**BASES AND RANK**



**Oleh :**

MOHAMMAD EKSA HARDIAN (23051204115)

ALIF RASYID FEBRIANSYAH (23051204131)

MU’IZUDDIN AHMAD GANI (23051204133)

**Dosen :** Ervin Yohannes, S.Kom., M.Kom., M.Sc., Ph.D.

**UNIVERSITAS NEGERI SURABAYA**

**FAKULTAS TEKNIK**

**JURUSAN TEKNIK INFORMATIKA**

**2023**

**KATA PENGANTAR**

Puji syukur kita panjatkan kehadirat Allah Swt. yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan tugas yang berjudul "Bases and rank" ini tepat pada waktunya.

Adapun tujuan dari penulisan dari laporan ini adalah untuk memenuhi tugas pada mata kuliah aljabar linear dan matriks. Selain itu, laporan ini juga bertujuan untuk menambah wawasan tentang basis dan peringkat sebuah matriks bagi para pembaca dan juga bagi penulis.

Terlebih dahulu, saya mengucapkan terima kasih kepada Bapak Ervin Yohannes, S.Kom., M.Kom., M.Sc., Ph.D., selaku Dosen aljabar linear dan matriks yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan sesuai dengan bidang studi yang saya tekuni ini.

Saya juga mengucapkan terima kasih kepada semua pihak yang tidak dapat saya sebutkan semua, terima kasih atas bantuannya sehingga sehingga saya dapat menyelesaikan tugas ini.

Kemudian, saya menyadari bahwa tugas yang saya tulis ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun kami butuhkan demi kesempurnaan laporan ini.

Surabaya, 17 Desember 2023

**BAB I**

**PENDAHULUAN**

**1.1 LATAR BELAKANG**

Sudah tidak asing yang namanya dengan matriks, hal ini telah kita pelajari sejak SMA/SMK di dalam pelajaran matematika. Matriks biasanya digunakan untuk pengaplikasikan permasalahan umum dalam kehidupan dunia maupun sistem komputer. Maka tidak heran matriks sudah tidak asing lagi dengar oleh kalangan umum. Permasalahan yang biasa ditemui biasanya berbentuk dalam sistem persamaan linear yang banyak sehingga dapat dibentuk ke dalam matriks.

Pengertian matriks itu sendiri adalah susunan bilangan, simbol, atau ekspresi yang disusun dalam baris dan kolom sehingga membentuk suatu bangun persegi. Baris dan kolom matriks akan menentukan dimensi matriks itu sendiri.

*Bases* dan *rank* dari sebuah matriks diterapkan dalam menentukan banyaknya solusi sistem persamaan linear. Salah satu penerapan perhitungan *rank* sebuah matriks, didalam teori kontrol, *rank* dari matriks daoat digunakan untuk menentukan sebuah sistem linear dapat dikontrol, atau dapat diobservasi.

Laporan ini bertujuan untuk mempretasikan program mencari nilai *rank* dan *bases* dari matriks dengan mudah dan cepat. Dengan adanya program ini, diharapkan meningkatkan efisiensi dan akurasi dala perhitungan matematika matriks, mendukung kemajuan teknologi, dan memudahkan berbagai kegiatan yang memerlukan analisis matriks dalam konteks komputasi modern.

**1.2 RUMUSAN MASALAH**

Rumusan masalah bedasarkan latar belakang dapat diurai sebagai berikut :

1. Bagaimana *basis* dan *rank* bisa terbentuk.
2. Cara apa saja yang bisa mencari nilai *basis* dan *rank*.
3. Bagaimana komputer mencari nilai *basis* dan *rank.*

**1.3 TUJUAN DAN MANFAAT**

Ada juga tujuan dan manfaat dari laporan ini sebagai berikut :

1. Mencari nilai *basis* dan *rank.*
2. Memahami cara yang dilakukan untuk mencari nilai *basis* dan *rank*.
3. Memahami cara kerja komputer dalam mencari nilai *basis* dan *rank*.

**BAB II**

**LANDASAN TEORI**

**2.1 MATRIKS**

Matriks adalah susunan angka atau objek matematika lainnya yang disusun dalam bentuk baris dan kolom sehingga membentuk suatu persegi. Sebagai contoh, matriks dibawah ini adalah matriks berukuran 2 x 3 (Karena terdiri dari 2 baris dan 3 kolom).

