

DASHBOARD VISUALISASI DATA ERP MEMANFAATKAN FRAMEWORK DJANGO

Arief Gunawan, *Departemen S1 Informatika Institut Sains dan Teknologi Terpadu Institusi Surabaya*, Dr. Joan Santoso, S.Kom, M.Kom, Department of Computer Science

Abstrak— Aplikasi visualisasi dashboard ini dibuat berbasis website. Pembuatan aplikasi ini bertujuan untuk user dapat mendesain dashboard lebih mudah dan sesuai dengan data yang disediakan oleh user. Aplikasi ini juga berfungsi sebagai pemilah data untuk suatu chart atau angka yang dikalkulasi. Selain itu, tujuan lain dari aplikasi ini adalah untuk menyajikan dashboard yang sudah diproses dengan cepat dan sesuai dengan desain user yang telah ditentukan.

Beberapa fitur utama yang dimiliki oleh aplikasi ini adalah user menentukan relasi setiap tabel yang ada pada sumber data, mengambil semua data yang terdapat pada sumber data yang disediakan, pengolahan data sesuai dengan konfigurasi user, visualisasi dashboard sesuai dengan konfigurasi user yang telah dibuat.

Fitur mengambil semua data dilakukan secara sebagian dan bertahap, tujuan dari fitur ini adalah supaya sistem dapat mengambil data yang besar tanpa menggunakan kebutuhan sistem yang besar. Pada fitur pengolahan data, data akan diolah sesuai dengan pilihan user yang sudah disediakan oleh sistem. Fitur visualisasi dashboard dapat menyajikan hasil visualisasi dengan cepat dengan menggunakan teknik *cached page*.

Hasil dari menggunakan teknik *cached page* untuk menghilangkan semua proses yang dibutuhkan dashboard saat akan dimuat ulang. Dashboard dapat dikirim dengan cepat dan tanpa adanya kendala dari sistem untuk menampilkan dashboard dari data yang besar.

Kata Kunci— Data Analyst, Django, MongoDB, Dashboard, Visualisasi Data.

I. PENDAHULUAN

Data merupakan bagian penting dari semua sistem. Data juga perlu diolah untuk menampilkan dan menyampaikan suatu laporan. Laporan dapat merupakan suatu bentuk penyampaian berita, keterangan, pemberitahuan ataupun pertanggung jawaban. Oleh karena itu diperlukannya visualisasi data yang digunakan untuk menampilkan data dengan bentuk yang dapat dipahami.

Visualisasi data sendiri sangat diperlukan untuk *user* dapat memahami kondisi atau gambaran tentang hal yang sedang terjadi pada suatu kumpulan data yang ada. Visualisasi dalam data sendiri dapat berupa *chart*, tabel, ataupun angka. Selain itu, diperlukan juga pengaturan peletakkan supaya data yang telah diproses dapat dipahami dengan jelas oleh si pembaca.

Aplikasi ini juga dapat menanggulangi berbagai macam model *database* yang digunakan oleh *user* seperti MySQL,

dan PostgreSQL. Selain itu, Aplikasi ini juga dapat menerima data dalam bentuk format csv yang memiliki *header*.

Tugas Akhir ini, merupakan sistem yang akan adalah sistem pembuatan visualisasi data *dashboard*. Dalam pembuatan *dashboard*, *user* membuat *dashboard* dengan memilih *template* yang dipilih dan memakai visualisasi data yang telah disediakan. Saat pembuatan *dashboard*, *user* juga dapat melihat *preview* halaman *dashboard* yang sedang dibuat. Aplikasi ini dijalankan dalam bentuk *website*. Sehingga orang lain yang ingin melihat *dashboard*, dapat mengaksesnya dimana saja melalui *browser*.

II. TEORI PENUNJANG

A. Django

Django adalah sebuah framework full-stack untuk membuat aplikasi web dengan bahasa pemrograman Python. Framework akan membantu kita membuat website lebih cepat, dibandingkan menulis kode dari awal¹. Full-stack artinya, django meliputi sisi front-end dan juga back-end. Front-end adalah sisi depan yang akan dilihat oleh pengguna, sedangkan back-end adalah sisi belakang yang berhubungan dengan database dan alur sistem.

Django menggunakan konsep Model-View-Template (MVT) sedikit berbeda dari MVC. Sebenarnya perbedaan utama antara kedua pola adalah bahwa Django sendiri menangani bagian Pengendali (Kode Perangkat Lunak yang mengontrol interaksi antara Model dan Tampilan). Template nya adalah file HTML dicampur dengan Django Template Language (DTL).

Django mengikuti prinsip KISS (Keep It Short and Simple). Ini berarti dalam Django kode harus singkat, mudah dimengerti, dan metode tidak boleh lebih dari 40-50 baris. Django juga mengikuti prinsip DRY (Don't Repeat Yourself) yang berarti bahwa software pattern yang sering muncul dapat digantikan dengan abstractions. Dengan cara ini, pihak pengembang dapat menyederhanakan proses pengembangan sehingga dapat membantu mempercepat waktu produksi secara keseluruhan

B. MongoDB

MongoDB adalah sebuah program database berbasis dokumen yang free, open-source dan cross-platform. MongoDB termasuk sebagai database NoSQL karena MongoDB menggunakan struktur data yang mirip dengan

