

**ANALISIS & PERANCANGAN PROGRAM APLIKASI
UNTUK KLASIFIKASI
MENGUNAKAN K-NN (K – Nearest Neighbor)**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Disusun oleh

Nama : Zeindri Saputra

No. Mahasiswa : 04.523.296

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2011

LEMBAR PENGESAHAN PEMBIMBING
ANALISIS & PERANCANGAN PROGRAM APLIKASI UNTUK KLASIFIKASI
MENGGUNAKAN K-NN (K – Nearest Neighbor)
LAPORAN TUGAS AKHIR



Telah Diterima Dan Disetujui Dengan Baik Oleh :

Dosen pembimbing

(Dr.Hj.Sri Kusumadewi, S.Si., M.T)

LEMBAR PENGESAHAN PEMBIMBING
ANALISIS & PERANCANGAN PROGRAM APLIKASI UNTUK KLASIFIKASI
MENGGUNAKAN K-NN (K – Nearest Neighbor)

LAPORAN TUGAS AKHIR



Disusun oleh

Nama : Zeindri Saputra

No. Mahasiswa : 04 523 296

Yogyakarta, April 2011

Telah Diterima Dan Disetujui Dengan Baik Oleh :

Dosen pembimbing

(Dr.Hi.Sri Kusumadewi, S.Si., MT)

LEMBAR PENGESAHAN PENGUJI
ANALISIS & PERANCANGAN PROGRAM APLIKASI UNTUK KLASIFIKASI
MENGGUNAKAN K-NN (K – Nearest Neighbor)

TUGAS AKHIR

Disusun oleh:

Nama : Zeindri Saputra

No.Mahasiswa : 04.523.296

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, April 2011

Tim Pengujian

Ketua

Dr. Sri Kusumadewi, S.Si., MT _____

Anggota I

Izzati Muhimah, ST., M.Sc., Ph.D _____

Anggota II

Zainudin Zukhri, ST., MIT _____

Mengetahui,
Ketua Program Studi Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom

LEMBAR PENGESAHAN PENGUJI

ANALISIS & PERANCANGAN PROGRAM APLIKASI UNTUK KLASIFIKASI MENGUNAKAN K-NN (K – Nearest Neighbor)

TUGAS AKHIR

Disusun oleh:

Nama : Zeindri Saputra

No.Mahasiswa : 04.523.296

Telah Dipertahankan di Depan **Sidang Penguji** Sebagai Salah Satu Syarat
Untuk Memperoleh **Gelar Sarjana** Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, April 2011

Tim Pengujian

Ketua


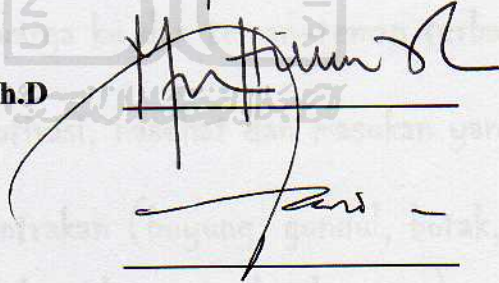
Dr. Sri Kusumadewi, S.Si., MT

Anggota I

Izzati Muhimah, ST., M.Sc., Ph.D

Anggota II

Zainudin Zukhri, ST., MIT

Mengetahui,

Ketua Program Studi Teknik Informatika
Universitas Islam Indonesia


Yudi Prayudi, S.Si., M.Kom

PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Terima kasih Allah SWT, karena hanya dengan rahmat dan hidayah-Mu semua dapat terjadi.

Kupersembahkan Untuk ;

Kedua orang tuaku yang luar biasa dalam mendampingi hidupku, karena beliaulah aku mencoba meraih segala cita-cita yang aku impikan.

Terima kasih ;

Buat seluruh keluarga ku dan teman-teman terbaik ku yang telah memberikan motivasi, nasehat dan masukan yang tiada henti.

Xploit, Teman-teman kontrakan (buyung, gundul, botak, young-yon, acep, adek, widy, paris, hardy, tara)

Seluruh teman-teman yang tidak dapat disebutkan

"You are the best people in my life"

MOTTO

“... Allah akan meninggikan orang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat” (QS.Al-Mujaadilah ayat 11)

“ Sesungguhnya sesudah kesulitan itu ada kemudahan ; Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain ”. (Q.S. Alam Nasyrah ayat 6 dan7)

“.. Bertolong-tolonglah kamu dalam kebaikan dan dalam melaksanakan takwa, jangan kamu tolong-menolong dalam dosa dan permusuhan ...”

(QS. AL-Maaidah ayat 2)

“ Raihlah lima perkara sebelum datangnya lima yang lain : 1. hidupmu sebelum matimu, 2. sehatmu sebelum sakitmu, 3. kesempatan sebelum sibukmu, 4. mudamu sebelum tuamu, 5. kayamu sebelum miskinmu”.

(Shahih jami'ush shaghir Al Albani dari Ibnu Abbas ra)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum. Wr. Wb

Dengan mengucapkan Alhamdulillah, puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga dapat menyelesaikan tugas akhir ini, yang berjudul “**Analisis & Perancangan Program Perancangan Aplikasi Untuk Klasifikasi Menggunakan Metode K-NN (K- Nearest Neighbor)**” dengan baik.

Laporan tugas akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Teknik Informatika pada Universitas Islam Indonesia dan atas apa yang telah diajarkan selama perkuliahan baik teori maupun praktek, disamping laporan itu sendiri yang merupakan rangkaian kegiatan yang harus dilakukan setelah tugas akhir ini selesai.

