BAB I

Pendahululuan

1.1. Latar Belakang Masalah

Kereta Rel Listrik atau lebih sering dikenal dengan KRL adalah salah satu alat transportasi yang sangat populer di kalangan pengguna Kereta Api terutama di daerah Jabotabek dan sekitarnya. Dibandingkan dengan transportasi umum darat lainnya, selain hargannya yang ekonomis, Kereta Rel Listrik (KRL) juga merupakan salah satu kendaraan yang efisien bebas dari kemacetan lalulintas yang sering menjadi kendala pengendara umum lainya di Jabotabek.

Banyaknya peminat pengguna Kereta Rel Listrik (KRL) menjadikan frekuensi operasional menjadi lebih banyak, dan untuk mengatasi hal tersebut dibutuhkan sebuah alat bantu untuk dapat mengatasi penjadwalan dan perawatan Kereta Rel Listrik dengan membangun sebuah aplikasi jadwal perdinasan KA(T18) untuk memudahkan pekerjaan penjadwalan Kereta Rel Listrik (KRL).

1.2. Perumusan Masalah

Berdasarkan latar belakang diatas diperlukan suatu aplikasi yang dapat menyelesaikan permasalahan operasional. Rumusan masalah yang dihadapi adalah bagaimana membuat aplikasi Penjadwalan Kereta Rel Listrik (KRL).

1.3. Maksud dan Tujuan

Maksudnya adalah membuat aplikasi Penjadwalan Kereta Rel Listrik (KRL).

Tujuan dari pembuatan aplikasi Penjadwalan Kereta Rel Listrik (KRL) adalah:

1. Dapat menangani masalah penjadwalan dan perawatan Kereta Rel Listrik (KRL).
2. Memudahkan dalam melakukan pekerjaan penjadwalan dan perawatan Kereta Rel Listrik (KRL) yang efektif dan efisien dalam menyajikan informasi serta memberikan solusi dalam mengatasi penjadwalan Kereta Rel Listrik (KRL).

1.4. Batasan Masalah

Aplikasi yang dibuat hanya menangani penjadwalan dan perawatan Kereta Rel Listrik (KRL), mulai dari pencatatan jadwal keberangkatan KRL, pencatatan no seri rangkaian dari KRL tersebut, pencatatan stasiun persinggahan yang dilewati oleh KRL, dan pembuatan laporan dari jadwal keberangkatan Kereta Rel Listrik (KRL). Namun aplikasi ini tidak menangani login untuk pengguna, grafik penjadwalan, laporan harian masinis, dinasan kereta api, dan pemeliharaan Kereta Rel Listrik.

Data yang akan diolah adalah data Kereta Rel Listrik (KRL), data penjadwalan, data stasiun singgah dan data detail rangkaian Kereta Rel Listrik (KRL).

Software yang digunakan untuk membuat aplikasi ini adalah Windows XP untuk sistem operasinya, c++ untuk bahasa pemrogramannya dengan compiler yang digunakan adalah borland c++ builder, dan DBMS yang digunakan adalah *Interbase*. *Hardware* minimum yang diperlukan agar *software* dapat bekerja adalah ram 256 MB, prosesor pentium 4, harddisk 20GB.

Perangkat lunak digunakan oleh karyawan petugas tiket PT Kereta Api Indonesia, operator, dan kepala stasiun. Tipe jaringan yang digunakan adalah client server yang menghubungkan database yang berada di server dengan yang ada di *client* dengan menggunakan topologi star.

1.5. Metodologi Penelitian

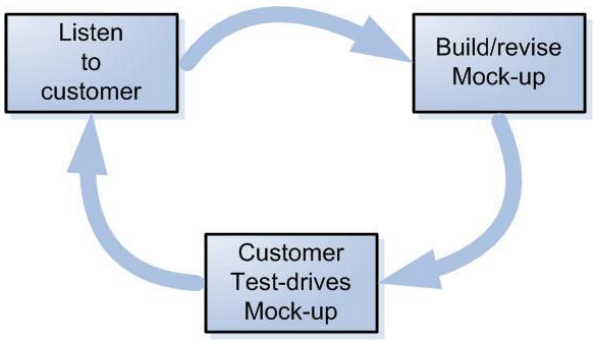
Metode yang dilakukan ada 2 yaitu:

a. Pengumpulan data

   Pengumpulan data yang dilakukan adalah dengan cara wawancara dengan *user* dan *developer* yang berada di PT.KAI.

b. Pembangunan aplikasi

   Metode yang digunakan adalah metode prototipe, untuk lebih jelasnya dapat dilihat pada gambar 1.1 dibawah ini.



Gambar 1.1 Model Prototipe

Proses pada model prototipe yang digambarkan pada gambar diatas , dapat dijelaskan sebagai berikut:

1. Listen to customer: *developer* dan *user* bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.
2. Build / revise mock-up : perancangan dilakukan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan prototipe.
3. Customer test-drives mock-up: *user* mengevaluasi prototipe yang dibuat dan digunakan untuk memperjelas kebutuhan software.

1.6. Sistematika Penulisan

BAB I Pendahuluan: bab ini membahas mulai dari latarbelakang masalah, identifikasi masalah, maksud dan tujuan, metodologi penelitian dan, batasan masalah.

BAB II Ruang Lingkup Perusahaan: bab ini memebahas mengenai sejarah, visi, misi, Struktur Organisasi dari Kereta Api.

BAB III Landasan teori: bab ini membahas mengenai prototype model,ERD ,DFD dan flowchat.

BAB IV Perancangan : bab ini membahas mengenai perancangan penjadwalan kereta rel listrik

BAB V Penutup: bab ini membahas mengenai kesimpulan

BAB II

Ruang Lingkup Perusahaan

1.1. Sejarah

Kehadiran kereta api di Indonesia ditandai dengan pencangkulan pertama pembangunan jalan KA didesa Kemijen Jum'at tanggal 17 Juni 1864 oleh Gubernur Jenderal Hindia Belanda, Mr. L.A.J Baron Sloet van den Beele. Pembangunan diprakarsai oleh "Naamlooze Venootschap Nederlandsch Indische Spoorweg Maatschappij" (NV. NISM) yang dipimpin oleh Ir. J.P de Bordes dari Kemijen menuju desa Tanggung (26 Km) dengan lebar sepur 1435 mm. Ruas jalan ini dibuka untuk angkutan umum pada Hari Sabtu, 10 Agustus 1867.

Keberhasilan swasta, NV. NISM membangun jalan KA antara Kemijen - Tanggung, yang kemudian pada tanggal 10 Februari 1870 dapat menghubungkan kota Semarang - Surakarta (110 Km), akhirnya mendorong minat investor untuk membangun jalan KA didaerah lainnya.

Setelah kemerdekaan Indonesia diproklamasikan pada tanggal 17 Agustus 1945, karyawan KA yang tergabung dalam "Angkatan Moeda Kereta Api" (AMKA) mengambil alih kekuasaan perkeretaapian dari pihak Jepang. Peristiwa bersejarah yang terjadi pada tanggal 28 September 1945, pembacaan pernyataan sikap oleh Ismangil dan sejumlah anggota AMKA lainnya, menegaskan bahwa

mulai tanggal 28 September 1945 kekuasaan perkeretaapian berada ditangan bangsa Indonesia. Orang Jepang tidak

diperkenankan lagi campur tangan dengan urusan perkeretaapian di Indonesia. Inilah yang melandasi ditetapkannya 28 September 1945 sebagai Hari Kereta Api di Indonesia, serta dibentuknya "Djawatan Kereta Api Republik Indonesia" (DKARI).

1.2. Visi

Terwujudnya Kereta Api sebagai Pilihan Utama Jasa Transportasi dengan fokus keselamatan dan pelayanan.

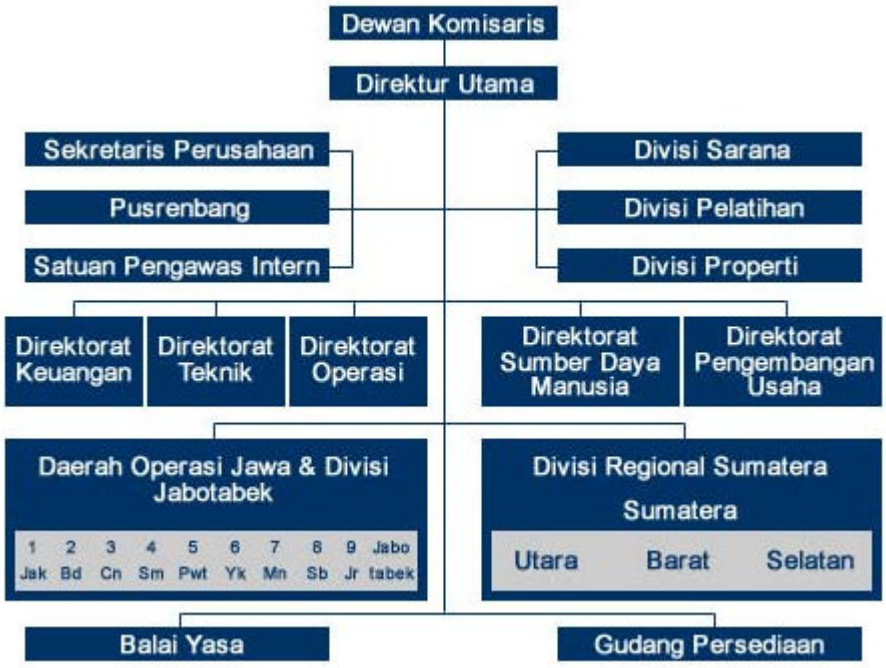
1.3. Misi

Menyelenggarakan jasa transportasi sesuai keinginan *stakeholder* dengan meningkatkan keselamatan dan pelayanan serta penyelenggaraan yang semakin efisien.

1.4. Struktur Organisasi

Struktur organisasi merupakan susunan seluruh organisasi dari PT Kereta Api Indonesia, mulai dari yang tertinggi yaitu dewan komisaris sampai ke daerah operasi dan divisi regional. Organisasi tertinggi dalam struktur organisasi PT.KAI

adalah dewan komisaris yang menjabat sebagai dewan tertinggi di atas direktur utama yang memiliki kemampuan dan keputusan menganai arah susunan organisasi PT.KAI sesuai dengan keputusan mentri transportasi indonesia, semua susunan organisasi di dalam PT.KAI mulai dari dewan komisaris sampai ke daerah operasi dan divisi regional memiliki fungsi dan perannnya masing-masing untuk menuju satu tujuan yang sama yaitu memiliki visi yang sama memberikan kenyamanan bertransportasi berkereta api kepada masyarakat. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini:



Gambar 2.1 – Struktur Organisasi

BAB III

Landasan Teori

3.1. Prototype Model

Kadang-kadang klien hanya memberikan beberapa kebutuhan umum software tanpa detil input, proses atau detil output. Di lain waktu mungkin dimana tim pembangun tidak yakin terhadap efisiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan form user interface. Ketika situasi seperti ini terjadi model prototype sangat membantu proses pembangunan software. Meskipun prototype dapat digunakan sebagai model proses berdiri sendiri (standalone), namun lebih sering digunakan sebagai teknik yang diimplementasikan bersama dengan model-model yang lain.

Prototype membantu pengembang dan pengguna untuk memiliki pemahaman yang lebih baik tentang apa yang akan dibangun ketika kebutuhan yang diinginkan tidak diuraikan secara jelas. Prototype diawali dengan komunikasi. Pengembang dan pengguna bertemu dan mendefinisikan sasaran-sasaran menyeluruh dari perangkat lunak yang akan dibangun, mengidentifikasi kebutuhan apa saja yang diinginkan. Iterasi prototype direncanakan secara cepat, demikian juga pemodelan dalam bentuk rancangan segera dibuat. Perancangan yang cepat berfokus pada penggambaran aspek-aspek perangkat lunak yang akan dilihat oleh pengguna, seperti tampilan antarmuka pengguna dengan sistem, atau format

tampilan output. Rancangan yang cepat ini akan membawa ke arah pembuatan program (konstruksi) dari prototype.

Prototype diserahkan dan dievaluasi oleh pengguna. Umpan balik dari pengguna digunakan untuk memperbaiki kriteria kebutuhan perangkat lunak. Hal ini dilakukan berulang-ulang dimana prototype disesuaikan untuk memenuhi kebutuhan pengguna, sementara pada saat yang samapengembang memiliki pemahaman yang lebih baik mengenai apa yang diinginkan pengguna untuk dipenuhi. Secara ideal, prototype adalah suatu mekanisme Untuk mengidentifikasi kebutuhan dari perangkat lunak yang akan dihasilkan. Pada saat prototype ini dikembangkan, pengembang berusaha menggunakan program atau took yang ada, seperti report generator, windows manager, yang memungkinkan prototype dibuat secara cepat. Prototype berlaku sebagai sistem pengenal, bukan sebagai system yang benarbenar dihasilkan untuk dioperasionalkan. Adalah benar bahwa banyak pengguna dan pengembang yang menyukai model prototype. Pengguna dapat merasakan sistem yang akan diwujudkan, dan pengembangan dapat membangun sesuatu dengan segera.

