LAMPIRAN A SPESIFIKASI APLIKASI PERBANKAN DALAM METODE-B

{ Deklarasi mesin Bank dengan maxCustomers dan max Accounts sebagai paramaternya*}*

MACHINE

Bank(*maxCustomers*, *maxAccounts*)

CONSTRAINTS

maxCustomers ∈1 ..100000 ∧ *maxAccounts* ∈1 ..200000

SEES

StrTokenType

DEFINITIONS

CUSTOMER == 0 ..*maxCustomers*-1;

ACCOUNT == 0 *maxAccounts*-1

{ Deklarasi variabel yang dipakai oleh mesin Bank*}*

VARIABLES

customers, *customerName*, *customerYob*,
accounts, *accountNumber*, *accountPin*, *accountBalance*,
accountOwner, *foundCustomers*

CONCRETE VARIABLES

fileOpen

INVARIANT

customers ⊆ *CUSTOMER* ∧
customerName ∈ *customers* → *STRTOKEN* ∧
customerYob ∈ *customers* → **NAT** ∧
customerName ⊗ *customerYob* ∈
customers ⤔ (*STRTOKEN* x **NAT**) ∧
accounts ⊆ *ACCOUNT* ∧
accountNumber ∈ *accounts* ⤔ **NAT** ∧
accountPin ∈ *accounts* → **NAT** ∧
accountBalance ∈ *accounts* → **NAT** ∧
accountOwner ∈ *accounts* → *customers* ∧
fileOpen ∈ **BOOL** ∧ *foundCustomers* ⊆ *customers*

{ inisialisasi Customer*}*
INITIALISATION
customers : = θ ‖*customerName* : = θ ‖*customerYob* := θ ‖

{ Operasi untuk membuat new Customer *}*
OPERATIONS
NewCustomer(*name*, *yob*) =
PRE
name ∈ *STRTOKEN* ∧ *yob* ∈ **NAT** ∧
(*name*, *yob*) ∉ **ran**(*customerName* ⊗ *customerYob*) ∧
Customers ≠ *CUSTOMER* ∧ *fileOpen* = **TRUE**
THEN
ANY *newCustomer* **WHERE**
newCustomer ∈ *CUSTOMER* - *customers*
THEN
customers : = *customers* ∪ {*newCustomer*} ||
customerName(*newCustomer*) : = *name* ||
customerYob(*newCustomer*) : = *yob*
END
END;

{ Operasi CustomerData digunakan untuk mengambil data tentang nama dan tahun lahir nasabah*}*
name, *yob* ← **CustomerData**(*cid*) =
PRE
cid ∈ *customers* ∧ *cid* ∈ *CUSTOMER* ∧ *fileOpen* = **TRUE**
THEN
name : = *customerName*(*cid*) || *yob* : = *customerYob*(*cid*)
END;
{ Operasi Balance digunakan untuk mengambil data tentang saldo dan PIN nasabah*}*
bal ← **Balance**(*aid*, *pin*) =
PRE
aid ∈ *accounts* ∧ *aid* ∈ *ACCOUNT* ∧ *pin* ∈ **NAT** ∧
accountPin(*aid*) = *pin* ∧ *fileOpen* = **TRUE**
THEN
bal : = *accountBalance*(*aid*)
END;
….

END
{ Deklarasi dari mesin RobustBank*}*
MACHINE
RobustBank(*maxCustomers*, *maxAccounts*)
CONSTRAINTS
maxCustomers ∈ 1 ..100000 ∧ *maxAccounts* ∈ 1 ..200000
INCLUDES
BK.Bank(*maxCustomers*, *maxAccounts*)
SEES
StrTokenType
DEFINITIONS
CUSTOMER == 0 ..*maxCustomers*-1;
ACCOUNT == 0 *maxAccounts*-1
SETS
RESULT = {*success*, *dbFull*, *dbError*,
customerAlreadyPresent, ..}