Setiap objek dalam matriks **A** berdimensi **n x m** sering dilambangkan dengan **An, m**, dimana nilai **n** adalah baris dan nilai **m** adalah kolom. Objek dalam matriks disebut elemen atau anggota matriks.

**2.2 RANK**

Dalam aljabar linear, peringkat atau *rank* dari seuatu matriks A adalah dimensi dari ruang vektor yang dibangun oleh kolom-kolom matriks tersebut. Hal ini berhubungan dengan banyak maksimal jumlah kolom matriks A yang saling bebas linear.

1. Menghitung *rank* dari sebuah matriks

Salah satu cara yang umum untuk menentukan peringkat adalah mengubah matriks menjadi bentuk yang lebih sederhana. Sebagai contoh, matriks A yang didefinisikan sebagai.

Dapat disusun menjadi bentuk eselon baris [tereduksi] dengan menerapkan operasi-operasi baris berikut:

Matriks terakhir yang dihasilkan memiliki dua barus tak-nol, sehingga peringkat dari matriks A adalah 2.

**2.3 BASES**

Basis atau *bases* adalah suatu ukuran tertentu yang menyatakan komponen dari sebuah vektor. Sebagai contoh himpunan vektor A sebagai berikut :

Himpunan vektor A akan dikatakan menghasilkan 3 kolom. Maka akan diketahui basis tersebut ada 3.

1. Menghitung *bases* dari sebuah matriks

Cara yang paling umum untuk menentukan basis adalah mengubah matriks menjadi bentuk yang lebih sederhana. Sama halnya seperti cara yang dilakukan untuk mencari *rank*. Sebagai contoh lain, matriks A yang didefinisikan sebagai.

Dapat disusun menjadi bentuk eselon baris [tereduksi] dengan menerapkan operasi-operasi baris berikut:

Matriks akhir akan terbentuk seperti ini, di contoh ini matriks berbentuk identitas yang dapat mempermudah menentukan basis dari sebuah matriks. Basis terdapat pada kolom 1, 2, dan 3. Maka masing-masing vektor dari basis akan ditentukan sebagai berikut :

**2.4 KESIMPULAN**

Dapat disimpulkan dari landasan teori. Cara yang paling efektif untuk mencari nilai *bases* dan *rank* adalah dengan cara penyederhanaan matriks yang biasa disebut juga dengan reduksi baris (*row reduction*).

**BAB III**

**PEMBAHASAN**

**3.1 KODE PROGRAM**

#include <iostream>

using namespace std;

int main() {

    int n, m;

    cout << "Masukan dimensi sebuah matriks : ";

    cin >> n >> m;

    //membuat matriks

    float a[n][m];

    float vektor[m][n];

    for(int i = 0; i < n; i++) {

        for(int j = 0; j < m; j++) {

            cin >> a[i][j];

            vektor[j][i] = a[i][j];

        }

    }

    //swap nol

    int countNol[n];

    for(int i = 0; i < n; i++) {

        countNol[i] = 0;

    }

    for(int i = 0; i < n; i++) {

        for(int j = 0; j < m; j++) {

            if(a[i][j] != 0) {

                break;

            }

            else {

                countNol[i]++;

            }

        }

    }

    for(int i = 0; i < n - 1; i++) {

        for(int j = 0; j < n - i - 1; j++) {

            if(countNol[j] > countNol[j + 1]) {

                swap(countNol[j], countNol[j + 1]);

                for(int k = 0; k < n; k++) {

                    swap(a[j][k], a[j + 1][k]);

                }

            }

        }

    }

    //gaussjordan

    for(int i = 0; i < n; i++) {

        float bagi = a[i][i];

        //agar diagonal matriks 1

        for(int j = 0; j < m; j++) {

            a[i][j] /= bagi;

        }

        for(int k = i + 1; k < n; k++) {

            float kali = a[k][i];

            for(int j = 0; j < m; j++) {

                a[k][j] -= kali \* a[i][j];

            }

        }

    }

    //rank

    int rank = n;

    for(int i = 0; i < n; i++) {

        bool cekRank = true;

        for(int j = 0; j < m; j++) {

            if(a[i][j] != 0) {

                cekRank = false;

