AnimLayer.java

```
package com.kyindo.game;

import java.util.ArrayList;
import java.util.List;

import android.content.Context;
import android.util.AttributeSet;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.view.animation.TranslateAnimation;
import android.widget.FrameLayout;

public class AnimLayer extends FrameLayout {

	public AnimLayer(Context context, AttributeSet attrs, int 
defStyle) {
		super(context, attrs, defStyle);
		initLayer();
	}

	public AnimLayer(Context context, AttributeSet attrs) {
		super(context, attrs);
		initLayer();
	}

	public AnimLayer(Context context) {
		super(context);
		initLayer();
	}

	private void initLayer(){
	}

	public void createMoveAnim(final Card from,final Card to,int 
fromX,int toX,int fromY,int toY){

		final Card c = getCard(from.getNum());

		LayoutParams lp = new LayoutParams(Config.CARD_WIDTH, 
Config.CARD_WIDTH);
		lp.leftMargin = fromX*Config.CARD_WIDTH;
		lp.topMargin = fromY*Config.CARD_WIDTH;
		c.setLayoutParams(lp);

		if (to.getNum()<=0) {
			to.getLabel().setVisibility(View.INVISIBLE);
		}
		TranslateAnimation ta = new TranslateAnimation(0, 
Config.CARD_WIDTH*(toX-fromX), 0, Config.CARD_WIDTH*(toY-fromY));
		ta.setDuration(100);
		ta.setAnimationListener(new Animation.AnimationListener() {
```

```
            @Override
            public void onAnimationStart(Animation animation) {}

            @Override
           public void onAnimationRepeat(Animation animation) {}

            @Override
            public void onAnimationEnd(Animation animation) {
                to.getLabel().setVisibility(View.VISIBLE);
                recycleCard(c);
            }
        });
        c.startAnimation(ta);
    }

    private Card getCard(int num){
        Card c;
        if (cards.size()>0) {
            c = cards.remove(0);
        }else{
            c = new Card(getContext());
            addView(c);
        }
        c.setVisibility(View.VISIBLE);
        c.setNum(num);
        return c;
    }
    private void recycleCard(Card c){
        c.setVisibility(View.INVISIBLE);
        c.setAnimation(null);
        cards.add(c);
    }
    private List<Card> cards = new ArrayList<Card>();

    public void createScaleTo1(Card target){
        ScaleAnimation sa = new ScaleAnimation(0.1f, 1, 0.1f, 1, 
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.5f);
        sa.setDuration(100);
        target.setAnimation(null);
        target.getLabel().startAnimation(sa);
    }

}
```

Card.java

```java
package com.kyindo.game;

import android.content.Context;
import android.view.Gravity;
import android.view.View;
import android.widget.FrameLayout;
import android.widget.TextView;

public class Card extends FrameLayout {

        public Card(Context context) {
                super(context);

                LayoutParams lp = null;

                background = new View(getContext());
                lp = new LayoutParams(-1, -1);
                lp.setMargins(10, 10, 0, 0);
                background.setBackgroundColor(0x33ffffff);
                addView(background, lp);

                label = new TextView(getContext());
                label.setTextSize(25);
                label.setGravity(Gravity.CENTER);

                lp = new LayoutParams(-1, -1);
                lp.setMargins(10, 10, 0, 0);
                addView(label, lp);

                setNum(0);
        }


        private int num = 0;

        public int getNum() {
                return num;
        }

        public void setNum(int num) {
                this.num = num;

                if (num<=0) {
                        label.setText("");
                }else{
                        label.setText(num+"");
                }

                switch (num) {
                case 0:
                        label.setBackgroundColor(0x00000000);
                        break;
                case 2:
```

```
                label.setBackgroundColor(0xffeee4da);
                break;
            case 4:
                label.setBackgroundColor(0xffede0c8);
                break;
            case 8:
                label.setBackgroundColor(0xfff2b179);
                break;
            case 16:
                label.setBackgroundColor(0xfff59563);
                break;
            case 32:
                label.setBackgroundColor(0xfff67c5f);
                break;
            case 64:
                label.setBackgroundColor(0xfff65e3b);
                break;
            case 128:
                label.setBackgroundColor(0xffedcf72);
                break;
            case 256:
                label.setBackgroundColor(0xffedcc61);
                break;
            case 512:
                label.setBackgroundColor(0xffedc850);
                break;
            case 1024:
                label.setBackgroundColor(0xffedc53f);
                break;
            case 2048:
                label.setBackgroundColor(0xffedc22e);
                break;
            default:
                label.setBackgroundColor(0xff3c3a32);
                break;
            }
    }

    public boolean equals(Card o) {
            return getNum()==o.getNum();
    }

    protected Card clone(){
            Card c= new Card(getContext());
            c.setNum(getNum());
            return c;
    }

    public TextView getLabel() {
            return label;
    }

    private TextView label;
    private View background;
```

```
}
```

