

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan koordinasi

Saat praktik kerja magang dilakukan, kedudukan sebagai *members* yang ditempatkan di bagian ABAP. Pembimbing dan pelatih bahasa pemrograman ABAP saat melakukan kerja magang adalah Bapak Albertus Reinandang selaku *technical director*, semua tugas dan *progress* berkoordinasi dengan *technical director*.

3.2 Tugas yang dilakukan

Perancangan *prototype sales online* menggunakan sebuah metode perancangan piranti lunak yaitu *Rapid Application Development (RAD)*. Menurut (Dennis, Wixom, & Roth, 2012) metode RAD terdapat 4 fase yaitu :

1. Tahap *planning* :

Pada tahap ini diadakan sebuah pertemuan antara *board of director*, *technical director*, dan *members* untuk mendiskusikan *scope* dari permasalahan yang ada seperti yang sudah dijelaskan di bagian latar belakang, sehingga mencapai konsensus untuk fitur yang akan disediakan serta batasannya untuk *prototype sales online*.

2. Tahap *user design* :

Pada tahap ini terjadi pertemuan yang intensif antara *member* dengan *technical director* untuk mendiskusikan UI yang akan dibuat seperti apa, membuat sebuah model rancangan dari *prototype sales online* dengan membuat sebuah *data flow diagram*, *entity relationship diagram*, dan pembuatan tabel beserta strukturnya.

3. Tahap *construct* :

Pembuatan *code program* atau tahap *coding* dari *user design* yang telah dibuat.

4. Tahap *cutover* :

Pada tahap ini diadakan *full scale testing* oleh *technical director* dan *finishing*.

3.3 Uraian

Proses pembuatan *prototype sales online* membutuhkan waktu sebanyak 8 minggu. Berikut ini diuraikan secara terperinci pekerjaan yang dilakukan tiap minggu dalam membuat sebuah *prototype sales online*.

Tabel 3.1 Deskripsi tugas per minggu

Minggu ke	Deskripsi
1	<i>Planning</i> untuk pembuatan <i>prototype sales online</i> bersama <i>board of director, technical director, dan members</i>
2	Perancangan <i>user design</i> , mencakup pembuatan <i>data flow diagram, entity relationship diagram</i> , menentukan dan menganalisis tabel SAP yang diperlukan untuk pembuatan <i>prototype sales online</i> , dan merancang tabel beserta strukturnya.
3	Pembuatan tabel baru di SAP, pembuatan <i>structure</i> dan <i>data element</i> , pembuatan <i>function module ABAP</i> untuk <i>login</i> , pembuatan <i>function module ABAP</i> untuk <i>search cust</i> , pembuatan <i>function module ABAP</i> untuk <i>search material</i> .
4	Pembuatan <i>function module</i> untuk <i>simulate sales order</i> , pembuatan tampilan <i>web</i> dan <i>PHP code</i> untuk <i>create sales order, simulate sales order, dan feedback</i> status pembuatan <i>sales document</i>
5	Pembuatan tampilan <i>web</i> dan <i>PHP code</i> untuk <i>create sales order, simulate sales order, feedback</i> status pembuatan <i>sales document</i> , dan pembuatan <i>function module ABAP display sales document</i>
6	Pembuatan tampilan <i>web</i> dan <i>PHP code</i> untuk <i>display sales document</i> , pembuatan <i>function module ABAP</i> untuk <i>generate sales report</i>
7	Pembuatan tampilan <i>web</i> dan <i>PHP code</i> untuk <i>generate sales report</i> dan melakukan <i>testing</i> untuk <i>bug and error</i> secara keseluruhan proses-proses utama dari <i>web</i> .
8	<i>Testing</i> oleh <i>technical director</i> dan <i>finishing</i> .

Tabel 3.2 *Gantt chart* aktivitas per minggu

Kegiatan	Minggu ke-							
	1	2	3	4	5	6	7	8
Studi literatur	■	■	■	■	■	■	■	■
<i>planning</i>	■							
Analisa dan perancangan	■	■						
Pembuatan tabel, <i>structure</i> , <i>data element</i>		■	■					
Pembuatan <i>function module</i> , tampilan <i>web</i> , dan PHP			■	■	■	■	■	
<i>Testing Website</i>							■	■
<i>Finishing</i>								■

3.3.1 Proses Pelaksanaan

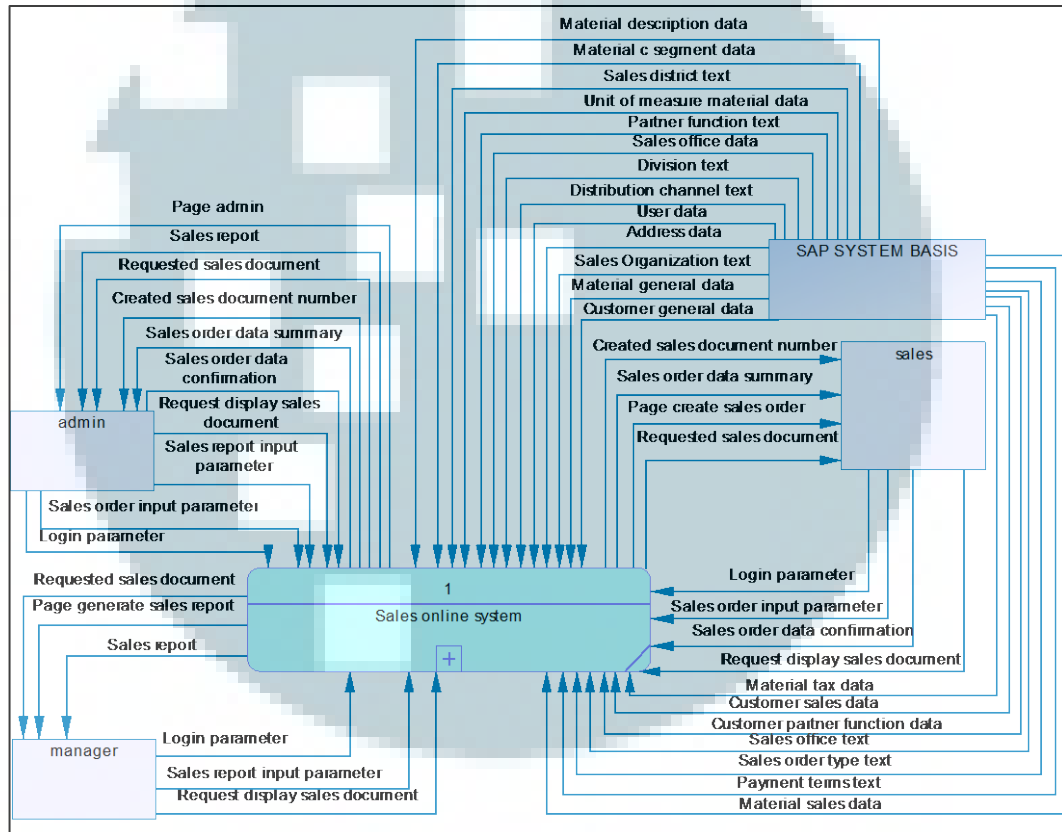
Pada saat proses pelaksanaan praktik kerja magang berlangsung, ada beberapa hal yang dilakukan seperti pembuatan desain *user interface* dan perancangan *prototype* yang meliputi struktur tabel dan *structure*, *entity relationship diagram*, *data flow diagram*, *flowchart*, dan pembuatan *prototype* yang meliputi pembuatan *function module* dan pembuatan PHP. Berikut dijelaskan secara rinci masing-masing proses yang dilakukan selama praktik kerja magang.

A. Perancangan Prototype

Dalam perancangan *prototype* dilakukan beberapa hal seperti pembuatan *data flow diagram*, *entity relationship diagram*, pembuatan tabel serta *structure* dan *flowchart*. Berikut dijelaskan secara rinci masing-masing perancangan yang dilakukan.

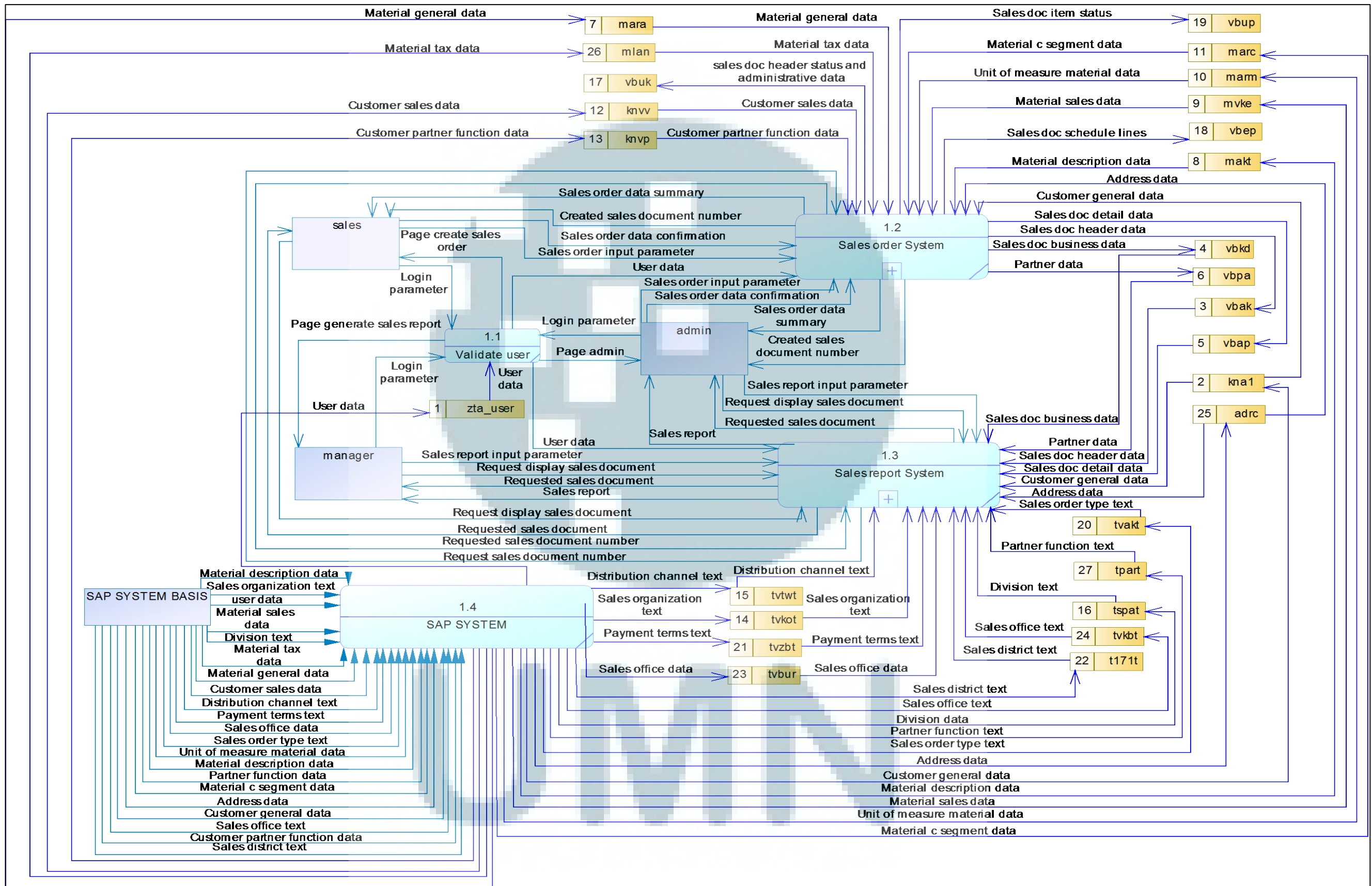
A.1 Data Flow Diagram

Context diagram DFD yang digambarkan dibawah ini, menggambarkan tentang keseluruhan proses *data flow* dari *prototype sales online*, dimana terdapat 3 entitas yakni *admin*, *sales*, *manager* sebagai user dari *prototype sales online* dan *sales online system* sebagai proses utama. Berikut dijabarkan pada gambar 3.1.



Gambar 3.1 Context diagram prototype sales online system

Dilanjutkan dengan DFD *level 0*, dimana memiliki 3 subproses yang terdapat pada proses utama terdiri dari *validate user*, *sales order system*, dan *sales report system*, berikut dijabarkan pada gambar 3.2.



