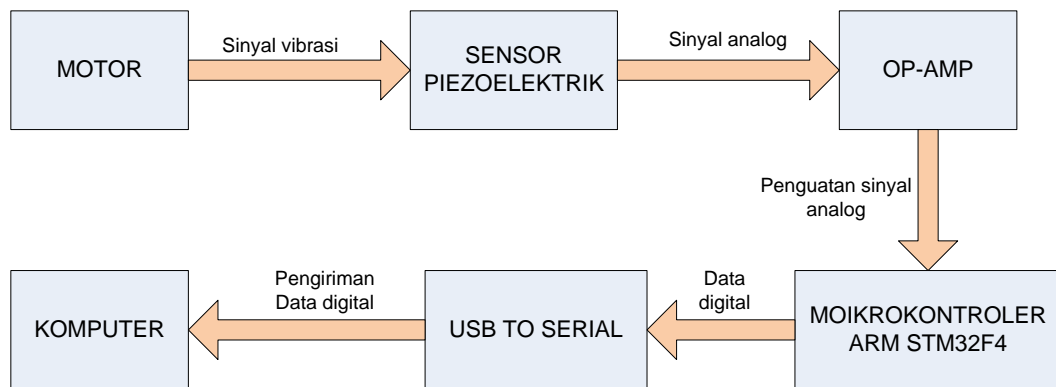


BAB III

PERENCANAAN DAN PERANCANGAN ALAT

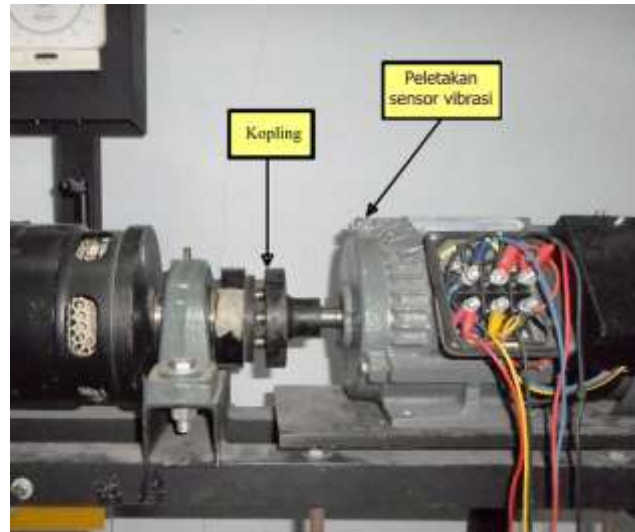
Perencanaan dan perancangan alat vibrasi meter berbasis mikrokontroler ARM STM32F4 memiliki berbagai macam sistem pendukung seperti : Motor, Sensor vibrasi piezoelektrik, Op-Amp sebagai penguat, mikrokontroler ARM STM32F4, USB to Serial dan komputer sebagai monitornya. Berikut ini gambar 3.1. adalah blok diagram perancangan dan pembuatan Alat secara keseluruhan:



Gambar 3.1. Blok Diagram Perancangan dan Pembuatan Alat

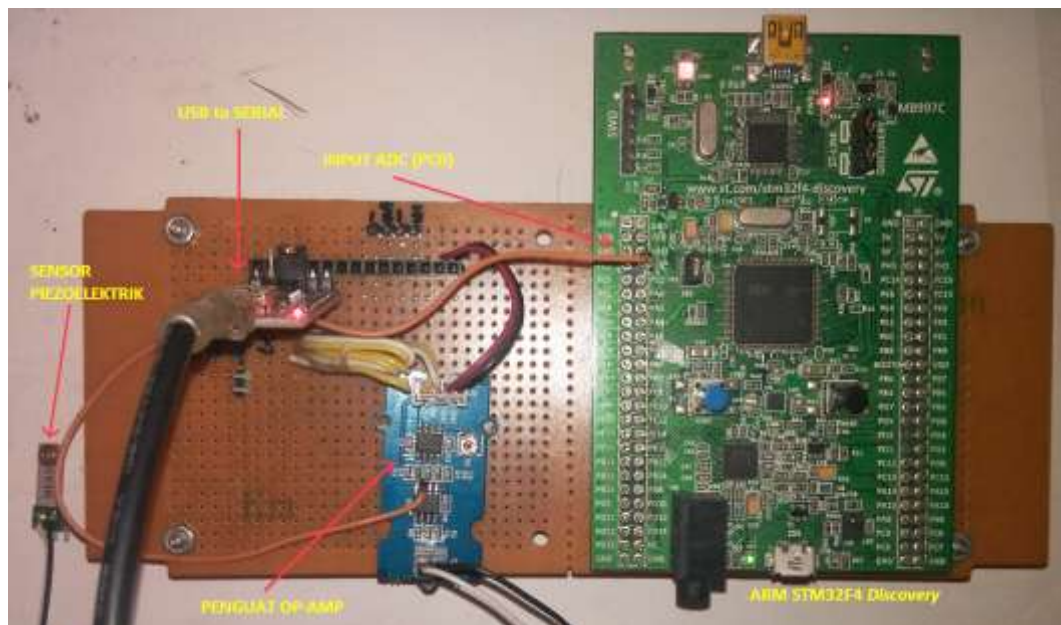
Pada gambar 3.1. vibrasi meter ini di rancang untuk mendeteksi getaran dengan meletakkan sensor pada motor (gambar 3.2.) dengan posisi sedekat mungkin dengan kopling sebagai sumber vibrasinya, kemudian *output* sensor piezoelektrik yang masih berupa sinyal analog ini dikuatkan oleh *op-amp* lalu di ubah menjadi data *digital* oleh ADC (*Analog to Digital Converter*) yang terdapat pada mikrokontroler *ARM STM32F4*, *software compiler* program yang digunakan adalah *Cocox*, Langkah selanjutnya sinyal yang sudah dalam bentuk data *digital* ini di kirim oleh *USB to Serial* ke komputer. Untuk *monitoring* getaran yang diukur oleh vibrasi meter pada sistem secara *real time*, dibuatlah sebuah *interface* pada *base*

station yaitu sebuah PC untuk memonitor getaran yang di visualisasikan sebagai plot grafik dalam domain frekuensi digunakan *software Delphi 7* sebagai *interfacenya*.



Gambar 3.2. Motor dan Generator

3.1. Perancangan *Hardware*



Gambar 3.3. *Hardware* Secara Keseluruhan

Rancangan *hardware* vibrasi meter ini terbagi atas beberapa bagian seperti terlihat pada gambar 3.3. diatas, yaitu bagian utama berupa mikrokontroler ARM STM32F4, bagian inputan berupa rangkaian piezoelektrik dengan *amplifiernya* dan bagian outputnya adalah USB to Serial yang berfungsi mengirim data digital kekomputer.

3.1.1. Sensor Vibrasi Piezoelektrik

Untuk mengetahui vibrasi yang akan di ukur digunakan sensor piezoelektrik. Sensor ini merupakan tranduser yang memiliki output tegangan analog berubah sesuai dengan besar kecilnya vibrasi yang di terima.



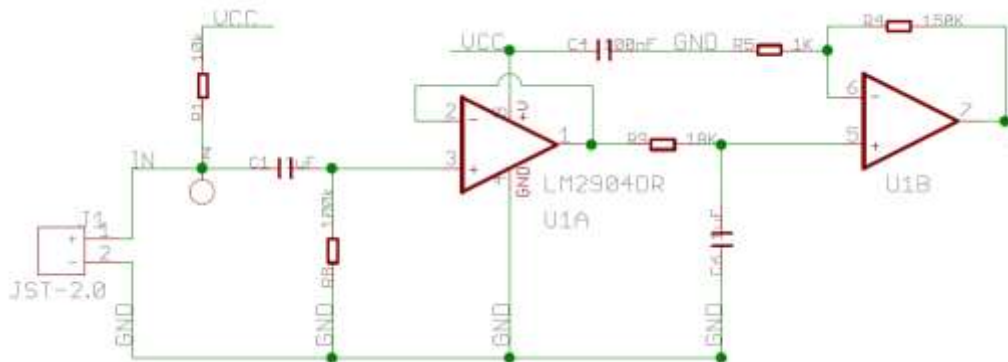
Gambar 3.4. Sensor Vibrasi Piezoelektrik

Pada gambar 3.4. diatas sensor dapat bekerja pada getaran sampai dengan 40 Hz atau 2.400 rpm dengan *high voltage sensivity* 1 V/g – 5 V/g pada arah vertical maupun horizontal. Output sensor sangat kecil sehingga perlu adanya penguatan terlebih dahulu sebelum di konversi kedalam ADC (*analog to digital converter*).

