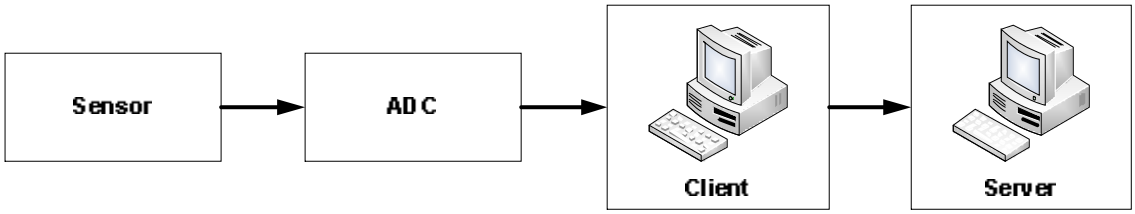
BAB III
PERANCANGAN SISTEM

3.1 Diagram Blok Sistem

Diagram blok dari sistem yang dirancang terdiri dari bagian sensor, ADC, komputer *client* dan komputer *server* beserta perangkat lunaknya, seperti yang ditunjukkan pada gambar 3.1.



Gambar 3.1. *Diagram Blok Sistem*

Sistem yang dibuat terdiri atas tiga bagian utama yaitu bagian sensor, akuisisi data dan bagian perangkat lunak yang dijalankan dengan komputer. Pada bagian sensor digunakan sensor tekanan flexiforce dimana masukan dari tekanan akan diterima oleh flexiforce, keluaran dari flexiforce tidak dapat langsung diterima oleh komputer karena keluarannya masih bersifat analog. Data berupa level tegangan tertentu yang masuk pada kaki masukan ADC kemudian akan dikonversi menjadi data digital 8 bit dan ditrasmisikan ke komputer dengan menggunakan *port* paralel.

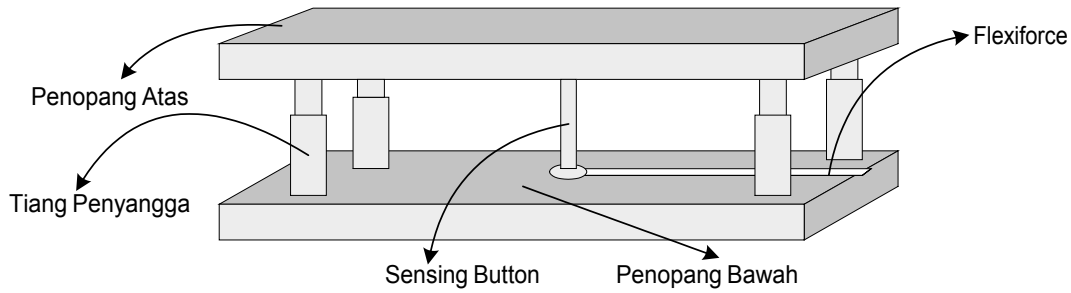
Dengan bantuan bahasa pemrograman Delphi dilakukan konversi terhadap data 8 bit yang dibaca dari *port* paralel menjadi data berupa bilangan berbasis 10 (bilangan desimal). Selanjutnya data tersebut dimasukkan ke persamaan atau formulasi tertentu untuk mendapatkan hasil berupa bilangan tertentu dalam satuan berat dan dibandingkan dengan data yang tersimpan pada *database*. Apabila data tersebut valid sesuai dengan kriteria tertentu, maka data hasil pengukuran dan data lain yang diperlukan akan disimpan di *database* komputer *client* yang bertujuan untuk menjaga apabila komputer *server* sedang tidak aktif atau terjadi gangguan pada jaringan. Data tersebut kemudian dikirim ke komputer *server* yang bertujuan sebagai monitoring sekaligus sebagai *backup*. Data tersebut kemudian disimpan di *database server*.

3.2 Perancangan Perangkat Keras

Alat ukur yang dirancang terdiri dari 6 subsistem utama, yaitu sistem mekanik, sensor, ADC, *port* paralel, catu daya, dan pemrograman (baik pemrograman pada komputer *client* maupun pemrograman pada komputer *server*). Ke-6 subsistem ini memiliki keterkaitan fungsi untuk memperoleh data hasil pengukuran berat kendaraan berbasis TCP/IP.

3.2.1 Mekanik

Desain mekanik merupakan salah satu faktor penting dalam membangun sebuah alat ukur berat kendaraan agar dapat menyatukan keseluruhan subsistem dalam satu kesatuan yang utuh.