Penulisan dan penyelesaian tugas akhir ini tidak lepas dari saran, bimbingan, dukungan serta bantuan dari berbagai pihak. Untuk itu pada kesempatan kali ini penulis menyampaikan ucapan terimakasih kepada :

1. Allah SWT. Atas segala hidayah, barokah dan taufiq-Nya
2. Bapak Ir. Gumbolo HS., M.Sc selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Jurusan Teknik Informatika.
4. Ibu Dr.Hj.Sri Kusumadewi, S.Si., M.T, selaku Dosen Pembimbing Tugas Akhir. Terima kasih atas segala bantuan, dukungan, dan pengetahuannya yang telah diberikan kepada penulis dalam penyusunan skripsi ini.
5. Kedua orangtuaku, yang selalu melimpahkan kasih sayang yang tulus, doa yang tiada henti-hentinya, serta dukungan yang begitu besar dalam kehidupanku.

6. Rully adek, Young-yon, Olly, Yudha, Husni, Fahmy, Acep, Widiyanto B.S, Heru Kular-Kilir dan teman-teman lainnya terimakasih.
7. Xploit, betapa bangga aku menjadi bagian dari kalian.
8. Semua pihak yang telah membantu dalam pembuatan hingga terselesaikannya tugas akhir ini, yang tidak dapat penulis sebutkan satu persatu.

Akhir kata dengan ketulusan hati penulis panjatkan doa semoga apa yang telah mereka berikan dengan keikhlasan, mendapat pahala yang setimpal dari Allah SWT. Penulis menyadari dalam penulisan laporan tugas akhir ini masih jauh dari sempurna, karena keterbatasan kemampuan dan pengalaman. Penulis mengharapkan saran dan kritik yang bersifat membangun untuk memperbaiki tugas akhir ini semoga dapat bermanfaat bagi penulis khususnya dan pembaca pada umumnya.

Wabillahiittaufiq wahi hidayah, Wassalamu'alaikum Wr. Wb

Yogyakarta,

April 2011

Zeindri Saputra

SARI

Meningkatnya era teknologi informasi dalam hal pengklasifikasian saat sekarang ini yang tidak terstruktur, analisis pengklasifikasian sudah banyak membantu dalam hal mempermudah menyelesaikan suatu perancangan program aplikasi yang tidak terstruktur menjadi suatu sistem aplikasi pengklasifikasian yang terstruktur. Dengan adanya penelitian analisis dalam proses pengklasifikasian menggunakan metode *K-NN* (*K-Nearest Neighbor*) diharapkan dapat mempermudah dalam pembuatan sistem aplikasi pengklasifikasian untuk suatu perancangan program aplikasi.

Dari permasalahan diatas, diperlukan suatu metode K-NN dimana K-NN merupakan suatu parameter K (jumlah tetangga terdekat) untuk menghitung jarak yang akan di evaluasi dengan semua data penelitian dengan cara memasukkan data-data pendukung dalam suatu perancangan analisis pengklasifikasian agar data yang diklasifikasi dapat lebih mudah dipahami dalam pembuatan sistem aplikasi.

Hasil analisis menggunakan metode K-NN dapat di peroleh dengan melakukan pengklasifikasian dengan rumus K-NN dan Apabila data-data tersebut telah diproses maka analisis ini akan memberikan hasil pengklasifikasian.

Kata Kunci : Analisis dan pengklasifikasian, *Metode KNN (K-Nearest Neighbor)*

TAKARIR

- Decision Support System* : Sistem pendukung keputusan
- k – nearest neighbor* : Metode klasifikasi yang digunakan sebagai metode SPK
- training data* : Data yang digunakan untuk pengklasifikasian.



DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN DOSEN PEMBIBING	ii
LEMBAR PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
KATA PENGANTAR	vi
SARI	viii
TAKARIR	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Metode Penelitian	2
1.7 Sistematika Penulisan Laporan Penelitian	3
BAB II LANDASAN TEORI	4
2.1 Metode Analisis	4
2.2 Metode K-NN (K-Nearest Neighbor)	4
2.3 Flowchart Metode KNN	7
2.4 Teori Mencari Jarak	8
BAB III ANALISIS DAN PERANCANGAN	9

3.1 PSEUDOCODE	9
3.1.1 Program mencari jarak	9
3.1.2 Program urutan jarak	9
3.1.3 Program mencari kelas	10
3.1.4 Program mencari jarak kelas	12
3.2. PERENCANAAN ANTAR MUKA	16
3.2.1 Halaman Utama	16
3.2.2 Halaman Menghitung Kadar Kalori Buah	17
3.2.3 Halaman Pengolahan Data Sampel	18
3.2.4 Halaman Pencarian Data Sampel	19
3.2.5 Halaman Pencarian Berdasarkan Status	20
3.2.6 Halaman Informasi	21
BAB IV SIMPULAN DAN SARAN	22
4.1 Kesimpulan	22
4.2 Saran	22
DAFTAR PUSTAKA	23

DATAR GAMBAR

Gambar 2.1 Flowchart Metode K-NN	7
Gambar 3. 1 Rancangan Halaman Utama	13
Gambar 3. 2 Rancangan Halaman Menghitung Kadar Kalori Buah	14
Gambar 3. 3 Rancangan Halaman Pengolahan Data Sampel	15
Gambar 3.4 Rancangan Halaman Pencarian Data Sampel	16
Gambar 3.5 Rancangan Halaman Pencarian Berdasarkan Status	17
Gambar 3.6 Racangan Halaman Informasi	18

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan perkembangan zaman sekarang ini membuat kebutuhan manusia akan informasi semakin meningkat pula. Sebagai contoh, dengan adanya komputer segala pekerjaan dapat lebih mudah dilakukan. Akibat dari perkembangan teknologi saat sekarang ini maka diperlukan suatu cara pengelompokan informasi untuk memudahkan para pengguna mengelompokan data yang dibutuhkan. Data pengelompokan ini dikenal dengan pengklasifikasian. Cara pengklasifikasian ini sudah banyak digunakan dalam berbagai aplikasi pencarian seperti google, yahoo, dan banyak lainnya.