Proses pada model prototype yang digambarkan pada gambar , bisa dijelaskan sebagai berikut:

1. Listen to customer: developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detil kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.

2. Build / revise mock-up : perancangan dilakukan cepat dan rancangan mewakili semua aspek software yang diketahui, dan rancangan ini menjadi dasar pembuatan prototype.
3. Customer test-drives mock-up: klien mengevaluasi prototype yang dibuat dan digunakan untuk memperjelas kebutuhan software.

Perulangan ketiga proses ini terus berlangsung hingga semusa kebutuhan terpenuhi. Prototype-prototype dibuat untuk memuaskan kebutuhan klien dan untuk memahami kebutuhan klien lebih baik. Prototype yang dibuat dapat dimanfaatkan kembali untuk membangun software lebih cepat, namun tidak semua prototype bisa dimanfaatkan. Sekalipun prototype memudahkan komunikasi antar developer dan klien, membuat klien mendapat gambaran awal dari prototype , membantu mendapatkan kebutuhan detil lebih baik namun demikian prototype juga menimbulkan masalah:

1. Dalam membuat prototype banyak hal yang diabaikan seperti efisiensi, kualitas, kemudahan dipelihara/dikembangkan, dan kecocokan dengan lingkungan yang sebenarnya. Jika klien merasa cocok dengan prototype yang disajikan dan berkeras terhadap produk tersebut, maka developer harus kerja keras untuk mewujudkan produk tersebut menjadi lebih baik, sesuai kualitas yang seharusnya.
2. Developer biasanya melakukan kompromi dalam beberapa hal karena harus membuat prototype dalam waktu singkat. Mungkin sistem operasi yang tidak sesuai, bahasa pemrograman yang berbeda, atau algoritma yang lebih sederhana. Agar model ini bisa berjalan dengan baik, perlu disepakati

bersama oleh klien dan developer bahwa prototype yang dibangun merupakan alat untuk mendefinisikan kebutuhan software.

3. Pengguna melihat bahwa apa yang muncul dan dilihat dari prototype adalah perangkat lunak yangk akan dioperasionalkan. Pengguna tidak menyadari bahwa prototype tersebut belum dibangun dengan memperhatikan kualitas perangkat lunak beserta maintainabilitasnya. Jika disampaikan bahwa produk yang akan dioperasionalkan harus dibangun ulang sehingga produk memiliki kualitas tinggi dan dapat dipelihara dengan baik, maka pengguna mengeluh dan bahkan meminta beberapa perbaikan untuk turut diterapkan dalam sistem yang akan dioperasionalkan.

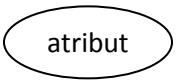
Meskipun permasalahan ada, prototype dapat menjadi model yang efektif untuk rekayasa perangkat lunak. Kuncinya adalah aturan permainan harus dijelaskan di awal proyek, bahwa pengembang dan pengguna harus memiliki kesepahaman bahwa prototype dibuat sebagai sarana untuk mendefinisikan kebutuhan. Sedangkan perangkat lunak sesungguhnya dibangun dengan berdasarkan kualitas.

3.2. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah model konseptual yang mendeskripsikan hubungan antara penympanan (dalam DFD). ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan ERD model dapat diuji dengan mengabaikan proses yang dilakukan.

ERD pertama kali dideskripsikan oleh Peter Chen yang dibuat sebagai bagian dari perangkat lunak CASE. Notasi yang digunakan dalam ERD dapat dilihat pada tabel dibawah ini:

Tabel 3.1 Notasi ERD

Notasi	Keterangan
	Entitas adalah suatu objek yang dapat dibedakan dari objek lainnya.
	Relasi menunjukan adanya hubungan diantara sejumlah entitas yang berbeda.
	Atribut merupakan cirri atau karakteristik yang dimiliki oelh sebuah entitas.atribut yang berfungsi sebagai key diberi garis bawah.
	Garis, sebagai penghubung antara relasi dengan entitas, relasi (apabila derajat relasi N ke N) dan entitas dengan atribut.

Di dalam atribut ada yang berfungsi sebagai key. Apa itu key? Key adalah suatu atribut yang sifatnya unik. Key dapat dibangun dari satu atribut atau gabungan dari beberapa atribut. Key terbagi menjadi beberapa jenis, diantarnya:

1. Super key, seluruh atribut dalam suatu entitas dijadikan key.

2. Primary key, 1 atribut dalam suatu entitas dijadikan key.
3. Candidat key, beberapa atribut dijadikan key.
4. Foreign key, suatu atribut yang berasal dari tabel yang berelasi.

Di dalam ERD, relasi, dapat terdiri dari sejumlah entitas yang disebut dengan derajat relasi. Derajat relase maksimum disebut dengan kardinalitas sedangkan derajat minimum disebut dengan modalitas. Jadi, kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa:

1. Satu ke satu (one to one / 1-1)

   Setiap entitas pada himpunan entitas A dapat berelasi dengan paling banyak satu entitas pada himpunan entitas B, demikian juga sebaliknya.

2. Satu ke banyak (one to many / 1-N)

   Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan B, tetapi tidak sebaliknya.

3. Banyak ke banyak (many to many / N-N)

   Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya. Relasi akan dipetakan menjadi tabel beserta atributnya.

3.3 Data Flow Diagram (DFD)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data keluar dari system, dimana data disimpan, proses apa yang menghasilkan data dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD sering digukan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau simana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama DFD yaitu:

1. Kebebasan dalam menjalankan implementasi tektik system.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam system dan subsistem.
3. Mengkomunikasikan pengetahuan system yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisa system yang diajukan untuk mementukan apakah data-data dan proses yang diperlukan sudah ditetapkan.
5. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik keterkaitan satu sama lain dalam system dan subsistem.
6. Membedakan system dari lingkungannya dengan menempatkna batas-batasnya.

7. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.
8. memungkinkan penganalisis menggambarkan setiap komponen yang digunakan dalam diagram.

DFD terdiri dari context diagram dan diagram rinci (DFD Levelled). Context diagram berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan system. DFD levelled menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan system dari sudut pandang fungsi.

Dalam DFD levelled akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikaso proses yang jelas. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran data yang masuk dan keluar pada level x+1 yang mendefinisikan proses pada level x tersebut.

Simbol-simbol yang digunakan dalam DFD dapa dilihat pada tabel di bawah ini:

Tabel 3.2 Simbol DFD

Yourdon / De Marco	Keterangan
Etitas Eksternal	Entitas eksternal, dapat berupa orang / unit yang terkait yang berinteraksi dengan system tetapi diluar sistem.
Proses	Orang, unit yang mempergunakan data. Komponen fisik yang tidak dapat diidentifikasikan.
Aliran data	Aliran data dengan arah khusus dari sumber ke tujuan.
Data store	Penyimpanan data.

Dalam penggambaran DFD, ada beberapa peraturan yang harus diperhatikan sehingga dalam penggambarannya tidak terjadi kesalahan, aturan tersebut yaitu:

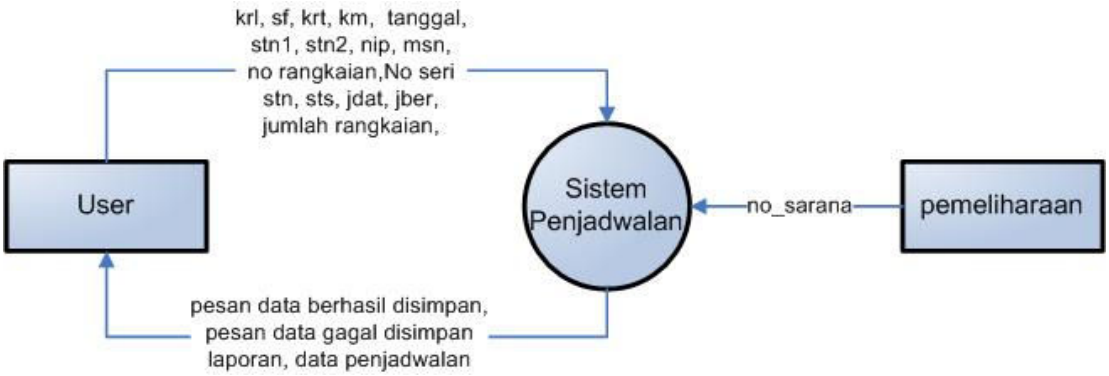
1. Antar entitas tidak diijinkan terjadi hubungan atau relasi.
2. Tidak boleh ada aliran data antara entitas eksternal dengan data store.
3. Untuk alasan kerapian, entitas eksternal atau data store boleh digambar beberapa kali dengan tanda khusus, misalnya diberi nomor atau garis miring.
4. Datu aliran data boleh mengalirkan beberapa paket data.
5. Bentuk anak panah aliran data boleh bervariasi.

6. Semua objek harus mempunyai nama.
7. Aliran data selalu diawali atau diakhiri dengan proses.
8. Semua aliran data harus mempunya tanda arah.
9. Jumlah proses tidak lebih dari sembilan proses dalam sistem, jika melebihi maka sebaiknya dikelompokan menjadi beberapa proses yang bekerja bersama-sama di dalam suatu subsistem.

BAB IV

Perancangan

4.1. Diagram Konteks



Gambar 4.1 – Diagram Konteks

4.2. DFD

4.2.1. DFD Level 1

Terdapat 3 proses yaitu info singgah, penjadwalan, dan info rangkaian.

1: Info singgah

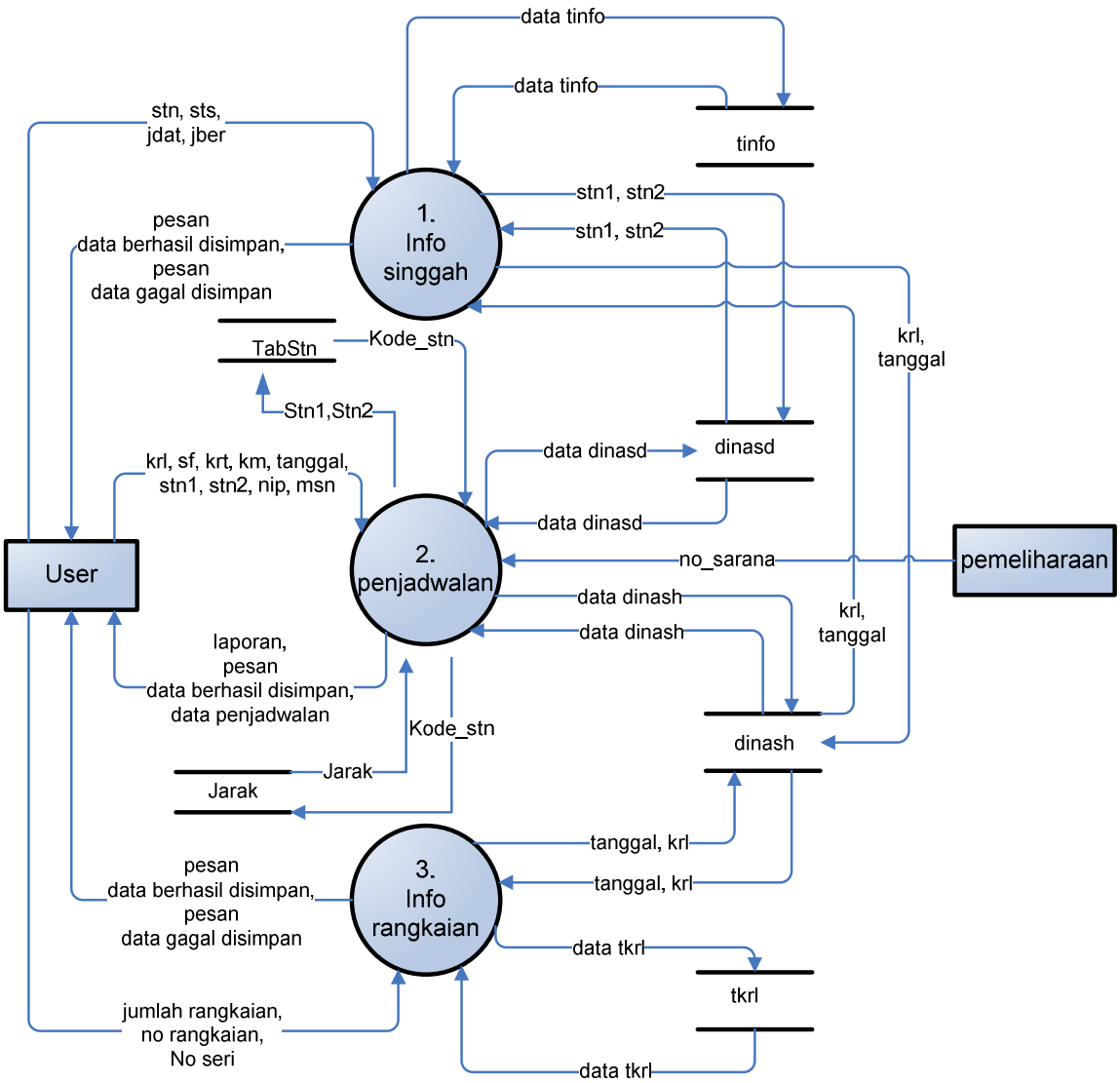
Digunakan untuk mencatat stasiun mana saja yang menjadi persinggahan oleh KRL. data disimpan pada tabel tinfo.