Gambar 3.2 DFD level 0 prototype sales online system

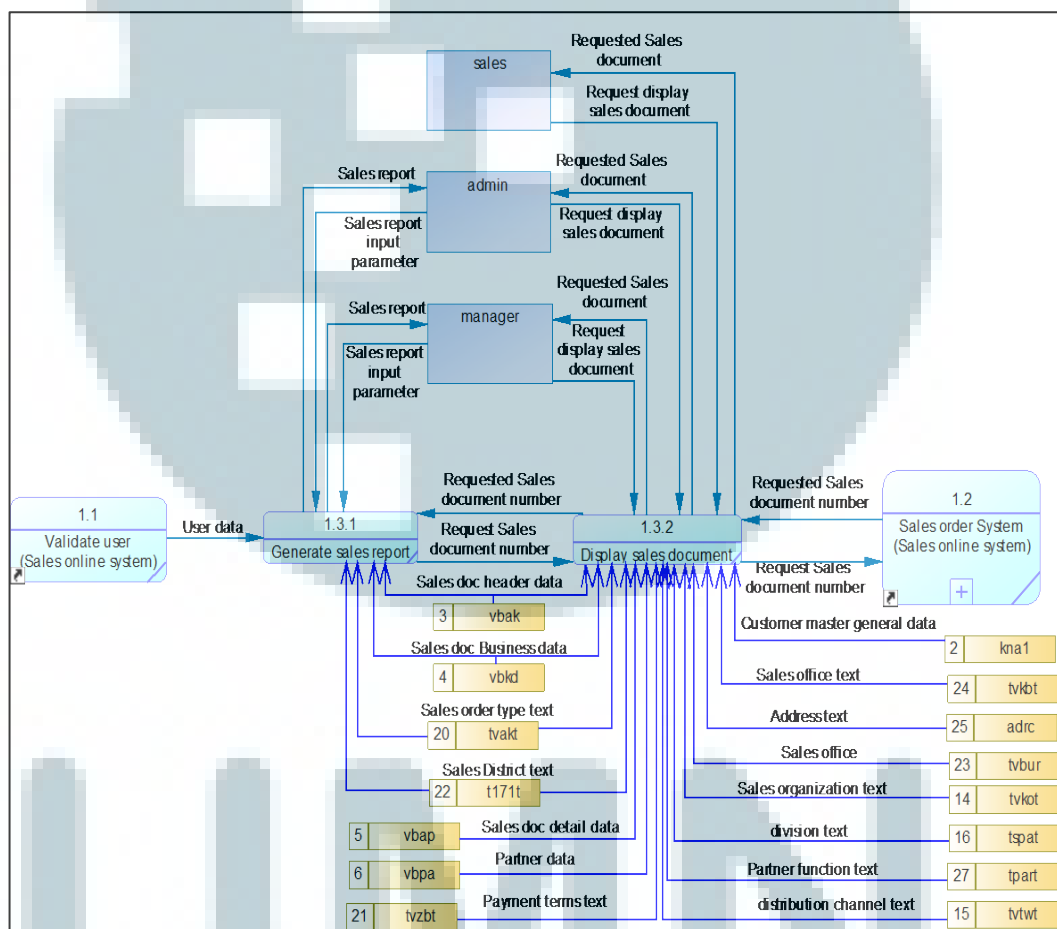
Secara rinci dijelaskan 2 subproses utama dalam DFD *level 1* yakni, *sales order system* dan *sales report system*. Pada subproses utama dari *sales order system* juga memiliki 2 subproses yakni, *simulate sales order* dan *create sales document*.

Subproses *simulate sales order* berguna untuk memberikan informasi *item list*, harga barang, dan total pembelanjaan yang dibeli oleh *customer*. Informasi tersebut diberikan kepada *admin/sales* sebelum dilakukan pembuatan *sales document* sehingga mengurangi kesalahan *input*.

Subproses *create sales document* berguna untuk membuat *sales document*, subproses ini menerima data dari subproses *simulate sales order* dan memberikan *sales document number* kepada *admin/sales* sebagai pemberitahuan bahwa *sales document* berhasil dibuat dan tersimpan dalam *database*. Apabila *admin/sales* ingin menampilkan *sales document* yang baru dibuat maka subproses *create sales document* memberikan data berupa *sales document number* kepada subproses *display sales document* yang ada didalam subproses utama *sales report system*. Berikut dijabarkan pada gambar 3.3.

UMMN

Untuk subproses utama dari *sales report system* memiliki 2 subproses yakni, proses *generate sales report* yang berfungsi untuk menampilkan *sales report* kepada *user admin* atau *manager* dan proses *display sales document* berfungsi untuk menampilkan *sales document* yang berdasarkan *sales document number* yang dipilih oleh *user (admin atau manager)* dari *sales report* yang ditampilkan sebelumnya oleh sistem kepada *admin/manager* dan menampilkan *sales document* berdasarkan *sales document number* yang baru saja dibuat oleh *user sales*.



Gambar 3.4 DFD level 1 sales report system

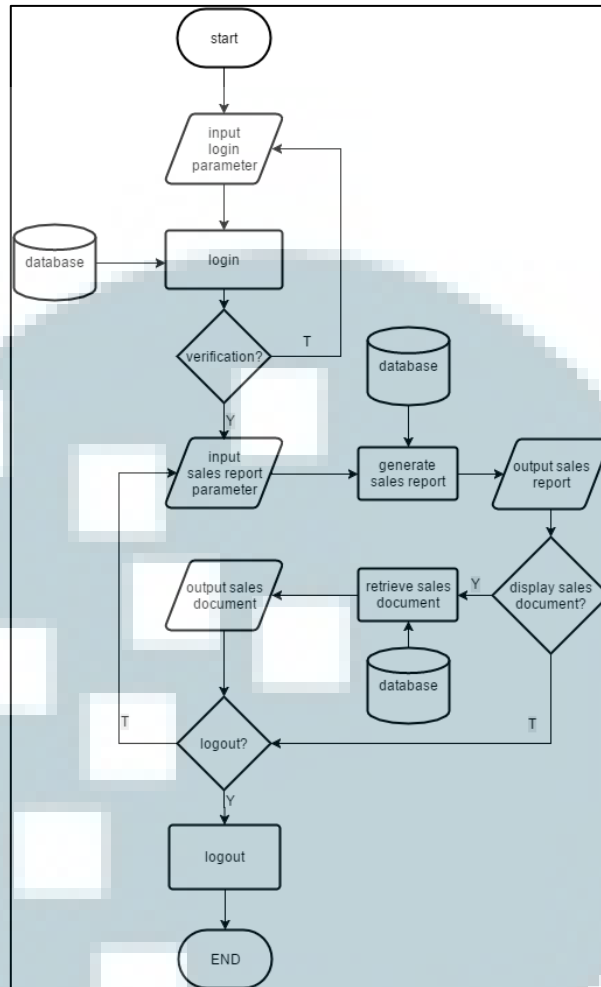
A.2 Flowchart

Prototype sales online memiliki beberapa proses dan *data flow* pada setiap proses. *Prototype sales online* memiliki 3 user yaitu *manager*, *admin*, dan *sales*. Berikut dijabarkan masing-masing *flowchart* dari user *admin*, *sales*, dan *manager*.

Pada gambar 3.5 dicantumkan *flowchart prototype sales online* dengan user sebagai *manager*. User *manager* hanya dapat melakukan proses *generate sales report* dan *display sales document*. Alur proses pada *flowchart user manager* dimulai dengan user melakukan *input login parameter*, kemudian sistem akan melakukan verifikasi apabila user terdaftar maka dapat melakukan proses *generate sales report*.

Untuk menampilkan *sales report*, user memberikan *input sales report parameter* ke dalam sistem, kemudian sistem akan memberikan *ouput sales report*. User dapat memilih untuk menampilkan *sales document* dari salah satu *sales document* yang ditampilkan pada *sales report*. User dapat melakukan *logout* dari sistem setelah melakukan proses *generate sales report* atau setelah menampilkan *sales document*.

UMMN

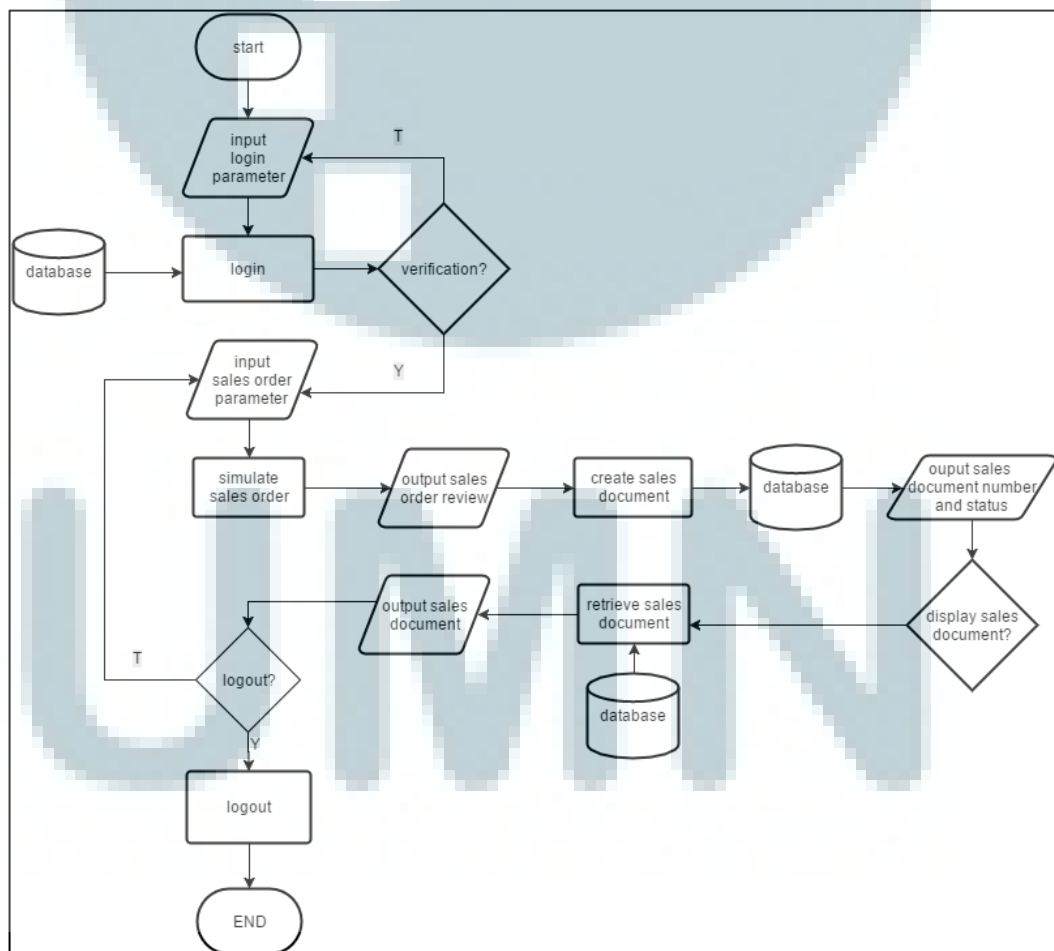


Gambar 3.5 Flowchart prototype sales online dengan user manager

Berikut ini dicantumkan *flowchart* dari *user sales* pada gambar 3.6. Pada *flowchart prototype sales online* dengan *user* sebagai *sales* terdapat beberapa proses yaitu melakukan *simulate sales order*, *create sales document*, dan dapat menampilkan *sales document* yang baru dibuat.

Proses alur pada *flowchart prototype sales online* dengan *user* sebagai *admin* adalah *user* akan melakukan proses *login* dengan *input login parameter* ke dalam sistem, kemudian sistem akan melakukan verifikasi *user*. Apabila *user* terdaftar dalam sistem, maka *user* dapat membuat *sales document* dengan memberikan *input sales order parameter* kepada sistem. Lalu sistem akan melakukan proses *simulate sales order*, proses ini akan memberikan *output sales order review* kepada *user*.

Fungsi dari *ouput sales order review* adalah memberikan fungsi yang serupa dengan *checkout* pada sistem *e-commerce*, agar mengurangi kesalahan *input* oleh *user* sebelum *user* melakukan pembuatan *sales document*. Setelah *user* yakin dengan *review* dari *sales order data* yang dimasukkan ke dalam sistem, sistem akan melakukan proses *create sales document*. Proses ini menyimpan data-data *sales order* ke dalam *database* kemudian akan menampilkan *output* status dari proses pembuatan *sales document* beserta *sales document number* yang dibuat kepada *user*. Proses alur terakhir *user* dapat memilih untuk menampilkan *sales document* yang baru dibuat atau tidak. Jika *user* memilih untuk tidak menampilkan *sales document*, maka *user* dapat *logout* atau kembali membuat *sales document* baru.



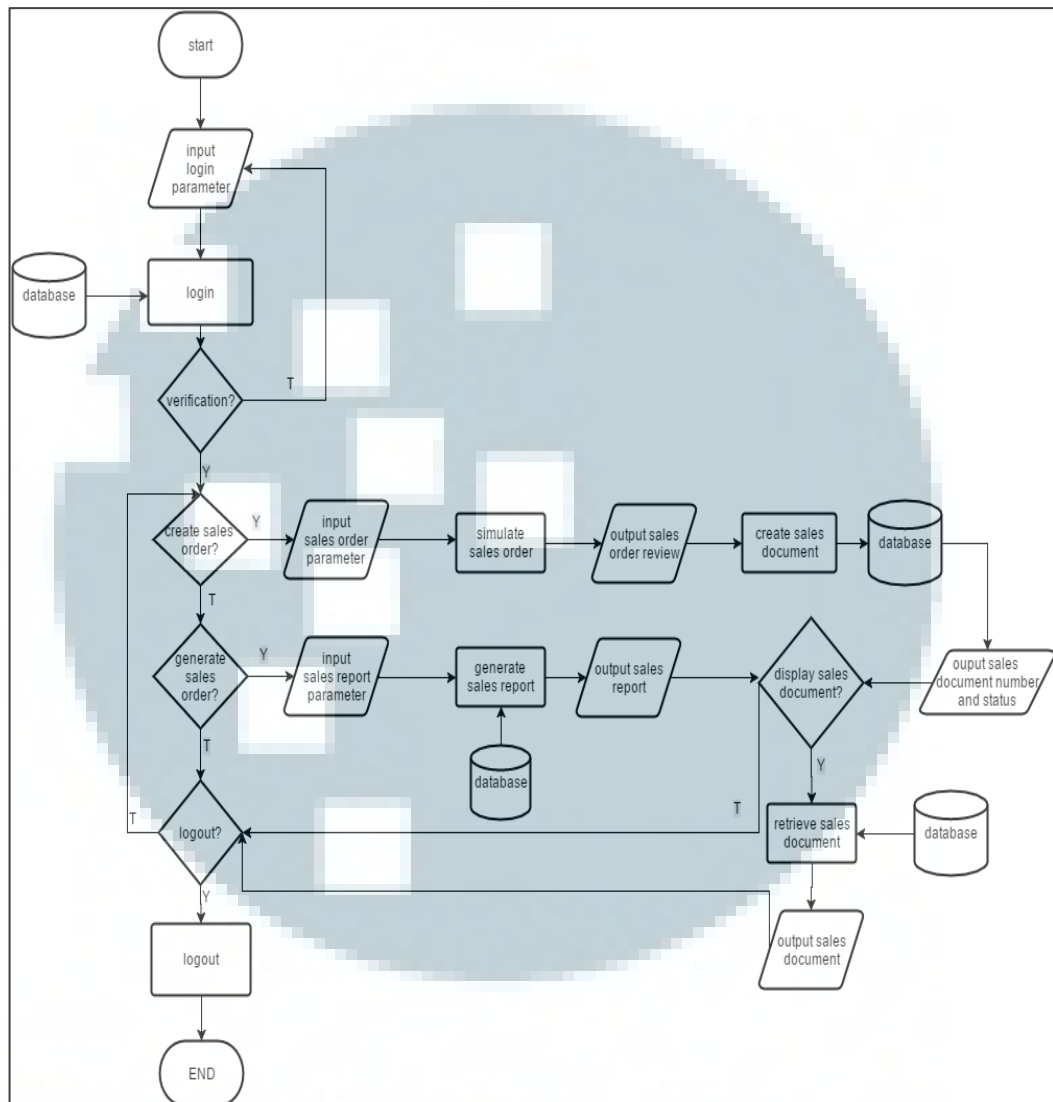
Gambar 3.6 Flowchart prototype sales online dengan user sales

Flowchart prototype sales online dengan user sebagai admin dijabarkan pada gambar 3.7. Proses alur pada *prototype sales online dengan user sebagai admin* dimulai dengan *user* melakukan *login* dengan memberi *input login parameter* ke dalam sistem. Sistem kemudian melakukan verifikasi *user*, apabila *user* terdaftar dalam sistem, maka *user* dapat memilih untuk melakukan *create sales order* atau *generate sales order*.

