

BAB I PENDAHULUAN

1.1 Latar Belakang

Setiap manusia memiliki nama diri untuk menunjukkan suatu identitas yang mudah diingat oleh orang lain. Nama diri kedudukannya berbeda dengan kata biasa. Nama diri, betapapun aneh ejaannya, tetap ditulis sesuai dengan ejaan yang digunakan oleh pemiliknya. Lain halnya dengan kata biasa, terutama kata-kata asing yang diserap menjadi kata dalam bahasa Indonesia, ejaannya disesuaikan dengan ejaan yang berlaku (Adisantoso, Rambe, & Kaliana, 2003).

Pencarian nama diri dalam basis data, katalog pengarang atau buku telepon merupakan suatu yang umum dalam sistem informasi. Jika diketahui dengan pasti ejaan nama diri itu, akan secara mudah ditemukan nama tersebut dalam daftar nama. Jika tidak diketahui ejaan yang tepat atau jika terjadi kesalahan mengetik saat mencari nama dalam daftar nama maka pencarian mungkin akan mengalami kegagalan, kecuali digunakan teknik pencarian tertentu (Adisantoso, Rambe, & Kaliana, 2003).

Dalam proses pencarian nama, sering terjadi kesalahan karena nama tidak diketahui secara lengkap atau terjadi kesalahan ejaan spesifik nama tersebut (Harsowiyono, 2013). Sejauh yang diketahui untuk saat ini salah satu solusi untuk mengatasi masalah tersebut adalah menggunakan *string matching*. Penggunaan teknik ini karena pencarian teks pada umumnya mencocokkan *string* yang sama sedangkan kita mencari *string* yang mempunyai kemiripan dari segi pengucapan (*phonetic string matching*). *Phonetic string matching* digunakan dalam proses pencarian nama karena algoritma fonetik mudah melakukan koreksi pada kata yang memiliki bunyi yang sama dan waktu yang digunakan untuk koreksi ejaan relatif lebih singkat. (Kurniawan, 2010).

Ada beberapa metode pencarian nama berbasis fonetik, antara lain algoritma *soundex*, *phonix*, dan *metaphone* (Harsowiyono, 2013). Contoh penggunaan algoritma fonetik adalah pada jasa *Customer Service*. Ketika ada pelanggan menyebutkan nama, *Customer Service* dapat mengetikkan nama tanpa harus mengetahui ejaan spesifik nama tersebut (Kurniawan, 2010). Sama seperti pada pencarian nama di kontak telepon, bila kita tidak mengetahui ejaan spesifik nama tersebut maka sulit untuk menemukannya sehingga pencarian nama mungkin akan mengalami kegagalan.

Maka dari itu sangat dibutuhkan suatu aplikasi yang dapat melakukan pencocokan *string* berdasarkan kemiripan ucapan dalam pencarian nama, sehingga memudahkan kita karena dapat menampilkan nama-nama yang mirip berdasarkan ucapan yang didengar. Pemilihan algoritma *soundex* dilatarbelakangi karena adanya sejarah *soundex* yang menyebutkan bahwa algoritma *soundex* juga merupakan algoritma yang mengatasi masalah pengejaan nama orang (Purnamasari, 2012).

Kelebihan dari algoritma *soundex* selain dari sisi kecepatan adalah kesederhanaan dalam pembentukan kode fonetik yang hanya berjumlah maksimal empat karakter sehingga tidak terlalu memperhatikan huruf pada akhir kata jika sudah membentuk empat karakter kode *soundex*. Jadi, jika huruf-huruf awal dari suatu kata benar dan tepat walaupun huruf terakhir terdapat kesalahan, maka *query* tersebut akan dilihat sebagai dua *query* yang sama, karena algoritma *soundex* hanya mengambil pengodean sampai berjumlah empat karakter. Sebagai contohnya adalah *query* uji *Orthosiphon Aristatus*, nama latin dari tumbuhan kumis kucing. Jika dilakukan pencarian menggunakan keseluruhan kata, maka bisa didapatkan dokumen yang sesuai dengan yang diinginkan. Namun, jika yang digunakan hanya bagian awal kata, misalnya ‘*Orthos*’, belum tentu semua algoritma fonetik dapat mengenalinya. Lain hal dengan *soundex*, karena kesederhanaannya, yang mampu mengenali potongan *query* tersebut sehingga bisa mengembalikan dokumen yang relevan sesuai dengan yang diinginkan (Harsowiyono, 2013).

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan masalah sebagai berikut: bagaimana mengimplementasikan algoritma *Soundex* dalam mencari suatu nama pada buku telepon elektronik?