2: Penjadwalan

Digunakan untuk mencatat data KRL dan data penjadwalan. data disimpan pada tabel dinasd dan dinash.

3: Info rangkaian

Digunakan untuk mencatat rangkaian dari suatu KRL berdasarkan dari SF. Data disimpan pada tabel tkrl.



Gambar 4.2 – DFD Level 1

4.2.2. DFD Level 2 untuk proses 2

Terdapat 6 proses yaitu buat baru, isi jadwal, buka jadwal, cetak laporan, ganti status, dan hapus jadwal.

2.1: Buat baru

Proses ini mingizinkan pengguna apabila ingin membuat jadwal baru. Proses ini akan menghapus jadwal yang ada pada tanggal sekarang..

2.2: Isi jadwal

Pengisian jadwal hanya mengizinkan user untuk mengisi data, baik data krl maupun data keberangkatan. Data krl akan disimpan pada tabel dinash, sedangkan data keberangkatan akan disimpan pada tabel dinasd.

2.3: Buka jadwal

Digunakan untuk membuka penjadwalan yang sebelumnya telah disimpan didalam database dengan hanya mengisikan tanggal.

2.4: Cetak laporan

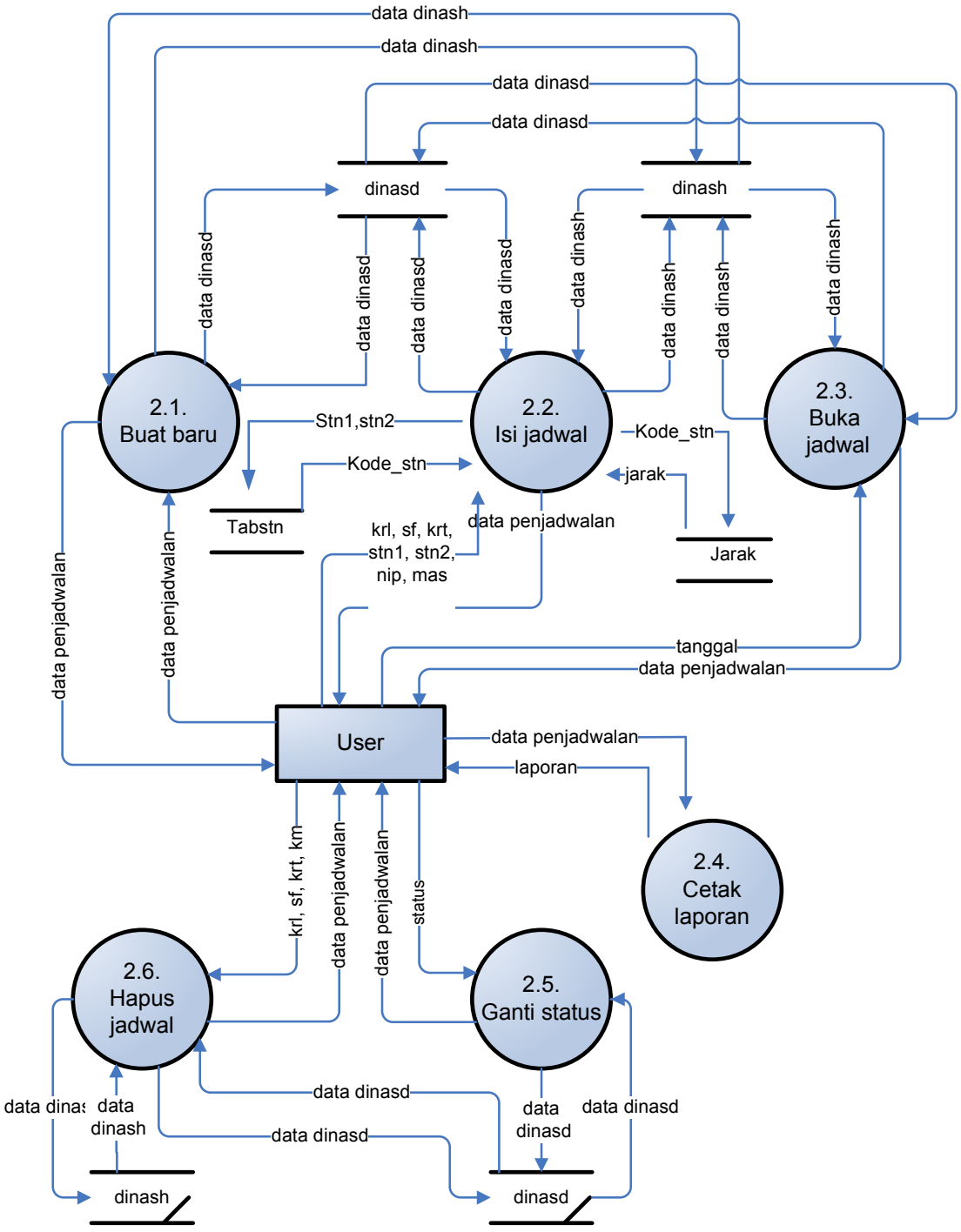
Digunakan untuk mencetak hasil dari pencatatan penjadwalan dalam bentuk laporan.

2.5: Hapus jadwal

Digunakan untuk menghapus data KRL yang tidak digunakan atau apabila adanya kesalahan pada saat pengisian data.

2.6: Ganti status

Untuk mengganti status KRL apakah dia beroperasi (status=0), batal beroperasi (status=1, atau dalam masa perawatan (status=2.



Gambar 4.3 DFD Level 2 Proses 2

4.3. Kamus Data

Kamus data merupakan deskripsi dari setiap elemen data yang terdapat dalam program. Berikut ini data pengolahan penjadwalan kereta api rel listrik

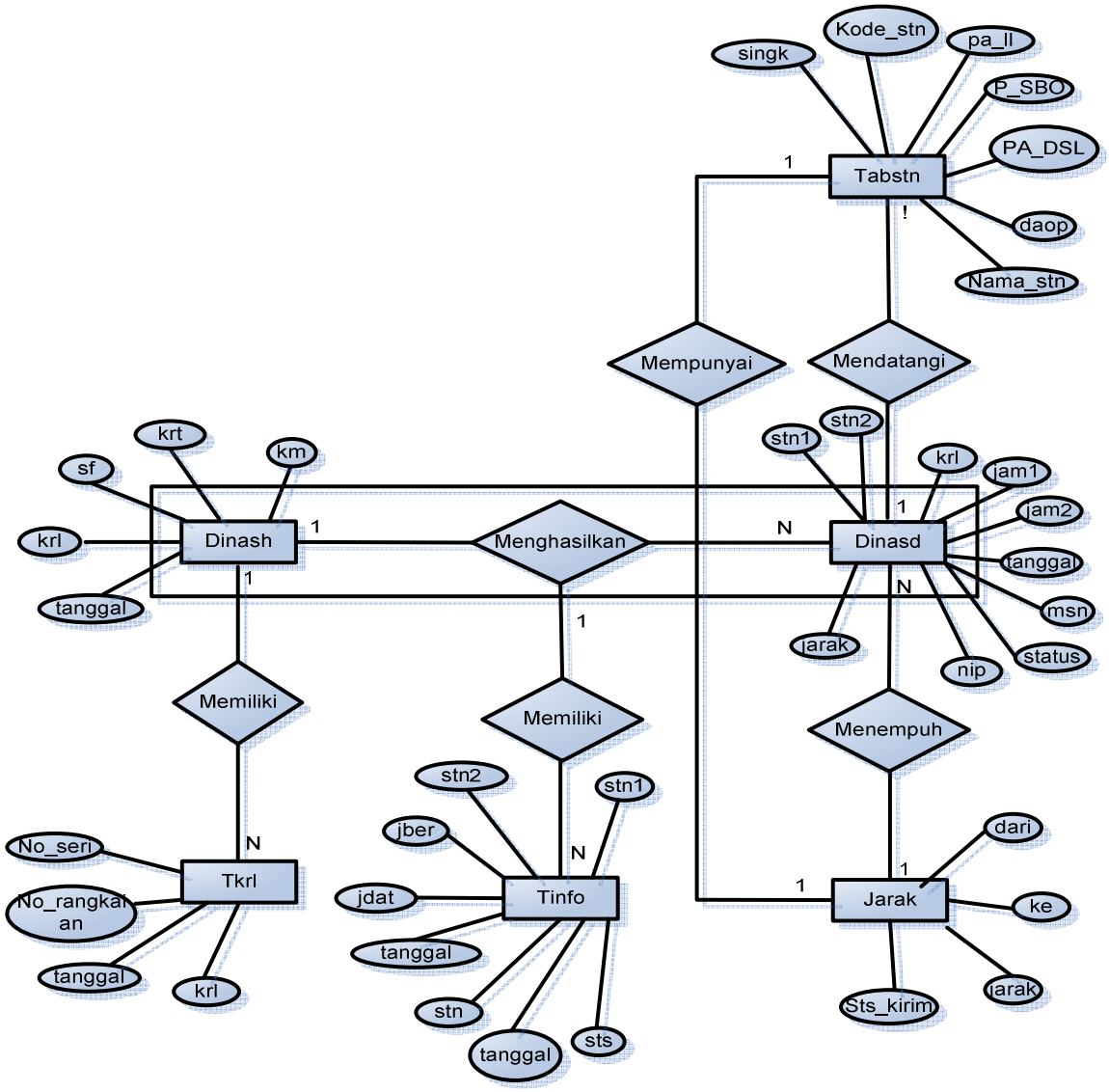
Tabel 4.1 Kamus Data

Nama : Data Dinash
Data Dinash = *berisi satu set rangkaian data krl* krl+sf+krt+tanggal+KM+Krt Krl = [A..Z|a..z|0..9|_|'|-] Sf = [A..Z|a..z|0..9|_|'|-] Tanggal=[0..9] KM=[0..9] Krt = [A..Z|a..z|0..9|_|'|-]
Nama : Data dinasd
Data Dinasd=*Berisi tentang data perjalanan KRL* Stn1+stn2+nip+msn+status+KRL+jam1+jam2+Tanggal+jarak Stn1=[A..Z|a..z|0..9|_|'|-] Stn2=[A..Z|a..z|0..9|_|'|-] Nip=[A..Z|a..z|0..9|_|'|-] Msn=[A..Z|a..z|0..9|_|'|-] Krl = [A..Z|a..z|0..9|_|'|-] jam1=[0..9] jam2=[0..9] Tanggal=[0..9] jarak=[0..9] Status=[A..Z|a..z]
Nama : Data Tinfo
Data Tinfo = *Berisi tentang informasi perjalan KRL* Stn+sts+jdat+jber+Tanggal+KRL++stn1+stn2 Stn = [A..Z|a..z|0..9|_|'|-]

Sts = [A..Z|a..z] Tanggal=[0..9] Krl = [A..Z|a..z|0..9|_|'|-] Jdat = [A..Z|a..z] Jber = [A..Z|a..z] Stn 1= [A..Z|a..z|0..9|_|'|-] Stn 2= [A..Z|a..z|0..9|_|'|-]
Nama : Data TKRL
Data rangkaian KRL =*berisi tentang informasi rangkaian data KRL* No_rangkaian+no_seri+Tanggal+KRL Tanggal=[0..9] Krl = [A..Z|a..z|0..9|_|'|-] No_rangkaian=[A..Z|a..z|0..9|_|'|-] No_seri=[A..Z|a..z|0..9|_|'|-]
Nama : Data TABSTN
Data TABSTN = *berisi data tentang stasiun* Nama_stn+Singk+Kode_Stn+Km+Daop Nama_stn= [A..Z|a..z|0..9|_|'|-] Singk = [A..Z|a..z|0..9|_|'|-] Kode_stn=[A..Z|a..z|0..9|_|'|-] KM=[0..9] Daop= [A..Z|a..z|0..9|_|'|-]
Nama : Data Jarak
Data Jarak = *berisi data tentang jarak antar stasiun* Dari+Ke+Jarak Dari=[0..9] Ke=[0..9] Jarak=[0..9]

