

## BAB IV

### REFLEKSI PELAKSANAAN MAGANG

#### 4.1 Sistem Monitoring dan Evaluasi Pembangunan Kota Probolinggo

SMEP merupakan sistem monitoring dan evaluasi pembangunan yang memiliki kurang lebih sebanyak 13 fitur. Sebagian besar dari fitur tersebut berupa laporan yang artinya tidak perlu memasukkan anggaran. Satu-satunya fitur yang mengharuskan untuk memasukkan anggaran adalah fitur Realisasi dan Kendala, fitur tersebut mengharuskan SKPD untuk memasukkan realisasi anggaran dan kendala kegiatan jika ada. Selain berupa isian formulir terdapat sekitar 8 fitur yang ada dalam sistem SIMSATKER (Tajuddin & Manan, 2015). Dari jumlah fitur yang ada menandakan bahwa SMEP memiliki lebih banyak fitur, berikut hasil lengkap daftar fitur yang ada pada Tabel 4.1

Tabel 4.1 Perbandingan Fitur

| SMEP                                | SIMSATKER                        |
|-------------------------------------|----------------------------------|
| 1. Dashboard                        | 1. Cetak Data Dinas              |
| 2. Realisasi & Capaian              | 2. Laporan Bulanan               |
| 3. Laporan Bulanan                  | 3. Laporan Triwulan              |
| 4. Laporan Triwulan                 | 4. Laporan Tahunan               |
| 5. Laporan Rekap Program            | 5. Laporan Proyek tahun Anggaran |
| 6. Laporan Mata Anggaran Per Satker | 6. Laporan Grafik                |
| 7. Laporan Rekap SKPD               | 7. Backup Data                   |
| 8. Laporan Jurnal                   | 8. Restore Data                  |
| 9. Rekapitulasi Masalah/Kendala     |                                  |
| 10. Pengguna                        |                                  |
| 11. Roles                           |                                  |
| 12. Akses Realisasi & Capaian       |                                  |
| 13. Log Aktivitas                   |                                  |

## 4.2 Query SQL Join

Implementasi *query* dalam SMEP merupakan salah satu hal yang menarik, pada proses pembuatan *query* melibatkan banyak tabel. Sebelum membuat *query* perlu dilakukan analisis untuk memastikan bahwa *query* yang dibutuhkan benar. Untuk melakukan *join* dapat dilakukan dengan menggunakan klausa *join* diikuti dengan mencocokkan antara *primary key* dengan *foreign key* dari dua tabel. Jika kita ingin menampilkan *foreign key*. Selain menggunakan klausa *Join* terdapat metode *join* lain yang dapat digunakan yaitu dengan menggunakan *subquery*. Dari hasil pembahasan pada bab sebelumnya dijelaskan bahwa ternyata *subquery* bisa digunakan untuk melakukan *join*, cara menggunakan *subquery* untuk *join* adalah dengan menambahkan klausa *in* setelah klausa *where* kemudian diikuti dengan *subquery* yang biasanya akan mengambil *foreign key* atau *primary key*.

Mahasiswa Informatika UII mendapatkan beberapa materi basis data dan *query sql*, mahasiswa mendapatkan materi tersebut melalui mata kuliah Pengembangan Sistem Informasi pada kurikulum 2016. Materi yang diajarkan berdasarkan hasil observasi penulis adalah *select*, *where*, *group function*, *join* dan *subquery*, mahasiswa mendapatkan materi itu berupa teori kemudian dilanjutkan dengan studi kasus. Pembahasan konsep *Query* dilakukan dalam enam pertemuan, dalam setiap pertemuan dijelaskan konsep dasarnya kemudian akan diikuti dengan contoh implementasinya. Diantara contoh *query* yang diajarkan ketika melaksanakan kuliah adalah seperti gambar Gambar 4.1

```

select department_id
from employment

select last_name, salary, job_id
from employees
where salary <= 3000 AND job_id = 1;

select avg(salary), max(salary), department_id
from employees
group by department_id;

select e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
from employees e
join departments d on e.department_id = d.department_id

select last_name, salary
from employees
where salary > (select salary from employees where las_name = 'Abel')

```

Gambar 4.1 Query Matakuliah Pengembangan Sistem Informasi

*Query* pada Gambar 4.1 merupakan beberapa *query* yang diajarkan ketika mengambil kelas Pengembangan Sistem Informasi, mulai dari penggunaan *select* hingga *subquery*. Namun pembahasan di kuliah tersebut hanya mencakup materi-materi konsep dasar penggunaan *sql query*, hal tersebut berbeda dengan apa yang penulis temui ketika melaksanakan magang. Hasil observasi yang dilakukan penulis *sql query* lebih kompleks, salah satunya *join table* lebih banyak tidak hanya satu tabel. Selain itu, banyak menggunakan klausa-klausa lain seperti *sum*, *case*, *if else* dan *self Join*.

Realita lain yang ditemui dari hasil observasi ketika melaksanakan magang adalah *programmer* pemula cenderung akan membuat *query* yang simpel kemudian pengolahan *query* seperti perhitungan atau *filtering* akan dilakukan saat implementasi *coding* melalui bahasa pemrograman atau *framework* tertentu seperti PHP atau Laravel. Hal tersebut kurang optimal sebab sebaiknya pengolahan data dilakukan saat *query*, jadi ketika implementasi *coding query* yang diolah adalah *query* yang sudah jadi atau sudah diolah datanya.

