

APLIKASI PENCARI KATA DALAM DATABASE KAMUS BAHASA KUTAI MENGGUNAKAN METODE BINARY SEARCH

Desryanti Djo Radja¹⁾, Awang Harsa Kridalaksana²⁾

¹⁾Mahasiswa S1 Program Studi Ilmu Komputer FMIPA Universitas Mulawarman

²⁾Dosen Program Studi Ilmu Komputer FMIPA Universitas Mulawarman

Email : desryradja@gmail.com

ABSTRAK

Perkembangan bahasa Indonesia sebagai bahasa persatuan dan bahasa negara tidak dapat dipisahkan dari perkembangan bahasa-bahasa daerah. Bahasa Kutai sangat rentan terhadap bahasa Indonesia lantaran tipisnya perbedaan antara kedua bahasa tersebut. Jika para penuturnya tidak konsisten dalam melestarikan maupun pengembangannya, maka bahasa Kutai pun terancam punah. Kamus merupakan media untuk memudahkan pencarian arti atau terjemah dari suatu kata. Kamus konvensional yang berbentuk buku cenderung monoton, tidak bisa ditambah ataupun diubah, serta membutuhkan waktu yang cukup lama dalam pencarian katanya. Perkembangan teknologi sekarang ini dapat membawa efek positif diantaranya adalah dengan adanya program aplikasi kamus.

Hasil penelitian berupa pembuatan sebuah software Aplikasi pencari kata menggunakan metode binary search dalam database Kamus Bahasa Kutai dengan maksud untuk mengenalkan perbendaharaan kata dalam bahasa Kutai sehari-hari yang telah banyak digunakan. Metode pencarian pada data terurut yang paling efisien, yaitu metode pencarian bagi dua atau pencarian biner (*binary search*). Metode ini digunakan untuk kebutuhan pencarian dengan waktu yang cepat. Diharapkan software ini dapat menarik minat pembaca, khususnya para pelajar SD dan masyarakat umum dan pada akhirnya mampu memberikan kontribusi dalam upaya pelestarian budaya asli Indonesia.

Kata kunci : Bahasa Kutai, Binary Search, Kamus.

Abstract

The development of the Indonesian language as state language of unity is can not be separated from the development of regional languages. Kutai language is very susceptible to Indonesian because of the thinness of the difference between the two languages. If the speakers are not consistent in preserving and development, the Kutai language was extinct. Dictionary is a for help people to search for meaning or translation of a word. Conventional dictionaries in the form of books is monotonous, can not be added or changed, and it requires quite a long time in the search said. The development of current technology can bring positive effects in the presence of which is a dictionary application program.

This paper is the research about a word search application software use the method of binary search in the database of Kutai dictionary to introduce vocabulary Kutai has been widely used. Search methods in the most efficient sequence data is binary search. This method is used for searches with a fast time. This software is expected to attract readers, especially elementary school students and the general public and ultimately are able to contribute in the effort to preserve the original culture of Indonesia.

Key words: Kutai, Binary Search, Dictionary.

PENDAHULUAN

Perkembangan bahasa Indonesia sebagai bahasa persatuan dan bahasa negara tidak dapat dipisahkan dari perkembangan bahasa-bahasa daerah. Bahasa daerah diyakini menjadi penyumbang kos kata terbesar dalam bahasa Indonesia.

Berdasarkan penelitian yang sudah ada sebelumnya dalam Kamus Besar Bahasa Indonesia (KBBI) terbitan Pusat Bahasa, Departemen Pendidikan dan Nasional pada 2008, terdapat 1.900 tambahan kata baru dari KBBI 2002 yang bersumber dari bahasa daerah, maka usaha mengembangkan bahasa Indonesia harus berbanding lurus dengan pengembangan bahasa-bahasa daerah di nusantara, sehingga bahasa Indonesia semakin kaya dari hasil penyerapan bahasa daerah.

(<http://www.antaraneews.com>, 2011).

Bahasa Kutai adalah bahasa Melayu yang hidup dan berkembang sejalan dengan perkembangan suku Kutai. Suku Kutai adalah suku yang mendiami alur sepanjang Sungai Mahakam, dan populasinya terbesar di wilayah bekas Kabupaten Kutai dahulu.

Bahasa Kutai sangat rentan terhadap bahasa Indonesia lantaran tipisnya perbedaan antara kedua bahasa tersebut. Jika para penuturnya tidak konsisten dalam melestarikan maupun pengembangannya, maka bahasa Kutai pun terancam punah.

Pada tahun 2011 ini Dinas Pendidikan Kutai Kartanegara turut serta mensosialisasikan di 18 kecamatan yang ada di Kutai Kartanegara dimana Bahasa Kutai akan menjadi bahasa yang akan dimasukkan dalam proses belajar para siswa untuk tingkat Sekolah Dasar (SD), sebagai pelajaran yang akan masuk di Muatan Lokal (Mulok).

Tidak dapat dipungkiri bahwa teknologi komputer saat ini terus berkembang, hal ini merupakan suatu indikasi bahwa komputer adalah salah satu media yang digemari dalam mencari informasi. Hal ini juga secara tidak langsung menandakan bahwa komputer adalah suatu media yang cocok untuk penyebaran informasi dan harus dapat dimanfaatkan dengan baik tak terkecuali adalah pengembangan bahasa. Mengembangkan dan melestarikan kebudayaan Indonesia melalui komputer adalah salah satu cara yang efektif.

Terdapat metode pencarian pada data terurut yang paling efisien, yaitu metode pencarian bagi dua atau pencarian biner (*binary search*). Metode ini digunakan untuk kebutuhan pencarian dengan waktu yang cepat, untuk mencari arti kata tertentu di dalam kamus kita tidak membuka kamus itu dari halaman awal sampai halaman akhir satu per satu, namun kita mencarinya dengan cara membelah atau membagi dua buku itu. Jika kata yang dicari tidak terletak di halaman pertengahan itu, kita mencari lagi di belahan bagian kiri atau belahan bagian kanan dengan cara membagi dua belahan yang di

maksud. Begitu seterusnya sampai kata yang di cari ditemukan. Hal ini hanya bisa dilakukan jika kata – kata di dalam kamus sudah terurut. Prinsip inilah yang di anut oleh metode *binary search* dan tepat di terapkan pada aplikasi yang akan penulis buat.

