# Penggunaan Hash Fitur Sebagai *Dimensionality Reduction* Fitur N-gram dalam Pembelajaran Mesin

M. Ferdi Ghozali
Sekolah Teknik Elektro dan Informatika
*Institut Teknologi Bandung*
*Bandung, Indonesia*
13515014@std.stei.itb.ac.id

**Abstrak—Aplikasi fungsi hash dalam dunia komputer sangatlah banyak. Seiring dengan majunya perkembangan teknologi, fungsi hash terus menerus dikembangkan untuk meningkatkan keamanan. Namun uniknya penggunaan hash sudah tidak hanya digunakan pada bidang keamanan saja, fungsi hash juga dapat digunakan untuk *dimensionality reduction* pada fitur N-gram. Pearson hash adalah sebuah fungsi hash yang didesain untuk digunakan pada processor 8-bit. Dalam proses hash, pearson hash menggunakan 256-byte *lookup table* yang berisi permutasi dari nilai 0 hingga 255. Pearson hash memberikan menghasilkan *message digest*. MD5 adalah fungsi hash satu-arah yang dibuat oleh Ron Rivest. MD5 menghasilkan message digest yang panjangnya 128 bit.**

**Kata kunci—hash; n-gram; Pearson Hash; MD5**

## I. Pendahuluan

Salah satu jenis algoritma kriptografi yang sampai saat ini masih dikembangkan adalah algoritma hash. Fungsi hash merupakan fungsi yang dapat mengubah suatu pesan menjadi string dengan panjang tertentu sesuai dengan algoritmanya. Misalnya, Pearson hash memiliki panjang hash 1 byte dan MD5 memiliki panjang hash 16 byte.

Fungsi hash saat ini sangat banyak pemakaiannya. Penggunaan fungsi hash paling umum adalah untuk menjaga integritas data, menghemat waktu pengiriman pesan, dan penyeragaman panjang data.

Namun seiring berkembangnya pembelajaran mesin, *dimensionality reduction* juga mengalami perkembangan. *Principal Component Analysis (PCA)* adalah salah satu contoh dari dimensionality reduction yang sering digunakan. Salah satu contoh pengaplikasian fungsi hash sebagai *dimensionality reduction* terjadi pada kompetisi Microsoft Malware Classification Challenge (BIG 2015), dimana tim "Mikhail & Dmitry & Stanislav" menggunakan SHA224 dari 10-gram sebagai solusi akhir dan mengklaim menjadi fitur paling kreatif dan penting. Alhasil tim "Mikhail & Dmitry & Stanislav" menduduki peringkat ke-tiga pada kompertiis tersebut.

*Dimensionality reduction* digunakan untuk mengurangi fitur yang dinilai terlalu banyak. Hal ini sering terjadi pada penggunaan N-gram di kasus *natural language processing.* Sebagai contoh pada kompetisi Microsoft Malware Classification Challenge, dengan menggunakan fitur 10-gram, algoritma akan menghasilkan sebanyak $(256+1)^{10}$ fitur atau sekitar $1.2 * 10^{24}$. sedangkan Jika menggunakan SHA-224 dari 10-gram maka jumlah fitur akan berkurang menjadi $2^{28}$.

Keuntungan penggunaan dari fungsi hash sebagai dimensionality reduction adalah kecepatan prosesnya dibanding PCA. PCA mengalami peningkatan waktu sangat tinggi jika data yang diolah berdimensi banyak, sedangkan fungsi hash lebih stabil kecepatannya pada data dimensional tinggi.