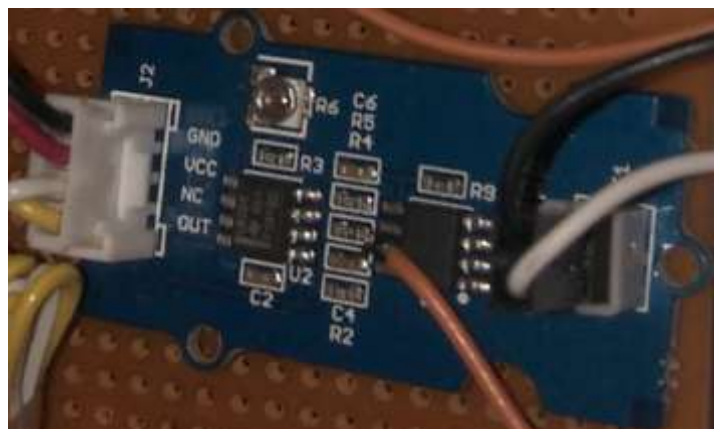
3.1.2. Penguat Op-Amp

Gambar 3.5. di bawah ini merupakan skema penguat *Inverting Amplifier* sedangkan gambar 3.6. adalah rangkaian *hardwarenya*, menggunakan ic LM2904DR dan resistor $R_f = 1K$ Ohm, $R_i = 150K$ Ohm, maka di peroleh 150 kali penguatan.

$$A = - \frac{R_f}{R_i} = - \frac{1 K}{150 K} = -150 \text{ kali}$$



Gambar 3.5. Skema penguat Op – Amp



Gambar 3.6. *Hardware* penguat Op – Amp

3.1.3. Mikrokontroler ARM STM32F4 *Discovery*

Pada gambar 3.7. dibawah ini mikrokontroler bekerja pada tegangan 5V dc yang di suplay dari USB komputer.

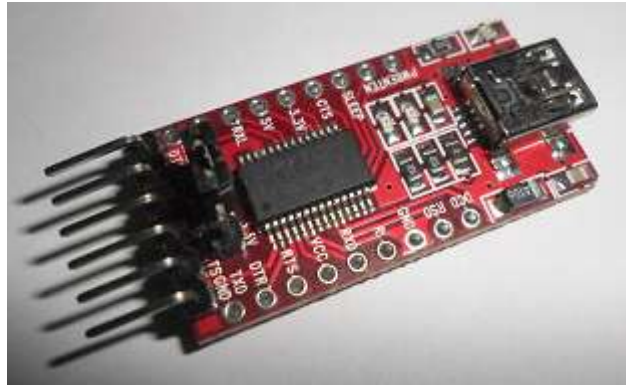


Gambar 3.7. Mikrokontroler ARM STM32F4

Mikrokontroler ARM STM32F4 dalam tugas akhir ini berfungsi untuk memproses data dari penguat op-amp yang berupa sinyal analog dan merubahnya menjadi sinyal digital melalui ADC portA0 yang ada pada mikrokontroler tersebut, kemudian mengirimnya ke komputer melalui USB to Serial dengan konfigurasi pin portD8 sebagai USART3_Tx dan portD9 sebagai USART3_Rx.

3.1.4. USB to Serial

Komunikasi data antara mikrontroler dan komputer dilakukan dengan menggunakan USB to Serial seperti yang terlihat pada gambar 3.8. di bawah ini.



Gambar 3.8. USB to Serial

3.2. Perancangan *Software*

Pada perancangan *software* vibrasi meter berbasis mikrokontroler ARM32F4 ini menggunakan bahasa C dan CooCox CoIDE sebagai *compilernya* dapat dilihat pada gambar 3.9. Bahasa C digunakan untuk membuat program ini, di butuhkan untuk mengatur kinerja *hardware* sehingga dapat berjalan sesuai dengan yang di harapkan. Sedangkan *Delphi 7* digunakan sebagai monitor dalam melakukan pengukuran.