Ditinjau dari kinerja pencarian, bahwa untuk kasus terburuk yaitu jika pencarian gagal menemukan x , algoritma pencarian beruntun memerlukan waktu yang sebanding dengan banyaknya data (n), sedangkan algoritma pencarian bagi dua (*binary search*) membutuhkan waktu yang sebanding dengan $2^{\log(n)}$. Karena $2^{\log(n)} < n$ untuk $n > 1$, maka jika n semakin besar waktu pencarian dengan algoritma bagi dua jauh lebih sedikit dari pada waktu pencarian dengan algoritma beruntun. Sebagai perbandingan antara algoritma pencarian beruntun dengan pencarian bagidua, misalkan untuk larik yang berukuran $n = 1024$ elemen, algoritma pencarian beruntun melakukan perbandingan elemen larik sebanyak 1024 kali, sedangkan algoritma pencarian bagidua melakukan perbandingan sebanyak $2^{\log(1024)} = 10$ kali. Karena itulah, algoritma pencarian bagi dua lebih baik untuk mencari data pada sekumpulan nilai yang sudah terurut ketimbang algoritma pencarian beruntun. (Munir, 2005)

Berdasarkan fakta – fakta yang ada dalam penulisan ini peneliti memilih salah satu alat bantu dan pengingat untuk memperluas wacana atau wawasan kosa kata dalam bahasa kutai yaitu kamus dengan menggunakan metode *binary search* atau pencarian bagidua.

TINJAUAN PUSTAKA

Definisi Kamus

Menurut Kamus Besar Bahasa Indonesia Kamus adalah buku acuan yg memuat kata dan ungkapan, biasanya disusun menurut abjad berikut keterangan tentang makna, pemakaian, atau terjemahannya; buku yg memuat kumpulan istilah atau nama yg disusun menurut abjad beserta penjelasan tentang makna dan pemakaiannya. Berdasarkan pada penggunaan bahasa kamus bisa dibagi menjadi beberapa jenis, yaitu:

1. Kamus Ekabahasa

Kamus ini hanya menggunakan satu bahasa. Kata-kata (entri) yang dijelaskan dan penjelasannya adalah terdiri daripada bahasa yang sama. Kamus ini mempunyai perbedaan yang jelas dengan kamus dwibahasa karena penyusunan dibuat berdasarkan pembuktian data korpus. Ini bermaksud definisi makna ke atas kata-kata adalah berdasarkan makna yang diberikan dalam contoh kalimat yang mengandung kata-kata berhubungan.

2. Kamus Dwibahasa

Kamus ini menggunakan dua bahasa, yakni kata masukan daripada bahasa yang dikamuskan diberi padanan atau pemerian takrifnya dengan menggunakan bahasa yang lain. Contohnya: *Kamus*

Inggris-Indonesia, Kamus Dwibahasa Oxford Fajar (Inggris-Melayu;Melayu-Inggris).

3. Kamus Aneka Bahasa

Kamus ini sekurang-kurangnya menggunakan tiga bahasa atau lebih. Misalnya, kata Bahasa Melayu Bahasa Inggris dan Bahasa Mandarin secara serentak. Contoh bagi kamus aneka bahasa ialah *Kamus Melayu-Cina-Inggris Pelangi* susunan Yuen Boon Chan pada tahun 2004

Bahasa Kutai

Sebagai salah satu bahasa Nusantara, bahasa kutai lebih dekat dengan bahasa Melayu, yang sudah sejak lama dipakai sebagai “lingua franca” di wilayah Nusantara. Kerajaan Kutai yaitu kerajaan tertua di Indonesia yang terletak di Provinsi Kalimantan Timur sudah lama mengenal bahasa Melayu. (Suryadikara, 1979).

Dalam cerita rakyat “Si Melengkar Pahlawan Kutai” (Legenda dan Cerita Rakyat Kutai, Seri III, 1974) diceritakan bahwa terjadi penyerbuan kapal – kapal pasukan Melayu ke kerajaan Kutai.

Dengan adanya orang – orang Melayu itu, tidak mustahil bahasa Melayu berpengaruh besar dalam kehidupan masyarakat Kutai sehari – hari. Oleh sebab itu, bahasa Melayu kemudian dipakai di kerajaan Kutai sebagai bahasa kerajaan. Pertumbuhan dan perkembangan bahasa Melayu di kerajaan Kutai ini lambat laun memberikan corak dan ragam tersendiri, sehingga menjadi apa yang kita kenal dengan bahasa Kutai sekarang. (Suryadikara, 1979).

Pencarian Biner (Binary Search)

Pencarian adalah salah satu hal penting dalam banyak kasus pemrograman. Terdapat bermacam – macam algoritma pencarian yang telah dikembangkan dengan ide dasar yang berbeda, salah satunya yaitu pencarian biner.

Selain algoritma yang memanfaatkan keterurutan elemen – elemen array, terdapat juga algoritma pencarian lain yang disebut pencarian biner (*binary search*). Algoritma ini juga disebut *dicotomic search*. Ide dasarnya adalah membandingkan harga yang dicari, x , dengan elemen tengah array. Jika x lebih besar dari elemen tengah array, karena elemen – elemen array terurut membesar, maka pencarian dilakukan pada setengah bagian yang nilainya lebih besar dari x sampai elemen terakhir, dengan teknik yang sama. Supaya lebih jelas, berikut diberikan gambaran kasus.

Diketahui sebuah tabel berisi harga integer $\text{TabInt}[1..n]$, yang telah terisi $.$, dan terurut membesar. Tuliskan program dengan bantuan fungsi yang jika diberikan sebuah x bernilai integer, maka akan dicari apakah harga x ada dalam TabInt secara dikotomik, dengan aturan berikut :

Bandingkan x dengan harga elemen tengah.

1. Jika sama berarti x ditemukan dalam tabel.

2. Jika lebih kecil, pencarian dilakukan pada elemen bagian bawah dengan cara yang sama.
3. Jika x lebih besar, pencarian dilakukan pada elemen bagian atas dengan cara yang sama.

Pencarian akan menghasilkan sebuah boolean ditemukan yang berharga true jika x ditemukan, dan false jika x tidak ditemukan serta indeks tempat x ditemukan. Pencarian dihentikan setelah harga pertama ditemukan.

