

BAB III

PELAKSANAAN MAGANG

3.1 Sistem Monitoring dan Evaluasi Pembangunan Kota Probolinggo

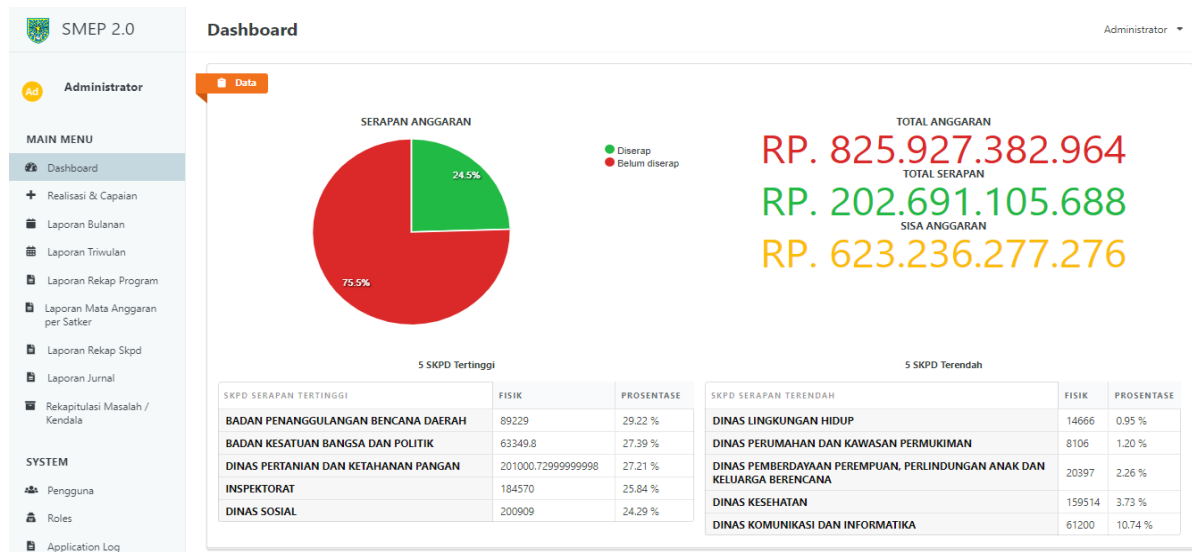
Bagian Administrasi Pembangunan Kota Probolinggo merupakan salah satu bagian pemerintahan Kota Probolinggo yang mengurus tentang program pembangunana Kota Probolinggo. Kota Probolinggo saat ini memiliki 39 SKPD, masing-masing SKPD memiliki program dan kegiatannya masing-masing. Anggaran dan realisasi keuangan yang ada pada program dan kegiatan tersebut wajib dimasukkan kedalam Sistem Monitoring dan Evaluasi Pembangunan. Setiap SKPD memiliki operator yang bertugas untuk memasukan input realisasi anggaran dari SKPD tersebut. Sementara itu, ada admin dari dinas pembangunan yang bertugas untuk mengelola pengguna dan melihat rekapitulasi data semua SKPD. SMEP memiliki 13 fitur dengan 5 fitur yang penulis kerjakan sebagaimana Tabel 3.1 fitur tersebut terbagi untuk dua aktor yaitu admin dan SKPD.

Tabel 3.1 Fitur SMEP

Fitur	Aktor Admin	Aktor SKPD
Dashboard	✓	✓
Manajemen Roles	✓	
Rekapitulasi Kendala	✓	
Manajemen Pengguna	✓	
Log Aktivitas	✓	

a. Dashboard

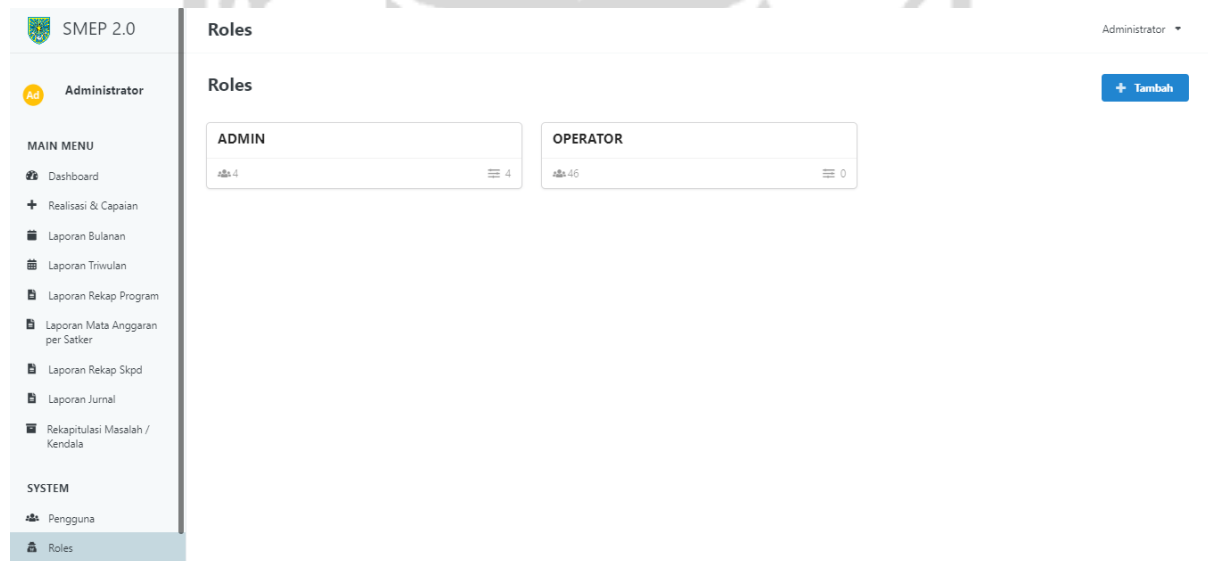
Fitur Dashboard Gambar 3.1 merupakan halaman awal yang akan menampilkan grafik anggaran, total anggaran, total serapan, sisa anggaran dan dua tabel yang berisikan serapan SKPD tertinggi dan terendah. Fitur ini dapat diakses oleh admin dan operator SKPD.



