**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : I Putu Eka Wira Mahardika**

**NRP : 0511140000025**

**DOSEN WALI : Anny Yuniarti, S.Kom., M.Comp.Sc.**

**DOSEN PEMBIMBING : 1. Bagus Jati Santoso, S.Kom., Ph.D.   
 2. Henning Titi Ciptaningtyas, S.Kom., M.Kom.**

# JUDUL TUGAS AKHIR

“Desain dan Implementasi Aplikasi untuk Menjawab Permasalahan *Why-Not on Reaching k Subscribers*”

# LATAR BELAKANG

Penelitian dan pengembangan performa, kualitas, dan penggunaan sistem basis data selama dekade terakhir ini telah mendapat perhatian lebih dalam beberapa tahun terakhir[1]. Berbagai macam algoritma kueri untuk membantu pengguna mencari data dengan nilai atribut terbaik seperti *Top-K Query*, *Skyline Query*, dan *Top-K Dominating Query* mulai dikembangkan. Dengan dikembangkannya berbagai macam algoritma kueri tersebut, maka mulai muncul berbagai permasalahan baru, salah satunya adalah dimana saat pengguna merasa hasil dari suatu kueri tidak sesuai dengan ekspektasi dan mulai bertanya kenapa data tertentu tidak muncul di hasil kueri. Permasalahan tersebut juga dikenal dengan “*Why-Not Question*”. Permasalahan tersebut sudah beberapa kali diangkat untuk menjadi topik jurnal, harapannya dengan dikembangkannya algoritma yang dapat menjawab permasalahan tersebut dengan memperbaiki kueri awal agar dapat mencapai ekspektasi namun tetap memperhitungkan penalti dari perubahannya.

Berangkat dari permasalahan “*Why-Not Question*”, terdapat masalah lain yang dapat diangkat untuk dicari jalan keluarnya. Diketahui bahwa pengguna memiliki sebuah produk dengan suatu nilai tertentu, dan terdapat sebuah dataset yang berisi data preferensi untuk masing-masing konsumen. Kemudian pengguna melakukan kueri untuk mengetahui berapa banyak data konsumen yang kriterianya cocok dengan produk tersebut. Permasalahan akan muncul saat jumlah konsumen hasil kueri tersebut lebih rendah dari ekspektasi pengguna. Permasalahan seperti ini selanjutnya akan disebut dengan istilah “*Why-Not on Reaching k Subscribers*”. Perbedaan mendasar dari permasalahan “*Why-Not Question*” dengan “*Why-Not on Reaching k Subscribers*” terletak pada masukan permasalahan. Pada “*Why-Not Question*” yang menjadi masukan adalah data mana yang hilang dari hasil kueri, sedangkan pada “*Why-Not on Reaching k Subscribers*” adalah jumlah data yang diharapkan dari hasil kueri,

Berdasarkan uraian di atas maka dapat dilihat bahwa dibutuhkan sebuah algoritma untuk menjawab permasalahan “*Why-Not on Reaching k Subscribers*”. Algoritma yang akan dirancang harapannya dapat memperbaiki nilai pada setiap atribut dari produk agar hasil kueri dapat mencapai jumlah data yang diharapkan pengguna. Dalam memperbaiki data produk, nilai penalti juga akan dipertimbangkan sebagai biaya dari suatu perubahan. Dengan begitu diharapkan juga hasil perbaikan nilai atribut juga memiliki nilai penalti serendah mungkin.

# RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mengindeks seluruh data konsumen untuk mendukung solusi atas permasalahan *Why-Not on Reaching k Subscribers*?
2. Bagaimana algoritma yang tepat untuk memperbaiki nilai atribut produk agar mencapai ekspektasi pengguna aplikasi terhadap jumlah konsumen?
3. Bagaimana pembuatan mekanisme penalti yang tepat untuk mengukur kualitas atas solusi perbaikan yang diberikan?