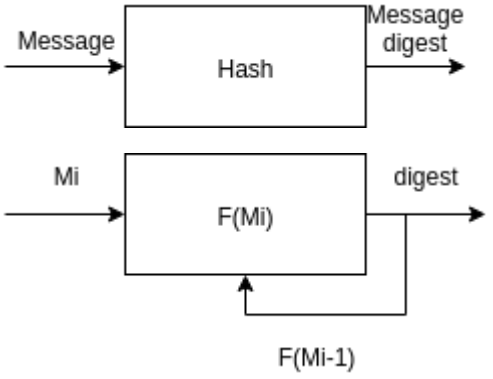
Pearson hashing adalah fungsi hash yang dirancang untuk prosesor dengan register 8-bit. Dalam proses hash, pearson hash menggunakan 256-byte *lookup table* yang berisi permutasi dari nilai 0 hingga 255. Pearson hashing cocok digunakan karena cepat dan menghasilkan message digest yang pendek. Kekurangan dari pearson hashing adalah *Non-cryptographic,* berarti metode ini mudah di proses balik jika mengetahui *lookup table* yang digunakan. Namun karena permasalahan utama disini bukan keamanan maka kekurangan tersebut dihiraukan.

## II. Dasar Teori

### A. Fungsi Hash

Fungsi hash adalah fungsi digunakan untuk mengubahan pesan berupa byte yang panjangnya sembarang menjadi byte yang memiliki panjang tetap. Fungsi hash memiliki sifat *irreversible*. Hal ini berarti pesan yang sudah diubah menjadi message digest tidak dapat diubah kembali menjadi pesan sebelumnya. Sehingga, sangat sulit untuk mencari dua pesan yang menghasilkan digest yang sama F(y) = F(x).

Masukan fungsi hash adalah berupa blok-blok pesan yang merupakan potongan-potongan dari satu kesatuan pesan dan keluaran dari hasil hash blok pesan sebelumnya. Berikut ini adalah skema fungsi hash.



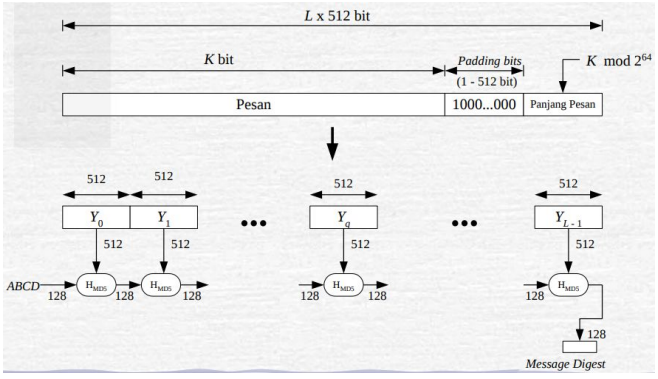

Mi merupakan blok pesan ke-i, F(Mi) adalah hasil keluaran fungsi hash / digest, dan F(Mi-1) adalah hasil keluaran fungsi hash sebelumnya. Apabila i = 1, maka F(Mi) digantikan dengan vektor awal yang diberi nilai tetap.

Pada pembelajaran mesin fungsi hash dapat digunakan sebagai pengurangan jumlah fitur pada n-gram sehingga

proses training tidak terlalu lama dan diharapkan juga dapat meningkatkan akurasi.

*B. MD5*

MD5 adalah fungsi hash satu-arah yang dibuat oleh Ron Rivest. MD5 merupakan perbaikan dari MD4 setelah MD4 ditemukan kolisinya. Algoritma MD5 menerima masukan berupa pesan dengan ukuran sembarang dan menghasilkan message digest yang panjangnya 128 bit.




Langkah-langkah pembuatan message digest secara garis besar:

1. Penambahan bit-bit pengganjal (padding bits).
2. Penambahan nilai panjang pesan semula.
3. Inisialisasi penyangga (buffer) MD.
4. Pengolahan pesan dalam blok berukuran 512 bit.

