



PENGUKURAN EFISIENSI PRODUKSI AIR MINERAL

DENGAN MENGGUNAKAN METODE
DATA ENVELOPMENT ANALYSIS



DAHAN ABDULLAH
MAYA SAVIRA
CUT ITA ERLIANA
BUNYAMIN
MUHAMMAD SYAHRUL KAHAR

Pengukuran Efisiensi Produksi Air Mineral dengan Menggunakan Metode *Data Envelopment Analysis*

**Dahlan Abdullah
Maya Savira
Cut Ita Erliana
Bunyami
Muhammad Syahrul Kahar**

Diterbitkan Oleh:



2020

Pengukuran Efisiensi Produksi Air Mineral dengan Menggunakan Metode *Data Envelopment Analysis*

Hak Cipta © 2020 pada

Penulis

Dahlan Abdullah
Maya Savira
Cut Ita Erliana
Bunyami
Muhammad Syahrul Kahar

Layout

T.M. siddiq^(SEFA)

Pracetak dan Produksi

CV. Sefa Bumi Persada

Hak Cipta dilindungi undang-undang.

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk memfotokopi, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis

Penerbit:

SEFA BUMI PERSADA

Jl. B. Aceh – Medan, Alue Awe - Lhokseumawe

email: sefabumipersada@gmail.com

Telp. 085260363550

Cetakan I : 2020

ISBN – 978-623-7648-43-7

1. Hal. 114: 16,5 x 7,5 cm

I. Judul

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah Subhanahu wa ta'ala, karena dengan limpahan rahmat dan hidayah-Nya penulis dapat menyelesaikan penelitian tepat pada waktunya. Dan tak lupa pula shalawat beriringi dengan salam marilah kita sama-sama sanjung sajikan atas kepangkuan Nabi Muhammad SAW yang mana telah membawa kita dari alam kebodohan ke alam yang berilmu pengetahuan seperti sekarang ini.

Dalam kesempatan ini penulis mengambil judul Penelitian **“Pengukuran Efisiensi Produksi Air Mineral dengan Menggunakan Metode *Data Envelopment Analysis*”**.

Penulis menyadari bahwa tanpa bantuan dari berbagai pihak, Penelitian ini tidak akan berjalan dengan baik. Penulis menyadari sepenuhnya bahwa dalam penulisan Penelitian ini masih jauh dari kesempurnaan, oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun demi kesempurnaan pada masa yang akan datang. Semoga Penelitian ini memberikan informasi dan bermanfaat untuk pengembangan wawasan dan peningkatan ilmu pengetahuan bagi kita semua. Akhir kata semoga Penelitian ini dapat bermanfaat bagi penulis khususnya dan pembaca pada umumnya.

Penulis

Dahlan Abdullah

DAFTAR ISI

| | Halaman |
|--|----------------|
| Kata Pengantar | i |
| Daftar Isi | ii |
| | |
| Bab I Pendahuluan | |
| A. Latar Belakang | 1 |
| B. Rumusan Masalah | 4 |
| C. Ruang Lingkup dan Batasan Penelitian | 4 |
| D. Tujuan Penelitian | 5 |
| E. Manfaat Penelitian | 6 |
| | |
| Bab II Tinjauan Pustaka | |
| A. Konsep Pengukuran Efisiensi..... | 7 |
| B. Produksi..... | 11 |
| C. Air Mineral Dalam Kemasan (AMDK)..... | 12 |
| D. Data Envelopment Analysis (DEA) | 13 |
| E. Flowchart | 16 |
| F. Unified Modeling Language (UML) | 18 |
| G. Web | 28 |
| H. XAMPP | 29 |
| I. Database | 32 |
| J. PHP (Page Hypertext Preprocessor)..... | 38 |
| K. SQL (Structured Query Language) | 43 |
| L. Mysql | 44 |
| M. Linear Interactive Discrete Optimizer (LINDO) | 58 |
| | |
| Bab III Metode Penelitian | |
| A. Tahapan (Waktu) Pelaksanaan Penelitian..... | 61 |
| B. Pengumpulan Data | 61 |
| C. Analisis dan Pengolahan Data | 62 |
| D. Skema Sistem | 64 |

Bab IV Hasil Dan Pembahasan

A. Hasil Penelitian/ Perancangan 66
B. Pembahasan..... 91
C. Implementasi Antar Muka 94

Bab V Kesimpulan Dan Saran

A. Kesimpulan 101
B. Saran 101

Daftar Pustaka..... 102

BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan di segala bidang telah menyebabkan perubahan gaya hidup dan pola konsumsi masyarakat. Ritme kehidupan yang menuntut segala sesuatu serba cepat dengan waktu yang terbatas, membutuhkan adanya jenis makanan dan minuman instan yang sehat untuk memenuhi asupan gizi masyarakat setiap hari. Dengan adanya makanan maupun minuman instan, akan tercipta suatu lingkungan hidup yang lebih efektif dan efisien.

Perkembangan teknologi dan industri yang pesat ternyata membawa dampak bagi kehidupan manusia, baik dampak positif maupun dampak negatif. Dampak yang positif memang sangat diharapkan bagi manusia dalam rangka meningkatkan kualitas dan kenyamanan hidup, namun dampak negatif yang tidak diharapkan karena dapat menurunkan kualitas dan kenyamanan hidup. Hal ini dapat dilihat dari perkembangan teknologi industri yakni, tersedianya air mineral yang banyak dikonsumsi oleh masyarakat. Dan juga penyediaan air minum, misalnya AMDK (Air Minum dalam Kemasan), serta persediaan Air Minum Isi Ulang (AMIU).

Secara tatanan ekonomi global, persaingan di dunia bisnis pada umumnya dilakukan berdasarkan produktivitas dan harga. Keberhasilan usaha ditentukan oleh kemampuan perusahaan untuk menghasilkan barang-barang dan jasa dalam jumlah yang makin besar dengan menggunakan sumber daya yang sama atau lebih sedikit. Meskipun keberhasilan usaha tidak hanya ditentukan oleh produktivitas dan harga saja, tetapi juga oleh kualitas, keanekaragaman, kesesuaian dengan selera kebutuhan pemakai, kegunaan, dan kemudahan yang ditawarkan serta ketepatan waktu.

Ketika mempertahankan bisnis perusahaan, seorang produsen dituntut untuk bekerja secara efisien agar keuntungan yang diperoleh menjadi lebih besar. Tuntutan bekerja secara efisien ini tidak dapat dihindari dalam bisnis modern, apalagi seringkali dijumpai biaya produksi dirasa terus meningkat sementara nilai produksi masih tetap.

Suatu perusahaan harus mampu memahami minat konsumen terhadap produk, mengenal keinginan konsumen dan hal-hal yang mempengaruhi minat konsumen terhadap produk yang ditawarkan. Di antaranya ada beberapa faktor yang mempengaruhi kepuasan konsumen seperti, kinerja dan keandalan produk atau jasa, citra merek (*brand image*), sistem antara *supplier*, perusahaan, perantara, total nilai, (*total customer value*) diterima konsumen. Karena kepuasan pelanggan merupakan suatu konsep yang telah lama dikenal dalam ilmu pemasaran. Manajern harus memilih kombinasi variabel marketing *mix* yang memberikan *cost effective* secara optimal. Dan produsen dalam hal tersebut mempunyai tanggung jawab yang besar.

Bisnis Air Minum Dalam Kemasan (AMDK) merupakan salah satu bisnis yang cukup menjanjikan pada saat ini, yang mana kebutuhan terhadap air minum dan pola hidup instan menjadi alternatif bagi konsumen untuk mencari Air Minum Dalam Kemasan (AMDK) dan menembus pasar sasaran. Peningkatan efisiensi di bagian produksi merupakan salah satu cara yang bisa dilakukan oleh pihak perusahaan agar selalu dapat memenuhi permintaan konsumen. Guna menghadapi banyaknya para pesaing perusahaan yang menghasilkan produk yang sama, perusahaan perlu melakukan suatu cara untuk menjalankan proses produksi yang efisien, yaitu bagaimana menggunakan *input* sehemat mungkin untuk menghasilkan *output* yang sesuai atau melebihi target permintaan yang telah ditetapkan. Menurut (Sardjono 2017) Produksi adalah mengubah *input* menjadi *output*.

Produksi merupakan hasil akhir dari proses atau aktivitas ekonomi dengan memanfaatkan beberapa masukan atau *input*. Kegunaan suatu barang akan bertambah bila memberikan manfaat baru atau lebih dari bentuk semula. Lebih spesifik lagi produksi adalah kegiatan perusahaan dengan mengkombinasikan berbagai *input* untuk menghasilkan *output* dengan biaya yang minimum. Dengan demikian produksi tidak terbatas pada pembuatan, tetapi sampai pada distribusi.

Oleh sebab itu, perlu dilakukan analisa efisiensi produktivitas, sehingga perusahaan tersebut dapat melakukan langkah-langkah perbaikan secara berkesinambungan ke arah peningkatan efisiensi tersebut. Hal penting dalam peningkatan efisiensi adalah penetapan target *input* dan *output* yang diperlukan bagi manajemen dalam melakukan monitoring setelah pengukuran kebutuhan pasar. Dengan menetapkan target, akan dapat dilakukan perbaikan *input* dan *output* untuk meningkatkan produktivitas pada tiap-tiap perusahaan tersebut.

Dengan adanya masalah tersebut maka dilakukan penelitian dengan menggunakan metode *Data Envelopment Analysis* (DEA) sehingga dapat diketahui efisiensi masing-masing perusahaan AMDK yang ada di wilayah Aceh dan menentukan strategi perbaikan bagi perusahaan tersebut yang inefisiensi.

DEA merupakan salah satu pendekatan yang dapat digunakan dalam pengukuran efisiensi perusahaan dengan kelebihan yaitu mengakomodasikan banyak *input* maupun *output* dalam banyak dimensi, sehingga akan didapatkan suatu pengukuran efisiensi yang lebih akurat sebagai langkah awal dalam meningkatkan pemasaran. DEA diaplikasikan secara luas dalam evaluasi kinerja pada institusi pendidikan, rumah sakit, cabang bank, *production plan* dan lain-lain.

Dalam menyelesaikan permasalahan ini, penulis menggunakan bahasa pemrograman PHP (*Personal Home Page*).

Bahasa pemrograman PHP adalah bahasa pemrograman yang bekerja dalam sebuah *web server*. *Script-script* PHP yang dibuat harus tersimpan dalam sebuah *server* dan dieksekusi atau diproses dalam *server* tersebut. Penggunaan program PHP memungkinkan sebuah *website* menjadi lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *website/computer client* akan diolah dan disimpan dalam *database web server* dan bisa ditampilkan kembali apabila di akses.

Untuk membantu penulis setelah menggunakan bahasa pemrograman PHP, penulis menggunakan *database Mysql* sebagai *database* dalam penelitian ini. Menurut (Bunafit 2005), *Mysql* adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar *SQL (Structured Query Language)*. *Mysql* dapat juga berperan sebagai *client*, sehingga sering disebut *database client server*.

B. Rumusan Masalah

Berdasarkan latar belakang masalah, maka penulis merumuskan masalah yang akan dibahas yaitu:

1. Bagaimana menerapkan metode *Data Envelopment Analysis* untuk mengukur efisiensi produksi air mineral di wilayah Aceh Utara dan Kota Lhokseumawe?
2. Bagaimana membangun aplikasi untuk mengukur efisiensi produksi air mineral di wilayah Aceh menggunakan metode *Data Envelopment Analysis* berbasis *web*?

C. Ruang Lingkup dan Batasan Penelitian/Perancangan

Sehubungan dengan keterbatasan yang dimiliki, setiap penelitian diperlukan penajaman kepada permasalahan agar tidak menyimpang dari tujuan yang ingin dicapai dengan membatasi permasalahan yang ada, adapun ruang lingkup dan batasan penelitian dalam penyelesaian penelitian ini adalah:

1. Data penelitian yang digunakan pada sistem ini hanya berfokus pada efisiensi produksi Air Mineral Dalam Kemasan (AMDK) di Wilayah Aceh Utara dan Kota Lhokseumawe.
2. Sistem ini dibangun dengan menggunakan bahasa pemrograman PHP, dengan *database Mysql*.
3. Metode yang digunakan dalam sistem ini ada metode *Data Envelopment Analysis*.
4. Perancangan sistem ini menggunakan *Unified Modelling Language*.
5. *Input* dari sistem ini adalah jumlah tenaga kerja, jumlah jam kerja produksi, biaya bahan baku produksi, jumlah *distributor/supplier*.
6. *Output* dari sistem ini adalah informasi dan nilai efisiensi suatu perusahaan produksi Air Mineral Dalam Kemasan (AMDK) di wilayah Aceh Utara dan Kota Lhokseumawe, yaitu jumlah produk, jumlah pelanggan, dan juga nilai kepuasan pelanggan.
7. Dalam pengambilan data nilai kepuasan akan dilakukan dengan cara pengisian kuisioner untuk 100 orang.
8. Perusahaan air mineral yang akan diteliti adalah Perusahaan Air Mineral AiniQua, Mount Aqua, Wendy's, dan Ie Ro.
9. Kemasan yang akan diteliti adalah kemasan air mineral 240 ml, 330 ml, 600 ml, dan 1500 ml.
10. Permasalahan *Linear Programming* diselesaikan menggunakan bentuk program komputer *Linear Interactive Discreate Optimizer - LINDO* versi 6.1.

D. Tujuan Penelitian

Berdasarkan perumusan masalah, maka yang menjadi tujuan dari penelitian ini adalah:

1. Untuk mengetahui bagaimana penerapan metode *Data Envelopment Analysis* untuk mengukur efisiensi produksi air mineral di wilayah Aceh Utara dan Kota Lhokseumawe.
2. Untuk membangun aplikasi pengukur efisiensi produksi air mineral di wilayah Aceh Utara dan Kota Lhokseumawe menggunakan metode *Data Envelopment Analysis* berbasis *web*.

E. Manfaat Penelitian

Dengan tercapainya tujuan penelitian di atas, maka diharapkan akan bermanfaat bagi semua pihak, yaitu:

1. Bagi Penulis

Sebagai sarana untuk menerapkan pengetahuan yang diperoleh selama menempuh studi, khususnya di dalam perancangan dan pembuatan aplikasi berbasis *web*.

2. Bagi pengguna

Manfaat penelitian ini bagi pengguna aplikasi aplikasi pengukur efisiensi produksi air mineral di wilayah Aceh Utara dan Kota Lhokseumawe menggunakan metode *Data Envelopment Analysis* berbasis *web* ini adalah untuk meningkatkan efisiensi suatu perusahaan agar memperoleh keuntungan yang lebih baik.

3. Bagi Pembaca

Adapun manfaat penelitian ini bagi pembaca adalah sebagai sarana informasi dan acuan untuk melakukan penelitian selanjutnya dan sebagai bahan referensi serta dapat memberikan informasi dan ilmu pengetahuan mengenai bagaimana cara mengetahui proses pembuatan aplikasi pengukuran efisiensi produksi ini.

BAB II

TINJAUAN PUSTAKA

A. Konsep Pengukuran Efisiensi

Pengukuran dapat diartikan sebagai proses memasangkan fakta-fakta suatu obyek dengan satuan-satuan ukuran tertentu. Sedangkan efisiensi merupakan perbandingan *output* dan *input* berhubungan dengan tercapainya *output* maksimum dengan sejumlah *input*, hal ini mengimplikasikan jika rasio *output-input* besar, maka efisiensi dikatakan semakin tinggi (Aumora, Bakce dan Dewi 2016).

Menurut Shafique dkk. dalam (Tuffahati et al, 2016) Pengukuran efisiensi dapat membantu suatu entitas untuk menilai dan mengevaluasi kinerja serta kemampuan daya saingnya dalam industri. Seberapa besar entitas tersebut dapat mengatasi tantangan dalam industrinya dan mampu bersaing serta bertahan bahkan mengembangkan entitasnya di masa depan. Pengukuran efisiensi dapat dilakukan dengan cara menilai daya saing dari bagian *output vs input* maupun beban *vs return*.

Konsep efisiensi terdiri dari beberapa pengertian, yaitu efisiensi teknis, efisiensi harga, dan efisiensi ekonomi. Efisiensi teknis adalah perbandingan antara produksi aktual dengan tingkat produksi potensial yang dapat dicapai. Efisiensi alokatif mengukur kemampuan perusahaan untuk menggunakan *input* dengan proporsi yang optimal pada masing-masing tingkat harga *input* dan teknologi yang dimiliki. Efisiensi ekonomi adalah besaran yang menunjukkan perbandingan antara keuntungan yang sebenarnya dengan keuntungan maksimum.

Menurut kamus besar Ekonomi (Wiharno dan Ismaya 2013) menyatakan bahwa efisiensi adalah hubungan atau perbandingan antara faktor keluaran (*output*) barang dan jasa dengan masukan (*input*) yang langka di dalam suatu unit kerja,

atau ketetapan cara (usaha, kerja) dalam menjalankan sesuatu (dengan tidak membuang-buang waktu, tenaga, biaya.

Efisiensi adalah suatu istilah yang sifatnya relatif, yaitu selalu harus dikaitkan dengan kriteria tertentu. Ahli ekonomi melihat efisiensi dari dua sudut pandang, sudut pandang positif dan normatif. Pandangan positif didasarkan pada perilaku manusia yang selalu mencari peningkatan nilai atau *value* (*utility maximization dan profit maximization theory*). Pencarian *value* adalah pendorong terciptanya mekanisme pasar.

Jika tercapai suatu situasi dimana masih ada *value* yang belum tereksplorasi, perilaku manusia adalah selalu berusaha mencari jalan untuk mencapai *value* tersebut. Pandangan normatif berakar dari keinginan untuk membuat kebijakan. Untuk menilai apakah kebijakan yang satu lebih baik dari pada kebijakan yang lainnya, dibutuhkan suatu dasar untuk perbandingan.

Efisiensi dikatakan sebagai kemampuan untuk menyelesaikan suatu pekerjaan dengan benar, atau dalam pandangan matematika didefinisikan sebagai perhitungan rasio *output* dan *input* atau jumlah keluaran yang dihasilkan dari suatu masukan yang digunakan. Suatu hasil dikatakan efisien apabila nilai efisiensi sama dengan satu (nilai efisiensi $(t) = 1$). Namun, jika nilai efisiensi $(t) > 1$ dapat diartikan bahwa penggunaan *input* belum efisien sehingga untuk mencapai nilai efisiensi perlu untuk menambah *input*. Jika nilai efisiensi $(t) < 1$ dapat diartikan bahwa penggunaan *input* tidak efisien sehingga untuk mencapai nilai efisiensi perlu untuk mengurangi *input*. Ada tiga faktor yang menyebabkan efisiensi tinggi yaitu:

1. Apabila dengan *input* yang sama dapat menghasilkan *output* yang lebih besar.
2. *Input* yang lebih kecil menghasilkan *output* yang sama.
3. Dengan *input* yang lebih besar dapat menghasilkan *output* yang jauh lebih besar.

Menurut (Israwan dan dkk. 2016) secara umum efisiensi merupakan perbandingan antara *output* dengan *input*, atau dalam rumus :

$$Efisiensi = \frac{Output}{Input} \dots\dots\dots (2.1)$$

Di mana :

Input = sumber daya yang digunakan

Output = hasil yang dicapai

Efisiensi sebuah perusahaan terdiri dari dua komponen yaitu: (1) *technical efficiency* dan (2) *allocative efficiency*. *Technical efficiency* menggambarkan kemampuan perusahaan untuk memilih kombinasi *input* yang optimal pada tingkat harga dan teknologi tertentu. Efisiensi teknis (*technical efficiency*) memusatkan perhatian pada kemampuan perusahaan menggunakan *input* dalam menghasilkan *output* dibandingkan dengan best practise.

Sedangkan efisiensi alokatif (*allocative efficiency*) mencerminkan kemampuan perusahaan dalam mengoptimalkan penggunaan *inputnya*, dengan struktur harga dan teknologi produksinya. Kedua ukuran ini yang kemudian dikombinasikan menjadi efisiensi ekonomi (*economic efficiency*). Suatu perusahaan dapat dikatakan efisien secara ekonomi jika dapat meminimalkan biaya produksi untuk menghasilkan *output* tertentu dengan suatu tingkat teknologi yang umumnya digunakan serta harga pasar yang berlaku.

Pada efisiensi terdapat tiga jenis pendekatan pengukuran yang biasa digunakan, yaitu:

1) Pendekatan rasio

Pendekatan rasio yaitu dengan membandingkan *output* yang dihasilkan dengan *input* yang digunakan. Dalam pendekatan ini, efisiensi yang tinggi apabila dapat memproduksi jumlah *output* yang maksimal dengan *input* yang seminimal mungkin.

2) Pendekatan regresi

Pendekatan regresi yaitu dalam mengukur efisiensi dalam pendekatan ini menggunakan sebuah metode dari tingkat *output* tertentu sebagai fungsi dari berbagai tingkat input tertentu. Dalam pendekatan ini, akan dinilai efisiensi apabila dapat menghasilkan *output* lebih banyak dibandingkan estimasinya. Namun dalam pendekatan ini tidak dapat mengukur efisiensi dengan jumlah *output* yang banyak, karena hanya satu *output* yang menjadi indikator.

3) Pendekatan *Frontier*

Pendekatan *Frontier* dalam mengukur tingkat efisiensi dibedakan menjadi dua jenis, yaitu pendekatan *Frontier parametric* dan *nonparametric*. Pendekatan parametrik diukur dengan tes statistik parametrik seperti menggunakan *Stochastic Frontier Approach* (SFA) dan *Distribusi Free Approach* (DFA). Sedangkan pendekatan *Frontier non parametric* diukur dengan tes statistik non parametrik yaitu dengan menggunakan metode Data Envelopment Analysis (DEA).

Selain tiga pendekatan yang telah tersebut, terdapat tiga pendekatan lain yang biasa digunakan dalam metode *parametric* dan metode *nonparametric* dalam mengukur hubungan antara *input* dan *output* dalam kegiatan suatu lembaga keuangan. Tiga pendekatan tersebut yaitu:

1) Pendekatan Aset (*The Assets Approach*)

Pendekatan aset menggambarkan fungsi utama suatu lembaga keuangan sebagai pemberi pinjaman. Pada pendekatan ini, *output* yang diukur benar-benar didefinisikan kedalam bentuk aset.

