BAB II

MATERI PENUNJANG

Kajian ilmiah dalam Kriptografi dibangun atas suatu konsep dasar, sebelum menjelaskan tentang Kriptografi dan Metode GOST (*Gosudarsvennyi Standard*) yang akan dibahas pada bab III perlu diketahui terlebih dahulu beberapa hal yang mendasari keduanya, yaitu konsep matematika.

2.1. Teori Fungsi

2.1.1. Fungsi

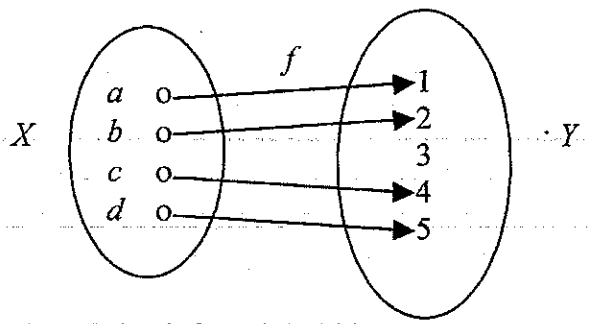
Sebuah himpunan berisi obyek yang berbeda dengan syarat keanggotaan yang jelas maka obyek tersebut dinamakan anggota atau elemen dari himpunan.

Definisi 2.1

Sebuah fungsi didefinisikan sebagai dua buah himpunan X , Y dan sebuah aturan f yang memasangkan setiap elemen X dengan tepat satu elemen pada Y.

Himpunan X disebut dengan domain fungsi dan himpunan Y sebagai kodomain. Fungsi dapat pula dinyatakan dalam bentuk diagram fungsi, setiap elemen pada domain X memiliki tepat satu garis anak panah yang berasal dari elemen tersebut. Setiap elemen pada kodomain Y bisa memiliki beberapa garis anak panah yang menuju padanya termasuk garis kosong.

Contoh 2.1



Gambar 2.1. Sebuah fungsi dari himpunan X terhadap himpunan Y

$X = \{a, b, c, d\}$, $Y = \{1, 2, 3, 4, 5)$ dengan aturan f didefinisikan $f(a) = 1$, $f(b) = 2$, $f(c) = 4$ dan $f(d) = 5$

Definisi 2.2

Jika $x \in X$ maka bayangan dari x adalah elemen di Y sesuai dengan aturan f, sedemikian sehingga $y = f(x)$.

Hal tersebut ekivalen : anggota himpunan Y yang paling sedikit memiliki satu pra bayangan adalah $\text{Im}(f)$.

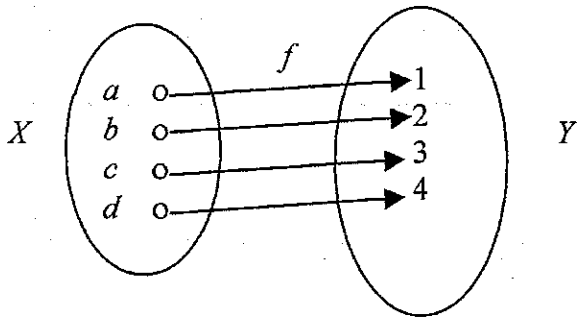
Contoh 2.2

$X = \{a, b, c, d\}$, $Y = \{1, 2, 3, 4, 5\}$, dan aturan f didefinisikan $f(a) = 1$, $f(b) = 2$, $f(c) = 4$, $f(d) = 5$ maka pra bayangan dari elemen 1 adalah a dan $\text{Im}(f)$ adalah $\{1, 2, 4, 5\}$.

Definisi 2.3

Sebuah fungsi $f : X \rightarrow Y$ dikatakan bijeksi jika setiap elemen pada kodomain Y merupakan bayangan dari tepat satu elemen pada X.