Config.java

```
package com.kyindo.game;

public class Config {

	public static final int LINES = 4;
	public static int CARD_WIDTH = 0;
}
```

GameView.java

```
package com.kyindo.game;

import java.util.ArrayList;
import java.util.List;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.graphics.Point;
import android.util.AttributeSet;
import android.view.MotionEvent;
import android.view.View;
import android.widget.LinearLayout;

public class GameView extends LinearLayout {

        public GameView(Context context) {
                super(context);

                initGameView();
        }

        public GameView(Context context, AttributeSet attrs) {
                super(context, attrs);

                initGameView();
        }

        private void initGameView(){
                setOrientation(LinearLayout.VERTICAL);
                setBackgroundColor(0xffbbada0);


                setOnTouchListener(new View.OnTouchListener() {

                        private float startX,startY,offsetX,offsetY;

                        @Override
                        public boolean onTouch(View v, MotionEvent event) {

                                switch (event.getAction()) {
                                case MotionEvent.ACTION_DOWN:
                                        startX = event.getX();
                                        startY = event.getY();
                                        break;
                                case MotionEvent.ACTION_UP:
                                        offsetX = event.getX()-startX;
                                        offsetY = event.getY()-startY;


                                       if (Math.abs(offsetX)>Math.abs(offsetY))
{
                                                if (offsetX<-5) {
    }

                            swipeLeft();
                        }else if (offsetX>5) {
                            swipeRight();
                        }
                    }else{
                        if (offsetY<-5) {
                            swipeUp();
                        }else if (offsetY>5) {
                            swipeDown();
                        }
                    }

                    break;
                }
                return true;
            }
        });
    }

    @Override
    protected void onSizeChanged(int w, int h, int oldw, int oldh) {
        super.onSizeChanged(w, h, oldw, oldh);

        Config.CARD_WIDTH = (Math.min(w, h)-10)/Config.LINES;

        addCards(Config.CARD_WIDTH,Config.CARD_WIDTH);

        startGame();
    }

    private void addCards(int cardWidth,int cardHeight){

        Card c;

        LinearLayout line;
        LinearLayout.LayoutParams lineLp;

        for (int y = 0; y < Config.LINES; y++) {
            line = new LinearLayout(getContext());
            lineLp = new LinearLayout.LayoutParams(-1,
cardHeight);
            addView(line, lineLp);

            for (int x = 0; x < Config.LINES; x++) {
                c = new Card(getContext());
                line.addView(c, cardWidth, cardHeight);

                cardsMap[x][y] = c;
            }
        }
    }

    public void startGame(){
        MainActivity aty = MainActivity.getMainActivity();
        aty.clearScore();
        aty.showBestScore(aty.getBestScore());

        for (int y = 0; y < Config.LINES; y++) {
            for (int x = 0; x < Config.LINES; x++) {
                cardsMap[x][y].setNum(0);
            }
        }

        addRandomNum();
        addRandomNum();
    }

    private void addRandomNum(){

        emptyPoints.clear();

        for (int y = 0; y < Config.LINES; y++) {
            for (int x = 0; x < Config.LINES; x++) {
                if (cardsMap[x][y].getNum()<=0) {
                    emptyPoints.add(new Point(x, y));
                }
            }
        }

        if (emptyPoints.size()>0) {

            Point p = 
emptyPoints.remove((int)(Math.random()*emptyPoints.size()));