{ Operasi RobustCustomerData digunakan untuk mengambil data tentang status nasabah dan basis data*}*
OPERATIONS
result ← **RobustNewCustomer**(*name*, *yob*) =
PRE *name* ∈ *STRTOKEN* ∧ *yob* ∈ **NAT THEN**
IF *BK.fileOpen* = **TRUE THEN**
IF *BK.customers* ≠ *CUSTOMER* **THEN**
IF (*name*,*yob*) ∉
ran(*BK.customerName* ⊗ *BK.customerYob*)
THEN
result : = *success* || *BK.NewCustomer*(*name*, *yob*)
ELSE *result* : = *customerAlreadyPresent*
END
ELSE *result* : = *dbFull*
END
ELSE *result* : = *dbError*
END
END
….
END

```
{* Deklarasi mesin BasicCGI*}
MACHINE
        BasicCGI
SEES
        StrTokenType

{* Operasi ReadNAT digunakan untuk memberikan statusinput program*}
OPERATIONS
        status, num ← ReadNat(name) =
                PRE name ∈ STRING THEN
                        status : ∈ BOOL || num :∈ NAT
                END;
        status, str ← ReadTokenString(name, maxLength) =
                PRE name ∈ STRING ∧ maxLength ∈ NAT THEN
                        status : ∈ BOOL || str :∈ STRTOKEN
                END;
                ..
END

{* Implementasi*}
IMPLEMENTATION
        OperationsBank_1
REFINES
        OperationsBank
IMPORTS
        RB.RobustBank(100, 200)
SEES
        BC.BasicCGI
DEFINITIONS
        CASHIER_HEADER(title) == HEADER(title, "#228B22");
        HEADER(title,color) == (
                BC.WriteLiteralContentType("text/html");
                BC.WriteLiteralString("<HTML><HEAD><TITLE>
                        B Bank: ");
                BC.WriteLiteralString(title);
                BC.WriteLiteralString("</TITLE></HEAD>nn");
                ..
        );
        FOOTER == ..
```

```
    NAME LEN == 256

OPERATIONS
{* pembuatan nasabah baru*}
NewCustomer =
    VAR st, name, yob, result IN
        CASHIER_HEADER("New Customer");
        st, name ← BC.ReadTokenString("name", NAME_LEN);
        IF st = TRUE THEN
            st, yob ← BC.ReadNat("yob");
            IF st = TRUE THEN
                result ← RB.RobustNewCustomer(name, yob);
                CASE result OF
                    EITHER success THEN
                    BC.WriteLiteralString("<P>Customer ");
                    BC.WriteLatin1TokenString(name);
                    BC.WriteLiteralString(" ("); BC.WriteNat(yob);
                    BC.WriteLiteralString(") has been added.</P>")
                OR customerAlreadyPresent THEN
                    ..
                END
            END
        ELSE BC.WriteLiteralString(
            "<P>Could not get year of birth.</P>")
        END
        ELSE
            BC.WriteLiteralString("<P>Could not get name.</P>")
        END;
        FOOTER
    END;
    ..
END
```

LAMPIRAN B VERIFICATION CONDITIONS (VCs) DAN HASIL VERIFIKASI

Modul Deposit

VCs:

```
STATUS: DONE
> val it = () : unit
> val it =
    ref [("MainSpec",
          [({VC =
                ``~SOMEof DepositTab (satisfy r. T) /\
                  SOMEof AccountsTab (satisfy r. T) ==>
                  SOMEof (r INSERT DepositTab) satisfy r. T``,
             VClabel = "init"},
            SOME|- ~SOMEof DepositTab (satisfy r. T) /\
                   SOMEof AccountsTab (satisfy r. T) ==>
                   SOMEof (r INSERT DepositTab) satisfy r. T)])] :
  (string * ({VC : term, VClabel : string} * thm option) list) list ref
```

Hasil Verifikasi:

```
> val it =
    [("MainSpec.init",
      |- ~SOMEof DepositTab (satisfy r. T) /\
         SOMEof AccountsTab (satisfy r. T) ==>
         SOMEof (r INSERT DepositTab) satisfy r. T)] : (string * thm) list


 DEPOSIT DONE
 > val it = () : unit
[closing file "deposit.smx"]
> val it = () : unit
```