Gambar 3.2. *Mekanik alat ukur*

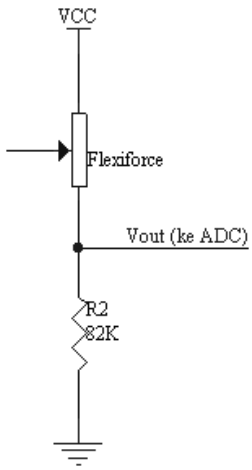
Sistem mekanik dari alat ukur berat kendaraan berbasis *client-server* terdiri dari : penopang atas yang berfungsi sebagai permukaan atau tempat untuk menyimpan benda yang akan diukur yang dalam hal ini adalah kendaraan, empat buah tiang penyangga yang berfungsi untuk menyangga penopang atas, *sensing button* berfungsi untuk memfokuskan pemberian beban terhadap sensor dan penopang bawah berfungsi sebagai alas untuk memperkokoh mekanik serta tempat untuk menyimpan sensor dan komponen elektronika.

3.2.2 Sensor

Sensor merupakan sebuah komponen atau alat yang berfungsi untuk merespon kondisi lingkungan. Pada alat yang dirancang menggunakan sensor tekanan flexiforce.

3.2.2.1 Flexiforce

Sebagaimana bahasan sebelumnya dijelaskan bahwa sensor pengukur berat beban flexiforce berperan sebagai resistor yang nilainya dapat diubah oleh beban yang diberikan. Oleh karena itu sensor ini dapat diaplikasikan sebagai rangkaian pembagi tegangan seperti terlihat pada gambar 3.3.



Gambar 3.3. *Rangkaian Flexiforce*

3.2.3 ADC (*Analog to Digital Converter*)

ADC 0804 digunakan untuk mengubah sinyal analog menjadi sinyal digital. ADC 0804 mampu menghasilkan data konversi sebesar 8 bit dengan level kuantitas 256. ADC 0804 menggunakan metode konversi *successive-approximation.* ADC 0804 ini mempunyai resolusi 0.0192 V/1D. Waktu konversi dari ADC 0804 adalah 100 μs. Awal konversi ditandai dengan pin *interupt* bernilai *high*, WR berfungsi untuk mengatur mode konversi *write* dan *read*. Gambar rangakaian ADC 0804 dapat dilihat pada gambar 3.4.

Persamaan berikut ini dapat digunakan untuk menentukan keluaran yang dihasilkan, yaitu :

$$\frac{A_{in}}{V_{reff}} = \frac{D_{out}}{256} \quad \text{.....(3.1)}$$

atau

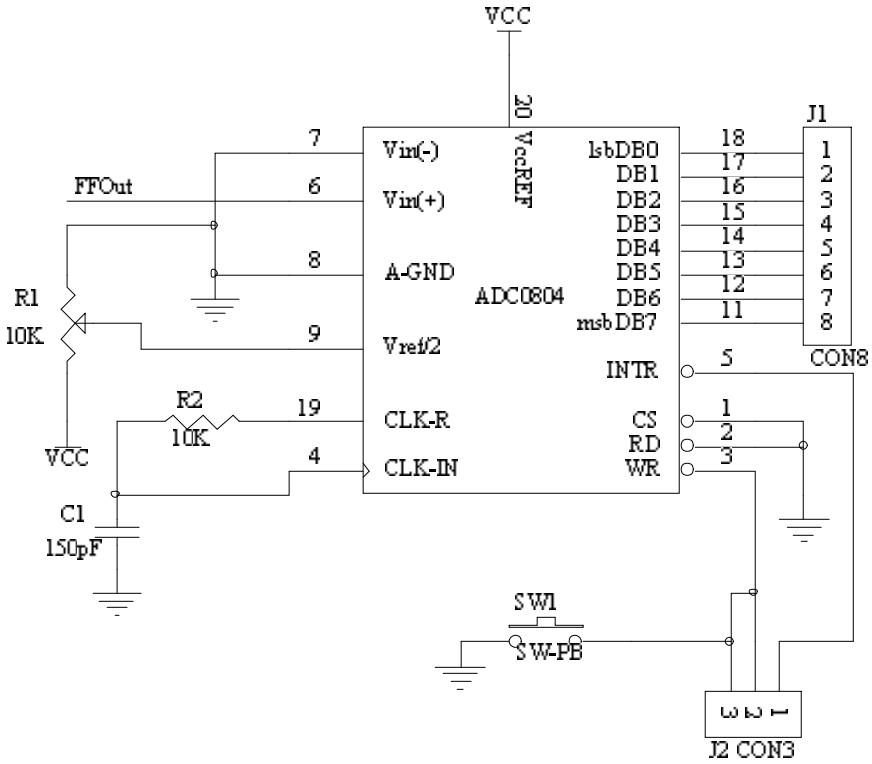
$$D_{out} = \frac{A_{in}}{V_{reff}} x\ 256 \quad \text{.....(3.2)}$$