            cardsMap[p.x][p.y].setNum(Math.random()>0.1?2:4);


    MainActivity.getMainActivity().getAnimLayer().createScaleTo1(card
sMap[p.x][p.y]);
        }
    }


    private void swipeLeft(){

        boolean merge = false;

        for (int y = 0; y < Config.LINES; y++) {
            for (int x = 0; x < Config.LINES; x++) {

                for (int x1 = x+1; x1 < Config.LINES; x1++) {
                    if (cardsMap[x1][y].getNum()>0) {

                        if (cardsMap[x][y].getNum()<=0) 
{
       MainActivity.getMainActivity().getAnimLayer().createMoveAnim(card
sMap[x1][y],cardsMap[x][y], x1, x, y, y);


       cardsMap[x][y].setNum(cardsMap[x1][y].getNum());
                                                      cardsMap[x1][y].setNum(0);

                                                      x--;
                                                      merge = true;

                                              }else if
(cardsMap[x][y].equals(cardsMap[x1][y])) {

       MainActivity.getMainActivity().getAnimLayer().createMoveAnim(card
sMap[x1][y], cardsMap[x][y],x1, x, y, y);

       cardsMap[x][y].setNum(cardsMap[x][y].getNum()*2);
                                                      cardsMap[x1][y].setNum(0);


       MainActivity.getMainActivity().addScore(cardsMap[x][y].getNum());
                                                      merge = true;
                                              }

                                              break;
                                      }
                              }
                      }
              }

              if (merge) {
                      addRandomNum();
                      checkComplete();
              }
       }
       private void swipeRight(){

              boolean merge = false;

              for (int y = 0; y < Config.LINES; y++) {
                      for (int x = Config.LINES-1; x >=0; x--) {

                              for (int x1 = x-1; x1 >=0; x1--) {
                                      if (cardsMap[x1][y].getNum()>0) {

                                              if (cardsMap[x][y].getNum()<=0)
{

       MainActivity.getMainActivity().getAnimLayer().createMoveAnim(card
sMap[x1][y], cardsMap[x][y],x1, x, y, y);

       cardsMap[x][y].setNum(cardsMap[x1][y].getNum());
                                                      cardsMap[x1][y].setNum(0);
```

```
                            x++;
                            merge = true;
                        }else if
(cardsMap[x][y].equals(cardsMap[x1][y])) {

    MainActivity.getMainActivity().getAnimLayer().createMoveAnim(card
sMap[x1][y], cardsMap[x][y],x1, x, y, y);

    cardsMap[x][y].setNum(cardsMap[x][y].getNum()*2);
                            cardsMap[x1][y].setNum(0);

    MainActivity.getMainActivity().addScore(cardsMap[x][y].getNum());
                            merge = true;
                        }

                        break;
                    }
                }
            }
        }

        if (merge) {
            addRandomNum();
            checkComplete();
        }
    }
    private void swipeUp(){

        boolean merge = false;

        for (int x = 0; x < Config.LINES; x++) {
            for (int y = 0; y < Config.LINES; y++) {

                for (int y1 = y+1; y1 < Config.LINES; y1++) {
                    if (cardsMap[x][y1].getNum()>0) {

                        if (cardsMap[x][y].getNum()<=0)
{

    MainActivity.getMainActivity().getAnimLayer().createMoveAnim(card
sMap[x][y1],cardsMap[x][y], x, x, y1, y);

    cardsMap[x][y].setNum(cardsMap[x][y1].getNum());
                            cardsMap[x][y1].setNum(0);

                            y--;

                            merge = true;
                        }else if
(cardsMap[x][y].equals(cardsMap[x][y1])) {

    MainActivity.getMainActivity().getAnimLayer().createMoveAnim(card
sMap[x][y1],cardsMap[x][y], x, x, y1, y);
```

