

No Dok. 6633

D3
658.787.028.5
Hid
R

RANCANG BANGUN SISTEM INFORMASI
PENERIMAAN DAN PENGELUARAN BAHAN BAKU
PADA DEPARTEMEN *WAREHOUSE* DI PT
TEKNIKATAMA KARYA MANDIRI DENGAN
MENGGUNAKAN *CODEIGNITER* DAN *MARIADB*

TUGAS AKHIR

Diajukan Untuk Memenuhi Sebagian Syarat Penyelesaian Program Sarjana
Terapan Program Studi Sistem Informasi Industri Otomotif Pada Politeknik
STMI Jakarta

OLEH
SARI MAHUDA HIDAYANTI
1315088



| DATA BUKU PERPUSTAKAAN | |
|------------------------|-------------------|
| Tgl Terima | 01/08/22 |
| No Induk Buku | 530/SLIO/BB/TA/22 |

POLITEKNIK STMI
KEMENTERIAN PERINDUSTRIAN RI
JAKARTA
2019

SUMBANGAN ALUMNI

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

JUDUL TUGAS AKHIR:
**RANCANG BANGUN SISTEM INFORMASI PENERIMAAN DAN
PENGELUARAN BAHAN BAKU PADA BAGIAN *WAREHOUSE* DI PT
TEKNIKATAMA KARYA MANDIRI DENGAN MENGGUNAKAN
*CODEIGNITER DAN MARIA DB***

Disusun Oleh :

Nama : Sari Mahuda Hidayanti
Nim : 1315088
Program Studi : Sistem Informasi Industri Otomotif

Telah Diuji Oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta Kementerian Perindustrian R.I pada hari Senin tanggal 20 September 2019.

Jakarta, 21 September 2019

Dosen Pembimbing

Fifi Lailasari Hadiahastuti, S.Kom, M.Kes
NIP: 19730162005022001

Ketua Penguji

Lucky Heriyanto, S.T, M.T.I
NIP: 197908202009011009

Dosen Penguji

Dr. Ridzky Kramanandita, S.Kom, M.T
NIP: 197403022002121001

Dosen Penguji

Ulil Hamida, ST,MT
NIP: 198103272005022001

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI**

TANDA PERSETUJUAN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR:

**RANCANG BANGUN SISTEM INFORMASI PENERIMAAN DAN
PENGELUARAN BAHAN BAKU PADA BAGIAN *WAREHOUSE* DI PT
TEKNIKATAMA KARYA MANDIRI MENGGUNAKAN CODEIGNITER
DAN MARIADB**

Disusun Oleh:

Nama : Sari Mahuda Hidayanti
Nim : 1315088
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 16 Agustus 2019
Tanggal Sidang : 20 September 2019
Tanggal Lulus : 20 September 2019

Telah diperiksa dan disetujui untuk diajukan dan dipertahankan dalam
Ujian Tugas Akhir Politeknik STMI Jakarta

Jakarta, 20 September 2019

Dosen Pembimbing,


Fifi Lailasari Hadianastuti, S.Kom, M.Kes
NIP : 197310162005022001



LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Sari Mahuda Hidayanti
NIM : 1315088
Judul Tugas Akhir : Rancang Bangun Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku Pada Departemen Warehouse di PT Teknikatama Karya Mandiri Menggunakan CodeIgniter Dan MariaDB
Pembimbing : Fifi LilaSari Hadianastuti, S.Kom, M.Kes

| Tanggal | BAB | Keterangan | Paraf |
|----------------|------------|------------|-------|
| 29 Mei 2019 | I, II, IV | Revisi | |
| 31 Mei 2019 | I, II, III | Revisi | |
| 26 Juni 2019 | I, II, III | Revisi | |
| 4 Juli 2019 | I - IV | Revisi | |
| 23 Juli 2019 | I - IV | Revisi | |
| 7 Agustus 2019 | I - V | Revisi | |
| 9 AGUSTUS 2019 | V | Pengecekan | |

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri
Otomotif

Noveriza Yuliasari, S.T, M.T
NIP.197811212009012003

Dosen Pembimbing

Fifi LilaSari Hadianastuti, S.Kom, M.Kes
NIP. 197310162005022001

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : SARI MAHUDA HIDAYANTI

NIM : 1315088

Berstatus mahasiswa Program Studi Sistem Informasi Industri Otomotif di Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul:

"Rancang Bangun Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku Pada Departemen Warehouse di PT Teknikatama Karya Mandiri Menggunakan Codeigniter dan MariaDB"

- Dibuat dan selesaikan sendiri, dengan menggunakan literatur hasil kuliah, survey lapangan, dibantu oleh dosen pembimbing maupun asisten pembimbing, serta buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya tugas akhir ini.
- Bukan merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- Bukan merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini.

Jika terbukti saya tidak memenuhi apa yang telah saya nyatakan di atas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai dengan peraturan yang berlaku.

Jakarta, 7 Agustus 2019



Sari Mahuda H

ABSTRAK

PT Teknikatama Karya Mandiri merupakan perusahaan manufaktur yang bergerak di bidang industri komponen otomotif khususnya kepada komponen *spare part* kendaraan roda empat. Dalam kegiatan penerimaan dan pengeluaran bahan baku di PT Teknikatama Karya Mandiri ini, ditemukan suatu permasalahan dalam sistem penerimaan dan pengeluaran bahan baku yang sedang berjalan. Seperti pada pencatatan aktivitas keluar/masuknya bahan baku pada daftar stok sering ditemukannya daftar stok yang tidak lengkap ataupun mengalami kerusakan. Dalam kegiatan permintaan produksi tidak ada dokumen khusus dan transaksi dilakukan secara lisan. Laporan bulanan yang dibuat masih menggunakan *Microsoft excel*. Untuk mengatasi permasalahan tersebut maka perlu dibuat sebuah sistem informasi penerimaan dan pengeluaran bahan baku pada Departemen *Warehouse* yang dapat memberikan informasi yang akurat sehingga stok bahan baku dapat terkontrol dengan baik. Membuat suatu sistem informasi dan basis data (*database*) untuk aktivitas keluar/masuknya bahan baku yang mana data tersebut dapat membantu Operator *Warehouse* untuk membuat sebuah laporan dengan cepat. Metodologi pengembangan sistem yang digunakan pada penelitian ini adalah *waterfall*. Adapun pemodelan sistem menggunakan diagram UML antara lain *usecase diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *deployment diagram*. Pemodelan data yang digunakan yaitu *diagram ERD* dan kamus data. Sistem informasi ini dibangun menggunakan *CodeIgniter* 3.1.10 dan *MariaDB* 10.1.38. Sistem ini diharapkan dapat membantu proses pencatatan bahan baku dan sebagai tempat penyimpanan data untuk mencegah terjadinya kehilangan atau kerusakan data.

Kata kunci: Sistem informasi, Penerimaan dan Pengeluaran Bahan Baku, *Waterfall*, *Unified Modeling Language*, *CodeIgniter*, *MariaDB*, Bahan Baku.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dewasa ini berkembang sangat cepat seiring dengan berkembangnya sistem informasi yang terkomputerisasi. Dewasa ini teknologi informasi telah menjadi salah satu bagian yang berperan penting dalam berjalannya suatu sistem informasi di suatu instansi atau perusahaan yang bergerak dalam berbagai bidang, termasuk dalam dunia industri otomotif. Dengan peranan teknologi komputer sebagai salah satu unsur penunjang berjalannya sistem informasi, kebutuhan akan informasi yang akurat dan relevan menjadi satu hal yang penting untuk membantu manajemen dalam suatu perusahaan. Tidak lepas dari peranan teknologi informasi, beberapa fungsi manajemen memerlukan peranan teknologi informasi, salah satunya adalah manajemen pergudangan dimana salah satu aktivitasnya adalah proses penerimaan dan pengeluaran bahan baku.

Proses pengeluaran bahan baku menjadi sangat penting karena mendukung kelancaran proses produksi dan mempengaruhi persediaan bahan baku. Rancangan sistem yang baik dalam proses pengeluaran pada bahan baku akan membuat proses lebih efektif dan efisien serta bisa didapatkan laporan yang *up to date*. (Heryanto, Budiman, & Nugroho, 2014)

Bahan baku adalah bahan yang digunakan dalam membuat produk di mana bahan tersebut secara menyeluruh tampak pada produk jadinya. Bagi suatu perusahaan, persediaan dan kualitas bahan baku merupakan suatu hal yang penting. Demi menjaga kualitas suatu produk mulai dari bahan baku hingga barang jadi, maka peranan gudang tidak dapat dipisahkan dari aktivitas tersebut. Gudang dapat digambarkan sebagai suatu sistem logistik dari perusahaan, mulai dari bahan baku dikirim oleh pemasok, hingga memasuki proses produksi dan dikirim ke *customer*. Suatu sistem informasi dapat berperan baik untuk Departemen *Warehouse* dalam mengelola penerimaan dan pengeluaran bahan baku yang ada dalam gudang.

PT Teknikatama Karya Mandiri adalah salah satu perusahaan menengah di bidang industri otomotif. Perusahaan ini disebut sebagai perusahaan menengah berdasarkan Undang- Undang Nomor 20 Tahun 2008 tentang Usaha Mikro, Kecil, dan Menengah (UMKM), dengan beberapa cirinya adalah memiliki karyawan 20-99 orang, dan memiliki pembagian tugas yang jelas antara lain, bagian keuangan, bagian pemasaran dan bagian produksi.

Perusahaan ini memiliki beberapa Departemen. Salah satunya yaitu Departemen *Warehouse*. Dalam menjalankan tugasnya, Departemen *Warehouse* sering kali menemukan kendala atau permasalahan dalam proses penerimaan dan pengeluaran bahan baku. Misalnya, stok bahan baku yang tidak terupdate, dikarenakan laporan bahan baku masuk dan keluar yang kurang terkontrol, hal ini juga menghambat Operator *Warehouse* untuk membuat laporan bulanan. Adapun kendala lain adalah pencatatan bahan baku yang masuk dan keluar masih menggunakan form yang diisi secara manual sehingga menyulitkan pengolahan data bahan baku, selain itu Departemen *Warehouse* masih menggunakan pengarsipan secara manual yaitu dengan disimpan dalam map yang ditumpuk sehingga kesulitan dalam hal pencarian data dan rentan rusak.

Untuk mengatasi masalah tersebut, sistem yang ada akan diperbaiki dengan merancang bangun suatu sistem informasi yang terkomputerisasi untuk membantu Operator *Warehouse* dalam proses penerimaan dan pengeluaran bahan baku, sehingga judul Tugas Akhir ini adalah “Rancang Bangun Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku pada Departemen *Warehouse* di PT Teknikatama Karya Mandiri Menggunakan *CodeIgniter* dan *MariaDB*”.

1.2 Permasalahan

Permasalahan yang terjadi pada Departemen *Warehouse* dalam penerimaan dan pengeluaran bahan baku pada PT Teknikatama Karya Mandiri adalah sebagai berikut:

1. Tidak terupdatenya catatan stok bahan baku sehingga mengganggu proses produksi.

2. Belum adanya *database* untuk penyimpanan dokumen yang mengakibatkan terjadinya kerusakan dokumen, *redundancy* data, atau kehilangan dokumen, dan sulitnya pencarian dokumen.
3. Pembuatan laporan bulanan masih menggunakan *Microsoft excel*. Hal tersebut membutuhkan waktu yang cukup lama.
4. Transaksi permintaan pembelian bahan baku dilakukan secara lisan oleh Departemen *Warehouse* kepada Departemen *Purchasing* dan tidak ada dokumen.

1.3 Tujuan Penelitian

Adapun tujuan pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Membuat suatu sistem informasi untuk memudahkan proses *update* stok sehingga aktivitas keluar dan masuknya bahan baku dapat terkontrol dengan baik.
2. Merancang *database* sistem informasi proses penerimaan dan pengeluaran bahan baku agar tidak kesulitan dalam pencarian data dan mencegah kerusakan dokumen, *redundancy* data, atau kehilangan dokumen.
3. Membuat laporan bulanan dengan tepat waktu.
4. Membuat suatu sistem dimana transaksi permintaan pembelian bahan baku dapat disimpan dan dapat dilihat sebagai bukti transaksi.

1.4 Batasan Masalah

Agar dalam penulisan Tugas Akhir ini lebih fokus dan terarah, maka perlu diadakan batasan-batasan sebagai berikut:

1. Penelitian dilakukan selama satu bulan bertempat di PT Teknikatama Karya Mandiri Jalan Raya Bahkilong No.88 Cikarang Selatan – Bekasi 17330.
2. Jangka waktu untuk penelitian ini, mulai dari 16 Juli 2018 sampai dengan 16 Agustus 2018.
3. Ruang lingkup yang dianalisis adalah menangani proses penerimaan dan pengeluaran bahan baku.

1.5 Manfaat Tugas Akhir

Adapun manfaat yang dapat diambil dari pembuatan Tugas Akhir ini adalah sebagai berikut:

Hasil penelitian ini agar dapat diimplementasikan di perusahaan untuk membantu kinerja perusahaan dalam melakukan pemantauan data bahan baku yang tersedia.

1.6 Sistematika Penulisan

Laporan ini disusun berdasarkan hal-hal yang berhubungan erat dengan hasil pengamatan sehingga dapat memperoleh gambaran yang lebih jelas mengenai isi laporan dengan praktik kerja lapangan yang dilaksanakan. Adapun tahapan-tahapan dalam laporan ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat latar belakang, permasalahan, tujuan penelitian, batasan masalah, manfaat tugas akhir, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang berbagai teori yang diperoleh dari buku-buku ataupun berbagai macam referensi yang berkaitan dengan tema yang diambil. Maupun analisis dokumen jurnal yang membahas tentang penerimaan dan pengeluaran bahan baku. Teori-teori yang dipaparkan pada laporan ini adalah seputar sistem informasi, penerimaan dan pengeluaran bahan baku dan metodologi pengembangan sistem.

BAB III METODOLOGI PENELITIAN

Pada bab ini membahas tentang jenis dan sumber data, metode pengumpulan data, metode pengembangan sistem yang digunakan, serta kerangka penelitian untuk membantu dalam pemecahan masalah.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini akan membahas mengenai data-data yang telah diperoleh berdasarkan hasil pengamatan selama melaksanakan penelitian di PT Teknikatama Karya Mandiri.

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis dari pengolahan data dengan pemodelan sistem menggunakan UML, pemodelan data menggunakan ERD, serta membuat perancangan antarmuka, sampai *hardware* dan *software* yang diperlukan.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari hasil penelitian yang dilakukan, serta mengemukakan saran-saran dalam penerapan sistem informasi penerimaan dan pengeluaran bahan baku untuk perusahaan dalam penelitian dan pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Berdasarkan Djahir dan Pratita (2014), sistem adalah kumpulan dari komponen-komponen yang saling berkaitan yang bekerja bersama untuk mencapai beberapa tujuan. Sistem apapun dapat mempunyai subsistem. Subsistem adalah sistem yang lebih kecil dari sebuah sistem. Maka, subsistem dapat menjadi satu cara untuk berpikir tentang komponen dari sebuah sistem. Pemahaman sistem sebagai gabungan antara subsistem sangat berguna untuk para analis karena memampukan para analis untuk memfokuskan diri pada suatu area tertentu. Supersistem adalah sistem yang lebih besar dan sangat kompleks. Supersistem mengacu kepada sistem apapun yang memiliki sistem-sistem yang lebih kecil.

Berdasarkan Rukun dan Hayadi (2018), kata “sistem” mengandung arti kumpulan dari komponen-komponen yang memiliki unsur keterkaitan antara satu dan lainnya.

Berdasarkan Anggraeni dan Irviani (2017), sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan.

Sementara menurut Rusdiana dan Irfan (2014), pendekatan sistem berusaha menjelaskan sesuatu berdasarkan sudut pandang tujuan, proses, dan struktur. Dengan memahami struktur, sistem dan proses sistem, seseorang dapat menjelaskan tujuan suatu sistem tidak tercapai. Pendekatan sistem merupakan jumlah keseluruhan dari bagian-bagian yang saling bekerja sama untuk mencapai hasil yang diharapkan berdasarkan kebutuhan tertentu.

Kata “sistem” berasal dari bahasa yunani yaitu, *systema*, yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, bisa diartikan sekelompok elemen yang

independen, namun saling berkaitan sebagai satu kesatuan. Berikut adalah definisi dari sistem menurut para ahli :

1. Menurut Ludwig (1991), sistem adalah seperangkat unsur yang saling berhubungan dan saling memengaruhi dalam satu lingkungan tertentu.
2. Menurut Davis (1995), sistem merupakan bagian-bagian yang beroperasi secara bersama-sama untuk mencapai beberapa tujuan.
3. Menurut Mcleod (2001), sistem adalah sekelompok elemen yang terintegrasi untuk mencapai suatu tujuan.
4. Menurut Jogiyanto (2005), sistem adalah kumpulan dari elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian dan kesatuan yang nyata, seperti tempat, benda, serta orang-orang yang ada dan terjadi.

2.1.1 Karakteristik Sistem

Menurut Ladjmudin (2005), Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah dan sasaran atau tujuan.

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai karakteristik dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung merupakan media yang menghubungkan antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini kemungkinan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintergrasi dengan subsitem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal *maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat berjalan. Sinyal input adalah energi yang diproses untuk mendapatkan keluaran dari sistem.

6. Keluaran Sistem

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila

mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

2.1.2 Klasifikasi Sistem

Menurut Setyorini (2016), suatu sistem dapat diklasifikasikan dari beberapa sudut pandang, yaitu:

1. Sistem Abstrak (*Abstract System*),
Sistem yang tidak tampak secara fisik, karena hanya berupa pemikiran atau ide-ide. Contoh, sistem Teologia yang merupakan suatu sistem yang menggambarkan hubungan manusia dengan Tuhan.
2. Sistem Fisik (*Physical System*)
Sistem yang tampak secara fisik. Contoh, sistem komputer, sistem produksi, sistem pendidikan, dll.
3. Sistem Alamiah (*Natural System*)
Sistem yang tediri dari proses-proses alam. Contoh sistem Geologi.
4. Sistem buatan Manusia (*Human made System*)
Sistem yang dirancang atau didesain oleh manusia. Contoh sistem informasi.
5. Sistem Deterministik (*Deterministic System*)
Sistem yang beroperasi dengan tingkah laku yang dapat diramalkan. Interaksi antar elemen-elemen dapat dideteksi, sehingga *outputnya* juga dapat diramalkan. Contoh sistem komputer.
6. Sistem Probabilitas (*Probabilistic System*)
Sistem yang tidak bisa diramalkan. Contoh sistem manusia.
7. Sistem Tertutup (*Closed System*),
Sistem yang tidak berhubungan dengan lingkungan luarnya.
8. Sistem Terbuka (*Open System*)
Sistem yang berhubungan atau dipengaruhi oleh lingkungan luarnya.

2.2 Konsep Dasar Informasi

Berikut ini adalah pendapat-pendapat yang dikemukakan para ahli mengenai definisi informasi yang dikutip dalam Muslihudin dan Oktafianto (2016):

- a. Jogyianto (2008), informasi adalah data yang diolah menjadi bentuk yang berguna bagi para pemakainya.
- b. Sutarman (2008), informasi merupakan hasil pengolahan data, sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang.
- c. Krismiaji dalam Zabaer (2005), informasi adalah data yang telah diorganisasikan, dan telah memiliki kegunaan dan manfaat.
- d. Mc. Leod, informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Menurut Muslihudin dan Oktafianto (2016), dapat disimpulkan bahwa informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan metode ataupun cara-cara tertentu.

Sedangkan berdasarkan Hutahaean (2015), informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah kejadian yang terjadi pada saat tertentu

Berdasarkan Anggraeni dan Irviani (2017), informasi adalah sekumpulan data atau fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima. Data yang diolah menjadi sesuatu yang berguna bagi si penerima. Maksudnya, yaitu dapat memberikan keterangan dan pengetahuan.

2.2.1 Kualitas Informasi

Berdasarkan Tyoso (2016), informasi yang berkualitas harus memenuhi beberapa persyaratan sebagai berikut:

a. Ketersediaan (*Availability*)

Informasi harus dapat diakses oleh orang yang membutuhkannya, maka dari itu informasi harus tersedia setiap saat pada “gudang data” yang terorganisasi rapi.

b. Mudah dipahami (*Comprehensibility*)

Informasi yang berbelit-belit atau tidak jelas koneksinya bahkan bersifat rumit, maka berakibat keputusan yang akan diambil tertunda, karena lebih banyak waktu yang digunakan untuk membahasnya.

c. Akurat (*Accuracy*)

Informasi harus bersih dari kesalahan dan kekeliruan. Artinya, informasi harus jelas dan tepat dalam mencerminkan makna yang terkandung dari data.

d. Relevan (*Relevant*)

Berkaitan dengan pengoperasian suatu organisasi, informasi yang dibutuhkan ialah informasi yang benar-benar relevan dengan permasalahan, misi dan tujuan organisasi yang bersangkutan.

e. Bermanfaat (*Benefits*)

Informasi sebaiknya dapat disajikan dalam bentuk-bentuk yang mudah dilihat dan dipelajari sehingga kepemanfaatannya terlihat jelas.

f. Tepat Waktu (*Being On/In Time*)

Informasi harus tersedia tepat pada waktunya sehingga saat organisasi membutuhkannya informasi sudah tersedia. Juga harus diperhatikan kapan informasi itu diperoleh pada peristiwa apa saat itu.

g. Keterandalan (*Reliability*)

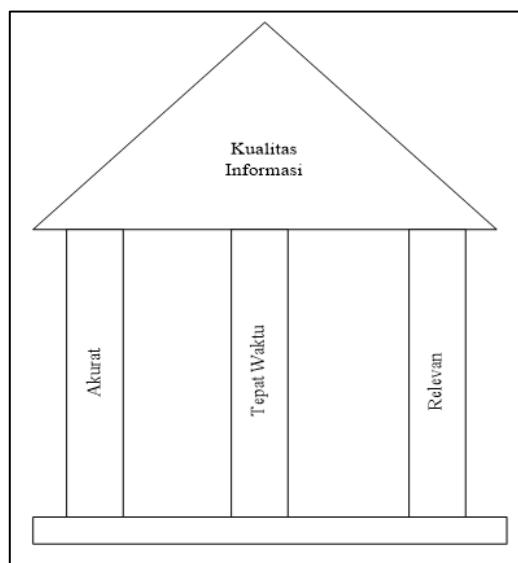
Informasi harus diperoleh dari sumber daya yang jelas dan dapat dipertanggungjawabkan kebenarannya. Demikian juga dengan pengelola atau pemberi informasi juga merupakan pihak-pihak yang dapat dipercaya.

h. Konsisten (*Consistent*)

Informasi tidak bermuatan hal-hal yang kontradiktif, sehingga bahasa yang digunakan harus sesuai dengan yang disajikan.

Sementara berdasarkan Anggareni dan Irviani (2017), berpendapat bahwa kualitas informasi dapat dianalogikan sebagai pilar-pilar dalam bangunan dan menentukan baik tidaknya pengambil keputusan. Berikut adalah pilar-pilar kualitas informasi dapat dilihat pada Gambar II.1 Kualitas informasi sering kali diukur berdasarkan:

- a. Relevansi
- b. Ketepatan waktu
- c. Keakurasian

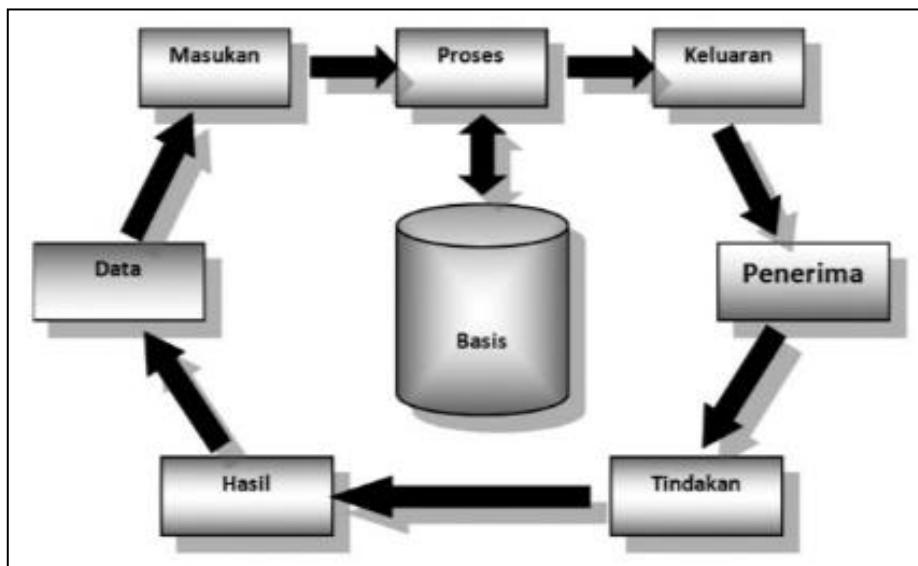


Gambar II.1 Pilar Kualitas Informasi

(Sumber: Anggraeni dan Irviani, 2017)

2.2.2 Siklus Informasi

Berdasarkan Anggraeni dan Irviana (2017), siklus informasi menggambarkan pengolahan data menjadi informasi dan pemakaian informasi untuk mengambil keputusan, hingga akhirnya dari tindakan hasil pengambilan keputusan tersebut dihasilkan data kembali. Berikut adalah gambar dari siklus informasi dalam Anggraeni dan Irviana (2017) dapat dilihat pada Gambar II.2.



Gambar II.2 Siklus Informasi dalam Anggraeni dan Irviana

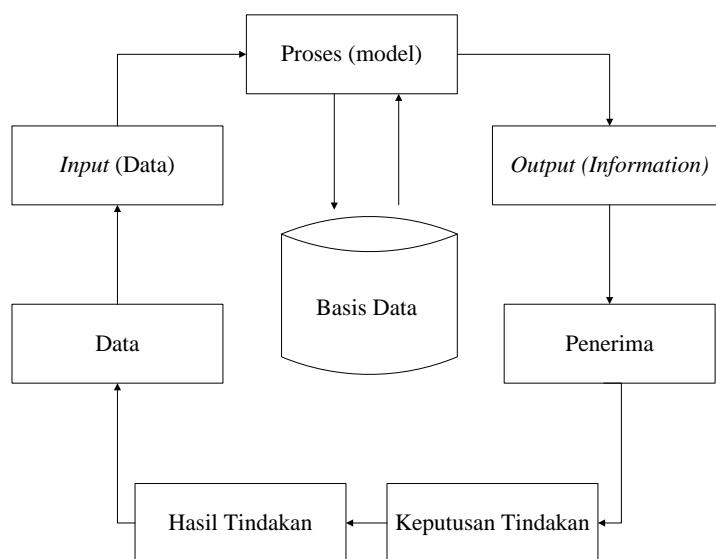
(Sumber: Anggareni dan Irviana, 2017)

Sementara berdasarkan Sutabri (2012), secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan lain sebagainya. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut

akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.

Sedangkan berdasarkan Hutahaean (2015), data yang diolah untuk menghasilkan informasi menggunakan model proses yang tertentu. Misalkan suhu dalam *Fahrenheit* diubah ke dalam *Celcius*. Dalam hal ini digunakan model matematik berupa rumus konversi dari derajat *Fahrenheit* menjadi satuan derajat *Celcius*. Data yang diolah melalui suatu model menjadi informasi, kemudian penerima menerima informasi tersebut, yang berarti menghasilkan keputusan dan melakukan tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya yang disebut dengan siklus informasi (*information cycle*). Siklus ini juga disebut dengan siklus pengolahan data (*data processing cycles*). Siklus informasi ini dapat digambarkan sebagai berikut pada Gambar II.3.



Gambar II.3 Siklus Informasi dalam Hutahaean

(Sumber: Hutahaean, 2015)

2.3 Konsep Dasar Sistem Informasi

Menurut Lucas (2017), sistem informasi adalah suatu kegiatan dari prosedur-prosedur yang diorganisasikan, bilamana dieksekusi, akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam

organisasi. Dan berikut ini adalah pendapat para ahli mengenai definisi dari sistem informasi yang dikutip dalam Rukun dan Hayadi (2018):

- a. Menurut Turban, McLean dan Wetherbe, sistem informasi adalah sebuah sistem informasi yang mempunyai fungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
- b. Menurut Bodnar dan Hopwood, sistem informasi adalah kumpulan perangkat keras dan lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna.
- c. Menurut Alter, sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah perusahaan.

2.3.1 Komponen Sistem Informasi

Berdasarkan Anggraeni dan Irviani (2017), komponen-komponen dari sistem informasi adalah sebagai berikut:

1. Komponen *input*, adalah data yang masuk ke dalam sistem informasi.
2. Komponen model, adalah kombinasi prosedur, logika dan model matematika yang memproses data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.
3. Komponen *output*, adalah hasil informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Komponen teknologi, adalah alat dalam sistem informasi, teknologi digunakan dalam menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan *output* dan memantau pengendalian sistem.
5. Komponen basis data, adalah kumpulan data yang saling terhubung yang tersimpan di dalam komputer dengan menggunakan *software database*.
6. Komponen kontrol, adalah komponen yang mengendalikan gangguan terhadap sistem informasi.

2.4 Konsep Dasar Penerimaan dan Pengeluaran

Sistem Penerimaan dan Pengeluaran bahan baku ditekankan pada proses penerimaan dan pengeluaran bahan baku, serta pembuatan laporan. Berikut akan dijabarkan lebih detail lagi mengenai ketiga proses tersebut.

1. Penerimaan bahan baku adalah kegiatan penerimaan bahan baku berdasarkan nota Surat Pesanan/Surat Perintah Kerja.
2. Pengeluaran bahan baku adalah kegiatan pengeluaran bahan baku sesuai dengan daftar permintaan bahan baku dari unit kerja atau departemen yang membutuhkan.
3. Pencatatan adalah kegiatan mencatat bahan baku yang diterima melalui buku penerimaan dan pengeluaran bahan baku.

2.4.1 Konsep Dasar Penerimaan

Pada Proses penerimaan bahan baku dapat di-*assign* tiga macam informasi *lot*, yaitu:

1. *Receiving/user lot*
Receiving/user lot adalah *lot* internal bahan baku.
2. *Vendor Lot*
Vendor lot adalah nomor lot internal *vendor/supplier* digunakan sebagai referensi *lot* sehingga jika terjadi masalah kualitas, masalah dapat dilacak sampai ke *supplier*.
3. *Purchase order lot*
Purchase order lot adalah nomor lot yang biasanya nilainya sama dengan *purchase order*.
Pada proses penerimaan bahan baku ini dapat diisikan informasi *lot number*, dan juga dapat diisikan informasi *serial number*.

2.4.2 Fungsi Penerimaan

Menurut Mulyadi (2008), Fungsi Penerimaan bertanggung jawab untuk melakukan pemeriksaan terhadap jenis, mutu, dan kuantitas barang yang diterima dari pemasok guna menentukan layak atau tidak barang tersebut diterima oleh

perusahaan. Fungsi ini juga bertanggung jawab untuk menerima barang dari pembeli yang berasal dari retur penjualan.

2.4.3 Proses Penerimaan Bahan Baku

Proses yang menjadi tahapan dalam proses penerimaan bahan baku masuk adalah:

1. Menerima informasi bahan baku masuk. Dalam tahapan ini, perusahaan terlebih dulu mendapatkan informasi bahan baku yang masuk. Informasi bisa didapatkan sebelumnya melalui *email*, telepon, sms dan media lainnya dengan jangka waktu tertentu sebelum bahan baku masuk. Selain itu informasi juga bisa didapatkan ketika bahan baku yang akan masuk sudah tiba di gudang. Aturan mengenai batasan waktu penerimaan informasi sangat tergantung dari aturan yang dibuat oleh masing-masing pengelola gudang.
2. Memastikan bahwa bahan baku yang akan masuk sudah siap diterima. Kesiapan ini berkaitan dengan sumber daya manusia, perlengkapan dan peralatan yang dibutuhkan untuk menangani bahan baku masuk dan juga lokasi penempatan bahan baku yang akan masuk.
3. Proses penerimaan bahan baku terjadi pada hari H, di mana bahan baku sudah tiba di gudang. Yang perlu dilakukan adalah menyiapkan *check list* bahan baku yang masuk. Jika sebelumnya sudah diinformasikan daftar bahan baku apa saja yang masuk dengan detail nama, kode, jumlah, ukuran volume, berat maka daftar bahan baku ini bisa dijadikan sebagai dokumen *check list* bahan baku. Pastikan bahwa ada petugas gudang yang bertanggung jawab untuk melakukan pengecekan bahan baku berdasarkan dokumen tersebut. Jika tidak ada informasi detail bahan baku yang akan masuk, maka bahan baku yang masuk dicatat sesuai detail yang diperlukan, misal seperti nama, kode, jumlah, volume dan berat. Detail informasi bahan baku yang perlu dilakukan pengecekan tergantung dari peraturan pengelolaannya. Semisal hanya perlu mencatatkan kode bahan baku dan jumlahnya saja, maka di petugas penerima bahan baku masuk di gudang

cukup melakukan pengecekan berdasarkan kode bahan baku dan jumlahnya saja. Yang perlu dipastikan selain detail informasi bahan baku yang masuk adalah kondisi bahan baku pada saat datang ke gudang. Jika diperlukan, sebaiknya bahan baku yang akan masuk tersebut difoto, baik pada saat bahan baku masih ada di atas kendaraan pengangkut maupun pada saat diturunkan.

2.4.4 Konsep Dasar Pengeluaran Barang

Pengeluaran barang dari gudang juga mempengaruhi mutasi persediaan pengeluaran barang dari gudang adalah atas permintaan dari bagian lain, seperti bagian produksi yang minta barang (bahan baku) untuk keperluan produksi. Bagian gudang melakukan pengeluaran barang dan mencatatnya pada kartu gudang.

2.4.5 Proses Pengeluaran Bahan Baku

Proses pengeluaran bahan baku secara garis besar adalah sebagai berikut:

1. Bagian produksi membuat bukti permintaan bahan baku sebanyak 3 lembar, dan harus disahkan oleh yang berwenang. Setiap bukti permintaan bahan baku berisi tentang:
 - a. Nomor pekerjaan atau bagian yang meminta bahan baku.
 - b. Jumlah dan keterangan bahan baku yang diperlukan.
 - c. Harga per unit dan harga total yang dikeluarkan. Bukti permintaan bahan baku (3 lembar diserahkan kepada bagian gudang).
2. Bagian gudang menyiapkan bahan baku sesuai dengan yang diminta dalam bukti permintaan untuk diserahkan kepada bagian produksi. Bahan baku yang diserahkan kepada bagian produksi, dicatat oleh bagian gudang pada tempat/lajur yang telah disediakan. Bukti permintaan bahan baku setelah disahkan oleh kepala bagian gudang, dibagikan sebagai berikut:
 - a. Lembar 1, kepada bagian akuntansi.
 - b. Lembar 2, sebagai arsip dibagian gudang.
 - c. Lembar 3, bersamaan dengan bahan diserahkan kepada bagian produksi.

2.4.6 Proses Pembuatan Laporan

1. Buku Penerimaan Bahan Baku

Buku penerimaan bahan baku merupakan buku yang terdiri dari lembaran-lembaran yang memuat informasi berkaitan dengan penerimaan bahan baku yang meliputi jenis, tanggal penerimaan, jumlah, nilai yang meliputi harga per satuan dan jumlah total, dan asal bahan baku. Setiap terjadi pemasukan bahan baku ke dalam gudang harus segera dilakukan pencatatan pemasukan ke dalam buku penerimaan, disamping harus pula melakukan pengisian pemasukan pada kartu persediaan (kartu stok) sehingga dapat diketahui jumlah persediaan bahan baku. Disamping itu, setiap terjadi pencatatan pemasukan bahan baku ke dalam buku penerimaan harus diikuti bukti-bukti penerimaan bahan baku (antara lain berupa nota, faktur, kwitansi, atau bukti yang lain, misalnya surat penyerahan bahan baku dari unit pembelian). Setiap bukti pemasukan bahan baku harus dibubuh nomor (sebagai nomor kode bukti masuk) sesuai urutan kronologis, yang kemudian nomor kode bukti masuk ini dituliskan pada kolom nomor kode bukti masuk dalam buku penerimaan bahan baku maupun kartu persediaan. penggunaan nomor kode bukti masuk ini dimasukkan untuk mempermudah pengecekan maupun pengawasan bahan baku.

2. Buku Pengeluaran Bahan Baku

Buku pengeluaran bahan baku merupakan buku yang terdiri atas lembaran-lembaran yang memuat informasi berkaitan dengan pengeluaran bahan baku yang meliputi jenis dan spesifikasi, tanggal pengeluaran, jumlah pengeluaran dan penerima. Setiap terjadi pengeluaran bahan baku dari gudang harus segera dilakukan pencatatan pengeluaran ke dalam buku pengeluaran bahan baku, disamping harus pula melakukan pengisian pengeluaran pada kartu stok sehingga nantinya dapat diketahui jumlah persediaan bahan baku. Buku pengeluaran harus diikuti bukti-bukti pengeluaran bahan baku yang dapat berupa surat atau bon gudang. Disamping itu setiap bukti pengeluaran bahan baku harus dihubungi nomor (sebagai nomor kode bukti keluar) sesuai urutan kronologis, yang kemudian nomor kode bukti keluar ini dituliskan pada kolom nomor kode bukti keluar dalam buku pengeluaran maupun kartu persediaan. Penggunaan nomor kode bukti keluar ini dimasukkan untuk mempermudah pengecekan maupun pengawasan bahan baku.

3. Kartu Persediaan/*Stock*.

Kartu persediaan merupakan formulir/lembaran untuk mencatat perubahan-perubahan jumlah persediaan bahan baku karena adanya pemasukan dan pengeluaran bahan baku. Adapun informasi yang harus tertuang dan tertulis dalam kartu persediaan bahan baku meliputi jenis dan spesifikasi, tanggal pemasukan/pengeluaran bahan baku, kode nomor surat bukti pemasukan/pengeluaran, asal/tujuan, jumlah pemasukan/pengeluaran, dan jumlah sisa (persediaan bahan baku). Dalam kegiatan pengelolaan administrasi pergudangan, kartu persediaan bahan baku dalam bentuk kartu ini dibuat rangkap dua, satu untuk arsip dan yang satu untuk kartu gantung (kartu yang digantungkan pada kelompok jenis bahan baku tertentu di mana bahan baku tersebut ditempatkan/disimpan sehingga hal ini akan mempermudah dalam pengecekan bahan baku, terutama pengecekan terhadap jumlah persediaan bahan baku).

2.5 Pengertian *Warehouse* atau Gudang

Berdasarkan Warman (2012), *Warehouse* atau gudang (kata benda) adalah bangunan yang dipergunakan untuk menyimpan barang dagangan. Penggudangan (kata kerja) ialah kegiatan menyimpan dalam gudang.

Sementara berdasarkan Wikipedia Indonesia (2017), Gudang adalah sebuah ruangan yang digunakan untuk menyimpan berbagai macam barang. Karena untuk menyimpan berbagai macam barang, biasanya gudang berpotensi untuk menyimpan debu. Karena itu, peletakan gudang perlu diperhatikan agar tidak menganggu aktivitas lain dalam bangunan tersebut. Gudang sebaiknya terletak di lokasi yang tidak lembap agar barang-barang yang disimpan tidak mudah rusak. Agar tidak menghabiskan lahan, gudang bisa diletakkan di bawah tangga atau di loteng. Gudang mencakup banyak jenis barang yang disimpan.

Menurut Sriyanto (2016) menerangkan bahwa gudang adalah suatu tempat atau bangunan yang digunakan untuk menimbun, menyimpan barang baik berupa bahan baku (*raw material*), barang setengah jadi (*work-in process*) ataupun barang jadi (*finished good*).

Berdasarkan Martono (2015) gudang adalah tempat penyimpanan sementara dan pengambilan *inventory* untuk mendukung kegiatan operasi bagi proses operasi berikutnya, ke lokasi distribusi, atau kepada konsumen akhir.

Berdasarkan Martono (2015), gudang memiliki keuntungan tersendiri. Keuntungan adanya gudang bagi *inventory* adalah:

1. Menyediakan tempat untuk meletakan dan melindungi.
2. Menyediakan tepat waktu sesuai pesanan (menjamin *service level*).
3. Memonitor status.
4. Sebagai alat komunikasi dengan konsumen.
5. Mengurangi biaya transportasi.

2.5.1 Alasan Diperlukannya Gudang

Menurut Trisna (2012), terdapat beberapa alasan gudang diperlukan dalam kegiatan logistik, yaitu:

1. *Transportation-Production & Cost Reduction*

Gudang mempunyai peranan penting di dalam proses pengendalian dan pengurangan biaya transportasi dan produksi. Gudang akan mempunyai peran di dalam menekan biaya transportasi maupun produksi.

2. *Coordination of Supply And Demand*

Gudang berfungsi salah satunya adalah sebagai unsur koordinasi antara penawaran dan permintaan, volume permintaan tidak selalu dapat diproyeksi secara relatif akurat sedangkan proses produksi harus tetap dilaksanakan.

3. *Production Needs*

Proses produksi untuk barang-barang dengan klasifikasi dan karakteristik tertentu membutuhkan gudang untuk menekan biaya produksi maupun untuk melengkapi proses akhir produksi barang. Untuk barang-barang dengan klasifikasi tertentu memerlukan waktu simpan agar memperoleh barang dengan kualitas lebih sempurna, sebagai contoh keju, anggur. Barang-barang lain yang memerlukan gudang untuk proses akhir

sesuai dengan klasifikasi dan karakteristiknya adalah barang yang harus dirakit (*assembling*) untuk penyempurnaan proses akhir produksi.

4. *Marketing Considerations*

Barang dengan tipe, klasifikasi dan karakteristik tertentu menuntut harus siap tersedia di pasar, agar pasokan barang tersebut di pasar tidak terhenti. Untuk itu diperlukan gudang yang relatif dekat dengan pasar sebagai media untuk mempercepat proses pendistribusian guna memenuhi kebutuhan pasar.

2.6 Pengertian Bahan Baku

Bahan baku merupakan bahan langsung (*direct material*) yaitu bahan yang membentuk suatu kesatuan yang tidak terpisahkan dari produk jadi. Bahan baku adalah bahan utama atau bahan pokok dan merupakan komponen utama dari suatu produk.

Bahan baku diperlukan oleh pabrik untuk diolah, yang setelah melalui beberapa proses diharapkan menjadi barang jadi (*finished goods*). Pengertian bahan baku menurut Sofjan (1999) adalah: "Semua bahan yang dipergunakan dalam perusahaan pabrik, kecuali terhadap bahan-bahan yang secara fisik akan digabungkan dengan produk yang dihasilkan oleh perusahaan pabrik tersebut". Jadi bahan baku merupakan bahan yang dipergunakan dalam perusahaan untuk menjadi bagian dari produk tertentu.

Persediaan bahan baku menurut Sofjan (1999) adalah persediaan dari barang-barang berwujud yang digunakan dalam proses produksi, barang mana dapat diperoleh dari sumber-sumber alam ataupun dibeli dari supplier atau perusahaan yang menghasilkan bahan baku bagi perusahaan pabrik yang menggunakananya.

Definisi lain dari bahan baku yang dikemukakan oleh beberapa ahli adalah sebagai berikut:

1. Menurut Mulyadi (2005), Bahan baku merupakan bahan yang memebentuk bagian menyeluruh.

2. Menurut Kholmi (2003), Bahan baku merupakan bahan yang membentuk bagian besar produk jadi, bahan baku yang diolah dalam perusahaan manufaktur dapat diperoleh dari pembelian lokal, impor atau hasil pengolahan sendiri.
3. Menurut Prawirosentono (2001), Bahan baku adalah bahan utama dari suatu produk atau barang.
4. Menurut Wibowo (2007), Bahan baku merupakan bahan yang utama didalam melakukan proses produksi sampai menjadi barang jadi. Bahan baku meliputi semua barang dan bahan yang dimiliki perusahaan dan digunakan untuk proses produksi .

2.6.1 Kebutuhan Bahan Baku

Menurut Ahyari (2013), Pada umumnya persediaan bahan baku yang diselenggarakan oleh suatu perusahaan akan dipergunakan untuk menunjang pelaksanaan proses produksi yang bersangkutan tersebut. Dengan demikian maka besarnya persediaan bahan baku tersebut akan disesuaikan dengan kebutuhan bahan baku tersebut untuk pelaksanaan proses produksi yang ada didalam perusahaan. Jadi untuk menentukan berapa banyak bahan baku yang akan dibeli oleh suatu perusahaan pada suatu periode akan banyak tergantung kepada berapa besarnya kebutuhan perusahaan tersebut akan masing-masing jenis bahan baku untuk keperluan proses produksi yang dilaksanakan dalam perusahaan yang bersangkutan.