Dimana A_{in} = masukan dari sensor

V_{reff} = tegangan referensi

D_{out} = keluaran digital (bilangan desimal)

256 = jumlah keseluruhan step keluaran digital (0 - 255)

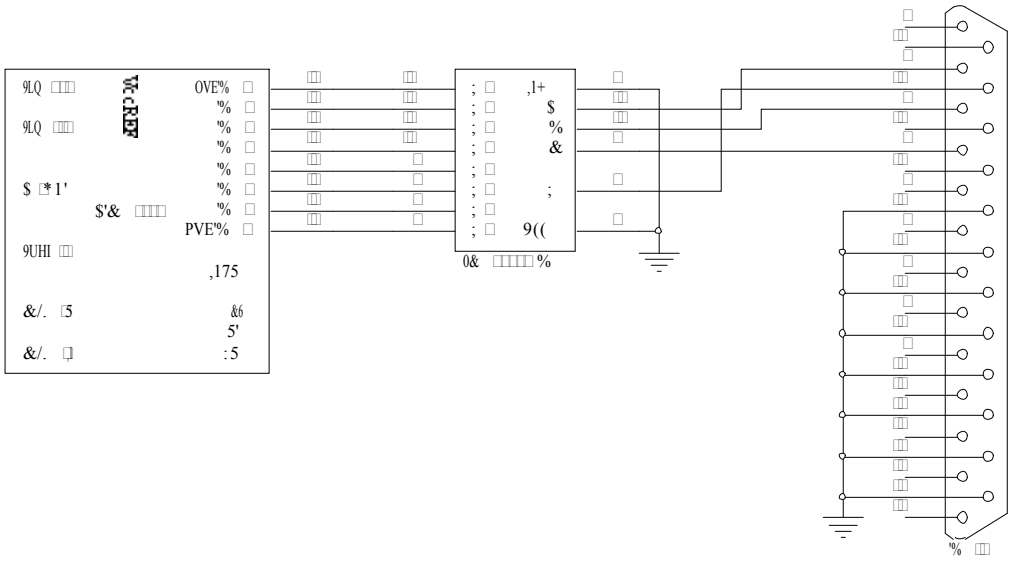


Gambar 3.4. *Rangkaian ADC0804*

Mode operasi yang digunakan dalam perancangan ini mode operasi kontinyu (proses membaca terus menerus dan tanpa proses operasi jabat tangan), maka pin CS dan RD ditanahkan. Prinsip kerja operasi kontinyu ini yaitu ADC akan memulai konversi ketika pin INTR kembali tidak aktif (*high*). Setelah proses konversi selesai, pin INTR akan aktif (*low*). Untuk memulai konversi pertama kali WR harus ditanahkan terlebih dahulu, hal ini digunakan untuk mereset SAR.Namun pada konversi berikutnya untuk mereset SAR dapat menggunakan sinyal INTR saat aktif (*low*) dan mulai konversi saat tidak aktif (*high*).Ketika selesai konversi data hasil konversi akan dikeluarkan secara langsung dari buffer untuk dibaca karena pin RD ditanahkan. Saat sinyal INTR aktif, sinyal ini digunakan untuk me-reset SAR. Saat pin INTR kembali tidak aktif (*high*) proses konversi dimulai kembali.