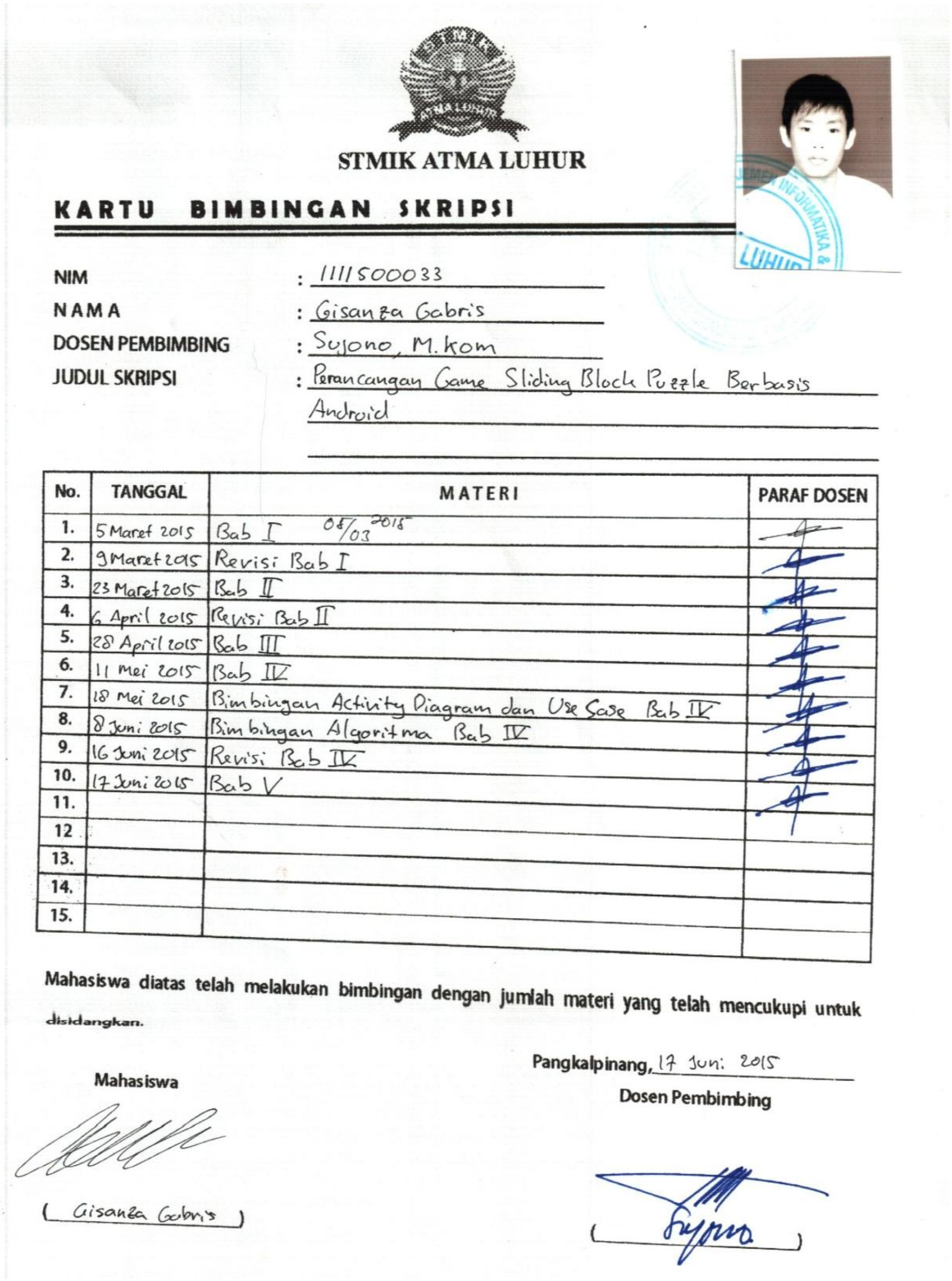
```
        cardsMap[x][y].setNum(cardsMap[x][y].getNum()*2);
                                                cardsMap[x][y1].setNum(0);

        MainActivity.getMainActivity().addScore(cardsMap[x][y].getNum());
                                                merge = true;
                                        }

                                        break;

                                }
                        }
                }
        }

                if (merge) {
                        addRandomNum();
                        checkComplete();
                }
        }
        private void swipeDown(){

                boolean merge = false;

                for (int x = 0; x < Config.LINES; x++) {
                        for (int y = Config.LINES-1; y >=0; y--) {

                                for (int y1 = y-1; y1 >=0; y1--) {
                                        if (cardsMap[x][y1].getNum()>0) {

                                                if (cardsMap[x][y].getNum()<=0)
{

        MainActivity.getMainActivity().getAnimLayer().createMoveAnim(card
sMap[x][y1],cardsMap[x][y], x, x, y1, y);

        cardsMap[x][y].setNum(cardsMap[x][y1].getNum());
                                                cardsMap[x][y1].setNum(0);

                                                y++;
                                                merge = true;
                                        }else if
(cardsMap[x][y].equals(cardsMap[x][y1])) {

        MainActivity.getMainActivity().getAnimLayer().createMoveAnim(card
sMap[x][y1],cardsMap[x][y], x, x, y1, y);

        cardsMap[x][y].setNum(cardsMap[x][y].getNum()*2);
                                                cardsMap[x][y1].setNum(0);

        MainActivity.getMainActivity().addScore(cardsMap[x][y].getNum());
                                                merge = true;
                                        }
                                        break;
                                    }
                                }
                            }
                }

                if (merge) {
                            addRandomNum();
                            checkComplete();
                }
        }

        private void checkComplete(){

                boolean complete = true;

                ALL:
                            for (int y = 0; y < Config.LINES; y++) {
                                for (int x = 0; x < Config.LINES; x++) {
                                    if (cardsMap[x][y].getNum()==0||

        (x>0&&cardsMap[x][y].equals(cardsMap[x-1][y]))||
                                (x<Config.LINES-
1&&cardsMap[x][y].equals(cardsMap[x+1][y]))||

        (y>0&&cardsMap[x][y].equals(cardsMap[x][y-1]))||
                                (y<Config.LINES-
1&&cardsMap[x][y].equals(cardsMap[x][y+1]))) {

                                        complete = false;
                                        break ALL;
                                    }
                                }
                            }

                if (complete) {
                            new
AlertDialog.Builder(getContext()).setTitle("Ky2048").setMessage("Game
Over").setPositiveButton("Start Again", new
DialogInterface.OnClickListener() {

                                @Override
                                public void onClick(DialogInterface dialog,
int which) {
                                    startGame();
                                }
                            }).show();
                }

        }

        private Card[][] cardsMap = new Card[Config.LINES][Config.LINES];
        private List<Point> emptyPoints = new ArrayList<Point>();
}
```

