oleh pihak struktural Universitas Janabadra.
3. Hasil dari penelitian dapat digunakan untuk penelitian sejenis.

1.6 Metodologi Penelitian

1. Pengumpulan data

Dalam penelitian ini metode pengumpulan data yang digunakan adalah sebagai berikut:

a. Wawancara

Wawancara terhadap pihak struktural Universitas Janabadra terkait proses bisnis dan desain dasar *database*.

b. Observasi

Data yang digunakan pada penelitian ini adalah data penerimaan mahasiswa baru Universitas Janabadra.

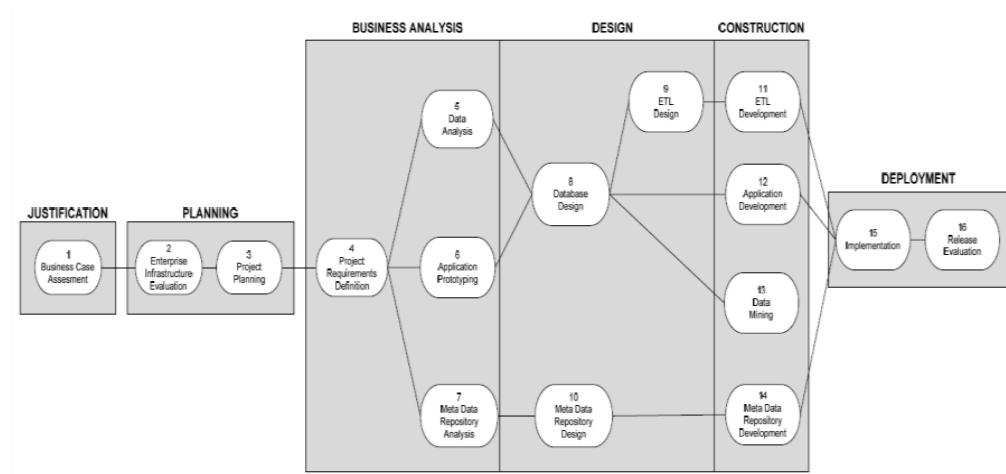
c. Studi Pustaka

Dilakukan dengan cara membaca dan mempelajari jurnal, buku dan materi yang ada di media *online* yang berhubungan dengan judul penelitian sejenis yang berkaitan dengan tema penulisan.

2. Perancangan model *Bussines Intelligence*

Prosedur dalam penelitian ini dilakukan dengan mendeskripsikan setiap urutan secara sistematis dan logis, sehingga dapat dijadikan pedoman yang

jelas untuk menyelesaikan permasalahan. Prosedur penelitian yang dilakukan dalam penelitian ini menggunakan metodologi *business intelligent roadmap* seperti pada gambar 1.1.



Gambar 1.1 Business Intelligence Roadmap (L.T. Moss, and S. Atre,2003)

Menurut (Wibowo & Andri, 2021) *Business Intelligence Roadmap* merupakan panduan untuk mengembangkan BI sebagai *decision support*. *Business Intelligence Roadmap* merupakan strategi atau langkah-langkah untuk mengimplementasikan *Business Intelligence* di dalam perusahaan atau organisasi. Ada pun langkah-langkah yang akan dilakukan dalam penelitian ini adalah sebagai berikut:

a. *Justification*

Pada tahap ini berisi evaluasi dan *assessment* kebutuhan serta mendefinisikan masalah dan peluang bisnis dari penerimaan mahasiswa baru Universitas Janabdra.

b. *Planning*

Pada tahapan ini berfokus dalam mengembangkan rencana strategis dan taktis yang menghasilkan bagaimana proyek BI untuk penerimaan mahasiswa baru Universitas Janabadra akan dikerjakan dan diselesaikan.

c. *Business Analysis*

Fase *business analysis* berfokus dalam melakukan analisis yang mendetail dari masalah dan peluang bisnis untuk mendapatkan pemahaman yang mendalam dari kebutuhan bisnis terhadap solusi produk.

d. *Design*

Tahapan *design* ini berisi proses perancangan *database* yang akan dimanfaatkan untuk pembangunan data *warehouse* dan *dashboard* dari data penerimaan mahasiswa baru Universitas Janabadra.

e. *Construction*

Tahapan *Construction* ini adalah proses membangun data *warehouse*, menggunakan *software airflow* untuk integrasi data dari data *source* ke *data lake* dan *python script* untuk proses *ETL* data menjadi data *warehouse* dan pembuatan dashboard menggunakan *metabase*

f. *Deployment*

Tahapan ini berisi pengujian yang dilakukan terhadap *dashboard* yang telah berhasil dibuat.

1.7 Sistematika Penyusunan

Penyusunan laporan tugas akhir ini akan disusun secara sistematik menjadi lima bab yaitu sebagai berikut:

BAB I : PENDAHULUAN

Bab ini berisi pengumpulan data yang digunakan dan tersusun dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini berisi tentang dasar teori yang mendukung penelitian dan teori yang melandasi proses pembuatan sistem.

BAB III: PERANCANGAN SISTEM

Bab ini menggambarkan tentang perancangan analisa dan perencanaan sistem yang memuat tentang analisa kebutuhan dari sistem yang akan dibuat, beserta rancangan sistem.

BAB IV : PEMBAHASAN DAN IMPLEMENTASI SISTEM

Bab ini berisi tentang implementasi sistem yang akan dibuat, melakukan pengujian, menganalisa kinerja dan penjelasan aturan cara pengoperasian.

BAB V : PENUTUP

Bab ini terdiri atas kesimpulan dan saran dari hasil implementasi sistem.

DAFTAR PUSTAKA**LAMPIRAN**

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Bagian ini mengulas penelitian-penelitian bidang sejenis yang telah dilakukan sebelumnya. Susunan tinjauan pustaka dari yang bersifat umum menuju khusus (*general to specific*). Tinjauan pustaka ini dipelajari dari skripsi, hasil seminar maupun jurnal.

Beberapa penelitian terdahulu yang menjadi referensi penelitian ini dapat dilihat pada tabel 2.1 yang berisi kekurangan dari hasil penelitian terdahulu dan tabel 2.2 berisi kelebihan dari hasil penelitian terdahulu.

Tabel 2.1 Kekurangan Tinjauan Pustaka

No	Penelitian	Metode	Hasil Analisa	Kekurangan
1.	Analisis Penerapan Model <i>Business Intelligence</i> pada Aplikasi PPOB (<i>Payment Point Online Banking</i>) dalam Meningkatkan Strategi Pemasaran (Studi Kasus: Aplikasi Apotikkuota). (Yunistira & Fudholi, 2020)	Prosedur dalam penelitian ini dilakukan dengan mendeskripsikan setiap urutan yaitu : 1.Mendeskripsikan objek dan subyek 2.Menentukan ruang lingkup penelitian 3.Jenis dan sumber data 4.Teknik pengumpulan data 5.Identifikasi dan analis kebutuhan system. 6. Menganalisis data 7. merancang data <i>warehouse</i> 8. Pengolahan data	Hasil penelitian ini, yaitu menyajikan informasi yang dibutuhkan oleh <i>stakeholder</i> dalam proses pengambilan keputusan untuk meningkatkan strategi pemasaran pada aplikasi Apotikkuota. Dalam proses pengambilan keputusan, penerapan strategi bauran pemasaran (marketing mix) yang didalamnya terdapat komponen 4P yaitu: <i>Price</i> (Harga), <i>Product</i> (Produk), <i>Place</i> (Distribusi), serta <i>Promotion</i> (Promosi) menjadi acuannya	1. Data <i>warehouse</i> sebagai sumber data BI, seharusnya tidak hanya bersumber dari OLTP, namun dapat juga diambil dari berbagai sumber baik <i>online</i> maupun <i>offline</i> . Sehingga dalam penerapannya informasi yang dihasilkan dapat lebih akurat dan sempurna. 2. Dalam membangun data <i>warehouse</i> perlu adanya mekanisme yang berkelanjutan, sehingga dapat terus diperbarui agar informasi yang didapatkan lebih sempurna. 3. Penelitian ini mengandalkan data OLTP yang selanjutnya diolah

		9. Kesimpulan dan saran.		pada data <i>warehouse</i> . Kedepannya diharapkan proses pengambilan data dapat dilakukan secara langsung bersumber pada data dalam SQL Server aplikasi Apotikkuota. Sehingga informasi yang didapat diharapkan <i>update</i> setiap waktu secara cepat dan akurat.
2.	Implementasi Aplikasi <i>Business Intelligence</i> Untuk Memonitor Efisiensi Pengelolaan Rumah Sakit. (Herwanto & Khumaidi, 2020)	Kegiatan penelitian ini dilakukan dengan tiga tahapan utama, yaitu proses konsolidasi data, pembangunan data <i>warehouse</i> , dan pembuatan aplikasi BI.	Penelitian ini menggali penggunaan teknologi informasi untuk membangun aplikasi BI, meninjau pendekatan yang tepat dalam membangun aplikasi BI, serta beberapa aspek penting yang harus dipertimbangkan agar sistem bekerja untuk lingkungan rumah sakit. Terdapat dua tahapan utama dalam membangun aplikasi ini, yaitu: membangun data <i>warehouse</i> yang berasal dari <i>database</i> rekam medis elektronik dan operasional rumah sakit, serta membangun aplikasi BI. Dengan terbentuknya aplikasi BI ini, sangat berguna bagi manajemen rumah sakit dalam mengelola	Pada penelitian ini masih membutuhkan penyempurnaan terutama pelaporan dalam bentuk grafis.

			institusinya dengan lebih baik.	
3.	Pemanfaatan <i>Business Intelligent</i> di Perguruan Tinggi(Prasetya & Meme Susilowati, 2020)	<i>Software Development Life Cycle</i> (SDLC)	<p>1.Penelitian ini bertujuan menyajikan sebuah informasi visualisasi yang lengkap sesuai proses bisnis</p> <p>2. visualisasi informasi digunakan untuk mengukur kinerja dan pengambil keputusan bagi pimpinan perguruan tinggi</p> <p>3. <i>Business Intelligence</i> pada Perguruan Tinggi untuk mengetahui statistik perkembangan dan kinerja dari pihak-pihak fakultas, jurusan, program studi, dosen, staf, mahasiswa, penelitian, pengabdian masyarakat dan sebagainya</p>	Aplikasi ini masih membutuhkan penyempurnaan terutama pelaporan dalam bentuk grafis.
4.	implementasi sistem <i>business intelligence</i> untuk data penelitian di perguruan tinggi(Hasan, 2019)	<i>Water fall</i>	<p>1. Perancangan meliputi perancangan data <i>warehouse</i>, perancangan OLAP <i>analysis</i>, perancangan report dan perancangan <i>dashboard</i>.</p> <p>2. Report yang dirancang membantu dalam membuat laporan penelitian dan dapat disesuaikan dengan kebutuhan.</p> <p>3. <i>Dashboard</i> membantu para</p>	<i>Dashboard</i> yang dirancang pada penelitian ini masih harus dilakukan pengembangan lebih lanjut agar dapat memenuhi kebutuhan informasi secara detail, akurat dan menyeluruh.

			pimpinan didalam menganalisis data untuk mempelajari tren penelitian yang dilakukan diperguruan tinggi.	
5.	Penerapan Business Intelligence pada CV. Tanggamas Chemical dengan metode OLAP (Berlin & Giap, 2020)	<i>OLAP,Bussiness Intelligent Roadmap</i>	<p>1.Dengan menggunakan web business intelligence ini permasalahan pada pengolahan data penjualan CV. Tangga Mas Chemical dapat diatasi karena penggunaannya yang mudah dan dapat berlangsung lebih optimal.</p> <p>2. Dengan menggunakan web business intelligence ini data dapat diolah secara cepat dan memudahkan dalam menentukan keputusan untuk meningkatkan proses bisnis.</p> <p>3. Penelitian ini menghasilkan business intelligence penjualan barang menggunakan web yang menampilkan informasi dalam bentuk grafik</p>	<p>masih terdapat keterbatasan dan kekurangan serta memerlukan perbaikan untuk meningkatkan manfaat dari business intelligence ini yang dapat dikembangkan pada penelitian selanjutnya adalah sebagai berikut :</p> <ol style="list-style-type: none"> 1. Memanfaatkan penggunaan teknik data mining sehingga pola dari data dapat dilihat dengan lebih jelas dan informasi yang didapat dari data yang ada lebih mendalam. 2. Menggunakan indikator pengukuran key performance indicator untuk pemantauan penjualan barang pada masing-masing barang, sehingga dapat diketahui tingkatannya.

Tabel 2.2 Kelebihan Tinjauan Pustaka

No	Penelitian	Metode	Hasil Analisa	Kelebihan
1.	Analisis Penerapan Model <i>Business Intelligence</i> pada Aplikasi PPOB	Prosedur dalam penelitian ini dilakukan dengan mendeskripsikan setiap urutan yaitu :	Hasil penelitian ini, yaitu menyajikan informasi yang dibutuhkan oleh <i>stakeholder</i> dalam proses pengambilan keputusan untuk	hasil analisis OLAP digunakan sebagai informasi dalam penerapan strategi bauran pemasaran (marketing mix) berdasarkan empat

	(Payment Point Online Banking) dalam Meningkatkan Strategi Pemasaran (Studi Kasus: Aplikasi Apotikkuota). (Yunistira & Fudholi, 2020)	<ol style="list-style-type: none"> 1. Mendeskripsikan objek dan subyek. 2. Menentukan ruang lingkup penelitian. 3. Jenis dan sumber data. 4. Teknik pengumpulan data. 5. Identifikasi dan analisis kebutuhan sistem. 6. Menganalisis data. 7. Merancang data <i>warehouse</i>. 8. Pengolahan data. 9. Kesimpulan dan saran. 	<p>meningkatkan strategi pemasaran pada aplikasi Apotikkuota. Dalam proses pengambilan keputusan, penerapan strategi bauran pemasaran (marketing mix) yang didalamnya terdapat komponen 4P yaitu: <i>Price</i> (Harga), <i>Product</i> (Produk), <i>Place</i> (Distribusi), serta <i>Promotion</i> (Promosi) menjadi acuannya.</p>	<p>komponen utamanya. Sehingga kebutuhan penerapan BI dalam hal peningkatan strategi pemasaran untuk menghasilkan informasi yang dibutuhkan sebagai acuan dalam pengambilan keputusan.</p>
2.	Implementasi Aplikasi <i>Business Intelligence</i> Untuk Memonitor Efisiensi Pengelolaan Rumah Sakit. (Herwanto & Khumaidi, 2020)	<p>Kegiatan penelitian ini dilakukan dengan tiga tahapan utama, yaitu proses konsolidasi data, pembangunan data <i>warehouse</i>, dan pembuatan aplikasi BI.</p>	<p>Penelitian ini menggali penggunaan teknologi informasi untuk membangun aplikasi BI, meninjau pendekatan yang tepat dalam membangun aplikasi BI, serta beberapa aspek penting yang harus dipertimbangkan agar sistem bekerja untuk lingkungan rumah sakit. Terdapat dua tahapan utama dalam membangun aplikasi ini, yaitu: membangun data <i>warehouse</i> yang berasal dari <i>database</i> rekam medis elektronik</p>	<p>Pada penelitian ini untuk mengukur tingkat efisiensi pengelolaan rumah rumah sakit diperlukan indikator pelayanan rumah sakit. Dengan adanya indikator tersebut dapat mengetahui tingkat pemanfaatan, mutu serta efisiensi pelayanan yang terdapat di rumah sakit. Penelitian ini berfokus pada penerapan aplikasi BI yang digunakan untuk memudahkan manajemen dalam memonitor efisiensi pengelolaan rumah sakit.</p>

			dan operasional rumah sakit, serta membangun aplikasi BI. Dengan terbentuknya aplikasi BI ini, sangat berguna bagi manajemen rumah sakit dalam mengelola institusinya dengan lebih baik.	
3.	Pemanfaatan <i>Business Intelligent</i> di Perguruan Tinggi(Prasetya & Meme Susilowati, 2020)	<i>Software Development Life Cycle</i> (SDLC)	<p>1. Penelitian ini bertujuan menyajikan sebuah informasi visualisasi yang lengkap sesuai proses bisnis.</p> <p>2. Visualisasi informasi digunakan untuk mengukur kinerja dan pengambil keputusan bagi pimpinan perguruan tinggi.</p> <p>3. <i>Business Intelligence</i> pada Perguruan Tinggi untuk mengetahui statistik perkembangan dan kinerja dari pihak-pihak Fakultas, Jurusan, Program Studi, dosen, staf, mahasiswa, penelitian, pengabdian masyarakat dan sebagainya.</p>	Menyajikan sebuah informasi yang lengkap yang dirancang dengan sedemikian rupa sehingga dapat menampilkan data-data sesuai proses bisnis yang terjadi yang dapat dipergunakan oleh perguruan tinggi untuk mengukur kinerjanya dan untuk pengambilan keputusan.
4.	Implementasi sistem <i>Business Intelligence</i> untuk data penelitian di perguruan	<i>Waterfall</i>	<p>1. Perancangan meliputi perancangan data <i>warehouse</i>, perancangan OLAP <i>analysis</i>, perancangan report</p>	Pada penelitian ini dengan adanya <i>dashboard</i> para pemegang keputusan mendapatkan ringkasan informasi sebagai landasan untuk menetapkan

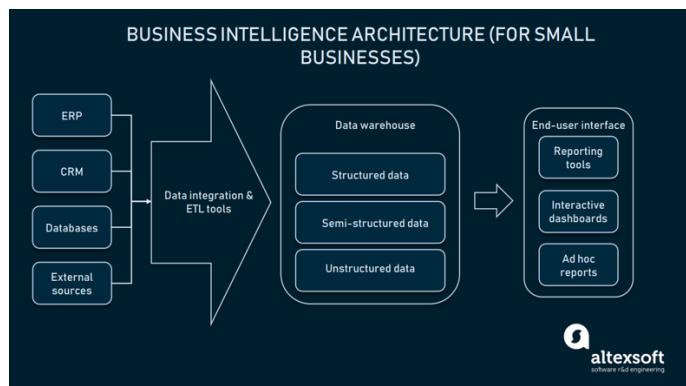
	tinggi. (Hasan, 2019)		<p>dan perancangan <i>dashboard</i>.</p> <p>2. Report yang dirancang membantu dalam membuat laporan penelitian dan dapat disesuaikan dengan kebutuhan.</p> <p>3. <i>Dashboard</i> membantu para pimpinan didalam menganalisis data untuk mempelajari tren penelitian yang dilakukan di Perguruan Tinggi.</p>	keputusan penting bagi Perguruan Tinggi.
5.	Penerapan Business Intelligence pada CV. Tanggamas Chemical dengan metode OLAP (Berlin & Giap, 2020)	<i>OLAP,Business Intelligent Roadmap</i>	<p>1.Dengan menggunakan web business intelligence ini permasalahan pada pengolahan data penjualan CV. Tangga Mas Chemical dapat diatasi karena penggunaannya yang mudah dan dapat berlangsung lebih optimal.</p> <p>2. Dengan menggunakan web business intelligence ini data dapat diolah secara cepat dan memudahkan dalam menentukan keputusan untuk meningkatkan proses bisnis.</p> <p>3. Penelitian ini menghasilkan business intelligence penjualan barang menggunakan web yang menampilkan informasi dalam bentuk grafik</p>	Hasil dari penelitian ini adalah dapat mengolah data yang banyak dan dapat menghasilkan laporan berupa dashboard web untuk membuat keputusan yang lebih baik lagi bagi manajer CV. Tangga Mas Chemical.

2.3 Landasan Teori

1. Business Intelligence

Business Intelligent atau BI adalah sekumpulan praktik pengumpulan, penataan, analisis, dan pengubahan data mentah menjadi wawasan bisnis yang dapat ditindaklanjuti. BI mempertimbangkan metode dan alat yang mengubah kumpulan data tidak terstruktur, menyusunnya menjadi laporan atau *dashboard* informasi yang mudah dipahami. Tujuan utama BI adalah untuk memberikan wawasan bisnis yang dapat ditindaklanjuti dan mendukung pengambilan keputusan berdasarkan data (AltexSoft Inc,2019).

Alur Proses Data *Business Intelligence* dapat dilihat pada gambar 2.1.



Gambar 2.1 Alur Proses Bussines Intellegence (AltexSoft Inc,2019)

Berikut penjelasan alur proses *Bussiness Intelligent*:

a. Proses *ETL (Extract, Transform, Load)*

Data dari berbagai macam sumber diolah melalui proses *ETL* yang meliputi :

1) *Data extraction* (Ekstraksi data).

Alat *ETL* mengambil data dari sumber data termasuk *ERP*, *CRM*, analitik, dan *spreadsheet*.

2) *Data transform* (Transformasi data).

Setelah diekstrak, alat *ETL* memulai pemrosesan data. Transformasi *ETL* merupakan pembersihan dan mempersiapkan agregasi untuk

analisis. Pada Langkah ini memastikan data yang akan diolah sepenuhnya siap dan kompatibel seperti pembersihan data, standardisasi data, deduplikasi data, verifikasi data, Pengurutan data dan aturan-aturan tambahan yang dapat meningkatkan kualitas data.

3) *Data Loading* (Pemuatan data).

Pada fase ini, data yang diubah diunggah ke data *warehouse*.

b. *Data Warehouse*

Menurut (AltexSoft Inc,2019) Data Warehouse adalah jenis *database* tertentu yang biasanya menyimpan informasi historis dalam format *SQL*. *Data Warehouse* terhubung dengan sumber data dan sistem *ETL* di satu ujung dan alat pelaporan atau antarmuka *dashboard* di sisi lain. Ini memungkinkan untuk menyajikan data dari berbagai sistem melalui satu antarmuka. Tujuan utama dari pembuatan *data warehouse* adalah untuk menyatukan data beragam ke dalam sebuah tempat penyimpanan, dimana pengguna dengan mudah menjalankan *query* (pencarian data), menghasilkan laporan, dan melakukan analisis.

c. *Bussiness Intelligent Result*

Hasil dari BI dapat digunakan untuk berbagai macam keperluan seperti pengolahan data multi dimensional, atau analisa dengan aplikasi *Dashboard*, atau ditampilkan ke laporan / *report* dalam bentuk grafik, dapat digunakan lagi untuk Data *mining* dimana data digali untuk dilihat polanya dan kemudian dapat digunakan untuk melakukan prediksi, dan bisa digunakan untuk sistem *alert* / *warning* sistem (Jimi,2019).

2. *Dashboard*

Dashboard merupakan sebuah alat untuk menyajikan informasi yang memberikan tampilan antarmuka dalam berbagai bentuk, baik berupa diagram, laporan, indikator visual maupun mekanisme *alert* yang dipadukan dengan sistem informasi yang dinamis dan relevan. Model yang disajikan mudah dipahami dan memberikan gambaran dengan jelas mengenai informasi yang ditampilkan (Yohanna & Rumapea, 2020).

Dashboard menyajikan informasi mengenai indikator utama dari aktifitas lembaga atau organisasi secara sekilas. Biasanya *dashboard* menghasilkan model yang digunakan untuk pengembangan kebutuhan monitoring dan evaluasi kinerja dari suatu perusahaan. Monitoring dan evaluasi kinerja dilakukan secara terus menerus oleh suatu perusahaan untuk memastikan bahwa proses bisnis yang dijalankannya dapat mencapai tujuan yang telah ditetapkan melalui strategi pengelolaan yang tepat (Hijriani, Aprilliana, Pribadi, & Sakethi, 2020).

4. Visualisasi Data

Visualisasi data adalah tampilan berupa grafis atau visual dari informasi dan data. Menurut (Perdana Arkan, 2021) Visualisasi data mengubah kumpulan data menjadi hal lebih sederhana untuk ditampilkan. Dengan menggunakan elemen visual, pembaca akan lebih mudah memahami tren, *outliers*, dan pola dalam suatu data.

Visualisasi data didalam bisnis memungkinkan para pembuat keputusan untuk melihat analitik yang disajikan secara visual sehingga para penggerak dibidang bisnis dapat memahami konsep yang sulit atau mengidentifikasi pola baru.

Penyajian data dalam bentuk mentah atau berupa tulisan saja tentu akan mempersulit orang awam, terutama pimpinan perusahaan untuk mengetahui maksudnya. Oleh karenanya, pentingnya visualisasi data adalah agar pengambil kebijakan dapat memahami dengan baik data tersebut dan dapat mengambil keputusan yang tepat.

Menurut (Perdana Arkan, 2021) Terdapat lima tipe visualisasi data yang sering digunakan yaitu:

a. Temporal

Tipe temporal cocok untuk menunjukkan hasil dari rangkaian data yang berbentuk linear, atau satu dimensi. Ciri utama dari visualisasi tipe ini adalah berupa garis, yang bermulai dan berakhir di titik tertentu. Garis tersebut dapat berdiri sendiri, atau bersinggungan dengan garis lainnya.

Contoh visualisasi data seperti ini adalah diagram *scatterplot*, *timeline*, *data time series*, dan diagram garis

b. *Hierarki*

Tipe hierarki umum digunakan untuk menunjukkan hubungan antara suatu kelompok terhadap kelompok lain yang lebih besar. Visualisasi hierarki cocok untuk menunjukkan munculnya data-data baru yang berasal dari suatu penyebab. Salah satu penggunaan yang umum adalah diagram pohon.

c. *Network*

Pada dasarnya, sekumpulan data atau *dataset* bisa saling berpengaruh satu sama lain. Penggunaan *network* pada *data visualization* adalah untuk mempermudah hubungan antar *dataset*. Salah satu contoh yang paling sering digunakan adalah word cloud dan node-link diagram.

d. Multidimensi

Visualisasi multidimensi umum digunakan untuk memvisualisasikan data yang memiliki banyak variabel atau dimensi.

Karena banyaknya *dataset* yang ditampilkan, visualisasi ini akan lebih menarik dan mencolok agar lebih mudah dipahami. Salah satu contoh visualisasi ini adalah *histogram*, *pie chart*, dan *stacked-bar*.

e. *Geospatial*

Visualisasi *geospatial* merepresentasikan wujud nyata dari suatu benda atau ruang yang memiliki data untuk ditampilkan. Visualisasi *geospasial* sering kali digunakan untuk menunjukkan penetrasi penjualan di suatu wilayah, peta pertumbuhan bisnis, hingga menunjukkan *flow* pada suatu *UX*. Contoh visualisasi *geospasial* yang paling sering digunakan adalah *heatmap* dan *kartograf*.

5. Aturan Pembuatan Grafik Analisis

Ketika membuat sebuah grafik, data yang ditunjukkan harus jelas, menarik, dan mudah untuk dimengerti. Berikut adalah 12 aturan yang digunakan dalam membuat grafik analisis (Meilach, 1986).

a. Simplicity

Membatasi elemen yang digunakan dan fokus pada tren.

b. Emphasis

Teknik untuk membuat penekanan pada suatu *chart* yaitu dengan membuat 1 elemen lebih menonjol dari elemen lainnya dengan memberikan warna yang lebih cerah, memberikan *outlines* yang berbeda serta membuat 1 elemen seperti bergerak keluar dari elemen lainnya.

c. Unity

Unity adalah hubungan antar elemen yang membuat beberapa elemen berfungsi secara keseluruhan dibandingkan secara terpisah.

d. Balance

Penempatan elemen terbagi menjadi 2 yaitu formal atau informal *balance*. Formal *balance* paling sering digunakan dalam bagan, tetapi informal *balance* dapat menjadi solusi desain ketika elemen bagan menciptakan ketidak seimbangan.

e. Spacing

Jarak antar elemen sama. Bar dan kolom harus lebih lebar dibandingkan dengan jarak diantara mereka.

f. Scale

Mengelola data sehingga akan tampak proporsional jika disajikan dalam bentuk gambar, dengan menggunakan *grid* pada sumbu horizontal dan vertikal. Pada diagram kurva, jika memilih proporsi yang buruk sumbu x (horizontal), dapat mempengaruhi pembaca dalam membaca data yang disajikan. contoh ketika menampilkan data penjualan, akan terlihat sangat buruk hasilnya jika ditampilkan grafik pada sumbu x yang pendek daripada sumbu x yang panjang.

g. Shade and color

Memberikan gradasi atau *pattern* pada bagan yang berurutan dari warna yang gelap ke warna yang cerah dengan begitu elemen – elemen akan

tampak bersebelahan. Hindari penggunaan warna merah pada bagan, karena warna merah melambangkan kegagalan bisnis.

h. Texture and Pattern

Menghindari penggunaan pola yang mencolok pada bar dan memberikan 2 pola yang sangat berbeda pada bar karena tidak enak untuk dilihat serta dapat mengakibatkan salah pemahaman pada data yang disampaikan. Dianjurkan menggunakan pola yang mirip pada bar karena lebih enak dilihat.

i. Grid Line Use

Grid line digunakan pada latar bagan untuk membantu mata dalam melihat data yang disajikan. Penggunaan *grid line* yang berlebihan dapat mempersulit pembaca dalam memahami data.

j. Line Thickness

Grafik garis tidak dapat menggunakan garis yang ketebalannya sama. Garis tren harus menggunakan garis yang tebal, sedangkan pada sumbu x dan y menggunakan garis yang memiliki ketebalan medium dan garis yang digunakan pada *grid* yaitu garis yang tipis. Ketika memiliki garis tren yang lebih dari satu, dapat menggunakan garis yang mempunyai ketebalan, warna atau tekstur yang berbeda. Urutan wana yang digunakan adalah dari warna yang terang ke warna yang gelap. Untuk garis yang menunjukkan sumbu x dan y dapat menggunakan warna - warna medium dan untuk garis *grid* menggunakan warna yang gelap atau dapat dikondisikan kembali apabila memiliki latar dengan warna yang gelap. Menggunakan garis yang solid apabila menunjukkan tren dan menggunakan garis putus - putus apabila menunjukkan data sekunder. Apabila data yang disajikan tidak tersedia, agar menggunakan garis titik - titik.

k. Data and Tic Mark Placement

Biasanya skala pada bagan berada di sisi kiri sumbu y. tapi ada kondisi dimana skala dapat di letakkan si sebelah kiri. Penyebabnya adalah

apabila posisi bar lebih cenderung ke arah kanan. Pemberian skala ini berguna untuk mempermudah dalam memahami data yang disajikan.

l. Placement of Number and Labels

Angka dan huruf yang digunakan harus mempunyai ukuran yang cukup besar untuk dapat dibaca dengan mudah. Label harus ditempatkan secara horizontal daripada vertikal sehingga dapat dibaca dengan mudah. Label pada bagan biasanya dimulai dengan angka 0 pada sudut sumbu x dan y. Jika penomoran tidak dimulai dari 0, pastikan angka yang digunakan memiliki ukuran yang besar dan jelas sehingga grafik tidak salah dibaca.

6. Grafik

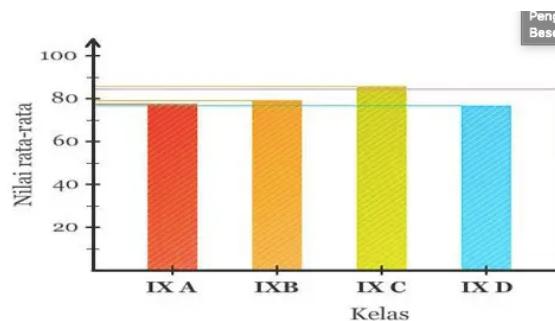
Grafik merupakan data yang ditampilkan dari sumber data yang disajikan ke dalam bentuk gambar dengan tujuan untuk memberikan informasi suatu data. Menurut (Ibeng Parta,2021) Grafik ini juga bisa diartikan ialah sebagai suatu kerangka atau juga gambar yang digunakan untuk membuat objek visualisasi dari data-data pada tabel dengan tujuan untuk memberikan informasi mengenai suatu data dari penyaji materi kepada penerima materi. Grafik ini juga dapat dibedakan itu dalam 3 jenis, yakni sebagai berikut:

a. Grafik Batang

Grafik batang merupakan jenis grafik yang menggambarkan beberapa data itu ke dalam bentuk batang. Grafik batang atau histogram ini merupakan jenis grafik yang paling sederhana serta juga sangat mudah untuk dapat dipahami. Grafik batang ini digunakan untuk dapat memperlihatkan perbedaan dengan jelas tentang tingkat nilai dari beberapa aspek pada sebuah data.

Panjang batang pada grafik batang ini menggambarkan presentase atau juga banyaknya data, sedangkan lebar batang ini berukuran sama semuanya. Pada umumnya data yang digunakan untuk perbandingan yakni tidak banyak, maksimal 8 data. Pada grafik batang, apabila data yang digunakan lebih dari satu maka untuk memperjelas data yang satu

dengan data yang lain ialah dengan memberi warna yang berbeda-beda pada tiap batang. Bentuk diagram batang dapat dilihat pada gambar 2.2.

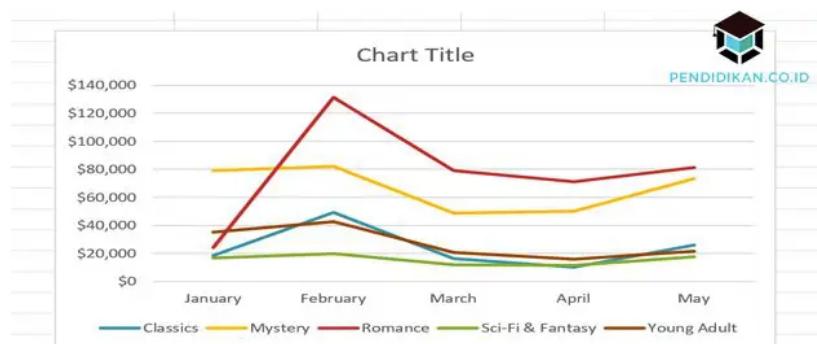


Gambar 2.2 Grafik batang (Ibeng Parta,2021)

b. Grafik Garis

Grafik garis ini merupakan jenis grafik yang menggambarkan beberapa data dalam bentuk garis atau kurva. Grafik garis ini digunakan untuk dapat menggambarkan suatu perkembangan atau juga perubahan data dari waktu ke waktu. Grafik garis ini mempunyai 2 sumbu utama yakni sumbu X serta sumbu Y.

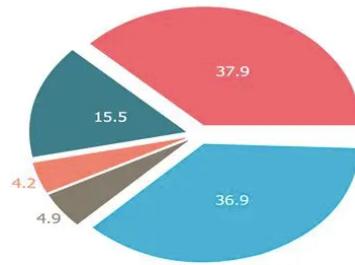
Sumbu X ini digunakan untuk dapat menunjukkan waktu pengamatan. Sedangkan untuk sumbu Y digunakan untuk dapat menunjukkan nilai dari hasil pengamatan pada waktu-waktu tertentu. Waktu dan hasil pengamatan dibuat dengan titik-titik pada bidang XY. Kemudian dari masing-masing titik yang berdekatan itu dihubungkan dengan garis, sehingga akan menghasilkan grafik garis. Bentuk Grafik garis dapat dilihat pada gambar 2.3.



Gambar 2.3 Grafik Garis (Ibeng Parta,2021)

c. Grafik Lingkaran

Grafik lingkaran ini merupakan jenis grafik yang menggambarkan beberapa data di dalam bentuk lingkaran. Grafik lingkaran tersebut digunakan untuk dapat menggambarkan persentase nilai total dari suatu data. Bentuk Grafik lingkaran dapat dilihat pada gambar 2.4.