            }

        }

        if(cekRank) {

            rank--;

        }

    }

    //bases

    int bases[n];

    for(int i = 0; i < n; i++) {

        bases[i] = -1;

    }

    for(int i = 0; i < n; i++) {

        bool cekBases = false;

        for(int j = 0; j < m; j++) {

            if(a[i][j] == 1) {

                bases[i] = j;

                break;

            }

        }

    }

    int banyakBases = 0;

    for(int i = 0; i < n; i++) {

        if(bases[i] != -1) {

            banyakBases++;

        }

    }

    cout << "Rank dari sebuah matriks adalah : " << rank << '\n';

    cout << "Terdapat " << banyakBases << " basis dari sebuah matriks" << '\n';

    int banyak = 1;

    for(int i = 0; i < n; i++) {

        if(bases[i] != -1) {

            cout << "v" << banyak << " :\n";

            for(int j = 0; j < m; j++) {

                cout << vektor[i][j] << '\n';

            }

            banyak++;

        }

    }

}

**3.2 TAMPILAN PROGRAM**

Masukan dimensi sebuah matriks : 3 3

1 2 1

-2 3 0

3 0 3

Rank dari sebuah matriks adalah : 3

Terdapat 3 basis dari sebuah matriks

v1 :

1

-2

3

v2 :

2

3

0

v3 :

1

0

3

**3.3 DESKRIPSI PROGRAM**

Program ini dibuat menggunakan bahasa C++ dengan software *Visual Basic Studio*. Dalam pemograman ini terdapat nilai *rank*, banyak *bases*, dan masing-masing vektor dari *bases*. Berikut adalah penjelasan program yang lebih rinci :

1. Library

*Library* adalah kumpulan kode yang menyediakan fungsionalitas tertentu untuk digunakan dalam aplikasi lain. Dalam C++, *library* biasanya memuat masing-masing fungsi untuk *input*, *output,* operasi matematika, dan masih banyak lagi. *Library* harus dipanggil menggunakan #*include* dan dibuka dengan tanda kurang dari (<) dilanjut dengan nama dari *library* dan ditutup dengan tanda lebih dari (>).

Program ini hanya memuat 1 *library* yaitu *iostream.*

#include <iostream>

using namespace std;

Guna *iostream* ini adalah untuk memudahkan dalam *input* dan *output* dalam suatu program. *Iostream* juga dibantu dengan *using namespace std;* dengan tujuan untuk mempermudah dalam pemanggilan *class* yang ada didalam *iostream.* Program yang digunakan dalam *iostream* adalah *cin* untuk input dan *cout* untuk output.

1. Fungsi *main*

Didalam fungsi *main* ada beberapa deklarasi yang harus dilakukan sebelum melakukan program.

1. Deklarasi

Sebuah program pasti memiliki variabel yang harus dideklarasi, ketika variabel tidak dideklarasi maka akan menghasilkan *error*. Ada beberapa deklarasi yang dibuat dalam program ini. *Int (interger)* digunakan untuk mendeklarasi bilangan bulat, *float* digunakan untuk mendeklarasi bilangan desimal.

Variabel yang dideklarasi juga bisa berupa *array*. *Array* dapat ditambahkan ke variabel dengan cara menambahkan kurung kotak setelah variabel. *Array* 2D juga dapat dibuat dengan cara menambahkan 2 kurung kotak, hal ini yang membuat *array* 2D tersebut matriks. Didalam kurung kotak terdapat variabel / angka yang dapat menentukan baris dan kolom.

1. *Input*

Setelah variabel di deklarasi maka variabel tersebut akan bisa diisi sebuah nilai. Nilai tersebut dapat berupa konstanta atau dapat diisi dengan pengguna program. Yang berperan sebagai input di program ini adalah *cin*. Input tersebut akan memasukan nilai variabel yang sesuai dengan kita mau. Format nya adalah *cin* lalu diikuti dengan 2 tanda lebih dari (>) lalu memasukan variabel yang mau kita isi.