Sebagai contoh, misalkan $n=8$ dan TabInt berisi (1, 3, 5, 8, 12, 90, 312, 500), sedang nilai yang dicari adalah $x = 5$, maka pemeriksaan akan dilakukan terhadap (8, 3, 5) dan ditemukan ber nilai true serta indeks yang dikirimkan adalah 3 karena nilai $x = 5$ ditemukan pada indeks ke- 3. Jika $n=7$, dan TabInt berisi (12, 30, 50, 83, 99, 123, 456), sedang nilai yang dicari $x=100$, maka pemeriksaan akan dilakukan terhadap (83, 99, 123) dan ditemukan bernilai *false* dan indeks yang dikirimkan adalah 0. (Wahid, 2004).

Proses yang terjadi pada pencarian dengan metode ini adalah sebagai berikut: Misalkan indeks kiri adalah i dan indeks kanan adalah j . Pada mulanya kita inialisasi i dengan 1 dan j dengan n . (Munir, 2005).

Langkah 1 : Bagi dua elemen larik pada elemen tengah. Elemen tengah adalah elemen dengan indeks $k = (i + j) \text{ div } 2$.

(Elemen tengah, $L[k]$, membagi larik menjadi dua bagian, yaitu bagian kiri $L[i..j]$ dan bagian kanan $L[k+1..j]$)

Langkah 2 : Periksa apakah $L[k] = x$. Jika $L[k] = x$, pencarian selesai sebab x

sudah ditemukan. Tetapi jika $L[k]$

$\neq x$, harus ditentukan apakah

pencarian akan dilakukan di larik

bagian kiri atau di bagian kanan.

Jika $L[k] < x$, maka pencarian

dilakukan lagi pada larik bagian kiri.

Sebaliknya, jika $L[k] > x$, pencarian dilakukan lagi pada larik bagian kanan.

Langkah 3 : Ulangi langkah 1 hingga x ditemukan atau $i > j$ (yaitu, ukuran larik sudah nol). (Munir, 2005)

HASIL DAN PEMBAHASAN

Aplikasi Pencari Kata dalam Kamus Bahasa Kutai dengan menggunakan metode pencarian *Binary search* ini merupakan aplikasi yang berbasis windows atau pemrograman visual. Dalam pemrograman visual, kita meningkatkan kecepatan dan kinerja aplikasi dengan mengoptimasi penggunaan memori, manajemen proses, dan pengaturan Input - Output. Pada aplikasi visual, aplikasi dibangun dengan menggunakan tool tertentu, kemudian dikompilasi dan hasilnya dapat langsung digunakan pada komputer tanpa harus

terhubung internet. Adapun beberapa fungsi yang dapat dihasilkan dari aplikasi ini adalah sebagai berikut :

1. Pengguna dapat mencari kata tertentu pada kamus dengan cara mengetikkan kata atau istilah yang ingin dicari, kemudian program dengan cepat akan mencantumkan pengertiannya.
2. Pengguna dapat menambahkan sendiri istilah beserta pengertian yang diinginkan bila kosakata tersebut belum tercantum pada database sistem.
3. Pengguna dapat merubah atau memodifikasi isi dari data – data pada aplikasi ini.

Gambaran umum dari sistem yang telah penulis rancang adalah : menu utama terdiri atas form pencarian istilah, manajemen data, menampilkan seluruh database serta petunjuk penggunaan aplikasi.

Pada database hanya diperlukan file dengan ekstensi txt yang akan menyimpan daftar kata yang diperlukan program yaitu perbendaharaan kata – kata dalam bahasa Kutai dan bahasa Indonesia dengan menggunakan notepad sehingga tidak terdapat penggunaan aplikasi DBMS (Database Management System) pada perancangan database sistem. Kemudian untuk bagian perancangan *Interface* Kamus Kutai dan menampilkan data digunakan bahasa pemrograman Visual Basic 6.0.

Pengaksesan kamus ini sangat mudah dan tidak perlu menggunakan koneksi internet, perbendaharaan kata – kata dalam bahasa Kutai dan bahasa Indonesia dapat langsung di cari pada halaman pencarian dengan menginputkan kata yang akan dicari dan menekan tombol find untuk menemukannya, kata yang dicari akan diproses dengan menggunakan logika algoritma binary search yang membagi data atas dua bagian sehingga untuk mencari arti kata tertentu dalam kamus menjadi lebih efisien dan cepat. Aplikasi pencari kata ini tidak bersifat case sensitive artinya kata yang diinput pada kolom pencarian baik huruf besar, huruf kecil atau huruf kecil besar tidak akan menjadi masalah, sistem secara otomatis akan mengkonversi kosa kata yang diinputkan ke lowercase atau huruf kecil dan mencari kata yang dimaksudkan.

Pencarian binary search dilakukan untuk :

1. Memperkecil jumlah operasi perbandingan yang harus dilakukan antara data yang dicari dengan data yang ada di dalam tabel, khususnya untuk jumlah data yang sangat besar ukurannya.
2. Prinsip dasarnya adalah melakukan proses pembagian ruang pencarian secara berulang-ulang sampai data ditemukan atau sampai ruang pencarian tidak dapat dibagi lagi, berarti ada kemungkinan data tidak ditemukan.

3. Syarat utama untuk pencarian biner adalah data di dalam tabel harus sudah terurut, misalkan terurut menaik.

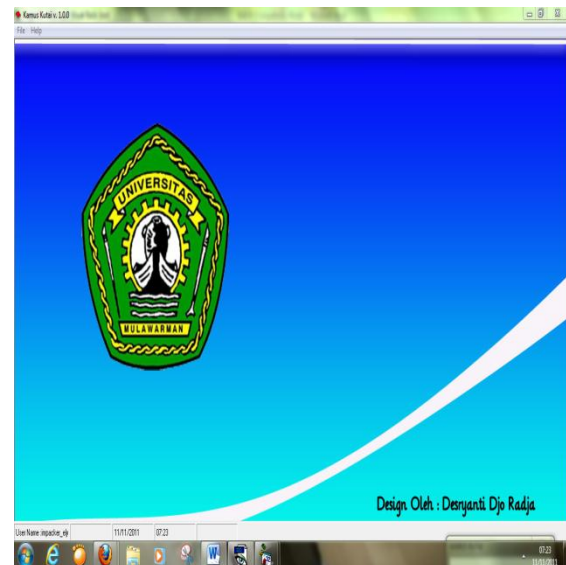
Implementasi sistem

Implementasi antarmuka sistem adalah hasil penerapan dari rancangan menu yang telah dibuat sebelumnya. Aplikasi pencari kata dalam kamus Bahasa Kutai ini diberi nama Kamus Kutai yang terdiri dari tiga form yang masing – masing memiliki fungsi tersendiri diantaranya :

1. Form MDI

Pada MDI Form terdiri dari dua MDI Child Form yaitu form untuk pencarian kata dan form untuk penambahan kata baru atau update database kamus Kutai tersebut. MDI Form inilah yang akan tampil pertama kali pada saat program *di load* atau *star up*.