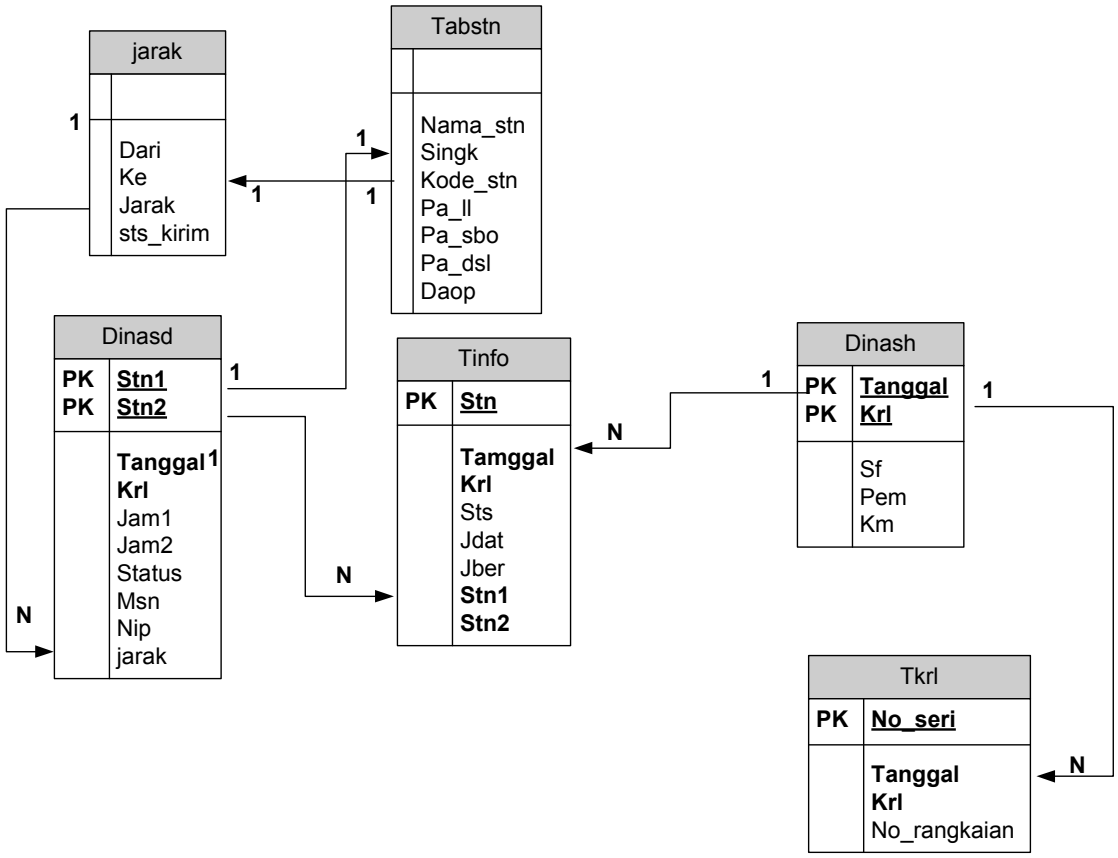
4.4. ERD

Gambar 4.4 berikut ini merupakan refrensi ERD,di mana akan terdapat beberapa tabel yang saling berelasi di antaranya tabel dinash yang berelasi terhadap tabel tkrl,tabel dinasd dan tabel dinasd berelasi juga terhadap tabel tinfo.



Gambar 4.4 ERD

4.5. Skema Relasi



Gambar 4.5 Skema Relasi

4.6. Struktur Tabel

4.6.1. Tabel Dinasd

Merupan tabel yang digunakan untuk menyimpan data-data mulai dari stasiun awal sampai satasiun tujuan beserta jam keberangkatannya.

Tabel 4.2 tabel dinasd

No	Nama field	Tipe	Panjang	Keterangan
1	Tanggal	Char	10	Digunakan untuk memisahkan jadwal.
2	Krl	Varchar	10	No krl dari suatu KRL
3	stn1	Varchar	4	Stasiun awal.
4	stn2	Varchar	4	Stasiun tujuan.
5	jam1	Char	5	Jam dari stasiun awal mulai berangkat.
6	jam2	Char	5	Jam dari stasiun tujuan tiba.
7	Status	Varchar	10	Status dari no krl. Angka 0 untuk jalan, 1 untuk batal, 2 untuk perawatan.
8	msn	Varchar	40	Nama masinis
9	Nip	Varchar	10	Nip dari masinis
10	Jarak	Integer	-	Jarak yang ditempuh dari stasiun awal ke stasiun tujuan.

4.6.2. Tabel Dinash

Merupakan tabel yang digunakan untuk menyimpan data-data KRL yang digunakan oleh penjadwalan.

Tabel 4.3 Tabel Dinash

No	Nama field	Tipe	Panjang	Keterangan
1	Tanggal	Char	10	Digunakan untuk memisahkan jadwal.
2	Krl	Varchar	10	No krl dari suatu KRL
3	Sf	Varchar	10	Untuk rangkaian KRL
4	Krt	Varchar	10	Untuk jumlah rangkaian KRL.
5	Km	Integer	-	Kilometer dari krl dihitun dari jumlah jarak dari keberangkatan dari suatu krl

4.6.3. Tabel Tinfo

Merupakan tabel yang digunakan untuk menyimpan data-data stasiun singgah.

Tabel 4.4 Tabel Tinfo

No	Nama field	Tipe	Panjang	Keterangan
1	Tanggal	Char	10	Digunakan untuk memisahkan jadwal.
2	Krl	Varchar	10	No krl dari suatu KRL
3	Stn	Varchar	4	Stasiun singgah
4	Sts	Varchar	10	Status dari persinggahan (fungsinya sama dengan status pada
5	Jdat	Char	5	Jam dari stasiun singgah mulai berangkat.
6	Jber	Char	5	Jam dari stasiun singgah tiba.
7	stn1	Varchar	4	Stasiun awal
8	stn2	Varchar	4	Stasiun tujuan

4.6.4 Tabel Tkrl

Merupakan tabel yang digunakan untuk menyimpan dan atau melihat informasi rangkaian dari suatu KRL yang digunakan untuk penjadwalan.

Tabel 4.5 Tabel Tkrl

No	Nama field	Tipe	Panjang	Keterangan
1	Tanggal	Char	10	Digunakan untuk memisahkan jadwal.
2	Krl	Varchar	10	No krl dari suatu KRL
3	no rangkaian	Varchar	5	No rangkaian dari KRL
4	no seri	Varchar	5	No seri rangkaian KRL

4.6.5 Tabel Tabstn

Merupakan tabel yang digunakan untuk mengetahui informasi tentang stasiun yang akan di lalui oleh KRL.

Tabel 4.6 Tabel Tabstn

No	Nama field	Tipe	Panjang	Keterangan
1	Nama_stn	Varchar	10	Untuk memberi nama stasiun

2	Singk	Varchar	10	Di gunakan untuk memberi singkatan nama stasiun
3	Kode_stn	Integer	-	Di gunakan untuk memberi kode pada stasiun
4	Pa_ll	Integer	-	Memberi nomer Pa_ll
5	Pa_sbo	Integer	-	Memberi no Pa_sbo
6	Pa_dsl	Integer	-	Memberi pa dsl
7	Daop	Integer	-	Di gunakan untuk memberi nama daerah operasional

4.6.6 Tabel Jarak

Merupakan tabel yang digunakan untuk mengetahui informasi tentang jarak antar stasiun yang akan di lalui oleh KRL.

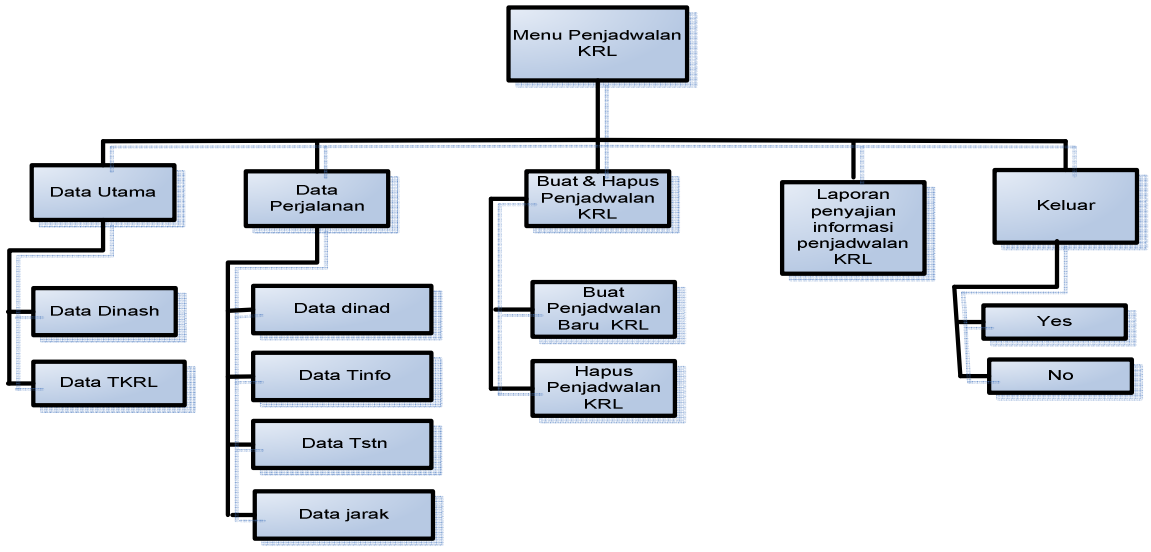
Tabel 4.7 Tabel jarak

No	Nama field	Tipe	Panjang	Keterangan
1	Dari	Integer	-	Digunakan untuk mengetahui kode stasiun awal
2	Ke	Integer	-	Digunakan untuk mengetahui

				kode stasiun awal tujuan
3	Jarak	Integer	-	Di gunakan untuk mengetahui jarak dari field dari dan field ke
4	Sts_kirim	Integer	-	Mengetahui status sudah terkirim atau belum

4.7 Perancangan Struktur Menu

Perancangan menu adalah perancangan antarmuka pilihan perintah pada program aplikasi untuk mengoperasikan dan memudahkan pemakai dalam menjalankan program. Struktur menu aplikasi dapat di lihat pada gambar berikut ini :