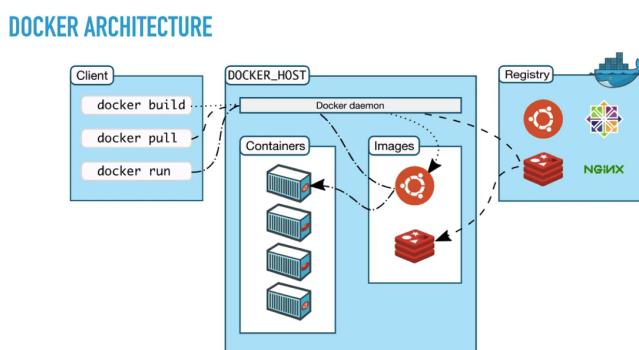


Gambar 2.4 Grafik Lingkaran (Ibeng Parta,2021)

2.4 Perangkat Lunak

1. Docker

Docker adalah *container management service*. Tujuan utamanya adalah untuk mempermudah *developer* atau *sysadmin* untuk membangun, mengemas, dan menjalankan aplikasi ke dalam sebuah *container* yang kemudian dapat digunakan dimana saja. *Container* tersebut berfungsi sebagai virtualisasi sebuah *Operating System* (OS) beserta komponen-komponen yang dibutuhkan dalam menjalankan aplikasi. Arsitektur *Docker* terdiri dari 3 bagian utama yaitu *Docker Client*, *Docker Server*, dan *Docker Registry* (Docker,2021). Arsitektur *Docker* dapat dilihat pada gambar 2.5.



Gambar 2.5 Arsitektur Docker (Docker,2021)

Docker menggunakan arsitektur *client-server*. *Docker Client* berbicara dengan *Docker Daemon*, yang melakukan tugas berat dalam membangun, menjalankan, dan mendistribusikan *container docker*. *Docker Client* dan *Docker Daemon*, dapat berjalan pada sistem yang sama, atau dapat menghubungkan *Docker Client* ke *daemon docker* jarak jauh. *Docker Client* dan *Docker Daemon*, berkomunikasi menggunakan *REST API*, melalui soket *UNIX* atau antarmuka jaringan. *Docker Client* lainnya adalah *Docker Compose*, yang memungkinkan bekerja dengan aplikasi yang terdiri dari sekumpulan *container*. Berikut bagian utama dari arsitektur docker (Docker,2021).

a. *Docker Client*

Docker client adalah sebuah aplikasi terminal yang berisikan kumpulan perintah untuk menjalani *docker*. Fungsinya adalah sebagai perantara untuk menghubungkan pengguna dengan *docker server*.

b. *Docker Daemon*

Docker daemon atau *docker server* adalah sebuah *service* yang dijalankan pada *host* dalam *Operating System (OS)* dimana *docker* itu diinstal. Fungsinya adalah membangun, mendistribusikan, dan menjalankan *docker container*. Berikut bagian dari *docker daemon*:

1) *Images*

Images adalah sebuah template yang bersifat *read only*. Template ini adalah sebuah *operating system* yang telah diinstal dengan berbagai *dependency library* yang dibutuhkan untuk menjalankan aplikasi. Fungsi dari *images* adalah membuat *container*, dengan satu *image* dapat dibuat menjadi beberapa *container*.

2) *Container*

Containers adalah *instances images docker* yang dapat dijalankan menggunakan perintah *run docker*, dimana tujuan dasar *docker* adalah menjalankan sebuah *container*.

c. *Docker Registry*

Docker registry adalah kumpulan *docker image* yang bersifat *private* maupun *public* yang dapat diakses di *docker.hub*. Dimana pengguna dapat *push* atau *pull image* yang dimiliki sendiri atau menggunakan *image* yang dibuat oleh pengguna lain yang di buka ke *public*.

2. *Google Cloud Platform*

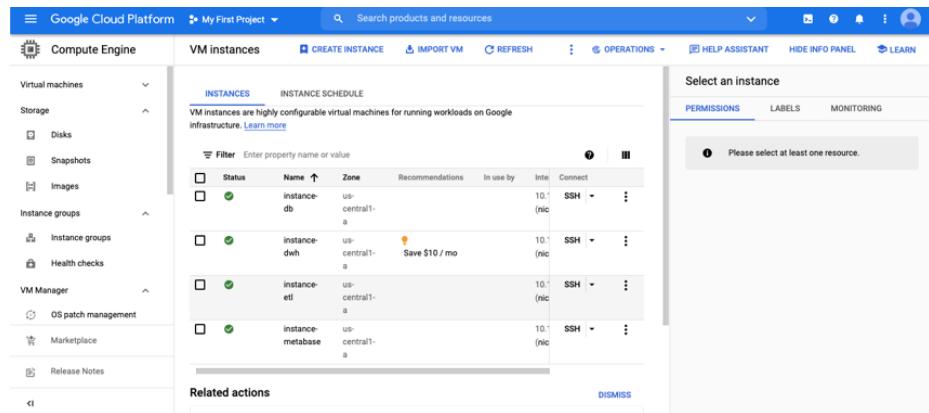
Google Cloud Platform atau GCP adalah kumpulan layanan komputasi awan yang ditawarkan oleh *Google*. GCP berjalan di atas infrastruktur yang sama yang digunakan oleh *Google* untuk produk internalnya, seperti *Google Search, YouTube* dan *Gmail*. Bersamaan dengan seperangkat alat manajemen, GCP menyediakan serangkaian layanan *cloud* modular termasuk komputasi, penyimpanan data, analisis data dan *machine learning*. Registrasi membutuhkan detail kartu kredit atau rekening bank (andre oliver, 2021).

Menurut (andre oliver, 2021) GCP menyediakan berbagai layanan komputasi untuk keperluan *development*. Namun, setelah melihat pertumbuhan di pasar teknologi, aplikasi ciptaan *Google* ini menambahkan bermacam-macam fitur baru. Mulai dari *cost management* hingga *security development tools*, layanan yang ditawarkan oleh *Google Cloud Platform* adalah seperti *Google Compute Engine, Google App Engine, Google Cloud Storage* dan *Google Container Engine*. *Google Cloud Platform* merupakan sistem *cloud computing* yang menawarkan banyak manfaat. Selain untuk keperluan *development*, aplikasi ini juga sering digunakan untuk kebutuhan manajemen *database*. *Google Cloud Platform* juga memiliki berbagai fungsi lainnya. manfaat-manfaat lain yang ditawarkan adalah seperti berikut:

- a. Harga cenderung lebih murah dari kompetitor.
- b. *Global fiber network* yang privat.
- c. Migrasi langsung dari *virtual machine*.
- d. Mampu meningkatkan kinerja *development*.
- e. Keamanan sistem yang mumpuni.

f. Layanan yang didedikasikan untuk ekspansi berkelanjutan.

Gambar tampilan *console Google Cloud Platform* dapat dilihat pada gambar 2.6.



Gambar 2.6 Tampilan Antarmuka *Google Cloud Platform*

3. MySQL

MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi *MySQL* adalah *database server* yang gratis dengan lisensi *GNU General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

MySQL adalah *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server* (Yasin K,2019).

4. PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional *open source* yang berjalan pada *platform Linux* dan berfungsi dengan objek sebagai komponen relasional dalam sistem manajemen basis data. *PostgreSQL* dikembangkan dan dikelola oleh *PostgreSQL Global Development Group*, yang merupakan grup pengembang *PostgreSQL*.

PostgreSQL mencoba menyesuaikan dengan standar *SQL* di mana kesesuaian tersebut tidak bertentangan dengan fitur tradisional atau dapat

menyebabkan keputusan arsitektur yang buruk. Banyak fitur yang diperlukan didukung oleh standar *SQL* meskipun terkadang dengan *syntask* atau fungsi yang sedikit berbeda. Pada rilis versi 14 di bulan September 2021, *PostgreSQL* memenuhi setidaknya 170 dari 179 fitur wajib untuk *SQL*.

Menurut (Ardian Dwi Praba,2020) Performansi *respon time PostgreSQL* lebih baik dibandingkan dengan *Mysql*. Dengan demikian maka dalam pembuatan *database* dalam skala besar dan jumlah *record* yang banyak, *PostgreSQL* bisa menjadi solusi yang lebih baik.

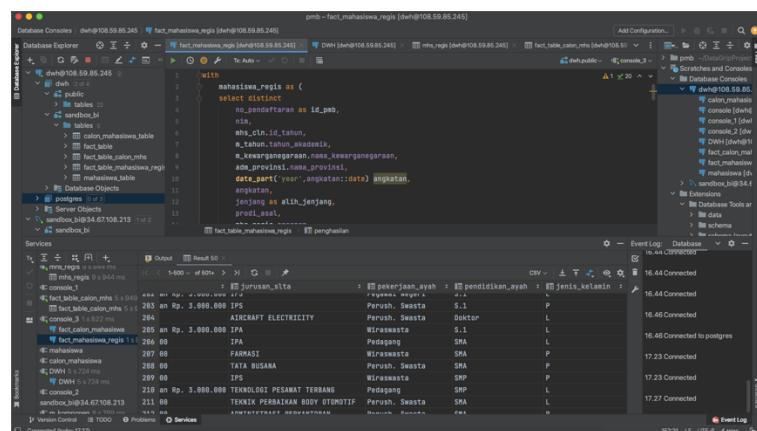
Menurut (Eril,2020) beberapa Kelebihan *PostgreSQL* sebagai sistem manajemen basis data adalah bersifat *open source*, aman karena dikendalikan oleh beberapa perusahaan, termasuk *cybertech* yang berarti tidak dapat dengan mudah dibeli atau diambil alih oleh entitas manapun tanpa persetujuan dari pihak yang mengendalikan serta *PostgreSQL* memiliki fitur bawaan yang mendukung berbagai alat eksternal yang dibutuhkan *DBA* untuk membuat sistem di berjalan dengan lancar dan efisien, *tools* yang disediakan kuat serta *postreSQL* bisa direplikasi. tanpa biaya tambahan dan lisensi.

5. *Python*

Python adalah bahasa pemrograman interpretatif yang bisa dipasang pada berbagai platform, khususnya platform yang berfokus pada keterbacaan kode. *Data science*, *internet of things (IoT)*, dan *machine learning* merupakan beberapa hal yang berkaitan langsung dengan *Python*. *programmer* biasa menggunakan *Python* untuk membuat *prototype*, *scripting* guna mengelola infrastruktur, maupun pembuatan *website* dalam skala besar. Sebuah penelitian yang diterbitkan dalam jurnal *Developer Economics – State of the Developer Nation* mengungkapkan, sudah 69% pengembang *machine learning* dan *data scientist* aktif memakai *Python* pada tahun 2018. Bahkan, laporan *IEEE Spectrum* tahun 2019 menyatakan bahasa pemrograman *Python* menjadi bahasa pemrograman paling populer di dunia (Tandika Bima,2020).

6. Datagrip

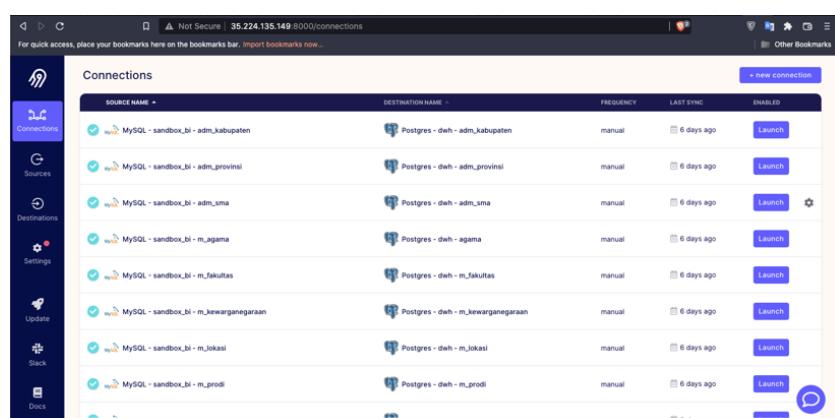
Datagrip adalah *database management tools* buatan dari perusahaan *Jetbrains*, *Datagrip* memang tidaklah gratis tetapi fitur yang ditawarkan sangat sepadan. *Datagrip* juga dilengkapi fitur yang sangat bermanfaat seperti *export-import database*, *Query beautify* (mempercantik *query*), *code checking*, *code generation*, dan lain-lain. *Datagrip* tidak hanya mendukung *database* bertipe *SQL* tetapi *database NOSQL* juga bisa di *remote* dengan *Datagrip* (JetBrains s.r.o,2021). Tampilan antarmuka *Datagrip* dapat dilihat pada gambar 2.7.



Gambar 2.7 Tampilan Software *Datagrip*

7. Airbyte

Airbyte merupakan sebuah platform khusus integrasi data.



Gambar 2.8 Tampilan antarmuka *Airbyte*

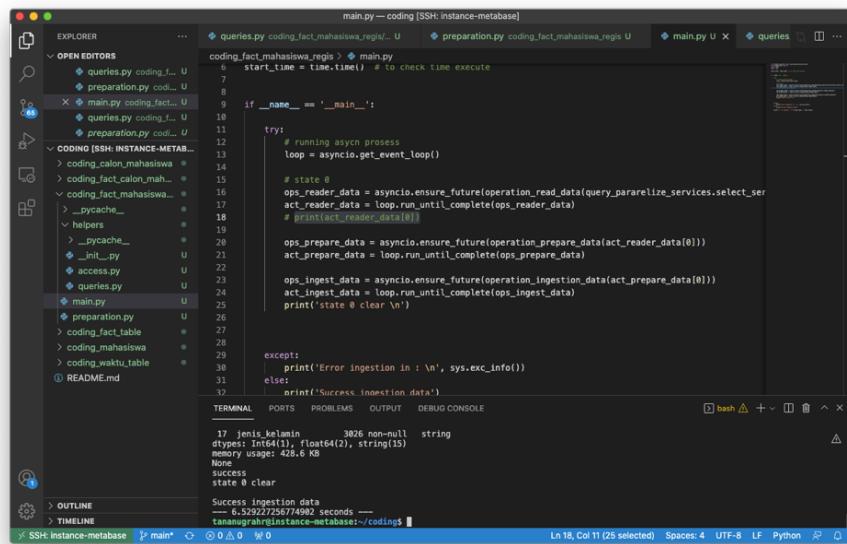
Menurut (Almadi W,2021) *Airbyte* merupakan standar *open source* baru untuk menyinkronkan data dari aplikasi, *API & database* ke *data warehouse*, data lakes dan destinasi lainnya. *Airbyte* sedang dalam misi untuk menjadikan jaringan pipa integrasi data sebagai komoditas. Tampilan antarmuka *Airbyte* dapat dilihat pada gambar 2.8

Fitur yang ditawarkan oleh *Airbyte* sebagai *open source* untuk data *integration* dan data *pipeline* adalah sebagai berikut:

- a. *Built for extensibility*, Menyesuaikan konektor yang ada dengan kebutuhan.
- b. *Optional normalized schema*, sepenuhnya dapat disesuaikan, mulai dengan data mentah atau dari beberapa saran data yang dinormalisasi.
- c. *Full-grade scheduler*, automasi replikasi data dengan frekuensi yang dibutuhkan.
- d. *Real-time monitoring*, *Airbyte* mencatat semua kesalahan dengan detail lengkap.
- e. *Incremental updates*, replikasi otomatis didasarkan pada pembaruan tambahan untuk mengurangi biaya transfer data.
- f. *Manual full refresh*, terkadang masih perlu menyinkronkan ulang semua data untuk memulai lagi.
- g. *Debugging autonomy*, ubah dan *debug pipeline* sesuai keinginan

8. *Visual Studio Code*

Visual Studio Code adalah alat *source code editor* yang ringan namun kuat yang berjalan di desktop tersedia untuk multiplatform seperti *Windows*, *macOS*, dan *Linux*. Muncul dengan dukungan bawaan untuk *JavaScript*, *TypeScript* dan *Node.js* dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain seperti *C++*, *C#*, *Java*, *Python*, *PHP*, *Go* dan *runtime* seperti *.NET* dan *Unity* (Microsoft,2021). Berikut tampilan antarmuka *Visual Studio Code* dapat dilihat pada gambar 2.9.



d. Code Debugging

salah satu fitur terkeren yang ditawarkan *Visual Studio Code* adalah dapat melakukan *debug* pada kode dengan cara mengawasi kode, variabel, *call stack* dan *expression* yang mana saja.

e. Source control

Visual Studio Code memiliki *integrated source control* termasuk *Git support in-the-box* dan penyedia *source code control* lainnya di pasaran.

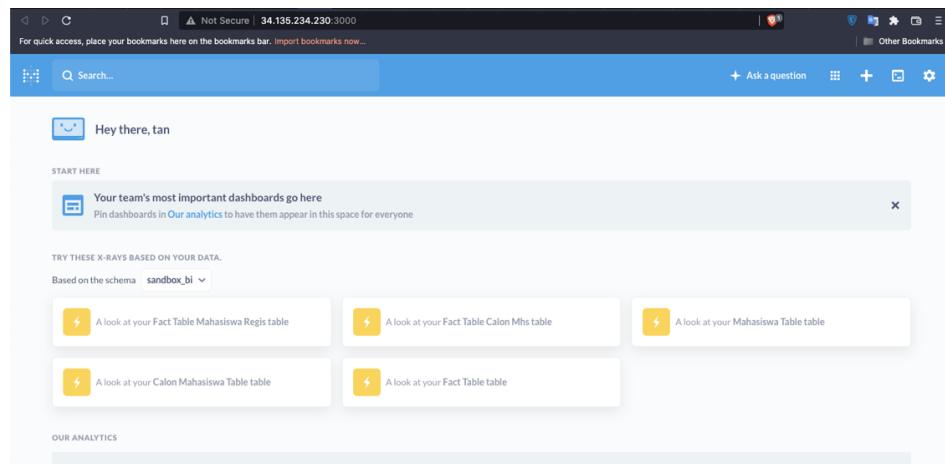
f. Integrated terminal

Tiada lagi *multiple windows* dan *alt-tabs*. *Visual Studio Code* dapat melakukan *command-line task* sekejap dan membuat banyak terminal di dalam editor.

9. Metabase

Metabase adalah sebuah *software* atau *tool* yang sering digunakan untuk keperluan *Business Intelligence*. Menurut (Korocz, T. 2020), *Metabase* merupakan *tool* yang bisa memvisualisasikan data lewat grafik sehingga proses analisis lebih mudah. Tidak seperti *tool* *Business Intelligence* lainnya, *Metabase* tidak selalu memerlukan kemampuan di bidang *SQL* untuk digunakan. Oleh karena itu, cara menggunakan *Metabase* dianggap mudah oleh orang-orang yang memanfaatkan *tool* ini sehari-hari. *Metabase* lebih mudah dipakai dibanding *tool* serupa lainnya seperti *Tableau* atau *Power Bussines Intellegence*. *Metabase* diperlukan untuk mengolah data dan mendapatkan *insight* tentang bisnis yang dimiliki.

Biasanya, sumber data untuk diolah sebuah perusahaan tidak hanya berasal dari satu tempat, tetapi bermacam-macam. Oleh karena itu, dibutuhkan data *warehouse* yang bisa menyimpan dan menyajikan data lebih mudah. *Metabase* adalah *tool* yang mampu mengekstraksi data dari data *warehouse* untuk menciptakan sebuah laporan yang dibutuhkan (Korocz, T. 2020). Tampilan antarmuka *Metabase* dapat dilihat pada gambar 2.10.



Gambar 2.10 Tampilan antarmuka *metabase*

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan melakukan evaluasi permasalahan-permasalahan, kesempatan, hambatan yang terjadi pada sistem dimana aplikasi dibangun yang meliputi perangkat keras dan perangkat lunak.

1. Kebutuhan perangkat keras.

Spesifikasi kebutuhan perangkat keras yang digunakan dalam membangun implementasi *dashboard* visualisasi penerimaan mahasiswa baru Universitas Janabadra adalah;

- a. *Processor intel core i5*.
- b. *Random Access Memory (RAM) 8 GB*.
- c. Piranti masukan berupa *mouse* dan *keyboard*.
- d. Piranti keluaran berupa monitor 13,3” dan *printer*.
- e. Media penyimpanan *SSD* dengan kapasitas 128 *GB*.

2. Kebutuhan perangkat lunak.

Adapun kebutuhan perangkat lunak yang digunakan dalam membangun implementasi *dashboard* visualisasi penerimaan mahasiswa baru Universitas Janabadra. Berikut beberapa perangkat lunak yang digunakan dalam proses pembangunannya:

- a. *Browser*
- b. *Google Cloud Platform Account*
- c. *Visual Studio Code*
- d. *Datagrip*
- e. *Docker*
- f. *Airbyte*
- g. *Python*
- h. *Metabase*

3.2 Perancangan Sistem

Perancangan sistem digunakan untuk memberikan gambaran kepada pengguna mengenai sistem yang akan dibuat. Perancangan implementasi *dashboard* visualisasi penerimaan mahasiswa baru Universitas Janabadra menggunakan metode *Business Intelligent Roadmap*. *Business Intelligence Roadmap* PMB Universitas Janabadra dapat dilihat pada tabel 3.1

Tabel 3.1 *Business Intelligent Roadmap* PMB Universitas Janabadra

Prosedur Pengerjaan			
	Input	Proses	Output
Justification	Latar belakang masalah yang dihadapi dalam memonitor penerimaan mahasiswa baru Universitas Janabadra	Bussiness Case Assesment Mengidentifikasi permasalahan yang ada serta menawarkan solusi yang diberikan	Memahami latar belakang masalah dan memberikan solusi untuk menyelesaikan masalah
Planning	1. Teori Bussiness Intelligent Roadmap 2. Teori penggunaan google cloud platform 3. pertanyaan terkait informasi penting yang ditampilkan pada dashboard	Bussiness Requirement Definition Mengidentifikasi kebutuhan pembuatan sistem yang berkaitan dengan proses bisnis dengan cara observasi dan wawancara	Hasil yang didapatkan dari observasi dan wawancara
Business Analysis	Hasil yang didapatkan dari observasi dan wawancara	Analisa Kebutuhan Bisnis Menganalisa informasi yang akan ditampilkan pada <i>dashboard</i> sesuai dengan hasil observasi maupun wawancara	Informasi - informasi penting yang akan ditampilkan pada <i>dashboard</i>
Design	Informasi - informasi penting yang akan ditampilkan pada <i>dashboard</i>	Rancangan Sistem Arsitektur <ul style="list-style-type: none"> Merancang Data Warehouse Menentukan teknologi yang akan dimanfaatkan dalam pembangunan sistem 	<ul style="list-style-type: none"> Rancangan Data Warehouse design Dashboard Teknologi yang dimanfaatkan
Construction	<ul style="list-style-type: none"> Rancangan Data Warehouse design Dashboard Teknologi yang dimanfaatkan 	Pembangunan Data Warehouse dan <i>dashboard</i> penerimaan mahasiswa baru	<ul style="list-style-type: none"> Data Warehouse Dashboard
Deployment	<ul style="list-style-type: none"> Data Warehouse Dashboard 	Pengujian Melakukan pengujian terhadap <i>dashboard</i> yang dibuat	Dokumentasi proses pengujian

Bussiness Intelligent Roadmap adalah panduan untuk mengembangkan aplikasi pendukung pengambilan keputusan *BI* menggunakan data terstruktur. (Sholiq et al., 2015). Berikut ini adalah tahapan – tahapan yang ada pada metode *Business Intelligent Roadmap* ini:

1. *Justification*

Tahapan ini berisi proses mengidentifikasi masalah yang ada berdasarkan dengan latar belakang serta memberikan solusi yang tepat terkait permasalahan yang ada. Latar belakang masalah yang dihadapi dalam memonitor penerimaan mahasiswa baru Universitas Janabadra yaitu adalah informasi yang dihasilkan belum dapat memenuhi sebagai alat penunjang untuk mendukung keputusan karena masih dalam bentuk data-data tabel yang berisi jumlah calon mahasiswa berdasarkan status penerimaan yaitu pendaftar dan registrasi.

Pelaporan masih dibuat sebatas teks atau angka-angka biasa sehingga kurang cepat untuk dimengerti dan diambil kesimpulannya. Sistem pelaporan yang kurang baik dan kurang menarik bisa menghambat dalam monitoring pencapaian target yang diinginkan

2. *Planning*

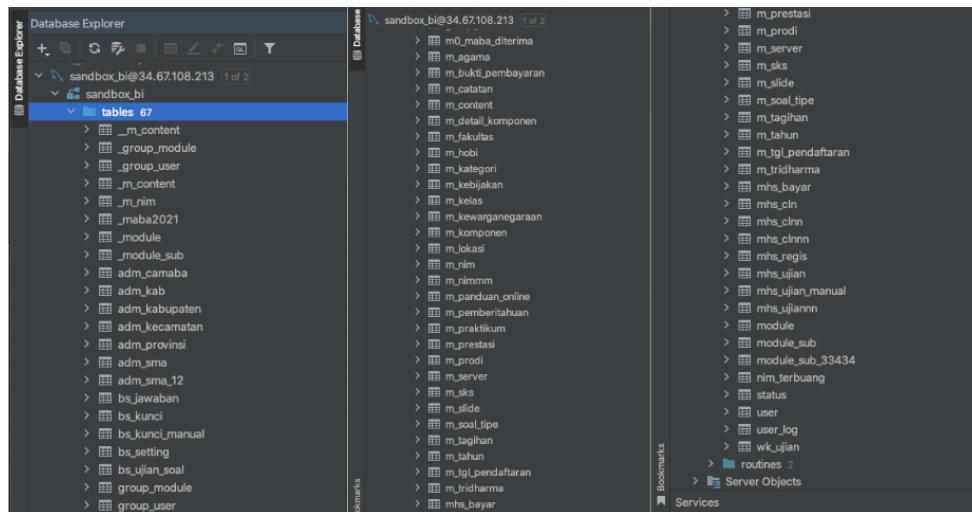
Tahap ini berisi proses mengidentifikasi kebutuhan data yang diperlukan dalam pembuatan sistem dengan cara melakukan wawancara dengan pihak struktural Universitas Janabadra. Hasil didapatkan adalah data penting yang harus ditampilkan pada sistem serta ketersediaan data yang diperlukan.

maka informasi yang didapat dari staf BAAK dan Admisi berupa :

- a. Jumlah calon mahasiswa baru yang mendaftar berdasarkan Program Studi dan Kelas (Reguler/Karyawan).
- b. Jumlah calon mahasiswa baru berdasarkan kelas tiap tahun masuk/angkatan.
- c. Jumlah calon mahasiswa baru yang diterima.
- d. Jumlah mahasiswa yang registrasi berdasarkan asal daerah asal sekolah per program studi.

- e. Jumlah mahasiswa baru berdasarkan kriteria data keluarga (penghasilan dan pendidikan orang tua)

Data PMB yang didapat dari data server PMB berupa 69 tabel Admisi, dapat dilihat pada gambar 3.1



Gambar 3.1 Database PMB

3. Business Analysis

Tahapan ini berisi proses menganalisa informasi yang akan ditampilkan pada *dashboard* sesuai dengan hasil observasi maupun wawancara. Informasi yang akan ditampilkan pada *Dashboard* penerimaan mahasiswa baru dapat dilihat pada tabel 3.2.

Tabel 3.2 Tabel Informasi yang akan ditampilkan

No	Informasi	Indikator
1	Jumlah Calon Mahasiswa PMB Universitas Janabadra	Lima tahun terakhir (2017-2021) <ul style="list-style-type: none"> • Program Studi • Tahun Angkatan • Program • Profile (Agama, Jenis Kelamin, jurusan slta, Kewarganegaraan,) • Kriteria Data Keluarga (Penghasilan orang tua, pendidikan orang tua, pekerjaan orang tua) • Pendapatan dari pendaftaran PMB
2	Jumlah Mahasiswa regis PMB Universitas Janabadra	Empat tahun terakhir (2018-2021) <ul style="list-style-type: none"> • Program Studi • Tahun Angkatan • Program

		<ul style="list-style-type: none"> • Profile (Agama, Jenis Kelamin, jurusan slta, Kewarganegaraan,) • Kriteria Data Keluarga (Penghasilan orang tua, pendidikan orang tua, pekerjaan orang tua)
3	Jumlah Total Calon Mahasiswa PMB	Lima tahun terakhir (2017-2021)
4	Jumlah Total Mahasiswa Regis PMB	Empat tahun terakhir (2018-2021)

4. Design

Tahapan ini berisi proses perencanaan pembangunan data *warehouse* PMB, teknologi yang dimanfaatkan dalam pembangunan data *warehouse* dan *dashboard* visualisasi.

a. Tabel yang digunakan untuk data warehouse.

Pada perancangan sistem data *warehouse* PMB Universitas Janabadra menggunakan *star schema*, dimana tabel fakta ditempatkan di tengah, dikelilingi oleh tabel-tabel dimensi. Tabel yang dibutuhkan untuk perancangan data *warehouse* penerimaan mahasiswa Universitas Janabadra yaitu:

1) Tabel mhs_cln

Fungsi : Berfungsi untuk menampung informasi tentang calon mahasiswa yang mendaftar di Universitas Janabadra

Nama Tabel : mhs_cln

Primary Key : id_pmb

Foreign Key : no_bayar, prodi1, sem_awal, id_lokasi

Jumlah Field : 59 field

Tabel mhs_cln dapat dilihat pada tabel 3.3.

Tabel 3.3 Tabel mhs_cln

No	Nama Kolom	Tipe Data	Size	Key	Keterangan
1	id_pmb	Varchar	10 byte	PK	Id pada tabel mhs_cln
2	nik	Varchar	50 byte		Data nik
3	tanggal_bayar	Varchar	15 byte		Tanggal bayar PMB
4	no_bayar	Varchar	50 byte	FK	no kode pembayaran pmb
5	sem_awal	Varchar	5 byte	FK	semester masuk PMB
6	Id_tahun	Int	5 byte		id tahun masuk
7	Id_lokasi	Varchar	5 byte	FK	id lokasi calon mahasiswa
8	tanggal	Varchar	15 byte		tanggal daftar
9	nama	Varchar	200 byte		nama calon mahasiswa
10	Jenis_kelamin	Varchar	5 byte		jenis kelamin calon mahasiswa
11	Kode_agama	Varchar	5 byte		kode agama calon mahasiswa
12	Prodi1	Int	5 byte	FK	kode program studi 1
13	Prodi2	Int	5 byte		kode program studi 2
14	program	Varchar	5 byte		program kelas
15	Alamat_asal	Text			alamat calon mahasiswa
16	Kec_asal	Varchar	5 byte		kode kecamatan
17	Kab_asal	Varchar	30 byte		kode kabupaten
18	Prop_asal	Varchar	30 byte		kode provinsi
19	warga	Varchar	5 byte		kewarganegaraan calon mahasiswa
20	Asal_slt	Varchar	200 byte		asal slda calon mahasiswa
21	Alamat_slt	Text			alamat slda calon mahasiswa
22	Kab_slt	Varchar	30 byte		kode kabupaten slda
23	Prop_slt	Varchar	30 byte		kode provinsi slda
24	telp	Varchar	100 byte		no telpon calon mahasiswa
25	email	Varchar	100 byte		alamat email calon mahasiswa
26	Nama_ortu	Varchar	100 byte		nama orang tua calon mahasiswa
27	pekerjaan	Varchar	200 byte		pekerjaan orang tua
28	Alamat_yogya	text			alamat mahasiswa di yogya
29	Kab_yogya	Varchar	30 byte		kode kabupaten
30	Prop_yogya	Varchar	30 byte		kode provinsi
31	Tempat	Varchar	50 byte		asal calon mahasiswa
32	Tanggal_lahir	Varchar	15 byte		tanggal lahir mahasiswa
33	Rekomendasi	Varchar	200 byte		nama pemberi rekomendasi
34	Nama_lengkap_info	Varchar	255 byte		nama lengkap mahasiswa

35	Fakultas_unit	Varchar	150 byte		kode fakultas
36	Nim_nidn	Varchar	100 byte		nim mahasiswa
37	Tanggal_ujian	Varchar	15 byte		tanggal ujian pmb
38	Waktu_ujian	Float			waktu ujian pmb
39	Start	Time			waktu mulai ujian pm
40	Finish	Time			waktu selesai ujian pmb
41	Sesi_ujian	Int	5 byte		kode sesi ujian
42	Lama	Double			waktu ujian pmb
43	Total_soal	int	5 byte		Jumlah soal pmb
44	Jum_soal	int	5 byte		Jumlah soal pmb
45	Nilai	double			nilai ujian pmb
46	Persentase	double			Persentase nilai pmb
47	Baru	tinyint			Kode pmb
48	Pil_tes	Varchar	10 byte		Pilihan tes pmb
49	Kode_soal	Varchar	255 byte		Kode soal pmb
50	Kode_prestasi	Varchar	11 byte		Kode prestasi calon mahasiswa
51	File_prestasi	Varchar	255 byte		file prestasi mahasiswa
52	Password	Varchar	255 byte		password pmb
53	Pass_biasa	Varchar	100 byte		password pmb
54	Trash	int	5 byte		-
55	D_time	datetime			-
56	Asal_perti	Varchar	200 byte		Asal perguruan tinggi calon mahasiswa
57	Alamat_perti	Text			Alamat pt calon mahasiswa
58	Kab_perti	Varchar	30 byte		Kode kabupaten pt
59	Prop_perti	Varchar	30 byte		Kode provinsi pt

2) Tabel m_tahun

Fungsi : Berfungsi untuk menampung informasi tahun akademik di Universitas Janabadra

Nama Tabel : m_tahun

Primary Key : id_tahun

Jumlah Field : 7 field

Tabel m_tahun dapat dilihat pada tabel 3.4.

Tabel 3.4 Tabel m_tahun

No	Nama_Kolom	Tipe_da_ta	Size	Key	Keterangan
1	Id_tahun	Int	3 byte	PK	id pada tabel tahun
2	Tahun1	Int	4 byte		tahun masuk semester 1
3	Tahun2	Int	4 byte		tahun masuk semester 2
4	Tahun_akademik	Varchar	15 byte		tahun akademik
5	Thun_sem	Varchar	11 byte		tahun akademik
6	semester	int	3 byte		informasi semester
7	aktif	tinyint	1 byte		-

3) Tabel adm_provinsi

Fungsi : Berfungsi untuk menampung informasi provinsi

Nama Tabel : adm_provinsi

Primary Key : id_prov

Jumlah Field : 4 field

Tampilan tabel adm_provinsi dapat dilihat pada tabel 3.5.

Tabel 3.5 Tabel adm_provinsi

No	Nama_Kolom	Tipe_data	Size	Key	Keterangan
1	Id_prov	Int	11 byte	PK	id pada tabel adm_prov
2	Kode_provinsi	Varchar	11 byte		kode provinsi
3	Nama_provinsi	Varchar	255 byte		nama provinsi
4	trash	Varchar	255 byte		-

4) Tabel mhs_regis

Fungsi : Berfungsi untuk menampung informasi mahasiswa registrasi

Nama Tabel : mhs_regis

Primary Key : no_pendaftaran

Jumlah Field : 83 field

Tampilan tabel mhs_regis dapat dilihat pada tabel 3.6.