Gambar 3.1 Fitur Dashboard

b. Manajemen Roles

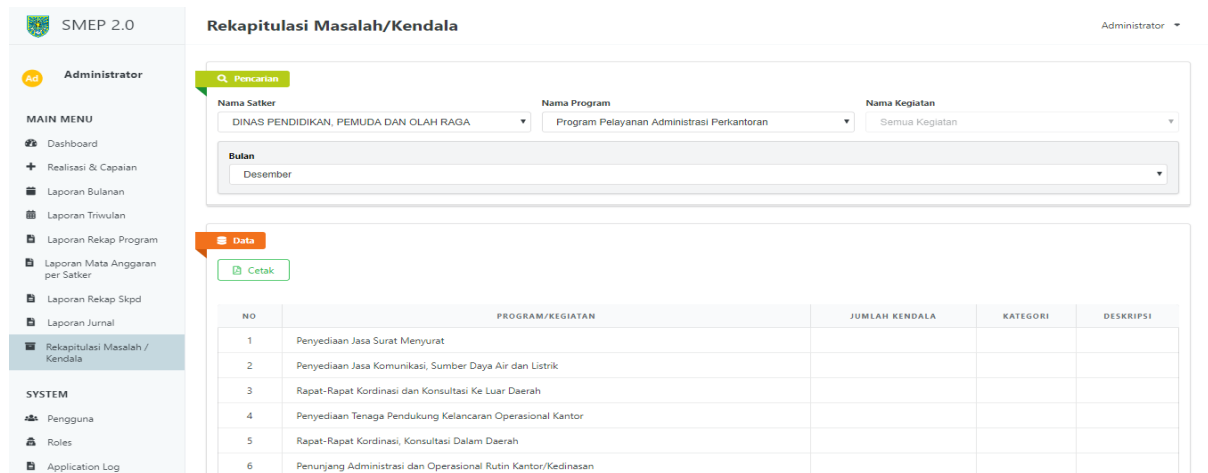
Admin dapat mengubah *role* pengguna sesuai dengan kebutuhan, terdapat dua *role* yaitu admin dan operator. Selain itu juga dapat menambah *role* seperti pada Gambar 3.2



Gambar 3.2 Fitur Manajemen Roles

c. Rekapitulasi Masalah/Kendala

Rekapitulasi Masalah/Kendala berisi kendala yang dihadapi oleh SKPD ketika melakukan penyerapan anggaran pada suatu program atau kegiatan. Fitur ini akan menampilkan kendala dari SKPD yang bersangkutan apabila diakses oleh operator SKPD, sementara dari sisi admin dapat menampilkan semua kendala dari satker. Tampilan halaman sebagaimana Gambar 3.3



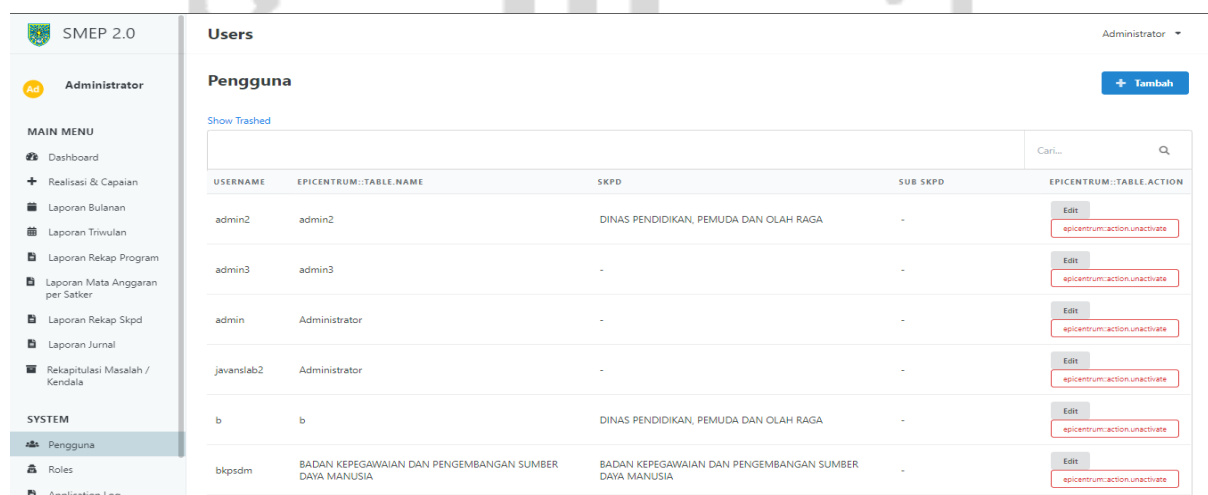
The screenshot displays the 'Rekapitulasi Masalah/Kendala' page in the SMEP 2.0 system. The interface includes a sidebar menu with options like 'Dashboard', 'Laporan Bulanan', and 'Rekapitulasi Masalah / Kendala'. The main content area features a search bar and several filter dropdowns: 'Nama Satker' (DINAS PENDIDIKAN, PEMUDA DAN OLAH RAGA), 'Nama Program' (Program Pelayanan Administrasi Perkantoran), 'Nama Kegiatan' (Semua Kegiatan), and 'Bulan' (Desember). Below the filters is a 'Data' section with a 'Cetak' button and a table listing issues.