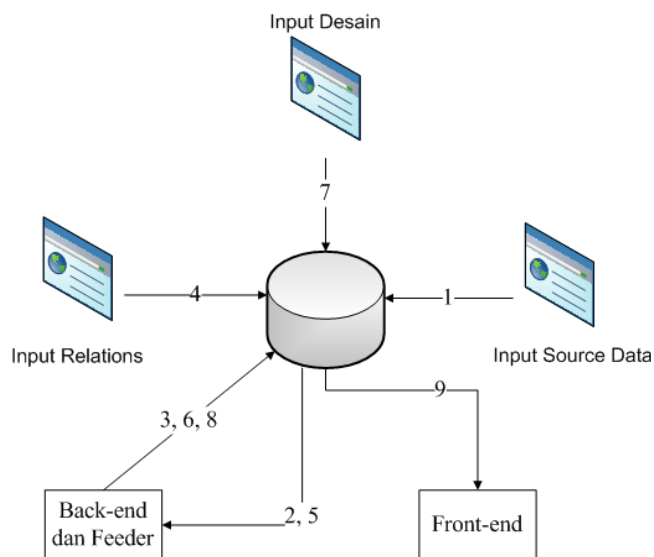
¹ Django, <https://docs.djangoproject.com/en/2.0/>.

JSON dan perpaduan dengan struktur RD BMS. Skema ini lebih dikenal dengan istilah BSON (*Binary JSON*). Database MongoDB merupakan database yang merupakan document.

MongoDB digunakan sebagai database pada sejumlah situs dan layanan terkenal, seperti Craigslist, eBay, Foursquare, SourceForge, dan The New York Times.

III. ARSITEKTUR SISTEM

Pada bagian ini akan dijelaskan mengenai desain arsitektural dari sistem yang akan dibuat. Pada desain arsitektur dapat dilihat sistem yang terdapat pada Dashboard Visualisasi Data ERP Memanfaatkan Framework Django.



Gambar. 1. Arsitektur Sistem Dashboard Visualisasi

Pada Gambar 1 merupakan desain sistem pada dashboard visualisasi. Terdapat 2 service pada sistem, yaitu Back-end dan Feeder, dan Front-end. Proses sebelum user untuk membuat dashboard diperlukan untuk input sumber data.

Pada sistem Back-end dan Feeder merupakan sistem yang menampung input dari user dan mengambil data yang terpadat pada sumber data yang telah diberikan oleh user. Sedangkan pada sistem Front-end merupakan sistem yang akan menampilkan dashboard yang telah dibuat oleh user dan dashboard tersebut telah dibuat public oleh pembuat.

Untuk input sumber data dapat dilihat pada Gambar 1. Penjelasan alur panah beserta nomer yang tertera akan dijelaskan sebagai berikut:

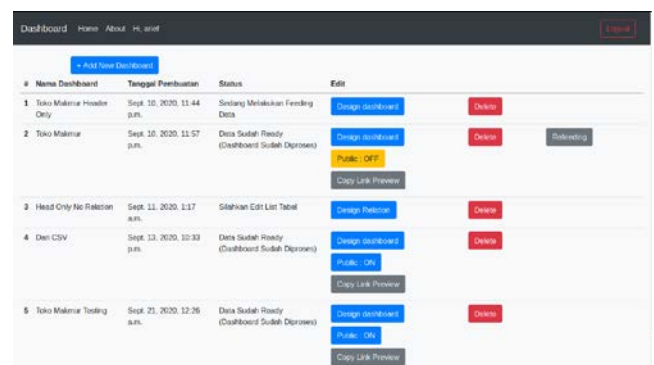
1. User memasukkan input data berupa informasi apakah sumber data tersebut berdasarkan dari RDBMS (PostgreSQL atau MySQL) atau CSV. Tujuan user menginput sumber data bertujuan untuk data tersebut valid dan dapat digunakan untuk memproses data yang ada pada dashboard.
2. Data yang sudah masuk akan dilakukan scheduling untuk mengambil list tabel yang ada dan kolom yang ada pada sumber data yang ada. Bila sumber data berupa csv, maka langsung lanjut ke step 7.
3. Untuk service Back-end dan Feeder setelah menerima sumber data akan membuat list tabel dan kolom dan disimpan dalam bentuk json pada aplikasi

untuk di proses lagi oleh user dalam memilah data yang akan digunakan.

4. User menginput data relation, data yang disimpan berupa relasi antar tabel dan primary key pada tabel yang diperlukan. Tujuan dalam hal ini untuk memudahkan user untuk filter data yang lebih variatif. Sedangkan untuk setting type column untuk membantu sistem dalam menyimpan tabel.
5. Setelah user menyimpan data relasi yang ada, maka data tersebut akan dimasukkan ke dalam scheduling untuk mengambil semua data sesuai relasi yang ada.
6. Service back-end dan feeder menyimpan seluruh data yang dapat di ambil pada tabel yang dipilih oleh user. Setiap tabel yang di ambil, juga di ambil data tabel yang memiliki relasi pada tabel yang dipilih.
7. User menginput desain dashboard melalui template yang sudah disediakan. Hasil yang disimpan dalam bentuk JSON ke dalam field pada document MongoDB.
8. Data design yang akan di proses, langsung akan diproses secara bertahap bila banyak yang langsung di input ke dalam design dashboard. Untuk hasil yang sudah di proses akan disimpan dalam bentuk JSON ke dalam field pada document dashboard yang bersangkutan dan membuat view cached page untuk front-end.
9. Saat user mengakses hasil tampilannya, aplikasi front-end hanya membaca dari view cached pages yang telah disediakan.