2) Pendekatan Produksi (*The Production Approach*)

Pendekatan produksi menggambarkan suatu lembaga keuangan disini sebagai produsen dari akun depositonya dan kredit pinjamannya, lalu mendefinisikan

output sebagai jumlah tenaga kerja, pengeluaran modal pada aset tetap dan material lainnya.

3) Pendekatan Intermediasi (*The Intermediation Approach*)

Pendekatan intermediasi menggambarkan suatu lembaga keuangan sebagai *intermediator* atau perantara antara unit *surplus* dengan unit *defisit*. Dalam kondisi ini biasanya input yang digunakan yaitu biaya tenaga kerja, modal dan pembayaran bunga pada deposito. Dan *output* yang digunakan biasanya kredit pinjaman ataupun pendapatan investasi.

Dengan adanya tiga pendekatan ini, maka dalam mengukur efisiensi lembaga keuangan keputusan untuk menentukan variabel *input* dan variabel *output* yang digunakan pun akan berbeda. Dalam penelitian ini, digunakan pendekatan produksi (*The Production Approach*) karena untuk menghitung tingkat efisiensi suatu produksi atau produk, yaitu Air Mineral.

B. Produksi

Produksi adalah suatu kegiatan yang mengubah *input* menjadi *output*. Kegiatan tersebut dalam ekonomi biasa dinyatakan dalam fungsi produk. Fungsi produk menunjukkan jumlah maksimum *output* yang dapat dihasilkan dari pemakaian sejumlah *input* dengan menggunakan teknologi tertentu. Produksi sering didefinisikan sebagai penciptaan guna, dimana guna berarti kemampuan barang atau jasa untuk memenuhi kebutuhan manusia (Manalu, et al. 2017)

Kegiatan produksi merupakan salah satu aktivitas ekonomi yang sangat menunjang selain kegiatan konsumsi. Tanpa kegiatan produksi, maka konsumen tidak akan dapat mengkonsumsi barang dan jasa yang dibutuhkannya. Kegiatan produksi dan konsumsi adalah satu mata rantai yang saling berkaitan dan tidak bisa saling dilepaskan.

Produksi adalah kegiatan yang dilakukan manusia dalam menghasilkan suatu produk baik barang, maupun jasa yang

kemudian dimanfaatkan oleh konsumen. Pada saat kebutuhan manusia masih sedikit dan sederhana, kegiatan produksi dan konsumsi sering kali dilakukan sendiri. Namun, seiring berjalannya waktu keterbatasan sumber daya, maka seseorang tidak dapat lagi memproduksi sendiri barang dan jasa yang dibutuhkannya, sehingga ia membutuhkan pihak lain untuk memproduksi apa yang menjadi kebutuhannya tersebut.

Menurut (Sardjono 2017) produksi adalah kegiatan mengubah *input* menjadi *output*. Produksi pada dasarnya merupakan proses penciptaan atau penambahan faedah bentuk, waktu dan tempat atas faktor-faktor produksi sehingga dapat lebih bermanfaat bagi pemenuhan kebutuhan manusia. Proses perubahan bentuk faktor-faktor produksi tersebut disebut proses produksi. Selain itu produksi dapat ditinjau dari dua pengertian, yaitu pengertian secara teknis dan pengertian secara ekonomis.

Ditinjau dari pengertian secara teknis, produksi merupakan proses pendayagunaan sumber-sumber yang telah tersedia guna memperoleh hasil yang lebih dari segala pengorbanan yang telah diberikan. Sedangkan bila ditinjau dari pengertian secara ekonomis, produksi merupakan suatu proses pendayagunaan segala sumber yang tersedia untuk memperoleh hasil yang terjamin kualitas maupun kuantitasnya, terkelola dengan baik sehingga merupakan komoditi yang dapat diperdagangkan. Adanya hubungan antara faktor-faktor produksi yang digunakan dengan *output* yang dihasilkan dinyatakan dalam suatu fungsi produksi.

C. Air Mineral Dalam Kemasan (AMDK)

Berdasarkan Surat Keputusan Menteri Perindustrian dan Perdagangan RI No.705/MPP/Kep/11/2003 AMDK atau Air Minum Dalam Kemasan adalah air baku yang sudah diproses tanpa bahan pangan lainnya dan bahan tambahan pangan, dikemas, sehingga aman untuk diminum dan air baku

itu sendiri merupakan air yang telah memenuhi persyaratan kualitas air bersih untuk diolah menjadi produk Air Minum Dalam Kemasan.

Air Minum Dalam Kemasan merupakan air baku yang sudah diproses, dikemas dan aman untuk diminum mencakup air mineral dan juga air *demineral*. Air mineral merupakan air minum dalam kemasan yang mengandung mineral dalam jumlah tertentu tanpa menambahkan mineral. Air *demineral* merupakan air minum di dalam kemasan yang didapatkan melalui proses pemurnian secara *destilasi, deionisasi, reverse osmosis* (RO).

D. Data Envelopment Analysis (DEA)

1. Pengertian DEA

Data Envelopment Analysis atau disingkat DEA pertama kali diperkenalkan pada tahun 1978 oleh Charnes A, Cooper WW dan Rhodes E dalam jurnal *Operational Research* dengan judul "*Measuring The Efficiency of Decision Making Units*". Jurnal tersebut membahas pengembangan langkah-langkah pengambilan keputusan efisiensi yang dapat digunakan dalam mengevaluasi Unit Pengambil Keputusan.

Data Envelopment Analysis (DEA) adalah sebuah teknik pemrograman matematis berdasarkan pada *Linear Programming* yang digunakan untuk mengevaluasi efisiensi dari suatu unit pengambilan keputusan (unit kerja) yang bertanggung jawab menggunakan sejumlah *input* untuk memperoleh suatu *output* yang ditargetkan. Metode DEA diciptakan sebagai alat evaluasi kinerja suatu aktivitas di sebuah unit *entitas* (organisasi) yang selanjutnya disebut *Decision Making Unit* (DMU). Secara sederhana, pengukuran ini dinyatakan dengan rasio: *output/input*, yang merupakan suatu pengukuran efisiensi atau produktivita (Filardo, Negoro dan Kunaifi 2017)s.

Menurut (Permatasari 2018) langkah pengukuran nilai efisiensi pada metode DEA :

1. Melakukan DMU penentuan dan mengidentifikasi DMU yang akan dievaluasi.
2. Memutuskan *input* dan *output* DMU.
3. Melakukan analisis untuk mendapatkan nilai efisiensi relatif.

Pengukuran berorientasi *input* menunjukkan sejumlah *input* dapat dikurangi secara proporsional tanpa mengubah jumlah *output* yang dihasilkan. Dengan menggunakan model berorientasi *input*, maka model tersebut akan menghitung pengurangan *input* yang diperlukan untuk menghasilkan kinerja yang efisien dengan *output* yang tetap.

Terdapat tiga manfaat yang diperoleh dari pengukuran efisiensi dengan DEA, yaitu:

1. Sebagai tolak ukur untuk memperoleh efisiensi relatif yang berguna untuk mempermudah perbandingan antar unit ekonomi yang sama.
2. Mengukur berbagai variasi efisiensi antar unit ekonomi untuk mengidentifikasi faktor-faktor penyebabnya.
3. Menentukan implikasi kebijakan sehingga dapat meningkatkan tingkat efisiensinya.

Keuntungan lainnya bahwa DEA dapat melihat sumber ketidakefisienan dengan ukuran peningkatan potensial (*potential improvement*) dari masing-masing *input*. DEA menghitung efisiensi teknis untuk seluruh unit. Skor efisiensi untuk setiap unit adalah relatif, tergantung pada tingkat efisiensi dari unit-unit lainnya di dalam sampel. Setiap unit dalam sampel dianggap memiliki tingkat efisiensi yang tidak *negatif* dan nilainya antara 0 hingga 1, di mana satu menunjukkan efisiensi yang sempurna.

2. Model DEA CCR (Charnes-Cooper-Rhodes)

Pertama kalinya model CCR ditemukan oleh Charnes, Cooper dan Rhodes pada tahun 1978. Pada model ini diperkenalkan suatu ukuran efisiensi untuk masing-masing *Decision Making Unit* (DMU) yang merupakan rasio maksimum antara *output* yang terbobot dengan *input* yang terbobot. Masing-masing nilai bobot yang digunakan dalam rasio tersebut ditentukan dengan batasan bahwa rasio yang sama untuk tiap DMU harus memiliki nilai yang kurang dari atau sama dengan satu.

Dengan demikian akan mereduksi *multiple input* dan *multiple output* ke dalam satu "virtual" *input* dan "virtual" *output* tanpa membutuhkan penentuan awal nilai bobot. Oleh karena itu ukuran efisiensi merupakan suatu fungsi nilai bobot dari kombinasi *virtual input* dan *virtual output*. Ukuran efisiensi DMU dapat dihitung dengan menyelesaikan permasalahan programming matematika berikut ini:

$$\max \frac{\sum_{k=1}^s v_k y_{kp}}{\sum_{j=1}^m u_j x_{jp}} \dots\dots\dots(2.2)$$

$$\text{s.t. } \frac{\sum_{k=1}^s v_k y_{ki}}{\sum_{j=1}^m u_j x_{ji}} \leq 1 \dots\dots\dots(2.3)$$

$$v_k, u_j \geq 0 \dots\dots\dots(2.4)$$

Di mana:

- x_{ji} = Nilai *input* ke- j yang digunakan DMU ke- i
- y_{ki} = Nilai *Output* ke - k yang digunakan DMU ke- i
- u_j = bobot untuk *input* j
- v_k = bobot untuk *output* k

persamaan (2.2), (2.3), dan (2.4) merupakan persamaan *non linear* atau persamaan *linear fraksional*, yang kemudian di transformasikan ke dalam bentuk *linear* sehingga dapat diaplikasikan dalam persamaan *linear* sebagai berikut:

$$\max \sum_{k=1}^s v_k y_{kp} \dots\dots\dots(2.5)$$

s.t.

$$\sum_{j=1}^m v_j x_{jp} = 1 \dots\dots\dots(2.6)$$

$$\sum_k v_k y_{ki} - \sum_{j=1}^m u_j x_{ji} \leq 0 \dots\dots\dots(2.7)$$

$$v_k, u_j \geq 0 \dots\dots\dots(2.8)$$



Perhitungan efisiensi dengan menggunakan model DEA CCR yang akan dilakukan akan diketahui DMU-DMU yang dianggap efisien maupun kurang efisien dengan mengacu pada hasil perhitungan nilai efisien model matematis DEA CCR dimana penentuannya berdasarkan ketentuan sebagai berikut: Jika efisiensi relatif $(hk) = 1$ maka DMU dinyatakan efisien, sedangkan jika efisiensi *relative* $(hk) < 1$ maka DMU tersebut dinyatakan tidak efisien.




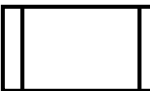
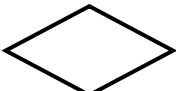
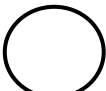



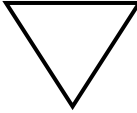

E. Flowchart

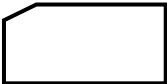




Diagram alir atau *Flowchart* merupakan bagian yang memperlihatkan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis (Iswandy 2015).

Fungsi *Flowchart* adalah untuk menggambarkan, menyederhanakan rangkaian proses atau prosedur sehingga mudah dipahami dan mudah dilihat berdasarkan urutan langkah dari suatu proses.

Tabel 2. 1 Simbol-Simbol *Flowchart*

| SIMBOL | NAMA | FUNGSI |
|---|-------------------------------------|---------------------------|
|  | TERMINATOR | Permulaan / akhir program |
|  | GARIS ALIRN (<i>FLOW LINE</i>) | Arah aliran program |

| | | |
|---|---|---|
|  | <i>PREPARATION</i> | Proses inisialisasi / pemberian harga awal |
|  | PROSES | Proses Perhitungan / proses pengolahan data |
|  | <i>INPUT/OUTPUT DATA</i> | Proses <i>input / output</i> , parameter, informasi |
|  | <i>PREDEFINED PROCESS (SUB PROGRAM)</i> | Permulaan <i>sub program</i> / proses menjalankan <i>sub program</i> |
|  | <i>DECISION</i> | Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya |
|  | <i>ON PAGE CONNECTOR</i> | Penghubung bagian-bagian <i>Flowchart</i> yang berada pada satu halaman |
|  | <i>OFF PAGE CONNECTOR</i> | Penghubung bagian-bagian <i>Flowchart</i> yang berada pada halaman berbeda |
|  | MANUAL | Simbol manual menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer |
|  | <i>KEYING OPERATION</i> | <i>Keying Operation</i> menyatakan segala jenis operasi yang diproses dengan menggunakan mesin yang mempunyai <i>keyboard</i> |
|  | <i>OFFLINE STORAGE</i> | Data dalam simbol ini akan disimpan ke suatu media tertentu |
|  | MANUAL INPUT | Memasukkan data secara manual dengan menggunakan <i>online keyboard</i> |

| | | |
|---|--------------------------|--|
|  | <i>PUNCHED CARD</i> | <i>Input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu |
|  | <i>MAGNETIC TAPE</i> | <i>Input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis |
|  | <i>DISK STORAGE</i> | <i>Input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> |
|  | <i>DOCUMENT</i> | Mencetak keluaran dalam bentuk dokumen melalui <i>printer</i> |
|  | <i>DISPLAY</i> | Mencetak keluaran dalam layar monitor |
| (sumber: Ali Ridho Barakbah dkk, 2013) | | |

F. Unified Modeling Language (UML)

1. Sejarah Unified Modeling Language (UML)

Tahun 1994, Grady Boch dan James Rumbaugh bergabung untuk menggunakan metode berorientasi objek. Ivan Jacobson bergabung pada tahun 1995, dan mereka bertiga fokus membuat suatu bahasa pemodelan objek standar sebagai ganti dari pendekatan atau metode objek standar. Berdasarkan kerja mereka dan hasil kerja lainnya pada industri, *Unified Modeling Language* (UML) versi 1.0 dirilis pada tahun 1997.

Unified Modeling Language (UML) tidak menentukan metode untuk sistem-sistem pengembangan, tetapi sudah diterima luas sebagai standar untuk pemodelan objek. *Object Management Group/OMG*, badan standar industri, mengadopsi UML pada bulan November 1997 dan terus bekerja sama untuk meningkatkannya berdasarkan kebutuhan

industri. Pada saat ini, salah satu industri telah merilis sebuah software yang mendukung UML yaitu Visual Paradigm Enterprise edition.

2. Definisi Unified Modelling Language (UML)

UML (*Unified Modelling Language*) adalah sebuah bahasa untuk menentukan visualisasi, konstruksi dan mendokumentasikan artifak (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak, Artifak dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya (Hardian 2014).

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya bidang yang membutuhkan pemodelan (Hardian 2014).

Unified Modeling Language (UML) biasa digunakan untuk (Henderi 2007):

- a. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *Use Case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur statik sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model *behavior* "yang menggambarkan kebiasaan atau sifat sebuah sistem" dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
- f. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk (Henderi et al., 2008):

1. Mengkomunikasikan ide
2. Melahirkan ide-ide baru dan peluang-peluang baru
3. Menguji ide dan membuat prediksi
4. Memahami struktur dan relasi-relasinya

3. Jenis-jenis Unified Modelling Language (UML)

a. *Class Diagram* (Diagram Kelas)

Class adalah kumpulan objek-objek dengan dan yang mempunyai struktur umum, *behavior* umum, relasi umum, dan *semantic*/kata yang umum. *Class-class* ditentukan/ditemukan dengan cara memeriksa objek-objek dalam *Sequence diagram* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan. *Class* sebaiknya diberi nama menggunakan kata benda sesuai dengan domain/bagian/ kelompoknya (Jeffery L. Whitten 2004).

Class Diagram adalah diagram yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Karena itu *class diagram* merupakan tulang punggung

atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML (Henderi 2008). Sementara (Jeffery L. Whitten 2004) *class diagram* adalah gambar grafis mengenai struktur objek statis dari suatu sistem, menunjukkan *class-class* objek yang menyusun sebuah sistem dan juga hubungan antara *class* objek tersebut.

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *Method* atau operasi. Berikut penjelasan atribut dan *Method*:

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *Method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

b. *Object Diagram* (Diagram Objek)




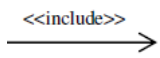
Diagram objek bersifat statis. Diagram ini memperlihatkan objek-objek serta relasi antar objek. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas.


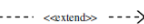
c. *Use Case Diagram*

Use Case atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*Behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Use Case diagram adalah model fungsional sebuah sistem yang menggunakan *actor* dan *use case*. *Use Case* adalah layanan (*services*) atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunanya (Henderi et al, 2008). *Use Case* adalah suatu pola atau gambaran yang menunjukkan kelakuan atau kebiasaan sistem. Setiap *Use Case* adalah suatu urutan (*Sequence*) transaksi yang saling berhubungan dan dilakukan oleh sebuah *actor* dan sistem dalam bentuk sebuah dialog (Henderi, 2007). *Use Case Diagram* dibuat untuk memvisualisasikan/ menggambarkan hubungan antara *Actor* dan *Use Case*. *Use Case* diagram mempresentasikan kegunaan atau fungsi-fungsi sistem dari perspektif pengguna.

Tabel 2. 2 Notasi *Use Case Diagram*

| Simbol | Nama Simbol | Keterangan |
|---|--------------------|--|
|  | <i>Use Case</i> | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>Use Case</i> . |
|  | <i>Association</i> | Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan aktor. |
|  | <i>Actor</i> | komponen yang dapat Berinteraksi dengan sistem |
|  | <i>Include</i> | Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini. |

| | | |
|---|-----------------------|--|
|  | <i>Generalization</i> | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya |
|  | <i>Extend</i> | Relasi <i>Use Case</i> tambahan ke-sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan itu, biasanya <i>Use Case</i> tambahan memiliki nama depan yang sama dengan <i>Use Case</i> yang ditambahkan. |
| (sumber: Rosa A.S dan M. Shalahuddin, 2014) | | |

d. *Sequence Diagram* (Diagram Urutan)

Sequence Diagram menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dengan *Message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *Sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram *Sequence* juga dibutuhkan untuk melihat skenario yang ada pada *Use Case*. Banyaknya diagram *Sequence* yang harus digambar adalah minimal sebanyak pendefinisian *Use Case* yang memiliki proses sendiri atau yang penting semua *Use Case* yang telah didefinisikan interaksinya pesan sudah dicakup dalam diagram *Sequence* sehingga semakin banyak *Use Case* yang didefinisikan maka diagram *Sequence* yang harus dibuat juga semakin banyak (Rosa and Shalahuddin 2014).

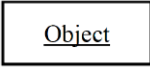



Sequence Diagram adalah suatu diagram yang memperlihatkan/menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah

urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa “pesan/*message*” (Henderi, 2008). *Sequence* diagram dapat juga didefinisikan sebagai suatu diagram yang menggambarkan interaksi-interaksi yang ada antar *class* dalam suatu hubungan perubahan dengan sebuah pesan pada akhir waktu (Miller Randy, 2008).

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/ *even* untuk menghasilkan *output* tertentu. *Sequence Diagram* diawali dari apa yang *metrigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan (Henderi, 2007).

Berikut adalah simbol-simbol yang ada pada diagram *Sequence*

Tabel 2. 3 Notasi *Sequence Diagram*

| Simbol | Nama Simbol | Keterangan |
|---|--------------------|---|
|  | <i>Object</i> | Merupakan <i>Instance</i> dari sebuah <i>Class</i> dan dituliskan tersusun secara <i>Horizontal</i> . |
|  | <i>Actor</i> | komponen yang dapat berkomunikasi dengan <i>Object</i> . |
|  | Waktu Aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. |
|  | <i>Final State</i> | Menunjukkan akhir dari aktivitas. |

| | | |
|--|------------------------------|--|
| | <i>Decision</i> | Adanya keputusan/kondisi tertentu. |
| | Pesan tipe <i>create</i> | Suatu objek membuat objek yang lain, arah panah meng-arah pada objek yang dibuat. |
| | Pesan tipe <i>call</i> | Suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. |
| | Pesan tipe <i>send</i> | Suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| | Pesan tipe <i>Return</i> | Suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |
| | Pesan tipe <i>Destroy</i> | Suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>Create</i> maka ada <i>Destroy</i> . |
| (Sumber: Rosa A.S dan M.Shalahuddin, 2014) | | |

e. *Collaboration Diagram*

Diagram ini bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (*Message*).

f. *Statechart Diagram*

Diagram ini bersifat dinamis. Diagram ini memperlihatkan *state-state* pada sistem memuat *state*, transisi, *event*, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari

antarmuka kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.

g. *Activity Diagram*

Activity diagram adalah diagram yang menggambarkan sifat dinamis secara alamiah sebuah sistem dalam bentuk model aliran dan kontrol dari aktivitas ke aktivitas lainnya (Henderi, 2007). Sebuah aktivitas merepresentasikan suatu operasi pada beberapa *class* dalam sistem yang menghasilkan suatu perubahan keadaan (*state*) dari sistem tersebut.

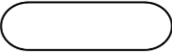




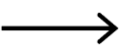
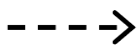
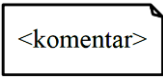
Secara khusus, *Activity diagram* biasa digunakan untuk memodelkan diagram alir sebuah sistem kerja (*workflow*) atau proses bisnis (prosedur bisnis) dan operasi-operasi secara internal (Miller Randy, 2008). Sementara menurut Whitten L. Jeffery et al (2004:428), *Activity diagram* adalah sebuah diagram yang dapat digunakan untuk menggambarkan secara grafis aliran proses bisnis, langkah-langkah sebuah *Use Case* atau logika *behavior* (metode) *object*. Karena sebuah *Activity diagram* adalah bentuk khusus dari statechart diagram, maka *Activity diagram* sering digunakan untuk memodelkan suatu kebiasaan sesuai dengan ketentuan/ kaedah bisnis.

Diagram ini bersifat dinamis diagram aktivitas atau *Activity Diagram* menggambarkan *Workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Hal yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang akan dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa and Shalahuddin 2014).

Untuk dapat membangun *Activity* diagram yang baik, berikut proses yang unggul untuk membangun *Activity* diagram:

1. Tambahkan poin awal dan akhir pada sebuah *Use Case*
2. Tambahkan sebuah kegiatan untuk tiap langkah utama pada *Use Case* (atau tiap langkah utama setiap pelaku yang menginisialisasi)
3. Tambahkan transisi dari setiap kegiatan ke kegiatan lain, poin keputusan, atau poin akhir
4. Tambahkan bar sinkronisasi di mana kegiatan dilakukan secara paralel.