Proses alur ketika *user* memilih melakukan *create sales order* dimulai dengan *user* memberi *input sales order parameter*, kemudian sistem akan melakukan proses *simulate sales order* dan memberi *output sales order review* kepada *user*. Setelah *user* melakukan *review sales order*, sistem akan memasukkan data *sales order* ke dalam *database*. Kemudian sistem akan memberikan *output* kepada *user* berupa status dari proses *create sales document* dan *sales document number* yang baru dibuat. *User* dapat memilih untuk menampilkan *sales document* atau tidak, jika *user* memilih tidak, maka *user* dapat melakukan *logout* atau kembali untuk melakukan *create sales order* atau *generate sales report*.

Proses alur ketika *user* memilih untuk melakukan *generate sales report* memiliki proses alur yang dimulai dengan *user* memasukkan data *input sales report parameter* ke dalam sistem, kemudian sistem melakukan proses *generate sales report* dan memberikan data *output sales report* kepada *user*. Setelah itu, *user* dapat memilih salah satu *sales document* yang ada pada *sales report* untuk ditampilkan data-data *sales document* tersebut secara keseluruhan. Jika *user* memilih untuk menampilkan *sales document* maka sistem melakukan proses *retrieve sales document* dan memberikan *output sales document* pada *user*. Jika *user* tidak memilih untuk menampilkan maka *user* dapat memilih untuk melakukan *logout*, apabila *user* belum

ingin melakukan *logout* maka *user* dapat kembali memilih untuk melakukan *create sales order* atau *generate sales report*.



Gambar 3.7 Flowchart prototype sales online dengan user admin

A.3 Entity Relationship Diagram

Menurut (Dennis, Wixom, & Roth, 2012), *entity relationship diagram* merupakan sebuah desain yang menunjukkan informasi yang dibuat, disimpan, dan digunakan oleh sistem. Berikut merupakan gambar ERD dari *prototype sales online*.

A.3 Pembuatan Tabel dan Structure

Dalam membuat *prototype sales online*, tabel yang digunakan hampir semua dari tabel SAP yang sudah ada. Namun ada tabel yang dibuat sendiri dalam SAP, tabel tersebut adalah tabel ZTA_USER. Tabel ini digunakan untuk menyimpan data-data *user* dan hak akses dari *user* tersebut. Struktur tabel ZTA_USER dapat dilihat pada tabel 3.3.

Tabel 3.3 Struktur tabel ZTA_USER

Component	Component Type	Data Type	Length	Decimal	Short Description
MANDT	MANDT	CLNT	3	0	Client
USERID	YD_USERID	CHAR	10	0	Element Data USERID
NAME	YD_NAME	CHAR	200	0	Element Data Untuk Nama
PASSWORD	YD_PASSWORD	CHAR	20	0	Element Data Untuk Password
BUKRS	BUKRS	CHAR	4	0	Company Code
EMAIL	CHAR40	CHAR	40	0	Character field of length 40
HANDPHONE	CHAR20	CHAR	20	0	Char20
VALID_TO	BEGDA	CHAR	8	0	Start Date
VALID_FROM	ENDDA	CHAR	8	0	End Date
CREATED_BY	USAME	CHAR	12	0	Entered by
CREATED_DATE	DATUM	DATS	8	0	Date
CREATED_TIME	UZEIT	TIMS	6	0	Time
LOCK_ID	YDLOCK_ID	CHAR	1	0	Data Element for YLOCK_ID
CHANGED_BY	USAME	CHAR	12	0	Entered by
CHANGED_DATE	DATUM	DATS	8	0	Date
CHANGED_TIME	UZEIT	TIMS	6	0	Time
SALES_ORG	VKORG	CHAR	4	0	Sales Organization
SALES_OFF	VKBUR	CHAR	4	0	Sales Office
SALES_GRP	VKGRP	CHAR	3	0	Sales Group

Tabel 3.3 Struktur tabel ZTA_USER (lanjutan)

Component	Component Type	Data Type	Length	Decimal	Short Description
DIVISION	SPART	CHAR	2	0	Division
DST_CHNL	VTWEG	CHAR	2	0	Distribution Channel
USER_MENU	ZD_USER_MENU	CHAR	1	0	Data Element Flag USER MENU

Menurut (O'Neill, 2015), *structure* merupakan bagian dari *complex types* yang artinya memiliki banyak *data types*. Sebuah *structure* memiliki banyak *component* yang memiliki tipe datanya masing-masing. *Structure* digunakan di dalam ABAP Program, *structure* dapat disamakan seperti *one dimension array* atau *struct* pada C++ dan nama lain *structure* disebut *workarea*. Berikut *structure* yang dibuat :

1. ZST_CUST_SALES

Structure ini dibuat untuk *function module search customer*, ZST_CUST_SALES bersifat sebagai *array* yang digunakan pada program *function module* ABAP untuk *search customer*. Isi dari *structure* ini didapat dari tabel-tabel *customer master* seperti tabel kna1 dan tabel knvv. Berikut dijabarkan pada tabel 3.4.

Tabel 3.4 Struktur ZST_CUST_SALES

Component	Component Type	Data Type	Length	Decimal	Short Description
KUNNR	KUNNR	CHAR	10	0	Customer Number
VKORG	VKORG	CHAR	4	0	Sales Organization
VTWEG	VTWEG	CHAR	2	0	Distribution Channel
SPART	SPART	CHAR	2	0	Division
VKGRP	VKGRP	CHAR	3	0	Sales Group
VKBUR	VKBUR	CHAR	4	0	Sales Office
LAND1	LAND1_G P	CHAR	3	0	Country Key
NAME1	NAME1_G P	CHAR	35	0	Name 1
NAME2	NAME2_G P	CHAR	35	0	Name 2

Tabel 3.4 Struktur ZST_CUST_SALES (lanjutan)

Component	Component Type	Data Type	Length	Decimal	Short Description
ORT01	ORT01_GP	CHAR	35	0	City
PSTLZ	PSTLZ	CHAR	10	0	Postal Code
REGIO	REGIO	CHAR	3	0	Region (State, Province, Country)
TELF1	TELF1	CHAR	16	0	First Telephone Number
TELFX	TELFX	CHAR	31	0	Fax Number
ADRNR	ADRNR	CHAR	10	0	Address

2. ZST_MATERIAL

Structure ini dibuat untuk *function module search material*, ZST_MATERIAL bersifat sebagai *array* yang digunakan pada program *function module* ABAP untuk *search material*. Isi dari *structure* ini didapat dari tabel-tabel *material master* seperti tabel mara, tabel mvke dan tabel makt. Berikut dijabarkan struktur dari ZST_MATERIAL pada tabel 3.5.

Tabel 3.5 Struktur ZST_MATERIAL

Component	Component Type	Data Type	Length	Decimal	Short Description
MATNR	MATNR	CHAR	18	0	Material Number
MAKTX	MAKTX	CHAR	40	0	Material Description
VRKME	VRKME	UNIT	3	0	Sales Unit
VTWEG	VTWEG	CHAR	2	0	Distribution Channel
VKORG	VKORG	CHAR	4	0	Sales Organization

3. YST_DISPLAY_SO

Structure ini dibuat untuk *function module display sales document*, YST_DISPLAY_SO bersifat sebagai *array* yang digunakan pada program *function module ABAP* untuk *display sales document*. Berikut dijabarkan pada tabel 3.6.

Tabel 3.6 Struktur YST_DISPLAY_SO

Component	Component Type	Data Type	Length	Decimal	Short Description
VBELN	VBELN_VA	CHAR	10	0	Sales Document
VKORG	VKORG	CHAR	4	0	Sales Organization
VKORG_TXT	VTXTK	CHAR	20	0	Description
SPART	SPART	CHAR	2	0	Division
SPART_TXT	VTXTK	CHAR	20	0	Description
VKBUR	VKBUR	CHAR	4	0	Sales Office
VTWEG	VTWEG	CHAR	2	0	Distribution Channel
VTWEG_TXT	VTXTK	CHAR	20	0	Description
VKGRP	VKGRP	CHAR	3	0	Sales Group
BZIRK	BZIRK	CHAR	6	0	Sales District
BZIRK_TXT	BZTXT	CHAR	20	0	Name of the district
AUART	AUART	CHAR	4	0	Sales Document Type
BEZEI	BEZEI20	CHAR	20	0	Description
AUDAT	AUDAT	DATS	8	0	Document Date (date received/sent)
VDATU	EDATU_VBAK	DATS	8	0	Requested delivery date
INCO1	INCO1	CHAR	3	0	Incoterms (part 1)
INCO2	INCO2	CHAR	28	0	Incoterms (part 2)
PRSDT	PRSDT	DATS	8	0	Date for pricing and exchange rate
VTEXT	DZTERM_BEZ	CHAR	30	0	Description of terms of payment

Tabel 3.6 Struktur YST_DISPLAY_SO (lanjutan)

Component	Component Type	Data Type	Length	Decimal	Short Description
ZTERM	DZTERM	CHAR	4	0	Terms of payment key
VKBUR_BEZEI	BEZEI20	CHAR	20	0	Description
VKBU_STREET	AD_STREET	CHAR	60	0	Street
VKBUR_ADRNR	AD_ADDRNUM	CHAR	10	0	Address Number
NETWR	NETWR_AK	CURR	15	2	Net Value of the Sales Order in Document Currency
WAERK	WAERK	CUKY	5	0	SD document currency
GUEBG	GUEBG	CHAR	8	0	Valid-from date (outline agreements, product proposals)
GUENN	GUEEN	CHAR	8	0	Valid-to date (outline agreements, product proposals)
BSTNK	BSTNK	CHAR	20	0	Customer purchase order number
BSTDK	BSTDK	DATS	8	0	customer purchase order date
FKDAT	FKDAT	DATS	8	0	Billing date for billing index and printout

4. YST_PARTNER

Structure ini dibuat untuk *function module display sales document*, YST_PARTNER bersifat sebagai *array* yang digunakan pada program *function module ABAP* untuk *display sales document* yang berisi data-data *partner (customer)*. Berikut dijabarkan struktur dari YST_PARTNER pada tabel 3.7.

Tabel 3.7 Struktur YST_PARTNER

Component	Component Type	Data Type	Length	Decimal	Short Description
KUNNR	KUNNR	CHAR	10	0	Customer Number
NAME1	NAME1_GP	CHAR	35	0	Name 1
ORT01	ORT01_GP	CHAR	35	0	City
STRAS	STRAS_GP	CHAR	35	0	House Number and Street
PARVW	PARVW	CHAR	2	0	Partner Function
VBELN	VBELN	CHAR	10	0	Sales and Distributin Document Number
VTEXT	VTXTK	CHAR	20	0	Description

5. YST_RANGE_VBELN

Structure ini dibuat untuk *function module display sales document* dan *function module generate sales report*, structure ini berfungsi untuk *input range parameter* berdasarkan *sales document* number sehingga memudahkan pencarian berdasarkan *sales document number*. Struktur dari YST_RANGE_VBELN dijabarkan pada tabel 3.8.

Tabel 3.8 Struktur YST_RANGE_VBELN

Component	Component Type	Data Type	Length	Decimal	Short Description
SIGN	CHAR1	CHAR	1	0	Single-Character Flag
OPTION	CHAR2	CHAR	2	0	Version Number Component
LOW	VBELN_VA	CHAR	10	0	Sales Document
HIGH	VBELN_VA	CHAR	10	0	Sales Document

6. YST_RANGE_DOC_DATE

Structure ini dibuat untuk *function module generate sales report*, *structure* ini berfungsi untuk *input range parameter* berdasarkan *sales document date* sehingga memudahkan pencarian berdasarkan *range sales document date* misalnya dari tanggal 1 Juni 2016 hingga 5 Juni 2016.