Latar belakang dibuatnya Analisis dan Perancangan Aplikasi untuk Klasifikasi menggunakan K-NN ini bertujuan bagaimana penulis dapat membuat suatu pengklasifikasian aplikasi dengan K-NN agar informasi pengklasifikasian ini dapat dipahami. Prinsip dasar pengklasifikasian ini berdasarkan kesamaan fitur atau kesamaan isi, secara umum pengklasifikasian dilakukan dengan cara mengkategorikan unsur-unsur nya kedalam satu atau beberapa objek dari sekumpulan objek-objek yang telah didefinisikan sebelumnya.

Dari latar belakang permasalahan diatas, untuk itu penulis bermaksud ingin membuat suatu analisis dan perancangan program aplikasi untuk klasifikasi menggunakan K-NN yang mampu memberikan informasi dan kemudahan dalam membuat suatu aplikasi yang menggunakan metode pengklasifikasian.

1.2 Rumusan Masalah

Rumusan masalah penelitian ini berdasarkan latar belakang diatas adalah bagaimana membuat suatu analisis pengklasifikasian dengan menggunakan K-NN dan memberikan informasi tentang pengklasifikasian.

1.3 Batasan Masalah

Pada analisis yang akan dibuat, agar permasalahan tidak terlalu luas dan lebih terarah sebagaimana tujuan yang ingin dicapai, maka diperlukan batasan-batasan sebagai berikut.

1. Analisis ini merupakan suatu analisis pengklasifikasian data dengan menggunakan metode K-NN.
2. Hasil dari analisis yang akan dibuat merupakan suatu pengklasifikasian data dari contoh sampel data dengan menggunakan metode K-NN.

1.4 Tujuan Penelitian

Penelitian yang akan dilakukan bertujuan untuk menganalisis sebuah sistem perancangan aplikasi dengan K-NN dan dari hasil analisis ini akan dibuat sebuah struktur pengklasifikasian yang dapat memberikan informasi tentang sebuah pengklasifikasian.

1.5 Manfaat Penelitian

Dengan adanya manfaat penelitian analisis ini, diharapkan dapat memberikan

1. Solusi dalam sebuah perancangan analisis yang baik.
2. Memberikan informasi pengklasifikasian dalam membangun sebuah aplikasi.

1.6 Metode Penelitian

Metode penelitian adalah langkah-langkah sistematis penelitian yang akan digunakan untuk membantu dalam pengerjaan suatu penelitian analisis. Adapun metode-metode yang akan digunakan untuk mencapai hasil dalam penelitian analisis ini.

1.6.1 Metode Pengumpulan Data

Langkah awal metode pengumpulan data dari sebuah analisis dan perancangan program aplikasi dengan K-NN ini.

1. Mencari informasi tentang khusus yang akan dibuat dalam analisis dan perancangan algoritma K-NN.
2. Membuat langkah algoritma dari khusus yang akan dianalisis.

1.6.2 Metode Pengembangan

1. Analisis Data.
2. Analisis K-NN.

1.7 Sistematika Penulisan Laporan Penelitian Analisis ini adalah sebagai berikut.

BAB I. PENDAHULUAN ; Berisi tentang latar belakang masalah, Rumusan masalah, Batasan masalah, Tujuan penelitian, Metodologi penelitian dan Sistematika penelitian.

BAB II LANDASAN TEORI ; Menerangkan tentang metode K-NN dan analisis algoritma dari penelitian yang akan dilakukan.

BAB III PSEUDOCODE ; Berisi tentang hasil penerangan dalam sebuah algoritma yang dianalisis.

BAB IV KESIMPULAN ; Ini merupakan bab terakhir yang berisikan kesimpulan yang diperoleh dari pemecahan masalah serta saran-saran sebagai masukan.

BAB II

LANDASAN TEORI

2.1 Metode Analisis

Metode yang akan digunakan untuk melakukan analisis dan perancangan program adalah metode K-NN. K-NN itu sendiri adalah suatu teknik yang dapat digunakan untuk menyelesaikan pengklasifikasian terhadap suatu objek. Pada metode pengklasifikasian input dan output data dinyatakan dengan diagram aliran data (data flow diagram) sebagai keseluruhan proses yang ada pada sistem. Pada tahapan ini digunakan notasi-notasi yang ada untuk menggambarkan arus data dari sistem, dimana akan sangat diperlukan untuk membantu dalam proses komunikasi dengan pengguna.

2.2 Metode K-NN (K-Nearest Neighbor)

Algoritma *k-nearest neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan (menggambarkan, menunjukkan atau menjelaskan) fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak Euclidean. [DAI01]

Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data test (yang klasifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut. Nilai k yang terbaik untuk algoritma ini tergantung pada data; secara umumnya, nilai k yang tinggi akan mengurangi efek *noise*

(kesalahan) pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus dimana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma *nearest neighbor*. Ketepatan algoritma k-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik.

- Algoritma metode ini adalah sebagai berikut :
 1. Tentukan parameter K (Jumlah tetangga yang terdekat).
 2. Hitung jarak antara data yang akan dievaluasi dengan semua data pelatihan.
 3. Urutkan jarak yang terbentuk (urut naik) dan tentukan jarak terdekat sampai urutan ke – K.
 4. Pasangkan kelas (c) yang bersesuaian.
 5. Cari jumlah kelas dari tetangga yang terdekat tersebut, dan tetapkan kelas tersebut sebagai kelas data yang dievaluasi.

Metode ini mempunyai rumus seperti persamaan (1) :

$$d_i = \sqrt{\sum_{i=1}^n (X_{ij} - P_j)^2} \quad \dots\dots\dots (1)$$

Keterangan :

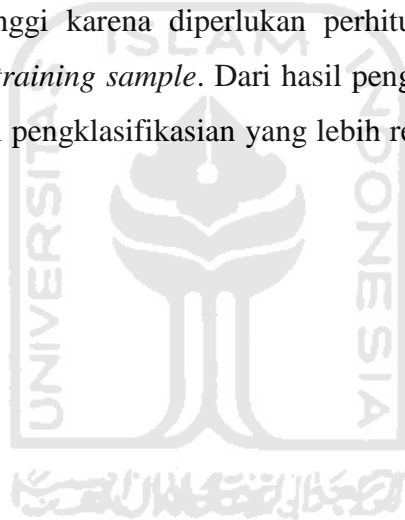
d_i = Jarak sampel

X_{ij} = Data sample yang sudah ada di dalam database.