MDI form juga memiliki dua menu utama yaitu menu file dan help, menu file memiliki sub menu dropdown yaitu pencarian kata, tambah kata, dan exit, sedangkan menu help terdiri dari sub menu cara menggunakan kamus dan about. Gambar 4.5 adalah tampilan pertama program saat dijalankan, terdapat logo Universitas Mulawarman dan nama pembuatnya.

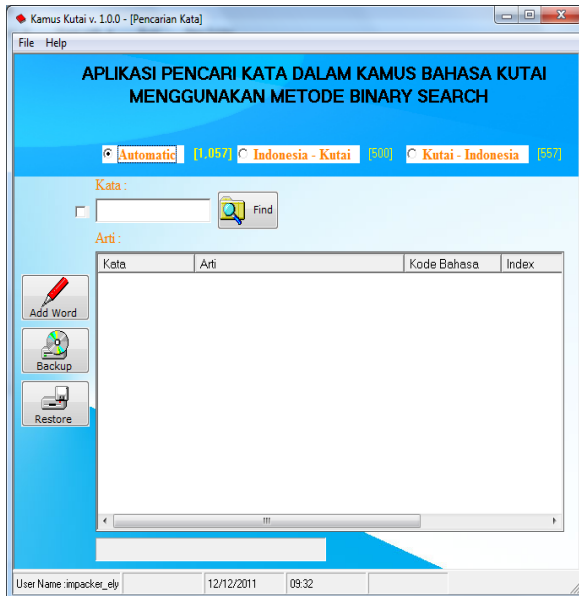


Gambar 4 Tampilan Awal Program

2. Halaman Pencarian Kata

Halaman pencarian kata yang berada pada Form 1 secara umum berfungsi untuk membantu *user* menemukan kata yang akan dicari dengan menginputkan kata yang diinginkan pada textbox kata dan dengan metode *binary search* database kamus akan diperiksa dan aplikasi ini dengan cepat akan menginformasikan ada tidaknya kata yang dicari.

Halaman pencarian kata ini memiliki tiga option button diantaranya tombol otomatis, tombol Indonesia – Kutai dan tombol Kutai – Indonesia dan terdapat tombol add word, backup, restore, find, listview untuk menampilkan kata yang di cari oleh *user* serta kontrol array lainnya untuk mempercantik tampilan *interface* seperti terlihat pada gambar 5



Gambar 5. Tampilan Halaman Pencarian Kata

Berikut adalah penjelasan dari atribut – atribut yang terdapat pada form pencarian kata seperti terlihat pada gambar 5

a. Button Automatic

Pada *option button automatic*, *user* dipersilahkan untuk memasukkan kata baik didalam bahasa Kutai ataupun bahasa Indonesia yang ingin diterjemahkan kedalam bahasa Kutai ataupun bahasa Indonesia, jadi akan membantu *user* mencari kata baik bahasa Indonesia maupun Bahasa Kutai secara otomatis dan inputan pada `txtKatadicari.text` bisa berupa kata bahasa Indonesia dan Kutai. Kata tersebut akan dicari di dalam database dan akan ditampilkan arti dari kata yang dicari. Apabila kata tersebut tidak terdapat dalam database, maka akan diberikan pemberitahuan bahwa kata tersebut tidak terdapat dalam database sistem, dan *user* diberikan pilihan untuk kembali mencari lagi dalam menu yang sama atau kembali ke menu lainnya.

b. Button Indonesia – Kutai

Pada *option Indonesia - Kutai* ini, *user* dipersilahkan untuk memasukkan kata baik didalam bahasa Indonesia yang ingin diterjemahkan kedalam bahasa Kutai pada kolom kata. Kata tersebut akan dicari di dalam database dan akan ditampilkan artinya kedalam bahasa Kutai. Apabila kata tersebut tidak terdapat dalam database, maka akan diberikan pemberitahuan bahwa kata tersebut tidak terdapat dalam database sistem, dan *user* diberikan pilihan untuk kembali mencari lagi dalam menu yang sama atau bisa memilih menu lainnya.

c. Button Kutai – Indonesia

Pada *option button Kutai - Indonesia* ini, *user* dipersilahkan untuk memasukkan kata didalam bahasa Kutai yang ingin diterjemahkan kedalam bahasa Indonesia pada kolom kata. Kata tersebut akan dicari di dalam database dan akan ditampilkan artinya dalam bahasa Indonesia. Apabila kata tersebut tidak terdapat dalam database, maka akan diberikan pemberitahuan bahwa kata tersebut tidak terdapat dalam database sistem, dan *user* diberikan pilihan untuk kembali mencari lagi dalam menu yang sama atau beralih ke menu lainnya.

Jadi pilihan kedua yaitu Indonesia – Kutai akan membantu *user* untuk mencari secara spesifik terjemahan ke Bahasa Kutai dengan inputan hanya bahasa Indonesia, sedangkan tombol pilihan Kutai – Indonesia membantu *user* secara spesifik mencari terjemahan ke Bahasa Indonesia dengan inputannya hanya bahasa Kutai dan

tombol *automatic* memberikan kebebasan bagi *user* untuk menginputkan kata mana yang akan dicari.

d. Add Word

Tombol add word pada form pencarian kata berfungsi sebagai *short cut* atau jalan pintas untuk berpindah ke halaman tambah kata, jadi ketika *user* memilih tombol ini, *user* akan dibawa ke halaman penambahan kata yang memungkinkan *user* untuk menambah kata, menyimpan, menghapus dan mengedit database yang ada pada sistem.

e. Backup

Tombol backup berguna untuk menyalin semua file sebelumnya, karena sewaktu-waktu file yang dipaka bisa saja rusak atau hilang. Tombol backup ini akan mengandakan file atau membuat cadangan file atau *mengcopy* file ke folder backup, fungsinya untuk mengamankan file tersebut, jika yang asli kena virus/rusak maka file backup/copy-annya masih ada dengan begitu tidak perlu memasukkan data baru untuk ditulis, karena data sebelumnya sudah tersimpan secara permanen.