# BATASAN MASALAH

Permasalahan yang dibahas dalam tugas akhir ini memiliki batasan antara lain:

1. Algoritma ini hanya memproses nilai atribut yang bertipe numerik.
2. Implementasi algoritma ini akan menggunakan bahasa pemrograman Phyton.
3. Dataset yang digunakan adalah data *real-life* dan sintetis.
4. Semua nilai atribut pada data akan dinormalisasi terlebih dahulu untuk mempermudah penghitungan nilai penalti.

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan pembuatan tugas akhir ini antara lain:

1. Menemukan formulasi yang tepat dalam mengindeks seluruh data konsumen untuk mendukung solusi atas permasalahan *Why-Not on Reaching k Subscribers*
2. Merancang algoritma yang tepat untuk memperbaiki nilai atribut produk agar mencapai ekspektasi pengguna aplikasi terhadap jumlah konsumen.
3. Membuat mekanisme penalti yang tepat untuk mengukur kualitas atas solusi perbaikan yang diberikan.

# MANFAAT TUGAS AKHIR

Manfaat dari pembuatan tugas akhir ini adalah:

1. Memberikan solusi bagi produsen untuk memperbaiki produknya secara efektif.
2. Memberikan kontribusi dalam pengembangan dan penelitian di bidang *data engineering*.

# TINJAUAN PUSTAKA

1. ***Query Refinement***

*Query Refinement* merupakan suatu pendekaatan untuk menghasilkan kueri baru agar dapat menjawab permasalahan pada hasil kueri seperti *Why* dan *Why-Not*, kueri baru tersebut selanjutnya akan disebut *refined query*. Hasil riset terakhir pada [2] tentang pendekatan *query-refinement* menjelaskan 2 kriteria untuk *refined query* yang baik. Pertama kueri tersebut bersifat *similar* — memiliki sedikit perubahan jika dibandingkan dengan kueri aslinya. Kriteria kedua adalah *refined query* yang baik bersifat *precise* — memiliki sedikit data tambahan pada hasil kuerinya.

Karena jumlah kemungkinan *refined query* terhadap sebuah kueri sangat banyak, jadi diperlukan suatu metrik (alat ukur) untuk membandingkan kualitas dari sebuah *refined query*. Metrik tersebut akan bermanfaat agar algoritma yang memproduksi *refined query* hanya mengembalikan sebuah kueri terbaik sebagai solusinya. Sesuai dengan 2 kriteria dari *refined query* yang baik, maka kriteria tersebut dapat dijadikan sebagai metrik atas kualitas *refined query*.

Yang pertama adalah metrik ketidakmiripan. *Refined query* yang baik seharusnya semirip mungkin dari kueri aslinya. Diketahui sebuah kueri asli *Q* dan *refined query Q’*, kita bandingkan tingkat kemiripan antara *Q* dan *Q’* dengan mengukur jarak/selisih dari perubahan yang paling minimal. Dengan demikian kueri tersebut dikatakan lebih mirip (atau lebih tidak mirip) terhadap satu sama lain jika jarak perbahannya kecil (begitu pula sebaliknya). Karena hasil dari *Q* dan *Q’* bersifat *union-compatible* (seluruh atribut dalam klausa *select* pada *Q* dan *Q’* adalah sama), kita hanya mempertimbangkan klausa *from* dan *where* sebagai operator perubahan untuk mengubah *Q* menjadi *Q’*. Empat kunci dalam operasi perubahan kueri adalah: (*O1*) ubah beberapa nilai konstan di predikat *select* pada klausa *where*, (*O2*) tambahkan sebuah predikat *select* pada klausa *where*, (*O3*) tambahkan/hilangkan predikat *join* pada klausa *where*, dan (*O4*) tambahkan/hilangkan *relation* pada klausa *from*. Perlu diketahui bahwa tidak ada perubahan operator secara eksplisit saat menghilangkan sebuah predikat *select* seperti yang dimodelkan pada *O1*. Proses menghilangkan predikat select secara efisiensi setara dengan mengubah jarak dari nilai select untuk menutupi domain dari seluruh atribut. Selanjutnya, saat *O4* digunakan untuk menghilangkan relasi *Ri* pada klausa from, seluruh predikat *select* dan *join* yang berasosiasi dengan *Ri* juga dihilangkan sebagai bagian dari operasi perubahan.