P_j = Data input yang akan dicari hasilnya.

n = Jumlah sample

K-Nearest Neighbor (K-NN) memiliki beberapa kelebihan yaitu bahwa tangguh terhadap *training data* yang *noisy* dan efektif apabila *training data*-nya besar. Sedangkan kelemahan dari K-NN adalah K-NN perlu menentukan nilai dari parameter K (jumlah dari tetangga terdekat). Metode *K-Nearest Neighbor* (KNN) kurang cocok untuk melakukan klasifikasi dokumen karena K-NN tidak menghasilkan pemodelan dari proses *learning* terhadap training data namun hanya berdasarkan perhitungan jarak terpendek dari *query instance* ke *training sample*. Proses pembelajaran berdasarkan perhitungan jarak menghasilkan representasi yang tidak jelas mengenai jenis jarak dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik. Selain itu biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample*. Dari hasil pengertian diatas, semakin tinggi nilai k maka diperoleh hasil pengklasifikasian yang lebih rendah dan waktu pemrosesan yang lebih lama.



Algoritma menghitung jarak terdekat dengan metode KNN (*K- Nearest Neighbor*) adalah.

1. Tentukan parameter K (jumlah tetangga terdekat).
2. Hitung jarak antara data yang akan dievaluasi dengan semua data pilihan.
3. Urutkan yang terbentuk (urut naik) dan tentukan jarak jarak terdekat sampai urutan ke – K.
4. Pasangkan kelas (c) yang bersesuaian.
5. Cari jumlah kelas dari tetangga yang terdekat tersebut, dan tetap kan kelas tersebut sebagai kelas data yang dievaluasi.

2.4 Teori Mencari jarak

a. Manhattan

Manhattan fungsi jarak yang akan menghitung jarak ke perjalanan untuk mendapatkan dari satu titik data lain jika garis grid sebagai berikut. Manhattan jarak antara dua item adalah jumlah komponen perbedaan terkait. Rumus untuk ini jarak antara titik $X = (X_1, X_2, \dots)$ dan titik $Y = (Y_1, Y_2, \dots)$ seperti pada persamaan (2) ;

$$d_i = \sum_{i=1}^n |X_i - Y_i| \dots\dots (2)$$

Dimana n adalah jumlah variabel dan X_i dan Y_i adalah nilai - nilai variabel i pada masing – masing titik-titik X dan Y.

b. Korelasi Pearson dan Pearson Squared

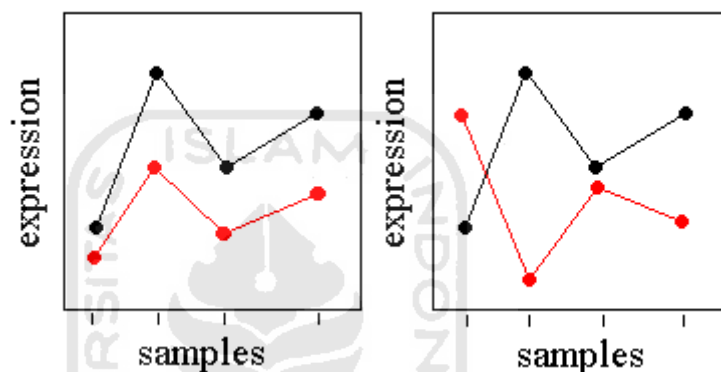
korelasi Pearson mengukur kesamaan dalam bentuk antara kedua profil. Rumus untuk jarak korelasi Pearson pada persamaan (3 dan 4)

$$d = 1 - r \dots\dots(3)$$

dimana

$$r = Z(x) \cdot Z(y) / n \dots\dots(4)$$

adalah titik skala Z - skor vektor X dan Y . Z adalah nilai x yang dibangun dengan cara mengurangkan dari nilai x rata-rata dan membagi dengan standar deviasi. Pearson Squared adalah mengukur jarak kesamaan dalam bentuk antara dua profil, tetapi juga untuk menangkap hubungan terbalik.



Pada gambar di sebelah kiri, profil profil hitam dan merah hampir sempurna, meskipun korelasi Pearson terdapat perbedaan tingkat ekspresi di basal dan skala. Gen ini akan cluster bersama-sama dengan baik korelasi Pearson atau jarak Squared Pearson. Pada gambar profil kanan, hitam dan merah hampir sempurna anti-berkorelasi. Gen ini akan ditempatkan dalam kelompok jarak jauh menggunakan korelasi Pearson, tetapi akan dimasukkan dalam cluster yang sama menggunakan Squared Pearson. Rumus untuk jarak Pearson squared pada persamaan (5)

$$d = 1 - 2r \dots\dots(5)$$

Dimana korelasi pearson adalah korelasi yang telah didefinisikan diatas.

c. Chebychev

Chebychev jarak antara dua titik adalah jarak maksimum antara titik-titik dalam dimensi tunggal. Jarak antara titik $\mathbf{X} = (\mathbf{X1}, \mathbf{X2}, \mathbf{dll})$ dan $\mathbf{Y} = (\mathbf{Y1}, \mathbf{Y2}, \mathbf{dll})$ dihitung dengan menggunakan rumus pada persamaan (6) ;

$$\mathbf{Maxi} |\mathbf{Xi} - \mathbf{Yi}| \dots\dots(6)$$

Dimana \mathbf{Xi} dan \mathbf{Yi} adalah nilai-nilai variabel \mathbf{i} pada titik-titik \mathbf{X} dan \mathbf{Y} masing-masing. Jarak Chebychev itu sendiri mungkin akan lebih tepat jika perbedaan antara titik–titik lebih tercermin oleh perbedaan dalam dimensi individu dibandingkan dengan semua dimensi yang akan dipertimbangkan secara bersama.