Tabel 2. 4 Notasi *Activity Diagram*

| Simbol | Nama Simbol | Keterangan |
|---|----------------------|---|
|  | <i>Activity</i> | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | <i>Initial state</i> | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
|  | <i>Final state</i> | Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
|  | <i>Decision</i> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
|  | <i>Join</i> | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
|  | <i>Control flow</i> | Alur yang menunjukkan arah berjalannya kendali. |
|  | <i>Object flow</i> | Alur yang menunjukkan arah objek. |
|  | <i>Note</i> | Memuat komentar/catatan yang sifatnya di luar sistem. |
| (Sumber: Rosa A.S dan M.Shalahuddin,2014) | | |

h. *Component Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan organisasi serta kebergantungan pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

i. *Deployment Diagram*

Diagram ini bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run time*). Dengan ini memuat simpul-simpul (*node*) beserta komponen-komponen yang ada didalamnya. *Deployment diagram* berhubungan erat dengan diagram komponen di mana *Deployment Diagram* memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi berlaku sebagai Aplikasi yang dijalankan pada banyak mesin (*Distributed Computing*).

G. Web

Web adalah jaringan komputer yang terdiri dari kumpulan situs *internet* yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui *hypertext transfer protocol* (Rerung, R. R, 2018).

Web dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). *Web* adalah sekumpulan halaman yang terdiri dari beberapa halaman yang berisi informasi dalam bentuk data digital yang

disediakan melalui jalur koneksi internet. *Web* adalah sebuah software yang berfungsi untuk menampilkan dokumen-dokumen pada suatu *web* yang membuat pengguna dapat mengakses internet melalui software yang terkoneksi dengan internet (Destiningrum and Adrian 2017)

Menurut (Nugroho dalam Hendini, 2016), menjelaskan bahwa *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang berasal dari file-file berisi bahasa pemrograman yang saling berhubungan digunakan untuk menampilkan informasi, gambar bergerak dan tidak bergerak, suara dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis.

H. XAMPP

XAMPP yang merupakan singkatan dari Apache, MySQL, PHP dan Perl sedangkan huruf "X" dimaksudkan sebagai suatu software yang dapat dijalankan di empat OS utama seperti Windows, Mac OS, Linux dan Solaris. Istilah ini seringkali disebut dengan cross platform (software multi OS).

Sesuai dengan namanya software yang satu ini merupakan gabungan dari beberapa software dengan fungsi yang sama yakni menunjang para pembuat web yang menginginkan adanya web server sendiri di PC atau laptopnya. Software ini juga berlisensi GNU dan dapat didownload secara gratis di internet mengingat peran vital yang dimilikinya terutama bagi pembuat web pemula.

Software XAMPP didirikan oleh suatu perusahaan bernama Apache Friends. Dengan adanya beberapa tools pemrograman seperti MySQL, PHP dan Perl yang dimilikinya tentu mengindikasikan jika anda menekuni salah satu atau semuanya berarti harus memiliki software yang bernama XAMPP ini. Maksud dari Apache yakni selain mengindikasikan nama pengembangnya juga merupakan suatu software yang

menghadirkan web server pada komputer anda layaknya web server sesungguhnya.

1. Fungsi Xampp

Apache adalah suatu software yang juga dikembangkan Apache Friends dengan tujuan untuk membuat web server pribadi sehingga anda dapat membuat tampilan web yang dinamis. Istilah ini biasa disebut Localhost. Banyak developer web yang terlebih dahulu mencoba menjalankan webnya di Localhost sebelum akhirnya diposting di web server yang sesungguhnya.

Selain sebagai web server, XAMPP juga menunjang beberapa Bahasa pemrograman khusus dalam website yakni PHP, MySQL dan Perl. PHP merupakan suatu Bahasa yang sering digunakan oleh programmer khusus Back End karena memang lebih mengutamakan logika dibanding tampilan, beda halnya dengan HTML atau CSS. Oleh karena itu script PHP tidak akan terlihat dalam tampilan website anda.

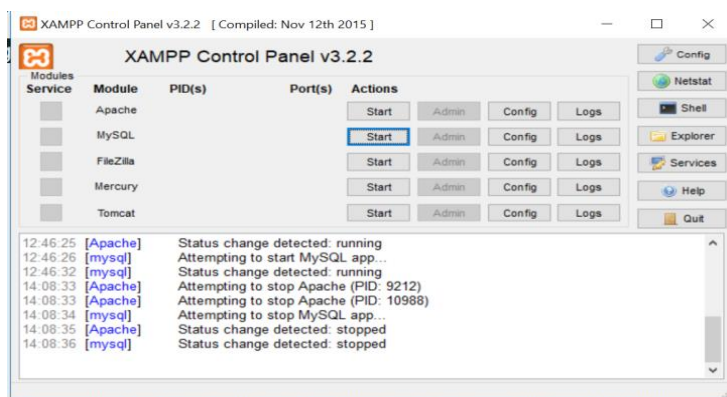
Selain sebagai web server, XAMPP juga menunjang beberapa Bahasa pemrograman khusus dalam website yakni PHP, MySQL dan Perl. PHP merupakan suatu Bahasa yang sering digunakan oleh programmer khusus Back End karena memang lebih mengutamakan logika dibanding tampilan, beda halnya dengan HTML atau CSS. Oleh karena itu script PHP tidak akan terlihat dalam tampilan website anda. Sampai saat ini sudah terdapat PHP versi 5.5.0.

MySQL merupakan suatu software yang digunakan untuk mengelola SQL (Structured Query Language). Bahasa ini biasa digunakan untuk keperluan database khusus pada website. Pengelolaan database yang dimaksudkan adalah untuk menambah data, mengubah,

menghapus dan lain-lain. Keberadaan MySQL juga biasanya identic dengan Bahasa PHP.

Selanjutnya adalah tools bernama Perl. Bahasa pemrograman yang satu ini tidak hanya digunakan untuk pengelolaan website saja namun juga dalam berbagai hal, juga merupakan salah satu Bahasa pemrograman versi jadul namun tetap bisa eksis sampai sekarang. Perl pertama kali dikenalkan pada tahun 1987 di mana saat itu masih menggunakan Unix.

2. Bagian-bagian penting xampp



Gambar 2.1 Bagian Xampp

a. Htdocs

Htdocs merupakan sebuah folder penyimpanan web server untuk halaman-halaman web yang sudah dibuat dan nantinya akan ditampilkan. Baik pada web server yang asli maupun XAMPP bentuk Htdocs-nya sama namun yang berbeda adalah di kapasitasnya. Karena XAMPP menggunakan penyimpanan internal komputer maka kapasitasnya menyesuaikan komputer anda. Sedangkan pada hosting berbayar kapasitas yang disediakan mengikuti ketentuan yang dibuat.

b. PhpMyAdmin

PhpMyAdmin merupakan suatu software khusus untuk mengelola administrasi MySQL. Jika pada Htdocs menyimpan file-file tampilan web anda maka di phpMyAdmin ini terdapat semua database yang anda gunakan untuk keperluan website.

c. Control panel

Sesuai dengan namanya, di Control Panel ini anda dapat mengontrol atau mengendalikan XAMPP dengan lebih efektif, mulai dari mengatur setting website, database, dan masih banyak lagi. Dalam dunia hosting lebih dikenal istilah CPanel.

I. Database

Database atau basis data adalah kumpulan berbagai data dan informasi yang tersimpan dan tersusun di dalam komputer secara sistematis yang dapat diperiksa, diolah atau dimanipulasi dengan menggunakan program komputer untuk mendapatkan informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil database disebut dengan sistem *database management system*. Istilah database sendiri mengacu pada koleksi data-data yang saling terkait satu sama lain dimana tujuan database adalah dapat digunakan untuk mengelola data dengan lebih efektif dan efisien.

1. Pengertian Database Menurut Para Ahli

Beberapa ahli di bidang teknologi informasi pernah menjelaskan tentang definisi database, di antaranya adalah:

a. Jogiyanto

Menurut Jogiyanto pengertian database adalah kumpulan informasi dan data yang saling berhubungan satu sama lain, dimana data tersebut tersimpan di

simpanan luar komputer dan diperlukan software tertentu untuk memanipulasinya.

b. Abdul Kadir

Menurut Abdul Kadir pengertian database atau basis data adalah suatu pengorganisasian sekumpulan data yang saling terhubung sehingga memudahkan aktivitas untuk memperoleh informasi.

c. S. Atte

Menurut S. Atte pengertian database atau basis data adalah sebuah koleksi data-data yang saling berhubungan yang ada dalam suatu organisasi atau enetrprise dengan berbagai penggunaan.

d. Fabbri dan schwab

Menurut Fabbri dan Schwab definisi database atau basis adalah suatu sistem berkas terpadu yang dirancang khusus untuk dapat meminimalkan pengulangan atau redundancy data.

e. Toni Fabbri

Menurut Toni Fabbri pengertian database atau basis data adalah suatu sistem file-file dan data yang terintegrasi dimana file dan data tersebut yang mempunyai sebuah primary key untuk melakukan pengulangan data.

f. Gordon C. Everest

Menurut Gordon C. Everest, definisi database adalah suatu kumpulan data yang bersifat mekanis, terbagi, terdefinisi secara formal, dan terkontrol. Pengontrolan tersebut terpusat pada suatu organisasi.

g. C.J. Date

Menurut C.J. Date pengertian basis data adalah koleksi data/ informasi operasional yang sengaja disimpan dan juga digunakan oleh sistem aplikasi sebuah organisasi.

2. Fungsi Database

Setelah memahami pengertian database, tentunya kita juga harus mengetahui apa fungsi dari database. Berikut ini adalah beberapa fungsi database:

- a. Mengelompokkan data dan informasi sehingga lebih mudah dimengerti.
- b. Mencegah terjadinya duplikat data maupun inkonsistensi data.
- c. Mempermudah proses penyimpanan, akses, pembaharuan, dan menghapus data.
- d. Menjaga kualitas data dan informasi yang diakses sesuai dengan yang di-input.
- e. Membantu proses penyimpanan data yang besar.
- f. Membantu meningkatkan kinerja aplikasi yang membutuhkan penyimpanan data.

3. Manfaat Database

Sebelum mengetahui apa saja jenis perangkat lunak yang bisa digunakan untuk menyusun database, berikut ini beberapa manfaat yang bisa didapatkan jika bekerja dengan sistem database:

- a. Tidak terjadi redundansi basis data

Seperti yang sudah disinggung pada pengertian database sebelumnya, database bisa membantu meminimalkan redundansi data. Redundansi sendiri merupakan terjadinya data-data ganda dalam berkas-berkas yang berbeda.

- b. Integritas data terjaga

Database memastikan integritas data yang tinggi dimana database akan memastikan keakuratan, aksesibilitas, konsistensi dan juga kualitas tinggi pada suatu data.

c. Independensi data terjaga

Database menjaga independensi data dimana orang lain tidak dapat merubah data meskipun data bisa diakses.

d. Kemudahan berbagi data

Menggunakan perangkat lunak database bisa digunakan untuk berbagi data atau informasi dengan sesama pengguna lainnya.

e. Menjaga keamanan data

Database menjamin keamanan suatu informasi dan data, dimana Anda bisa menyisipkan kode akses untuk data-data tertentu yang tidak bisa diakses bersama.

f. Kemudahan akses data

Dengan database bisa memudahkan untuk mengakses dan mendapatkan data karena semua data terorganisir dengan baik.

4. Tipe-tipe database

Dibutuhkan software khusus untuk menyimpan dan mengambil data dan informasi dari database. Software ini sering disebut dengan System Managemen Basis Data (DBMS). Berikut ini adalah tipe database:

a. Analytical database; yaitu database untuk menyimpan informasi dan data yang diambil dari operasional dan eksternal database.

b. Operational database; yaitu database yang menyimpan data mendetail yang dibutuhkan untuk mendukung operasi suatu organisasi secara keseluruhan.

c. Distributed database; yaitu kelompok kerja lokal database dan departemen di berbagai kantor dan lokasi kerja yang lainnya.

- d. Data warehouse; yaitu sebuah gudang data yang menyimpan berbagai data dari tahun-tahun sebelumnya hingga saat ini.
- e. End-user database; yaitu basis data pengguna akhir yang terdiri dari berbagai file data yang dikembangkan dari end-user dalam workstation mereka.
- f. Real time database; yaitu sistem pengolahan yang dirancang dalam menangani beban kerja suatu negara yang bisa berubah-ubah, mengandung data terus menerus dan sebagian tidak berpengaruh terhadap waktu.
- g. Document oriented database; yaitu salah satu perangkat lunak komputer yang dibuat untuk sebuah aplikasi dan berorientasi pada dokumen.
- h. In memory database; yaitu database yang tergantung pada memori untuk menyimpan informasi/ data pada komputer.
- i. Navigational database; pada navigasi database, queries menemukan benda bagi yang mengikuti referensi dari objek tertentu.
- j. Hypermedia database on the web; sekumpulan halaman multimedia yang saling berhubungan dalam sebuah website, yang terdiri dari homepage dan hyperlink dari multimedia (gambar, teks, grafik audio, dan lain-lain).
- k. External database; database yang menyediakan akses ke luar, dan data pribadi online.
- l. Relational database; merupakan standar komputasi bisnis, dan basis data yang paling umum dipakai saat ini.

5. Jenis-jenis software database

Setelah memahami pengertian database dan keuntungan yang bisa didapatkan jika menggunakan

perangkat lunak database, maka berikut ini jenis-jenis software database terbaik yang bisa Anda gunakan:

a. Microsoft access

Salah satu software database ini adalah yang paling sering digunakan. Microsoft access sangat cocok digunakan untuk sebagian besar komputer yang relasional. Selain itu, jika Anda berbisnis dalam skala rumahan, bisa memilih sistem database ini karena sangat ringan digunakan dan format datanya sangat umum sehingga memudahkan pembacaan.

b. Oracle

Salah satu software database ini sangat mampu untuk menyimpan data dengan ukuran yang maksimum hingga tera byte. Oracle paling banyak digunakan pada perusahaan-perusahaan terutama yang sedang berkembang karena memang untuk mengaksesnya tersedia secara gratis.

c. Ms SQL Server

Software database ini merupakan manajemen basis data yang umum digunakan pada Microsoft dengan bahasa pemograman yang digunakan adalah Transact-SQL. Tipe data yang digunakan cukup banyak sehingga sangat efektif untuk mendukung kinerja Anda.

d. MySQL

Salah satu software database yang open access untuk umum dan kompatibel pada sistem operasi Windows maupun Linux. Keunggulan yang bisa Anda gunakan dengan menggunakan program MySQL adalah bisa digunakan untuk multi user. Kelebihan lainnya dari MySQL yaitu tersedia gratis, query data yang cepat dan berlisensi resmi.

e. Firebird

Bisa dibilang software database ini memiliki fitur sistem yang standar dan ringan yaitu fitur ANS

SQL-99 dan SQL-2003. Kompatibel untuk digunakan pada sistem operasi Windows, Linux maupun Unix.

f. Postgre SQL

Menawarkan sistem database opensource dengan lisensi GPL/ General Public License. Software ini menggunakan bahasa pemrograman C++, C, SQL, PHP dan lainnya. Jika digunakan untuk pekerjaan pribadi, maka software ini sangat recommended digunakan.

J. PHP (*Page Hypertext Preprocessor*)

PHP atau Personal Home Page merupakan salah satu sumber bahasa pemrograman di server yang digunakan untuk mengatasi masalah dan pengembangan suatu web dan bisa digunakan bersamaan dengan HTML (HyperText Markup Language).

PHP singkatan dari PHP yaitu Hypertext Preprocessor. PHP ini merupakan suatu bahasa pemrograman server-side yang dirancang untuk pengembangan web. PHP dikatakan server-side lantaran program yang diberikan kan dijalankan atau diproses pada komputer yang bertindak sebagai server. Contohnya saat pengguna mengakses suatu situs maka web browser akan melakukan request ke server. Pengertian PHP (Hypertext Preprocessor) menurut para ahli:

1. Arief

PHP ialah suatu bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan server-side-scripting maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya akan dikirimkan ke browser dengan format HTML.

2. Nugroho

PHP atau singkatan dari Personal Home Page merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side.

3. Betha Sidik

PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman script-script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman server side.

4. Sibero

PHP yaitu pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan.

1. Sejarah PHP (Hypertext Preprocessor)

Pada awalnya PHP muncul pada tahun 1995, PHP tersebut dibuat oleh Rasmus Lerdor. Saat itu, PHP masih bernama Form Interpreted (FI) yang berwujud dalam bentuk sekumpulan skrip yang digunakan untuk pengolahan data formulir dari web.

Kemudian Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan rilis kode sumber tersebut menjadi sumber terbuka (open source), maka banyak pemrograman yang tertarik untuk mengikuti perkembangan PHP.

Pada November 1997, dikeluarkan PHP/FI 2.0. Saat rilis, interpreter PHP telah diimplementasikan dalam program C. Pada rilis ini dilampirkan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, perusahaan bernama Zend melakukan perlisian ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat.

Selanjutnya, pada Juni 1998, perusahaan Zend tadi merilis interpreter baru untuk PHP dan meresmikan rilis tersebut menjadi PHP 4.0. PHP 4.0 merupakan versi PHP yang penggunaannya paling banyak di awal abad ke-21. Versi ini sering digunakan karena kemampuannya untuk membangun aplikasi web kompleks namun tetap mempunyai kecepatan dan stabilitas tinggi.

Pada Juni 2014, Zend merilis PHP 5.0. Pada versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga menyertakan model pemrograman dengan orientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman pada orientasi objek.

Server objek bawaan ditambahkan pada versi 5.4 untuk memudahkan pengembang dalam menjalankan kode PHP tanpa melakukan instalasi software server. Versi terbaru dan stabil dari PHP sekarang ini yaitu versi 7.0.16 dan 7.1.2 yang sudah resmi dirilis pada 17 Februari 2017.

2. Fungsi PHP (Hypertext Preprocessor)

a. Memprsingkan tatanan HTML dan CSS

Untuk membuat suatu halaman web yang dinamis, PHP bisa dipakai untuk mempersingkat penggunaan tatanan HTML dan CSS. Contohnya dalam suatu sistem karyawan mempunyai jumlah baris 100. Bila memakai HTML dan CSS tentu barisan tersebut akan menjadi sangat panjang. Sedangkan bila ditambah dengan memakai PHP, maka pemrogram bisa mengatur beberapa baris yang diperlukan atau ditampilkan.

b. Input data

Dengan memakai bahasa pemrograman PHP, pemrogram dapat memasukkan data dan menyimpannya dalam sistem database seperti MySQL.

c. Manajemen cookie dan session

Dalam PHP, Cookie dan Session dipakai untuk menyimpan informasi pengguna. Fungsi `session_start()` dipakai untuk memulai session dan cookie dengan fungsi `setcookie()`. Contohnya proses cookie, menyimpan username dan password pengguna di browser sehingga tidak harus mengisinya kembali saat membuka situs yang sama. Session contohnya menyimpan informasi login yang hanya berlaku dalam satu sesi saja.

d. Compress teks

Dalam PHP pemrogram bisa mengompres teks panjang sehingga menjadi lebih pendek dengan fungsi `gzcompress()`. Sedangkan untuk mengembalikannya memakai fungsi `gzuncompress()`.

3. Jenis Data PHP (Hypertext Preprocessor)

a. Integer

Merupakan tipe data PHP berupa bilangan bulat, tipe data ini sering sekali digunakan pada pemrograman PHP khususnya yang berkaitannya dengan bilangan bulat.

b. Float

Dapat dikatakan juga bilangan berkoma (,) tapi dalam pemrograman tidak menggunakan koma melainkan titik (.).

c. String

Merupakan tipe data berupa karakter, di dalamnya bisa berupa teks atau kata.

d. Boolean

Merupakan tipe data yang bisa bernilai, True atau False, tipe data boolean biasanya digunakan untuk logika, seperti pada kondisi if atau looping "pengulangan".

e. Array

Merupakan kumpulan dari tipe data, didalamnya memiliki banyak nilai, jadi dapat dikatakan array merupakan tipe data yang terdiri dari kumpulan tipe data.

f. Objek

Tipe daya objek dapat berupa bilangan, fungsi, maupun variable. Tipe data ini digunakan supaya programmer terbiasa dengan Objek Oriented Programming “OOP”, sehingga data yang dimasukkan pada script program dapat diringkas.

4. Kelebihan dan kekurangan PHP (Hypertext Preprocessor)

a. Kelebihan PHP (Hypertext Preprocessor)

1. Banyaknya web server yang mendukung bahasa pemrograman PHP sehingga konfigurasinya semakin mudah.
2. Beberapa web server misalnya Apache, IIS, Xitami, Nginx, dan Lain-Lain.
3. Pengembangan bahasa pemrograman PHP tergolong lebih mudah, karena banyak developer yang membantu mengembangkannya maupun menggunakannya.
4. Relatif mudah untuk dipahami, karena sekarang ini banyak sekali tersebar materi-materi/referensi untuk mempelajari PHP.
5. Bahasa pemrograman PHP juga dapat di sisipkan kedalam HTML.
6. Cocok digunakan untuk pemrograman web dinamis, walau bisa juga untuk membuat program komputer lainnya.
7. PHP merupakan bahasa pemrograman bersifat Open Source, sehingga dapat di gunakan di banyak sistem operasi komputer dan tentunya gratis.

- b. Kekurangan PHP (Hypertext Preprocessor)
 - a. Keamanan yang kurang baik, jika programmer tidak memperhatikan keamanan dari program yang dibuatnya.
 - b. Saat ini untuk encoding kode PHP tergolong sangat mahal, karena membutuhkan tool tertentu.
 - c. Kurang cocok untuk pengembangan program komputer berskala besar.

K. SQL (Structured Query Language)

SQL adalah bahasa komputer standar yang digunakan untuk komunikasi dengan sistem *database relasional*. Yang mendefinisikan *SQL* ialah *ANSI (American National Standard Institute)* dan *ISO (International Standard Organization)* (Toledo and Pauline 2007). Sebagai bahasa pemrograman *SQL* dilengkapi dengan perintah-perintah *Query*. *Query* adalah sebuah mekanisme utama untuk mengambil informasi dari *Database* dalam bentuk tertentu. *SQL* merupakan perintah atau bahasa yang melekat di dalam satu *Database*. Sebagai bahasa perintah *SQL* didukung oleh beberapa *DBMS (Database Management System)* seperti: *Mysql, Msq, PostgreSQL, Interbase dan Oracle*.