Contoh 2.3



Gambar 2.2. Sebuah fungsi bijeksi dari himpunan X terhadap himpunan Y

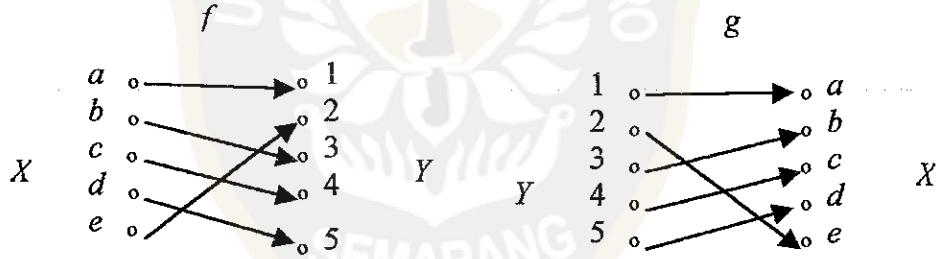
$X = \{a, b, c, d\}$, $Y = \{1, 2, 3, 4)$ dengan aturan f didefinisikan $f(a) = 1$, $f(b) = 2$, $f(c) = 3$ dan $f(d) = 4$

Definisi 2.4

Jika f adalah fungsi bijeksi dari X terhadap Y maka dapat didefinisikan sebuah fungsi g dari Y ke X sebagai berikut : untuk setiap $y \in Y$ *didefinisikan fungsi* $g(y) = x$ *dengan* $x \in X$ *dan* $f(x) = y$. *Fungsi g ini diperoleh dari fungsi f yang disebut fungsi invers dari f .*

Contoh 2.4

$X = \{a, b, c, d, e\}$, dan $Y = \{1, 2, 3, 4, 5\}$ dan aturan f diberikan sesuai dengan garis anak panah pada *Gambar 2.3*. Dengan melihatnya maka diketahui bahwa f adalah fungsi bijeksi dan fungsi inversnya yaitu g dibentuk dengan membalik garis anak panah pada fungsi f.



Gambar 2.3. Sebuah fungsi bijeksi f dan inversnya $g = f^{-1}$

Jika fungsi f merupakan fungsi bijeksi maka inversnya (f^{-1}) juga merupakan fungsi bijeksi. "*Dalam kriptografi fungsi bijeksi digunakan sebagai alat untuk mengenkripsi pesan plainteks dan fungsi invers digunakan untuk mendekripsi sandi* ".

2.1.2. Permutasi

Permutasi merupakan fungsi yang sering digunakan dalam berbagai bentuk metode kriptografi, karena merupakan pemetaan suatu himpunan terhadap dirinya sendiri.

Definisi 2.5

S adalah sebuah himpunan berhingga, permutasi p pada S adalah fungsi bijeksi dari S terhadap dirinya sendiri ($p : S \rightarrow S$)

Contoh 2.5

Himpunan $S = \{1,2,3,4,5\}$. Sebuah permutasi $p : S \rightarrow S$ didefinisikan sebagai berikut : $p(1) = 5, p(2) = 3, p(3) = 4, p(4) = 1, p(5) = 2$.

sebuah permutasi dapat dideskripsikan dengan berbagai cara, selain seperti dinyatakan di atas dapat dinyatakan pula sebagai sebuah array berikut ini :

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 4 & 1 & 2 \end{pmatrix}$$

baris pertama dalam array merupakan domain sedangkan baris kedua merupakan bayangan sesuai dengan pemetaan oleh p.

Oleh karena sebuah permutasi merupakan fungsi bijeksi maka memiliki invers. Untuk mengetahui inversnya yaitu dengan cara menukarkan baris-baris dalam array dan menyusun elemen-elemen baris pertama yang baru disesuaikan dengan pasangannya. Sehingga invers dari p pada contoh di atas adalah :

$$p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 2 & 3 & 1 \end{pmatrix}$$

2.2. Teori Bilangan

2.2.1. Integer

Himpunan dari semua bilangan bulat {...,-3,-2,-1,0,1,2,3,...} dinyatakan dengan simbol Z_n dan dinamakan himpunan bilangan integer.