d. Korelasi Tingkat Spearman

Korelasi Rank Spearman mengukur korelasi antara dua urutan nilai. Kedua urutan adalah peringkat secara terpisah dan perbedaan peringkat dihitung pada setiap posisi \mathbf{i} . Jarak antara urutan $\mathbf{X} = (\mathbf{X1}, \mathbf{X2}, \mathbf{dll})$ dan $\mathbf{Y} = (\mathbf{Y1}, \mathbf{Y2}, \mathbf{dll})$ dihitung dengan menggunakan rumus pada persamaan (7) :

$$\mathbf{1} - \frac{\sum_{i=1}^n (\mathbf{rank}(X)_i - \mathbf{rank}(Y_i))^2}{n(n^2 - 1)} \dots\dots(7)$$

Dimana \mathbf{Xi} dan \mathbf{Yi} adalah nilai-nilai dari urutan \mathbf{X} dan \mathbf{Y} masing-masing. rentang korelasi Spearman adalah dari -1 sampai 1. korelasi Spearman dapat mendeteksi korelasi linear dan non-linear tertentu. Namun, korelasi Pearson mungkin akan lebih tepat untuk menemukan korelasi linear.

BAB III

ANALISIS DAN PERANCANGAN

3.1 PSEUDOCODE

3.1.1 Program mencari jarak :

Penjelasan coding dari implementasi prosedural sebagai berikut, program diawali dengan melakukan pencarian jarak dengan memasukkan input data ke metrix status dimana sebagai berikut :

```
function mencari_jarak (A, B, C, D : real) : real
var
    jarak : real
    berat, karbohidrat, protein, lemak : real
    {Nilai berat, karbohidrat, protein, lemak sudah terdefinisi nilainya}
begin
    jarak ← sqrt(((A-berat)*(A-berat))+((B-karbohidrat)*(B-karbohidrat))+((C-
    protein)*(C-protein))+((D-lemak)*(D-lemak)))
    return jarak
end.
```

Penghitungan jarak dilakukan diantara data yang akan dievaluasi dengan semua data pilihan, selanjutnya program melakukan pengurutan data berdasarkan status kalori.

3.1.2 Program urutan jarak :

Langkah selanjutnya melakukan pengurutan data sampai urutan ke - K setelah melakukan pencarian jarak. Berikut ini langkah pengurutan jarak berdasarkan status :

```
functionurut_jarak (K, i, j : integer) : integer
Var
```

```

urut_jarak : integer
  i ← 0 to K do
    i ← 0
  Begin
    For ( i ← 0; i < K - 1 ) do
      Begin
        If j [ i ] > j [ i+1 ] then
          Begin
            if ( j [ i ], j [ i + 1 ] ) then
              K ← i + 1
            End if
          Next
        End for
      while K > 1
    End
  
```

3.1.3 Program mencari kelas :

Selanjutnya program melakukan penentuan kondisi status, dimana langkahnya dapat dilihat dalam pseudocode dibawah ini :

```

Function kelas (kecil, sedang, besar : integer) : integer
var
status, kecil, sedang, besar : integer
{definisi untuk input dan output}
J, K : real
{nilai J dan K sudah terdefinisi nilainya}
begin
  input (J, status) ← 0
  if (status kecil > status sedang) dan (status kecil > status besar) then
    begin
  
```

```

    output.text ← “rendah”
end if
else if (status sedang > status kecil) dan (status sedang > status besar) then
    begin
        output.text ← “sedang”
    end if
else if (status besar > status kecil) dan (status besar > status sedang) then
    begin
        output.text ← “besar”
    end if
else if status kecil ← status sedang then
    for J ← 0 to K do
        begin
            if status (J) ← “rendah” then
                begin
                    status (J) ← “rendah”
                end if
            else status (J) ← “sedang”
            end if
        end if
    begin
        output.text ← tentukan status (0)
    end if
else if status kecil ← status besar then
    for J ← 0 to K do
        begin
            if status (J) ← “rendah” then
                begin
                    status (J) ← “rendah”
                end if
            end if
        end if
    end if
end if

```

```

else if (J) ← “besar”
end if
begin
output.text ← tentukan status (0)
end if
else if status sedang ← status besar then
for J ← 0 to K do
begin
if status (J) ← “sedang” then
begin
status (J) ← “sedang”
end if
else status (J) ← “besar”
end if
begin
output.text ← tentukan status (0)
end if
else
for J ← 0 to K do
begin
if status (J) “rendah” then
begin
tentukan status (J) ← “rendah”
end if
else if status (J) ← “sedang” then
begin
tentukan status (J) ← “sedang”
end if
else if (J) ← “besar”

```

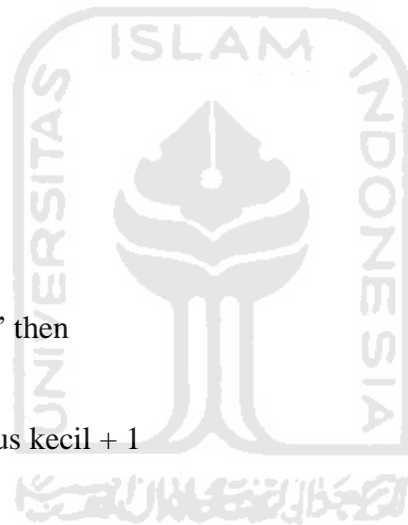
end if

end.