IV. DESAIN INTERFACE

Pada bagian ini akan dijelaskan mengenai tampilan dari sisi user saat pembuatan dashboard dalam website Tugas Akhir ini. Ada beberapa fitur pada tugas akhir ini dalam pembuatan dashboard pada tugas akhir ini. Dalam bagian ini akan dijelaskan interface yang diterima oleh user saat mendesain dashboard yang merupakan fitur dari tugas akhir ini.



Gambar. 2. Interface Dashboard Controller

Pada Gambar 2 merupakan tampilan saat user setelah login. List dashboard yang ditampilkan merupakan dashboard yang telah dibuat oleh user yang sedang login. User melakukan input pada dashboard dapat dilihat pada kolom edit yang bertombol warna biru. Pada button delete digunakan untuk menghapus dashboard yang telah dibuat oleh user.

Pada button refeeding bertujuan untuk mengambil data ulang pada sumber data bila sumber data dari PostgreSQL dan MySQL. Bila dashboard telah siap untuk ditampilkan, terdapat tombol untuk mematikan atau menyalakan dashboard dapat diakses selain dari pemilik. Beserta tombol untuk melakukan copy link pada dashboard.

Gambar. 3. Form Input Menambahkan Dashboard Baru

Saat user membuat dashboard sumber data dapat selain dari database. Untuk sumber data berasal dari CSV user hanya tinggal mengupload file CSV yang memiliki delimited “,” (titik dan koma). Sedangkan untuk tipe koneksi yang ada pada combobox pada sumber data dari database adalah PostgreSQL dan MySQL. Pada saat user menekan tombol save, sistem akan melakukan pengecekan pada input text semua harus terisi (bila sumber data berasal dari database).

Gambar. 4. Fitur Edit Relation pada Sumber Data

Pada Gambar 4 user dapat memilih setiap tipe column pada setiap column tabel yang ada. Hal ini bertujuan untuk membantu sistem dalam pengolahan dan filtering data yang dilakukan oleh user. Dalam hal ini terdapat 3 tipe yang sudah disiapkan oleh sistem. Yaitu text, number, decimal. Selain bertujuan untuk membantu user dalam melakukan filtering data, tujuan dari menentukan tipe column adalah mempermudah sistem untuk menyimpan hasil data yang didapatkan dari sumber data yang telah disiapkan.

Setelah user menginput relation yang diperlukan oleh sistem. User mulai dapat membuat dashboard yang akan diinginkan. Feeding data dilakukan secara paralel sehingga tidak mengganggu user saat mendesain dashboard.

Gambar. 5. Form Input Menambahkan Dashboard Baru

Sebelum user memilih suatu template untuk dipilih, user diminta untuk menginput berapa banyak kolom pada suatu row. Sistem grid pada tugas akhir ini maximal satu baris menampung 4 column. Column yang dipilih membagi tempat yang ada, contohnya dapat dilihat pada Gambar 5 di atas tombol “Add row”. Dalam contoh tersebut satu baris di isi 2 kolom, dan satu baris yang panjangnya mengikuti ukuran layar nantinya akan terbagi menjadi 2 bagian sesuai dengan konfigurasi user setiap barisnya.

Gambar. 6. Interface User Memilih Grid Setiap Baris Dashboard.

Setelah user menginput banyak kolom pada suatu baris, user dapat memilih template apa yang akan terisi dengan template yang sudah disediakan. Pada baris template yang telah disediakan dapat di geser dengan menahan template dan digeser sesuai dengan arah yang diinginkan. Pada gambar template dan template text dapat ditekan untuk menampilkan form input template.

Gambar. 7. Form Input Template Text

Pada Gambar 3.13 merupakan form input bila template Text dipilih. Dalam form template text, user dapat mensetting border apa saja dapat dipilih, beserta isi text dan data yang akan ditampilkan. Saat pemrosesan data user dapat melakukan filter untuk data yang diolah untuk melakukan perhitungan yang di inginkan. Perhitungan yang dapat dilakukan pada template ini adalah sum, max, min, count, Average.

Gambar. 8. Form Input Template Chart

Pada Gambar 8 merupakan form input bila template yang dipilih merupakan bukan text template. Selain itu user dapat mengganti kalkulasi data yang memenuhi syarat terbesar ke terkecil dan peringkat lebih kecil dari konfigurasi user sebagai “others”, filter ini ada bila pada template terdapat perhitungan menjadikan others. Selain itu untuk filtering data dapat dilakukan juga sama seperti template text.

TABEL I
LIST TEMPLATE DESAIN DASHBOARD

Type Template	Regular
Line Chart	
Bar Chart	
Pie Chart	
Doughnut Chart	

Template yang terdapat pada desain dashboard

TABEL I
(LANJUTAN)

Polar Area Chart	
Multiple Line Chart	
Multiple Line Area Chart	
Stacked Bar Chart	

Template yang terdapat pada desain dashboard

Pada Tabel 1 merupakan list chart-chart yang tersedia pada Tugas Akhir ini. Setiap chart memiliki input yang hampir sama pada Gambar 8.

V. UJI COBA

Pada bagian ini menampilkan hasil uji coba dengan menggunakan studi kasus yang tersedia. Sumber data yang dicoba pada Tugas Akhir ini menggunakan PostgreSQL. Data yang diambil bagian transaksi penjualan pada ERP. Data penjualan yang diberikan dicampur dengan transaksi POS.