Tabel 3.9 Struktur YST_RANGE_DOC_DATE

Component	Component Type	Data Type	Length	Decimal	Short Description
SIGN	CHAR1	CHAR	1	0	Single-Character Flag
OPTION	CHAR2	CHAR	2	0	Version Number Component
LOW	AUDAT	DATS	8	0	Document date
HIGH	AUDAT	DATS	8	0	Document date

7. YST_REPORT_SO

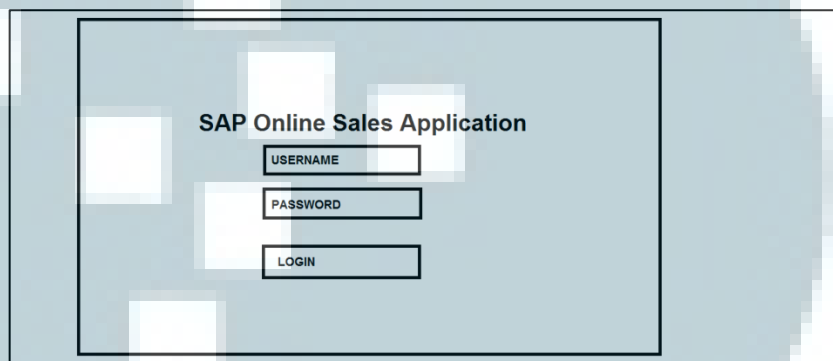
Structure ini dibuat untuk *function module generate sales report*, structure ini berfungsi untuk menampilkan *sales report* yang berisi data-data *sales document* yang telah dibuat pada rentang tanggal tertentu.

Tabel 3.10 Struktur YST_REPORT_SO

Component	Component Type	Data Type	Length	Decimal	Short Description
VBELN	VBELN_VA	CHAR	10	0	Sales Document
AUDAT	AUDAT	DATS	8	0	Document date
AUART	AUART	CHAR	4	0	Sales Document Type
BEZEI	BEZEI20	CHAR	20	0	Description
BZTXT	BZTXT	CHAR	20	0	Sales District
BZIRK	BZIRK	CHAR	6	0	Name of the district
VKORG	VKORG	CHAR	4	0	Sales Organization
VKBUR	VKBUR	CHAR	4	0	Sales office
VKGRP	VKGRP	CHAR	3	0	Sales group
SPART	SPART	CHAR	2	0	Division
VTWEG	VTWEG	CHAR	2	0	Distribution Channel
NETWR	NETWR_AK	CURR	15	2	Net Value of the Sales Order in Document Currency
WAERK	WAERK	CUKY	5	0	SD document currency

B. Desain Antarmuka

Prototype sales online memiliki halaman utama *login*, sistem ini memiliki 3 *user* utama yakni *admin*, *sales person*, dan *manager*. Setiap *user* memiliki hak akses atas setiap laman, karena itu ketika *login* akan diarahkan secara langsung ke laman yang merupakan hak dari *user* tersebut. Alasan tidak dibuatkan menu *sign up* dikarenakan *user* telah dibuat terlebih dahulu oleh BASIS di dalam SAP R/3, hal ini termasuk dalam *requirement* pembuatan *prototype*.



SAP Online Sales Application

USERNAME

PASSWORD

LOGIN

Gambar 3.9 Desain halaman *login*

Gambar 3.10 merupakan desain tampilan dari *user admin*, *admin* memiliki hak untuk melihat laman *create sales order* dan *sales report*.



Welcome, Admin

logout

create order

Sales Report

Gambar 3.10 Desain halaman *admin*

Tampilan untuk membuat sebuah *sales order* merupakan hak akses dari *sales person* dan *admin*. Berikut merupakan desain tampilan dari *create sales order*.

icon home welcome, admin **logout**

create sales order

sales organization **valid from** click to insert date
distribution channel **valid to** click to insert date
division **po number**
sales group **po date** click to insert date
Req delivery date click to insert date **sales to party** ...
ship to party ...

item	material	quantity	sales unit	description

simulate

Gambar 3.11 Desain halaman *create sales order*

Pada tampilan laman *create sales order* terdapat fitur untuk memunculkan sebuah kalender kecil dan laman *pop-up* untuk mencari *material* dan *customer*. Berikut contoh desain tampilan kalender kecil dan *pop-up* untuk mencari *material* dan *customer*.

show entries search

Material Number	Description	Sales Unit

Material Number Description Sales Unit

showing 1 to 10 of 500 entries Previous 1 2 3 4 ... 50 Next

Gambar 3.12 Desain *pop-up search material*

show entries search

Customer Number	Customer Name	City

Customer Number Customer Name City

showing 1 to 10 of 500 entries Previous 1 2 3 4 ... 50 Next

Gambar 3.13 Desain *pop-up search customer*

June 2016						
Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Gambar 3.14 Desain *pop-up kalender*

Sebelum membuat *sales document* akan ditampilkan *checkout* sehingga mengurangi kesalahan *input item* sebelum membuat sebuah *sales document*. Berikut gambar desain tampilan *simulate sales order* yang merupakan sebuah fitur seperti *checkout*.

Item	Material	Description	Quantity	Unit	Price	Currency	Amount

Create

Gambar 3.15 Desain *simulate sales order*

Apabila *item* yang dimasukkan sudah sesuai, maka setelah menekan tombol *create*, sistem akan memberikan sebuah tampilan yang berisi pesan bahwa *sales document* berhasil dibuat atau tidaknya beserta *sales document number*.

Sales Document 139588 Has Been Created

Display Document

Gambar 3.16 Desain *notifikasi create sales document*

Sales document yang baru dibuat dapat dilihat secara keseluruhan data-datanya, seperti *sales data*, *partner data*, *billing data* dan *item detail* yang merupakan barang yang dibeli oleh seorang *customer*.

Gambar 3.17 Desain *display sales document*

User manager hanya diberikan hak akses untuk melihat laporan penjualan yang berupa daftar-daftar *sales document* yang telah dibuat. Contoh tampilan laporan penjualan dapat dilihat pada gambar 3.18. *Sales document* yang ditampilkan pada *sales report* dapat dipilih dan dilihat seperti pada gambar 3.17.

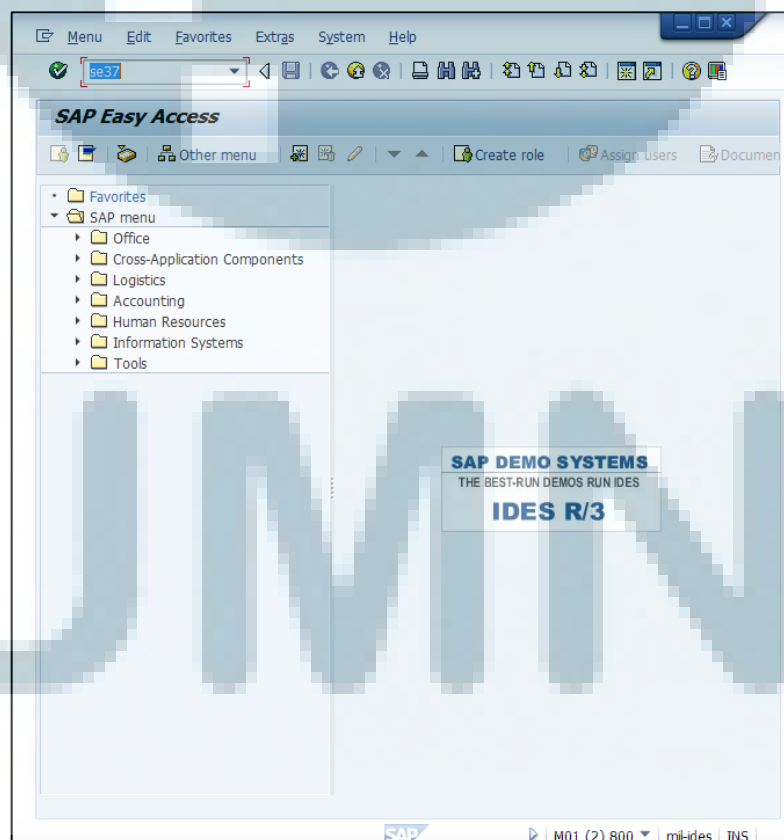
Gambar 3.18 Desain *generate sales report*

C. Pembuatan Prototype

Setelah melakukan perancangan *prototype sales online* dan membuat desain antarmuka dari *prototype sales online*, maka dilakukan pembuatan *prototype* dengan membuat *function module* pada SAP dan membuat kode HTML dan PHP. Berikut ini dijelaskan secara rinci masing-masing proses yang dilakukan.

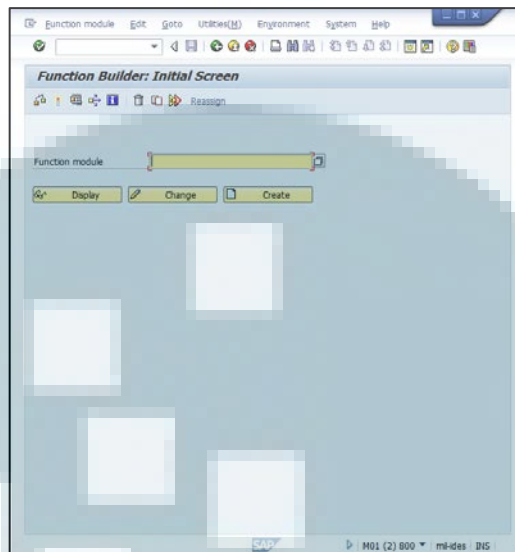
C.1 Pembuatan Function Module

Function module diperlukan sebagai penghubung antara SAP dan PHP. *Function module* berisi kode program ABAP dan dapat menggunakan fungsi-fungsi yang terdapat dalam SAP system. *Function module* dibuat di SAP R/3, *function builder* merupakan halaman untuk membuat sebuah *function module*, halaman ini dapat diakses dengan mencari kode halaman *function builder* (se37) seperti pada gambar 3.19.



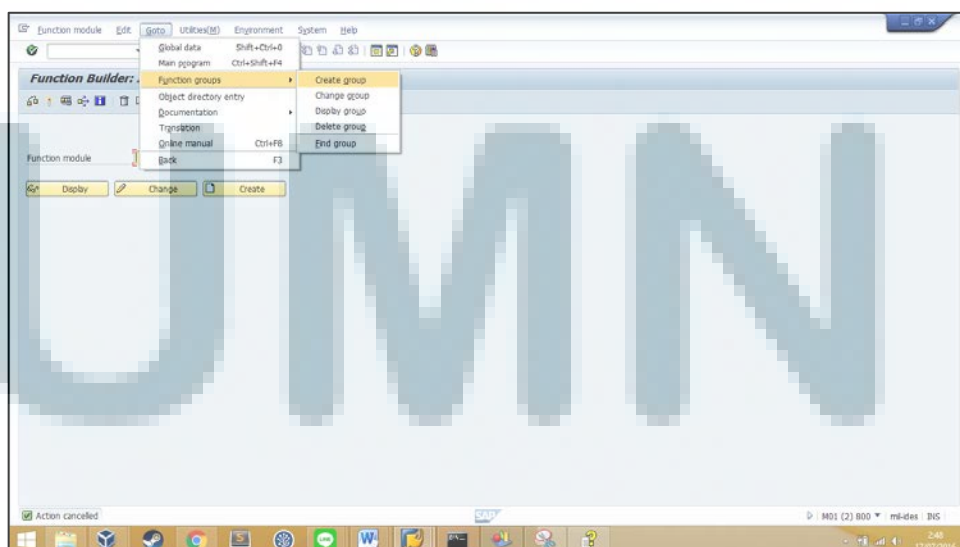
Gambar 3.19 Halaman awal menu SAP

Pada gambar 3.20 merupakan tampilan *function builder* untuk membuat sebuah *function module*.



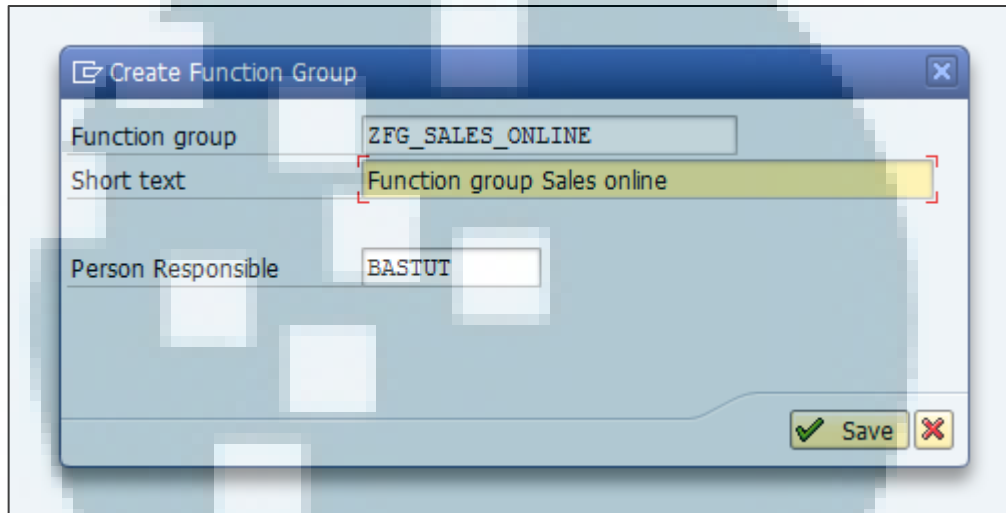
Gambar 3.20 Halaman *function builder*

Terdapat aturan *naming convention* dalam membuat sebuah kostumisasi dalam SAP yakni *Z convention* dan *Y convention*, seperti *ZFM_XXXX* untuk nama sebuah *function module* dan *ZFG_XXXX* untuk nama sebuah *function group*. Sebelum membuat *function module*, diperlukan membuat sebuah *function group*. Cara membuat *function group* dijelaskan pada gambar 3.21 dan gambar 3.22



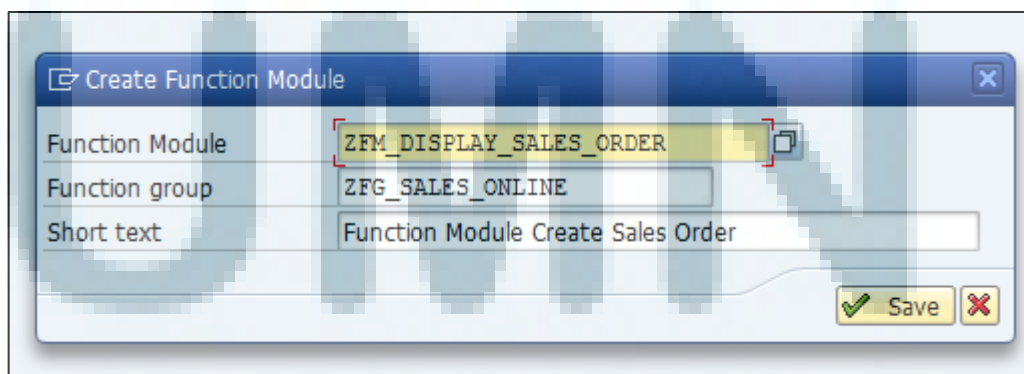
Gambar 3.21 Cara membuat *function group*

Nama *function group* yang akan digunakan untuk membuat *function group* baru akan disimpan dan diingat (*reserved*) oleh SAP system, sehingga untuk membuat *function group* baru dengan nama yang sudah ada di dalam SAP system tidak akan bisa dibuat. Fungsi dari *function group* adalah untuk mengelompokkan *function module*.