2.6.2 Jenis-Jenis Bahan Baku

Menurut Yayat dan Komara (2013) jenis-jenis bahan baku yaitu:

1. Bahan baku langsung atau *direct material* adalah semua bahan baku yang merupakan bagian daripada barang jadi yang dihasilkan. Biaya yang di keluarkan untuk membeli bahan baku langsung ini mempunyai hubungan yang erat dan sebanding dengan jumlah barang jadi yang dihasilkan.

2. Bahan baku tidak langsung atau disebut juga dengan *indirect material*, adalah bahan baku yang ikut berperan dalam proses produksi tetapi tidak secara langsung tampak pada barang jadi yang dihasilkan.

2.7 System Development Life Cycle (SDLC)

Menurut Dennis *et al* (2015) *System Development Life Cycle* (SDLC) memiliki 4 langkah dasar atau fase untuk membentuk suatu sistem informasi. Fase tersebut yaitu *planning*, *analysis*, *design*, dan *implementation* atau dapat disebut dengan perencanaan, analisis, perancangan, dan penerapan. Perbedaan proyek dalam membangun sebuah sistem infromasi mungkin dapat berbeda-beda sesuai dengan caranya masing-masing. Akan tetapi, semua proyek pasti memiliki elemen dari 4 fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah yang mengandalkan pada teknik dan menghasilkan suatu hasil berupa dokumen yang spesifik atau file yang menjelaskan tentang proyek.

Tahapan SDLC diproses dalam suatu alur yang dimulai dari awal hingga akhir. Dalam suatu proyek, tim proyek memindahkan fase tersebut dengan teratur, secara bertahap, secara iteratif, atau dalam pola lainnya. Dalam SDLC, fase tersebut dijelaskan dalam beberapa teknik yang digunakan untuk menjelaskannya secara umum.

Ada dua hal penting untuk dipahami tentang SDLC:

- a. Pertama, yang harus dipahami adalah mendapatkan dan memahami pengertian umum dari tahapan dan langkah-langkah yang digunakan dalam proyek sistem informasi serta beberapa teknik yang menghasilkan dokumen.
- b. Kedua, hal yang penting untuk dipahami bahwa SDLC adalah proses penyempurnaan yang dilakukan secara bertahap.

Hasil yang diperoleh dalam tahap analisis adalah suatu ide umum dari bagian sistem yang baru. Hasil ini lalu digunakan sebagai *input* pada tahap *design*, yang kemudian disempurnakan atau diperbaiki untuk menghasilkan dokumen yang menjelaskan detail dari bentuk sistem yang dibuat. Dokumen ini, akan digunakan dalam tahap implementasi untuk menghasilkan sistem yang sebenarnya. Setiap

tahapan menyempurnakan dan menguraikan pekerjaan yang telah dilakukan sebelumnya. Berikut dibawah ini merupakan setiap tahapan dalam SDLC:

a. *Planning* (Perencanaan)

Tahap *planning* / perencanaan merupakan proses dasar yang menjelaskan mengapa sebuah sistem informasi harus dibangun dan menentukan bagaimana sebuah tim proyek akan membangunnya. Terdapat dua langkah dalam *planning* (perencanaan) yaitu:

1. Inisiasi proyek, pada tahap ini nilai bisnis suatu sistem terhadap organisasi diidentifikasi: Apakah sistem tersebut akan menurunkan biaya atau meningkatkan pendapatan? Sebagian besar gagasan untuk sistem baru berasal dari luar area Departemen IT (misalnya dari Departemen Pemasaran, Departemen Akuntansi) dalam suatu bentuk *system request* (permintaan sistem). Permintaan sistem menyajikan ringkasan singkat tentang kebutuhan sistem dalam suatu bisnis dan menjelaskan bagaimana suatu sistem akan menciptakan nilai bisnis. Departemen IT bekerja sama dengan departemen lain yang menghasilkan permintaan (disebut sponsor proyek) untuk melakukan analisis kelayakan sistem informasi.

System request (permintaan sistem) dan analisis kelayakan disajikan dan dipresentasikan kepada pihak komite persetujuan *Information System* (komite pengarah), yang menentukan apakah proyek tersebut harus diambil atau tidak. Setelah proyek disetujui, manajer proyek membuat sebuah rencana kerja, membentuk anggota *staff* proyek, dan anggota teknisi lainnya di tempat, untuk membantu tim proyek mengendalikan dan mengarahkan proyek pada keseluruhan tahap SDLC.

b. *Analisis* (Analisis)

Berdasarkan Wikipedia Indonesia (2018), dalam linguistik, analisis adalah kajian yang dilaksanakan terhadap sebuah bahasa guna meneliti struktur bahasa tersebut secara mendalam. Sedangkan pada kegiatan laboratorium, kata analisis dapat juga berarti kegiatan yang dilakukan di laboratorium untuk memeriksa kandungan suatu zat dalam cuplikan.

Sedangkan berdasarkan Kamus Besar Bahasa Indonesia (KBBI) (2018), analisis adalah penyelidikan terhadap suatu peristiwa (karangan, perbuatan, dan sebagainya) untuk mengetahui keadaan yang sebenarnya (sebab-musabab, duduk perkaranya, dan sebagainya).

Berdasarkan Muslihudin dan Oktafianto (2016), analisis sistem adalah sebuah istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka. Analisis sistem merupakan tahapan paling awal dari pengembangan sistem yang menjadi fondasi dalam menentukan keberhasilan sistem informasi yang dihasilkan nantinya. Tahapan ini merupakan tahap yang mudah jika klien sangat paham dengan masalah yang dihadapi organisasinya dan tahu betul fungsionalitas dari sistem informasi yang akan dibuat. Tetapi tahap ini bisa menjadi tahap paling sulit jika klien tidak bisa mengidentifikasi kebutuhannya dan tertutup terhadap pihak luar yang ingin mengetahui detail proses-proses bisnisnya.

Tahap analisis dalam SDLC akan mengidentifikasi siapa yang akan memakai sistem, apa yang dikerjakan oleh sistem, kapan dan dimana sistem akan digunakan. Selama pada tahap ini, tim proyek menganalisis sistem yang digunakan, mengidentifikasi peluang untuk melakukan perbaikan sistem tersebut, dan mengembangkan konsep untuk sistem baru. Tahap ini memiliki tiga langkah:

1. Menyusun strategi analisis untuk mengarahkan usaha tim proyek. Seperti misalnya strategi untuk menganalisis sistem yang telah ada (disebut sebagai *as-is system*) beserta masalah-masalah yang ada dan untuk merancang suatu sistem baru (disebut sebagai *to-be system*).
2. Langkah selanjutnya adalah pengumpulan dokumen (misalnya, melalui wawancara atau kuesioner). Analisis informasi ini bersamaan dengan masukan dari sponsor proyek dan banyak orang lainnya yang mengarah pada pengembangan konsep untuk sistem baru. Konsep sistem baru ini kemudian digunakan sebagai dasar untuk mengembangkan seperangkat model analisis

bisnis, yang menggambarkan bagaimana bisnis akan beroperasi jika sistem baru dikembangkan.

3. Analisis, konsep sistem, dan model digabungkan menjadi dokumen yang disebut sebagai proposal sistem, yang dipresentasikan ke sponsor proyek dan divisi yang mengambil keputusan lainnya yang memutuskan apakah proyek dilanjutkan atau tidak.

Proposal sistem adalah dokumen awal yang menjelaskan apa kebutuhan yang harus dipenuhi untuk sistem baru. Karena ini benar-benar langkah pertama dalam perancangan sistem baru, beberapa ahli berpendapat bahwa tidak tepat menggunakan istilah "analisis" sebagai nama untuk fase ini. Beberapa ahli berpendapat lebih baik menggunakan nama "analisis dan desain awal." Tetapi sebagian besar organisasi menggunakan "analisis" untuk fase ini.

c. *Design (Desain)*

Tahap perancangan / *design* untuk mengidentifikasi bagaimana sistem akan beroperasi dalam perangkat keras (*hardware*), perangkat lunak (*software*) dan infrastruktur jaringan. Seperti *interface* pengguna, formulir, dan laporan. Serta program yang spesifik seperti *database*, dan file yang akan dibutuhkan. Langkah-langkah dalam tahap perancangan atau *design* menentukan secara tepat bagaimana sistem akan beroperasi. Berikut merupakan fase *design* yang memiliki empat langkah:

1. Strategi perancangan/*design*. Menjelaskan apakah sistem akan dikembangkan oleh *programmer* perusahaan itu sendiri, apakah sistem akan diberikan ke perusahaan lain (biasanya perusahaan konsultan), atau apakah perusahaan akan membeli paket perangkat lunak yang ada.
2. Pengembangan desain yang menggambarkan perangkat keras, perangkat lunak, dan infrastruktur jaringan yang akan digunakan. Dalam kebanyakan kasus, sistem akan menambah atau mengubah infrastruktur yang sudah ada dalam suatu organisasi. Desain antarmuka menentukan bagaimana pengguna akan bergerak melalui sistem (misalnya, metode navigasi seperti menu dan tombol di layar) dan formulir dan laporan yang akan digunakan sistem.

3. Pengembangan *database* dan spesifikasi file. Pengembangan ini menentukan dengan tepat data apa yang akan disimpan dan di mana mereka akan disimpan.
4. Tim analis mengembangkan desain program, mendefinisikan program tersebut dan apa yang akan dilakukan program.

Kumpulan dari hasil kegiatan ini seperti desain arsitektur, desain antarmuka, spesifikasi *database* dan *file*, dan perancangan program merupakan spesifikasi sistem yang diserahkan ke tim pemrograman untuk diimplementasi.

Pada akhir tahap perancangan, analisis kelayakan dan rencana proyek diperiksa ulang dan diperbaiki, dan keputusan lain dibuat oleh sponsor proyek dan divisi yang melakukan persetujuan untuk memutuskan apakah proyek akan dihentikan atau dilanjutkan.

d. *Implementation (Implementasi)*

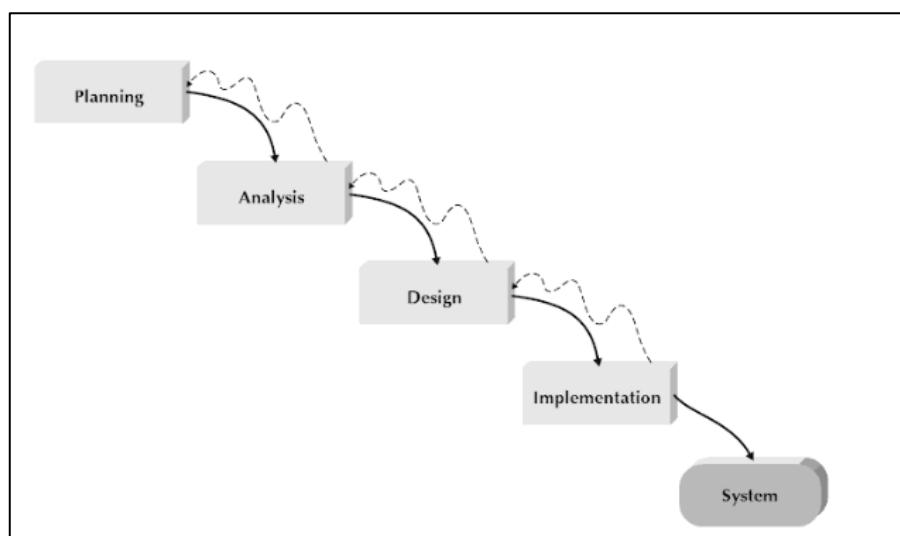
Tahap akhir dalam SDLC adalah tahap implementasi, di mana sistem benar-benar dibangun (atau dibeli, dalam kasus paket *design software*). Implementasi adalah fase yang paling mendapat perhatian, karena tahapan ini merupakan bagian terlama dan paling mahal dari proses pengembangan. Fase implementasi ini memiliki tiga langkah:

1. Pertama merekonstruksi sistem. Sistem ini dibangun dan diuji untuk memastikan kinerjanya sebagaimana yang telah dirancang. Karena biaya *bugs* bisa sangat besar, pengujian adalah salah satu langkah paling kritis dalam implementasi. Sebagian besar organisasi memberi lebih banyak waktu dan perhatian untuk menguji dari pada mengubah program di tempat pertama.
2. Instalasi sistem. Instalasi adalah proses dimana sistem lama dimatikan dan yang baru dinyalakan atau dihidupkan. Salah satu aspek terpenting adalah pengembangan rencana pelatihan untuk mengajarkan pengguna cara menggunakan sistem baru dan membantu mengelola perubahan-perubahan yang disebabkan oleh sistem yang baru.
3. Tim analisis menetapkan rencana dukungan untuk sistem. Rencana ini biasanya mencakup tinjauan formal atau informal serta cara sistematis untuk

mengidentifikasi perubahan besar dan kecil yang diperlukan untuk sistem tersebut.

2.8 Metode Pengembangan Sistem *Waterfall*

Berdasarkan Dennis *et al* (2015), contoh model yang masih digunakan hingga saat ini adalah *waterfall development* atau disebut juga dengan model pengembangan “air mancur”. Dengan metode pengembangan *waterfall*, analis dan pengguna atau *user* memproses secara berurutan dari satu fase ke fase berikutnya. Biasanya proses dalam satu fase sangat panjang dan ditampilkan dalam sponsor proyek atau *project sponsor* untuk disetujui sehingga dapat berlanjut ke fase selanjutnya. Suatu fase atau tahapan akan berlanjut jika sponsor proyek telah menyetujui tahapan tersebut. Model ini disebut dengan *waterfall development*, karena tahap demi tahapan bergerak maju dengan cara yang sama seperti air mancur. Tetapi, dalam model ini suatu tahapan sulit untuk bergerak mundur, (contohnya dari tahap desain mundur ke tahap analisis). Tahapan-tahapan metode *waterfall* menurut Dennis *et al* (2015) dapat dilihat pada Gambar II.4.

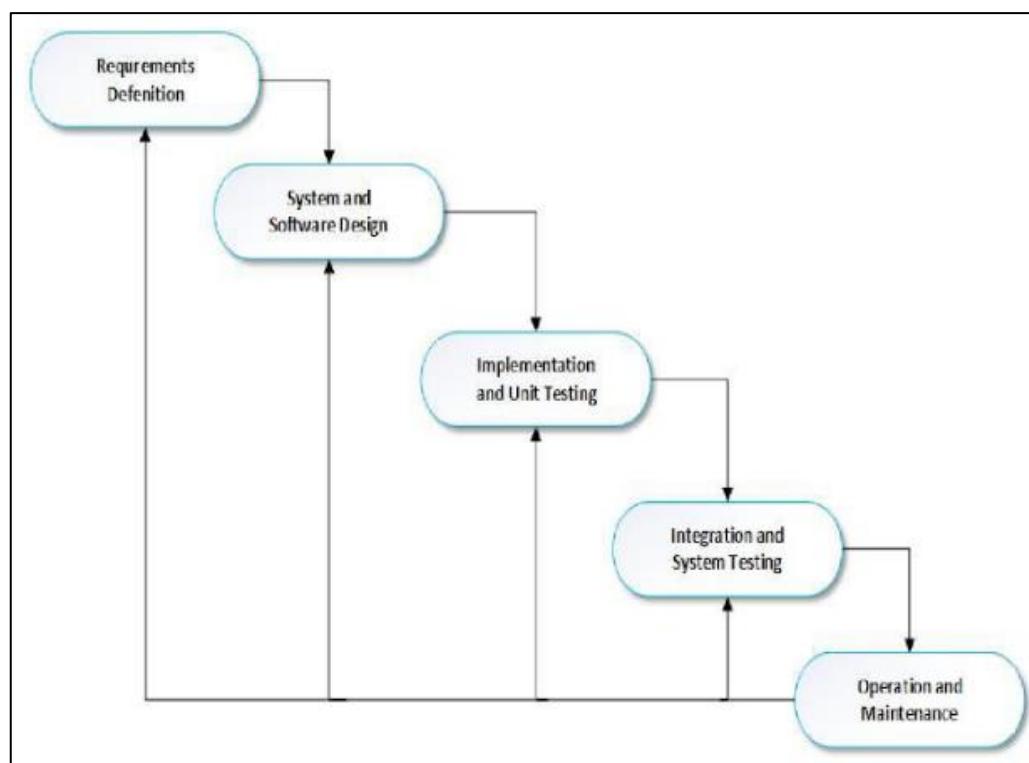


Gambar II.4 Metode *Waterfall* Berdasarkan Dennis *et al*

(Sumber: Dennis *et al*, 2015)

Berdasarkan Muharto dan Ambarita (2016), model *waterfall* dikenal juga dengan “*Linier Sequential Model*” yang sering juga disebut dengan “*classic life cycle*” atau model *waterfall*. Metode ini muncul pertama kali sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model/metode yang paling

banyak dipakai di dalam *Software Engineering* (SE). Metode ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan *system* lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement*. Menurut Pressman dan Sommerville (2010) pada Muharto dan Ambarita (2016), metode ini terdiri dari beberapa langkah, seperti ditunjukkan pada Gambar II.5.



Gambar II.5 Metode *Waterfall* Berdasarkan Pressman dan Sommerville

(Sumber: Pressman dan Sommerville, 2010)

Adapun tahap yang ada pada gambar dijelaskan sebagai berikut:

- 1) Analisis dan definisi kebutuhan. Layanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan *user* atau pemakai.
- 2) Perancangan sistem dan perangkat lunak. Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan

- melibatkan identifikasi dan deskripsi sistem perangkat lunak yang mendasar.
- 3) Implementasi dan pengujian unit. Pada tahap ini, perancangan perangkat lunak direalisasikan dengan program atau unit program.
 - 4) Integrasi dan pengujian sistem. Unit program diuji sebagai sistem yang lengkap untuk menjamin bahwa kebutuhan sistem telah dipenuhi.
 - 5) Operasi dan pemeliharaan, yaitu mengoperasikan program di lingkungannya dan melakukan pemeliharaan. Biasanya ini merupakan fase siklus hidup yang paling lama. Pemeliharaan mencakup koreksi dari berbagai *error* yang ditemukan pada tahap-tahap sebelumnya, melakukan perbaikan atas implementasi unit sistem dan pengembangan sistem.

2.9 Flowchart

Berdasarkan Sitorus (2015), untuk menggambarkan sebuah proses agar mudah dipahami oleh orang lain maka dibutuhkan alat bantu yang berbentuk diagram alir (*flowchart*). *Flowchart* menggambarkan urutan logika dari suatu prosedur pemecahan masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu. Diagram alir ini selain dibutuhkan sebagai alat komunikasi, juga diperlukan sebagai dokumentasi.

Berdasarkan Indrajani (2015), *flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya mempermudah penyelesaian masalah, khususnya yang perlu dipelajari dan dievaluasi lebih lanjut.

Berdasarkan Jalinus dan Ambiyar (2016), ada dua model *flowchart* yaitu (1) *system flowchart* dan (2) *program flowchart*. *System flowchart* tidak digunakan untuk menggambarkan urutan langkah pemecahan masalah serta hanya untuk menggambarkan prosedur dalam sistem yang dibentuk. Sementara *program flowchart* menggambarkan urutan logika dari suatu prosedur pemecahan masalah.

Tujuan dari *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas menggunakan

simbol-simbol. Simbol-simbol yang terdapat dalam sebuah *flowchart* dapat dilihat pada Tabel II.1.

Tabel II.1 Simbol-simbol *Flowchart*

| Simbol | Nama | Keterangan |
|--------|---------------------------|--|
| | Terminal | Menyatakan permulaan atau akhir suatu program. |
| | <i>Input-Output</i> | Meyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya. |
| | <i>Process</i> | Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer. |
| | <i>Decision</i> | Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak. |
| | <i>Connector</i> | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama. |
| | <i>Off-line Connector</i> | Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda. |
| | <i>Predefined Process</i> | Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal. |
| | <i>Punched Card</i> | Menyatakan input berasal dari kartu atau output ditulis ke kartu. |
| | <i>Punch Tape</i> | Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang. |
| | <i>Document</i> | Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak di kertas. |
| | <i>Flow</i> | Menyatakan jalannya arus suatu proses. |

(Sumber: Sitorus, 2015)

Keterangan: Penggambaran pada *flowmap* Penerimaan dan Pengeluaran Bahan



Baku menggunakan *Microsoft Visio* dimana simbol arsip adalah

2.10 *Unified Modeling Language* (UML)

Berdasarkan Mulyani (2016), *Unified Modelling Language* (UML) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem.

Berdasarkan Dennis *et al* (2015), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu *Structure diagram* dan *Behavior diagram*. *Structure diagram* biasanya digunakan untuk mempresentasikan data dan hubungan statik yang ada di dalam sebuah sistem informasi. Sedangkan *Behavior diagram* untuk menyediakan para analis dengan sebuah gambaran hubungan yang dinamis antara instansi atau objek yang merepresentasikan sistem informasi bisnis. UML ini mendefinisikan suatu set dari 15 teknik untuk pemodelan sistem, dapat dilihat pada Tabel II.2.

Tabel II.2 Ringkasan Diagram UML

| Nama Diagram | Fungsi |
|----------------------------------|--|
| <i>Structure Diagrams</i> | |
| <i>Class</i> | Menggambarkan hubungan antara pemodelan-pemodelan <i>class</i> di dalam sistem. |
| <i>Object</i> | Menggambarkan hubungan antara pemodelan-pemodelan objek di dalam sistem. |
| <i>Package</i> | Mengelompokkan elemen UML lainnya menjadi satu untuk membentuk level konstruksi yang lebih tinggi. |
| <i>Deployment</i> | Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk |

| Nama Diagram | Fungsi |
|---------------------------------|--|
| <i>Structure Diagrams</i> | |
| | menunjukkan komponen <i>software</i> sedang dijadikan arsitektur fisik suatu sistem. |
| <i>Component</i> | Menggambarkan hubungan fisik diantara komponen-komponen <i>software</i> . |
| <i>Composite</i> | Menggambarkan struktur internal dari suatu class dan hubungan diantara bagian-bagian dari suatu <i>class</i> . |
| <i>Profile</i> | Digunakan untuk mengembangkan eksistensi UML sendiri. |
| <i>Behavioral Diagrams</i> | |
| <i>Activity</i> | Menggambarkan proses bisnis masing-masing <i>class</i> . Aliran aktivitas dalam <i>use case</i> , atau <i>detail</i> desain dari suatu metode. |
| <i>Sequence</i> | Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam aktivitas berdasarkan urutan waktu. |
| <i>Communication</i> | Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu aktivitas. |
| <i>Interaction Overview</i> | Menggambarkan suatu ikhtisar alur dari kontrol suatu proses. |
| <i>Timing</i> | Menggambarkan interaksi yang terjadi diantara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu. |
| <i>Behavioral State Machine</i> | Menguji kebiasaan dari suatu <i>class</i> . |

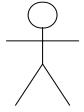
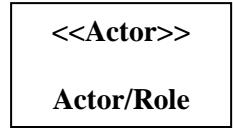
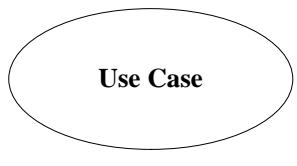
| Nama Diagram | Fungsi |
|-------------------------------|--|
| <i>Structure Diagrams</i> | |
| <i>Protocol State Machine</i> | Menggambarkan ketergantungan diantara perbedaan <i>interface</i> dari suatu <i>class</i> . |
| <i>Use Case</i> | Menggambarkan kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi antara sistem dan lingkungannya. |

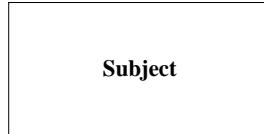
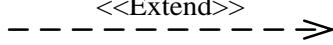
(Sumber: Dennis *et al.* 2015)

2.10.1 Use Case Diagram

Diagram *use case* dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam diagram *use case* ini adalah “apa” yang dapat diperbuat oleh sistem, dan bukan “bagaimana” sistem melakukannya. Diagram *use case* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut. Simbol-simbol *use case* dijelaskan pada Tabel II.

Tabel II.3 Simbol-simbol *Use-Case Diagram*

| Simbol | Fungsi |
|--|--|
|  Actor/Role  <>Actor>> Actor/Role | <ul style="list-style-type: none"> ▪ Seseorang atau sistem yang mendapatkan keuntungan dari sistem. ▪ Dilabelkan dengan peran. ▪ Bisa diasosiasikan dengan <i>actor</i> lainnya berdasarkan spesialisasi. ▪ Diletakan diluar batas sistem. |
|  Use Case | <ul style="list-style-type: none"> ▪ Merepresentasikan bagian utama dari sistem secara fungsional. ▪ Dapat memperpanjang <i>use case</i> lain. |

| Simbol | Fungsi |
|---|--|
| | <ul style="list-style-type: none"> ▪ Dapat termasuk <i>use case</i> lain. ▪ Diletakan didalam batas sistem. ▪ Dilabelkan dengan frasa kata kerja deskriptif. |
|  | <ul style="list-style-type: none"> ▪ Menyertakan nama subjek didalam maupun diatas. ▪ Merepresentasikan ruang lingkup dari sistem. |
|  | <ul style="list-style-type: none"> ▪ Merepresentasikan eksistensi dari <i>use case</i> untuk menyertakan perilaku optional. ▪ Memiliki anak panah yang digambarkan dari eksistensi <i>use case</i> ke <i>use case</i> dasar. |
|  Generalisasi | <ul style="list-style-type: none"> ▪ Merepresentasikan <i>use case</i> khusus ke yang lebih umum ▪ Memiliki anak panah yang digambarkan dari <i>use case</i> khusus ke <i>use case</i>. |

(Sumber: Dennis *et al.* 2015)

2.10.2 Activity Diagram

Berdasarkan Dennis *et al* (2015), *activity diagram* digunakan untuk memodelkan perilaku dalam proses bisnis yang independen terhadap objek. Dalam banyak hal, *activity diagram* dapat dipandang sebagai data flow diagram yang digunakan pada konjungsi dengan analisis terstruktur. Walaupun begitu, *activity diagram* tidak seperti data flow diagram, *activity diagram* menggunakan notasi yang mengalamatkan pemodelan paralel.

Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas

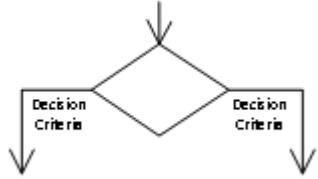
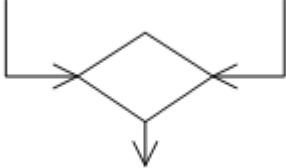
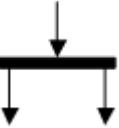
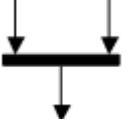
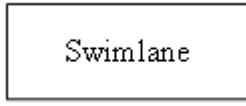
yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang akan ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada *Activity Diagram* dapat dilihat pada Tabel II.4.

Tabel II.4 Simbol-Simbol *Activity Diagram*

| No | Simbol | Nama | Deskripsi |
|----|--------|--------------|---|
| 1. | | Action | Merupakan notasi yang <i>simple</i> , dan tidak untuk dianalisis perilakunya. Dinamakan sesuai aksi yg dilakukan. |
| 2. | | Activity | Digunakan untuk mewakili satu set aksi yang dilakukan. |
| 3. | | Object Node | Digunakan untuk mewakili objek yang terhubung ke satu set arus objek. |
| 4. | | Control Flow | Menampilkan urutan eksekusi. |
| 5. | | Object Flow | Menunjukkan aliran dari sebuah objek dari satu aktivitas ke aktivitas lainnya. |
| 6. | | Initial Node | Menggambarkan permulaan dari suatu set aktivitas atau aksi. |

| No | Simbol | Nama | Deskripsi |
|-----|---|----------------------------|---|
| 7. |  | <i>Final-activity Node</i> | Menggambarkan akhir dari setiap aktivitas. |
| 8. |  | <i>Final-flow Node</i> | Menggambarkan akhir aliran control spesifik atau aliran objek. |
| 9. |  | <i>Decision Node</i> | Menggambarkan suatu kondisi untuk memastikan bahwa arah panah atau arah objek hanya memiliki satu jalur ke bawah. |
| 10. |  | <i>Merge Node</i> | Digunakan untuk membawa aliran keputusan yang berbeda ke satu <i>decision node</i> . |
| 11. |  | <i>Fork Node</i> | Menggambarkan percabangan dari satu aliran aktivitas. |
| 12. |  | <i>Join Node</i> | Menyatukan beberapa percabangan dari aliran aktivitas. |
| 13. |  | <i>Swimlane</i> | Menggambarkan akhir aliran kontrol spesifik atau aliran objek. |

(Sumber: Dennis, *et.al.* 2015)

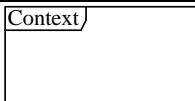
2.10.3 Sequence Diagram

Berdasarkan Dennis *et al* (2015), *sequence diagram* adalah salah satu dari dua jenis diagram interaksi. Mereka menggambarkan benda-benda yang berpartisipasi dalam kasus penggunaan dan pesan yang melewati antara mereka dari waktu ke waktu untuk satu *use case*. Sebuah diagram *sequence* adalah model

dinamis yang menunjukkan urutan eksplisit pesan yang lewat di antara objek dalam interaksi didefinisikan. Karena urutan diagram menekankan pemesanan berbasis waktu kegiatan yang terjadi di antara *set* benda, mereka sangat membantu untuk memahami spesifikasi *real-time* dan kompleks menggunakan kasus. Berikut simbol yang ada pada *Sequence Diagram* dapat dilihat pada Tabel II.5.

Tabel II.5 Simbol-Simbol *Sequence Diagram*

| No | Simbol | Nama | Deskripsi |
|----|--------|----------------------|---|
| 1. | | Aktor | Orang atau sistem yang berasal dari manfaat dan eksternal ke sistem yang berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan. |
| 2. | | Lifeline | Menyatakan kehidupan suatu objek. |
| 3. | | Objek | Berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan yang ditempatkan diatas diagram. |
| 4. | | Execution Occurrence | Menyatakan objek dalam keadaan aktif dan berinteraksi pesan. |
| 5. | | Message | Pesan yang menggambarkan komunikasi yang terjadi antar objek. |
| 6. | | Message (return) | Pesan yang dikirim untuk diri sendiri. |
| 7. | | Guard Condition | Menggambarkan tes yang harus dipenuhi untuk pesan yang akan dikirim. |
| 8. | | Object Destruction | Ditempatkan di akhir lifeline suatu objek untuk menunjukkan akhir dari eksistensi. |

| No | Simbol | Nama | Deskripsi |
|----|---|---------------------------|---|
| 9. |  | <i>Frame</i> | Mengindikasikan konteks dari <i>sequence diagram</i> . |
| 8. | x | <i>Object Destruction</i> | Ditempatkan di akhir <i>lifeline</i> suatu objek untuk menunjukkan akhir dari eksistensi. |

(Sumber: Dennis, *et.al* 2015)

2.10.4 Class Diagram

Berdasarkan Dennis *et al* (2015), sebuah *class diagram* digunakan untuk mengatur dan menyimpan informasi yang ada dalam sistem. Selama analisis, *class* mengarah kepada orang, tempat, atau sesuatu yang akan ditangkap oleh sistem. Sebuah *class* digambarkan dengan tiga buah persegi panjang, dimana di bagian atas diisi dengan nama *class*, di bagian tengah diisi dengan *attribute*, dan di bagian bawah diisi dengan *operation*. Sebuah *class* merupakan deskripsi dari sekumpulan objek yang memiliki properti (*attribute*), operasi (*method*), relasi (*association*), dan tingkah laku (*behavior*) yang sama. Sebuah *class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda /fungsi). Berikut simbol-simbol yang ada pada diagram kelas dapat dilihat pada Tabel II.6.

Tabel II.6 Simbol-Simbol *Class Diagram*

| No. | Simbol | Deskripsi | | | |
|----------------|---|---|-------------|---------------|---|
| 1. | <p><i>Class</i></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Class 1</td> </tr> <tr> <td>-attribute1</td> </tr> <tr> <td>+operation1()</td> </tr> </table> | Class 1 | -attribute1 | +operation1() | <p><i>Class</i> berfungsi untuk menggambarkan orang/aktor, tempat, atau sesuatu yang dibutuhkan oleh sistem untuk mendokumentasikan dan menyimpan informasi.</p> <p>Judul <i>class</i> ditulis dengan huruf tebal dan berada di tengah atas.</p> <p><i>Attribute</i> ditempatkan di tengah kotak.</p> <p><i>Operation</i> ditempatkan di bawah kotak.</p> |
| Class 1 | | | | | |
| -attribute1 | | | | | |
| +operation1() | | | | | |
| 2. | Nama <i>attribute</i> | Mewakili suatu properti untuk mendeskripsikan keadaan dari suatu objek. | | | |

| No. | Simbol | Deskripsi |
|-----|---|---|
| | | Dapat diturunkan dari <i>attribute</i> lain dengan menempatkan garis miring sebelum nama <i>attribut</i> . |
| 3. | Nama operasi/ <i>operation</i> () | Mewakili suatu aksi atau fungsi dari suatu <i>class</i> . Dapat diklasifikasikan sebagai konstruktor (<i>method</i> yang secara otomatis dipanggil/dijalankan pada sebuah <i>class</i>), <i>query</i> , atau perubahan operasi. Tanda kurung mengandung informasi atau parameter yang dibutuhkan untuk menampilkan operasi. |
| 4. | <i>AssociatedWith</i>  | Mewakili sebuah hubungan antara beberapa <i>class</i> atau <i>class</i> dengan <i>class</i> . Digambarkan dengan kata kerja atau nama, yang mewakili sebuah hubungan. Tersedia diantara satu atau lebih <i>class</i> . |
| 5. | <i>Generalization</i>  | Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus Mewakili hubungan antara beberapa <i>class</i> (<i>class</i> banyak). |
| 6. | <i>Aggregation</i> 0..* <i>IsPartOf</i> ▶ 1  | Mewakili bagian hubungan yang logis antara beberapa <i>class</i> atau <i>class</i> dengan <i>class</i> . Merupakan bagian dari <i>association</i> . |
| 7. | <i>Composition</i> 1..* <i>IsPartOf</i> ▶ 1  | Mewakili bagian hubungan yang fisik antara beberapa <i>class</i> atau <i>class</i> dengan <i>class</i> . Merupakan bagian dari <i>association</i> . |

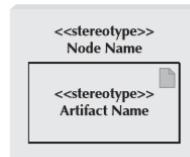
(Sumber: Dennis *et al.* 2015)

2.10.5 Deployment Diagram

Berdasarkan Dennis *et al* (2015) menyebutkan bahwa *deployment diagram* digunakan untuk merepresentasikan hubungan antara komponen *hardware* yang digunakan dalam infrastruktur fisik dari suatu sistem informasi. *Deployment*

diagram juga dapat digunakan untuk mewakili komponen *software* dan bagaimana mereka ditempatkan di atas arsitektur fisik atau infrastruktur dari suatu sistem informasi.

Tabel II.7 *Deployment Diagram*

| No. | Simbol | Istilah dan Definisi |
|-----|---|--|
| 1. |  | <p><i>Node:</i></p> <ul style="list-style-type: none"> – Apakah sumber daya komputasi, misalnya, komputer klien, server, jaringan terpisah, atau perangkat jaringan individu. – Dilabeli dengan namanya. – Dapat berisi stereotip untuk secara khusus memberi label jenis node yang diwakili, misalnya, perangkat, workstation klien, server aplikasi, perangkat seluler, dll. |
| 2. |  | <p><i>Artifact:</i></p> <ul style="list-style-type: none"> – Adalah spesifikasi perangkat lunak atau basis data, misalnya, basis data atau tabel atau tampilan database, komponen atau lapisan perangkat lunak. – Dilabeli dengan namanya. – Dapat berisi stereotip untuk secara khusus melabeli jenis artefak, misalnya, file sumber, tabel basis data, file yang dapat dieksekusi, dll. |
| 3. |  | <p><i>Node dengan Deployed artifact:</i></p> <ul style="list-style-type: none"> – Menampilkan artefak yang ditempatkan pada simpul fisik. |
| 4. |  | <p><i>Communication path:</i></p> <ul style="list-style-type: none"> – Mewakili hubungan antara dua <i>node</i>. – Memungkinkan <i>node</i> untuk bertukar pesan. – Dapat berisi stereotip untuk secara khusus melabeli jenis jalur komunikasi yang diwakili, (misalnya, LAN, Internet, serial, paralel). |

(Sumber: Dennis *et al*, 2015)

2.11 Sistem Manajemen Basis Data

Berdasarkan Pamungkas (2017), sistem manajemen basis data merupakan perangkat lunak yang didesain untuk melakukan penyimpanan dan pengaturan basis data. Sementara basis data, merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan *software* untuk memanipulasi dengan tujuan tertentu.

2.11.1 *Database*

Berikut adalah pengertian dari *database* menurut para ahli yang tercantum dalam Jalinus dan Ambiyar (2016):

1. Berdasarkan Connolly dan Begg (2010), *database* adalah sekumpulan data tersebar yang berhubungan secara logis, dan penjelasan dari data ini dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.
2. Menurut Gottschalk dan Saether dalam jurnal (2010), *database* adalah sekumpulan data yang terorganisasi untuk mendukung banyak aplikasi secara efisien dengan memusatkan data dan mengontrol *data redundant*.

Maka dapat disimpulkan *database* adalah sekumpulan data yang saling terhubung dan terorganisasi yang disimpan berdasarkan skema dengan memusatkan data dan mengontrol *data redundant* untuk memenuhi kebutuhan informasi dari suatu organisasi.

Berdasarkan Suryo (2012), *database* adalah kumpulan informasi yang disimpan dalam computer secara sistematik untuk memperoleh informasi dari basis data tersebut. Konsep dasar dari *database* adalah kumpulan dari catatan, atau potongan dari pengetahuan. Sebuah *database* memiliki penjelasan terstruktur dari jenis fakta yang disimpan didalamnya, penjelasan itu disebut dengan skema. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur *database*, yang dikenal dengan *database* model atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah yaitu mewakili semua informasi dalam bentuk tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Contoh dari tabel barang dapat dilihat pada Tabel II.8.

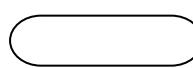
Tabel II.8 Contoh Tabel Barang

| Field | Field Name | Type | Width | Description |
|--------------|-------------------|-------------|--------------|--------------------|
| 1 | Kode_Barang | Varchar | 7 | Kode Barang* |
| 2 | Nama_Barang | Varchar | 35 | Nama Barang |
| 3 | Satuan | Varchar | 15 | Satuan |
| 4 | Harga | Varchar | 9 | Harga |

(Sumber: Suryo, 2012)

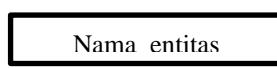
2.11.2 Entity Relationship Diagram (ERD)

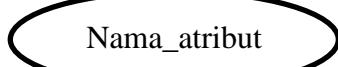
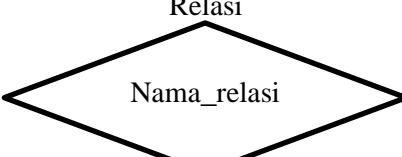
Berdasarkan Yanto (2016), ERD adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek dunia nyata yang sering dikenal dengan hubungan antar entitas. Sebagai contoh jika membuat ERD dari sistem perpustakaan maka bahan sebagai objek ERD bias berupa anggota, buku, peminjaman, pengembalian, dan lain sebagainya. ERD terdiri dari tiga komponen utama yaitu:

| Simbol | Nama |
|---|--------------------------------|
|  | Entitas (<i>entity</i>) |
|  | Atribut (<i>attribute</i>) |
|  | Relasi (<i>Relationship</i>) |

Berikut ini adalah symbol-simbol dari *Entity Relationship Diagram* (ERD), dapat dilihat pada Tabel II.9.

Tabel II.9 Simbol-simbol *Entity Relationship Diagram*

| No. | Simbol | Deskripsi |
|-----|--|--|
| 1. | Entitas  Nama entitas | Entitas merupakan data inti yang akan disimpan |

| No. | Simbol | Deskripsi |
|-----|---|---|
| 2. | Atribut  | Field atau kolom data yang butuh disimpan dalam suatu entitas |
| 3. | Atribut kunci primer  | Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan. |
| 4. | Relasi  | Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja |
| 5. | Asosiasi  | Penghubung antara relasi dan entitas |

(Sumber: Rosa dan Shalahuddin, 2015)

2.11.3 Kamus Data

Berdasarkan Indrajani (2015), kamus data adalah katalog fakta tentang data dan kebutuhan informasi suatu sistem informasi. Kamus data terdapat pada tahap analisis dan perancangan. Pada tahap analisis, kamus data berfungsi untuk mendefinisikan data yang mengalir pada sistem. Sedangkan pada tahap perancangan, kamus data digunakan untuk merancang masukan dan keluaran seperti laporan serta basis data. Sumber kamus data yaitu Data *Store* (file-file), Data *Flow* (aliran data), dan Data *Element* yang dinyatakan dalam spesifikasi data dan berasal dari file. Berikut adalah notasi-notasi yang digunakan dalam kamus data dapat dilihat pada Tabel II.10.

Tabel II.10 Notasi-notasi Kamus Data

| Notasi | Keterangan |
|--------|--|
| = | <i>Is composed of</i> |
| + | <i>And</i> |
| () | <i>Optional (may be present or absent)</i> |
| { } | <i>Iteration</i> |
| [] | <i>Select one of several alternative choices</i> |
| ** | <i>Comment</i> |

| Notasi | Keterangan |
|--------|---|
| @ | <i>Identifier (key field) for a store</i> |
| | <i>Separates alternative choices</i> |

(Sumber: Indrajani, 2015)

Contoh kamus data antara lain:

Name = courtesy-title + first-name + (middle-name) + last-name

Courtesy-title = [Mrs. / Mr. / Miss / Ms. / Dr. / Professor]

First-name = {legal-character}

Middle-name = {legal-character}

Last-name = {legal-character}

Legal-character = [A-Z / a-z / 0-9]

2.12 MariaDB

Berdasarkan mariadb.com (2018), MariaDB adalah *database* sumber terbuka yang paling cepat berkembang. Pertumbuhan MariaDB dipercepat oleh 12 juta pengguna di 45 negara yang tidak hanya menjalankan aplikasi *mission-critical* tetapi juga terus berinovasi dan secara aktif menyumbangkan teknologi baru untuk proyek-proyek MariaDB. Selain itu, MariaDB adalah *database default* untuk distribusi Linux seperti Red Hat dan SUSE, dan tersedia melalui penyedia *cloud* besar seperti Amazon AWS dan Microsoft Azure.

MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih mandiri yakni MariaDB.

MariaDB adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MariaDB, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

MariaDB dibangun di atas arsitektur modern yang dapat diperluas di setiap lapisan: klien, kluster, kernel, dan penyimpanan. Perpanjangan ini memberikan dua keuntungan besar. Hal ini memungkinkan inovasi komunitas berkelanjutan melalui *plugin*, yang berarti bahwa berbagai mesin penyimpanan - seperti MariaDB ColumnStore atau Facebook MyRocks - dapat diintegrasikan melalui arsitektur diperluas milik MariaDB. Selain itu, memudahkan pelanggan untuk mengkonfigurasi MariaDB untuk mendukung berbagai macam kasus penggunaan dari OLTP ke OLAP.

