Apriori: uses a generate-and-test approach generates candidate itemsets and tests if they are frequent
- Generation of candidate itemsets is expensive (in both space and time)
- Support counting is expensive
  Subset checking (computationally expensive)
  Multiple Database scans (I/O)

FP-Growth: allows frequent itemset discovery without candidate itemset generation. Two step approach:
- Step 1: Build a compact data structure called the FP-tree
  Built using 2 passes over the data-set.
- Step 2: Extracts frequent itemsets directly from the FP-tree
  Traversal through FP-Tree
<table>
<thead>
<tr>
<th>Name</th>
<th>NIK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jadiel Project</td>
<td>09.02.06</td>
</tr>
</tbody>
</table>

**Buku**

**Tanda**

**Perppu**

**Facultas Ilmu Komputer**