3.1.4 Program mencari jarak kelas :

Langkah terakhir, program melakukan pencarian jarak kelas terdekat sampai ke – K untuk selanjutnya dapat menentukan status, pseudoce nya sebagai berikut :

```
function jarak_kelas
output (rendah, sedang, besar : integer) : integer
var
i ← status : integer
J, K : integer
begin
  for (i) ← 0 to K do
    begin
      if (J) ← “rendah” then
        begin
          status kecil ← status kecil + 1
        end
      else if (J) ← “sedang” then
        begin
          status sedang ← status sedang + 1
        end
      else status besar ← status besar + 1
    end if
  end
end.
```



Tabel 3.1 Perhitungan jarak

No	Berat (gram)	Protein	Lemak	Karbohidrat	Status	Jarak
1	100	10	0.2	2	Tinggi	50.81
2	100	0	2.7	1.2	Tinggi	50.08
3	100	0	7.5	2	Tinggi	50.57
4	100	0	2	1	Tinggi	50.04
5	100	14	4.4	1	Tinggi	51.84
6	100	0	3.4	1	Tinggi	50.12
7	100	0.1	0	8.2	Tinggi	50.52
8	100	1.6	0.4	23.5	Tinggi	54.83
9	100	0.4	0	11.8	Sedang	51.16
10	100	0	8.4	1	Sedang	50.71
11	100	0	1.2	0	Sedang	50.03
12	100	0	2.2	0.6	Sedang	50.06
13	100	0	5.3	1	Sedang	50.29
14	100	0	1.8	1	Sedang	50.04
15	100	0	2.1	0.3	Sedang	50.06
16	100	0	1.2	0.4	Sedang	50.03
17	100	0	1	0	Sedang	50.03
18	100	0	2.2	0.8	Sedang	50.05
19	100	0	0	10	Rendah	50.81
20	100	0	2.1	1	Rendah	50.05
21	100	0	8.7	2	Rendah	50.77
22	100	0	3.3	0.6	Rendah	50.12
23	100	0	4.2	0	Rendah	50.19
24	100	0	1.4	0.9	Rendah	50.03
25	100	0	2.1	1.1	Rendah	50.05
26	100	0	1.8	0	Rendah	50.05
27	100	0	0.3	0	Rendah	50.02
28	100	0	2.2	0.7	Rendah	50.06
29	100	0	1.4	0.9	Rendah	52.03
30	100	3.2	0.5	17	Rendah	52.54

- a. Proses KNN : mengurutkan jarak
 Proses berikut adalah mengurutkan data sampel dari data yang terdekat jaraknya hingga jarak yang terjauh.

no	Berat	Protein	Lemak	Karbohidrat	Status	Jarak
1	100	0	0.3	0	Rendah	50.02
2	100	0	1.2	0	Sedang	50.03
3	100	0	1.2	0.4	Sedang	50.03
4	100	0	1	0	Sedang	50.03
5	100	0	1.4	0.9	Rendah	50.03
6	100	0	1.4	0.9	Rendah	50.03
7	100	0	2	1	Tinggi	50.04
8	100	0	1.8	1	Sedang	50.04
9	100	0	2.2	0.8	Sedang	50.05
10	100	0	2.1	1	Rendah	50.05
11	100	0	2.1	1.1	Rendah	50.05
12	100	0	1.8	0	Rendah	50.05
13	100	0	2.2	0.6	Sedang	50.06
14	100	0	2.1	0.3	Sedang	50.06
15	100	0	1.4	0.9	Rendah	50.06
16	100	0	2.7	1.2	Tinggi	50.08
17	100	0	3.4	1	Tinggi	50.12
18	100	0	3.3	0.6	Rendah	50.12
19	100	0	4.2	0	Rendah	50.19
20	100	0	5.3	1	Sedang	50.29
21	100	0.1	0	8.2	Tinggi	50.52
22	100	3.2	0.5	17	Rendah	50.54
23	100	0	7.5	2	Tinggi	50.57
24	100	0	8.4	1	Sedang	50.71
25	100	0	8.7	2	Rendah	50.77
26	100	10	0.2	2	Tinggi	50.81
27	100	0	0	10	Rendah	50.81
28	100	14	4.4	1	Tinggi	51.81
29	100	0	1.4	0.9	Rendah	52.03
30	100	3.2	0.5	17	Rendah	52.54

Tabel 3.2 Pengurutan Jarak dari yang terdekat hingga yang terjauh

b. Proses KNN : menyaring hasil keputusan

Hasil keputusan disaring berdasarkan nilai K. nilai merupakan jumlah kelas untuk memilih data yang diambil berdasarkan jarak yang terdekat. Didalam sistem ini nilai K dapat di tentukan.

Rumus mencari nilai K :

$$\mathbf{K = 1/8 \times \text{jumlah sampel data}}$$

Dalam kasus ini nilai $K = 1/8 \times 30 = 4$

Tabel 3.3 Penyaringan keputusan berdasarkan $K = 1/8 \times$ Data sampel

No	Berat	Protein	Lemak	Karbohidrat	Status	Jarak
1	100	0	0.3	0	Rendah	50.02
2	100	0	1.2	0	Sedang	50.03
3	100	0	1.2	0.4	Sedang	50.03
4	100	0	1	0	Sedang	50.03

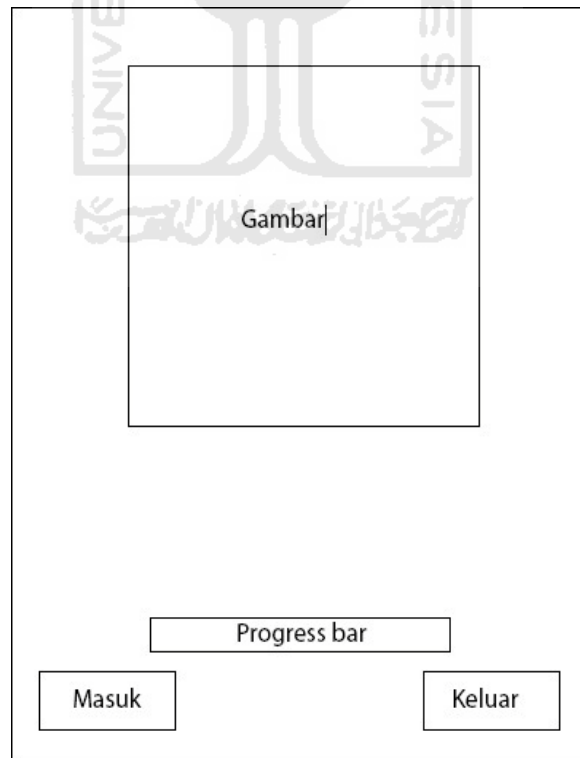
Sehingga hasil keputusan yang diambil berdasarkan jumlah keputusan terbanyak yakni : jumlah kadar kalori (sedang) > jumlah kadar kalori (rendah). Hasil keputusan pengklasifikasian : **Sedang**