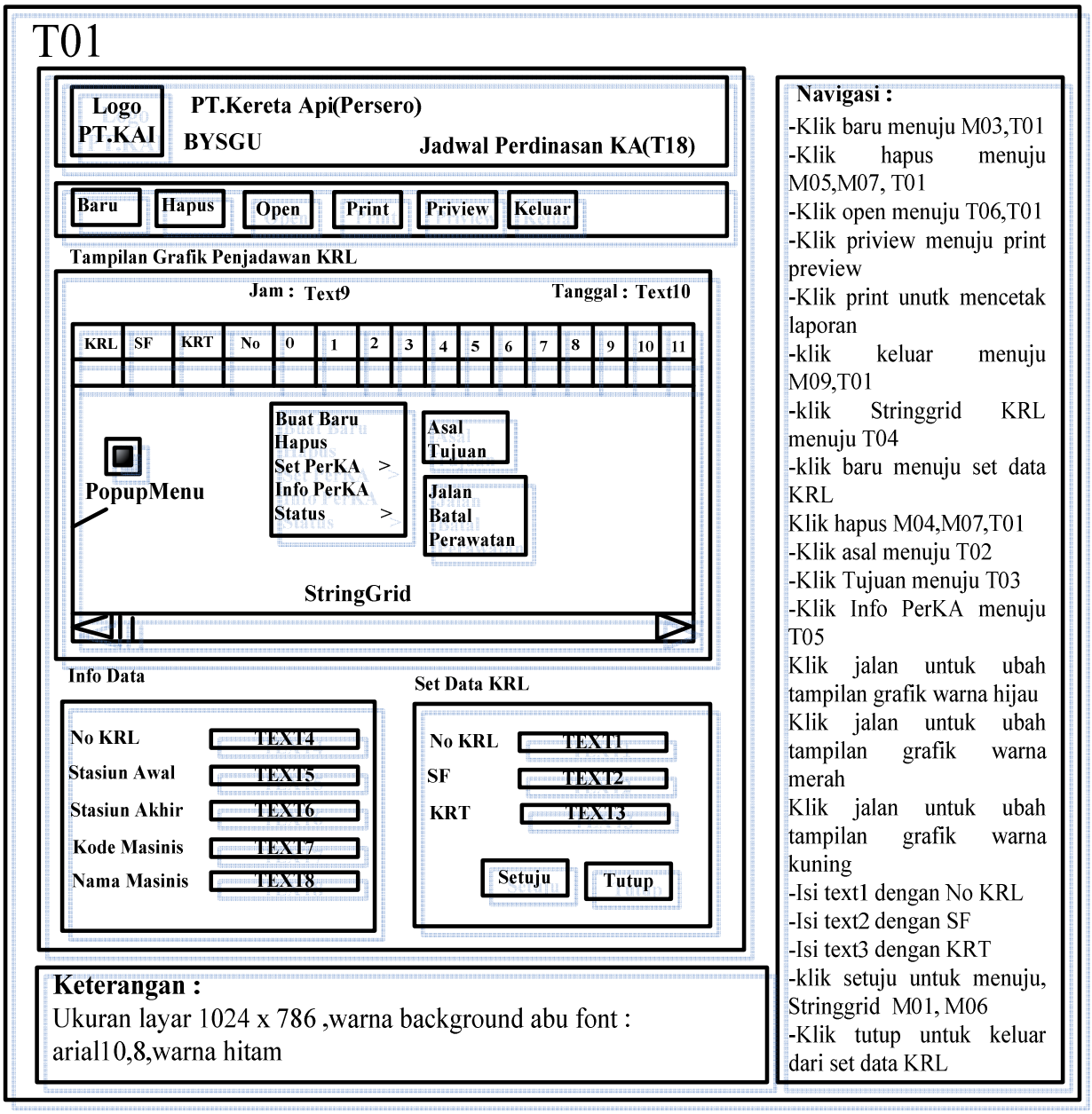


Gambar 4.6 Struktur Menu

4.8 Perancangan Antar Muka

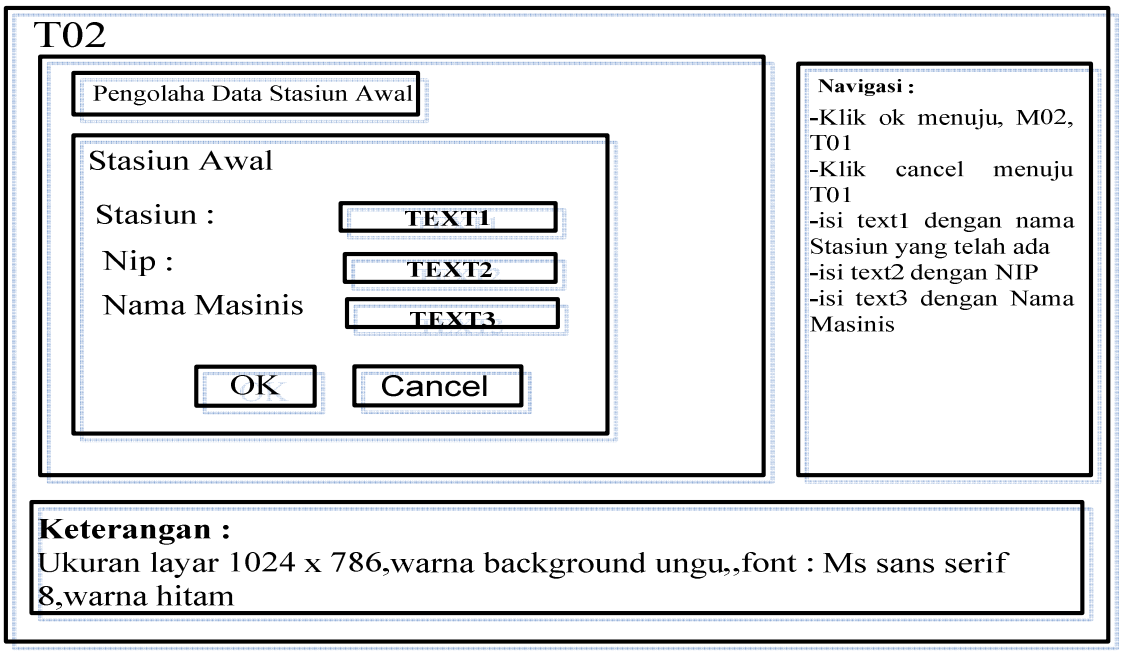
Perancangan antar muka terdiri dari perancangan struktur menu,perancangan tampilan input, perancangan tampilan output serta perancangan tampilan pesan.

4.8.1 Perancangan Form Menu Utama Pengolahan Input Data KRL



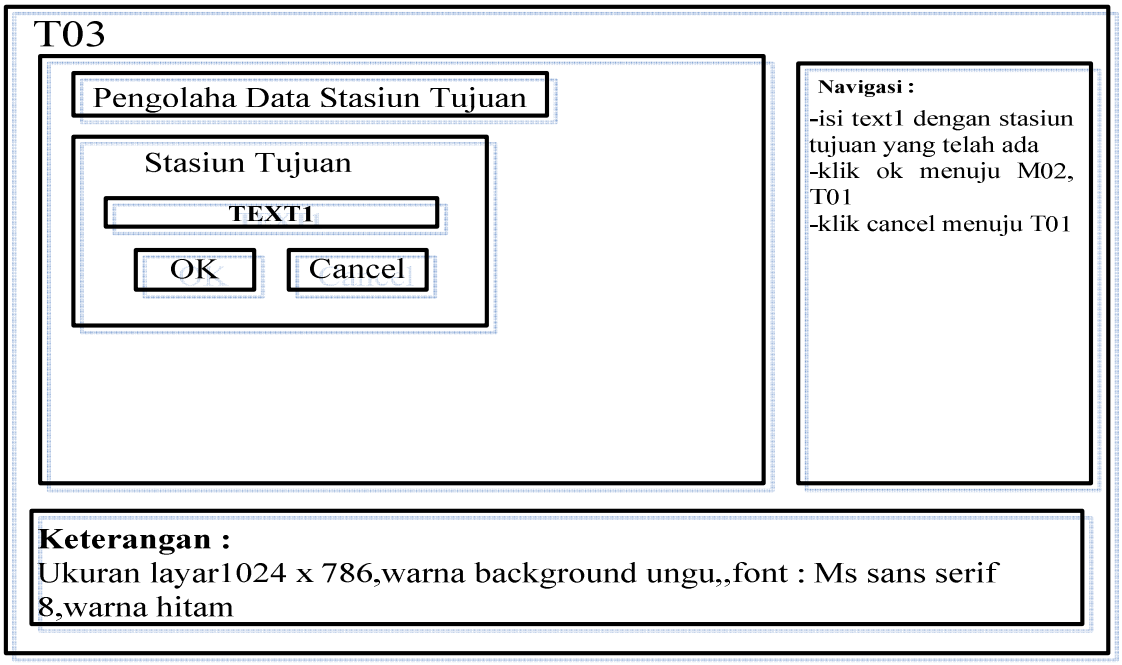
Gambar 4.7 Perancangan antamuka Form Menu Utama Pengolahan Data KRL

4.8.2 Perancangan Form Pengolahan Input Data Stasiun Asal



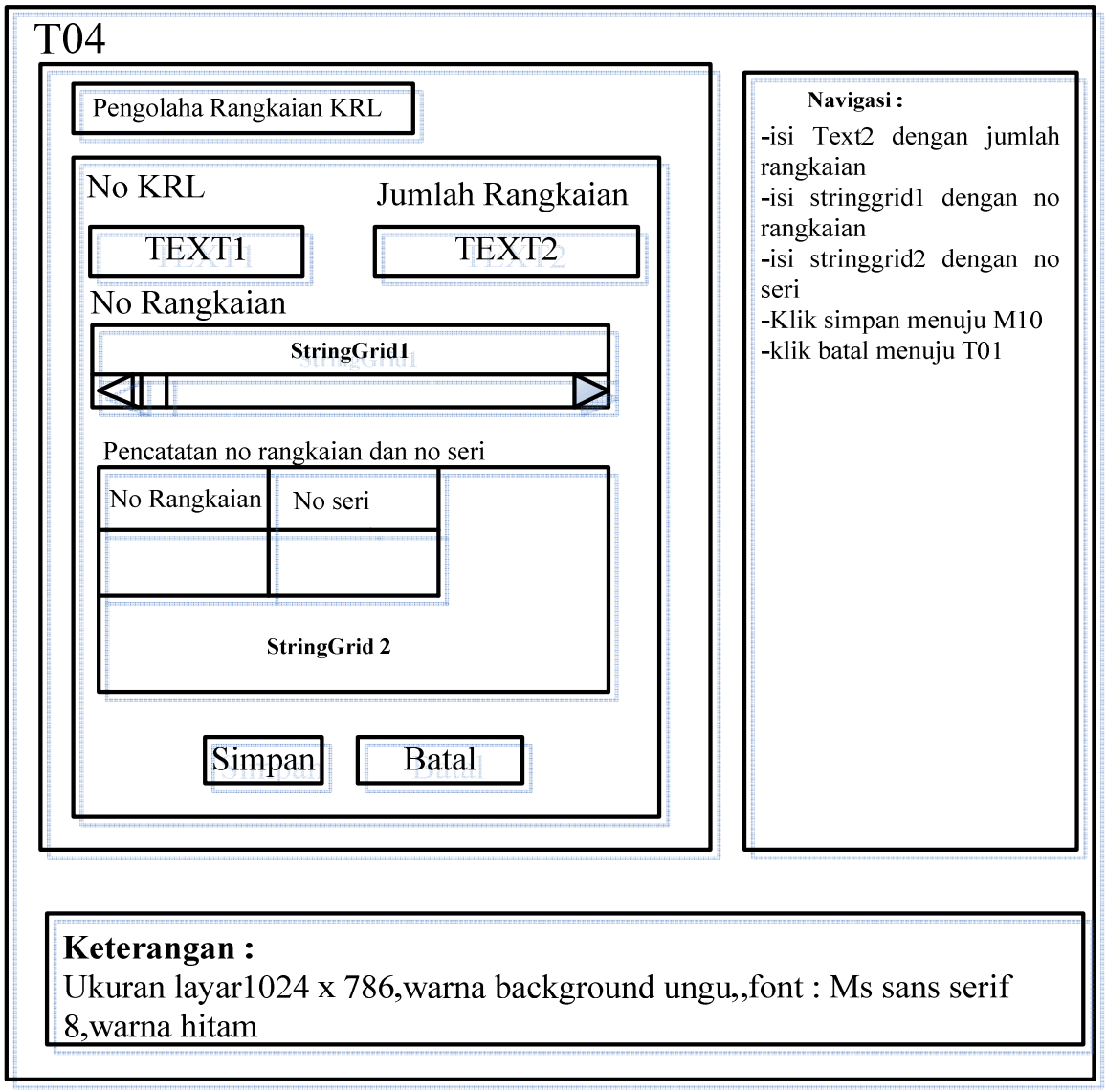
Gambar 4.8 Perancangan antamuka Form Pengolahan Data Stasiun Asal

4.8.3 Perancangan Form Pengolahan Input Data Stasiun Tujuan



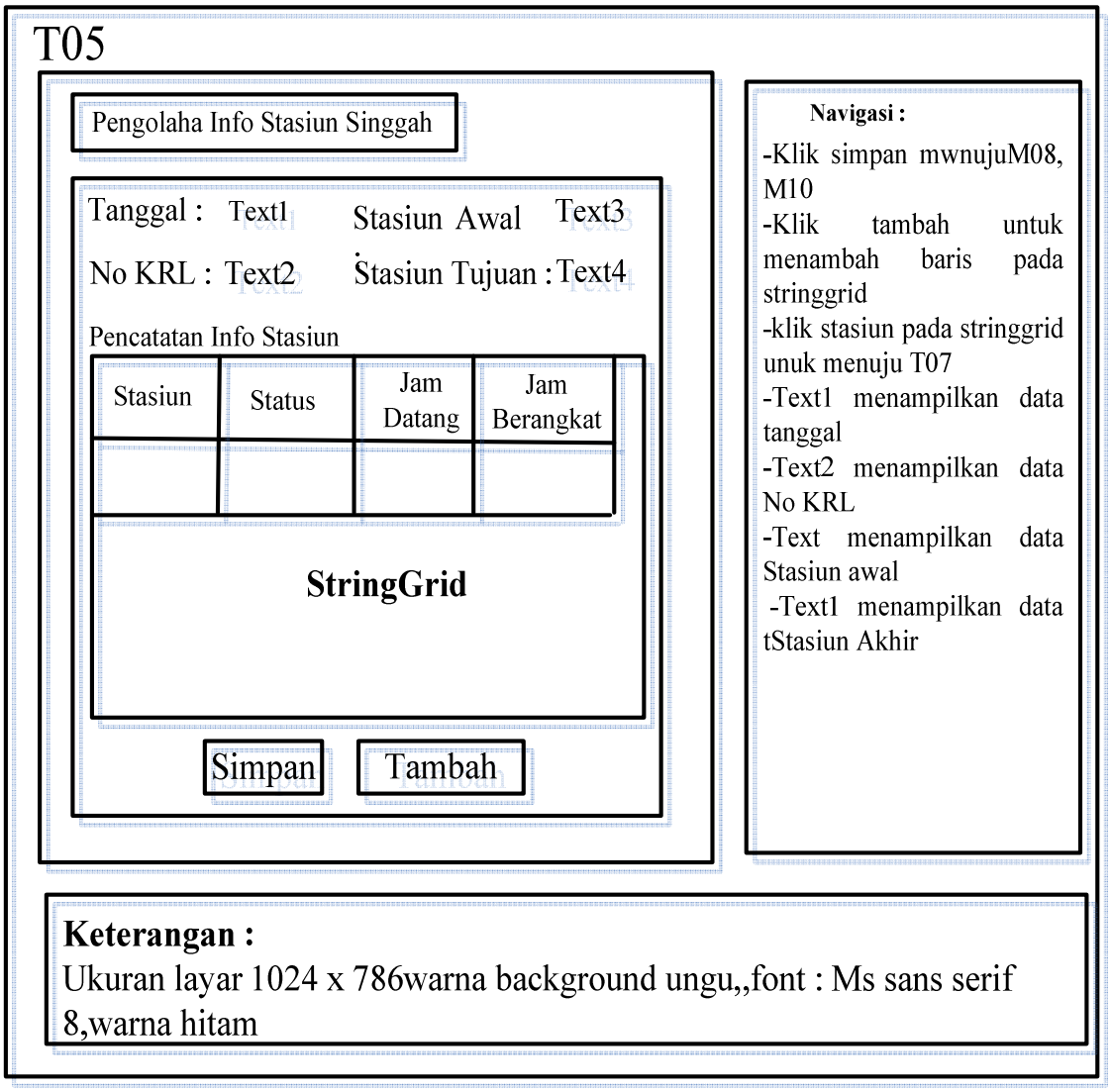
Gambar 4.9 Perancangan antamuka Form Pengolahan Data Stasiun Tujuan