2.12.1 Tipe Data

Berdasarkan Jubilee (2014), Tipe data dibagi menjadi tiga kategori: numerik, tanggal, dan tipe string.

a. Tipe Data Numerik

Berikut ini daftar data numerik yang digunakan:

1. **INT** – nilai integer dengan ukuran normal. *Signed* berkisar -2147483648 sampai 2147483647. *Unsigned* berkisar dari 0 sampai 4294967295. *Unsigned* dimaksudkan, hanya untuk kisaran non minus.
2. **TINYINT** – nilai integer yang sangat kecil. *Signed* berkisar dari -128 sampai 127. *Unsigned* berkisar dari 0 sampai 255.
3. **SMALLINT** – nilai integer yang kecil. *Signed* berkisar dari -32768 sampai 32767. *Unsigned* berkisar dari 0 sampai 65535.
4. **MEDIUMINT** – nilai integer medium. *Signed* berkisar dari -8388608 sampai 8388607. *Unsigned* berkisar 0 sampai 16777215.
5. **BIGINT** – nilai integer dengan ukuran besar. *Signed* berkisar dari -9223372036854775808 sampai 9223372036854775807. *Unsigned* berkisar dari 0 sampai 18446744073709551615.
6. **FLOAT** – berdasarkan Sutaji (2012), bilangan *floating point* yang kecil (presisi tunggal). Jangkauan nilai yang diizinkan adalah -3.402823466E+38 s/d -1.175494351e-38,0 dan 1.175494351E-38 s/d 3.402823466E+38.
7. **DOUBLE** – berdasarkan Sutaji (2012), bilangan *floating point* dengan ukuran normal (presisi ganda). Jangkauan nilai yang diizinkan adalah -

1.7976931348623157E+308 s/d -2.225073858507201E-308,0 dan
2.225073858507201E-308 s/d 1.7976931348623157E+308.

b. Tipe Data Tanggal dan Waktu

Berikut ini daftar tipe data dan waktu:

1. **DATE** – tanggal dengan format YYY-MM-DD diantara 1000-01-01 sampai 9999-12-31. Sebagai contoh, tanggal 30 Desember 1973 akan disimpan sebagai 1973-12-30.
2. **DATETIME** – kombinasi tanggal dan waktu dengan format YYY-MM-DD HH:MM:SS, diantara 1000-01-01 00:00:00 sampai 9999-12-31 23:59:59. Sebagai contoh, pukul 15:30 sore pada tanggal 30 Desember 1973 akan disimpan sebagai 1973-12-30 15:30:00.
3. **TIME** – menyimpan nilai waktu dengan format HH:MM:SS. Contohnya, 15:30:00.
4. **YEAR** – menyimpan tahun dengan format 2 digit atau 4 digit. YEAR(2) Memiliki rentang nilai dari tahun 1970 sampai 2069 (70 sampai 69). YEAR(4) memiliki rentang nilai dari tahun 1901 sampai 2155.

c. Tipe Data String

Berikut ini daftar tipe data string:

1. **CHAR** – string dengan panjang tetap antara 1 sampai 255 karakter.
2. **VARCHAR** – string dengan panjang bervariasi antara 1 sampai 255 karakter.
3. **BLOB/TEXT** – sebuah *field* dengan panjang maksimum 65535 karakter. BLOB (*Binary Large Objects*) digunakan untuk menyimpan data binary yang besar, seperti gambar atau tipe file lainnya. *Field* yang didefinisikan sebagai TEXT juga menyimpan jumlah data yang besar; perbedaannya, BLOB bersifat *case sensitive* yang artinya huruf besar dan kecil akan diartikan secara berbeda. Sebagai contoh penulisan alert () huruf kecil semua. Jika ditulis dengan ALERT () maka kode tidak bekerja. Sedangkan TEXT tidak.

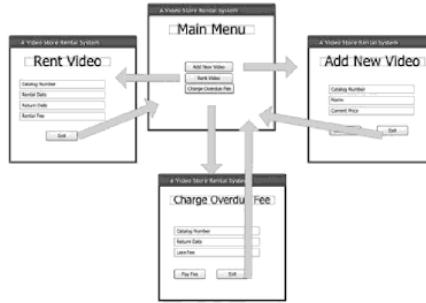
4. **TINYBLOB/TINYTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 255 karakter.
5. **MEDIUMBLOB/MEDIUMTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 16777215 karakter.
6. **LONGLOB/LONGTEXT** – sebuah kolom BLOB atau TEXT dengan panjang maksimum 4294967295.
7. **ENUM** – berdasarkan Sutaji (2012), enum adalah sebuah *enumeration*. Sebuah obyek string yang hanya boleh memiliki satu nilai, yang terambil dari ‘value1’, ‘value2’, ..., NULL atau nilai spesial “”error. Sebuah *enum* dapat menampung 65535 pilihan nilai.

2.13 *Windows Navigation Diagram (WND)*

Berdasarkan Dennis *et al* (2011), *Windows Navigation Diagram* merupakan *statechart* diagram khusus yang berfokus pada *user interface*.

1. Menunjukkan bagaimana semua layar, formulir, dan laporan terkait.
2. Menunjukkan bagaimana pengguna bergerak dari satu ke yang lain.
3. Seperti diagram keadaan untuk user interface:
 - a. Kotak mewakili komponen.
 - b. Panah mewakili transisi.
 - c. Stereotipe menunjukkan tipe antarmuka.

Berdasarkan Dennis *et al* (2015), *Windows Layout Diagram* adalah suatu gambar yang menyerupai antar-muka pengguna. Biasanya dibuat dalam *Microsoft Visio*. Dengan *visio*, perancang dapat menarik komponen antar-muka dengan cepat ke dalam kanvas. Contoh *Windows Navigation Diagram* dapat dilihat pada Gambar II.
5.



Gambar II.5 *Windows Navigation Diagram*

(Sumber: Dennis *et al*, 2015)

2.14 PHP

Berdasarkan Tim EMS (2016), PHP adalah singkatan dari *Hypertext Preprocessor*. Merupakan bahasa *scripting* untuk web yang cukup popular. PHP adalah bahasa *script*, artinya ditanamkan atau disisipkan ke dalam HTML (*Hypertext Markup Language*). Salah satu kelebihan PHP dibandingkan dengan bahasa pemrograman web lainnya adalah banyak web server yang mendukung PHP mulai dari Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah. Ada juga paket web server yang memudahkan melakukan *instalasi* sekali klik, seperti XAMPP yang tersedia dalam berbagai system operasi.

Berdasarkan Winarno dan Zaki (2013). PHP adalah bahasa pemrograman yang memungkinkan *programmer* menggenerate kode HTML secara dinamis, artinya *programmer* bisa membuat tampilan halaman web yang dinamis, bisa berubah-ubah sesuai dengan keinginan *programmernya*.

Berdasarkan Jubilee (2017), PHP merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis website. Sebagai sebuah aplikasi, website tersebut hendaknya memiliki sifat dinamis dan interaktif. Memiliki sifat dinamis artinya, website tersebut bisa berubah tampilan kontennya sesuai kondisi tertentu (misalnya menampilkan produk berbeda-beda untuk setiap pengunjung). Interaktif artinya, website tersebut dapat member *feedback* bagi *user* (misalnya, menampilkan hasil pencarian produk). Kode standar untuk mendesain website sebenarnya adalah HTML. Namun, tanpa memanfaatkan PHP, kode HTML tidak

dapat digunakan untuk membuat website dengan dua sifat diatas. Dengan kata lain jika mengandalkan HTML saja, tampilan situs yang dilihat para pengunjung satu dan lainnya relative sama dan statis. Dengan memanfaatkan PHP, *programmer* dapat membuat sistem *username login, shopping card, database*, dan sebagainya.

Berdasarkan Adelheid dan Khairil (2012), PHP atau *Hypertext Preprocessor* merupakan bahasa berbentuk *script* yang ditempatkan dalam server dan dieksekusi di dalam server untuk selanjutnya ditransfer dan dibaca oleh *client*. PHP juga bisa disisipkan dalam bahasa HTML. PHP pertama kali diciptakan oleh seorang pria berkewarganegaraan Denmark yang bernama Rasmus Lerdorf pada tahun 1995.

Berdasarkan Wikipedia Indonesia, PHP atau *Hypertext Preprocessor* adalah bahasa skrip yang dapat disisipkan ke dalam HTML. PHP banyak digunakan untuk membuat program web dinamis. Dalam PHP, mempunyai berbagai sintaks dasar. Sintaks tersebut yaitu antara lain:

1. Pembatas

PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun yang di luar pembatas tidak diproses oleh PHP. Pembatas yang paling umum adalah “<?php” untuk membuka dan “?>” untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, Javascript.

2. Variabel

Variabel diawali dengan simbol *dollar* \$. Pada versi PHP 5, diperkenalkan jenis isyarat yang memungkinkan fungsi untuk memaksa \$ menjadi parameter objek dari *class* tertentu seperti array, atau fungsi. Contoh variabel dapat ditulis sebagai \$nama_variabel. Penulisan fungsi, penamaan kelas, nama variabel adalah peka akan huruf besar dan huruf kecil.

3. Komentar

PHP memiliki 3 jenis sintaks sebagai komentar pada kode yaitu tanda blok / * * /, komentar 2 baris // serta tanda pagar # digunakan untuk komentar

satu baris. Komentar bertujuan untuk meninggalkan catatan pada kode PHP dan tidak akan diterjemahkan ke program.

4. Fungsi

Berikut adalah contoh dari penulisan fungsi atau *method* yang ada di PHP.

Function tampilkan (\$data=" ") // mendefinisikan fungsi, "tampilkan" adalah nama sebuah fungsi

If (\$data) *return* \$data;

Else return 'tidak ada data'; // melakukan proses pengolahan data

Echo tampilkan("isi halaman") // menjalankan fungsi

PHP memiliki 8 tipe data, yaitu: *Boolean, Integer, Float/Double, String, Array, Object, Resource, NULL*.

2.15 *Framework*

Berdasarkan Supono dan Putratama (2016), *framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau *class* dari awal. Seorang *programmer* tidak perlu membuat dari awal fungsi-fungsi seperti fungsi koneksi ke *database*, fungsi *string*, dan lainnya.

Berdasarkan Wikipedia Indonesia, PHP *framework* adalah sebagai sebuah kerangka kerja yang disusun oleh berbagai komunitas pengembang web diseluruh dunia. Kerangka ini bertujuan untuk mempermudah dalam membuat sebuah aplikasi web bagi yang sering menulis *script* PHP secara keseluruhan dan itu pun akan diulang pada halaman yang lain. *Framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur-prosedur dari *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau *class* dari awal. Contoh *framework* PHP seperti *CakePHP, Codeigniter, Laravel, PRADO, Symfony, Zend Framework, Yii, Akelos, QPHP, ZooP*. Ada beberapa alasan mengapa menggunakan *framework* yaitu:

- b. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- c. Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada)
- d. Umumnya *framework* menyediakan fasilitas-fasilitas umum yang dipakai sehingga *programmer* tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, *multiple database*, *scaffolding*, pengaturan *session*, *error handling*, dan lain-lain).

Berdasarkan Jubilee (2015), *framework* merupakan banyak kode, yang disimpan dalam beberapa file yang terpisah, dan memudahkan dalam penggunaan kode yang digunakan secara berulang-ulang. Dengan *framework*, *programmer* tidak perlu menulis baris kode yang panjang untuk fungsi tertentu. *Programmer* hanya perlu menggunakan fungsi yang sudah dituliskan ke dalam *framework*.

2.16 CodeIgniter

Berdasarkan Supono dan Putratama (2016), *CodeIgniter* adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model*, *View*, *Controller*) untuk membangun *website* dinamis dengan PHP. Ada 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi yaitu:

- 2 *View*, merupakan bagian yang menangani *presentation logic*. *View* berfungsi untuk menerima dan mempresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian *model*.
- 3 *Model*, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert*, *update*, *delete*, *search*) menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

Controller, merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang diproses oleh aplikasi.

Berdasarkan Jubilee (2015), CI merupakan *framework* yang dibuat oleh Rick Ellis, CEO pMachine. CI dibuat dan terus dikembangkan, dan dapat digunakan

secara gratis. CI juga mempunyai forum yang dapat digunakan untuk berbagi tips, mencari, dan mengatasi *bug* dalam kode. Tersedia pada situs www.codeigniter.com.

Berdasarkan Komputer (2014), *codeigniter* adalah *framework* bahasa pemrograman PHP. Sebagai contoh, situs kompas.com dan okezone.com mengadopsi sistem ini sebagai *framework* websitenya. *Codeigniter* menawarkan kemudahan serta standarisasi dalam proses pengembangan aplikasi berbasis website. Dengan *codeigniter*, proses pengembangan menjadi lebih cepat dan terstandar. Standard ini sangat berguna dalam pengembangan yang dikerjakan oleh tim. *Codeigniter* juga menyediakan *library*, dan *helper* yang berguna dan mempermudah proses. Terdapat beberapa kelebihan penggunaan *codeigniter* dalam pengembangan suatu proyek, yaitu:

a. Menghemat Waktu

Dengan struktur dan *library* yang telah disediakan, membuat *programmer* hanya fokus pada logika pemrograman yang sedang dikerjakan.

b. *Code Reuse*

Dengan *codeigniter*, suatu pekerjaan akan memiliki standard yang baku. Sehingga *programmer* dapat menggunakan kembali pada proyek-proyek yang akan datang.

c. Dukungan Komunitas

CI mempunyai banyak komunitas yang tersebar untuk membantu masalah *programmer*. Dengan adanya komunitas, *programmer* dapat saling berkonsultasi mengenai masalah yang dihadapi.

Berdasarkan Wikipedia Indonesia, *codeigniter* merupakan aplikasi sumber terbuka yang berupa *framework* PHP dengan model MVC (*Model*, *View*, *Controller*) untuk membangun website dinamis dengan menggunakan PHP. *Codeigniter* memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. *Codeigniter* pertama kali dirilis pada 28 Februari 2006. Model *view controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *small talk*, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user*

interface, dan bagian yang menjadi kontrol aplikasi. 3 jenis komponen MVC adalah:

1. *View*, bagian ini biasanya berupa file *template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. *Model*, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

CodeIgniter adalah Kerangka Pengembangan aplikasi (sebuah toolkit) untuk orang-orang yang membangun situs *web* dengan menggunakan bahasa PHP (*Hypertext Preprocessing*). Tujuannya adalah untuk memungkinkan *programmer* mengembangkan proyek lebih cepat dari pada menulis kode dari awal dengan menyediakan kumpulan pustaka untuk tugas-tugas yang biasanya dibutuhkan, serta antarmuka yang sederhana dan struktur logis untuk mengakses pustaka-pustaka ini. *CodeIgniter* memungkinkan *programmer* fokus pada proyek dengan meminimalkan jumlah kode yang diperlukan untuk tugas yang diberikan.

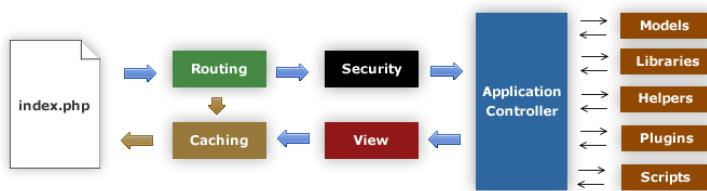
CodeIgniter didasarkan pada pola pengembangan *Model-View-Controller*. MVC adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi. Dalam praktiknya, memungkinkan halaman *web* pengguna mengandung *scripting minimal* karena presentasi terpisah dari *scripting PHP*.

- a. *Model*, mewakili struktur data. Biasanya kelas model akan berisi fungsi-fungsi yang membantu untuk mengambil, memasukkan, dan memperbarui informasi dalam *database*.
- b. *View*, adalah informasi yang sedang disajikan kepada pengguna. *View* biasanya adalah halaman *web*, tetapi di *CodeIgniter*, tampilan juga bisa

berupa fragmen halaman seperti *header* atau *footer*. Hal ini juga dapat berupa halaman RSS, atau jenis "halaman" lainnya.

- c. *Controller*, berfungsi sebagai perantara antara *Model*, *View*, dan sumber daya lain yang diperlukan untuk memproses permintaan HTTP dan menghasilkan halaman web.

CodeIgniter memiliki pendekatan yang cukup longgar untuk MVC karena *Model* tidak diperlukan. Jika pengguna tidak memerlukan pemisahan tambahan, atau menemukan bahwa mempertahankan *model* memerlukan lebih banyak kerumitan daripada yang diinginkan, pengguna dapat mengabaikannya dan membangun aplikasi secara *minimal* menggunakan Pengontrol dan Tampilan.



Gambar II.6 Alur Kerja *CodeIgniter*

(Sumber: *CodeIgniter*, 2015)

Indeks.php berfungsi sebagai pengontrol depan, menginisialisasi sumber daya dasar yang dibutuhkan untuk menjalankan *CodeIgniter*.

1. *Router* memeriksa permintaan HTTP untuk menentukan apa yang harus dilakukan.
2. Jika *file* cache ada, akan dikirim langsung ke *browser*, melewati eksekusi sistem normal.
3. Keamanan. Sebelum pengontrol aplikasi dimuat, permintaan HTTP dan data yang dikirim oleh pengguna disaring untuk keamanan.
4. *Controller* memuat *model*, pustaka inti, pembantu, dan sumber daya lain yang diperlukan untuk memproses permintaan khusus.
5. Tampilan akhir ditampilkan kemudian dikirim ke *browser web* untuk dilihat. Jika cache diaktifkan, tampilan akan di-cache terlebih dahulu sehingga pada permintaan berikutnya dapat dilayani.

2.17 Black Box Testing

Berdasarkan Jurnal Mustaqbal dan Firdaus (2015), *black box testing* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan *input* dan melakukan pengetesan pada spesifikasi fungsional program. *Black box testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).

Berdasarkan Utami dan Asnawati (2015), kategori *error* yang akan diketahui melalui *black box testing* adalah:

- a. Fungsi yang hilang atau tak benar.
- b. *Error* dari antar-muka.
- c. *Error* dari struktur data atau akses *database*.
- d. *Error* dari kinerja atau tingkah laku.

2.17.1 Metode Black Box Testing

- a. *Equivalence partitioning* merupakan metode *blackbox testing* yang membagi domain masukan dari suatu program ke dalam kelas-kelas data. Contoh *black box testing* dengan *equivalence partitioning* adalah pemeliharaan data untuk aplikasi bank yang sudah diotomatiskan. Pemakai dapat memutar nomor telepon bank dengan menggunakan mikro komputer yang terhubung dengan *password* yang telah ditentukan dan diikuti dengan perintah-perintah. Data yang diterima adalah:

Kode area : kosong atau 3 digit

Prefix : 3 digit atau tidak diawali 0 atau 1

Suffix : 4 digit

Password : 6 digit alfanumerik

Perintah : *check, deposit, dll,*

Selanjutnya kondisi *input* digabungkan dengan masing-masing data elemen dapat ditentukan sebagai berikut:

- Kode area : kondisi *input, boolean*-kode area mungkin ada atau tidak kondisi *input, range* – nilai ditentukan antara 200 dan 999.
- Prefix : kondisi *input range > 200* atau tidak diawali 0 atau 1
- Suffix : kondisi *input* nilai 4 digit
- Password* : kondisi *input boolean- password* mungkin diperlukan atau tidak kondisi *input* nilai dengan 6 karakter *string*
- Perintah : kondisi *input set* berisi perintah-perintah yang telah Didefinisikan

2.17.2 Perbedaan *White Box* dan *Black Box*

White Box (Struktural)

1. Dilakukan oleh penguji yang mengetahui tentang QA.
2. Melakukan *testing* pada *software/program* aplikasi menyangkut *security* dan *performance* program tersebut (meliputi tes *code*, desain implementasi, *security*, *data flow*, *software failure*).
3. Dilakukan seiring dengan tahapan pengembangan *software* atau pada tahap *testing*.

Black Box (Fungsional)

1. Dilakukan oleh penguji *independent*.
2. Melakukan pengujian berdasarkan apa yang dilihat, hanya fokus terhadap fungsionalitas dan *output*. Pengujian lebih ditunjukkan pada desain *software* sesuai standar dan reaksi apabila terhadap celah-celah *bug/vulnerabilitas* program aplikasi tersebut setelah dilakukan *white box testing*.

2.18 Analisis *Pieces*

Menurut Ragil (2010), metode PIECES adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan

pelayanan pelanggan. Analisis ini disebut dengan PIECES Analysis (*Performance, Information, Economy, Control, Efficiency, and Service*).

Analisis PIECES ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variabel evaluasi, yaitu:

1. *Performance* (kinerja)

Kinerja merupakan variable pertama dalam metode analisis PIECES. Dimana memiliki peran penting untuk menilai apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya, dan melihat sejauh mana dan seberapa handalkah suatu sistem informasi dalam berproses untuk menghasilkan tujuan yang diinginkan. Dalam hal ini kinerja diukur dari *throughput*, yaitu jumlah pekerjaan/*output/deliverables* yang dapat dilakukan/ dihasilkan pada saat tertentu.*response time*, yaitu waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan *output/deliverables* tertentu.

2. *Information* (informasi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik. Informasi yang disajikan haruslah benar–benar mempunyai nilai yang berguna. Hal ini dapat diukur dengan :

Keluaran (outputs): Suatu sistem dalam memproduksi keluaran.

Masukan (inputs): Dalam memasukkan suatu data sehingga kemudian diolah untuk menjadi informasi yang berguna.

3. *Economic* (ekonomi)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau diturunkan biaya penyelenggarannya.

4. *Control* (pengendalian)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan/ kecurangan menjadi semakin baik pula.

5. *Efficiency* (efisiensi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi, dan harus lebih unggul dari pada sistem manual.

6. *Service* (layanan)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Buatlah kualitas layanan yang sangat *user friendly* untuk *end-user* (pengguna) sehingga pengguna mendapatkan kualitas layanan yang baik.

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian adalah tata cara bagaimana suatu penelitian dilaksanakan atau cara yang ditempuh sehubungan dengan penelitian yang dilakukan, dan memilih langkah-langkah sistematis. Metodologi penelitian mempelajari cara-cara melakukan pengamatan dengan pemikiran yang tepat secara terpadu melalui tahapan-tahapan yang disusun secara ilmiah untuk mencari, menyusun, serta menganalisis dan menyimpulkan data-data berdasarkan fakta-fakta secara ilmiah.

Untuk menghasilkan penelitian tugas akhir yang lebih lengkap diperlukan suatu metode dalam penelitian yang telah dipersiapkan sesuai dengan masalah yang akan dibahas.

3.2 Jenis dan Sumber Data

Data yang diperlukan dalam penelitian ini ada dua jenis, yaitu data primer dan data sekunder. Sumber dari data-data ini berasal dari tempat yang diamati untuk penelitian adalah di PT Teknikatama Karya Mandiri:

1. Data Primer

Data Primer adalah data yang diperoleh secara langsung dari PT Teknikatama Karya Mandiri, dimana pengumpulan data atau informasi dilakukan langsung dari objek yang diteliti. Data-data tersebut adalah data yang digunakan dalam proses penerimaan dan pengeluaran bahan baku diantaranya, analisis dokumen yang berjalan, proses bisnis sistem saat ini dan yang akan diusulkan, dan kebutuhan pengguna sistem.

2. Data Sekunder

Data sekunder adalah data yang diperoleh dari data yang tersedia dan telah terlebih dahulu dikumpulkan dan dilaporkan oleh orang lain, buku-buku dan

kajian ilmiah dari berbagai sumber yang berkaitan dengan penelitian. Data tersebut adalah data umum perusahaan, profil perusahaan, dan struktur organisasi perusahaan.

3.3 Metode Pengumpulan Data

Penelitian ini menggunakan beberapa teknik pengumpulan data, yaitu:

a. Studi Lapangan

Studi lapangan adalah usaha melakukan pengumpulan data secara langsung pada objek yang dilakukan dengan cara sebagai berikut:

1. Pengamatan, tahap ini dilakukan secara langsung pada departemen *warehouse* di PT Teknikatama Karya Mandiri dengan mengamati proses penerimaan dan pengeluaran bahan baku mulai dari bagian *warehouse* menerima dan mengecek bahan baku yang masuk, lalu mencatat arus bahan baku, dan bahan baku dikirim ke departemen produksi untuk proses pengolahan menjadi barang jadi.
2. Wawancara, Pengambilan data dengan cara berdialog dan bertanya dengan karyawan departemen *warehouse* tentang proses bisnis yang berjalan pada sistem penerimaan dan pengeluaran bahan baku. Sebagai data yang diperlukan untuk penyusunan tugas akhir.

b. Studi Kepustakaan

Tahap ini merupakan tahap dimana dilakukannya pengumpulan referensi dari sumber-sumber seperti buku yang berhubungan dengan masalah yang sedang dianalisis. Studi pustaka yang dilakukan berkaitan dengan penerimaan dan pengeluaran bahan baku, *warehouse*, pengembangan sistem, analisis dan desain sistem informasi, *CodeIgniter* dan *database* dari berbagai referensi, baik itu referensi elektronik yang didapat dari internet maupun referensi dari buku teks.

3.4 Metode Pengembangan Sistem

Metode pengembangan sistem informasi penerimaan dan pengeluaran bahan baku ini menggunakan metode pengembangan sistem *waterfall*. Metode *waterfall* ini terdiri dari tahap *planning*, *analysis*, *design*, dan *implementation*.

Berikut ini akan dijelaskan secara singkat mengenai tahapan-tahapan yang dilakukan dalam pengembangan sistem dengan metode *waterfall*.

1. Tahap Perencanaan (*Planning*)

Pada tahap ini dilakukan diskusi dengan operator *warehouse* mengenai perencanaan untuk membangun suatu sistem informasi dengan membuat sebuah *system request*.

2. Tahap Analisis (*Analysis*)

Menganalisis kebutuhan sistem dengan wawancara, observasi, dan membuat analisis permasalahan yang didapat dari tahap identifikasi masalah sebagai bahan pengembangan sistem.

3. Tahap Desain (*Design*)

Membuat pemodelan sistem dengan *Unified Modeling Language* (UML) yaitu menggunakan *usecase* diagram, *activity* diagram, *sequence* diagram dan *class* diagram dan membuat rancangan antarmuka. Data-data yang didapat dari tahap analisis diterapkan dalam tahap desain.

4. Tahap Implementasi (*Implementation*)

Pada tahap ini, pengembang akan membangun sebuah sistem dengan rancangan yang sudah ada diterjemahkan ke dalam sebuah *code* atau aktivitas *coding*. Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *framework* *CodeIgniter* dan MariaDB sebagai *database* yang digunakan.

3.5 Kerangka Penelitian

Dalam penelitian yang dibahas dalam Tugas Akhir ini, dilakukan langkah-langkah atau tahapan dalam penelitian yang ada pada Gambar III.1 sesuai dengan metodologi penelitian yang digunakan dalam penelitian tersebut. Penjelasan langkah-langkah atau tahapan dalam penelitian tersebut adalah sebagai berikut:

1. Studi Pendahuluan

Studi Pendahuluan merupakan langkah awal yang dilakukan untuk memulai penelitian. Studi pendahuluan dilakukan dengan turun langsung ke lokasi untuk mengetahui gambaran yang jelas mengenai sistem yang sedang

berjalan pada departemen *warehouse*. Studi dapat dilakukan dengan melakukan observasi, wawancara, dan studi pustaka.

2. Identifikasi Masalah

Pada tahap ini, peneliti mempersempit ruang penelitian dengan menganalisis sistem penerimaan dan pengeluaran bahan baku yang sedang berjalan lalu mengidentifikasi permasalahan yang ada pada sistem yang sedang berjalan dengan melakukan wawancara dan observasi. Ketika wawancara dilakukan, operator *warehouse* menyampaikan masukan jika sistem yang tengah berjalan belum terkomputerisasi sehingga aktivitas keluar/masuknya bahan baku kurang terkontrol dengan baik dan mengakibatkan stok bahan baku tidak terupdate.

3. Identifikasi Solusi

a. Maksud dan tujuan dari penelitian ini yakni:

Membuat rancangan bangun sistem informasi sehingga pencatatan stok bahan baku dapat terkontrol dengan baik. Membuat suatu sistem informasi dan *database* untuk aktivitas keluar/masuknya bahan baku yang mana data tersebut dapat membantu operator *warehouse* untuk membuat sebuah laporan dengan cepat.

4. Batasan Masalah

Pada tahap ini bertujuan untuk membatasi masalah agar penelitian lebih terarah sehingga dapat menunjukkan gambaran yang lebih spesifik mengenai arah pemecahannya. Berikut batasan dalam penulisan tugas akhir ini:

- a. Penelitian dilakukan selama satu bulan bertempat di PT Teknikatama Karya Mandiri Jalan Raya Bahkilong No.88 Cikarang Selatan – Bekasi 17330.
- b. Pengamatan dilakukan pada bagian departemen *warehouse*. Jangka waktu untuk penelitian ini, mulai dari 16 Juli 2018 sampai dengan 16 Agustus 2018.

5. Penerapan Metode *Waterfall*

a. Perencanaan (*Planning*)

Pada tahap ini akan dilakukan perencanaan untuk membangun suatu sistem informasi.

b. Melakukan Analisis (*Analysis*)

Pada tahap ini penulis melakukan identifikasi kebutuhan sistem dengan beberapa cara seperti:

1. Mengumpulkan data-data pada proses bisnis yang berjalan, serta menganalisis dokumen pada proses penerimaan dan pengeluaran bahan baku.
2. Identifikasi proses bisnis yang berjalan lalu menganalisis proses bisnis tersebut.
3. Identifikasi kebutuhan sistem informasi penerimaan dan pengeluaran bahan baku.

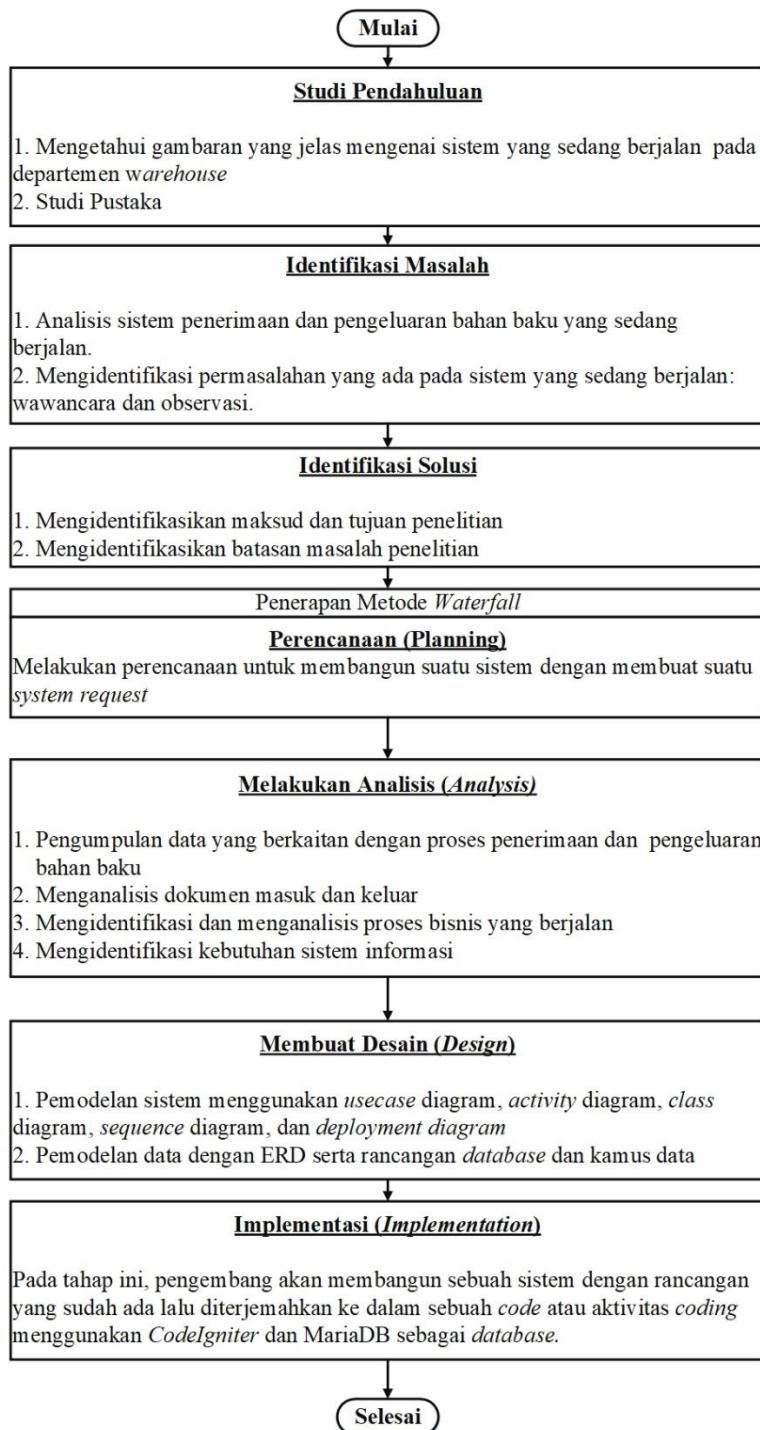
c. Membuat Desain (*Design*)

Membuat pemodelan sistem dengan *Unified Modeling Language* (UML) yaitu menggunakan *usecase* diagram, *activity* diagram, *sequence* diagram, *class* diagram, dan *deployment* diagram dan membuat rancangan antarmuka. Data-data yang didapat dari tahap analisis diterapkan dalam tahap desain.

d. Implementasi (*Implementation*)

Pada tahap ini, pengembang akan membangun sebuah sistem dengan rancangan yang sudah ada diterjemahkan ke dalam sebuah *code* atau aktivitas *coding*. Bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *framework* *CodeIgniter* dan MariaDB sebagai *database* yang digunakan.

diagram alir kerangka penelitian yang digunakan dalam laporan tugas akhir ini dapat dilihat pada Gambar III.1.



Gambar III.1 Kerangka Penelitian

(Sumber: Pengolahan Data, 2019)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sekilas Perusahaan

PT Teknikatama Karya Mandiri merupakan salah satu perusahaan kecil menengah yang bergerak dalam bidang industri otomotif yang memproduksi komponen untuk kendaraan roda empat seperti *pin shaft*, *gusset*, *end housing*, dan *reinforcement*.

PT Teknikatama Karya Mandiri didirikan oleh seorang pemilik perusahaan bernama Bapak Anton Saputra. Pertama kali didirikan di kawasan industri Jababeka, tepatnya di Jl. Jababeka III Blok C, No.45, kawasan industri Jababeka I, Cikarang Selatan - Bekasi. PT Teknikatama Karya Mandiri awalnya didirikan dengan bantuan tenaga kerja hanya 2 orang saja yaitu di bagian produksi.

Untuk awal pendirian PT Teknikatama Karya Mandiri memiliki 8 pelanggan. Pemilik memilih kawasan industri Jababeka sebagai tempat pendirian usaha dengan alasan untuk mendekati pelanggannya dan untuk meminimalisir biaya distribusi pengiriman barang kepada pelanggan. Untuk pendistribusian produk ke pelanggan, perusahaan memiliki kendaraan berupa mobil *pick up* ukuran kecil untuk proses pengiriman.

Seiring berjalananya waktu, pada maret 2016, PT Teknikatama Karya Mandiri akhirnya membeli gedung untuk pabrik dan kantor yang berada di Cikarang Selatan. Dan PT Teknikatama Karya Mandiri pun berpindah lokasi menjadi di Jalan Raya Bahkilong No.88, Cikarang Selatan-Bekasi sampai dengan saat ini. Hingga sampai saat ini PT Teknikatama Karya Mandiri sudah memiliki 22 karyawan untuk bagian produksi dan *staff* kantor. Sudah sejak lama pula PT Teknikatama Karya Mandiri menerapkan Sistem Manajemen Mutu ISO 9000:2008 yang menjamin segala bentuk kegiatan industri yang berlaku sesuai dengan standar sistem manajemen mutu ISO 9001:2008, dan tidak sedikit pula penghargaan sertifikat yang telah didapat oleh PT Teknikatama Karya Mandiri.

Mesin-mesin yang digunakan untuk produksi saat ini 99% sudah berteknologi CNC. Sehingga produk yang dihasilkan sangat mampu untuk bersaing dengan perusahaan pesaing otomotif lainnya. Ditambah dengan kemampuan *skill* karyawan yang terus didukung oleh perusahaan untuk diikutsertakan dalam pelatihan dan pengembangan.

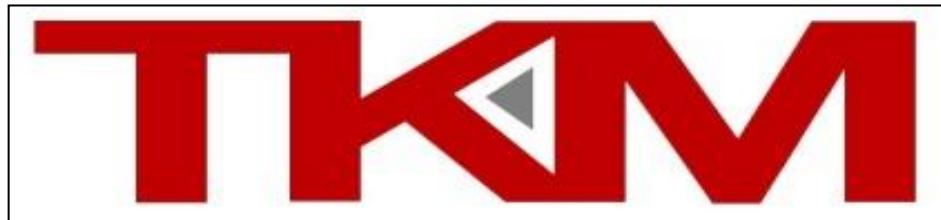
4.2 Profil Umum Perusahaan

Adapun profil usaha lebih rinci sebagai berikut:

| | | |
|-----------------|---|---|
| Nama perusahaan | : | PT Teknikatama Karya Mandiri |
| Nama pemilik | : | Anton Saputra |
| Jabatan | : | Direktur |
| Tahun berdiri | : | 2002 |
| Jenis Usaha | : | Manufaktur suku cadang kendaraan roda empat |
| Daerah kerja | : | Cikarang Selatan - Bekasi |
| Alamat usaha | : | Jalan Raya Bahkilong No. 88 Cikarang Selatan - Bekasi 17330 |
| Telepon | : | 62-21-22104845 |
| Fax | : | 62-21-22104846 |
| Nomor SIUP | : | 10261 – 03 / PK / 1.824.271 |
| Nomor TDP | : | 09.02.1.46.46396 |
| Nomor NPWP | : | 02.187.829.3-036.000 |
| Jumlah pekerja | : | 22 karyawan |
| Email | : | marketing@teknikatama.co.id. |

4.3 Logo Perusahaan

Logo simbol dari organisasi, kelompok, atau perorangan adalah hal yang mencerminkan makna atau pesan yang ingin disampaikan dari organisasi/kelompok tersebut. Logo untuk sebuah perusahaan juga sangatlah penting keberadaannya karena dengan logo, suatu perusahaan dapat dikenal oleh konsumen atau khalayak. Logo dari PT Teknikatama Karya Mandiri dapat dilihat pada Gambar IV.1.



Gambar IV.1 Logo PT Teknikatama Karya Mandiri

(Sumber: PT Teknikatama Karya Mandiri, 2018)

4.4 Visi dan Misi Perusahaan

PT Teknikatama Karya Mandiri sendiri mempunyai visi, misi, dan motto untuk perusahaannya. Adapun visi, misi dan motto tersebut adalah sebagai berikut:

Visi :

“Menjadi perusahaan *manufacturing* yang berkelas dunia dan pemimpin pasar di Indonesia”.

Misi :

Menjadi industri pendukung penghasil part OEM, untuk industri otomotif yang dapat dipercaya dengan cara :

1. Menghasilkan produk yang berkualitas tinggi melalui sistem manajemen mutu.
2. Pengembangan sumber daya manusia yang profesional dan inovatif.
3. Mensejahterakan karyawan dan memberikan nilai tambah kepada setiap *stakeholder*.

Motto :

TEAM WORK, KUALITAS DAN MANFAAT.

TEAM WORK

Kami menyadari bahwa tanpa adanya *team work*, kami tidak akan menghasilkan produk yang baik.

KUALITAS

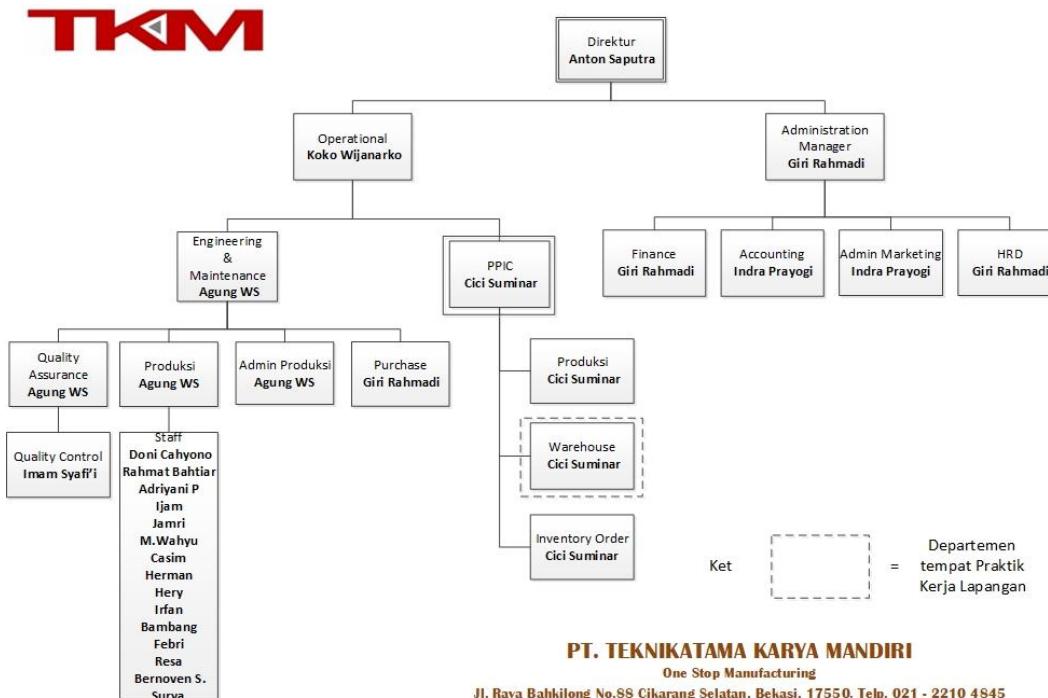
Kami sepakat menghasilkan produk dengan kualitas terbaik

MANFAAT

Kami selalu berusaha agar dimanapun kami berada, kami dapat memberikan mafaat bagi orang lain.

4.5 Struktur Organisasi Perusahaan

Setiap organisasi atau perusahaan pasti mempunyai struktur organisasi yang ada didalamnya untuk saling berkoordinasi mengurus jalannya suatu perusahaan dalam mencapai tujuan. Adapun struktur dari PT Teknikatama Karya Mandiri dapat dilihat pada Gambar IV.2. Keterangan :



Gambar IV.2 Struktur Organisasi Perusahaan

(Sumber: PT Teknikatama Karya Mandiri, 2018)

4.5.1 Tugas dan Wewenang Jabatan di PT Teknikatama Karya Mandiri

Setiap perusahaan pada umumnya pasti mempunyai sebuah struktur organisasi. Struktur organisasi tersebut sangatlah berguna untuk melaksanakan fungsi perencanaan, pengorganisasian, pengarahan, dan pengawasan. Dengan kata lain, struktur organisasi menjelaskan tentang wewenang, tugas, dan tanggung jawab

seseorang yang berada pada setiap departemen. Dan berikut ini adalah gambaran mengenai tugas dan tanggung jawab masing-masing bagian atau departemen yang berada di PT Teknikatama Karya Mandiri:

1. Departemen Produksi dan PPIC *Job Order* :

- a. Bertugas untuk mempersiapkan, merencanakan dan melakukan produksi yang bersifat *Job Order* (pesanan tertentu).
- b. Membangun kerjasama yang baik antar bagian produksi.
- c. Bertanggungjawab terhadap proses produksi.
- d. Mengatur jalannya operasional sehingga berjalan dengan lancar.
- e. Mengawasi proses produksi.
- f. Mencatat seluruh jadwal produksi pengiriman barang.

2. Departemen Produksi dan PPIC *Mass Production* :

- a. Bertugas untuk mempersiapkan, merencanakan, dan melakukan produksi pekerjaan yang bersifat massal.
- b. Membangun kerjasama yang baik antar bagian produksi.
- c. Bertanggungjawab terhadap proses produksi.
- d. Mengatur jalannya operasional sehingga berjalan dengan lancar.
- e. Mengawasi proses produksi.

3. Departemen *Warehouse* :

- a. Membuat catatan persediaan.
- b. Menyiapkan barang yang akan dikirimkan kepada *customer* berdasarkan surat jalan.
- c. Bertanggungjawab atas kebersihan dan kerapihan di dalam area gudang barang.
- d. Memproses permintaan bahan baku yang diminta oleh departemen produksi untuk diolah.
- e. Mengontrol dan bertanggungjawab atas penyimpanan barang yang ada di gudang.

- f. Bertanggungjawab atas pelaksanaan bongkar muat barang di gudang dan menandatangi surat penerimaan barang dan surat jalan.
- g. Merapikan setiap penempatan barang yang ada di gudang berdasarkan kelompok barang dengan baik dan teratur.

4. Departemen *Purchasing* :

- a. Bertugas sebagai penyedia barang dan jasa agar proses produksi berjalan dengan baik.
- b. Melakukan seleksi *supplier* untuk didaftarkan ke daftar *supplier* terpilih.
- c. Membuat *purchase order* dan pembelian langsung dari permintaan pembelian.
- d. Melakukan *monitoring* pemenuhan pengadaan barang dan jasa dari *supplier*.
- e. Menginformasikan pembayaran dari tagihan kepada *supplier* ke Bagian Keuangan.