Tabel 3.6 Tabel mhs_regis

No	Nama_Kolom	Tipe_data	Size	key	Keterangan
1	id_regis	int	11 byte	PK	id tabel mhs regis
2	No_pendaftaran	Varchar	20 byte		informasi no pendaftaran mahasiswa
3	Id_prodi	int	5 byte		id tabel prodi
4	Jalur_pmb	Int	5 byte		kode jalur PMB
5	program	Varchar	5 byte		program kelas
6	nik	Varchar	50 byte		informasi NIK mahasiswa
7	nim	Varchar	15 byte		informasi nim mahasiswa
8	angkatan	year			tahun angkatan mahasiswa
9	Nama_lengkap	Varchar	255 byte		nama lengkap mahasiswa
10	Tempat_lahir	Varchar	100 byte		tempat lahir mahasiswa
11	Tgl_lahir	Date			tanggal lahir mahasiswa
12	Jenis_kelamin	Varchar	5 byte		jenis kelamin mahasiswa
13	Agama	Int	5 byte		kode agama mahasiswa
14	Status	int	5 byte		kode status aktif
15	Warga	Varchar	5 byte		kode kewarganegaraan
16	Alamat_asal	text			alamat asal mahasiswa
17	Kec_asal	Varchar	5 byte		kode kecamatan
18	Kab_asal	Varchar	5 byte		kode kabupaten
19	Prov_asal	Varchar	5 byte		kode provinsi
20	Rt_asal	Varchar	5 byte		asal rt
21	Rw_asal	Varchar	5 byte		asal rw
22	Kode_pos_asal	Varchar	10 byte		kode pos asal
23	Tlp_asal	Varchar	20 byte		no telpon rumah
24	Alamat_yogya	text			alamat di yogyakarta
25	Rt_yogya	Varchar	5 byte		rt yogya
26	Rw_yogya	Varchar	5 byte		rw yogya
27	Kode_pos_yogya	Varchar	5 byte		kode pos yogya
28	Tlp_yogya	Text			telepon di yogyakarta
29	Pekerjaan	Varchar	100 byte		informasi pekerjaan mahasiswa
30	Nama_instansi	Varchar	255 byte		nama instansi
31	Alamat_instansi	Varchar	255 byte		alamat instansi
32	Kode_pos_instansi	Varchar	10 byte		kode pos instansi
33	Tlp_instansi	Varchar	20 byte		no telepon instansi orang tua
34	Ijazah_slta	Varchar	10 byte		no ijazah
35	Jurusu_slta	Varchar	150 byte		jurusu slta mahasiswa

36	No_ijazah	Varchar	50 byte		no ijazah mahasiswa
37	Tahun_ijazah	Varchar	4 byte		tahun keluar ijazah
38	Kab_ijazah	Varchar	10 byte		kode kabupaten
39	Prov_ijazah	Varchar	10 byte		kode kabupaten
40	Alamat_slt	text			alamat slta
41	Kode_pos_ijazah	Varchar	10 byte		kode pos slta
42	Nama_ayah	Varchar	150 byte		nama ayah
43	Pekerjaan_ayah	Varchar	100 byte		pekerjaan ayah
44	Pendidikan_ayah	Varchar	150 byte		pendidikan ayah
45	Nama_ibu	Varchar	100 byte		nama ibu
46	Pekerjaan_ibu	Varchar	100 byte		pekerjaan ibu
47	Pendidikan_ibu	Varchar	10 byte		pendidikan ibu
48	Alamat_ortu	text			alamat orang tua
49	Kab_ortu	Varchar	5 byte		kode kabupaten orang tua
50	Prov_ortu	Varchar	5 byte		kode provinsi orang tua
51	Rt_ortu	Varchar	5 byte		rt orang tua
52	Rw_ortu	Varchar	5 byte		rw orang tua
53	Kode_pos_ortu	Varchar	5 byte		kode pos orang tua
54	Tlp_ortu	Varchar	20 byte		no telpon orang tua
55	Penghasilan	Varchar	150 byte		penghasilan orang tua
56	Hobi_1	Varchar	10 byte		kode hobi mahasiswa
57	Hobi_2	Varchar	10 byte		kode hobi mahasiswa
58	Hobi_3	Varchar	10 byte		kode hobi mahasiswa
59	Paskibra	Varchar	10 byte		mengikuti paskibra
60	Lkir	Varchar	10 byte		mengikuti Lkir
61	Osis	Varchar	10 byte		mengikuti Osis
62	Pertukaran	Varchar	10 byte		-
63	pmr	Varchar	10 byte		mengikuti pmr
64	Jambore	Varchar	10 byte		mengikuti jambore
65	pon	Varchar	10 byte		mengikuti pon
66	Prodi_asal	Varchar	150 byte		prodi asal perguruan tinggi
67	Nama_pt	Varchar	255 byte		nama perguruan tinggi asal
68	Jenjang	Varchar	10 byte		jenjang di perguruan tinggi asal
69	Nim_asal	Varchar	30 byte		nim di perguruan tinggi asal
70	Alamat_pt	Text			alamat pt asal
71	Kode_pos_pt	Varchar	10 byte		kode pos pt asal
72	Thun_awal_aktif	Varchar	10 byte		tahun masuk pt asal
73	Thun_akhir_aktif	Varchar	10 byte		tahun akhir pt asal
74	Acc	Int	5 byte		-

75	File_ijazah	Varchar	255 byte		upload file ijazah
76	File_skhun	Varchar	255 byte		upload file skhun
77	File_ktp	Varchar	255 byte		upload file ktp
78	File_akta	Varchar	255 byte		upload file akta
79	File_bukti_regis	Varchar	255 byte		upload file bukti regis
80	File_bebas_narkoba	Varchar	255 byte		upload file bebas narkoba
81	File_surat_pindah	Varchar	255 byte		upload file surat pindah
82	File_ket_kerja	Varchar	255 byte		upload file keterangan kerja
83	Tgl_regis	timestamp			tanggal registrasi

5) Tabel m_fakultas

Fungsi : Berfungsi untuk menampung informasi fakultas

Nama Tabel : m_fakultas

Primary Key : id_fakultas

Jumlah Field : 4 field

Tampilan tabel m_fakultas dapat dilihat pada tabel 3.7.

Tabel 3.7 Tabel m_fakultas

No	Nama Kolom	Tipe data	Size	Key	Keterangan
1	Id_fakultas	Int	5 byte	PK	Id tabel m_fakultas
2	fakultas	Varchar	50 byte		nama fakultas
3	aktif	tinyint	1 byte		Kode aktif fakultas
4	Sia_univ	tinyint	1 byte		-

6) Tabel m_prodi

Fungsi : Berfungsi untuk menampung informasi program studi di Universitas Janabadra

Nama Tabel : m_prodi

Primary Key : id_prodi

Jumlah Field : 13 field

Tabel m_fakultas dapat dilihat pada tabel 3.8.

Tabel 3.8 Tabel m_prodi

No	Nama_Kolom	Tipe_data	Size	Key	Keterangan
1	Id_prodi	int	5 byte	PK	Id tabel m_prodi
2	Id_fakultas	int	5 byte		Id fakultas
3	Kode_fakultas_sia	int	2 byte		Kode sia fakultas
4	Kd_pmb	int	2 byte		Kode pmb
5	prodi	Varchar	50 byte		Nama program studi
6	Aktif	Tinyint	1 byte		-
7	Sks_paket	tinyint	4 byte		Jumlah sks
8	Sia_univ_prd	Tinyint	4 byte		Kode sia program studi
9	Sia_univ_fak	Tinyint	4 byte		Kode sia fakultas
10	Sia_th_kuri	year	4 byte		Kode sia tahun kurikulum
11	Kode_fak_univ	int	3 byte		Kode fakultas universitas
12	Kode_jenjang	Varchar	5 byte		Kode jenjang
13	Kode_prg	Tinyint	4 byte		Kode prg

7) Tabel m_agama

Fungsi : Berfungsi untuk menampung informasi agama

Nama Tabel : m_agama

Primary Key : kode_agama

Jumlah Field : 2 field

Tabel m_agama dapat dilihat pada tabel 3.9.

Tabel 3.9 Tabel m_agama

No	Nama_Kolom	Tipe_data	Size	Key	Keterangan
1	Kode_agama	Varchar	1 byte	PK	Id tabel m_agama
2	Nama_agama	Varchar	20 byte		Informasi nama agama

8) Tabel m_kewarganegaraan

Fungsi : Berfungsi untuk menampung informasi kewarganegaraan

Nama Tabel : m_kewarganegaraan

Primary Key : kode_kewarganegaraan

Jumlah Field : 2 field

Tabel m_kewarganegaraan dapat dilihat pada tabel 3.10.

Tabel 3.10 Tabel m_kewarganegaraan

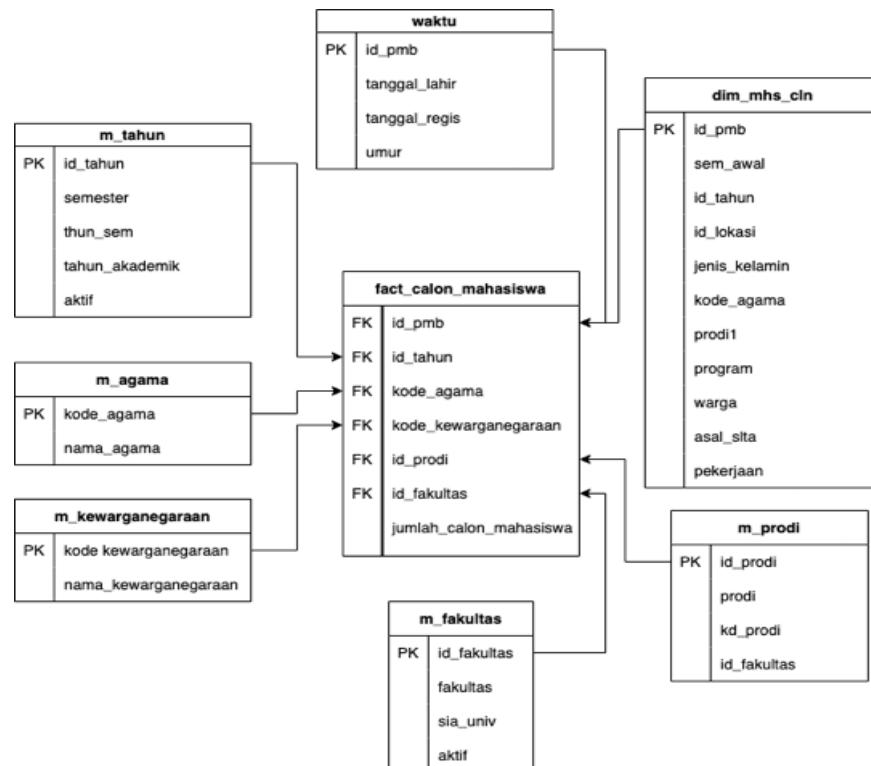
No	Nama Kolom	Tipe_data	Size	Key	Keterangan
1	Kode_kewarganegaraan	Varchar	1 byte	PK	Id tabel m_Kewarganegaraan
2	Nama_kewarganegaraan	Varchar	255 byte		Informasi nama kewarganegaraan

b. Design Data Warehouse

Berdasarkan tabel yang dipilih untuk membangun data *warehouse* pada proses pembangunan *dashboard* penerimaan mahasiswa baru Universitas Janabadra menggunakan *star schema*. *Design* skema yang dirancang terdiri dari dua skema, yaitu skema calon mahasiswa dan skema mahasiswa. berikut rancangan skema calon mahasiswa dan mahasiswa.

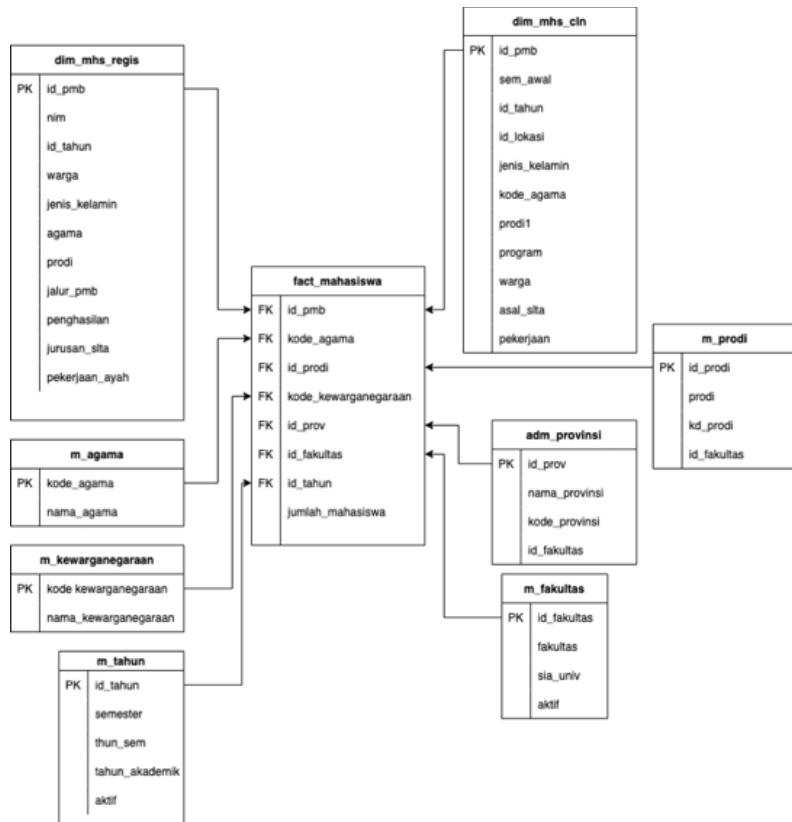
1. Star skema Calon Mahasiswa

Tabel fakta calon mahasiswa dikelilingi oleh 7 tabel dimensi yaitu tabel tahun, agama, mhs_calon, kewarganegaraan, waktu, prodi dan fakultas. Star skema calon mahasiswa dapat dilihat pada gambar 3.2.

Gambar 3.2 *Star schema* calon mahasiswa

2. Star skema Mahasiswa

Tabel fakta mahasiswa dikelilingi oleh 8 tabel dimensi yaitu tabel tahun, agama, mhs_regis, mhs_calon, adm_prov, kewarganegaraan, prodi dan fakultas. Rancangan skema mahasiswa dapat dilihat pada gambar 3.3.



Gambar 3.3 Star schema mahasiswa

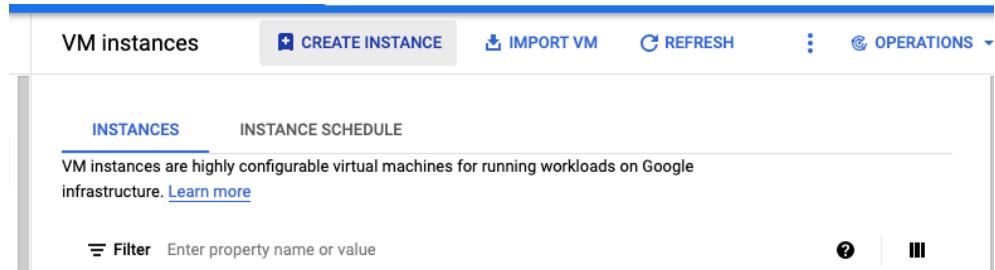
5. Construction

Tahapan ini berisi proses pembangunan data *warehouse* penerimaan mahasiswa baru Universitas Janabadra yang berdasarkan rancangan skema yang telah dibuat pada proses *design*. Sebelum membangun *datawarehouse* pada proses ini melakukan *setup* perangkat lunak yang akan digunakan untuk prosesnya adalah:

a. Setup Google Cloud Platform

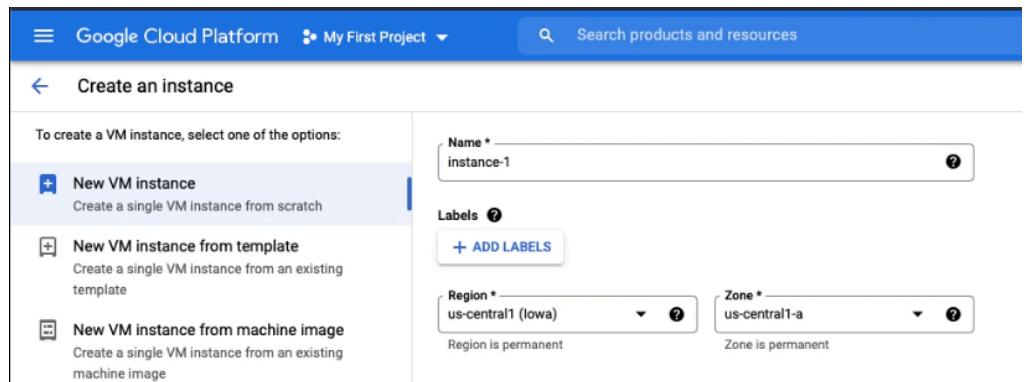
Pada Akun *Google Cloud Platform* yang sudah registrasi pilih *console* untuk masuk ke menu untuk memulai pengoperasian dan pilih *Compute*

Engine untuk menginstal *VM Engine* kemudian pilih *Create Instance* untuk membuat Virtual Machine. Dapat dilihat pada gambar 3.4



Gambar 3.4 *Create Instance*

Isi nama *Instance* yang ingin kita buat sesuai kebutuhan dan fungsinya dapat dilihat pada gambar 3.5.



Gambar 3.5 Nama *Instance*

Pada *Machine Configuration* pilih *Series* dan *Machine Type* yang diinginkan untuk *Instance* yang telah dibuat sesuai dengan yang dibutuhkan operasi yang akan kita jalankan. Dapat dilihat pada gambar 3.6.

Machine configuration

Machine family

[GENERAL-PURPOSE](#) [COMPUTE-OPTIMIZED](#) [MEMORY-OPTIMIZED](#) [GPU](#)

Machine types for common workloads, optimized for cost and flexibility

Series

E2

CPU platform selection based on availability

Machine type

e2-small (2 vCPU, 2 GB memory)



vCPU

1 shared core

Memory

2 GB

[▼ CPU PLATFORM AND GPU](#)

Display device

Enable to use screen capturing and recording tools.

Enable display device

Gambar 3.6 Machine Configuration

Pada *Public Image* Pilih *Operating Sistem* yang akan kita gunakan pada *Virtual Machine* yang akan dibangun dan ukuran penyimpanannya. Dapat dilihat pada gambar 3.7.

Boot disk

Select an image or snapshot to create a boot disk; or attach an existing disk. Can't find what you're looking for? Explore hundreds of VM solutions in [Marketplace](#)

[PUBLIC IMAGES](#)

[CUSTOM IMAGES](#)

[SNAPSHOTS](#)

[EXISTING DISKS](#)

Operating system

Ubuntu

Version *

Ubuntu 20.04 LTS

amd64 focal image built on 2021-12-12, supports Shielded VM features

Boot disk type *

Standard persistent disk

Size (GB) *

100

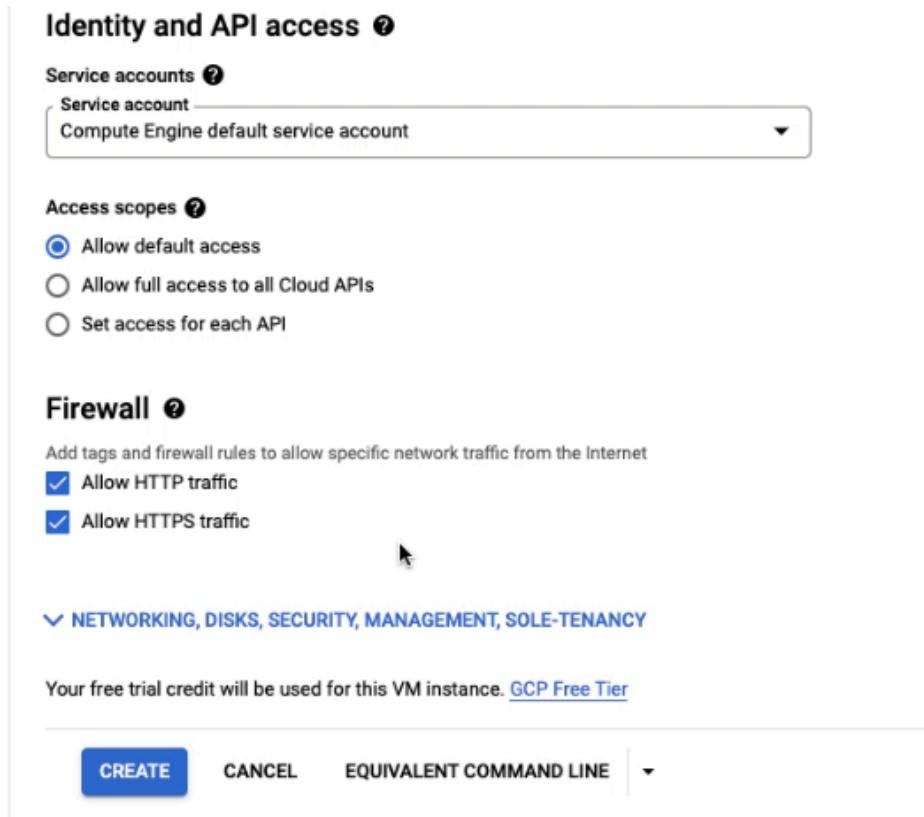
[▼ SHOW ADVANCED CONFIGURATION](#)

[SELECT](#)

[CANCEL](#)

Gambar 3.7 Public Image

Pada Identity dan API access *Set Firewall* dan *Allow HTTP traffic* dan *HTTPS traffic* untuk mendapatkan izin akses. Dan selanjutnya klick Create untuk membuat VM. Dapat dilihat pada gambar 3.8.



Gambar 3.8 *Identity and API access*

Pada proses setup *VMInstance* membuat empat *VM* yakni *Instance-db*, *Instance-dwh*, *Instance-etl* dan *Instance-metabase* yang prosesnya sama seperti diatas dengan konfigurasi *Instance* seperti pada tabel 3.11.

Tabel 3.11 Konfigurasi *Instance*

Nama	Machine Type	Size
Instance-db	E2-Small	20 GB
Instance-dwh	E2-Small	50 GB
Instance-etl	E2-Medium	80 GB
Instance-Metabase	E2-Small	100 GB

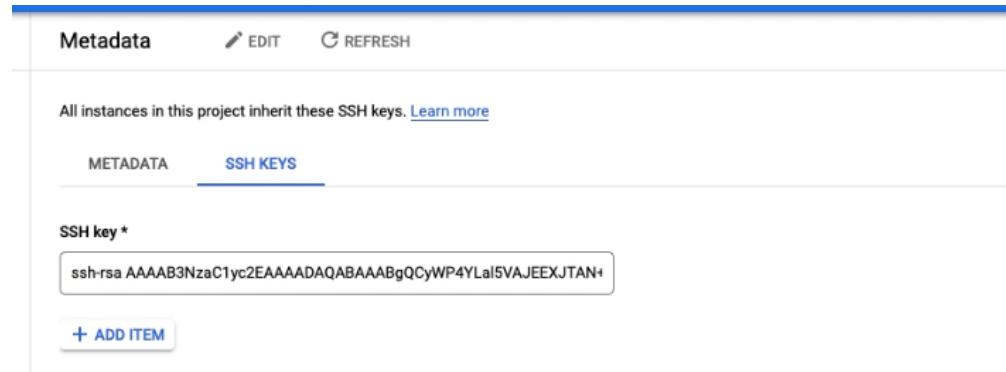
Set RSA Key, *RSA Key* adalah kunci pribadi berdasarkan algoritma *RSA*. *Private Key* digunakan untuk otentikasi dan pertukaran kunci simetris selama pembentukan sesi *SSL/TLS*. Untuk *RSA Key* dapat dilihat

menggunakan terminal dengan perintah `$ cat ~/.ssh/id_rsa.pub`. dapat dilihat seperti pada gambar 3.9

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[TAR:~ tanaanugrahr$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQCyWP4YLal5VAJEEXJTAN+ObeEF5bwcmDbqg8qkSze9
BM+vbwCjZqVRTtPdVaAzHZMX05dMUpaQKSDIqnBG8dLmfhnoiARSBn1L6HRpAE8pAXXMiLH4j57qefn
Kw8DjbANx4ULfnWB/U4Y5Tp603hSTpY/igWsjrbYrTjSMbLWa7dZpkUqz1iaYZZBSj4Q3iN7Q6tlkMdJ
dKKC9BpLHVFHij/zKfxqR/rNXpIp3xqLyU33W00bilUEOjv0008JCIXB0ZbzQk0MVxB0qVKUmJG1BKaP
6gJbhBf+ZsxlvCoDYmis4juDs4gBRsmGI0n6pBX20tZVdwCJU1xt6z6MjfCgHdf2rLiYENFDSELg0aP
lnHdm6Kj3pFWt5t5Nm7/Ah3TdM3e02Dzvc5lrisOHn+Wt7edb7x1JCckbjyOZfc2AYDkvDsf0y6CaM
PwdscLAoMvGFK1mOfJLWwX89+Cct1G2w+4yK8N35jmIYtmUUG/lzvHuW5NQ9Ip4IEfik= tanaanugrahr
rahr@TAR:~ tanaanugrahr$ ]
```

Gambar 3.9 Set RSA Key

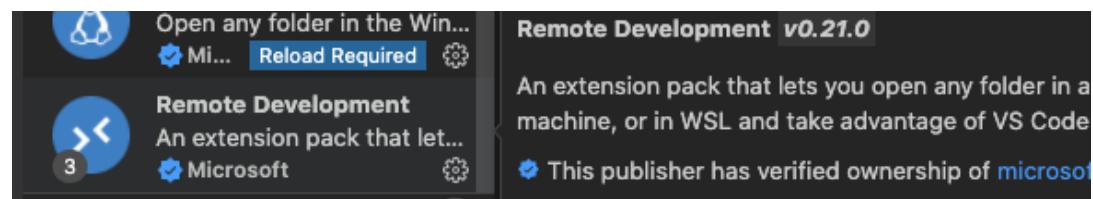
Masukkan *Key RSA* komputer ke dalam *Metadata* di halaman *Google Cloud Platform* seperti pada gambar 3.10



Gambar 3.10 Input Key RSA komputer ke dalam *Metadata*

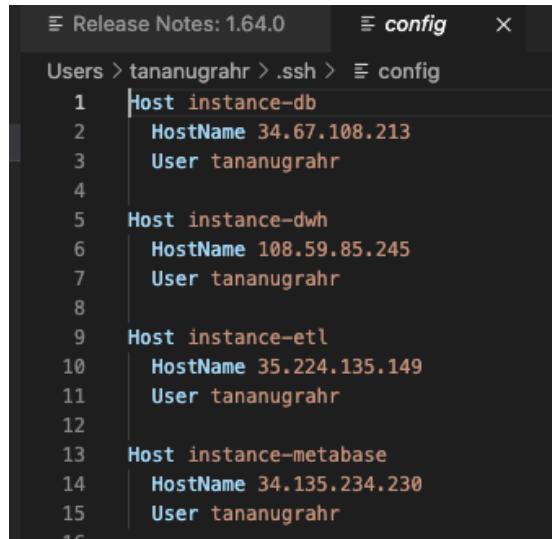
b. Set Up VSCode

Pada aplikasi *visual studio code install extension Remote Development* yang nantinya digunakan sebagai alat untuk remote server yang telah dibuat pada *Google Clod Platfrom*. Dapat dilihat pada gambar 3.11.



Gambar 3.11 Install Extension Remote Development

Setelah *Remote Developmen* dipasang pada *VSCode* pilih *SSH Config*. Kemudian masukkan nama *Host* yang telah dibuat pada *Google Cloud Platform*, *Hostname* yaitu alamat *IP Eksternal* pada *Instance* di *GCP* dan nama user. Dapat dilihat pada gambar 3.12.



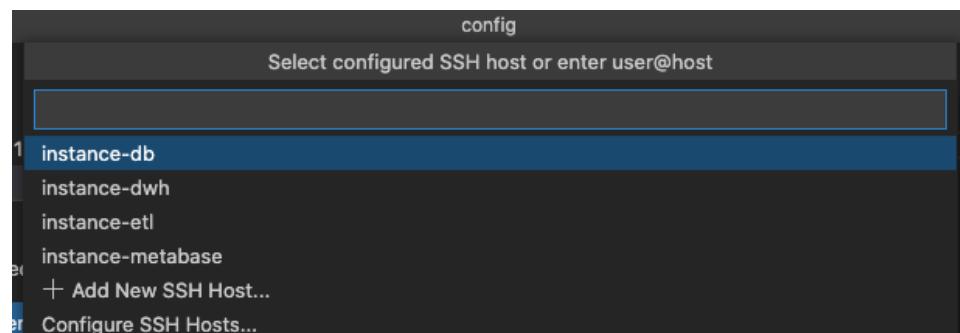
```

    Release Notes: 1.64.0      config
Users > tananugrahr > .ssh > config
1 | Host instance-db
2 | HostName 34.67.108.213
3 | User tananugrahr
4 |
5 | Host instance-dwh
6 | HostName 108.59.85.245
7 | User tananugrahr
8 |
9 | Host instance-etl
10 | HostName 35.224.135.149
11 | User tananugrahr
12 |
13 | Host instance-metabase
14 | HostName 34.135.234.230
15 | User tananugrahr
16

```

Gambar 3.12 *SSH Config*

Pada proses selanjutnya pilih *connect to host* dan pilih *VMInstace* yang akan kita *remote*. Dapat dilihat pada gambar 3.13



Gambar 3.13 *Connect to host*

c. Instalasi Docker Pada *VMInstace*

Setelah masuk pada instace yang dipilih kemudian install docker menggunakan terminal pada *VSCode* untuk perintah instalasi docker adalah seperti berikut:

- 1) Pertama, perbarui daftar paket Anda yang ada dengan perintah:

```
$ sudo apt update
```

- 2) Selanjutnya, instal beberapa paket prasyarat yang memungkinkan apt menggunakan paket melalui *HTTPS* dengan perintah:

```
$ sudo apt install apt-transport-https ca-certificates
curl software-properties-common
```

- 3) Kemudian tambahkan kunci GPG untuk repositori Docker resmi ke sistem dengan perintah:

```
$ curl -fsSL
https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
```

- 4) Tambahkan repositori *Docker* ke sumber *APT* dengan perintah:

```
$ sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu focal stable"
```

- 5) Terakhir, instal *Docker* dengan perintah:

```
$ sudo apt install docker-ce
```

- 6) Memeriksa *docker* yang diinstal berjalan pada *VMInstance* dengan perintah:

```
$ sudo systemctl status docker
```

d. Instalasi Tools Pada Docker

Pada proses ini menginstal *image* dan *container Mysql, PostgreSql, Airbyte dan Metabase* pada *docker* dengan *docker-compose* pada setiap *VMInstance* yang telah dibuat.

1) *Mysql*

Untuk menginstal Mysql pada Instace-db menggunakan perintah docker Compose seperti berikut:

```
version: '3.9'
services:
  db:
    container_name: mysql8
    image: mysql:8.0
    command: mysqld --default-authentication-
      plugin=mysql_native_password --character-set-
      server=utf8mb4 --collation-
      server=utf8mb4_unicode_ci
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: sandbox_bi
      MYSQL_ALLOW_EMPTY_PASSWORD: "yes"
    ports:
```

```

    - '3306:3306'
volumes:
- './docker/db/data:/var/lib/mysql'
- './docker/db/my.cnf:/etc/mysql/conf.d/my.cnf'
- './docker/db/sql:/docker-entrypoint-initdb.d'

```

Kemudian simpan dengan nama *docker-compose.yaml* pada direktori *Instace* kemudian install *docker-compose.yml* dengan perintah:

```
$ sudo docker-compose up -d
```

Selanjutnya periksa apakah container telah terinstal pada *Instace-db* dapat dilihat pada gambar 3.14.

```

TERMINAL PORTS OUTPUT DEBUG CONSOLE
> ▾ TERMINAL
△ tananugrahr@instance-db:~$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAME
6dd3b026ad9d mysql:8.0 "docker-entrypoint.s..." 6 weeks ago Up 3 weeks 0.0.0.0:3306->3
306/tcp, :::3306->3306/tcp, 3306/tcp mysql8
tananugrahr@instance-db:~$ █

```

Gambar 3.14 Container Docker pada *Instace-db*

2) PostgreSql

Untuk menginstal *Postresql* pada *Instace-dwh* menggunakan perintah *docker compose* seperti berikut:

```

version: "3.3"
services:
  db:
    image: "postgres:13"
    ports:
      - "5432:5432"
    volumes:
      - ./pgdata:/var/lib/postgresql/data
    environment:
      - POSTGRES_USER=dbuser
      - POSTGRES_PASSWORD=admin2021
      - POSTGRES_DB=dwh

```

Kemudian simpan dengan nama *docker-compose.yaml* pada direktori *Instace-dwh* kemudian install *docker-compose.yml* dengan perintah:

```
$ sudo docker-compose up -d
```

Selanjutnya periksa apakah container telah terinstal pada Instance-dwh dapat dilihat pada gambar 3.15.