1.3 Batasan Masalah

Aspek-aspek yang akan menjadi batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Menggunakan algoritma *Soundex* dalam pencocokan *string*.
2. Nama orang yang menggunakan ejaan Bahasa Indonesia yang disempurnakan.
3. Sampel data yang digunakan adalah kontak telepon mahasiswa/i STT Musi angkatan 2010-2014 sebanyak 465 buah.
4. Aplikasi dirancang untuk mengenali nama di buku telepon elektronik.
5. Menggunakan bahasa pemrograman *Visual Basic 2010* dan *database SQL Server 2008*.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengimplementasikan algoritma *soundex* dalam mencari suatu nama pada buku telepon elektronik.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah mempermudah pengguna dalam pencarian nama pada buku telepon elektronik dengan menggunakan algoritma *soundex*.

1.6 Metodologi Penelitian

Perangkat lunak yang akan dirancang memerlukan persyaratan yang jelas, rincian proses yang jelas, dan verifikasi yang ketat untuk memasuki tahap selanjutnya. Semua persyaratan sudah ditentukan dari awal, yaitu pola *phonetic* yang bagaimana agar aplikasi dapat menampilkan pencarian *phonetic* berbahasa Indonesia. Persyaratan tersebut nantinya tidak boleh berubah saat di tengah atau

akhir pengerjaan. Oleh karena itu, model pengembangan sistem yang akan digunakan dalam perancangan adalah model sekuensial linear.

Sekuensial linear mengusulkan sebuah pendekatan kepada pengembangan perangkat lunak yang sistematis dan sekuensial. Siklus model sekuensial dapat dilihat pada Gambar 1.1. Aktivitas-aktivitas yang dilingkupi oleh model sekuensial linear adalah sebagai berikut (Pressman, 2002):

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya perangkat lunak. Pada tahap ini akan dilakukan pengumpulan data kontak telepon mahasiswa/i STT Musi angkatan 2010-2014 sebanyak 465 buah, pemodelan proses (*flowchart*) dan pemodelan data (ERD).

2. Desain

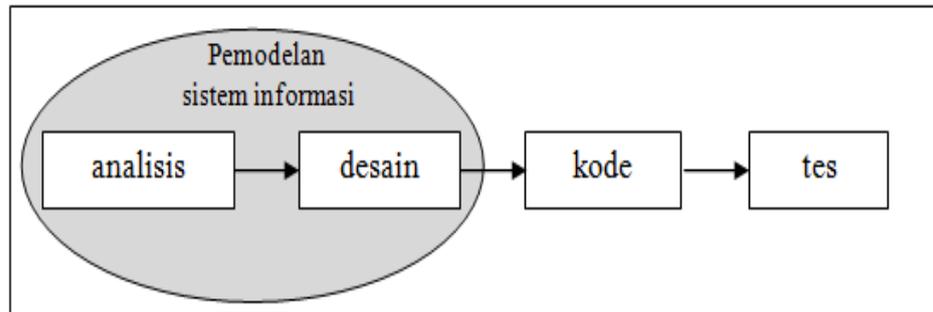
Proses desain menerjemahkan kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan sebelum dimulai pemunculan kode. Pada tahap ini akan dibuat rancangan program (*flowchart*), desain layar masukan, layar keluaran, dan rancangan tabel basis data pada aplikasi yang akan dibuat. Desain layar masukan dan layar keluaran yang akan dibuat meliputi *form* utama, *form master buku telepon*, dan *form* pencarian.

3. Kode

Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis. Berdasarkan tahap desain, maka aplikasi akan diimplementasikan ke dalam bahasa pemrograman *Visual Studio 2010* dan *SQL Server 2008* sebagai *database*.

4. Tes

Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak dan eksternal fungsional, mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil yang dibutuhkan. Pengujian akan dilakukan pada seluruh sistem yang terdapat di dalam aplikasi. Jenis pengujian yang akan digunakan adalah *black box testing* dan *white box testing*.



Gambar 1.1 Siklus Model Sekuensial Linear (sumber: Pressman, 2002)

1.7 Sistematika Penulisan

Pembahasan tugas akhir ini dibagi menjadi lima bab, masing-masing bab terdiri dari sub bab yang disusun secara sistematis. Secara garis besar, isi dari masing-masing bab adalah sebagai berikut.

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang permasalahan, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian serta sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab ini berisi penjelasan mengenai landasan teori serta referensi yang dijadikan sarana pendukung oleh penulis dalam mengimplementasikan algoritma *Soundex* untuk pencarian nama pada buku telepon elektronik.

BAB 3 ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas mengenai desain sistem menggunakan *Flowchart*, desain *database* menggunakan *Entity Relationship Diagram* (ERD), serta rancangan antarmuka sistem yang akan dibangun.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Pada bab ini dibahas mengenai proses pembuatan perangkat lunak, tampilan perangkat lunak yang dibuat, dan hasil pengujian perangkat lunak.

BAB 5 PENUTUP

Bab ini berisi kesimpulan secara umum serta saran-saran yang dapat digunakan dalam mengembangkan lebih lanjut di masa mendatang.