



**PERANCANGAN SIMULASI HASHING DIVISION REMAINDER
DAN LINEAR PROBING UNTUK PEMETAAN ALAMAT REKORD
DAN MENGATASI KOLISI PADA SISTEM BERKAS**

Manorang Gultom^{1*}

¹Sistem Informasi, Fakultas Teknologi Informasi, Universitas Widya Dharma Pontianak, Jln. HOS
Cokrominoto No. 445 Pontianak, Kalimantan Barat, Indonesia

*email: manorangtm@gmail.com

Received: 23 Maret 2021 Accepted: 30 Juni 2021 Published: 30 Juni 2021

Abstrak

Tujuan Penelitian ini adalah untuk menerapkan proses pemetaan nilai kunci dari suatu record menjadi alamat fisik record dalam media penyimpanan. Penerapan konsep pemetaan nilai kunci menjadi alamat fisik record disajikan dalam bentuk program simulasi dengan menggunakan Bahasa pemrograman VB NET 2010 dan Mysql. Pada proses pemetaan nilai kunci menjadi alamat fisik record, program simulasi menggunakan pendekatan Division Remainder dan jika terjadi kolisi, program simulasi akan menerapkan pendekatan Linear Probing untuk mencari alamat alternatif baru bagi kunci yang dipetakan. Proses pemetaan nilai kunci menjadi alamat fisik, program simulasi membutuhkan data berupa ukuran tabel dan kunci yang akan dipetakan. Ukuran tabel menjadi bilangan pembagi yang digunakan pada pendekatan Division Remainder dengan jumlah kunci yang dipetakan lebih kecil dari ukuran tabel. Kunci akan dipetakan satu demi satu dan jika terjadi kolisi, program simulasi langsung menerapkan Linear Probing pertama untuk mencari alamat alternatif baru bagi kunci yang mengalami kolisi. Jika masih kolisi dengan kunci yang lain, Linear probing akan dilakukan lagi dengan menambah nilai indeks i secara linier sampai alamat baru kosong ditemukan.

Kata kunci: Division Remainder, Linear Probing, Kolisi, Program Simulasi

Abstract

The purpose of this research is to apply the process of mapping the key value of a record to the physical address of the record in the storage medium. The application of the concept of mapping key values to physical addresses of records is presented in the form of a simulation program using the VB NET 2010 programming language and Mysql. In the process of mapping key values to physical addresses of records, the simulation program uses the Division Remainder approach and in the event of collision, the simulation program will apply the Linear Probing approach to find new alternatif addresses for the mapped keys. In the process of mapping key values to physical addresses, the simulation program requires data in the form of tabel sizes and keys to be mapped. The tabel size becomes the divisor used in the Division Remainder approach with the number of keys mapped that is smaller than the tabel size. The keys are mapped one by one and in case of collision, the simulation program immediately applies the first Linear Probing to find a new alternatif address for the collision key. If there are still collisions with other keys, Linear probing will be carried out again by increasing the index value i linearly until a new blank address is found.

Keywords: Division Remainder, Linear Probing, Collision, Simulation Program.

How to cite (in APA style): Gultom, M. (2021). Perancangan simulasi hashing division remainder dan linear probing untuk pemetaan alamat record dan mengatasi kolisi pada sistem berkas. *Jurnal Pendidikan Informatika dan Sains*, 10(1), 64-79.



PENDAHULUAN

Pengelolaan data pada sistem berkas merupakan salah satu aktifitas rutin yang dilakukan oleh para pengguna komputer. Pengelolaan data dapat berupa membaca, menyimpan, mengubah dan menghapus data dari dan ke media penyimpanan. Semua pengguna komputer tidak akan menghendaki data yang di kelola mengalami kerusakan atau hilang saat di operasikan. Kerusakan data dapat di sebabkan beberapa hal, misalnya kerusakan perangkat media penyimpanan, virus computer, kolisi dan lain lain. Kolisi (*Collision*) adalah suatu kejadian dimana saat penyimpanan data dilakukan, maka data yang baru di simpan akan menempati lokasi yang sama dengan data yang sudah tersimpan di lokasi yang sama sebelumnya atau dengan kata lain terjadi kolisi atau benturan alamat data, dimana dua data yang berbeda menempati alamat fisik yang sama di media penyimpanan.

Organisasi berkas adalah suatu cara atau metoda untuk mengakses data dari dan ke media penyimpanan yang artinya bagaimana suatu data di simpan ke media penyimpanan sehingga mudah untuk di akses saat di perlukan. Dalam sistem berkas dikenal beberapa jenis organisasi file yaitu Sequential, Relatif, Index sequensial dan Multi Key. Dalam organisasi File Relatif, terdapat beberapa jenis metoda yang dapat digunakan untuk memetakan nilai *key* menjadi alamat fisik data di media penyimpanan, diantaranya Pemetaan Langsung (*Direct Mapping*), Pencarian Tabel (*Direktory Lookup*) dan Kalkulasi Alamat (*Calculation*). Pada metoda Kalkulasi Alamat, terdapat suatu proses pemetaan nilai *key* suatu data menjadi alamat fisik data di media penyimpanan atau dikenal dengan istilah *Hashing*. *Hashing* adalah suatu proses pemetaan nilai *key* suatu data menjadi alamat fisik data tersebut di media penyimpanan.

Hashing dapat diterapkan dengan menggunakan berbagai metoda, yaitu *Division Remainder*, *Mid Square* dan *Folding*. Penerapan *Hashing* dengan metoda *Division Remainder* maka Alamat relatif dari suatu nilai *key* di dapat dari sisa hasil pembagian nilai *key* tersebut dengan suatu bilangan yang disebut sebagai bilangan pembagi. Terdapat beberapa faktor yang harus dipertimbangkan dalam pemilihan bilangan pembagi diantaranya yaitu pembagi harus diseleksi untuk mengurangi benturan, sebaiknya pembagi itu merupakan bilangan prima. Sekalipun bilangan pembagi telah ditentukan dengan baik untuk mengatasi benturan, bila ruang alamat dari berkas relatif sudah hampir penuh, maka peluang terjadinya kolisi (benturan) akan meningkat.

Mengatasi kolisi dalam organisasi file Relatif dapat dilakukan dengan berbagai cara, yaitu *Linear Probing*, *Linear Quotient* dan *Linear Double Hashing*. Dalam penelitian ini metoda yang digunakan dalam mengatasi kolisi adalah dengan menggunakan *Linear Probing*.

(Muharto & Ambarita, 2016) mengungkapkan bahwa perancangan sistem merupakan tahap selanjutnya setelah analisa sistem, mendapatkan gambaran dengan jelas tentang apa yang dikerjakan pada analisa sistem, dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut. Pendapat (Lee, 2014) mengatakan bahwa Sebuah proyek aplikasi windows dapat berupa satu program yang termasuk antar muka pengguna, sebuah jendela di layar computer yang menggunakan system operasi windows. Sedangkankan (Nurseto, 2011) mengatakan bahwa Media pembelajaran adalah wahana penyalur pesan dan informasi belajar. Media pembelajaran yang dirancang secara baik akan sangat membantu peserta didik mencapai tujuan pembelajaran. (Saputra, Wawan & P, E, 2012) mengatakan bahwa kebutuhan media pembelajaran yang aplikatif diperlukan dalam usaha untuk memudahkan proses belajar mengajar, seperti halnya media pembelajaran interaktif organisasi komputer

(Jeperson, 2014) mengungkapkan bahwa masukan adalah energi yang dimasukkan kedalam sistem yang dapat berupa perawatan (*maintenace input*), dan masukan sinyal (*signal input*).