```
tananugrahr@instance-dwh:~/docker_pmb_skripsi/postgres$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND             CREATED          STATUS          PORTS
NAMES
53fef2ac2956   postgres:13 "docker-entrypoint.s..."  6 weeks ago   Up 6 weeks   0.0.0.0:5432->5432/tcp, ::::5432->5432/tcp
tananugrahr@instance-dwh:~/docker_pmb_skripsi/postgres$
```

Gambar 3.15 Container Docker pada Instance-dwh

3) Airbyte

Untuk menginstal *Airbyte* pada *Instace-etl* menggunakan perintah *docker compose* seperti berikut:

```
version: "3.7"
#https://github.com/compose-spec/compose-
spec/blob/master/spec.md#using-extensions-as-fragments
x-logging: &default-logging
  options:
    max-size: "100m"
    max-file: "5"
    driver: json-file
services:
  # hook in case we need to add init behavior
  # every root service (no depends_on) should depend on
  init
    init:
      image: airbyte/init:${VERSION}
      logging: *default-logging
      container_name: init
      command: /bin/sh -c
        ./scripts/create_mount_directories.sh /local_parent
        ${HACK_LOCAL_ROOT_PARENT} ${LOCAL_ROOT}"
      environment:
        - LOCAL_ROOT=${LOCAL_ROOT}
        -
HACK_LOCAL_ROOT_PARENT=${HACK_LOCAL_ROOT_PARENT}
volumes:
  - ${HACK_LOCAL_ROOT_PARENT}:/local_parent
db:
  image: airbyte/db:${VERSION}
  logging: *default-logging
  container_name: airbyte-db
  restart: unless-stopped
  environment:
    - POSTGRES_USER=${DATABASE_USER}
    - POSTGRES_PASSWORD=${DATABASE_PASSWORD}
  volumes:
```

```

    - db:/var/lib/postgresql/data
scheduler:
  image: airbyte/scheduler:${VERSION}
  logging: *default-logging
  container_name: airbyte-scheduler
  restart: unless-stopped
  environment:
    - WEBAPP_URL=${WEBAPP_URL}
    - DATABASE_USER=${DATABASE_USER}
    - DATABASE_PASSWORD=${DATABASE_PASSWORD}
    - DATABASE_URL=${DATABASE_URL}
  CONFIG_DATABASE_USER=${CONFIG_DATABASE_USER:-}
  CONFIG_DATABASE_PASSWORD=${CONFIG_DATABASE_PASSWORD:-}
  }-
  CONFIG_DATABASE_URL=${CONFIG_DATABASE_URL:-}
  RUN_DATABASE_MIGRATION_ON_STARTUP=${RUN_DATABASE_MIGRATION_ON_STARTUP}
    - WORKSPACE_ROOT=${WORKSPACE_ROOT}
    -
  WORKSPACE_DOCKER_MOUNT=${WORKSPACE_DOCKER_MOUNT}
    - LOCAL_ROOT=${LOCAL_ROOT}
    - LOCAL_DOCKER_MOUNT=${LOCAL_DOCKER_MOUNT}
    - CONFIG_ROOT=${CONFIG_ROOT}
    - TRACKING_STRATEGY=${TRACKING_STRATEGY}
    - AIRBYTE_VERSION=${VERSION}
    - AIRBYTE_ROLE=${AIRBYTE_ROLE:-}
    - TEMPORAL_HOST=${TEMPORAL_HOST}
    - WORKER_ENVIRONMENT=${WORKER_ENVIRONMENT}
    - S3_LOG_BUCKET=${S3_LOG_BUCKET}
    - S3_LOG_BUCKET_REGION=${S3_LOG_BUCKET_REGION}
    - AWS_ACCESS_KEY_ID=${AWS_ACCESS_KEY_ID}
    - AWS_SECRET_ACCESS_KEY=${AWS_SECRET_ACCESS_KEY}
    - GCP_STORAGE_BUCKET=${GCP_STORAGE_BUCKET}
    - LOG_LEVEL=${LOG_LEVEL}
    - SUBMITTER_NUM_THREADS=${SUBMITTER_NUM_THREADS}
    - RESOURCE_CPU_REQUEST=${RESOURCE_CPU_REQUEST}
    - RESOURCE_CPU_LIMIT=${RESOURCE_CPU_LIMIT}
    -
  RESOURCE_MEMORY_REQUEST=${RESOURCE_MEMORY_REQUEST}
    - RESOURCE_MEMORY_LIMIT=${RESOURCE_MEMORY_LIMIT}
    - MAX_SYNC_JOB_ATTEMPTS=${MAX_SYNC_JOB_ATTEMPTS}
    - MAX_SYNC_TIMEOUT_DAYS=${MAX_SYNC_TIMEOUT_DAYS}
    - INTERNAL_API_HOST=${INTERNAL_API_HOST}
volumes:
  - /var/run/docker.sock:/var/run/docker.sock
  - workspace:${WORKSPACE_ROOT}
  - ${LOCAL_ROOT}:${LOCAL_ROOT}
server:
  image: airbyte/server:${VERSION}
  logging: *default-logging
  container_name: airbyte-server
  restart: unless-stopped
  environment:
    - WEBAPP_URL=${WEBAPP_URL}
    - DATABASE_USER=${DATABASE_USER}
    - DATABASE_PASSWORD=${DATABASE_PASSWORD}
    - DATABASE_URL=${DATABASE_URL}

```

```

    - CONFIG_DATABASE_USER=${CONFIG_DATABASE_USER:-}
    -
    CONFIG_DATABASE_PASSWORD=${CONFIG_DATABASE_PASSWORD:-}
    - CONFIG_DATABASE_URL=${CONFIG_DATABASE_URL:-}
    -
    RUN_DATABASE_MIGRATION_ON_STARTUP=${RUN_DATABASE_MIGRATION_ON_STARTUP}
    - WORKSPACE_ROOT=${WORKSPACE_ROOT}
    - CONFIG_ROOT=${CONFIG_ROOT}
    - TRACKING_STRATEGY=${TRACKING_STRATEGY}
    - AIRBYTE_VERSION=${VERSION}
    - AIRBYTE_ROLE=${AIRBYTE_ROLE:-}
    - TEMPORAL_HOST=${TEMPORAL_HOST}
    - WORKER_ENVIRONMENT=${WORKER_ENVIRONMENT}
    - S3_LOG_BUCKET=${S3_LOG_BUCKET}
    - S3_LOG_BUCKET_REGION=${S3_LOG_BUCKET_REGION}
    - AWS_ACCESS_KEY_ID=${AWS_ACCESS_KEY_ID}
    - AWS_SECRET_ACCESS_KEY=${AWS_SECRET_ACCESS_KEY}
    - GCP_STORAGE_BUCKET=${GCP_STORAGE_BUCKET}
    - LOG_LEVEL=${LOG_LEVEL}
    - RESOURCE_CPU_REQUEST=${RESOURCE_CPU_REQUEST}
    - RESOURCE_CPU_LIMIT=${RESOURCE_CPU_LIMIT}
    -
    RESOURCE_MEMORY_REQUEST=${RESOURCE_MEMORY_REQUEST}
    - RESOURCE_MEMORY_LIMIT=${RESOURCE_MEMORY_LIMIT}
  ports:
    - 8001:8001
  volumes:
    - workspace:${WORKSPACE_ROOT}
    - data:${CONFIG_ROOT}
    - ${LOCAL_ROOT}://${LOCAL_ROOT}
  webapp:
    image: airbyte/webapp:${VERSION}
    logging: *default-logging
    container_name: airbyte-webapp
    restart: unless-stopped
    ports:
      - 8000:80
    environment:
      - AIRBYTE_ROLE=${AIRBYTE_ROLE:-}
      - AIRBYTE_VERSION=${VERSION}
      - API_URL=${API_URL:-}
      - IS_DEMO=${IS_DEMO:-}
      - PAPERUPS_STORYTIME=${PAPERUPS_STORYTIME:-}
      - FULLSTORY=${FULLSTORY:-}
      - OPENREPLAY=${OPENREPLAY:-}
      - TRACKING_STRATEGY=${TRACKING_STRATEGY}
      - INTERNAL_API_HOST=${INTERNAL_API_HOST}
  airbyte-temporal:
    image: temporalio/auto-setup:1.7.0
    logging: *default-logging
    container_name: airbyte-temporal
    restart: unless-stopped
    ports:
      - 7233:7233
    environment:

```

```

- DB=postgresql
- DB_PORT=${DATABASE_PORT}
- POSTGRES_USER=${DATABASE_USER}
- POSTGRES_PWD=${DATABASE_PASSWORD}
- POSTGRES_SEEDS=${DATABASE_HOST}
-
DYNAMIC_CONFIG_FILE_PATH=config/dynamicconfig/development.yaml
  - LOG_LEVEL=${LOG_LEVEL}
volumes:
  -
    ./temporal/dynamicconfig:/etc/temporal/config/dynamicconfig
    volumes:
      workspace:
        name: ${WORKSPACE_DOCKER_MOUNT}
        # the data volume is only needed for backward
        compatibility; when users upgrade
        # from an old Airbyte version that relies on file-
        based configs, the server needs
        # to read this volume to copy their configs to the
        database
      data:
        name: ${DATA_DOCKER_MOUNT}
    db:
      name: ${DB_DOCKER_MOUNT}

```

Selanjutnya simpan dengan nama *docker-compose.yaml* pada direktori *Instace-etl* kemudian install *docker-compose.yml* dengan perintah:

```
$ sudo docker-compose up -d
```

Periksa apakah container telah terinstal pada Instace-etl dengan perintah `$ sudo docker ps -a`. dapat dilihat pada gambar 3.16.

COLUMN	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
	fd0f17e26072	airbyte/webapp:0.29.14-alpha	"/docker-entrypoint..."	6 weeks ago	Up 6 w
	eeks		0.0.0.0:8000->80/tcp, :::8000->80/tcp		
		airbyte-webapp			
	e3a217aebc0d	airbyte/scheduler:0.29.14-alpha	"/bin/bash -c bin/\${...}"	6 weeks ago	Up 6 w
	eeks		airbyte-scheduler		
	8810207d81bc	airbyte/server:0.29.14-alpha	"/bin/bash -c bin/\${...}"	6 weeks ago	Up 6 w
	eeks		8000/tcp, 0.0.0.0:8001->8001/tcp, :::8001->8001/tcp		
		airbyte-server			
	88e9552327cb	temporalio/auto-setup:1.7.0	"/entrypoint.sh /bin..."	6 weeks ago	Up 6 w
	eeks		6933-6935/tcp, 6939/tcp, 7234-7235/tcp, 7239/tcp, 0.0.0.0:7233->7233/tcp, :::7233->7233/tcp		
	0712b38145f5	airbyte/db:0.29.14-alpha	"docker-entrypoint.s..."	6 weeks ago	Up 6 w
	eeks		5432/tcp		
		airbyte-db			
	79189bfa3fda	airbyte/init:0.29.14-alpha	"/bin/sh -c './script...'	6 weeks ago	Exited
	(0) 3 weeks ago	init			
	tananugrahr@instance-etl:~\$				

Gambar 3.16 Container pada Instace-etl

4) Metabase

Untuk menginstal *Metabase* pada Instace-metabase menggunakan perintah *Docker Compose* seperti berikut:

```
version: '3.9'
services:
  metabase-secrets:
    image: metabase/metabase:v0.40.2
    container_name: metabase-secrets
    hostname: database.rata.id
    volumes:
      - metabase-data:/dev/random:ro
    ports:
      - 3000:3000
    environment:
      MB_DB_TYPE: postgres
      MB_DB_DBNAME: metabase
      MB_DB_PORT: 5432
      MB_DB_USER: /run/secrets/db_user
      MB_DB_PASS: /run/secrets/db_password
      MB_DB_HOST: postgres-secrets
    networks:
      - metanet1-secrets
    depends_on:
      - postgres-secrets
    secrets:
      - db_password
      - db_user
  postgres-secrets:
    image: postgres:latest
    container_name: postgres-secrets
    hostname: postgres-secrets
    environment:
      POSTGRES_USER: /run/secrets/db_user
      POSTGRES_DB: metabase
      POSTGRES_PASSWORD: /run/secrets/db_password
    networks:
      - metanet1-secrets
    secrets:
      - db_password
      - db_user
  volumes:
    metabase-data:
networks:
  metanet1-secrets:
    driver: bridge
secrets:
  db_password:
    file: db_password.txt
  db_user:
    file: db_user.txt
```

Kemudian simpan dengan nama *docker-compose.yaml* pada direktori *Instace-metabase* kemudian install *docker-compose.yml* dengan perintah:

```
$ sudo docker-compose up -d
```

Selanjutnya periksa apakah container telah terinstal pada Instace - *metabase* dapat dilihat pada gambar 3.17.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
2540cab53719	metabase/metabase:v0.40.2	"/app/run_metabase.sh"	6 weeks ago	Up 6 weeks
0.0.0.0:3000->3000/tcp, :::3000->3000/tcp		metabase-secrets		
5399f94aa172	postgres:latest	"docker-entrypoint.s..."	6 weeks ago	Up 6 weeks
5432/tcp		postgres-secrets		

Gambar 3.17 Docker Container Pada Instace -metabase

5) Python

Untuk menjalankan Python Script diperlukan menginstal *Python* pada *instance*. Pada proses ini *Python* diinstal pada *Instance-metabase*. Untuk menginstal *Python* pada *Instance-Metabase* menggunakan perintah:

```
$ sudo apt install python3-pip
```

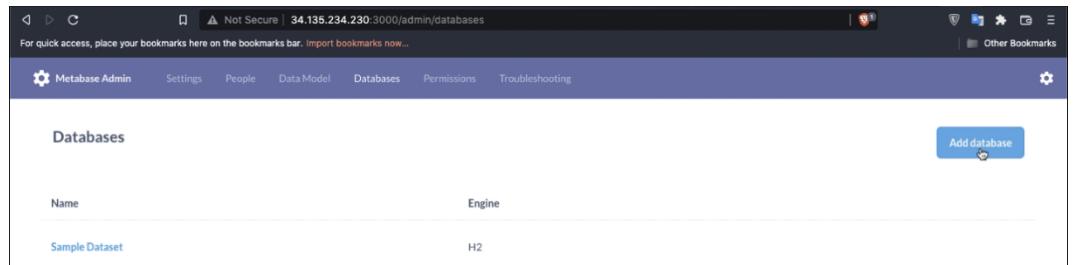
6) Aws Data Wrangler

Untuk menjalankan *AWS Data Wrangler* diperlukan menginstal *AWS Data Wrangler* pada instance. Pada proses ini *AWS Data Wrangler* diintal pada *VMInstace-metabase*. Untuk mengintal *AWS Data Wrangler* pada *Instance-Metabase* menggunakan perintah:

```
$ pip install pyarrow==2 awswrangler
```

e. Setup Metabase

Pada proses ini setelah metabase di install pada docker kemudian jalankan metabase di browser dengan alamat url (ip eksternal pada instance metabase:port) yaitu <http://34.135.234.230:3000/>. Kemudian masuk ke menu admin dan pilih *databases* kemudian *add databases* seperti pada gambar 3.18.



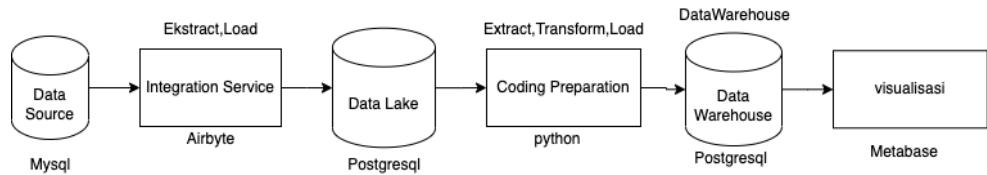
Gambar 3.18 *Setup Metabase*

Selanjutnya koneksi *data warehouse postgresql* di metabase dan isi *Host* sesuai dengan *IP Eksternal Instance-dwh* dan pada data lainnya sesuai dengan *docker-compose postgresql* yang telah dibuat seperti pada gambar 3.19.

The screenshot shows a 'ADD DATABASE' form under the 'DATABASES' tab. The 'Database type' dropdown is set to 'PostgreSQL'. The 'Name' field contains 'dwh'. The 'Host' field contains '108.59.85.245'. The 'Port' field contains '5432'. The 'Database name' field contains 'dwh'. The 'Username' field contains 'dbuser'. The 'Password' field is redacted with dots. At the bottom right of the form is a blue 'Save' button.

Gambar 3.19 Koneksi *data warehouse postgresql*

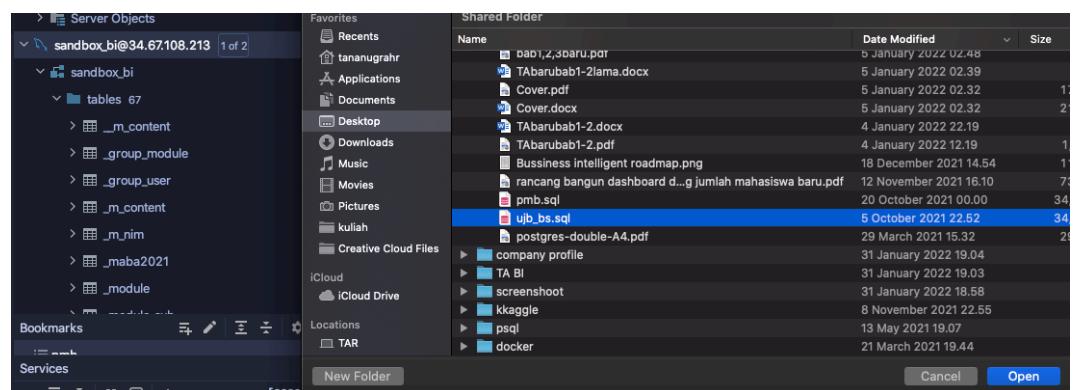
Setelah melakukan tahapan *setup* perangkat lunak kemudian lanjut pada proses pembangunan *data warehouse* penerimaan mahasiswa baru Universitas Janabadra yang berdasarkan rancangan skema yang telah dibuat pada proses *design*. Alur data pembuatan *data warehouse* penerimaan mahasiswa baru Universitas Janabadra dapat dilihat pada gambar 3.20.



Gambar 3.20 Alur data pembuatan data *warehouse*

a. Data Source

Data source PMB Universitas Janabadra yang didapat dari Siskom Universitas Janabadra berbentuk *SQL dump* dengan format *MySQL*. Data dari *sql dump* di *import* pada *instance-db* pada menggunakan *Datagrip* pada server *data source* yang telah dikoneksikan di *datagrip* klik *run sql script* kemudian pilih file *sql* dan klik *open*. untuk prosesnya dapat dilihat pada gambar 3.21.



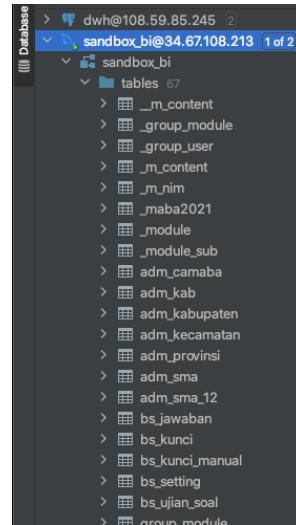
Gambar 3.21 Import Data source

b. Integration Service

Pada proses integrasi melakukan *Extract* (mengambil data dari sumber) kemudian *Loading* (memasukkan data ke target) target dari proses ini adalah *Data Lake*. Pada. Proses integrasi ini mengubah jenis data dari data *MySQL* menjadi *PostgreSQL*. dan *loading* tabel yang dibutuhkan untuk analisis ke *data lake*.

1) Ekstract data Source

Pada tahapan ini *Ekstract data source* PMB ke dalam server data *source* pada *vm instance di google cloud platform*. Maka data akan muncul seperti gambar 3.22.



Gambar 3.22 Data Source PMB

2) Connection data

Pada proses ini menghubungkan server *data source* ke server data tujuan. Pada proses ini memilih tabel yang dibutuhkan dalam pembangunan data *warehouse* PMB untuk diload kedalam *data lake* PMB yaitu tabel mhs_cln, mhs_regis, m_tahun, m_kewarganegaraan, m_agama, m_prodi, m_fakultas dan adm_provinsi. Pada proses ini *data source MySQL* dirubah menjadi *PostgreSQL*. Untuk proses *set data source* dan *set data destination* dapat dilihat pada gambar 3.23 Dan 3.24.

Gambar 3. 23 Set data source

Destination Settings

Name * - Pick a name to help you identify this destination in Airbyte
dwh - mbs.clp

Destination type
Postgres

Host * - Hostname of the database.
108.59.85.245

Port * - Port of the database. (e.g. 5432)
5432

DB Name * - Name of the database.
dwh

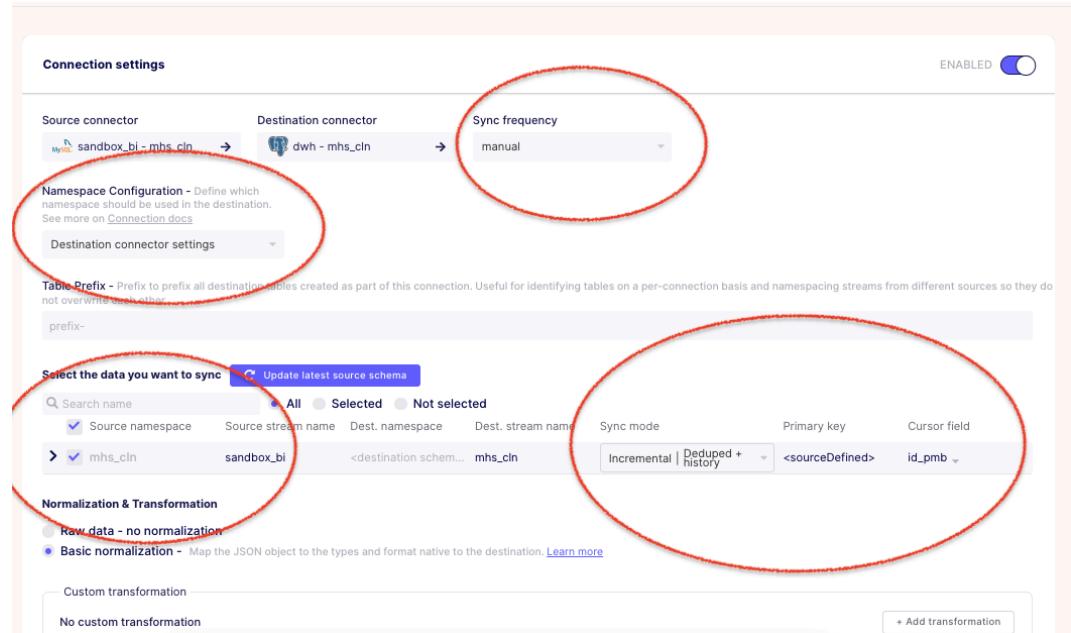
Default Schema * - The default schema tables are written to if the source does not specify a namespace. The usual value for this field is "public", (e.g. public)
public

User * - Username to use to access the database.
dbuser

Password - Password associated with the username.
***** Edit

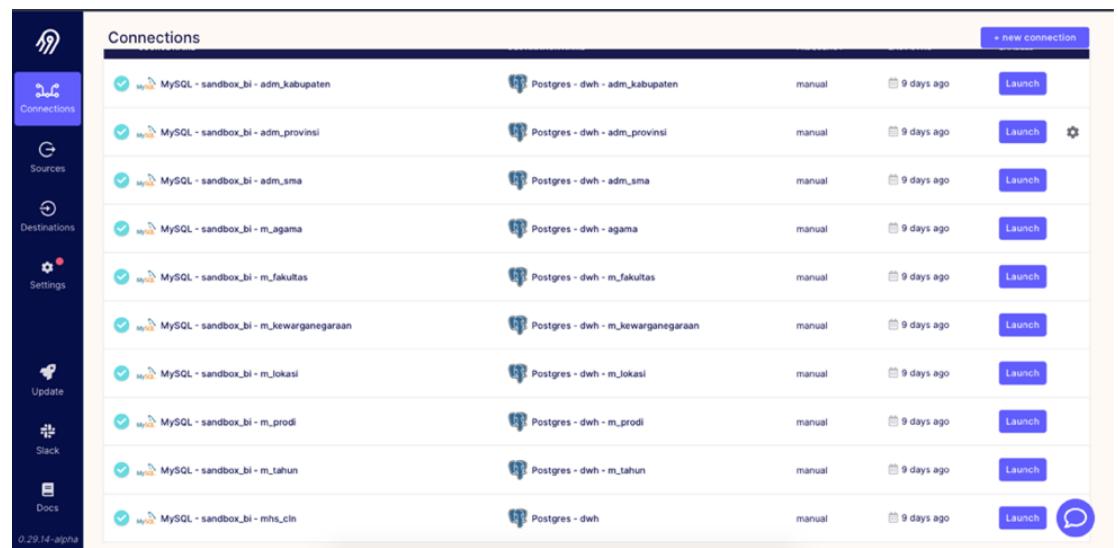
Gambar 3.24 Set data destination

Pada proses *connection setting* setelah *set data source* dan *data destination* pada proses *Sync frequency* set menjadi manual agar data disinkronisasikan hanya saat dibutuhkan. Kemudian *Namespace Configuration set* pada *Destination Connection setting* agar data yang diintegrasikan tidak membuat skema baru dan masuk ke dalam skema *public* lanjut pilih tabel yang akan diintegrasikan selanjutnya *sync mode* dirubah menjadi *Incremental | Deduped + History* agar apabila melakukan sinkronisasi data tidak menduplikasi data yang sudah ada dan hanya menambahkan data baru dan selanjutnya *Cursor field* pada *primary key* pada tabel yang dipilih. Prosesnya dapat dilihat pada gambar 3.25.



Gambar 3.25 Connection Setting

Kemudian lakukan proses seperti pada gambar diatas terhadap data tabel yang ingin diintegrasikan. Hasil dari tabel tabel yang dipilih untuk proses pembangunan *data warehouse* penerimaan mahasiswa baru Universitas Janabadra dapat dilihat pada gambar 3.26.



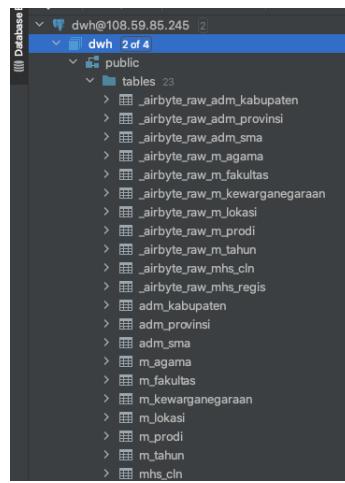
Gambar 3.26 Connection data pada Airbyte

3) Loading data

Setelah melakukan proses *connection*, data yang telah pilih untuk analisis akan otomatis masuk pada server *Data Lake*.

c. Data Lake

Data yang terdapat dalam *data lake* PMB adalah data hasil dari proses integrasi menggunakan *airbyte*. Berikut tampilan *data lake* yang telah di *load* menggunakan *airbyte* dapat dilihat pada gambar 3.27.



Gambar 3.27 *Data Lake* PMB

Proses selanjutnya menambah tabel baru dengan nama *dim_waktu* yang mengelompokkan waktu dari tabel yang ada. Tabel *dim_waktu* dapat dilihat pada tabel 3.11.

Fungsi : Berfungsi untuk menampung informasi waktu

Nama Tabel : *dim_waktu*

Primary Key : *Id_pmb*

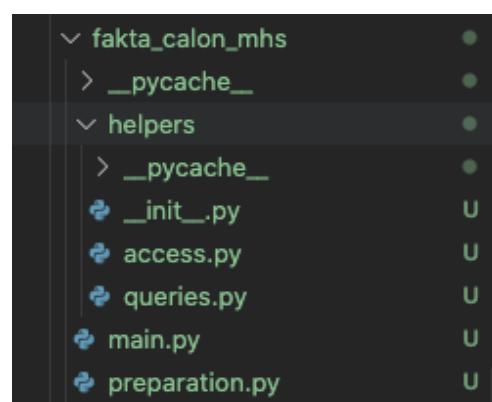
Jumlah *Field* : 8 *field*

Tabel 3.12 Tabel dim_waktu

No	Nama Kolom	Tipe_data	Size	Key	Keterangan
1	N_pmb	Bigint			Jumlah Id yang terhitung per field
2	Id_pmb	varchar	11 byte	PK	Id pada tabel waktu
3	Tanggal_lahir	Date			Tanggal lahir calon mahasiswa
4	Tanggal_bayar	Date			Tanggal bayar calon mahasiswa
5	Tanggal_ujian	Varchar	50 byte		Tanggal ujian mahasiswa
6	Id_tahun	Float8			Id tahun akademik
7	Tahun_akademik	Varchar	10 byte		Informasi tahun akademik
8	umur	Float8			Umur calon mahasiswa

d. Coding Preparation

Pada proses ini mengekstrak data dari *data lake* PMB. Setelah tahap ekstraksi selesai, data perlu dipindahkan ke sistem target atau ke sistem perantara untuk diproses lebih lanjut. Langkah selanjutnya adalah proses transformasi *ETL* untuk membangun data *Warehouse* terstruktur. Pada Langkah ini memastikan data yang akan diolah sepenuhnya siap dan kompatibel. Setelah data telah dipastikan bersih kemudian *load* kedalam data *warehouse*. Pada proses *Coding Preparation* menggunakan *Syntaks python* yang dibangun menggunakan *AWS Data Wrangler* untuk menjalankan proses *ETL*. Pada proses ini susunan modul pada *script python* dapat dilihat pada gambar 3.28.



Gambar 3.28 Susunan Modul Script Python

Berikut proses pembuatan data *warehouse* calon mahasiswa dan mahasiswa:

1. Calon Mahasiswa yang Mendaftar PMB

Pada proses ini untuk membangun data calon mahasiswa harus melakukan *query* sesuai dengan *star* skema calon mahasiswa yang telah dirancang sebelumnya. *Star* skema calon mahasiswa dapat dilihat pada gambar 3.2. Proses yang dilakukan adalah pembuatan tabel baru yang berisi fakta calon mahasiswa.

a. Membuat tabel fakta calon mahasiswa

Membuat tabel baru dengan nama *fakta_calon_mhs* Tabel fakta calon mahasiswa dapat dilihat pada tabel 3.11.

Fungsi : Berfungsi untuk menampung informasi fakta calon mahasiswa PMB Universitas Janabadra

Nama Tabel : *fakta_calon_mhs*

Primary Key : *Id_pmb*

Foreign Key : *Id_pmb,id_tahun,kode_agama,kode_kewarganegaraan, id_prodi, id_fakultas*

Jumlah *Field* : 7 *field*

Tabel 3.13 Tabel fakta calon mahasiswa

No	Nama Kolom	Tipe Data	Size	Key	Keterangan
1	N_pmb	bigint			Jumlah id_pmb
2	Id_pmb	varchar	11 byte	PK, FK	Id tabel fakta
3	Id_tahun	Float8		FK	Id tahun
4	Kode_agama	varchar	2 byte	FK	Kode agama mahasiswa
5	Kode_kewarganegaraan	varchar	2 byte	FK	Kode kewarganegaraan mahasiswa
6	Id_prodi	Float8		FK	Id program studi
7	Id_fakultas	Float8	2 byte	FK	Id fakultas

Source code sql untuk pembuatan tabel fakta calon mahasiswa adalah sebagai berikut:

```
create table sandbox_bi.fakta_calon_mahasiswa (
    n_pmb bigint,
    id_pmb varchar(11) PRIMARY KEY ,
    id_tahun float8,
    kode_agama varchar(2),
    kode_kewarganegaraan varchar(2),
    id_prodi float8,
    id_fakultas float8)
```

b. *Load Data Calon Mahasiswa*

Menjalankan *query statement* untuk menggabungkan tabel mhs_cln, tabel m_prodi, tabel_m_fakultas, tabel kewarganegaraan, tabel_tahun, tabel_dim_waktu dan tabel_agama berdasarkan dengan skema yang telah dibuat, kemudian *load* data ke dalam tabel fakta calon mahasiswa menggunakan *python script*. Berikut python script yang akan dijalankan:

1) *init.py*

file dieksekusi secara implisit, dan *Python* akan menambahkan beberapa atribut tambahan ke modul saat diimpor.

```
from helpers.queries import
query_pararelize_services
__all__ = ['query_pararelize_services']
```

2) *Access.py*

syntaks ini berfungsi sebagai *connection* pada server dwh.

Berikut *syntask connection access* server dwh:

```
host_dwh='108.59.85.245'
database_dwh='dwh'
user_dwh='dbuser'
password_dwh='admin2021'
```

3) *Queries.py*

pada *syntask* ini berfungsi sebagai penampung *query* pembangunan *data warehouse* calon mahasiswa. Berikut *query data warehouse* calon mahasiswa:

```

class query_pararelize_services :
    select_service_dbsources = (
"""
With calon_mahasiswa_fact as (
    select distinct
        mhs_cln.id_pmb,m_agama.kode_agama,
        m_kewarganegaraan.kode_kewarganegaraan,
        m_prodi.id_prodi,m_fakultas.id_fakultas,
        dim_waktu.id_tahun
    from mhs_cln
    left join m_prodi on mhs_cln.prodi1 =
        m_prodi.id_prodi
        left join m_fakultas on m_prodi.id_fakultas
        = m_fakultas.id_fakultas
        left join m_agama on mhs_cln.kode_agama =
            m_agama.kode_agama
        left join m_kewarganegaraan on mhs_cln.warga
        = m_kewarganegaraan.kode_kewarganegaraan
        left join dim_waktu on mhs_cln.id_pmb =
            dim_waktu.id_pmb
    where mhs_cln.id_pmb != '' and
        mhs_cln.id_pmb is not null and
        dim_waktu.id_tahun >= 7 and
        mhs_cln.tanggal_bayar !='' and
        mhs_cln.tanggal_bayar is not null and
        mhs_cln.tanggal_bayar != '0000-00-00'),
fact_table_calon_mahasiswa as (
    select n_pmb,
        id_pmb,
        id_tahun,
        kode_agama,
        kode_kewarganegaraan,
        id_prodi,
        id_fakultas
    from (select count(id_pmb) n_pmb,
        id_pmb,
        id_tahun,
        kode_agama,
        kode_kewarganegaraan,
        id_prodi,
        id_fakultas
    from calon_mahasiswa_fact
    group by 2, 3, 4, 5,6,7) a
    group by n_pmb,2, 3, 4, 5, 6 ,7)
select
    n_pmb,
    id_pmb,
    id_tahun,
    kode_agama,
    kode_kewarganegaraan,
    id_prodi,
    id_fakultas
from fact_table_calon_mahasiswa
group by n_pmb,2,3,4,5,6,7
order by 3 desc;
""")

```

4) Preparation.py

Pada *syntask* ini berisi beberapa fungsi yakni membaca data dari sumber data, menyiapkan data *mengganti string = null* ke nilai kosong, *ingestion* data ke dwh yakni proses pengangkutan data dari satu atau lebih sumber ke situs target untuk diproses dan dianalisis lebih lanjut, menyusun kerangka data , mengonversi tipe data, menampilkan standar informasi data dan pernyataan *ingestion* berikut *syntask* yang disusun:

```

import awswrangler as wr
from helpers.access import *
import asyncio
import pandas as pd
import numpy as np
import pg8000
import sys
def connection_postgres():
    try:
        con = pg8000.connect(
            host=host_dwh,
            port='5432',
            database = database_dwh,
            user = user_dwh,
            password = password_dwh)
    except pg8000.Error as er:
        print("Opps: Something Else:", er)
    else:
        return con
async def read_data(sql_statement): #read data from sources
    try:
        connection_=connection_postgres()
        datareader = wr.postgresql.read_sql_query(
            sql=sql_statement,con=connection_)
        connection_.close()
    except:
        print('Opps: Check your statemen query:', sys.exc_info())
    else:
        return datareader
async def operation_read_data(sql_statement):
    #thread
    try:
        tasks = []
        tasks.append(asyncio.ensure_future(read_data(sql_statement)))
        original_result_read_data = await
        asyncio.gather(*tasks)
    except:

```

```

print('operations read_data exception in :',
      sys.exc_info())
else:
    return original_result_read_data
# prepare data replace string = null to empty
value
async def prepare_data(reader_data):
    df_dirty= pd.DataFrame(reader_data)
    for col in
        list(df_dirty.select_dtypes(include=['string',
            'object']).columns):
            df_dirty[col].fillna(value=np.nan, inplace=True)
    for col in
        list(df_dirty.select_dtypes(include=['Int64',
            'boolean', 'float64']).columns):
            df_dirty[col].fillna(value=np.nan, inplace=True)
    return df_dirty
async def
operation_prepare_data(data_accessibility):
    try:
        tasks = []
    tasks.append(asyncio.ensure_future(prepare_data(
        data_accessibility)))
    original_result_prepare_data = await
    asyncio.gather(*tasks)
    except:
        print('operations prepare_data exception in :',
              sys.exc_info())
    else:
        return original_result_prepare_data
async def ingestion_data_to_dwh (df): #ingests
data to dwh
try:
    # componen dataframe
    df_postgresql = pd.DataFrame(df)
    if len(df_postgresql) != 0:
        # convert string
        df_postgresql[[ 'id_pmb',
            'kode_agama',
            'kode_kewarganegaraan']] = df_postgresql[[ 'id_pmb',
            'kode_agama',
            'kode_kewarganegaraan']].astype('string')
        # convert int
        df_postgresql[[ 'n_pmb']] = df_postgresql[[ 'n_pmb']].astype('Int64')
        # convert float
        df_postgresql[[ 'id_tahun',
            'id_prodi',

```

```

'id_fakultas'
]] = df_postgresql[[
'id_tahun',
'id_prodi',
'id_fakultas'
]].astype('float64')
print('dataframe info after prepare: \n')
print(df_postgresql.info()) #show the
information standard of data
# ingestion statemen
wr.postgresql.to_sql(
df_postgresql,
con=connection_postgres(),
schema="sandbox_bi",
table="fakta_calon_mahasiswa",
mode="upsert",
chunksize=200,
upsert_conflict_columns=[
'id_pmb'
])
else:
print('Dataframe empty dataset')
except:
print('error in \n', sys.exc_info())
else:
print('success')
connection_postgres().close()
async def
operation_ingestion_data(data_ingestion_df):
try:
tasks = []
tasks.append(asyncio.ensure_future(ingestion_dat
a_to_dwh(data_ingestion_df)))
original_result_ingestion_data = await
asyncio.gather(*tasks)
except:
print('operations ingestion_data
exception in :', sys.exc_info())
else:
return original_result_ingestion_data

```

5) Main.py

Pada *syntask* ini berfungsi untuk menjalankan modul dan fungsi-fungsi yang digunakan berikut *syntask* yang digunakan:

```

from helpers.queries import
query_pararelize_services
from preparation import *
import time
import sys
start_time = time.time() # to check time
execute
if __name__ == '__main__':

```

```

try:
    # running asycn prosess
    loop = asyncio.get_event_loop()
    # state 0
    ops_reader_data =
    asyncio.ensure_future(operation_read_data(query_
pararelize_services.select_service_dbsources))
        act_reader_data =
    loop.run_until_complete(ops_reader_data)
        # print(act_reader_data[0])
        ops_prepare_data =
    asyncio.ensure_future(operation_prepare_data(act
_reader_data[0]))
        act_prepare_data =
    loop.run_until_complete(ops_prepare_data)
        # print(act_reader_data[0])
        ops_ingest_data =
    asyncio.ensure_future(operation_ingestion_data(a
ct_prepare_data[0]))
        act_ingest_data =
    loop.run_until_complete(ops_ingest_data)
        print('state 0 clear \n')
except:
    print('Error ingestion in : \n',
sys.exc_info())
else:
    print('Success ingestion data')
    print("--- %s seconds ---" % (time.time() -
start_time))

```

Kemudian Jalankan Main.py dengan perintah \$ Python3 (path lokasi main.py calon mahasiswa). Apabila file eror saat di ingestion maka akan terlihat pesan eror pada terminal. Sesuaikan urutan data pada file preparation.py di ingestion data. apabila file sukses di ingestion maka akan terlihat pada gambar 3.29.