3.2.4 *Port* Paralel

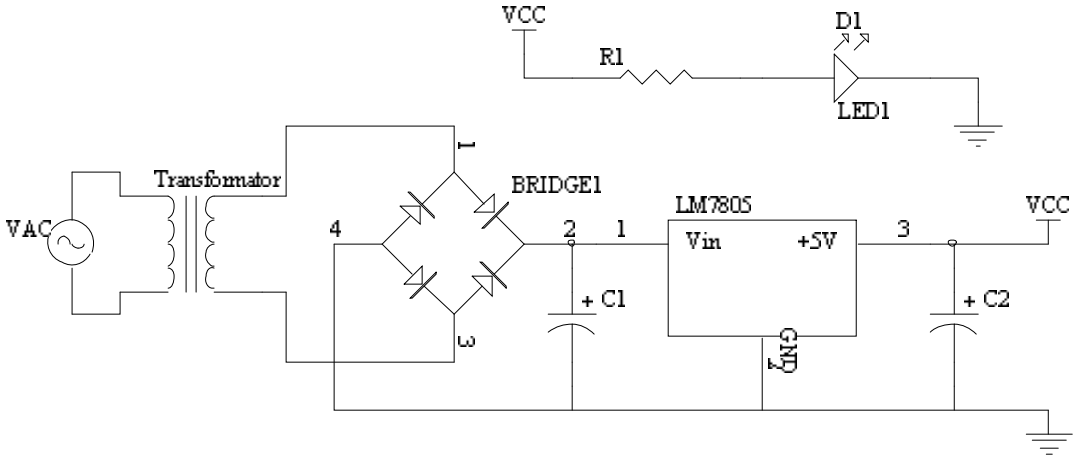
Alasan pemanfaatan *port* paralel sebagai jalur komunikasi antara alat yang dibuat dengan PC diantaranya karena kecepatan transfer data yang lebih baik daripada menggunakan *port serial* karena pengiriman data dari PC langsung 8 bit sekaligus. Dan dalam hal pengiriman data dari alat, cukup dengan memanfaatkan multiplexer 8 ke 1, maka data yang dikirim 8 bit langsung dari alat diambil secara *serial* oleh PC. Selain itu pembuatan program juga lebih sederhana dan lebih pendek karena tidak harus mengatur *baudrate* dan *paritas*-nya, serta tidak perlu memilih jenis COM yang akan digunakan.



Gambar 3.5. *Rangkaian antarmuka port paralel*

3.2.5 Catu Daya

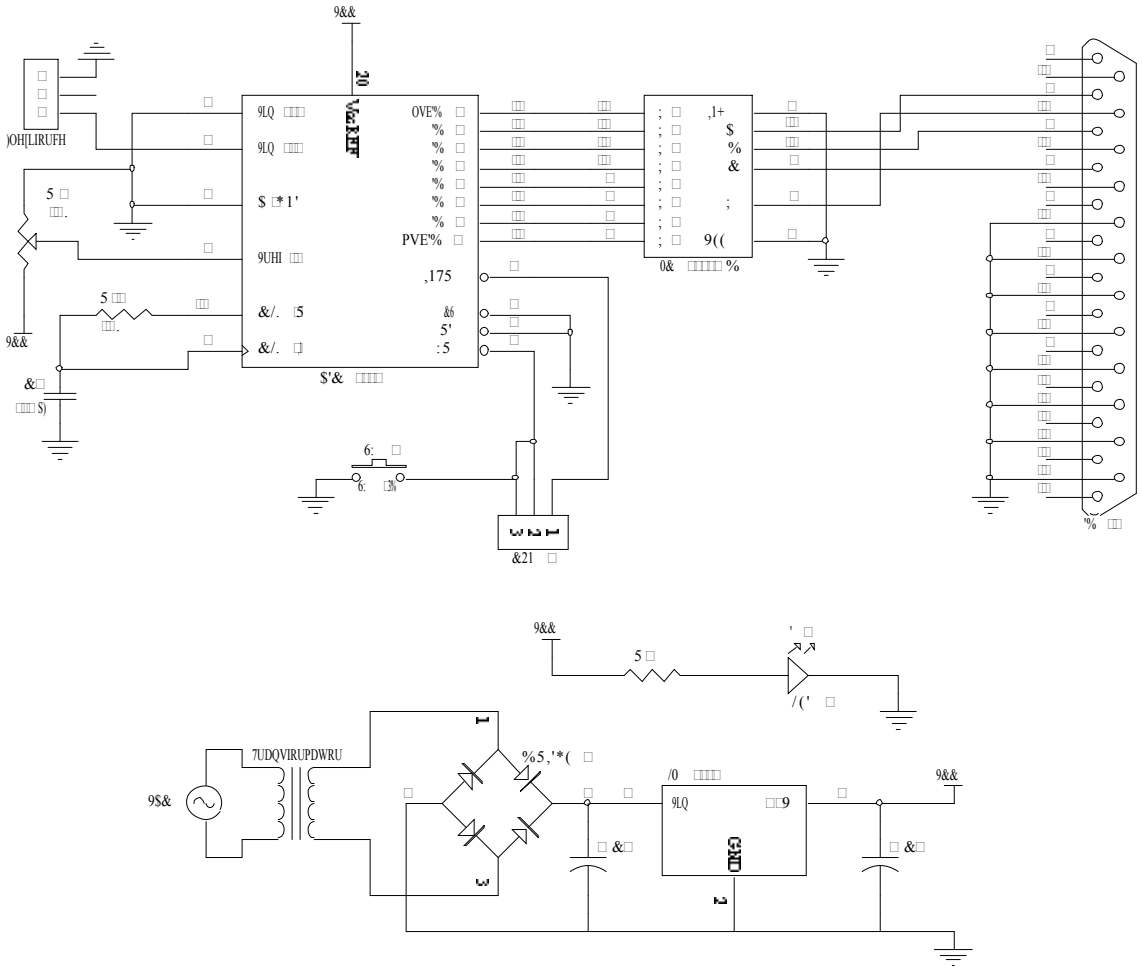
Catu daya berfungsi untuk memberikan suplai tegangan, pada alat yang dibuat terutama pada IC CMOS ADC 0804 yang bekerja pada level tegangan 5V DC. Regulator DC tersebut menggunakan rangkaian *Wheatstone* dan bekerja dalam mode penyearah gelombang penuh dari AC ke DC dengan sistem jembatan. Transformator yang digunakan adalah jenis tanpa CT (*Center Tap*). Kapasitor elektrolit digunakan untuk menghilangkan *ripple* akibat penyearahan yang belum sempurna, dengan adanya muatan dari elko maka *ripple* dapat ditutupi. Tegangan keluaran penyearah belum dapat stabil pada satu titik yang diinginkan, misalnya pada 5 Volt DC. Untuk mengatasi hal ini maka dibuatlah catu daya yang dilengkapi dengan IC regulator. IC regulator yang digunakan adalah LM 7805, di mana IC ini akan meregulasi tegangan mendekati 5 Volt DC sesuai kebutuhan rangkaian.



