

## BAB II

### DASAR TEORI

Pada bab ini dibahas tentang pengujian fatigue, teknik instrumental sensor getar, mikrokontroller arduino uno R3 dan *software arduino*.

#### 2.1 Uji Fatigue

Pengujian material merupakan salah satu cara yang digunakan untuk mengetahui sifat suatu material. Salah satunya adalah uji *fatigue* atau uji kelelahan pengujian ini bertujuan untuk mengetahui ketahanan suatu material akibat pembebanan yang dinamis atau berulang dimana beban yang diberikan dibawah kekuatan luluh suatu material. Kelelahan suatu material di pengaruhi oleh banyak faktor seperti ; ukuran specimen, bentuk specimen, proses pengerjaan akhir atau *finishing*, jenis pembebanan yang diberikan.

*Fatigue* adalah kerusakan material yang diakibatkan oleh adanya tegangan yang menumpuk atau terkumpul yang besarnya kelelahan material lebih kecil

dari tegangan tarik maksimum (*ultimate tensile strength*) ( $\sigma_u$ ) maupun tegangan luluh (*yield*) material yang diberikan beban konstan. Terdapat tiga fase dalam kelelahan fatik yaitu :

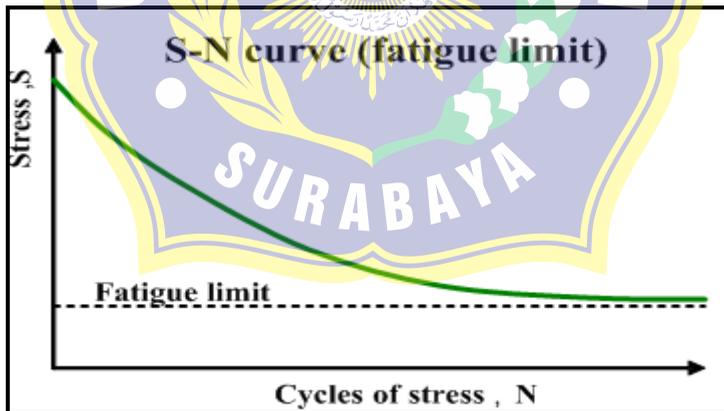
1. Awal retak yang terjadi pada uji fatik umumnya dimulai dari *crack initiation* yang terjadi di permukaan material yang lemah atau daerah dimana terjadi konsentrasi tegangan di permukaan (seperti goresan, notch, lubang-pits dll) akibat adanya pembebanan berulang.

2. Penyebaran retak *Crack initiation* ini bertambah menjadi *microcracks*. Penyebaran atau paduan *microcracks* ini akan membentuk *macrocracks* yang akan terbentuk failure.

3. Patah terjadi ketika material telah mengalami siklus tegangan dan regangan yang menghasilkan kerusakan yang permanen.

## 2.2.1 Prinsip Kerja Pengujian Fatigue

Pengujian ini dilakukan dengan memberikan beban yang relative tinggi, *failure* akan terjadi dengan siklus yang rendah. material lain diuji pada tegangan yang lebih rendah dari spesimen pertama, dan bertambahnya siklus yang menyebabkan *failure* juga bertambah. Besarnya beban tertinggi adalah 0,8 dari kekuatan luluh untuk logam besi sebesar 0,75% dari kekuatan luluh untuk logam *non* besi. Dengan variasi beban atau tegangan maka akan dihasilkan siklus yang berbeda, sehingga grafik S-Nnya dapat dibuat. (Iii, 2008)



Gambar 2.1. skema kurva s-n

## 2.1.2 Kurva uji fatigue

Penyajian data fatigue biasanya menggunakan kurva S-N yaitu pemetaan tegangan ( $S$ ) terhadap jumlah siklus sampai terjadi kegagalan ( $N$ ). Kurva S-N ini lebih dianjurkan menggunakan skala semi log seperti pada gambar 2.2. Untuk beberapa bahan teknik yang penting.



Gambar 2.2 Kurva S-N

Kurva tersebut diperoleh dari pemetaan tegangan terhadap jumlah siklus sampai terjadi kegagalan pada benda uji. Pada kurva ini menggunakan skala logaritma.

Batas ketahanan fatigue (endurance limit ) baja ditentukan pada jumlah siklus  $N > 10^7$  (Dieter,1992).

Persamaan umum kurva S-N dinyatakan oleh persamaan ( *dowling,1991*).

$$S = B + C \ln (N_f) \dots\dots\dots(1)$$

Dengan :

B dan C adalah konstanta *empiris* material

Pengujian fatigue dilakukan dengan cara memberikan stress level tertentu sehingga spesimen patah pada siklus tertentu. Dieter (1992) menyatakan untuk mendapatkan kurva S-N dibutuhkan 8-12 spesimen. Retak fatigue biasanya dimulai pada permukaan di mana lentur dan torsi menyebabkan terjadinya tegangan-tegangan yang tinggi atau di tempat-tempat. yang tidak rata menyebabkan terjadinya konsentrasi tegangan. Oleh karena itu, batas ketahanan (endurance limit) sangat tergantung pada kualitas penyelesaian permukaan (*Van Vlack,2005*).