1. *For loop*

*For loop* banyak digunakan didalam program ini yang berfungsi untuk mengakses masing-masing baris dan kolom dari sebuah *array* 2D yang sudah di deklarasi. *For loop* akan dilakukan ketika variabel yang dideklarasi memenuhi syarat dari program tersebut. Format dari *for loop* itu sendiri adalah *for* lalu buka dengan kurung, terdapat 3 bagian dalam *for*. Bagian pertama adalah deklarasi sebuah variabel, bagian kedua adalah syarat suatu variabel, bagian ketiga adalah operasi varibel. Ketika variabel tidak di operasikan maka akan terjadi *error* yang dimana *loop* akan terjadi terus menerus tanpa henti.

1. Kondisi *if*

Kondisi *if* digunakan ketika kondisi variabel yang sudah ditentukan benar atau salah. Di program ini kondisi *if* digunakan untuk mengecek nilai yang dicari benar atau salah. Kondisi *if* ditulis dengan *if* itu sendiri, dibuka dengan kurung lalu

1. *Boolean*

*Boolean* singkatnya adalah benar atau tidak. Ketika suatu program setelah dijalankan pasti akan menghasilkan jawaban benar atau tidak. Di program ini *boolean* digunakan untuk mengecek apakah program tersebut bertemu dengan syarat yang telah ditentukan. *Boolean* biasa ditulis *bool* untuk mendeklarasi sebuah variabel agar program tersebut.

1. Fungsi program

Program tanpa adanya fungsi maka program itu tidak bisa berjalan. Berikut fungsi progra yang dibuat :

1. *Swap* nol

//swap nol

    int countNol[n];

    for(int i = 0; i < n; i++) {

        countNol[i] = 0;

    }

    for(int i = 0; i < n; i++) {

        for(int j = 0; j < m; j++) {

            if(a[i][j] != 0) {

                break;

            }

            else {

                countNol[i]++;

            }

        }

    }

    for(int i = 0; i < n - 1; i++) {

        for(int j = 0; j < n - i - 1; j++) {

            if(countNol[j] > countNol[j + 1]) {

                swap(countNol[j], countNol[j + 1]);

                for(int k = 0; k < n; k++) {

                    swap(a[j][k], a[j + 1][k]);

                }

            }

        }

    }

Fungsi *swap* nol ini adalah memindahkan baris yang memiliki nol lebih ke bawah, sehingga matriks akhir yang dibentuk adalah matriks segitiga atas.

Hal ini akan mempermudah program dalam mencari nilai *rank* dan *bases*.

1. *Row Reduction*

    //rowreduction

    for(int i = 0; i < n; i++) {

        float bagi = a[i][i];

        //agar diagonal matriks 1

        for(int j = 0; j < m; j++) {

            a[i][j] /= bagi;

        }

        for(int k = i + 1; k < n; k++) {

            float kali = a[k][i];

            for(int j = 0; j < m; j++) {

                a[k][j] -= kali \* a[i][j];

            }

        }

    }

*Row reduction* (reduksi baris), inti dari program ini. Reduksi baris berperan sanagat penting dalam mencari nilai *rank* dan *basis.* Cara kerjanya dengan menjadikan baris yang dibawah nya menjadi nol. Sehingga hasil akhir matriks adalah segitiga atas.

1. *Rank*

    //rank

    int rank = n;

    for(int i = 0; i < n; i++) {

        bool cekRank = true;

        for(int j = 0; j < m; j++) {

            if(a[i][j] != 0) {

                cekRank = false;

            }

        }

        if(cekRank) {

            rank--;

        }

    }

Fungsi ini bertugas untuk mencari nilai *rank* dari sebuah matriks. *Rank* awalnya adalah banyak baris tersebut, tetapi jika bertemu baris yang berisikan nilai nol. Maka *rank* akan berkurang nilai nya.

1. *Bases*

    //bases

    int bases[n];

    for(int i = 0; i < n; i++) {

        bases[i] = -1;

    }

    for(int i = 0; i < n; i++) {

        bool cekBases = false;

        for(int j = 0; j < m; j++) {

            if(a[i][j] == 1) {

                bases[i] = j;

                break;

            }

        }

    }

    int banyakBases = 0;

    for(int i = 0; i < n; i++) {

        if(bases[i] != -1) {

            banyakBases++;

        }

    }

Fungsi ini bertugas sebagai mencari *bases* serta posisi vektor yang akan di *output*. Fungsi ini akan mencari nilai 1 disetiap baris yang akan menghitung basis dari matriks tersebut.