Gambar 3.6. *Rangkaian catu daya*

3.2.6 Skema Rangkaian Keseluruhan

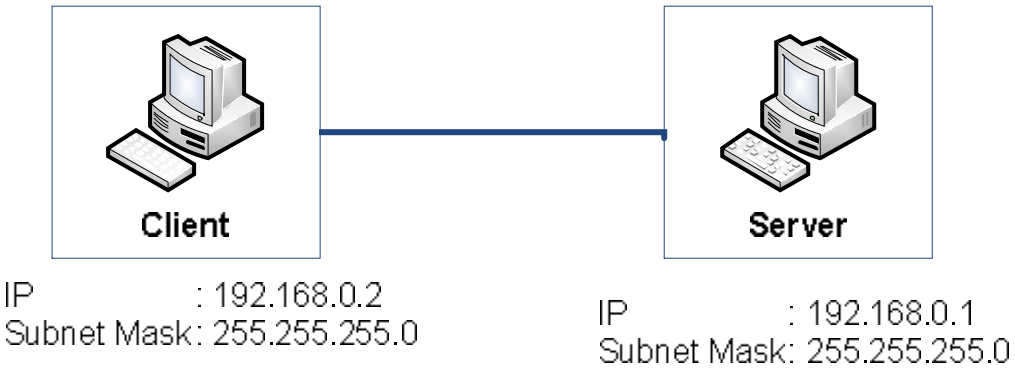
Berikut ini adalah skema rangkaian secara keseluruhan dari sistem yang dirancang.



Gambar 3.7. *Skema rangkaian keseluruhan*

3.3 Perancangan Jaringan Komputer

Perancangan jaringan komputer pada tugas akhir ini menggunakan topologi star, terdiri dari 1 komputer *server* dan 2 komputer *client*, dan 1 buah hub, hub adalah perangkat dengan banyak port yang memungkinkan beberapa titik (dalam hal ini komputer yang sudah memasang kartu jaringan).