```
tananugrahr@instance-metabase:~$ python3 /home/tananugrahr/coding/fakta_calon_mhs/main.py
      n_pmb    id_pmb  id_tahun kode_agama kode_kewarganegaraan  id_prodi  id_fakultas
0       1  2111010001     16.0        2           1       1.0        3.0
1       1  2111020003     16.0        2           1       1.0        3.0
2       1  2111020004     16.0        3           1       1.0        3.0
3       1  2111020006     16.0        1           1       1.0        3.0
4       1  2111020007     16.0        2           1       1.0        3.0
...
4839     1  1741142014     7.0        3           1       8.0        4.0
4840     1  1741142015     7.0        1           1       8.0        4.0
4841     1  1741142016     7.0        1           1       8.0        4.0
4842     1  1741150001     7.0      <NA>           1       8.0        4.0
4843     1  1741150002     7.0      <NA>           1       8.0        4.0
[4844 rows x 7 columns]
dataframe info after prepare:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4844 entries, 0 to 4843
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype  
--- 
 0   n_pmb             4844 non-null   Int64  
 1   id_pmb            4844 non-null   string  
 2   id_tahun          4844 non-null   float64 
 3   kode_agama         4792 non-null   string  
 4   kode_kewarganegaraan  4833 non-null   string  
 5   id_prodi          4844 non-null   float64 
 6   id_fakultas       4844 non-null   float64 
dtypes: Int64(1), float64(3), string(3)
memory usage: 269.8 KB
None
success
state 0 clear

Success ingestion data
--- 0.6214568614959717 seconds ---
tananugrahr@instance-metabase:~$
```

Gambar 3.29 Run Main.py Fakta Calon Mahasiswa

2. Mahasiswa Registrasi

Pada proses ini untuk membangun data mahasiswa registrasi PMB Universitas Janabadra harus melakukan *query* sesuai dengan *star* skema mahasiswa yang telah dirancang sebelumnya. *Star* skema mahasiswa dapat dilihat pada gambar 3.3. Proses yang dilakukan yaitu pembuatan tabel baru yang berisi fakta mahasiswa registrasi PMB Universitas Janabadra.

a. Membuat tabel fakta_mahasiswa

Membuat tabel baru dengan nama fakta_mahasiswa. Tabel fakta mahasiswa regis dapat dilihat pada tabel 3.12.

Fungsi : Berfungsi untuk menampung informasi fakta mahasiswa registrasi

Nama Tabel : fakta_mahasiswa

Primary Key : No_pendaftaran

Foreign Key : Id_pmb, kode_agama, id_prodi, id_fakultas,
 kode_kewarganegaraan, id_prov, id_tahun
Jumlah Field : 9 field

Tabel 3.14 Tabel fakta mahasiswa

No	Nama Kolom	Tipe Data	Size	Key	Keterangan
1	N_mb	bigint			Jumlah id_pmb
2	Id_pmb	varchar	11 byte	PK,FK	Id tabel fakta
3	Kode_agama	varchar	2 byte	FK	Kode agama mahasiswa
4	Id_prodi	Float8		FK	Id Program studi
5	Id_fakultas	Float8		FK	Id fakultas
6	kode_kewarganegaraan	varchar	2 byte	FK	kode kewarganegaraan
7	id_prov	Float8		FK	Id asal provinsi mahasiswa
8	Id_tahun	Float8		FK	Id tahun masuk
9	umur	Float8			Umur mahasiswa saat registrasi

Source code sql untuk pembuatan tabel fakta mahasiswa adalah sebagai berikut:

```
create table sandbox_bi.fakta_mahasiswa (
  n_mb bigint,
  id_pmb varchar(11) PRIMARY KEY ,
  kode_agama varchar(2),
  id_prodi float8,
  id_fakultas float8,
  kode_kewarganegaraan varchar(2),
  id_prov float8,
  id_tahun float8,
  umur float8)
```

b. *Load* data Mahasiswa Registrasi

Menjalankan *query statement* untuk menggabungkan tabel mhs_regis, tabel mhs_cln, tabel m_prodi, tabel_m_fakultas, tabel tabel_m_tahun, m_kewarganegaraan, tabel_agama dan tabel adm_provinsi kemudian *load* data ke dalam tabel fakta mahasiswa menggunakan *python script*. Berikut python script yang akan dijalankan:

1) *Init.py*

file dieksekusi secara implisit , *import query* konfigurasi
server paralelisme

```
from helpers.queries import
query_pararelize_services
__all__ = ['query_pararelize_services']
```

2) *Access.py*

Syntax ini berfungsi sebagai *connection* pada server dwh.

Berikut *syntaks connection access* server dwh:

```
database_dwh='dwh'
user_dwh='dbuser'
password_dwh='admin2021'
```

3) *Queries.py*

Pada *syntax* ini berfungsi sebagai penampung *query* pembangunan *data warehouse* calon mahasiswa. Berikut *query data warehouse* calon mahasiswa:

```
class query_pararelize_services :
    select_service_dbsources = (
    """
    with mahasiswa_fact as(
        select distinct no_pendaftaran as id_pmb,
        mhs_cln.kode_agama,public.m_prodi.id_prodi,
        public.m_fakultas.id_fakultas,
        public.m_kewarganegaraan.kode_kewarganegaraan,
        public.adm_provinsi.id_prov,public.mhs_cln.id_ta
        hun,public.dim_waktu.umur
        from public.mhs_regis
        left join mhs_cln on
        mhs_regis.no_pendaftaran = mhs_cln.id_pmb
        left join public.m_agama on
        public.mhs_regis.no_pendaftaran =
        public.m_agama.kode_agama
        left join public.m_prodi on
        public.mhs_regis.id_prodi =
        public.m_prodi.id_prodi
        left join public.m_fakultas on
        public.m_prodi.id_fakultas =
        public.m_fakultas.id_fakultas
        left join public.m_kewarganegaraan on
        public.mhs_regis.warga =
        public.m_kewarganegaraan.kode_kewarganegaraan
        left join public.adm_provinsi on
        public.mhs_regis.prov_asal =
        public.adm_provinsi.kode_provinsi
```

```

        left join public.dim_waktu on
public.mhs_regis.no_pendaftaran =
public.dim_waktu.id_pmb
        left join public.m_tahun on
public.mhs_cln.id_tahun =
public.m_tahun.id_tahun
where
        no_pendaftaran !=''
        and no_pendaftaran is not null),
fact_table_mahasiswa_regis as (
select n_mb,id_pmb,kode_agama,id_prodi,
id_fakultas,kode_kewarganegaraan,id_prov,
id_tahun,umur
from (
select count(id_pmb) n_mb,id_pmb,kode_agama,
id_prodi,id_fakultas,kode_kewarganegaraan,
id_prov,id_tahun,umur
from mahasiswa_fact
group by 2, 3, 4, 5,6,7,8,9 ) a
group by n_mb,2, 3, 4, 5, 6 ,7,8,9)
select
n_mb,id_pmb,kode_agama,id_prodi,id_fakultas,
kode_kewarganegaraan,id_prov,id_tahun,umur
from fact_table_mahasiswa_regis
group by n_mb,2,3, 4,5,6,7,8,9
order by 8 desc;
"""
)

```

4) Preparation.py

Pada *syntax* ini berisi beberapa fungsi yakni membaca data dari sumber data, menyiapkan data mengganti *string = null* ke nilai kosong, *ingestion* data ke *dwh* yakni proses pengangkutan data dari satu atau lebih sumber ke situs target untuk diproses dan dianalisis lebih lanjut, menyusun kerangka data , mengonversi tipe data, menampilkan standar informasi data dan pernyataan *ingestion* berikut *syntax* yang disusun:

```

import awswrangler as wr
from helpers.access import *
import asyncio
import pandas as pd
import numpy as np
import pg8000
import sys
def connection_postgres():
    try:
        con = pg8000.connect(
            host=host_dwh,

```

```

        port='5432',
        database = database_dwh,
        user = user_dwh,
        password = password_dwh
    )
except pg8000.Error as er:
    print("OOps: Something Else:", er)
else:
    return con
async def read_data(sql_statement): #read data
from sources
    try:
        connection_ = connection_postgres()
        datareader =
wr.postgresql.read_sql_query(
            sql=sql_statement,
            con=connection_
        )
        connection_.close()
    except:
        print('Opps: Check your statemen
query:', sys.exc_info())
    else:
        return datareader
async def operation_read_data(sql_statement):
#thread
    try:
        tasks = []
tasks.append(asyncio.ensure_future(read_data(sql
_
statement)))
        original_result_read_data = await
asyncio.gather(*tasks)
    except:
        print('operations read_data exception
in :', sys.exc_info())
    else:
        return original_result_read_data
# prepare data replace string = null to empty
value
async def prepare_data(reader_data):
    df_dirty= pd.DataFrame(reader_data)
    for col in
list(df_dirty.select_dtypes(include=['string',
'object']).columns):
        df_dirty[col].fillna(value=np.nan,
inplace=True)
        for col in
list(df_dirty.select_dtypes(include=['Int64',
'boolean', 'float64']).columns):
        df_dirty[col].fillna(value=np.nan,
inplace=True)
    return df_dirty
async def
operation_prepare_data(data_accessibility):
    try:

```

```

        tasks = []
tasks.append(asyncio.ensure_future(prepare_data(
data_accessibility)))
    original_result_prepare_data = await
asyncio.gather(*tasks)
except:
    print('operations prepare_data
exception in :', sys.exc_info())
else:
    return original_result_prepare_data
async def ingestion_data_to_dwh (df): #ingests
data to dwh
try:
    # componen dataframe
    df_postgresql = pd.DataFrame(df)
if len(df_postgresql) != 0:

    # convert string
    df_postgresql[[

        'id_pmb',
        'kode_agama',
        'kode_kewarganegaraan'
    ]] = df_postgresql[[

        'id_pmb',
        'kode_agama',
        'kode_kewarganegaraan'
    ]].astype('string')
    # convert int
    df_postgresql[[

        'n_mb'
    ]] = df_postgresql[[

        'n_mb'
    ]].astype('Int64')
    # convert float
    df_postgresql[[

        'id_prodi',
        'id_fakultas',
        'id_prov',
        'id_tahun',
        'umur'
    ]] = df_postgresql[[

        'id_prodi',
        'id_fakultas',
        'id_prov',
        'id_tahun',
        'umur'
    ]].astype('float64')
    print('dataframe info after prepare:
\\n')
    print(df_postgresql.info()) #show
the information standard of data

    # ingestion statemen
    wr.postgresql.to_sql(
        df_postgresql,
        con=connection_postgres(),

```

```

        schema="sandbox_bi",
        table="fakta_mahasiswa",
        mode="upsert",
        chunksize=200,
        upsert_conflict_columns=[
            'id_pmb'])
    else:
        print('Dataframe empty dataset')
except:
    print('error in \n', sys.exc_info())
else:
    print('success')
    connection_postgres().close()
async def
operation_ingestion_data(data_ingestion_df):
    try:
        tasks = []
    tasks.append(asyncio.ensure_future(ingestion_dat
a_to_dwh(data_ingestion_df)))
        original_result_ingestion_data = await
asyncio.gather(*tasks)
    except:
        print('operations ingestion_data
exception in :', sys.exc_info())
    else:
        return original_result_ingestion_data

```

5) Main.py

Pada *syntax* ini berfungsi untuk menjalankan modul dan fungsi-fungsi yang digunakan berikut *syntax* yang digunakan:

```

from helpers.queries import
query_pararelize_services
from preparation import *
import time
import sys
start_time = time.time() # to check time
execute
if __name__ == '__main__':
    try:
        # running asycn prosess
        loop = asyncio.get_event_loop()
        # state 0
        ops_reader_data =
asyncio.ensure_future(operation_read_data(query_
pararelize_services.select_service_dbsources))
        act_reader_data =
loop.run_until_complete(ops_reader_data)
        # print(act_reader_data[0])
        ops_prepare_data =
asyncio.ensure_future(operation_prepare_data(act_
reader_data[0]))
        act_prepare_data =
loop.run_until_complete(ops_prepare_data)
        # print(act_reader_data[0])

```

```

        ops_ingest_data =
asyncio.ensure_future(operation_ingestion_data(a
ct_prepare_data[0]))
        act_ingest_data =
loop.run_until_complete(ops_ingest_data)
        print('state 0 clear \n')
    except:
        print('Error ingestion in : \n',
sys.exc_info())
    else:
        print('Success ingestion data')
        print("--- %s seconds ---" % (time.time() -
start_time))

```

Kemudian Jalankan *Main.py* dengan perintah \$ Python3 (path lokasi main.py mahasiswa regis) dapat dilihat pada gambar 3.30.

```

tananugrahr@instance-metabase:~$ python3 /home/tananugrahr/coding/fakta_mahasiswa/main.py
   n_mb      id_pmb kode_agama  id_prodi  id_fakultas kode_kewarganegaraan  id_prov  id_tahun  umur
0   1 1831200002    <NA>       5.0       1.0           1  20.0    NaN  NaN
1   1 1833130006    <NA>       7.0       1.0           1  13.0    NaN  NaN
2   1 1931190005    <NA>       5.0       1.0           1  19.0    NaN  NaN
3   1 1936130003    <NA>       9.0       1.0           1  13.0    NaN  NaN
4   1 2026140011    <NA>      10.0       2.0           1  14.0    NaN  NaN
...
...  ...  ...  ...
3026  1 1841310002     1       8.0       4.0           1  32.0    8.0  20.0
3027  1 1841320001     3       8.0       4.0           1  33.0    8.0  18.0
3028  1 1841330001     2       8.0       4.0           1  34.0    8.0  19.0
3029  1 1841330002     2       8.0       4.0           1  34.0    8.0  21.0
3030  1 1841350001     2       8.0       4.0           1  24.0    8.0  18.0
[3031 rows x 9 columns]
dataframe info after prepare:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3031 entries, 0 to 3030
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
_____
 0   n_mb              3031 non-null   Int64  
 1   id_pmb            3031 non-null   string 
 2   kode_agama         3026 non-null   string 
 3   id_prodi          3031 non-null   float64
 4   id_fakultas       3031 non-null   float64
 5   kode_kewarganegaraan 3031 non-null   string 
 6   id_prov            2990 non-null   float64
 7   id_tahun           3026 non-null   float64
 8   umur              2896 non-null   float64
dtypes: Int64(1), float64(5), string(3)
memory usage: 216.2 KB
None
success
state 0 clear

Success ingestion data
--- 0.4033217430114746 seconds ---
tananugrahr@instance-metabase:~$ █

```

Gambar 3.30 Run Main.py Fakta Mahasiswa

e. Data Warehouse

Setelah melakukan proses *ETL* pada tahapan *coding preparation*, data *warehouse* PMB calon mahasiswa dan mahasiswa regis siap untuk digunakan pada proses visualisasi. Berikut adalah tampilan data *warehouse* yang sudah dibangun:

1. Data Warehouse Calon Mahasiswa

Setelah melakukan proses *load* maka data akan tampil pada skema *sandbox_bi*. Data tabel fakta calon mahasiswa dapat dilihat pada gambar 3.31.

	n_pmb	id_pmb	id_tahun	kode_agama	kode_kewarganegaraan	id_prodi	id_fakultas
1	1	2111010001	16	2	1		1
2	1	2111020003	16	2	1		1
3	1	2111020004	16	3	1		1
4	1	2111020006	16	1	1		1
5	1	2111020007	16	2	1		1
6	1	2111020008	16	1	1		1
7	1	2111020010	16	1	1		1
8	1	2111020011	16	2	1		1
9	1	2111020012	16	2	1		1
10	1	2111020013	16	2	1		1
11	1	2111030002	16	1	1		1
12	1	2111040001	16	2	1		1
13	1	2111040003	16	2	1		1
14	1	2111050001	16	1	1		1
15	1	2111060003	16	4	1		1
16	1	2111060004	16	1	1		1
17	1	2111080002	16	1	1		1
18	1	2111090001	16	1	1		1

Gambar 3.31 Tampilan tabel fakta calon mahasiswa

2. Data Warehouse Mahasiswa

Setelah melakukan proses *load* maka data akan tampil diskema *sandbox_bi*. Data tabel fakta mahasiswa dapat dilihat pada gambar 3.32.

	n_mb	id_pmb	kode_agama	id_prodi	id_fakultas	kode_kewarganegaraan	id_prov	id_tahun	umur
1	1	1831200002	<null>	5	1	1	20	<null>	<null>
2	1	1833130006	<null>	7	1	1	13	<null>	<null>
3	1	1931190005	<null>	5	1	1	19	<null>	<null>
4	1	1936130003	<null>	9	1	1	13	<null>	<null>
5	1	2026140011	<null>	10	2	1	14	<null>	<null>
6	1	2111010001	2	1	3	1	1	16	18
7	1	2111020003	2	1	3	1	2	16	23
8	1	2111020004	3	1	3	1	2	16	18
9	1	2111020006	1	1	3	1	2	16	17
10	1	2111020007	2	1	3	1	2	16	18
11	1	2111020008	1	1	3	1	2	16	18
12	1	2111020010	1	1	3	1	2	16	21
13	1	2111020013	2	1	3	1	2	16	25
14	1	2111030002	1	1	3	1	3	16	18
15	1	2111040003	2	1	3	1	4	16	19
16	1	2111050001	1	1	3	1	5	16	20
17	1	2111060003	4	1	3	1	6	16	17
18	1	2111060004	1	1	3	1	6	16	17

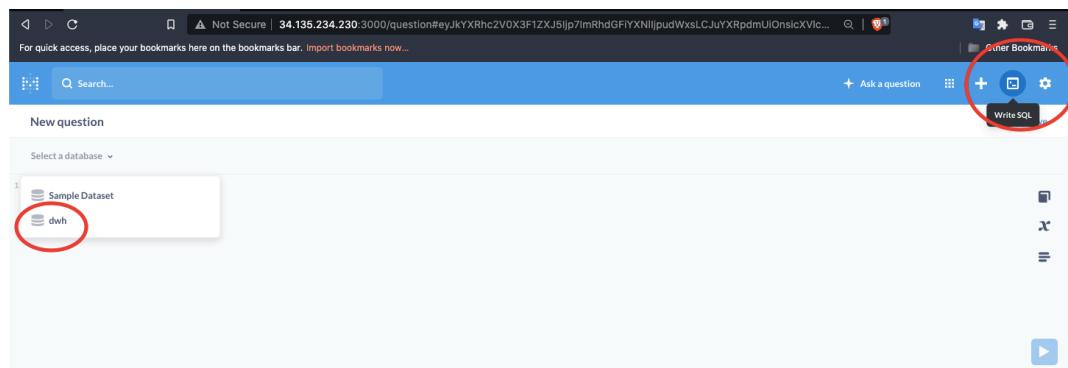
Gambar 3.32 Tampilan tabel fakta mahasiswa

f. Visualisasi

Setelah selasai membagun *data warehouse* penerimaan mahasiswa baru Universitas Janabadra, kemudian dilanjut ke proses visualisasi. Pada proses visualisasi ini menyajikan data yang telah diolah pada *data warehouse* menjadi bentuk gambar, diagram atau animasi untuk mempermudah dalam pembacaan suatu informasi. Pada proses visualisasi ini menggunakan *metabase*.

1. Membuat *Question* baru

Pada proses ini membuat *question* baru untuk nantinya ditampilkan pada halaman *dashboard*. Pada halaman *metabase* klik *write sql* kemudian pilih database yang akan digunakan. Dapat dilihat pada gambar 3.33.



Gambar 3.33 *Create Question*

2. *Create Question* Total calon mahasiswa PMB Universitas Janabadra lima tahun terakhir.

Pada tampilan ini dibangun dengan query:

```
select sum(n_pmb)
from sandbox_bi.fakta_calon_mahasiswa
pada Pada tampilan ini jenis visualisasi menggunakan number.
Menampilkan total keseluruhan calon mahasiswa yang mendaftar
pmb diuniversitas janabadra. Dapat dilihat pada gambar 3.34.
```



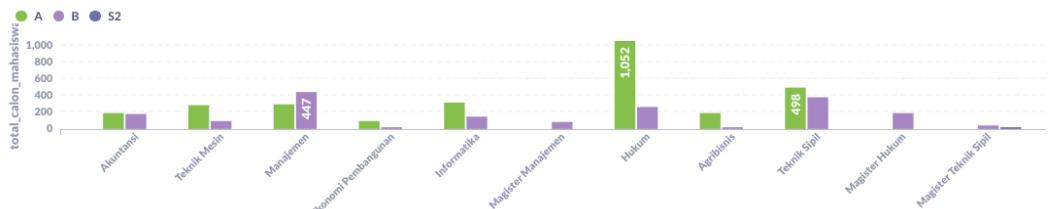
Gambar 3.34 Total calon mahasiswa PMB

3. *Create Question* Jumlah Calon Mahasiswa PMB Universitas Janabadra Berdasarkan Program Studi.

Pada tampilan ini dibangun dengan *query*:

```
select sum(n_pmb) total_calon_mahasiswa,
m_prodi.prodi,
dim_waktu.tahun_akademik,
mhs_cln.program
from sandbox_bi.fakta_calon_mahasiswa
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_calon_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
left join public.mhs_cln on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.mhs_cln.id_pmb
where 1=1
[[and{{prodi}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4
```

Pada tampilan visualisasi ini menggunakan *bar chart*. Pada visualisasi ini menampilkan jumlah pendaftar berdasarkan program studi dan kelas. Dapat dilihat pada gambar 3.35



Gambar 3.35 Jumlah Calon Mahasiswa Berdasarkan Program Studi dan Kelas

4. *Create Question* Jumlah Calon Mahasiswa PMB Universitas Janabadra Berdasarkan Tahun Akademik dan Semester.

Jumlah Calon Mahasiswa Berdasarkan Tahun Akademik Pada tampilan ini dibangun dengan *query* :

```
select sum(n_pmb) total_calon_mahasiswa,
public.m_prodi.prodi,
public.m_fakultas.fakultas,
public.dim_waktu.tahun_akademik
from sandbox_bi.fakta_calon_mahasiswa
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_calon_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.dim_waktu on
fakta_calon_mahasiswa.id_pmb = public.dim_waktu.id_pmb
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4
order by tahun_akademik asc
```

Pada tampilan ini jenis visualisasinya menggunakan *line chart*. Pada visualisasi ini menampilkan total calon mahasiswa yang mendaftar PMB berdasarkan tahun akademik dan dapat difilter pada program studi dan fakultas. Untuk *chart* jumlah pendaftar berdasarkan tahun akademik dapat dilihat pada gambar 3.36.



Gambar 3.36 Jumlah Calon Mahasiswa PMB UJB Berdasarkan Tahun Akademik dan Semester

5. Jumlah Calon Mahasiswa Berdasarkan Tahun Akademik Tiap Prodi

Pada tampilan ini dibangun dengan *query* :

```
select distinct
sum(n_pmb) ,
public.m_prodi.prodi,
public.m_fakultas.fakultas,
public.dim_waktu.tahun_akademik
from sandbox_bi.fakta_calon_mahasiswa
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_calon_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by n_pmb,prodi,3,4
```

Pada tampilan visualisasi ini menggunakan tampilan *bar chart*. Pada visualisasi ini menampilkan jumlah pendaftar tiap Prodi yang dapat difilter berdasarkan tahun akademik. dapat dilihat pada gambar 3.37.



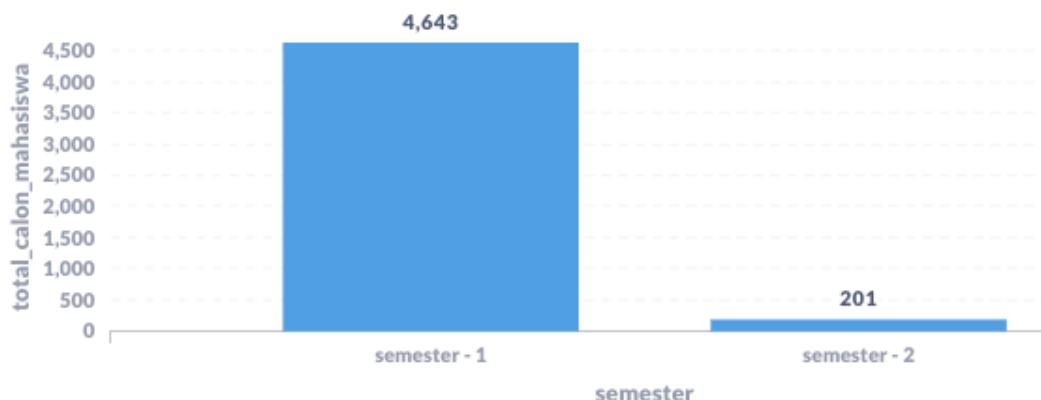
Gambar 3.37 Tampilan Jumlah pendaftar berdasarkan tahun akademik tiap Prodi

6. Jumlah Calon Mahasiswa Berdasarkan Semester

Pada tampilan ini dibangun dengan *query* :

```
select sum(n_pmb) total_calon_mahasiswa,
m_prodi.prodi,
m_tahun.semester,
dim_waktu.tahun_akademik
from sandbox_bi.fakta_calon_mahasiswa
left join public.m_tahun on
sandbox_bi.fakta_calon_mahasiswa.id_tahun =
public.m_tahun.id_tahun
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
where 1=1
[[and{{prodi}}]]
[[and{{tahun_akademik}}]]
[[and{{fakultas}}]]
group by 2,3,4
```

Pada tampilan visualisasi ini menggunakan tampilan *bar chart*. Pada visualisasi ini menampilkan jumlah pendaftar berdasarkan semester di Universitas Janabadra. Tampilan *question* dapat dilihat pada gambar 3.38.



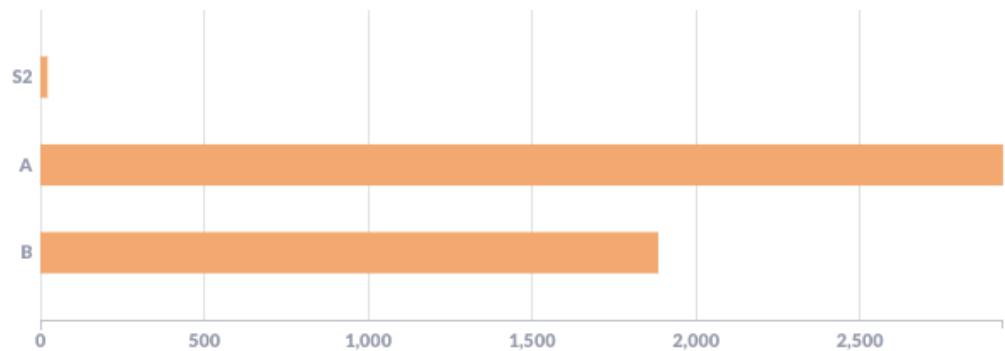
Gambar 3.38 Tampilan jumlah pendaftar berdasarkan semester

7. Jumlah Calon Mahasiswa PMB Universitas Janabadra Berdasarkan Program Kelas.

Pada tampilan ini dibangun dengan *query* :

```
select sum(n_pmb) total_calon_mahasiswa,
m_prodi.prodi,
mhs_cln.program,
dim_waktu.tahun_akademik
from sandbox_bi.fakta_calon_mahasiswa
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.mhs_cln on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.mhs_cln.id_pmb
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
where 1=1
[[and{{prodi}}]]
[[and{{tahun_akademik}}]]
[[and{{program}}]]
group by 2,3,4
```

Pada tampilan visualisasi ini menggunakan tampilan *row chart*. Pada visualisasi ini menampilkan jumlah calon mahasiswa berdasarkan program kelas di Universitas Janabadra. Kelas A merupakan kelas regular, kelas B merupakan kelas malam dan S2 merupakan program magister. *Question* dapat dilihat pada gambar 3.39



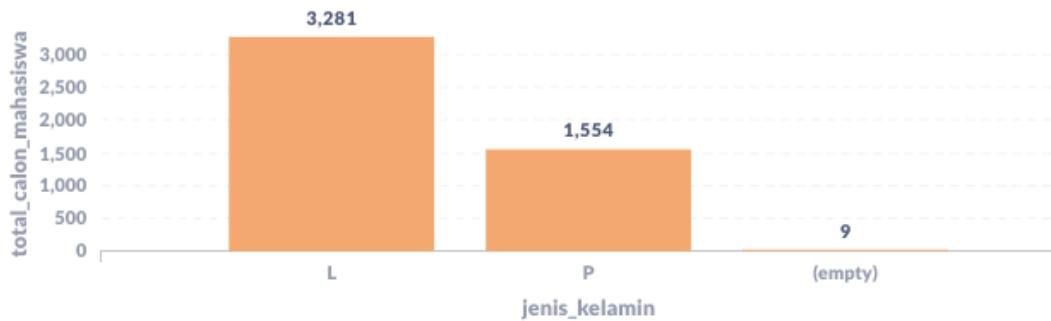
Gambar 3.39 Tampilan Jumlah pendaftar berdasarkan kelas

8. Jumlah Calon Mahasiswa PMB Universitas Janabadra Berdasarkan Jenis kelamin

Jumlah Calon Mahasiswa Berdasarkan Jenis Kelamin Pada tampilan ini dibangun dengan *query* :

```
select sum(n_pmb) total_calon_mahasiswa,
mhs_cln.jenis_kelamin,
m_prodi.prodi,
m_fakultas.fakultas,
dim_waktu.tahun_akademik
from sandbox_bi.fakta_calon_mahasiswa
left join public.mhs_cln on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.mhs_cln.id_pmb
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_calon_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
```

Pada tampilan visualisasi ini menggunakan *bar chart*. Pada visualisasi ini menampilkan jumlah calon mahasiswa berdasarkan jenis kelamin. *Chart* calon mahasiswa berdasarkan jenis kelamin dapat dilihat pada gambar 3.40.



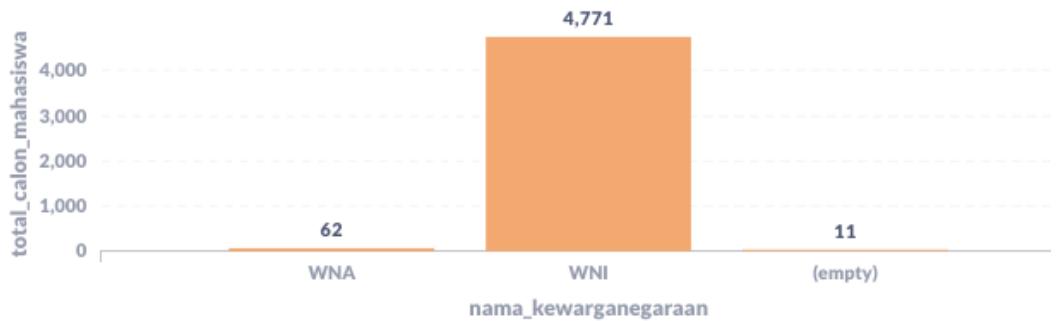
Gambar 3.40 Tampilan jumlah calon mahasiswa berdasarkan jenis kelamin

9. Jumlah Calon Mahasiswa Berdasarkan Kewarganegaraan

Pada tampilan ini dibangun dengan *query* :

```
select sum(n_pmb) total_calon_mahasiswa,
public.m_kewarganegaraan.nama_kewarganegaraan,
public.dim_waktu.tahun_akademik,
public.m_prodi.prodi,
public.m_fakultas.fakultas
from sandbox_bi.fakta_calon_mahasiswa
left join public.m_kewarganegaraan on
sandbox_bi.fakta_calon_mahasiswa.kode_kewarganegaraan
= public.m_kewarganegaraan.kode_kewarganegaraan
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_calon_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
where 1=1
[[and{{tahun_akademik}}]]
[[and{{prodi}}]]
[[and{{fakultas}}]]
group by 2,3,4,5
```

Pada tampilan visualisasi ini menggunakan *bar chart*. Pada visualisasi ini menampilkan jumlah calon mahasiswa berdasarkan status kewarganegaraan. Tampilan dapat dilihat pada gambar 3.41.



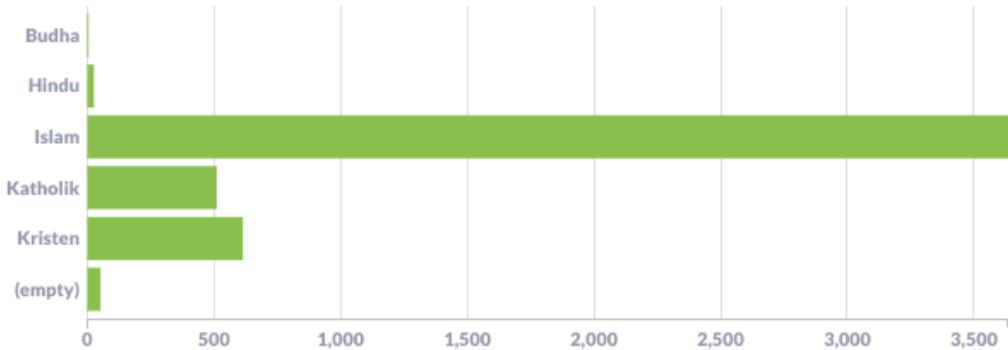
Gambar 3.41 Tampilan jumlah calon mahasiswa berdasarkan status kewarganegaraan

10. Jumlah Calon Mahasiswa Berdasarkan Agama

Pada tampilan ini dibangun dengan *query* :

