**Nama : Abdillah Mufki Auzan Mubin**

**NPM : 40621100046**

**TUGAS ALGORITMA DAN PEMROGRAMAN II 5.5.1 STORING HIGH SCORES FOR A GAME**

|  |  |
| --- | --- |
| **Bahasa Python** | **Gambar** |
| class GameEntry:  """Represents one entry of a list of high scores."""  def \_\_init\_\_(self, name, score):  """Create an entry with given name and score."""  self.\_name = name  self.\_score = score  def get\_name(self):  """Return the name of the person for this entry."""  return self.\_name    def get\_score(self):  """Return the score of this entry."""  return self.\_score  def \_\_str\_\_(self):  """Return string representation of the entry."""  return '({0}, {1})'.format(self.\_name, self.\_score) # e.g., '(Bob, 98)'  class Scoreboard:  """Fixed-length sequence of high scores in nondecreasing order."""  def \_\_init\_\_(self, capacity=10):  """Initialize scoreboard with given maximum capacity.  All entries are initially None.  """  self.\_board = [None] \* capacity # reserve space for future scores  self.\_n = 0 # number of actual entries  def \_\_getitem\_\_(self, k):  """Return entry at index k."""  return self.\_board[k]  def \_\_str\_\_(self):  """Return string representation of the high score list."""  return '\n'.join(str(self.\_board[j]) for j in range(self.\_n))  def add(self, entry):  """Consider adding entry to high scores."""  score = entry.get\_score()  # Does new entry qualify as a high score?  # answer is yes if board not full or score is higher than last entry  good = self.\_n < len(self.\_board) or score > self.\_board[-1].get\_score()  if good:  if self.\_n < len(self.\_board): # no score drops from list  self.\_n += 1 # so overall number increases  # shift lower scores rightward to make room for new entry  j = self.\_n - 1  while j > 0 and self.\_board[j-1].get\_score() < score:  self.\_board[j] = self.\_board[j-1] # shift entry from j-1 to j  j -= 1 # and decrement j  self.\_board[j] = entry # when done, add new entry  if \_\_name\_\_ == '\_\_main\_\_':  board = Scoreboard(5)  for e in (  ('Asep', 800), ('bagas',1115), ('Bagus', 590), ('Raffi', 740),  ('Mufki', 510), ('Gilang', 660), ('indra', 720), ('Deni', 400),  ):  ge = GameEntry(e[0], e[1])  board.add(ge)  print('After considering {0}, scoreboard is:'.format(ge))  print(board)  print() |  |