Komponen-komponen yang ada pada *SQL* ialah sebagai berikut:

1. *Data Definition Language (DDL)*

Merupakan bahasa *SQL* untuk membangun kerangka *Database*. Terdapat tiga perintah yang digunakan:

- a. *Create*: digunakan untuk membuat *Database* baru, tabel baru, *view* baru dan kolom.
- b. *Alter*: digunakan untuk mengubah struktur tabel yang telah dibuat seperti mengganti nama tabel, menambah kolom, menghapus kolom, maupun memberikan atribut pada kolom.

c. *Drop*: digunakan untuk menghapus *Database* dan tabel.

2. *Data Manipulation Language (DML)*

Merupakan sub bahasa *SQL* untuk memanipulasi data dalam *database* yang telah dibuat. Perintah yang digunakan yaitu:

- a. *Insert*: untuk menyisipkan atau memasukkan data baru ke dalam tabel yang sudah dibuat.
- b. *Select*: untuk mengambil data atau menampilkan data dari suatu tabel atau beberapa tabel dalam relasi.
- c. *Update*: untuk memperbaharui data lama menjadi data terkini.
- d. *Delete*: untuk menghapus data dari tabel.

3. *Data Control Language (DCL)*

Sub bahasa *SQL* yang digunakan untuk melakukan pengontrolan data dan *server database*. Perintah *DCL* adalah sebagai berikut:

- a. *Grant*: untuk memberikan hak/izin akses oleh administrator (pemilik utama) *server* kepada *user* (pengguna biasa).
- b. *Revoke*: untuk menghilangkan atau mencabut hak akses yang telah diberikan kepada *user* oleh administrator.

L. *Mysql*

MySQL adalah sebuah database management system (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. Database management system (DBMS) MySQL multi pengguna dan multi alur ini sudah dipakai lebih dari 6 juta pengguna di seluruh dunia.

MySQL adalah DBMS yang open source dengan dua bentuk lisensi, yaitu Free Software (perangkat lunak bebas) dan Shareware (perangkat lunak berpemilik yang penggunaannya

terbatas). Jadi MySQL adalah database server yang gratis dengan lisensi GNU General Public License (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Seperti yang sudah disinggung di atas, MySQL masuk ke dalam jenis RDBMS (Relational Database Management System). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah database terdapat satu atau beberapa tabel.

SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada relational database atau database yang terstruktur. Jadi MySQL adalah database management system yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan database server.

MySQL adalah pengembangan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Monty Widenius dan TcX (perusahaan perangkat lunak asal Swedia).

Sayangnya, UNIREG belum terlalu kompatibel dengan database dinamis yang dipakai di website. TcX kemudian mencari alternatif lain dan menemukan perangkat lunak yang dikembangkan oleh David Hughes, yaitu miniSQL atau mSQL. Namun, ditemukan masalah lagi karena mSQL tidak mendukung indexing sehingga belum sesuai dengan kebutuhan TcX.

Pada akhirnya muncul kerjasama antara pengembang UNIREG (Michael Monty Widenius), mSQL (David Hughes), dan TcX. Kerjasama ini bertujuan untuk mengembangkan sistem database yang baru, dan pada 1995 dirilislah MySQL seperti yang dikenal saat ini. Saat ini pengembangan MySQL berada di bawah Oracle.

1. Kelebihan dan Kekurangan MySQL

Setelah mempelajari pengertian MySQL dan sejarahnya, ada baiknya jika Anda juga mengetahui kelebihan dan kekurangannya juga.

Meskipun menjadi database yang cukup populer, MySQL tentu mempunyai beberapa kelebihan dan kekurangan dibandingkan dengan database server lainnya. Salah satu kekurangan MySQL adalah performanya turun di saat beberapa database manajemen sistem mampu bekerja baik pada pengelolaan database yang besar.

a. Kelebihan MySQL

MySQL mempunyai beberapa kelebihan yang bisa Anda manfaatkan untuk mengembangkan perangkat lunak yang andal seperti:

1. Mendukung Integrasi Dengan Bahasa Pemrograman Lain.

Website atau perangkat lunak terkadang dikembangkan dengan menggunakan berbagai macam bahasa pemrograman, jadi Anda tidak perlu khawatir jika menggunakan MySQL. Maka dari itu, MySQL bisa membantu Anda untuk mengembangkan perangkat lunak yang lebih efektif dan tentu saja lebih mudah dengan integrasi antara bahasa pemrograman.

2. Tidak Membutuhkan RAM Besar.

MySQL dapat dipasang pada server dengan spesifikasi kecil. Jadi tidak perlu khawatir jika Anda hanya mempunyai server dengan kapasitas 1 GB karena Anda masih bisa menggunakan MySQL sebagai database Anda.

3. Mendukung Multi User.

MySQL dapat dipakai oleh beberapa user dalam waktu bersamaan tanpa membuatnya crash

atau berhenti bekerja. Ini dapat Anda manfaatkan ketika mengerjakan proyek yang sifatnya tim sehingga seluruh tim dapat bekerja dalam waktu bersamaan tanpa harus menunggu user lain selesai.

4. Bersifat Open Source

MySQL adalah sistem manajemen database gratis. Meskipun gratis, bukan berarti database ini mempunyai kinerja buruk. Apalagi lisensi gratis yang dipakai adalah GPL di bawah pengelolaan Oracle sehingga kualitasnya termasuk baik. Selain itu, Anda juga tidak perlu khawatir jika terjadi masalah karena banyak komunitas dan dokumentasi yang membahas soal MySQL.

5. Struktur Tabel yang Fleksibel.

MySQL mempunyai struktur tabel yang mudah dipakai dan fleksibel. Contohnya saat MySQL memproses ALTER TABLE dan lain sebagainya. Jika dibandingkan dengan database lain seperti Oracle dan PostgreSQL, MySQL tergolong lebih mudah.

6. Tipe Data yang Bervariasi.

Kelebihan lain dari MySQL adalah mendukung berbagai macam data yang bisa Anda gunakan di MySQL. Contohnya float, integer, date, char, text, timestamp, double, dan lain sebagainya. Jadi manajemen database sistem ini sangat membantu Anda untuk mengembangkan perangkat lunak yang berguna untuk pengelolaan database di server.

7. Keamanan yang Terjamin.

Open source bukan berarti MySQL menyediakan keamanan yang buruk. Malah sebaliknya, MySQL mempunyai fitur keamanan yang cukup apik. Ada beberapa lapisan keamanan yang

diterapkan oleh MySQL, seperti level nama host, dan subnetmask. Selain itu MySQL juga dapat mengatur hak akses user dengan enkripsi password tingkat tinggi.

b. Kekurangan MySQL

Sayangnya, meskipun memiliki segudang kelebihan, masih ada beberapa kelemahan yang dimiliki oleh MySQL sehingga Anda perlu mempertimbangkannya juga sebelum memakainya.

1. Kurang Cocok untuk Aplikasi Game dan Mobile

Anda yang ingin mengembangkan aplikasi game atau perangkat mobile ada baiknya jika mempertimbangkan lagi jika ingin menggunakan MySQL. Kebanyakan pengembang game maupun aplikasi mobile tidak menggunakannya karena memang database manajemen sistem ini masih kurang bagus dipakai untuk sistem aplikasi tersebut.

2. Sulit Mengelola Database yang Besar

Jika Anda ingin mengembangkan aplikasi atau sistem di perusahaan dengan database yang cukup besar, ada baiknya jika menggunakan database manajemen sistem selain MySQL. MySQL dikembangkan supaya ramah dengan perangkat yang mempunyai spesifikasi rendah, itulah mengapa MySQL tidak memiliki fitur yang lengkap seperti aplikasi lainnya

3. Technical Support yang Kurang Bagus

Sifatnya yang open source terkadang membuat aplikasi tidak menyediakan technical support yang memadai. Technical support MySQL diklaim kurang bagus. Hal ini membuat pengguna kesulitan. Apalagi jika pengguna mengalami masalah yang berhubungan dengan pengoperasian perangkat

lunak tersebut dan membutuhkan bantuan technical support.

Anda sudah belajar pengertian, sejarah, kekurangan, dan kelebihan mysql. Anda bisa melanjutkan membaca untuk belajar MySQL dari dasar.

2. Belajar MySQL Lengkap

Mempelajari MySQL cukup mudah. Sesuai dengan pengertian MySQL yang sudah dibahas sebelumnya, bahwa sistem manajemen server ini menggunakan SQL sebagai bahasa pemrogramannya. Jadi akan lebih baik jika Anda harus memahami bagaimana SQL bekerja dan cara menggunakannya.

a. Mengenal SQL

SQL merupakan bahasa pemrograman yang perlu Anda pahami karena dapat merelasikan antara beberapa tabel dengan database maupun antar database. Ada tiga bentuk SQL yang perlu Anda ketahui, yaitu *Data Definition Language (DDL)*, *Data Manipulation Language (DML)*, dan *Data Control Language (DCL)*.

| | |
|--------|---|
| CREATE | Dipakai untuk membuat tabel dan database. |
| DROP | Dipakai untuk menghapus database dan tabel. |
| ALTER | Dipakai untuk mengubah struktur tabel yang sudah ada. Alter dapat mengganti field menggunakan perintah "Change", menambahkan field menggunakan perintah "Add", atau menghapus field menggunakan perintah "drop", dan mengubah namanya menggunakan perintah "Rename". |

1. Data Definition Language (DDL)

DDL berguna pada saat Anda ingin mendefinisikan data di dalam database. Terdapat

beberapa query yang dikelompokkan ke dalam DDL, yaitu:

Argumen DDL di atas perlu Anda pahami karena merupakan dasar penggunaan SQL di bagian awal pembuatan database. Contohnya saja jika belum menjalankan perintah "CREATE", Anda belum bisa melanjutkan penggunaan argumen yang lainnya.

2. Data Manipulation Language (DML)

DML dapat Anda pakai setelah menjalankan perintah DDL. DML berfungsi untuk memanipulasi, mengubah, atau mengganti isi dari database (tabel) yang sudah ada.

Terdapat beberapa perintah DML yang perlu Anda ketahui, yaitu:

| | |
|--------|--|
| INSERT | Dipakai untuk memasukkan data ke dalam tabel pada database. |
| UPDATE | Dipakai untuk mengubah data yang ada di dalam tabel pada database. |
| DELETE | Dipakai untuk menghapus data di dalam tabel pada database. |

3. Data Control Language (DCL)

Jika Anda sudah mempunyai user dan ingin mengatur hak akses masing-masing user, Anda sebaiknya memahami berbagai macam jenis DCL dan cara penggunaannya. DCL berguna untuk memberikan hak akses database, mendefinisikan space, mengalokasikan space, dan melakukan audit penggunaan database.

Terdapat beberapa perintah DCL yang perlu Anda ketahui, yaitu:

| | |
|----------|---|
| GRANT | Dipakai untuk memberikan izin kepada user untuk mengakses database. |
| REVOKE | Dipakai untuk membatalkan izin user untuk mengakses database. |
| COMMIT | Dipakai untuk menetapkan penyimpanan pada database. |
| ROLLBACK | Dipakai untuk membatalkan penyimpanan pada database. |

2. Cara Menginstall MySQL di Windows dan Linux

MySQL server mendukung berbagai macam sistem operasi (*cross-platform*), khususnya Linux dan Windows. Proses instalasinya pun sangat mudah. Jika menggunakan Windows, Anda bisa menginstall XAMPP untuk menjalankan MySQL server yang di dalamnya sudah terdapat juga modul untuk menjalankan Apache, PHP, FileZilla, dan Tomcat.

Sedangkan di Linux, Anda dapat menginstall MySQL secara terpisah atau menginstall LAMP (Linux, Apache, MySQL, PHP) yang sudah ada modul Apache dan PHP juga. Selain itu, XAMPP juga sudah tersedia di Linux. Jadi Anda bisa menyesuaikan proses instalasi sesuai dengan yang Anda nyaman dan suka.

1. Cara Install MySQL Server di Windows

Proses instalasi MySQL server menggunakan XAMPP sangat mudah.

a. Langkah 1. Unduh file instalasi XAMPP

Anda cukup mengunduh file instalasi XAMPP melalui halaman website apachefriends.org.

b. Langkah 2. Install XAMPP

Setelah selesai proses unduh, silakan eksekusi file instalasi XAMPP untuk menjalankan

proses instalasi. Kemudian ikuti instruksi yang ada.

c. Langkah 3. Aktifkan MySQL server menggunakan GUI XAMPP

Saat Anda selesai menjalankan proses instalasi akan muncul panel kontrol XAMPP. Klik “Start” pada modul Apache dan MySQL. Sampai di sini MySQL server sudah dapat berjalan dan sudah dapat dipakai untuk menjalankan perintah SQL.

Anda bisa mengikuti artikel [Cara Menggunakan MySQL di Windows](#) untuk panduan selengkapnya.

2. Cara Install MySQL Server di Linux

Pada panduan cara install MySQL Server di Linux ini akan menggunakan Terminal sebagai contohnya. Anda bisa menggunakan cara ini atau jika ingin lebih mudah bisa langsung mengunduh file instalasi XAMPP dan menginstallnya seperti contoh Windows di atas.

a. Langkah 1. Install Server Web Apache

Sebelum Anda menjalankan proses instalasi, ada baiknya untuk memperbarui repositorinya terlebih dahulu. Untuk memperbarui repositori pada Linux caranya cukup mudah, Anda tinggal membuka Terminal kemudian mengetikkan baris komentar berikut:

```
$ sudo apt-get update
```

Kemudian jalankan perintah berikut untuk menginstall Apache.

```
$ sudo apt-get install apache2
```

b. Langkah 2. Mengecek Proses Instalasi Apache

Jika proses instalasi Apache sudah selesai dan berhasil dijalankan, Anda bisa mengeceknya dengan mengakses web browser kemudian menuliskan alamat IP server atau alamat domain.

```
http://<alamat_ip_address_server>.
```

c. Langkah 3. Instalasi MySQL

Proses instalasi MySQL server di Linux cukup menggunakan satu perintah saja.

```
$ sudo apt-get install mysql-server
```

Proses instalasi ini akan meminta Anda untuk memasukkan username dan password untuk akses ke dalam MySQL server.

d. Langkah 4. Pengecekan Instalasi MySQL

Anda dapat memastikan apakah proses instalasi MySQL sudah berjalan baik atau belum dengan mengetikkan perintah di bawah ini.

```
$ sudo systemctl status mysql
```

Perintah di atas untuk mengecek apakah MySQL sudah berjalan atau belum. Jika belum, Terminal akan menampilkan status 'Error' atau 'Failed'. Anda dapat mengikuti artikel Cara Instal LAMP di Linux untuk melihat panduan lengkapnya.

3. Fungsi-fungsi di MySQL Server

Jika MySQL sudah dapat berjalan dengan baik di server atau perangkat, ada beberapa fungsi yang bisa Anda jalankan menggunakan teks perintah (*command prompt*). Untuk masuk ke dalam MySQL server buka 'CMD' di Windows dan 'Terminal' di Linux.

a. Login dan Logout MySQL Server

Pengertian MySQL adalah database manajemen server yang cukup aman sehingga mempunyai aturan

hak akses yang ketat. Jadi ketika ingin mengakses MySQL, Anda harus menggunakan password yang sudah diatur sebelumnya.

Jika sebelumnya belum mengatur user untuk masuk ke dalam MySQL, Anda dapat menggunakan user root dengan mengetikkan perintah berikut.

```
$ mysql -u root -p
```

Opsi '-u' merupakan tanda jika ingin login menggunakan user dengan menggunakan password yang dinyatakan dengan opsi '-p'. Jika ingin keluar, Anda hanya perlu mengetikkan perintah "quit" atau "\q".

b. Opsi di MySQL Server

MySQL server menyediakan beberapa bantuan yang bisa Anda akses untuk mengubah dan mendokumentasikan server, yaitu dengan mengetikkan perintah "\h" atau "\?" ke dalam koneksi MySQL yang sedang aktif.

Perhatian! Semua koneksi harus diakhiri tanda titik koma (;). Tanda ini sebagai petunjuk bahwa perintah atau query yang dimasukkan sudah selesai dan siap dieksekusi.

| | | |
|---------|------------|---|
| Bantuan | \h atau \? | Dipakai untuk menampilkan opsi bantuan yang tersedia di dalam MySQL server. |
| Clear | \c | Dipakai untuk menghapus atau membatalkan semua perintah yang berjalan pada satu perintah. |
| Connect | \r | Dipakai untuk merefresh koneksi ke database yang ada di dalam Server Host. |
| Ego | \G | Dipakai untuk menampilkan data di dalam database server secara horizontal |
| Go | \g | Dipakai untuk mengeksekusi perintah yang sudah dimasukkan. |
| Tee | \T | Dipakai untuk mengatur lokasi file untuk perintah yang ingin didokumentasikan. |

| | | |
|--------|----|---|
| Note | \t | Dipakai untuk mengakhiri perintah \T yang berguna untuk mendokumentasikan semua perintah yang sudah dijalankan. |
| Print | \p | Dipakai untuk menampilkan seluruh perintah yang sudah dijalankan ke layar. |
| Prompt | \R | Dipakai untuk mengubah 'prompt' sesuai dengan keinginan. |
| Source | \. | Dipakai untuk mengeksekusi perintah dari luar yang berbentuk file .sql |
| Use | \u | Dipakai untuk masuk ke dalam database (mirip dengan perintah "cd" di dalam bash script). |

c. Perintah Administrasi di MySQL Server

Pengertian MySQL server lainnya yaitu sebuah database manajemen sistem yang bersifat client/server. Jadi sangat penting untuk mengatur hak akses setiap user. Pengaturan hak akses ini supaya semua user mempunyai wewenang yang berbeda. Hak akses juga dapat mengatur user supaya hanya dapat mengakses database tertentu saja.

Pengaturan hak akses di MySQL tentu akan sangat berguna apalagi karena perangkat lunak ini berjalan di jaringan publik yang kemungkinan semua user bisa mengaksesnya melalui internet. Sebelum mengatur hak akses, Anda harus membuat terlebih dahulu user MySQL selain 'root'.

1. Membuat User Baru

Anda dapat menggunakan perintah SQL bernama "INSERT" untuk membuat user baru ke dalam database di MySQL server. Adapun sintaks yang dapat Anda gunakan contohnya:

```
INSERT INTO user(host, user, password)
VALUES('%','nama_user','password');
```

Perintah di atas akan menambahkan user ke dalam tabel 'user'.

Selain perintah di atas, Anda juga bisa memanfaatkan perintah “GRANT” untuk membuat user dan database. Langkah-langkahnya sebagai berikut.

1. Langkah 1. Masuk ke Dalam MySQL Server Menggunakan User ‘Root’.

Masuk ke dalam database MySQL dengan menggunakan perintah berikut ini.

```
$ mysql -u root -p
```

Kemudian masukkan password untuk user ‘root’ untuk masuk ke dalam MySQL server.

2. Langkah 2. Membuat User Untuk Database.

Anda dapat menggunakan contoh perintah berikut ini untuk membuat user untuk database.

```
GRANT ALL PRIVILEGES ON *.* TO  
'username'@'localhost' IDENTIFIED BY 'password';
```

Perintah di atas akan membuat user dengan hak akses secara menyeluruh seperti ‘root’. Namun Anda juga dapat menyesuaikan batasan hak akses dari masing-masing user yang dibuat dengan menggunakan perintah “SELECT” seperti perintah berikut.

```
GRANT SELECT ON *.* TO 'username'@'localhost';
```

Contoh dari penggunaan GRANT dan SELECT seperti beberapa perintah di bawah ini.

```
CREATE USER 'yasin'@'localhost' IDENTIFIED BY  
    'password';  
GRANT ALL ON db1.* TO 'yasin'@'localhost';  
GRANT SELECT ON db2.karyawan TO  
    'yasin'@'localhost';  
GRANT USAGE ON *.* TO 'yasin'@'localhost' WITH  
    MAX_QUERIES_PER_HOUR 90;
```

Setelah selesai menambahkan user, akhiri *session* menggunakan perintah “quit” atau “\q”.

3. Langkah 3. Membuat Database Menggunakan User Baru.

Masuk ke kembali ke dalam database server menggunakan user yang sudah dibuat.

```
$ mysql -u username -p
```

Kemudian buat database menggunakan perintah "CREATE".

```
CREATE DATABASE nama_database;
```

4. Langkah 4. Memasukkan Tabel ke Dalam Database.

Jika database sudah dibuat, Anda harus masuk ke dalam database tersebut untuk membuat tabel menggunakan perintah "USE".

```
USE karyawan;
```

Setelah masuk, gunakan perintah "CREATE TABLE" untuk membuat tabel sesuai dengan kebutuhan data yang ingin dimasukkan, contohnya di bawah ini.

```
CREATE TABLE contoh( id smallint unsigned not null auto_increment, name varchar(20) not null, constraint pk_example primary key (id) );
```

```
INSERT INTO contoh( id, name ) VALUES ( null, 'Sample data' );
```

2. Memberikan Hak Akses ke User

Jika sudah membuat user tetapi lupa untuk membuat hak akses, Anda masih dapat mengubahnya dan memberikan akses menggunakan perintah "GRANT". Contoh penggunaan sintaksnya seperti berikut.

a. Langkah 1. Melihat Seluruh User yang Sudah Ada.

Anda dapat melihat seluruh user yang ada di dalam MySQL server menggunakan perintah "SELECT" seperti baris perintah berikut ini.

```
SELECT user, host from mysql.user;
```


b. Langkah 2. Melihat Hak Akses User.

Jika sudah menemukan user yang dimaksud, Anda bisa mengecek hak akses yang saat ini berlaku untuk user tersebut menggunakan perintah “GRANT”.

```
show grants for 'user'@'host';
```

c. Langkah 3. Membatalkan Hak Akses User.

Sebelum mengatur ulang hak akses baru untuk user yang sudah ada, terlebih dahulu Anda harus membatalkan hak akses yang berlaku saat ini menggunakan perintah “REVOKE”.

```
revoke all privileges on *.* from 'user'@'host';
```

d. Langkah 4. Memberikan Hak Akses.

