



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam membangun aplikasi ini antara lain adalah:

1. Studi Literatur

Penulis melakukan studi tentang perbengkelan, algoritma *greedy*, dan *knapsack problem*. Studi dilakukan melalui artikel, jurnal, dan buku.

2. Perancangan Aplikasi dan Sistem

Perancangan aplikasi dan sistem menggunakan aplikasi *Microsoft Visio* dan *Power Designer* untuk membuat *flowchart* dan *data flow diagram*.

3. Pembuatan Desain dan Pemrograman

Pada tahap ini penulis merancang desain antarmuka dan membuat aplikasi dengan netbeans.

4. Uji Coba

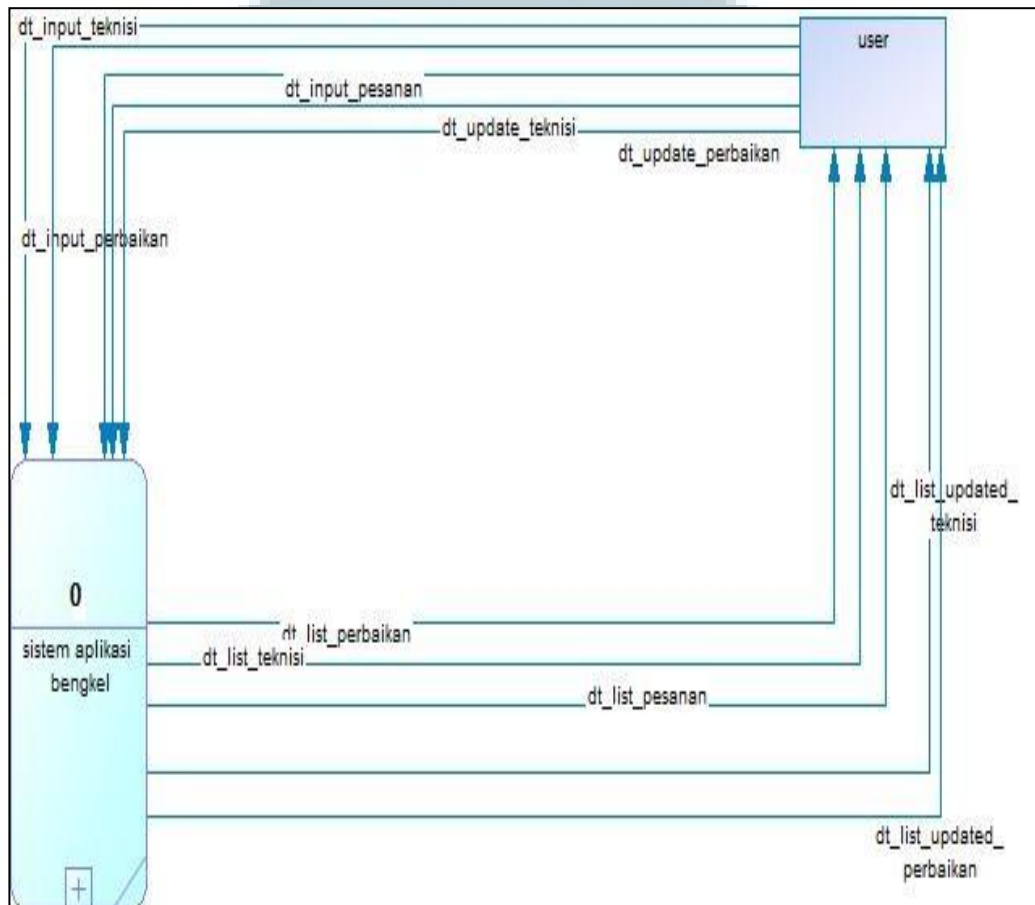
Aplikasi yang telah dibuat diuji coba apakah sesuai hasil, dan apakah ada keasalahan dalam program.