Gambar 3.22 Halaman awal membuat *function group*

Gambar 3.23 merupakan cara membuat sebuah *function module* pada SAP. Nama yang digunakan untuk *function module* yang ingin dibuat harus belum terdaftar di dalam SAP system, karena nama yang telah terdaftar di dalam SAP system tidak dapat digunakan.



Gambar 3.23 Cara membuat *function module*

Tab import merupakan *parameter input* yang digunakan pada *function module*, *tab export* merupakan hasil proses dari *function module*, *tab tables* merupakan internal tabel atau *workarea* yang bisa disebut sebagai *array* dalam bahasa pemrograman lain yang digunakan untuk proses dalam *function module*, dan *tab source code* merupakan kode program yang merupakan proses dari *function module* tersebut.

Pada gambar 3.26 merupakan tampilan *source code* dari *function module display sales order* dengan total jumlah *lines of code* sebesar 194 *lines*.

```

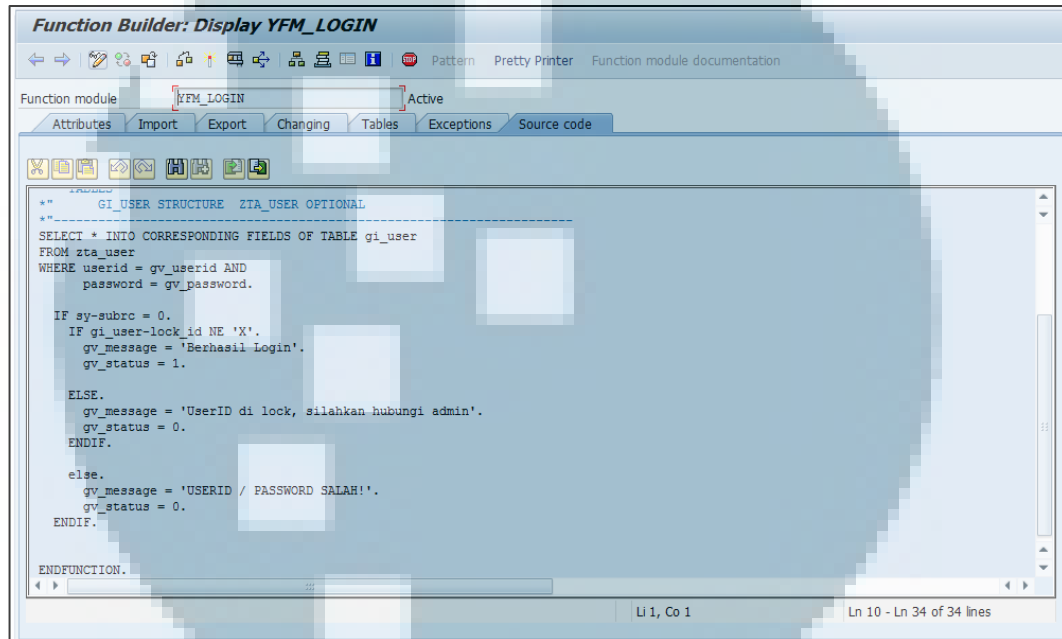
Function Builder: Change ZFM_DISPLAY_SALES_ORDER
Function module: ZFM_DISPLAY_SALES_ORDER Active
Source code
FUNCTION ZFM_DISPLAY_SALES_ORDER.
-----
**"Local interface:
** IMPORTING
** VALUE(GV_VBELN) TYPE VBELN OPTIONAL
** EXPORTING
** VALUE(GV_MESSAGE) LIKE BAPIRETURN STRUCTURE BAPIRETURN
** TABLES
** GR_VBELN STRUCTURE YST_RANGE_VBELN OPTIONAL
** GI_DISPLAY STRUCTURE YST_DISPLAY_SO
** GI_VBAP STRUCTURE VBAP
** GI_PARTNER STRUCTURE YST_PARTNER
-----
data : gi_ykorg like standard table of tvkot with header line,
gi_vtweg like standard table of tvwt with header line,
gi_ykbur like standard table of tvbur with header line,
gi_spart like standard table of tspot with header line,
gi_adrnr like standard table of adrc with header line,
gi_tvzbt like standard table of tvzbt with header line,
gi_knal like standard table of KNAL with header line,
gi_vbkd like standard table of vbkd with header line,
gi_tvkbt like standard table of tvkbt with header line,
gi_t171t like standard table of t171t with header line,
gi_tvakt like standard table of tvakt with header line,

```

Gambar 3.26 Kode *function module display sales order*

Nama *function module* yang digunakan untuk *prototype sales online* adalah ZFM_DISPLAY_SALES_ORDER, ZFM_GENERATE_SO_REPORT, ZFM_BASTUT_GET_MATERIAL, ZFM_SEARCH_CUST, ZFM_SIMULATE_ORDER, DAN YFM_LOGIN. Berikut ini dicantumkan potongan kode dari masing-masing *function module* yang digunakan dalam pembuatan *prototype sales online*.

Potongan kode pada gambar 3.27 merupakan fungsi *login* pada *prototype sales online*. Cara kerja *function module login* adalah mengisi *structure gi_user* yang diambil data-datanya dari tabel *ZTA_USER* berdasarkan *userid* dan *password*, apabila berhasil maka mengisi data *message* ke variabel *gv_message* dan data *status* ke variabel *gv_status*.



```
Function Builder: Display YFM_LOGIN
Function module YFM_LOGIN Active
Attributes Import Export Changing Tables Exceptions Source code
GI_USER STRUCTURE ZTA_USER OPTIONAL
SELECT * INTO CORRESPONDING FIELDS OF TABLE gi_user
FROM zta_user
WHERE userid = gv_userid AND
password = gv_password.

IF sy-subrc = 0.
IF gi_user-lock_id NE 'X'.
gv_message = 'Berhasil Login'.
gv_status = 1.

ELSE.
gv_message = 'UserID di lock, silahkan hubungi admin'.
gv_status = 0.
ENDIF.

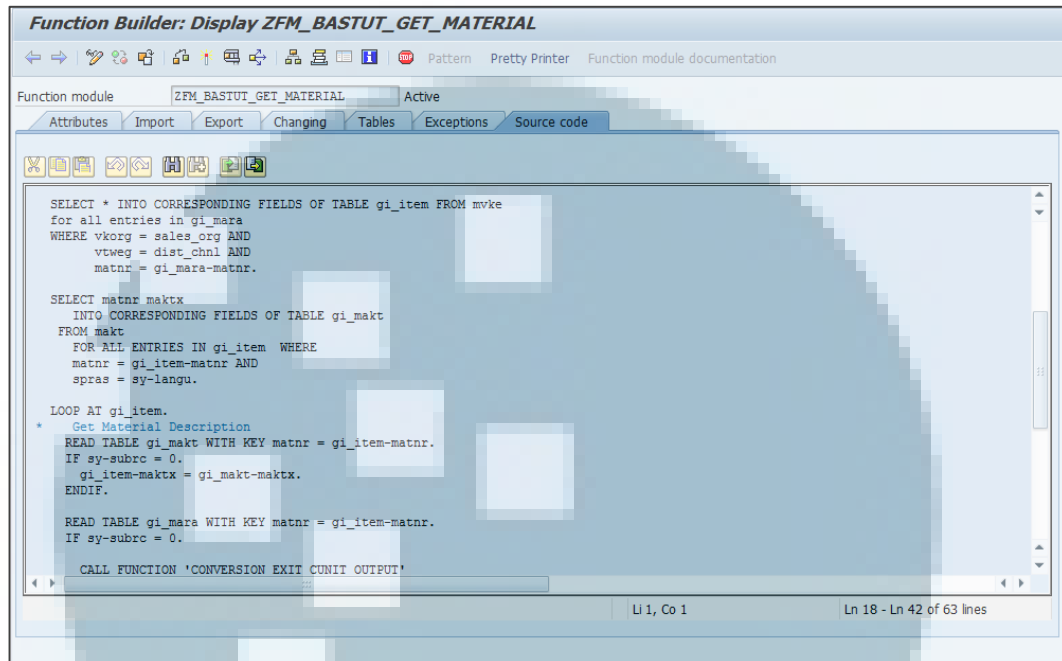
else.
gv_message = 'USERID / PASSWORD SALAH!'.
gv_status = 0.
ENDIF.

ENDFUNCTION.
Ln 10 - Ln 34 of 34 lines
```

Gambar 3.27 Kode *function module login*

Potongan kode pada gambar 3.28 merupakan fungsi untuk mencari *material*. Cara kerja *function module get material* adalah mengambil data dari tabel *mvke* lalu mengisi data tersebut ke dalam *structure gi_item* berdasarkan *parameter input sales organization*, *parameter input distribution channel* dan *material number* dari *structure gi_mara*. Kemudian mengambil data dari tabel *makt* dan mengisi data ke dalam *structure gi_makt* berdasarkan *material number* dari *gi_item* dan *system language SAP* yang dipakai. Setelah itu melakukan *looping* pada *gi_item*, setiap iterasi melakukan pembacaan data pada *gi_makt* dengan kriteria *material number*

pada gi_makt sama dengan gi_item. Apabila terdapat kecocokan, maka komponen maktx pada gi_item diisi dengan data dari komponen maktx pada gi_makt.



```
Function Builder: Display ZFM_BASTUT_GET_MATERIAL
Function module: ZFM_BASTUT_GET_MATERIAL Active
Attributes Import Export Changing Tables Exceptions Source code

SELECT * INTO CORRESPONDING FIELDS OF TABLE gi_item FROM mvke
for all entries in gi_mara
WHERE vkorg = sales_org AND
      vtweg = dist_chnl AND
      matnr = gi_mara-matnr.

SELECT matnr maktx
INTO CORRESPONDING FIELDS OF TABLE gi_makt
FROM makt
FOR ALL ENTRIES IN gi_item WHERE
      matnr = gi_item-matnr AND
      spras = sy-langu.

LOOP AT gi_item.
* Get Material Description
  READ TABLE gi_makt WITH KEY matnr = gi_item-matnr.
  IF sy-subrc = 0.
    gi_item-maktx = gi_makt-maktx.
  ENDIF.

  READ TABLE gi_mara WITH KEY matnr = gi_item-matnr.
  IF sy-subrc = 0.

CALL FUNCTION 'CONVERSION_EXIT_CUNIT_OUTPUT'
```

Gambar 3.28 Kode function module get material

Potongan kode pada gambar 3.29 merupakan fungsi pencarian *customer*. Cara kerja dari fungsi ini dengan mengisi data terlebih dahulu yang diambil dari tabel knvv ke *structure* gi_cust berdasarkan *parameter input sales organization, distribution channel, division, dan customer statistics group*, lalu mengisi *structure* gi_kna1 dari tabel kna1 berdasarkan *customer number* dari gi_cust. Kemudian melakukan *looping* pada gi_cust, setiap iterasi dilakukan pembacaan data pada gi_kna1 dengan kriteria *customer number* sama dengan *customer number* pada gi_cust. Apabila ditemukan kecocokan, maka komponen-komponen gi_cust seperti land1, name1, name2, dan lainnya diisi dengan data dari komponen-komponen pada gi_kna1.

```

Function Builder: Display ZFM_SEARCH_CUST
Function module: ZFM_SEARCH_CUST Active
Source code

select * into CORRESPONDING FIELDS OF TABLE GI_CUST from KQVV
where VKORG = GV_VKORG and
VTWEG = GV_VTWEG and
INCO1 = 'CIF' and
VERSG = '1' and
SPART = GV_SPART.

select * into corresponding fields of table gi_kna1 from kna1
for all entries in gi_cust where
kunnr = gi_cust-kunnr.

LOOP AT gi_cust.
* Get Material Description
READ TABLE gi_kna1 WITH KEY kunnr = gi_cust-kunnr.
IF sy-subrc = 0.
    gi_cust-LAND1 = gi_kna1-LAND1.
    gi_cust-NAME1 = gi_kna1-NAME1.
    gi_cust-NAME2 = gi_kna1-NAME2.
    gi_cust-ORT01 = gi_kna1-ORT01.
    gi_cust-POSTLZ = gi_kna1-POSTLZ.
    gi_cust-REGIO = gi_kna1-REGIO.
    gi_cust-TELF1 = gi_kna1-TELF1.
    gi_cust-TELEFX = gi_kna1-TELEFX.
    gi_cust-ADDDP = gi_kna1-ADDDP.