A. Studi Kasus

Pada studi kasus ini akan difokuskan pada memproses transaksi pada semester 1 2020. Tujuan akhir dari studi kasus ini untuk menentukan kemampuan sistem Tugas Akhir ini untuk mengolah data yang akan menampilkan sesuai dengan konfigurasi user untuk menampilkan Key Index Performance pada semester 1 2020.

Table	Engine	Collation	Data Length ¹	Index Length ²	Data Free	Auto Increment	Rows ³	Comment
MAKMUR_BARANG	table		6,758,400	647,168	?	?	21,046	
MAKMUR_BRANCH	table		8,192	24,576	?	?	17	
MAKMUR_DPENJUALAN	table		901,562,368	203,735,040	?	?	5,137,012	
MAKMUR_HPENJUALAN	table		504,889,344	46,055,424	?	?	2,049,159	
MAKMUR_SUPPL	table		3,416,064	417,792	?	?	12,216	
5 in total		Indonesian_Indonesia.1252	1,416,634,368	250,880,000	0			

Gambar. 9. Informasi Sumber Data.

Pada Gambar 9 merupakan tabel yang akan digunakan sebagai uji coba pada Tugas Akhir ini, tabel yang difokuskan adalah MAKMUR_DPENJUALAN, dan tabel MAKMUR_HPENJUALAN.



Gambar. 10. Konfigurasi Desain Desain Dashboard Baris 1-4

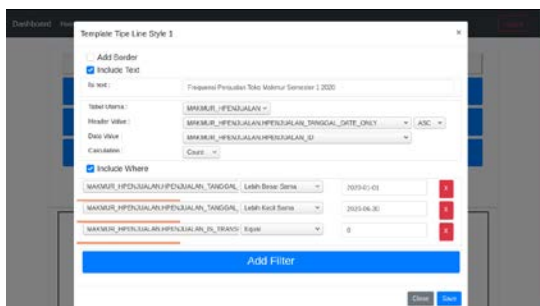
Pada Gambar 10 merupakan konfigurasi untuk menampilkan desain yang memfokuskan pada tabel MAKMUR_HPENJUALAN. Konfigurasi desain dashboard pada baris 1 memiliki 1 kolom saja yang berisi text yang menampung informasi text yang menjadi informasi pemilik dari sumber data. Pada kolom 2 merupakan line chart yang menampung frekuensi penjualan dalam satu semester. Informasi setiap bulan dibagi pada dashboard baris 3-4 yang berisi setiap kolom 3 pie chart.

Konfigurasi yang dilakukan pada dashboard baris 3-4 memiliki tipe chart yang sama. Konfigurasi yang diberikan pada masing-masing kolom setiap baris desain dashboard hampir sama. Perbedaan masing-masing kolom pada desain dashboard baris 3-4 adalah filter tanggalnya saja menyesuaikan dengan informasi yang akan disampaikan.



Gambar. 11. Konfigurasi Desain Desain Dashboard pada Baris 1

Konfigurasi desain yang diberikan pada gambar 11 menggunakan template text pada baris 1. Konfigurasi pada penambahan garis di bawah untuk sebagai pembatas di desain dashboard baris dibawahnya.

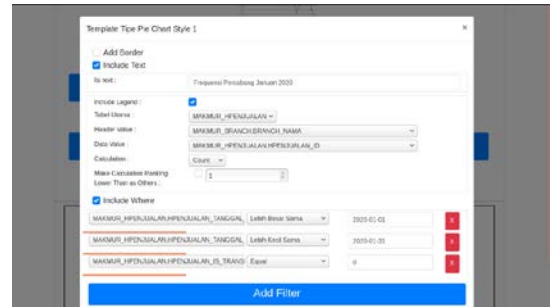


Gambar. 12. Konfigurasi Desain Desain Dashboard pada Baris 2

Desain dashboard yang diperlukan pada baris 2 diperlukan untuk menampilkan keseluruhan transaksi pada

semester 1 2020. Tabel utama yang digunakan pada baris 2 adalah MAKMUR_HPENJUALAN. Informasi tanggal yang didapatkan dapat diambil dari tabel sumber data HPENJUALAN_TANGGAL_DATE_ONLY. Perhitungan yang digunakan merupakan perhitungan transaksi yang dikelompokkan berdasarkan data header.

Selain itu transaksi yang ditampilkan bukan merupakan transfer barang yang merupakan perpindahan barang antar cabang. Hal ini difilter berdasarkan dari HPENJUALAN_IS_TRANSFER.



Gambar. 13. Konfigurasi Desain Desain Dashboard pada Baris 3-4

Konfigurasi yang digunakan pada setiap kolom desain dashboard baris 3-4. Konfigurasi diperlukan menyediakan nama cabang yang telah diproses. Nama cabang yang diproses didapatkan dari tabel MAKMUR_BRANCH, dengan didapatkan dari kolom BRANCH_NAMA. Hal ini dapat dilakukan dikarenakan relasi pada MAKMUR_HPENJUALAN telah dikonfigurasi memiliki foreign key ke tabel MAKMUR_BRANCH. Konfigurasi perhitungan menggunakan perhitungan count.