```
select sum(n_pmb) total_calon_mahasiswa,
public.m_prodi.prodi,public.m_fakultas.fakultas,
public.m_agama.nama_agama,
public.dim_waktu.tahun_akademik
from sandbox_bi.fakta_calon_mahasiswa
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_calon_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_agama on
sandbox_bi.fakta_calon_mahasiswa.kode_agama =
public.m_agama.kode_agama
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
```

Pada tampilan visualisasi ini menggunakan *row chart*. Pada visualisasi ini menampilkan jumlah calon mahasiswa berdasarkan agama yang dianut. *Chart* calon mahasiswa berdasarkan agama yang dianut dapat dilihat pada gambar 3.42.



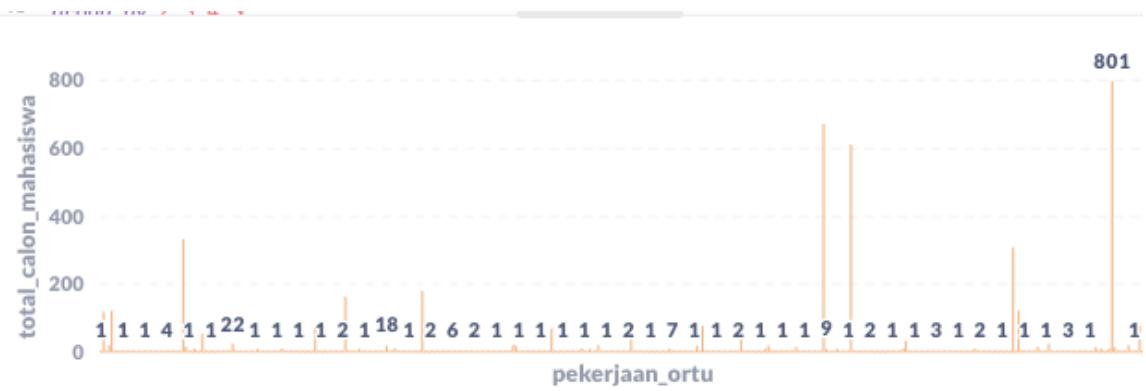
Gambar 3.42 Tampilan jumlah calon mahasiswa berdasarkan agama yang dianaut

11. Jumlah calon mahasiswa PMB Universitas Janabadra berdasarkan data keluarga (pekerjaan orang tua)

Pada tampilan ini dibangun dengan *query* :

```
select sum(n_pmb) total_calon_mahasiswa,
UPPER(mhs_cln.pekerjaan) as pekerjaan_ortu,
m_prodi.prodi,
m_fakultas.fakultas,
dim_waktu.tahun_akademik
from sandbox_bi.fakta_calon_mahasiswa
left join public.mhs_cln on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.mhs_cln.id_pmb
left join public.m_prodi on
sandbox_bi.fakta_calon_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_calon_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
```

Pada tampilan visualisasi ini menggunakan *bar chart*. Pada visualisasi ini menampilkan calon mahasiswa PMB berdasarkan program pekerjaan orang tua. Untuk visualisasinya dapat dilihat pada gambar 3.43.



Gambar 3.43 Tampilan jumlah calon mahasiswa berdasarkan pekerjaan orang tua

12. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra

Pada tampilan ini dibangun dengan *query*:

```
select sum(n_mb) tota_mahasiswa_regis
from sandbox_bi.fakta_mahasiswa
```

Pada tampilan ini jenis visualisasi menggunakan *number* yang dimana pada visualisasi ini hanya menampilkan total mahasiswa registrasi yang diterima pada PMB Universitas Janabadra pada tahun 2018 hingga 2021. Untuk visualisasinya dapat dilihat pada gambar 3.44.



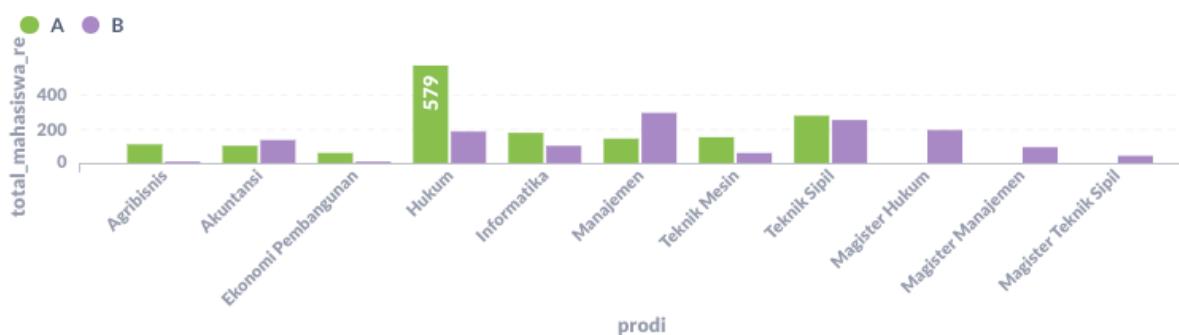
Gambar 3.44 Tampilan jumlah mahasiswa registrasi

13. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra Berdasarkan Program Studi.

Jumlah Mahasiswa Registrasi Berdasarkan Program Studi dan Kelas Pada tampilan ini dibangun dengan *query*:

```
select sum(n_mb) total_mahasiswa_regis,
public.m_prodi.prodi,
public.m_fakultas.fakultas,
public.m_tahun.tahun_akademik,
public.mhs_regis.program
from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
order by program asc
```

Pada visualisasi mahasiswa registrasi berdasarkan program studi dan kelas menggunakan *bar chart* pada bar ini warna biru diartikan sebagai kelas A dan warna hijau sebagai kelas B. Gambar visualisasinya dapat dilihat pada gambar 3.45.



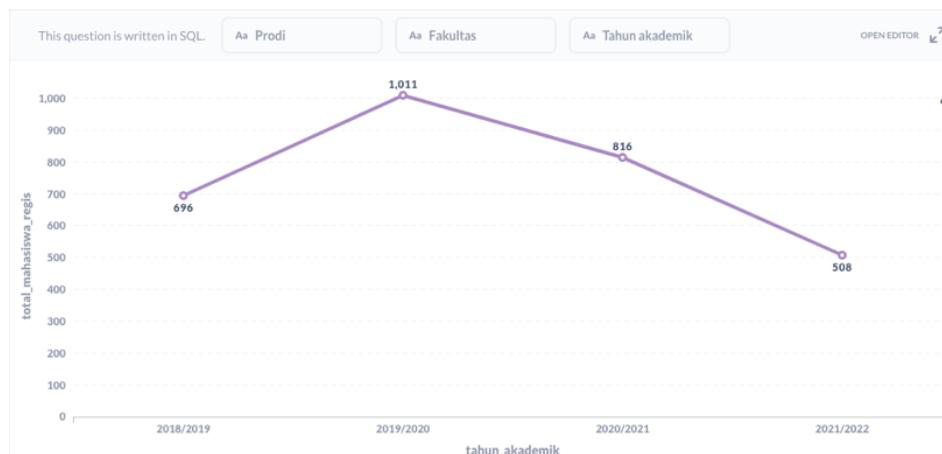
Gambar 3.45 Tampilan jumlah mahasiswa registrasi berdasarkan program studi dan kelas

14. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra Berdasarkan Tahun Akademik.

Pada tampilan ini dibangun dengan *query*:

```
select sum(n_mb) total_mahasiswa_regis,
public.m_prodi.prodi,public.m_fakultas.fakultas,
fakta_mahasiswa.id_tahun,public.m_tahun.tahun_akademik
from sandbox_bi.fakta_mahasiswa
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.mhs_cln on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_cln.id_pmb
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
order by tahun_akademik asc
```

Pada tampilan ini jenis visualisasi menggunakan *line chart* yang dimana pada visualisasi ini hanya menampilkan total mahasiswa registrasi yang diterima pada PMB Universitas Janabadra dari tiap tahun akademik. Untuk visualisasinya dapat dilihat pada gambar 3.46.



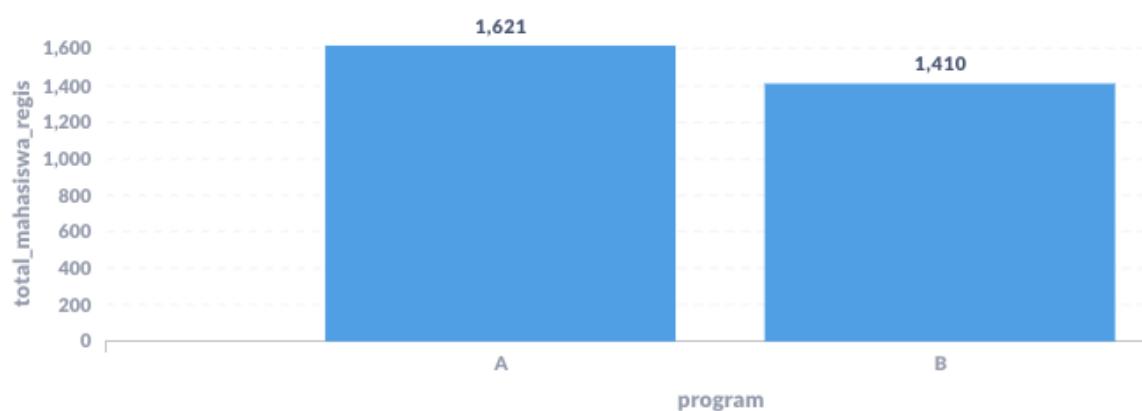
Gambar 3.46 Tampilan jumlah mahasiswa registrasi berdasarkan tahun akademik

15. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra Berdasarkan Program Kelas

Jumlah Mahasiswa Registrasi Berdasarkan Kelas Pada tampilan ini dibangun dengan *query*:

```
select sum(n_mb) total_mahasiswa_regis,
public.mhs_regis.program,public.m_prodi.prodi,
public.m_fakultas.fakultas,public.m_tahun.tahun_akademik
from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran left join
public.m_prodi on sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi left join public.m_fakultas
on sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
```

Pada visualisasi mahasiswa registrasi berdasarkan kelas menggunakan *bar chart* pada visualisasi ini dapat dilihat mahasiswa pada tahun akademik 2021/2022 pada Prodi Informatika jumlah mahasiswa kelas A berjumlah 38 orang dan kelas B berjumlah 13 orang. Untuk gambar visualisasinya dapat dilihat pada gambar 3.47.



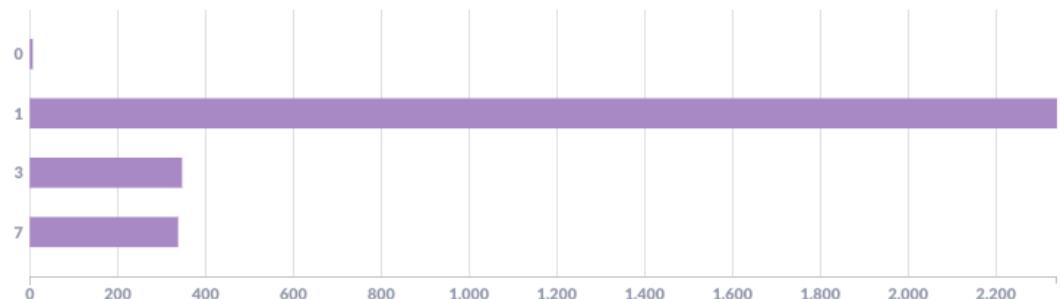
Gambar 3.47 Tampilan jumlah mahasiswa registrasi berdasarkan kelas

16. Jumlah Mahasiswa Registrasi Berdasarkan Jalur Pmb

Pada tampilan ini dibangun dengan *query*:

```
select sum(n_mb) total_mahasiswa_regis,
public.mhs_regis.jalur_pmb, public.m_prodi.prodi,
public.m_fakultas.fakultas, public.m_tahun.tahun_akademik
from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
order by jalur_pmb asc
```

Pada visualisasi ini menggunakan *row chart* menampilkan jumlah mahasiswa registrasi berdasarkan jalur PMB Teknik Informatika tahun akademik 2021/2022 Untuk tampilan Visualisasinya dapat dilihat pada gambar 3.48.



Gambar 3.48 Tampilan jumlah mahasiswa registrasi berdasarkan jalur PMB

17. Jumlah Mahasiswa Registrasi Berdasarkan Alih Jenjang

Pada tampilan ini dibangun dengan *query*:

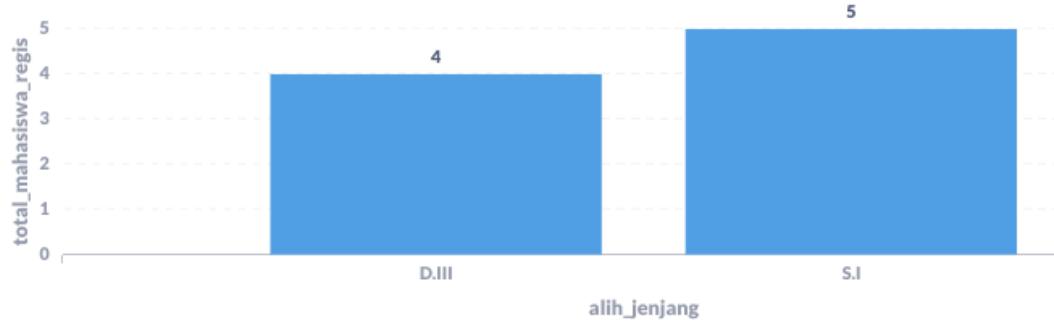
```
select sum(n_mb) total_mahasiswa_regis,
public.mhs_regis.jenjang as alih_jenjang,
public.m_prodi.prodi, public.m_fakultas.fakultas,
public.m_tahun.tahun_akademik
```

```

from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran left join
public.m_prodi on sandbox_bi.fakta_mahasiswa.id_prodi
= public.m_prodi.id_prodi left join public.m_fakultas
on sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas left join public.m_tahun
on sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]] and
jenjang != '' and
jenjang IS NOT null
group by 2,3,4,5

```

Pada visualisasi mahasiswa alih jenjang menggunakan *bar chart* yang dimana pada visualisasi ini hanya menampilkan total mahasiswa registrasi alih jenjang yang diterima pada PMB Universitas Janabadra dari tiap tahun akademik. Untuk visualisasinya dapat dilihat pada gambar 3.49.



Gambar 3.49 Tampilan jumlah mahasiswa registrasi berdasarkan alih jenjang

18. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra Berdasarkan Jenis Kelamin

Jumlah Mahasiswa Registrasi Berdasarkan Jenis Kelamin Pada tampilan ini dibangun dengan *query*:

```

select sum(n_mb) tota_mahasiswa_regis,
public.mhs_regis.jenis_kelamin,public.m_prodi.prodi,
public.m_fakultas.fakultas,public.m_tahun.tahun_akadem
ik from sandbox_bi.fakta_mahasiswa

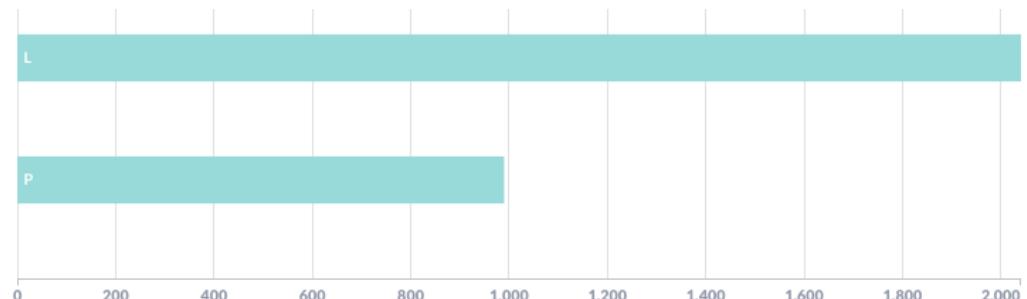
```

```

left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5

```

Pada visualisasi mahasiswa registrasi berdasarkan jenis kelamin menggunakan *row chart* pada bar ini jenis kelamin yang dapat difilter berdasarkan program studi , fakultas dan tahun akademik. Gambar visualisasinya dapat dilihat pada gambar 3.50.



Gambar 3.50 Tampilan jumlah mahasiswa registrasi berdasarkan jenis kelamin

19. Jumlah Mahasiswa Registrasi Berdasarkan Status Kewarganegaraan

Pada tampilan ini dibangun dengan *query*:

```

select sum(n_mb) total_mahasiswa_regis,
public.m_kewarganegaraan.nama_kewarganegaraan,
public.m_prodi.prodi,
public.m_fakultas.fakultas,
public.m_tahun.tahun_akademik,
public.adm_provinsi.nama_provinsi
from sandbox_bi.fakta_mahasiswa
left join public.m_kewarganegaraan on
sandbox_bi.fakta_mahasiswa.kode_kewarganegaraan =
public.m_kewarganegaraan.kode_kewarganegaraan

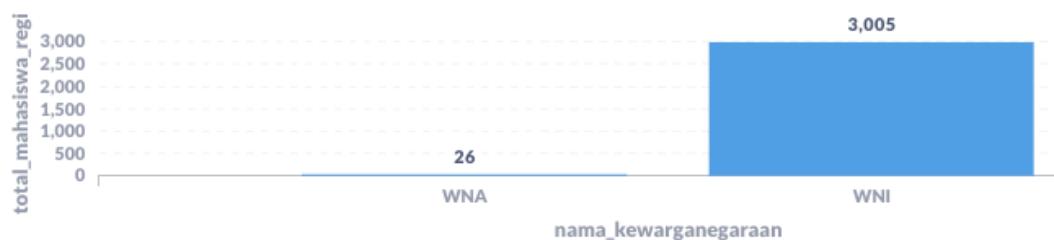
```

```

left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
left join public.adm_provinsi on
sandbox_bi.fakta_mahasiswa.id_prov =
public.adm_provinsi.id_prov
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
[[and{{nama_provinsi}}]]
group by 2,3,4,5,6

```

Pada tampilan visualisasi ini menggunakan *bar chart* menampilkan status kewarganegaraan. Visualisasinya dapat dilihat pada gambar 3.51.



Gambar 3.51 Tampilan jumlah mahasiswa registrasi status kewarganegaraan

20. Jumlah Mahasiswa Registrasi Berdasarkan Jurusan SLTA

Pada tampilan ini dibangun dengan *query*:

```

select sum(n_mb) total_mahasiswa_regis,
UPPER(public.mhs_regis.jurusanslta) AS jurusanslta,
public.m_prodi.prodi,public.m_fakultas.fakultas,
public.m_tahun.tahun_akademik
from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas

```

```

left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]][[and{{fakultas}}]][[and{{tahun_akademik}}]]group by 2,3,4,5

```

Pada visualisasi ini menggunakan *bar chart* menampilkan jumlah mahasiswa registrasi berdasarkan jurusan SLTA Visualisasinya dapat dilihat pada gambar 3.52.



Gambar 3.52 Tampilan jumlah mahasiswa registrasi berdasarkan jurusan SLTA

21. Jumlah Mahasiswa Registrasi Berdasarkan Provinsi

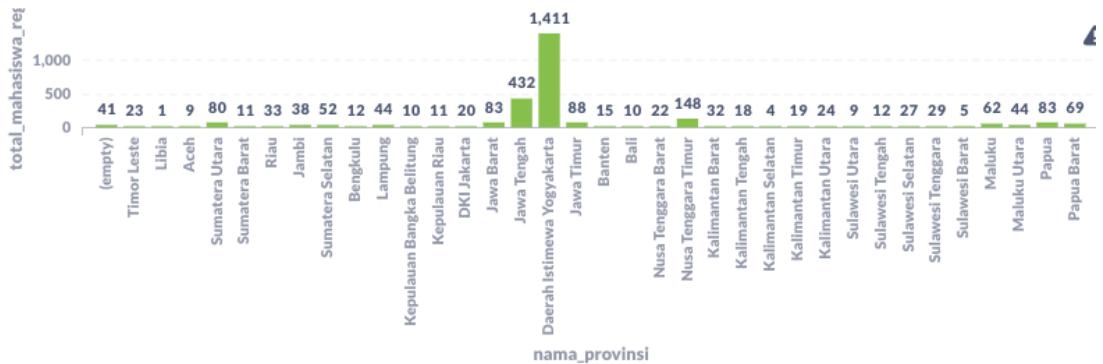
Pada tampilan ini dibangun dengan *query*:

```

select sum(n_mb) total_mahasiswa_regis,
public.mhs_regis.prov_asal,public.adm_provinsi.nama_provinsi,public.m_prodi.prodi,public.m_fakultas.fakultas,public.m_tahun.tahun_akademik
from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb = public.mhs_regis.no_pendaftaran
left join public.adm_provinsi on
sandbox_bi.fakta_mahasiswa.id_prov = public.adm_provinsi.id_prov
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi = public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas = public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun = public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5,6

```

Pada visualisasi ini menggunakan *bar chart* menampilkan mahasiswa registrasi berdasarkan asal provinsi. Visualisasinya dapat dilihat pada gambar 3.53.



Gambar 3.53 Tampilan jumlah mahasiswa registrasi berdasarkan provinsi

22. Jumlah mahasiswa registrasi PMB Universitas Janabadra berdasarkan data keluarga Berdasarkan Pendidikan Orang Tua

Pada tampilan ini dibangun dengan *query*:

```
select sum(n_mb) total_mahasiswa_regis,
UPPER(public.mhs_regis.pendidikan_ayah) as
pendidikan_orang_tua,public.m_prodi.prodi,
public.m_fakultas.fakultas,
public.m_tahun.tahun_akademik
from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
```

Pada tampilan visualisasi ini menggunakan *bar chart*. Pada visualisasi ini menampilkan calon mahasiswa PMB berdasarkan

program pendidikan orang tua. *Chart* dapat dilihat pada gambar 3.54.



Gambar 3.54 Tampilan jumlah mahasiswa registrasi berdasarkan Pendidikan orang tua

23. Jumlah Mahasiswa Registrasi Berdasarkan Pendapatan Orang Tua

Pada tampilan ini dibangun dengan *query*:

```
select sum(n_mb) total_mahasiswa_regis,
public.mhs_regis.penghasilan,
public.m_prodi.prodi,
public.m_fakultas.fakultas,
public.m_tahun.tahun_akademik
from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
order by penghasilan desc
```

Pada visualisasi ini menggunakan *bar chart* menampilkan pendapatan orang tua. Visualisasinya dapat dilihat pada gambar 3.55.



Gambar 3.55 Tampilan jumlah mahasiswa registrasi berdasarkan pendapatan orang tua

24. Jumlah Mahasiswa Registrasi Berdasarkan Pekerjaan Orang tua

Pada tampilan ini dibangun dengan *query*:

```
select sum(n_mb) total_mahasiswa_regis,
public.mhs_regis.pekerjaan_ayah,
public.m_prodi.prodi,
public.m_fakultas.fakultas,
public.m_tahun.tahun_akademik
from sandbox_bi.fakta_mahasiswa
left join public.mhs_regis on
sandbox_bi.fakta_mahasiswa.id_pmb =
public.mhs_regis.no_pendaftaran
left join public.m_prodi on
sandbox_bi.fakta_mahasiswa.id_prodi =
public.m_prodi.id_prodi
left join public.m_fakultas on
sandbox_bi.fakta_mahasiswa.id_fakultas =
public.m_fakultas.id_fakultas
left join public.m_tahun on
sandbox_bi.fakta_mahasiswa.id_tahun =
public.m_tahun.id_tahun
where 1=1
[[and{{prodi}}]]
[[and{{fakultas}}]]
[[and{{tahun_akademik}}]]
group by 2,3,4,5
```

Pada visualisasi ini menggunakan *bar chart* menampilkan pekerjaan orang tua yaitu ayah yang dapat difilter berdasarkan program studi, fakultas dan tahun akademik. Visualisasinya dapat dilihat pada gambar 3.56.



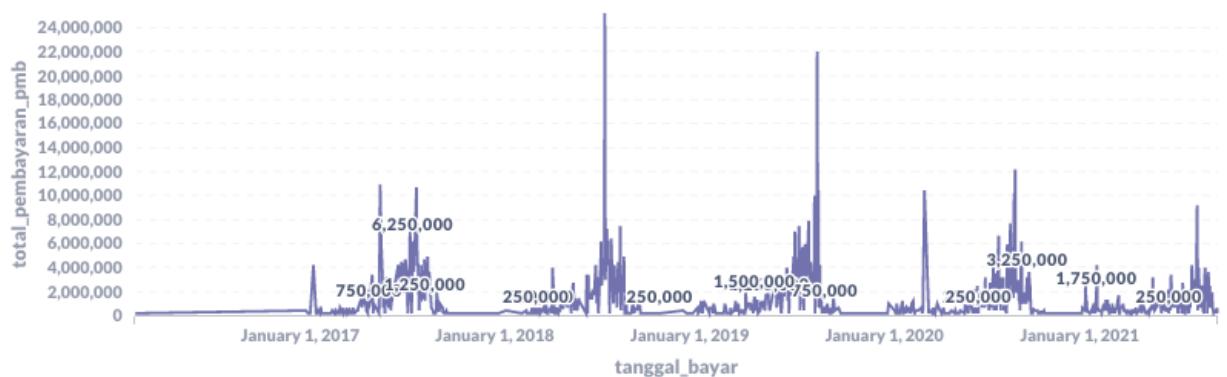
Gambar 3.56 Tampilan jumlah mahasiswa berdasarkan pekerjaan orang tua

25. Jumlah Pendapatan Pendaftaran PMB Berdasarkan Tanggal Bayar

Pada tampilan ini dibangun dengan *query*:

```
select
sum(n_pmb) total_calon_mahasiswa, sum(n_pmb)*250000
total_pembayaran_pmb,public.dim_waktu.tanggal_bayar
from sandbox_bi.fakta_calon_mahasiswa
left join public.dim_waktu on
sandbox_bi.fakta_calon_mahasiswa.id_pmb =
public.dim_waktu.id_pmb
where 1=1
[[and {{tanggal_bayar}}]]
group by 3
```

Pada tampilan pendapatan berdasarkan tanggal bayar menggunakan *line chart*. Pada tampilan visualisasi ini menampilkan total pembayaran per-hari yang dapat ditampilkan berdasarkan periode tanggal yang diinginkan.Untuk gambar pendapatan berdasarkan tanggal bayar dapat dilihat pada gambar 3.57.



Gambar 3.57 Tampilan jumlah Pendapatan berdasarkan tanggal bayar

26. Menambahkan Filter Pada *Question*

Pada proses penambahan filter pada *question* menggunakan fungsi

Where 1=1

```
[[and{{kategori filter yang akan ditambahkan}}]]
```

Contoh kasus pada jumlah mahasiswa per tahun akademik menambahkan filter provinsi dapat dilihat pada gambar 3.58

New question Started from Jumlah Mahasiswa per Tahun akademik Save

```

dwh v
Aa Prodi Aa Fakultas Aa Tahun akademik
Aa Nama provinsi D

1 select sum(n_mb) total_mahasiswa_regis,
2 public.m_prodi.prodi,
3 public.m_fakultas.fakultas,
4 fakta_mahasiswa.id_tahun,
5 public.m_tahun_tahun_akademik,
6 public.adm_provinsi.nama_provinsi B
7 from sandbox.bi.fakta_mahasiswa
8 left join public.m_prodi on sandbox.bi.fakta_mahasiswa.id_prodi = public.m_prodi.id_prodi
9 left join public.m_fakultas on sandbox.bi.fakta_mahasiswa.id_fakultas = public.m_fakultas.id_fakultas
10 left join public.mhs_cln on sandbox.bi.fakta_mahasiswa.id_pmb = public.mhs_cln.id_pmb
11 left join public.m_tahun on sandbox.bi.fakta_mahasiswa.id_tahun = public.m_tahun.id_tahun
12 left join public.adm_provinsi on sandbox.bi.fakta_mahasiswa.id_prov = public.adm_provinsi.id_prov A
13 where 1=1
14 [[and{{prodi}}]]
15 [[and{{fakultas}}]]
16 [[and{{nama_provinsi}}]] C
17 [[and{{tahun_akademik}}]] D
18
19 group by 2,3,4,5,6
20 order by tahun_akademik asc
21

```

Gambar 3.58 Menambahkan Filter Pada *Question*

Tahapan yang harus dilakukan:

- Menghubungkan tabel fakta dan tabel dimensi provinsi dengan

klausa *left join* dengan *syntax*:

```
Left join public.adm_provinsi on
sandbox.bi.fakta_mahasiswa.id_prov =
public.adm_provinsi.id_prov
```

- Menambahkan *colom* yang akan ditampilkan dengan *syntax*:

```
Public.adm_provinsi.nama_provinsi
```

- Menambahkan filter yang akan digunakan dengan *syntax*:

```
[[and{{nama_provinsi}}]]
```

- Tampilan filter akan muncul pada bagian atas halaman editor

Set variable :

- Variable type = Field Filter*

- Field to Map to = public → adm_provinsi → nama_provinsi*

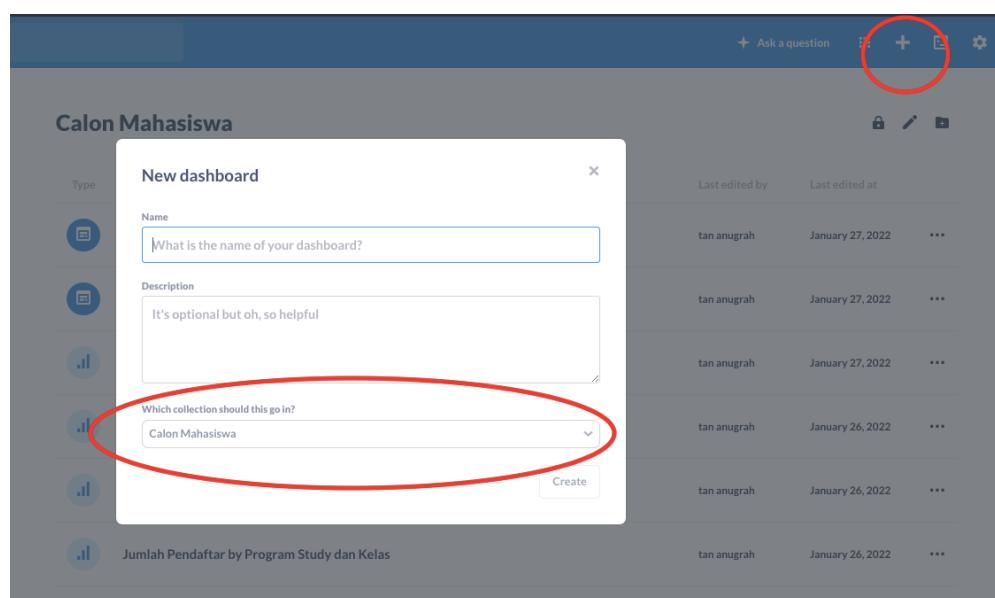
- 3) Filter widget type = Category
- 4) Filter widget label = nama provinsi

27. Implementasi Dashboard

Untuk pembangunan *dashboard* memiliki beberapa tahap dari pembuatan halaman *dashboard*, penambahanan *question* pada *dashboard* dan pembuatan filter berikut tahapan pembangunan *dashboard* untuk penerimaan mahasiswa baru Universitas Janabadra:

1) Halaman *Dashboard* Baru

Untuk membuat dashboard baru dan menyusun question yang telah dibangun klik create pilih new dashboard selanjutnya isi nama , deskripsi tentang dashboard dan lokasi penyimpanan dashboard selanjutnya klik create. Dapat dilihat pada gambar 3.59.

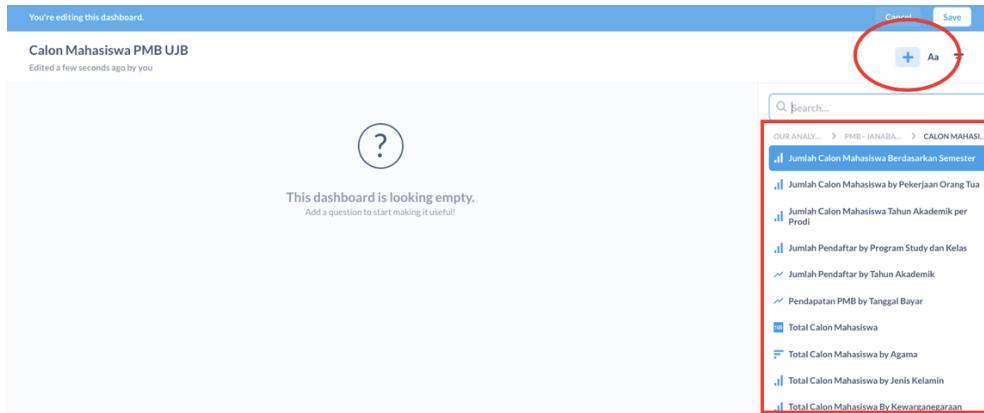


Gambar 3.59 New Dashboard

2) Menambahkan *Question*

Dashboard baru yang telah dibuat kemudian klik *add question* selanjutnya pilih *question* yang telah dibangun untuk dimasukkan kedalam halaman *dashboard*. Kemudian susun

question sesuai dengan yang diinginkan. Dapat dilihat pada gambar 3.60.



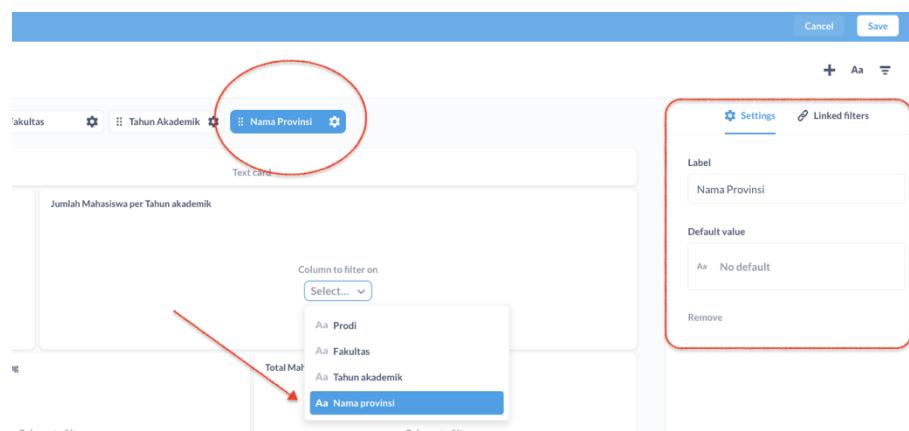
Gambar 3.60 *Add Question*

3) *Add Filter*

Setelah *dashboard* disusun sesuai dengan yang diinginkan selanjutnya menambahkan filter untuk menyeleksi tampilan sesuai dengan periode ataupun kategori yang diinginkan. Untuk menambahkan filter klik *add filter* dan pilih *other categories* untuk menambahkan filter berdasarkan kategori dan *time* untuk memfilter berdasarkan waktu. Contoh kasus untuk menambahkan filter pada *question* jumlah mahasiswa per tahun akademik seperti pada gambar 3.58. proses penambahan filter pada *dashboard* dapat dilihat pada gambar 3.61. Selanjutnya Isi nama label untuk filter yang akan dibuat dan pilih sesuai dengan kategori filter yang diinginkan pada contoh kasus penambahan filter nama provinsi pada pilihan *colom to filter on* pada *question* pilih nama provinsi dan klik *save* untuk menyimpan. Dapat dilihat pada gambar 3.62.

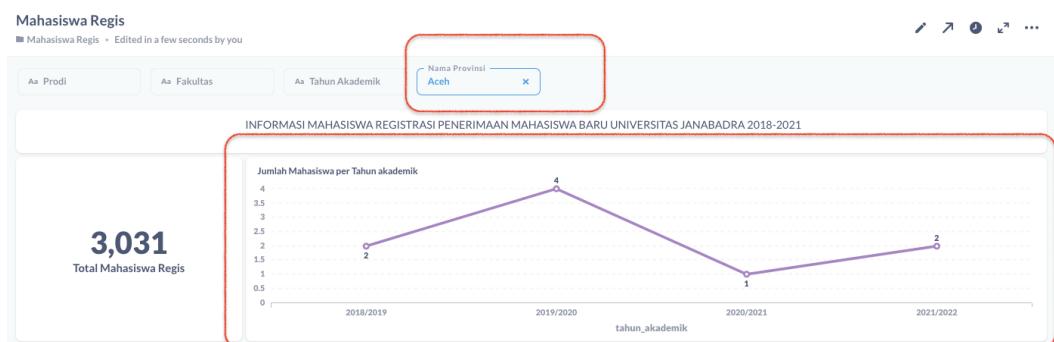


Gambar 3.61 Add Filter



Gambar 3.62 Set Filter

Selanjutnya tes fungsi filter yang ditambahakan pada *dashboard* mahasiswa. Contoh kasus menambahkan filter asal provinsi berasal dari aceh *pada question* jumlah mahasiswa per tahun akademik maka hasil akan filter dapat dilihat pada gambar 3.63.



Gambar 3.63 Tes Fungsi Filter

6. *Deployment*

Tahapan ini melakukan *functional test* terhadap *dashboard* yang telah dibangun berdasarkan *business case* yang telah dibuat.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Dashboard PMB Universitas Janabadra

Dengan mengimplementasikan metode *Business Intelligence Roadmap* telah berhasil membangun sebuah data *warehouse* yang digunakan untuk membangun *dashboard* visualisasi penerimaan mahasiswa baru Universitas Janabadra. *Dashboard* dibangun berjumlah empat halaman *dashboard* yang dapat difilter sesuai Program Studi, Fakultas dan tahun akademik.

1. Implementasi *Dashboard* Calon Mahasiswa

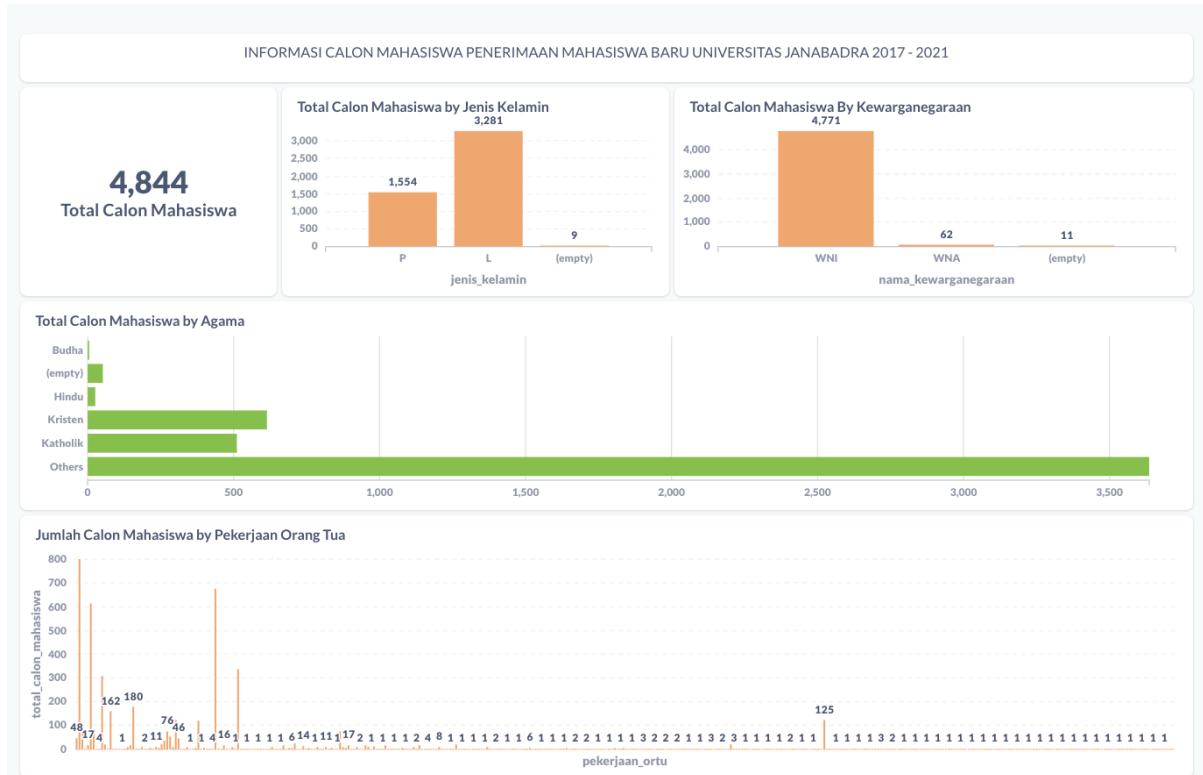
Pada tampilan *dashboard* calon mahasiswa menampilkan informasi data dari tahun 2017-2021. *Dashboard* calon mahasiswa menampilkan informasi calon mahasiswa yang dikelompokkan berdasarkan total keseluruhan pendaftar PMB, jumlah pendaftar berdasarkan tahun akademik, pendapatan dari pendaftaran PMB berdasarkan tanggal bayar, jumlah calon mahasiswa berdasarkan tahun akademik tiap program studi, jumlah pendaftar berdasarkan kelas, jumlah pendaftar PMB berdasarkan semester dan jumlah pendaftar PMB berdasarkan program studi dan kelas. Tampilan *dashboard* calon mahasiswa dapat dilihat pada gambar 4.1.



Gambar 4.1 *Dashboard* informasi calon mahasiswa PMB

2. Implementasi *Dashboard* Profil Calon Mahasiswa

Pada tampilan *dashboard* profil calon mahasiswa menampilkan informasi calon mahasiswa yang dikelompokkan berdasarkan jumlah calon mahasiswa berdasarkan jenis kelamin, jumlah calon mahasiswa berdasarkan status kewarganegaraan, jumlah calon mahasiswa berdasarkan agama yang dianut dan jumlah calon mahasiswa berdasarkan pekerjaan orang tua. Tampilan *dashboard* calon mahasiswa dapat dilihat pada gambar 4.2.



Gambar 4.2 *Dashboard* profil calon mahasiswa PMB Universitas Janabadra

3. Implementasi *Dashboard* Mahasiswa Registrasi

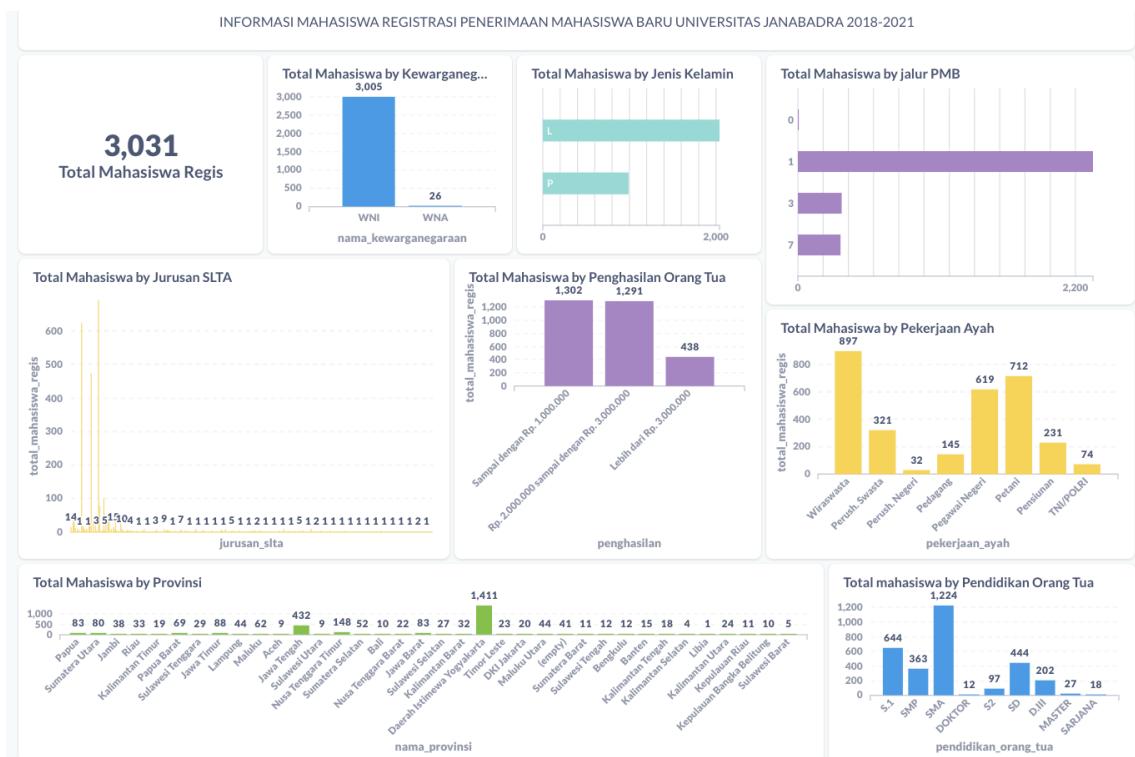
Pada tampilan *dashboard* mahasiswa registrasi menampilkan informasi mahasiswa yang telah diterima dan registrasi sebagai mahasiswa Universitas Janabadra yang dikelompokkan berdasarkan jumlah mahasiswa berdasarkan tahun akademik, jumlah mahasiswa berdasarkan alih jenjang, jumlah mahasiswa berdasarkan kelas dan jumlah mahasiswa berdasarkan program studi dan kelas. Tampilan *dashboard* calon mahasiswa dapat dilihat pada gambar 4.3.



Gambar 4.3 *Dashboard* mahasiswa registrasi PMB Universitas Janabadra

4. Implementasi *Dashboard* Profil Mahasiswa Registrasi

Pada tampilan *dashboard* profil mahasiswa registrasi menampilkan informasi mahasiswa yang telah diterima dan registrasi sebagai mahasiswa Universitas Janabadra yang dikelompokkan berdasarkan jumlah mahasiswa berdasarkan status kewarganegaraan, jumlah mahasiswa berdasarkan jenis kelamin, jumlah mahasiswa berdasarkan jalur PMB, jumlah mahasiswa berdasarkan jurusan SLTA, jumlah mahasiswa berdasarkan penghasilan orang tua, jumlah mahasiswa berdasarkan pekerjaan ayah, dan total mahasiswa berdasarkan provinsi. Tampilan *dashboard* calon mahasiswa dapat dilihat pada gambar 4.4.



Gambar 4.4 *Dashboard profil mahasiswa registrasi PMB Universitas Janabadra*

4.2 Deployment

Pengujian yang dilakukan pada penelitian ini adalah *functional testing*. Pengujian ini dilakukan untuk mengetahui *dashboard* yang telah dibangun dapat menampilkan informasi sesuai dengan *business case* yang telah dibuat yaitu:

1. Jumlah calon mahasiswa PMB Universitas Janabadra lima tahun terakhir.

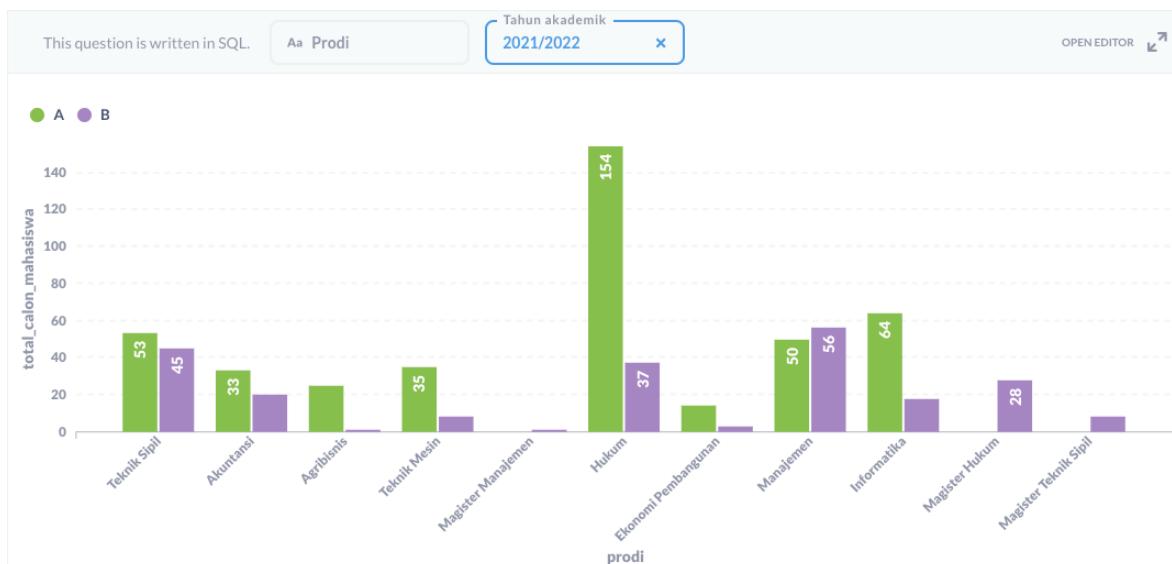
Pada visualisasi ini hanya menampilkan total calon mahasiswa yang mengikuti PMB pada tahun 2017 hingga 2021. Untuk visualisasinya dapat dilihat pada gambar 4.9.



Gambar 4.5 Tampilan visualisasi jumlah calon mahasiswa PMB

2. Jumlah Calon Mahasiswa PMB Universitas Janabadra Berdasarkan Program Studi.

Contoh bar yang menampilkan jumlah calon mahasiswa pada tahun 2021/2022 dapat dilihat pada gambar 4.10.



Gambar 4.6 Jumlah pendaftar berdasarkan program studi dan kelas

3. Jumlah Calon Mahasiswa PMB Universitas Janabadra Berdasarkan Tahun Akademik dan Semester.

- a. Jumlah Calon Mahasiswa Berdasarkan Tahun Akademik

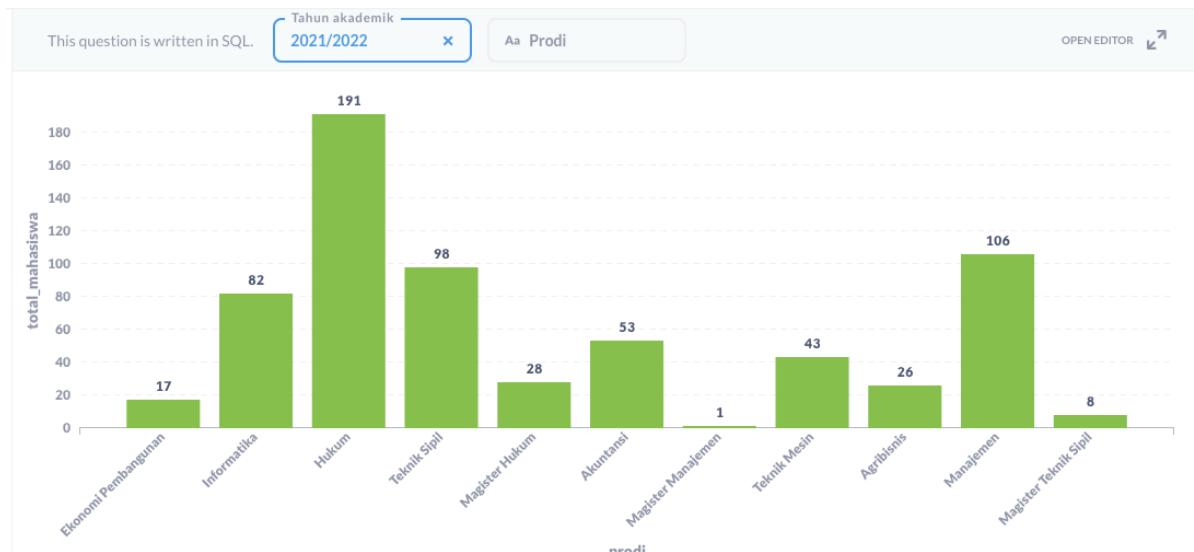
Pada *chart* ini pada tahun ajaran 2017/2018 total calon mahasiswa yang mendaftar sebanyak 963 orang dan mengalami kenaikan pada tahun 2018/2019 menjadi 1.004 orang selanjutnya pada tahun 2019/2020 mengalami kenaikan dari tahun sebelumnya yakni 1.261 orang dan pada tahun 2019/2020 pada masa pandemi *corona* mengalami penurunan menjadi 963 orang dan selanjutnya pada tahun 2020/2021 mengalami penurunan jauh dari tahun-tahun sebelumnya yakni hanya sebanyak 653 orang. Untuk *chart* jumlah pendaftar berdasarkan tahun akademik dapat dilihat pada gambar 4.11.



Gambar 4.7 Jumlah pendaftar PMB berdasarkan tahun akademik

b. Jumlah Calon Mahasiswa Berdasarkan Tahun Akademik Tiap Prodi

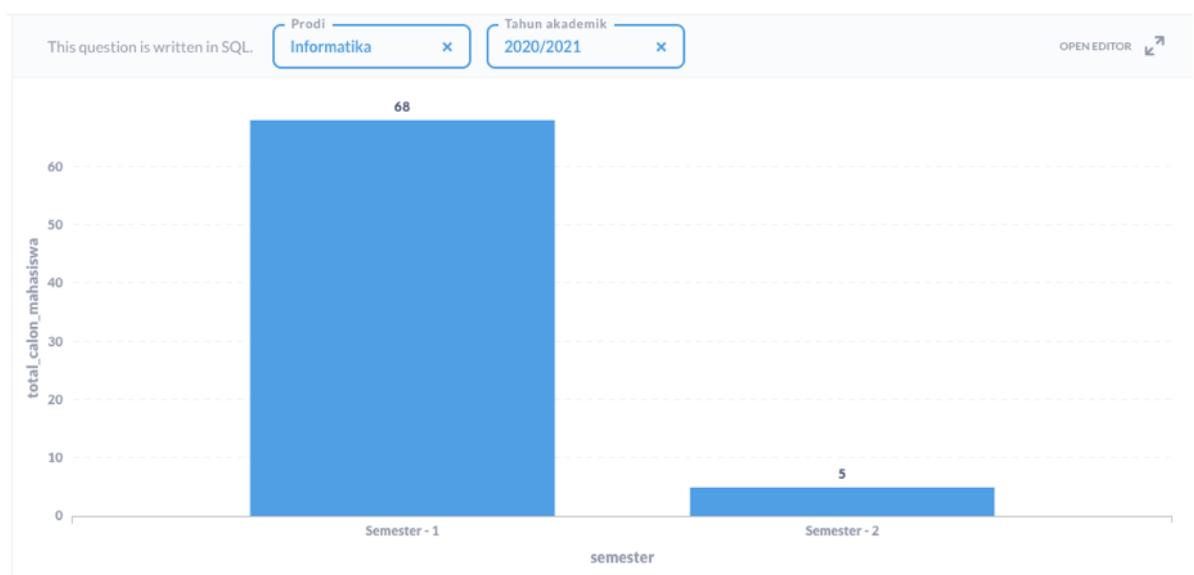
Pada tampilan ini mengambil contoh jumlah calon mahasiswa pada tahun akademik 2021/2022. Tampilan jumlah calon mahasiswa berdasarkan tahun akademik tiap Prodi dapat dilihat pada gambar 4.12.



Gambar 4.8 Jumlah pendaftar berdasarkan tahun akademik tiap Prodi

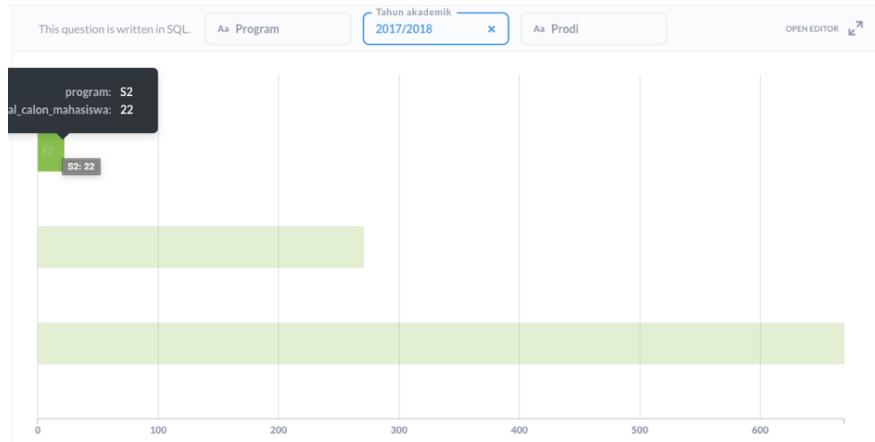
c. Jumlah Calon Mahasiswa Berdasarkan Semester

Untuk contoh menampilkan jumlah calon mahasiswa yang mengikuti PMB memilih Prodi Teknik Informatika pada tahun akademik 2020/2021 pada semester 1 berjumlah 68 orang dan pada semester 2 berjumlah 5 orang. Tampilan jumlah pendaftar berdasarkan semester dapat dilihat pada gambar 4.9.



Gambar 4.9 Jumlah pendaftar berdasarkan semester

4. Jumlah Calon Mahasiswa PMB Universitas Janabadra Berdasarkan Kelas A merupakan kelas regular, kelas B merupakan kelas malam dan S2 merupakan program magister. Untuk contoh menampilkan jumlah calon mahasiswa berdasarkan kelas pada tahun akademik 2017/2018 jumlah calon mahasiswa yang mendaftar program S2 sebanyak 22 orang. Kelas A lebih dari 600 orang dan kelas B lebih dari 200 orang. Dapat dilihat pada gambar 4.14.



Gambar 4.10 Jumlah pendaftar berdasarkan kelas

5. Jumlah Calon Mahasiswa PMB Universitas Janabadra Berdasarkan Profil (Jenis kelamin, Kewarganegaraan dan Agama)
 - a. Jumlah Calon Mahasiswa Berdasarkan Jenis Kelamin

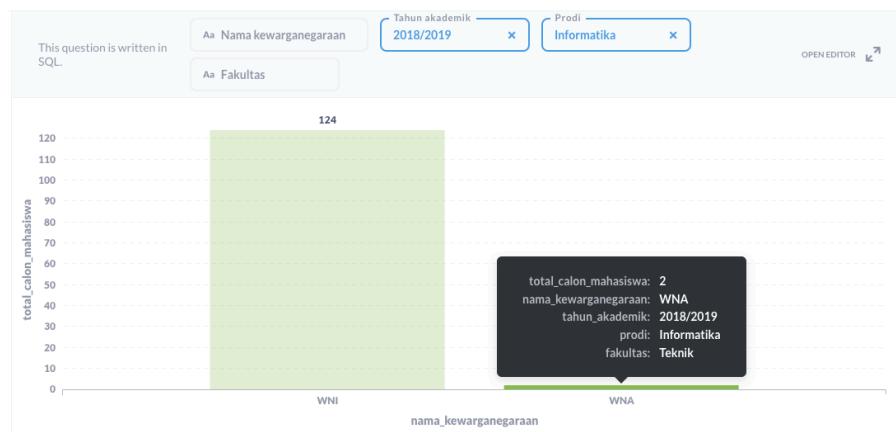
Pada visualisasi ini menampilkan jumlah calon mahasiswa berdasarkan jenis kelamin. Contoh bar yang menampilkan jumlah calon mahasiswa pada tahun 2021/2022 pada Prodi Hukum menampilkan jumlah laki-laki sejumlah 111 orang dan perempuan 80 orang. *Chart* calon mahasiswa berdasarkan jenis kelamin dapat dilihat pada gambar 4.15.



Gambar 4.11 Tampilan jumlah calon mahasiswa berdasarkan jenis kelamin

b. Jumlah Calon Mahasiswa Berdasarkan Kewarganegaraan

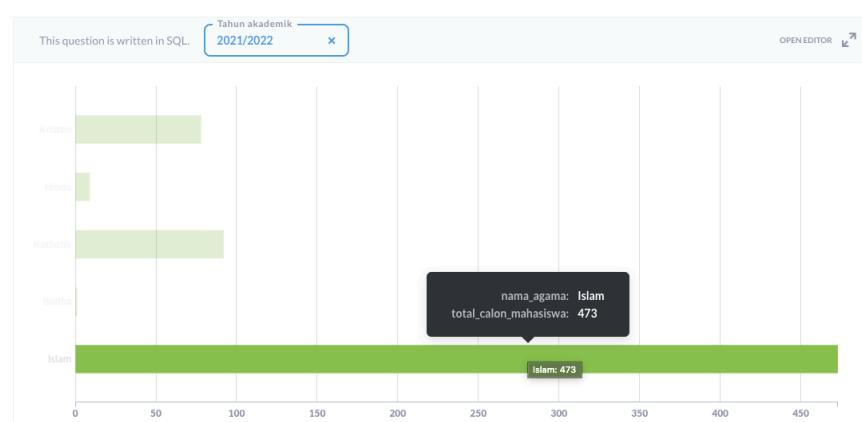
Contoh bar yang menampilkan jumlah calon mahasiswa pada tahun 2018/2019 pada Prodi Informatika menampilkan jumlah WNI sejumlah 124 orang dan WNA 2 orang. *Chart* calon mahasiswa berdasarkan status kewarganegaraan dapat dilihat pada gambar 4.16.



Gambar 4.12 Calon mahasiswa berdasarkan status kewarganegaraan

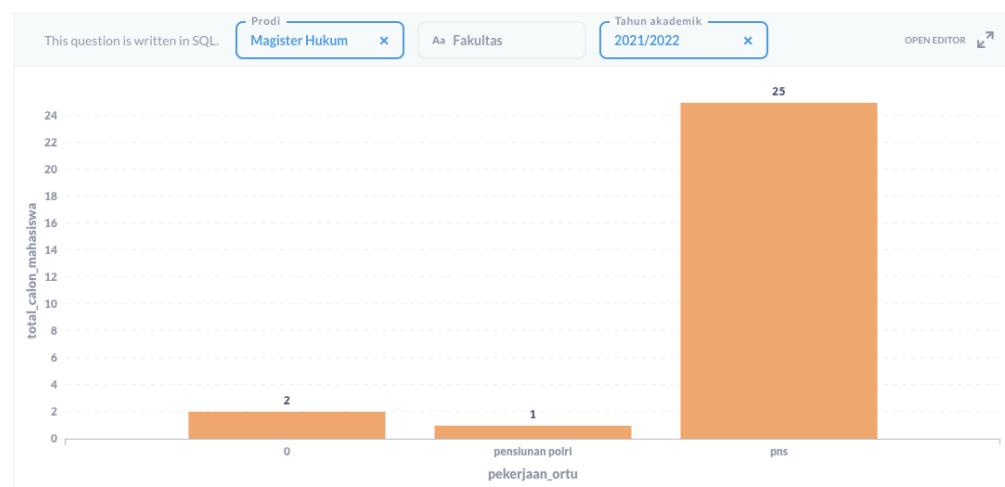
c. Jumlah Calon Mahasiswa Berdasarkan Agama

Row chart yang menampilkan jumlah calon mahasiswa pada tahun 2021/2022 pada menampilkan jumlah agama Islam sejumlah 461 orang, Budha kurang dari 50 orang, Katolik kurang dari 100 orang, Hindu kurang dari 50 orang dan Kristen kurang dari 100 orang. *Chart* calon mahasiswa berdasarkan agama yang dianut dapat dilihat pada gambar 4.17.



Gambar 4.13 Jumlah calon mahasiswa berdasarkan agama

6. Jumlah calon mahasiswa PMB Universitas Janabadra berdasarkan data Contoh bar yang menampilkan jumlah calon mahasiswa pada tahun 2021/2022 pada Prodi Magister Hukum dengan pekerjaan orang tua PNS sebanyak 25 orang, pensiunan polri 1 orang dan 2 orang data null. Untuk visualisasinya dapat dilihat pada gambar 4.18.



Gambar 4.14 Tampilan jumlah calon mahasiswa berdasarkan pekerjaan orang tua

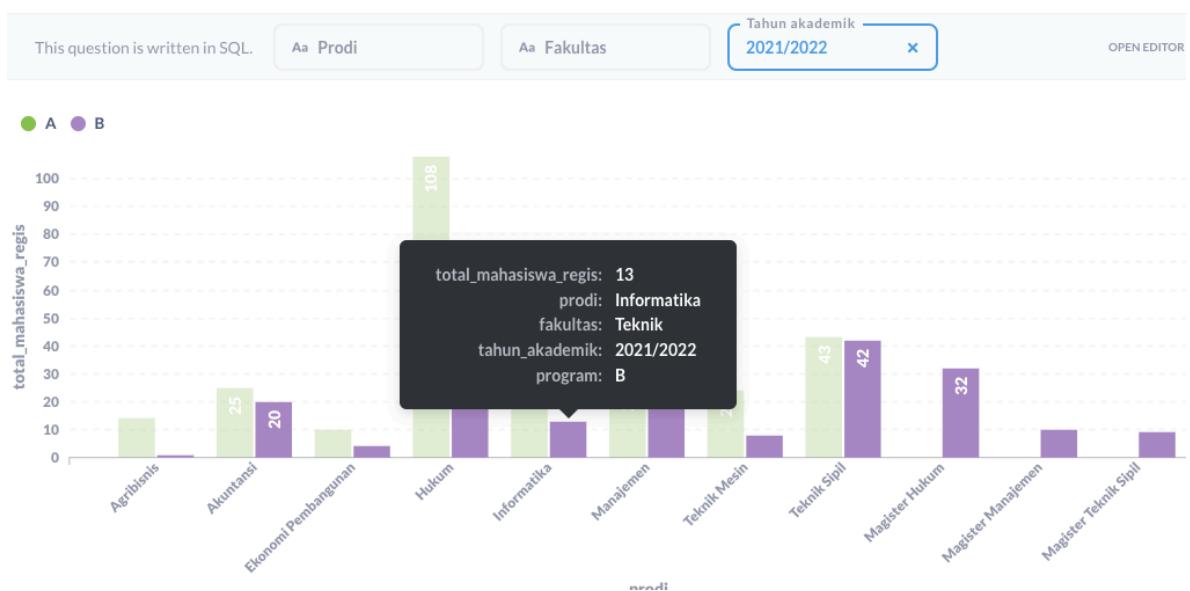
7. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra pada visualisasi ini hanya menampilkan total mahasiswa registrasi yang diterima pada PMB Universitas Janabadra pada tahun 2018 hingga 2021. Untuk visualisasinya dapat dilihat pada gambar 4.19.



Gambar 4. 15 Jumlah mahasiswa registrasi

8. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra Berdasarkan Program Studi.

Pada bar ini warna biru diartikan sebagai kelas A dan warna hijau sebagai kelas B. Informasi mahasiswa registrasi pada tahun 2021/2022 kelas A yang paling banyak adalah Prodi Hukum dengan jumlah 108 orang dan kelas B yang paling banyak adalah Prodi Manajemen dengan jumlah 46 orang. Gambar visualisasinya dapat dilihat pada gambar 4.20.



Gambar 4.16 Jumlah mahasiswa registrasi berdasarkan program studi dan kelas

9. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra Berdasarkan Tahun Akademik.

Pada tahun 2018/2019 total mahasiswa Universitas Janabarda 694 orang dan mengalami kenaikan pada tahun 2019/2020 menjadi 1.011 orang kemudian pada tahun 2020/2021 pada masa dimulainya pandemi *Corona* mengalami penurunan menjadi 813 orang dan terus mengalami penurunan ditahun 2021/2022 menjadi 508 orang dan ada 5 orang yang data tahun akademik nya *null*. Untuk visualisasinya dapat dilihat pada gambar 4.21.



Gambar 4.17 Jumlah mahasiswa registrasi berdasarkan tahun akademik

10. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra Berdasarkan Program Kelas dan Jalur PMB

a. Jumlah Mahasiswa Registrasi Berdasarkan Kelas

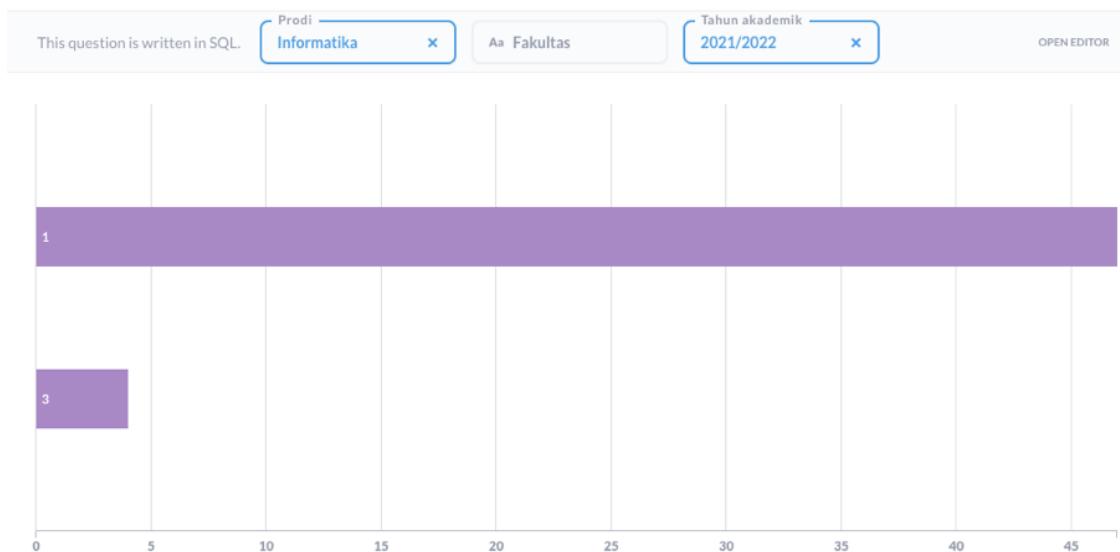
Pada visualisasi ini dapat dilihat mahasiswa pada tahun akademik 2021/2022 pada Prodi Informatika jumlah mahasiswa kelas A berjumlah 38 orang dan kelas B berjumlah 13 orang. Untuk gambar visualisasinya dapat dilihat pada gambar 4.22



Gambar 4.18 Tampilan jumlah mahasiswa registrasi berdasarkan kelas

b. Jumlah Mahasiswa Registrasi Berdasarkan Jalur Pmb

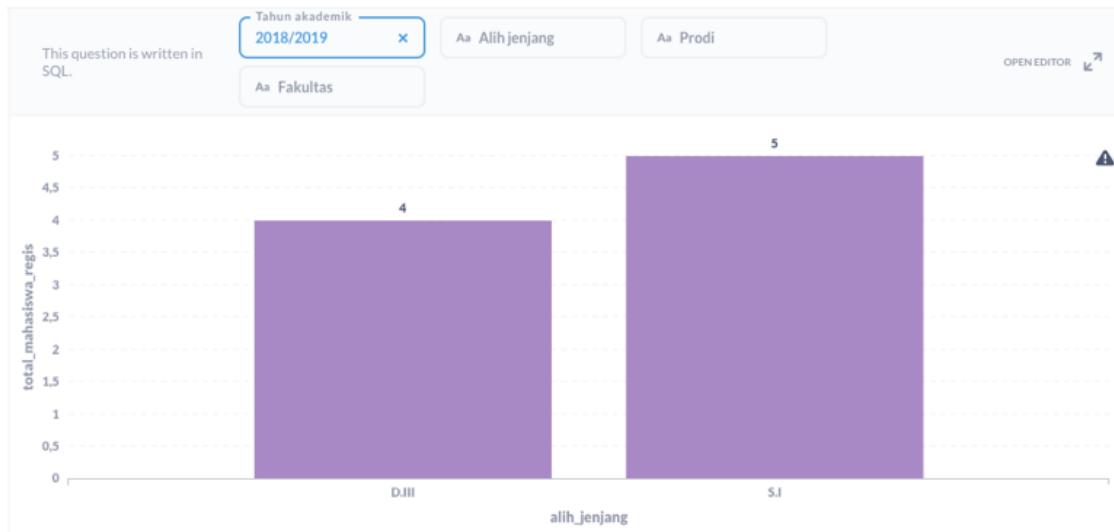
Jumlah mahasiswa registrasi berdasarkan jalur PMB Teknik Informatika tahun akademik 2021/2022 dengan kode 1 sebanyak 47 orang dan kode 3 sebanyak 4 orang. Pada visualisasi ini dapat difilter berdasarkan program studi, fakultas dan tahun akademik. Untuk tampilan Visualisasinya dapat dilihat pada gambar 4.23



Gambar 4.19 Jumlah mahasiswa registrasi berdasarkan jalur PMB

11. Jumlah Mahasiswa Registrasi Berdasarkan Alih Jenjang

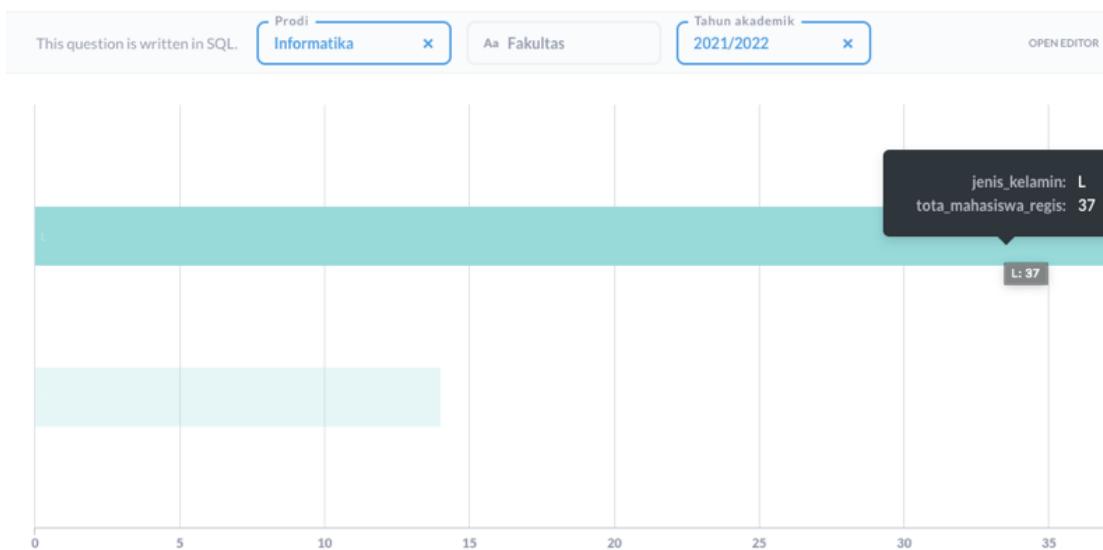
PMB Universitas Janabadra dari tiap tahun akademik. Pada tahun 2018/2019 jumlah mahasiswa alih jenjang S1 sebanyak 5 orang dan DIII sebanyak 4 orang dan pada tahun 2019/2020 hingga 2021/2022 tidak ada mahasiswa alih jenjang yang masuk Universitas Janabadra. Untuk visualisasinya dapat dilihat pada gambar 4.24.



Gambar 4.20 Jumlah mahasiswa registrasi berdasarkan alih jenjang

12. Jumlah Mahasiswa Registrasi PMB Universitas Janabadra Berdasarkan Profil (Jenis Kelamin, Kewarganegaraan, Jurusan SLTA, Provinsi)
- Jumlah Mahasiswa Registrasi Berdasarkan Jenis Kelamin

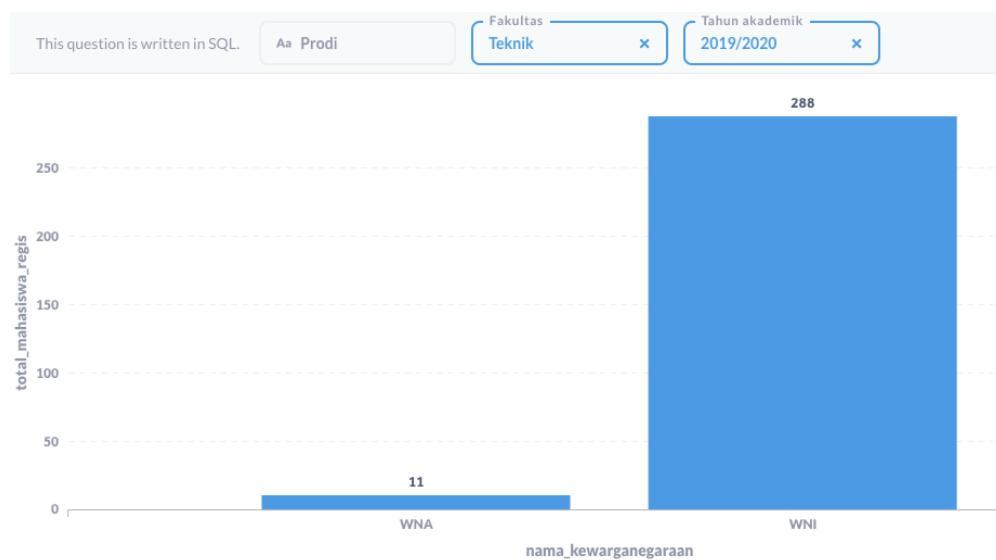
Pada bar ini jenis kelamin mahasiswa pada Prodi Teknik informatika tahun akademik 2021/2022 jenis kelamin laki-laki berjumlah 37 orang dan perempuan 14 orang. Gambar visualisasinya dapat dilihat pada gambar 4.25.



Gambar 4.21 Jumlah mahasiswa registrasi berdasarkan jenis kelamin

b. Jumlah Mahasiswa Registrasi Berdasarkan Status Kewarganegaraan

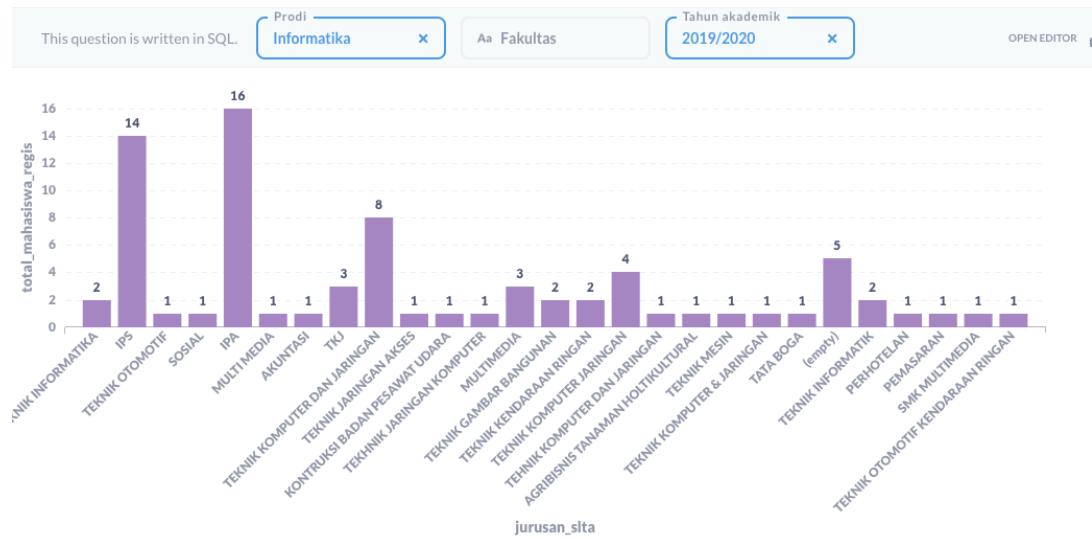
Pada tampilan visualisasi ini menampilkan status kewarganegaraan mahasiswa pada tahun akademik 2019/2020 pada Fakultas Teknik dengan WNI sebanyak 288 orang dan WNA sebanyak 11 orang. Visualisasinya dapat dilihat pada gambar 4.26.



Gambar 4.22 Jumlah mahasiswa registrasi status kewarganegaraan

c. Jumlah Mahasiswa Registrasi Berdasarkan Jurusan SLTA

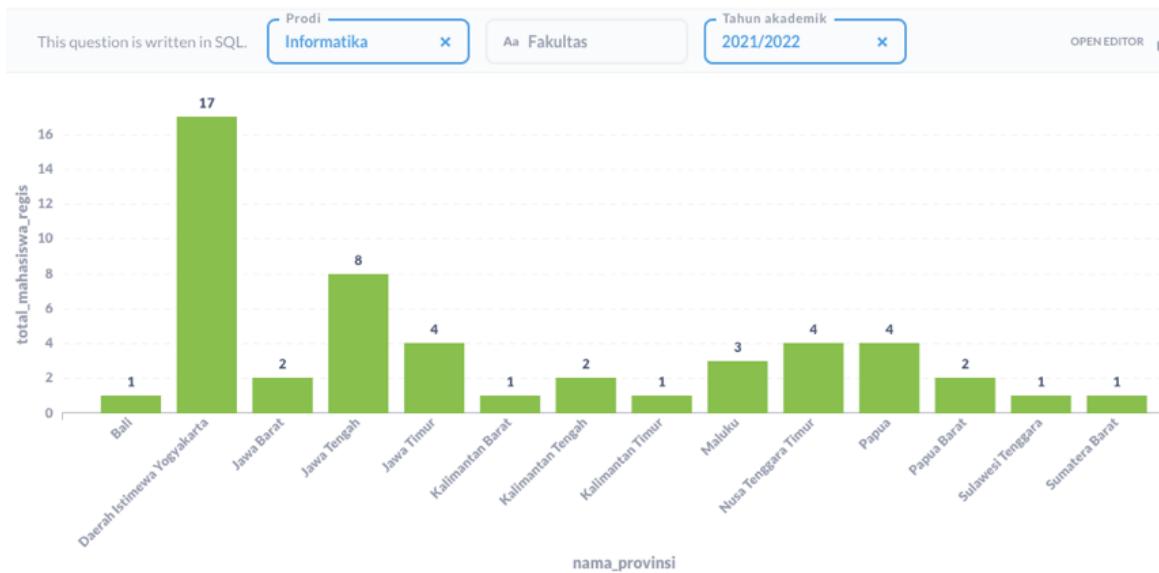
Pada visualisasi ini menggunakan *bar chart* menampilkan jumlah mahasiswa registrasi berdasarkan jurusan SLTA Teknik Informatika 2019/2020 dengan jurusan IPA yang paling dominan berjumlah 16 orang disusul jurusan IPS sebanyak 14orang. Visualisasinya dapat dilihat pada gambar 4.27.



Gambar 4.23 Jumlah mahasiswa registrasi berdasarkan jurusan SLTA

d. Jumlah Mahasiswa Registrasi Berdasarkan Provinsi

Pada tahun akademik 2021/2022 program studi Teknik Informatika mayoritas mahasiswa berasal Provinsi Daerah Istimewa Yogyakarta dengan jumlah 17 orang dan terbesar kedua berasal dari provinsi Jawa Timur dengan jumlah 4 orang. Visualisasinya dapat dilihat pada gambar 4.28.

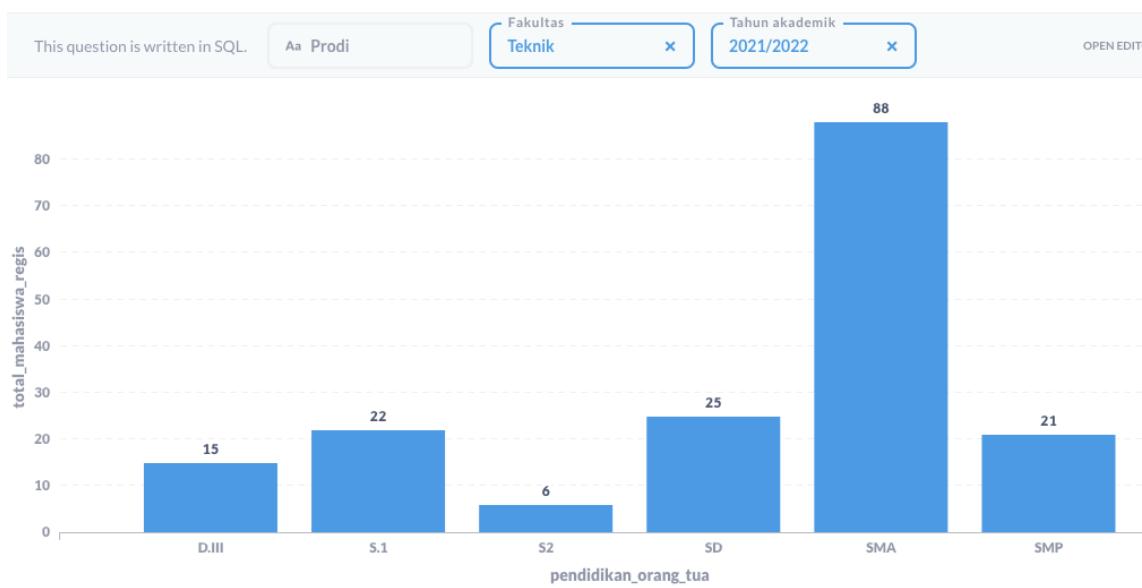


Gambar 4.24 Jumlah mahasiswa registrasi berdasarkan provinsi