Pengujian *fatigue* dilakukan dengan alat uji *fatigue* (*Rotary Bending Machine*). benda uji diputar dan diberi beban, maka terjadi momen lentur pada benda uji. Momen lentur ini menyebabkan adanya beban lentur di permukaan benda uji dan besarnya dihitung dengan persamaan (*international for use of ONO'S,-*).

$$\sigma = \frac{WL/2}{\pi/3d^3} K/C^2 \dots\dots\dots(2)$$

Dengan:

$\sigma$  = Tegangan lentur ( kg/cm<sup>2</sup> )

W = Beban lentur (kg)

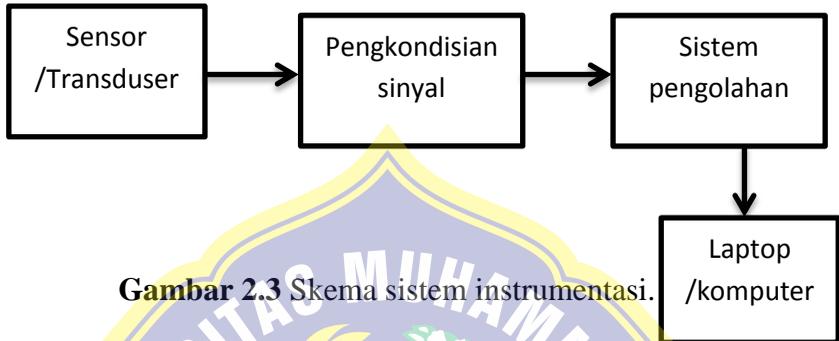
d = Diameter benda uji (cm)

## 2.2 Teknik Instrumentasi

Instrumentasi adalah peralatan dan piranti (*device*) yang digunakan untuk pengukuran dan pengendalian suatu sistem dalam rancangan ini. Sistem instrumentasi merupakan gabungan dari beberapa peralatan instrumentasi yang berupa *hardware* dan *software* tersebut.

Secara umum, sistem instrumentasi terdiri dari empat peralatan, yaitu peralatan masukan (*input*), pengkondisi sinyal (*sinyal conditioning*), sistem pengolah, dan peralatan pencatat. Peralatan masukan (*input*) merupakan peralatan pertama yang menerima data atau sinyal yang akan diukur. *Output* yang dihasilkan dari peralatan masukan berupa sinyal-sinyal listrik yang di ubah kedalam data analog berupa angka.

Pengkondisi sinyal merupakan peralatan yang berfungsi untuk memperkuat atau mempertajam sinyal-sinyal listrik yang dihasilkan dari peralatan masukan (*input*). Sistem pengolah adalah peralatan yang berfungsi mengolah hasil penguatan sinyal-sinyal listrik yang dilakukan oleh pengkondisi sinyal. Peralatan pencatat merupakan peralatan yang berfungsi mencatat dan menampilkan hasil pengolahan dari sistem pengolah. Skema sistem instrumentasi dapat dilihat pada gambar



**Gambar 2.3** Skema sistem instrumentasi.

### 2.3 Sensor Getar

Sensor Getaran merupakan salah satu sensor yang dapat mengukur getaran suatu benda yang nantinya dimana data tersebut akan diproses untuk kepentingan percobaan ataupun di gunakan untuk mengantisipasi sebuah kemungkinan adanya mara bahaya. Salah satu jenis sensor getaran yang saat ini sering di gunakan adalah accelerometer, alat ini merupakan alat yang dapat berfungsi untuk mengukur percepatan dari sebuah benda. Percepatan tersebut di ukur bukan dengan menggunakan koordinat dari percepatan tersebut, melainkan dengan mengukur percepatan berdasarkan fenomena pergerakan benda yang di hubungkan dengan perubahan massa yang terjadi di dalam alat pengukur tersebut.



**Gambar 2.4** Sensor Getar

## 2.4 Arduino

Arduino adalah alat pengendali mikro *single-board* yang bersifat *open-source*, yang dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Di Arduino memiliki *prosesor Atmel AVR* dan menggunakan software dan bahasa pemrograman tersendiri

### 2.4.1 Hardware

*Hardware* dalam *arduino* memiliki beberapa jenis, yang mempunyai kelebihan dan kekurangan dalam setiap papannya. Penggunaan jenis arduino disesuaikan dengan kebutuhan, hal ini yang akan mempengaruhi dari jenis prosesor yang digunakan. Jika semakin kompleks perancangan dan program yang dibuat, maka harus

sesuai pula jenis kontroler yang digunakan. Yang membedakan antara arduino yang satu dengan yang lainnya adalah penambahan fungsi dalam setiap boardnya dan jenis mikrokontroler yang digunakan. Di tugas akhir ini, jenis arduino yang dipakai adalah arduino uno r3 clone.