NO	PROGRAM/KEGIATAN	JUMLAH KENDALA	KATEGORI	DESKRIPSI
1	Penyediaan Jasa Surat Menyurat			
2	Penyediaan Jasa Komunikasi, Sumber Daya Air dan Listrik			
3	Rapat-Rapat Kordinasi dan Konsultasi Ke Luar Daerah			
4	Penyediaan Tenaga Pendukung Kelancaran Operasional Kantor			
5	Rapat-Rapat Kordinasi, Konsultasi Dalam Daerah			
6	Penunjang Administrasi dan Operasional Rutin Kantor/Kedinasan			

Gambar 3.3 Fitur Rekapitulasi Masalah/Kendala

d. Manajemen Pengguna

Admin sebagai yang mengurus sistem memiliki akses untuk menambah, menghapus dan memberikan akses kepada pengguna. Daftar pengguna ditampilkan dalam sebuah tabel dengan tampilan Gambar 3.4



The screenshot shows the 'Users' management page in the SMEP 2.0 system. The page title is 'Pengguna' and it includes a '+ Tambah' button. Below the title is a search bar and a table listing users. The table has columns for USERNAME, EPICENTRUM::TABLE.NAME, SKPD, SUB SKPD, and EPICENTRUM::TABLE.ACTION. Each row includes an 'Edit' button and a red 'epicentrum:action_unactivate' button.

USERNAME	EPICENTRUM::TABLE.NAME	SKPD	SUB SKPD	EPICENTRUM::TABLE.ACTION
admin2	admin2	DINAS PENDIDIKAN, PEMUDA DAN OLAH RAGA	-	Edit epicentrum:action_unactivate
admin3	admin3	-	-	Edit epicentrum:action_unactivate
admin	Administrator	-	-	Edit epicentrum:action_unactivate
javanslab2	Administrator	-	-	Edit epicentrum:action_unactivate
b	b	DINAS PENDIDIKAN, PEMUDA DAN OLAH RAGA	-	Edit epicentrum:action_unactivate
bkpsdm	BADAN KEPEGAWAIAN DAN PENGEMBANGAN SUMBER DAYA MANUSIA	BADAN KEPEGAWAIAN DAN PENGEMBANGAN SUMBER DAYA MANUSIA	-	Edit epicentrum:action_unactivate

Gambar 3.4 Fitur Manajemen Pengguna

e. Log Aktivitas

Admin dapat membuka akses realisasi dan capaian pada bulan tertentu apabila operator belum menginputkan data setelah batas waktu input berakhir. Bulan yang dapat dibuka aksesnya adalah bulan yang sudah terlewati sebelum bulan saat ini. Halaman fitur ini seperti yang terpampang pada Gambar 3.5

NO	NAMA PENGGUNA	TANGGAL	NO DPA	PROGRAM	KEGIATAN	BULAN	TAHUN	DESKRIPSI
1	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2020-01-10 15:24:46	300800.300801.08.005.5.2	Program Peningkatan Kapasitas Lembaga Perwakilan Rakyat Daerah	Kegiatan Reses	11	2019	Terdapat realisasi yang melebihi LK pada DPA yaitu ATK senilai 157.500,-
2	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2020-01-10 14:40:25	300800.300801.02.023.5.2	Program Peningkatan Sarana dan Prasarana Aparatur	Pemeliharaan Rutin/Berkala Mobil Jabatan	12	2019	Terdapat realisasi yang melebihi LK pada DPA yaitu Bahan Kebersihan senilai 525.000,-
3	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2020-01-10 14:15:55	300800.300801.01.027.5.2	Program Pelayanan Administrasi Perkantoran	Penunjang Administrasi dan Operasional Rutin Kantor/Kedinasan	12	2019	Terdapat realisasi yang melebihi LK pada DPA yaitu ATK senilai 10.348.150,- dan Cetak senilai 4.200.000,-
4	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2020-01-09 10:21:20	300800.300801.06.010.5.2	Program Peningkatan Kuantitas dan Kualitas Pelayanan Publik	Pengelolaan Informasi dan Data	12	2019	Terdapat realisasi yang melebihi LK pada DPA yaitu Publikasi senilai 37.500.000,-
5	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2020-01-09 10:14:54	300800.300801.06.010.5.2	Program Peningkatan Kuantitas dan Kualitas Pelayanan Publik	Pengelolaan Informasi dan Data	11	2019	Terdapat realisasi yang melebihi LK pada DPA yaitu Publikasi senilai 22.810.000,-
6	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2020-01-08 08:51:14	300800.300801.02.032.5.2	Program Peningkatan Sarana dan Prasarana Aparatur	Pemeliharaan Rutin/Berkala Alat Kantor, Alat Rumah Tangga, Alat Studio dan Alat Komunikasi	11	2019	Terdapat realisasi yang melebihi LK pada DPA senilai 17.485.000,- yaitu Alat Listrik (5.300.000,-) dan Habis Pakai (12.185.000,-)
7	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2020-01-08 15:53:08	300800.300801.01.027.5.2	Program Pelayanan Administrasi Perkantoran	Penunjang Administrasi dan Operasional Rutin Kantor/Kedinasan	12	2019	Terdapat realisasi yang melebihi LK pada DPA yaitu ATK senilai 10.348.150,-
8	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2020-01-08 10:24:26	300800.300801.01.022.5.2	Program Pelayanan Administrasi Perkantoran	Penyediaan Peralatan dan Bahan Kebersihan	12	2019	Terdapat realisasi yang melebihi LK pada DPA senilai 2.660.000,-
9	SEKRETARIAT DEWAN PERWAKILAN RAKYAT DAERAH	2019-11-03	300800.300801.08.004.5.2	Program Peningkatan Kapasitas Lembaga Perwakilan Rakyat Daerah	Rapat-papat Paripurna	10	2019	Terdapat Realisasi yang melebihi LK pada DPA yaitu Penggandaan Risalah Fraksi, Sumpah/Janji Pimp. dan Pengumuman/Pretapan Calon Pimp. senilai