2.2.2. Integer Modulo n

Diambil n adalah sebuah integer positif

Definisi 2.6

Jika a dan b merupakan integer, maka a dikatakan ***kongruen*** *terhadap b modulo n, ditulis* $a \equiv b \pmod{n}$ *jika integer n membagi integer (a-b). Integer n disebut modulus kongruensi.*

Contoh 2.6

(i) $24 \equiv 9 \pmod{5}$ karena $24 - 9 = 3 \cdot 5$

(ii) $-13 \equiv 17 \pmod{5}$ karena $-13-17 = -6 \cdot 5$

Definisi 2.7

Integer modulo n, ditulis Z_n *adalah himpunan yang anggota-anggotanya adalah integer* $\{0,1,2,3,\ldots,n-1\}$. *Operasi penjumlahan, pengurangan dan perkalian dalam* Z_n *dibentuk oleh modulo n.*

Contoh 2.7

$Z_{25} = \{0,1,2,3,\ldots,24\}$. Dalam Z_{25}, $23 + 16 = 14$, karena $23 + 16 = 39 \equiv 14 \pmod{25}$. Begitu pula untuk operasi perkalian $23 \cdot 16 = 18$ dalam Z_{25}.

Definisi 2.8

$a \in Z_n$, *invers perkalian dari a modulo n adalah integer* $x \in Z_n$ *sedemikian sehingga* $ax \equiv 1 \pmod{n}$. *Jika x tersebut ada, maka a dikatakan dapat diinverskan, invers dari a dituliskan* a^{-1}.

Contoh 2.8

Elemen yang dapat diinverskan dalam Z_9 adalah 1, 2, 4, 5, 7, dan 8.

Misalnya $4^{-1} = 7$, karena $4 \cdot 7 \equiv 1 \pmod 9$

Definisi 2.9

$a,b \in Z_n$, pembagian a oleh b modulo n adalah hasil kali dari a dan b^{-1} modulo n, dan hanya didefinisikan jika b dapat diinverskan modulo n.

Misal $a \in Z_n$, maka a dapat diinverskan jika dan hanya jika $\gcd(a,n) = 1$.

2.2.3. Bilangan Biner

Definisi 2.10

Basis atau radix suatu sistem bilangan adalah menyatakan banyaknya lambang yang dipergunakan dalam sistem bilangan tersebut.

$L_n = \{l_1, l_2, l_3, \ldots, l_n\}$ adalah himpunan lambang yang digunakan dalam sistem bilangan basis n.

Contoh 2.9

Sistem bilangan desimal mempunyai basis atau radix sepuluh (10) karena sistem ini menggunakan 10 lambang angka, yaitu: $L_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9.\}$

Untuk menyajikan notasi lambang bilangan yang lebih dari basisnya maka dibentuk kombinasi lambang dari lambang dasar dalam sistem bilangan.

$l_2l_1, l_2l_2, \ldots l_2ln, l_3l_1, l_3l_2, l_3l_3, \ldots l_2l_n, \ldots l_nl_n, l_2l_1l_1, l_2l_1l_2, \ldots$

Definisi 2.11

Bilangan biner ialah sistem bilangan yang mempunyai basis atau radix dua, dengan menggunakan lambang 0 dan 1. Dan dinotasikan dengan β.

Dalam sistem bilangan biner digunakan pendekatan yang sama dalam menyajikan lambang yang lebih dari dua. Setelah mencapai lambang 1, maka akan kehabisan lambang biner (tidak ada 2, 3, ... dalam sistem bilangan biner). Untuk menyatakan nilai dua, gunakan angka biner kedua diikuti oleh angka biner pertama untuk mendapatkan 10. Untuk menyatakan nilai tiga, gunakan 11. Oleh karenanya dalam biner dicacah sebagai berikut: 0, 1, 10, 11. Untuk menghindari kekacauan dengan bilangan-bilangan desimal, ada baiknya untuk membaca bilangan-bilangan biner ini dengan nol, satu, satu-nol, dan satu-satu.

Komputer digital merepresentasikan semua datanya dalam bentuk biner, hal ini disebabkan mesin komputer secara umum hanya mengenal bahasa dari sinyal-sinyal elektrik yang diterimanya. Representasi biner merupakan representasi paling sederhana yang memudahkan mekanisme penyimpanan data karena terdiri dari dua nilai : nol atau satu (on/off, in/out, ya/tidak, benar atau salah).