Pada saat *user* memilih tombol backup maka sistem akan memberikan pesan konfirmasi apakah database mau di backup. Jika *user* memilih "No" maka proses backup dibatalkan.

Jika *user* memilih "Yes" maka database akan di backup dan sistem akan menginformasikan bahwa database sudah di backup.

f. Restore

Tombol restore berfungsi untuk mengembalikan database ke kondisi terakhir sebelum ada pembaharuan database.

g. Find

kata yang di input akan di convert menjadi *lowcase* atau huruf kecil semua, jadi sistem ini tidak bersifat *case sensitive*.

Setelah menginputkan kata yang dicari selanjutnya *user* menekan tombol Find atau Enter pada

keyboard, dengan demikian sistem akan memeriksa database dengan menjalankan method cari kata yaitu algoritma *binary search*.

h. Progress Bar

Progress bar ini menunjukkan presentase pergerakan atau loading pada saat pencarian kata.

i. Check

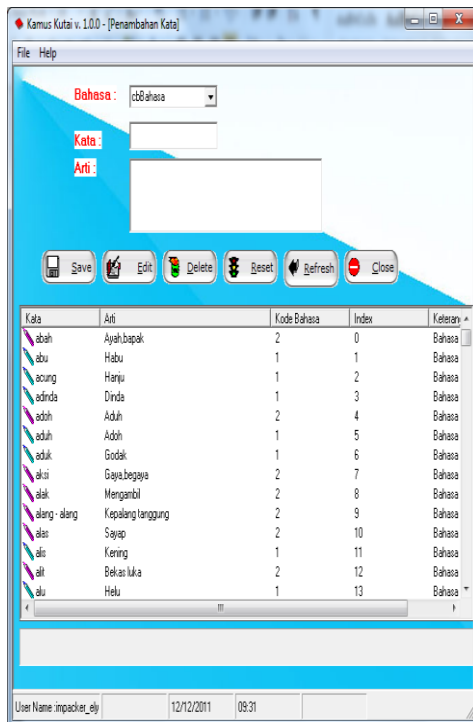
Tombol check juga dapat digunakan untuk pencarian kata namun pada bagian ini proses pencarian dilakukan dengan cara berbeda, kata yang dicari di input pada text box kata dan klik, dengan demikian pencarian akan diproses menggunakan timer yang ada, dimana kata akan dicari dengan melakukan pemeriksaan kata demi kata pada database dan akan berhenti pada saat kata ditemukan.

3. Halaman Tambah Kata

Sesuai dengan namanya halaman ini berfungsi untuk memperbaharui database kamus, mulai dari menambah data, mengubah dan menyimpan data kamus. Adapun tombol – tombol yang dimaksud adalah seperti yang terlihat pada gambar 4.8. Halaman ini hanya dapat diakses oleh admin, sehingga pada waktu akan masuk ke halaman ini admin diminta login terlebih dahulu. Berikut adalah gambar halaman tambah kata dan tampilan *login* yang ada.



Gambar 6. Halaman *Login*



Gambar 7. Tampilan halaman tambah kata

Berikut ini adalah penjelasan dari fungsi tombol yang terdapat pada halaman tambah kata.

a. Tombol save

Tombol save seperti berfungsi untuk menyimpan data baru yang sebelumnya tidak terdapat pada database. Jika data yang di input telah ada sebelumnya maka sistem akan memberikan *popup warning* kepada pengguna bahwa data sudah ada.

b. Tombol Edit

Tombol ini dibuat dengan memanggil method pada tombol edit agar dapat mengubah data yang ada dalam tabel kamus. Pada saat kata yang akan diedit dipilih maka kolom bahasa, arti dan kata akan terisi sesuai dengan inputan database yang sudah ada sebelumnya dan *caption* pada tombol edit akan berganti dengan *update*.

c. Tombol Delete

Tombol delete berfungsi menghapus data tertentu yang ada pada tabel. Seperti halnya tombol edit, maka sebelum menghapus data yang ada pada tabel, pengguna harus memilih data mana yang akan dihapus. Tombol ini apabila dipilih akan menampilkan message dialog sebagai peringatan kepastian kepada pengguna apakah data yang ada atau yang dipilih tersebut yakin akan

dihapus. Setelah itu program dijalankan sesuai dengan method delete untuk menghapus data yang diinginkan dari tabel dan apabila pengguna menekan tombol "Yes" maka data yang dipilih tersebut akan dihapus dari tabel dan terdapat message yang menginformasikan kepada pengguna bahwa data telah berhasil di hapus. Dalam form akan muncul data yang ada dalam tabel dimana letaknya sebelum data yang baru dihapus itu tersimpan. Bila pengguna menekan tombol "No", maka tampilan form akan kembali ke awal.

d. Tombol Reset

Apabila *user* memilih menu reset ini maka Combox pemilihan bahasa, pengisian kata dan arti akan dikosongkan atau dikembalikan ke nilai awal.

e. Tombol Refresh

Refresh digunakan untuk memperbaharui tampilan atau memuat kembali, menyegarkan tampilan yang ada setelah terjadinya perubahan, misalnya setelah terjadi penambahan ataupun pengurangan database kamus, posisi *index* kata akan berubah sehingga untuk melihat perubahan yang terjadi gunakan tombol refresh ini.

f. Tombol Close

Apabila *user* memilih tombol close maka form2 akan di tutup dan form 1 akan ditampilkan kembali

g. Prview Database

Pada bagian ini database ditampilkan menggunakan list view yang terurut menaik atau ascending yang terdiri dari kolom kata, kolom arti, kode bahasa, *index* dan keterangan.