```

Gambar 3.29 Kode *function module search customer*

Potongan kode pada gambar 3.30 merupakan *function module* untuk membuat rekap *sales document (sales report)* yang pernah dibuat. Pada potongan kode di gambar 3.30 terdapat fungsi *CONVERSION_EXIT_ALPHA_INPUT*. Fungsi ini berguna untuk mengonversikan parameter input *sales document number*, kemudian hasil konversi tersebut dimasukkan ke dalam *structure range sales document number*. Setelah itu, mengambil data pada tabel *vbak* dan mengisi ke dalam *structure gi_report* berdasarkan parameter input *sales organization, distribution channel, division, sales office, sales group, range document number*, dan *range tanggal dokumen*.

The screenshot shows the SAP Function Builder interface for the function module 'ZFM_GENERATE_SO_REPORT'. The 'Source code' tab is active, displaying the following code:

```

CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
  EXPORTING
    input      = GV_VBELN
  IMPORTING
    OUTPUT     = LV_VBELN
  .
GR_VBELN-LOW = LV_VBELN.
GR_VBELN-SIGN = 'I'.
GR_VBELN-OPTION = 'EQ'.
GR_VBELN-HIGH = SPACE.
APPEND GR_VBELN.
ENDIF.

select * into corresponding fields of table gi_report from VBAK
where VKORG = GV_VKORG AND
      VTIWEG = GV_VTIWEG AND
      SPART = GV_SPART AND
      VKBUR = GV_VKBUR AND
      VKGRP = GV_VKGRP AND
      VBELN in GR_VBELN AND
      AUDA1 in GR_DOCDATE.

select * into corresponding fields of table gi_vbkd from vbkd
for all entries in gi_report where

```

The status bar at the bottom indicates 'Ln 1, Co 1' and 'Ln 27 - Ln 51 of 93 lines'.

Gambar 3.30 Kode *function module* generate sales order report

Potongan kode pada gambar 3.31 adalah pemanggilan *function module* BAPI_SIMULATE_SALES_ORDER pada SAP system. Parameter input untuk BAPI_SIMULATE_SALES_ORDER berupa *structure*.

The screenshot shows the SAP Function Builder interface for the function module 'ZFM_SIMULATE_ORDER'. The 'Source code' tab is active, displaying the following code:

```

EXPORTING
  order_header_in      = order_header_in
  * CONVERT_FARVW_AUART = ' '
IMPORTING
  * SALESDOCUMENT      =
  * SOLD_TO_PARTY      =
  * SHIP_TO_PARTY      =
  * BILLING_PARTY      =
  return               = return
TABLES
  order_items_in      = order_items_in
  order_partners      = order_partners
  * ORDER_SCHEDULE_IN =
  order_items_out     = order_items_out
  * ORDER_CFGS_REF    =
  * ORDER_CFGS_INST   =
  * ORDER_CFGS_PART_OF =
  * ORDER_CFGS_VALUE  =
  * ORDER_CFGS_BLOB   =
  * ORDER_CCARD       =
  * ORDER_CCARD_EX    =
  * ORDER_SCHEDULE_EX =
  order_condition_ex  = li_order_condition_ex
  * ORDER_INCOMPLETE =

```

The status bar at the bottom indicates 'Ln 1, Co 1' and 'Ln 18 - Ln 42 of 56 lines'.

Gambar 3.31 Kode *function module* simulate sales order

C.2 Pembuatan PHP

Setiap *function module* yang telah dibuat akan dipakai untuk satu PHP *file*. Seperti *page login.php* menggunakan *function module* YFM_LOGIN. Berikut potongan kode dari *page login* pada gambar 3.32.

```
30     echo "<div class='container'>";
31     echo "<a href='index.php'><img src='bootstrap/image/login.png' alt='login'
32         style='position: absolute left: 20px; top: 200px; width: 60px; height: 60px;' /></a>";
33     echo "</div>";
34 }
35 else {
36     $fce = &$sap->NewFunction ("YFM_LOGIN");
37     if ($fce == false ) {
38         $sap->PrintStatus();
39         exit;
40     }
41     $_POST['userid'] = strtoupper($_POST['userid']);
42     $fce->GV_USERID = $_POST['userid'];
43     $fce->GV_PASSWORD = $_POST['pass'];
44
45     $fce->Call();
46
47     if ($fce->GetStatus() == SAPRFC_OK) {
48         $fce->GI_USER->Reset();
49         if ($fce->GV_STATUS) {
50
51             while ($fce->GI_USER->Next() ) {
52                 $_SESSION['USERID'] = $fce->GI_USER->row['USERID'];
53                 $_SESSION['NAME'] = $fce->GI_USER->row['NAME'];
54                 $_SESSION['PASSWORD'] = $fce->GI_USER->row['PASSWORD'];
55                 $_SESSION['SALES_ORG'] = $fce->GI_USER->row['SALES_ORG'];
56                 $_SESSION['SALES_GRP'] = $fce->GI_USER->row['SALES_GRP'];
57                 $_SESSION['SALES_OFF'] = $fce->GI_USER->row['SALES_OFF'];
58                 $_SESSION['DST_CHNL'] = $fce->GI_USER->row['DST_CHNL'];
59                 $_SESSION['DIVISION'] = $fce->GI_USER->row['DIVISION'];
60                 $_SESSION['USER_MENU'] = $fce->GI_USER->row['USER_MENU'];
61             }
62         }
63     }
64 }
```

Gambar 3.32 Kode PHP login

Pada gambar 3.33 ditampilkan potongan kode HTML dari *page login*. Pada *page login* terdapat *input user id*, *input password*, dan *login button*. Namun tidak ada tombol *sign up* dikarenakan *user* telah ditentukan pada tabel ZTA_USER pada SAP oleh BASIS. Hal ini merupakan kriteria yang telah ditentukan pada perancangan *prototype sales online*.

```

5 <title>Login</title>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <!-- Bootstrap CSS and bootstrap datepicker CSS used for styling the demo pages-->
8 <link rel="stylesheet" href="bootstrap/css/style.css">
9 <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
10 <script src="bootstrap/js/jquery.min.js"></script>
11 <script src="bootstrap/js/prefixfree.min.js"></script>
12 <script src="http://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
13 <script type="text/javascript">
14     function upperMe() {
15         document.getElementById("userid").value = document.getElementById("userid").value.
16             toUpperCase();
17     }
18 </script>
19 </head>
20 <body>
21 <div class="body"></div>
22 <div class="grad"></div>
23 <div class="header">
24     <div><span>SAP</span> Online Sales Application</div>
25 </div>
26 <br>
27 <div class="login">
28     <form method="post" action="login.php">
29         <input type="text" placeholder="username" name="userid" id="userid" style="text-
30             transform:uppercase; onchange="upperMe()" "><br>
31         <input type="password" placeholder="password" name="pass" id="pass"><br>
32         <button type="submit" class="btn btn-default">Login</button>
33     </form>
34 </div>
35 </body>
36 </html>

```

Gambar 3.33 Kode HTML login

Untuk pembuatan kode PHP *create sales order* dibagi menjadi 3 yakni pembuatan tampilan awal untuk *input data sales order*, lalu *simulate order* PHP, dan *create order* PHP. Pada gambar 3.34 merupakan kode HTML dari *create sales order*.

```

149 <td><input type="text" name="material10" id="material10" readonly="TRUE"
150     value="" style="width:200px;"></td>
151 <td><input type="button" style="width:30px;" name="material_browse10" id="
152     material_browse10" value="..." onclick="pop_up('search_material.php',10);"></
153     td>
154 <td><input type="text" name="quantity10" id="quantity10" style="width:100%;"
155     value="1"></td>
156 <td><input type="text" name="sales_unit10" id="sales_unit10" style="width:100
157    %;" readonly="TRUE"></td>
158 <td><input type="text" name="description10" id="description10" readonly="TRUE
159     " style="width:400px;"></td>
160 </tr>
161 <tr>
162 <td><input type="text" name="item20" value="20" id="item20" style="width:
163     100%;" readonly="TRUE"></td>
164 <td><input type="text" style="width:200px;" name="material20" id="material20"
165     readonly="TRUE" value=""></td>
166 <td><input type="button" style="width:30px;" name="material_browse20" id="
167     material_browse20" value="..." onclick="pop_up('search_material.php',20);"></
168     td>
169 <td><input type="text" style="width:100%;" name="quantity20" id="quantity20"
170     value="1"></td>
171 <td><input type="text" style="width:100%;" name="sales_unit20" id="
172     sales_unit20" readonly="TRUE"></td>
173 <td><input type="text" style="width:100%;" name="description20" id="
174     description20" readonly="TRUE"></td>
175 </tr>
176 <tr>
177 <td><input type="text" style="width:100%;" name="item30" value="30" id="
178     item30" readonly="TRUE"></td>
179 <td><input type="text" style="width:200px;" name="material30" id="material30"
180     readonly="TRUE" value=""></td>

```

Gambar 3.34 Kode HTML create sales order

Setelah dimasukan semua data-data yang diperlukan untuk membuat sebuah *sales order* maka akan diberikan simulasi data sebelum dibuat menjadi *sales document*. Pada gambar 3.35 dicantumkan potongan kode PHP *simulate order*

```

29     $sap = new SAPConnection();
30     $sap->Connect("logon_data.conf");
31     if ($sap->GetStatus() == SAPRFC_OK ) {
32         $sap->Open ();
33     }
34     if ($sap->GetStatus() != SAPRFC_OK ) {
35         $sap->PrintStatus();
36         exit;
37     }
38     $fce = &$sap->NewFunction ("ZFM_SIMULATE_ORDER");
39     if ($fce == false ) {
40         $sap->PrintStatus();
41         exit;
42     }
43
44     $podate = date("Y-m-d",strtotime($_POST['podate']));
45     $newPodate = str_replace("-", "", $podate);
46     $_SESSION['podate'] = $newPodate;
47
48     $validfrom = date("Y-m-d",strtotime($_POST['validfrom']));
49     $newValidfrom = str_replace("-", "", $validfrom);
50     $_SESSION['validfrom'] = $newValidfrom;
51
52     $validto = date("Y-m-d",strtotime($_POST['validto']));
53     $newValidto = str_replace("-", "", $validto);
54     $_SESSION['validto'] = $newValidto;
55
56     $reqdelivdate = date("Y-m-d",strtotime($_POST['reqdelivdate']));
57     $newreqdelivdate = str_replace("-", "", $reqdelivdate);
58     $_SESSION['reqdelivdate'] = $newreqdelivdate;
59
60     $fce->ORDER_HEADER_IN['DOC_TYPE'] = 'TA';
61     $fce->ORDER_HEADER_IN['SALES_ORG'] = $_POST['sales_org'];

```

Gambar 3.35 Kode PHP *simulate order*

Proses terakhir dalam membuat sales order adalah membuat *sales document*, dimana data-data *sales order parameter* yang telah dimasukan sebelumnya akan tersimpan di dalam tabel SAP. *Function module* yang digunakan untuk membuat *sales document* merupakan BAPI_SALESORDER_CREATEFROMDAT1. *Function module* ini telah disediakan oleh SAP.

```

41     $sap = new SAPConnection();
42     $sap->Connect("logon_data.conf");
43
44     if ($sap->GetStatus() == SAPRFC_OK )
45         $sap->Open ();
46
47     if ($sap->GetStatus() != SAPRFC_OK ) {
48         $sap->PrintStatus();
49         exit;
50     }
51     $fce = &$sap->NewFunction ("BAPI_SALESORDER_CREATEFROMDAT1");
52
53     if ($fce == false ) {
54         $sap->PrintStatus();
55         exit;
56     }
57
58     $fce->ORDER_HEADER_IN['DOC_TYPE'] = 'TA';
59     $fce->ORDER_HEADER_IN['SALES_ORG'] = $_SESSION['SALES_ORG'];
60     $fce->ORDER_HEADER_IN['DISTR_CHAN'] = $_SESSION['DST_CHNL'];
61     $fce->ORDER_HEADER_IN['DIVISION'] = $_SESSION['DIVISION'];
62     $fce->ORDER_HEADER_IN['SALES_GRP'] = $_SESSION['SALES_GRP'];
63     $fce->ORDER_HEADER_IN['SALES_OFF'] = $_SESSION['SALES_OFF'];
64     $fce->ORDER_HEADER_IN['PURCH_NO'] = $_SESSION['ponumb'];
65     $fce->ORDER_HEADER_IN['PURCH_DATE'] = $_SESSION['podate'];
66     $fce->ORDER_HEADER_IN['CT_VALID_F'] = $_SESSION['validfrom'];
67     $fce->ORDER_HEADER_IN['CT_VALID_T'] = $_SESSION['validto'];
68     $fce->ORDER_HEADER_IN['REQ_DATE_H'] = $_SESSION['reqdelivdate'];
69
70     $fce->ORDER_PARTNERS->row['PARTN_ROLE'] = 'AG';
71     $fce->ORDER_PARTNERS->row['PARTN_NUMB'] = $_SESSION['salestoparty'];
72     $fce->ORDER_PARTNERS->Append($fce->ORDER_PARTNERS->row);
73     $fce->ORDER_PARTNERS->row['PARTN_ROLE'] = 'WE';