Representasi biner merupakan bentuk khusus dari representasi basis secara umum, maka sebuah bilangan biner dapat direpresentasikan sebagai :

$j_{n-1}j_{n-2}\cdots j_1j_0$

dengan j_i adalah bilangan 0 atau 1, sesuai dengan pengertian bilangan yaitu :

$$\sum_{i=0}^{n-1} j_i * 2^i = \begin{cases} j_{n-1} * 2^{n-1} + \\ j_{n-2} * 2^{n-2} + \\ \cdots + \\ \cdots + \\ j_1 * 2^1 + \\ j_0 * 2^0 \end{cases}$$

Contoh 2.10

bilangan 10101_2 dalam basis 10 adalah :

$1*2^4$ +
$0*2^3$ +
$1*2^2$ +
$0*2^1$ +
$1*2^0$

atau dapat ditulis pula sebagai berikut :

$1*2^4$ +
$1*2^2$ +
$1*2^0$ = $16 + 4 + 1 = 21_{10}$

2.2.4. Bilangan Heksadesimal

Sistem bilangan heksadesimal mempunyai basis 16. Walaupun dapat digunakan 16 buah lambang yang manapun, namun lazimnya digunakan 0 sampai 9 dan A sampai F seperti terlihat pada tabel 2.1. Setelah mencapai 9 pada sistem heksadesimal, pencacahan dilanjutkan sebagai berikut:

A, B, C, D, E, F

Setelah kehabisan lambang dasar, dibentuk kombinasi 2 angka, dengan mengambil angka kedua diikuti oleh angka pertama, kemudian kedua diikuti oleh angka kedua, dan seterusnya. Maka bilangan berikutnya setelah F dalam heksadesimal adalah 10.

Kemudian dikuti oleh 11, 12, 13, 14, 15, 16, 17, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, dan seterusnya.

Salah satu bidang pengembangan yang paling luas dewasa ini adalah mikrokomputer. Pada saat memprogram, manganalisa, maupun memeriksa sebuah

komputer, akan dibutuhkan bilangan heksadesimal. Disamping itu juga harus menguasai perubahan-perubahan heksadesimal sebagai berikut.

Tabel 2.1. Konversi desimal, biner dan heksadesimal

n	Biner	hexadesimal	desimal
	0	0	0
0	1	1	1
1	10	2	2
2	100	4	4
3	1000	8	8
4	10000	10	16
5	100000	20	32
6	1000000	40	64
7	10000000	80	128
8	100000000	100	256
9	1000000000	200	512
10	10000000000	400	1024
11	100000000000	800	2048
12	1000000000000	1000	4096
13	10000000000000	2000	8192
14	100000000000000	4000	16384
15	1000000000000000	8000	32768
16	10000000000000000	10000	65536

2.2.5. Konversi Antar Basis

Komputer akan melakukan proses perhitungan dalam bentuk biner akan tetapi dari sudut pandang pengguna komputer mungkin melakukan proses perhitungan dalam bentuk desimal. Oleh sebab itu dalam komputer terdapat mekanisme konversi bilangan yang memudahkan komunikasi antara komputer digital dengan pengguna komputer tersebut. Usaha mengkonversi bilangan tersebut dapat dilakukan dengan metode pembagian yang berdasarkan pada aritmatika modular, proses pembagian tersebut bersifat rekursif hal ini dapat dinyatakan dalam algoritma berikut ini :

* n adalah sebuah bilangan (sesuai dengan basis asal , misalnya basis 10)

* Jika (n > 0) maka lakukan proses berikut :

  1. Lakukan pembagian pada n oleh basis yang dinginkan (basis 2)
  2. Hasilnya adalah sisa pembagian : (n mod 2)
  3. n⇐Hasil bagi : (n Div 2), selanjutnya proses berulang sampai diperoleh bilangan yang dimaksud dalam konversi tersebut.