4. Sub menu about

Pada saat menu about di klik maka akan ditampilkan pesan atau informasi tentang judul dan pembuat aplikasi kamus ini.

5. Sub menu cara menggunakan

Fasilitas ini berisi tentang bagaimana cara menggunakan kamus Kutai ini sehingga user yang belum begitu memahami cara menggunakan aplikasi ini dapat menggunakan fasilitas "cara menggunakan" untuk membantunya.

Cara Kerja Binary Search

Pembahasan berikut ini dibatasi pada pencarian data pada array terurut menaik dimana harga elemen – elemennya tidak ada yang sama.

Pencarian data dengan metode pencarian biner pada pencarian biner pada array tersebut dapat dijelaskan sebagai berikut ini. Jika array data memuat N buah elemen data, maka mula – mula kita tetapkan harga batas bawah dan batas atas interval. Dari dua harga tersebut kemudian dapat ditentukan titik tengah intervalnya. Misal harga batas bawah interval bawah dinyatakan dengan variabel *bawah*, harga batas atas interval dinyatakan dengan variabel *atas*, dan titik tengah interval dinyatakan dengan variabel *tengah*. Dengan demikian, maka harga awal untuk bawah adalah sama dengan 0 dan atas sama dengan arrKata. Sedangkan tengah dihitung dengan cara sebagai berikut :

Tengah = (atas + bawah) div 2

Jika cacah data dalam larik data adalah ganjil, maka titik tengah interval akan membagi larik data tersebut menjadi dua bagian yang persis sama. Sebaliknya, jika cacah data dalam larik data adalah genap, maka titik tengah interval akan membagi larik data menjadi dua bagian dimana salah satu bagian akan mempunyai cacah data lebih banyak.

Data – data yang mempunyai harga lebih kecil dari harga pada titik tengah interval akan berada disebelah kiri titik tengah interval dan data – data yang mempunyai harga lebih besar dari harga data pada titik tengah interval akan berada di sebelah kanan titik tengah interval. Data yang dicari, kemudian dibandingkan dan dicocokkan dengan data pada posisi tengah. Jika sama, maka data telah ditemukan dan proses selesai. Namun, jika data yang dicari lebih kecil dari harga data pada posisi tengah, berarti data yang dicari berada pada interval di sebelah kiri. Selanjutnya, proses pencarian hanya akan difokuskan pada interval tersebut dan kita tidak perlu lagi membandingkan dengan data – data pada interval kanan. Untuk itu harga atas perlu digeser menjadi tengah – 1.

Sebaliknya, jika elemen data yang dicari mempunyai harga lebih besar dari harga pada posisi tengah, berarti data yang dicari adalah berada di interval sebelah kanan. Selanjutnya proses pencarian hanya akan difokuskan pada interval tersebut. Harga bawah kemudian dinaikkan menjadi tengah + 1

Selanjutnya harga tengah akan ditentukan kembali berdasarkan harga bawah dan atas yang baru pada interval dimana data yang dicari berada. Perhitungannya dilakukan sebagaimana formula di atas. Perbandingan dan pencocokan antara data yang dicari dengan data pada posisi tengah akan dilakukan kembali. Dengan demikian akan terjadi tiga kemungkinan sebagaimana sebelumnya, yaitu data yang dicari mempunyai harga sama dengan harga data pada posisi tengah, lebih kecil, atau lebih besar. Pergeseran harga bawah dan atas akan dilakukan kembali berdasarkan harga – harga batas interval yang baru tersebut.

Proses seperti ini akan dilakukan secara terus menerus selama harga bawah kurang dari atau sama dengan harga atas. Pada akhirnya, jika data yang dicari memang berada dalam array data maka data tersebut akan dapat ditemukan, yaitu ketika data yang dicari mempunyai harga yang sama dengan harga data pada posisi tengah. Sebaliknya, apabila harga bawah telah sama dengan harga data pada posisi tengah, berarti data tersebut tidak ada dalam array data. Dengan demikian, maka proses pencarian dapat dihentikan. Dengan cara seperti ini, maka proses pencarian data dapat dilakukan dengan lebih cepat, karena proses perbandingan dan pencocokan data tidak harus dilakukan pada semua elemen data. Pencarian dengan metode pencarian biner akan memerlukan proses tambahan untuk perhitungan batas – batas interval, namun demikian, dalam cacah data yang cukup banyak, proses tambahan tersebut tetaplah lebih efisien dibandingkan jika harus melakukan perbandingan dan pencocokan satu per satu pada semua elemen data dalam array data sebagaimana dalam metode pencarian langsung ataupun pencarian menggunakan timer.

Berikut ini dituliskan solusi dalam bentuk algoritma pencarian data dengan metode *binary search*. Dari array kata arrKata dengan n buah elemen data dan data yang dicari sebagai katadicari maka dapat dijelaskan bahwa prosenya adalah sebagai berikut

1. Inisialisasikan
 $atas = arrKata$
 $bawah = 0$
2. Proses berulang langkah-3 s/d langkah-4
 $While\ bawah < -tinggi$
3. Hitung harga titik tengah intervalnya
 $tengah = (bawah + atas) \div 2$
4. Bandingkan harga data yang dicari dengan harga tengah interval
 $If\ katadicari = arrKata[tengah]$
 Jika ya, data ketemu dan cetak hasil
 (“Data “,katadicari,” ada pada urutan ke :”,tengah)
 Ke langkah-6
 Jika tidak, cetak lokasi data
 $If\ katadicari < arrKata[tengah]$
 Jika ya, tentukan batas atas interval baru
 $atas = tengah + 1$
 Jika tidak, tentukan batas bawah interval baru
 $bawah = tengah - 1$
5. Data tidak ketemu dan cetak pesan
 (“data”,katadicari,” tidak ditemukan dalam array kata”
6. selesai

berikut juga akan diberikan contoh penerapan pencarian *binary search* dalam pencarian data. Jika

diketahui sebuah array kata arrKata yang mempunyai tujuh elemen data string yang terurut menaik, sebagai berikut :

1. abu = habu
2. acak = ejek
3. adoh = aduh
4. aduh = adoh
5. aduk = godak
6. alak = ambil
7. alang – alang = kepalang tanggung

Berdasarkan data – data dalam array kata arrKata diatas, maka akan dicari elemen data yang mempunyai harga adoh = aduh. Penentuan lokasi titik tengah interval pada setiap langkah untuk membandingkan harga data yang dicari dengan harga data pada lokasi titik tengah interval tersebut ditunjukkan pada tabel 1 dan tabel 2 dibawah ini.