Gambar 3.5 Fitur Log Aktivitas

SMEP selain memiliki 13 fitur, memiliki 38 tabel dalam basis data MySQL. Tabel dengan jumlah *record* terbanyak adalah tabel audits yang berisi *log* dari aktivitas pengguna dengan jumlah *record* 1.050.411, sementara itu tabel *sikd_satker* adalah tabel dengan *record* paling sedikit yaitu 35. Untuk tabel dengan *field* paling sedikit adalah tabel *acl_roles*, tabel tersebut berisi aktor apa saja yang ada dalam SMEP. Sementara itu tabel dengan *field* paling banyak adalah tabel *dpa_dpa* dengan 41 *field*. Terakhir, relasi yang ada dalam basis data SMEP dilakukan secara implisit.

3.2 Multi Join Query

SMEP merupakan sebuah sistem yang memiliki database dengan tabel dan data yang banyak. Database yang digunakan untuk SMEP adalah MySQL. MySQL merupakan salah satu database relasional di mana satu tabel dapat memiliki relasi dengan satu atau banyak tabel. Untuk itu ketika melakukan pengolahan data perlu diperhatikan bagaimana cara melakukannya. Sebelum membuat query terdapat beberapa tahapan yang perlu dilakukan yaitu analisis kebutuhan sistem dan data, analisis table, pembuatan skema *query*, implementasi *query* sampai dengan pengujian *query*.

3.2.1 Tahapan *Query*

Sebelum membuat atau implementasi *query* terdapat beberapa tahapan yang perlu dilakukan. Tahapan yang perlu dilakukan adalah analisis kebutuhan sistem dan data, analisis tabel, pembuatan skema *query*, implementasi dan pengujian. Hal tersebut untuk memastikan data apa yang diperlukan dan sebaiknya bagaimana cara melakukan *query*. Berikut adalah detail dari setiap tahapan

a. Analisis Kebutuhan Sistem dan Data

Sebelum membuat sebuah *query* perlu dilakukan analisis data apa saja yang sistem butuhkan kemudian data apa saja yang perlu untuk ditampilkan. Untuk melakukannya perlu dibagi menjadi masing-masing fitur. Hasilnya seperti Tabel 3.2

Tabel 3.2 Hasil Analisis Kebutuhan Sistem dan Data

Fitur	<i>Query</i>	Data yang ditampilkan
Dashboard	SKPD Serapan Tertinggi	Nama SKPD, Realisasi Fisik, Prosentase
	SKPD Serapan Terendah	Nama SKPD, Realisasi Fisik, Prosentase
	Total Anggaran	Total Anggaran
	Total Serapan	Total Serapan
	Sisa Anggaran	Sisa Anggaran
Log Aktivitas	Log Kendala	Nama SKPD, Tanggal Log, No DPA, Program, Kegiatan, Bulan, Tahun, Deskripsi Kendala

b. Analisis Tabel

Setelah mendapatkan data apa yang perlu ditampilkan, langkah selanjutnya adalah memetakan dari tabel apa saja data tersebut. Hasilnya ada pada Tabel 3.3

Tabel 3.3 Hasil Analisis Tabel

<i>Query</i>	Tabel
SKPD Serapan Tertinggi	smep_realisasi, dpa_dpa, mata_anggaran, rincian_mata_anggaran, sikd_satker
SKPD Serapan Terendah	smep_realisasi, dpa_dpa, mata_anggaran, rincian_mata_anggaran, sikd_satker
Total Anggaran	dpa_dpa, dpa_mata_anggaran
Total Serapan	dpa_dpa, dpa_mata_anggaran
Sisa Anggaran	dpa_dpa, dpa_mata_anggaran
Log Kendala	audits, kendala_kegiatan, sikd_prog, sikd_kgtn, dpa_dpa

c. Skema *Query*

Tabel yang sudah diketahui tersebut untuk menentukan apakah perlu *join* dan *join* seperti apa yang diperlukan. Selain itu, menentukan klausa apa saja yang diperlukan untuk membuat *query*. Daftar skema tersebut ada dalam Tabel 3.4

Tabel 3.4 Skema Query

<i>Query</i>	Klausa <i>Query</i>
SKPD Serapan Tertinggi	<i>Sum, Subquery, Inner Join, Group By, Order By, Limit</i>
SKPD Serapan Terendah	<i>Sum, Subquery, Inner Join, Group By, Order By, Limit</i>
Total Anggaran	<i>Sum, Inner Join</i>
Total Serapan	<i>Sum</i>
Sisa Anggaran	<i>Sum, Inner Join, Subquery</i>
Log Kendala	<i>Inner Join, Order By</i>

d. Implementasi

Langkah selanjutnya setelah didapatkan data yang perlu ditampilkan, tabel yang diperlukan dan klausa apa saja yang dibutuhkan, tahapan selanjutnya adalah implementasi *query*. *Query* tersebut dibuat dengan menggunakan bahasa SQL melalui *console* yang ada pada *database client*.