Berikan kembali hak akses ke user yang sudah dihapuskan hak aksesnya menggunakan perintah “GRANT” dengan beberapa opsi yang berlaku seperti ‘SELECT’, ‘INSERT’, ‘UPDATE’, atau ‘DELETE’.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON  
`db`.* TO 'user'@'host';
```

e. Langkah 5. Melakukan flush.

Setelah selesai jangan lupa untuk melakukan flush. Perintah ini berguna untuk memperbarui dan menerapkan aturan baru yang berlaku.

```
flush privileges;
```

M. Linear Interactive Discrete Optimizer (LINDO)

LINDO adalah sebuah program komputer yang digunakan untuk dapat menyelesaikan permasalahan *Linear Programming*, yaitu suatu pemodelan matematik yang digunakan untuk mengoptimalkan suatu tujuan dengan berbagai kendala yang ada. Software ini merupakan program interaktif yang memudahkan bagi penggunaanya. Masalah

program *linear* dengan variabel yang cukup banyak akan lebih mudah diselesaikan menggunakan LINDO.

Program ini merupakan suatu paket program yang berorientasi kepada perintah-perintah dan bukanlah berorientasi pada menu program. Ini berarti bahwa pemakai tidak dituntut dalam suatu urutan pilihan, melainkan terdapat sejumlah perintah-perintah yang harus dipilih dan dijalankan. Prinsip kerja utama LINDO adalah memasukkan data, menyelesaikan serta menaksir kebenaran dan kelayakan data berdasarkan penyelesaiannya.

Untuk menggunakan LINDO ada beberapa tahapan yang perlu dilakukan, yaitu :

1. Merumuskan masalah dalam kerangka pemrograman linear
2. Menuliskan dalam bentuk persamaan matematik
3. Menuliskan rumusan ke dalam LINDO dan mengeksekusinya
4. Interpretasi keluaran LINDO

Langkah-langkah yang dilakukan dalam menggunakan LINDO adalah sebagai berikut.

1. Pilih Start-Program-Winston-LINDO.
2. Pada layar akan muncul untitled baru yang siap untuk digunakan. Kata pertama untuk mengawali pengetikan formula pada LINDO adalah MAX atau MIN. Persamaan yang diketikkan setelah kata MAX atau MIN disebut fungsi tujuan. Setelah itu diketikkan suatu batasan yang berupa pertidaksamaan diawali dengan mengetikkan kata SUBJECT TO dan diakhir batasan diketikkan kata END. Contoh penulisannya adalah sebagai berikut:

MIN/MAX $c_1x_1 + c_2x_2 + \dots + c_nx_n$

SUBJECT TO

$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n$

\leq atau $\geq d_1$. $a_{21}x_1 + a_{22}x_2 +$

$\dots + a_{2n}x_n \leq$ atau $\geq d_2$.