5. Departemen *Engineering & Maintenance*

- a. Bertugas untuk menjamin kondisi mesin dapat beroperasi dengan baik.
- b. Membuat jadwal mesin produksi

4.6 Daftar Fasilitas Mesin Produksi

Daftar mesin yang digunakan untuk memproduksi barang mentah menjadi barang jadi yang siap dipasarkan ke konsumen dapat dilihat pada Tabel IV.1.

Tabel IV.1. Daftar Mesin Produksi PT Teknikatama Karya Mandiri

| No | Type | Maker | Capacity | QTY |
|----|----------------------|------------------------|--------------------|--------|
| 1 | CNC Machining Center | Kiwa Triple V 2li-S | 610 x 410 x 510 mm | 1 unit |
| 2 | CNC Machining Center | Brother Speedio S500Xi | 500 x 420 x 320 mm | 1 unit |
| 3 | CNC Machining Center | OKK RPM-3V | 380 x 600 x 300 mm | 1 unit |

| No | Type | Maker | Capacity | QTY |
|----|----------------------|-----------------------|--------------------|--------|
| 4 | CNC Machining Center | Mill Seiki MV-400 | 300 x 600 x 300 mm | 1 unit |
| 5 | CNC Machining Center | Tongtai Titan TMV 400 | 250 x 400 x 600 mm | 1 unit |
| 6 | CNC Lathe | Mazak QTS-200 | Ø 280'' x 200 mm | 1 unit |
| 7 | CNC Lathe | Headman HCL 300 | Ø 6'' x 300 mm | 1 unit |
| 8 | CNC Lathe | Headman CK 6232 | Ø 8'' x 300 mm | 1 unit |
| 9 | CNC Lathe | Headman CK 6632 | Ø 8 x 300 mm | 1 unit |
| 10 | CNC Lathe | Senday SDC-32T | Ø 4'' x 190 mm | 1 unit |
| 11 | Milling Machine | POLO S-5 | 254 x 1270 mm | 1 unit |
| 12 | Lathe Machine | Lei Shin LA 530 | Ø 300 x 1270 mm | 1 unit |
| 13 | Surface Grinding | Proth PSGS 3060 AH | 300 x 600 mm | 1 unit |
| 14 | Welding Machine | OTC | 350 A | 1 unit |
| 15 | Hardening Oven | Naberthem | 300 x 300 x 300 | 1 unit |

(Sumber: PT Teknikatama Karya Mandiri, 2018)

4.7 Bahan Baku Dalam Proses Produksi dan Supplier

PT Teknikatama Karya Mandiri mendapatkan bahan baku (*raw materials*) dari pemasoknya sebelum diolah. Jenis bahan baku untuk *Mass Production* PT Teknikatama Karya Mandiri berasal dari S45C. S45C adalah baja medium dengan kandungan *carbon* 0.45%. Banyak digunakan dalam manufaktur mesin, seperti pembuatan komponen mekanis. Berikut ini adalah produk yang dihasilkan oleh PT Teknikatama Karya Mandiri seperti pada Tabel IV.2. Sedangkan nama-nama Supplier ada pada Tabel IV.4.

Tabel IV.2. Produk *Mass Production* PT Teknikatama Karya Mandiri

| No . | Nama Produk | Gambar Bahan Baku | Gambar Produk |
|------|----------------------|---|---|
| 1. | <i>Gusset</i> |  |  |
| 2. | <i>Reinforcement</i> |  |  |
| 3. | <i>Pin Shaft</i> |  |  |

| No | Nama Produk | Gambar Bahan Baku | Gambar Produk |
|----|------------------------|--|--|
| 4. | <i>Connector Joint</i> |  |  |
| 5. | <i>End Housing</i> |  |  |

(Sumber: PT Teknikatama Karya Mandiri, 2018)

Tabel IV.3. Produk *Job Order* PT Teknikatama Karya Mandiri

| No. | Nama Produk | Gambar Produk | Nama Bahan Baku |
|-----|---------------------------------|--|-----------------|
| 1. | <i>Cooling Pipe 650 dan 350</i> |  | Pipa Tembaga |

| No. | Nama Produk | Gambar Produk | Nama Bahan Baku |
|-----|-------------------------|--|-----------------|
| 2. | <i>Special Tool</i> |  | S45C |
| 3. | <i>Slodgin Rod 1800</i> |  | Asental Stenlis |
| 4. | <i>Hex Shock</i> |  | Asental SS400 |

(Sumber: PT Teknikatama Karya Mandiri, 2018)

Tabel IV.4. *Supplier* PT Teknikatama Karya Mandiri

| No | Nama Bahan Baku | <i>Supplier</i> |
|----|-----------------|---|
| 1 | S45C | PT Gemala Kempa Daya PT Stillmetindo Prima |
| 2 | Pipa Tembaga | PT Allindo Coin Emas |
| 3 | Asental Stenlis | PT Sinar Jaya Steel |
| 4 | Asental SS400 | Citra Baja |

(Sumber: PT Teknikatama Karya Mandiri, 2018)

4.8 Jaringan Pemasaran

Jaringan pemasaran adalah seseorang atau kelompok yang membeli produk yang telah dihasilkan. Atau dapat juga disebut dengan *customer*/pembeli dari produk tersebut.

PT Teknikatama Karya Mandiri sendiri memasarkan produknya ke perusahaan-perusahaan besar yang terdapat didalam negeri. Adapun nama-nama dari perusahaan tersebut dapat dilihat pada Tabel I.4.

Tabel IV.5 *Customer*

| No | Nama <i>Customer</i> |
|----|----------------------------------|
| 1. | PT Gemala Kempa Daya |
| 2. | PT Inti Ganda Perdana |
| 3. | PT Astra Honda Motor |
| 4. | PT Akebono Brake Astra Indonesia |
| 5. | PT Wahana Eka Paramitha |
| 6. | PT Indomobil Suzuki Motor |
| 7. | PT Showa Indonesia Manufacturing |
| 8. | PT Indopoly Swakarsa Mandiri |

(Sumber: PT Teknikatama Karya Mandiri, 2018)

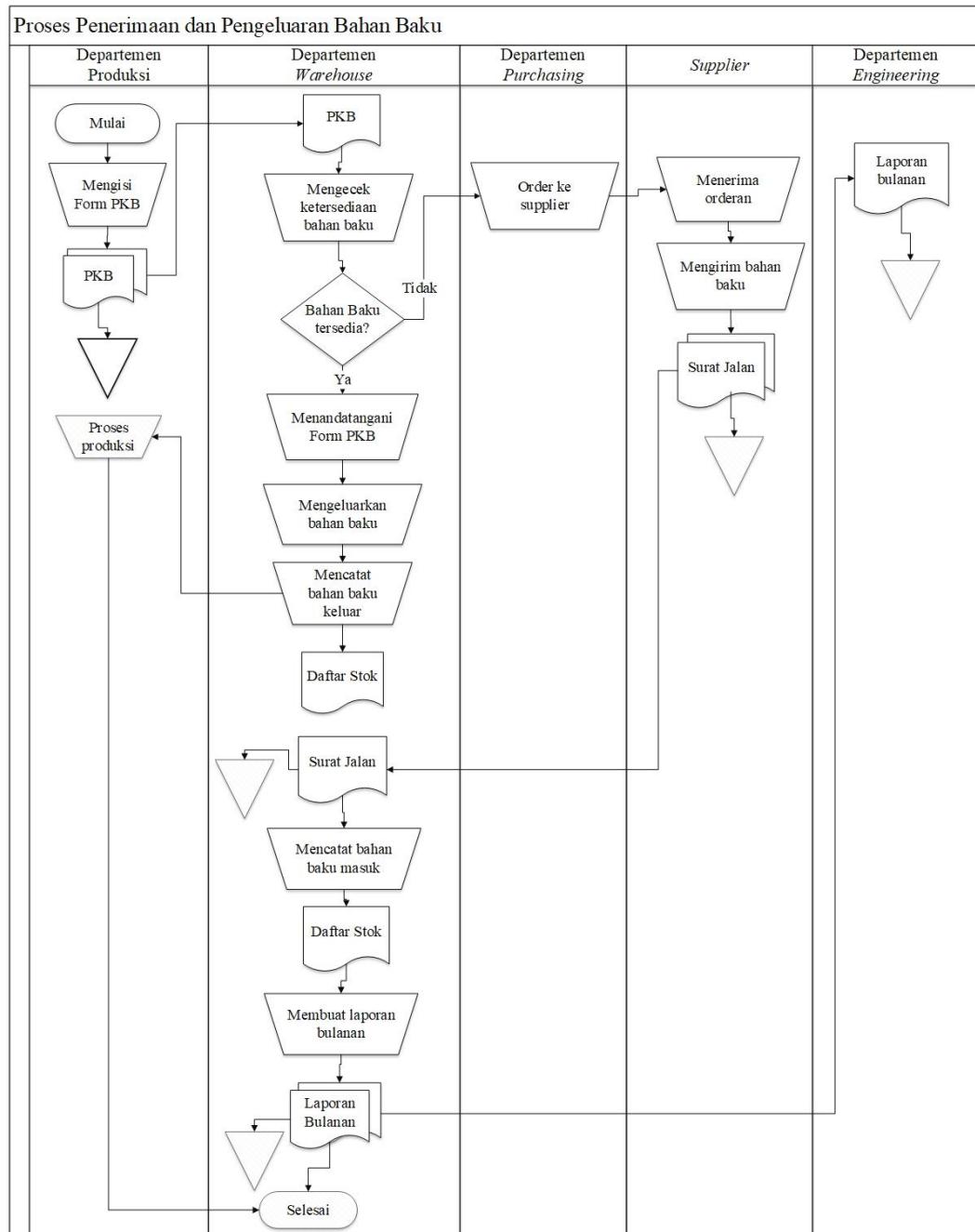
4.9 Prosedur Penerimaan dan Pengeluaran Bahan Baku

Penerimaan bahan baku adalah kegiatan penerimaan bahan baku berdasarkan nota. Pengeluaran bahan baku merupakan pengeluaran bahan baku sesuai dengan daftar permintaan bahan baku dari departemen yang membutuhkan.

Dalam proses penerimaan dan pengeluaran bahan baku yang ada di PT Teknikatama Karya Mandiri, dapat diurutkan sebagai berikut ini:

1. Adanya permintaan bahan baku dari departemen produksi untuk kebutuhan memproduksi barang.
2. Petugas dari departemen produksi mengisi dan menyerahkan lembar PKB (Permintaan Kebutuhan Barang) kepada operator *warehouse*.
3. Operator *warehouse* akan mengecek ketersediaan bahan baku yang diminta.
4. Jika bahan baku tersedia, operator *warehouse* akan menyetujui surat PKB dari departemen produksi.
5. Jika bahan baku yang diminta tidak tersedia atau kurang dari titik minimum, maka operator *warehouse* akan menghubungi departemen *purchasing*.
6. Departemen *purchasing* akan menghubungi pihak *supplier* untuk melakukan pengiriman bahan baku.
7. Pihak *supplier* mengirimkan bahan baku yang dipesan dan diterima oleh departemen *warehouse* disertai dengan surat jalan. Selanjutnya departemen *warehouse* akan melakukan *update* pada kartu stok gudang.
8. Bahan baku disiapkan dan dikirim ke departemen produksi untuk proses produksi menjadi barang jadi.
9. Operator *Warehouse* akan merekap arus bahan baku masuk dan keluar selama satu bulan dan diserahkan ke Departemen *Engineering* sebagai dasar perencanaan *load* mesin.

Berikut ini adalah penggambaran prosedur penerimaan dan pengeluaran bahan baku dengan menggunakan *flowmap* dapat dilihat pada Gambar IV.3.



Gambar IV.3 Proses Penerimaan dan Pengeluaran Bahan Baku

(Sumber: Hasil Analisis, 2019)

Untuk tata letak barang di gudang, dapat dilihat pada Gambar IV. 4.



Gambar IV.4 Tata Letak Barang di Gudang PT Teknikatama Karya Mandiri

(Sumber: PT Teknikatama Karya Mandiri, 2018)

4.9.1 Dokumen pada Penerimaan dan Pengeluaran Bahan Baku yang Berjalan

Dalam proses penerimaan dan pengeluaran bahan baku, tentunya terdapat dokumen-dokumen yang berperan untuk membantu Operator *Warehouse*. Dokumen yang terlibat pada Departemen *Warehouse* ini tidak hanya satu dokumen. Tetapi ada beberapa macam dokumen. Berikut ini adalah dokumen-dokumen yang terlibat dalam sistem penerimaan dan pengeluaran bahan baku:

1. Form Permintaan Kebutuhan Barang

Form permintaan kebutuhan barang ini digunakan ketika Departemen Produksi meminta bahan baku dari gudang. Jika Operator *Warehouse* sudah menandatangani dan menyetujui form permintaan kebutuhan barang, maka bahan baku yang ada di gudang dapat langsung dikirim untuk diproduksi. Untuk keterangan yang ada dalam form permintaan kebutuhan barang adalah sebagai berikut:

- Dept : Untuk mencatat nama Departemen yang mengajukan PKB
- Tgl permintaan : Untuk mencatat tanggal saat membuat form

PKB

- No. PKB : Nomor urut dari form permintaan kebutuhan Bahan baku
- NO : Untuk menuliskan nomor urut
- Nama barang : Untuk mencatat nama dari bahan baku yang diminta
- *Qty* : Untuk mencatat jumlah bahan baku yang diminta
- WO : Untuk menuliskan kode nomor WO
- Tgl dibutuhkan : Untuk mencatat kapan bahan baku dibutuhkan
- *Note* : Untuk catatan tambahan
- Diterima : Ditandatangani oleh yang menerima form PKB yaitu Departemen *Warehouse*
- Disetujui : ditandatangani oleh pihak yang menyetujui adanya permohonan form PKB yaitu departemen *warehouse*
- Dibuat : ditandatangani oleh pihak yang mengajukan form permintaan kebutuhan barang

Satu lembar akan disimpan oleh operator *warehouse*, sementara satu lembar lainnya akan disimpan oleh Departemen atau pihak yang mengajukan form tersebut. Untuk gambar dari form permintaan kebutuhan barang dapat dilihat pada Gambar IV.5.



PT.TEKNIKATAMA KARYA MANDIRI

PERMINTAAN KEBUTUHAN BARANG

Dept. :
Tgl Permintaan :

No. PKB :

FR-MPP-08-Rev 00

Gambar IV.5 *Form* Permintaan Kebutuhan Barang

(Sumber: PT Teknikatama Karya Mandiri, 2018)

2. Daftar Stok

Daftar stok ini digunakan oleh Operator *Warehouse* untuk mencatat arus bahan baku yang keluar maupun masuk ke dalam gudang. Seperti untuk mencatat berapa jumlah bahan baku yang keluar, berapa jumlah bahan baku yang masuk, berapa stok bahan baku terakhir di gudang. Berikut adalah keterangan yang terdapat dalam daftar stok yang digunakan oleh Operator *Warehouse* untuk mencatat stok dalam gudang:

- Nama Barang : untuk mencatat nama bahan baku yang disimpan dalam gudang
 - No.Part : untuk mencatat nomor dari bahan baku tersebut
 - Tanggal : kolom tanggal untuk mencatat kapan tanggal aktivitas keluar-masuk bahan baku tersebut terjadi
 - No. Bon : diambil dari nomor surat jalan
 - Qty : kolom *qty* masuk, dimaksudkan untuk mencatat berapa banyak bahan baku yang masuk. Kolom *qty*

keluar dan *user*, dimaksudkan untuk mencatat berapa banyak bahan baku yang keluar dan ditujukan ke siapa bahan baku tersebut keluar

- Stok : stok digunakan untuk mencatat berapa stok bahan baku yang ada dalam gudang

Untuk lebih jelasnya, gambar dari daftar *stock* gudang dapat dilihat pada Gambar IV.6.barang yang masuk. Kolom *qty* keluar dan *user*, dimaksudkan untuk mencatat

Gambar IV.6 Daftar Stok

(Sumber: PT Teknikatama Karya Mandiri, 2018)

3. Form Surat Jalan dari Pemasok/*Supplier*

Form surat jalan dari pemasok ini berisi kejelasan dan rincian terhadap bahan baku yang dikirimkan kepada PT TKM. Untuk keterangan yang ada dalam form surat jalan, adalah sebagai berikut:

- Dikirim ke : Untuk mencatat nama *customer* yang menerima bahan baku
 - Tanggal : Untuk mencatat tanggal saat bahan baku dikirimkan
 - Kendaraan : Untuk mencatat jenis kendaraan yang

- digunakan
- No : Untuk menuliskan nomor urut
 - Kode : Untuk menuliskan kode bahan baku yang dikirimkan
 - Nama Barang : Untuk menuliskan nama bahan baku yang dikirimkan
 - Jumlah : Untuk mencatat jumlah bahan baku yang dipesan
 - Keterangan : Untuk menambahkan pesan lain
 - *Security* : Tanda tangan petugas keamanan yang ada saat proses keluarnya barang
 - Sopir : Tanda tangan sopir yang mengirimkan bahan baku
 - Penerima : Tanda tangan *customer*
 - Pengirim : Tanda tangan yang mengirimkan (*supplier*)

Untuk gambar dari form surat jalan dari *supplier*, dapat dilihat pada Gambar IV.7.

| PT. GEMALA KEMPA DAYA | | | | | | | | | | | | |
|--|---|---|---|---|----------|-------|----------|----------|---|---|---|---|
| Jl. Raya Pegangsaan Dua Blok A1, Pulogadung Jakarta-Indonesia, PO Box 1038 / Jat, Jakarta 13065 Phone : 460 2755 (Hunting System) Telek : 46580GEMALA IA, Fax : 460 2765 Asal : (Bogor) alih Jakarta Tujuan : Bandung TKM tanggal : 13-DEC-18 / 13:50 Kendaraan : 09973 FAN | | | | | | | | | | | | |
| SURAT PENGANTAR | | | | | | | | | | | | |
| NO : 0280/PAG/12/2018 No. : 00827/000/12/2018  | | | | | | | | | | | | |
| NO | KODE | NAMA BARANG | JUMLAH | KETERANGAN | | | | | | | | |
| 1 | 173102260 | STICK SHEET # HTP 540 12.0 X 97 X 133 (GRIS MET L/H PE-TD) | 600 SHEET | 21226/2E/12/2018(360), 31227/2E/12/2018(360) | | | | | | | | |
| Note : Lembar pertama harap dikembalikan ke GKD * Nama Terang 1. File | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Security</th> <th>Sopir</th> <th>Penerima</th> <th>Pengirim</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> | | | | | Security | Sopir | Penerima | Pengirim |  |  |  |  |
| Security | Sopir | Penerima | Pengirim | | | | | | | | | |
|  |  |  |  | | | | | | | | | |

Gambar IV.7 Form Surat Jalan dari *Supplier*

(Sumber: PT Teknikatama Karya Mandiri, 2018)

4.10 Penggambaran Sistem Penerimaan dan Pengeluaran Bahan Baku dengan UML (*Unified Modeling Language*)

Berdasarkan Mulyani (2016), *Unified Modelling Language* (UML) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem. Berdasarkan Dennis, Wixom, dan Tegarden (2015), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu *Structure diagram* dan *Behavior diagram*. Merepresentasikan sebuah sistem yang tengah berjalan ke dalam UML melalui diagram *usecase*, dimaksudkan agar memudahkan *user* mengerti akan jalannya sistem tersebut.

4.10.1 *Usecase Diagram*

Diagram *use case* dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam diagram *use case* ini adalah “apa” yang dapat diperbuat oleh sistem, dan bukan “bagaimana” sistem melakukannya. Diagram *use case* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut.

Dalam sistem penerimaan dan pengeluaran bahan baku ini, terdapat empat aktor. Antara lain adalah operator produksi, operator *warehouse*, divisi *purchasing* dan *supplier*. Adapun detail dari analisis *usecase* diagram ini, adalah sebagai berikut:

a. Definisi Aktor

Pendefinisian aktor pada *use case diagram* penerimaan dan pengeluaran bahan baku yang sedang berjalan pada departemen *warehouse* PT Teknikatama Karya Mandiri dapat dilihat pada Tabel IV.6 berikut:

Tabel IV.6 Definisi Aktor

| No | Aktor | Definisi |
|----|---------------------|--|
| 1. | Departemen Produksi | <ul style="list-style-type: none"> • Mengisi Form PKB (Permintaan Kebutuhan Barang) |

| No | Aktor | Definisi |
|----|----------------------------------|--|
| | | <ul style="list-style-type: none"> • Melakukan proses produksi |
| 2. | Departemen <i>Warehouse</i> | <ul style="list-style-type: none"> • Mengecek ketersediaan bahan baku • Menerima dan menyetujui <i>form</i> kebutuhan barang • Mengeluarkan bahan baku dari gudang • Mencatat aliran bahan baku keluar dan masuk serta jumlah <i>stock</i> dalam gudang • Membuat Laporan Bulanan |
| 3 | Departemen <i>Purchasing</i> | Memesan atau membeli bahan baku kepada <i>Supplier</i> apabila bahan baku yang dibutuhkan untuk proses produksi tidak tersedia |
| 4 | <i>Supplier</i> | <ul style="list-style-type: none"> • Menerima orderan • Mengirim bahan baku yang telah dipesan oleh divisi <i>purchasing</i>. |
| 5 | Departemen <i>Engineering</i> | Melihat laporan untuk dapat mengetahui aktivitas barang yang keluar atau masuk ke dalam gudang sekaligus untuk perencanaan <i>loading</i> mesin. |

(Sumber: Hasil Analisis, 2019)

b. Definisi *Use Case Diagram*

Penggambaran *Unified Modeling Language* (UML) Sistem Penerimaan dan Pengeluaran Bahan Baku pada PT Teknikatama Karya Mandiri dengan menggunakan *Use Case Diagram* dapat dilihat pada Gambar IV.7 dan Definisi *Use Case diagram* dapat dilihat pada Tabel IV.7.

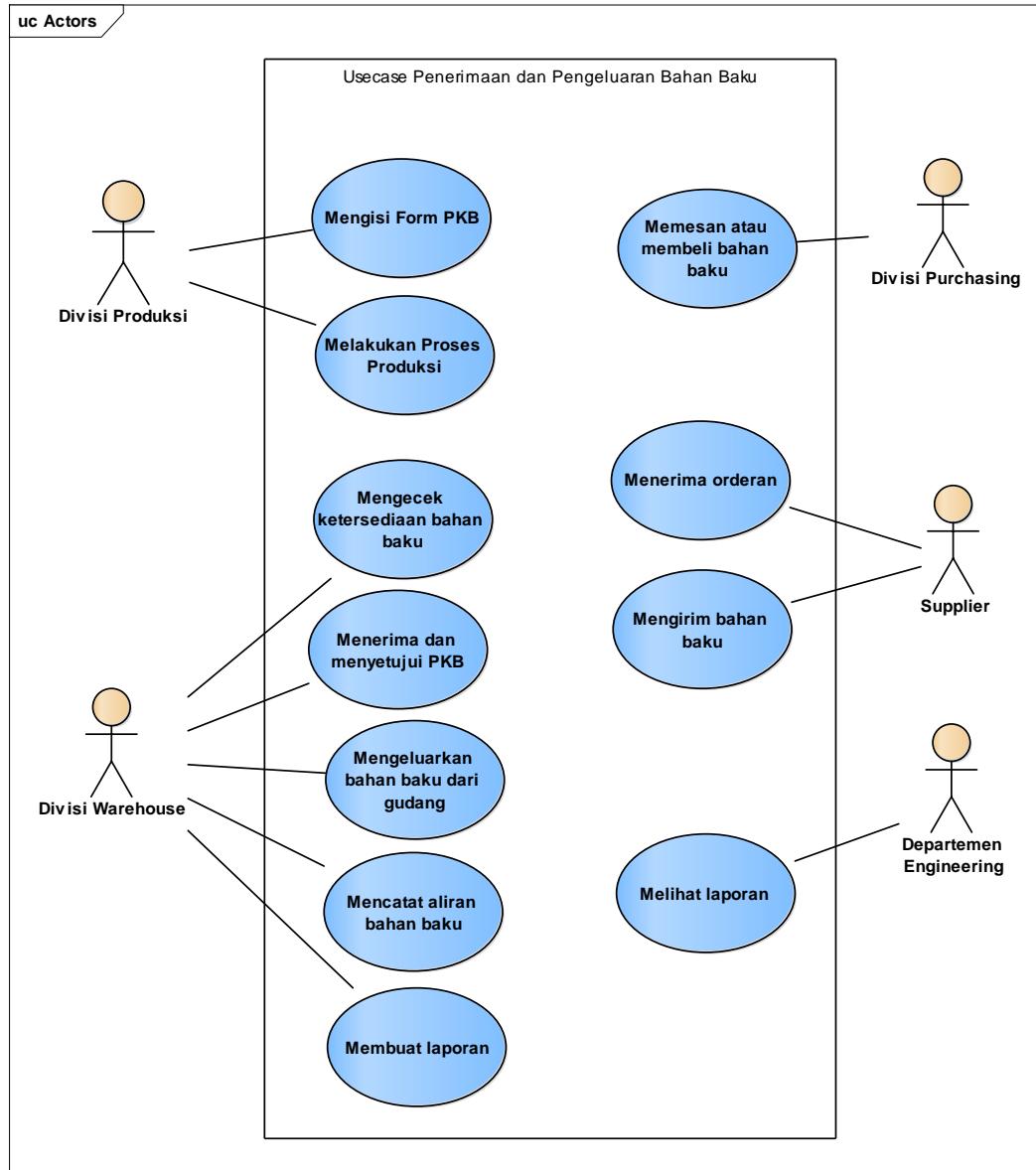
Tabel IV.7 Definisi *Use Case Diagram*

| No | <i>Use Case</i> | Deskripsi |
|----|---|--|
| 1. | Mengisi form permintaan kebutuhan barang | Hal ini dilakukan oleh Operator Produksi guna meminta bahan baku yang ada dalam gudang kepada operator <i>warehouse</i> |
| 2. | Melakukan proses produksi | Proses dimana bahan baku diolah menjadi barang jadi |
| 3. | Mengecek ketersediaan bahan baku | Hal ini dilakukan untuk memastikan apakah bahan baku yang diminta oleh departemen produksi tersedia atau tidak |
| 4. | Menerima dan menyetujui form kebutuhan barang | Proses dimana operator <i>warehouse</i> menandatangani form PKB dan menyetujui permintaan bahan baku |
| 5. | Mengeluarkan bahan baku dari gudang | Setelah memastikan bahan baku yang dibutuhkan tersedia di gudang, bahan baku tersebut dikirim ke departemen produksi |
| 6. | Mencatat aliran bahan baku | Proses dimana operator <i>warehouse</i> mencatat bahan baku yang masuk dan bahan baku yang keluar serta jumlah stok yang ada di gudang |
| 7. | Melakukan pemesanan atau pembelian bahan baku | Hal ini dilakukan apabila bahan baku yang diminta oleh departemen produksi tidak tersedia di gudang |
| 8. | Menerima Orderan | Proses dimana supplier menerima list bahan baku untuk dikirimkan |

| No | Use Case | Deskripsi |
|-----------|--|--|
| 9. | Mengirim bahan baku yang telah dipesan | Proses ini dilakukan oleh <i>supplier</i> ketika divisi <i>purchasing</i> telah memesan bahan baku untuk proses produksi |
| 10 | Membuat Laporan Bulanan | Proses ini dilakukan oleh operator <i>warehouse</i> untuk merekap arus bahan baku masuk dan keluar selama satu bulan. |
| 11 | Melihat laporan bulanan | Proses ini dilakukan oleh Departemen <i>engineering</i> setelah operator <i>warehouse</i> membuat laporannya. |

(Sumber: Hasil Analisis, 2019)

Berikut adalah *Usecase* Penerimaan dan Pengeluaran bahan baku pada Departemen *warehouse*:



Gambar IV.8 *Usecase* Diagram Penerimaan dan Pengeluaran Bahan Baku

(Sumber: Hasil Analisis, 2019)

4.11 Analisis *Pieces*

Analisis *Pieces* adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan pelayanan pelanggan.

Analisis ini disebut dengan *Pieces Analysis (Performance, Information, Economy, Control, Efficiency and Service)*.

Analisis *Pieces* ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variable evaluasi yaitu :

1. *Performance* (Kinerja)
2. *Information* (Informasi)
3. *Economic* (Ekonomi)
4. *Control* (Pengendalian)
5. *Efficiency* (Efisiensi)
6. *Service* (Layanan)

Berikut ini adalah Analisis *Pieces* untuk penerimaan dan pengeluaran bahan baku yang ada pada Departemen *Warehouse* PT Teknikatama Karya Mandiri dapat dilihat pada Tabel IV.8.

Tabel IV.8 Analisis *Pieces* Penerimaan dan Pengeluaran Bahan Baku

| Analisis | Uraian Masalah | Akibat | Sebab | Solusi |
|-------------------|--|--|--|--|
| <i>Perfomance</i> | minimnya informasi detail bahan baku yang keluar/masuk dalam gudang. | Stok bahan baku yang tidak terupdate, mengakibatkan proses persiapan bahan baku yang lama karena adanya proses pengecekan stok terlebih dahulu dan mengganggu proses produksi. | Daftar stok bahan baku sering hilang | Pencatatan daftar stok dilakukan secara terkomputerisasi dan disimpan ke dalam basis data/ <i>database</i> . |

| Analisis | Uraian Masalah | Akibat | Sebab | Solusi |
|--------------------|---|---|--|---|
| <i>Information</i> | laporan bulanan masih menggunakan <i>Microsoft excel</i> dan belum terintegrasi | operator <i>warehouse</i> harus bekerja dua kali dan membutuhkan waktu yang cukup lama. | operator <i>warehouse</i> harus membuka beberapa daftar stok kemudian dilakukan input ulang saat melakukan rekap data. | Pembuatan laporan bulanan dibuat secara terkomputerisasi. Data yang ada diambil berdasarkan data stok, data transaksi bahan bakukeluar dan masuk. |

Sumber: Hasil Analisis (2019)

BAB V

HASIL ANALISIS DAN PEMBAHASAN

5.1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem untuk sistem informasi penerimaan dan pengeluaran bahan baku dapat dilihat pada Tabel V.1 sebagai berikut:

Tabel V.1 Analisis Kebutuhan Sistem Usulan

| No. | Permasalahan | Kebutuhan <i>User</i> | Solusi | Kebutuhan Sistem | |
|-----|--|--|--|---|--|
| | | | | <i>Functional Requirement</i> | <i>Non- Functional Requirement</i> |
| 1. | Data stok bahan baku yang tidak terupdate sehingga mengganggu proses produksi | Sistem terkomputerisasi yang dapat membantu mengelola proses keluar masuk bahan baku. | Merancang dan membangun suatu sistem yang dapat membantu proses pengelolaan stok bahan baku. | Sistem dapat mengelola stok bahan baku. | Transaksi keluar dan masuk wajib diisi. |
| 2. | Belum adanya <i>database</i> untuk penyimpanan dokumen yang mengakibatkan terjadinya kerusakan dokumen, <i>redundancy</i> data, atau | Sistem dengan <i>database</i> untuk menghindari kemungkinan kerusakan dokumen, <i>redundancy</i> | Perancangan <i>database</i> sistem informasi proses penerimaan dan pengeluaran bahan baku agar tidak | Sistem dapat mengelola data bahan baku dengan nama bahan baku, satuan, stok, stok minimum dengan aksi simpan, | Mengelola data bahan baku dengan nama bahan baku, satuan, stok, stok minimum dengan aksi simpan, |

| No. | Permasalahan | Kebutuhan <i>User</i> | Solusi | Kebutuhan Sistem | |
|-----|--|---|--|---|--|
| | | | | <i>Functional Requirement</i> | <i>Non- Functional Requirement</i> |
| | kehilangan dokumen, dan sulitnya pencarian dokumen. | data, atau kehilangan | kesulitan dalam pencarian data dan mencegah kerusakan dokumen, <i>redundancy</i> data, atau kehilangan dokumen. | | tambah, ubah, hapus. |
| 3. | Pembuatan laporan bulanan masih menggunakan <i>Microsoft excel</i> . Hal tersebut mengakibatkan operator <i>warehouse</i> harus bekerja dua kali dan membutuhkan waktu yang cukup lama | Mencetak laporan bulanan melalui sistem berbasis <i>web</i> . | Membuat sistem dengan <i>output</i> laporan bahan baku. | Sistem mampu mencetak laporan bahan baku. | Laporan bahan baku yang dicetak diantaranya adalah laporan transaksi masuk dan keluar. |
| 4. | Transaksi permintaan pembelian bahan baku | Adanya sistem yang dapat melakukan | Merancang suatu sistem yang dapat membantu | Sistem menyediakan form permintaan | Transaksi bahan baku harus diisi sebagai bukti |

| No. | Permasalahan | Kebutuhan <i>User</i> | Solusi | Kebutuhan Sistem | |
|-----|--|--|--|--|---|
| | | | | <i>Functional Requirement</i> | <i>Non-Functional Requirement</i> |
| | dilakukan secara lisan oleh departemen <i>warehouse</i> kepada departemen <i>purchasing</i> dan tidak ada dokumen. | transaksi permintaan pembelian bahan baku yang dapat disimpan dan dapat dilihat sebagai bukti transaksi. | dalam proses permintaan pembelian bahan baku yang dilengkapi dengan aksi simpan, ubah, hapus, dan detail | pembelian bahan baku yang dilengkapi dengan aksi simpan, ubah, hapus, dan detail | transaksi dan memudahkan informasi untuk daftar stok. |

(Sumber: Hasil Analisis, 2019)

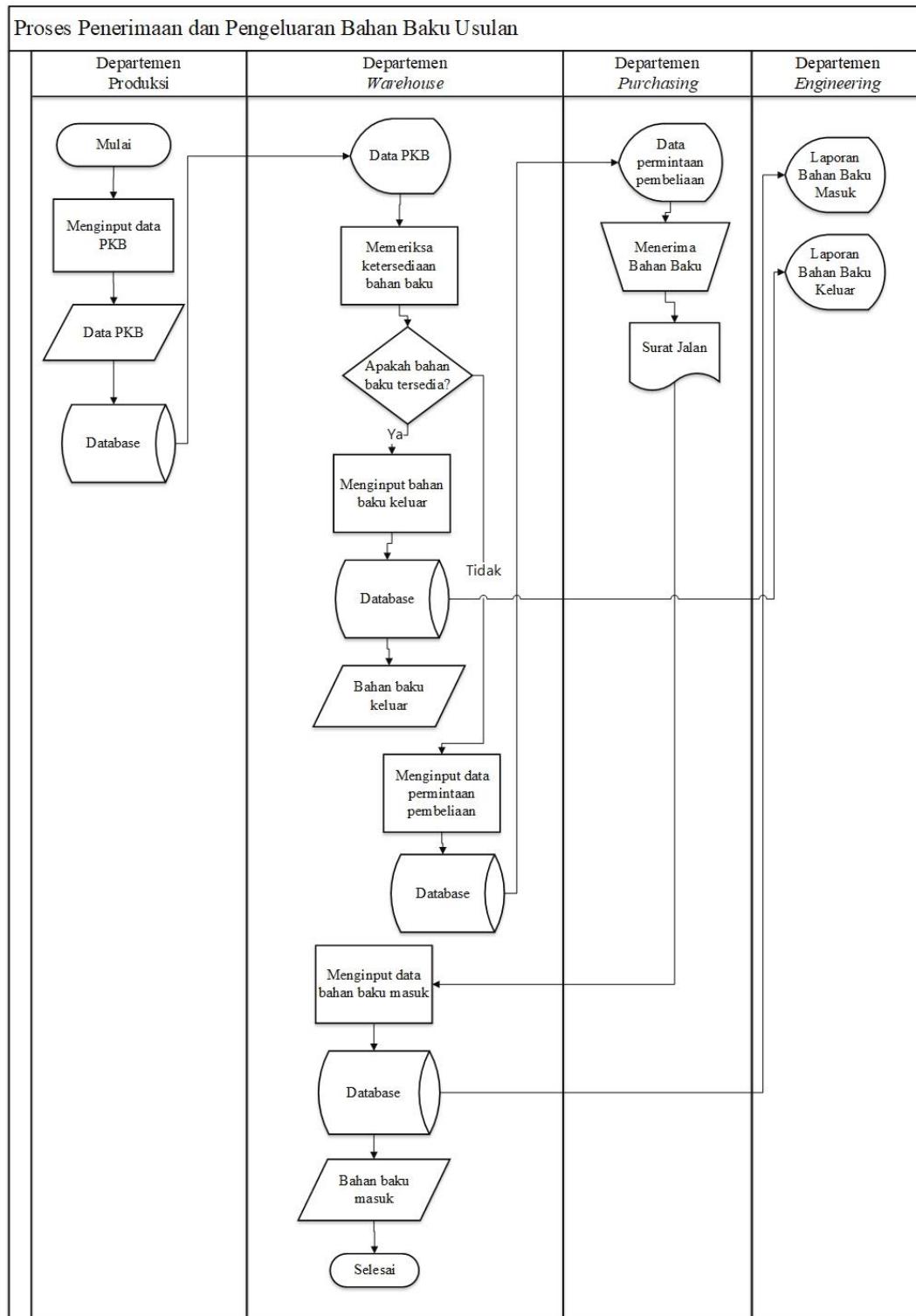
5.2. Prosedur Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku Usulan

Prosedur sistem informasi penerimaan dan pengeluaran bahan baku yang diusulkan melalui tahapan berikut:

1. Departemen produksi akan menginput data permintaan kebutuhan barang (PKB), kemudian tersimpan ke dalam *database*.
2. Operator *warehouse* akan menerima PKB dan mengecek ketersediaan bahan baku di gudang.
3. Apabila stok bahan baku tersedia maka Operator *warehouse* akan menyetujui PKB.
4. Operator *warehouse* akan menginput bahan baku keluar yang diajukan oleh departemen produksi kemudian tersimpan dalam *database*.
5. Apabila diketahui bahan baku telah mencapai titik minimum maka operator *warehouse* akan membuat permintaan pembelian bahan baku.
6. Operator *warehouse* akan mengirimkan permintaan pembelian bahan baku ke departemen *purchasing*.

7. Departemen *Purchasing* akan menerima permintaan pembelian bahan baku.
8. Operator *warehouse* akan menginput bahan baku masuk dari *supplier* berdasarkan surat jalan.
9. Operator *warehouse* mencetak laporan bulanan transaksi bahan baku masuk dan keluar yang ditunjukan kepada Departemen *Engineering* sebagai dasar perencanaan *loading* mesin.
10. Departemen *Engineering* dapat melihat dan mencetak laporan bulanan transaksi bahan baku masuk dan keluar sebagai dasar perencanaan *loading* mesin.

Berikut merupakan *Flowmap* sistem informasi penerimaan dan pengeluaran bahan baku usulan yang dapat dilihat pada Gambar V.1.



Gambar V.1 *Flowmap Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku Usulan*

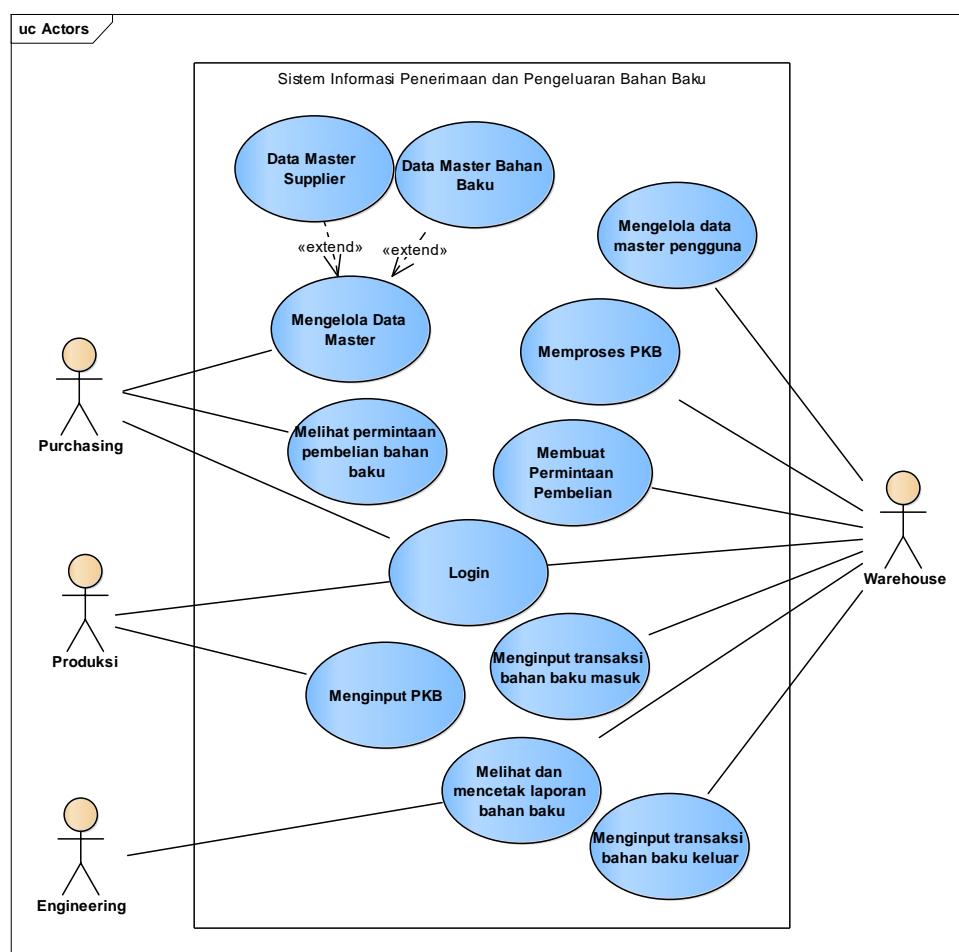
(Sumber: Hasil Analisis, 2019)

5.3. Analisis Sistem Usulan

Analisis proses sistem informasi penerimaan dan pengeluaran bahan baku menggunakan pemodelan sistem UML (*unified modeling language*) yang meliputi pembuatan *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, *deployment diagram* dan juga pembuatan kamus data.

5.3.1. Use Case Diagram

Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Rancangan *use case diagram* sistem informasi penerimaan dan pengeluaran bahan baku dapat dilihat pada Gambar V.2



Gambar V.2 *Use Case Diagram* Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku Usulan
Baku Usulan

(Sumber: Hasil Analisis Data, 2019)

1. Definisi Aktor

Pendefinisian aktor pada *use case* penerimaan dan pengeluaran bahan baku usulan dapat dilihat pada Tabel V.2 berikut:

Tabel V.2 Definisi Aktor *Use Case Diagram* Sistem Usulan

| No | Aktor | Definisi |
|----|-------------------------------|--|
| 1. | Departemen Produksi | Departemen Produksi adalah aktor yang menginput data permintaan bahan baku untuk proses produksi serta mengolah data master <i>user</i> . |
| 2. | Operator <i>Warehouse</i> | Operator <i>Warehouse</i> adalah aktor yang menerima permintaan bahan baku dari departemen produksi, memproses permintaan bahan baku dari produksi, membuat permintaan pembelian bahan baku, menginput bahan baku masuk dan keluar, melihat laporan bulanan serta mengolah data master <i>user</i> . |
| 3. | Departemen <i>Purchasing</i> | Departemen <i>Purchasing</i> adalah aktor yang menerima permintaan pembelian bahan baku serta mengolah data master bahan baku dan <i>supplier</i> . |
| 4. | Departemen <i>Engineering</i> | Departemen <i>Engineering</i> adalah aktor yang dapat melihat dan mencetak laporan bulanan yang digunakan untuk perencanaan <i>loading</i> mesin. |

(Sumber: Hasil Analisis Data, 2019)

2. Definisi *Use Case*

Pendefinisian *use case* pada sistem informasi penerimaan dan pengeluaran bahan baku usulan dapat dilihat pada Tabel V.3:

Tabel V.3 Definisi *Use Case Diagram* Sistem Usulan

| No. | Use Case | Deskripsi |
|-----|---------------------------------------|--|
| 1. | <i>Login</i> | Proses untuk melakukan <i>login</i> pada aplikasi. |
| 2. | Menginput data permintaan bahan baku. | Proses memasukan data permintaan bahan baku yang dilakukan oleh Departemen Produksi. |

| No. | Use Case | Deskripsi |
|-----|--|--|
| 3. | Mengelola data master | Proses mengelola data master berupa menambah data master, mengubah, dan menghapus data master dilakukan oleh Operator <i>Warehouse</i> dan Departemen <i>Purchasing</i> . |
| 4. | Memproses PKB | Proses yang dilakukan terhadap permintaan bahan baku yang masuk. Apakah menyetujui atau melakukan permintaan pembelian. |
| 5. | Menginput bahan baku keluar. | Proses menginput data bahan baku keluar yang tersedia setelah transaksi permintaan bahan baku oleh Departemen Produksi dan proses ini dilakukan oleh Operator <i>Warehouse</i> . |
| 6. | Membuat Permintaan Pembelian | Proses pembuatan permintaan pembelian bahan baku kepada Departemen <i>Purchasing</i> dikarenakan bahan baku sudah di tahap minimum. |
| 7. | Menginput bahan baku masuk. | Proses menginput data bahan baku masuk dari <i>supplier</i> berdasarkan surat jalan. |
| 8. | Melihat data Permintaan pembelian bahan baku | Proses untuk melihat data permintaan pembelian bahan baku yang diajukan Operator <i>Warehouse</i> kepada Departemen <i>Purchasing</i> . |
| 9. | Melihat dan mencetak laporan | Proses melihat dan mencetak laporan bahan baku masuk dan bahan baku keluar yang dilakukan oleh Departemen <i>Warehouse</i> dan Departemen <i>Purchasing</i> . |

(Sumber: Hasil Analisis Data, 2019)

3. Skenario Use Case

Skenario jalannya masing-masing *Use case* pada sistem informasi penerimaan dan pengeluaran bahan baku usulan dapat dilihat pada poin-poin berikut:

a. Use Case Login

Berikut adalah skenario *use case login* yang terdapat pada Tabel V.4.