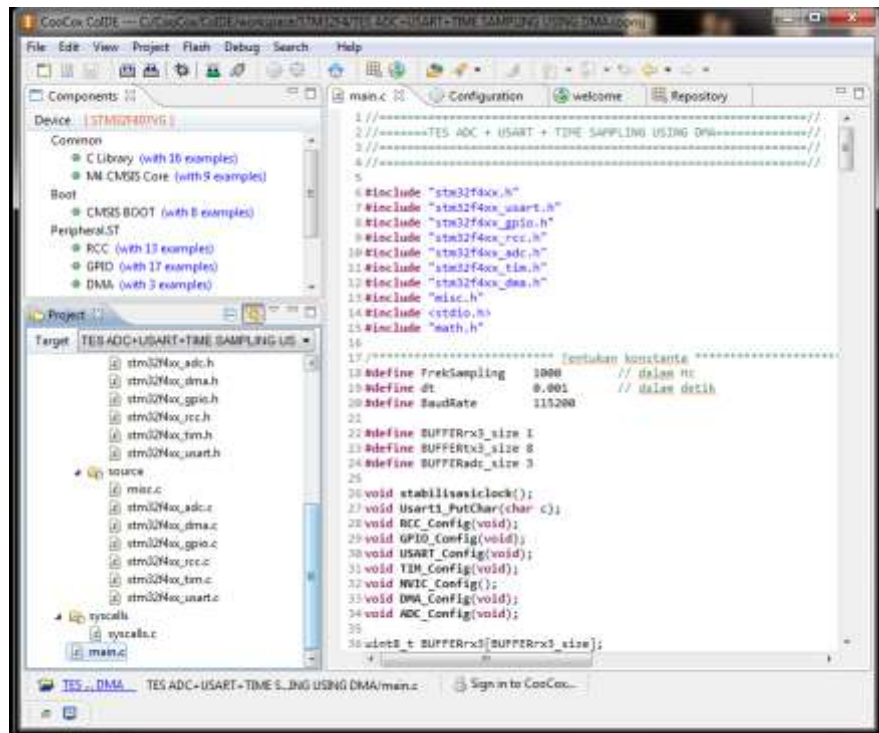
Gambar 3.9. *Compiler CooCox CoIDE*

Proses awal pemrograman adalah inialisasi mikrokontroler kemudian dilanjutkan pada proses pengambilan data dari sensor vibrasi piezoelektrik, data

tersebut di baca oleh komputer dari port ADC (*analog to digital converter*) yang ada pada mikrokontroler. Data hasil proses tadi merupakan besaran vibrasi kemudian dikirim ke komputer menggunakan USB to Serial.

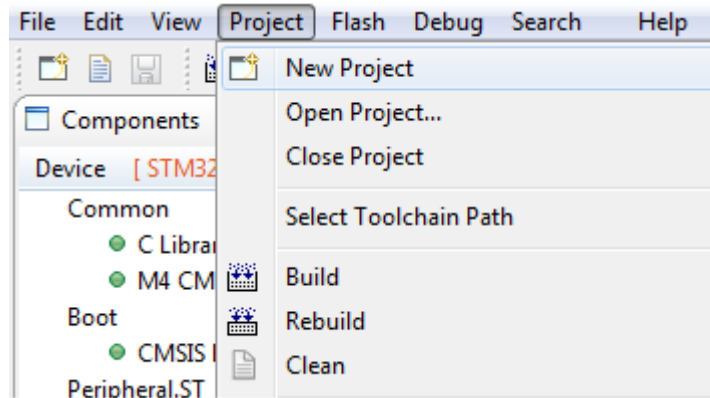
3.2.1. Pemrograman Menggunakan *Coccox CoIDE*

Coccox CoIDE merupakan *compiler* dimana program dapat di tulis menggunakan bahasa C, dengan menggunakan bahasa C diharapkan waktu disain (*developing time*) akan menjadi lebih singkat. Setelah program di tulis dan dilakukan kompilasi tidak terdapat kesalahan *error* , maka proses download dapat dilakukan kedalam mikrokontroler ARM STM32F4.



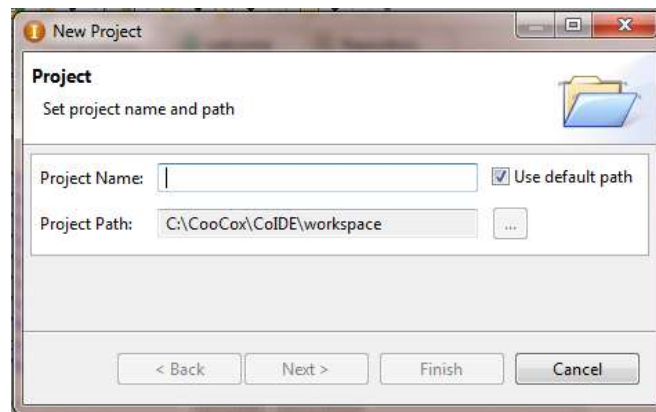
Gambar 3.10. Tampilan *Coccox CoIDE*

Untuk memulai pemrograman pada Coccox CoIDE pilih pada menu **project>new perobject** seperti gambar 3.11. dibawah ini.