Pesan ditambah dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 (mod 512). Jika panjang pesan 448 bit, maka pesan tersebut ditambah dengan 512 bit menjadi 960 bit. Jadi, panjang bit-bit pengganjal adalah antara 1 sampai 512. Bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

Pesan yang telah diberi bit-bit pengganjal selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula. Jika panjang pesan > $2^{64}$ maka yang diambil adalah panjangnya dalam modulo $2^{64}$. Dengan kata lain, jika panjang pesan semula adalah K bit, maka 64 bit yang ditambahkan menyatakan K modulo $2^{64}$. Setelah ditambah dengan 64 bit, panjang pesan sekarang menjadi kelipatan 512 bit.

MD5 membutuhkan 4 buah penyangga (buffer) yang masing-masing panjangnya 32 bit. Total panjang penyangga adalah 4  32 = 128 bit. Keempat penyangga ini menampung hasil antara dan hasil akhir. Keempat penyangga ini diberi nama A, B, C, dan D. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) sebagai berikut:

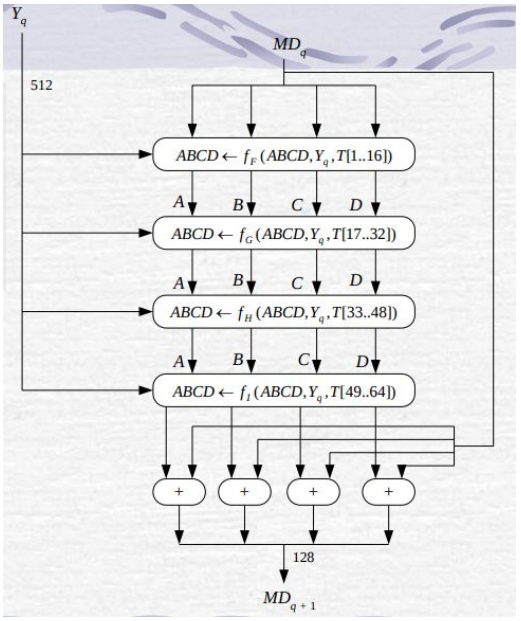
A = 01234567

B = 89ABCDEF

C = FEDCBA98

D = 76543210

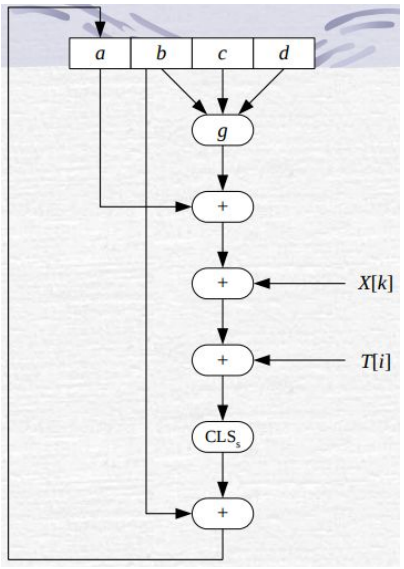
Pesan dibagi menjadi L buah blok yang masingmasing panjangnya 512 bit ($Y_0$ sampai $Y_{L-1}$ ). Setiap blok 512-bit diproses bersama dengan penyangga MD menjadi keluaran 128-bit, dan ini disebut proses $H_{MD5}$. Gambaran proses $H_{MD5}$ diperlihatkan pada Gambar berikut:




Pada Gambar tersebut, $Y_q$ menyatakan blok 512-bit ke-q. $MD_q$ adalah nilai message digest 128-bit dari proses $H_{MD5}$ ke-q. Pada awal proses, $MD_q$ berisi nilai inisialisasi penyangga MD. Proses $H_{MD5}$ terdiri dari 4 buah putaran, dan masing-masing putaran melakukan operasi dasar MD5 sebanyak 16 kali dan setiap operasi dasar memakai sebuah elemen T. Jadi setiap putaran memakai 16 elemen Tabel T.

| Nama | Notasi | $g(b, c, d)$ |
|---|---|---|
| $f_F$ | $F(b, c, d)$ | $(b \wedge c) \vee (\sim b \wedge d)$ |
| $f_G$ | $G(b, c, d)$ | $(b \wedge d) \vee (c \wedge \sim d)$ |
| $f_H$ | $H(b, c, d)$ | $b \oplus c \oplus d$ |
| $f_I$ | $I(b, c, d)$ | $c \oplus (b \wedge \sim d)$ |