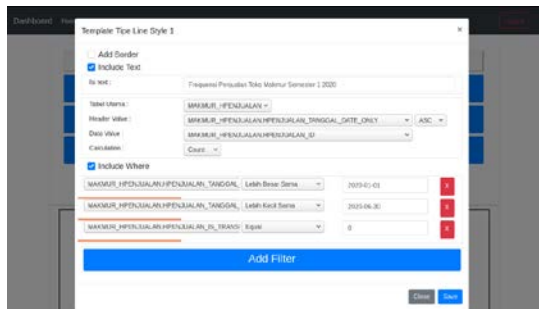


Gambar. 14. Konfigurasi Desain Desain Dashboard Baris 5-9

Pada gambar 14 informasi dibagi menjadi 3 bagian. Pada bagian pertama, terdapat pada baris 5 dan memiliki konfigurasi line chart yang menampung omset perhari yang ditampilkan pada line chart. Pada bagian kedua menampung kuartal 1 2020 pada baris 6-7, pada baris 6 terisi dengan template bertipe text dan baris ke 7 berisi pie chart.

Sedangkan pada bagian ketiga, menampung kuartal 2 2020 pada baris 8-9. Template yang dibutuhkan pada bagian ketiga adalah pada baris 8 menggunakan template text yang berisi tentang ringkasan total omset yang didapatkan pada bulan tersebut. Sedangkan pada pie chartnya menampung total omset yang dibedakan berdasarkan cabang yang ada. Konfigurasi yang digunakan pada bagian ketiga akan digunakan juga pada bagian kedua yang dibedakan

berdasarkan periodenya.

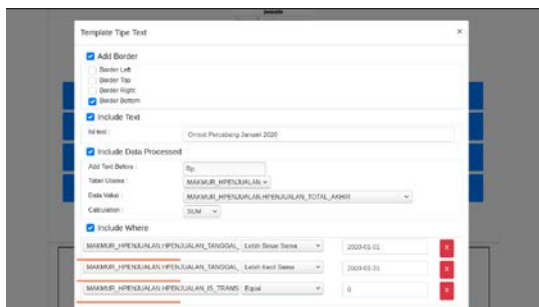


Gambar. 15. Konfigurasi Desain Desain Dashboard pada Baris 5

Konfigurasi desain yang diberikan pada gambar 5.7 menggunakan template line chart. Konfigurasi pada penambahan garis di bawah dan bawah untuk sebagai pembatas.. Tabel utama yang digunakan pada baris 2 adalah MAKMUR_HPENJUALAN. Informasi tanggal didapatkan dari HPENJUALAN_TANGGAL_DATE_ONLY dan diurutkan berdasarkan ASC.

Pada proses perhitungan, perhitungan menggunakan perintah sum yang menghitung dari kolom HPENJUALAN_TOTAL_AKHIR. Proses perhitungan transaksi yang dilakukan, akan dikelompokkan berdasarkan tanggal transaksi dilakukan pada kolom HPENJUALAN_TANGGAL_DATE_ONLY.

Selain itu transaksi yang ditampilkan bukan merupakan transfer barang yang merupakan perpindahan barang antar cabang. Hal ini difilter berdasarkan dari HPENJUALAN_IS_TRANSFER. Sedangkan filter tanggal difilter pada HPENJUALAN_TANGGAL_DATE_ONLY dengan data yang telah disimpan dengan format (yyyy-mm-dd).



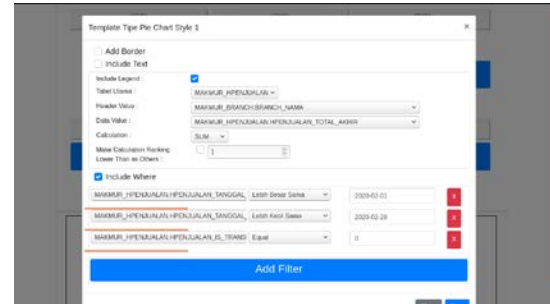
Gambar. 16. Konfigurasi pada Baris 6 Kolom 1

Pada konfigurasi template baris 6 merupakan informasi perhitungan perperiode bulan. Konfigurasi juga akan sama dilakukan pada baris 8 yang hanya membedakan konfigurasi pemilahan data yang akan dihitung. Pada setiap baris 6 memiliki konfigurasi terdapat garis dibawahnya untuk membantu data terlihat rapi. Saat menampilkan data kalkulasi, akan ditambahkan sebuah text mata uang rupiah yang akan ditambahkan sebelum tulisan hasil transaksi ditampilkan.

Perhitungan data akan dilakukan terhadap tabel MAKMUR_HPENJUALAN dan data yang akan dihitung berasal dari kolom HPENJUALAN_TOTAL_AKHIR.

Perhitungan dilakukan dengan tipe kalkulasi sum, dan kalkulasi ini dilakukan tanpa pengelompokan seperti template desain dashboard yang lain.

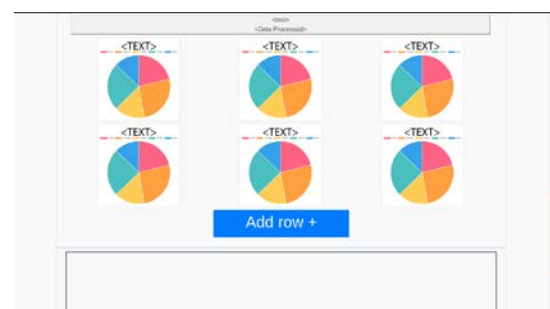
Pemilihan data yang digunakan sebagai kalkulasi mengambil antara periode sesuai dengan informasi yang akan disampaikan. Selain itu, transaksi yang dihitung juga bukan merupakan transfer barang antar cabang yang dipilih pada kolom HPENJUALAN_IS_TRANSFER dengan value 0.