(Muslihudin, 2016) mengungkapkan bahwa sistem keluaran adalah energi yang diolah, diklasifikasikan menjadi keluaran yang berguna untuk subsistem lain.

(Id/entri, 2019), 1 simulasi adalah metode pelatihan yang meragakan sesuatu dalam bentuk tiruan yang mirip dengan keadaan yang sesungguhnya; 2 penggambaran suatu sistem atau proses dengan peragaan berupa model statistik atau pemeranan. Simulasi adalah suatu aplikasi yang meniru proses sebenarnya dari suatu metode untuk menyelesaikan suatu permasalahan. Simulasi dapat disajikan dengan menggunakan perangkat lunak sedemikian rupa sehingga menampilkan proses persis seperti keadaan nyata.

(W Sri, 2011) mengungkapkan bahwa Pada Hash Tertutup terdapat kemungkinan lebih dari satu data memiliki nilai fungsi yang sama sehingga terjadi tabrakan (*collision*). Cara untuk mengatasi tabrakan ini dapat dilakukan dengan beberapa strategi seperti, Resolusi Linier (*Linear Resolution*). Sedangkan menurut (Yanti, 2016) Metoda pencarian Hash Tertutup (*Closed Hash*) dengan beberapa strategi untuk mengatasi tabrakan (*collision*) pada saat penempatan data antara lain, Resolusi Linier (*Linear Resolution*) dan *Overflow*, yang terbagi dua yaitu *Rehashing* dan Sekuensia serta *Double Hashing*, terbagi dua yaitu *Rehashing* dan Sekuensial.

Menurut (P, Eko & Y, Anton & F, 2016) *Hashing* merupakan metode pengaksesan data yang dilakukan dengan cara memetakan/mengkonversikan himpunan kunci record menjadi himpunan alamat memori, sehingga *range address* menjadi kecil. Fungsi untuk mengkonversikan nilai kunci aktual menjadi lokasi alamat disebut fungsi hash, sedangkan metode pencarian yang memanfaatkan fungsi hash disebut sebagai *hashing* atau *hash addressing*. (Y, 2014) mengungkapkan bahwa resolusi terhadap tabrakan merupakan proses untuk menangani kejadian dua atau lebih key dipetakan ke alamat yang sama pada tabel. Caranya yaitu dengan mencari lokasi yang kosong dalam tabel hash secara terurut.

Menurut (Wulandari & Rusjdi, 2016) *Division Remainder* adalah salah satu jenis hash yang dapat digunakan untuk memetakan nilai *key* suatu record menjadi alamat fisik dari record tersebut. *Division Remainder* bekerja dengan cara Alamat relatif dari suatu nilai *key* merupakan sisa dari hasil pembagian nilai *key* tersebut dengan suatu bilangan yang disebut sebagai bilangan pembagi dimana bilangan pembagi ini merupakan bilangan prima. Rumus untuk melaksanakan pemetaan nilai *key* menjadi alamat fisik dari record dapat dilihat pada Persamaan 1.

$$ADDR = KEY \text{ MOD } DIV \quad (1)$$

Dimana *ADDR* adalah alamat relatif dari record, *KEY* adalah kunci dari record tersebut dan *DIV* adalah bilangan pembagi dan merupakan bilangan prima.

Linear Probing adalah salah satu metode yang digunakan untuk mengatasi kolisi atau benturan, dimana terdapat dua atau lebih record yang berbeda menempati alamat yang sama sebagai hasil dari proses *hashing* nilai *key*. *Linear Probing* akan melakukan hash yang kedua terhadap nilai *key* dengan menambah satu secara linear sampai alamat relatif tidak ada yang sama dengan alamat relatif sebelumnya. Menurut (Ridho & Prianto, 2020) *Linear Quotient* merupakan konsep pemetaan dengan kalkulasi untuk mendapatkan alamat address, sehingga pencarian dapat langsung merujuk pada lokasi record tersebut

Menurut (Wulandari & Rusjdi, 2016) dalam Perancangan Simulasi Media Pembelajaran Dengan Metode *Division Remainder* Untuk Pencarian Alamat Relatif Pada Proses Penempatan Data, rumus untuk *Linear Probing* dalam mengatasi kolisi atau benturan dapat dilihat seperti pada Persamaan 2.

$$h^i(X) = (h^{i-1}(X) + 1) \text{ mod } [ukuran\ tabel] + 1 \quad (2)$$

Dimana $h_i(X)$ adalah alamat relatif yang dihasilkan pada proses perhitungan awal. Nilai i akan bertambah satu secara linier sampai alamat kosong yang belum ditempati oleh record lain ditemukan.

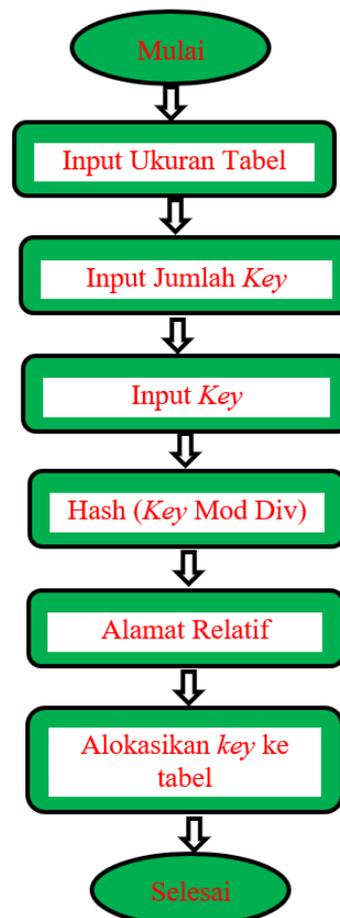
METODE

Penulisan karya ilmiah ini dilakukan dengan beberapa tahapan yaitu mencari topik penulisan, studi pustaka, penentuan bahasa pemrograman dan perancangan program simulasi. Topik yang dipilih pada penulisan karya ilmiah ini adalah tentang pemetaan nilai *key* menjadi alamat fisik record dalam media penyimpanan dan cara mengatasi kolisi yang terjadi setelah proses pemetaan dilakukan dalam system berkas. Pada tahap perancangan program aplikasi dilakukan dengan beberapa tahapan, yaitu perancangan database untuk user, perancangan menu utama, perancangan antarmuka program dan implementasi algoritma *Division Remainder* dan *Linear Probing*.