1. *Query* SKPD Serapan Tertinggi

Query ini akan menampilkan 5 SKPD dengan serapan tertinggi dengan melibatkan dua table untuk mendapatkan anggaran dan rinciannya yaitu *dpa_dpa* dan *sikd_satker*. Sementara itu untuk mendapatkan hasil perhitungan lainnya dibutuhkan lima *subquery* dengan tabel *dpa_dpa*, *smep_realisasi*, *dpa_mata_anggaran* dan *dpa_rincian_mata_anggaran*. Kemudian akan dilakukan *group by* berdasarkan *id_satker* dan akan diambil lima yang tertinggi. *Query* yang sudah dibuat ada pada Gambar 3.6

```

SELECT sikd_satker.id_sikd_satker,
       sikd_satker.nama,
       (SELECT Sum(realisasi_fisik)
        FROM dpa_dpa
         INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
        WHERE tahun = '2019'
         AND dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker),
       (SELECT SUM(volume)
        FROM dpa_mata_anggaran
         INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
        where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker),
       (
         (SELECT Sum(realisasi_fisik)
          FROM dpa_dpa
           INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
          WHERE tahun = '2019'
           AND dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
        /
         (
           (SELECT SUM(volume)
            FROM dpa_mata_anggaran
             INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
            where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
          +
           (SELECT SUM(dpa_rincian_mata_anggaran.volume)
            FROM dpa_rincian_mata_anggaran
             INNER JOIN dpa_mata_anggaran
              ON dpa_mata_anggaran.id_dpa_mata_anggaran =
                 dpa_rincian_mata_anggaran.dpa_mata_anggaran_id
             INNER JOIN dpa_dpa
              ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
            where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
          )
        )
        * 100
       ) as presentase
FROM dpa_dpa
     INNER JOIN sikd_satker ON sikd_satker.id_sikd_satker = dpa_dpa.sikd_satker_id
GROUP BY sikd_satker.id_sikd_satker
ORDER BY presentase desc
LIMIT 5;

```

Gambar 3.6 *Query* SKPD Serapan Tertinggi

2. Query SKPD Serapan Terendah

Query ini akan menampilkan 5 SKPD dengan serapan terendah dengan melibatkan dua table untuk mendapatkan anggaran dan rinciannya yaitu dpa_dpa dan sikd_satker. Untuk mencari SKPD dengan serapan terendah, cara yang dilakukan sama dengan mencari SKPD Serapan tertinggi. Hanya saja yang membedakan adalah akan diurutkan dari yang terendah seperti pada Gambar 3.7

```

SELECT sikd_satker.id_sikd_satker,
       sikd_satker.nama,
       (SELECT Sum(realisasi_fisik)
        FROM dpa_dpa
         INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
        WHERE tahun = '2019'
         AND dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker),
       (SELECT SUM(volume)
        FROM dpa_mata_anggaran
         INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
        where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker),
       (
         (SELECT Sum(realisasi_fisik)
          FROM dpa_dpa
           INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
          WHERE tahun = '2019'
           AND dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
         /
         (
           (SELECT SUM(volume)
            FROM dpa_mata_anggaran
             INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
            where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
           +
           (SELECT SUM(dpa_rincian_mata_anggaran.volume)
            FROM dpa_rincian_mata_anggaran
             INNER JOIN dpa_mata_anggaran
              ON dpa_mata_anggaran.id_dpa_mata_anggaran =
                dpa_rincian_mata_anggaran.dpa_mata_anggaran_id
             INNER JOIN dpa_dpa
              ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
            where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
         )
         * 100
       ) as presentase
FROM dpa_dpa
     INNER JOIN sikd_satker ON sikd_satker.id_sikd_satker = dpa_dpa.sikd_satker_id
GROUP BY sikd_satker.id_sikd_satker
ORDER BY presentase asc
LIMIT 5;

```

Gambar 3.7 Query SKPD Serapan Terendah

3. Query Total Anggaran

Total Anggaran merupakan keseluruhan anggaran pembangunan yang dimiliki oleh Kota Probolinggo, untuk mendapatkannya dengan melakukan *join* dua tabel antara *dpa_dpa* dengan *dpa_mata_anggaran* untuk mendapatkan jumlah anggaran. Selain itu terdapat *filter* hanya untuk *dpa* yang berstatus 1 dan yang jenis kegiatannya adalah 'DpaSkpdKgtn'. *Query* untuk menampilkan seperti pada Gambar 3.8

```
SELECT SUM(jumlah)
FROM dpa_dpa
INNER JOIN dpa_mata_anggaran ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
WHERE status_dpa = 1 AND jns_dpa = 'DpaSkpdKgtn';
```

Gambar 3.8 Query Total Anggaran

4. Query Totap Serapan

Total Serapan adalah jumlah serapan yang mampu dilakukan oleh SKPD dari anggaran pembangunan yang ada. Untuk melakukan penjumlahan digunakan klausa *sum* untuk kolom *realisasi_keuangan* dari tabel *smep_realisasi*. Khusus untuk *query* yang ini bukan merupakan *multi join query* karena data yang didapatkan hanya dari satu tabel saja. *Query* terpampang pada Gambar 3.9

```
SELECT SUM(realisasi_keuangan)
FROM smep_realisasi WHERE tahun = now();
```