Tabel 1. Contoh pencarian data dengan metode pencarian biner (contoh 1)

Iterasi Ke :	Bawah	Atas	Tengah	arrKata [tengah]
1	1	7	4	Aduh=adoh
2	1	3	2	Acak = ejek
3	3	3	3	Adoh = aduh

Tabel 2. Contoh pencarian data dengan metode pencarian biner (contoh 2)

Iterasi Ke :	Bawah	Atas	Tengah	arrKata [tengah]
1.00	1	7.00	4.00	Aduh = adoh
2.00	1	7.00	6.00	Alak = ambil
3.00	3	7.00	7.00	Alang - alang = kepalang tanggung

Sedangkan tabel menunjukkan penentuan lokasi titik tengah interval pada pencarian data alang – alang = kepalang tanggung dalam array kata arrKata. Dalam kedua tabel tersebut, harga data yang dicari akan selalu dibandingkan dengan harga data pada lokasi titik tengah interval yang diperoleh. Jika keduanya mempunyai harga yang sama, berarti data telah ditemukan dan tinggal mencetak hasilnya untuk kemudian proses dapat dihentikan. Jika tidak sama, maka proses dapat diulangi kembali untuk menentukan titik tengah interval yang baru sampai harga atas sama dengan harga bawah sehingga titik tengah intervalnya juga akan sama.

Dalam contoh pertama, data “adoh = aduh” mempunyai posisi aktual pada urutan ke – 3. Data tersebut jika dicari menggunakan metode pencarian langsung akan ditemukan pada langkah iterasi ke – 3. Demikian juga jika diselesaikan dengan metode

pencarian biner juga akan ditemukan pada langkah iterasi ke -3. Dalam contoh kedua, data alang – alang = kepalang tanggung mempunyai waktu aktual pada urutan ke -7. Jika menggunakan metode pencarian langsung, maka data tersebut baru akan ditemukan pada langkah ke -7. Tetapi jika menggunakan metode pencarian *binary search* akan ditemukan pada langkah iterasi ke -3. Berdasarkan kedua contoh sederhana tersebut, kita dapat membandingkan efisiensi masing – masing metode pencarian langsung dan pencarian biner jika diterapkan dalam program – program aplikasi. Secara umum, faktor utama yang menentukan dalam pemilihan metode pencarian data adalah kondisi data dan cacah data. Dalam kondisi dimana data dalam array kata belum diurutkan, maka pencarian data tidak mungkin dilakukan dengan metode pencarian langsung. Namun jika cacah datanya cukup besar, maka akan lebih baik jika kita bisa menerapkan metode pencarian *binary search* dengan syarat harus mengurutkan datanya terlebih dahulu. Cara demikian lazim dilakukan dalam suatu sistem pengolahan data, karena kenyataannya pengolahan data akan selalu melibatkan sejumlah data yang besar. Sehingga operasi pengurutan yang harus dilakukan tersebut tidak selalu merugikan.

Dari pembahasan diatas secara ringkas urutan langkah-langkah logis penyelesaian masalah yang ada disusun secara sistematis dan logis dalam algoritma sebagai berikut :

```

Kamus
    ConstN : integer =
    8
    { misalkanjumlahelemen
    array maksimum = 8 }
    TypearrKata =
    array[ 1 .....
    N ]ofinteger
    atas,        bawah,
    tengah : Integer
    katadicari :string
    Ketemu :string
ALGORITMA
    Input      (cari)
    { memintanilai data
    yang akandicari}
    atas← 1 { indeks
    array dimulaidari0 }
    bawah← N
    Ketemu← False
    While (atas<bawah) and
    (notketemu) do
    
```

```

        tengah←
        (atas + bawah) div 2
        IfarrKata
        [tengah] =
        katadicarithen
            Ketemu←true
            Else
                If( arrKata
                [Tengah] <katadicari )
                then { cari di
                bagiankanan }
                atas← Tengah + 1
            Else
                bawah← Tengah - 1
                { cari di bagiankiri }
            Endif
        Endif
    EndWhile
    If (ketemu) then
        Output
        ('Kata','Arti','In
        dex','Keterangan',
        Tengah )
    Else
        Output( 'Datatidak
        ditemukan' )
    Endif

```

4.5 Tahapan Pengujian Sistem

Berikut adalah tahapan pengujian sistem pencarian kata berdasarkan jumlah array kata yang terdapat pada tabel 4.4 dan algoritma pencarian *binary search*.

Tabel 3. Contoh Nilai-Nilai data yang sudah terurut

arr Kata	A b u = h a b u	A c a k = e j e k	A d o h = a d u h	A d u h = a d o h	A d u k = g o d a k	Alak = m e n g a m b i l	Ala n g = k e p a l a n g t a n g u n g	Al i t = b e k a s l u k a s
	1	2	3	4	5	6	7	8

1. Data Uji 1 :

Cari = "aduh = adoh"

Loop pertama : Tengah = (BatasAtas + BatasBawah) div 2 = (1 + 8) div 2 = 4

arrKata [Tengah] = arrKata [4] = (aduh = adoh) , berarti loop pertama data langsung ditemukan

2. Data Uji 2

Cari = "aduk = godak"

Loop pertama : Tengah = (BatasAtas + BatasBawah) div 2 = (1 + 8) div 2 = 4

arrKata [Tengah] = arrKata [4] = (aduh = adoh) < cari = (aduk = godak), berarti

BatasAtas = Tengah - 1 = 4 - 1 = 3

Loop kedua : Tengah = (BatasAtas + BatasBawah) div 2 = (3 + 8) div 2 = 5

arrKata [Tengah] = arrKata [5] = (Aduk = godak) = cari (aduk = godak), berarti

BatasAtas = Tengah - 1 = 6 - 1 = 5

berarti setelah loop kedua, data ditemukan

3. Data uji 3

Cari = "abcd = abc"

Loop pertama : Tengah = (BatasAtas + BatasBawah) div 2 = (1 + 8) div 2 = 4

arrKata [Tengah] = arrKata [4] = (Aduh = adoh) > cari = (abcd = abc), berarti BatasBawah = Tengah + 1 = 4 + 1 = 5

Loop kedua : Tengah = (BatasAtas + BatasBawah) div 2 = (1 + 5) div 2 = 3

arrKata [Tengah] = arrKata [3] = (Adoh = aduh) > cari = (abcd = abc) , berarti BatasAtas = Tengah + 1 = 3 + 1 = 4

Loop ketiga : Tengah = (BatasAtas + BatasBawah) div 2 = (4 + 1) div 2 = 2

arrKata [Tengah] = arrKata [2] = (acak = ejek) > cari = (abcd = abc), berarti BatasAtas = Tengah + 1 = 2 + 1 = 3, berarti setelah loop ketiga data tidak ditemukan.