HASIL DAN PEMBAHASAN

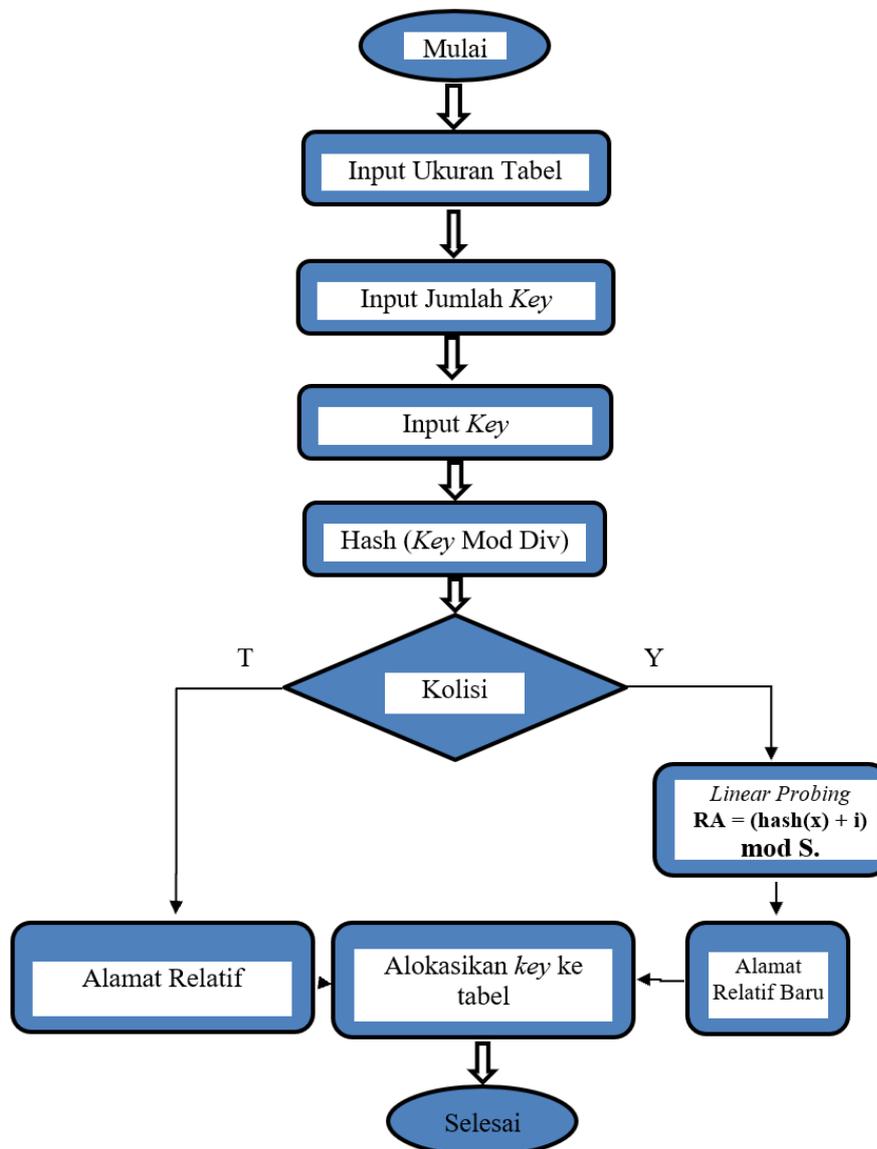
Analisis Algoritma *Division Remainder* dan *Linear Probing*

Sebelum melakukan perancangan program simulasi, penulis terlebih dahulu menganalisis kinerja algoritma *Division Remainder* dan *Linear Probing*. Proses pemetaan nilai *key* menjadi alamat fisik dengan algoritma ini memerlukan beberapa data yang harus di ketahui yaitu berupa ukuran tabel, jumlah *key* dan kumpulan *key* yang akan dipetakan. Dalam perancangan program simulasi ini, ukuran tabel digunakan sebagai bilangan pembagi. Flowchart dari *Division Remainder* dalam memetakan nilai *key* menjadi alamat fisik record disajikan pada Gambar 1.



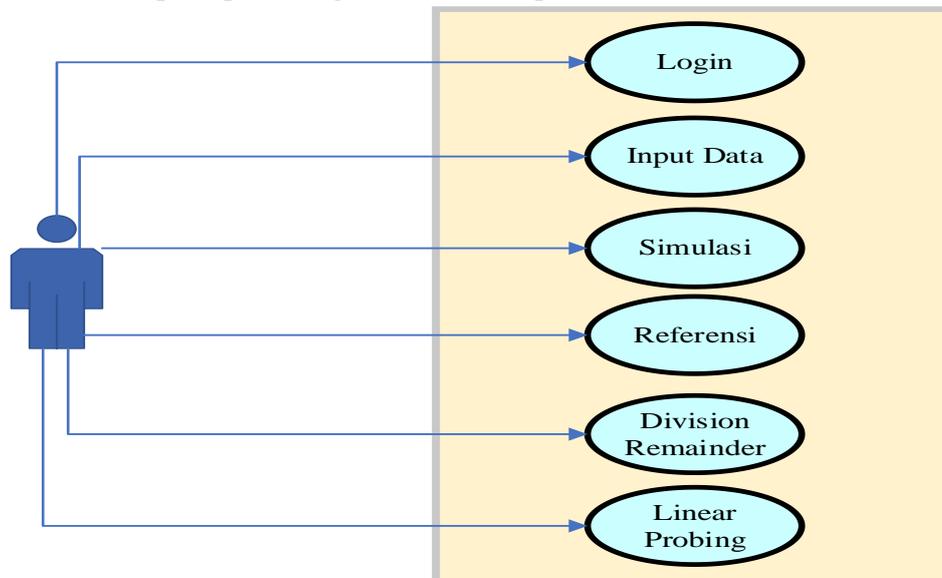
Gambar 1. Flowchart Aplikasi Simulasi *Division Remainder*

Apabila pada proses pemetaan nilai *key* menjadi alamat menghasilkan angka yang sama dengan pemetaan *key* sebelumnya, maka algoritma *Linear Probing* akan mencari alamat alternatif baru untuk *key* tersebut dengan mengikuti flowchart pada gambar 3.2. Nilai integer *i* dimulai dari 1 dan akan menaik secara otomatis bertambah satu selama alamat baru belum ditemukan tanpa terjadi kolisi. Setelah alamat alternatif baru ditemukan maka record akan disimpan. Semakin kecil factor muat dari tabel atau media penyimpanan yang digunakan, maka kemungkinan terjadinya kolisi akan tinggi karena cadangan alamat yang tersedia semakin sedikit. *Flowchart Linear Probing* pada aplikasi yang dibuat dapat dilihat pada Gambar 2.



Gambar 2. Flowchart *Linear Probing*

Untuk menjelaskan mengenai proses-proses sistem berjalan yang ada di aplikasi simulasi, maka penulis menggunakan teknik pemodelan *Unified Modeling Language* (UML) yang merupakan salah satu pemodelan teknik analisis untuk sebuah sistem. Penulis akan menguraikan sistem berjalan pada aplikasi simulasi seperti pada diagram Use Case pada Gambar 3.

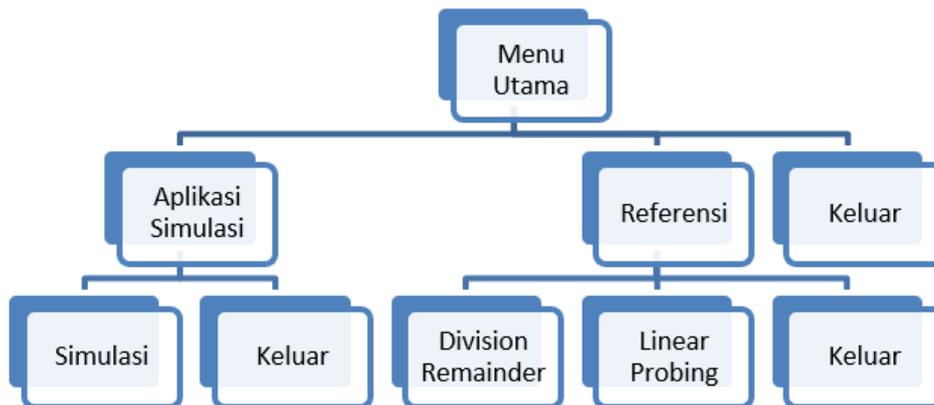


Gambar 3. Diagram Use Case Aplikasi

Berdasarkan Gambar 3, dapat diketahui bahwa dalam sistem terdapat satu aktor yang berperan sebagai pengguna aplikasi. Aktor dapat melakukan aktifitas berupa login sebagai hak akses ke aplikasi, dimana hak akses ini disimpan dalam database mysql. Setelah login sukses, pengguna akan diarahkan ke menu utama untuk menjalankan aplikasi dengan memilih menu yang ada pada sistem. Untuk menjalankan program simulasi, maka aktor memilih menu aplikasi selanjutnya memilih sub menu Simulasi. Aktor akan melakukan proses input data yaitu berupa ukuran tabel hash, jumlah kunci yang akan dipetakan, menginput semua kunci, melakukan proses *Division Remainder* dan proses *Linear Probing* jika terjadi kolisi.