4.8.4 Perancangan Form Pengolahan Rangkaian KRL



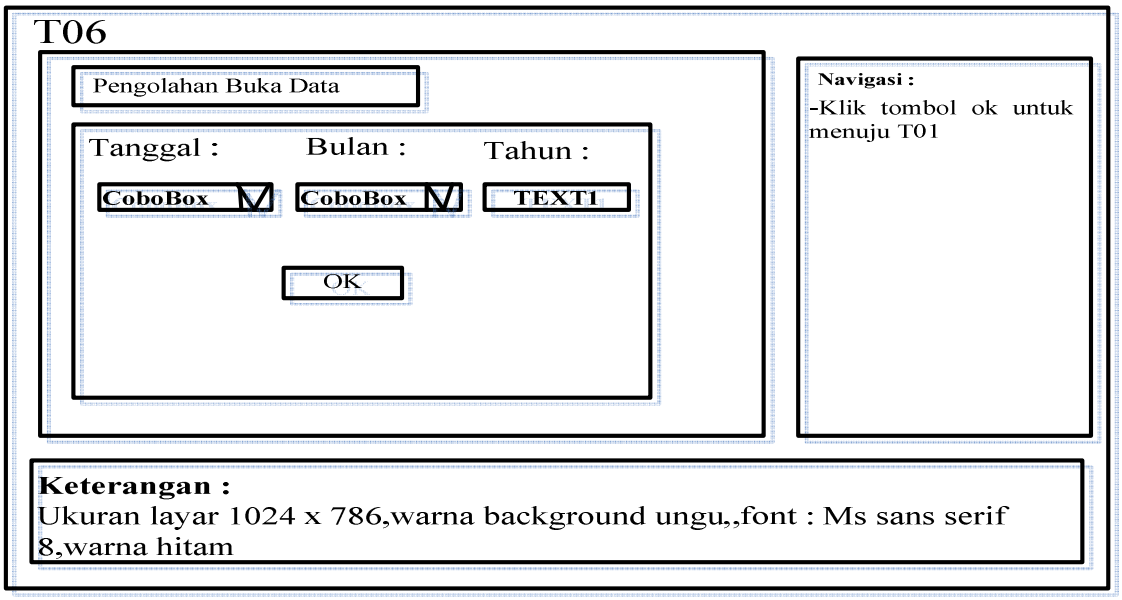
Gambar 4.10 Perancangan antamuka Form Pengolahan Rangkaian KRL

4.8.5 Perancangan Form Pengolahan Informasi Stasiun Singgah



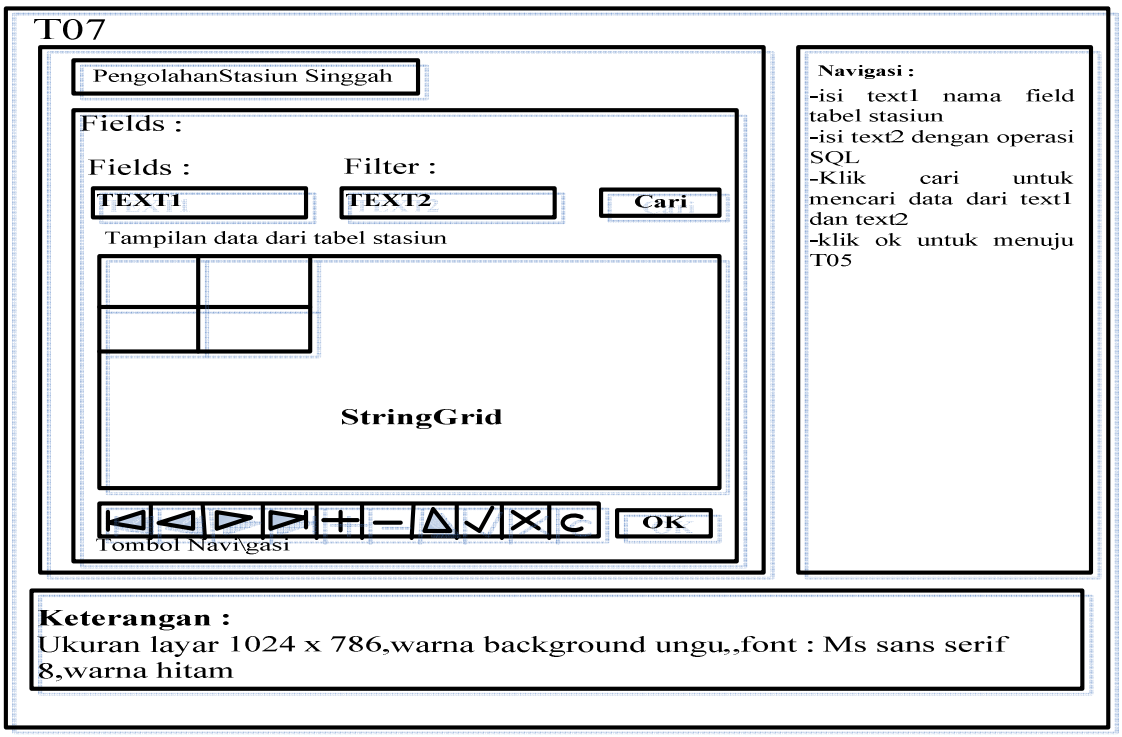
Gambar 4.11 Perancangan antamuka Form Informasi Stasiun Singgah

4.8.6 Perancangan Form Pengolahan Buka Data



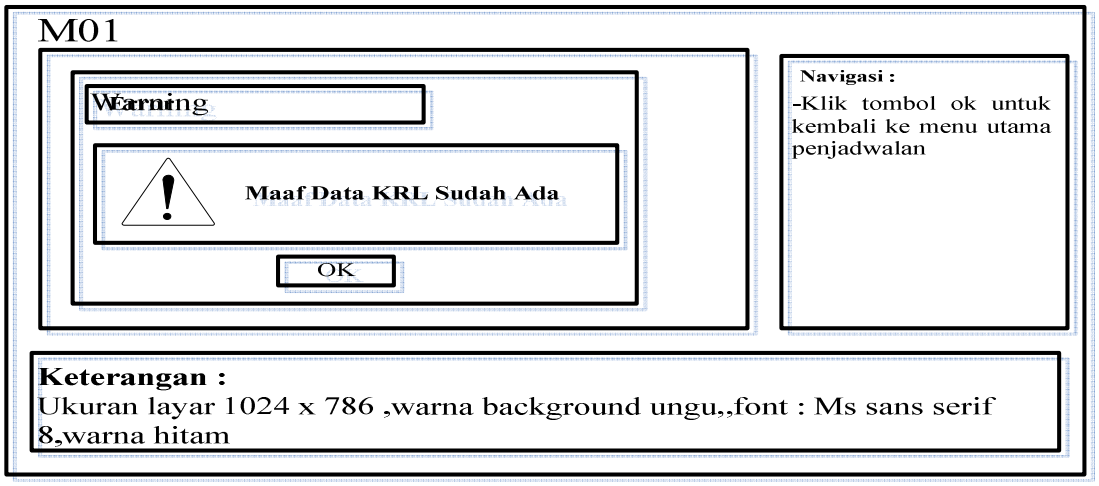
Gambar 4.12 Perancangan antamuka Form Pengolahan Buka Data

4.8.7 Perancangan Form Pengolahan Stasiun Singgah



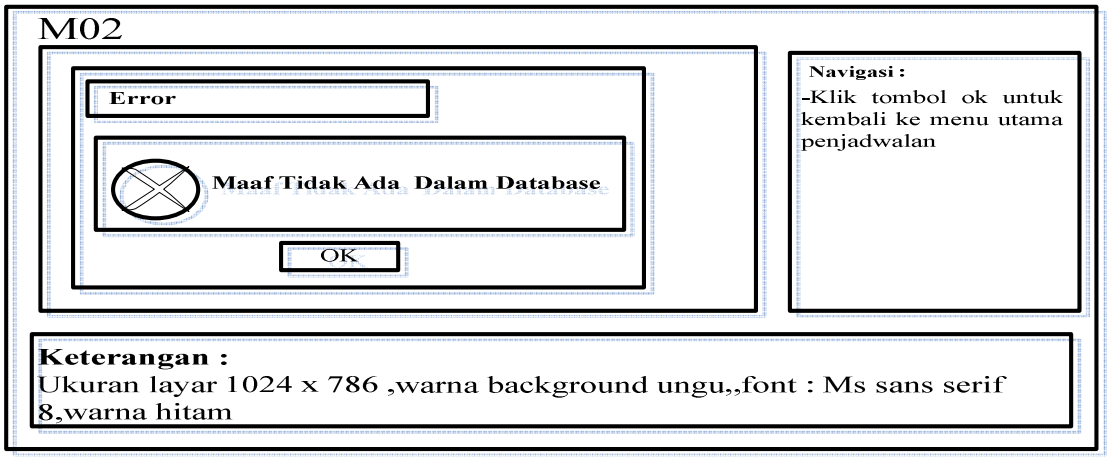
Gambar 4.13 Perancangan antamuka Form Pengolahan Stasiun Singgah

4.8.8 Tampilan Pesan Ketika Memasukan Data KRL Yang Sama



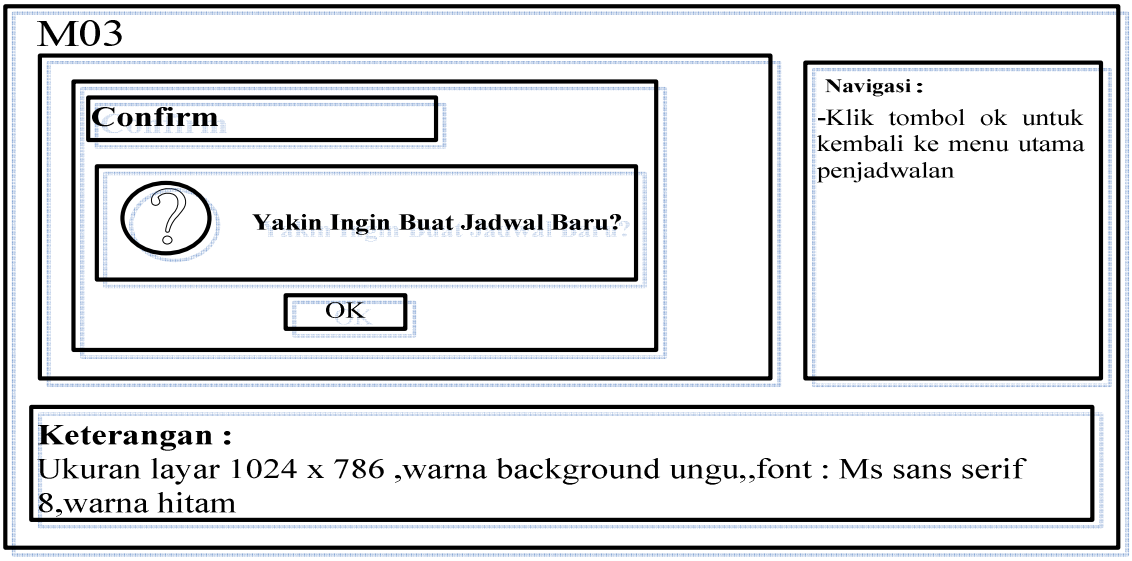
Gambar 4.14 Perancangan antamuka Pesan Ketika Memasukan Data KRL Yang Sama