Gambar 3.9 Query Totap Serapan

5. Query Sisa Anggaran

Sisa Anggaran merupakan anggaran yang masih tersisa dari anggaran yang digunakan oleh SKPD. Caranya dengan mencari jumlah anggaran dengan klausa *sum* yang didapatkan dari *join* antara tabel *dpa_dpa* dan *dpa_mata_anggaran*. Serta dengan menambahkan *filter* yang status *dpa*nya 1 dan jenis *dpa* adalah 'DpaSkpdKgtn', kemudian hasilnya akan dikurangi dengan jumlah realisasi keuangan tahun ini yang ada pada Gambar 3.10

```
SELECT SUM(jumlah) - (SELECT SUM(realisasi_keuangan) FROM smep_realisasi WHERE tahun = now()) as sisa_anggaran
FROM dpa_dpa
INNER JOIN dpa_mata_anggaran ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
WHERE status_dpa = 1 AND jns_dpa = 'DpaSkpdKgtn';
```

Gambar 3.10 Query Sisa Anggaran

6. Log Kendala

Log Kendala merupakan query yang akan menampilkan senarai kendala yang dimasukkan oleh operator skpd, berupa tanggal, kegiatan, program dan kendala yang dihadapi. Pada Gambar 3.11 terdapat lima tabel yang akan digabungkan, kemudian akan dilakukan *filtering* untuk *auditable_type* yang sama dengan 'App\Models\KendalaKegiatan'. Setelah itu akan diurutkan dengan klausa *order by* berdasarkan yang terakhir kali diubah.

```
select
    dpa_dpa.no_dpa,
    sikd_prog.nm_prog,
    sikd_kgtn.nm_kgtn,
    kendala_kegiatan.bulan,
    kendala_kegiatan.tahun,
    audits.new_values
from audits
    join kendala_kegiatan on kendala_kegiatan.id = audits.auditable_id
    join sikd_prog on sikd_prog.id_sikd_prog = kendala_kegiatan.sikd_prog_id
    join sikd_kgtn on sikd_kgtn.id_sikd_kgtn = kendala_kegiatan.sikd_kgtn_id
    join dpa_dpa on dpa_dpa.id_dpa_dpa = kendala_kegiatan.dpa_dpa_id
where audits.auditable_type = 'App\Models\KendalaKegiatan'
order by audits.updated_at desc;
```

Gambar 3.11 Query Log Kendala

e. Pengujian

Langkah terakhir dari pembuatan *query* adalah dengan melakukan pengujian, akan diuji apakah berapa lama waktu yang dibutuhkan untuk melakukan *query*. Hasil dari pengujian waktu eksekusi *query* ada pada Tabel 3.5

Tabel 3.5 Hasil Pengujian Eksekusi Query

Query	Jumlah Data	Waktu Eksekusi
SKPD Serapan Tertinggi	465.210	10 Detik
SKPD Serapan Terendah	465.210	10 Detik
Total Anggaran	38.307	55 Milidetik
Total Serapan	248.096	700 Milidetik

Sisa Anggaran	286.403	700 Milidetik
Log Kendala Kegiatan	1.072.402	300 Milidetik

Setelah dilakukan pengujian untuk mengetahui waktu eksekusi sebuah *query*, terdapat beberapa hal yang dapat mempengaruhi waktu eksekusi suatu *query*. Selain banyaknya data, hal lain yang dapat mempengaruhi waktu eksekusi antara lain *join*, *subquery* dan *group by*. Hal tersebut dapat dilihat pada tabel Tabel 3.5 dimana pada Query untuk mencari SKPD Tertinggi dan SKPD Terendah memiliki waktu eksekusi sekitar 10 detik. Hal tersebut karena pada kedua *query* tersebut didalamnya terdapat lima *subquery* untuk melakukan operasi perhitungan, ditambah lagi dengan adanya *group by* untuk mengelompokkan berdasarkan id dari satker. Jika dibandingkan dengan *query* Sisa Anggaran yang sama-sama memiliki *subquery* perbedaan waktu eksekusi cukup jauh, hal itu dikarenakan *query* Sisa Anggaran hanya memiliki satu *subquery*, tidak ada *group by* dan jumlah data lebih sedikit. *Query* dengan data terbanyak yaitu Log Kendala Kegiatan memiliki waktu eksekusi yang lebih singkat dari pada SKPD Serapan Tertinggi dan Terendah. Hal tersebut karena tidak terdapat *subquery* maupun agregasi pada *query* tersebut, sehingga eksekusinya bisa lebih cepat walaupun dengan data yang lebih banyak.

3.2.2 Perbandingan Query Join dengan Subquery

Query yang sudah dilakukan untuk membuat beberapa fitur di SMEP melibatkan lebih dari satu tabel dalam penerapannya. *Join* tabel dapat dilakukan dengan beberapa cara yaitu dengan menggunakan klausa *join* atau bisa juga dengan menggunakan klausa *subquery*.

1. Query Total Anggaran

Untuk mencari total anggaran dari SMEP Probolinggo dapat dilakukan dengan cara melakukan *query join* seperti pada Gambar 3.12. Cara lain yang dapat dilakukan adalah dengan menggunakan *subquery* untuk mencari *primary key* dari tabel *dpa_dpa* yang sama dengan *foreign key* dari tabel *dpa_mata_anggaran* yang ada pada Gambar 3.13

```
SELECT SUM(jumlah)
FROM dpa_dpa
INNER JOIN dpa_mata_anggaran ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
WHERE status_dpa = 1 AND jns_dpa = 'DpaSkpdKgtm';
```

Gambar 3.12 Join Total Anggaran

```
SELECT sum(jumlah)
FROM dpa_mata_anggaran
WHERE dpa_dpa_id IN (SELECT id_dpa_dpa FROM dpa_dpa WHERE status_dpa = 1 AND jns_dpa = 'DpaSkpdKgtm');
```