Tabel V.4 Skenario Use Case Login

| Nama Use Case | Login |
|------------------------------|---|
| <i>Primary Actor</i> | Departemen Produksi, Departemen <i>Warehouse</i> , Departemen <i>Purchasing</i> serta Departemen <i>Engineering</i> . |
| <i>Use Case Description</i> | <i>Use case</i> ini menggambarkan <i>login</i> sesuai dengan hak akses dari <i>user</i> . Dalam hal ini hak aksesnya sesuai dengan jabatan. |
| <i>Relationship</i> | <i>Association</i> : Departemen produksi, Departemen <i>Warehouse</i> , Departemen <i>Purchasing</i> , serta Departemen <i>Engineering</i> . |
| <i>Normal Flow of Events</i> | <ol style="list-style-type: none"> 1. <i>User</i> membuka aplikasi 2. Sistem menampilkan <i>form Login</i> 3. <i>User</i> mengisi <i>username</i> dan <i>password</i> pada <i>form login</i>. 4. Sistem akan melakukan validasi <i>login</i>. 5. Jika <i>username</i> dan <i>password</i> valid, maka akan muncul halaman utama. 6. Jika <i>username</i> dan <i>password</i> salah (tidak valid) maka akan menampilkan <i>form login</i>. |

(Sumber: Hasil Analisis Data, 2019)

b. *Use Case* Menginput Data Permintaan Kebutuhan Bahan Baku

Berikut ini adalah *use case* menginput data permintaan kebutuhan bahan baku yang dapat dilihat pada Tabel V.5.

Tabel V.5 Skenario Use Case Menginput Data Permintaan kebutuhan Bahan Baku

| Nama Use Case | Menginput Data Permintaan Kebutuhan Bahan Baku |
|-----------------------------|--|
| <i>Primary Actor</i> | Departemen Produksi |
| <i>Use Case Description</i> | <i>Use case</i> ini menggambarkan proses memasukan data permintaan kebutuhan bahan baku dari |

| | |
|------------------------------|--|
| Nama Use Case | Menginput Data Permintaan Kebutuhan Bahan Baku |
| | departemen produksi kepada departemen <i>warehouse</i> . |
| <i>Relationship</i> | <i>Association:</i> Departemen produksi |
| <i>Normal Flow of Events</i> | <ol style="list-style-type: none"> 1. Departemen Produksi memilih menu permintaan, kemudian memilih sub menu <i>form</i> permintaan. 2. Sistem menampilkan <i>form</i> permintaan 3. Bagian Produksi memasukan data permintaan berupa nama bahan baku, nama customer dan jumlah, kemudian klik tambah dan sistem akan menampilkan permintaan detail. 4. Proses memasukan data permintaan dapat dilakukan lebih dari satu kali. 5. Bagian Produksi memilih tombol simpan permintaan untuk menyimpan permintaan bahan baku. |

(Sumber: Hasil Analisis Data, 2019)

c. *Use Case* Mengelola Data Master

Berikut ini adalah skenario *use case* mengelola Data Master yang dapat dilihat pada Tabel V.6.

Tabel V.6 Skenario *Use Case* Mengelola Data Master

| | |
|-----------------------------|---|
| Nama Use Case | Mengelola Data Master |
| <i>Primary Actor</i> | Departemen <i>Warehouse</i> , Departemen <i>Engineering</i> |
| <i>Use Case Description</i> | <i>Use case</i> ini menggambarkan proses mengelola data master. Yaitu data master bahan baku, <i>supplier</i> , dan <i>user</i> . |
| <i>Relationship</i> | <i>Association:</i> Departemen <i>Warehouse</i> , Departemen <i>Engineering</i> |

| Nama Use Case | Mengelola Data Master |
|------------------------------|---|
| <i>Normal Flow of Events</i> | <ol style="list-style-type: none"> 1. Departemen <i>warehouse</i> dan Departemen <i>Engineering</i> membuka menu Data Master. 2. Memilih submenu Data Master. 3. Menambah Data Master. <ol style="list-style-type: none"> a. Memilih tombol Tambah. b. Mengisi <i>form input</i> yang ditampilkan oleh sistem. c. Menyimpan data yang telah diinput. d. Sistem menampilkan tabel data master 4. Mengubah Data Master <ol style="list-style-type: none"> a. Memilih tombol ubah yang terdapat pada tabel data master. b. Sistem akan menampilkan <i>form edit</i>. c. Mengisi data yang akan diubah. d. Memilih tombol simpan. e. Sistem akan memperbarui data yang telah diubah. f. Sistem akan menampilkan tabel data master. 5. Menghapus Data Master <ol style="list-style-type: none"> a. Memilih tombol hapus yang telah ada di tabel data master. b. Sistem akan menghapus data yang ingin dihapus. |

(Sumber: Hasil Analisis 2019)

d. *Use Case* Memproses Permintaan Kebutuhan Bahan Baku

Berikut ini adalah skenario *use case* memproses permintaan bahan baku yang dapat dilihat pada Tabel V.7.

Tabel V.7 Skenario *Use Case* Memproses Permintaan Bahan Baku

| | |
|------------------------------|---|
| Nama Use Case | Memproses Permintaan Bahan Baku |
| <i>Primary Actor</i> | <i>Departemen Warehouse</i> |
| <i>Use Case Description</i> | <i>Use case ini menggambarkan pemrosesan permintaan bahan baku dari Departemen produksi.</i> |
| <i>Relationship</i> | <i>Association: Departemen Warehouse</i> |
| <i>Normal Flow of Events</i> | <ol style="list-style-type: none"> 1. Departemen <i>Warehouse</i> akan membuka menu data permintaan bahan baku, Sistem akan menampilkan data permintaan bahan baku beserta stok bahan baku. 2. Departemen <i>Warehouse</i> akan memeriksa ketersediaan bahan baku pada data permintaan bahan baku. 3. Apabila diketahui jumlah persediaan bahan baku lebih besar dari permintaan maka akan bagian Departemen <i>Warehouse</i> memilih tombol Transaksi keluar. 4. Sistem akan menampilkan <i>form</i> transaksi bahan baku keluar. 5. Apabila diketahui jumlah permintaan lebih banyak dari jumlah persediaan bahan baku maka Bagian Inventory memilih tombol PP. 6. Sistem akan menampilkan <i>form</i> Permintaan Pembelian Bahan Baku. |

(Sumber: Hasil Analisis Data, 2019)

e. *Use Case Menginput Transaksi Bahan Baku Keluar*

Berikut ini adalah skenario *use case* menginput transaksi bahan baku keluar yang dapat dilihat pada Tabel V.8.

Tabel V.8 Skenario *Use Case* Menginput Transaksi Bahan Baku Keluar

| | |
|----------------------|--|
| Nama Use Case | Menginput Transaksi Bahan Baku Keluar |
| <i>Primary Actor</i> | <i>Departemen Warehouse</i> |

| Nama Use Case | Menginput Transaksi Bahan Baku Keluar |
|------------------------------|--|
| <i>Use Case Description</i> | <i>Use case</i> ini menggambarkan proses menginput data bahan baku keluar setelah adanya permintaan bahan baku. |
| <i>Relationship</i> | <i>Association:</i> Departemen <i>Warehouse</i> |
| <i>Normal Flow of Events</i> | <ol style="list-style-type: none"> 1. Departemen <i>Warehouse</i> memilih menu Data Transaksi, kemudian pilih submenu <i>Form</i> Bahan Baku Keluar. 2. Sistem akan menampilkan <i>Form</i> Bahan Baku Keluar. 3. Mengisi <i>field</i> nama bahan baku, jumlah, dan keterangan 4. Departemen <i>Warehouse</i> menekan tombol Tambah. 5. Sistem akan menampilkan data detil yang telah diinput. 6. Apabila data sudah benar tekan tombol Selesai input. |

(Sumber: Hasil Analisis Data, 2019)

f. *Use Case* Membuat Permintaan Pembelian

Berikut ini adalah skenario *use case* membuat permintaan pembelian yang dapat dilihat pada Tabel V.9.

Tabel V.9 Skenario *Use Case* Membuat Permintaan Pembelian

| Nama Use Case | Membuat Permintaan Pembelian |
|-----------------------------|--|
| <i>Primary Actor</i> | Departemen <i>Warehouse</i> |
| <i>Use Case Description</i> | <i>Use case</i> ini menggambarkan proses membuat permintaan pembelian yang dilakukan oleh Departemen <i>Warehouse</i> untuk Departemen <i>Purchasing</i> . |
| <i>Relationship</i> | <i>Association:</i> Departemen <i>Warehouse</i> |

| Nama Use Case | Membuat Permintaan Pembelian |
|------------------------------|---|
| <i>Normal Flow of Events</i> | <ol style="list-style-type: none"> 1. Departemen <i>Warehouse</i> membuka menu <i>form</i> permintaan pembelian. 2. Sistem akan menampilkan <i>form</i> permintaan pembelian. 3. Departemen <i>Warehouse</i> mengisi nama bahan baku, jumlah, <i>request schedule</i>, dan keterangan. 4. Klik tombol buat permintaan kemudian sistem akan menampilkan permintaan detil. 5. Apabila data yang diinput sudah benar maka pilih tombol selesai input. 6. Sistem akan menampilkan data permintaan pembelian bahan baku. |

(Sumber: Hasil Analisis, 2019)

g. *Use Case* Menginput bahan baku masuk.

Berikut ini adalah skenario *use case* menginput bahan baku masuk yang dapat dilihat pada Tabel V.10.

Tabel V.10 Skenario *Use Case* Menginput Transaksi Bahan Baku Masuk.

| Nama Use Case | Menginput Transaksi Bahan Baku Masuk |
|------------------------------|---|
| <i>Primary Actor</i> | Departemen <i>Warehouse</i> |
| <i>Use Case Description</i> | <i>Use case</i> ini menggambarkan proses memasukan data bahan baku ketika kedatangan bahan baku |
| <i>Relationship</i> | Association: Departemen <i>Warehouse</i> |
| <i>Normal Flow of Events</i> | <ol style="list-style-type: none"> 1. Departemen <i>Warehouse</i> memilih menu Data Transaksi, kemudian pilih submenu <i>Form</i> Bahan Baku Masuk. 2. Sistem akan menampilkan <i>Form</i> Bahan Baku Masuk. 3. Mengisi <i>field</i> |

| Nama Use Case | Menginput Transaksi Bahan Baku Masuk |
|----------------------|---|
| | <p>4. Departemen <i>Warehouse</i> menekan tombol Tambah Data</p> <p>5. Sistem akan menampilkan data detil yang telah diinput.</p> <p>6. Apabila data sudah benar tekan tombol Selesai input.</p> <p>7. Sistem akan menampilkan <i>pop up form</i>.</p> <p>8. Departemen <i>Warehouse</i> mengisi <i>pop up form</i> tersebut.</p> <p>9. Departemen <i>Warehouse</i> memilih tombol simpan bahan baku masuk.</p> <p>10. Sistem akan menyimpan data bahan baku masuk.</p> |

(Sumber: Hasil Analisis Data, 2019)

h. *Use Case* Melihat Data Permintaan Pembelian Bahan Baku

Berikut ini merupakan skenario *use case* melihat permintaan pembelian bahan baku yang dapat dilihat pada Tabel V.11.

Tabel V.11 Skenario *Use case* Melihat data Permintaan Pembelian

| Nama Use Case | Menerima Permintaan Pembelian Bahan Baku |
|------------------------------|---|
| <i>Primary Actor</i> | Departemen <i>Purchasing</i> |
| <i>Use Case Description</i> | <i>Use case</i> ini menggambarkan proses untuk melihat data permintaan pembelian oleh Departemen <i>purchasing</i> dari Departemen <i>warehouse</i> . |
| <i>Relationship</i> | Association: Departemen <i>Purchasing</i> |
| <i>Normal Flow of Events</i> | <p>1. Departemen <i>Purchasing</i> membuka menu Data Permintaan pembelian.</p> <p>2. Sistem akan menampilkan data permintaan pembelian.</p> |

(Sumber: Hasil Analisis Data, 2019)

i. *Use Case* Mencetak Laporan

Berikut ini adalah skenario *use case* mencetak laporan yang dapat dilihat pada Tabel V.12.

Tabel V.12 Skenario *Use Case* Mencetak Laporan

| Nama Use Case | Mencetak Laporan |
|------------------------------|--|
| <i>Primary Actor</i> | Departemen <i>Warehouse</i> |
| <i>Use Case Description</i> | <i>Use case</i> ini menggambarkan proses mencetak laporan transaksi bahan baku masuk dan transaksi bahan baku keluar. |
| <i>Relationship</i> | Association: Departemen <i>Warehouse</i> dan Departemen <i>Engineering</i> . |
| <i>Normal Flow of Events</i> | <ol style="list-style-type: none"> 1. Memilih menu Laporan. 2. Sistem akan menampilkan <i>form</i> laporan. 3. Memilih bulan. 4. Sistem akan menampilkan data transaksi pada bulan yang dipilih oleh Bagian <i>Inventory</i>. 5. Memilih tombol cetak. 6. Sistem menampilkan tampilan laporan. |

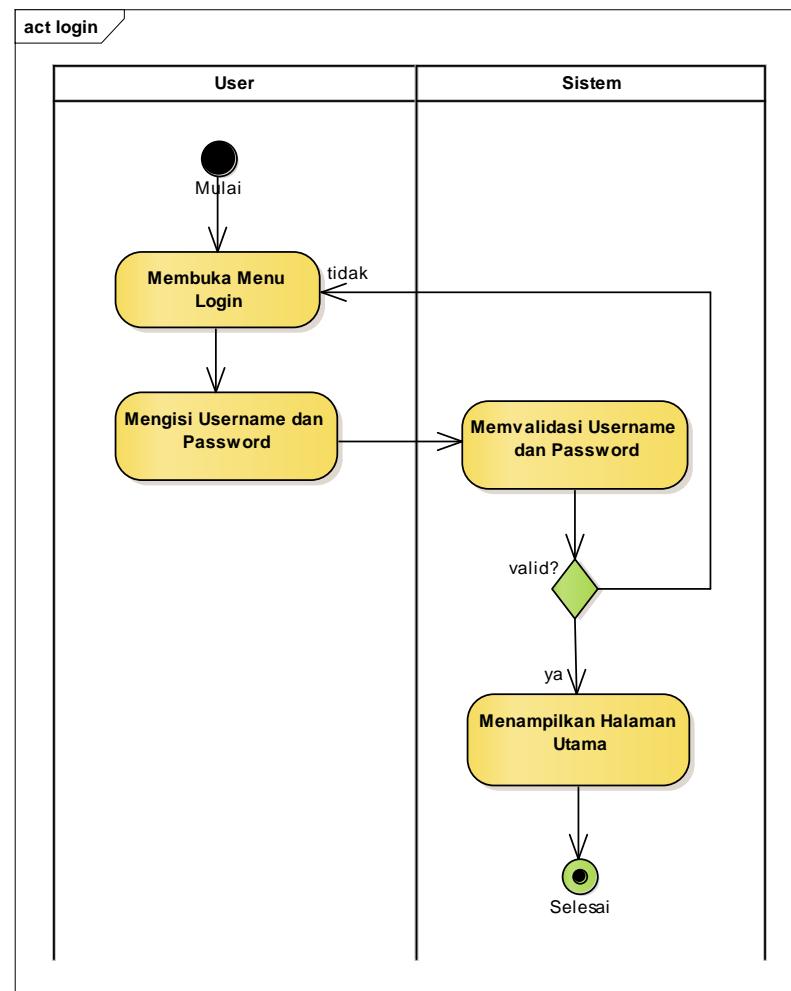
(Sumber: Hasil Analisis Data, 2019)

5.3.2 *Activity Diagram*

Activity diagram merupakan sebuah diagram yang menggambarkan urutan aktivitas proses bisnis pada suatu sistem. Berikut merupakan *activity diagram* usulan dari sistem informasi penerimaan dan pengeluaran bahan baku:

1. *Activity Diagram Login*

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu departemen *warehouse*, departemen produksi , dan departemen *purchasing* untuk mendapatkan akses kedalam sistem informasi penerimaan dan pengeluaran bahan baku. Berikut Gambar V.3 merupakan *activity diagram login*:

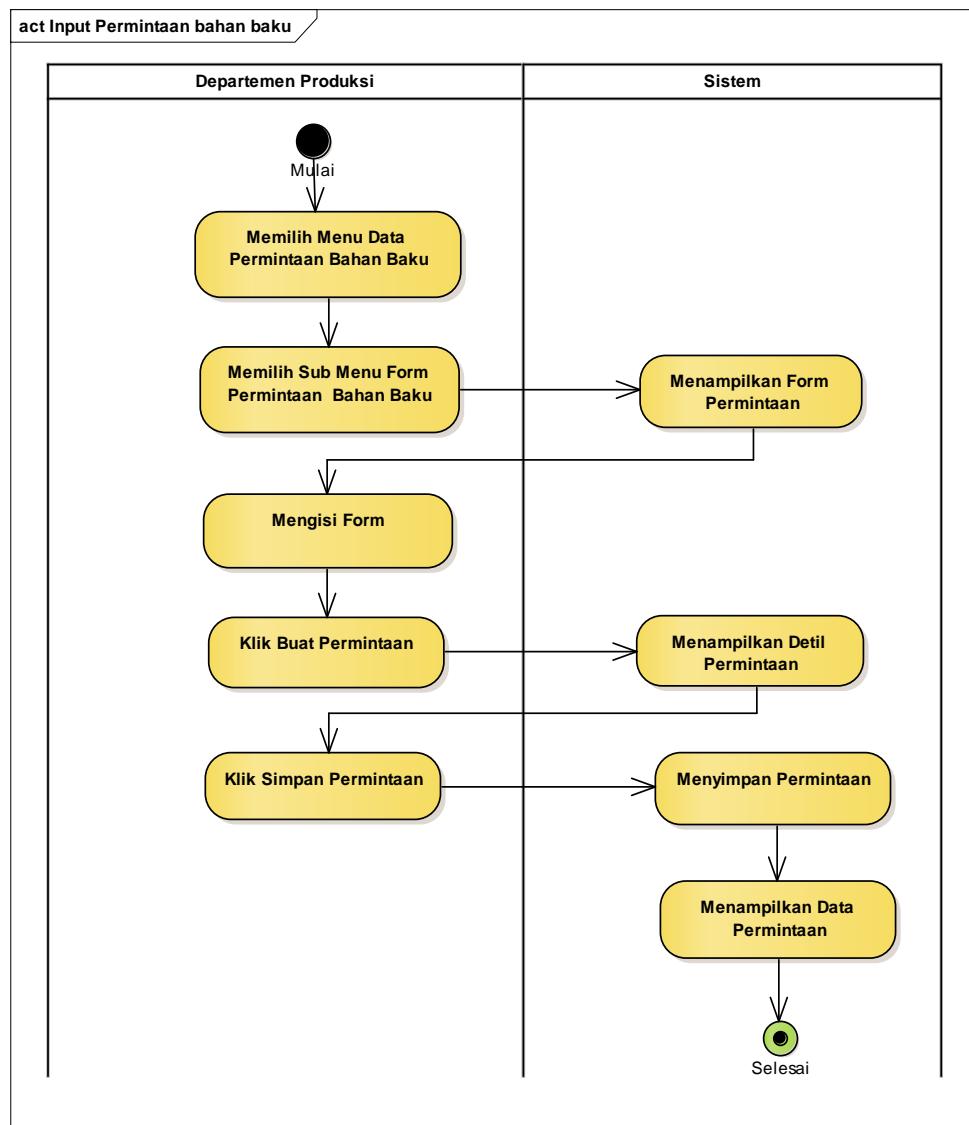


Gambar V.3 activity diagram login

(Sumber: Hasil Analisis, 2019)

2. Activity Diagram Menginput Data Permintaan Bahan Baku

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu departemen produksi untuk menginput data permintaan bahan baku dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.4:

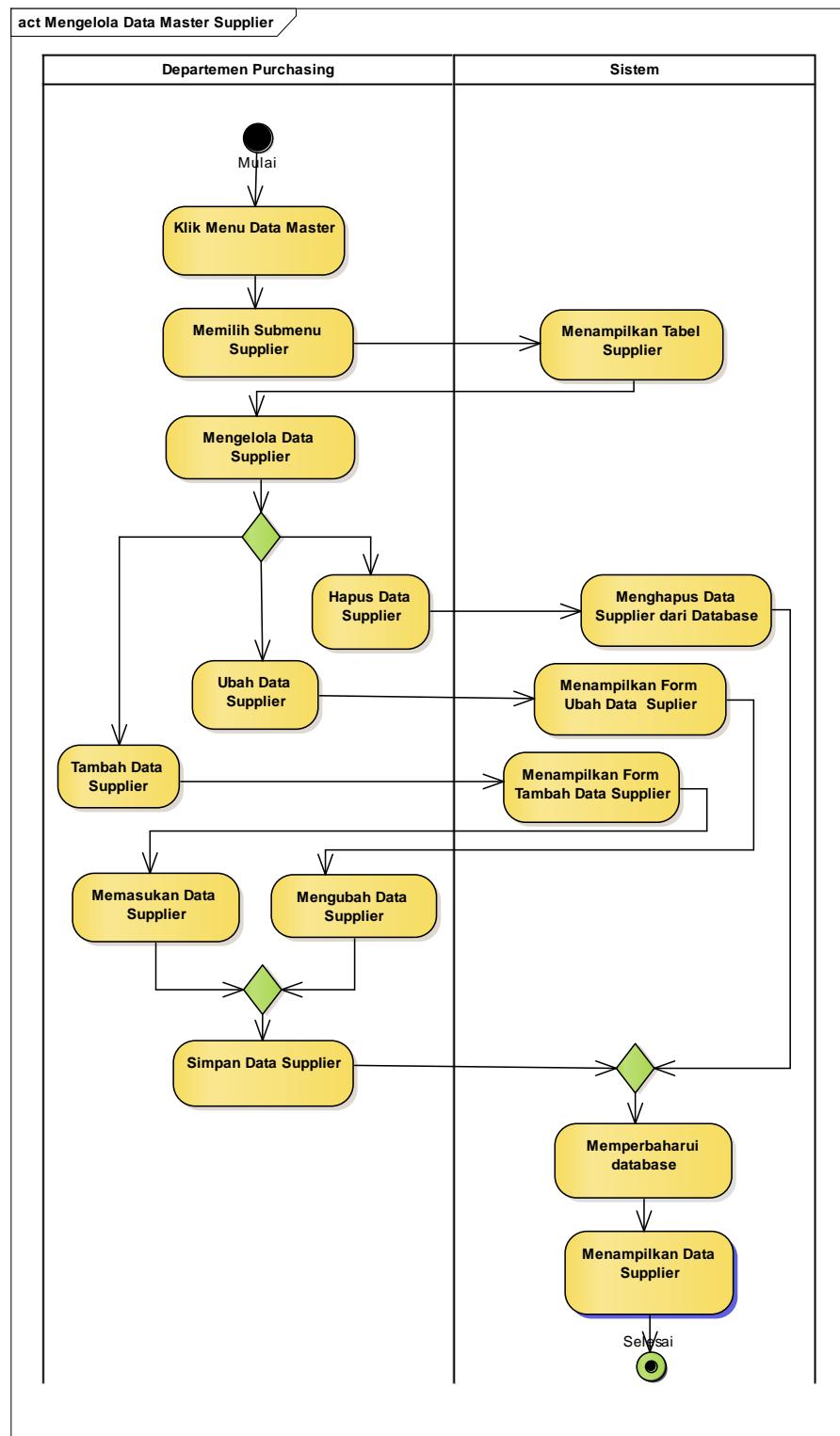


Gambar V.4 *Activity Diagram* Menginput Data Permintaan Bahan Baku

(Sumber: Hasil Analisis, 2019)

3. *Activity Diagram* Mengelola Data Master *Supplier*

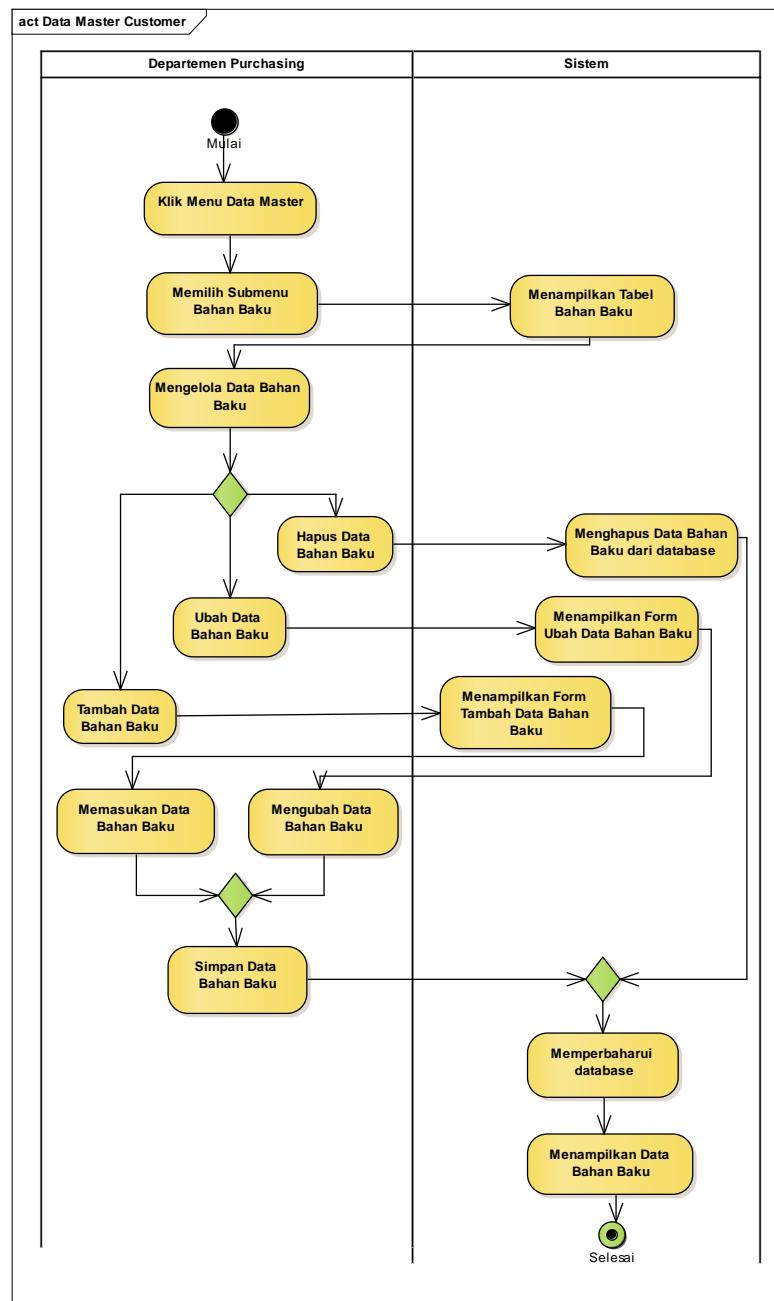
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu departemen *purchasing* untuk mengelola data master *supplier* dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.5:

Gambar V.5 *Activity Diagram Mengelola Data Master Supplier*

(Sumber: Hasil Analisis, 2019)

4. Activity Diagram Mengelola Data Master Bahan Baku

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu departemen *purchasing* untuk mengelola data master dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.6:

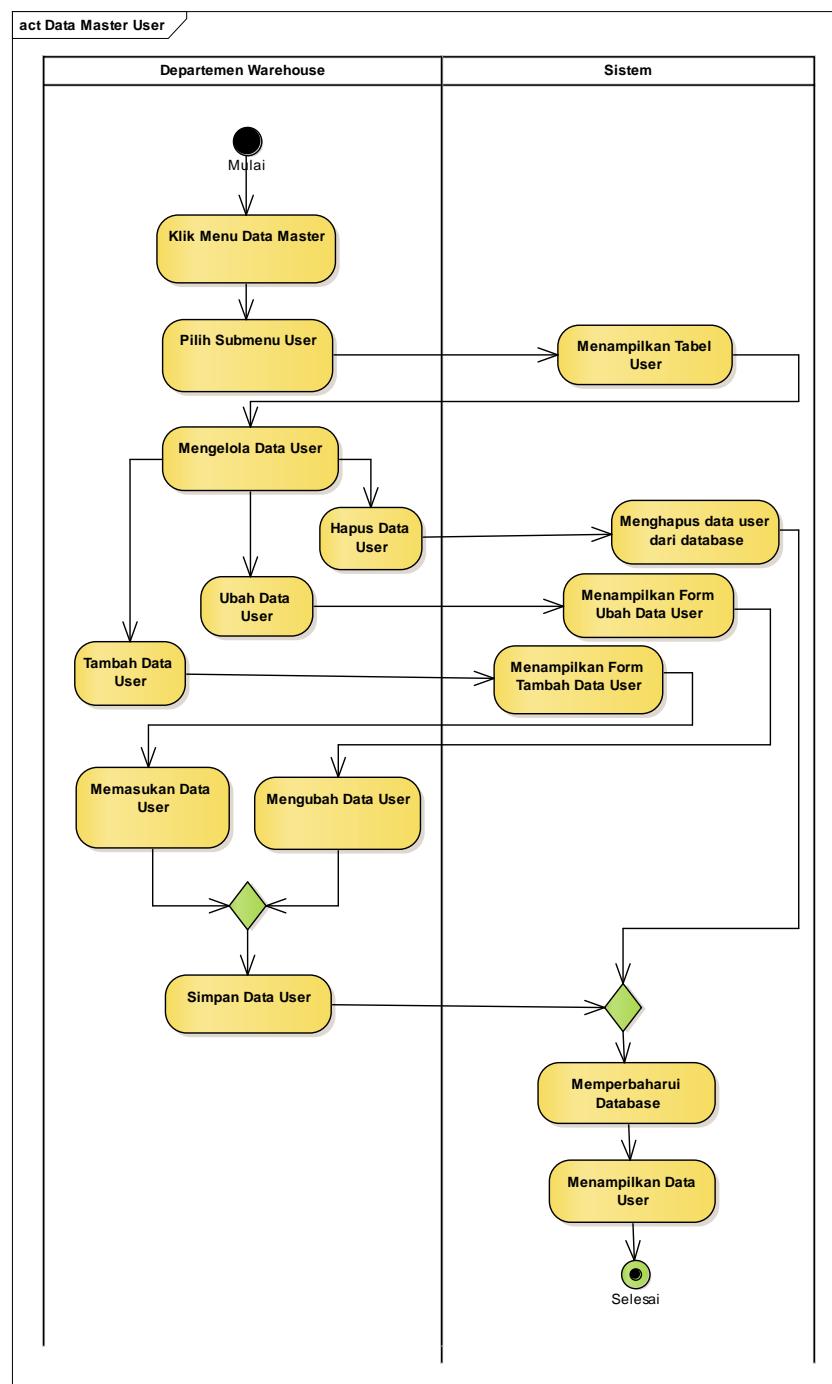


Gambar V.6 *Activity Diagram* Mengelola Data Master Bahan Baku

(Sumber: Hasil Analisis, 2019)

5. Activity Diagram Mengelola Data Master User

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu departemen *warehouse* untuk mengelola data master dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.7:

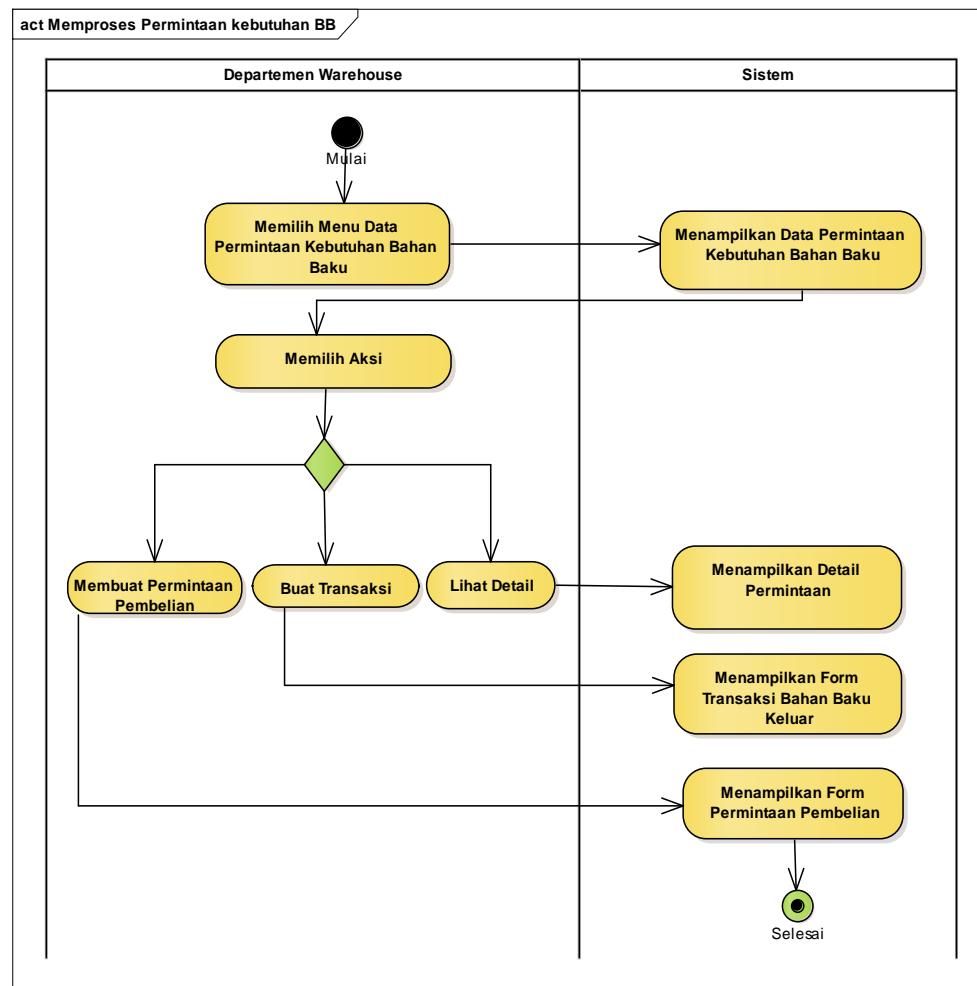


Gambar V.7 *Activity Diagram* Mengelola Data Master User

(Sumber: Hasil Analisis, 2019)

6. *Activity Diagram* Memproses Permintaan Bahan Baku

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu departemen *warehouse* untuk memproses permintaan bahan baku dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.8:

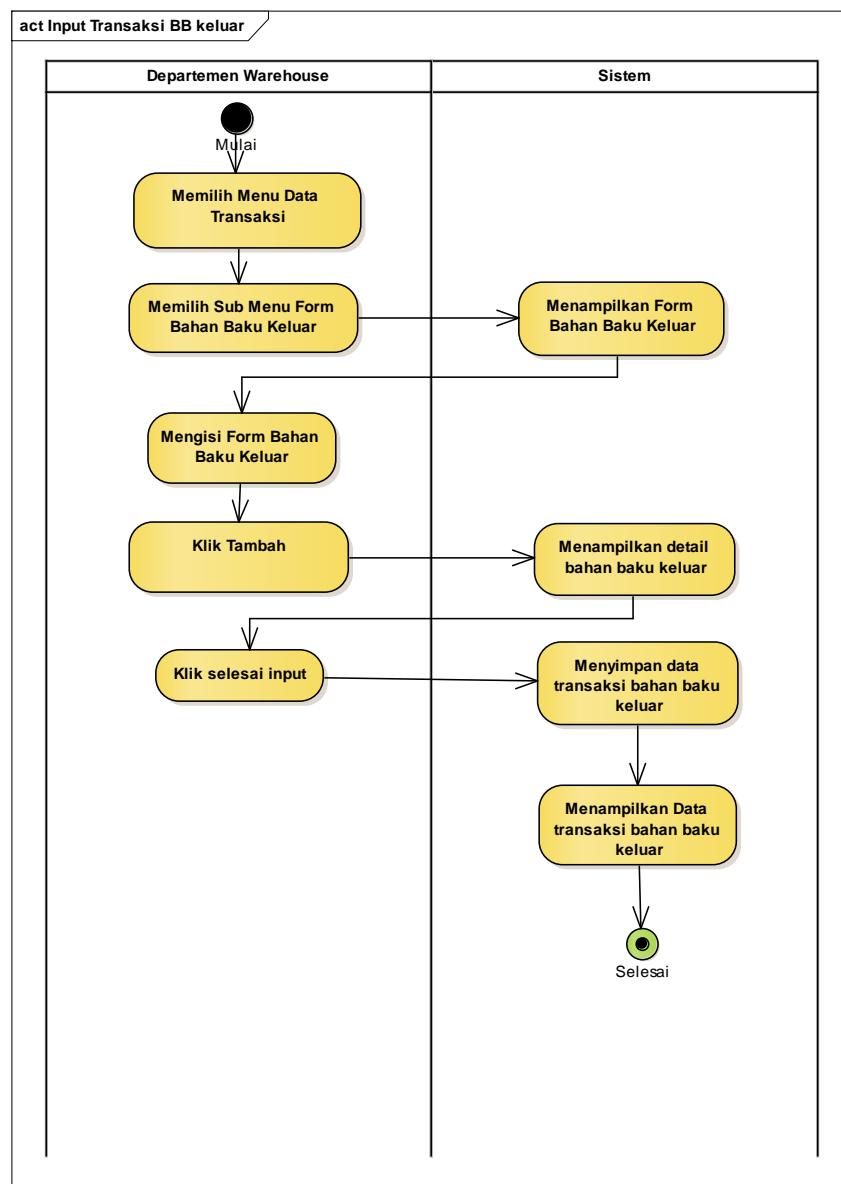


Gambar V.8 *Activity Diagram* Memproses Permintaan Bahan Baku

(Sumber: Hasil Analisis, 2019)

7. *Activity Diagram* Menginput Transaksi Bahan Baku Keluar

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu departemen *warehouse* untuk menginput transaksi bahan baku keluar dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.9:

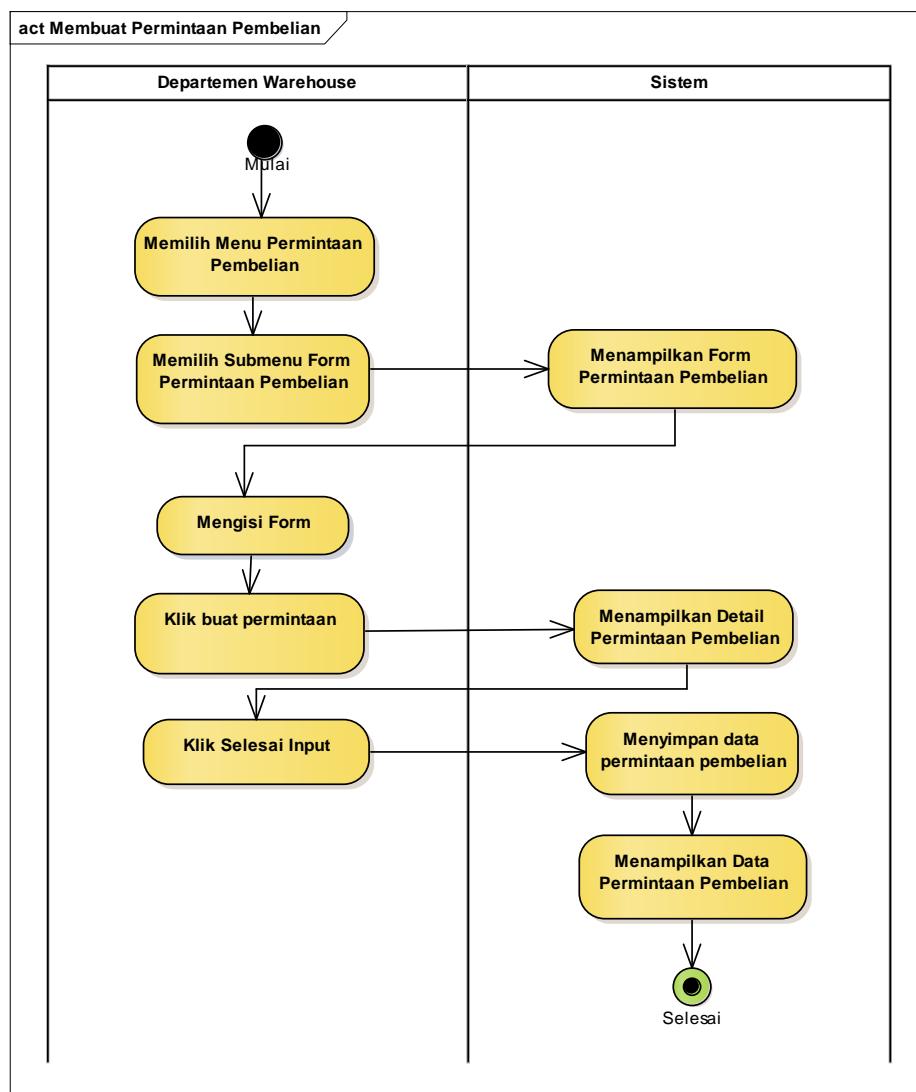


Gambar V.9 *Activity Diagram* Menginput Transaksi Bahan Baku Keluar

(Sumber: Hasil Analisis, 2019)

8. *Activity Diagram* Membuat Permintaan Pembelian

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu departemen *warehouse* untuk membuat permintaan pembelian dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.10:

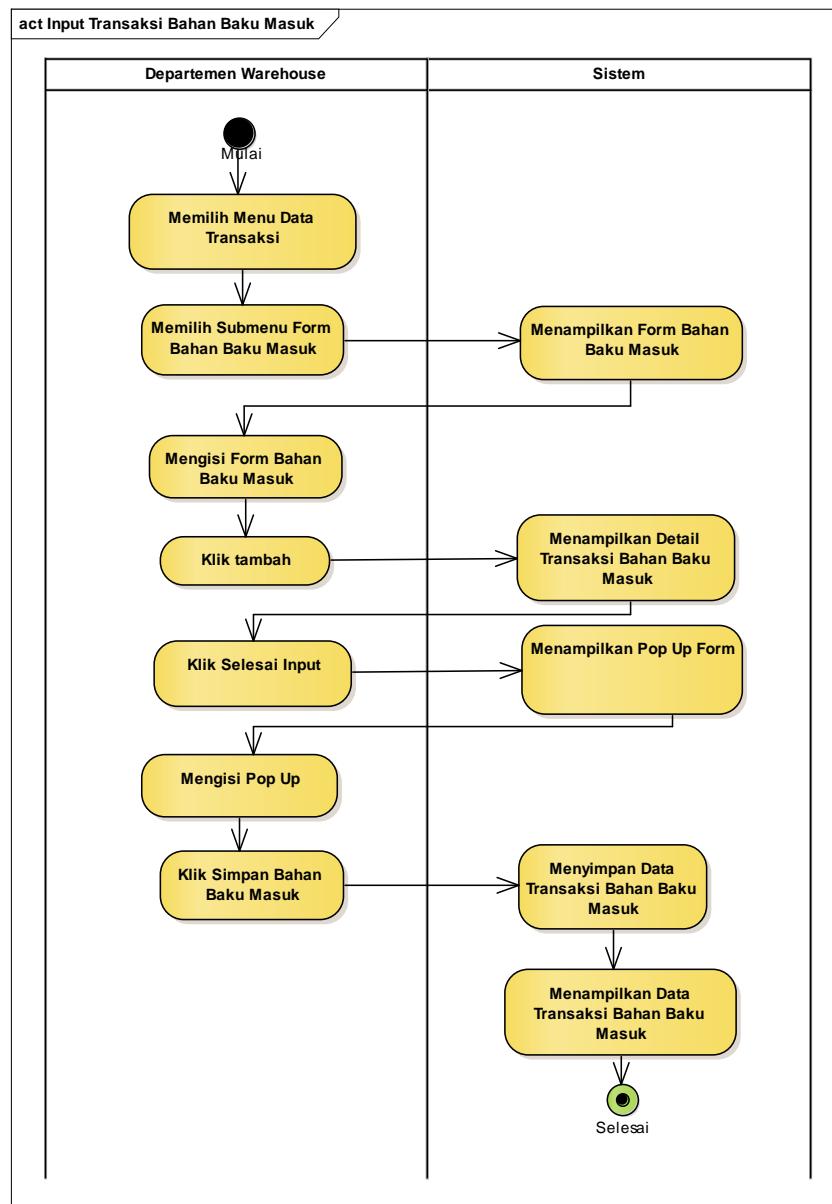


Gambar V.10 *Activity Diagram* Membuat Permintaan Pembelian

(Sumber: Hasil Analisis, 2019)

9. *Activity Diagram* Menginput Transaksi Bahan Baku Masuk

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu Departemen *warehouse* untuk menginput transaksi bahan baku masuk dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.11:

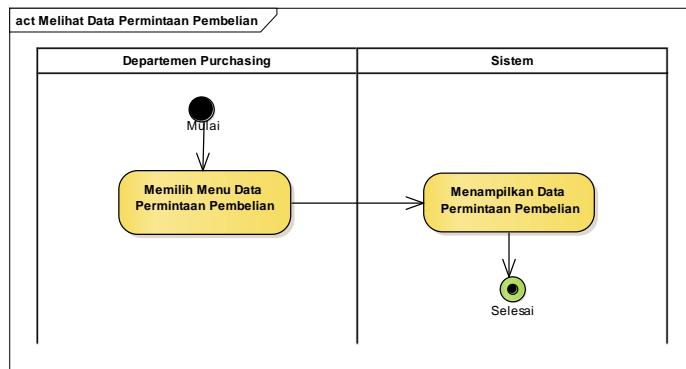


Gambar V.11 *Activity Diagram* Menginput Transaksi Bahan Baku Masuk

(Sumber: Hasil Analisis, 2019)

10. *Activity Diagram* Melihat Data Permintaan Pembelian Bahan Baku

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu Departemen *Purchasing* untuk melihat data permintaan pembelian bahan baku dalam sistem informasi penerimaan dan pengeluaran bahan baku yang dapat dilihat pada Gambar V.12:

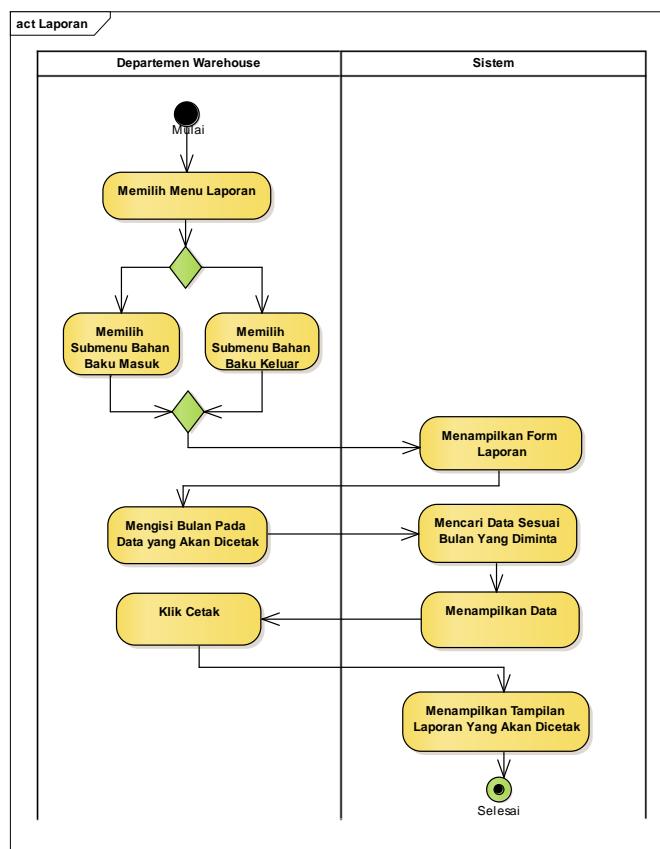


Gambar V.12 *Activity Diagram* Melihat Data Permintaan Pembelian

(Sumber: Hasil Analisis, 2019)

11. *Activity Diagram* Melihat dan Mencetak Laporan

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user* yaitu Departemen *warehouse* dan Departemen *Engineering* untuk melihat dan mencetak laporan transaksi bahan baku.