Fungsi-fungsi $f_F$ , $f_G$ , $f_H$ , dan $f_I$ masing-masing berisi 16 kali operasi dasar terhadap masukan, setiap operasi dasar menggunakan elemen Tabel T. Operasi dasar MD5 diperlihatkan pada Gambar berikut:




*C. Pearson Hash*

Pearson hashing adalah sebuah metode hash yang didesain untuk keperluan processor 8-bit. Pearson hashing

bukan merupakan *cryptographic* sehingga mudah untuk dibalik prosesnya. Namun karena keperluan penggunaan bukan untuk keamanan, kekurangan ini dihiraukan.

fungsi hash ini menggunakan 8-bit cipher substitusi dalam bentuk tabel sebagai media transformasinya. Karena hanya menggunakan 8-bit substitusi tabel maka algoritma ini tidak tergolong kuat dalam keamanan. Namun ada beberapa kelebihan yang diberikan algoritma ini, antara lain:

1. Proses kerjanya sangatlah sederhana.
2. Cepat.
3. hasil digestnya singkat (8 bit / 1 byte).
4. sangat jarang *collision*.

Berikut *pseudocode* dari pearson hash.

```
h := 0
  for each c in C loop
    h := T[ h xor c ]
  end loop
return h
```

*D. Spam Mails Detection*

ADCG SS14 Challenge 02 merupakan sebuah kompetisi yang diselenggarakan pada tahun 2014 di kaggle. kompetisi ini tidak memiliki hadiah hanya untuk media eksperimen dan belajar. Kompetisi ini menyediakan 5 file.

1. TR-mails.zip - email + label untuk *training*
2. TT-mails.zip - email + label untuk *testing*
3. test.submission.txt - contoh hasil label
4. readme.txt - deskripsi data
5. ExtractContent.py - kode python untuk mengekstrak email

Terdapat sekitar 2500 data latih dan 1800 data test pada kompetisi ini. Kompetisi ini dipilih karena data tidak terlalu banyak sehingga tidak terlalu memberatkan apabila digunakan untuk eksperimen.

*E. N-Gram*

Language model yang sudah dikenal dan banyak digunakan oleh peneliti adalah n-gram. N-gram (adjacent n-gram) merupakan kumpulan dari item sejumlah n yang disusun secara berurutan dari text atau speech.

Sebagai Contoh kalimat awal nya adalah sbb.

```
"Natural-language processing (NLP) is an area of
computer science and artificial intelligence concerned
with the interactions between computers and human
(natural) languages."
```

maka nilai dari 5-gram kalimat tersebut adalah sbb.

```
['natural language processing nlp is',
 'language processing nlp is an',
 'processing nlp is an area',
 'nlp is an area of',
 'is an area of computer',
 'an area of computer science',
 'area of computer science and',
 'of computer science and artificial',
 'computer science and artificial intelligence',
 'science and artificial intelligence concerned',
 'and artificial intelligence concerned with',
 'artificial intelligence concerned with the',
 'intelligence concerned with the interactions',
 'concerned with the interactions between',
 'with the interactions between computers',
 'the interactions between computers and',
 'interactions between computers and human',
 'between computers and human natural',
 'computers and human natural languages']
```

Dapat dilihat bahwa satu kalimat teks saja bisa menghasilkan 19 fitur, tentu jika digunakan pada lebih banyak data maka jumlah fitur akan membludak. Oleh karena itu perlu pemilihan fitur atau pengurangan dimensi pada fitur.