3.2 Perancangan

Perancangan aplikasi ini menggunakan *flowchart* diagram untuk menggambarkan alur proses jalannya aplikasi, *data flow* diagram untuk menggambarkan aliran data dari suatu proses ke proses lain, dan *entity*

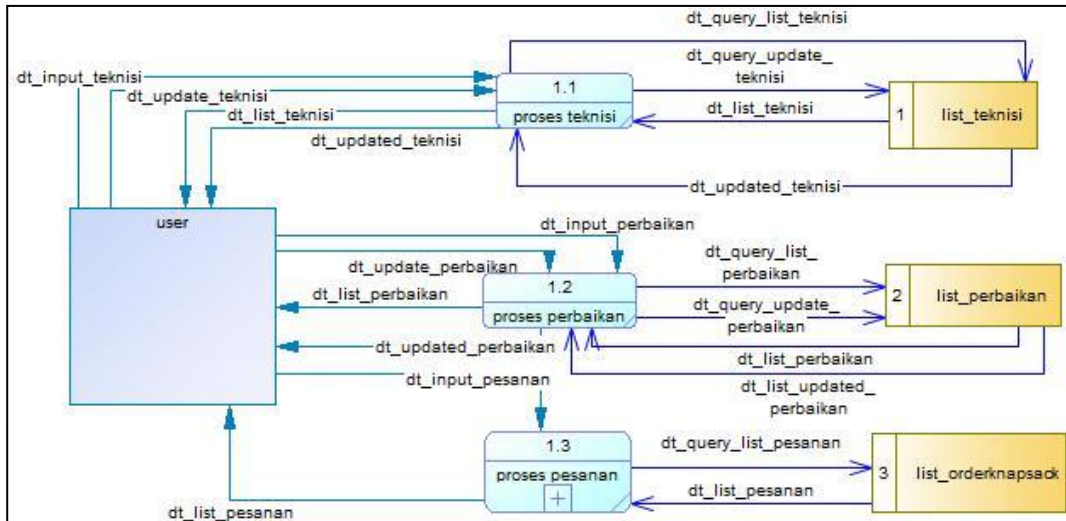
relationship diagram yang menggambarkan hubungan relasi antar tabel yang ada dalam aplikasi.

3.2.1 Data Flow Diagram



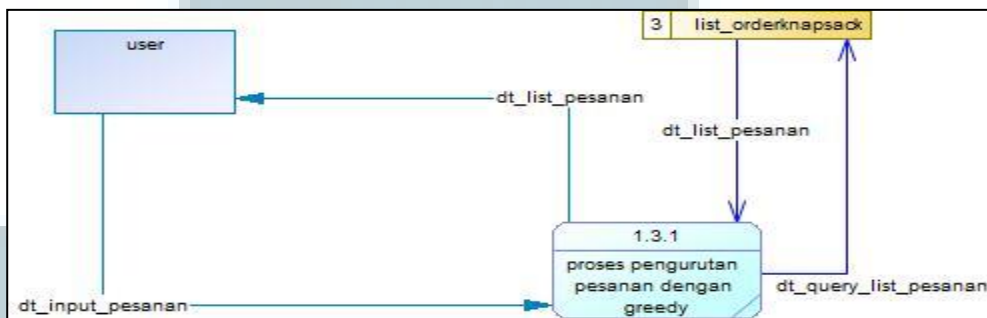
Gambar 3.1 Context Diagram

Pada gambar 3.1 Menjelaskan hubungan data antara entitas *user* dengan sistem aplikasi bengkel, dimana entitas *user* mengirimkan 5 data, dan sistem aplikasi bengkel mengembalikan 5 data ke entitas *user*. Untuk memperjelas aliran di buat dfd level 1