```

... ..
am1x1 + am2x2 + ... + amnxn ≤ atau ≥ dm.
x1 ≥ 0
x2 ≥ 0
... ..
xn ≥ 0
END

```

3. Setelah formulasi diketikkan, maka langkah selanjutnya adalah pilih menú *solve*. *LINDO* akan mengoreksi kesalahan pada formula terlebih dahulu, kemudian apabila terjadi kesalahan pada pengetikan, maka kursor akan menunjukan ke arah kesalahan tersebut.
4. Selanjutnya ada pertanyaan untuk menentukan tingkat kesensivitasan solusi. Apabila memilih *yes*, maka nantinya pada *output* akan diberikan keterangan tentang analisis sensitivitas dari persoalan tadi. Tetapi apabila memilih *no*, maka pada *output* tidak akan muncul keterangan tentang analisis sensitivitas.

BAB III

METODE PENELITIAN

A. Tahapan (waktu) Pelaksanaan Penelitian

Pada penelitian aplikasi pengukuran efisiensi Produksi Air Mineral ini, penulis mengambil data pada beberapa Pabrik produksi Air Mineral yang ada di wilayah Aceh Utara dan Kota Lhokseumawe. Proses penelitian ini dilakukan pada bulan Maret 2019 sampai dengan Juni 2019.

1. Lokasi (Tempat) Penelitian/Perancangan

Lokasi (tempat penelitian ini dilakukan di masing-masing Perusahaan Produksi Air Mineral di daerah Kota Lhokseumawe dan Aceh Utara. Perusahaan tersebut adalah PT. Aini Sejahtera, PT. Ima Mountaz Sejahtera, CV. Wendy's, dan CV. Ie Ro. Lokasi ini di ambil karena memiliki segala aspek yang mendukung keperluan penelitian berjalan dengan baik.

B. Pengumpulan Data

1. Data Primer

Data Primer berupa data produksi air mineral dari masing-masing perusahaan, yang di dapat dari Perusahaan PT. Aini Sejahtera, PT. Ima Mountaz Sejahtera, CV. Wendy's, dan CV. Ie Ro. Data yang di ambil memiliki kriteria seperti jumlah produksi, jumlah pelanggan, nilai kepuasan, jumlah tenaga kerja, jumlah jam kerja produksi, biaya bahan baku produksi dan jumlah distributor/supplier. Data tersebut akan menjadi data utama yang akan diolah dalam penelitian ini yang menjadi acuan efisiensi produksi air mineral.

2. Data Sekunder

Untuk menambah kelengkapan dalam penyusunan penelitian ini, penulis mengambil bahan-bahan referensi

pendukung dari beberapa buku, jurnal, maupun *literature online* dan *offline* yang sesuai dengan isi penulisan skripsi ini yaitu tentang efisiensi produksi air mineral metode *Data Envelopment Analysis*, dan UML (*Unified Modelling Language*).

C. Analisis dan Pengolahan Data

Tahap analisa kebutuhan sistem ini merupakan tahap menganalisa sistem yang akan dibangun. Setelah analisa didapatkan, maka langkah selanjutnya adalah membuat sebuah hasil analisis. Tujuan dari analisa kebutuhan sistem ini untuk memahami dengan sesungguhnya kebutuhan dari sistem yang akan dibangun dan mengembangkan sebuah sistem yang memadai atau memutuskan bahwa pengembangan sistem yang baru tidak dibutuhkan. Hasil analisa tersebut akan menjadi acuan dari perancangan sistem yang akan di bangun.

1. Analisa Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras (*Hardware*) merupakan perangkat yang sangat diperlukan di dalam sistem komputer, perangkat keras yang dengan spesifikasi yang dapat digunakan pada penelitian ini adalah sebagai berikut.

- a. ASUS Intel Core i5 A455L
- b. HDD 500 GB
- c. RAM 4 GB

2. Analisa Kebutuhan Perangkat Lunak (*Software*)

Analisis kebutuhan merupakan langkah awal untuk menentukan gambaran perangkat yang akan dihasilkan ketika pengembang melaksanakan sebuah pembuatan sistem atau aplikasi yang menggunakan perangkat lunak lainnya. Perangkat lunak (*Software*) yang baik dan sesuai dengan kebutuhan pengguna sangat berpengaruh pada keberhasilan dalam melakukan analisis kebutuhan.

Perangkat lunak (*Software*) adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer,

data elektronik yang disimpan oleh komputer itu dapat berupa program atau instruksi yang akan menjalankan suatu perintah. Perangkat lunak (*Software*) disebut juga sebagai penerjemah perintah-perintah yang dijalankan pengguna komputer untuk diteruskan atau diproses oleh perangkat keras. Melalui perangkat lunak (*Software*) inilah suatu komputer dapat menjalankan suatu perintah.

Perangkat lunak (*Software*) inilah merupakan salah satu faktor penunjang untuk membuat dan merancang sebuah sistem. Perangkat lunak (*Software*) berfungsi sebagai pengolah data. Perangkat lunak yang digunakan pada perancangan sistem ini adalah sebagai berikut.

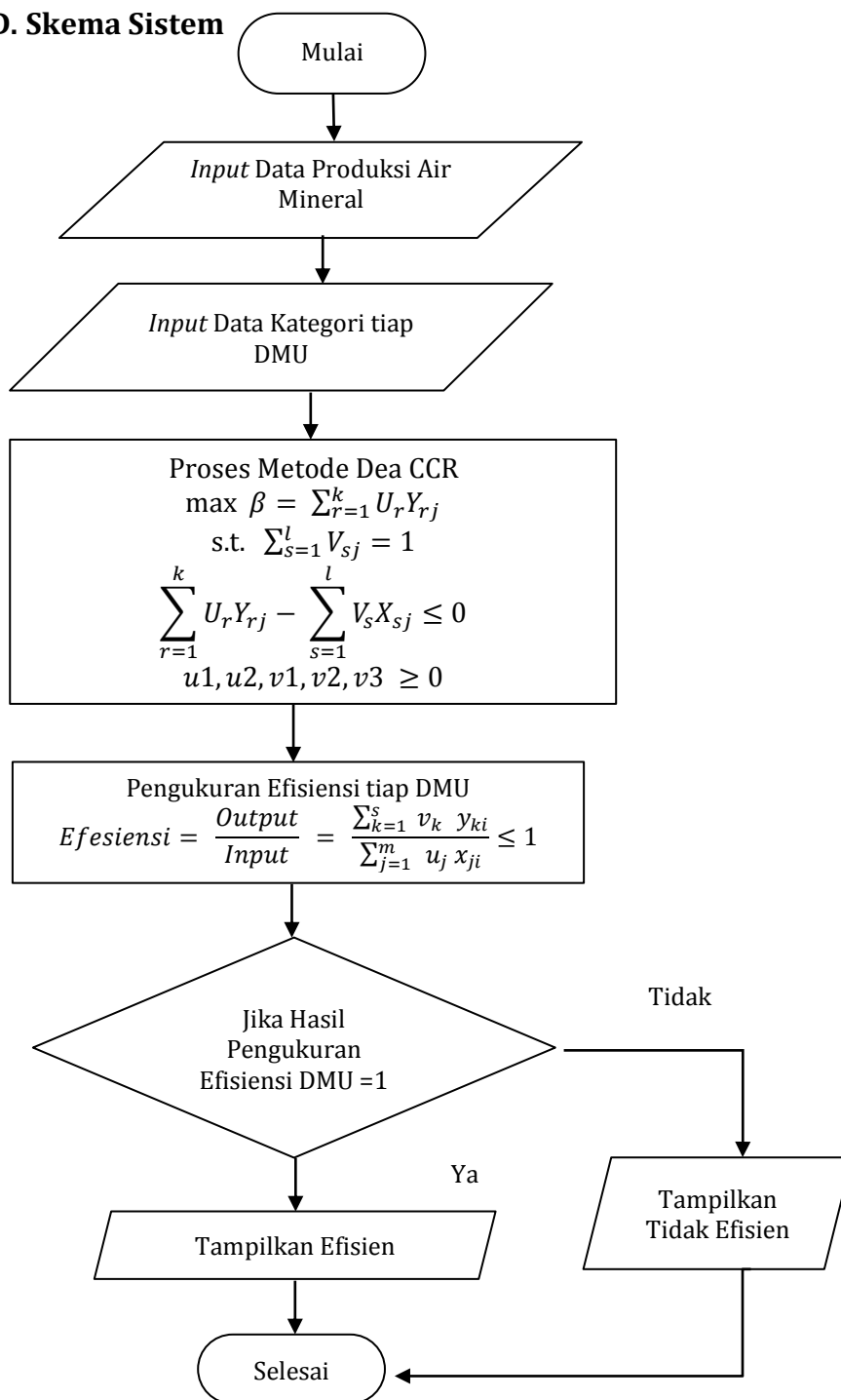
- a. Sistem Operasi Windows 10 Professional 64 bit
 - b. Microsoft Office 2016
 - c. Visual Paradigm 16.1
 - d. Google Chrome
 - e. XAMPP Versi 7.3.12
 - f. Sublime Text 3 Versi 3.2.2
 - g. LINDO 6.1
 - h. *Mysql* dan PHP
3. Analisa Kebutuhan *Input*

Data-data yang dimasukkan untuk menentukan efisiensi produksi air mineral adalah variabel *input* yang meliputi jumlah tenaga kerja, jumlah jam kerja produksi, biaya bahan baku produksi, jumlah *distributor/supplier*, dan variabel *output* yaitu jumlah produksi, jumlah pelanggan dan jumlah nilai kepuasan konsumen.

4. Analisa Kebutuhan *Output*

Output data yang ditampilkan oleh Aplikasi ini adalah berupa daftar efisiensi dari kedua variabel *input* dan *output* yang menggunakan metode *Data Envelopment Analysis* (DEA) untuk proses menghitung nilai efisiensi tersebut. Adapun rancangan *interface output* dapat dilihat pada BAB IV.

D. Skema Sistem



Gambar 3. 1 Skema Sistem

Skema sistem Pengukuran Efisiensi Produksi Air Mineral Dengan Menggunakan *Metode Data Envelopment Analysis* (DEA) Di Wilayah Aceh Utara dan Kota Lhokseumawe. Pada skema sistem yang telah disebutkan menjelaskan tentang alur kerja sistem yang akan dirancang. Adapun langkah-langkah proses pencarian DEA pada sistem yang akan dirancang adalah sebagai berikut.

1. Langkah yang pertama dimulai dengan menginput data produksi air mineral.
2. Selanjutnya adalah menginput data kategori DMU (*Decision Making Unit*)
3. Program akan melakukan proses pencarian menggunakan Metode DEA CCR. Menentukan fungsi tujuan dan fungsi kendala .
4. Untuk mengukur efisiensi tiap DMU akan dihitung menggunakan rumus *output/input*
5. Jika hasil efisiensi DMU = 1, maka status DMU tersebut adalah efisien
6. Jika hasil efisiensi DMU $\neq 1$, maka status DMU tersebut adalah tidak efisien.

BAB IV

HASIL DAN PEMBAHASAN

A. Hasil Penelitian/Perancangan

Pada penelitian ini, penulis akan menguji metode *Data Envelopment Analysis* untuk menghitung efisiensi produksi air mineral. Untuk mempertahankan bisnis perusahaan, seorang produsen harus bekerja secara efisien. Perusahaan harus mampu memahami minat konsumen terhadap produk, mengenal keinginan konsumen dan hal-hal yang mempengaruhi minat konsumen terhadap produk yang ditawarkan.

Perusahaan perlu melakukan suatu cara untuk menjalankan proses produksi yang efisien, yaitu bagaimana menggunakan *input* sehemat mungkin untuk menghasilkan *output* yang sesuai atau melebihi target permintaan yang telah ditetapkan. Metode ini akan menghitung efisiensi produksi. Data yang digunakan adalah data dari 4 perusahaan air mineral, yaitu data Ainiqua, Mount Aqua, Wendy's dan Ie Ro. Data-data tersebut berupa, jumlah produksi per tahun, jumlah pelanggan, jumlah nilai kepuasan konsumen, jumlah tenaga kerja, jumlah jam kerja produksi, jumlah biaya bahan baku produksi per box, dan jumlah *distributor/supplier*.

1. Analisa Sistem

Analisis sistem merupakan tahapan paling awal dari pengembangan sistem yang menjadi fondasi menentukan keberhasilan sistem informasi yang dihasilkan nantinya. Analisa sistem memiliki tiga tahapan dalam mendeskripsikan pengembangan sistem yaitu : analisis masalah, analisis kebutuhan, analisis proses.

1. Analisis Masalah

Setiap tahunnya jumlah produsen air mineral yang dipasarkan di Aceh jumlahnya terus bertambah dengan berbagai kriteria. Bisnis air minum dalam kemasan menjadi bisnis yang cukup menjanjikan pada saat ini, yang mana kebutuhan terhadap air minum dan pola hidup instan menjadi alternatif bagi konsumen. Peningkatan efisiensi di bagian produksi merupakan salah satu cara yang bisa dilakukan oleh pihak perusahaan agar selalu dapat memenuhi permintaan konsumen, agar mampu bersaing dengan perusahaan lain yang menghasilkan produk yang sama. Perusahaan harus mampu memproduksi produk secara efisien dimana *input* yang digunakan bisa sehemat mungkin dengan menghasilkan *output* yang sesuai atau melebihi target permintaan yang telah ditetapkan.

2. Analisis Kebutuhan

Untuk mempermudah menganalisis sebuah sistem dibutuhkan dua jenis kebutuhan, yaitu kebutuhan fungsional dan kebutuhan non fungsional.

a. Kebutuhan Fungsional

Analisis kebutuhan fungsional disini menjelaskan tentang sistem yang disediakan. Sistem ini melakukan perhitungan efisiensi pada produksi air mineral. Untuk menerapkan sistem tersebut, kebutuhan fungsional yang harus dipenuhi antara lain adalah:

1. Sistem memerlukan data jumlah produksi per tahun, jumlah pelanggan, jumlah nilai kepuasan konsumen, jumlah tenaga kerja, jumlah jam kerja produksi, jumlah biaya bahan baku produksi per box, dan jumlah *distributor/supplier*.

2. Pengelompokan data berdasarkan jenis merek atau perusahaan produksi air mineral.
3. Perhitungan dilakukan menggunakan metode DEA CCR

b. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi antara lain sebagai berikut:

1. *Performa* sistem yang akan dibangun *output* yang dihasilkan berupa hasil efisiensi produksi air mineral.
2. Sistem ini menampilkan hasil dari analisa data yang ada. Sistem akan menampilkan produksi air mineral yang efisien maupun tidak efisien.
3. Sistem yang dibangun tidak memerlukan perangkat tambahan yang dapat mengeluarkan biaya dan bebas digunakan sehingga hemat biaya.

3. Analisis Proses

Membangun sistem perhitungan efisiensi produksi air mineral menggunakan metode *Data Envelopment Analysis* (DEA), hasil dari keluaran sistem perhitungan efisiensi ini adalah status efisiensi/tidak efisien pada tiap perusahaan air mineral.

2. Pemilihan *Decision Making Unit* (DMU)

Decision Making Unit (DMU) yang akan diukur dan dianalisa efisiensinya adalah 3 Merek Produksi Air Mineral di Wilayah kota Lhokseumawe dan 1 di Aceh Utara yang keseluruhannya berjumlah 4 Merek Air Mineral atau 4 DMU. *Decision Making Unit* (DMU) yang sudah dipilih akan dikonversi untuk proses pengolahan data selanjutnya. Konversi Merek Air Mineral ke dalam DMU dapat dilihat pada tabel berikut ini.

Tabel 4. 1 *Decision Making Unit (DMU)*

| Merek Air Mineral | <i>Decision Making Unit (DMU)</i> |
|--------------------------|--|
| AiniQua | DMU 1 |
| Mount Aqua | DMU 2 |
| Wendy's | DMU 3 |
| Ie Ro | DMU 4 |

3. Pengelompokan *Input* dan *Output*

Setelah dilakukan klasifikasi DMU maka yang akan dilakukan selanjutnya adalah dengan menganalisa dan mengelompokkan data *input* dan data *output*. Data yang digunakan dalam penelitian ini merupakan variabel-variabel yang saling terkait. Berdasarkan hasil pengamatan dilapangan maka didapatkan variabel *input* dan *output* yang akan digunakan dalam pengolahan data adalah sebagai berikut.

Tabel 4. 2 Pengelompokan *Input* dan *Output*

| No. | Kode Variabel | Nama Variabel | Status |
|------------|----------------------|-----------------------------------|---------------|
| 1. | V1 | Jumlah Tenaga Kerja | <i>Input</i> |
| 2. | V2 | Jumlah Jam Kerja Produksi | <i>Input</i> |
| 3. | V3 | Biaya Bahan Baku Produksi Per Box | <i>Input</i> |
| 4. | V4 | Jumlah Distributor/Supplier | <i>Input</i> |
| 5. | U1 | Jumlah Produksi | <i>Output</i> |
| 6. | U2 | Jumlah Pelanggan | <i>Output</i> |
| 7. | U3 | Nilai Kepuasan Konsumen | <i>Output</i> |

4. Model Matematis DEA

Pemodelan matematis DEA ini dilakukan untuk memperoleh nilai efisiensi dari masing-masing DMU dengan menggunakan data yang telah didapatkan sebelumnya. Berikut adalah contoh model matematis DEA

CCR dengan menggunakan software LINDO sebagai pendukung perhitungan program *linear* (Abdullah 2018)

$$\text{Maksimum } Z = \sum_{k=1}^s V_k Y_{kp}$$

$$\text{s.t.} \quad \sum_{j=1}^m V_j X_{jp} = 1$$

$$\sum_{k=1}^s V_k Y_{ki} - \sum_{j=1}^m U_j X_{ji} \leq 0$$

$$V_k, U_j \geq 0$$

5. Data Awal *Decision Making Unit* (DMU)

Berikut ini data yang digunakan sebagai masukan (*input*) untuk masing-masing DMU yang di ambil dari jumlah keseluruhan setiap *input* dan *output* dari setiap produk air mineral yang sudah disebutkan.

Tabel 4. 3 Data Awal DMU

| DMU | V1 | V2 | V3 | V4 | U1 | U2 | U3 |
|------------|-----|------|-------|----|--------|-----|------|
| AiniQua | 29 | 2051 | 10339 | 25 | 432140 | 150 | 4050 |
| Mount Aqua | 121 | 2051 | 11602 | 30 | 651422 | 350 | 4631 |
| Wendy's | 25 | 2637 | 10120 | 25 | 334055 | 100 | 4163 |
| Ie Ro | 20 | 2051 | 10050 | 20 | 185000 | 55 | 3946 |

6. Perhitungan DEA CCR

a. Menghitung Rasio Bobot

Rasio bobot didapatkan menggunakan Linear Programming yang didapat dengan menggunakan program pendukung LINDO, adapun Linear Programming rasio bobot setiap DMU adalah sebagai berikut:

1. *Linear Programming* Bobot Ainiqua

$$\text{Max } 432140U_1 + 120U_2 + 4454U_3$$

St

$$29V_1 + 2051V_2 + 10339V_3 + 25V_4 = 1$$

$$432140U_1 + 120U_2 + 4454U_3 - 29V_1 - 2051V_2 - 10339V_3 - 25V_4 \leq 0$$

$$651422U_1 + 250U_2 + 4631U_3 - 121V_1 - 2051V_2 - 11602V_3 - 30V_4 \leq 0$$

$334055U_1 + 100U_2 + 4163U_3 - 25V_1 - 2637V_2 - 10120V_3 - 25V_4 \leq 0$
 $185000U_1 + 55U_2 + 3946U_3 - 20V_1 - 2051V_2 - 10050V_3 - 20V_4 \leq 0$
 $V_1 \geq 0, V_2 \geq 0, V_3 \geq 0, V_4 \geq 0, U_1 \geq 0, U_2 \geq 0, U_3 \geq 0$
 End

Hasil Komputasi Model:

LP OPTIMUM FOUND AT STEP 0

| OBJECTIVE FUNCTION VALUE | | |
|--------------------------|----------|--------------|
| 1) 1.000000 | | |
| VARIABLE | VALUE | REDUCED COST |
| U1 | 0.000000 | 0.000000 |
| U2 | 0.000000 | 0.000000 |
| U3 | 0.000185 | 0.000000 |
| V1 | 0.000000 | 0.000000 |
| V2 | 0.000000 | 0.000000 |
| V3 | 0.000097 | 0.000000 |
| V4 | 0.000000 | 0.000000 |

| ROW | SLACK OR SURPLUS | DUAL PRICES |
|-----|------------------|-------------|
| 2) | 0.000000 | 1.000000 |
| 3) | 0.000000 | 1.000000 |
| 4) | 0.000000 | 0.000000 |
| 5) | 0.072638 | 0.000000 |
| 6) | 0.166786 | 0.000000 |
| 7) | 0.000000 | 0.000000 |
| 8) | 0.000000 | 0.000000 |
| 9) | 0.000097 | 0.000000 |
| 10) | 0.000000 | 0.000000 |
| 11) | 0.000000 | 0.000000 |
| 12) | 0.000000 | 0.000000 |
| 13) | 0.000185 | 0.000000 |

NO. ITERATIONS = 0

RANGES IN WHICH THE BASIS IS UNCHANGED:

| OBJ COEFFICIENT RANGES | | | |
|------------------------|---------------|--------------------|--------------------|
| VARIABLE | CURRENT COEF | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| U1 | 432140.000000 | 0.000000 | 0.000000 |
| U2 | 120.000000 | 0.000000 | INFINITY |
| U3 | 4454.000000 | 0.000000 | 0.000000 |
| V1 | 0.000000 | 0.000000 | INFINITY |

| | | | |
|----|----------|----------|----------|
| V2 | 0.000000 | 0.000000 | INFINITY |
| V3 | 0.000000 | INFINITY | 0.000000 |
| V4 | 0.000000 | 0.000000 | INFINITY |

| RIGHTHAND SIDE RANGES | | | |
|-----------------------|-------------|--------------------|--------------------|
| ROW | CURRENT RHS | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| 2 | 1.000000 | INFINITY | 1.000000 |
| 3 | 0.000000 | 0.056134 | 0.255583 |
| 4 | 0.000000 | 0.385274 | 0.082419 |
| 5 | 0.000000 | INFINITY | 0.072638 |
| 6 | 0.000000 | INFINITY | 0.166786 |
| 7 | 0.000000 | 0.000000 | INFINITY |
| 8 | 0.000000 | 0.000000 | INFINITY |
| 9 | 0.000000 | 0.000097 | INFINITY |
| 10 | 0.000000 | 0.000000 | INFINITY |
| 11 | 0.000000 | 0.000000 | INFINITY |
| 12 | 0.000000 | 0.000000 | INFINITY |
| 13 | 0.000000 | 0.000185 | INFINITY |

Hasil yang di dapat dari LINDO adalah perolehan nilai bobot untuk masing-masing variabel yang akan digunakan untuk pencarian nilai virtual *input* dan nilai virtual *output* pada pencarian selanjutnya. Berikut adalah tabel dari nilai bobot yang diperoleh untuk Ainiqua:

Tabel 4. 4 Input-Output Linear Programming Bobot Ainiqua

| INPUT | | | | OUTPUT | | |
|--------------|----|----------|----|---------------|----|----------|
| V1 | V2 | V3 | V4 | U1 | U2 | U3 |
| 0 | 0 | 0.000097 | 0 | 0 | 0 | 0.000185 |

2. Linear Programming Bobot Mount Aqua

$$\text{Max } 651422U_1 + 250U_2 + 4631U_3$$

St

$$121V_1 + 2051V_2 + 11602V_3 + 30V_4 = 1$$

$$432140U_1 + 120U_2 + 4454U_3 - 29V_1 - 2051V_2 - 10339V_3 - 25V_4 \leq 0$$

$$651422U_1 + 250U_2 + 4631U_3 - 121V_1 - 2051V_2 - 11602V_3 - 30V_4 \leq 0$$

$334055U_1 + 100U_2 + 4163U_3 - 25V_1 - 2637V_2 - 10120V_3 - 25V_4 \leq 0$
 $185000U_1 + 55U_2 + 3946U_3 - 20V_1 - 2051V_2 - 10050V_3 - 20V_4 \leq 0$
 $V_1 \geq 0, V_2 \geq 0, V_3 \geq 0, V_4 \geq 0, U_1 \geq 0, U_2 \geq 0, U_3 \geq 0$
 End

Hasil Komputasi Model:

LP OPTIMUM FOUND AT STEP 2

| OBJECTIVE FUNCTION VALUE | | |
|--------------------------|----------|--------------|
| 1) 1.000000 | | |
| VARIABLE | VALUE | REDUCED COST |
| U1 | 0.000000 | 0.000000 |
| U2 | 0.004000 | 0.000000 |
| U3 | 0.000000 | 0.000000 |
| V1 | 0.000000 | 0.000000 |
| V2 | 0.000000 | 0.000000 |
| V3 | 0.000086 | 0.000000 |
| V4 | 0.000000 | 0.000000 |

| ROW | SLACK OR SURPLUS | DUAL PRICES |
|-----|------------------|-------------|
| 2) | 0.000000 | 1.000000 |
| 3) | 0.411139 | 0.000000 |
| 4) | 0.000000 | 1.000000 |
| 5) | 0.472263 | 0.000000 |
| 6) | 0.646230 | 0.000000 |
| 7) | 0.000000 | 0.000000 |
| 8) | 0.000000 | 0.000000 |
| 9) | 0.000086 | 0.000000 |
| 10) | 0.000000 | 0.000000 |
| 11) | 0.000000 | 0.000000 |
| 12) | 0.004000 | 0.000000 |
| 13) | 0.000000 | 0.000000 |

NO. ITERATIONS = 2

RANGES IN WHICH THE BASIS IS UNCHANGED:

| OBJ COEFFICIENT RANGES | | | |
|------------------------|---------------|--------------------|--------------------|
| VARIABLE | CURRENT COEF | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| U1 | 651422.000000 | 0.000000 | INFINITY |
| U2 | 250.000000 | INFINITY | 0.000000 |
| U3 | 4631.000000 | 0.000000 | INFINITY |
| V1 | 0.000000 | 0.000000 | INFINITY |
| V2 | 0.000000 | 0.000000 | INFINITY |

| | | | |
|----|----------|----------|----------|
| V3 | 0.000000 | INFINITY | 0.000000 |
| V4 | 0.000000 | 0.000000 | INFINITY |