Contoh 2.11

Akan dikonversikan bilangan 100_{10} terhadap basis 2 :

Perhitungan	Output
N⇐100	
	(100 mod 2)⇒0
N⇐(100 Div 2) = 50	
	(50 mod 2) ⇒0
N⇐(50 Div 2) = 25	
	(25 mod 2) ⇒1
N⇐(25 Div 2) = 12	
	(12 mod 2) ⇒0
N⇐(12 Div 2) = 6	
	(6 mod 2) ⇒0
N⇐(6 Div 2) = 3	
	(3 mod 2) ⇒1
N⇐(3 Div 2) = 1	
	(1 mod 2) ⇒1
N⇐(1 Div 2) = 0	

Berdasarkan algoritma rekursif ini, maka hasil konversi binernya adalah 1100100_2 , dengan penulisan terbalik yaitu dari hasil output paling bawah sampai pada output paling atas.

2.2.6. Bit *(Binary digit)*

Definisi 2.12

Bit ialah jumlah dari angka biner dalam satu kesatuan. Kata ini merupakan singkatan dari binary digit (angka biner).

Contoh 2.12

Bilangan biner seperti 1101 mempunyai 4 angka biner, atau 4 bit. Bilangan 111010 mempunyai 6 angka biner, atau 6 bit.

Bit Paritas

Dalam sistem digital *kata (word)* adalah sekelompok bit yang diperlakukan, disimpan, dan dipindahkan sebagai suatu kesatuan. Sebagai contoh, misalkan sebuah komputer akan menambahkan 0101 1000 0011 dan 0010 0100 0110. (Ini adalah 586 dan 246) Masing-masing bilangan ini merupakan sebuah kata. Komputer membawa masing-masing kata ini keluar dari memori dan meletakkannya ke dalam suatu aritmatika. Jumlahnya merupakan sebuah kata baru yang kemudian diletakkan kembali ke dalam memori.

Pada saat kata-kata sedang dipindahkan dan disimpan, kesalahan dapat masuk ke dalam kata-kata tersebut. Sebagai contoh, salah satu 0 dalam suatu kata secara tidak disengaja mungkin berubah menjadi 1 akibat pemutusan sesaat, akibat derau, akibat peralihan, dan sebagainya. Dibawah kondisi operasi normalnya perubahan semacam itu tidak mungkin terjadi, namun suatu kesalahan dapat merusakkan segalanya. Oleh karenanya, dibutuhkan metode-metode untuk mendeteksi kesalahan yang timbul pada saat suatu kata sedang dipindahkan atau disimpan.

Pendekatan yang paling luas penggunaannya untuk mendeteksi kesalahan yang timbul selama penyimpanaan dan pemindahan kata-kata adalah membubuhkan suatu *bit paritas* pada kata tersebut.

Definisi 2.13

Paritas genap ialah bit paritas tambahan yang membuat banyaknya bit 1 menjadi genap pada sekelompok bit.

Definisi 2.14

Paritas ganjil ialah bit paritas tambahan yang membuat banyaknya bit 1 menjadi ganjil pada sekelompok bit.

Contoh 2.13

Misalkan diberikan sebuah kata seperti 0111. Terdapat tiga buah 1 dalam kata ini, banyaknya 1 yang ganjil. Kemudian dibubuhkan sebuah 1 tambahan pada kata tersebut untuk mendapatkan 01111. Sekarang banyaknya 1 menjadi genap. Kata baru ini dapat dipindahkan dan disimpan oleh komputer, dan dapat diperiksa paritas genapnya pada berbagai titik untuk menyakinkan bahwa tidak ada kesalahan yang telah memasuki kata tersebut.

Sebagai gambaran lain bagi paritas genap, Tabel 2.2. memperlihatkan sekelompok bit dengan sebuah bit paritas genap dan paritas ganjil.