Modul Withdraw

VCs:

```
STATUS: DONE
> val it = () : unit
> val it =
    ref [("MainSpec",
          [({VC =
                ``~SOMEof WithdrawTab (satisfy r. T) /\
                  SOMEof AccountsTab (satisfy r. T) ==>
                  SOMEof (r INSERT WithdrawTab) satisfy r. T``,
             VClabel = "init"},
            SOME|- ~SOMEof WithdrawTab (satisfy r. T) /\
                   SOMEof AccountsTab (satisfy r. T) ==>
                   SOMEof (r INSERT WithdrawTab) satisfy r. T)])] :
  (string * ({VC : term, VClabel : string} * thm option) list) list ref
```

Hasil Verifikasi:

```
> val it =
    [("MainSpec.init",
      |- ~SOMEof WithdrawTab (satisfy r. T) /\
         SOMEof AccountsTab (satisfy r. T) ==>
         SOMEof (r INSERT WithdrawTab) satisfy r. T)] : (string * thm) list


 WITHDRAW DONE
 > val it = () : unit
[closing file "withdraw.smx"]
> val it = () : unit
```

Modul Filter Invalid Customer

VCs:

```
STATUS: DONE
> val it = () : unit
> val it =
    ref [("MainSpec",
          [({VC =
               ``~SOMEof InvalidTransactionTab (satisfy r. T) ==>
               ALLof
                 (insertX
                    (deleteX WithdrawTab
                       drop r.
                         SOMEof CustomersTab
                           satisfy t.
                             SOMEof AccountsTab
                               satisfy s.
                                 (r.CustomerID = t.ID) /\ (s.CustomerID = t.ID))
                    (insertX
                       (deleteX DepositTab
                          drop r.
                            SOMEof CustomersTab
                              satisfy t.
                                SOMEof AccountsTab
                                  satisfy s.
                                    (r.CustomerID = t.ID) /\ (s.CustomerID = t.ID))
                       InvalidTransactionTab (\x. x) only r. T) (\x. x) only r. T)
                  satisfy r.
                    ~SOMEof CustomersTab
                      satisfy t.
                        SOMEof AccountsTab
                          satisfy s. (t.ID = r.CustomerID) /\ (s.CustomerID = t.ID)``,
             VClabel = "init"},
            SOME[ ~SOMEof InvalidTransactionTab (satisfy r. T) -->
                  ALLof
                    (insertX
                       (deleteX WithdrawTab
                          drop r.
                            SOMEof CustomersTab
                              satisfy t.
                                SOMEof AccountsTab
                                  satisfy s.
                                    (r.CustomerID = t.ID) /\ (s.CustomerID = t.ID))
                       (insertX
                          (deleteX DepositTab
                             drop r.
                               SOMEof CustomersTab
                                 satisfy t.
                                   SOMEof AccountsTab
                                     satisfy s.
                                       (r.CustomerID = t.ID) /\ (s.CustomerID = t.ID))
                          InvalidTransactionTab (\x. x) only r. T) (\x. x) only r. T)
                     satisfy r.
                       ~SOMEof CustomersTab
                         satisfy t.
                           SOMEof AccountsTab
                             satisfy s. (t.ID = r.CustomerID) /\ (s.CustomerID = t.ID))])]
    : (string * ({VC : term, VClabel : string} * thm option) list) list ref
```

Hasil Verifikasi:

```
> val it =
    [("MainSpec.init",
      !- ~SOMEof InvalidTransactionTab (satisfy r. T) ==>
         ALLof
           (insertX
              (deleteX WithdrawTab
                 drop r.
                   SOMEof CustomersTab
                    satisfy t.
                      SOMEof AccountsTab
                       satisfy s.
                         (r.CustomerID = t.ID) /\ (s.CustomerID = t.ID))
            (insertX
               (deleteX DepositTab
                  drop r.
                    SOMEof CustomersTab
                     satisfy t.
                       SOMEof AccountsTab
                        satisfy s.
                          (r.CustomerID = t.ID) /\ (s.CustomerID = t.ID))
               InvalidTransactionTab (\x. x) only r. T) (\x. x) only r. T)
          satisfy r.
            ~SOMEof CustomersTab
             satisfy t.
               SOMEof AccountsTab
                satisfy s. (t.ID = r.CustomerID) /\ (s.CustomerID = t.ID))]
     : (string * thm) list

 FILTER INVALID CUSTOMER DONE
```