Mari nyatakan *wi* sebagai biaya untuk operasi perubahan Oi, . Dengan begitu dapat diasumsikan bahwa . Mari nyatakan *ni* sebagai total dari operasi *Oi* yang digunakan untuk merubah *Q* menjadi *Q’*, . Jarak perubahan untuk transformasi ini diketahui sebagai . Kita mengacu pada jarak perubahan minimum untuk mengubah *Q* menjadi *Q’* sebagai ukuran untuk perbedaan antara *Q* dan *Q’*, yang dapat dihitung secara efisien untuk *Q* dan *Q’*.

Yang kedua adalah metrik ketidaktepatan. *Refined query* yang baik seharusnya hanya menghasilkan data yang dihasilkan pada kueri aslinya dan dengan tambahan data yang memang diinginkan (dalam hal ini data yang hilang pada permasalahan *Why-Not*). Data-data yang tidak diinginkan untuk tampil di dalam hasil kueri seharusnya dapat diminimalisir.

*Skyline refined query* adalah *refined query* yang memiliki nilai terbaik (paling rendah) pada metrik ketidakmiripan maupun pada metrik ketidaktepatan. Dari seluruh hasil *refined query* yang mungkin akan menjadi solusi, maka *Skyline refined query* adalah solusi terbaik [2].

1. **Kueri Berbasis Preferensi**

Preferensi telah menjadi peran utama dalam berbagai disiplin ilmu, termasuk pada ilmu computer seperti kecerdasan buatan, interaksi manusia dan komputer, dan basis data. Pada basis data, representasi dari preferensi memiliki dua kategori utama yaitu preferensi kualitatif dan preferensi kuantitatif. Preferensi kualitatif dispesifikasikan menggunakan predikat biner yang dapat membandingkan antara suatu data dengan data lainnya secara relatif. Sedangkan preferensi kuantitatif diekspresikan dengan menetapkan skor pada suatu data atau pada elemen kueri yang mendeskripsikan sebuah kumpulan data, yang dimana skor tersebut mengekspresikan derajat ketertarikan (*degree of interest*). Pada preferensi kuantitatif, suatu data *a* dikatakan lebih disukai (*prefered*) apabila skor untuk data *a* lebih tinggi dari skor untuk data *b* [3].

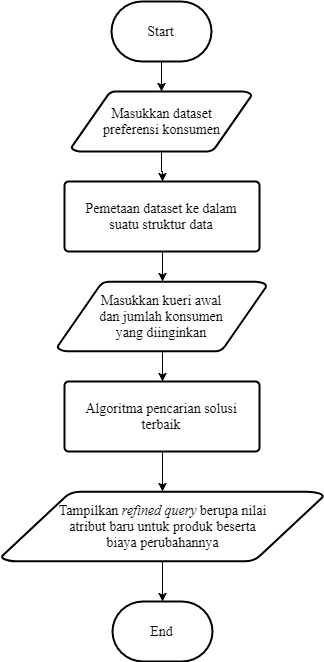
1. **Dominasi Data**

Dominasi data adalah suatu kondisi dimana sebuah objek memiliki nilai atribut lebih baik secara keseluruhan objek. Sebagai contoh, objek *a* dikatakan mendominasi objek *b* jika seluruh nilai atribut pada a tidak ada yang lebih buruk dari *b* dan setidaknya satu atribut pada *a* bernilai lebih baik daripada *b* [4].

# RINGKASAN ISI TUGAS AKHIR

Tugas Akhir ini disusun untuk menangani masalah *Why-Not on Reaching k Subscribers*. Pada dunia nyata aktor dalam algoritma ini adalah produsen dari suatu produk. Dataset untuk permasalahan ini adalah sebuah data produk, dan kumpulan data preferensi konsumen. Permasalahan yang ingin diselesaikan adalah saat hasil kueri menunjukkan bahwa jumlah konsumen yang cocok dengan kriteria produk ternyata lebih rendah dari ekspektasi produsen. Jadi tugas akhir ini akan mencari solusi terbaik untuk memperbaiki data produk agar hasil kueri memenuhi ekspektasi produsen, namun dengan biaya serendah-rendahnya.