Gambar V.13 *Activity Diagram* Mencetak Laporan

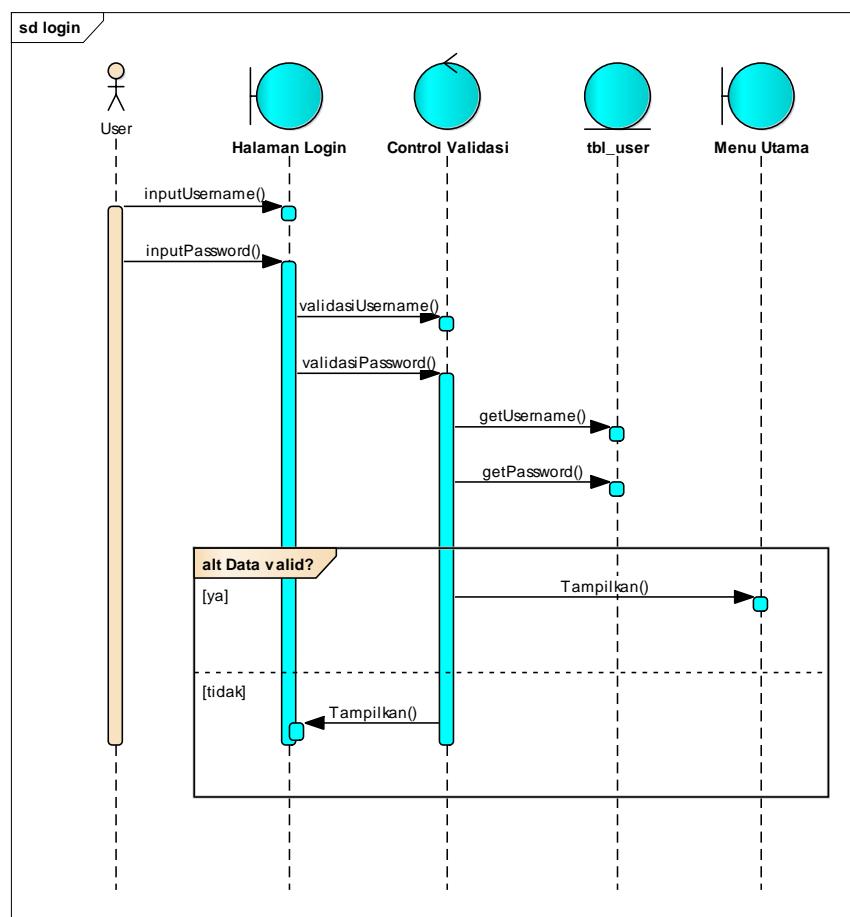
(Sumber: Hasil Analisis, 2019)

5.3.3. Sequence Diagram

Sequence diagram merupakan diagram untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Berikut merupakan beberapa *sequence diagram* usulan dari sistem informasi penerimaan dan pengeluaran bahan baku:

1. Sequence Diagram Login

Sequence diagram login merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses *login*. Proses *login* dalam sistem informasi penerimaan dan pengeluaran bahan baku ini dilakukan oleh *user* agar dapat mengakses sistem. Berikut merupakan *sequence diagram login* yang dapat dilihat pada Gambar V.14:

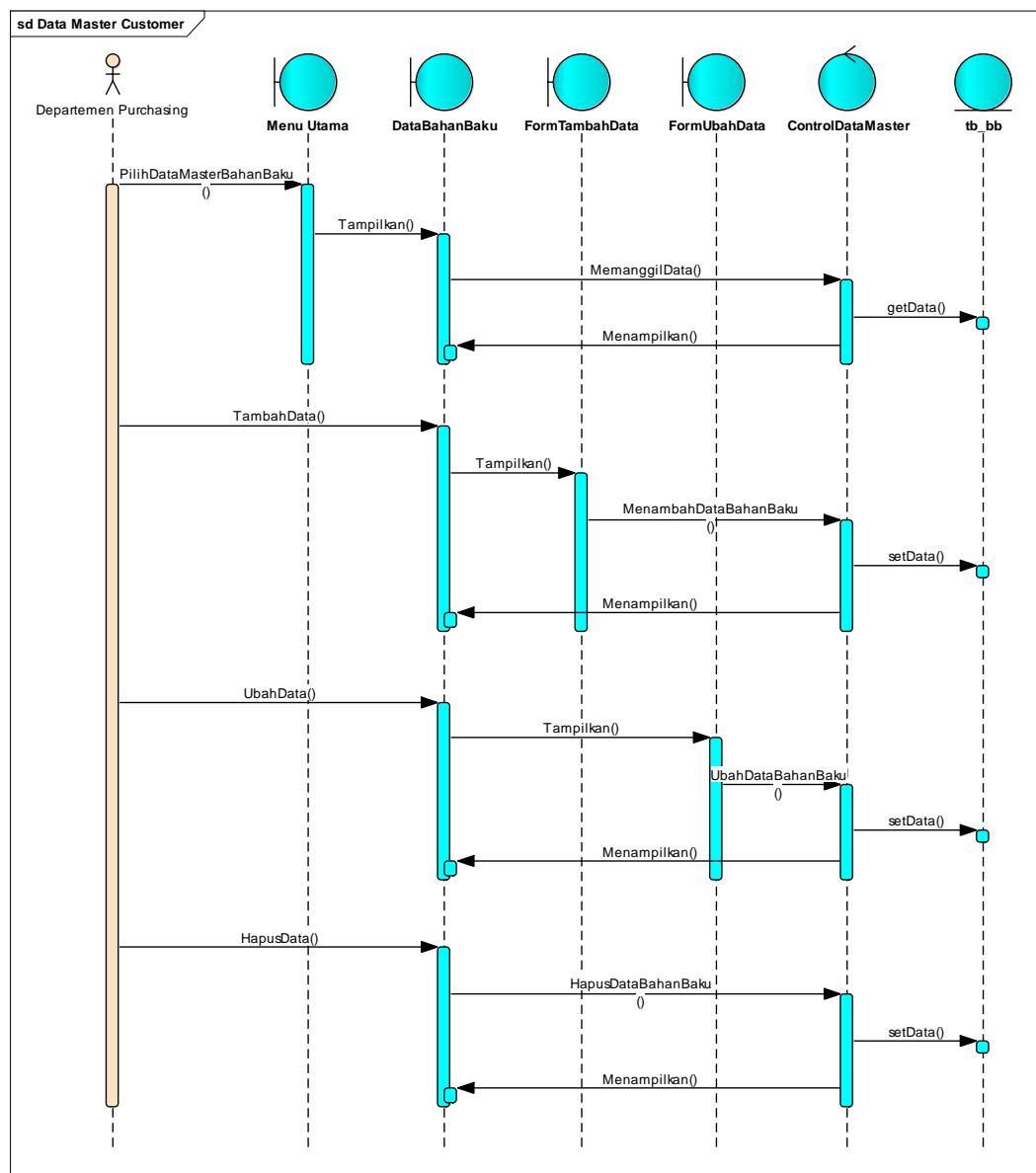


Gambar V.14 Sequence Diagram Proses Login

(Sumber: Hasil Analisis, 2019)

2. Sequence Diagram Mengelola Data Master Bahan Baku

Sequence diagram mengelola data master merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data master. Proses mengelola data master dalam sistem informasi penerimaan dan pengeluaran bahan baku ini dilakukan oleh Departemen *Purchasing* agar dapat mengelola data master. Berikut merupakan *sequence diagram* mengelola data master bahan baku yang dapat dilihat pada Gambar V.15:

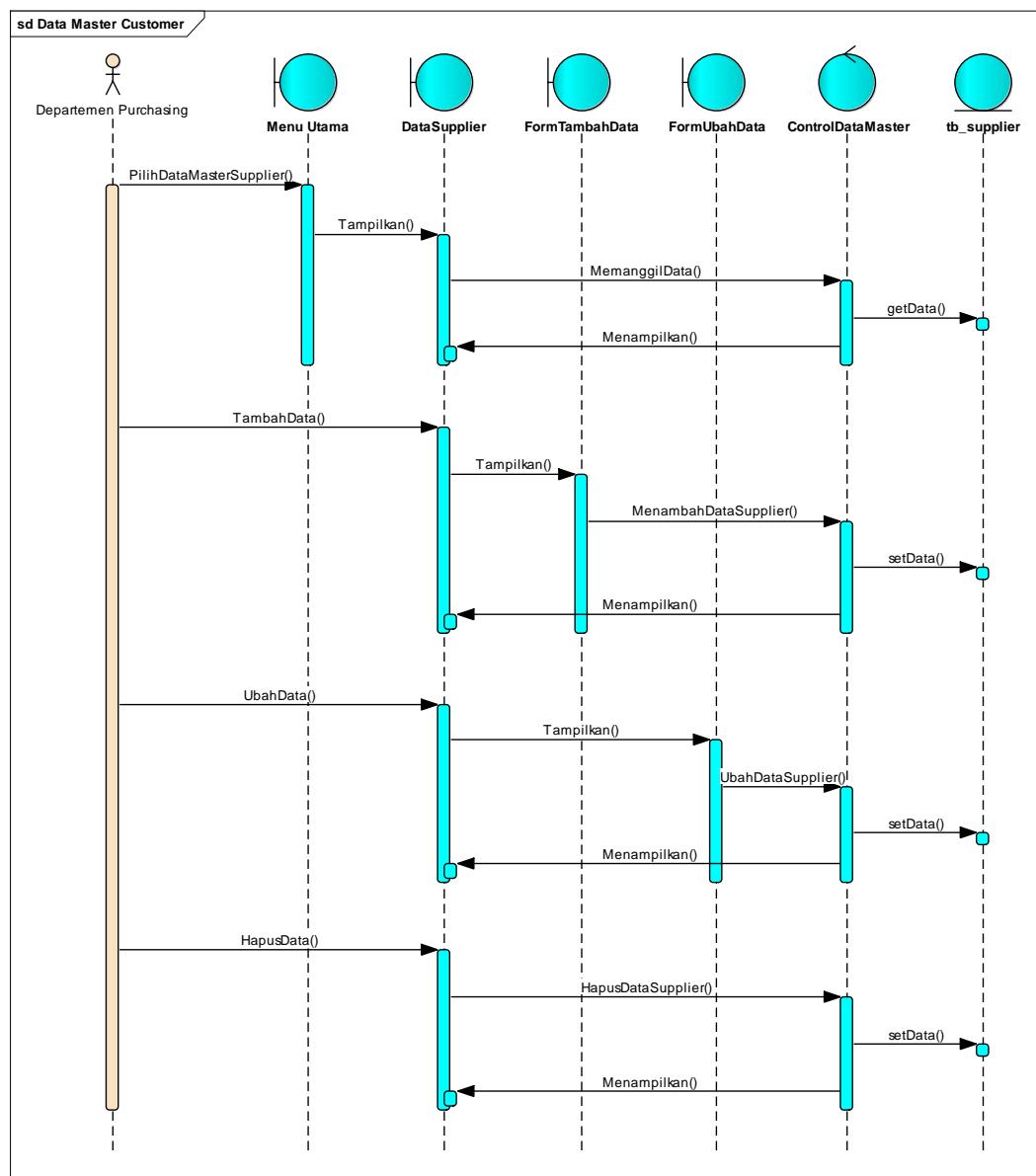


Gambar V.15 *Sequence Diagram* Proses Mengelola Data Master Bahan Baku

(Sumber: Hasil Analisis, 2019)

3. Sequence Diagram Mengelola Data Master Supplier

Sequence diagram mengelola data master merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data master. Proses mengelola data master dalam sistem informasi penerimaan dan pengeluaran bahan baku ini dilakukan oleh Departemen *Purchasing* agar dapat mengelola data master *supplier*. Berikut merupakan *sequence diagram* mengelola data master *supplier* yang dapat dilihat pada Gambar V.16:

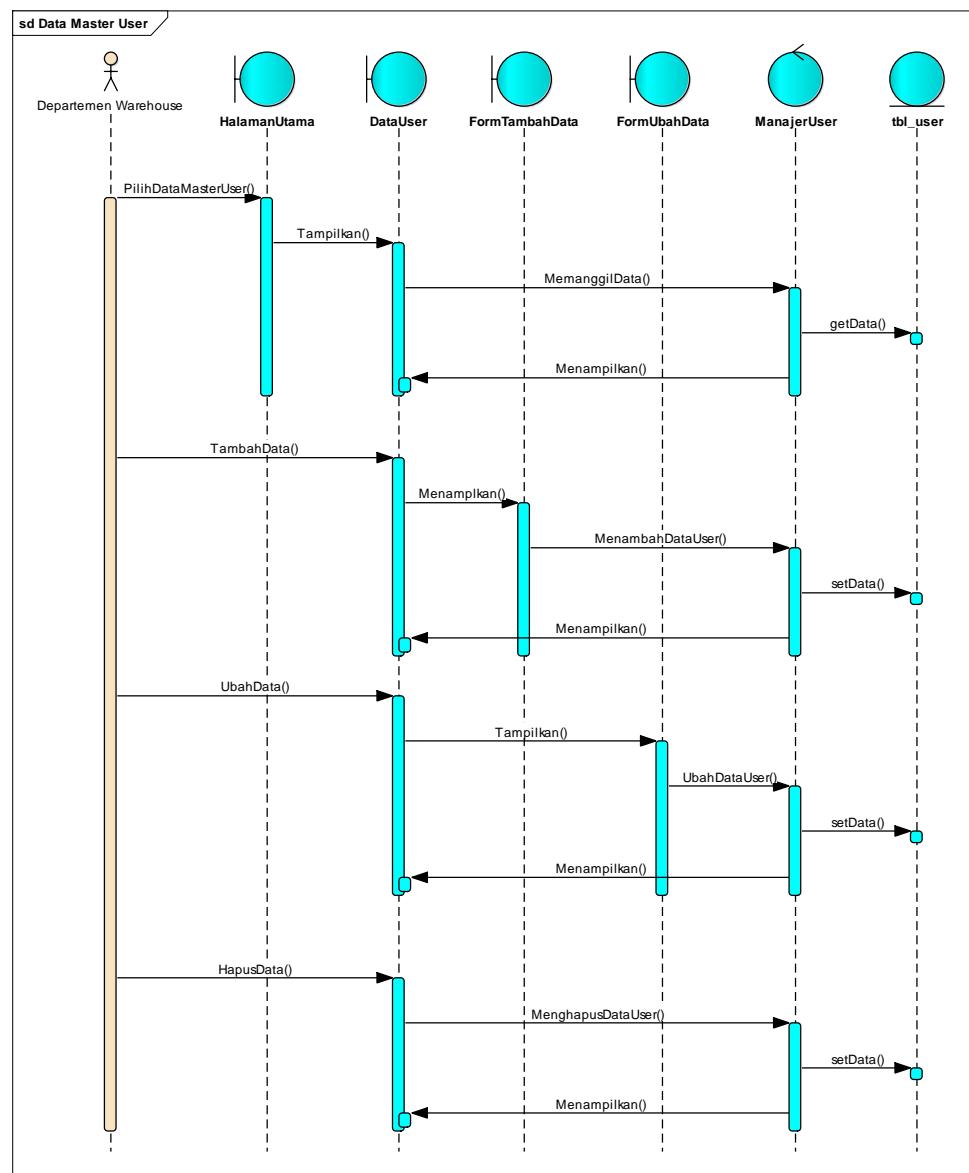


Gambar V.16 *Sequence Diagram* Proses Mengelola Data Master *Supplier*

(Sumber: Hasil Analisis, 2019)

4. Sequence Diagram Mengelola Data Master User

Sequence diagram mengelola data master merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mengelola data master. Proses mengelola data master dalam sistem informasi penerimaan dan pengeluaran bahan baku ini dilakukan oleh Departemen *Warehouse* agar dapat mengelola data master. Berikut merupakan *sequence diagram* mengelola data master *user* yang dapat dilihat pada Gambar V.17:

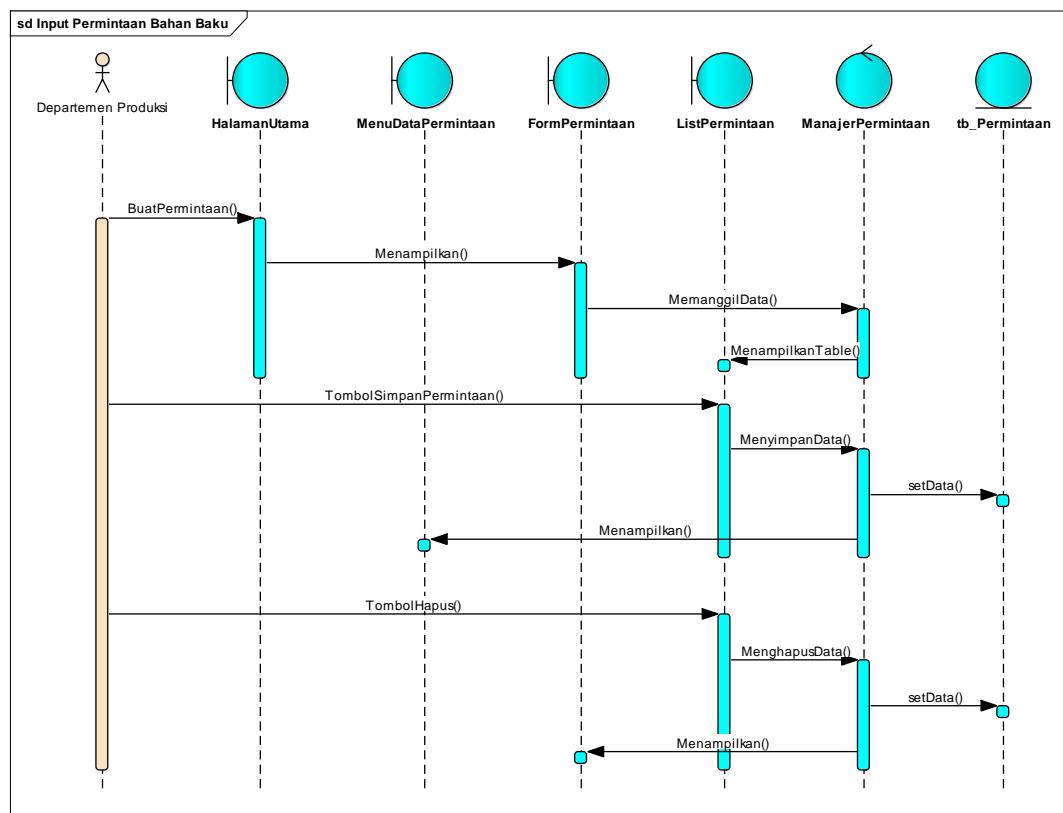


Gambar V.17 *Sequence Diagram* Proses Mengelola Data Master User

(Sumber: Hasil Analisis, 2019)

5. Sequence Diagram Menginput Permintaan Bahan Baku

Sequence diagram menginput permintaan bahan baku merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses untuk menginput permintaan bahan baku. Proses tersebut dilakukan oleh Departemen produksi agar dapat melakukan permintaan bahan baku. Berikut merupakan *sequence diagram* menginput permintaan bahan baku yang dapat dilihat pada Gambar V.18:



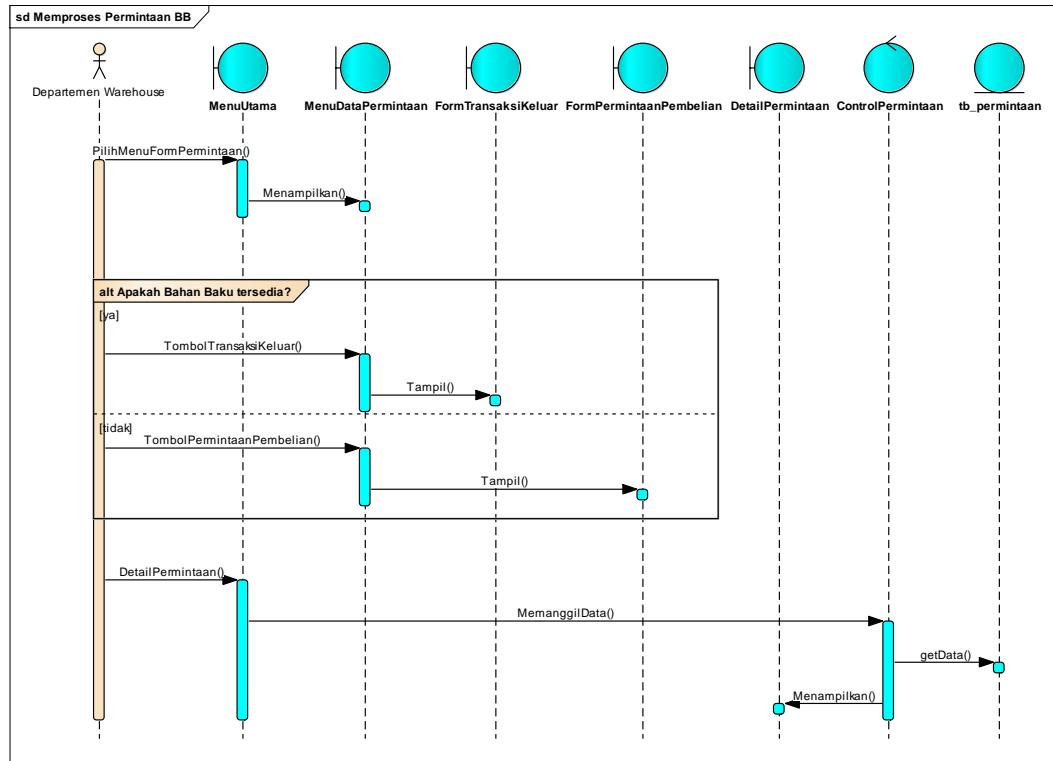
Gambar V.18 *Sequence Diagram* Proses Menginput Permintaan Bahan Baku

(Sumber: Hasil Analisis, 2019)

6. Sequence Diagram Memproses Permintaan Bahan Baku

Sequence diagram memproses permintaan bahan baku merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses untuk memproses permintaan bahan baku. Proses tersebut dilakukan oleh Departemen *warehouse* agar dapat memproses permintaan bahan baku. Berikut

merupakan *sequence diagram* memproses permintaan bahan baku yang dapat dilihat pada Gambar V.19:

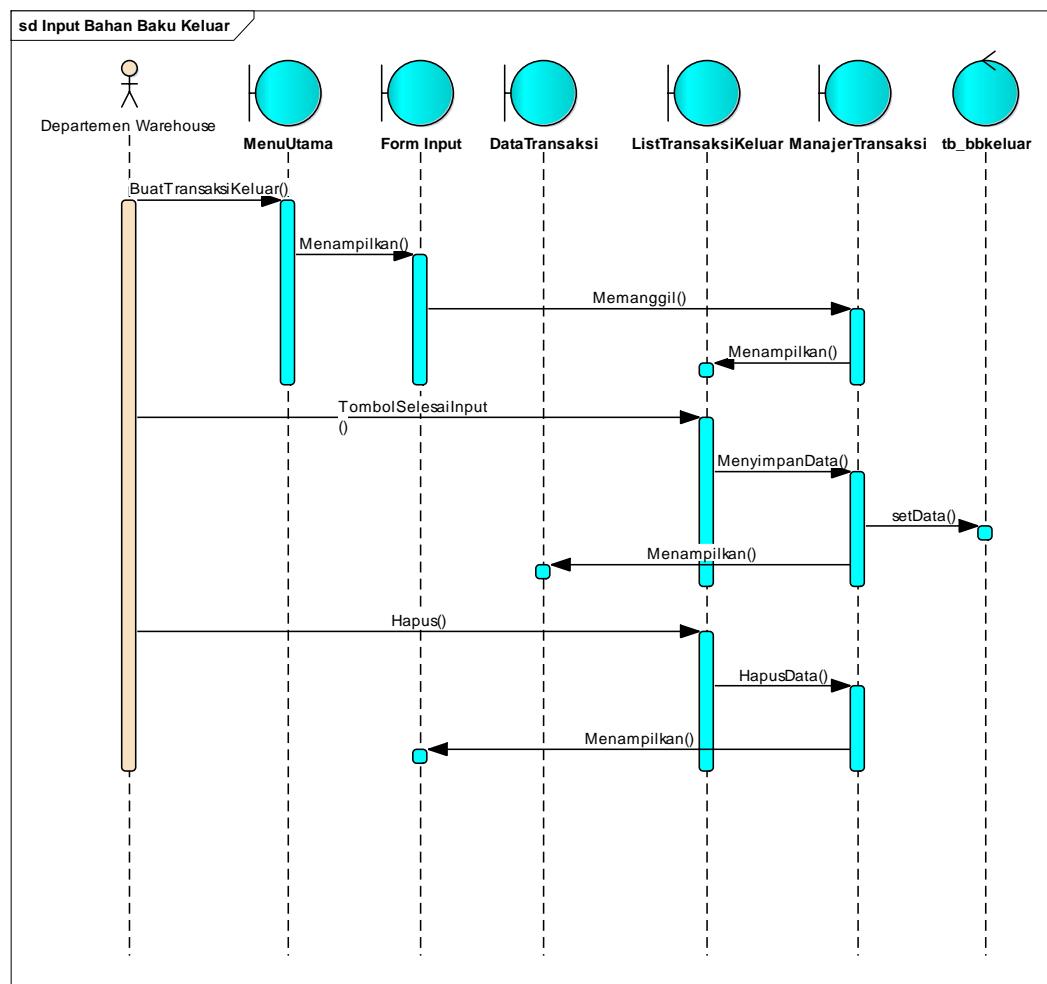


Gambar V.19 Sequence Diagram Memproses Permintaan Bahan Baku

(Sumber: Hasil Analisis, 2019)

7. Sequence Diagram Menginput Transaksi Bahan Baku Keluar

Sequence diagram menginput transaksi bahan baku keluar merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses transaksi bahan baku keluar. Proses tersebut dilakukan oleh departemen *warehouse* agar dapat melakukan transaksi bahan baku keluar. Berikut merupakan *sequence diagram* memproses permintaan bahan baku yang dapat dilihat pada Gambar V.20:

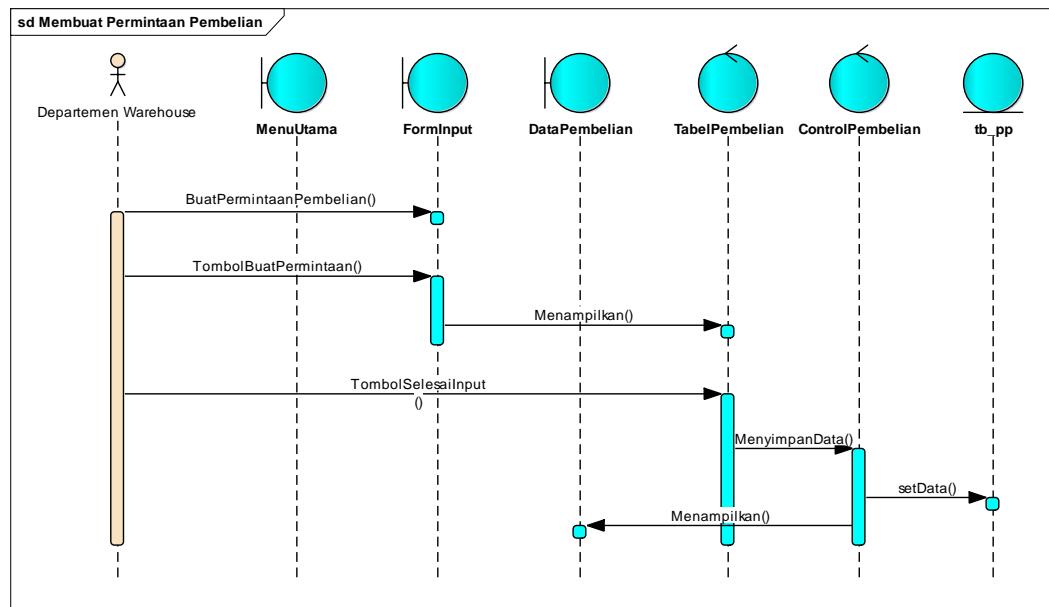


Gambar V.20 *Sequence Diagram* Proses Menginput Transaksi Bahan Baku Keluar

(Sumber: Hasil Analisis, 2019)

8. *Sequence Diagram* Membuat Permintaan Pembelian

Sequence diagram membuat permintaan pembelian merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses membuat permintaan pembelian. Proses tersebut dilakukan oleh Departemen warehouse agar dapat melakukan permintaan pembelian bahan baku. Berikut merupakan *sequence diagram* membuat permintaan pembelian yang dapat dilihat pada Gambar V.21:

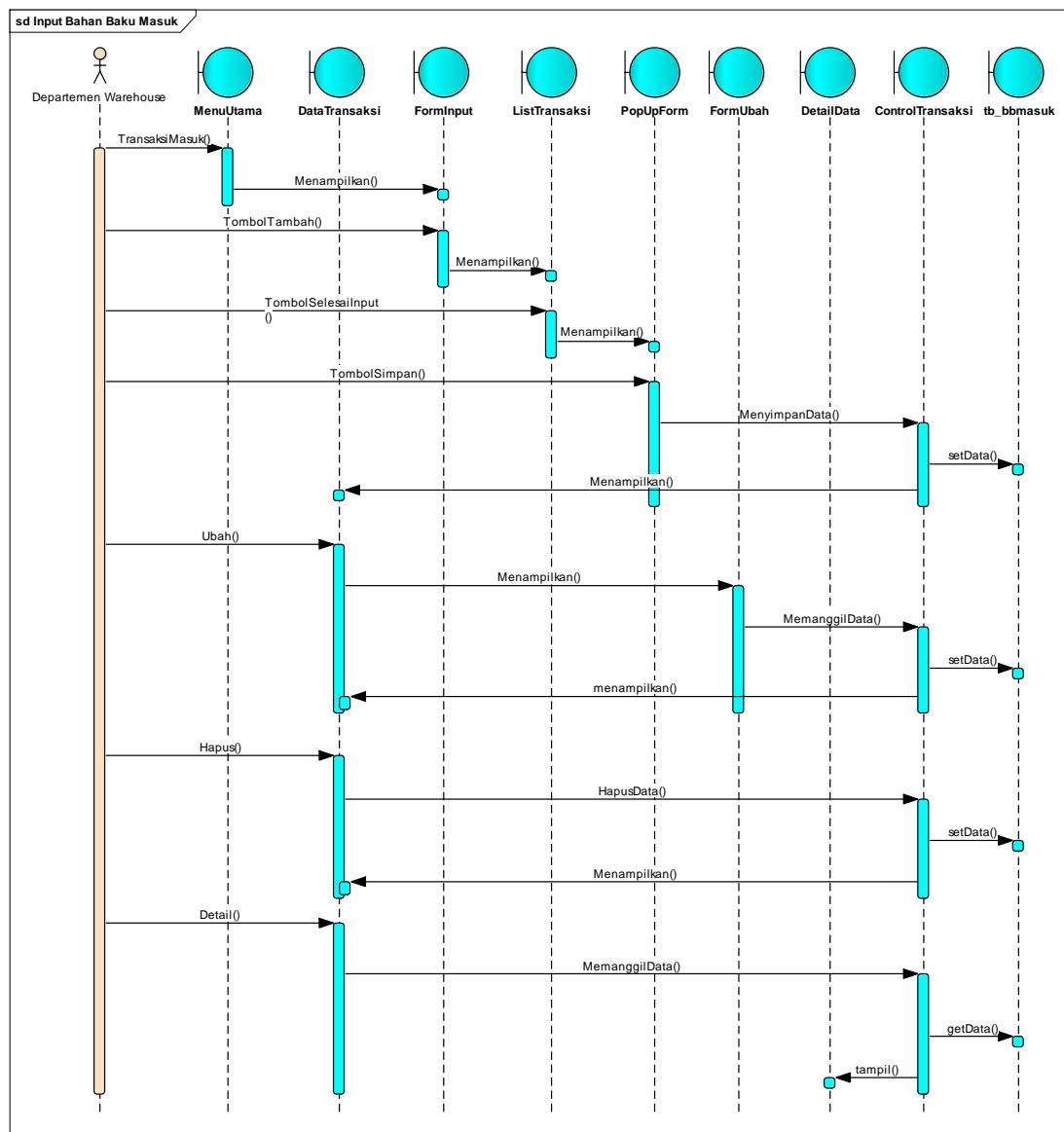


Gambar V.21 Sequence Diagram Proses Membuat Permintaan Pembelian

(Sumber: Hasil Analisis, 2019)

9. Sequence Diagram Menginput Transaksi Bahan Baku Masuk

Sequence diagram menginput transaksi bahan baku masuk merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses menginput transaksi masuknya bahan baku. Proses tersebut dilakukan oleh Departemen warehouse agar dapat melakukan *input* data bahan baku masuk. Berikut merupakan *sequence diagram* menginput transaksi bahan baku masuk yang dapat dilihat pada Gambar V.22:

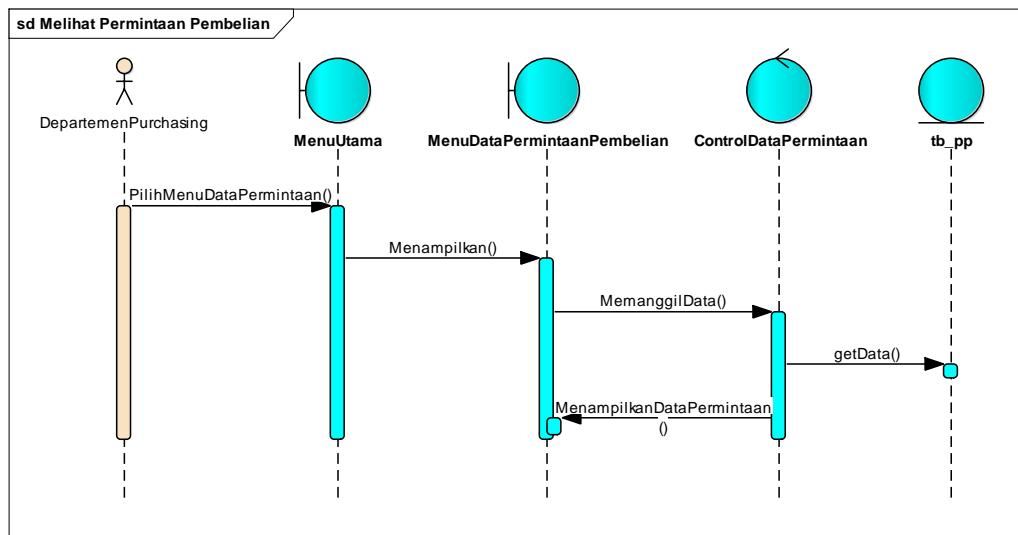


Gambar V.22 Sequence Diagram Proses Menginput Transaksi Bahan Baku Masuk

(Sumber: Hasil Analisis, 2019)

10. Sequence Diagram Melihat Permintaan Pembelian

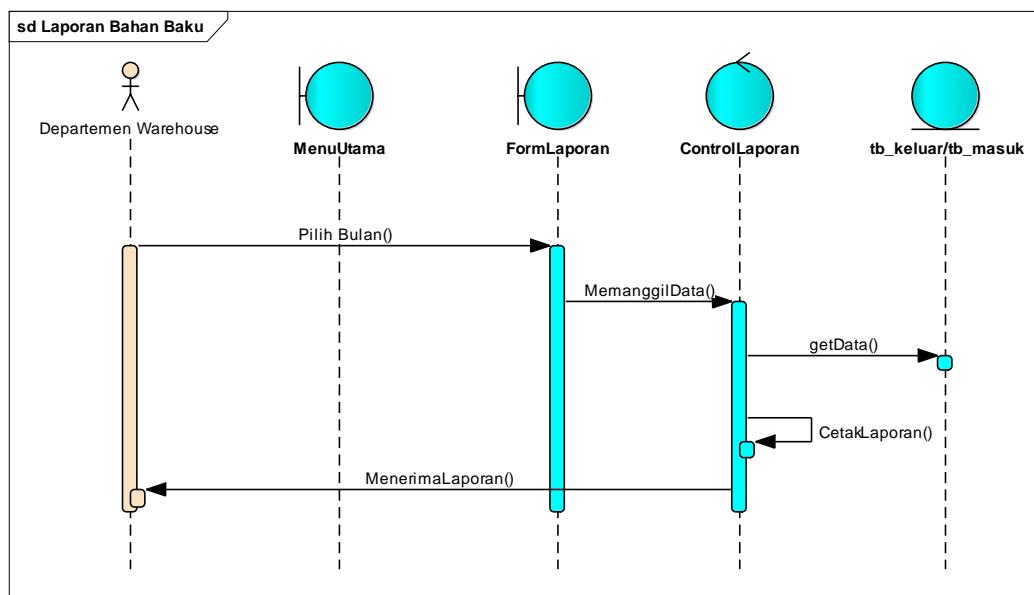
Sequence diagram melihat permintaan pembelian merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses untuk melihat permintaan pembelian. Proses tersebut dilakukan oleh departemen *purchasing* agar dapat melihat data permintaan pembelian. Berikut merupakan *sequence diagram* melihat data permintaan pembelian yang dapat dilihat pada Gambar V.23:



Gambar V.23 *Sequence Diagram* Proses Melihat Data Pemintaan Pembelian
 (Sumber: Hasil Analisis, 2019)

11. *Sequence Diagram* Melihat dan Mencetak Laporan

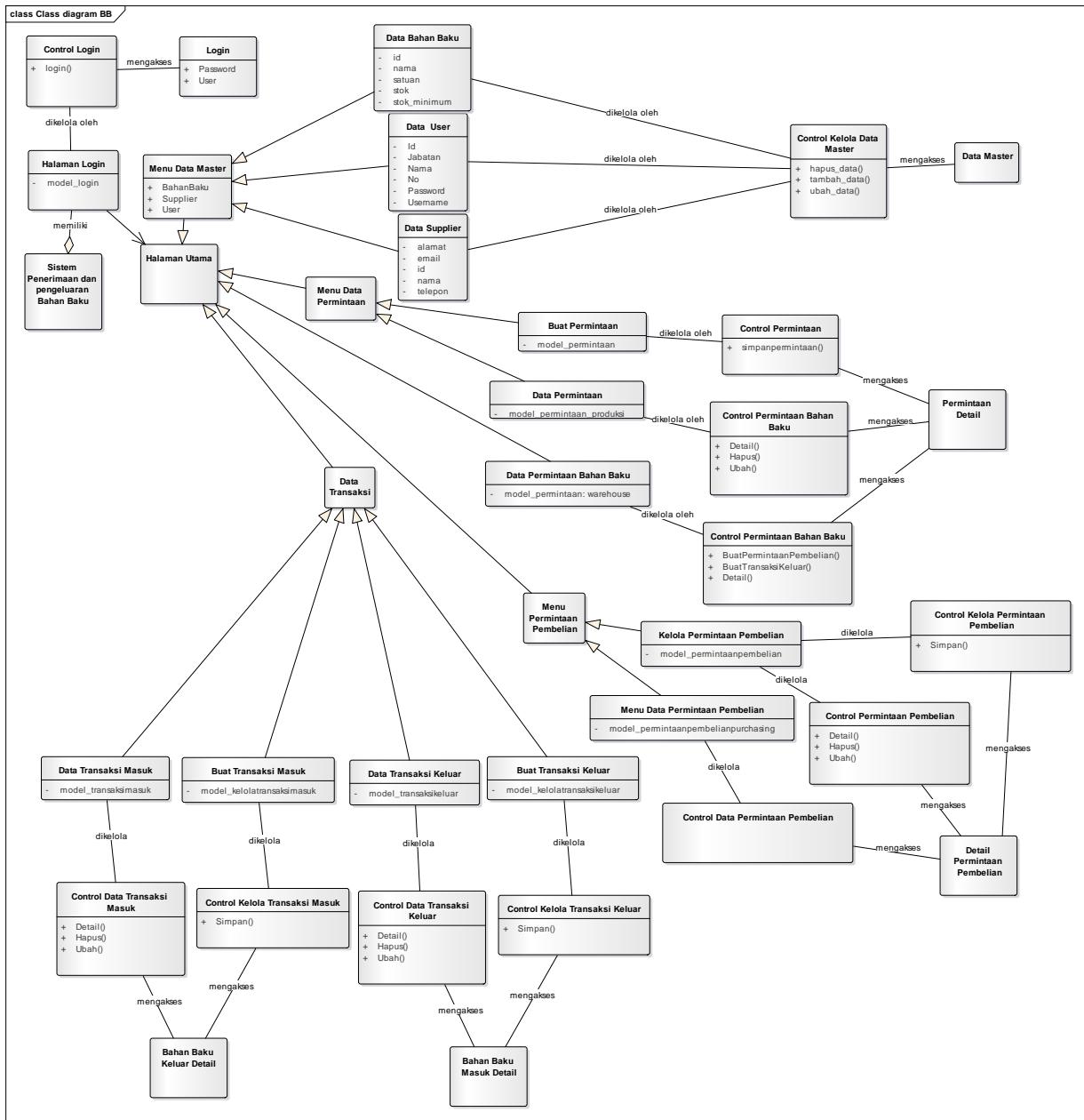
Sequence diagram melihat dan mencetak laporan merupakan sebuah diagram *sequence* yang menjelaskan interaksi objek-objek dalam sebuah proses mencetak laporan. Berikut merupakan *sequence diagram* mencetak laporan yang dapat dilihat pada Gambar V.24:



Gambar V.24 *Sequence Diagram* Proses Melihat dan Mencetak Laporan
 (Sumber: Hasil Analisis, 2019)

5.3.4. Class Diagram

Class diagram pada usulan sistem informasi penerimaan dan pengeluaran bahan baku digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat, untuk membangun sistem *class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. Berikut adalah *class diagram* usulan sistem informasi penerimaan dan pengeluaran bahan baku pada Gambar V.25.