Aktor juga dapat melihat teori tentang *Division Remainder* dan *Linear Probing* yang disajikan dalam bentuk form dengan memilih menu Referensi. Apabila aktor ingin keluar dari aplikasi, maka dapat dilakukan dengan memilih menu keluar pada menu utama.

Selain rancangan alur program, pada aplikasi ini juga dirancang menu utama untuk mempermudah pengguna menjalankan program. Berdasarkan rancangan menu pada Gambar 4 terdapat menu pilihan Aplikasi Simulasi dengan submenu Simulasi yang merupakan menu untuk menjalankan program simulasi. Menu Referensi memiliki submenu *Division Remainder* yang berfungsi untuk menampilkan teori singkat tentang *Division Remainder* dan submenu *Linear Probing* untuk menampilkan teori singkat tentang *Linear Probing*.



Gambar 4. Rancangan Menu Utama Aplikasi

Setelah perancangan alur program dan menu utama, selanjutnya dilakukan perancangan antarmuka program (Gambar 5). Antarmuka adalah sarana komunikasi antara program dengan pengguna program. Antarmuka diharapkan dapat membantu pengguna program lebih mudah untuk menjalankan program dan tidak mengalami kesalahan saat mengoperasikan program.



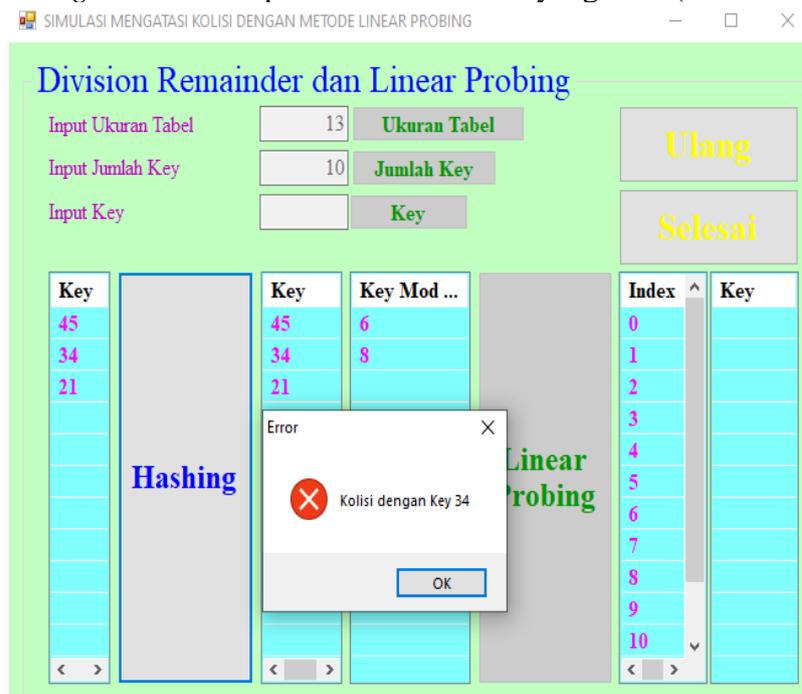
Gambar 5. Antarmuka Program Simulasi

Kotak text Ukuran Tabel digunakan sebagai sarana untuk menginput ukuran tabel (S). Saat ukuran tabel di input dan menekan tombol Ukuran Tabel, maka tabel Index akan terbentuk sesuai dengan angka yang di input di kotak text Ukuran Tabel yang diinput.

Kotak text Jumlah Key digunakan sebagai sarana menginput jumlah key yang akan di kalkulasi. Penginputan key yang akan di kalkulasi akan berulang sesuai dengan jemplah key yang di input.,Setelah key di input dan tampil di tabel Key, maka langsung menekan tombol Hashing untuk melakukan kalkulasi nilai key dengan metoda *Division Remainder*.

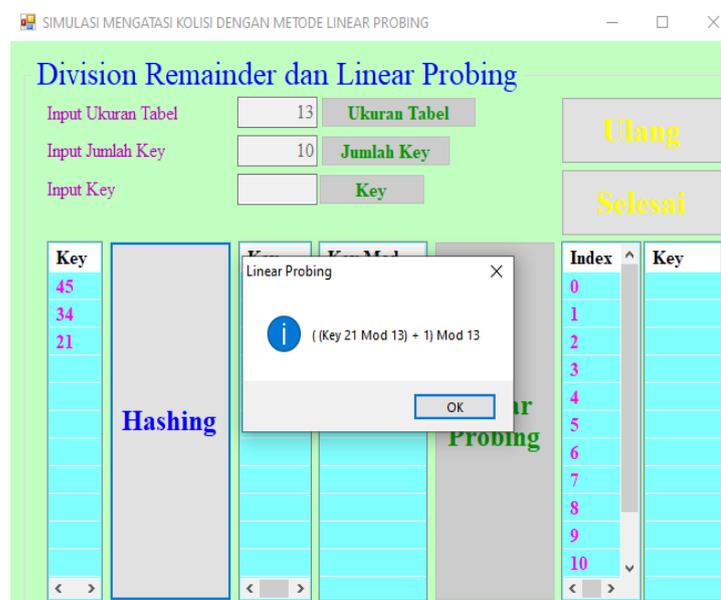
Setiap proses kalkulasi alamat, program akan langsung melakukan validasi apakah alamat key mengalami kolisi. Jika terjadi kolisi, maka program akan memberitahu ke pengguna dan pengguna wajib menekan tombol *Linear Probing* untuk mengatasi kolisi tersebut hingga mendapatkan alamat

alternatif yang baru. Berikut adalah tampilan proses *Linear Probing*. Dalam contoh ini *key* 21 mengalami kolisi dengan *key* 34 yaitu sama sama mendapatkan alamat 8, maka untuk *key* 21 akan di terapkan *Linear Probing* untuk mendapatkan alamat relatif yang baru (Gambar 6.)



