

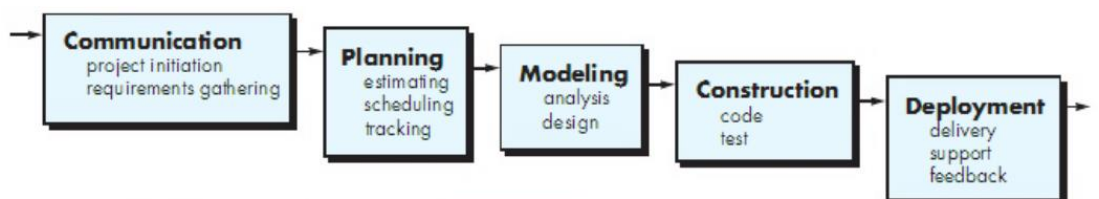
Gambar 2. 1 Model-view-controller architecture

(Sumber: Sarnath Ramnath & Brahma Dathan, 2010:240)

Model, merupakan *object passive* yang relatif, yaitu menyimpan data. Setiap object memiliki aturan tersendiri dalam *model*. *View* menerjemahkan *model* kedalam bentuk format yang ditentukan, biasanya yang cocok untuk interaksi dengan *end user*. Misalnya jika *model* menyimpan informasi tentang rekening bank, pada *view* tertentu saja yang dapat menampilkan jumlah rekening dan total saldo rekening. Sedangkan, *controller* mendapatkan input dari penggunaan dan jika diperlukan, lalu *method calls* pada *model* akan mengubah data yang tersimpan. Ketika *model* diubah, maka *view* akan respon dengan apa yang diubah lalu ditampilkan.

2.3.6 Waterfall Model

Menurut Pressman (2015:42), *waterfall model* adalah pendekatan sistematis dan sekuensial ke pengembangan *software* yang dimulai dengan spesifikasi *requirement* oleh pengguna dan berlanjut ke proses *planning*, *modeling*, *construction*, dan *deployment*, yang berpuncak dengan dukungan terus-menerus terhadap *software* yang telah diselesaikan.



Gambar 2. 15 Waterfall Model

(Sumber: Pressman, 2015:42)