```

Gambar 3.36 Kode PHP *create sales document*

```

78     </div>
79     <div class = "header_text">REPORT SALES ORDER</div>
80     <div class = "container">
81         <?php if ($_SESSION['USERID'] == 'ADMIN'){
82             echo "<a href='admin.php' class='noPrint'><img src='bootstrap/image/home.png'
83                 alt='logout menu' width='100' height='70' class='noPrint'
84                 style='position:absolute; top:-105px;left:30px;' /></a>";
85             echo "<div style='position:relative; top:-80px; left:130px; font-family:arial; font-
86                 size:16px; font-weight:bold; color:white;'>WELCOME " . $_SESSION['NAME'] . "</div>";
87         }
88         else {
89             echo "<div style='position:relative; top:-85px; left:15px; font-family:arial; font-
90                 size:16px; font-weight:bold; color:white;'>WELCOME " . $_SESSION['NAME'] . "</div>";
91         }
92     }
93     <a href="logout.php">&nbsp;</a>
95     <table class = "report_table">
96         <form action="generate_report.php" method="post">
97             <tr>
98                 <td>Sales Organization</td>
99                 <td style="left:5px; position:relative;"></td>
100                <td><input type = "text" value = <?php echo "" . $_SESSION['SALES_ORG'] . "" ?> name=
101                "sales_org" id="sales_org" readonly="TRUE" style="background-color:#bebec3; left:10
                px; position:relative;"></td>
102            </tr>
103            <tr><td><br/></td></tr>
104            <tr>
105                <td>Distribution Channel</td>
106                <td style="left:5px; position:relative;"></td>
107                <td><input type = "text" value = <?php echo "" . $_SESSION['DST_CHNL'] . "" ?> name="
108                dst_chnl" id="dst_chnl" readonly="TRUE" style="background-color:#bebec3; left:10px;
109                position:relative;"></td>

```

Gambar 3.37 Kode HTML *report page*

Untuk proses *report* digunakan *function module* ZFM_SO_REPORT pada *report.php*. Page ini yang digunakan untuk *print report sales*.

```

47
48     $fce = &$sap->NewFunction ("ZFM_GENERATE_SO_REPORT");
49     if ($fce == false ) {
50         $sap->PrintStatus();
51         exit;
52     }
53
54     $fce->GV_VKORG = $_SESSION['SALES_ORG'];
55     $fce->GV_VTWEG = $_SESSION['DST_CHNL'];
56     $fce->GV_SPART = $_SESSION['DIVISION'];
57     $fce->GV_VKBUR = $_SESSION['SALES_OFF'];
58     $fce->GV_VKGRP = $_SESSION['SALES_GRP'];
59     $fce->GV_VBELN = $_SESSION['sodocnumb'];
60     $fce->GV_AUDATFROM = $_SESSION['sodatefrom'];
61     $fce->GV_AUDATTO = $_SESSION['sodateto'];
62
63
64     $fce->Call();
65
66
67     if ($fce->GetStatus() == SAPRFC_OK) {
68     }
69     <div style="position:absolute; top:100px; left:250px; font-family:arial; font-size:20px; font-weight:bold;"
70     >REPORT SALES ORDER</div>
71     <div style="position:absolute; top:120px; left:350px; font-family:arial; font-size:20px; font-weight:bold;"
72     ><?php echo $_SESSION['datefromori'];?> to <?php echo $_SESSION['datetoori'];?></div>
73     <div style="position:absolute; top:150px; left:20px;">
74     <?php
75         echo "<table class='table table-bordered'>
76             <tr>
77                 <th>Document Number</th>
78                 <th>Document Type</th>

```

Gambar 3.38 Potongan kode PHP *report*

Gambar 3.39 merupakan potongan kode PHP untuk tampilan *display sales document* beserta prosesnya yang menggunakan *function module* ZFM_DISPLAY_SALES_ORDER. Pada potongan kode tersebut, *sales document*

number dijadikan sebagai *input parameter* dengan variabel `GV_VBELN` pada *function module* `ZFM_DISPLAY_SALES_ORDER`. Lalu melakukan *looping* pada `GI_DISPLAY` dan membaca data yang ada satu per satu saat iterasi.

```
94 }
95
96 $fce = &$sap->NewFunction ("ZFM_DISPLAY_SALES_ORDER");
97
98 if ($fce == false ) {
99     $sap->PrintStatus();
100     exit;
101 }
102
103 $fce->GV_VBELN = $_GET['postvbeln'];
104 $fce->Call();
105 if ($fce->GetStatus() == SAPRFC_OK) {
106     $fce->GI_DISPLAY->Reset();
107     $fce->GI_VBAP->Reset();
108     $fce->GI_PARTNER->Reset();
109     echo "<section class='wrapper'>";
110     echo "<h2>Header</h2>";
111     echo "<ul class='tabs'>";
112     echo "<li><a href='#tab1'>Sales Data</a></li>";
113     echo "<li><a href='#tab2'>Partner Data</a></li>";
114     echo "<li><a href='#tab3'>Billing Data</a></li>";
115     echo "</ul>";
116     echo "<div class='clr'></div>";
117     echo "<section class='block'>";
118     echo "<article id='tab1'>";
119     while ( $fce->GI_DISPLAY->Next() ) {
120         echo "<div style='position:relative; margin-top:5px; margin-left:-6px;'><label
121             for='docNumb' style='color:black; font-family:arial;'>$fce->GI_DISPLAY->row[
122             'BEZEI'] "</label><span class='label label-default' id='docNumb' style='font-
123             family:arial; font-size:12px; position:relative; left:69px; margin-left:10px;
124             width:50px; height:20px; display:inline-block;'>".fm_delete_zero_value($fce->
125             GI_DISPLAY->row['VBELN'])."</span></div>";
126         echo "<div style='position:relative; margin-top:5px; margin-left:-6px; left:314px;
127             top:-30px;'><label for='docDate' style='color:black; font-family:arial;'>
```

Gambar 3.39 Kode PHP *display sales document*

Terakhir merupakan potongan kode PHP untuk *search customer* dan *search material* dalam membuat *sales order*, untuk PHP *search customer* menggunakan *function module* `ZFM_SEARCH_CUST` dan PHP *search material* menggunakan *function module* `ZFM_BASTUT_GET_MATERIAL`.

UMMN

```

79
80
81     $fce = &$sap->NewFunction ("ZFM_SEARCH_CUST");
82     if ($fce == false) {
83         $sap->PrintStatus();
84         exit;
85     }
86
87     $fce->GV_VKORG = $_SESSION['SALES_ORG'];
88     $fce->GV_VTWEK = $_SESSION['DST_CHNL'];
89     $fce->GV_SPART = $_SESSION['DIVISION'];
90     $fce->GV_VKBUR = $_SESSION['SALES_OFF'];
91     $fce->GV_VKGRP = $_SESSION['SALES_GRP'];
92
93     $fce->Call();
94
95     if ($fce->GetStatus() == SAPRFC_OK) {
96         echo "<table id='example' class='display' cellspacing='0' width='100%'>
97             <thead>
98                 <tr>
99                     <th>Customer Number</th>
100                    <th>Customer Name</th>
101                    <th>City</th>
102                </tr>
103            </thead>
104
105            <tfoot>
106                <tr>
107                    <th>Customer Number</th>
108                    <th>Customer Name</th>
109                    <th>City</th>
110                </tr>
111            </tfoot>

```

Gambar 3.40 Kode PHP *search customer*

```

70
71     $fce = &$sap->NewFunction ("ZFM_BASTUT_GET_MATERIAL");
72     if ($fce == false) {
73         $sap->PrintStatus();
74         exit;
75     }
76
77     $fce->SALES_ORG = $_SESSION['SALES_ORG'];
78     $fce->DIST_CHNL = $_SESSION['DST_CHNL'];
79     $fce->DIVISION = $_SESSION['DIVISION'];
80
81     $fce->Call();
82
83     if ($fce->GetStatus() == SAPRFC_OK) {
84         echo "<table id='example' class='display' cellspacing='0' width='100%'>
85             <thead>
86                 <tr>
87                     <th>Material Number</th>
88                     <th>Description</th>
89                     <th>Sales Unit</th>
90                 </tr>
91            </thead>
92
93            <tfoot>
94                <tr>
95                    <th>Material Number</th>
96                    <th>Description</th>
97                    <th>Sales Unit</th>
98                </tr>
99            </tfoot>
100            <tbody>";
101     $fce->GI_ITEM->Reset();
102

```

Gambar 3.41 Kode PHP *search material*

Untuk dapat menggunakan *function module* dari SAP dan dapat dipakai PHP maka diperlukan sebuah penghubung, gambar 3.42 merupakan potongan kode PHP untuk koneksi ke SAP *server*.

```

8
9 // The type of SAP connection
10 define ("SAPRFC_CONN_NONE", 0); // no connection
11 define ("SAPRFC_CONN_APPSERVER", 1); // client connection to the single
12 // application server
13 define ("SAPRFC_CONN_LOGONGROUP", 2); // client load balanced connection
14 // to the logon group
15 define ("SAPRFC_CONN_USERDEFINED", 3); // client user defined
16 define ("SAPRFC_CONN_ACCEPT", 4); // server connection
17
18 /**
19 * Class SAPConnection,
20 */
21 class SAPConnection extends SAP {
22 /**
23 * PUBLIC VARS
24 */
25 var $rfc = false; // RFC handle for connection
26 var $connType = SAPRFC_CONN_NONE; // Connection type
27 var $appServer = ""; // Hostname of the application server
28 var $systemNumber = ""; // System number, 00 - default
29 var $gatewayHost = ""; // Hostname of the SAP Gateway
30 var $gatewayService = ""; // Service name of the SAP Gateway (sapgwXX)
31 var $messageServer = ""; // Hostname of the message server for Load Balancing
32 var $R3SystemName = ""; // Name (<SID>) of R/3 System
33 var $logonGroup = ""; // Name of the logon group
34 var $userLogonData = false; // User defined logon data
35 var $trace = false; // RFC trace
36 var $codePage = ""; // SAP Codepage for connection, empty = default
37 var $serverReg = false; // Registration parameters for RFC server
38 var $systemInfo = false; // System information about Remote SAP system
39 var $fclList = array(); // Function list
40
41 /**

```

Gambar 3.42 Kode PHP *sap_connection*

Untuk dapat memanggil sebuah *function module* dalam SAP maka diperlukan *sap_function* PHP. Gambar 3.43 merupakan potongan kode PHP dari *sap_function*.

```

313 } //switch
314 } //for
315 }
316
317 /**
318 * Call function module
319 *
320 * @return SAPRFC_OK on success, SAPRFC_EXCEPTION if exception raised
321 */
322 function Call () {
323     if ( $this->fcl == false )
324         return $this->setStatus (SAPRFC_ERROR,"SAPFunction::Call: Function is ".
325                                 "not defined.");
326     $this->ExportVars();
327     $rc = @saprfc_call_and_receive ( $this->fcl );
328     if ( $rc == SAPRFC_OK )
329         $this->ImportVars();
330     elseif ( $rc == SAPRFC_EXCEPTION )
331         $this->exception = @saprfc_exception ( $this->fcl );
332
333     if ( $this->bapi ) {
334     $bapi_return = $this->RETURN;
335     if ( isset( $bapi_return ) &&
336         is_array( $bapi_return ) &&
337         isset( $bapi_return["MESSAGE"] ) &&
338         $bapi_return["NUMBER"] != 0 ) {
339         return $this->setStatus(SAPRFC_APPL_ERROR,$bapi_return);
340     }
341     }
342     return $this->setStatus ( $rc,@saprfc_error() );
343 }
344
345 /**

```

Gambar 3.43 Kode PHP *sap_function*

Untuk dapat melakukan *query* pada internal tabel di SAP seperti *insert*, *delete*, *modify* atau *read* dari PHP, maka diperlukan *sap_table.php*. Gambar 3.44 merupakan potongan kode dari *sap_table.php*.

```

85  * Insert row into internal table
86  */
87  function Insert ($row, $index=1 ) {
88      if ($this->__CheckParams($index,true,true) == false ) return false;
89      unset ($this->row);
90      $this->row = array();
91      foreach ( $this->rowStruct as $member ) {
92          if ( is_array($row) && isset ($row[$member]) )
93              $this->row[$member] = $row[$member];
94          else
95              $this->row[$member] = "";
96      }
97      return @saprfc_table_insert ($this->fce, $this->name, $this->row, $index);
98  }
99
100 /**
101  * Modify row in internal table
102  */
103  function Modify ($row, $index=1 ) {
104      if ($this->__CheckParams($index,true,true) == false ) return false;
105      unset ($this->row);
106      $this->row = @saprfc_read_table ($this->fce, $this->name, $index );
107      foreach ( $this->rowStruct as $member ) {
108          if ( is_array($row) && isset ($row[$member]) )
109              $this->row[$member] = $row[$member];
110      }
111      return @saprfc_table_modify ($this->fce, $this->name, $this->row, $index);
112  }
113
114 /**
115  * Delete row in internal table
116  */
117  function Delete ($index=1 ) {

```

Gambar 3.44 Kode PHP *sap_table*

Gambar 3.45 merupakan potongan kode php untuk *enable* SAP RFC *extension*, sehingga memungkinkan untuk *interfacing* antara SAP dan PHP. Potongan kode ini memanggil SAP *connection* PHP, SAP *function* PHP, dan SAP *table* PHP.

```

7  */
8
9  if (!defined('SAPCLASSES_DIR')) define('SAPCLASSES_DIR',dirname(__FILE__));
10
11 // Define status values
12 if (!defined("SAPRFC_OK"))      define("SAPRFC_OK",0);
13 if (!defined("SAPRFC_ERROR"))  define("SAPRFC_ERROR",1);
14 if (!defined("SAPRFC_APPL_ERROR")) define("SAPRFC_APPL_ERROR",99);
15
16 /**
17  * Abstract SAP class,
18  * define status handling for derived classes
19  */
20 class SAP {
21     /*
22     * PUBLIC VARS
23     */
24     var $status = SAPRFC_OK; // The status of last method executed
25                             // SAPRFC_OK (0) means success
26     var $statusInfos = ""; // More status info (messages, BAPI struct)
27     var $debug = false; // Debug output
28
29     /**
30     * Constructor
31     */
32     function SAP()
33     {
34         if ( $this->CheckExtension() == false )
35             die('SAPRFC extension is not installed. Exiting....');
36     }
37
38     /**
39     * Set status, show errors if debug set

```

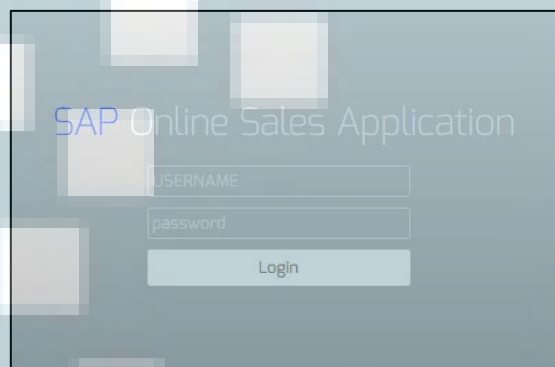
Gambar 3.45 Kode PHP *class* SAP RFC *extension*

D. Implementasi

Berikut ini merupakan hasil implementasi dari pengembangan *prototype sales online* untuk kebutuhan *proof of concept* selama melakukan kerja magang di CV. Difo Lestari. Hasil implementasi berupa gambar dari tampilan *prototype sales online*.