MainActivity.java

```
package com.kyindo.game;

import com.kyindo.game.R;

import android.app.Activity;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

public class MainActivity extends Activity {

        public MainActivity() {
                mainActivity = this;
        }

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);

                root = (LinearLayout) findViewById(R.id.container);
                root.setBackgroundColor(0xfffaf8ef);

                tvScore = (TextView) findViewById(R.id.tvScore);
                tvBestScore = (TextView) findViewById(R.id.tvBestScore);

                gameView = (GameView) findViewById(R.id.gameView);

                btnNewGame = (Button) findViewById(R.id.btnNewGame);
                btnNewGame.setOnClickListener(new View.OnClickListener()
{@Override public void onClick(View v) {
                        gameView.startGame();
                }});

                animLayer = (AnimLayer) findViewById(R.id.animLayer);
        }


        @Override
        public boolean onCreateOptionsMenu(Menu menu) {

                // Inflate the menu; this adds items to the action bar if
it is present.
                getMenuInflater().inflate(R.menu.main, menu);
                return true;
        }

        public void clearScore(){
                score = 0;
        showScore();
    }

    public void showScore(){
        tvScore.setText(score+"");
    }

    public void addScore(int s){
        score+=s;
        showScore();

        int maxScore = Math.max(score, getBestScore());
        saveBestScore(maxScore);
        showBestScore(maxScore);
    }

    public void saveBestScore(int s){
        Editor e = getPreferences(MODE_PRIVATE).edit();
        e.putInt(SP_KEY_BEST_SCORE, s);
        e.commit();
    }

    public int getBestScore(){
        return
getPreferences(MODE_PRIVATE).getInt(SP_KEY_BEST_SCORE, 0);
    }

    public void showBestScore(int s){
        tvBestScore.setText(s+"");
    }

    public AnimLayer getAnimLayer() {
        return animLayer;
    }

    private int score = 0;
    private TextView tvScore,tvBestScore;
    private LinearLayout root = null;
    private Button btnNewGame;
    private GameView gameView;
    private AnimLayer animLayer = null;

    private static MainActivity mainActivity = null;

    public static MainActivity getMainActivity() {
        return mainActivity;
    }

    public static final String SP_KEY_BEST_SCORE = "bestScore";

}
```