3.2 PERENCANAAN ANTAR MUKA

Desain antar muka (*interface*) merupakan hasil implementasi dari kebutuhan pengguna. Desain *interface* yang utama ditujukan kepada pengguna, dimana *interface* didesain sedemikian rupa untuk memudahkan penggunaan sistem aplikasi ini. Desain sederhana aplikasi ini akan memberikan kenyamanan oleh pengguna itu sendiri. Berikut desain *interface* pada suatu analisis pengklasifikasian.

3.2.1 Halaman Utama

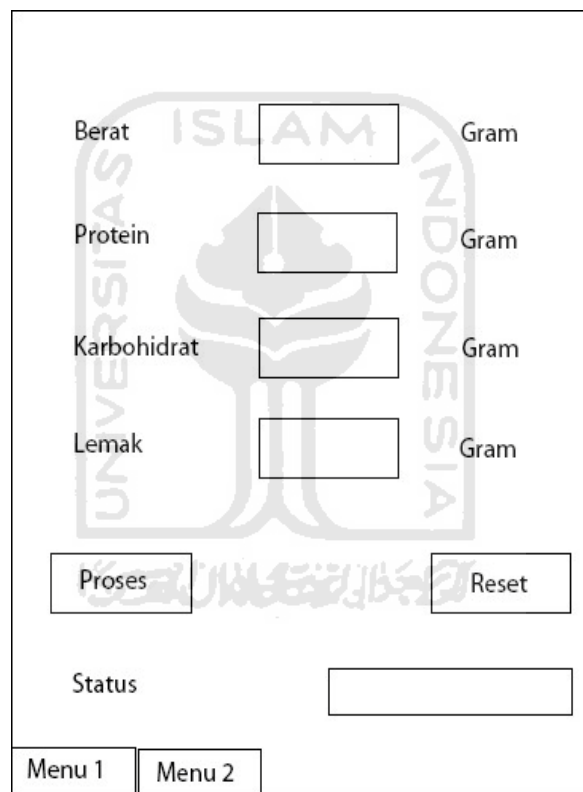
Halaman utama merupakan halaman depan dari Aplikasi Sistem Pendukung Keputusan Penentu Kadar Kalori Buah. Halaman ini didesain sebaik mungkin agar pengguna lebih mudah memahami atau bersifat *user friendly*. Berikut adalah rancangan dari halaman utama pada gambar 3.1



Gambar 3. 1 Rancangan Halaman Utama

3.2.2 Halaman Menghitung Nilai

Halaman Menghitung Kadar Kalori Buah merupakan halaman untuk menghitung kadar ,kalori pada tiap jenis buah yang akan dilakukan oleh pengguna. Di halaman ini untuk mengetahui jenis kadar kalori pada buah, pengguna terlebih dahulu harus menginput berat buah, nilai karbohidrat, protein, dan lemak. Berikut adalah rancangan dari halaman utama pada gambar 3.2



The image shows a user interface for calculating the calorie content of fruit. It features a central form with four input fields for 'Berat' (Weight), 'Protein', 'Karbohidrat' (Carbohydrate), and 'Lemak' (Fat), each followed by a 'Gram' unit label. Below these fields are two buttons: 'Proses' (Process) and 'Reset'. At the bottom, there is a 'Status' label followed by an empty text box, and two menu buttons labeled 'Menu 1' and 'Menu 2'. A large, faint watermark of the Universitas Islam Indonesia logo is visible in the background.

Gambar 3. 2 Rancangan Halaman Menghitung Nilai K

3.2.3 Halaman Pengolahan Data Sampel

Halaman Pengolahan Data Sampel merupakan halaman untuk mengetahui data yang telah ada didalam *database*. Di halaman ini terdapat inputan data diantaranya, nama buah, nilai berat, nilai protein, nilai karbohidrat, nilai lemak dan nilai kalori dimana pengguna dapat menambahkan dan mengedit data tersebut. Berikut adalah rancangan dari halaman utama pada gambar 3.3

The image shows a web form for sample data processing. It includes the following elements:

- Nama Buah:** A text input field.
- Berat:** A text input field followed by the unit "Gram" and a "Save" button.
- Protein:** A text input field followed by the unit "Gram" and an "Edit" button.
- Karbohidrat:** A text input field followed by the unit "Gram" and a ">" button.
- Lemak:** A text input field followed by the unit "Gram" and a "<" button.
- Kalori:** A text input field followed by the unit "Gram" and a "Reset" button.
- Jumlah Sampel:** A text input field.
- List Input Data:** A large rectangular area intended for displaying the data list.
- Navigation:** Two buttons labeled "Menu 1" and "Menu 2" at the bottom.