4.8.9 Tampilan Pesan Ketika Salah Memasukan Nama Stasiun Dalam From Input Data Stasiun



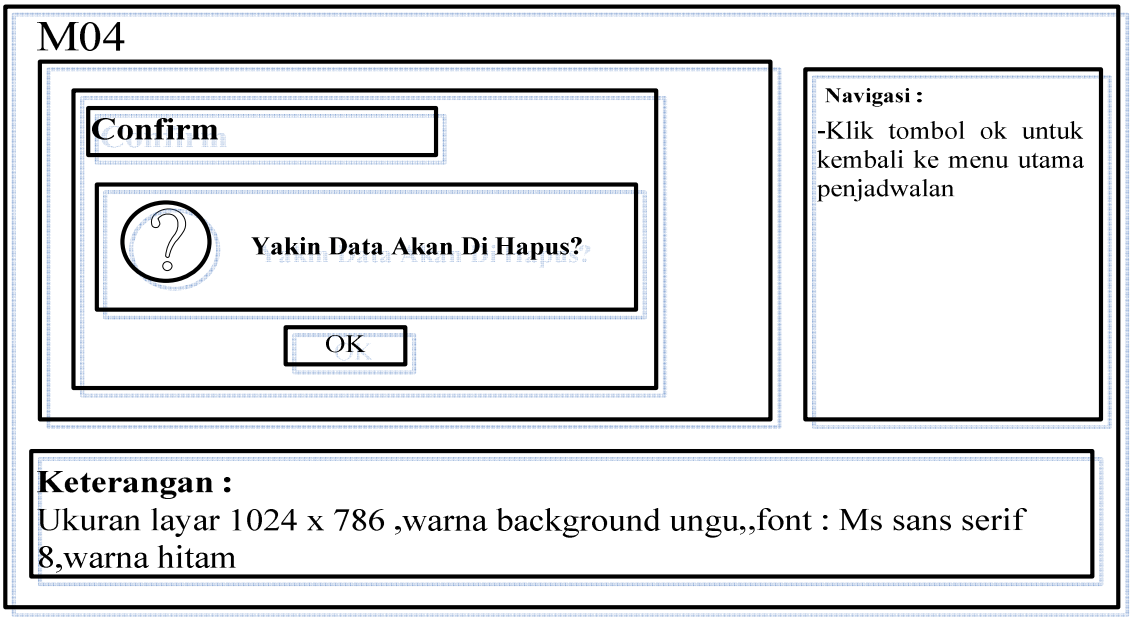
Gambar 4.15 Perancangan antamuka Pesan Ketika Salah Memasukan Nama Stasiun Dalam From Input Data Stasiun

4.8.10 Tampilan Pesan Ketika Menekan Tombol Jadwal Baru



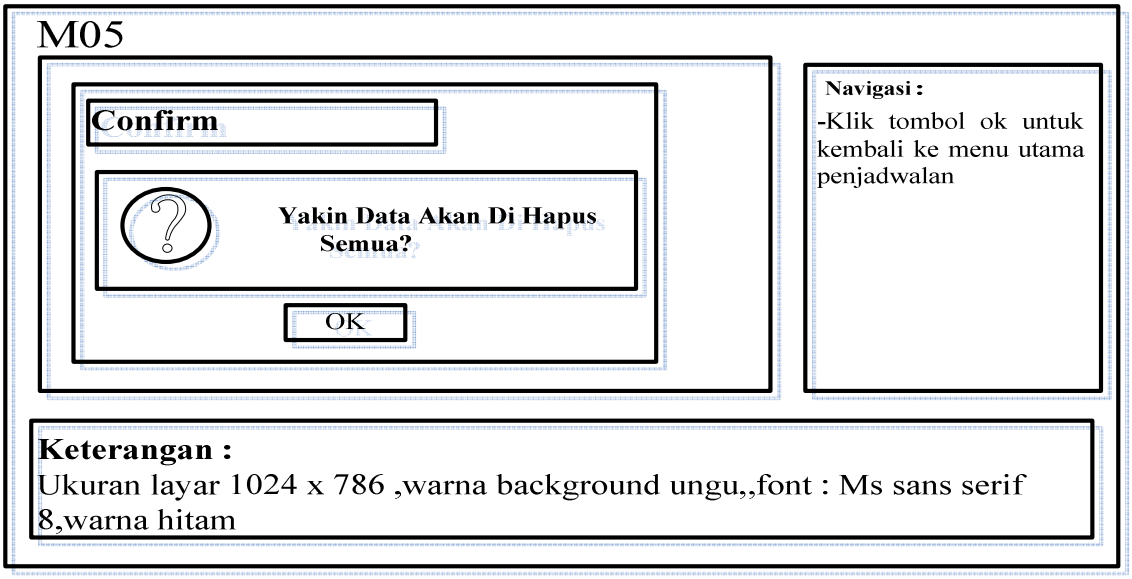
Gambar 4.16 Perancangan antamuka Pesan Ketika Menekan Tombol Jadwal Baru

4.8.11 Tampilan Pesan Menghapus Data KRL



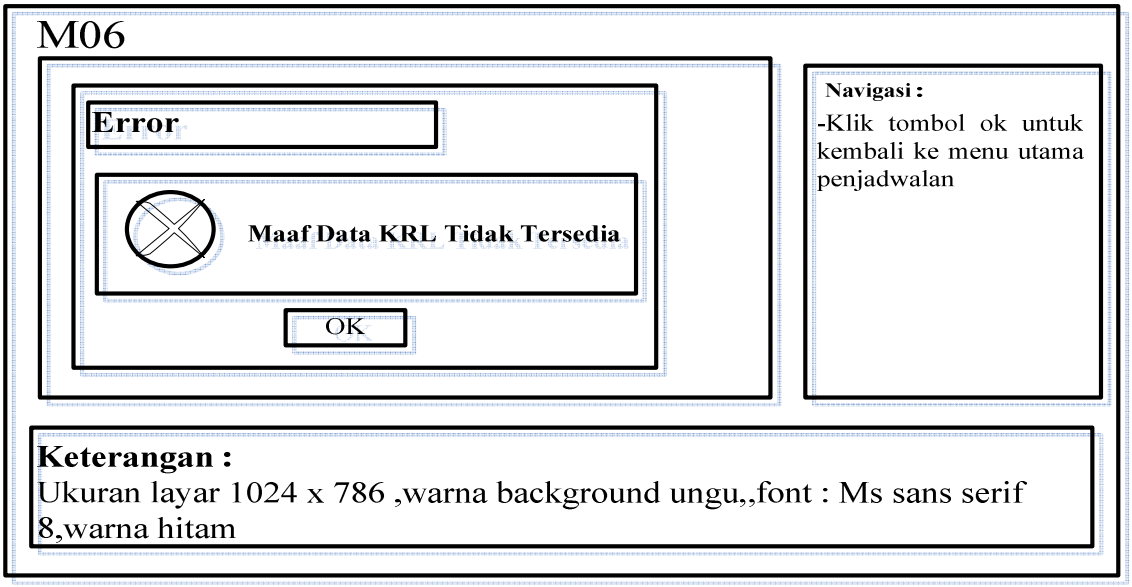
Gambar 4.17 Perancangan antamuka Pesan Menghapus Data KRL

4.8.12 Tampilan Pesan Ketika Tekan Tombol Hapus



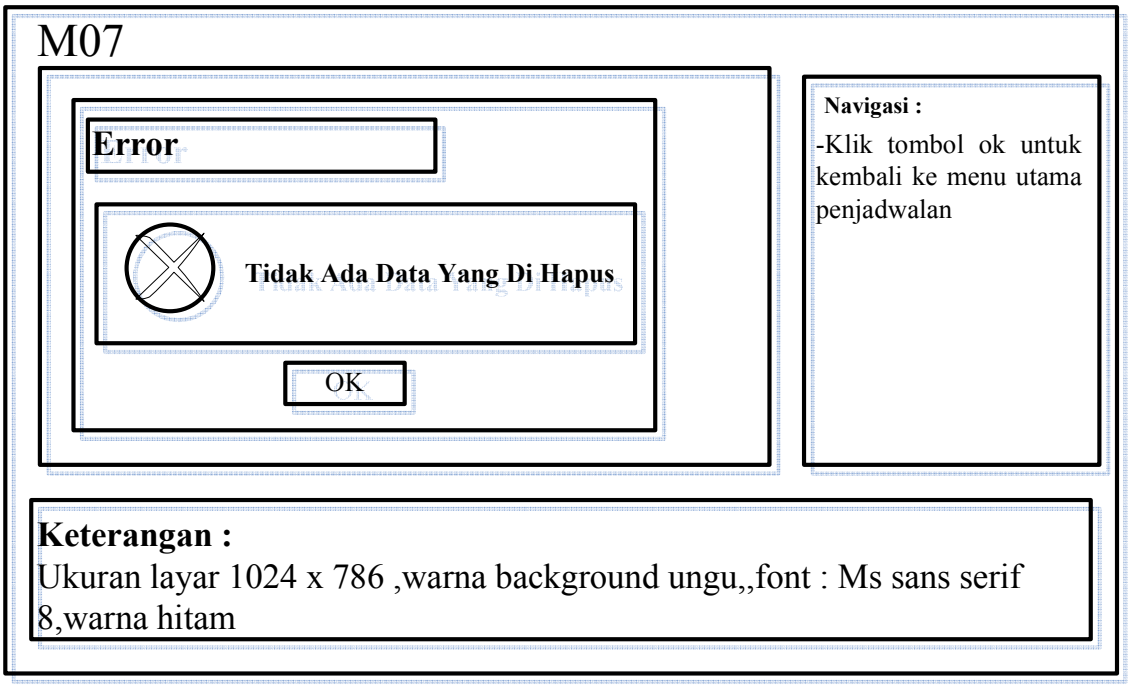
Gambar 4.18 Perancangan antamuka Pesan Ketika Tekan Tombol Hapus

4.8.13 Tampilan Pesan Ketika Data KRL Tidak Ada



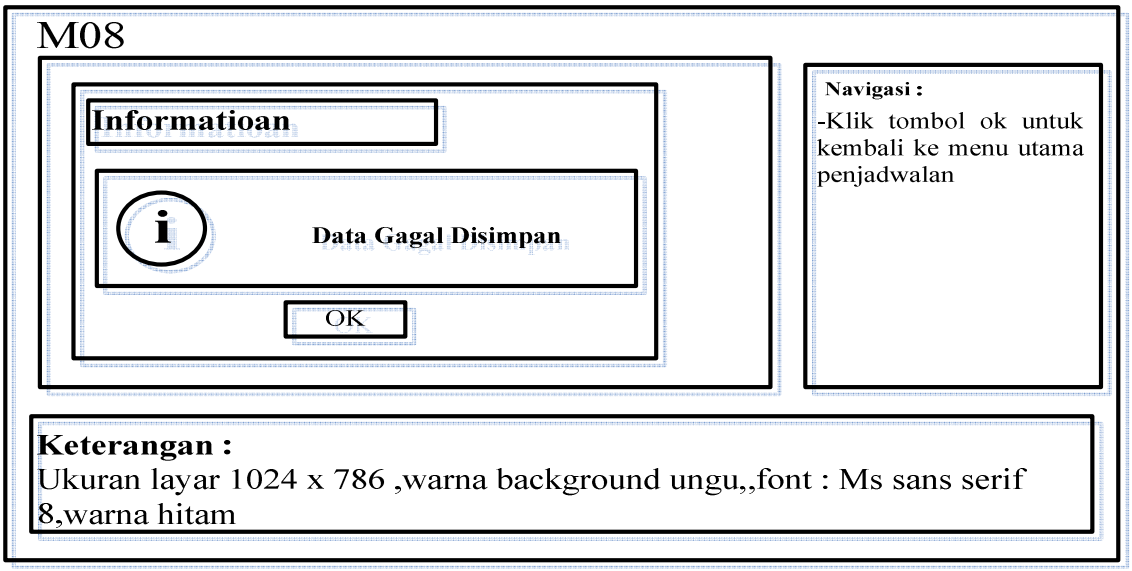
Gambar 4.19 Perancangan antamuka Pesan Ketika Data KRL Tidak Ada

4.8.14 Tampilan Pesan Ketika Tidak Ada Data Yang Di Hapus



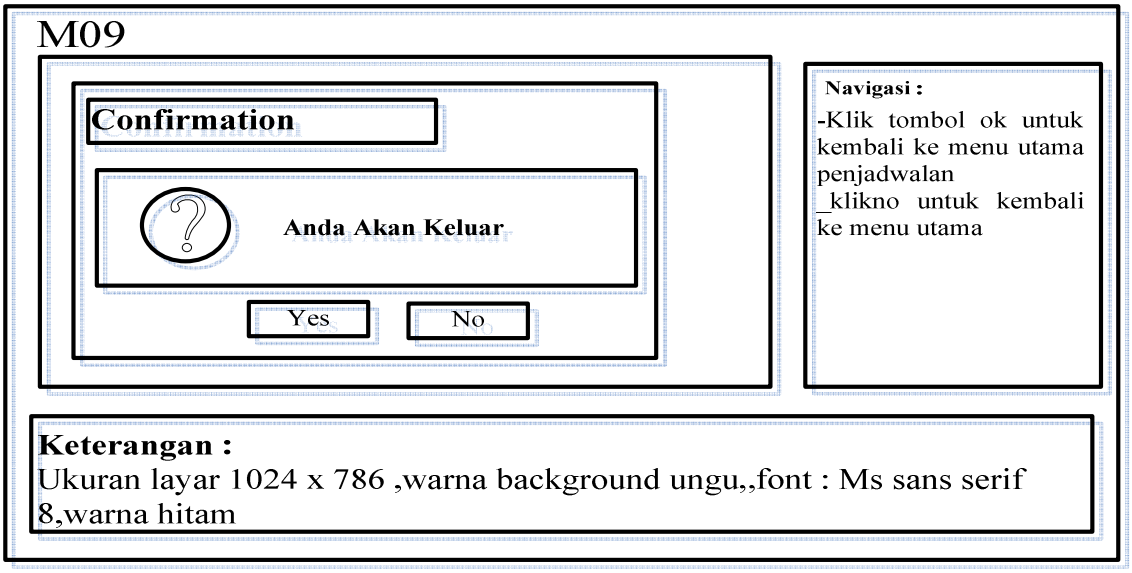
Gambar 4.20 Perancangan antamuka Pesan Ketika Tidak Ada Data Yang Di Hapus