Gambar 3.13 Subquery Total Anggaran

2. Query Sisa Anggaran

Selain untuk mencari total anggaran, untuk mencari sisa anggaran dari SMEP Kota Probolinggo juga diperlukan *query* dengan klausa *join* seperti pada Gambar 3.14

```
SELECT SUM(jumlah) - (SELECT SUM(realisasi_keuangan) FROM smep_realisasi WHERE tahun = now()) as sisa_anggaran
FROM dpa_dpa
INNER JOIN dpa_mata_anggaran ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
WHERE status_dpa = 1 AND jns_dpa = 'DpaSkpdKgtm';
```

Gambar 3.14 Join Sisa Anggaran

```
select sum(jumlah) - (SELECT sum(realisasi_keuangan) from smep_realisasi where tahun = '2019') as sisa_anggaran
from dpa_mata_anggaran
where dpa_dpa_id IN (SELECT id_dpa_dpa from dpa_dpa where status_dpa = 1 AND jns_dpa = 'DpaSkpdKgtm');
```

Gambar 3.15 Subquery Sisa Anggaran

Untuk mendapatkan hasil yang sama, dapat pula menggunakan *query* Gambar 3.15 yaitu dengan menggunakan *subquery* untuk mencocokkan *primary key* tabel *dpa_dpa* dengan *foreign key* tabel *dpa_mata_anggaran*.

3. Query SKPD Serapan Tertinggi

Query terakhir yang menggunakan *join* adalah untuk mencari SKPD dengan serapan tertinggi, tabel yang akan digabungkan adalah *dpa_dpa* dengan *sikd_satker*. Kolom untuk menggabungkan yaitu *primary key* dari tabel *sikd_satker* kolom *id_sikd_satker* dengan *foreign key* pada tabel *dpa_dpa* kolom *sikd_satker_id*. Alasan digabung dengan tabel *sikd_satker* karena data hasil *query* akan diurutkan berdasarkan satker dengan serapan tertinggi. Hasil *query* ada pada Gambar 3.16

```

SELECT sikd_satker.id_sikd_satker,
       sikd_satker.nama,
       (SELECT Sum(realisasi_fisik)
        FROM dpa_dpa
         INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
        WHERE tahun = '2019'
         AND dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker),
       (SELECT SUM(volume)
        FROM dpa_mata_anggaran
         INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
        where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker),
       (
         (SELECT Sum(realisasi_fisik)
          FROM dpa_dpa
           INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
          WHERE tahun = '2019'
           AND dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
        /
         (
           (SELECT SUM(volume)
            FROM dpa_mata_anggaran
             INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
             where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
          +
           (SELECT SUM(dpa_rincian_mata_anggaran.volume)
            FROM dpa_rincian_mata_anggaran
             INNER JOIN dpa_mata_anggaran
              ON dpa_mata_anggaran.id_dpa_mata_anggaran =
                 dpa_rincian_mata_anggaran.dpa_mata_anggaran_id
             INNER JOIN dpa_dpa
              ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
             where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
          )
        ) * 100
       ) as presentase
FROM dpa_dpa
     INNER JOIN sikd_satker ON sikd_satker.id_sikd_satker = dpa_dpa.sikd_satker_id
GROUP BY sikd_satker.id_sikd_satker
ORDER BY presentase desc
LIMIT 5;

```

Gambar 3.16 Join SKPD Serapan Tertinggi

Terdapat perbedaan saat menggunakan *subquery* sebagai pengganti *join*, kolom dari tabel yang akan diambil datanya ditukar untuk diletakkan setelah klausa *from* yang ada dalam Gambar 3.17

```

SELECT sikh_satker.id_sikh_satker,
       sikh_satker.nama,
       (SELECT Sum(realisasi_fisik)
        FROM dpa_dpa
         INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
        WHERE tahun = '2019'
         AND dpa_dpa.sikh_satker_id = sikh_satker.id_sikh_satker),
       (SELECT SUM(volume)
        FROM dpa_mata_anggaran
         INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
        where dpa_dpa.sikh_satker_id = sikh_satker.id_sikh_satker),
       (
         (SELECT Sum(realisasi_fisik)
          FROM dpa_dpa
           INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
          WHERE tahun = '2019'
           AND dpa_dpa.sikh_satker_id = sikh_satker.id_sikh_satker)
        /
         (
           (SELECT SUM(volume)
            FROM dpa_mata_anggaran
             INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
            where dpa_dpa.sikh_satker_id = sikh_satker.id_sikh_satker)
          +
           (SELECT SUM(dpa_rincian_mata_anggaran.volume)
            FROM dpa_rincian_mata_anggaran
             INNER JOIN dpa_mata_anggaran
              ON dpa_mata_anggaran.id_dpa_mata_anggaran =
                 dpa_rincian_mata_anggaran.dpa_mata_anggaran_id
             INNER JOIN dpa_dpa
              ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
            where dpa_dpa.sikh_satker_id = sikh_satker.id_sikh_satker)
          )
        * 100
       ) as presentase
FROM sikh_satker
WHERE id_sikh_satker IN (SELECT dpa_dpa.sikh_satker_id from dpa_dpa)
GROUP BY sikh_satker.id_sikh_satker
ORDER BY presentase desc
LIMIT 5;

```

Gambar 3.17 Subquery SKPD Serapan Tertinggi

Setelah dilakukan perbandingan dari segi SQL, selanjutnya adalah membandingkan dari segi waktu eksekusi *query* hasilnya pada Tabel 3.6