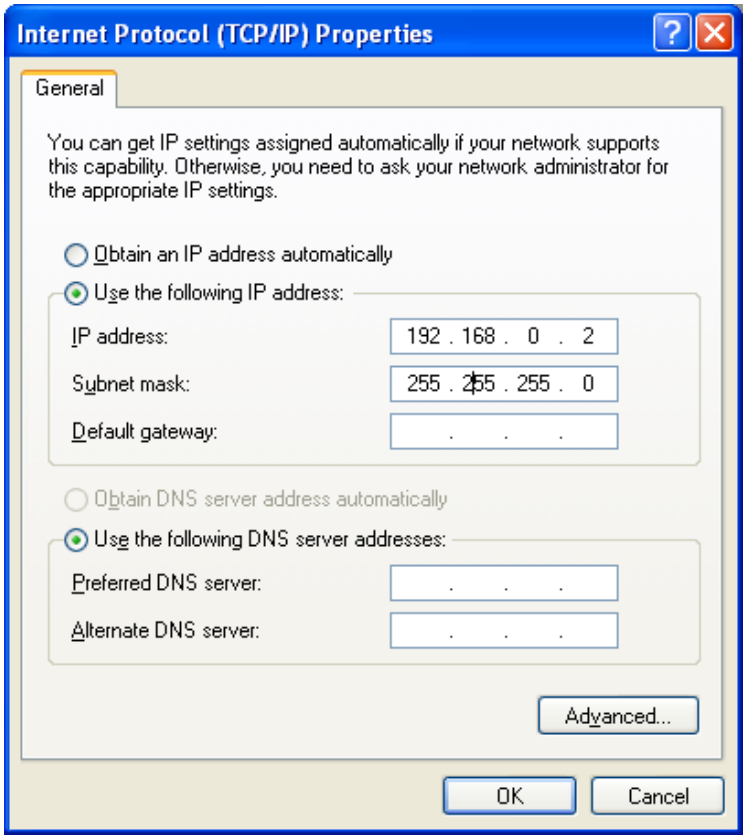


Gambar 3.8. *Perancangan jaringan komputer*

Protokol jaringan yang digunakan adalah TCP/IP *(Transmission Control Protocol/Internet Protocol)*, protokol merupakan kumpulan aturan-aturan yang mengatur dan menentukan bagaimana sebuah jaringan dapat bertukar informasi. protokol TCP/IP dirancang untuk memberikan identitas kepada setiap komputer saat terkoneksi pada suatu jaringan.

Pembagian IP address pada sebuah ethernet card tergantung dari koneksi komputer dengan komputer yang lain, format IP address yang diperhatikan untuk beberapa komputer yang terkoneksi, sebagai contoh 192.168.X.Y, dimana X adalah nomor yang harus sama untuk semua komputer termasuk *server* dan Y adalah nomor spesifik dan tidak boleh ada yang sama. Dalam mengatasi permasalahan jumlah komputer yang sangat banyak, TCP/IP dapat melakukan pengalamatan secara otomatis yang diatur oleh komputer *server* yaitu dengan menggunakan DHCP *Server (Dynamic Host Configuration Protocol)*.

Tampilan dari konfigurasi IP address pada windows XP dapat dilihat pada gambar 3.9.



Gambar 3.9. *konfigurasi jaringan windows XP*

3.4 Perancangan Perangkat Lunak

Sistem pengukuran ini direalisasikan berbasis komputer sehingga diperlukan perangkat lunak untuk mengendalikan perangkat keras, mengambil data, serta diinginkan. Program pengukuran yang telah dirancang direalisasikan dengan bahasa pemrograman Borland Delphi versi 7.0.