Proses pencarian solusi diawali dengan produsen memasukkan data produk, dataset preferensi konsumen, dan jumlah data hasil kueri yang diinginkan. Sebelum masuk ke algoritma untuk menjawab *Why-Not on Reaching k Subscribers*, dataset preferensi konsumen dibuat struktur datanya terlebih dahulu untuk mengoptimalkan algoritma. Algoritma akan dijalankan setelah seluruh data preferensi konsumen dipetakan ke dalam suatu struktur data. Kombinasi nilai atribut produk dengan biaya (nilai penalti) paling rendah akan dikembalikan ke pengguna sebagai solusi. Berikut adalah diagram alur untuk menggambarkan alur aplikasi:



Gambar 1. Diagram alur aplikasi

Diketahui sebuah objek produk *P* memiliki sekumpulan nilai atribut *a* sebanyak *j*. Untuk penghitungan nilai penalti kita akan mendefinisikan sebuah set nilai tetap sebagai bobot pada masing-masing nilai atribut produk dimana diasumsikan dan . Dari beberapa definisi tersebut dapat kita formulasikan model dasar untuk menghitung nilai penalti adalah . Nilai penalti ini nantinya akan digunakan untuk mengukur biaya yang harus dibebankan pada produsen atas suatu perubahan kueri (*query refinement*).

# METODOLOGI

## Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalahuntuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

## Studi literatur

Pada studi literatur ini, akan dipelajari beberapa sejumlah referensi yang akan diperlukan untuk merancang algoritma yaitu mengenai *refined query.*

## Analisis dan desain perangkat lunak

Pada tahap ini permasalahan ke dimodelkan dalam beberapa bentuk notasi matematika untuk mempermudah proses perancangan desain dari algoritma dan struktur data.

## Implementasi perangkat lunak

Implementasi algoritma ini akan dibuat dalam bentuk program konsol yang dibangun dengan menggunakan bahasa pemrograman Python. IDE yang akan digunakan adalah PyCharm.

## Pengujian dan evaluasi

Pengujian dalam algoritma ini akan dilakukan dalam beberapa cara, antara lain:

1. Pengujian Akurasi

Pengujian ini akan berfokus pada ketepatan solusi yang diberikan dalam menjawab permasalahan *Why-Not on Reaching k Subscribers*.

1. Pengujian Waktu Eksekusi

Pengujian ini akan berfokus pada seberapa lama waktu yang dibutuhkan untuk mengeksekusi algoritma dalam menjawab permasalahan *Why-Not on Reaching k Subscribers*.

1. Pengujian Penggunaan Memori

Pengujian ini akan berfokus pada pengukuran besarnya memori yang digunakan saat aplikasi dijalankan.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tahapan** | **2017** | | | | **2018** | | | | | | | | | | | | | | | | | | | | |
| **Desember** | | | | **Januari** | | | | **Februari** | | | | **Maret** | | | | **April** | | | | **Mei** | | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |
| Perancangan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |
| Pengujian dan Evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |
| Penyusunan Buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | H. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi dan C. Yu, “Making Database Systems Usable,” *SIGMOD,* pp. 13-24, 2007. |
| [2] | Q. Tran dan C.-Y. Chan, “How to ConQueR Why-not Questions,” *Proceedings of the ACM SIGMOD International Conference on Management of Data,* pp. 15-26, 2010. |
| [3] | G. Koutrika, E. Pitoura dan K. Stefanidis, “Preference-Based Query Personalization,” *Advanced Query Processing,* vol. I, no. 36, pp. 57-81, 2013. |
| [4] | S. Borzsonyi, D. Kossmann dan K. Stocker, “The Skyline Operator,” *ACM Trans. Database System.,* vol. 25 , no. 2, p. 129–178, 2000. |
| [5] | Z. He dan E. Lo, “Answering Why-Not Questions on Top-K,” *IEEE Transactions on Knowledge and Data Engineering,* 2014. |