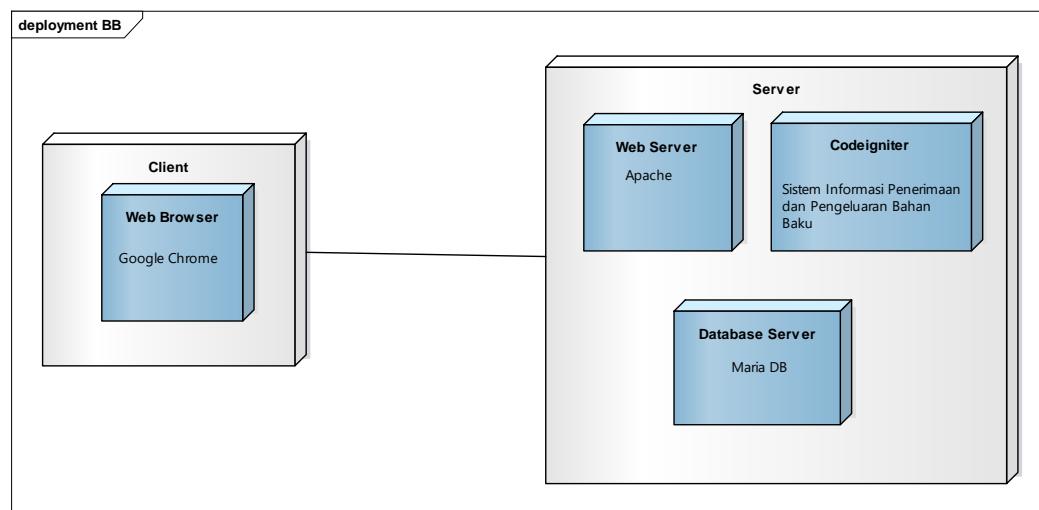


Gambar V.25 *Class Diagram* Usulan untuk Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku

(Sumber: Hasil Analisis, 2019)

5.3.5. Deployment Diagram

Deployment diagram pada usulan sistem informasi penerimaan dan pengeluaran bahan baku digunakan untuk mewakili *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi, berikut Gambar V.26 merupakan *deployment diagram* sistem informasi penerimaan dan pengeluaran bahan baku.



Gambar V.26 *Deployment Diagram* Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku

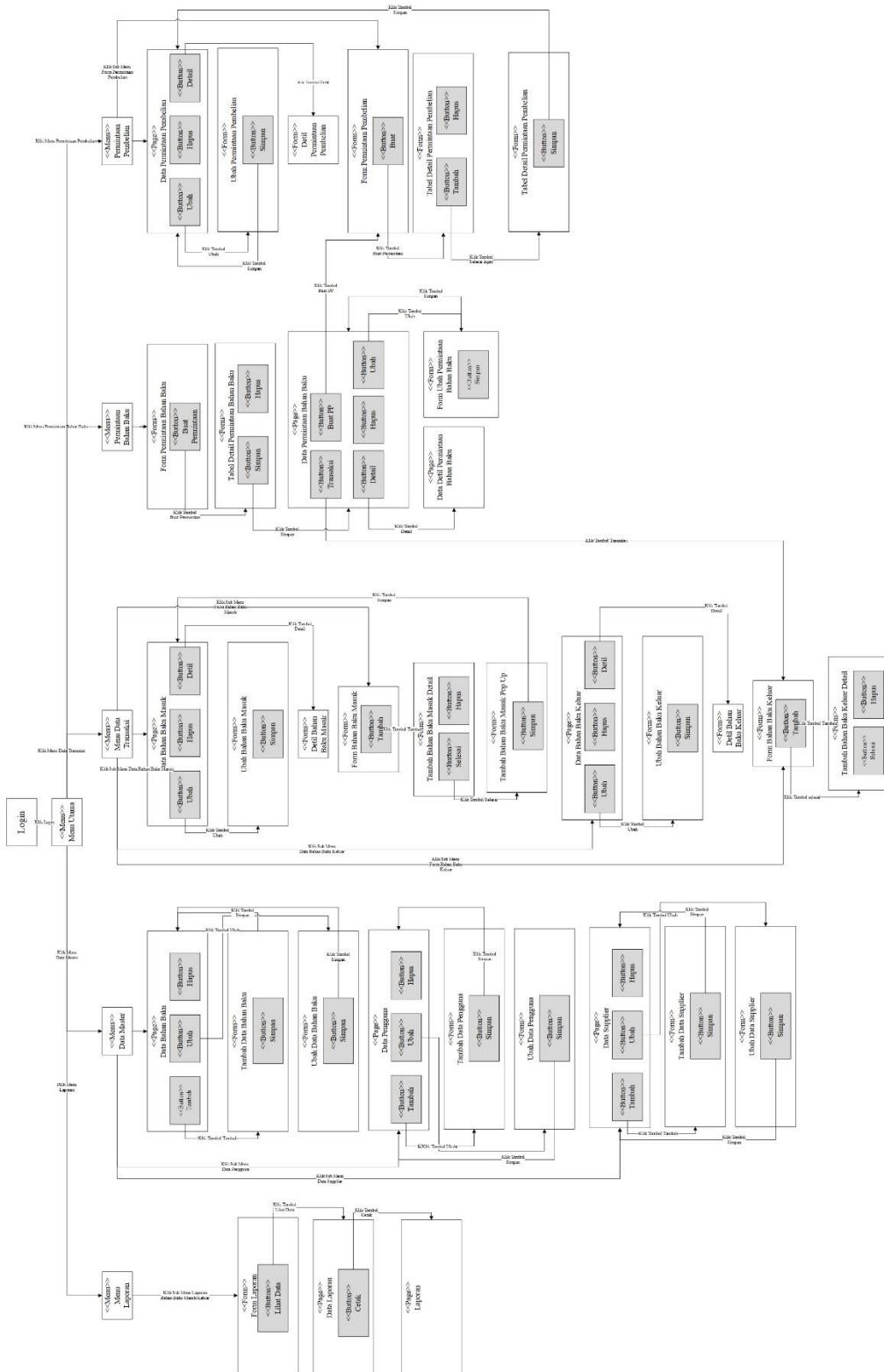
(Sumber: Hasil Analisis, 2019)

5.4 Windows Navigation Diagram (WND)

Windows Navigation Diagram merupakan *statechart* diagram khusus yang berfokus pada *user interface*. Yang mana menunjukkan bagaimana semua layar, formulir, dan laporan terkait, menunjukkan bagaimana pengguna bergerak dari satu ke yang lain dan seperti diagram keadaan untuk *user interface*:

- Kotak mewakili komponen.
- Panah mewakili transisi.
- Stereotype menunjukkan tipe antarmuka.

Windows Navigation Diagram usulan dapat dilihat pada Gambar V.27.



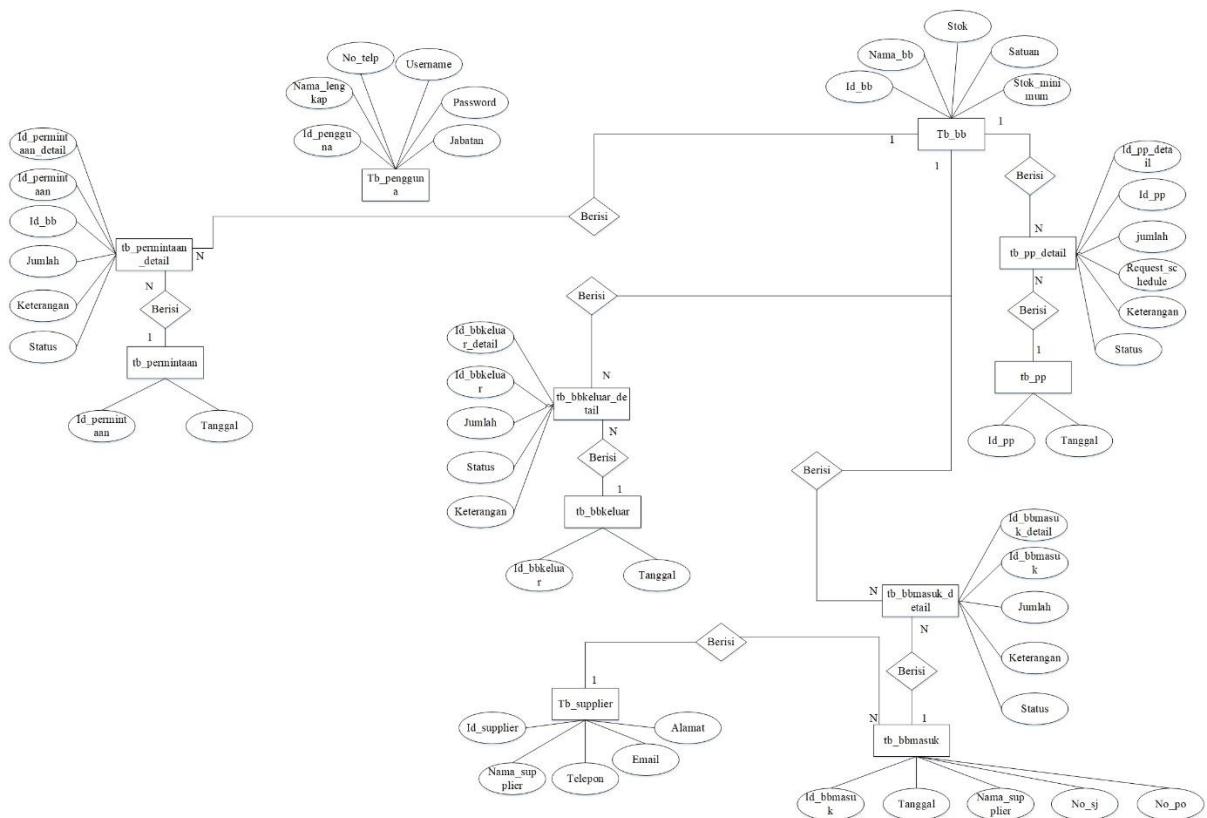
Gambar V.27 Windows Navigation Diagram Sistem Informasi Penerimaan dan

Pengeluaran Bahan Baku

(Sumber: Hasil Analisis, 2019)

5.5 Pemodelan Data Sistem Usulan

Pemodelan data pada sistem informasi penerimaan dan pengeluaran bahan baku di PT Tekniktama Karya Mandiri menggunakan dua cara yaitu *Entity Relationship Diagram* (ERD) yang digunakan untuk menjelaskan hubungan antar data dalam basis data dan kamus data yang digunakan untuk menjelaskan isi dari *database* yang digunakan dalam sistem usulan.



Gambar V.28 *Entity Relationship Diagram* Sistem Informasi Penerimaan dan Pengeluaran Bahan Baku
 (Sumber: Hasil Analisis, 2019)

5.5.1 Kamus Data

Kamus data merupakan daftar data yang terdapat dalam sebuah sistem dengan maksud untuk mendefinisikan aliran data di dalam sebuah sistem dengan lengkap dan sesuai dengan sistem, sehingga pengguna mempunyai pengertian yang sama tentang *input*, *output* dan komponen data *store*. Pada tahap perancangan

sistem, kamus data digunakan untuk merancang basis data. Berikut adalah kamus data pada sistem informasi penerimaan dan pengeluaran bahan baku.

1. Spesifikasi Tabel Data Bahan Baku

Nama Tabel : Data Bahan Baku
 Akronim : tb_bb
 Fungsi : Untuk menyimpan data bahan baku agar dapat mengetahui stok bahan baku.
 Tipe : Data Master

Tabel V.13 Tabel Data Bahan Baku

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|-----------------|--------------|---------|---------|-------------|
| 1. | Id Bahan Baku | id_bb | Varchar | 10 | Primary Key |
| 2. | Nama Bahan Baku | Nama_bb | Varchar | 150 | |
| 3. | Satuan | Satuan | Varchar | 30 | |
| 4. | Stok | Stok | Integer | 11 | |
| 5. | Stok Minimum | Stok_minimum | Integer | 11 | |

(Sumber: Hasil Analisis,2019)

2. Spesifikasi Tabel Data Pengguna

Nama Tabel : Data Pengguna
 Akronim : tb_pengguna
 Fungsi : Untuk menyimpan data pengguna dan digunakan untuk hak akses ketika melakukan login.
 Tipe : Data Master

Tabel V.14 Tabel Data Pengguna

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|---------------|--------------|---------|---------|-------------|
| 1. | Id User | Id_pengguna | Varchar | 11 | Primary Key |
| 2. | Nama Pengguna | Username | Varchar | 50 | |
| 3. | Nama Lengkap | Nama_lengkap | Varchar | 50 | |
| 4. | Jabatan | Jabatan | Text | | |

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|-------------|----------|---------|---------|------------|
| 5. | Password | Password | Varchar | 30 | |

(Sumber: Hasil Analisis, 2019)

3. Spesifikasi Tabel Data *Supplier*

- Nama Tabel : Data *Supplier*
 Akronim : tb_supplier
 Fungsi : Untuk menyimpan data *supplier* dan digunakan saat akan melakukan permintaan pembelian bahan baku.
 Tipe : Data Master

Tabel V.15 Tabel Data *Supplier*

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|---------------|---------------|---------|---------|-------------|
| 1. | Id Supplier | Id_supplier | Varchar | 10 | Primary Key |
| 2. | Nama Supplier | Nama_supplier | Varchar | 50 | |
| 3. | Email | Email | Varchar | 50 | |
| 4. | Nomor Telepon | Nomor | Varchar | 12 | |
| 5. | Alamat | Alamat | Varchar | 150 | |

(Sumber: Hasil Analisis, 2019)

4. Spesifikasi Tabel Transaksi Bahan Baku Keluar

- Nama Tabel : Data Transaksi Bahan Baku Keluar
 Akronim : tb_bbkeluar
 Fungsi : Untuk menyimpan data transaksi bahan baku yang keluar dari gudang.
 Tipe : Data Transaksi

Tabel V.16 Tabel Data Transaksi Bahan Baku Keluar

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|----------------------|-------------|---------|---------|-------------|
| 1. | Id Bahan Baku Keluar | Id_bbkeluar | Varchar | 10 | Primary Key |
| 2. | Tanggal | Tanggal | Date | | |

(Sumber: Hasil Analisis, 2019)

5. Spesifikasi Tabel Transaksi Bahan Baku Keluar Detail

Nama Tabel : Data Transaksi Bahan Baku Keluar Detail
 Akronim : tb_bbkeluar_detail
 Fungsi : Untuk menyimpan data transaksi bahan baku yang keluar dari gudang secara detail.
 Tipe : Data Transaksi

Tabel V.17 Tabel Data Transaksi Bahan Baku Keluar Detail

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|-----------------------------|--------------------|---------|---------|-------------|
| 1. | Id Bahan Baku Keluar Detail | Id_bbkeluar_detail | Varchar | 11 | Primary Key |
| 2. | Id Bahan Baku Keluar | Id_bbkeluar | Varchar | 10 | Foreign Key |
| 3. | Id Bahan Baku | Id_bb | Varchar | 10 | Foreign Key |
| 4. | Jumlah | Jumlah | Integer | 11 | |
| 5. | Keterangan | Keterangan | Varchar | 50 | |
| 6. | Status | Status | Varchar | 20 | |

(Sumber: Hasil Analisis, 2019)

6. Spesifikasi Tabel Transaksi Bahan Baku Masuk

Nama Tabel : Data Transaksi Bahan Baku Masuk
 Akronim : tb_bbmasuk
 Fungsi : Untuk menyimpan data transaksi bahan baku yang masuk dari *supplier*.
 Tipe : Data Transaksi

Tabel V.18 Tabel Data Transaksi Bahan Baku Masuk

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|---------------------|---------------|---------|---------|-------------|
| 1. | Id Bahan Baku Masuk | Id_bbmasuk | Varchar | 10 | Primary Key |
| 2. | Tanggal | Tanggal | Date | | |
| 3. | Nama Supplier | Nama_Supplier | Varchar | 50 | |

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|-------------------|---------|---------|---------|------------|
| 4. | Nomor Surat jalan | No_sj | Varchar | 11 | |
| 5. | Nomor PO | No_po | Varchar | 50 | |

(Sumber: Hasil Analisis, 2019)

7. Spesifikasi Tabel Transaksi Bahan Baku Masuk Detail

- Nama Tabel : Data Transaksi Bahan Baku Masuk Detail
 Akronim : tb_bbmasuk_detail
 Fungsi : Untuk menyimpan data transaksi bahan baku yang masuk dari *supplier* secara detail.
 Tipe : Data Transaksi

Tabel V.19 Tabel Data Transaksi Bahan Baku Masuk Detail

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|----------------------------|-------------------|---------|---------|-------------|
| 1. | Id Bahan Baku Masuk Detail | Id_bbmasuk_detail | Varchar | 11 | Primary Key |
| 2. | Id Bahan Baku Masuk | Id_bbmasuk | Varchar | 10 | Foreign Key |
| 3. | Id Bahan Baku | Id_bb | Varchar | 10 | Foreign Key |
| 4. | Jumlah | Jumlah | Integer | 11 | |
| 5. | Keterangan | Keterangan | Varchar | 50 | |
| 6. | Status | Status | Varchar | 20 | |

(Sumber: Hasil Analisis, 2019)

8. Spesifikasi Tabel Permintaan Bahan Baku

- Nama Tabel : Data Permintaan Bahan Baku
 Akronim : tb_permintaan
 Fungsi : Untuk menyimpan data permintaan bahan baku.
 Tipe : Data Transaksi

Tabel V.20 Tabel Permintaan Bahan Baku

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|---------------|---------------|---------|---------|-------------|
| 1. | Id Permintaan | Id_permintaan | Varchar | 10 | Primary Key |
| 2. | Tanggal | Tanggal | Date | | |

(Sumber: Hasil Analisis, 2019)

9. Spesifikasi Tabel Permintaan Bahan Baku Detail

- Nama Tabel : Data Permintaan Bahan Baku Detail
 Akronim : tb_permintaan_detail
 Fungsi : Untuk menyimpan data permintaan bahan baku secara detail.
 Tipe : Data Transaksi

Tabel V.21 Tabel Permintaan Bahan Baku Detail

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|----------------------|----------------------|---------|---------|-------------|
| 1. | Id Permintaan Detail | Id_permintaan_detail | Varchar | 11 | Primary Key |
| 2. | Id Permintaan | Id_permintaan | Varchar | 10 | Foreign Key |
| 3. | Id Bahan Baku | Id_bb | Varchar | 10 | Foreign Key |
| 4. | Jumlah | Jumlah | Integer | 11 | |
| 5. | Keterangan | Keterangan | Varchar | 50 | |
| 6. | Status | Status | Varchar | 20 | |

(Sumber: Hasil Analisis, 2019)

10. Spesifikasi Tabel Permintaan Pembelian

- Nama Tabel : Data Permintaan Pembelian Bahan Baku
 Akronim : tb_pp
 Fungsi : Untuk menyimpan data permintaan pembelian bahan baku.
 Tipe : Data Transaksi

Tabel V.22 Tabel Permintaan Pembelian

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|-------------------------|---------|---------|---------|-------------|
| 1. | Id Permintaan Pembelian | Id_pp | Varchar | 10 | Primary Key |
| 2. | Tanggal | Tanggal | Date | | |

(Sumber: Hasil Analisis, 2019)

11. Spesifikasi Tabel Permintaan Pembelian Detail

Nama Tabel : Data Permintaan Pembelian Bahan Baku

Akronim : tb_pp_detail

Fungsi : Untuk menyimpan data permintaan pembelian bahan baku.

Tipe : Data Transaksi

Tabel V.23 Tabel Permintaan Pembelian Detail

| No. | Nama Elemen | Akronim | Tipe | Panjang | Keterangan |
|-----|--------------------------------|------------------|---------|---------|-------------|
| 1. | Id Permintaan Pembelian Detail | Id_pp_detail | Varchar | 11 | Primary Key |
| 2. | Id Permintaan Pembelian | Id_pp | Varchar | 10 | Foreign Key |
| 3. | Id Bahan Baku | Id_bb | Varchar | 10 | Foreign Key |
| 4. | Jumlah | Jumlah | Integer | 11 | |
| 5. | <i>Request Schedule</i> | Request_schedule | Date | | |
| 6. | Keterangan | Keterangan | Varchar | 50 | |
| 7. | Status | Status | Varchar | 20 | |

(Sumber: Hasil Analisis, 2019)

5.6 Implementasi Sistem

Tahap ini adalah tahap pengkodean program menggunakan Bahasa pemrograman PHP dengan *framework* *CodeIgniter* dan dibantu dengan *Sublime Text 3* sebagai aplikasi *editor*. Setiap *interface* berisikan kode program agar dapat dijalankan sesuai fungsinya. Untuk mendukung kebutuhan implementasi sistem diperlukan suatu spesifikasi perangkat lunak (*software*) maupun perangkat keras (*hardware*). Adapun spesifikasinya sebagai berikut.

1. Analisis Kebutuhan *Software*
 - a. Sistem Operasi : *Microsoft Windows 7*
 - b. *Database Server* : MariaDB
 - c. Bahasa Pemrograman : PHP 5 keatas
 - d. Browser : Google Chrome (*best view*), Mozilla Firefox
 - e. Framework : CodeIgniter
 2. Analisis Kebutuhan *Hardware*
 - a. *Processor* : Minimal Processor Pentium IV
 - b. RAM : Minimal RAM 512 MB
 - c. *Harddisk* : Minimal Harddisk 64 GB

Mouse, Keyboard, Monitor sebagai peralatan muka.

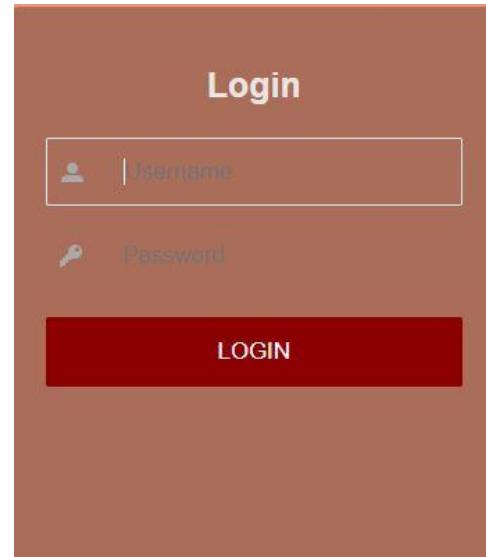
5.7 Perancangan Sistem Antarmuka

Rancangan *interface* (antarmuka) dari program sistem informasi penerimaan dan pengeluaran bahan baku ini bertujuan untuk mengambarkan antarmuka aplikasi yang telah dibuat. Rancangan antarmuka ini merupakan bagian yang berhubungan langsung antara pengguna dengan sistem, berikut rancangan antarmuka sistem informasi penerimaan dan pengeluaran bahan baku:

- ## 1. Form Login

Form Login merupakan form yang digunakan oleh pengguna sebagai tampilan awal sebelum masuk ke dalam halaman utama aplikasi. Pengguna harus

memasukan *username* dan *password* yang terdapat di halaman *login* agar dapat masuk ke dalam halama utama aplikasi.



Gambar V.29 Tampilan Form *Login*

(Sumber: Hasil Analisis, 2019)

2. Tampilan Halaman Utama

Tampilan Halaman Utama merupakan tampilan yang muncul pertama kali ketika pengguna telah *login* ke dalam aplikasi. Berikut ini Tampilan halaman utama yang dapat dilihat pada Gambar V.30.



Gambar V.30 Tampilan Halaman Utama

(Sumber: Hasil Analisis, 2019)

3. Tampilan Halaman Data Pengguna

Halaman data master pengguna merupakan halaman untuk pengguna (Departemen *warehouse*) untuk melihat, mengubah, mencari, dan menambah data pengguna. Berikut ini merupakan tampilan halaman data master pengguna yang dapat dilihat pada Gambar V.31. Form untuk tambah data pengguna dapat dilihat pada Gambar V.32.

The screenshot shows a web application interface for managing user data. On the left, there is a vertical sidebar with a red background containing navigation links: 'Halaman Utama', 'Data Master', 'Data Permintaan BB', 'Data Transaksi', 'Permintaan Pembelian', 'Laporan', and 'Log Out'. The main content area has a white background. At the top, it says 'Data User'. Below that is a green button labeled 'Tambah Data'. A table displays five user entries with columns: Nama Lengkap, No Telepon, Username, Password, Jabatan, and Aksi (Actions). Each entry has 'Ubah' and 'Hapus' buttons. At the bottom of the table, it says 'Showing 1 to 5 of 5 entries'. The footer of the page includes the text 'PT Teknikatama Karya Mandiri © 2019 Politeknik STMI'.

Gambar V.31 Tampilan Halaman Data Pengguna

(Sumber: Hasil Analisis, 2019)

This screenshot shows the same application interface as in Gambar V.31, but with a modal window open in the center. The modal is titled 'Tambah Data Pengguna'. It contains input fields for 'Nama Lengkap', 'Nomor Telepon', 'Username', 'Password', and 'Jabatan'. Below these fields is a blue 'Simpan' button. In the background, the 'Data User' table is visible, showing the same five user entries as in the previous screenshot. The footer 'PT Teknikatama Karya Mandiri © 2019 Politeknik STMI' is also present.

Gambar V.32 Tampilan Halaman *Form Input* Data Pengguna

(Sumber: Hasil Analisis, 2019)

4. Tampilan Halaman Data *Supplier*

Halaman data master *supplier* merupakan tampilan untuk pengguna (Departemen *warehouse*) untuk melihat, mengubah, menambah, mencari data *supplier*. Berikut ini merupakan tampilan halaman data master *supplier* yang dapat dilihat pada Gambar V.33 *Form* untuk tambah data *supplier* dapat dilihat pada Gambar V.34.

The screenshot shows a web-based application interface for managing supplier data. On the left is a vertical navigation menu with items like 'Halaman Utama', 'Data Master', 'Data Permintaan BB', 'Data Transaksi', 'Permintaan Pembelian', 'Laporan', and 'Log Out'. The main content area has a title 'Data Supplier' and a green 'Tambah Data' button. Below it is a table with columns: ID Supplier, Nama Supplier, Email, Telepon, Alamat, and Aksi. One row is visible with data: SP-001, PT Gemala Kempa Daya, GKD@gmail.com, 082122758900, Cikarang, and two buttons: 'Ubah' (Change) and 'Hapus' (Delete). At the bottom of the table, it says 'Showing 1 to 2 of 2 entries'. The footer of the page includes the text 'PT Teknikatama Karya Mandiri © 2019 Politeknik STMI'.

Gambar V.33 Tampilan Halaman Data *Supplier*

(Sumber: Hasil Analisis, 2019)

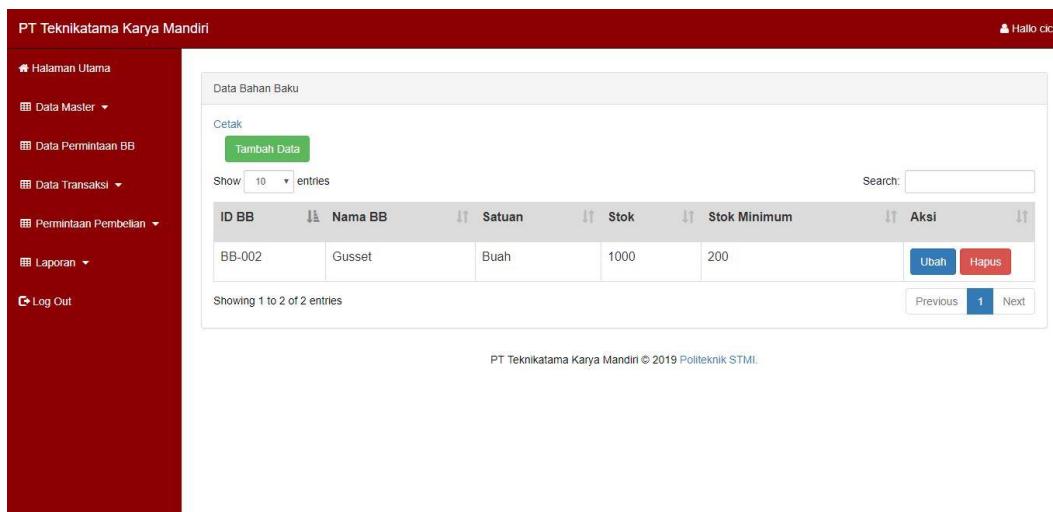
This screenshot shows a modal dialog box titled 'Tambah Data Supplier'. It contains fields for inputting supplier information: 'Kode Supplier' (SP-002), 'Nama Supplier' (Nama Supplier), 'Email' (Email), 'Telepon' (Telepon), and 'Alamat' (Alamat). At the bottom of the form is a blue 'Simpan' (Save) button. The background of the modal is semi-transparent, showing the 'Data Supplier' table from the previous screenshot.

Gambar V.34 Tampilan Halaman *Form Input* Data *Supplier*

(Sumber: Hasil Analisis, 2019)

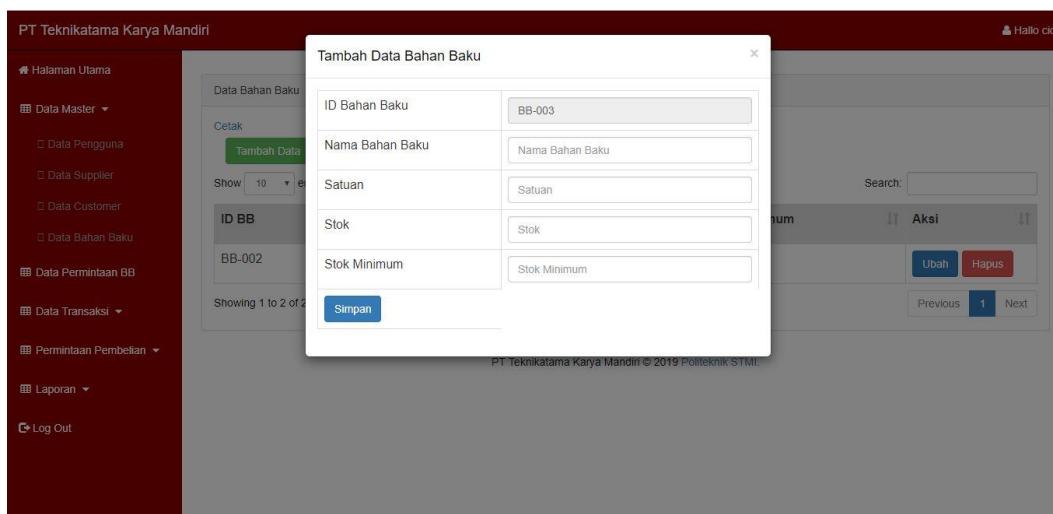
5. Tampilan Halaman Data Bahan Baku

Halaman data master bahan baku merupakan tampilan untuk pengguna (*Departemen warehouse*) untuk melihat, mengubah, menambah, mencari data bahan baku. Berikut ini merupakan tampilan halaman data master bahan baku yang dapat dilihat pada Gambar V.35. Form untuk tambah data bahan baku dapat dilihat pada Gambar V.36.



Gambar V.35 Tampilan Halaman Data Bahan Baku

(Sumber: Hasil Analisis, 2019)



Gambar V.36 Halaman *Form Input* Data Bahan Baku

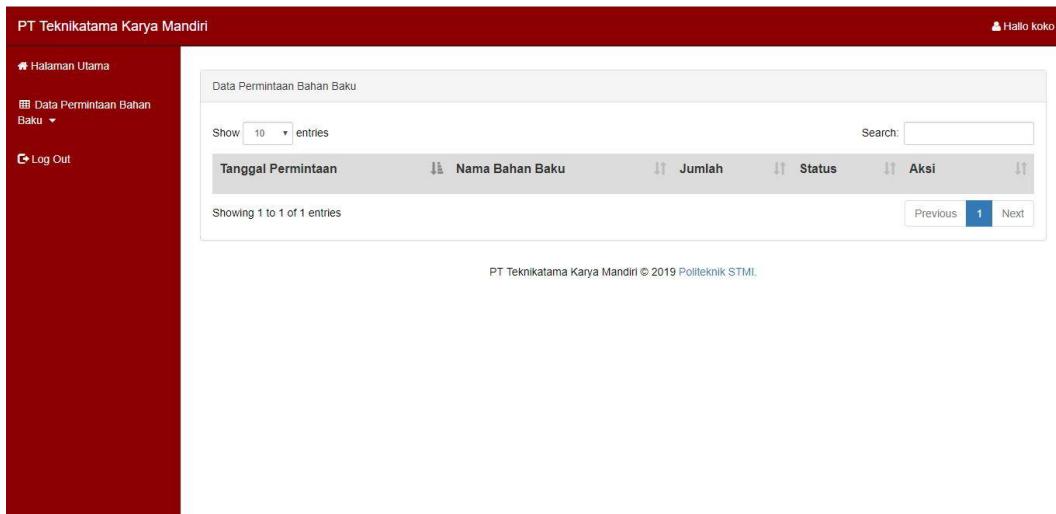
(Sumber: Hasil Analisis, 2019)

6. Tampilan Halaman Data Permintaan Bahan Baku

Halaman data permintaan baku merupakan tampilan yang muncul ketika pengguna (Departemen *warehouse*) memilih menu Data Permintaan Bahan Baku. Halaman ini juga dapat diakses oleh Departemen produksi untuk melihat data permintaan bahan baku yang telah dilakukan oleh Departemen produksi. Perbedaan tampilan Data Permintaan Bahan Baku yang dapat diakses oleh Departemen *warehouse* dan Departemen produksi terdapat pada aksi. Departemen *warehouse* dapat melihat detail, membuat transaksi keluar dan membuat permintaan pembelian seperti yang terdapat pada Gambar V.37. Aksi yang dapat dilakukan oleh Departemen produksi Pada Halaman Data Permintaan Bahan Baku adalah mengubah, menghapus, dan melihat detail Gambar V.38.

Gambar V.37 Tampilan Halaman Data Permintaan Bahan Baku pada Departemen *Warehouse*

(Sumber: Hasil Analisis, 2019)

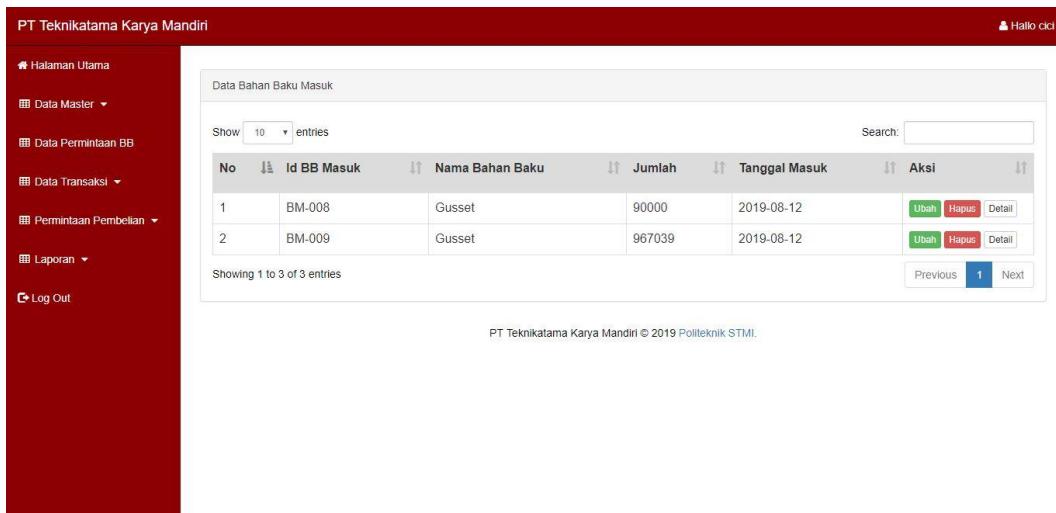


Gambar V.38 Tampilan Halaman Data Permintaan Bahan Baku pada Departemen Produksi

(Sumber: Hasil Analisis, 2019)

7. Tampilan Halaman Data Transaksi Masuk

Halaman data transaksi masuk merupakan tampilan untuk pengguna (Departemen *warehouse*) untuk melihat detail, mengubah, mencari data transaksi masuk. Berikut ini merupakan tampilan halaman data transaksi masuk yang dapat dilihat pada Gambar V.39.

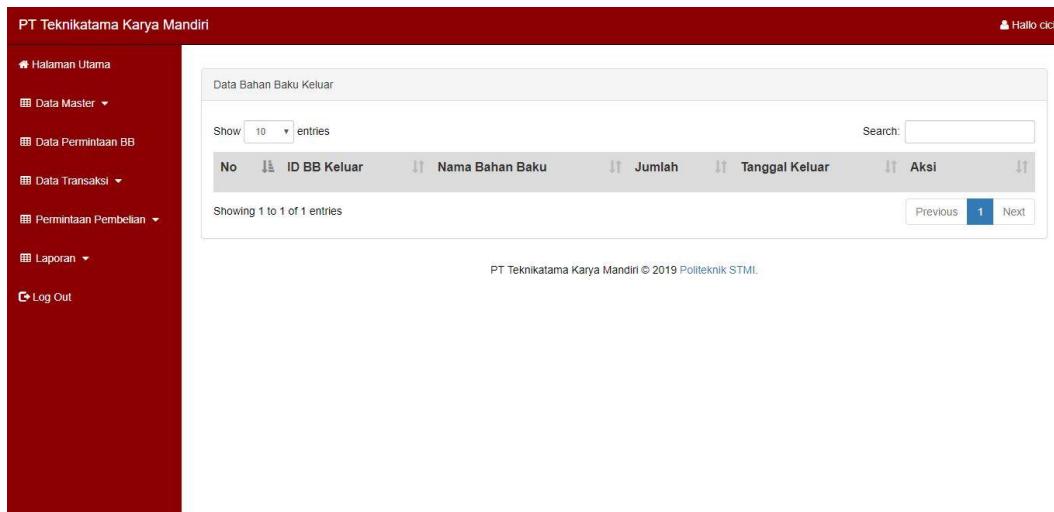


Gambar V.39 Tampilan Halaman Data Transaksi Masuk

(Sumber: Hasil Analisis, 2019)

8. Tampilan Halaman Data Transaksi Keluar

Halaman data transaksi keluar merupakan tampilan yang muncul ketika pengguna (Departemen warehouse) memilih menu Data Transaksi Bahan Baku Keluar. Halaman ini merupakan halaman untuk pengguna (Departemen warehouse) untuk melihat detail, mengubah, mencari data transaksi keluar. Berikut ini merupakan tampilan halaman data transaksi keluar yang dapat dilihat pada Gambar V.40.



Gambar V.40 Tampilan Halaman Data Transaksi Keluar

(Sumber: Hasil Analisis, 2019)

9. Tampilan Halaman *Form* Bahan Baku Masuk

Halaman *form* bahan baku masuk merupakan tampilan yang muncul ketika pengguna (Departemen warehouse) memilih menu Form Bahan Baku Masuk. Halaman ini merupakan halaman untuk pengguna (Departemen warehouse) untuk menambah data transaksi masuk. Berikut ini merupakan tampilan halaman *form* transaksi masuk yang dapat dilihat pada Gambar V.41.

Gambar V.41 Tampilan Halaman *Form* Bahan Baku Masuk

(Sumber: Hasil Analisis, 2019)

Setelah menekan tombol selesai input maka akan muncul *pop up form* untuk mengisi nama *supplier*, nomor surat jalan, nomor po, dan tanggal masuk. Tampilan *pop up form* dapat dilihat pada Gambar V.42.

Gambar V.42 Tampilan Halaman *Pop Up Form* Bahan Baku Masuk

(Sumber: Hasil Analisis, 2019)

10. Tampilan Halaman *Form* Bahan Baku Keluar

Halaman *form* bahan baku keluar merupakan tampilan yang muncul ketika pengguna (Departemen warehouse) memilih menu Form Bahan Baku

Keluar. Halaman ini merupakan halaman untuk pengguna (Departemen *warehouse*) untuk menambah data transaksi keluar.

Gambar V.43 Tampilan Halaman *Form* Bahan Baku Keluar

(Sumber: Hasil Analisis, 2019)

11. Tampilan Halaman *Form* Permintaan Pembelian

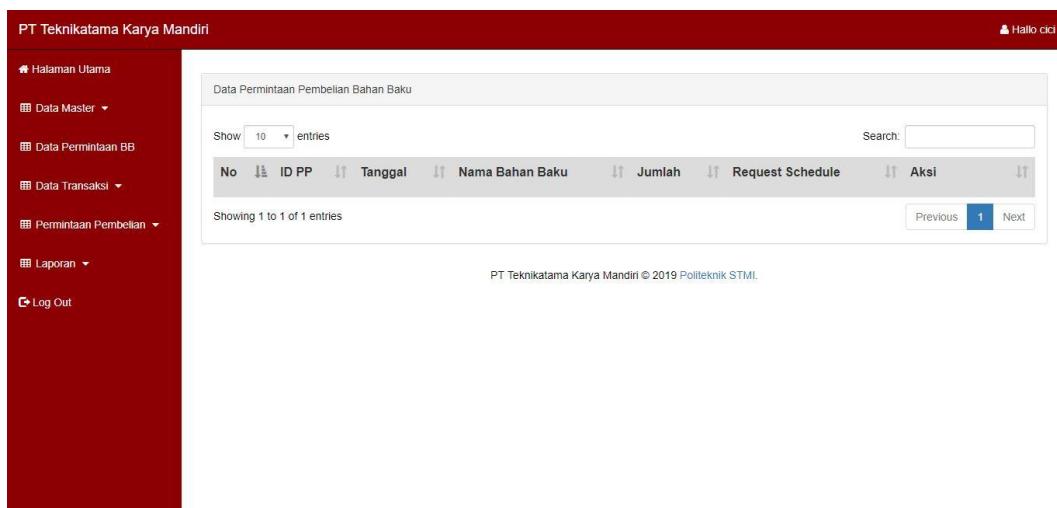
Halaman *form* permintaan pembelian bahan baku merupakan tampilan untuk pengguna untuk membuat permintaan bahan baku. Berikut ini merupakan tampilan halaman *form* permintaan pembelian yang dapat dilihat pada Gambar V.44.