#### **2.4.2 Arduino Uno**

Menurut (Kadir, 2013), Arduino Uno adalah salah satu produk berlabel arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler ATmega328 (sebuah keping yang secara fungsional bertindak seperti sebuah komputer). Piranti ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks. Pengendalian LED hingga pengontrolan robot dapat dilakukan dengan menggunakan papan berukuran relatif kecil ini. Dengan penambahan komponen tertentu, piranti ini dapat dipakai untuk pemantauan kondisi pasien di rumah sakit atau pengendalian alat-alat yang ada di rumah. (Publishing & Guo, 2014)

**Tabel 2.1** Index Board Arduino

Mikrokontroler	ATmega328
Tegangan pengoperasian	5V
Tegangan input yang disarankan	7-12V
Batas tegangan input	6-20V
Jumlah pin I/O digital	14 ( 6 di antaranya menyediakan keluaran PWM )
Jumlah pin input analog	6
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Memori flash	32 KB ( ATmega328 ), sekitar 0,5 KB digunakan oleh bootloader
SRAM	2 KB ( ATmega328 )
EEPROM	1 KB ( ATmega328 )
Clock Speed	16 MHz

( Sumber: B. Gustom,2015 )



**Gambar 2.5** Arduino Uno

## 2.5 Software Arduino

*Software arduino* yang umum digunakan adalah *driver* dan IDE, walaupun ada beberapa *software* lain yang sangat berguna selama pengembangan arduino. *Integrated Development Environment* (IDE), suatu bahasa program khusus untuk suatu komputer supaya dapat membuat suatu rancangan atau *skets* program untuk papan *Arduino*. IDE *arduino* merupakan *software* yang sangat canggih ditulis dengan menggunakan *java*. IDE *arduino* terdiri dari :

### 1. Editor Program

Sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

### 2. Compiler

Berfungsi untuk kompilasi *sketch* tanpa unggah ke *board* bisa dipakai untuk pengecekan kesalahan kode *sintaks sketch*. Sebuah modul yang mengubah kode program menjadi kode *biner* bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*.

### 3. Uploader

Berfungsi untuk mengunggah hasil kompilasi *sketch* ke *board* target. Pesan *error* akan terlihat jika *board* belum terpasang atau alamat *port* COM belum terkonfigurasi dengan benar. Sebuah modul yang memuat kode *biner* dari komputer ke dalam *memory* didalam papan *arduino*.(Sumber: B.Gustomo, 2015 )



**Gambar 2.6** Contoh Program Arduino Ide

Kode Program *Arduino* biasa disebut *sketch* dan dibuat menggunakan bahasa pemrograman C. Program atau

*sketch* yang sudah selesai ditulis di *Arduino IDE* bisa langsung *compile* dan *upload* ke *Arduino Board*.

Secara sederhana, *sketch* dalam *Arduino* dikelompokkan menjadi 3 blok (lihat gambar di atas):

1. Header
2. Setup
3. Loop

### **2.5.1. Header**

Pada bagian ini biasanya ditulis definisi-definisi penting yang akan digunakan selanjutnya dalam program, misalnya penggunaan *library* dan pendefinisian *variable*. *Code* dalam blok ini dijalankan hanya sekali pada waktu *compile*. Di bawah ini contoh *code* untuk mendeklarasikan *variable led* (integer) dan sekaligus diisi dengan angka 13;

```
int led = 13;
```

## 2.5.2 Setup

Di sinilah awal program *Arduino* berjalan, yaitu di saat awal, atau ketika *power on Arduino board*. Biasanya di blok ini diisi penentuan apakah suatu pin digunakan sebagai *input* atau *output*, menggunakan perintah *pinMode*. *Initialisasi variable* juga bisa dilakukan di blok ini

```
// the setup routine runs once when you press reset:  
void setup() { // initialize the digital pin as an output.  
pinMode(led, OUTPUT); }
```

OUTPUT adalah suatu makro yang sudah didefinisikan *Arduino* yang berarti = 1. Jadi perintah di atas sama dengan `pinMode(led, 1);`

Suatu pin bisa difungsikan sebagai *OUTPUT* atau *INPUT*. Jika difungsikan sebagai output, dia siap mengirimkan arus listrik (maksimum 100 mA) kepada beban yang disambungkannya. Jika difungsikan sebagai *INPUT*, pin tersebut memiliki *impedance* yang tinggi dan siap menerima arus yang dikirimkan kepadanya.

### 2.5.3. Loop

Blok ini akan dieksekusi secara terus menerus. Apabila program sudah sampai akhir blok, maka akan dilanjutkan dengan mengulang eksekusi dari awal blok. Program akan berhenti apabila tombol *power Arduino* di matikan. Di sinilah fungsi utama program *Arduino* kita berada.

```
void setup()

{Serial.begin(9600); //Open the serial to set the baud rate
for 9600bps}

void loop()

{ Serial.print ("DATA, DATE, TIME,");

int val;

val=analogRead(0); //Connect the analog vibration
sensor to analog interface 0

Serial.print(" ");

Serial.println(val,DEC);//Print the analog value read via
serial port

delay(100);
```