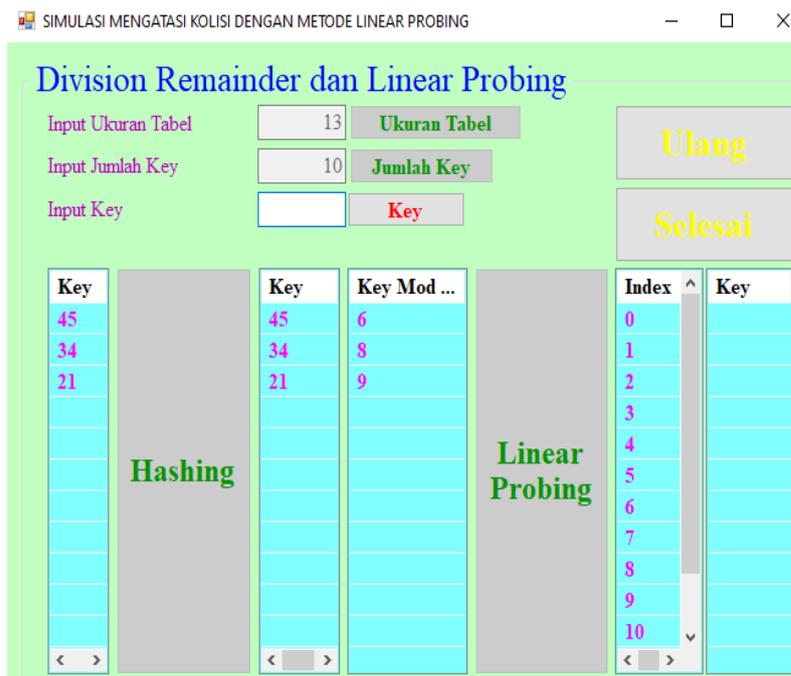
Gambar 6. Kolisi *key* 21 dengan 34

Informasi pada Gambar 6 merupakan hasil validasi program terhadap alamat yang didapat setelah melakukan kalkulasi terhadap *key* 21 ternyata kolisi dengan *key* 34, maka dengan menekan tombol OK, program akan melakukan proses *Linear Probing* dengan nilai $i=1$ seperti gambar 7.



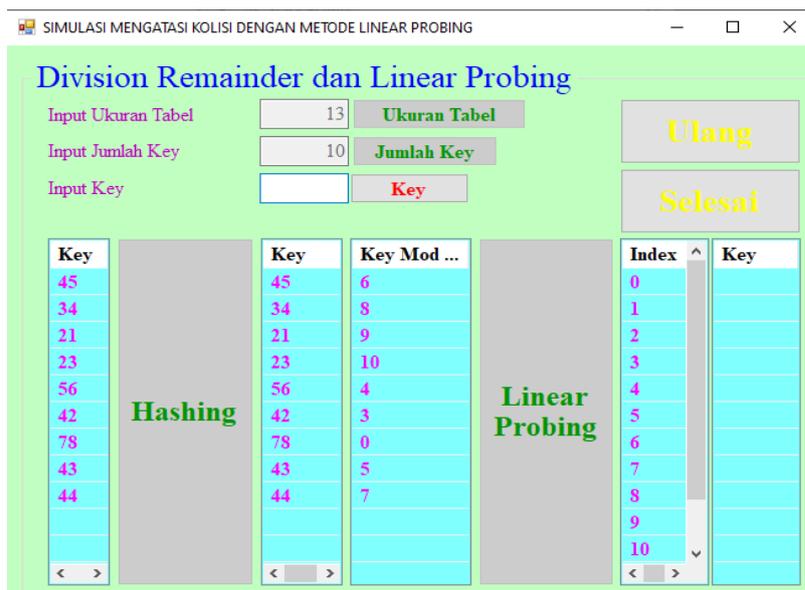
Gambar 7. Proses *Linear Probing* *key* 21

Dengan menekan tombol OK, maka akan di dapat alamat relative baru untuk *key* 21 yaitu index 9 seperti Gambar 8.



Gambar 8. Alamat Relatif baru Key 21 Setelah proses *Linear Probing*

Secara berurut saat insert *key* 23, 56, 42, 78, 43 dan 44 dan telah melalui proses Linear Probing, maka di dapat alamat relatif seperti Gambar 9.



Gambar 9. Insert Key 23,56,42,78,43 dan 44

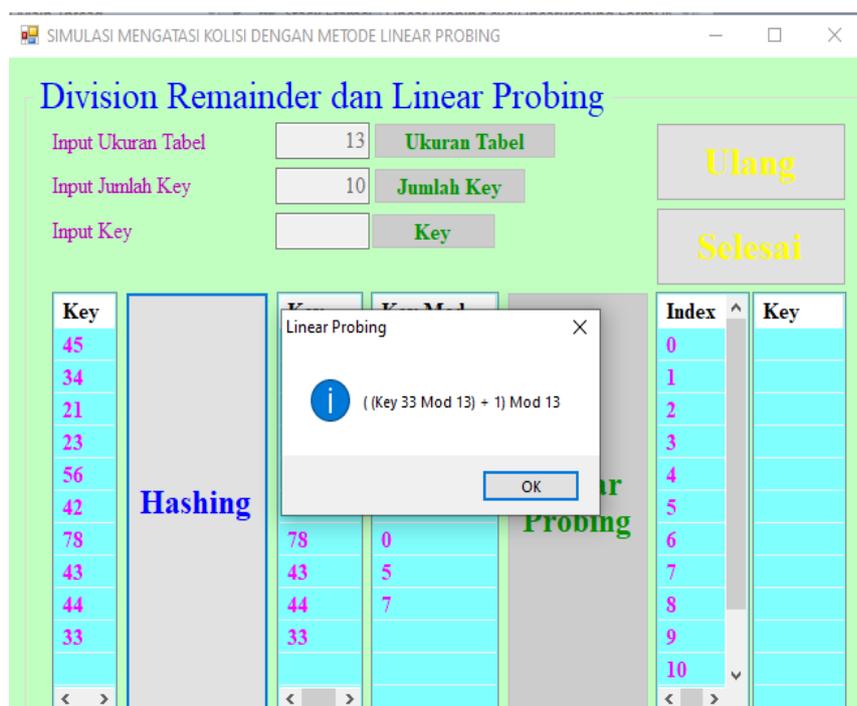
Dalam aplikasi ini apabila proses *Linear Probing* pertama masih mengalami kolisi, maka proses akan di ulang dengan menambah nilai *i* mulai dari 1 sampai N secara berurut hingga didapatkan alamat relatif yang belum di tempati oleh record lain. Misalnya saat memetakan *key* 33, maka kolisi berlangsung secara berulang dimana index *i* bertambah 1 mulai dari $i=1$ sampai $i=4$.

Berikut tampilan proses *Linear Probing* yang berlangsung mulai dari $i=1$ hingga $i=4$ untuk mendapatkan alamat relatif key 33 karena kolisi yang terjadi secara berulang (Gambar 10).



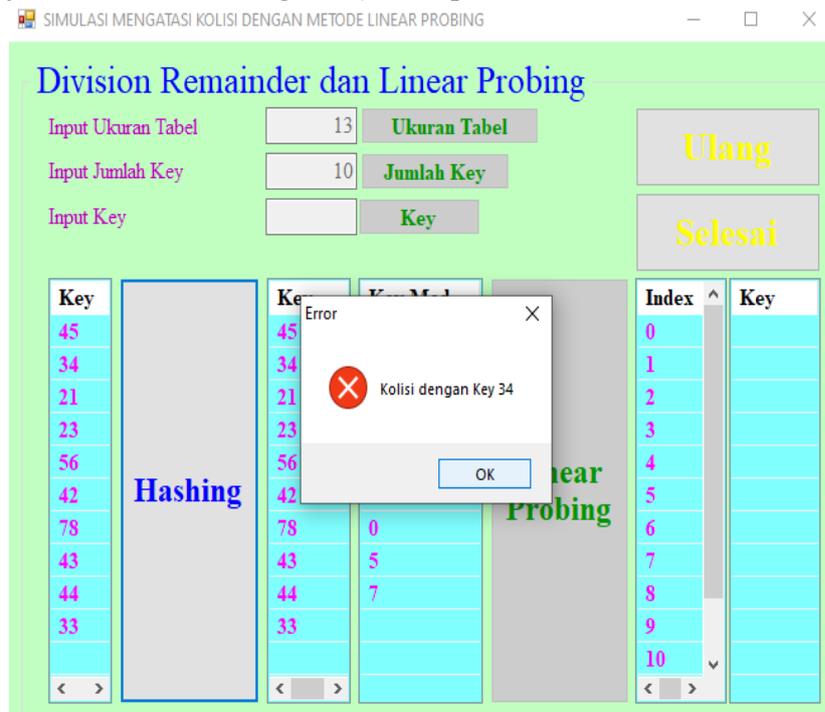
Gambar 10. Kolisi key 33 dengan 44

Dengan menekan tombol OK, maka akan dilakukan proses *Linear Probing* dengan nilai index $i=1$, namun masih terjadi kolisi seperti Gambar 11.