Tabel 2.2. Bit paritas

Sekelompok bit	Bit tambahan	
	Paritas genap	Paritas ganjil
0000	0	1
0001	1	0
0010	1	0
0011	0	1
0100	1	0
0101	0	1
0110	0	1
0111	1	0
1000	1	0
1001	0	1
1010	0	1
1011	1	0

1100	0	1
1101	1	0
1110	1	0
1111	0	1

Kedua jenis paritas ini biasa digunakan, dan tidak ada alasan yang kuat untuk lebih memilih salah satu jenis.

Pengguanaan bit paritas untuk mendeteksi kesalahan berdasarkan dua asumsi yang berlaku pada kebanyakan sistem digital:

- Probabilitas kesalahan sangat kecil.

  Dalam hal suatu kesalahan masuk ke dalam sebuah kata, kesalahan tersebut hampir dapat dipastikan merupakan kesalahan 1-bit. Kemungkinan 2 atau lebih bit berubah secara tak disengaja sangatlah kecil kecuali jika terjadi gangguan total, yang akan terdeteksi oleh sarana lain. Dengan kata lain pemeriksaan paritas akan menangkap semua kesalahan 1-bit namun tidak menangkap kesalahan ganda. Dalam kebanyakan sistem digital kemungkinan terjadi kesalahan ganda sangat kecil.

- Pemeriksaan paritas khususnya biasa dilakukan dalam piranti penyimpanan seperti disket, pita, memori inti magnetis.

2.2.7. Ukuran Bilangan

Sistem penyimpanan data dalam komputer digital adalah dalam satuan bit atau byte (1 byte sama dengan 8 bit). Logaritma adalah pengukuran terhadap besarnya sebuah bilangan, pemakaian logaritma basis 10 atau logaritma basis e (natural) lebih dikenal bagi kebanyakan orang. Tetapi dalam ilmu komputer penggunaan logaritma basis 2 merupakan hal yang mendasar, karena sebuah logaritma adalah ukuran terhadap jumlah digit yang diperlukan oleh komputer

untuk merepresentasikan sebuah nilai dari basis tertentu. Berikut ini adalah tabel yang memperlihatkan bilangan yang dapat diwakili oleh digit biner :

Digit	Biner	Ekivalen dalam desimal
1	$1=2^1-1$	1
2	$11=2^2-1$	3
3	$111=2^3-1$	7
4	$1111=2^4-1$	15
5	$11111=2^5-1$	31
6	$111111=2^6-1$	63

Bilangan yang cukup penting dalam satuan penyimpanan data diantaranya adalah 2^{10} = 1024, oleh karena bilangan tersebut mendekati 1000, para pakar komputer sepakat bilangan tersebut sebagai satuan Kilobit disingkat K, misalnya data/file komputer dengan kapasitas 2048 bit dapat ditulis 2K.

2.3. Operasi Logika

Operasi logika digunakan oleh komputer terutama pada bahasa pemrograman tingkat tinggi misalnya Pascal, Basic, C dan sebagainya. Operasi logika berguna dalam menentukan proses-proses yang dilakukan oleh komputer digital yaitu untuk pengetesan kondisi sehingga prosessor komputer dapat mengeksekusi suatu program aplikasi atau membatalkan suatu perintah eksekusi. Berikut ini adalah tabel dari operasi logika AND, OR, XOR dan NOT yang biasa dipakai dalam proses enkripsi data :

Tabel 2.3. Operasi logika AND

P	Q	P ∧ Q
1	1	1
1	0	0
0	1	0
0	0	0

Tabel 2.4. Operasi logika OR

P	Q	P ∨ Q
1	1	1
1	0	1
0	1	1
0	0	0

Tabel 2.5. Operasi logika NOT

P	~ P
1	0
0	1

Tabel 2.6. Operasi logika X-OR

P	Q	P ⊕ Q
1	1	0
1	0	1
0	1	1
0	0	0

2.4. File

Definisi 2.15

Item atau elemen data ialah nilai dari suatu data yang mempunyai tipe tertentu, seperti integer, real, karakter, atau tipe yang terstruktur.

Definisi 2.16

Record adalah struktur data yang tersusun dari satu atau lebih item (elemen data) yang mewakili suatu data.

Definisi 2.17

File adalah kumpulan dari berkas (record) yang mempunyai hubungan satu sama lain.