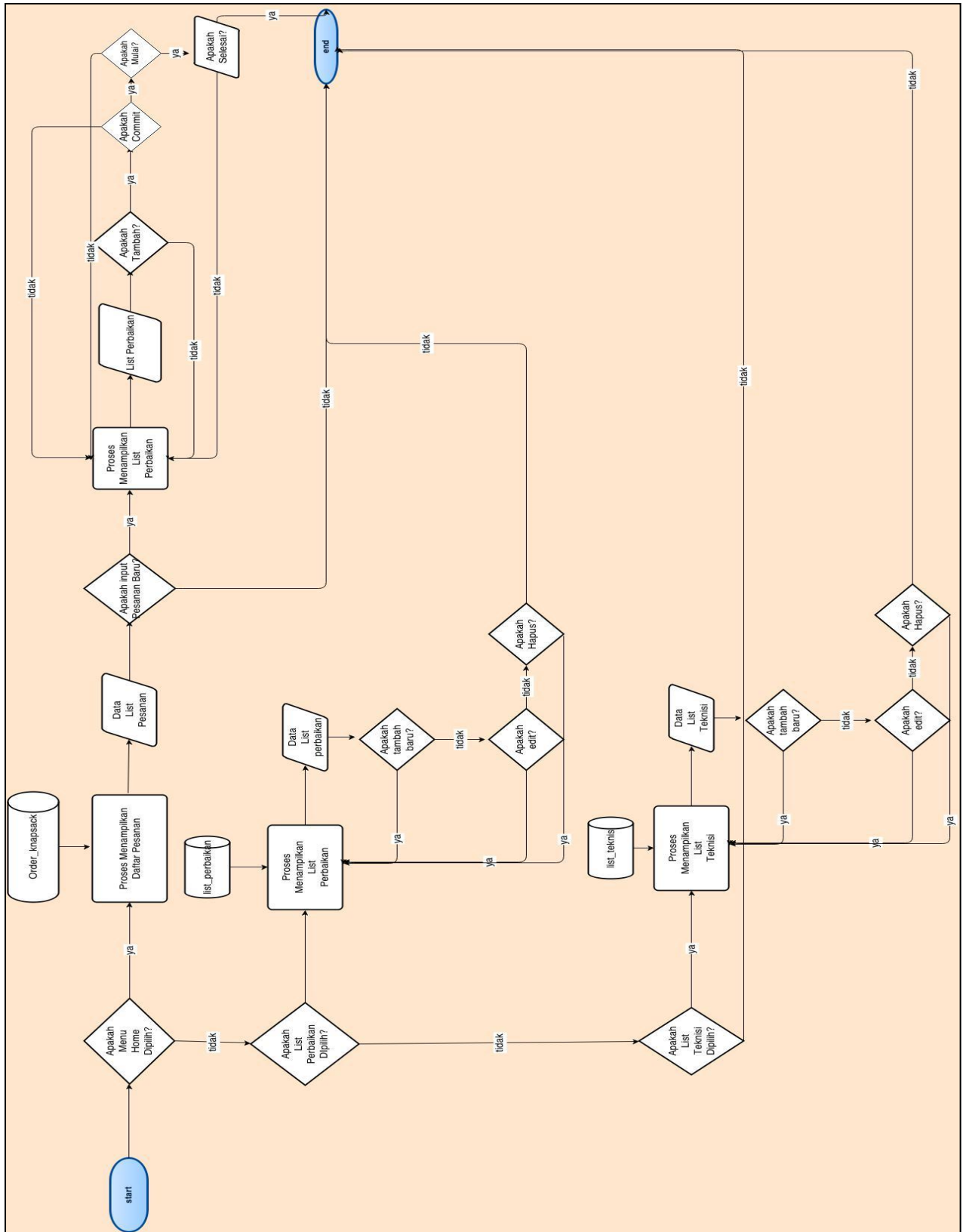
Gambar 3.2 data flow diagram level 1

Gambar 3.2 menjelaskan aliran data dari entitas *user* ke proses pesanan, teknisi, dan perbaikan jika *user* melakukan perubahan data. Proses terhubung dengan datastore penyimpanan yaitu tabel *list_perbaikan* yang terhubung dengan proses perbaikan, tabel *list_teknisi* yang terhubung dengan proses teknisi, tabel *order_knapsack* yang terhubung dengan proses pesanan.