1. Halaman *login*

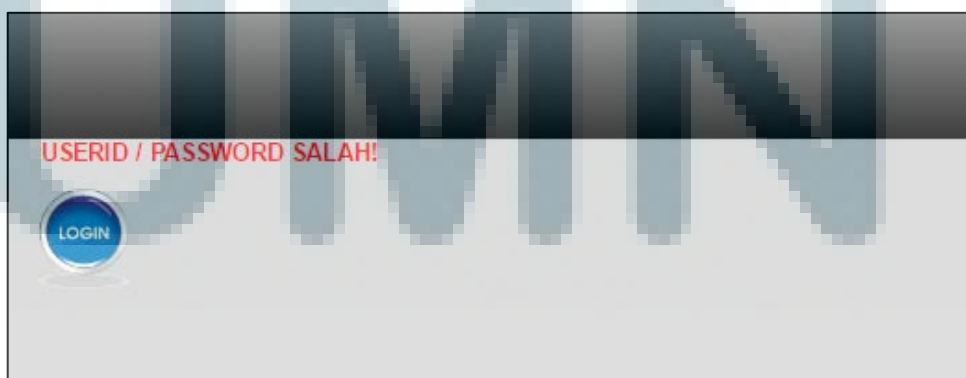
Halaman ini yang akan ditampilkan pertama kali saat mengakses *website prototype sales online*. *User* terbagi menjadi 3 dengan hak akses masing-masing yakni *admin*, *sales*, dan *manager*. *User* harus melakukan *login* sebelum mengakses *website*.

The image shows a login form for the SAP Online Sales Application. The form is centered on a light blue background. It features the title 'SAP Online Sales Application' at the top. Below the title, there are two input fields: 'USERNAME' and 'password'. A 'Login' button is positioned below the password field. The entire form is enclosed in a black rectangular border.

Gambar 3.46 *Form login*

2. Halaman notifikasi *error*

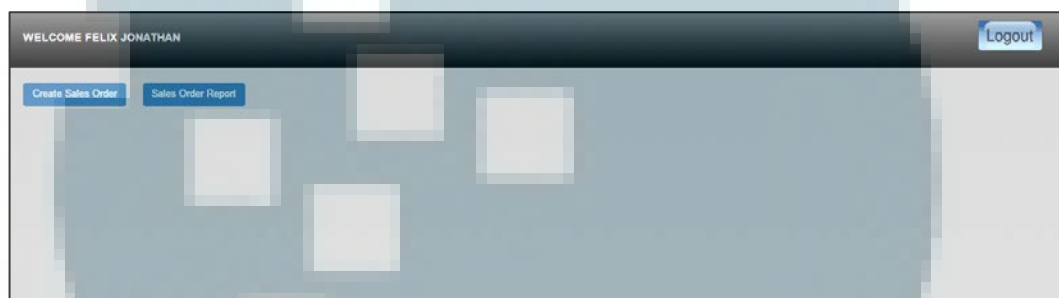
Sebuah sistem yang baik harus memberikan pesan atau *feedback* apabila terjadi kesalahan sehingga *user* dapat mengetahui kesalahannya dan diharapkan dapat memperbaikinya. Berikut contoh tampilan pesan apabila terjadi kesalahan



Gambar 3.47 Tampilan pesan kesalahan

3. Halaman *admin*

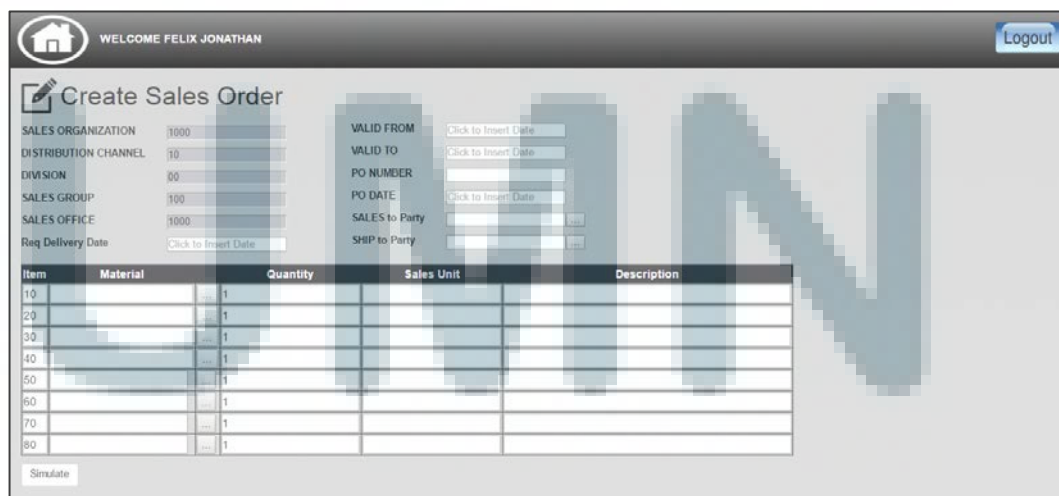
Setiap *user* memiliki hak akses yang berbeda-beda, *admin* dapat membuat *sales document* dan melihat *sales report*, akan tetapi untuk *sales person* dan *manager* hanya dapat membuat *sales document* untuk *sales person* dan *manager* hanya dapat melihat *sales report*. Setelah *admin* melakukan proses verifikasi / *login* maka sistem akan otomatis mengarahkan ke halaman *admin*. Berikut contoh tampilan halaman *admin*.



Gambar 3.48 Tampilan halaman *admin*

4. Halaman *create sales order*

User sebagai *sales person* ketika berhasil melakukan proses *login* maka sistem akan otomatis mengarahkan ke halaman *create sales order*. Berikut contoh tampilan *create sales order*.

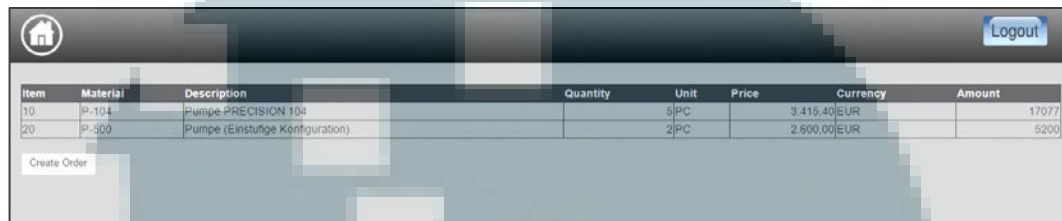
The screenshot shows the 'Create Sales Order' form. At the top left, there is a home icon and 'WELCOME FELIX JONATHAN'. At the top right, there is a 'Logout' button. The main heading is 'Create Sales Order'. Below the heading, there are several input fields for selection and date picking. A table is visible at the bottom of the form, and a 'Simulate' button is at the bottom left.

Item	Material	Quantity	Sales Unit	Description
10		1		
20		1		
30		1		
40		1		
50		1		
60		1		
70		1		
80		1		

Gambar 3.49 Tampilan halaman *create sales order*

5. Halaman *simulate sales order*

Simulate sales order merupakan fitur pengecekan ulang sebelum melakukan pembuatan *sales order* (*sales document*). Pada halaman ini yang ditampilkan adalah beberapa *material* yang ingin dijual beserta harganya. Berikut contoh tampilan dari *simulate sales order*.



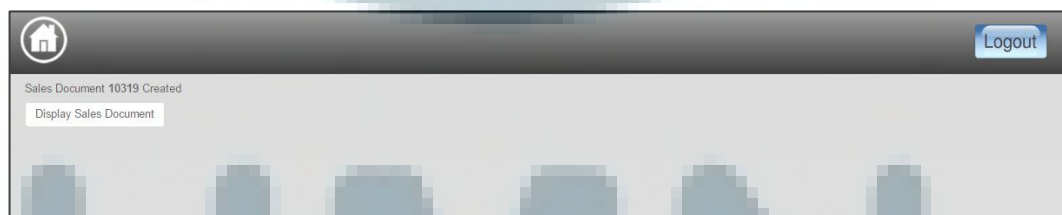
Item	Material	Description	Quantity	Unit	Price	Currency	Amount
10	P-104	Pumpe PRECISION 104		5/PC	3.415,40	EUR	17077
20	P-500	Pumpe (Einstufige Konfiguration)		2/PC	2.600,00	EUR	5200

Create Order

Gambar 3.50 Tampilan halaman *simulate sales order*

6. Halaman notifikasi pembuatan *sales document*

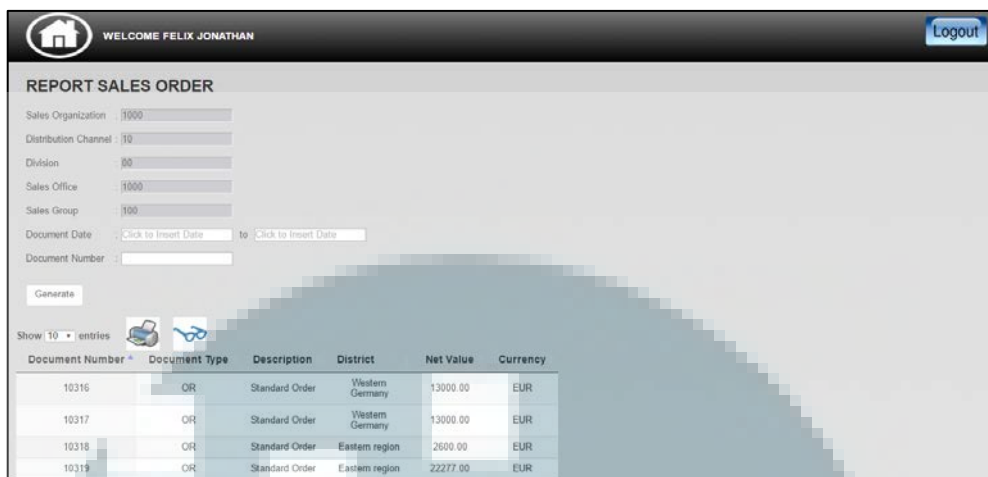
Setelah *user* yakin dengan *material* yang dimasukkan ke dalam sistem, maka sistem akan melakukan penyimpanan dan pembuatan *sales document*. Apabila tidak ada kesalahan dalam proses memasukan data, maka sistem akan memberikan pesan berhasil dalam membuat dokumen penjualan dan menampilkan nomor dokumen penjualan. Berikut contoh tampilan notifikasi pembuatan *sales document*.



Gambar 3.51 Tampilan notifikasi pembuatan *sales document*

7. Halaman *generate sales report*

Halaman ini hanya dapat diakses oleh *user* sebagai *manager* dan *admin*. *User* yang melakukan proses *login* sebagai *manager* maka sistem akan otomatis mengarahkan ke halaman ini setelah berhasil melakukan proses verifikasi. Berikut contoh tampilan laporan penjualan.



Gambar 3.52 Form generate sales report

8. Halaman *display sales document*

Halaman ini dapat diakses ketika berhasil melakukan pembuatan *sales document* dan saat melakukan permintaan pemunculan laporan penjualan dengan cara memilih dari salah satu dokumen yang ditampilkan dalam daftar penjualan tersebut. Berikut contoh tampilan dari *display sales document*.



Gambar 3.53 Tampilan *display sales document*

9. Menu *logout*

Menu *logout* akan membuat *user* keluar dari sistem dan otomatis akan kembali ke halaman awal yaitu halaman *login*. Menu ini berupa tombol *logout* disetiap *header page*.

3.3.2 Kendala yang ditemukan

Terdapat kendala yang dialami selama proses kerja magang khususnya dari segi teknis pembuatan aplikasi dikarenakan banyaknya perubahan-perubahan *requirement* yang dikarenakan perbedaan pendapat antara dengan *board of director*, *technical director*, dan *member*. Tidak hanya itu saja ada kesulitan dalam membuat desain antarmuka agar mempunyai tampilan yang menarik dan mudah untuk digunakan.

3.3.3 Solusi atas kendala yang ditemukan

Kendala tersebut sebenarnya sudah menjadi bagian dari resiko pengembangan suatu aplikasi. Oleh karena itu, solusi dari kendala tersebut adalah ikut terlibat dalam *brainstorming* dengan *technical director*, *board of director*, dan beberapa *member* untuk menentukan fitur, desain tampilan, dan beberapa struktur data seperti tabel, *data element*, *structure*, dan lain-lain. Serta menggunakan bootstrap untuk membuat tampilan *website* agar terlihat menarik.

UMMN