Gambar. 17. Konfigurasi pada Baris 7 Kolom 1

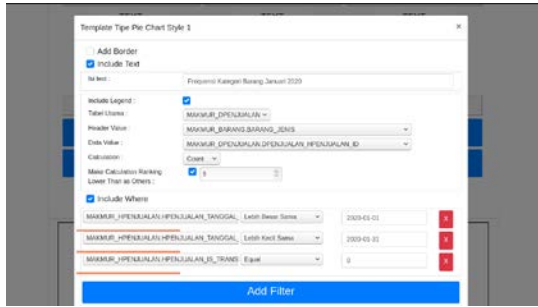
Pada konfigurasi baris 7 merupakan konfigurasi menampung pie chart yang berisi pada setiap kolomnya periode perbulan. Dalam perhitungan yang dilakukan, tabel data utama menggunakan MAKMUR_HPENJUALAN, sedangkan untuk data header menggunakan tabel dari relasi lain yaitu dari tabel MAKMUR_BRANCH dengan kolom BRANCH_NAMA.

Perhitungan yang akan disajikan pada pie chart, didapatkan pada kolom HPENJUALAN_TOTAL_AKHIR. Perhitungan dengan menggunakan tipe kalkulasi sum. Filter data yang dilakukan disesuaikan dengan template text yang ada di atas pie chart. Konfigurasi pada baris 7 juga akan sama digunakan pada baris 9 yang dibedakan penyesuaian filter periode yang akan diproses.



Gambar. 18. Desain Baris 10-12

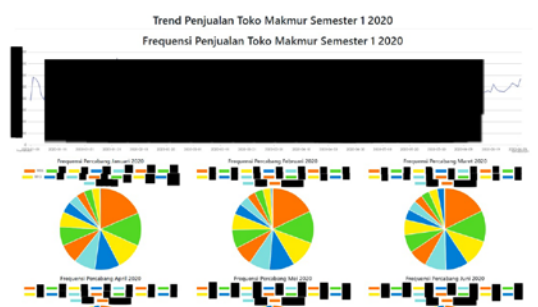
Pada Gambar 18 merupakan desain lanjutan setelah menampilkan omset penjualan. Konfigurasi yang dilakukan pada baris 10 merupakan informasi text yang hanya menampilkan tulisan saja. Sedangkan pada baris 11-12 merupakan pie chart yang menampung keseluruhan perhitungan kategori yang terjadi transaksi pada sumber data dalam studi kasus.



Gambar. 19. Konfigurasi pada Baris 11 Kolom 1

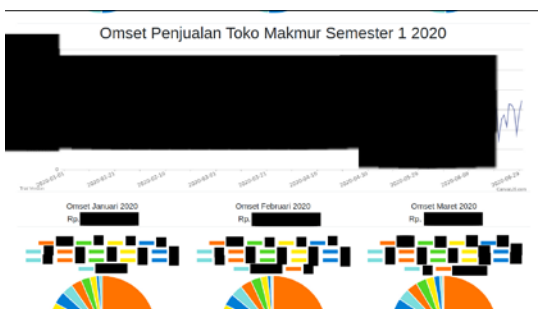
Konfigurasi pada Gambar 19 merupakan konfigurasi yang akan digunakan pada setiap kolom dan baris untuk menampung frekuensi penjualan perkategori barang, pembeda pada setiap chart yang ditampilkan adalah periode yang akan ditampilkan.

Tabel data utama dari sumber data yang digunakan menggunakan tabel MAKMUR_DPENJUALAN. Data kategori barang didapatkan pada tabel MAKMUR_BARANG kolom BARANG_JENIS yang memiliki relasi dengan MAKMUR_DPENJUALAN kolom DPENJUALAN_BARANG_KODE. Kolom yang digunakan akan diabaikan dikarenakan menggunakan tipe kalkulasi count.



Gambar. 20. Hasil Dashboard pada Frekuensi Penjualan

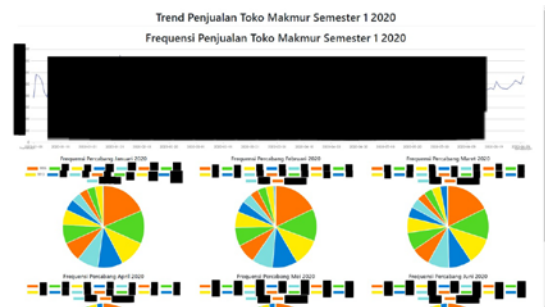
Pada Gambar 20 merupakan hasil yang didapatkan dari konfigurasi desain pada Gambar 10 yang telah diberikan pada bagian frekuensi yang diproses pada studi kasus yang diberikan. Chart pada dashboard yang ditampilkan sesuai dengan konfigurasi desain dashboard user.



Gambar. 21. Hasil Hasil Dashboard pada Omset Perkategori

Pada Gambar 21, merupakan hasil yang didapatkan dari konfigurasi desain yang telah diberikan dari Gambar 14. Dashboard menyajikan data yang sesuai dengan konfigurasi

beserta input text yang diberikan pada masing-masing pie chart.



Gambar. 22. Hasil Dashboard pada Penjualan Frekuensi Perkategori

Gambar di atas pada Gambar 22 merupakan hasil dari konfigurasi yang diberikan pada Gambar 18 yang merupakan konfigurasi untuk menyajikan data frekuensi penjualan berdasarkan perkategori item. Data yang diberikan mendapatkan value tambahan others yang menampung jumlah seluruh data yang lebih kecil dari 9 penjualan kategori terbesar.