| RIGHTHAND SIDE RANGES | | | |
|-----------------------|-------------|--------------------|--------------------|
| ROW | CURRENT RHS | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| 2 | 1.000000 | INFINITY | 1.000000 |
| 3 | 0.000000 | INFINITY | 0.411139 |
| 4 | 0.000000 | 0.856541 | 1.000000 |
| 5 | 0.000000 | INFINITY | 0.472263 |
| 6 | 0.000000 | INFINITY | 0.646230 |
| 7 | 0.000000 | 0.000000 | INFINITY |
| 8 | 0.000000 | 0.000000 | INFINITY |
| 9 | 0.000000 | 0.000086 | INFINITY |
| 10 | 0.000000 | 0.000000 | INFINITY |
| 11 | 0.000000 | 0.000000 | INFINITY |
| 12 | 0.000000 | 0.004000 | INFINITY |
| 13 | 0.000000 | 0.000000 | INFINITY |

LP OPTIMUM FOUND AT STEP 0

| OBJECTIVE FUNCTION VALUE | | |
|--------------------------|----------|--------------|
| 1) 1.000000 | | |
| VARIABLE | VALUE | REDUCED COST |
| U1 | 0.000000 | 0.000000 |
| U2 | 0.004000 | 0.000000 |
| U3 | 0.000000 | 0.000000 |
| V1 | 0.000000 | 0.000000 |
| V2 | 0.000000 | 0.000000 |
| V3 | 0.000086 | 0.000000 |
| V4 | 0.000000 | 0.000000 |

| ROW | SLACK OR SURPLUS | DUAL PRICES |
|-----|------------------|-------------|
| 2) | 0.000000 | 1.000000 |
| 3) | 0.411139 | 0.000000 |
| 4) | 0.000000 | 1.000000 |
| 5) | 0.472263 | 0.000000 |
| 6) | 0.646230 | 0.000000 |
| 7) | 0.000000 | 0.000000 |
| 8) | 0.000000 | 0.000000 |
| 9) | 0.000086 | 0.000000 |
| 10) | 0.000000 | 0.000000 |
| 11) | 0.000000 | 0.000000 |
| 12) | 0.004000 | 0.000000 |
| 13) | 0.000000 | 0.000000 |

NO. ITERATIONS = 0

RANGES IN WHICH THE BASIS IS UNCHANGED:

| OBJ COEFFICIENT RANGES | | | |
|------------------------|---------------|--------------------|--------------------|
| VARIABLE | CURRENT COEF | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| U1 | 651422.000000 | 0.000000 | INFINITY |
| U2 | 250.000000 | INFINITY | 0.000000 |
| U3 | 4631.000000 | 0.000000 | INFINITY |
| V1 | 0.000000 | 0.000000 | INFINITY |
| V2 | 0.000000 | 0.000000 | INFINITY |
| V3 | 0.000000 | INFINITY | 0.000000 |
| V4 | 0.000000 | 0.000000 | INFINITY |

| RIGHTHAND SIDE RANGES | | | |
|-----------------------|-------------|--------------------|--------------------|
| ROW | CURRENT RHS | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| 2 | 1.000000 | INFINITY | 1.000000 |
| 3 | 0.000000 | INFINITY | 0.411139 |
| 4 | 0.000000 | 0.856541 | 1.000000 |
| 5 | 0.000000 | INFINITY | 0.472263 |
| 6 | 0.000000 | INFINITY | 0.646230 |
| 7 | 0.000000 | 0.000000 | INFINITY |
| 8 | 0.000000 | 0.000000 | INFINITY |
| 9 | 0.000000 | 0.000086 | INFINITY |
| 10 | 0.000000 | 0.000000 | INFINITY |
| 11 | 0.000000 | 0.000000 | INFINITY |
| 12 | 0.000000 | 0.004000 | INFINITY |
| 13 | 0.000000 | 0.000000 | INFINITY |

Table 4. 5 *Input-Output Linear Programming* Bobot Mount Aqua

| INPUT | | | | OUTPUT | | |
|-------|----|----------|----|--------|----------|----|
| V1 | V2 | V3 | V4 | U1 | U2 | U3 |
| 0 | 0 | 0.000086 | 0 | 0 | 0.004000 | 0 |

3. *Linear Programming* Bobot Wendy's

Max $334055U_1 + 100U_2 + 4163U_3$

St

$25V_1 + 2637V_2 + 10120V_3 + 25V_4 = 1$

$432140U_1 + 120U_2 + 4454U_3 - 29V_1 - 2051V_2 - 10339V_3 -$

$25V_4 \leq 0$

$$651422U1+250U2+4631U3-121V1-2051V2-11602V3-30V4 \leq 0$$

$$334055U1+100U2+4163U3-25V1-2637V2-10120V3-25V4 \leq 0$$

$$185000U1+55U2+3946U3-20V1-2051V2-10050V3-20V4 \leq 0$$

$$V1 \geq 0, V2 \geq 0, V3 \geq 0, V4 \geq 0, U1 \geq 0, U2 \geq 0, U3 \geq 0 \text{ end}$$

Hasil Komputasi Model:

LP OPTIMUM FOUND AT STEP 4

| OBJECTIVE FUNCTION VALUE | | |
|--------------------------|----------|--------------|
| 1) 1.000000 | | |
| VARIABLE | VALUE | REDUCED COST |
| U1 | 0.000001 | 0.000000 |
| U2 | 0.000000 | 0.000000 |
| U3 | 0.000175 | 0.000000 |
| V1 | 0.031443 | 0.000000 |
| V2 | 0.000000 | 0.000000 |
| V3 | 0.000021 | 0.000000 |
| V4 | 0.000000 | 0.000000 |

| ROW | SLACK OR SURPLUS | DUAL PRICES |
|-----|------------------|-------------|
| 2) | 0.000000 | 1.000000 |
| 3) | 0.000000 | 0.000000 |
| 4) | 2.710909 | 0.000000 |
| 5) | 0.000000 | 1.000000 |
| 6) | 0.000000 | 0.000000 |
| 7) | 0.031443 | 0.000000 |
| 8) | 0.000000 | 0.000000 |
| 9) | 0.000021 | 0.000000 |
| 10) | 0.000000 | 0.000000 |
| 11) | 0.000001 | 0.000000 |
| 12) | 0.000000 | 0.000000 |
| 13) | 0.000175 | 0.000000 |

NO. ITERATIONS = 4

RANGES IN WHICH THE BASIS IS UNCHANGED:

| OBJ COEFFICIENT RANGES | | | |
|------------------------|---------------|--------------------|--------------------|
| VARIABLE | CURRENT COEF | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| U1 | 334055.000000 | 0.000000 | 0.000000 |
| U2 | 100.000000 | 0.000000 | INFINITY |
| U3 | 4163.000000 | 0.000000 | 0.000000 |
| V1 | 0.000000 | 0.000000 | 0.000000 |

| | | | |
|----|----------|----------|----------|
| V2 | 0.000000 | 0.000000 | INFINITY |
| V3 | 0.000000 | 0.000000 | 0.000000 |
| V4 | 0.000000 | 0.000000 | INFINITY |

| RIGHTHAND SIDE RANGES | | | |
|-----------------------|-------------|--------------------|--------------------|
| ROW | CURRENT RHS | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| 2 | 1.000000 | INFINITY | 1.000000 |
| 3 | 0.000000 | 0.015865 | 0.006149 |
| 4 | 0.000000 | INFINITY | 2.710909 |
| 5 | 0.000000 | 0.003824 | 0.009070 |
| 6 | 0.000000 | 0.022140 | 0.010832 |
| 7 | 0.000000 | 0.031443 | INFINITY |
| 8 | 0.000000 | 0.000000 | INFINITY |
| 9 | 0.000000 | 0.000021 | INFINITY |
| 10 | 0.000000 | 0.000000 | INFINITY |
| 11 | 0.000000 | 0.000001 | INFINITY |
| 12 | 0.000000 | 0.000000 | INFINITY |
| 13 | 0.000000 | 0.000175 | INFINITY |

Tabel 4. 6 *Input-Output Linear Programming* Bobot Wendy's

| INPUT | | | | OUTPUT | | |
|----------|----|----------|----|----------|----|----------|
| V1 | V2 | V3 | V4 | U1 | U2 | U3 |
| 0.031443 | 0 | 0.000021 | 0 | 0.000001 | 0 | 0.000175 |

4. *Linear Programming* Bobot Ie Ro

Max $185000U_1 + 55U_2 + 3946U_3$

St

$20V_1 + 2051V_2 + 10050V_3 + 20V_4 = 1$

$432140U_1 + 120U_2 + 4454U_3 - 29V_1 - 2051V_2 - 10339V_3 - 25V_4 \leq 0$

$651422U_1 + 250U_2 + 4631U_3 - 121V_1 - 2051V_2 - 11602V_3 - 30V_4 \leq 0$

$334055U_1 + 100U_2 + 4163U_3 - 25V_1 - 2637V_2 - 10120V_3 - 25V_4 \leq 0$

$185000U_1 + 55U_2 + 3946U_3 - 20V_1 - 2051V_2 - 10050V_3 - 20V_4 \leq 0$

$V_1 \geq 0, V_2 \geq 0, V_3 \geq 0, V_4 \geq 0, U_1 \geq 0, U_2 \geq 0, U_3 \geq 0$

end

Hasil Komputasi Model:
 LP OPTIMUM FOUND AT STEP 3

| OBJECTIVE FUNCTION VALUE | | |
|--------------------------|----------|--------------|
| 1) 1.000000 | | |
| VARIABLE | VALUE | REDUCED COST |
| U1 | 0.000000 | 0.000000 |
| U2 | 0.000000 | 0.000000 |
| U3 | 0.000253 | 0.000000 |
| V1 | 0.011867 | 0.000000 |
| V2 | 0.000000 | 0.000000 |
| V3 | 0.000076 | 0.000000 |
| V4 | 0.000000 | 0.000000 |

| ROW | SLACK OR SURPLUS | DUAL PRICES |
|-----|------------------|-------------|
| 2) | 0.000000 | 1.000000 |
| 3) | 0.000000 | 0.000000 |
| 4) | 1.142793 | 0.000000 |
| 5) | 0.009657 | 0.000000 |
| 6) | 0.000000 | 1.000000 |
| 7) | 0.011867 | 0.000000 |
| 8) | 0.000000 | 0.000000 |
| 9) | 0.000076 | 0.000000 |
| 10) | 0.000000 | 0.000000 |
| 11) | 0.000000 | 0.000000 |
| 12) | 0.000000 | 0.000000 |
| 13) | 0.000253 | 0.000000 |

NO. ITERATIONS = 3

RANGES IN WHICH THE BASIS IS UNCHANGED:

| OBJ COEFFICIENT RANGES | | | |
|------------------------|---------------|--------------------|--------------------|
| VARIABLE | CURRENT COEF | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| U1 | 185000.000000 | 0.000000 | INFINITY |
| U2 | 55.000000 | 0.000000 | INFINITY |
| U3 | 3946.000000 | INFINITY | 0.000000 |
| V1 | 0.000000 | 0.000000 | 0.000000 |
| V2 | 0.000000 | 0.000000 | INFINITY |
| V3 | 0.000000 | 3750.649170 | 0.000000 |
| V4 | 0.000000 | 0.000000 | INFINITY |

| RIGHTHAND SIDE RANGES | | | |
|-----------------------|-------------|--------------------|--------------------|
| ROW | CURRENT RHS | ALLOWABLE INCREASE | ALLOWABLE DECREASE |
| 2 | 1.000000 | INFINITY | 1.000000 |
| 3 | 0.000000 | 0.016738 | 0.321262 |
| 4 | 0.000000 | INFINITY | 1.142793 |
| 5 | 0.000000 | INFINITY | 0.009657 |
| 6 | 0.000000 | 0.023917 | 0.088578 |
| 7 | 0.000000 | 0.011867 | INFINITY |
| 8 | 0.000000 | 0.000000 | INFINITY |
| 9 | 0.000000 | 0.000076 | INFINITY |
| 10 | 0.000000 | 0.000000 | INFINITY |
| 11 | 0.000000 | 0.000000 | INFINITY |
| 12 | 0.000000 | 0.000000 | INFINITY |
| 13 | 0.000000 | 0.000253 | INFINITY |

Tabel 4. 7 *Input-Output Linear Programming* Bobot le Ro

| INPUT | | | | OUTPUT | | |
|----------|----|----------|----|--------|----|----------|
| U1 | U2 | U3 | U4 | V1 | V2 | V3 |
| 0.011867 | 0 | 0.000076 | 0 | 0 | 0 | 0.000253 |

Dari hasil perhitungan rasio bobot menggunakan *Linear Programming* menggunakan LINDO maka di dapat nilai bobot setiap DMU, adapun rasio bobot setiap DMU tersebut adalah sebagai berikut.

Tabel 4. 8 Rasio Bobot Setiap DMU

| TIPE | NAMA VARIABEL | BOBOT | | | |
|------|-----------------------------|----------|----------|----------|----------|
| | | D1 | D2 | D3 | D4 |
| V1 | Jumlah Tenaga Kerja | 0 | 0 | 0.031443 | 0.011867 |
| V2 | Jumlah Jam Kerja Produksi | 0 | 0 | 0 | 0 |
| V3 | Biaya Bahan Baku Produksi | 0.000097 | 0.000086 | 0.000021 | 0.000076 |
| V4 | Jumlah Distributor/Supplier | 0 | 0 | 0 | 0 |

| | | | | | |
|----|-------------------------|----------|----------|----------|----------|
| U1 | Jumlah Produk | 0 | 0 | 0.000001 | 0 |
| U2 | Jumlah Pelanggan | 0 | 0.004000 | 0 | 0 |
| U3 | Nilai Kepuasan Konsumen | 0.000185 | 0 | 0.000175 | 0.000253 |

b. Menghitung *Virtual Input* dan *Output*

Setelah bobot masing-masing variabel DMU diketahui, langkah selanjutnya adalah menghitung nilai *virtual input* dan *output*, adapun perhitungan *virtual input* dan *output* menggunakan formula berikut:

$$\text{Virtual Input} = \sum_{i=1}^I V_i X_i$$

$$\text{Virtual Output} = \sum_{j=1}^J U_j y_j$$

1. Virtual Input Ainiqua

Input

$$\begin{aligned}
 &= \text{bobot V1 (nilai V1)} + \text{bobot V2 (nilai V2)} + \text{bobot V3 (nilai V3)} + \text{bobot V4 (nilai V4)} \\
 &= 0(29) + 0(2051) + 0,000097(10339) + 0(25) \\
 &= 1
 \end{aligned}$$

Output

$$\begin{aligned}
 &= \text{bobot U1 (nilai U1)} + \text{bobot U2 (nilai U2)} + \text{bobot U3 (nilai U3)} \\
 &= 0(432140) + 0(150) + 0,000185(4050) \\
 &= 1
 \end{aligned}$$

2. *Virtual Input* Mount Aqua

Input

$$\begin{aligned}
 &= \text{bobot V1 (nilai V1)} + \text{bobot V2 (nilai V2)} + \text{bobot V3 (nilai V3)} + \text{bobot V4 (nilai V4)} \\
 &= 0(121) + 0(2051) + 0,000086(11602) + 0(30) \\
 &= 1
 \end{aligned}$$

Output

$$\begin{aligned} &= \text{bobot U1 (nilai U1)} + \text{bobot U2 (nilai U2)} + \text{bobot U3 (nilai U3)} \\ &= 0(651422) + 0,004000(350) + 0(4631) \\ &= 1 \end{aligned}$$

3. *Virtual Input Wendy's***Input**

$$\begin{aligned} &= \text{bobot V1 (nilai V1)} + \text{bobot V2 (nilai V2)} + \text{bobot V3 (nilai V3)} + \text{bobot V4 (nilai V4)} \\ &= 0,031443(25) + 0(2637) + 0,000021(10120) + 0(25) \\ &= 1 \end{aligned}$$

Output

$$\begin{aligned} &= \text{bobot U1 (nilai U1)} + \text{bobot U2 (nilai U2)} + \text{bobot U3 (nilai U3)} \\ &= 0,000001(334055) + 0(100) + 0,000175(4163) \\ &= 1 \end{aligned}$$

4. *Virtual Input Ie Ro***Input**

$$\begin{aligned} &= \text{bobot V1 (nilai V1)} + \text{bobot V2 (nilai V2)} + \text{bobot V3 (nilai V3)} + \text{bobot V4 (nilai V4)} \\ &= 0,011867(20) + 0(2051) + 0,000076(10050) + 0(20) \\ &= 1 \end{aligned}$$

Output

$$\begin{aligned} &= \text{bobot U1 (nilai U1)} + \text{bobot U2 (nilai U2)} + \text{bobot U3 (nilai U3)} \\ &= 0(185000) + 0(55) + 0,000253(3946) \\ &= 1 \end{aligned}$$

c. Menghitung Nilai Efisiensi

Nilai efisiensi dihitung menggunakan persamaan matematika berikut:

$$\text{Efisiensi} = \frac{\text{Virtual Output}}{\text{Virtual Input}} = \frac{\sum_{i=1}^I U_i X_i}{\sum_{j=1}^J v_j Y_j}$$

$$\text{Efisiensi D1} = \frac{1}{1} = 1$$

$$\text{Efisiensi D2} = \frac{1}{1} = 1$$

$$\text{Efisiensi D3} = \frac{1}{1} = 1$$

$$\text{Efisiensi D4} = \frac{1}{1} = 1$$

Dari hasil perhitungan nilai rasio efisiensi maka dapat diketahui masing-masing DMU yang memiliki nilai efisien, nilai rasio efisiensi < 1 merupakan DMU yang tidak efisien. Adapun hasil perhitungan rasio efisiensi dapat di lihat pada tabel berikut:

Tabel 4. 9 Hasil Perhitungan Rasio Efisiensi

| Kode | DMU | Rasio Efisiensi | Status |
|------|------------|-----------------|---------|
| D1 | Ainiqua | 1 | Efisien |
| D2 | Mount Aqua | 1 | Efisien |
| D3 | Wendy's | 1 | Efisien |
| D4 | Ie Ro | 1 | Efisien |

7. Manajemen Basis Model

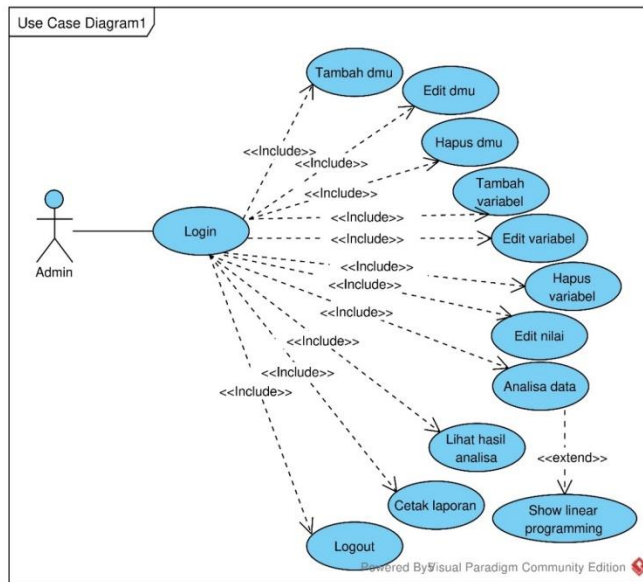
Sistem ini didesain menggunakan UML untuk menjelaskan proses sistem secara visual agar mudah dimengerti bagaimana sistem ini bekerja. Desain ini nantinya akan sangat membantu dalam implementasi ke bahasa pemrograman.

a. Use Case Diagram

Use Case digunakan untuk mengetahui fungsi-fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsional itu. *Use Case Diagram*

ini akan menjelaskan apa-apa fungsi yang akan dikerjakan oleh sistem. Oleh karena itu *Use Case Diagram* akan mempresentasikan bagaimana interaksi antara *user* atau admin dengan sistem.

Berikut Diagram *Use Case* untuk aplikasi pengukuran efisiensi produksi air mineral dengan menggunakan metode DEA.



Gambar 4. 1 *Use Case Diagram*

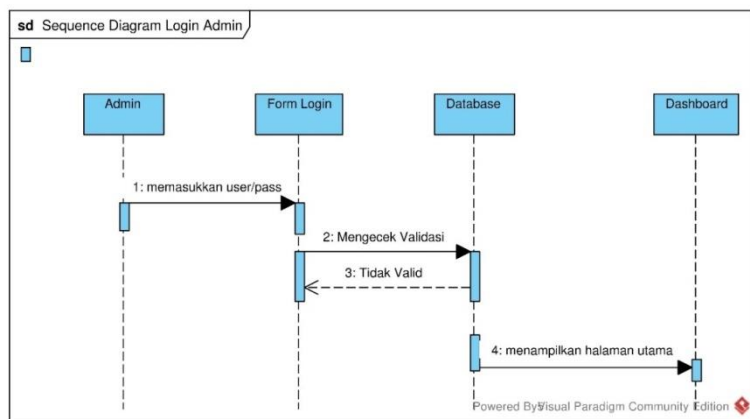
b. *Sequence Diagram*

Sequence diagram yaitu penggambaran kolaborasi antara objek dari kelas-kelas yang ada serta pesan dan jawaban yang diterima atau dikirim oleh objek, oleh karena itu untuk menggambar *Sequence diagram* maka harus diketahui objek-objek yang terlibat di dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek tersebut, membuat diagram ini juga dibutuhkan untuk melihat skenario yang ada pada *Use Case*

diagram. Pada sistem ini *Sequence* diagram terdiri dari beberapa bagian, yaitu sebagai berikut:

1. *Sequence Diagram* Login Admin

Admin membuka sistem, kemudian sistem akan menampilkan halaman *login*, kemudian admin melakukan *login* dengan *username* dan *password*. Setelah admin memasukkan data *login*, sistem akan memvalidasi data login operator pada basis data, jika data login admin salah, maka admin harus memasukkan ulang data login admin. Jika data sesuai maka sistem akan masuk ke halaman *dashboard*. Berikut gambar *Sequence Diagram* Login Admin.

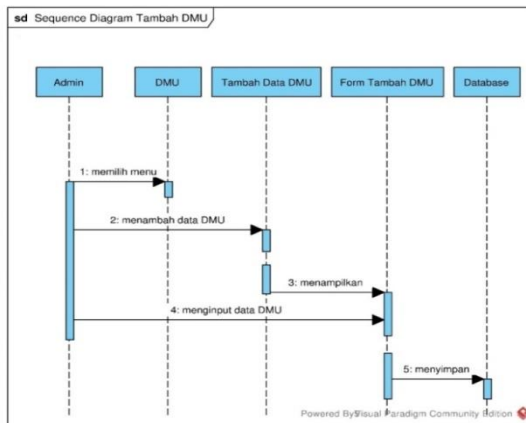


Gambar 4. 2 *Sequence Diagram* Login Admin

2. *Sequence Diagram* Tambah DMU

Diagram *sequence* tambah DMU adalah diagram yang menunjukkan jalannya proses penambahan data DMU oleh admin. Pada diagram ini dijelaskan cara untuk menambah data DMU. Hal yang pertama dilakukan admin adalah memilih menu DMU. Selanjutnya memilih menu tambah DMU. Form DMU akan ditampilkan, admin dapat

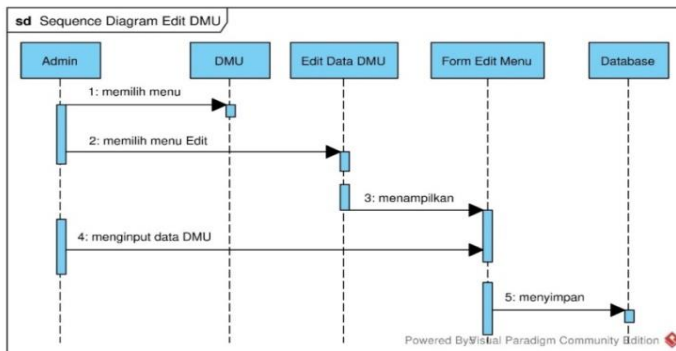
mengisi atau meng-*input* data untuk DMU baru yang akan disimpan pada *database*



Gambar 4. 3 Sequence Diagram Tambah DMU

3. Sequence Diagram Edit DMU

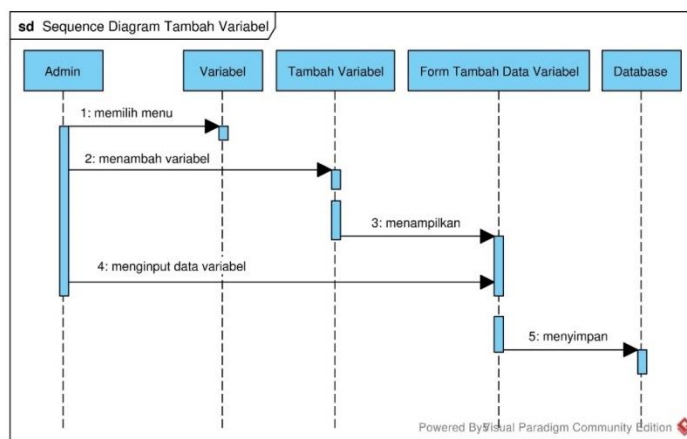
Pada diagram *sequence* edit DMU dijelaskan proses admin untuk mengedit data DMU yang telah di *input* sebelumnya. Jalan kerjanya adalah admin memilih menu DMU lalu setelah berada pada data DMU, admin bisa memilih menu edit untuk DMU yang akan di edit datanya. Setelah form data DMU ditampilkan admin dapat memasukkan data baru untuk DMU dan memilih simpan agar data dapat tersimpan pada *database*.



Gambar 4. 4 Sequence Diagram Edit DMU

4. Sequence Diagram Tambah Variabel

Diagram *Sequence* tambah variabel adalah diagram yang menjelaskan tentang alur penambahan data variabel. Proses kerjanya yaitu admin memilih menu variabel, lalu memilih menu tambah variabel untuk menambah data variabel yang diinginkan. Setelah itu form data variabel akan ditampilkan. Admin dapat mengisi data pada form yang telah tersedia lalu memilih simpan untuk menyimpan data tersebut ke *database*.



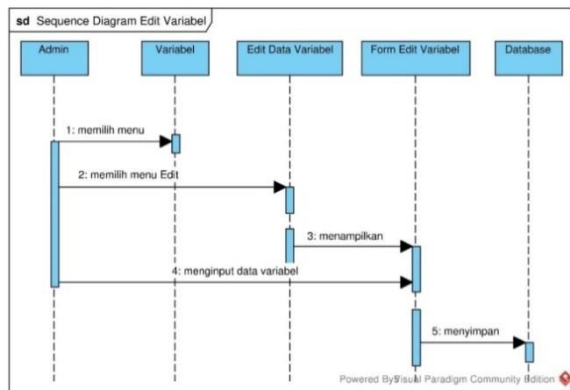
Gambar 4. 5 Sequence Diagram Tambah Variabel

5. Sequence Diagram Edit Variabel

Diagram *sequence* edit variabel adalah diagram yang menjelaskan tentang cara mengedit data variabel yang telah di *input* sebelumnya oleh admin.

Langkah pertama adalah admin memilih menu variabel, setelah itu memilih edit data variabel untuk variabel yang akan di edit datanya. Form edit data variabel akan ditampilkan, admin dapat mengisi atau mengganti data awal dengan data baru

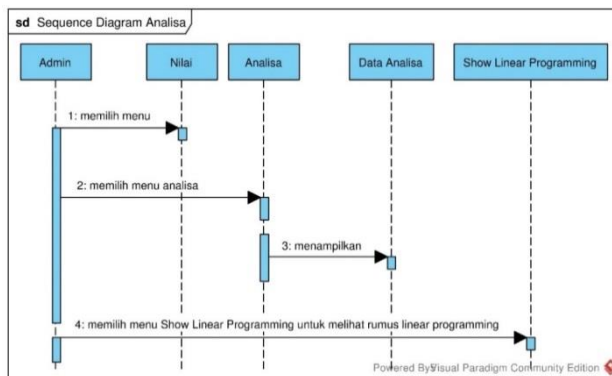
pada form ini Setelah itu memilih simpan untuk menyimpan data yang telah di edit ke *database*.



Gambar 4. 6 *Sequence Diagram* Edit Variabel

6. *Sequence Diagram* Analisa

Digram *sequence* analisa adalah diagram yang menjelaskan tentang proses analisa DEA. Langkah pertama adalah admin memilih menu nilai, setelah itu memilih menu analisa. Data analisa akan ditampilkan. Admin juga dapat memilih *show linear programming* untuk melihat *linear programming*



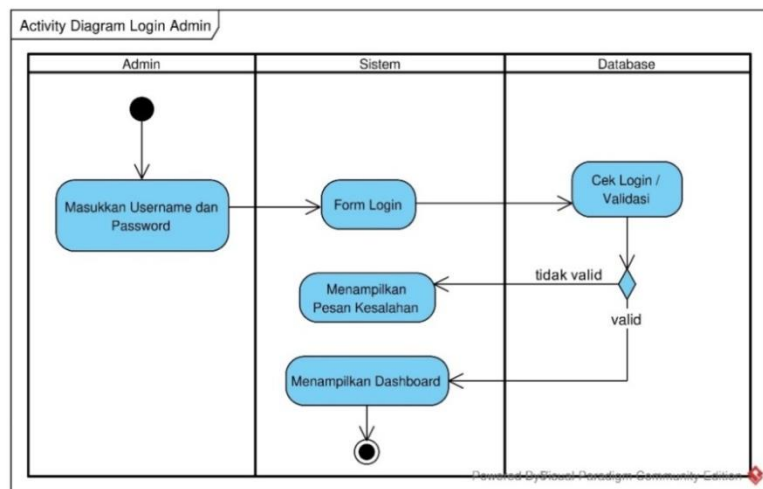
Gambar 4. 7 *Sequence Diagram* Analisa

c. *Activity Diagram*

Activity diagram atau diagram aktivitas menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem maupun proses menu yang ada pada perangkat lunak. Berikut *Activity diagram* pada aplikasi sistem Pengukuran efisiensi produksi air mineral dengan menggunakan metode *Data Envelopment Analysis* di wilayah Lhokseumawe dan Aceh Utara.

1. *Activity diagram login admin*

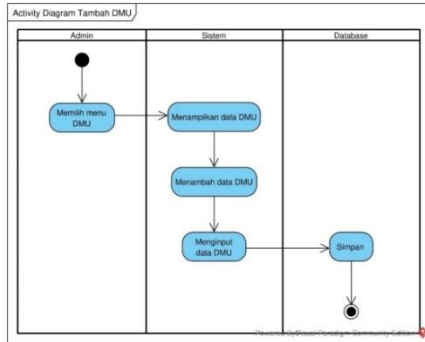
Diagram *activity login admin* adalah diagram yang menjelaskan tentang proses aktivitas *login admin*. Sistem akan menampilkan halaman *Login*. Admin akan diminta untuk memasukkan *username* dan *password*. Lalu sistem akan memvalidasi data *login* pada *database*. Jika data yang dimasukkan salah, maka sistem akan menampilkan pesan kesalahan, dan admin harus memasukkan data *login* kembali. Jika data yang dimasukkan sudah benar, maka sistem akan menampilkan halaman *Dashboard*.



Gambar 4. 8 *Activity diagram login admin*

2. Activity diagram tambah DMU admin

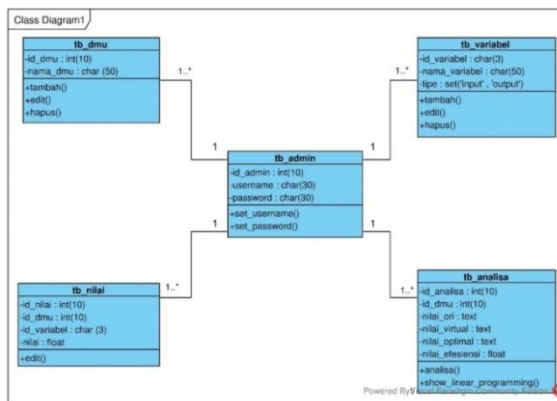