Gambar 3. 3 Rancangan Halaman Pengolahan Data Sampel

3.2.4 Halaman Pencarian Data Sampel

Halaman Pencarian Data Sampel merupakan halaman untuk mengetahui data keseluruhan yang terdapat didalam *database* Aplikasi Penentu Kadar Kalori Buah. Di halaman ini pengguna dapat mencari informasi yang ingin diketahui berdasarkan pencarian nama buah. Berikut adalah rancangan dari halaman utama pada gambar 3.4

Nama Buah	<input type="text"/>
<input type="button" value="Cari"/>	
List Data	
Nama Buah	<input type="text"/>
Berat	<input type="text"/> Gram
Protein	<input type="text"/> Gram
Kerbohidrat	<input type="text"/> Gram
Lemak	<input type="text"/> Gram
<input type="button" value="Menu 1"/>	<input type="button" value="Menu 2"/>

Gambar 3.4 Rancangan Halaman Pencarian Data Sampel

3.2.5 Halaman Pencarian Berdasarkan Status

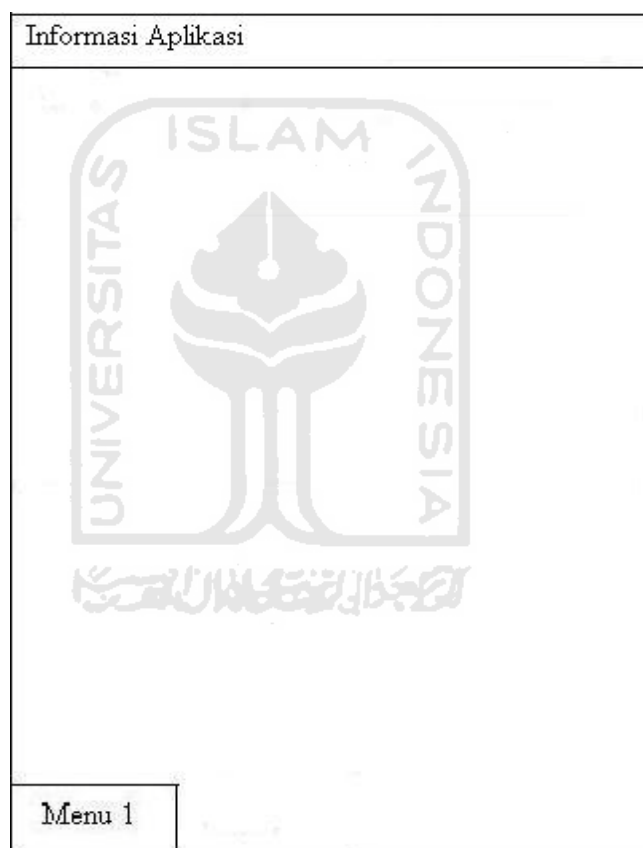
Halaman Pencarian Berdasarkan Status merupakan halaman untuk mengetahui berbagai macam data sample yang sudah ditentukan dalam *database* berupa status kadar kalori pada Aplikasi Penentu Kadar Kalori Buah. Di halaman ini pengguna dapat mencari informasi status kadar kalori buah yang diinginkan berdasarkan kategorinya. Berikut adalah rancangan dari halaman utama pada gambar 3.5

The image shows a wireframe of a web page. At the top, there is a header area with the text 'Status Kalori' on the left and a search input field on the right. Below the header is a large, empty rectangular area, likely intended for search results or a list of items. At the bottom of the page, there is a footer area containing two buttons labeled 'Menu 1' and 'Menu 2'. A large, faint watermark of the 'UNIVERSITAS ISLAM INDONESIA' logo is visible in the background of the page.

Gambar 3.5 Rancangan Halaman Pencarian Data

3.2.6 Halaman Informasi

Halaman Informasi merupakan halaman untuk mengetahui informasi-informasi yang terdapat dalam dalam sebuah sistem aplikasi. Halaman Informasi ini menerangkan cara kerja dan keterangan langkah yang harus dilakukan pengguna (*user*) sebelum menjalankan sebuah aplikasi. Berikut adalah rancangan dari halaman utama pada gambar 3.6



Gambar 3.6 Rancangan Halaman Informasi

BAB IV

SIMPULAN DAN SARAN

4.1 Kesimpulan

Setelah melalui tahapan analisis desain, maka dapat diambil beberapa kesimpulan sebagai berikut :

- a. Analisis dan Perancangan dari sebuah pengklasifikasian dapat memberikan suatu gambaran bagaimana sistem dapat dibuat.
- b. Analisis yang dibuat mampu memberikan informasi yang dapat mempermudah pengklasifikasian menjadi sebuah program.
- c. Analisis ini berguna untuk memberikan solusi dalam sebuah perancangan agar dapat menjadi lebih baik dan efisien.

4.2 Saran

Berdasarkan analisis yang telah dilakukan, masih banyak kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar kinerja perancangan lebih baik. Oleh karena itu untuk pengembangan ke depan disarankan agar analisis dan perancangan ini dapat :

- a. Memberikan informasi–informasi yang lebih akurat lagi agar dapat dianalisis dengan kemajuan teknologi yang sudah ada.
- b. Memberikan kemudahan dalam suatu perumusan analisis dan perancangan agar dapat lebih dipahami pada saat melakukan analisis.

DAFTAR PUSTAKA

- [ANI05a] "Anis. 2005. *Sistem Pendukung Keputusan Klinis (On-Line)* available at <http://fuadanis.blogspot.com>"
- [DAI01] "Dailini, D. Umar. 2001. *Komputerisasi Pengambilan Keputusan Jakarta Elex Media*
- [RNL06] [www.informatika.org/~rinaldi/Matdis/2006-2007/Makalah/Makalah0607-121.pdf].
- [ANI05b] Anis, 2005. *Sistem Pendukung Keputusan Klinis* available at <http://anisfuad.wordpress.com/2005/09/13/sistem-pendukung-keputusan-klinik/>
- [HAR05] Hartono, Andry. 2005. *Terapi Gizi dan Diet Rumah Sakit*, Yogyakarta; Buku Kedokteran.
- [KUS07] Kusumadewi, Sri. 2007. *Bahan Kuliah Informatika Kedokteran*, Yogyakarta; UII.
- [TRF04] Tom Radcliffe, 2003. *Manajemen Proyek dan Analisis Data. Bioinformatic* at <http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/>