B. Performa System

Pada bagian ini akan menjelaskan bagaimana performa sistem mengolah data, menyajikan data. Saat pengolahan data waktu yang dihitung adalah waktu dalam pembuatan suatu chart pada baris / kolom dengan skenario filter pengambilan data besar yang diambil dan pengambilan data kecil yang diambil.

```
Start Mongo Pipeline :1600896289.9749134
target_pipeline : 41a337a3dd1abf9e3c80cc3378f8a18d_3
pipeline:
[{'$project': {'_id': 0, 'MAKMUR_HPENJUALAN-- --HPENJUALAN-- --HPENJUALAN_IS_TRANSFER': 1}}, {'$match': {'$and': ['2020-06-30']}}, {'MAKMUR_HPENJUALAN-- --HPENJUALAN_ANGGAL_DATE_ONLY', 'total': {'$sum': 1}}}, {'$sort': {'_id': 1}}]
generate template html
End Mongo Pipeline :1600896334.9642544
update kode template : a78a0186832cfae9285adc6f1e5c168a
```

Gambar. 23. Proses Pembuatan Data Utama pada Gambar 12

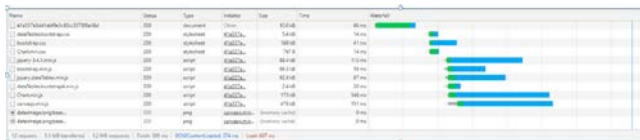
Pengecekan performa sistem dalam pembuatan untuk pengambilan data yang besar dilakukan berdasarkan studi kasus pada Gambar 12. Filter yang diambil untuk menampilkan data semester 1 2020 dengan pipeline data menyesuaikan dengan konfigurasi chart yang telah dikonfigurasi oleh user.

Waktu proses dilakukan pada format timestamp, untuk waktu awal dilakukan pada timestamp 1600896289 dengan selesai mengambil data dan mengolah data untuk disimpan diselesaikan dalam waktu 1600896334. Waktu yang dibutuhkan didapatkan dari selisih timestamp pengerjaan awal dan selesainya Dalam hal ini didapatkan lama waktu sistem untuk memproses data untuk disimpan menjadi sebuah chart yang siap adalah 45 detik. Perlu digaris bawahi, waktu ini tidak mempengaruhi lama waktu dashboard disajikan ke user.


```
Start Mongo Pipeline :1600899238.0980093
target_pipeline : 41a337a3dd1abf9e3c80cc3378f8a18d_3
pipeline:
[{'$project': {'_id': 0, 'MAKMUR_HPENJUALAN--__--HPENJUA
NGGAL_DATE_ONLY': 1, 'MAKMUR_HPENJUALAN--__--HPENJUA
TE_ONLY': {'$gte': '2020-01-01', '$lte': '2020-01-31'}}},
{'$MAKMUR_BRANCH--__--BRANCH_NAMA', 'total': {'$sum': 1}}]
-----
generate template html
End Mongo Pipeline :1600899271.3440733
update kode template : 55636abfc89dba6921c860f219984e9f
```

Gambar. 24. Proses Pembuatan Data Utama pada Gambar 13

Gambar diatas pada gambar 5.16 merupakan pengecekan bila data yang diolah merupakan data yang hasilnya sedikit setelah difilter. Waktu pengerjaan awal didapatkan dengan waktu timestamp 1600899238, dengan sistem setelah selesai menyimpan data tersebut ke dasboard dengan waktu timestamp 1600899271. Lama waktu pengerjaan didapatkan 33 detik.



Gambar. 21. Hasil Hasil Dashboard pada Omset Perkategori

Pada gambar diatas gambar 5.17 merupakan waktu yang diperlukan untuk membaca keseluruhan data yang didapatkan saat menampilkan keseluruhan dashboard pada studi kasus yang diberikan. Ukuran dari data yang telah dioleh memiliki ukuran 10kB dengan waktu loadnya 60 ms. Pada setiap garis timeline memiliki informasi per 100ms.

Total waktu yang diperlukan untuk menerima data dari server adalah 389ms. Proses dari sisi client untuk menampung data yang telah diberikan oleh server dengan waktu 374ms. Sedangkan waktu dari sisi client memproses keseluruhan data yang diperlukan untuk menampilkan dari data menjadi chart adalah 607ms.

VI. KESIMPULAN

Pada bagian ini akan dijelaskan mengenai beberapa kesimpulan yang didapat dari pembuatan website Tugas Akhir. Kesimpulan didapatkan berdasarkan perkembangan selama proses pembuatan website. Kesimpulan-kesimpulan yang didapat antara lain :

1. Disarankan penambahan query “order by” saat mengambil data dari sumber data database RDBMS, hal ini dikarenakan sistem akan mendapatkan nilai yang berubah-ubah untuk dapat men-generate data untuk data yang akan di ambil selanjutnya. Nilai ini, yang bisa kita sebut sebagai kursor karena fungsinya sebagai penunjuk suatu data untuk diambil selanjutnya. Maka dari itu, dalam Tugas Akhir ini melakukan proses query “order by” dengan settingan user yang user anggap kolom yang telah dipilih merupakan primary key. Dalam kasus ini, primary key dapat lebih dari 1, hal ini bertujuan untuk membantu sistem dalam pengambilan

data.