13. Jumlah mahasiswa registrasi PMB Universitas Janabadra berdasarkan data keluarga (penghasilan orang tua, pendidikan orang tua, pekerjaan orang tua)

a. Jumlah Mahasiswa Registrasi Berdasarkan Pendidikan Orang Tua

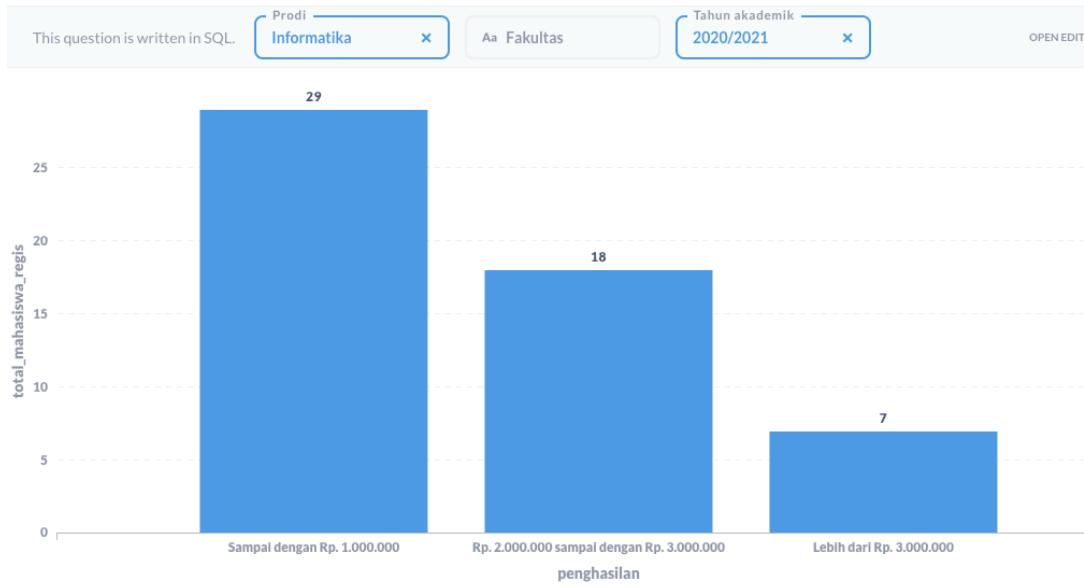
Bar yang menampilkan jumlah calon mahasiswa pada tahun 2021/2022 pada Fakultas Teknik berdasarkan Pendidikan orang tua pada *chart* ini Pendidikan orang tua yang memiliki nilai paling besar adalah SMA dengan jumlah 88 dan yang paling kecil adalah S2 dengan jumlah 6 orang. *Chart* dapat dilihat pada gambar 4.25.



Gambar 4.25 Jumlah mahasiswa registrasi berdasarkan Pendidikan orang tua

b. Jumlah Mahasiswa Registrasi Berdasarkan Pendapatan Orang Tua

Pada Prodi Teknik Informatika 2020/2021 dengan pendapatan orang tua sampai dengan Rp 1.000.000 sebanyak 29 orang Rp 2.000.000 sampai Rp 3.000.000 sebanyak 18 orang dan lebih dari Rp 3.000.000 sebanyak 7 orang. Visualisasinya dapat dilihat pada gambar 4.30.



Gambar 4.26 Jumlah mahasiswa registrasi berdasarkan pendapatan orang tua

c. Jumlah Mahasiswa Registrasi Berdasarkan Pekerjaan Orang tua

Pada Prodi Teknik Informatika 2021/2022 dengan pekerjaan orang tua yang paling banyak adalah wiraswasta dengan jumlah 16 orang dan disusul dengan petani dengan jumlah 10 orang. Visualisasinya dapat dilihat pada gambar 4.31.



Gambar 4.27 Jumlah mahasiswa berdasarkan pekerjaan orang tua

14. Jumlah Pendapatan Pendaftaran PMB Berdasarkan Tanggal Bayar

Pada tampilan visualisasi ini menampilkan total pembayaran per-hari yang dapat ditampilkan berdasarkan periode tanggal yang diinginkan. Pada tampilan ini mengambil contoh pendapatan PMB dari tanggal 10 Januari 2021 hingga 10 Maret 2021. Pada tanggal 3 Maret 2021 ada 4 calon mahasiswa yang membayar biaya pendaftaran untuk PMB dan total pendapatan sebesar 1.000.000 dengan biaya PMB sebesar Rp 250.000. Untuk gambar pendapatan berdasarkan tanggal bayar dapat dilihat pada gambar 4.32.



Gambar 4.28 Jumlah Pendapatan berdasarkan tanggal bayar

4.3 Validasi Data

Validitas data berfungsi untuk membandingkan data hasil penelitian dengan data yang sebenarnya. Proses yang dilakukan adalah dengan membandingkan data pada *data source* yaitu data yang belum mengalami proses *ETL* dengan menggunakan *query SQL* dengan data yang telah mengalami proses *ETL*. Berikut adalah perubahan data dan atribut dari tabel calon mahasiswa dan mahasiswa registrasi dapat dilihat pada tabel 4.1 dan 4.2.

Tabel 4.1 Validasi data calon mahasiswa

Perbandingan	Nama Tabel	
	Mhs_cln	Fakta_calon_mahasiswa
Jumlah Atribut	59	7
Jumlah Row	7990	4844

Tabel 4.2 Validasi data mahasiswa regis

Perbandingan	Nama Tabel	
	Mhs_regis	Fakta_mahasiswa
Jumlah Atribut	83	9
Jumlah Row	3032	3031

Pada tabel 4.1 menunjukkan perubahan jumlah data dan juga atribut pada tabel mhs_cln. Pada saat data belum mengalami proses *ETL* data berjumlah 7990 dan memiliki atribut berjumlah 59. Setelah data difilter berdasarkan tahun masuk yaitu 2017-2021 dan penghapusan data yang kosong data berubah menjadi 4844 dan perubahan dari 59 menjadi 7 dikarenakan hanya 7 atribut yang dibutuhkan untuk analisis.

Pada tabel 4.2 menunjukkan perubahan jumlah data dan juga atribut pada tabel mhs_regis. Pada saat data belum mengalami proses *ETL* data berjumlah 3032 dan memiliki atribut berjumlah 83. Setelah data difilter berdasarkan tahun masuk yaitu 2018-2021 dan penghapusan data yang kosong data berubah menjadi 3031 dan perubahan dari 83 menjadi 9 dikarenakan hanya 9 atribut yang dibutuhkan untuk analisis.

4.4 Analisis Dashboard

Pada *dashboard* calon mahasiswa dapat dilihat total calon mahasiswa yang mendaftar program PMB Universitas Janabadra dari tahun akademik 2017/2018 yang berjumlah 963 orang dan tahun akademik 2018/2019 berjumlah 1.004 orang dan terus mengalami peningkatan hingga tahun akademik 2019/2020 menjadi 1.261 orang dan selanjutnya mengalami penurunan sejak pandemi covid-19 pada tahun akademik 2020/2021 berjumlah 963 orang dan total pendaftar PMB terus mengalami penurunan hingga hingga titik terendah dalam 5 tahun terakhir yakni tahun akademik 2021/2022 berjumlah 653 orang.

Sebaran jumlah calon mahasiswa yang mendaftar PMB Universitas Janabadra berdasarkan program studi S1 yang paling banyak diminati adalah program studi Hukum dengan jumlah 1.314 orang dan disusul dengan program studi Teknik Sipil yang berjumlah 873 orang sedangkan program studi yang paling sedikit diminati adalah program studi Ekonomi Pembangunan dengan jumlah 119 orang. pada program Magister dengan program studi yang paling banyak diminati adalah Magister Hukum dengan jumlah 192 orang dan paling sedikit peminat adalah Magister Teknik Sipil dengan jumlah 66 orang. Dari segi jumlah pendaftaran calon mahasiswa berdasarkan Fakultas, tren pendaftaran PMB mirip mengikuti jumlah pendaftaran mahasiswa secara keseluruhan tetapi hanya Fakultas Teknik yang terus mengalami penurunan dari tahun akademik 2017/2018 hingga 2021/2022.

Pada *dashboard* calon mahasiswa berdasarkan profile mayoritas calon mahasiswa yang mendaftar adalah berjenis kelamin laki-laki dengan jumlah 3.281 orang dan perempuan 1.554 orang dan berdasarkan status kewarganegaraan calon mahasiswa dengan status WNA paling banyak mendaftar pada tahun akademik 2019/2010.

Pada *dashboard* mahasiswa registrasi dapat dilihat total mahasiswa yang melakukan registrasi pada program PMB Universitas Janabadra dari tahun akademik 2018/2019 berjumlah 694 orang dan terus mengalami peningkatan hingga tahun akademik 2019/2020 menjadi 1.011 orang dan selanjutnya mengalami penurunan sejak pandemi covid-19 pada tahun akademik 2020/2021 berjumlah 813 orang dan total mahasiswa registrasi PMB terus mengalami penurunan hingga hingga titik terendah dalam 4 tahun terakhir yakni tahun akademik 2021/2022 berjumlah 508 orang.

Dari segi mahasiswa registrasi berdasarkan Fakultas memiliki pola yang sama seperti mahasiswa registrasi secara keseluruhan. hanya fakultas Teknik yang mengalami penurunan dari titik tertinggi dalam 4 tahun terakhir yakni 2018/2019 dan terus menurun hingga tahun 2021/2022.

Pada *dashboard* mahasiswa registrasi berdasarkan profile, mayoritas mahasiswa registrasi adalah berjenis kelamin laki-laki dengan jumlah 2.041 orang

dan perempuan 990 orang dan berdasarkan status kewarganegaraan mahasiswa registrasi dengan status WNA paling banyak mendaftar pada tahun akademik 2019/2010.

Dari *dashboard* calon mahasiswa dan mahasiswa registrasi mayoritas calon mahasiswa dan mahasiswa yang melakukan registrasi berasal dari provinsi DIY diikuti dengan provinsi Jawa Tengah dan fakultas yang paling banyak diminati mahasiswa adalah Fakultas Hukum dan yang paling sedikit peminat adalah Fakultas Pertanian.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian implementasi *dashboard* visualisasi penerimaan mahasiswa baru Universitas Janabadra dapat disimpulkan:

1. *Dashboard* informasi visualisasi penerimaan mahasiswa baru Universitas Janabadra dapat dirancang menggunakan metodologi *business intelligent roadmap* dengan 6 tahapan yaitu *justification* mendefinisikan masalah dan peluang bisnis dari PMB kemudian lanjut tahapan *planning* yaitu mengidentifikasi kebutuhan data dan melakukan wawancara dengan pihak struktural Universitas Janabadra kemudian lanjut pada proses *business analysis* yaitu menganalisa informasi yang akan ditampilkan pada dashboard sesuai dengan hasil observasi dan wawancara kemudian lanjut pada tahap *design* yaitu merancang database yang akan dimanfaatkan, perancangan *star schema* untuk pembangunan tabel fakta calon mahasiswa dan mahasiswa registrasi pada data warehouse PMB. kemudian lanjut pada tahapan *construction* yaitu proses membangun data *warehouse* dan *dashboard*. menggunakan *software airflow* untuk integrasi data yang dipilih dari data *source* ke *data lake* dan *python script* untuk proses *ETL* data menjadi data *warehouse* sesuai dengan *star schema* yang telah dirancang.
2. *Dashboard* penerimaan mahasiswa baru Universitas Janabadra dapat dibangun menggunakan *metabase*. *dashboard* yang dibangun berjumlah empat halaman *dashboard* yaitu dashboard calon mahasiswa, dashboard profil calon mahasiswa, dashboard mahasiswa registrasi dan dashboard profil mahasiswa registrasi.
3. Pada tahapan *deployment* yaitu melakukan *functional test* terhadap *dashboard* yang telah dibangun berdasarkan *business case* yang telah dibuat. *Dashboard* yang sudah dibangun menggunakan *metabase* berfungsi memonitoring penerimaan mahasiswa baru dalam bentuk

informasi visual yang mudah dibaca dan dipahami serta dapat dilihat berdasarkan program studi, fakultas dan tahun akademik yang dapat digunakan pihak struktural dalam memonitoring kegiatan admisi dan promosi penerimaan mahasiswa baru.

5.2 Saran

Berdasarkan hasil penelitian pada visualisasi *dashboard* penerimaan mahasiswa baru Universitas Janabadra ini, maka saran untuk penelitian selanjutnya:

1. Mengintegrasikan data terkait pembayaran seperti tridarma dan status pembayaran mahasiswa terkait penerimaan mahasiswa baru sehingga dapat menampilkan pendapatan Universitas dari program PMB.
2. Mengimplementasikan data *source* langsung dari server Universitas yang terkait penerimaan mahasiswa baru sehingga data dapat dilihat secara *real-time* saat program PMB berlangsung.

DAFTAR PUSTAKA

- Almadi, W. (2021). Airbyte Open Source Data Integration Pipeline. Available at: <https://ilmuprogram.com/2021/08/10/airbyte-open-source-data-integration-pipeline/> diakses 04 Januari 2022.
- Akbar, R., Rasyiddah, D., Anrisya, M., Fritania Julyazti, N., & Syaputri, S. (2018). JEPIN (Jurnal Edukasi dan Penelitian Informatika) Penerapan Aplikasi Power Business Intelligence Dalam Menganalisis Prioritas Pekerjaan di Indonesia. *JEPIN (Jurnal Edukasi Dan Penelitian Informatika) Penerapan Aplikasi Power Business Intelligence Dalam Menganalisis Prioritas Pekerjaan Di Indonesia*, 4, 54–59.
- AltexSoft Inc. (2019)." Complete Guide to Business Intelligence and Analytics: Strategy, Steps, Processes, and Tools". available at: <https://medium.com/swlh/complete-guide-to-business-intelligence-and-analytics-strategy-steps-processes-and-tools-3b3f068b8a67> diakses 05 mei 2021
- Ardian Dwi Praba, M. S.(2020). Studi Perbandingan Performansi Antara Mysql Dan Postgresql. *Jurnal Khatulistiwa Informatika*, VIII(2), 1–6.
- Ariani, T. R., Tania, K. D., & Indah, D. R. (2016). Penerapan Business Intelligence Pada Sistem Informasi Penjualan Barang PT. WINSA (STUDI KASUS DI PT. WINSA PALEMBANG). *Jurnal Sistem Informasi*, 103–110.
- Arumsari, M.(2019). Microsoft Visual Studio Code: Seperti Apa Fitumya?. Available at: <https://www.dicoding.com/blog/microsoft-visual-studio-code/> diakses 04 Januari 2022.
- Bororing, J.E.(2021).*Implementasi Dashboard Untuk Visualisasi Data Penerimaan Mahasiswa Baru Universitas Janabadra.*
- Choiri, Eril Obeit.(2020). Mengenal Lebih Lengkap Database PostgreSQL. Available at: <https://qwords.com/blog/database-adalah/>. diakses 04 Januari 2022.
- Docker. (2021). Docker Overview. Available at: <https://github.com/docker/docker.github.io/blob/master/get-started/overview.md> diakses tanggal 16 Januari 2022
- Fikri Muhammad. (2021)." Sudah Tahu Apa Itu Pendidikan?? Ini Dia Penjelasannya". Available on: <https://smansatebingtinggi.sch.id/2021/03/19/sudah-tahu-apa-itu-pendidikan-ini-dia-penjelasannya/> diakses tanggal 13 Januari 2022
- Hasan, F. N. (2019). Implementasi Sistem Business Intelligence Untuk Data Penelitian di Perguruan Tinggi. *Prosiding Seminar Nasional Teknoka*, 4(2502), I1–I10. <https://doi.org/10.22236/teknoka.v4i1.3943>

- Herwanto, H., & Khumaidi, A. (2020). Implementasi Aplikasi Business Intelligence Untuk Memonitor Efisiensi Pengelolaan Rumah Sakit. *Jurnal Media Informatika Budidarma*, 4(3), 495. <https://doi.org/10.30865/mib.v4i3.2090>
- Hijriani, A., Aprilliana, E., Pribadi, R. I. A., & Sakethi, D. (2020). Business Intelligence Dashboard (BID) Pada Usaha Mikro Bidang Retail Studi Kasus CV Duta Square Bandar Lampung. *Matrix : Jurnal Manajemen Teknologi Dan Informatika*, 10(1), 11–18. <https://doi.org/10.31940/matrix.v10i1.1616>
- Ibeng Parta. (2021). Pengertian Grafik, Contoh, Jenis, Tujuan Beserta Fungsinya . Available at: <https://pendidikan.co.id/pengertian-grafik-contoh-jenis-tujuan-beserta-fungsinya/> diakses 05 Januari 2022.
- JetBrains s.r.o. (2021). "Introduction". Available at: <https://www.jetbrains.com/help/datagrip> diakses 14 Januari 2022
- Korocz, T. (2020). Introducing Metabase, a Simple and Powerful Analytics Tool. Available from <https://www.percona.com/blog/2020/04/01/introducing-metabase-a-simple-and-powerful-analytics-tool/> diakses 04 Januari 2022.
- Larissa T. Moss, S. A. (2003). Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications. Addison-Wesley Professional.
- Microsoft. (2021). Getting Started. Available at <https://code.visualstudio.com/docs> diakses 04 Januari 2022.
- Oliver, A. (2021). Tingkatkan Kualitas Development Perusahaanmu dengan Google Cloud Platform. Available at: <https://glints.com/id/lowongan/gcp-google-cloud-adalah/> diakses 04 Januari 2022.
- Perdana Arkan. (2021). Data Visualization: Cara Tampilkan Data agar Mudah Dipahami. Available at: <https://glints.com/id/lowongan/data-visualization-adalah/> diakses 10 Februari 2022.
- Postgresql.org. (n.d.)."What is PostgreSQL?". Available at<https://www.postgresql.org/about/> diakses 14 Januari 2022
- Prasetya, H. P., & Meme Susilowati. (2020). Pemanfaatan Business Intelligence Di Perguruan Tinggi. *KURAWAL Jurnal Teknologi, Informasi Dan Industri*, 3, 40–57. Retrieved from <https://jurnal.machung.ac.id/index.php/kurawal/article/view/303/170>
- Prasetya, H. P., & Susilowati, M. (2016). Visualisasi Informasi Data Perguruan Tinggi Dengan Data. *Jurnal Teknik Informatika Dan Sistem Informasi*, 2(3), 298–308. Retrieved from <https://journal.maranatha.edu/index.php/jutisi/article/view/634>
- Tandika,B. (2020). Apa Itu Bahasa Pemrograman Python?. Available from <https://glints.com/id/lowongan/apa-itu-bahasa-pemrograman-python/>. diakses 04 Januari 2022
- Utami, C. B., Faruk, L., & Musyarrofah, O. (2015). Pengaruh Business Intelligence Terhadap Dunia Bisnis. *ICIT Journal*, 1(1), 16–25. <https://doi.org/10.33050/icit.v1i1.6>

- Wibowo, A. S., & Andri, A. (2021). Dashboard Business Intelligence Vusialisasi Data Akreditasi Sekolah Pada SMP Negeri 1 Sembawa. *Jurnal Nasional Ilmu Komputer*, 2(4), 249–256. <https://doi.org/10.47747/jurnalnik.v2i4.536>
- Yasin, K. (2019). Pengertian MySQL, Fungsi, dan Cara Kerjanya (Lengkap). Available at: <https://www.niagahoster.co.id/blog/mysql-adalah/> diakses 04 Januari 2022.
- Yohanna, M., & Rumapea, Y. Y. P. (2020). Rancang Bangun Dashboard Dalam Memonitoring Jumlah Mahasiswa Baru. *CESS (Journal of Computer Engineering, System and Science)*, 5(1), 57. <https://doi.org/10.24114/cess.v5i1.15511>
- Yunistira, A., & Fudholi, D. H. (2020). Analisis Penerapan Model Business Intelligence pada Aplikasi Payment Point Online Banking dalam Meningkatkan Strategi Pemasaran (Studi Kasus: Aplikasi ApotikKuota). *Jurnal Ilmu Komputer Dan Agri-Informatika*, 7(1), 1–10. <https://doi.org/10.29244/jika.7.1.1-10>