LAMPIRAN B

KARTU BIMBINGAN MATERI



STMIK ATMA LUHUR

KARTU BIMBINGAN SKRIPSI

JEMEN INFORMATIKA & LUHUR

NIM : 1111500033

NAMA : Gisanza Gabris

DOSEN PEMBIMBING : Sujono, M.kom

JUDUL SKRIPSI : Perancangan Game Sliding Block Puzzle Berbasis Android

No.	TANGGAL	MATERI	PARAF DOSEN
1.	5 Maret 2015	Bab I 08/03 2015	
2.	9 Maret 2015	Revisi Bab I	
3.	23 Maret 2015	Bab II	
4.	6 April 2015	Revisi Bab II	
5.	28 April 2015	Bab III	
6.	11 Mei 2015	Bab IV	
7.	18 Mei 2015	Bimbingan Activity Diagram dan Use Case Bab IV	
8.	8 Juni 2015	Bimbingan Algoritma Bab IV	
9.	16 Juni 2015	Revisi Bab IV	
10.	17 Juni 2015	Bab V	
11.			
12			
13.			
14.			
15.			

Mahasiswa diatas telah melakukan bimbingan dengan jumlah materi yang telah mencukupi untuk disidangkan.

Pangkalpinang, 17 Juni 2015

Mahasiswa

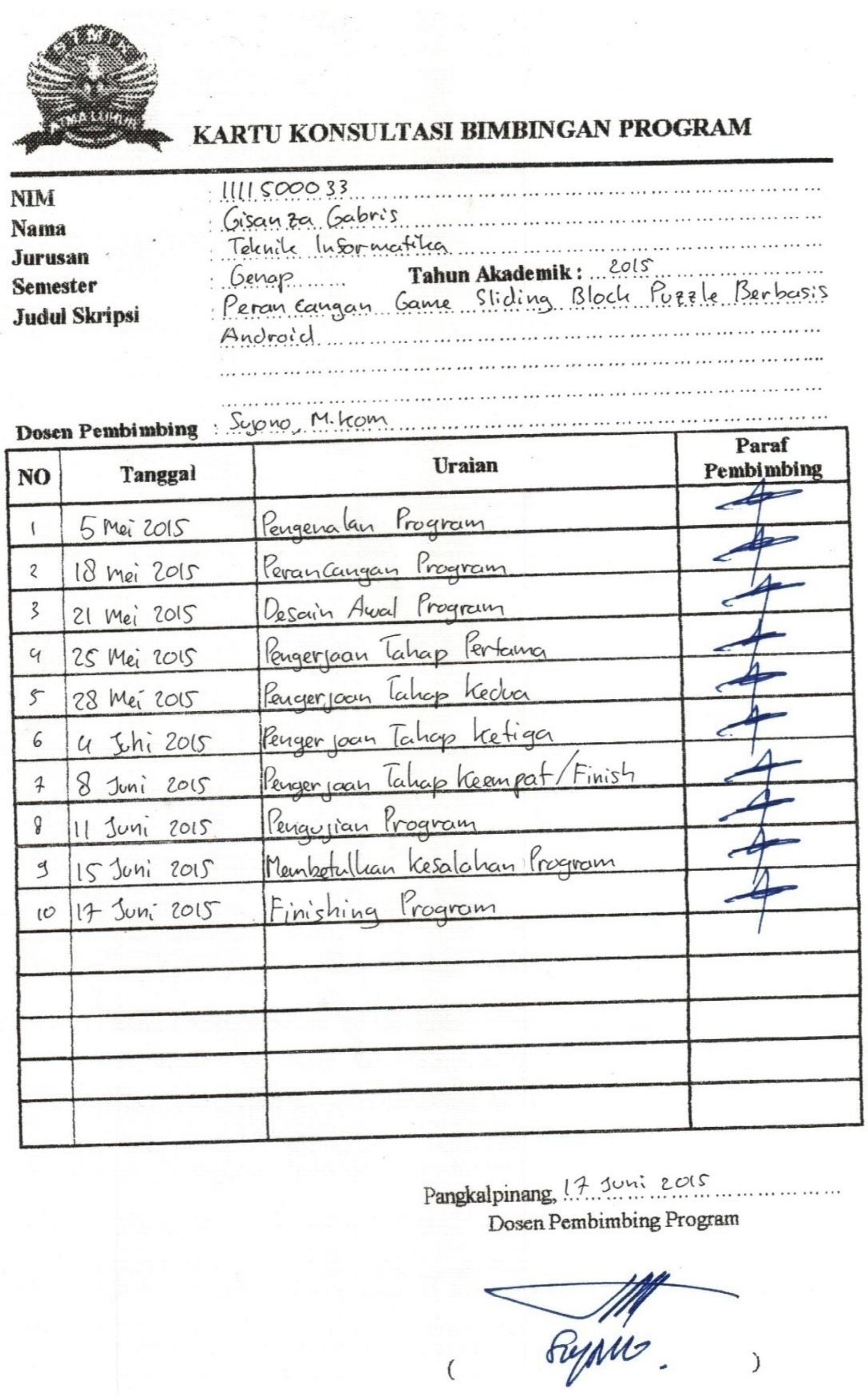
(Gisanza Gabris)

Dosen Pembimbing

(Sujono)

LAMPIRAN C

KARTU BIMBINGAN PROGRAM



KARTU KONSULTASI BIMBINGAN PROGRAM

NIM : 1111500033

Nama : Gisanza Gabris

Jurusan : Teknik Informatika

Semester : Genap Tahun Akademik : 2015

Judul Skripsi : Perancangan Game Sliding Block Puzzle Berbasis Android

Dosen Pembimbing : Sujono, M.Kom

NO	Tanggal	Uraian	Paraf Pembimbing
1	5 Mei 2015	Pengenalan Program	
2	18 Mei 2015	Perancangan Program	
3	21 Mei 2015	Desain Awal Program	
4	25 Mei 2015	Pengerjaan Tahap Pertama	
5	28 Mei 2015	Pengerjaan Tahap Kedua	
6	4 Juni 2015	Pengerjaan Tahap Ketiga	
7	8 Juni 2015	Pengerjaan Tahap Keempat/Finish	
8	11 Juni 2015	Pengujian Program	
9	15 Juni 2015	Membetulkan Kesalahan Program	
10	17 Juni 2015	Finishing Program	