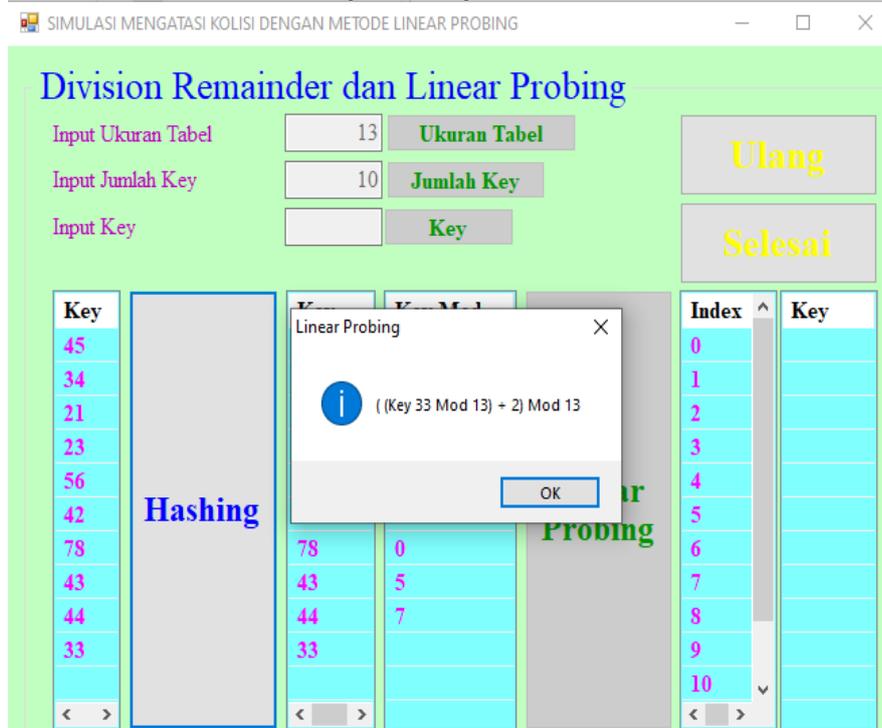
Gambar 11. *Linear Probing* key 33 dengan $i=1$

Dengan menekan tombol OK, proses *Linear Probing* berikutnya dilakukan untuk *key* 33 dengan $i=2$ dan ternyata masih kolisi dengan *key* 34 seperti Gambar 12.



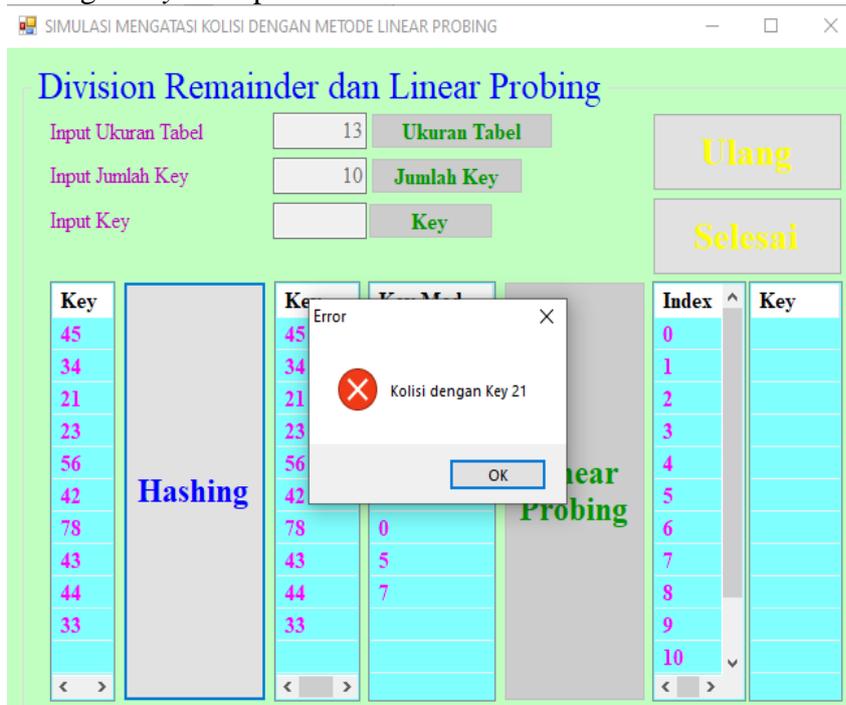
Gambar 12. Kolisi *key* 33 dengan *key* 34

Dengan menekan tombol OK, maka *Linear Probing* akan dilakukan untuk *key* 33 dengan $i-2$ untuk mendapatkan alamat alternatif baru seperti tampilan Gambar 13.



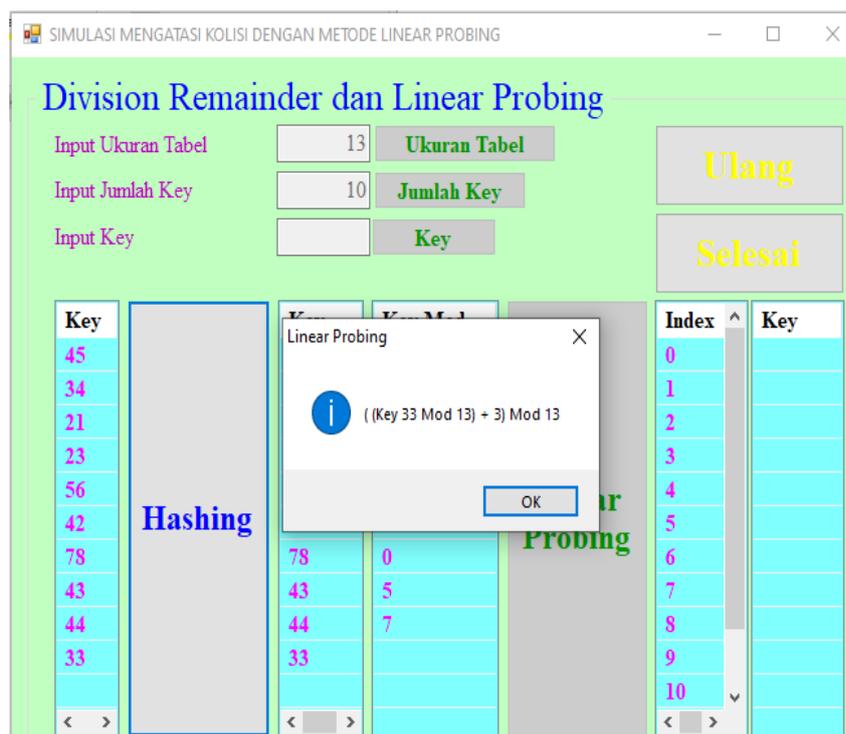
Gambar 13. *Linear Probing* *Key* 33 dengan $i=2$

Dengan menekan tombol OK, maka akan dilakukan proses *Linear Probing* key 33 dan ternyata masih kolisi dengan *key* 21 seperti Gambar 14.