3.4.1 Bagian Utama Perangkat Lunak

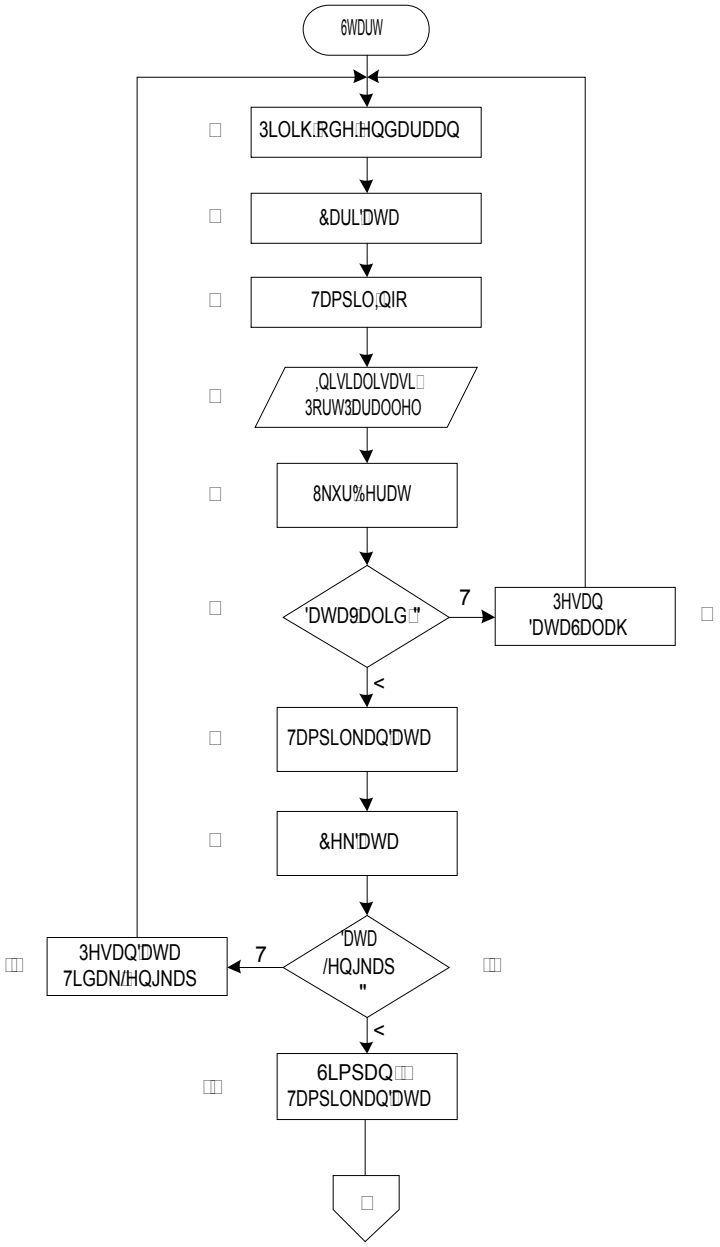
Perangkat lunak yang telah dirancang terdiri dari bagian-bagian :

1. Perangkat lunak pada bagian *client*, terdiri dari proses :
   a. Memilih kode dan menampilkan informasi kendaraan.
   b. Membaca data dari *port* paralel, melakukan konversi data dan membandingkan data tersebut dengan data pada database.
   c. Menampilkan data yang akan dikirim.
   d. Menyimpan data ke dalam database *client* dan mengaktifkan mode pengiriman data.
   e. Mengirim data ke *server* dan menunggu respon dari *server*.

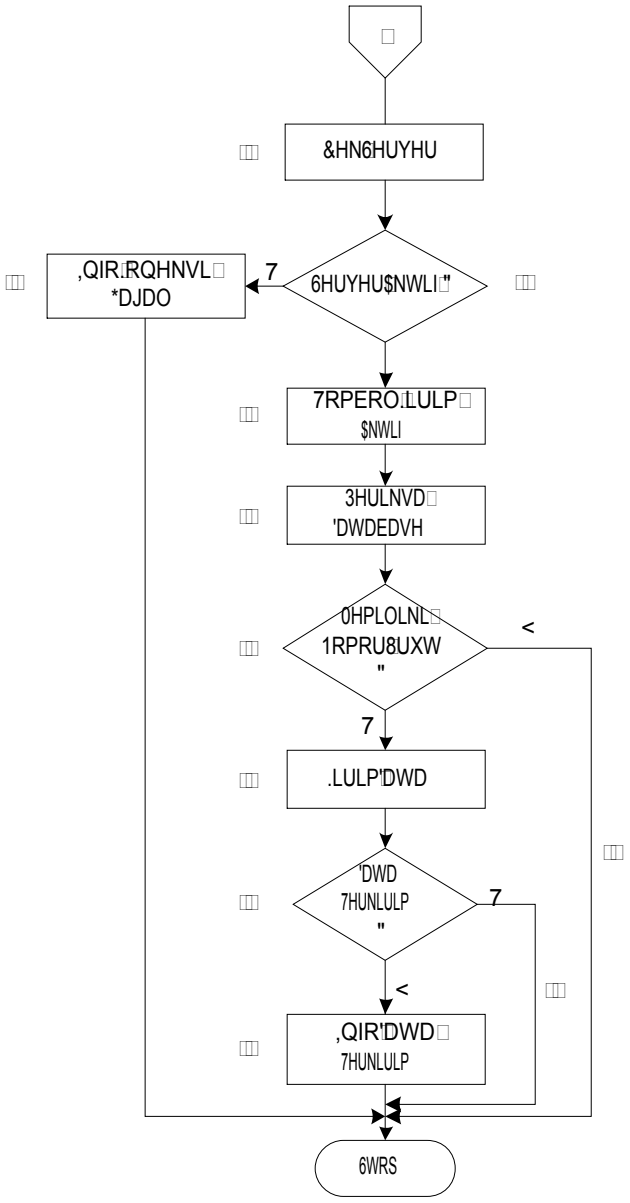
2. Perangkat lunak untuk bagian *server*, terdiri dari bagian :
    a. Membaca data yang dikirim oleh *client*.
    b. Menampilkan data yang dikirim oleh *client*.
    c. Menyimpan data yang dibaca ke dalam *database server*.
    d. Mengirim respon ke *client*.
    e. Menampilkan data dari *database* yang telah tersimpan.