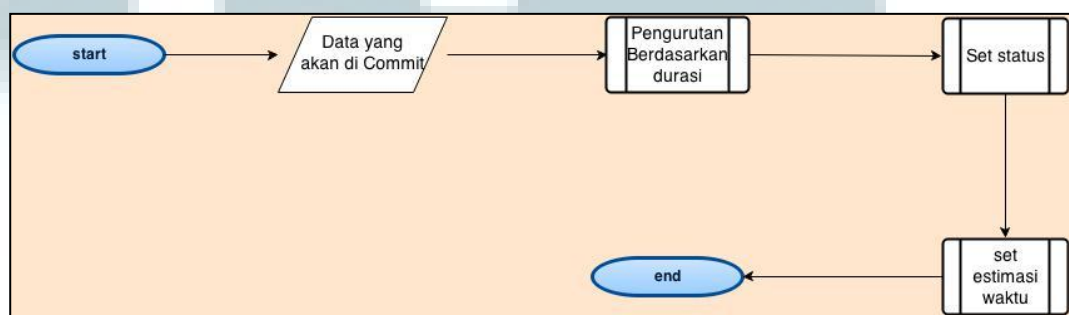
Gambar 3.3 data flow diagram level 2 proses pengurutan dengan algoritma greedy

3.2.2 Flowchart



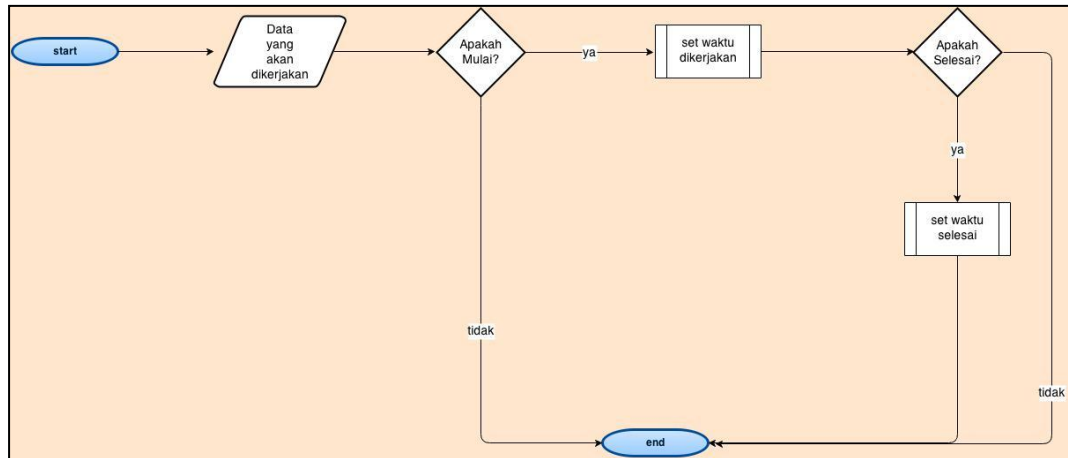
Gambar 3.4 Flowchart sistem dan sub sistem

Gambar 3.4 di atas menjelaskan diagram aliran proses dari sistem aplikasi optimasi waktu kerja bengkel ini secara keseluruhan. Pada *menu* utama terdapat daftar list pesanan yang di ambil dari tabel *order_knapsack*, *user* dapat menambah pesanan baru dengan memasukkan nama pemesanan juga perbaikan yang akan di lakukan. Setelah itu akan ada proses *commit* yang akan di jelaskan pada gambar 3.5, setelah di *commit* maka dapat mulai dikerjakan, dan setelah selesai dapat memilih selesai dikerjakan. Selain menu utama terdapat 2 menu lainnya yaitu *list* perbaikan dan *list* teknisi, disana *user* dapat melakukan proses *edit data*, *delete data*, dan tambah *data*.