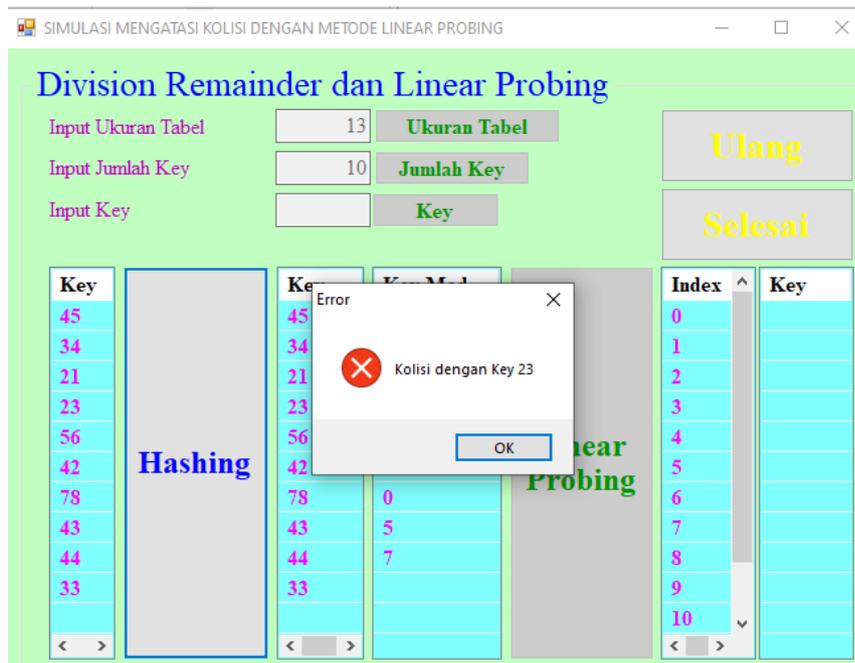
Gambar 14. Kolisi *key* 33 dengan *key* 21

Dengan menekan tombol OK, maka proses *Linear Probing* akan dilakukan terhadap *key* 33 dengan nilai indeks $i=3$ seperti Gambar 15.



Gambar 15. *Linear Probing* *key* 33 dengan $i=3$

Dengan menekan tombol OK, proses *Linear Probing* akan dilakukan terhadap *key* 33 dengan nilai index $i=3$, namun masih kolisi dengan *key* 23 seperti Gambar 16.



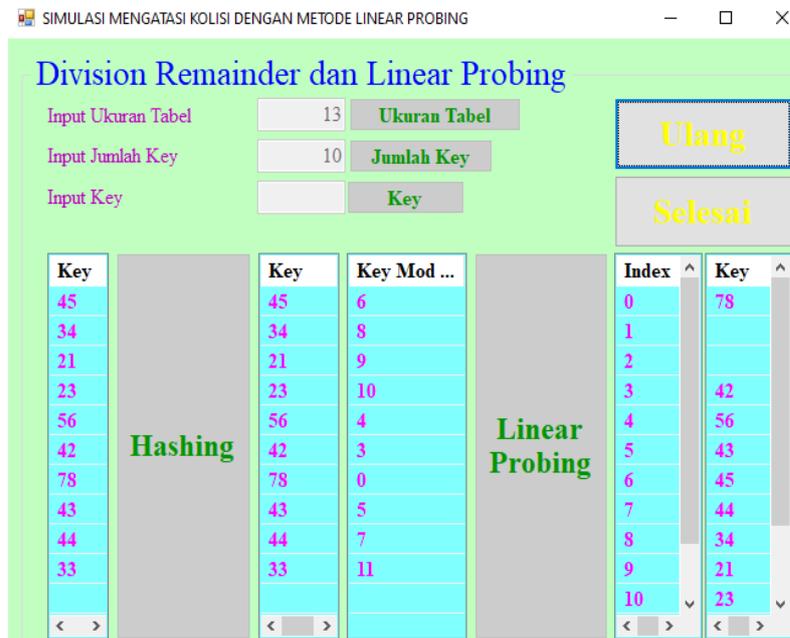
Gambar 16. Kolisi *key* 33 dengan *key* 23

Dengan menekan tombol OK, maka akan dilakukan proses *Linear Probing* *key* 33 dengan nilai $i=4$ seperti Gambar 17.



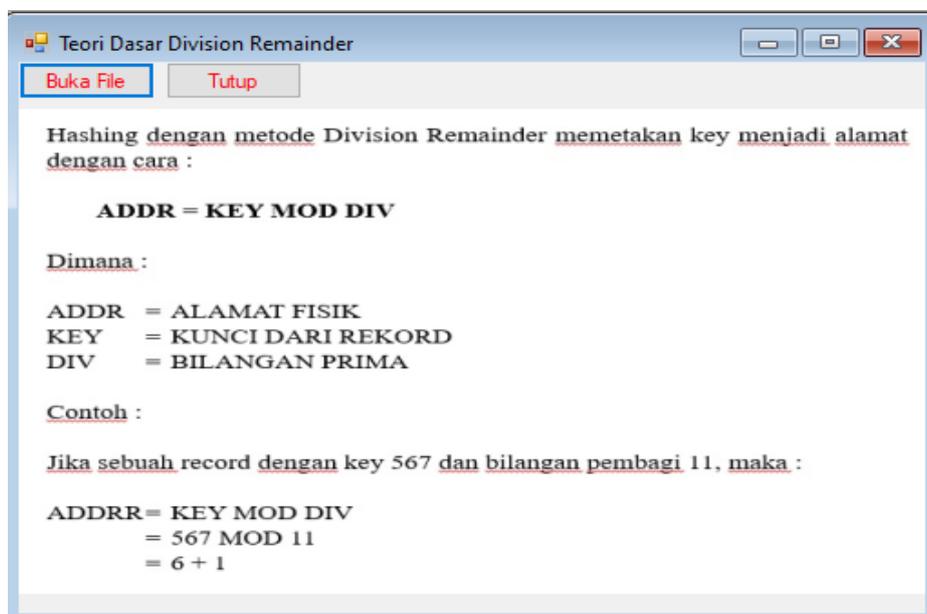
Gambar 17. *Linear Probing* *key* 33 dengan $i=4$

Dengan menekan tombol OK, maka akan dilakukan proses *Linear Probing* terhadap *key* 33 dengan nilai indeks $i=4$ untuk mendapatkan alamat alternatif baru untuk *key* 33. Proses *Linear Probing* ke 4 terhadap *key* 33 memberikan alamat ditabel yaitu index 11. Sehingga hasil akhir pemetaan nilai *key* secara keseluruhan dapat dilihat seperti Gambar 18.