3.4.2 Diagram Alir Perangkat Lunak

3.4.2.1 Diagram Alir Pada Bagian *Client*



Gambar 3.10. *Diagram alir pada bagian client*

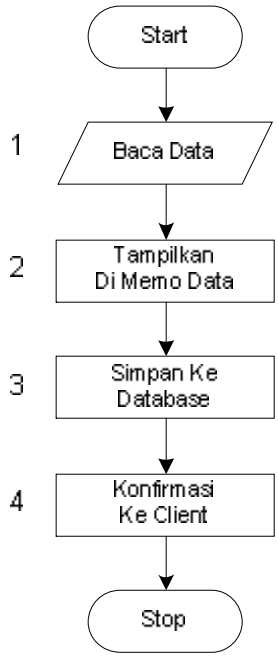


Gambar 3.10. *Lanjutan diagram alir pada bagian client*

1. Memilih kode kendaraan dari daftar.
2. Mencari data pada *database* kendaraan, Kode tersebut kemudian dibandingkan dengan daftar kode yang terdapat pada *database* kendaraan.
3. Menampilkan informasi kendaraan.
4. Inisialisasi *port* paralel dan pembacaan *PortIn* dari *port* paralel.
5. Apabila ada data yang terbaca maka data tersebut akan dikonversi menjadi satuan berat dengan rumus tertentu.
6. Data hasil konversi kemudian dibandingkan dengan data yang terdapat pada *database* sebagai proses validasi.

7. Apabila data tersebut tidak valid maka akan ditampilkan pesan kesalahan, dan akan dilakukan pengukuran ulang.
8. Apabila data tersebut valid maka data tersebut akan ditampilkan sebagai data hasil pengukuran.
9. Periksa data hasil pengukuran.
10. Periksa kelengkapan data yang dibutuhkan.
11. Apabila data tidak lengkap maka akan ditampilkan pesan data tidak lengkap dan harus dilakukan proses pengukuran ulang.
12. Apabila semua data yang disyaratkan sudah lengkap, maka data tersebut akan disimpan ke *database* dan ditampilkan pada *memo* data.
13. Melakukan koneksi terhadap *server* dengan membaca alamat IP *server.*
14. Memeriksa apakah *server* sedang aktif atau tidak aktif.
15. Apabila *server* tidak aktif maka akan ditampilkan pesan "Koneksi Gagal".
16. Apabila *server* sedang aktif maka tombol kirim akan diaktifkan dan akan ditampilkan pesan "*Connecting*", dan pada saat *client* mengaktifkan mode pengiriman, pada *server* akan ditampilkan informasi tentang alamat IP *client* dan siap untuk membaca data yang akan dikirim oleh *client* tersebut.
17. *Record* pada *database client* berisi semua data yang diisikan oleh *client*, sedangkan data yang akan dikirim hanya data yang belum mempunyai nomor data.
18. memeriksa semua *record* apakah sudah memiliki nomor data atau belum.
19. Apabila tidak ada *record* yang belum memiliki nomor data maka tidak ada data yang dikirim ke *server*.
20. Mengirim *record* yang belum memiliki nomor data.
21. Setelah mengirim *record*, *client* akan menunggu jawaban dari *server* berupa nomor data.
22. Apabila tidak ada respon dari *server* maka data belum diterima oleh *server*.
23. Apabila *server* telah mengirim respon berupa nomor data, maka nomor data tersebut akan dimasukkan ke *record* yang telah dikirim, kemudian ditampilkan informasi bahwa data telah terkirim..

3.4.2.1 Diagram Alir Pada Bagian *Server*



Gambar 3.11. *Diagram alir pada bagian server*

a. Membaca data yang dikirim oleh *client*.
b. Menampilkan data pada *memo* data.
c. Menyimpan data tersebut ke dalam *database server*
d. Mengirim respon/konfirmasi kepada *client* berupa nomor data sebagai tanda bahwa data telah diterima.