Diagram *activity* tambah DMU adalah diagram yang menjelaskan tentang aktivitas tambah DMU. Untuk menambah DMU Admin harus memilih menu DMU, setelah itu sistem akan menampilkan data DMU. Admin dapat menginput data DMU pada form yang tersedia dan memilih simpan untuk menyimpan data tersebut ke dalam *database*.



Gambar 4. 9 Activity diagram tambah dmu admin

d. Class Diagram

Class diagram digunakan untuk menampilkan beberapa kelas yang ada di dalam sistem yang sedang dibangun. *Class diagram* menunjukkan hubungan antar kelas di dalam sistem dan bagaimana mereka berkolaborasi untuk mencapai suatu tujuan.



Gambar 4. 10 Class Diagram

8. Manajemen Basis Data

a. Tabel tb_admin

Tabel 4. 10 Tabel tb_admin

| No | Nama | Type | Width | Keterangan |
|----|----------|---------|-------|----------------|
| 1 | Id_admin | Int | 11 | Primary key |
| 2 | username | Varchar | 30 | Username admin |
| 3 | password | Varchar | 30 | Password admin |

b. Tabel tb_analisa

Tabel 4. 11 Tabel tb_analisa

| No | Nama | Type | Width | Keterangan |
|----|-----------------|-------|-------|----------------------|
| 1 | Id_analisa | Int | 11 | Primary key |
| 2 | Id_dmu | int | 10 | Id dmu |
| 3 | Nilai_ori | text | - | nilai ori |
| 4 | Nilai_virtual | text | - | Nilai <i>virtual</i> |
| 5 | Nilai_optimal | text | - | Nilai optimal |
| 6 | Nilai_efisiensi | float | - | Nilai efisiensi |
| 7 | slack | float | - | slack |

c. Tabel tb_dmu

Tabel 4. 12 Tabel tb_dmu

| No | Nama | Type | Width | Keterangan |
|----|----------|---------|-------|-------------|
| 1 | Id_dmu | Int | 10 | Primary key |
| 2 | Nama_dmu | Varchar | 50 | Nama dmu |

d. Tabel tb_nilai

Tabel 4. 13 Tabel tb_nilai

| No | Nama | Type | Width | Keterangan |
|----|-------------|---------|-------|-------------|
| 1 | Id_nilai | Int | 11 | Primary key |
| 2 | Id_dmu | int | 10 | Id dmu |
| 3 | Id_variabel | varchar | 3 | Id variabel |
| 4 | Nilai | float | - | Nilai |

e. Tabel tb_variabel

Tabel 4. 14 Tabel tb_variabel

| No | Nama | Type | Width | Keterangan |
|----|---------------|------------------------|-------|---------------|
| 1 | Id_variabel | varchar | 3 | Primary key |
| 2 | Nama_variabel | varchar | 50 | Nama variabel |
| 3 | Tipe | Set ('input','output') | 3 | Tipe data |

B. Pembahasan


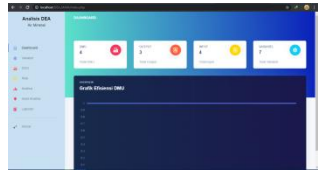

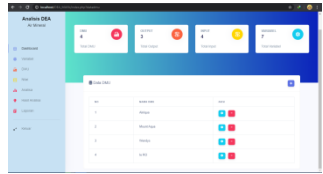
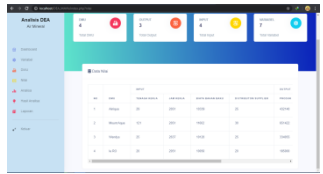

Pada bagian ini, akan membahas berbagai proses dari penelitian yang dilakukan dari data sebenarnya, dimana sistem akan diuji dengan mengimplementasikan rancangan desain program menggunakan bahasa pemrograman. Perancangan sistem ini berawal dari analisa kebutuhan dan masalah-masalah yang ada hingga menemukan solusi praktis menggunakan metode dan algoritma komputer, mendesain proses-proses yang akan dilakukan nanti, implementasi dan pengujian sistem.


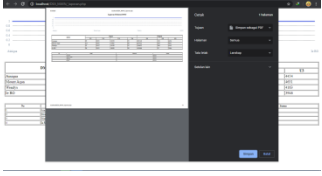
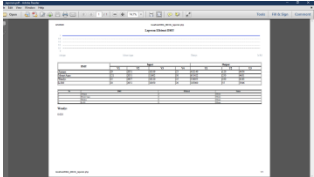
1. Pengujian Sistem

Pengujian sistem merupakan proses yang bertujuan untuk memastikan apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Ada 2 jenis pengujian sistem yang dapat dilakukan pada tahapan ini, yaitu:

a. *White-box testing*


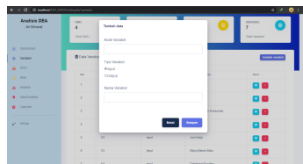
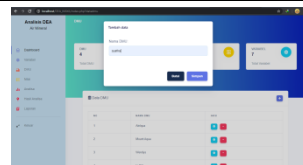
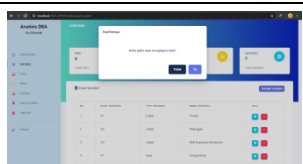
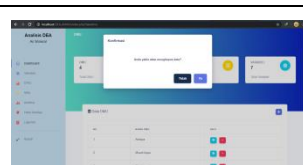
White-box testing berfokus pada struktur *control* program untuk memastikan bahwa semua *statemen* pada program telah di eksekusi paling tidak satu kali selama pengujian dan bahwa semua kondisi logis telah diuji.

| No | Aksi | Hasil yang diterima | Hasil yang diterima | Screenshoot hasil pengujian sistem |
|----|---|--|---------------------|---|
| 1 | Memulai dengan membuka sistem tingkat efisiensi Air Mineral | Menampilkan halaman <i>login</i> | Valid |  |
| 2 | Memilih menu <i>dashboard</i> | Menampilkan grafik dan jumlah <i>input</i> , <i>output</i> dan jumlah variabel | Valid |  |
| 3 | Memilih <i>list</i> menu variabel | Menampilkan <i>list</i> kode variabel, tipe variabel, dan nama variabel | Valid |  |
| 4 | Memilih <i>list</i> menu DMU | Menampilkan <i>list</i> nama DMU dan aksi | Valid |  |
| 5 | Memilih <i>list</i> menu nilai | menampilkan <i>list</i> DMU, <i>input</i> , <i>output</i> , <i>action</i> . | Valid |  |
| 6 | Memilih <i>list</i> menu analisa | Menampilkan <i>list</i> analisa DEA beserta <i>linear programming</i> | Valid |  |

| | | | | |
|---|---|--|-------|--|
| 7 | Memilih <i>list</i> menu hasil analisa | Menampilkan <i>list</i> hasil dari analisa | Valid |  |
| 8 | Memilih <i>list</i> menu laporan | Menampilkan <i>list</i> laporan untuk dicetak | Valid |  |
| 9 | Memilih menu simpan PDF pada menu laporan | laporan akan terdownload dalam <i>format</i> PDF | Valid |  |

b. *Black-box testing*

Black-box testing berfokus pada domain informasi dari perangkat lunak, dengan melakukan *test case* dengan mempartisi domain *input* dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

| No | Aksi | Hasil yang diharapkan | Hasil yang diterima | Screenshot Program |
|----|--|--------------------------------------|---------------------|---|
| 1 | Menginput <i>username</i> dan <i>password</i> benar | Menampilkan halaman <i>Dashboard</i> | Valid |  |
| 2 | Memilih button simpan pada form tambah data variabel | Menyimpan data yang diinputkan | Valid |  |
| 3 | Menekan button simpan pada form tambah data DMU | Menyimpan data yang akan diinputkan | Valid |  |
| 4 | Menekan button hapus pada form tambah variabel | Menghapus data | Valid |  |
| 5 | Menekan button hapus pada form tambah data DMU | Menghapus Data | Valid |  |

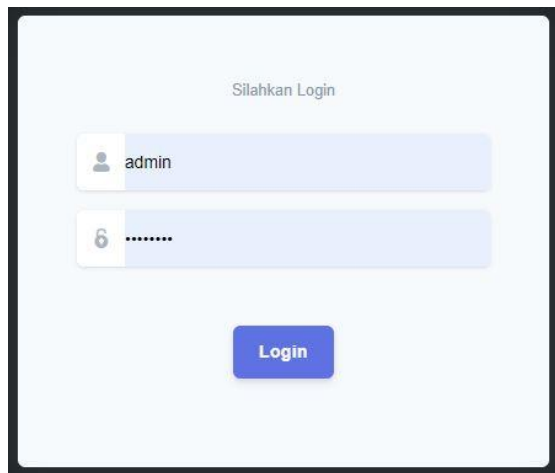
C. Implementasi Antar Muka

Implementasi sistem pada tahap ini melanjutkan konstruksi aplikasi (Construction) dari metode prototype yaitu implementasi dari perancangan sistem yang telah didefinisikan sebelumnya. Tampilan program akan digunakan pengguna untuk berinteraksi dengan perangkat lunak yang dibangun. Implementasi sistem digunakan sebagai tolak ukur pengujian

dari hasil program yang sudah dibuat untuk pembangunan sistem.

1. Halaman *Login*

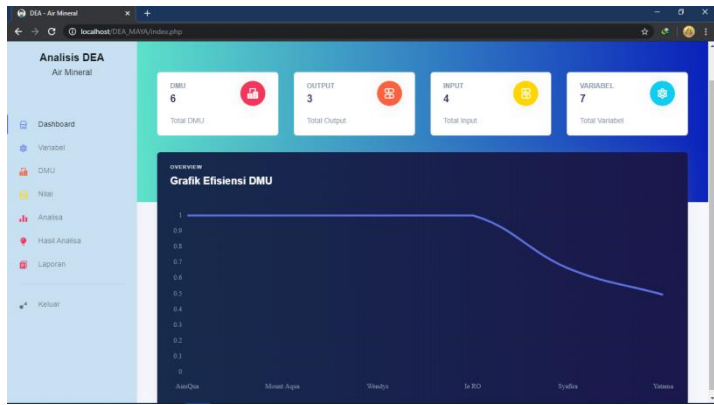
Halaman *Login* merupakan halaman yang digunakan untuk admin atau *user* masuk atau *Login* ke sistem, dengan cara memasukkan *email* dan *password* sistem akan mengecek kevalidan *email* dan *password* yang dimasukkan jika benar maka sistem akan menampilkan halaman *Dashboard*.



Gambar 4. 11 Gambar *Login* Admin

2. Halaman *Dashboard*

Pada halaman *dashboard* terdapat beberapa menu pilihan seperti menu variabel, menu DMU, menu nilai, menu analisa, menu hasil analisa, menu laporan dan menu *log out*. *Dashboard* menampilkan jumlah DMU, jumlah *output*, jumlah *input* dan jumlah variabel. *Dashboard* juga menampilkan grafik trend dari efisiensi tiap DMU yang telah dianalisa.



Gambar 4. 12 Halaman *Dashboard*

3. Halaman Variabel

Halaman variabel menampilkan data variabel yang telah diinput. Pada halaman ini juga dapat ditambahkan data variabel, mengedit atau menghapus data variabel. Data variabel yang ditampilkan antara lain kode variabel, tipe variabel dan nama variabel. Variabel yang telah di *input* dapat di edit maupun dihapus pada pilihan aksi. Hal ini bertujuan untuk memudahkan admin dalam mengedit data yang telah diinput sebelumnya.

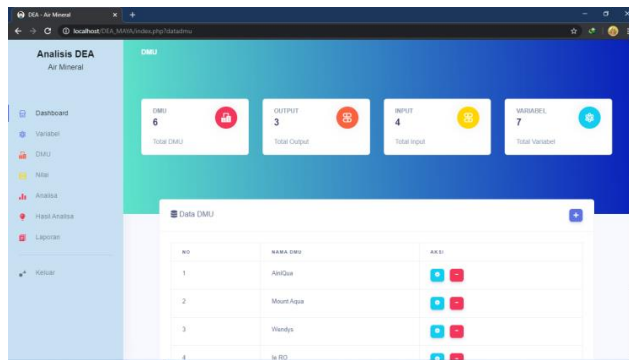
The screenshot shows the 'VARIABEL' page with a table of variable data. The table has five columns: 'NO', 'KODE VARIABEL', 'TIPE VARIABEL', 'NAMA VARIABEL', and 'AKSI'. There are five rows of data, each with a red '+' icon and a red '-' icon in the 'AKSI' column.

| NO | KODE VARIABEL | TIPE VARIABEL | NAMA VARIABEL | AKSI |
|----|---------------|---------------|---------------------------|------|
| 1 | u1 | input | Tanpa Kerja | + - |
| 2 | u2 | input | Jam Kerja Produksi | + - |
| 3 | u3 | input | Biaya Bahan Baku Produksi | + - |
| 4 | u4 | input | Distribusi | + - |
| 5 | u5 | output | Hasil Produksi | + - |

Gambar 4. 13 Halaman Variabe

4. Halaman DMU

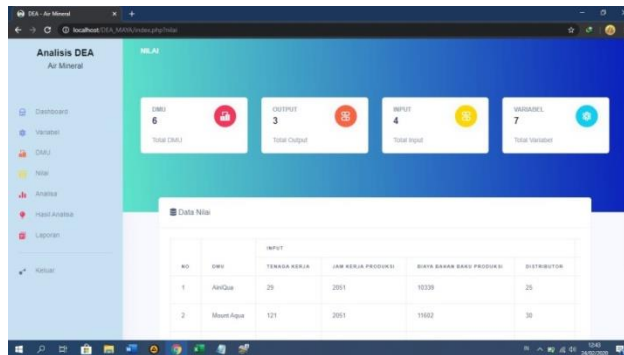
Halaman DMU menampilkan data DMU yang telah diinput. Pada halaman ini admin juga dapat menambahkan DMU yang diinginkan. Data DMU yang tersedia pada halaman ini adalah nama DMU. DMU yang telah di *input* dapat diedit atau dihapus pada pilihan aksi yang tersedia dalam halaman tersebut. Hal ini bertujuan untuk memudahkan admin dalam mengelola dan mengedit maupun menghapus data DMU yang telah di *input*.



Gambar 4. 14 Halaman DM

5. Halaman Nilai

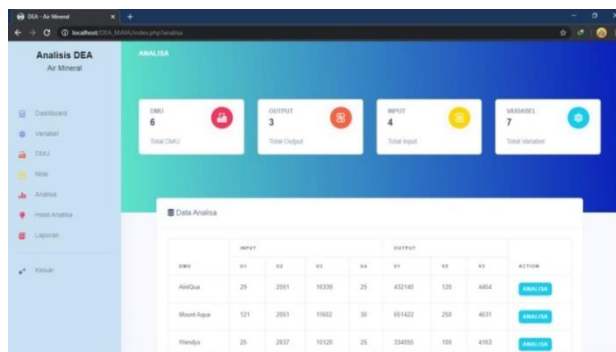
Halaman nilai menampilkan data nilai yang telah ditambahkan oleh admin. Data nilai yang terdapat pada halaman ini antara lain DMU, *input* yang meliputi tenaga kerja, jam kerja, biaya bahan baku, *distributor/supplier* dan *output* yang meliputi jumlah produk, pelanggan, nilai kepuasan konsumen. Pada halaman ini admin hanya mampu mengisi nilai dari variabel yang telah ditentukan sebelumnya.



Gambar 4. 15 Halaman Nilai

6. Halaman Analisa

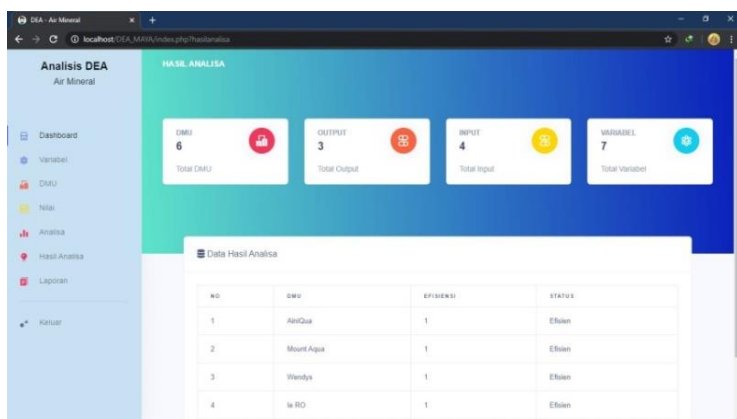
Halaman Analisa ditujukan untuk menganalisis data yang telah tersedia atau di *input* sebelumnya. Halaman ini menampilkan data analisa. Data analisa tersebut akan dianalisis oleh program untuk melihat nilai efisien tidaknya suatu DMU. Data analisa yang ditampilkan antara lain DMU, variabel *input* dan variabel *output*. Setelah memilih aksi analisa, hasil analisa dari DMU akan ditampilkan. Pada halaman ini admin juga dapat melihat *linear programming* dari masing-masing DMU yg telah di analisa. *Linear programming* ini yang akan memudahkan peneliti untuk membuktikan hasil dengan menggunakan program pendukung lain yaitu LINDO.



Gambar 4. 16 Halaman Analisa

7. Halaman Hasil Analisa

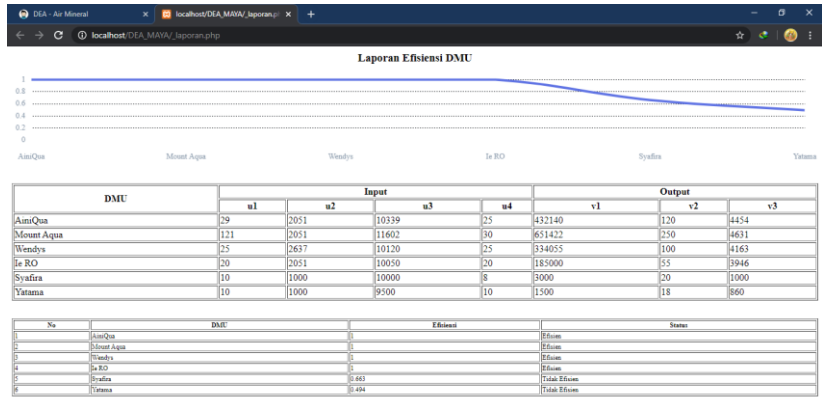
Halaman hasil analisa menampilkan data hasil analisa yang telah dianalisis pada halaman analisa sebelumnya. Data hasil analisa yang ditampilkan antara lain DMU, nilai efisiensi dan juga status atau keterangan dari nilai efisiensi tersebut. Disini akan terlihat apakah suatu DMU itu efisien atau tidak efisien. Halaman ini hanya memperlihatkan data yang telah di uji atau analisis, halaman ini tidak dapat melakukan perubahan seperti edit atau hapus.



Gambar 4. 17 Halaman Hasil Analisa

8. Halaman Cetak Laporan

Halaman cetak laporan menampilkan info dari data akhir dari analisis efisiensi DEA air mineral. Laporan yang dicetak menampilkan grafik trend efisiensi DMU, nilai dari masing-masing variabel *input* maupun *output* dan juga status dari nilai-nilai tersebut. Hal ini bertujuan untuk membantu admin dalam mengelola laporan data analisis DEA air mineral. Laporan yang dicetak atau di *save* tersimpan dalam format PDF.



Gambar 4. 18 Halaman Cetak Laporan

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil pengukuran efisiensi dengan menggunakan Model DEA CCR maka didapatkan hasil bahwa DMU yang dinilai Efisiensi Relatifnya adalah 1. Sedangkan DMU yang nilai relatifnya kurang dari 1 di kategorikan tidak efisien. Perolehan hasil komputasi dari Model DEA CCR pada ke empat DMU adalah 1, hal itu merupakan perolehan dari pembagian variabel *output* dengan variabel *input*, maka diperoleh hasil dari ke empat DMU adalah efisien.

Aplikasi yang dibangun untuk mengukur efisiensi Produksi Air Mineral di wilayah Aceh Utara dan Kota Lhokseumawe adalah aplikasi berbasis *web*, dengan menggunakan bahasa pemrograman PHP. Aplikasi ini diharapkan dapat membantu untuk memudahkan perusahaan atau industri produksi dalam mengukur tingkat efisiensi produksi Air Mineral.

B. Saran

Adapun beberapa saran dari penulis yang mungkin berguna yang ingin penulis sampaikan ialah sebagai berikut :

1. Peneliti mengharapkan aplikasi ini dapat berguna bagi pihak perusahaan dalam membantu untuk mengetahui bagaimana agar suatu produksi dapat dihasilkan dengan lebih efisien.
2. Di masa mendatang, diharapkan untuk pengembangan lebih lanjut pada aplikasi ini dapat dilakukan dengan menggunakan metode-metode yang lain untuk mendapatkan perbandingan metode mana yang lebih baik.

DAFTAR PUSTAKA

- Abdullah, Dahlan. *Data Envelopment Analysis Dengan Menggunakan Upper Bound Pada Output Untuk Mengukur Efisiensi Kinerja Perguruan Tinggi*. Medan: Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, 2018.
- Anhar. *Panduan Menguasai PHP & MySQL Secara Otodidak*. Jakarta Selatan: Mediakita, 2010.
- Aumora, Nova Sri, Djaimi Bakce, and Novia Dewi. "Analisis Efisiensi Produksi Usahatani Kelapa di Kecamatan Pulau Burung Kabupaten Indragiri Hilir." *urnal SOROT*, 2016: Volume 11, 47-59.
- Bunafit, Nugroho. *Database Relasional Dengan MySQL*. Yogyakarta: Andi, 2005.
- Dadan, and Kerendi Developers. *Membuat CMS Multifitur*. Jakarta: PT Elex Media Komputindo, 2015.
- Destiningrum, Mara, and Qadhli Jafar Adrian. "Sistem Informasi Penjadwalan Dokter Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)." *Jurnal TEKNOINFO*, 2017: Vol 11, No. 2.
- EMS, Tim. *PHP 5 dari Nol*. Jakarta: PT Elex edia Komputindo, 2016.
- Enterprise, Jubilee. *Pemograman Database dengan Python dan MySQL*. Jakarta: PT Elex Media Komputindo, 2018.
- Filardo, Andrew, Nugroho Priyo Negoro, and Aang Kunaifi. "Penerapan Data Envelopment Analysis dalam Pengukuran Efisiensi Retailer Produk Kendaraan Merek Toyota." *JURNAL SAINS DAN SENI ITS*, 2017: Vol. 6, No. 1, 73-77.
- Hardian, Banu, et al. *Diagram Sequence UML*. Depok, 2014.
- Henderi. *Analysis and Design System with Unified Modeling Language (UML)*. Tangerang: STIMIK Raharja, 2007.

- Henderi, et al. *Unified Modelling Language*. Tangerang: Raharja Enrichment, 2008.
- Hendini, Ade. "Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)." *Jurnal Khatulistiwa Informatika*, 2016: Vol IV, No. 2, 107-116.
- Hendini, Ade. "Pemodelan UML Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak)." *Jurnal Khatulistiwa Informatika*, 2016: Vol. IV, No. 2.
- Hudiyanti, Chintia Vairra, Fitra A Bachtiar, and Budi Darma Setiawan. "Perbandingan Double Moving Average dan Double Exponential Smoothing untuk Peramalan Jumlah Kedatangan Wisatawan Mancanegara di Bandara Ngurah Rai." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* (Andi), 2019: 2667- 2672.
- HUTAHAEAN, Jeperson. *Konsep Sistem Informasi*. Yogyakarta: CV BUDI UTAMA, 2014.
- Israwan, LM. Fajar, and dkk. "Implementasi Model CCR Data Envelopment Analysis (DEA) Pada Pengukuran Efisiensi Keuangan Daerah." *Jurnal Sistem Informasi Bisnis*, 2016: 76-83.
- Iswandy, Eka. "Sistem Penunjang Keputusan Untuk Menentukan Penerimaan Dana Santunan Sosial Anak Nagari dan Penyalurannya bagi Mahasiswa dan pelajar kurang mampu di Kenagarian Barung." *TEKNOIF*, 2015: 3.
- Jeffery L. Whitten, L. D. *Metode Desain & Analisis Sistem*. Yogyakarta: Andi, 2004.
- Kanisius. *22 Panduan Perjalanan Aman & nyaman*. Yogyakarta: Kanisiun (Anggota IKAPI), 2009.
- Manalu, Effrida, Fricles Ariwisanto Sianturi, and Mamed Rofendy Manalu. "Penerapan Algoritma Naive Bayes untuk Memprediksi Jumlah Produksi Barang Berdasarkan Data Persediaan dan Jumlah Pemesanan Pada CV.

- PADADAN MAMA PATRIES."* *Jurnal Manajemen Dan Informatika Pelita Nusantara*, Desember 2017: Vol 1 No 2.
- Miftakhul, Huda, and Bunafit. *Membuat Aplikasi Database dengan Java, MySQL dan Neatbeans*. 2008.
- Permatasari, Maya Fauziah. *Analisis Efisiensi Kinerja Pada UMKM Klaster Alat Rumah Tangga di Kabupaten Sragen dengan Metode Data Envelopment Analysis (DEA)*. Surakarta: Universitas Muhammadiyah Surakarta, 2018.
- Rerung, Rintho Rante. *Pemograman Web Dasar*. Yogyakarta: CV BUDI UTAMA, 2018.
- Rosa, A S, and M Shalahuddin. *Rekayasa Perangkat Lunak*. Bandung: Informatika, 2014.
- Sardjono, Sigit. *Ekonomi Mikro-Teori dan Aplikasi*. Yogyakarta: ANDI (Anggota KAPI), 2017.
- Shafique, M. N., Ahmad, H., & Adil, M. Y. "A Comparative Study Of The Efficiency Of Takaful And Conventional Insurance In Pakistan." *International Journal Of Accounting Research*, 2015: Vol. 2 No. 5.
- Toledo, Mata, and K Pauline. *Schaum's Outline Dasar-dasar Database Relational*. Jakarta: Erlangga, 2007.
- Tuffahati, Hulwah, Sepky Mardian, and Edy Supranto. "Pengukuran Efisiensi Asuransi Syariah Dengan Data Envelopment Analysis (DEA)." *Jurnal Akuntansi dan Keuangan Islam*, 2016: Vol. 4, No. 1.
- Wiharno, Sigit, and and Sujana Ismaya. *Kamus Besar Ekonomi*. Bandung: CV Pustaka Grafika, 2013.
- Yanto, Robi;. *Manajemen Basis Data Menggunakan MySQL*. Yogyakarta: CV BUDI UTAMA, 2016.



PENGUKURAN EFISIENSI PRODUKSI AIR MINERAL

DENGAN MENGGUNAKAN METODE
DATA ENVELOPMENT ANALYSIS

Bisnis Air Minum Dalam Kemasan (AMDK) merupakan salah satu bisnis yang cukup menjanjikan pada saat ini, yang mana kebutuhan terhadap air minum dan pola hidup instan menjadi alternatif bagi konsumen untuk mencari Air Minum Dalam Kemasan (AMDK) dan menembus pasar sasaran. Peningkatan efisiensi di bagian produksi merupakan salah satu cara yang bisa dilakukan oleh pihak perusahaan agar selalu dapat memenuhi permintaan konsumen. Guna menghadapi banyaknya para pesaing perusahaan yang menghasilkan produk yang sama, perusahaan perlu melakukan suatu cara untuk menjalankan proses produksi yang efisien, yaitu bagaimana menggunakan *input* sehemat mungkin untuk menghasilkan *output* yang sesuai atau melebihi target permintaan yang telah ditetapkan. Efisiensi Produksi dihitung menggunakan model CCR *Data Envelopment Analysis* (DEA) dengan empat variabel *input* dan 3 variabel *output*. CCR DEA menggunakan *Linear Programming* untuk mengevaluasi efisiensi relatif Unit Pengambil Keputusan atau *Decision Making Unit* (DMU). Total sampel yang digunakan adalah 4 DMU. Penelitian ini akan menghasilkan nilai efisiensi produksi Air Mineral di Aceh Utara dan Lhokseumawe. Tujuan dari penelitian ini adalah untuk membantu perusahaan produksi Air Minum dalam Kemasan (AMDK) agar dapat lebih optimal dan efisien.



SEFA BUMI PERSADA
Jl. Malikussaleh No. 3 Bayu - Aceh Utara
email: sefabumipersada@gmail.com
Telp. 085260363550

ISBN 978-623-7648-43-7