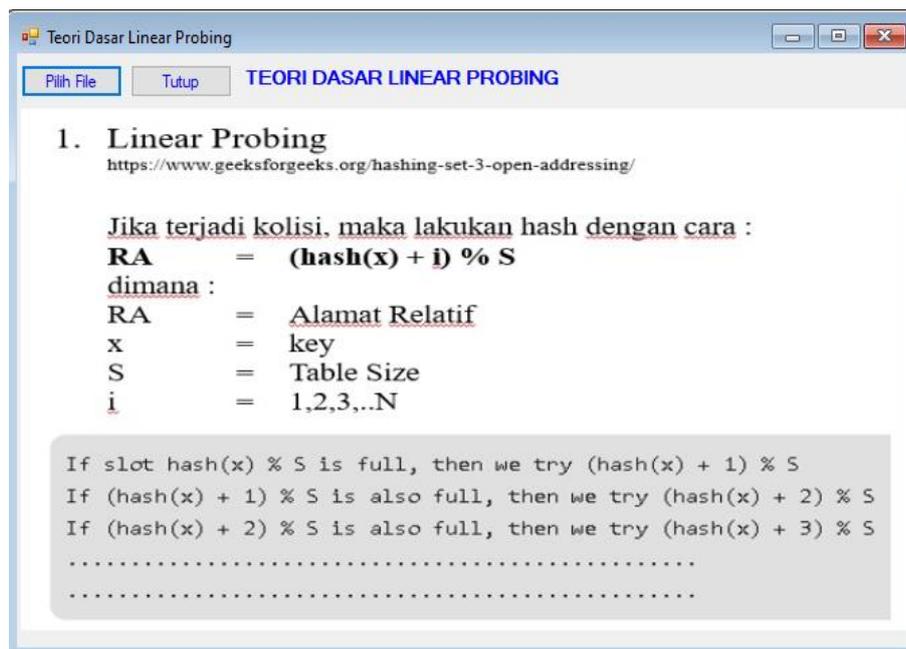
Gambar 18. Hasil Akhir Pemetaan Nilai *Key*

Untuk melihat teori singkat tentang Division Remainder dan Linear Probing, maka pengguna dapat memilih menu Referensi dan memilih teori singkat yang ingin di lihat. Jika pengguna memilih submenu Division Remainder, maka pengguna akan diarahkan ke jendela pemilihan file dari media penyimpanan yang akan ditampilkan. Berikut adalah tampilan jika submenu Division Remainder dipilih (Gambar 19).



Gambar 19. Tampilan teori Dasar Division Remainder

Demikian juga halnya untuk pilihan sub menu Linear Probing, pengguna akan diarahkan untuk memilih file yang akan ditampilkan. Berikut adalah tampilan hasil sub menu Linear Probing (Gambar 20).



Gambar 20. Tampilan sub menu Linear Probing

Untuk mengakhiri penggunaan aplikasi, maka pengguna memilih menu Keluar dan menekan tombol Yes pada jendela konfirmasi untuk memastikan pengguna mengakhiri penggunaan aplikasi

Berdasarkan hasil uji coba program simulasi yang ditampilkan diatas, maka dapat diidentifikasi kelebihan dan kekurangan dari program simulasi yang dibuat. Kelebihan program simulasi adalah:

1. Program simulasi dapat menampilkan cara kerja Metode *Division Remainder* untuk memetakan nilai key record menjadi alamat relative dengan cepat dan mudah untuk dipahami.
2. Program simulasi akan langsung menerapkan *linear Probing* jika saat pemetaan alamat terjadi kolisi
3. Program simulasi melakukan proses *Linear Probing* secara linier sampai ditemukan alamat baru yang kosong pada tabel *hash*.
4. Program simulasi dapat digunakan sebagai bahan tambahan untuk pembelajaran pemetaan alamat record pada perkuliahan system berkas.

Adapun kekurangan program simulasi adalah :

1. Program simulasi belum dilengkapi dengan fitur audio yang dapat membantu menjelaskan proses pemetaan alamat dan mengatasi kolisi yang terjadi agar lebih mudah lagi untuk dipahami.
2. Proses input data key tidak dapat dilakukan sekaligus untuk beberapa key.
3. Tampilan antarmuka program simulasi masih kurang baik.

SIMPULAN

Berdasarkan hasil pengujian program simulasi dengan ukuran tabel 13 dan jumlah *key* yang dipetakan sebanyak 10 atau dengan factor muat 0.76, maka tingkat terjadinya kolisi sangat tinggi, dimana terjadi kolisi sebanyak enam kali. Hal ini di sebabkan jumlah record yang akan dipetakan hampir mendekati ukuran tabel. Sehingga untuk menghindari kolisi dengan tingkat tinggi, maka di upayakan factor muat 0.5.

REFERENSI

- Wulandari, D.W., & Rusjdi, D. (2016). Perancangan Simulasi Media Pembelajaran dengan Metode Division Remainder untuk Pencarian Alamat Relatif pada Proses Penempatan Data. *Petir: Jurnal Pengkajian dan Penerapan Teknik Informatika*, 9(2), 157-162.
- Muharto & Ambarita, A. (2016). Metode Penelitian Sistem Informasi: Mengatasi Kesulitan Mahasiswa Dalam Menyusun Proposal Penelitian. *Deepublish*. Yogyakarta.
- Lee, C. (2014). *Buku Pintar Pemrograman Visual Basic 2010*. Elex Media Komputindo.
- Sutanta, E. (2006). Beberapa Metode Penyelesaian Collision Pada Organisasi Berkas Secara Hashing. *J. WAHANA Ilm*, 4.
- Ridho, F., & Prianto, E. (2020). Query Pemetaan Kunci Metode Linear Quotient Menggunakan Alokasi Basis Data. *Jurnal Manajemen Informatika dan Komputer*, 1(1), 10-15.
- Id/entri, <https://kbbi.kemdikbud.go.id>. (2019). *Kamus Besar Bahasa Indonesia Daring*.
- Prestiliano, J. (2007). Aplikasi Tabel Hash dalam Pengarsipan dan Pencarian Data. *J. Teknol*, 4(1), 41-56.
- Jeperson, H. (2014). Konsep Sistem Informasi. *Yogyakarta: Deepublish*.
- Muslihudin, M. (2016). *Analisis Dan Perancangan Sistem Informasi Menggunakan Model Terstruktur Dan UML*. Penerbit Andi.
- Nurseto, T. (2011). Membuat media pembelajaran yang menarik. *Jurnal Ekonomi dan pendidikan*, 8(1).
- Prianto, E., Yudhana, A., & Fadlil, A. (2016). Analisis Empiris Perbandingan Kinerja Metode Hashing Progressive Overflow dan Linear Quotient dalam Studi Pembuatan Aplikasi Dekstop Administrasi Kepegawaian. *ILKOM Jurnal Ilmiah*, 8(3), 174-181.
- Saputra, W., & Purnama, B. E. (2015). Pengembangan multimedia pembelajaran interaktif untuk mata kuliah organisasi komputer. *Speed-Sentra Penelitian Engineering dan Edukasi*, 4(2).
- Sri, W., & Sumardi, S. S. (2011). IMPLEMENTASI PERANGKAT LUNAK DENGAN PENERAPAN PENCARIAN RELATIF (HASH SEARCH). *Semantik*, 1(1).
- Mutiawani, V. (2014). Hashtable Sebagai Alternatif Dari Algoritma Pencarian Biner Pada Aplikasi E-Acesia. *Universitas Syiah Kuala, Banda Aceh*.
- Yanti, I. S. D. (2016). ANALISA METODE PENCARIAN HASH. *METHODIKA: Jurnal Teknik Informatika dan Sistem Informasi*, 2(1), 87-95.