Gambar 3.5 *Flowchart* proses *commit* setelah *input* pesanan

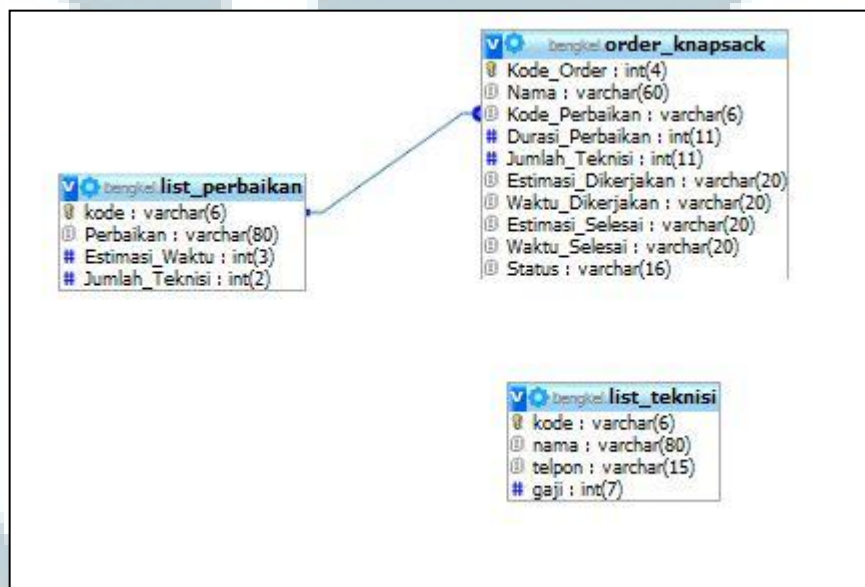
Gambar 3.5 Menjelaskan aliran proses saat sebuah data di-*commit* maka akan langsung diurutkan berdasarkan durasi pengerjaan sesuai algoritma *greedy*, lalu akan di *set status*, *waiting* jika semua teknisi sedang sibuk, *assign* jika ada teknisi yang tersedia, lalu *set* estimasi waktu pengerjaan dan selesai.



Gambar 3.6 Flowchart proses pengerjaan

Gambar 3.6 menjelaskan aliran proses pengerjaan setelah di lakukan *commit* mulai dari set waktu pengerjaan , lalu set waktu selesai setelah selesai dikerjakan

3.2.4 Entity Relationship Diagram



Gambar 3.7 Entity Relationship Diagram

Dari gambar 3.7 terlihat adanya relasi antara kode perbaikan pada tabel list_perbaikan yang bersifat unik dengan kode perbaikan pada tabel order_knapsack yang mempunyai kode yang sama untuk setiap pemilihan perbaikan.

3.2.4 Struktur Tabel

Pada aplikasi ini menggunakan Mysql sebagai *database* yang terdiri dari 3 tabel yaitu list_perbaikan , list_teknisi, dan order_knapsack berikut tabel-tabel yang digunakan :

Nama Tabel : list_perbaikan

Fungsi : menyimpan tipe-tipe perbaikan yang ada pada bengkel

Tabel 3.1 Tabel list_perbaikan

No.	Nama <i>field</i>	Tipe data	Keterangan
1.	Kode	varchar	Kode perbaikan unik dan <i>primary key</i>
2.	Perbaikan	varchar	Nama perbaikan
3.	Estimasi_Waktu	int	Estimasi waktu lama perbaikan
4.	Jumlah_Teknisi	int	Jumlah teknisi yang digunakan untuk perbaikan

Nama Tabel : list_teknisi

Fungsi : menyimpan data teknisi yang ada pada bengkel

Tabel 3.2 Tabel list_perbaikan

No.	Nama <i>field</i>	Tipe data	Keterangan
1.	Kode	varchar	Kode teknisi unik dan <i>primary key</i>
2.	Nama	varchar	Nama teknisi
3.	Telpon	varchar	No telpon teknisi
4.	Gaji_	int	Gaji

