

# Scheme untuk Edgy

Aziz Amerul Faozi

February 16, 2020

## Contents

<b>1</b>	<b>Pengantar</b>	<b>1</b>
<b>2</b>	<b>Installasi</b>	<b>1</b>
2.1	Scheme . . . . .	1
<b>3</b>	<b>Aritmatika</b>	<b>2</b>
3.1	Aritmatika Standar . . . . .	2
3.1.1	Latihan 1 . . . . .	3
3.2	Constanta . . . . .	3
3.3	Fungsi . . . . .	4
3.4	Aritmatika Anak Teknik . . . . .	4
3.5	String . . . . .	4
3.6	Conditional Statement . . . . .	5

## 1 Pengantar

LISP adalah bahasa pemrograman yang baik untuk edgy selain secara dialektika ini sangat mantap dan susah, karena lebih cenderung mirip sebagai bahasa computer daripada sebagai bahasa manusia

## 2 Installasi

### 2.1 Scheme

Petama kita mencoba untuk menggunakan scheme dalam part 1 kali ini untuk memulai scheme kita bisa menggunakan mode interpreter terlebih dahulu. Seperi contoh berikut ini :

**scheme**

maka akan muncul code seperti berikut

```
1]=> (+ 1 1)
```

code diatas merupakan contoh untuk ekseskusi interpreter dalam mode scheme. Hasilnya jelas muncul angka 2.

Sama seperti halnya dengan python dimana selain memiliki interpreter scheme juga bisa dibuild dengan menggunakan editor, salah satu editor favorit saya adalah emacs. Tapi kali ini saya tidak akan fokuskan untuk penggunaan emacs, walau nanti dalam buku ini juga mungkin akan ada screenshot dari emacs. Berikut contoh code yang disimpan dalam bentuk file berekstensi scm.

```
(+ 1 1)
```

Simpan file tersebut dalam file ujischeme.scm

### 3 Aritmatika

Pada bagian ini saya mencoba untuk menggunakan buku yang dibuat oleh kurikum MIT untuk memberikan tutorial saja, karena contohnya saya takut akan bisa membuat saya terjerat pasal plagiasi maka yang saya buat akan sedikit berbeda dari yang para dosen MIT buat.

#### 3.1 Aritmatika Standar

Pertama kita akan mencoba melakukan penjumlahan, pengurangan, perkalian dengan menggunakan scheme.

```
(+ 1 1); ini adalah contoh code untuk penjumlahan
```

bagian setelah tanda semicolon ";" merupakan komentar maka code tersebut hanya sebagai dokumentasi dari bahasa pemrograman sajah. Tidak akan dieksekusi oleh komputer. Untuk membuat anda lebih familiar dengan bahasa pemrograman scheme, anda bisa menebak hasil keluaran tersebut dengan menggunakan bahasa scheme

```
(/ 6 2) ; contoh dengan pembagian  
(* 3 8); contoh dengan perkalian  
(- 4 1); Contoh dengan pengurangan
```

Sebagai informasi untuk anda bahwa Scheme itu mirip dengan bahasa lisp maka dari itu penulisannya juga sama untuk kasus penulisan dengan negatif misalkan anda tidak bisa langsung bisa menggunakan perintah berikut

```
(-1); contoh salah mengesign angka negatif  
(- 1); contoh benar
```

hal ini dikarenakan pada variable pertama pada tanda kurung digunakan sebagai operand dari program. Berikut contoh salah ketika kita menjumlahkan angka empat dengan min 1.

```
(+ 4 -1) ; ini contoh salah  
(+ 4 (- 1)); ini contoh benar  
(+ 4 (-1)); ini contoh benar
```

dengan code tersebut anda bisa menilai sendiri lah yah, bagaimana cara memasukkan perintah aritmatika kedalam dialektika emacs. Oke gengs biar tambah edgy mari kita coba kerjakan soal MIT berikut,

### 3.1.1 Latihan 1

1. Tentukan dari keluaran code berikut.

```
11  
(+ 6 4 5)  
(- 9 1)  
(/ 6 2)  
(+ (* 2 4) (- 6 7))
```

### 3.2 Constanta

Lalu bagaimana yah kalau mau assign constanta di scheme, jadi gampang nih gengs, semisal kita punya fungsi  $E = mc^2$ , lalu nilai  $m = 2$  nilai  $c = 3 * 10^8$  maka kamu bisa menulis seperti ini gengs.

```
(define m 2) ; memasukkan nilai m  
(define c (* 3 (expt 10 8))); memasukkan nilai c  
(* m (expt c 2)); mengjadikan e = mc quadrat
```

### 3.3 Fungsi

Seperti halnya dengan bahasa pemrograman lain bahasa pemrograman Scheme juga memiliki fungsi deklarasinya mirip seperti deklarasi konstanta tapi dia memiliki variable masukkan.

```
(define (pangkat2 x) (expt x 2)) ; deklarasi fungsi
```

dan eksekusinya tinggal panggil ajah fungsinya sebagai berikut

```
(pangkat2 9) ; pemanggilan fungsi
```

### 3.4 Aritmatika Anak Teknik

Oke gengs ternyata Scheme bisa support dengan komputasi anak teknik, dia ternyata mampu untuk memanipulasi variable complex. Berikut contohnya

```
(* 1+7i 1-7i) ; maka hasilnya pasti 50
```

ini berarti memang schemers bisa terbantu sekali dengan menggunakan bahasa ini, apalagi jika schemers adalah seorang anak teknik.

### 3.5 String

Untuk bisa lucu-lucuan harus ada string di bahasa pemrograman gengs, memang sih schemers akan cukup ribet untuk memanipulasi ini di scheme. Karena memang ini dibuat anak teknik untuk ngitung yang aneh-aneh ketimbang buat curhat makanya schemers harus mengurangi curhatnya dengan banyak berhitung. Tapi bukan berarti schemers tidak bisa menggunakan string loh gengs

```
(string "mantap")
```

untuk memanggilnya dalam bentuk variable kamu bisa menggunakan konstanta atau fungsi

```
(define teks1 (string "mantap"))
(define teks2 (string "tidak mantap"))
```

### **3.6 Conditional Statement**

Jika maka pasti merupakan suatu hal yang wajib untuk para pemrogram (AKA programmer) pastinya sih kalau scheme udah support untuk conditional Statement yah, berikut adalah contoh penggunaan conditional statement dengan menggunakan scheme

```
(define x 10)  
()
```