2. Saat memproses data yang besar, penulis juga mencoba melakukan query pengolahan data yang hampir sama ke database RDBMS PostgreSQL. Hal yang didapatkan di PostgreSQL justru lebih cepat untuk mengirimkan hasil query dari MongoDB. Kecepatan pengolahan data pada MongoDB tidak lebih cepat dari RDBMS PostgreSQL, tetapi struktur yang fleksibel pada MongoDB sangat dibutuhkan pada Tugas Akhir.
3. Penggunaan teknik cached page sangat membantu saat menampilkan tampilan yang tidak menerima input, contohnya pada menampilkan dashboard pada Tugas Akhir ini.
4. Bila sistem membutuhkan struktur data yang fleksibel, MongoDB adalah pilihan tepat sebagai penampung data.
5. Kecepatan insert data pada MongoDB sangat cepat, dan tipe field yang di tambahkan menyesuaikan dengan tipe data pada variabel yang di tambahkan.
6. Bootstrap CSS merupakan sebuah framework CSS yang sangat baik dalam menangani pembuatan desain website dengan sistem grid dan maupun yang responsif pada smartphone.
7. Penggunaan penyimpanan dengan tipe data JSON sangat memantu pada suatu sistem yang membutuhkan penyimpanan data yang fleksibel.

VII. SARAN

Pada bagian ini akan menjelaskan saran-saran yang didapat saat pembuatan Tugas Akhir ini. Berikut saran-saran yang dibagi menjadi beberapa poin, diantaranya :

1. Kecepatan proses pada MongoDB tidak lebih bagus daripada PostgreSQL. Tetapi bila penyimpanan sumber data dilakukan pada sistem, MongoDB sangat direkomendasikan.
2. Bila sistem melakukan pengambilan data besar yang memiliki fungsi join, sangat disarankan untuk sistem tersebut jalan parallel dengan sistem yang lainnya. Karena akan sangat mengganggu sistem yang lain dikarenakan bila tidak jalan secara parallel, sistem akan lebih banyak menunggu untuk mengambil data dari sumber data.
3. Sangat disarankan untuk arsitektur sistem multi server. Dengan pembagian sebagai berikut schedulling feeding, sistem utama, schedulling processing pipeline MongoDB, dan schedulling pembuatan hasil desain dashboard.
4. Diperlukan konsultasi dengan orang UI/UX untuk masalah tampilan supaya terlihat bagus.

UCAPAN TERIMA KASIH / ACKNOLEDGMENT

Puji syukur kepada Tuhan Yang Maha Esa atas berkat yang telah diberikannya, sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Selama penyusunan hingga selesainya Tesis ini, penulis telah banyak menerima dorongan, bantuan, dan perhatian dari berbagai pihak. Oleh

karena itu pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada :

1. Kedua orang tua, yang telah memberikan dukungan dan motivasi baik berupa material maupun spiritual dalam penyelesaian Tugas Akhir ini.
2. Bapak Dr. Joan Santoso, S.Kom., M.Kom. selaku dosen pembimbing yang mau membimbing dengan sabar hati dan mengajarkan banyak hal mengenai pembuatan aplikasi serta memberikan banyak nasihat dalam proses pengerjaan Tugas Akhir.
3. Bapak Dr. Decky Avrulukito, S.Sn, M.M. selaku dosen estetika yang membimbing penulis dalam pendesainan poster yang digunakan untuk Tugas Akhir ini.
4. Ibu Mrr. Dwi Yanti H., S.Pd., M.M. selaku dosen ecc yang membimbing penulis dalam pembuatan abstrak berbahasa Inggris.
5. Evan Leonardi, Alfonsus Januardi, Daniel Christanto, Muhammad Shodiq, S.Kom, Chandra Reza Gannes Prasetya yang merupakan teman seperjuangan yang turut ambil bagian memberi motivasi, semangat, dukungan dan ide kepada penulis.
6. Teman-teman di iSTTS yang tidak dapat disebutkan satu per satu atas dukungan, bantuan dan semangat yang telah diberikan pada penulis.

DAFTAR PUSTAKA

- [1] Aloto, Esayas, 2018. *Who's Using MongoDB and Why?*, Datavail, [Online]. Available at: <https://www.datavail.com/blog/whos-using-MongoDB-and-why/>
- [2] Niska. Christoffer, 2014, *Extending Bootstrap*, Packt Publishing.
- [3] Newbold. Paul, William L. Carlson and Betty, Thorne. 2007. *Statistics for Business and Economics 6th Edition*, Singapore: Pearson Education
- [4] Gutttag. John V, 2019. *Introduction to Computation and Programming Using Python: With Application to Understanding Data*, MIT Press.
- [5] Munzner. Tamara, 2014. *Visualization Analysis & Design*. CRC Press
- [6] Loyse. Greg, 2017. *Introduction to web development with Python and Django Documentation*
- [7] Django Software Foundation, 2019. *Django*. [Online]. Available: <https://djangoproject.com>
- [8] Chen. Hsinchun, H.L. Chiang. Roger, C. Ctorey. Veda, 2012, *Business Intelligence And Analytics: From Big Data To Big Impact*, MIS Quarterly
- [9] Deqing Li, Honghui Mei, Yi Shen, Shuang Su, Wenli Zhang, Juntao Wang, Ming Zu, Wei Chen, 2018. *ECharts: A declarative framework for rapid construction of web-based visualization*. State Key Lab of CAD&CG, Zhejiang University, China.

Arief Gunawan lahir di Surabaya, Jawa Timur, pada tahun 1995. Dia menyelesaikan studi S1 di program studi Teknik Informatika iSTTS pada tahun 2020. Minat penelitiannya adalah bidang Data Analyst, dan Big Data.