Gambar V.44 Tampilan Halaman *Form* Permintaan Pembelian

(Sumber: Hasil Analisis, 2019)

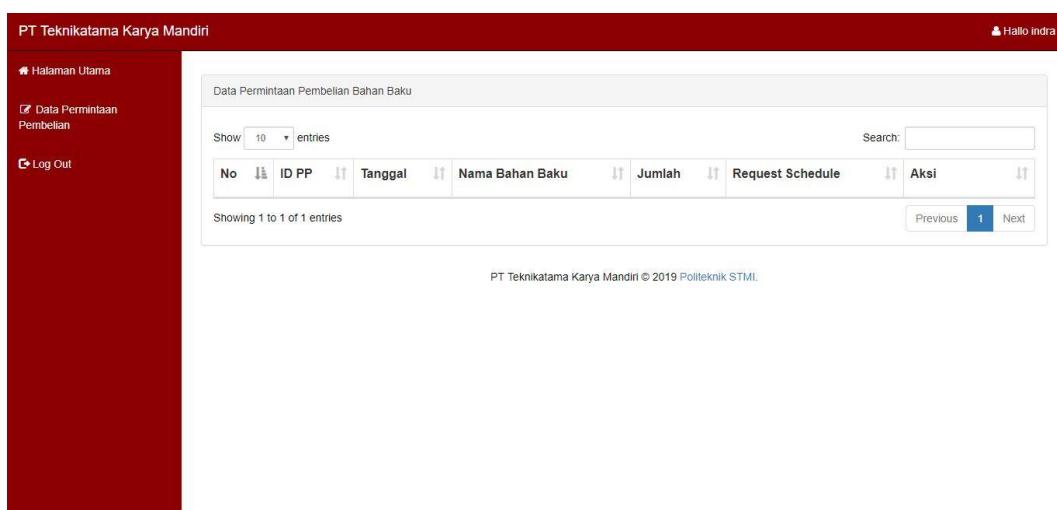
12. Tampilan Halaman Data Permintaan Pembelian

Halaman data permintaan pembelian bahan baku merupakan tampilan yang muncul ketika pengguna (Departemen *warehouse*) dan pengguna (Departemen *purchasing*) memilih menu Data Permintaan Pembelian. Halaman ini merupakan halaman untuk pengguna (Departemen *warehouse*) untuk membuat permintaan pembelian bahan baku seperti yang dapat dilihat pada Gambar V.45. Untuk pengguna (Departemen *purchasing*) hanya dapat melihat data pembelian bahan baku seperti yang dapat dilihat pada Gambar V.46.



Gambar V.45 Tampilan Halaman Data Permintaan Pembelian di Departemen *Warehouse*

(Sumber: Hasil Analisis, 2019)



Gambar V.46 Tampilan Halaman Data Permintaan Pembelian di Departemen *Purchasing*

(Sumber: Hasil Analisis, 2019)

13. Tampilan Laporan Data Transaksi Bahan Baku Masuk dan Keluar

Menu ini hanya terdapat untuk akun Departemen warehouse untuk mencetak laporan transaksi bahan baku masuk maupun keluar berdasarkan periode tertentu. Berikut ini merupakan gambar dari tampilan halaman laporan transaksi bahan baku masuk yang dapat dilihat pada Gambar V.47. Tampilan halaman laporan transaksi bahan baku keluar yang dapat dilihat pada Gambar V.48.

Gambar V.47 Tampilan Halaman Laporan Transaksi Bahan Baku Masuk

(Sumber: Hasil Analisis, 2019)

Gambar V.48 Tampilan Halaman Laporan Transaksi Bahan Baku Keluar

(Sumber: Hasil Analisis, 2019)

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil pengumpulan, pengolahan dan analisis data yang telah dilakukan dalam penelitian ini dapat disimpulkan bahwa:

1. Dengan adanya aplikasi sistem penerimaan dan pengeluaran bahan baku ini departemen *warehouse* dapat menginput data bahan baku masuk dan keluar ke dalam sistem secara cepat dan lebih tepat.
2. Dengan adanya sistem basis data untuk penerimaan dan pengeluaran bahan baku yang diharapkan dapat melakukan proses penyimpanan data dan mencegah terjadinya kehilangan data karena setiap data yang dimasukan ke dalam sistem diharapkan tersimpan dalam *database*.
3. Aplikasi sistem informasi penerimaan dan pengeluaran bahan baku ini dapat mengintegrasikan antar bagian sehingga diharapkan dapat memudahkan departemen *warehouse* dan pihak-pihak lain yang berkepentingan terhadap informasi penerimaan dan pengeluaran bahan baku.

6.2. Saran

Saran yang diberikan untuk pengembangan sistem informasi penerimaan dan pengeluaran bahan baku adalah sebagai berikut:

1. Dalam penerapan sistem baru terhadap sistem lama sebaiknya dilakukan secara bertahap, untuk menghindari terjadinya kegagalan terhadap sistem yang baru dan diperlukan adanya sosialisasi serta pengenalan untuk penggunaan sistem ini kepada bagian yang menerapkannya.
2. Sebaiknya dilakukan pemeliharaan terhadap aplikasi secara berkala sehingga aplikasi dapat berjalan dengan baik.

DAFTAR PUSTAKA

- Adelheid, A. & Nst, Khairil., 2012, *Buku Pintar Menguasai PHP MySQL*, Mediakita, Jakarta.
- Ahyari, Agus. 2013. Manajemen Produksi Perencanaan Sistem Produksi.
- Anggraeni, E. Y. & Irviani, R., 2017, *Pengantar Sistem Informasi*, CV Andi Offset, Yogyakarta.
- Davis, Gordon. 1995. Sistem Informasi Manajemen. Jakarta: PT. Pustaka Binaman Pressindo.
- Dennis, A., Wixom, Barbara Haley, & Tegarden, D., 2015, *System Analysis & Design. An Object-Oriented Approach with UML (Fifth Edition)*, Wiley, USA.
- Djahir, Y. & Pratita, D., 2014, *Sistem Informasi Manajemen*, Deepublish, Yogyakarta.
- EMS, Tim., 2016, *PHP 5 dari Nol*, PT Elex Media Komputindo, Jakarta.
- Enterprise, J., 2015., *Membuat Website PHP dengan CodeIgniter*, PT Elex Media Komputindo, Jakarta.
- Enterprise, J., 2017, *PHP Komplet*, PT Elex Media Komputindo, Jakarta.
- Enterprise,J., 2014, *MySQL Untuk Pemula*, PT Elex Media Komputindo, Jakarta.
- Fauzi, R. A., 2017, *Sistem Informasi Akuntansi (Berbasis Akuntansi)*, Deepublish, Yogyakarta.
- Hutahaean, J., 2015, *Konsep Informasi Sistem*, Deepublish, Yogyakarta.
- Indrajani., 2015, *Database Design*, PT Elex Media Komputindo, Jakarta.
- Jakarta: Bumi Aksara.

- Jalinus, N. & Ambiyar., 2016, *Media dan Sumber Pembelajaran*, Kencana, Jakarta.
- Jogiyanto, 2005, *Sistem Teknologi Informasi*, CV Andi Offset, Yogyakarta.
- Khotimah, Khusnul and Sriyanto. 2016. Perancangan dan implementasi data warehouse untuk mendukung sistem akademik (Studi kasus pada STKIP Muhammadiyah Kota Bumi. Jurnal TIM Darmajaya Vol. 02, Halaman 96-107
- Komputer, Wahana, 2010, *Paduan Belajar MySQL Database Server*, Mediakita, Jakarta.
- Komputer, Wahana., 2014, *Membuat Aplikasi SMS Gateway dengan Codeigniter*, PT Elex Media Komputindo, Jakarta.
- Ladjamudin, Al-Bahra. (2005). Analisis dan Desain Sistem Informasi. Yogyakarta : Graha Ilmu.
- Leod, Mc & P. Schell, G., 2012, *Sistem Informasi Manajemen Edisi Sepuluh*, Salemba Empat, Jakarta.
- Ludwig, Ernest E., 1991, Applied Process Design For Chemical And Petrochemical Plants, Volume I, 2nd Edition. Houston: Gulf Publishing Company Book Division.
- Martono, R., 2015, *Manajemen Logistik Terintegrasi*, PPM, Jakarta.
- Martono, Ricky. 2015. *Manajemen Logistik Terintegrasi*. Jakarta: PPM Manajemen.
- Masiyal Kholmi, 2003 "Akuntasi Biaya", Edisi Empat, Yogyakarta, BPFE.
- McLeod, Raymond Jr. 2001. Sistem Informasi Manajemen jilid Dua, Edisi Bahasa Indonesia. Jakarta: PT. Bhuana Ilmu Populer.
- Muharto & Ambarita A., 2016, *Metode Penelitian Sistem Informasi: Mengatasi Kesulitan Mahasiswa dalam Menyusun Proposal Penelitian*, Deepublish, Yogyakarta.

- Mulyani, S., 2016, *Sistem Informasi Manajemen Rumah Sakit: Analisis dan Perancangan*, Abdi Sistematika, Bandung.
- Muslihudin, M. & Oktafianto., 2016, *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*, CV Andi Offset, Yogyakarta.
- Mustaqbal M. Sidi & Firdaus F. Roeri, 2015, ‘Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis’.
- Pamungkas, Canggih A., 2017, *Pengantar dan Implementasi Basis Data*, Deepublish, Yogyakarta.
- Prawirosentono, Suyadi. 2001. Manajemen Operasi, Analisis dan Studi Kasus.
- Ragil, Wukil, 2010, *Pedoman Sosialisasi Prosedur Standar*, Mitra Wacana Media, Jakarta.
- Rukun, Kasman dan Hayadi, B. Herawan, 2018, *Sistem Informasi Berbasis Expert System*, Deepublish, Yogyakarta.
- Rusdiana & Irfan, M., 2014, *Sistem Informasi Manajemen*, CV Pustaka Setia, Bandung.
- Setyorini. 2016. Pengantar Sistem Informasi Akutansi . Deepublish.
- Sitorus, L., 2015, *Algoritma dan Pemrograman*, Andi Offset, Yogyakarta.
- Supono & Putratama, V., 2016, *Pemrograman Web dengan Menggunakan PHP dan Framework Code Igniter*, Deepublish, Yogyakarta.
- Sutabri, T., 2012, *Analisis Sistem Informasi*, CV Andi Offset, Yogyakarta.
- Tyoso, Jaluanto Sunu Punjul, 2016, *Sistem Informasi Manajemen*, Deepublish, Yogyakarta.
- Utami, Feri H. & Asnawati, 2015, *Rekayasa Perangkat Lunak*, Deepublish, Yogyakarta.

Warman, J., 2012, *Warehouse Management Cetakan Ketujuh*, Pustaka Sinar Harapan, Jakarta.

Wibowo, Singgih. 2007. Manajemen Produksi. Yogyakarta: BPFE.

Winarno, E. & Zaki, A., 2013, *Buku Sakti Pemrograman PHP*, PT Elex Media Komputindo, Jakarta.

Yayat Nurhayati, H. Acep Komara (2013), Pengaruh Pasokan Bahan Baku Terhadap Proses Produksi Dan Tingkat Penjualan Terhadap Industri Rotan Kabupaten Cirebon.

Situs:

CodeIgniter, *CodeIgniter*, dilihat 15 Juni 2019, dari www.codeigniter.com/user_guide/index.html.

<https://www.bacanulis.com/2018/07/prosedur-pembelian-permintaan-dan.html>. Diakses 08 maret 2019.

<https://www.ilmu-ekonomi-id.com/2017/07/penerimaan-dan-pengeluaran-barang-di-gudang.html/>. Diakses 08 maret 2019.

KBBI, *Kamus Besar Bahasa Indonesia*, dilihat 11 Juni 2019, dari Kbbi.web.id.

MariaDB, *Definisi Mariadb*, dilihat 15 Juni 2019, dari <https://mariadb.com/products/why-mariadb>.

Master Pendidikan, *19 Pengertian Metodologi Penelitian Menurut Para Ahli*, dilihat 24 Mei 2019, dari masterpendidikan.com/2016/03/19-pengertian-metodologi-penelitian-menurut-para-ahli.html

Wikipedia Indonesia, Analisis, dilihat 17 Juni 2019, dari Id.wikipedia.org/wiki/Analisis.

Wikipedia Indonesia, *CodeIgniter*, dilihat 05 Juni 2019, dari Id.wikipedia.org/wiki/CodeIgniter.

Wikipedia Indonesia, Gudang, dilihat 20 Juni 2019, dari [Id.m.wikipedia.org/wiki/Gudang](https://id.m.wikipedia.org/wiki/Gudang).

LAMPIRAN A

FUNCTIONAL TESTING

1. Functional Testing

Functional testing merupakan pengujian yang dilakukan untuk memeriksa apakah fungsionalitas dari aplikasi yang dibangun sesuai dengan apa yang diharapkan. *Functional testing* menguji sistem berdasarkan *use case* yang telah dibuat berikut merupakan pengujian yang dilakukan terhadap aplikasi sistem persediaan bahan baku.

a. *Functional Test Use Case Login*

Berikut merupakan *functional test case* login yang dapat dilihat pada Tabel 1.

Function : Operasi validasi login

Data Assumptions : Fungsi operasi validasi login sudah berjalan dengan baik, penggunaan huruf kapital dan huruf kecil tidak mempengaruhi pada validasi login meskipun tidak sesuai dengan data.

Deskripsi : Melakukan *login* ke dalam sistem dengan menguji kesalahan *password* dan *username*

Tabel 1 *Test Case Login*

| No. | <i>Test Case Name</i> | <i>Description</i> | <i>Expected Result</i> | <i>Actual Record</i> | <i>Kesimpulan</i> |
|-----|-----------------------|---|---|----------------------|-------------------|
| 1. | Validasi <i>Login</i> | <i>Username</i> dan <i>Password</i> tidak diisi dan klik tombol login | Gagal ke halaman utama dan kembali ke form login. | Sesuai | Valid |
| 2. | Validasi <i>Login</i> | Mengisi <i>Username</i> yang benar dan <i>Password</i> yang salah dan klik tombol login | Gagal ke halaman utama dan kembali ke form login. | Sesuai | Valid |

| No. | <i>Test Case Name</i> | <i>Description</i> | <i>Expected Result</i> | <i>Actual Record</i> | Kesimpulan |
|-----|-----------------------|--|---|----------------------|------------|
| 3. | Validasi Login | Mengisi Username yang salah dan Password yang benar dan klik tombol login. | Gagal ke halaman utama dan kembali ke form login. | Sesuai | Valid |
| 4. | Validasi Login | Mengisi Username dan password yang benar | Masuk ke halaman utama | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

b. *Functional Test Use Case Menginput Data Permintaan Bahan Baku*

Berikut merupakan *functional test case* menginput data permintaan bahan baku yang dapat dilihat pada Tabel 2.

Function : Operasi menyimpan data permintaan bahan baku

Data Assumptions : Fungsi operasi menginput dan menyimpan data permintaan bahan baku sudah berjalan dengan baik.

Deskripsi : Melakukan penginputan data permintaan bahan baku.

Tabel 2 *Test Case Menginput Data Permintaan Bahan Baku*

| No. | <i>Test Case Name</i> | <i>Description</i> | <i>Expected Result</i> | <i>Actual Record</i> | Kesimpulan |
|-----|-------------------------------|---|---|----------------------|------------|
| 1. | Membuat permintaan bahan baku | Mengisi data permintaan dengan mengisi semua kolom pada form. Dan klik tombol buat permintaan | Data yang telah diinput akan tampil di tabel. | Sesuai | Valid |
| 2. | Menyimpan data | Klik tombol simpan permintaan pada data | Data tersimpan dan sistem menampilkan data | Sesuai | Valid |

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|-------------------------------|---|--|---------------|------------|
| | permintaan bahan baku | yang telah tampil di tabel form. | permintaan bahan baku | | |
| 3. | Membuat permintaan bahan baku | Mengisi data permintaan tanpa mengisi jumlah di kolom form. Dan klik tombol buat permintaan | Data tidak tampil pada tabel form dan memberi peringatan agar mengisi jumlah permintaan. | Sesuai | Valid |
| 4. | Membuat permintaan bahan baku | Mengisi data permintaan tanpa mengisi keterangan di kolom form. Dan klik tombol buat permintaan | Data yang telah diinput akan tampil di tabel | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

c. *Functional Test Use Case Melihat Data Permintaan Bahan Baku*

Berikut merupakan *functional test case* melihat data permintaan bahan baku yang dapat dilihat pada Tabel 3.

Function : Operasi melihat data permintaan bahan baku

Data Assumptions : Fungsi operasi melihat data permintaan bahan baku.

Deskripsi : Melihat data permintaan bahan baku.

Tabel 3 *Test Case* Melihat Data Permintaan Bahan Baku

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|----------------|--|--|---------------|------------|
| 1. | Melihat data | User memilih submenu data permintaan dan sistem akan | Sistem akan menampilkan data permintaan. | Sesuai | Valid |

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|--------------------------------------|--|--|---------------|------------|
| | permintaan bahan baku | menampilkan data permintaan. | | | |
| 2. | Melihat detil permintaan bahan baku. | Klik tombol detail pada tabel permintaan bahan baku. | Sistem menampilkan detil permintaan. | Sesuai | Valid |
| 3. | Menghapus data permintaan bahan baku | Klik tombol hapus pada tabel permintaan bahan baku | Sistem akan menghapus data permintaan yang ingin dihapus dan sistem menampilkan data permintaan. | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

d. *Functional Test Use Case Mengelola Data Master*

Berikut merupakan *functional test case* mengelola data master yang dapat dilihat pada Tabel 4.

Tabel 4 *Test Case* Mengelola Data Master

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|--------------------|--|--|---------------|------------|
| 1. | Tambah data master | Menambah data master dengan mengisi semua kolom pada form. | Data berhasil ditambah | Sesuai | Valid |
| 2. | Tambah data master | Menambah data master dengan mengosongkan salah satu kolom pada form. | Muncul pesan untuk input data pada box yang dikosongkan. | Sesuai | Valid |

| No. | <i>Test Case Name</i> | <i>Description</i> | <i>Expected Result</i> | <i>Actual Record</i> | Kesimpulan |
|-----|-----------------------|---|--|----------------------|-------------------|
| 3. | Mengubah data master | <i>User</i> mengisi semua data master yang ingin diubah pada form ubah setelah klik tombol ubah pada tabel data master. | Data terubah dan sistem menampilkan data master | Sesuai | Valid |
| 4. | Mengubah data master | <i>User</i> mengosongkan salah satu data master yang ingin diubah pada form ubah setelah klik tombol ubah pada tabel data master. | Muncul pesan untuk input data pada box yang dikosongkan. | Sesuai | Valid |
| 5. | Menghapus data master | <i>User</i> klik tombol hapus pada data yang ingin dihapus. | Data master terhapus | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

e. *Functional Test Use Case Memproses Permintaan Bahan Baku*

Berikut merupakan *functional test case* memproses permintaan bahan baku yang dapat dilihat pada Tabel 5.

Function : Operasi memproses permintaan bahan baku.

Data Assumptions : Fungsi operasi memproses permintaan bahan baku sudah berjalan dengan baik.

Deskripsi : Memproses Permintaan bahan baku berdasarkan jumlah permintaan dan stok bahan baku yang tersedia.

Tabel 5 *Test Case* Memproses Permintaan Bahan Baku

| No. | <i>Test Case Name</i> | <i>Description</i> | <i>Expected Result</i> | <i>Actual Record</i> | Kesimpulan |
|-----|--|---|---|----------------------|------------|
| 1. | Proses Permintaan bahan baku berdasarkan stok dan jumlah permintaan. | <i>User</i> akan melihat stok dan jumlah permintaan apabila stok lebih besar dari permintaan maka <i>user</i> akan memilih tombol transaksi, apabila permintaan lebih besar dari stok maka <i>user</i> akan memilih tombol buat permintaan pembelian. | Tombol buat permintaan pembelian dan tombol transaksi berjalan dengan benar | Sesuai | Valid |
| 2. | Melihat detail permintaan bahan baku | <i>User</i> klik tombol detail dan sistem akan menampilkan detail permintaan. | Tombol detail berjalan dengan benar | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

f. *Functional Test Use Case* Membuat Laporan

Berikut merupakan *functional test case* membuat laporan yang dapat dilihat pada Tabel 6.

Function : Operasi membuat laporan

Data Assumptions : Fungsi operasi membuat laporan sudah berjalan dengan baik.

Deskripsi : Melakukan pembuatan laporan perbulan dengan menginput nama bulan.

Tabel 6 *Test Case* Membuat Laporan

| No. | <i>Test Case Name</i> | <i>Description</i> | <i>Expected Result</i> | <i>Actual Record</i> | Kesimpulan |
|-----|-----------------------------------|--|---|----------------------|------------|
| 1. | Membuat laporan bahan baku masuk | <i>User</i> akan memilih data pada bulan yang ingin dibuat laporan, kemudian klik tombol lihat data | Tombol lihat data berjalan dengan benar dan sistem menampilkan data bahan baku masuk pada bulan yang dipilih | Sesuai | Valid |
| 2. | Membuat laporan bahan baku masuk | <i>User</i> klik tombol cetak, dan sistem akan menampilkan tampilan laporan. | Tombol cetak berjalan dengan benar dan sistem menampilkan tampilan laporan | Sesuai | Valid |
| 3. | Membuat laporan bahan baku keluar | <i>User</i> akan memilih data pada bulan yang ingin dibuat laporan, kemudian klik tombol lihat data. | Tombol lihat data berjalan dengan benar dan sistem menampilkan data bahan baku keluar pada bulan yang dipilih | Sesuai | Valid |
| 4. | Membuat laporan bahan baku keluar | <i>User</i> klik tombol cetak, dan sistem akan menampilkan tampilan laporan. | Tombol cetak berjalan dengan benar dan sistem menampilkan tampilan laporan | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

g. *Functional Test Use Case* Membuat Permitaan Pembelian

Berikut merupakan *functional test case* membuat permintaan pembelian yang dapat dilihat pada Tabel 7.

Function : Operasi membuat permintaan pembelian.

Data Assumptions : Fungsi operasi menginput dan menyimpan data permintaan baku sudah berjalan dengan baik.

Deskripsi : Melakukan penginputan data permintaan bahan baku.

Tabel 7 *Test Case* Membuat Permintaan Pembelian

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|------------------------------|---|---|---------------|------------|
| 1. | Membuat permintaan pembelian | User mengisi semua kolom pada form permintaan pembelian dan klik tombol buat permintaan | Tombol buat permintaan berjalan dengan benar, dan sistem menampilkan tabel permintaan | Sesuai | Valid |
| 2. | Membuat permintaan pembelian | User klik tombol selesai input dan sistem akan menampilkan <i>pop up form input</i> . | Tombol selesai input berjalan dengan benar. | Sesuai | Valid |
| 3. | Membuat permintaan pembelian | Setelah mengisi tanggal pada <i>pop up form</i> , user klik tombol simpan dan sistem akan menyimpan data permintaan | Tombol simpan berjalan dengan benar dan data permintaan tersimpan | Sesuai | Valid |
| 4. | Membuat permintaan pembelian | Menghapus data permintaan pembelian pada form permintaan | Tombol hapus berjalan dengan benar dan data terhapus. | Sesuai | Valid |

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|------------------------------|---|---|---------------|------------|
| | | pembelian dengan tombol hapus. | | | |
| 5. | Membuat permintaan pembelian | User mengosongkan kolom jumlah pada form permintaan pembelian dan klik tombol buat permintaan | Sistem akan menampilkan permintaan untuk mengisi kolom yang dikosongkan | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

h. *Functional Test Use Case Menginput Bahan Baku Masuk*

Berikut merupakan *functional test case* menginput bahan baku masuk yang dapat dilihat pada Tabel 8.

Function : Operasi menginput bahan baku masuk.

Data Assumptions : Fungsi operasi menginput dan menyimpan data bahan baku masuk sudah berjalan dengan baik.

Deskripsi : Melakukan penginputan data bahan baku masuk.

Tabel 8 *Test Case* Menginput Bahan Baku Masuk

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|---------------------------------|---|---|---------------|------------|
| 1. | Menginput data bahan baku masuk | User mengisi semua kolom pada form bahan baku masuk dan klik tombol tambah data | Tombol tambah berjalan dengan benar, dan sistem menampilkan data bahan baku masuk | Sesuai | Valid |
| 2. | Menginput data bahan baku masuk | User klik tombol selesai input dan sistem akan | Tombol selesai <i>input</i> berjalan dengan benar. | Sesuai | Valid |

| No. | <i>Test Case Name</i> | <i>Description</i> | <i>Expected Result</i> | <i>Actual Record</i> | Kesimpulan |
|-----|---------------------------------|--|---|----------------------|-------------------|
| | | menampilkan <i>pop up form input</i> . | | | |
| 3. | Menginput data bahan baku masuk | Setelah mengisi semua kolom pada <i>pop up form</i> , <i>user</i> klik tombol simpan dan sistem akan menyimpan data bahan baku masuk | Tombol simpan berjalan dengan benar dan data permintaan tersimpan | Sesuai | Valid |
| 4. | Menginput data bahan baku masuk | Menghapus data bahan baku masuk pada form bahan baku masuk dengan tombol hapus. | Tombol hapus berjalan dengan benar dan data terhapus. | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

i. *Functional Test Use Case Menginput Transaksi Bahan Baku Keluar*

Berikut merupakan *functional test case* menginput transaksi bahan baku keluar yang dapat dilihat pada Tabel 9.

Function : Operasi menginput data transaksi bahan baku keluar

Data Assumptions : Fungsi operasi menginput dan menyimpan data transaksi bahan baku keluar sudah berjalan dengan baik.

Deskripsi : Melakukan penginputan data transaksi bahan baku keluar.

Tabel 9 *Test Case* Menginput Transaksi Bahan Baku Keluar

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|----------------------------------|---|--|---------------|------------|
| 1. | Menginput data bahan baku keluar | User mengisi semua kolom pada form bahan baku keluar dan klik tombol tambah data | Tombol tambah berjalan dengan benar, dan sistem menampilkan data bahan baku keluar | Sesuai | Valid |
| 2. | Menginput data bahan baku keluar | User klik tombol selesai input dan sistem akan menampilkan <i>pop up form input</i> . | Tombol selesai <i>input</i> berjalan dengan benar. | Sesuai | Valid |
| 3. | Menginput data bahan baku keluar | Setelah mengisi semua kolom pada <i>pop up</i> form, user klik tombol simpan dan sistem akan menyimpan data bahan baku keluar | Tombol simpan berjalan dengan benar dan data permintaan tersimpan | Sesuai | Valid |
| 4. | Menginput data bahan baku keluar | Menghapus data bahan baku keluar pada form bahan baku keluar dengan tombol hapus. | Tombol hapus berjalan dengan benar dan data terhapus. | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

j. *Functional Test Use Case Melihat Data Permintaan Pembelian Bahan Baku*

Berikut merupakan *functional test case* melihat data permintaan pembelian bahan baku yang dapat dilihat pada Tabel 10.

Function : Operasi melihat data permintaan pembelian bahan baku

Data Assumptions : Fungsi operasi melihat data permintaan pembelian bahan baku sudah berjalan dengan baik.

Deskripsi : Melakukan kegiatan untuk melihat data permintaan pembelian bahan baku.

Tabel 10 *Test Case* Melihat Data Permintaan Pembelian

| No. | Test Case Name | Description | Expected Result | Actual Record | Kesimpulan |
|-----|-------------------------------------|---|--|---------------|------------|
| 1. | Melihat data permintaan pembelian | User klik menu data permintaan pembelian dan sistem akan menampilkan data permintaan pembelian. | Sistem menampilkan data permintaan pembelian | Sesuai | Valid |
| 2. | Melihat detail permintaan pembelian | User klik tombol detail pada data permintaan pembelian, dan sistem menampilkan detail data permintaan pembelian | Tombol detail berjalan dengan benar dan sistem menampilkan detail permintaan pembelian | Sesuai | Valid |

(Sumber: Hasil Analisis, 2019)

LAMPIRAN B

KODE PROGRAM

1. Halaman Utama

```
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>PT Teknikatama Karya Mandiri</title>

    <!-- Bootstrap Core CSS -->
    <link href=<?php echo base_url() ;?>assets/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom CSS -->
    <link href=<?php echo base_url() ;?>assets/css/sb-admin.css" rel="stylesheet">

    <!-- Custom Fonts and icon -->
    <link href=<?php echo base_url() ;?>assets/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">

    <!-- jQuery for 1. tombol data tambah/update/delete dan pengaruh ke table-->
    <script src=<?php echo base_url() ;?>assets/calendar/jquery.min.js"></script>
```

```

<!-- DataTables pengaruh ke table-->
<link rel="stylesheet" href="php echo base_url()?assets/datatables/dataTables.bootstrap.css">

<!-- Bootstrap Core JavaScript pengaruh ke menu dropdown sidebar -->
<script src="php echo base_url(); ?&gt;assets/js/bootstrap.min.js"&gt;&lt;/script&gt;

&lt;!--Fullcalendar Start--&gt;
&lt;link href='&lt;?php echo base_url(); ?&gt;assets/calendar/fullcalendar.min.css' rel='stylesheet' /&gt;
&lt;link href='&lt;?php echo base_url(); ?&gt;assets/calendar/fullcalendar.print.min.css' rel='stylesheet' media='print' /&gt;

&lt;script src='&lt;?php echo base_url(); ?&gt;assets/calendar/moment.min.js'&gt;&lt;/script&gt;
&lt;script src='&lt;?php echo base_url(); ?&gt;assets/calendar/fullcalendar.min.js'&gt;&lt;/script&gt;

&lt;!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries --&gt;
&lt;!-- WARNING: Respond.js doesn't work if you view the page via file:// --&gt;
&lt;!--[if lt IE 9]&gt;
    &lt;script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"&gt;&lt;/script&gt;
    &lt;script
src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"&gt;&lt;/script&gt;
&lt;![endif]--&gt;

&lt;/head&gt;</pre

```



```

<a class="navbar-brand" style="color: #fffcfc;"> PT Teknikatama Karya
Mandiri </a>
</div>
<!-- Top Menu Items -->
<ul class="nav navbar-right top-nav">

<li class="dropdown" >
    <a href="#" class="dropdown-toggle" data-toggle="dropdown"
style="color: #fffcfc;"><i class="fa fa-user" style="color: #fffcfc;"></i> Hallo <?php
echo $this->session->userdata('username'); ?> <b class=""></b></a>

</li>
</ul>
</a>
<!-- Sidebar Menu Items - These collapse to the responsive navigation menu on
small screens -->
<div class="collapse navbar-collapse navbar-ex1-collapse">
    <ul class="nav navbar-nav side-nav" style="background-color: #8B0000">
        <li>
            <a href="<?php echo site_url('Dashboard');?>" style="color:
#fffcfc;"><i class="fa fa-home"></i> Halaman Utama </a>
        </li>
    </ul>
<?php if ($akses['jabatan']=='Operator')
{
?>
```

```
<li>
    <a href="javascript:;" data-toggle="collapse" data-target="#demo"
style="color: #fffffc;"><i class="fa fa-fw fa-table"></i> Data Master <i class="fa fa-
fw fa-caret-down"></i></a>

    <ul id="demo" class="collapse">
        <li>
            <a href="<?php echo site_url('Pengguna');?>"> <i class="far fa-
circle"></i> Data Pengguna</a>
        </li>

        <li>
            <a href="<?php echo site_url('Supplier');?>"> <i class="far fa-
circle"></i> Data Supplier</a>
        </li>

<!--          <li>
            <a href="<?php echo site_url('Supplier');?>"> <i class="fa fa-fw fa-
edit"></i> Data Supplier</a>
        </li>

        <li>
            <a href="<?php echo site_url('Customer');?>"> <i class="fa fa-fw fa-
edit"></i> Data Customer</a>
        </li> -->

        <li>
```

```
<a href="<?php echo site_url('Bb');?>"> <i class="far fa-circle"></i>
Data Persediaan Bahan Baku</a>
</li>
</ul>
</li>

<li>
<a href="<?php echo site_url('Permintaan/Tampil_permintaan');?>" style="color: #fffffc;"><i class="fa fa-fw fa-table"></i> Data Permintaan BB</a>
</li>

<li>
<a href="javascript:;" data-toggle="collapse" data-target="#demos" style="color: #fffffc;"><i class="fa fa-fw fa-table"></i> Data Transaksi<i class="fa fa-fw fa-caret-down"></i></a>
<ul id="demos" class="collapse">
<li>
<a href="<?php echo site_url('Bbmasuk/Form_input');?>"> <i class="far fa-circle"></i> Form Bahan Baku Masuk</a>
</li>

<li>
<a href="<?php echo site_url('Bbmasuk/lihat_data');?>"> <i class="far fa-circle"></i> Data Bahan Baku Masuk
</a>
</li>
<li>
<a href="<?php echo site_url('Bbkeluar/Form_input');?>"> <i class="far fa-circle"></i> Form Bahan Baku Keluar</a>
```

```
</li>
<li>
<a href=<?php echo site_url('Bbkeluar/index');?>> <i class="far fa-circle"></i> Data Bahan Baku Keluar
</a>
</li>
</li>
</ul>
</li>

<li>
<a href="javascript:;" data-toggle="collapse" data-target="#demos1" style="color: #fffffc;"><i class="fa fa-fw fa-table"></i> Permintaan Pembelian <i class="fa fa-fw fa-caret-down"></i></a>
<ul id="demos1" class="collapse">
<li>
<a href=<?php echo site_url('PP/Form_input');?>> <i class="far fa-circle"></i> Form Permintaan Pembelian</a>
</li>
<li>
<a href=<?php echo site_url('PP/lihat_data');?>> <i class="far fa-circle"></i> Data Permintaan Pembelian</a>
</li>
</li>
</ul>
</li>
```

```

<li>
    <a href="javascript:;" data-toggle="collapse" data-target="#laporan"
style="color: #fffffc;"><i class="fa fa-fw fa-table"></i> Laporan <i class="fa fa-fw
fa-caret-down"></i></a>

    <ul id="laporan" class="collapse">
        <li>
            <a href=<?php echo site_url('Laporan/bbmasuk');?>"> <i class="far fa-
circle"></i> Laporan BB Masuk</a>
        </li>
        <li>
            <a href=<?php echo site_url('Laporan/bbkeluar');?>"> <i class="far fa-
circle"></i> Laporan BB Keluar</a>
        </li>
        </ul>
    </li>
    <li>

<?php
}
?>
<?php if ($akses['jabatan']=='Produksi')
{
?>

<li>
    <a href="javascript:;" data-toggle="collapse" data-target="#demos2"
style="color: #fffffc;"><i class="fa fa-fw fa-arrows-v"></i> Data Permintaan Bahan
Baku <i class="fa fa-fw fa-table"></i></a>
    <ul id="demos2" class="collapse">

```

```

<li>
<li>
<a href="php echo site_url('Permintaan');?&gt;" <i class="far fa-circle"></i> Form Permintaan</a>
</li>
<li>
<a href="php echo site_url('Permintaan/lihat_data');?&gt;" <i class="far fa-circle"></i> Data Permintaan</a>
</li>
</ul>
</li>
<li>

<?php
}
?>
<?php if ($akses['jabatan']=='Purchasing')
{
?>

<li>
<a href="php echo site_url('PP/Tampil_pp');?&gt;" style="color:#fffcfc;"&gt;&lt;i class="fa fa-fw fa-edit"&gt;&lt;/i&gt; Data Permintaan Pembelian&lt;/a&gt;
&lt;/li&gt;

&lt;?php
}
?&gt;
</pre

```

```
<li>
    <a href="<?php echo site_url('auth/logout');?>" style="color: #fffccfc;"><i
        class="glyphicon glyphicon-log-out"></i> Log Out

    </a>
</li>
</ul>
</div>
<!-- /.navbar-collapse -->
</nav>

<div id="page-wrapper">

<div class="container-fluid">

    <div class="panel-heading"></div>

    <?php
    echo $this->template->content;

?>

</body>

    <!-- /.table-responsive -->
    </div>
    <!-- /.panel-body -->
</div>
```

```
<div>
  <body>
    <center>

      </center>
    </body>
  </div>
<!-- /.panel -->
<div align="center">
  
</div>
<div>
  <div style="text-align: center; " id="non"><font color="black">PT Teknikatama Karya Mandiri </font><span class="copyright">&copy; <font color="black">2019 <a href="http://stmi.ac.id">Politeknik STMI</a></font></a>.</span></div>
</div>
<!-- jQuery -->

<!-- DataTables JavaScript -->
<!-- DataTables -->
<script src="<?php echo base_url();?>assets/datatables/jquery.dataTables.min.js"></script>
<script src="<?php echo base_url();?>assets/datatables/dataTables.bootstrap.min.js"></script>

<script>
```

```

$(document).ready(function(){
    $('#example1').DataTable({ columnDefs: [{ "defaultContent": "-", "targets": "_all" }]}));
}

</script>

```

2. A. CONTROLLER LAPORAN

```

<?php
class Laporan extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->model(array('M_laporan','M_bb','M_customer'));
    }

    function bbkeluar(){
        $bulan = $this->input->post('bulan');
        $tahun = $this->input->post('tahun');
        $data['bulan'] = $bulan;
        $data['tahun'] = $tahun;
        $data['record'] = $this->M_laporan->laporan_bbkeluar($bulan,$tahun);
        $this->template->content->view('laporan/laporan_bbkeluar',$data);
        $this->template->publish();
    }

    function bbmasuk(){
        $bulan = $this->input->post('bulan');
        $tahun = $this->input->post('tahun');
        $data['bulan'] = $bulan;
    }
}

```

```

    $data['tahun'] = $tahun;
    $data['record']           = $this->M_laporan-
>laporan_bbmasuk($bulan,$tahun);
    $this->template->content->view('laporan/laporan_bbmasuk',$data);
    $this->template->publish();
}

function cetakbbmasuk(){
$bulan = $this->uri->segment(3);
$tahun = $this->uri->segment(4);
$data_bulan = array('1' => 'JANUARI',
                    '2' => 'FEBRUARI',
                    '3' => 'MARET',
                    '4' => 'APRIL',
                    '5' => 'MEI',
                    '6' => 'JUNI',
                    '7' => 'JULI',
                    '8' => 'AGUSTUS',
                    '9' => 'SEPTEMBER',
                    '10' => 'OKTOBER' ,
                    '11' => 'NOVEMBER',
                    '12' => 'DESEMBER');

// die;
$data['bulan'] = $data_bulan[$bulan];
    $data['record']           = $this->M_laporan-
>laporan_bbmasuk($bulan,$tahun);
    $this->load->view('laporan/cetak_bbmasuk',$data);
    // $this->template->content->view('laporan/cetak_bbmasuk',$data);
    // $this->template->publish();
}

```

```

}

function cetakbbkeluar(){
    $data['bahankbaku'] = $this->M_bb->lihat_data()->result();
    $bulan = $this->uri->segment(3);
    $tahun = $this->uri->segment(4);
    $data_bulan = array('1' => 'JANUARI',
                        '2' => 'FEBRUARI',
                        '3' => 'MARET',
                        '4' => 'APRIL',
                        '5' => 'MEI',
                        '6' => 'JUNI',
                        '7' => 'JULI',
                        '8' => 'AGUSTUS',
                        '9' => 'SEPTEMBER',
                        '10' => 'OKTOBER' ,
                        '11' => 'NOVEMBER',
                        '12' => 'DESEMBER');

    // die;
    $data['bulan'] = $data_bulan[$bulan];
    $data['record'] = $this->M_laporan->laporan_bbkeluar($bulan,$tahun);
    $this->load->view('laporan/cetak_bbkeluar',$data);
    // $this->template->content->view('laporan/cetak_bbkeluar',$data);
    // $this->template->publish();
}

}

```

B. MODELS LAPORAN

```
<?php
```

```
class M_laporan extends CI_Model
{
    //fungsi ini untuk menampilkan laporan
    function laporan_bbkeluar($bulan,$tahun){
        $query = "SELECT * FROM tb_bbkeluar, tb_bbkeluar_detail, tb_bb
WHERE tb_bb.id_bb = tb_bbkeluar_detail.id_bb
        AND tb_bbkeluar_detail.id_bbkeluar = tb_bbkeluar.id_bbkeluar
        AND MONTH(tb_bbkeluar.tanggal) ='".$bulan."' AND YEAR(tb_bbkeluar.tanggal)
='".$tahun."';
        return $this->db->query($query);
    }

    function laporan_bbmasuk($bulan,$tahun){
        $query = "SELECT * FROM tb_bbmasuk_detail, tb_bbmasuk,
tb_bb      WHERE      tb_bb.id_bb      =      tb_bbmasuk_detail.id_bb      AND
tb_bbmasuk.id_bbmasuk      =      tb_bbmasuk_detail.id_bbmasuk      AND
MONTH(tb_bbmasuk.tanggal)  =  '".$bulan."'  AND  YEAR(tb_bbmasuk.tanggal)
='".$tahun."';
        return $this->db->query($query);
    }
}
```

C. VIEW LAPORAN (CETAK BB MASUK)

```
<script type="text/javascript">
```

```

        window.print();

</script>

<h3>PT Teknikatama Karya Mandiri</h3>
<h5>Jl. Raya Bahkilong</h5>
<h5>Desa Sukadami, Cikarang Selatan</h5>
<h3>LAPORAN BAHAN BAKU MASUK PADA BULAN <?php echo $bulan
?></h3>
<hr>
<table border="1px" cellspacing="2" cellpadding="1" width="100%" >
<thead>
<tr style="background-color: #daa520">
<th> <p> No</th> </p>
<th> <p> ID BB Masuk</th> </p>
<th> <p> Nama BB</th> </p>
<th> <p> No Surat Jalan</th> </p>
<th> <p> No PO</th> </p>
<th> <p> Tanggal</th> </p>
<th> <p> Jumlah</th> </p>
<th> <p> Satuan</th> </p>
<th> <p> Keterangan</th> </p>
</tr>
</thead>
<?php
if ($record->result()) {
    $no = 1;
    foreach ($record->result() as $d)
    {
        echo "<tr align ='center'>
<td>$no</td>
<td>$d[No]</td>
<td>$d[Id_BB_Masuk]</td>
<td>$d[Nama_BB]</td>
<td>$d[No_Surat_Jalan]</td>
<td>$d[No_PO]</td>
<td>$d[Tanggal]</td>
<td>$d[Jumlah]</td>
<td>$d[Satuan]</td>
<td>$d[Keterangan]</td>
</tr>
";
    }
}

```

```

<td>$d->id_bbmasuk</td>
<td>$d->nama_bb</td>
<td>$d->no_sj</td>
<td>$d->no_po</td>
<td>$d->tanggal</td>
<td>$d->jumlah</td>
<td>$d->satuan</td>
<td>$d->keterangan</td>
</tr>";

$no++;
}

}else {
    echo "<tr><td colspan = '7' align = 'center'>No records</td></tr>";
}

?>
</table>

<table>
<tr height="200">
    <td align="center"> Dicetak Oleh </td>
</tr>
<tr>
    <td align="center"> Departemen Warehouse </td>
</tr>
</table>

```

LAMPIRAN C

WAWANCARA DENGAN OPERATOR WAREHOUSE

Peneliti : “selamat pagi bu.. saya ingin melakukan wawancara terkait dengan penerimaan dan pengeluaran bahan baku dalam gudang. Apakah ibu bersedia?.”

Op W : “Baik. Akan saya jelaskan mengenai prosedur penerimaan dan pengeluaran bahan baku yang dalam gudang yang saya tangani. Jadi di dalam gudang ada 4 jenis barang, NG, RM, FG dan *repair*. FG itu kumpulan barang yang sudah jadi dengan kondisi barang yang bagus. NG itu kumpulan barang yang sudah jadi tapi kondisinya reject atau jelek. RM itu kumpulan bahan baku. Sedangkan *repair* adalah barang dengan kecacatan sedikit dan masih bisa diperbaiki.”

Peneliti : “Oke bu..”

Op W : “Untuk prosedur bahan baku yang ada di gudang sendiri, seperti ini. PT TKM mendapatkan bahan baku dari suppliernya sebelum diolah. Setelah bahan baku dari supplier sampai, dikasih ke gudang RM. Disimpan dulu di gudang buat nunggu waktu produksi Terus nanti operator produksinya mengajukan PKB (permintaan kebutuhan barang) ke operator *warehouse*. Jika sudah di ACC, lalu barang di produksi. Nanti kalau waktu pengirimannya tiba, saya buatkan surat jalan untuk delivery ordernya.

Peneliti : “Untuk bahan baku sekali keluar dari gudang bisa berapa pcs ya Bu?”

Op W : “300pcs.

Peneliti : “Untuk daftar stoknya memang masih manual ya Bu?”

Op W : “Oh ya. Karna saya biasa nulis di form. Laporan bulanan juga masih dalam bentuk excel.”

Peneliti : “Oke bu.. Terimakasih banyak untuk informasi yang diberikan. Selamat bekerja kembali”