

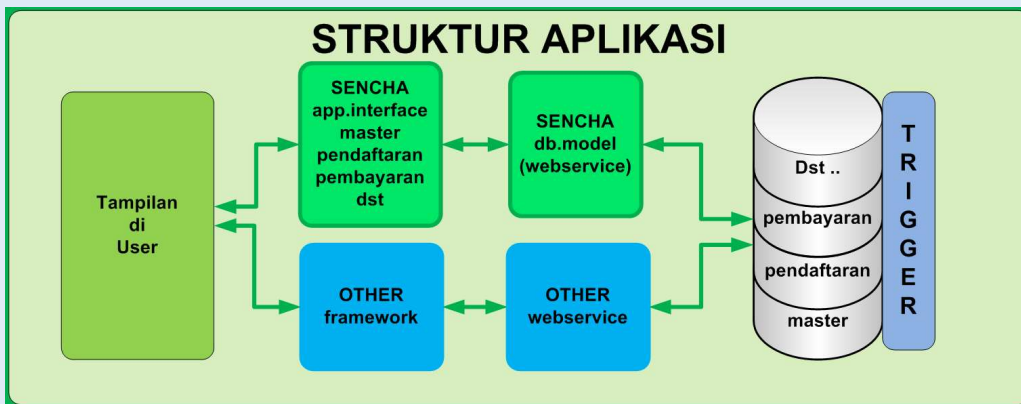
# SELAYANG PANDANG HASIL ANALISA APLIKASI SIMGOS

8 Agustus 2022

## ANALISA

### 1. Struktur Aplikasi

**i** SENCHA dan MySql



Struktur aplikasi yang ada :

menggunakan database mysql dengan memanfaatkan trigger sebagai otak untuk melakukan proses-proses aturan aplikasi.

Menggunakan framework Sencha untuk membuat tampilan layar kepada pengguna aplikasi, Dengan perantara db.model yang juga build in dalam framework Sencha.

Catatan :

Salah satu keuntungan dari aturan-aturan aplikasi yang dilakukan oleh database (trigger), adalah adanya kemungkinan pengembangan aplikasi dengan framework yang sama sekali berbeda tetapi menghasilkan perilaku data yang sama pada database

### 2. Pengelompokan data dan modul

**i** Struktur pengelompokan data sudah terkelompok hingga framework Sencha

Hal ini terlihat dari penamaan database 'pendaftaran' dan di framework Sencha juga terdapat kelompok objek 'pendaftaran', dan seterusnya. Hal ini tentunya akan memudahkan dalam melakukan pembuatan, penambahan, perbaikan terhadap modul-modul aplikasi, sehingga tidak mengganggu modul-modul yang lain. Juga memudahkan dalam melakukan identifikasi posisi aplikasi yang akan dikembangkan atau diperbaiki.

### 3. Trigger



#### Contoh trigger insert pada tabel pendaftaran

```
drop trigger pendaftaran.onAfterInsertPendaftaran;

create trigger onAfterInsertPendaftaran
on pendaftaran.pendaftaran for each row
BEGIN
    DECLARE VTAGIHAN CHAR(10);
    DECLARE VTARIF_ID INT;
    DECLARE VTARIF DECIMAL(60,2);

    SET VTAGIHAN = pembayaran.buatTagihan(NEW.NORM, NEW.NOMOR);
    IF pembayaran.isFinalTagihan(VTAGIHAN) = 0 THEN
        CALL cetakan.storeKarcis(NEW.NOMOR, 1, NEW.OLEH);
        IF NEW.PAKET > 0 OR NOT NEW.PAKET IS NULL THEN
            IF VTAGIHAN != '' THEN
                BEGIN
                    DECLARE VQTY DECIMAL(60,2);
                    DECLARE VPAKET_DETIL INT DEFAULT 0;
                    DECLARE VKARTU CHAR(11);

                    CALL master.getTarifPaket(NEW.PAKET, NEW.TANGGAL, VTARIF_ID, VTARIF);
                    CALL pembayaran.storeRincianTagihan(VTAGIHAN, NEW.NOMOR, 5, VTARIF_ID, 1, VTARIF, 0, 0, 0);

                    CALL master.inPaket(NEW.PAKET, 3, 1, VQTY, VPAKET_DETIL);
                    IF VPAKET_DETIL > 0 THEN
                        SELECT rt.REF_ID INTO VKARTU
                        FROM pembayaran.rincian_tagihan rt,
                             master.tarif_administrasi ta
                        WHERE rt.TAGIHAN = VTAGIHAN
                              AND rt.JENIS = 1
                              AND ta.ID = rt.TARIF_ID
                              AND ta.ADMINISTRASI = 1
                              AND rt.STATUS = 1;

                        IF FOUND_ROWS() > 0 THEN
                            CALL pembayaran.storeRincianTagihanPaket(VTAGIHAN, VPAKET_DETIL, VKARTU, NEW.TANGGAL, 1, 1);
                            CALL pembayaran.batalRincianTagihan(VTAGIHAN, VKARTU, 1);
                        END IF;
                    END IF;
                END;
            END IF;
        END IF;
    END IF;
END;
```

Pada contoh trigger di atas, trigger akan dijalankan setelah ada data yang di-insert ke tabel pendaftaran.