*F. Hash Function untuk dimensionality reduction*

Meskipun kolisi sangat jarang terjadi pada algoritma hash, tapi perlu disadari bahwa panjang output hash selalu sama dan terbatas. Hal ini menjadikan bahwa algoritma hash pasti akan terjadi kolisi jika diinputkan dengan lebih banyak dari kemungkinan output hash. Sebagai contoh algoritma MD5 memiliki 16 byte output, Jika saya coba input sembarang string berbeda sebanyak permutasi 16 byte + 1 input, maka mau tidak mau akan ada setidaknya satu output yang sama, hal ini tentu sulit untuk diprediksi atau diatur agar sama oleh karena itu dianggap random. Jika pada pearson panjang output diatur 1 byte (256 kemungkinan output), maka jika jumlah hasil N-Gram adalah 10000, maka akan banyak nilai N-gram yang output hash nya sama dan akhirnya digabungkan sehingga fitur N-Gram yang seharusnya 10000 macam menjadi 256 macam saja.

## III. Rancangan Eksperimen

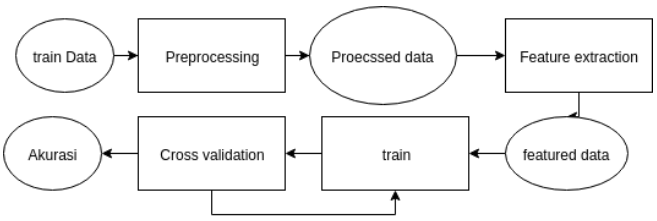
Sebelum melakukan eksperimen, dilakukan ekstraksi data untuk mengubah file email menjadi dataframe.

```
srcdir = "../dataset/TR-mails/"
dstfile = "../destination/training.csv"
if not os.path.exists(srcdir):
        print ('The source directory %s does not exist,
exit...' % (srcdir))
        sys.exit()
ExtractBodyFromDir(srcdir, dstfile)

srcdir = "../dataset/TT-mails/"
dstfile = "../destination/predict.csv"
if not os.path.exists(srcdir):
        print ('The source directory %s does not exist,
exit...' % (srcdir))
        sys.exit()
ExtractBodyFromDir(srcdir, dstfile)
```

Setelah itu untuk mengurangi variasi kata yang berbeda, diperlukan preprocessing sehingga kelak hasil fitur tidak terlalu banyak. Setelah di preprocessing, maka data akan diambil fiturnya (N-gram) dan kemudian dimasukan ke dalam *dimensional reduction* untuk dipotong jumlah

fiturnya, dan kemudian dimasukan ke *classifier* untuk dilatih dan diuji.




### A. *Preprocessing*

Preprocessing adalah tahap manipulasi data agar data lebih mudah diolah. Pada kasus ini preprocessing digunakan agar jenis kata yang muncul pada data lebih sedikit sehingga hasil dari N-gram tidak terlalu banyak. Terdapat 3 metode preprocessing yang digunakan pada eksperimen ini, yaitu stemming, lemmatizing, dan stopword *elimination*.

1. stemming

   stemming adalah proses mengembalikan bentuk kata menjadi kata dasar, meskipun kata tersebut bukan kata yang valid. sebagai contoh menuliskan menjadi nulis.

2. lemmatizing

   lemmatizing adalah proses mengembalikan bentuk kata ke kata dasar, berbeda dengan stem, lemmatizing mengharuskan kata dasar merupakan kata yang valid. Contoh nya menuliskan menjadi tulis.

3. *stopword elimination*

   menghilangkan kata kata yang sering digunakan seperti “adalah”, “the”, “dan”, dan tanda baca. Tujuan nya karena kata kata tersebut muncul di semua konteks sehingga tidak berarti untuk dijadikan fitur klasifikasi.

### B. *feature extraction*

feature extraction adalah proses mengambil fitur pada data sehingga model lebih mudah mengenali pola yang terkandung didalamnya. Feature extraction yang digunakan adalah n-gram.