Contoh 2.14

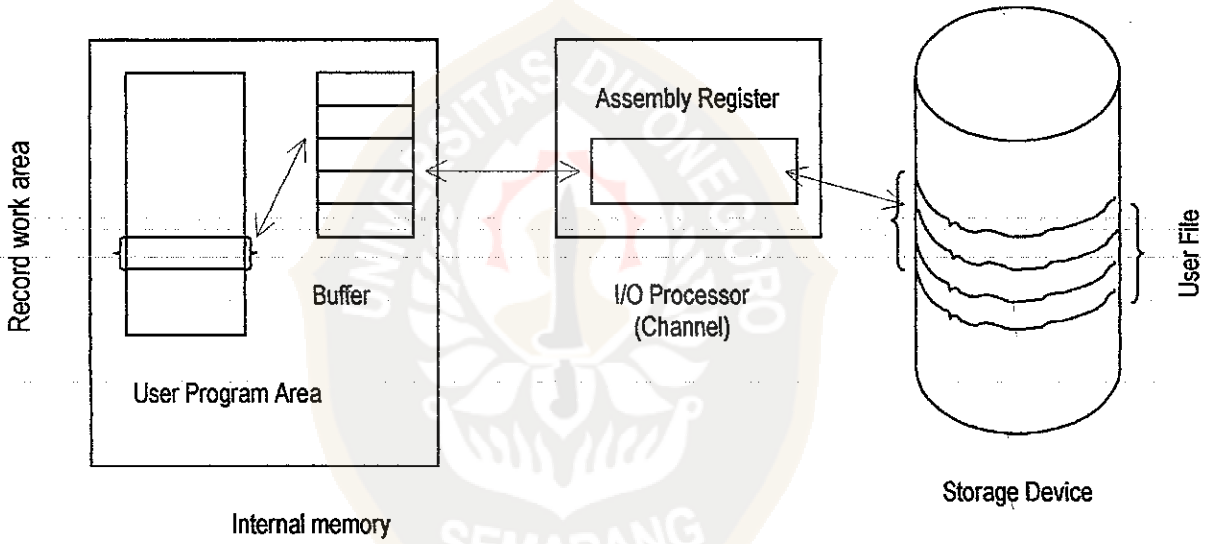
Kartu Mahasiswa bisa dikatakan sebuah record yang terdiri dari item-item: Nama, Nim, Jurusan, Alamat, Golongan darah. Sedangkan File yang menghimpunannya adalah data mahasiswa pada suatu Fakultas.

Penggunaan komputer selalu terkait dengan data-data yang bisa dibaca dari beberapa media antara lain media kartu, terminal komputer interaktif, media penyimpanan eksternal (misalnya disket, CD dan Hard disk), dan data tersebut dapat juga dibangkitkan oleh suatu program. Ukuran data yang cukup besar biasanya disimpan pada memori eksternal, data tersebut dinamakan dengan file.

Sebuah alat penyimpanan eksternal terdiri dari bagian drive dan media penyimpan. Bagian drive berupa perangkat keras misalnya disk drive komputer,

sedangkan media penyimpan adalah material tempat data atau file tersebut disimpan misalnya pita magnetik, disket , CD-compact disc, dan hard disk.

Dalam sudut pandang pengguna komputer, file merupakan bagian pertukaran antara memori internal dengan media penyimpanan eksternal. Adapun aliran data antara memori internal dan media penyimpanan eksternal dapat digambarkan sebagai berikut :



Gambar 2.4. Aliran data antara memori internal dengan media penyimpanan eksternal

Sebuah file secara logika merupakan data yang dibaca dari suatu media atau dibangkitkan oleh suatu program, pada saat data itu dibaca maka memori internal komputer akan menyimpannya untuk sementara waktu sehingga user dapat berinteraksi dengan data tersebut secara langsung, namun dalam memori internal ini data tersebut bersifat sementara saja sehingga apabila data tersebut tidak disimpan pada media simpan maka data tersebut tidak dapat diakses pada waktu yang akan datang.