Tabel 3.6 Perbandingan Waktu Eksekusi *Join* dan *Subquery*

	<i>Join</i>	<i>Subquery</i>
Total Anggaran	150 Milidetik	150 Milidetik
Total Serapan	900 Milidetik	900 Milidetik
SKPD Serapan Tertinggi	7 Detik	7 Detik

Berdasarkan Tabel 3.6 *query* antara klausa *join* dan klausa *subquery* tidak terdapat perbedaan yang cukup signifikan dengan penggunaan dua klausa tersebut. Sementara itu penggunaan *subquery* lebih cocok digunakan jika hanya untuk *join* dengan dua tabel saja karena hanya dapat mengambil data pada tabel *dpa_mata_anggaran* seperti pada *query* Gambar 3.18

```
select sum(jumlah) - (SELECT sum(realisasi_keuangan) from smep_realisasi where tahun = '2019') as sisa_anggaran
from dpa_mata_anggaran
where dpa_dpa_id IN (SELECT id_dpa_dpa from dpa_dpa where status_dpa = 1 AND jns_dpa = 'DpaSkpdKgtn');
```

Gambar 3.18 Query Sisa Anggaran dengan Subquery

Query pada Gambar 3.18 hanya akan mengambil data dari kolom jumlah, sementara data yang ada pada tabel *dpa_dpa* hanya akan dipakai untuk klausa *where* yang digunakan untuk mencocokkan *primary key* dan *foreign key* yaitu *id* dari *dpa* pada Gambar 3.19

```
select
    dpa_dpa.no_dpa,
    sikd_prog.nm_prog,
    sikd_kgtn.nm_kgtn,
    kendala_kegiatan.bulan,
    kendala_kegiatan.tahun,
    audits.new_values
from audits
join kendala_kegiatan on kendala_kegiatan.id = audits.auditable_id
join sikd_prog on sikd_prog.id_sikd_prog = kendala_kegiatan.sikd_prog_id
join sikd_kgtn on sikd_kgtn.id_sikd_kgtn = kendala_kegiatan.sikd_kgtn_id
join dpa_dpa on dpa_dpa.id_dpa_dpa = kendala_kegiatan.dpa_dpa_id
where audits.auditable_type = 'App\\Models\\KendalaKegiatan'
order by audits.updated_at desc;
```

Gambar 3.19 Query Join Log Kendala

Tabel yang digabungkan sebanyak lima dan masing-masing dari tabel tersebut terdapat kolom dengan data yang akan diambil. Hal tersebut tidak bisa dilakukan hasilnya akan seperti pada Gambar 3.20

```
select
    dpa_dpa.no_dpa,
    sikd_prog.nm_prog,
    sikd_kgtn.nm_kgtn,
    kendala_kegiatan.bulan,
    kendala_kegiatan.tahun,
    audits.old_values,
    audits.new_values
from kendala_kegiatan
where id IN (SELECT auditable_id from audits where auditable_type = 'App\\Models\\KendalaKegiatan') and
sikd_prog_id IN (SELECT id_sikd_prog from sikd_prog) and
sikd_kgtn_id IN (SELECT sikd_kgtn_id from sikd_kgtn) and
dpa_dpa_id IN (SELECT id_dpa_dpa from dpa_dpa)
order by audits.updated_at desc;
```

Gambar 3.20 Query Subquery Log Kendala

Subquery akan lebih cocok digunakan untuk mencari serapan SKPD tertinggi dan SKPD yang menampilkan beberapa perhitungan. Masing-masing berbeda tabel yang ada pada Gambar 3.21. Sementara itu untuk menggabungkan tabel dengan *subquery* hanya bisa dilakukan jika melibatkan dua tabel saja.

```
SELECT sikd_satker.id_sikd_satker,
       sikd_satker.nama,
       (SELECT Sum(realisasi_fisik)
        FROM dpa_dpa
         INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
        WHERE tahun = '2019'
         AND dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker),
       (SELECT SUM(volume)
        FROM dpa_mata_anggaran
         INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
        where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker),
       (
         (SELECT Sum(realisasi_fisik)
          FROM dpa_dpa
           INNER JOIN smep_realisasi ON smep_realisasi.dpa_dpa_id = dpa_dpa.id_dpa_dpa
          WHERE tahun = '2019'
           AND dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
         /
         (
           (SELECT SUM(volume)
            FROM dpa_mata_anggaran
             INNER JOIN dpa_dpa ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
             where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
           +
           (SELECT SUM(dpa_rincian_mata_anggaran.volume)
            FROM dpa_rincian_mata_anggaran
             INNER JOIN dpa_mata_anggaran
              ON dpa_mata_anggaran.id_dpa_mata_anggaran =
                 dpa_rincian_mata_anggaran.dpa_mata_anggaran_id
             INNER JOIN dpa_dpa
              ON dpa_dpa.id_dpa_dpa = dpa_mata_anggaran.dpa_dpa_id
             where dpa_dpa.sikd_satker_id = sikd_satker.id_sikd_satker)
         )
       ) * 100
       ) as presentase
FROM dpa_dpa
     INNER JOIN sikd_satker ON sikd_satker.id_sikd_satker = dpa_dpa.sikd_satker_id
GROUP BY sikd_satker.id_sikd_satker
ORDER BY presentase desc
LIMIT 5;
```

Gambar 3.21 Query SKPD Tertinggi

Subquery tersebut akan langsung menghasilkan query yang sudah jadi, atau tidak perlu diolah ketika melakukan implementasi *coding*, jika tidak menggunakan *subquery* maka *query* tersebut akan dipisah-pisah dan akan diolah operasi perhitungannya saat melakukan *implementasi coding*.