Modul Filter Invalid Account

VCs:

```
STATUS: DONE
> val it = () : unit
> val it =
    ref [("MainSpec",
          [({VC =
               ``~SOMEof InvalidTransactionTab (satisfy r. T) ==>
                ALLof
                  (insertX
                     (deleteX WithdrawTab
                        drop r. SOMEof AccountsTab satisfy t. r.AccountID = t.ID)
                     (insertX
                        (deleteX DepositTab
                           drop r. SOMEof AccountsTab satisfy t. r.AccountID = t.ID)
                        InvalidTransactionTab (\x. x) only r. T) (\x. x) only r. T)
                  satisfy r. ~SOMEof AccountsTab satisfy t. r.AccountID = t.ID``,
            VClabel = "init"},
           SOME!- ~SOMEof InvalidTransactionTab (satisfy r. T) ==>
                  ALLof
                    (insertX
                       (deleteX WithdrawTab
                          drop r. SOMEof AccountsTab satisfy t. r.AccountID = t.ID)
                       (insertX
                          (deleteX DepositTab
                             drop r. SOMEof AccountsTab satisfy t. r.AccountID = t.ID)
                          InvalidTransactionTab (\x. x) only r. T) (\x. x) only r. T)
                    satisfy r. ~SOMEof AccountsTab satisfy t. r.AccountID = t.ID)])]
    : (string * ({VC : term, VClabel : string} * thm option) list) list ref
```

Hasil Verifikasi:

```
> val it =
    [("MainSpec.init",
      !- ~SOMEof InvalidTransactionTab (satisfy r. T) ==>
         ALLof
           (insertX
              (deleteX WithdrawTab
                 drop r. SOMEof AccountsTab satisfy t. r.AccountID = t.ID)
              (insertX
                 (deleteX DepositTab
                    drop r. SOMEof AccountsTab satisfy t. r.AccountID = t.ID)
                 InvalidTransactionTab (\x. x) only r. T) (\x. x) only r. T)
           satisfy r. ~SOMEof AccountsTab satisfy t. r.AccountID = t.ID)] :
  (string * thm) list


 FILTER INVALID ACCOUNT DONE
```

Modul Check PIN

VCs:

```
STATUS: DONE
> val it = () : unit
> val it =
    ref [("MainSpec",
          [((VC =
                ``~SOMEof InvalidTransactionTab (satisfy r. T) ==>
                 ALLof
                   (insertX
                      (deleteX WithdrawTab
                         drop r. SOMEof AccountsTab satisfy t. r.NomorPIN = t.NomorPIN)
                      InvalidTransactionTab (\x. x) only r. T)
                   satisfy r. ~SOMEof AccountsTab satisfy t. r.NomorPIN = t.NomorPIN``,
            VClabel = "init"),
           SOME!- ~SOMEof InvalidTransactionTab (satisfy r. T) ==>
                  ALLof
                    (insertX
                       (deleteX WithdrawTab
                          drop r.
                            SOMEof AccountsTab satisfy t. r.NomorPIN = t.NomorPIN)
                       InvalidTransactionTab (\x. x) only r. T)
                    satisfy r. ~SOMEof AccountsTab satisfy t. r.NomorPIN = t.NomorPIN)])]
    : (string * ((VC : term, VClabel : string) * thm option) list) list ref
```

Hasil Verifikasi:

```
> val it =
    [("MainSpec.init",
      !- ~SOMEof InvalidTransactionTab (satisfy r. T) ==>
         ALLof
           (insertX
              (deleteX WithdrawTab
                 drop r.
                   SOMEof AccountsTab satisfy t. r.NomorPIN = t.NomorPIN)
              InvalidTransactionTab (\x. x) only r. T)
           satisfy r. ~SOMEof AccountsTab satisfy t. r.NomorPIN = t.NomorPIN)]
    : (string * thm) list


 CHECK PIN DONE
 > val it = () : unit
[closing file "checkpin.smx"]
> val it = () : unit
```