Terlihat, salah satu logikanya adalah buatTagihan pada database pembayaran. Sehingga, perilaku tambah data di tabel pendaftaran dapat dipelajari pada trigger ini. Dengan mempelajari fungsi-fungsi trigger ini, akan mudah memahami perilaku database. Secara keseluruhan, dikarenakan seluruh aturan-aturan aplikasi tercatat dalam fungsi (tidak di dalam aplikasi), maka dengan memahami fungsi-fungsi dalam database ini secara otomatis akan mempunyai kemampuan untuk menguasai proses-proses aplikasi SIMGOS.

Selain itu pula, hal ini menyederhanakan proses di framework Sencha atau yang berkaitan dengan tampilan pengguna. Sehingga aplikasi hanya berfokus pada proses-proses dasar crud (Create Read Update Delete) dan desain tampilan pengguna.

# PROSES

## 4. Proses Implementasi

**i** Bagian terberat adalah proses implementasi, karena disinilah aplikasi digunakan oleh pengguna yang mungkin langsung juga berhadapan dengan pelanggan (pasien atau keluarganya). Proses implementasi banyak pola dan caranya. Tentunya dengan faktor resiko masing-masing.

Yang perlu menjadi perhatian bahwa kesalahan kecil pada proses coding bisa berakibat besar. Kesalahan ketik lebih dari dan kurang dari bisa merubah logika program. Hal yang tidak dapat dipungkiri bahwa manusia adalah makhluk yang mempunyai kekurangan. Untuk mengatasi hal-hal ini tentunya bisa disiasati dengan proses-proses testing, proses update aplikasi / database dari kegiatan development ke lingkungan testing maupun ke lingkungan produksi, membuat daftar kegiatan / Langkah-langkah implementasi, dan seterusnya.

Perlu juga dibuatkan catatan-catatan kegiatan implementasi, masukan pengguna, catatan-catatan error dan penyelesaiannya. Hal ini akan sangat bermanfaat dalam pengembangan aplikasi selanjutnya, agar lebih terukur terstruktur dan tepat sasaran.

## 5. Proses Update

**i** Proses Update biasanya terbagi atas 2 bagian utama, yaitu : update aplikasi dan update database.

Git adalah salah satu tools dalam pengembangan aplikasi yang sangat populer saat ini. Sangat membantu team development dalam memelihara kode sumber program. Sangat disarankan ada admin yang mengontrol baik proses pembagian tugas maupun proses update pada lingkungan test dan produksi.

## 6. Pembagian Tugas

**i** Berdasarkan Struktur Aplikasi, secara penugasan teknis terbagi atas :

1. *Programmer Interface, sebagai sumber daya untuk melakukan pembuatan atau pengembangan tampilan layar sesuai dengan kebutuhan.*
2. *DBA / database administrator, sebagai sumber daya yang selalu berkoordinasi dengan programmer interface dalam penyediaan database. Dengan Batasan pekerjaan dari fisik server, sistem operasi, instalasi database, proses backup restore, pembuatan tabel, hingga fungsi-fungsi trigger. Mengingat semua aturan aplikasi ada pada fungsi trigger, sehingga perlu dilakukan penguatan Sumber Daya Manusia pada sisi ini, baik berupa peningkatan kemampuan maupun penambahan sdm dengan melakukan pergeseran tanggung jawab dari programmer interface, ataupun kepala programmer. Hal ini diperlukan karena besarnya tanggungjawab pada bagian ini.*

*Hal tersebut diatas adalah fokus terhadap penugasan teknis. Tentunya hal pendukung lainnya yang tak kalah penting adalah bagian implementator yang langsung berhadapan dengan pengguna, bagian testing yang memastikan aplikasi siap untuk dijalankan di produksi, bagian master data yang memastikan kesesuaian master data. Selain itu pula biasanya catatan-catatan kegiatan ada pada bagian ini.*

*Menjadi penekanan khusus untuk bagian DBA, karena ini merupakan otak dari aplikasi SIMGOS. Dengan penguatan di bagian ini, diharapkan meningkatkan kepercayaan diri dalam memegang kendali terhadap aplikasi SIMGOS ini.*

## SELANJUTNYA

### 7. Pekerjaan selanjutnya

**i** Adapun pekerjaan-pekerjaan yang dapat dilakukan selanjutnya adalah :

1. *Optimasi dan Penyesuaian, meliputi pekerjaan optimasi terhadap kecepatan proses dan tampilan pengguna agar pengguna semakin nyaman cepat dalam bekerja, dan sesuai dengan pola kerja di lingkungan Rumah Sakit Fatmawati. Disinilah pentingnya catatan-catatan implementasi, sebagai dasar dalam melakukan optimasi dan penyesuaian.*
2. *Pembuatan Aplikasi Pengguna, meliputi pekerjaan pembuatan aplikasi yang sifatnya belum ada. Sehingga pada akhirnya aplikasi ini menjadi lengkap seluruh aspek kebutuhan pengguna di Rumah Sakti Fatmawati.*
3. *Pelaporan atau Data Warehouse. Mengingat dari struktur databasenya yang ada dan tingginya transaksi di Rumah Sakit Fatmawati, diperlukan database pelaporan tersendiri. Transaksi dan Laporan merupakan hal yang saling Tarik menarik kebutuhan sumber daya server dan database, sehingga disarankan untuk membuat database tersendiri, dengan menarik data transaksi secara berkala, misalnya setiap 24 jam.*
4. *Dokumentasi, hal ini biasanya sering terabaikan dan bisa menjadi berat dilakukan. Fungsi terbesar dari bagian ini adalah adanya pengembangan berkesinambungan, sekalipun telah berganti sumber daya manusia.*