1. N-gram

   N-gram merupakan salah satu proses yang secara luas digunakan dalam text mining (pengolahan teks) dan pengolahan bahasa. N-gram merupakan sekumpulan kata yang ada dalam sebuah paragraf dan ketika menghitung N-gram biasanya dilakukan dengan menggerakkan satu kata maju ke depan (Meskipun dalam prosesnya terdapat suatu proses dimana kata yang dimajukan sejumlah X kata). Sebagai contoh terdapat sebuah kalimat “The cow jumps over the moon”. Jika n=2 makan dikenal dengan bigram. Dimana n-gram menjadi : The cow, Cow jumps, Jumps over, Over the, The moon.

### C. *Dimensionality Reduction*

*Dimensionality reduction* adalah proses mengurangi dimensi fitur data agar *classifier* dapat memproses fitur dengan lebih efektif dan tidak memakan waktu lama. Ada tiga metode *Dimensionality reduction* yang akan digunakan, yaitu PCA, Pearson hash, MD5 hash.

### D. *Train and Test*

algoritma *classifier* yang akan digunakan adalah *support vector machine*. SVM merupakan algoritma yang robust digunakan untuk data yang tidak terlalu banyak dan banyak fitur.

Eksperimen diawali dengan melakukan klasifikasi spam email dari fitur 1-gram dan 1-gram + PCA sebagai pembanding. Setelah itu dilanjutkan dengan eksperimen dengan 1-gram + Pearson hash dan 1-gram + MD5. dari keempat eksperimen akan dihitung akurasi, jumlah fitur, waktu training dan waktu pengurangan dimensi fitur. Eksperimen dilakukan dengan metode 5-fold cross validation. Data yang akan digunakan hanya 1000 Data Training yang dipilih berdasarkan indeks awal.

## IV. Hasil Eksperimen

Berikut hasil dari eksperimen yang dilakukan menggunakan dataset dan metode yang sudah dijelaskan pada bab-bab sebelumnya.

| | Jumlah Fitur | Akurasi (rata rata 5-fold) | Waktu Proses |
|---|---|---|---|
| n-gram | 31335 | 0.81 | 0.13 |
| PCA + n-gram | 256 | 0.7 | 0.29 |
| n-gram + Pearson | 256 | 0.65 | 0.32 |
| MD5 + n-gram | 31335 | - | - |

## V. Analisis dan Kesimpulan

Dari hasil eksperimen, penggunaan PCA dan Pearson hash dari fitur N-gram mengalami penurunan akurasi. Hal ini mungkin dikarenakan penurunan jumlah fitur yang cukup signifikan sehingga banyak informasi yang hilang.

Selain itu MD5 tidak menurunkan jumlah feature sehingga tidak dilakukan eksperimen lebih lanjut. Hal ini dikarenakan jumlah input yang jauh lebih sedikit dari seluruh kemungkinan output dari MD5.

Penelitian ini memberikan kesimpulan bahwa PCA lebih baik digunakan untuk dimensionality reduction dari pada fungsi hash. Hal ini dikarenakan hash menyatuk fitur secara acak sedangkan PCA menyatukan fitur dengan aturan tertentu sehingga memperkecil kehilangan informasi.

## VI. Ucapan terimakasih

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa, karena atas bantuan, rahmat, dan berkat-Nya, makalah ini dapat selesai pada waktunya. Tak lupa juga,

penulis ingin menyampaikan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir selaku dosen mata kuliah IF4020 Kriptografi yang telah membagikan ilmunya kepada penulis. Selain itu, penulis juga ingin menyampaikan terima kasih kepada kedua orang tua yang selalu mendukung penulis.

.

## References

[1] R. Munir, Slide Kuliah IF4020 Kriptografi, Fungsi Hash, 2018.

[2] R. Munir, Slide Kuliah IF4020 Kriptografi, SHA, 2018.

[3] R. Munir, Slide Kuliah IF4020 Kriptografi, MD5, 2018.

[4] https://en.bitcoinwiki.org/wiki/Pearson_hashing

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Mei 2019

Mokhammad Ferdi Ghozali/13515014