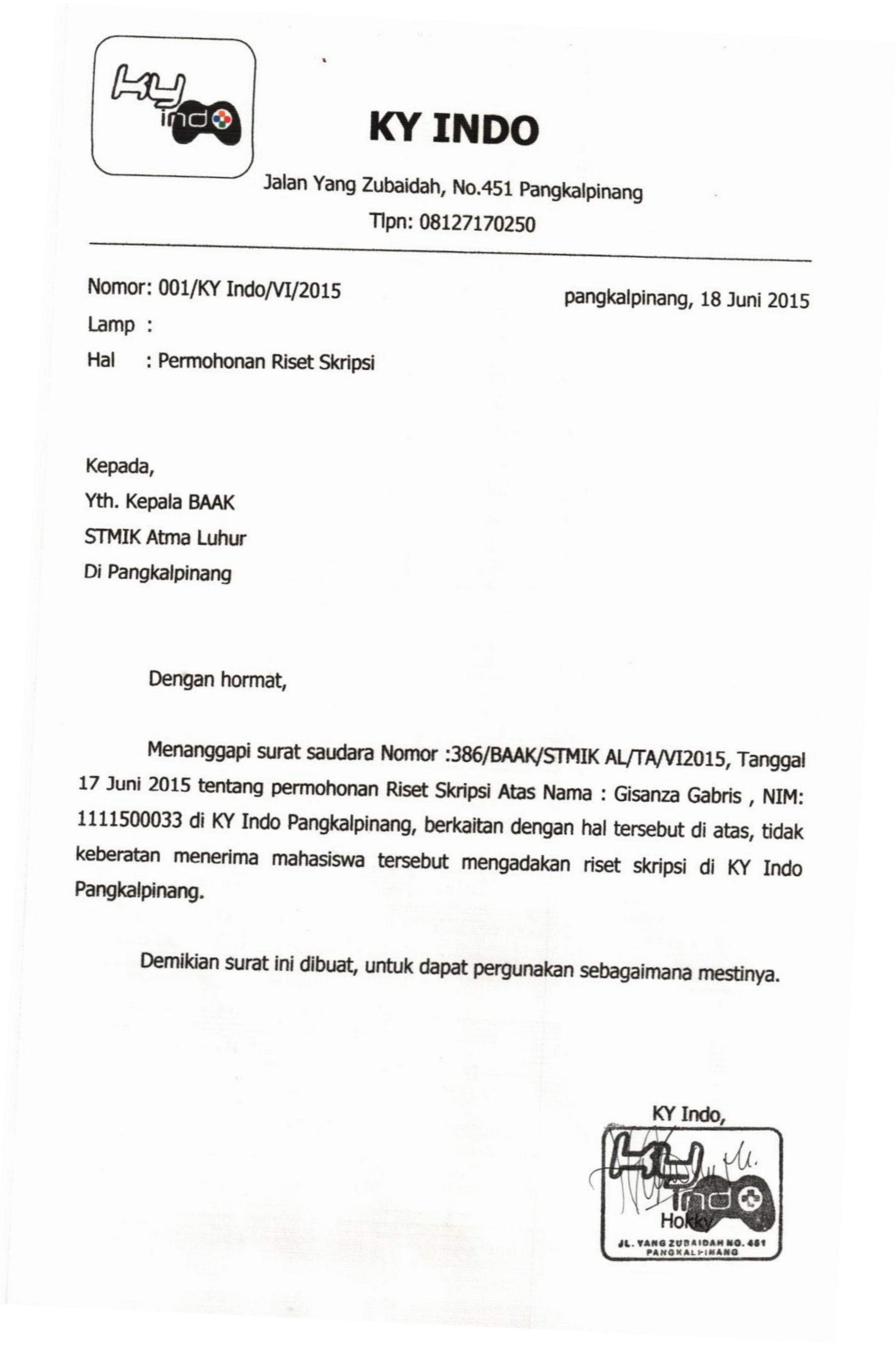
Pangkalpinang, 17 Juni 2015

Dosen Pembimbing Program

( )

LAMPIRAN D

SURAT PENGANTAR



KY INDO

Jalan Yang Zubaidah, No.451 Pangkalpinang
Tlpn: 08127170250

Nomor: 001/KY Indo/VI/2015 pangkalpinang, 18 Juni 2015
Lamp :
Hal : Permohonan Riset Skripsi