4.8.15 Tampilan Pesan Ketika Data Gagal Di simpan



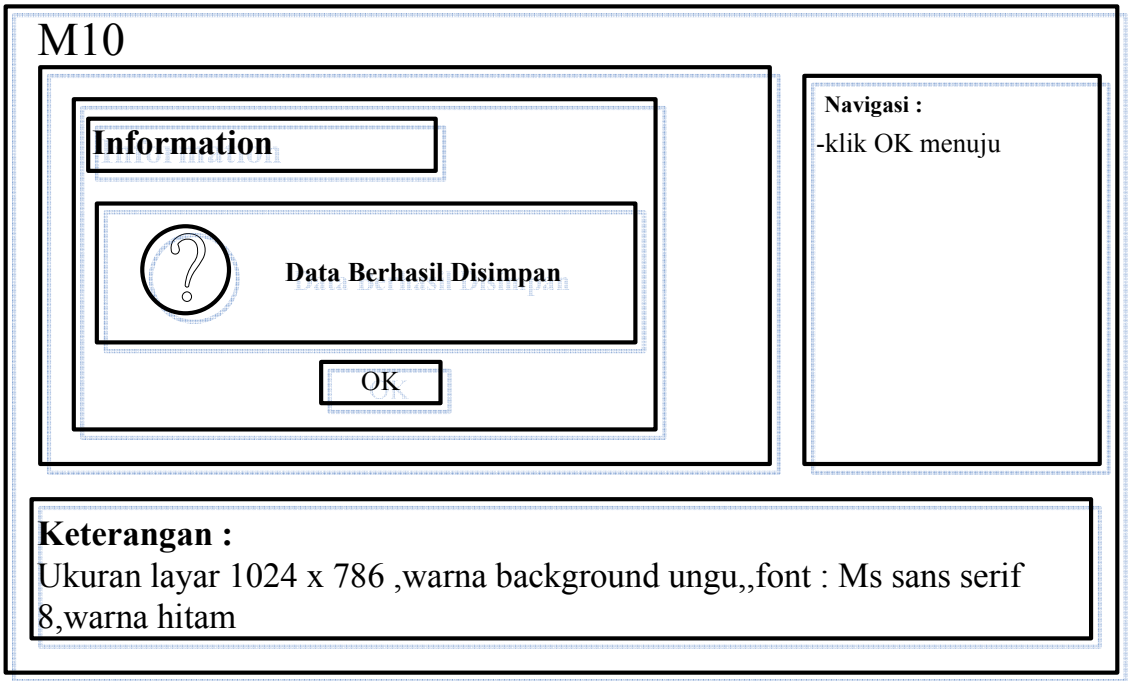
Gambar 4.21 Perancangan antamuka Pesan Ketika Data Gagal Di simpan

4.8.16 Tampilan Pesan Ketika Menekan Tombol Keluar



Gambar 4.22 Perancangan antamuka Pesan Ketika Menekan Tombol Keluar

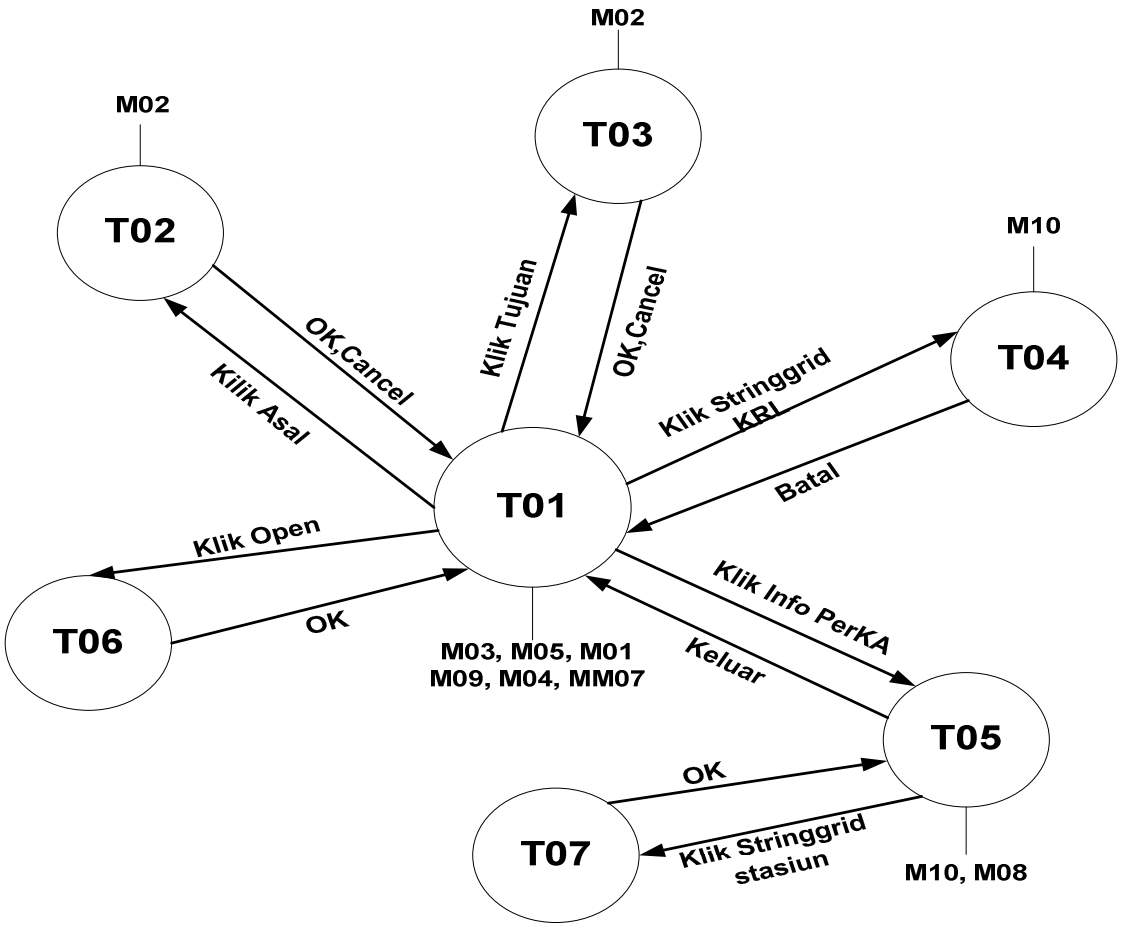
4.8.17 Tampilan Pesan Ketika Data Berhasil Di Simpan



Gambar 4.23 Perancangan antamuka Pesan Ketika Data Berhasil Di Simpan

4.9 Jaringan Semantik

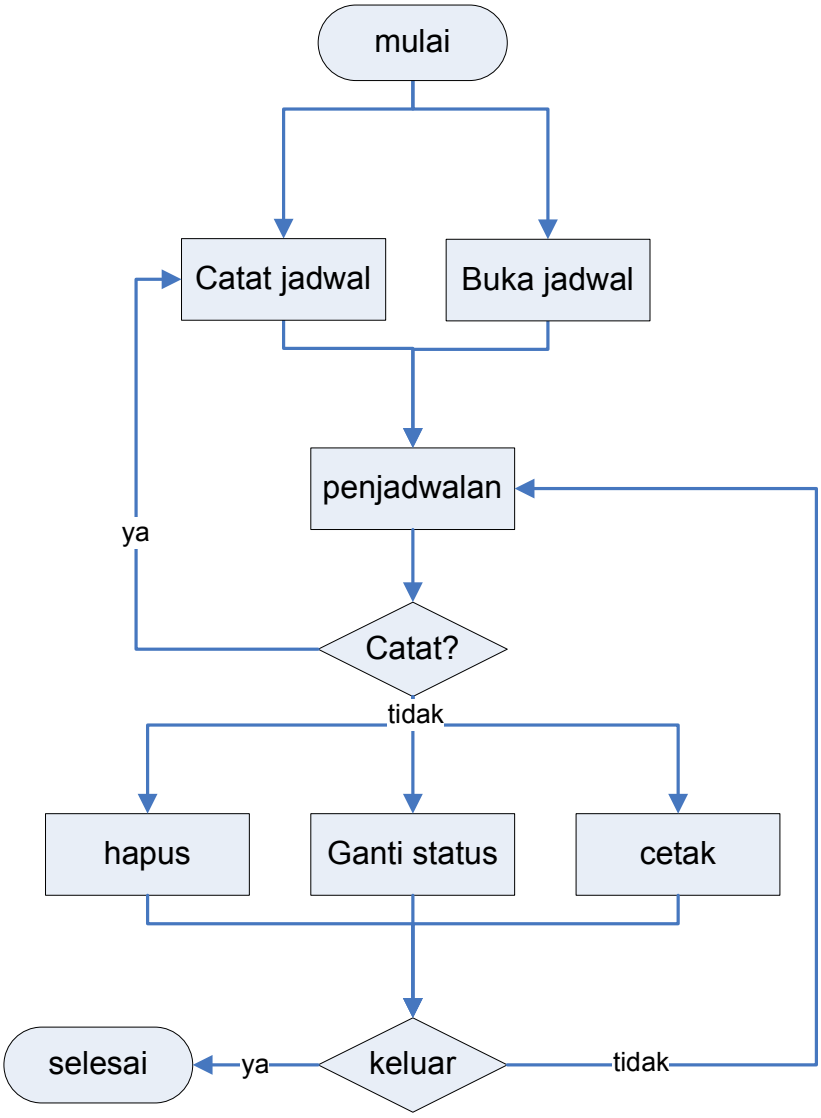
Jaringan semantik dari penjadwalan kereta api rel listrik yang akan di bangun adalah sebagai berikut :



Gambar 4.24 Perancanga Jaringan Semantik Kereta Api Rel Listrik Yang Akan Di Bangun

4.10 Perancangan Prosedural

Perancangan procedural merupakan perancanagn yang di lakukan untuk menetapkan detail algoritma yang akan dinyatakan ke dalam suatu program. Adapun perancangan procedural yang akan di bangun adalah sebagai berikut :



Gambar 4.25 FlowChart Penjadwalan Kereta Api Rel Listrik

BAB V

Penutup

5.1. Kesimpulan

1. Sistem penjadwalan ini adalah sebuah sistem yang akan memberikan solusi kepadatan penjadwalan kereta rel listrik di jabotabek meliputi info singgah kereta api,jarak tempuh kereta dan info rangkaian kereta api.
2. Setiap keberangkatan Kereta rel Listrik(KRL) memungkinkan dapat melakukan pembatalan dan perawatan.
3. Perangkat lunak pengolahan penjadwalan KRL ini dapat membuat laporan hasil dari penjadwalan yang ada.

5.2 Saran

Perancangan penjadwalan kereta api rel listrik ini juga dapat membantu untuk membuat implementasi kereta api rel listrik sesuai dengan perancangan yang telah ada.

Perangkat lunak yang dibuat telah memenuhi standar dari gambaran umum Kereta Rel Listrik (KRL) yang telah dipaparkan diatas.

Mudah-mudahan dengan perangkat lunak ini dapat membantu pekerjaan penyelesaian dalam penjadwalan operasional Kereta Rel Listrik (KRL) di Jabotabek

Daftar Pustaka

Roger P. (2002), Rekaya Perangkat Lunak, Andi, Yogyakarta, 4, 39-42

Fathansyah (2004)**,** *Basis Data*, Penerbit INFORMATIKA Bandung,52-65.

Heric Hendarto (Minggu, 22 Desember 2008), *Kereta Api Rel Listrik*, *http://www.wikipedia.co.id/*

Meirino (3 januari 2009), *Prototype Model*, http://one.indoskripsi.com/judul-skripsi-tugas-makalah/tugas-kuliah-lainnya/prototype-mode

Ian Sommerville (2003), *Software Engineering Jilid 2,* Erlangga, Jakarta, 87-92