Untuk jumlah data sebanyak n, maka proses perbandingan maksimal sebanyak (log n) kali. Untuk contoh di atas, jumlah data 8, maka proses perbandingan maksimal sebanyak 3 kali.

Dalam *binary search*, pertama kali kita membandingkan item dalam posisi tengah array. Jika cocok kita dapat mengembalikannya secepatnya.

Jika data yang dicari lebih kecil dari data tengah, maka item yang dicari berada pada setengah array kecil, jika lebih besar berada pada setengah array

$$31.25:2 = 15.625 \sim 15$$

1. $15.625:2 = 7.8125 \sim 7$
2. $7.8125:2 = 3.90625 \sim 3$
3. $3.90625:2 = 1.953125 \sim 1$
4. $1.953125:2 = 0.9765625 \sim 0$

Pembaca dapat melihat bahwa pada setiap kali pencarian, larik dibagi menjadi dua bagian yang berukuran hampir sama. Pada setiap pembagian, elemen tengah dibandingkan apakah sama dengan katadacari (if (arrKata[tengah] = katadacari). Pada kasus terburuk, yaitu pada kasus katadacari tidak terdapat didalam larik atau katadacari ditemukan setelah ukuran larik tinggal 1 elemen, larik akan dibagi sebanyak $^2\log(n)$ kali, sehingga jumlah perbandingan yang dibutuhkan adalah sebanyak $^2\log(n)$ kali. Kita katakan bahwa waktu pencarian sebanding dengan $^2\log(n)$ untuk $n=256$ elemen misalnya, kasus terburuk menghasilkan pembagian larik sebanyak $^2\log(256)=8$ kali dibandingkan dengan metode pencarian beruntun yang pada kasus terburuk melakukan perbandingan sebanyak 256 kali. Jadi untuk larik yang terurut, algoritma pencarian bagidua jauh lebih cepat dari pada algoritma pencarian.

KESIMPULAN

Berdasarkan hasil perancangan dan penerapan, berikut adalah kesimpulan dari hasil penelitian yang dilakukan:

1. Telah dihasilkan aplikasi pencari kata dengan menggunakan metode binary search dalam database kamus bahasa Kutai.
2. *Binary search* adalah algoritma pencarian yang efisien dengan jumlah perbandingan sebanyak $^2\log n$ kali.

DAFTAR PUSTAKA

- [1] Andayani, Yuliana, dan Santoso. "Penggunaan Metode Binary Search Pada Translator Bahasa Indonesia – Bahasa Jawa". Politeknik Elektronika Negeri Surabaya : Surabaya.
- [2] Bahrah, A. 1992. *Kamus Bahasa Daerah Kutai Umum – Indonesia*. Lembaga Pembinaan Kebudayaan Kutai : Tenggarong.
- [3] MADCOMS. 2010. *Mahir Dalam 7 Hari Microsoft Visual Basic 6.0 & Crystal Report 2008*. Penerbit ANDI : Madiun.
- [4] Munir, A. 2005. *Algoritma & Pemrograman dalam bahasa pascal dan C*. Penerbit Informatika : Bandung.
- [5] Pusat Bahasa. *Kamus Besar Bahasa Indonesia Edisi Keempat*. Gramedia Pustaka Utama : Jakarta.
- [6] Putra Indra. 2004. *Membuat Program Aplikasi Nyata Dengan Visual Basic 6.0*. Penerbit ANDI : Yogyakarta.
- [7] Rosa dan Shalahuddin. 2010. *Modul Pembelajaran Algoritma dan Pemrograman*. Penerbit MODULA : Bandung.
- [8] Sanjaya, R. 2005. *Membuat Menu Cantik Untuk Aplikasi Visual Basic 6.0*. Penerbit ANDI : Yogyakarta.
- [9] Sismoro,H dan Iskandar,K. 2004. *Struktur Data & Pemrograman Dengan Pascal*. Penerbit ANDI : Yogyakarta.
- [10] Sjartuni,A. 2000. *Tuntunan praktis pemrograman dasar – dasar pemrograman dengan visual basic 5.0*. Elex Media Komputindo : Jakarta .
- [11] Suryadikara, F. 1979. *Bahasa Kutai*. Pusat Pembinaan dan Pengembangan Bahasa, Departemen Pendidikan dan Kebudayaan : Tenggarong.
- [12] Tim Wahana Komputer. 2000. *Pemrograman visual basic 6.0*. Penerbit ANDI : Yogyakarta.
- [13] Wahid, F. 2004. *Dasar – dasar Algoritma dan Pemrograman*. Penerbit ANDI : Yogyakarta.
- [14] Wibisono, K. *Buku Bahasa Kutai di Luncurkan*.2009.URL: <http://www.antaranews.com/berita/1249925614/buku-bahasa-kutai-diluncurkan> diakses pada tanggal 10 Mei 2011.
- [15] Anonymous. *Bahasa Kutai*. 2011. URL:<http://id.wikipedia.org/wiki/BahasaKutai> diakses pada tanggal 10 Mei 2011.
- [16] Anonymous. *Bahasa Kutai Terancam Punah*. 2007. URL:<http://www.kutaikartanegara.com/news.php?id=1744> diakses pada tanggal 19 mei 2011.
- [16] Anonymous. *Saat Erau dianjurkan Berbahasa Kutai*. 2010. http://humas.kutaikartanegarakab.go.id/index.php/read/saat-erau-dianjurkan_berbahasa-kutai/ diakses pada tanggal 19 mei 2011