Kepada,
Yth. Kepala BAAK
STMIK Atma Luhur
Di Pangkalpinang

Dengan hormat,

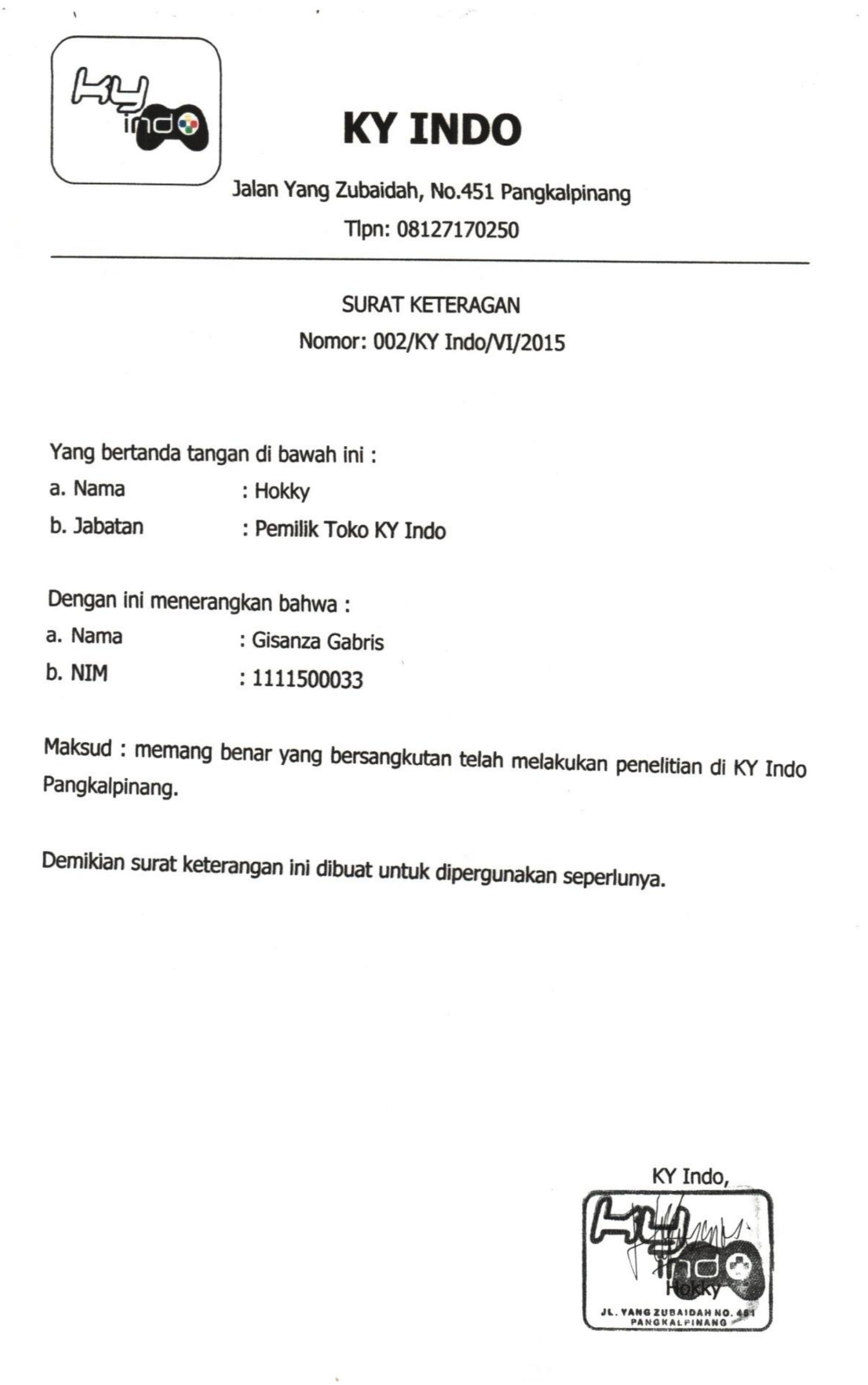
Menanggapi surat saudara Nomor :386/BAAK/STMIK AL/TA/VI2015, Tanggal 17 Juni 2015 tentang permohonan Riset Skripsi Atas Nama : Gisanza Gabris , NIM: 1111500033 di KY Indo Pangkalpinang, berkaitan dengan hal tersebut di atas, tidak keberatan menerima mahasiswa tersebut mengadakan riset skripsi di KY Indo Pangkalpinang.

Demikian surat ini dibuat, untuk dapat pergunakan sebagaimana mestinya.

KY Indo,

JL. YANG ZUBAIDAH NO. 451
PANGKALPINANG

Hokky



KY INDO

Jalan Yang Zubaidah, No.451 Pangkalpinang
Tlpn: 08127170250

SURAT KETERAGAN
Nomor: 002/KY Indo/VI/2015

Yang bertanda tangan di bawah ini :

a. Nama : Hokky
b. Jabatan : Pemilik Toko KY Indo

Dengan ini menerangkan bahwa :

a. Nama : Gisanza Gabris
b. NIM : 1111500033

Maksud : memang benar yang bersangkutan telah melakukan penelitian di KY Indo Pangkalpinang.

Demikian surat keterangan ini dibuat untuk dipergunakan seperlunya.

KY Indo,

JL. YANG ZUBAIDAH NO. 451
PANGKALPINANG

Hokky