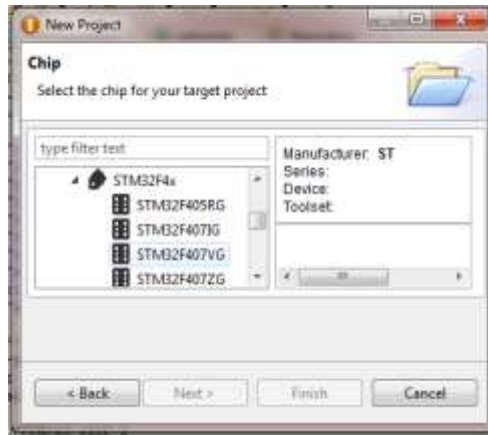
Gambar 3.11. New project

Kemudian akan muncul tampilan seperti gambar 3.12. di bawah ini, tentukan nama *project* dan simpan pada folder *workspace*, biasanya sudah tersetting secara *default*.



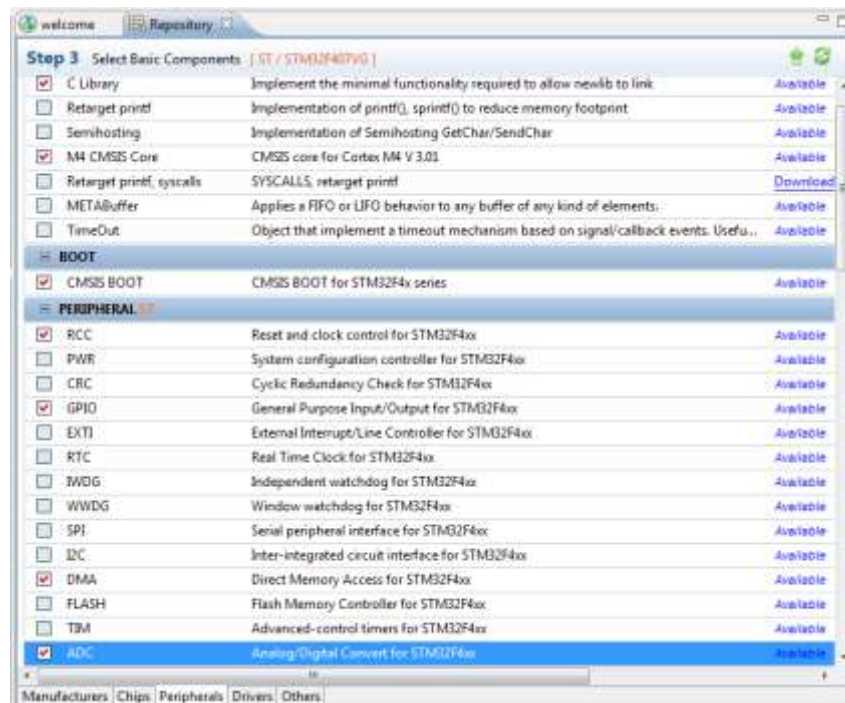
Gambar 3.12. Menentukan Nama *Project*

Setelah menentukan nama *project* kemudian pilih **chip>next**, kemudian akan tampil seperti pada gambar 3.13. di bawah ini. Cari nama *chip* yang di gunakan yaitu **STM32F407VG>finish**.



Gambar 3.13. Memilih *chip*

Menentukan *basic components* pada *repository* seperti pada gambar 3.14. di bawah ini, pilih komponen yang di butuhkan seperti *C library*, *M4 CMSIS core*, *CMSIS BOOT*, *RCC*, *GPIO*, *DMA*, *ADC*.



Gambar 3.14. *Panel Repository*

Sesudah memilih komponen yang kita pilih pada *repository* maka sekarang siap menulis program dengan cara **klik>main.c**. Untuk meng*compile* klik **build** atau **rebuild** hingga tidak ada error yang muncul, dan untuk mendownload ke *chip* klik **download code to flash** seperti terlihat pada gambar 3.15.dibawah ini.



Gambar 3.15. Panel *compile* dan *download*