Nama Tabel : list_teknisi

Fungsi : menyimpan data pengerjaan

Tabel 3.3 Tabel order_knapsack

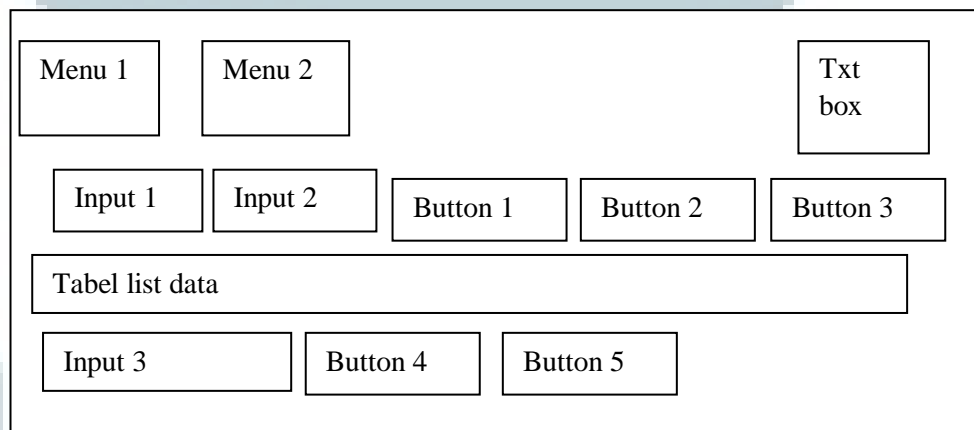
No.	Nama <i>field</i>	Tipe data	Keterangan
1.	Kode order	int	Kode pesanan unik, <i>primary key</i>
2.	Nama	varchar	Nama pemesan

Tabel 3.3 Tabel order_knapsack(lanjutan)

No.	Nama <i>field</i>	Tipe data	Keterangan
3.	Kode_perbaikan	varchar	Kode perbaikan yang dilakukan, <i>foreign key</i>
4.	durasi_perbaikan	int	Lama waktu perbaikan
5.	Jumlah_teknisi	int	Jumlah teknisi yang di pakai
6.	Estimasi_dikerjakan	varchar	Estimasi dimulai pengerjaan
7.	Waktu_dikerjakan	varchar	Waktu mulai pengerjaan
8.	Estimasi_selesai	varchar	Estimasi waktu selesai pengerjaan
9.	Waktu_selesai	varchar	Waktu selesai pengerjaan
10.	Status	varchar	Status pesanan

3.3 Sketsa Layar

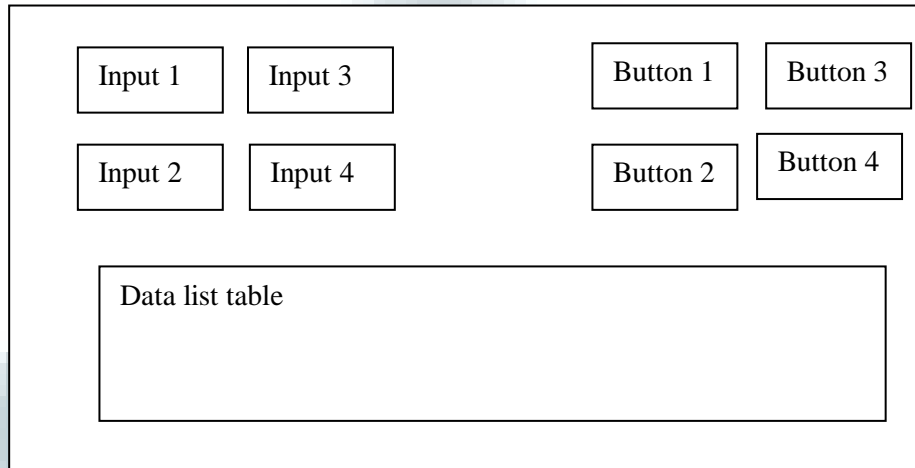
Beberapa tampilan sketsa menu-menu yang ada pada aplikasi ini adalah:



Gambar 3.8 Sketsa Menu utama

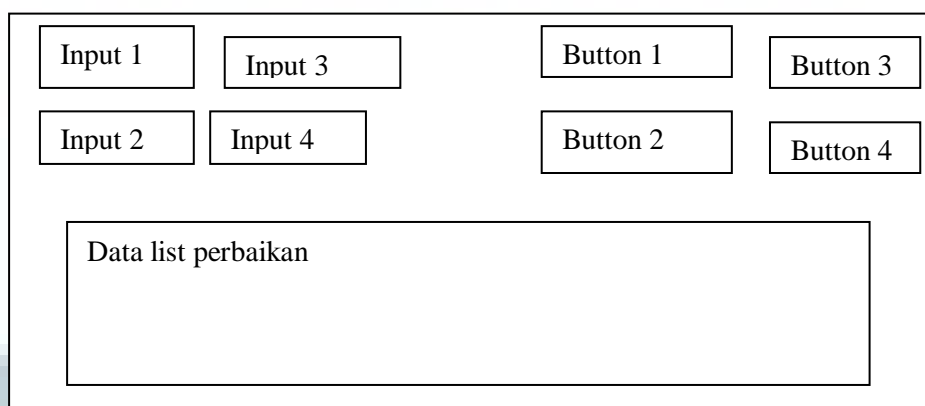
Sketsa *menu* utama dari aplikasi ini terdapat 2 menu yaitu menu utama dan menu untuk *list* perbaikan dan *list* teknisi yang ada, lalu ada *text box* untuk menampilkan jumlah teknisi yang tersedia. Ada 3 *input* untuk memasukkan nama pelanggan, perbaikan yang ingin dilakukan, dan *input* kode perbaikan yang sudah

selesai. Tabel *list data* untuk menampilkan semua pekerjaan yang di lakukan. Terdapat 5 *button* untuk menu utama ini.



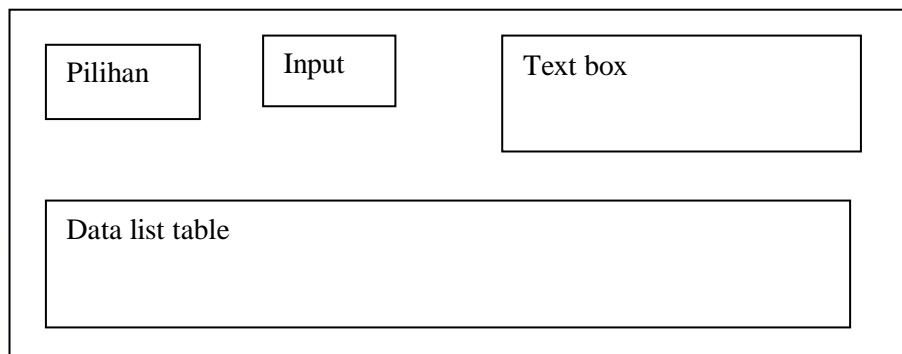
Gambar 3.9 Sketsa *Menu list* teknisi

Terdapat 4 *input* dan *button* untuk aplikasi ini yang berguna untuk memasukkan data-data teknisi yang ada. *Button* digunakan untuk menambah, menghapus, delete, dan refresh. Data list table untuk menampilkan data yang ada.



Gambar 3.10 Sketsa *Menu list* perbaikan

Menu ini digunakan untuk *list* perbaikan yang ada. Memasukkan data-data perbaikan serta menghapus, *refresh*, *edit*, dan menambah data.



Gambar 3.11 Sketsa *Search*

Sketsa ini menggambarkan pop up search untuk mencari perbaikan berdasarkan kode atau nama perbaikan.